



# AIB-2941M 使用手冊

Version 1.0.0, Sep. 2024

本手冊的技術服務與操作資訊適用於



授權者: Weikai Wang

編輯者: Anna Huang

## 保固說明

---

泓格科技股份有限公司 (ICP DAS) 所生產的產品，均保證原始購買者對於有瑕疵之材料，於交貨日起保有為期一年的保固。

## 免責聲明

---

泓格科技股份有限公司對於因為應用本產品所造成的損害並不負任何法律上的責任。本公司保留有任何時間未經通知即可變更與修改本文件內容之權利。本文所含資訊如有變更，恕不予另行通知。本公司盡可能地提供正確與可靠的資訊，但不保證此資訊的使用或其他團體在違反專利或權利下使用。此處包涵的技術或編輯錯誤、遺漏，概不負其法律責任。

## 版權所有

---

2024 © 泓格科技股份有限公司版權所有。

## 商標識別

---

本文件提到的所有公司商標、商標名稱及產品名稱分別屬於該商標或名稱的擁有者所有。

### 技術服務

如有任何問題，請與本公司客服聯絡，我們將盡速為您服務。

E-mail 信箱：[service@icpdas.com](mailto:service@icpdas.com)

# 目錄

目錄.....	3
1. 產品簡介.....	5
1.1. 產品特色.....	6
1.2. 產品規格.....	11
1.3. 產品概述.....	14
1.4. 產品尺寸.....	20
2. 快速上手.....	21
2.1. AIB-2941M 安裝.....	22
2.2. 擴充配件安裝.....	24
2.2.1. 安裝 AI 模組/SSD.....	24
安裝 XV-Board/XB-Board.....	26
2.2.2.....	26
2.2.3. 插入 SIM 卡.....	29
2.2.4. 安裝天線.....	30
2.3. 管理者登入.....	31
2.4. 使用者介面語言變更.....	32
2.5. IP 設定.....	34
3. 工具與任務.....	36
3.1. 系統工具.....	36
3.1.1. 使用者介面語言變更.....	37
3.1.2. IP 設定.....	39
3.1.3. 螢幕解析度設定.....	41
3.1.4. 使用者帳戶管理.....	44
3.2. 軟體工具.....	48
3.2.1. Utility – 檢視系統資訊.....	48
3.2.2. Text Edit 文字編輯器.....	51
3.2.3. 效能監控管理.....	52
3.2.4. 網路芬鄰設定.....	53
3.2.5. 自動網路校時.....	56
3.2.6. CuteCOM.....	58
3.3. 其他支援工具.....	59
3.3.1. 遠端連線軟體.....	59

3.3.2.	SSH 檔案傳輸 .....	62
4.	AIB-2941M 程式開發 .....	64
4.1.	開發工具準備 .....	65
4.2.	用 Python 建立第一個 AIB-2941M 應用程式 .....	67
4.2.1.	建立一個新專案 .....	68
4.2.2.	在視窗上加入控制選項 .....	72
4.2.3.	在控制項目中加入事件 .....	74
4.2.4.	執行編譯好的應用程式 .....	76
4.2.5.	Python 範例 .....	77
5.	I/O 擴充模組與 SDK 的選擇 .....	78
6.	AIB-2941M 系統更新與升級 .....	81
附錄 A.	Tips –實用技巧 .....	88
A.1.	如何設定 AIB-2941M 自動網路校時 .....	88
A.2.	如何更換 RTC 電池 .....	90
附錄 B.	Hailo-8L 開發及應用 .....	91
B.1.	環境設置 .....	93
B.1.1.	AIB-2941M runtime 環境設置: HailoRT 及 Hailo PCIe driver 安裝 .....	93
B.1.2.	PC 環境設置: CUDA 及深度學習框架安裝 .....	95
B.2.	AI 學習模型訓練和效能評估 .....	99
B.2.1.	模型訓練 .....	99
B.2.2.	模型效能評估 .....	101
B.3.	模型轉換 .....	103
B.3.1.	模型轉換工具安裝及模型供 Hailo-8L 使用 .....	103
B.4.	使用及開發 AI 模型程式 .....	106
附錄 C.	XV-Board 和 XB-Board 系列 I/O 擴充卡 .....	107
附錄 D.	手冊修訂記錄 .....	109

# 1. 產品簡介

本章提供了 AIB-2941M 系列模組和其配備組件的概覽，並介紹了基本概念，協助使用者熟悉 AIB-2941M 模組。



AIB-2941M 是一無風扇、安全、緊湊、低功耗的邊緣 AI 控制器，具有基於 ARM 的 CortexR-A72 處理器。提供 M.2 插槽擴充不同的 AI 加速器模組，在邊緣實現可擴展的高效能低功耗人工智慧功能，擴展 AI 計算能力，將 AI 和邊緣計算引入工業應用。可在 I/O 擴充方面，有高保護性的 XV-Board/XB-Board 可供選購搭配使用。另外提供一個 microSD 插槽及 M.2 Key M 插槽 (擴充 NVEM SSD 卡使用)，可儲存常用的應用程式、圖檔及資料，以及方便攜帶與擴充的儲存工具。

AIB-2941M 邊緣 AI 控制器可配備 GbE LAN、HDMI、USB 3.1、DIO 和 4G/Lora/NB-IoT 無線連接等週邊設備。堅固的金屬外殼和散熱使其適用於 -20 至 +70 °C 環境。

## 1.1. 產品特色

AIB-2941M 具有全面性的系統配置功能及便利性的遠端系統更新服務，可提供多種用途，滿足使用者的各種需求。以下列出 AIB-2941M 的主要特色。

### Ubuntu

Ubuntu 它是一種易於使用的操作系統，非常適合初學者和經驗豐富的用戶。它具有直觀的界面和豐富的功能，多種用途，是目前最多使用者的 Linux 版本。應用於 AIB-2941M，可以充分利用其功能和性能。(Ubuntu 為出廠預設安裝 OS)



- 良好的使用者經驗
- 易用性、多功能性、安全性
- 豐富的社群支援
- 嵌入式服務：Web Server, FTP Server, Telnet Server, SSH Server
- 豐富的軟體支援及開發工具：專屬的 SDK, Perl, Java, GUI 軟體...等

### Raspberry Pi OS



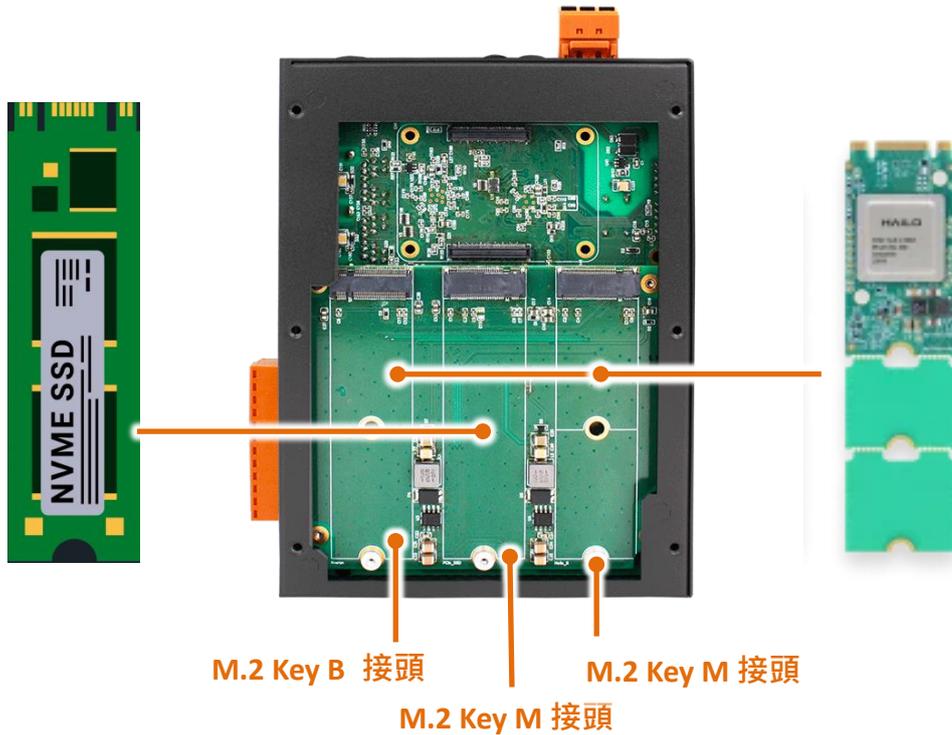
EdgeAI 控制器內建樹莓派 64 位元操作系統。Raspberry Pi OS 基於 Debian Linux，採用文本行命令方式操作，適合熟悉 Linux 操作的用戶。同樣適合熟悉 Windows 的用戶，Raspberry Pi OS 運行 Windows 用戶熟悉的簡化桌面界面，顯示桌布、任務欄、快捷方式和應用程序菜單等。

- 豐富的樹莓派軟體生態系統
- 超過 35,000 個軟體套件易於安裝
- 預裝了多個應用程序——Python3/Python IDE/等
- 可於網路上取得許多現成的應用程序

## 人工智慧模組

- M.2 Key M 接頭，用於擴充 AI 加速模組
- M.2 Key B 接頭，用於擴充 AI 加速模組
- M.2 Key M 接頭，用於擴充 SSD 模組

提供 M.2 Key M/ Key B 接頭，彈性擴充不同人工智慧模組



## 硬體序號鎖保護軟體

AIB-2941M 提供一 64 位元硬體鎖，提供防軟體盜拷保護技術。



## 多種磁碟儲存介面，擴充儲存空間

AIB-2000 提供多種可靠的儲存裝置，如、microSD，使用者可依據需求，選擇儲存裝置。

- 一個 microSD 插槽 (出貨不附 microSD 卡)
- 一個 M.2 Key M 插槽(擴充 M.2 NVEM SSD 卡使用) (出貨不附 SSD 卡)



因 SSD 有眾多品牌，以下兩款是我們測試過可以正常運作的工規 SSD

### Innodisk M.2 (P80) 3TE4

<https://www.innodisk.com/tw/products/flash-storage/m2-pcie/m2-p80-3te4-b-m-key>

### APacer PV220-M280

[https://www.apacer.com/en/product/industrial-product/detail/industrial\\_ssd/pv220-m280](https://www.apacer.com/en/product/industrial-product/detail/industrial_ssd/pv220-m280)

其他品牌和型號的 SSD 可能會有各式各樣的問題，不建議使用

## 支援多種 I/O 擴充卡



AIB-2000 系列配備有一個 I/O 擴充匯流排，可供用戶選購一片 XV-board 或 XB-board 擴充卡。它可用來發展各種 I/O 功能，例如：DI、DO、A/D、D/A、Timer/Counter 與各種通訊介面，例如：RS-232/422/485... 等。

更多有關於 XV-board I/O 模組，請參閱

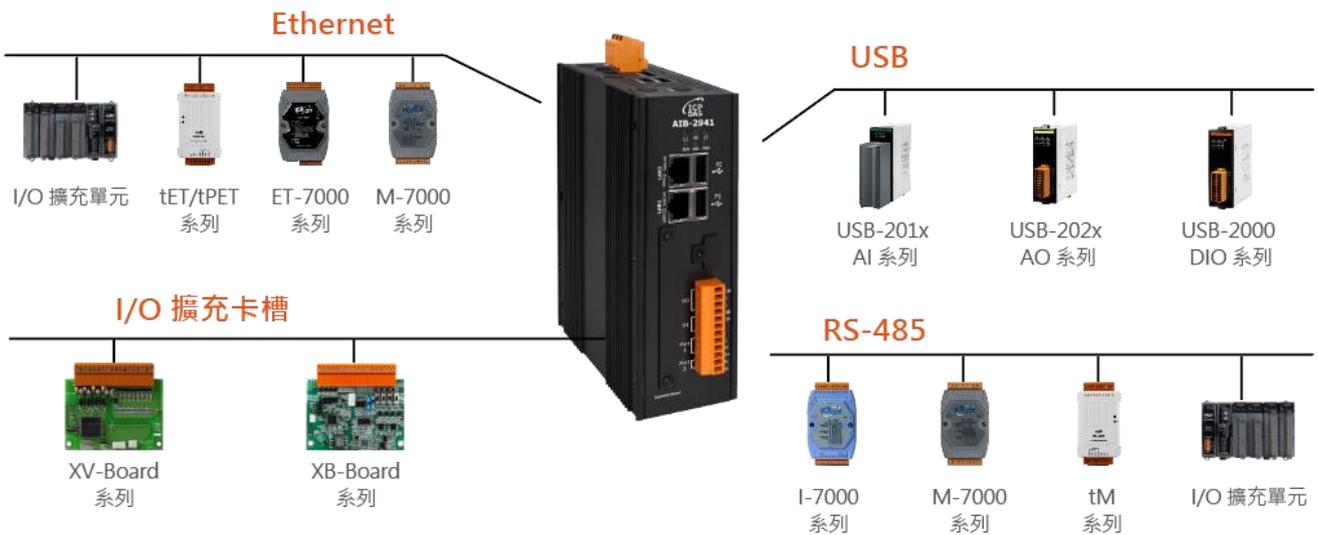
[https://www.icpdas.com/tw/product/guide+Remote I O Module and Unit+PAC I O Modules+XV-board](https://www.icpdas.com/tw/product/guide+Remote+I+O+Module+and+Unit+PAC+I+O+Modules+XV-board)

更多有關於 XB-board I/O 模組，請參閱

[https://www.icpdas.com/tw/product/guide+Remote I O Module and Unit+PAC I O Modules+XB-board](https://www.icpdas.com/tw/product/guide+Remote+I+O+Module+and+Unit+PAC+I+O+Modules+XB-board)

## 豐富的 I/O 通訊介面

AIB-2000 配備多個 RS-485 序列通訊埠、兩個乙太網路及 USB 連接埠，可擴充 I/O 及連接外部設備。



更多有關於 Ethernet 介面 I/O 模組，請參閱

<https://www.icpdas.com/en/product/p02.php?root=537&kind=539>

更多有關於 USB 介面 I/O 模組，請參閱

<https://www.icpdas.com/en/product/p02.php?root=537&kind=541>

更多有關於 RS-485 介面 I/O 模組，請參閱

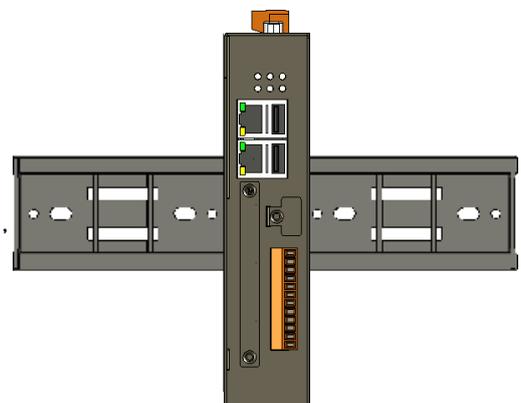
<https://www.icpdas.com/en/product/p02.php?root=537&kind=538>

## 金屬外殼的設計選擇

AIB-2000 配備金屬材質外殼。

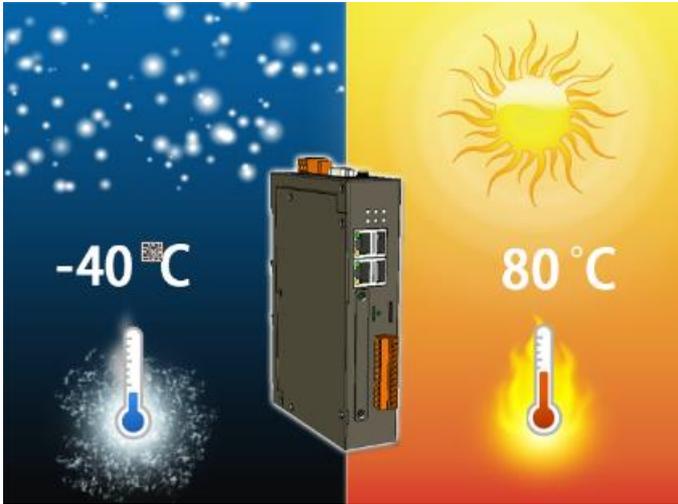
## 輕巧設計，安裝簡易

AIB-2000 的機體精實輕巧，可搭配導軌 (DIN-Rail) 使用，架設於狹窄空間。



## 運作於嚴苛的環境之中，仍保有高可靠度及穩定性

AIB-2000 擁有在極端溫度和濕度下運作的能力。



- 操作環境溫度：-20 °C 至 +70 °C
- 儲存環境溫度：-40 °C 至 +80 °C
- 操作環境相對濕度：10 % 至 90 % RH，無冷凝

## 1.2. 產品規格

下表為 AIB-2941M 的規格表。

系統軟體	
OS	Ubuntu 作業系統(預設) /標準 Raspberry PI 4 64-bit OS
SDK 提供	C / python 函式庫
多語系支援	英文、簡體中文、繁體中文
中央處理器模組 中央處理器模組、繁體中文 I 4 64-b	
CPU	Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
GPU	VideoCoreVI (OpenGL ES 3.0), H.265 (up to 4Kp60fps decode), H.264 (up to 1080p60fps, 1080p30 encode)
SDRAM	4GB LPDDR4
儲存	內建 32GB eMMC
其它週邊	
其它儲存	1 個 M.2 Key M 接頭(PCIe 2.0 介面) · 支援安裝 PCIe SSD 2230/2242/2260/2280 (沒附 M.2 SSD 卡)
FRAM	16 KB
即時時鐘(RTC)	提供秒、分、時、日、星期、月、年資訊
看門狗	有
LED 指示燈	1 個電源 LED 指示燈、1 個系統運作 LED 指示燈和 3 個可程式控制 LED 指示燈
AI Accelerator 模組	
M.2connector Key M	1 個 M.2 Key M 接頭 x (PCIe 2.0 介面) 支援安裝 Hailo-8L M.2 AI 加速模組 2230/2242/2260/2280 (選配) Frameworks: TensorFlow/Lite, ONNX, Keras, & Pytorch
M.2connector Key B	1 個 M.2 Key B 接頭 x (USB3.0 介面) 支援安裝 Kneron M.2 AI 加速模組 2230/2242/2260/2280(選配) Frameworks: TensorFlow/Lite, Caffe, Keras, & Pytorch

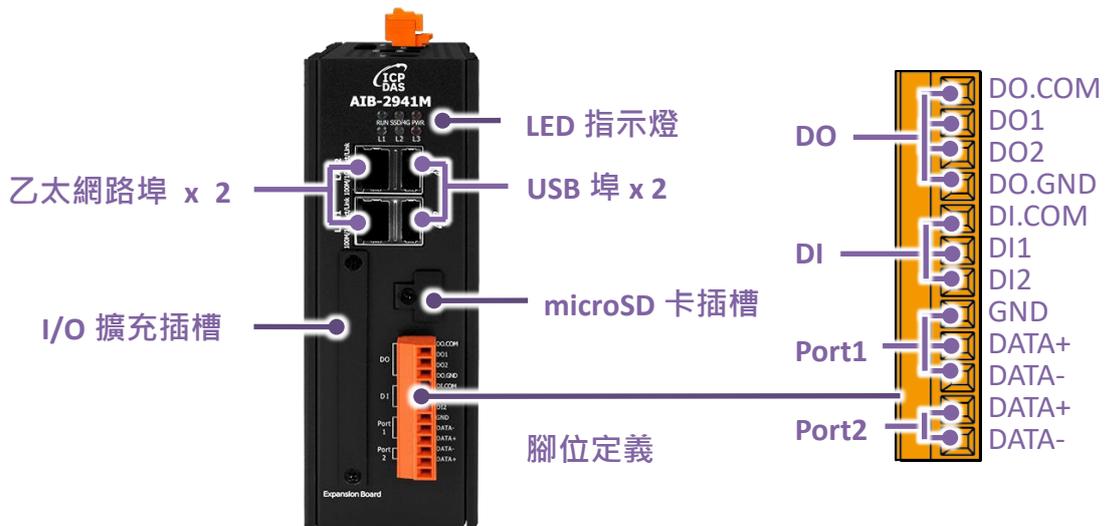
顯示及通訊介面	
顯示	HDMI 埠 x 1 (4Kp 60fps 帶音頻)
乙太網路	RJ-45 x 2, 10/100/1000 Based-TX (Auto-negotiating LED 指示燈和自動 MDI/MDI-X LED 指示燈)
USB 2.0 (host)	2
USB 3.0 (host)	2
串列埠 1	RS-485 (Data+, Data-); 2500 VDC 隔離
串列埠 2	RS-485 (Data+, Data-); 2500 VDC 隔離
無線通訊	1 個擴充接頭，安裝 4G/LoRa/NB-IoT 模組(USB2.0 介面)(選配) 及 Nano SIM 插槽 x 1 (選配)
數位輸入	
通道數	2
類型	濕接點
Sink/Source (NPN/PNP)	Sink/Source
ON 電壓準位	濕: +19 VDC ~ +24 VDC
OFF 電壓準位	濕: +11 VDC 最大.
數位輸出	
通道數	2
類型	隔離開路集極
負載電壓	+24 VDC
負載電流	100 mA/通道
過電壓保護	60 VDC
過載保護	1.1 A
短路保護	是
I/O 擴充	
I/O 類型	XV-board/XB-board (選配)
電源	
輸入範圍	+12 ~ +48 VDC
功耗	4.8 W (0.2 A @ 24 VDC)

機構	
外殼	金屬
尺寸 (mm)	52 mm x 177 mm x 128 mm
安裝	標準導軌安裝 (DIN-Rail)/ 壁掛式安裝
環境	
工作溫度	-20 ~ +70 °C
儲存溫度	-40 ~ +80 °C
濕度	10 ~ 90 % RH, 無凝露

### 1.3. 產品概述

AIB-2941M 具備多種傳輸元件，可透過乙太網路、USB、RS-232/RS-485 等介面來連接外部裝置做整合。以下對 AIB-2941M 外部組成元件及其傳輸元件做概覽及簡要說明。

#### 前視圖



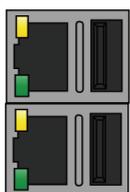
#### LED 指示燈



AIB-2941M 系列模組具有 6 個 LED 指示燈：

LED 指示燈	顏色	訊息說明
RUN	綠	系統運作中
PWR	紅	電源供電正常
4G/SSD	橘	4G 通訊/SSD 儲存碟存取
L1	綠	使用者程式控制
L2	橘	
L3	紅	

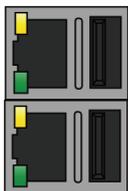
## 乙太網路埠



AIB-2941M 支援 10/100/1000 Based-TX (Auto-negotiating LED 指示燈和自動 MDI/MDI-X LED 指示燈 )網路，由連接埠的燈號可以得知 AIB-2941M 目前使用的網路速度。

LED 指示燈	狀態 (顏色)	說明
1G	恆亮 (橘色)	1 G 速度連線
	恆滅	10/100 M 速度連線
Lin/Act	恆亮 (綠色)	已建立網路通訊連線
	恆滅	未建立網路通訊連線
	閃爍 (綠色)	資料封包傳輸中

## USB 埠



AIB-2941M 提供兩個 USB 2.0 連接埠來連接 USB 設備，如滑鼠、鍵盤或其他 USB 裝置。

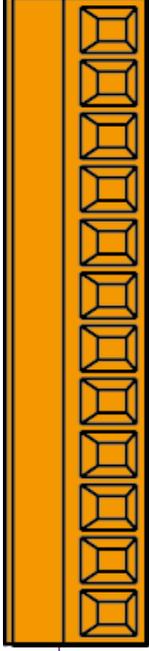
## microSD 卡插槽

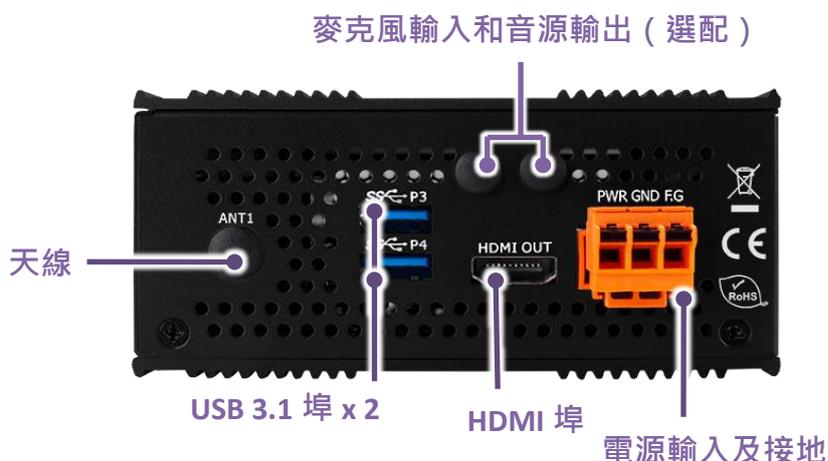
microSD 插槽的配置，用來提供使用者使用 microSD 記憶卡做系統回復及儲存空間擴充用。microSD 記憶卡擴充可支援至 32 GB。

## XV-Board/XB-Board 插槽

安裝 XV-board / XB-Board I/O 擴充模組後，連線端子的位置。  
安裝流程請參考第 25~27 頁，章節 2.2.2.安裝 XV-Board/XB-Board

## 腳位配置

Pin 腳位	標示	說明	
	1	DO.PWR	Digital Output x 2
	2	DO0	
	3	DO1	
	4	DO.GND	
	5	DI.COM	Digital Input x 2
	6	DI0	
	7	DI1	
	8	RS485.GND	串列埠 2 (RS485)
	9	D2-	
	10	D2+	
	11	D1-	串列埠 1 (RS485)
	12	D1+	



### HDMI 埠

HDMI 輸出埠 x 1 (4Kp 60fps 帶音頻)

解析度格式包括：1920x1080、1680x1050、1600x1200、1440x900、1440x576、1440x480、1400x1050、1280x1024、1280x960、1280x800、1280x720、1152x864、1024x768、832x624 和 800x600。

### USB 3.1 埠

AIB-2941M 提供兩個 USB 3.1 連接埠用來連接 USB 高速設備裝置。

### 天線

一天線孔，搭配 4G 模組使用 (選配)

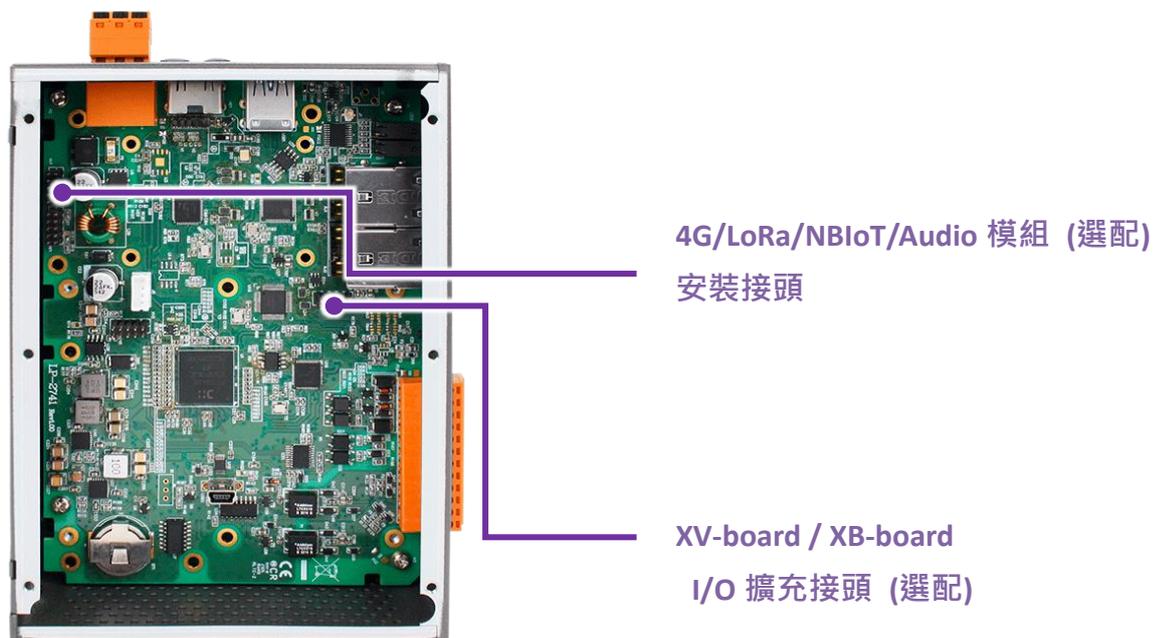
### 麥克風輸入和音源輸出 (選配)

### 電源輸入及接地

3 Pin 的連接器。2 Pin 為電源輸入，1 Pin 為接地(F.G)。



## 左視圖



### 4G/LoRa/NB-IoT/Audio 模組接頭(選配)

安裝 4G/LoRa/NB-IoT 模組(USB2.0 介面) 及 Nano SIM 插槽 x 1 (選配)

### XV-board / XB-board I/O 擴充接頭

I/O 擴充槽→ 安裝方式請參考 “2.2.2 安裝 XV-Board/XB-Board”



### M.2 Key B 接頭

1 個 M.2 Key B 接頭 (USB3.0 介面) · 安裝 M.2 Key B AI 加速模組 2230/2242/2260/2280

### M.2 Key M 接頭

2 個 M.2 Key M 接頭 (PCIe x1 2.0 介面) · 安裝 M.2 Key M AI 加速模組或 M.2 NVEM SSD 2230/2242/2260/2280 (沒附 M.2 SSD 卡)

#### SSD 的建議使用

因 SSD 有眾多品牌，以下兩款是我們測試過可以正常運作的工規 SSD

#### Innodisk M.2 (P80) 3TE4

<https://www.innodisk.com/tw/products/flash-storage/m2-pcie/m2-p80-3te4-b-m-key>

#### APacer PV220-M280

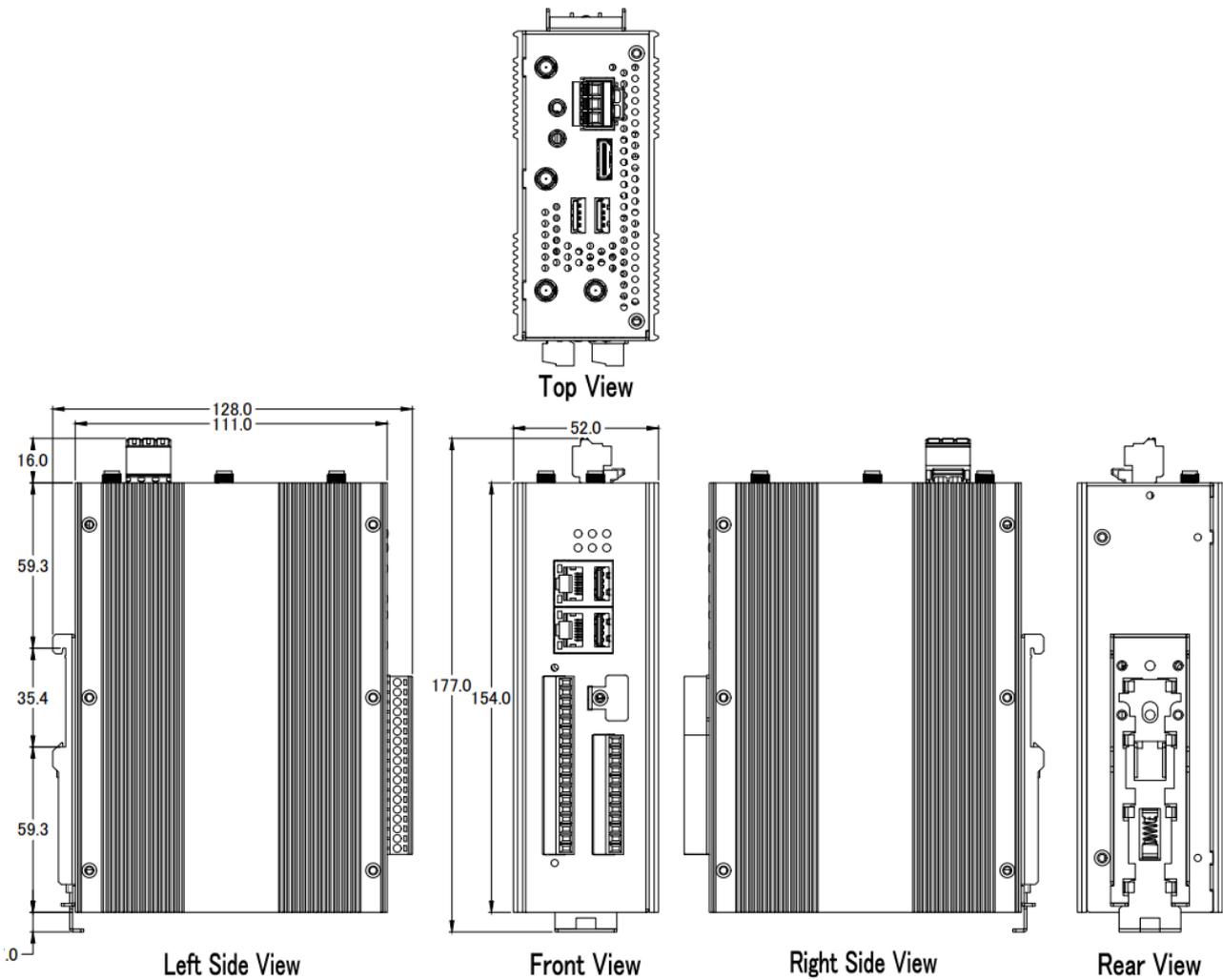
[https://www.apacer.com/en/product/industrial-product/detail/industrial\\_ssd/pv220-m280](https://www.apacer.com/en/product/industrial-product/detail/industrial_ssd/pv220-m280)

其他品牌和型號的 SSD 可能會有各式各樣的問題，不建議使用

## 1.4. 產品尺寸

下圖為 AIB-2941M 的尺寸規格，安裝至機箱內時，請預留足夠的散熱空間。

所有尺寸均以 mm 為單位。



## 2. 快速上手

本章說明 AIB-2941M 安裝與設定作業，將一步一步帶領首次使用 AIB-2941M 的使用者，瞭解安裝、設定及執行 AIB-2941M。

在開始使用 AIB-2941M 之前，請先檢查產品內容是否包括以下項目。如有短缺或因運送造成貨品的損毀，請儘快向您所購買的經銷商聯絡。



AIB-2941M



44 mm 導軌固定架



螺絲起子  
(1C016) 2.4 mm



M3x6L 螺絲\* 4



歐式壓著端子\* 16

### 選配行動網路擴充模組：天線安裝套件



GSM/GPRS 天線  
(ANT-421-04)

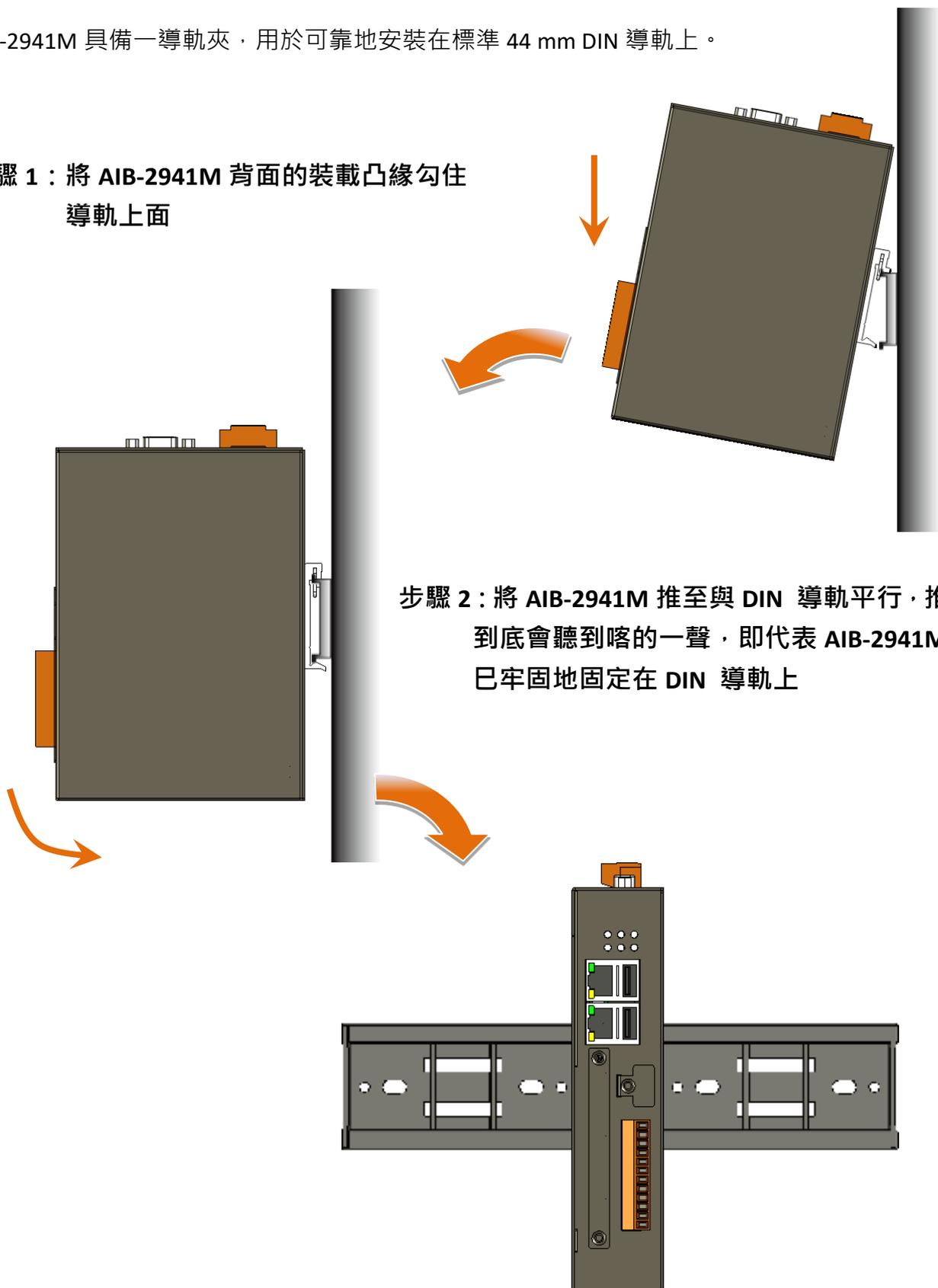


GPS Active 天線  
(ANT-115-03)

## 2.1. AIB-2941M 安裝

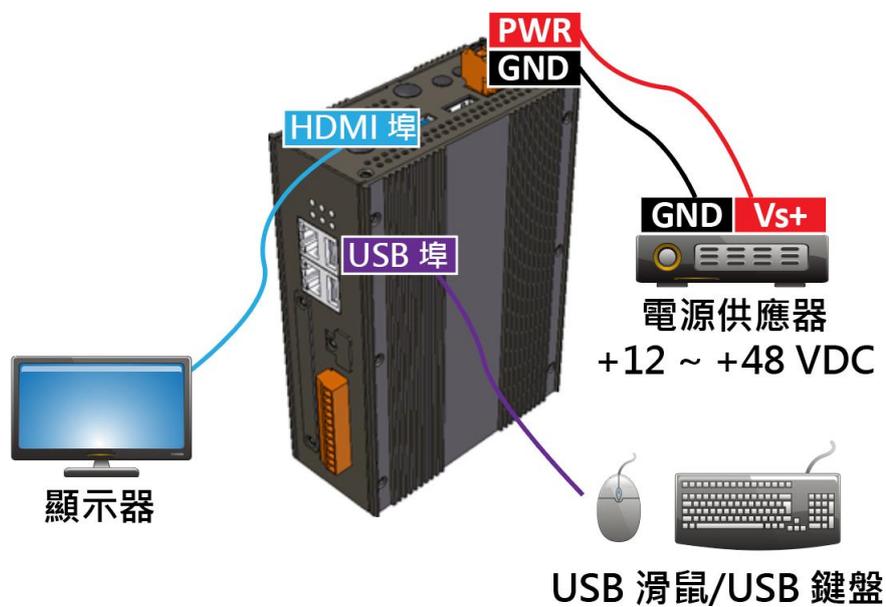
AIB-2941M 具備一導軌夾，用於可靠地安裝在標準 44 mm DIN 導軌上。

**步驟 1：** 將 AIB-2941M 背面的裝載凸緣勾住導軌上面



#### 步驟 4：設置電源及連接 PC 及 USB 裝置

- i. 連接 USB 滑鼠或 USB 鍵盤至 USB 埠
- ii. 連接顯示器至 HDMI 埠
- iii. 連接+12 至+48 VDC 的電源供應器至接頭的 PWR 端及 GND 端



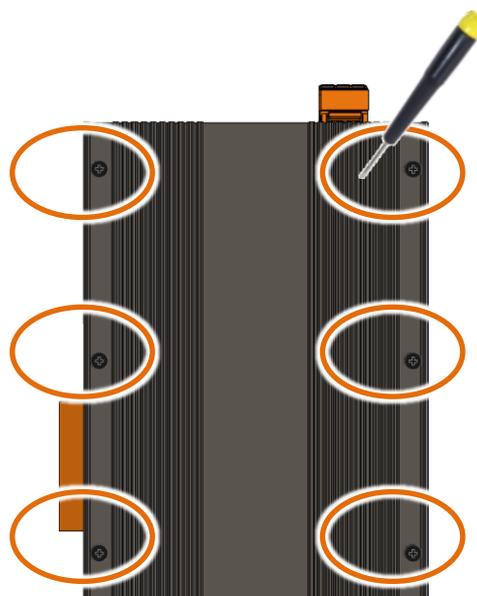
## 2.2. 擴充配件安裝

### 2.2.1. 安裝 AI 模組/SSD

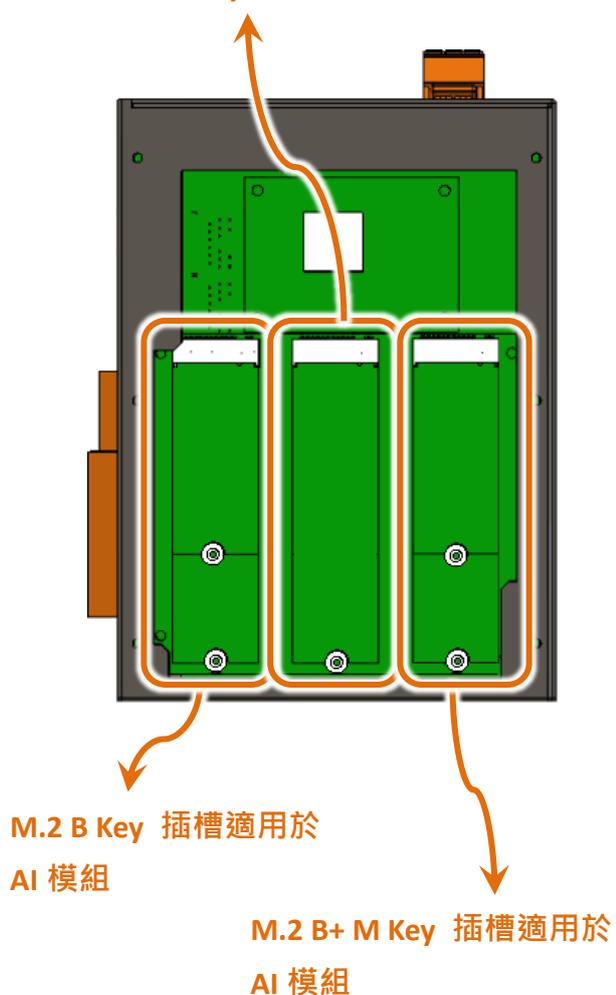
步驟 1：使用螺絲起子，將右側蓋上的螺絲卸下

步驟 2：移除右側蓋

步驟 3：確認 AI 模組和 SSD 可用的 M.2 插槽



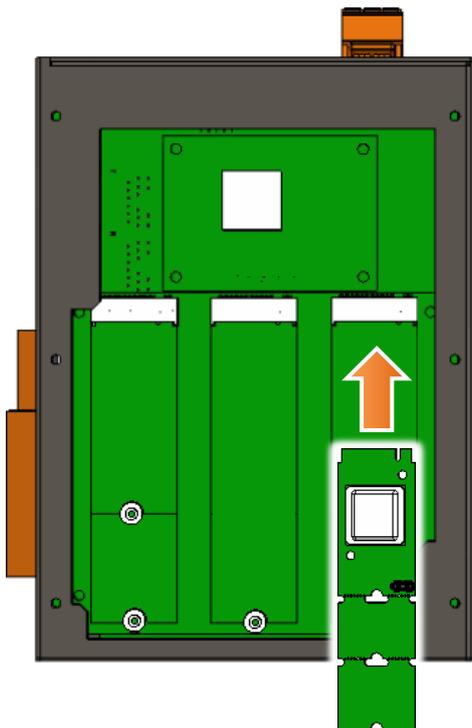
M.2 B+ M Key 插槽適用於 SSD



M.2 B Key 插槽適用於  
AI 模組

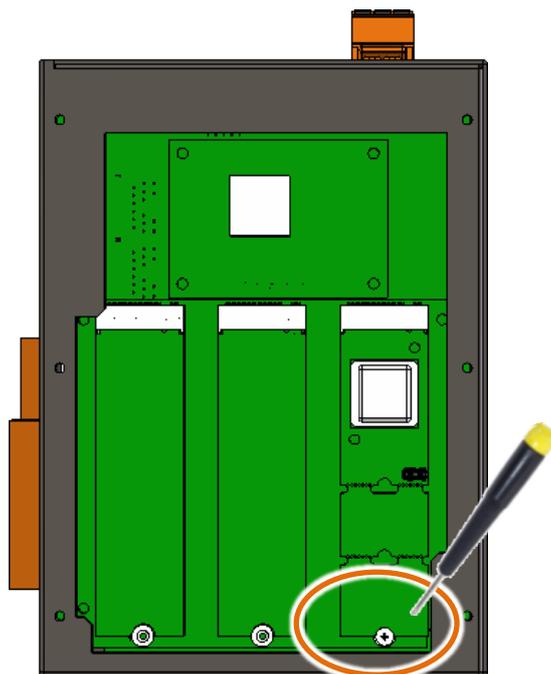
M.2 B+ M Key 插槽適用於  
AI 模組

步驟 4：插入 AI 模組/SSD



步驟 5：使用隨附的 M2 x 3L 螺絲固定 AI 模組/SSD

步驟 6：合上右側蓋，然後旋緊螺絲



## 2.2.2. 安裝 XV-Board/XB-Board

AIB-2941M 具有一 I/O 擴充插槽，可支援一 XV-Board/XB-Board 擴充模組。有關於 XV-Board/XB-Board 的規格資訊，請參閱

### XV-board I/O 模組 (選配)

<https://www.icpdas.com/tw/product/guide+Remote I O Module and Unit+PAC I O Modules+XV-board>

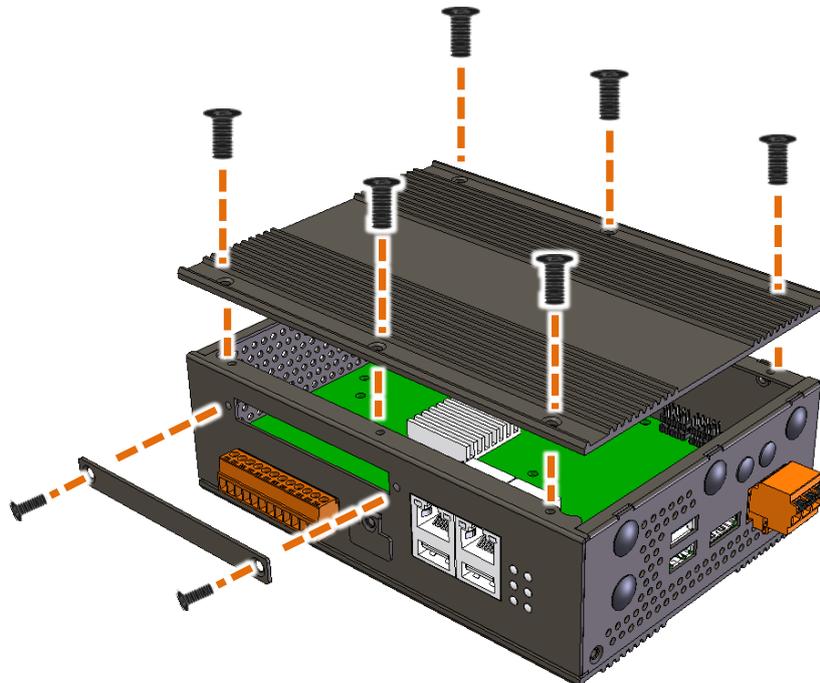
### XB-board I/O 模組(選配)

<https://www.icpdas.com/tw/product/guide+Remote I O Module and Unit+PAC I O Modules+XB-board>

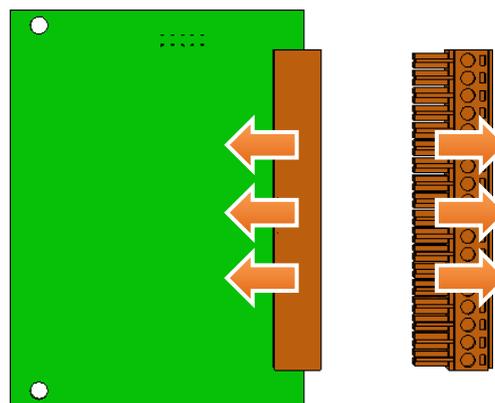
步驟 1：將左側蓋上的螺絲卸下

步驟 2：移除左側蓋

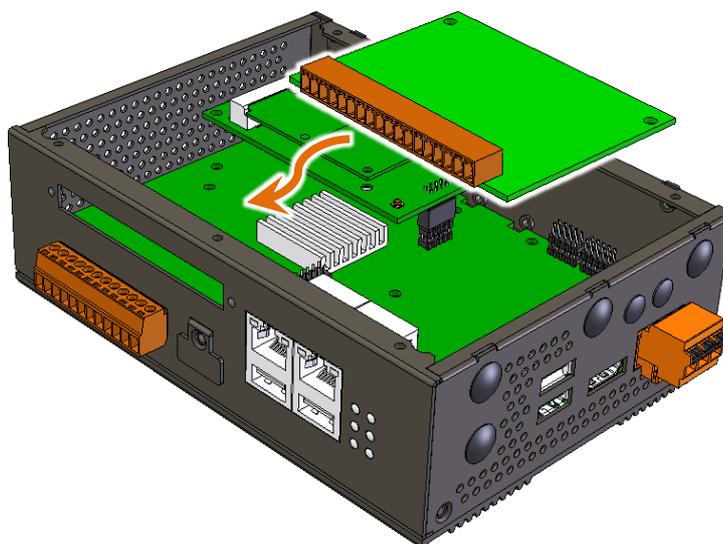
步驟 3：將前蓋上 I/O 擴充槽上的螺絲卸下



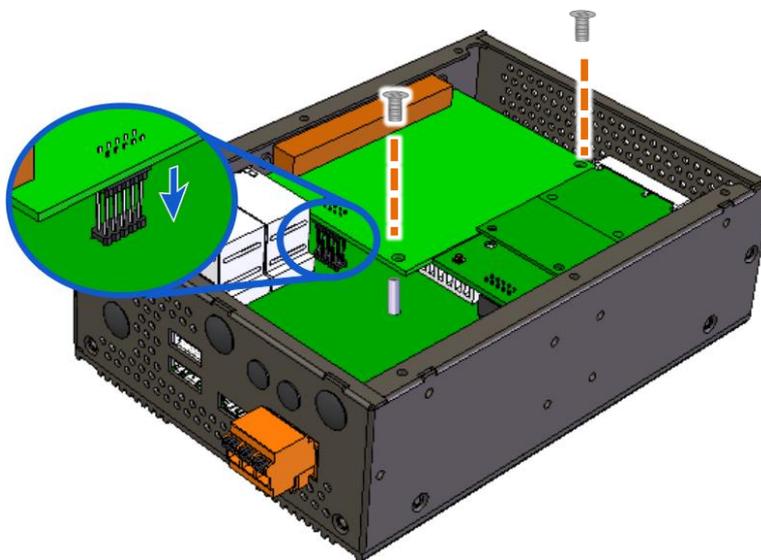
步驟 4：將 XV-board/XB-board 與 I/O 端子分開



步驟 5：垂直握住 XV-board/XB-board 並對準 I/O 插槽，然後小心地將 XV-board/XB-board 壓到 I/O 擴充插槽上



步驟 6：小心地將 XV-board/XB-board 連接至 I/O 擴充接頭上，並使用隨附的螺絲固定 XV-board/XB-board



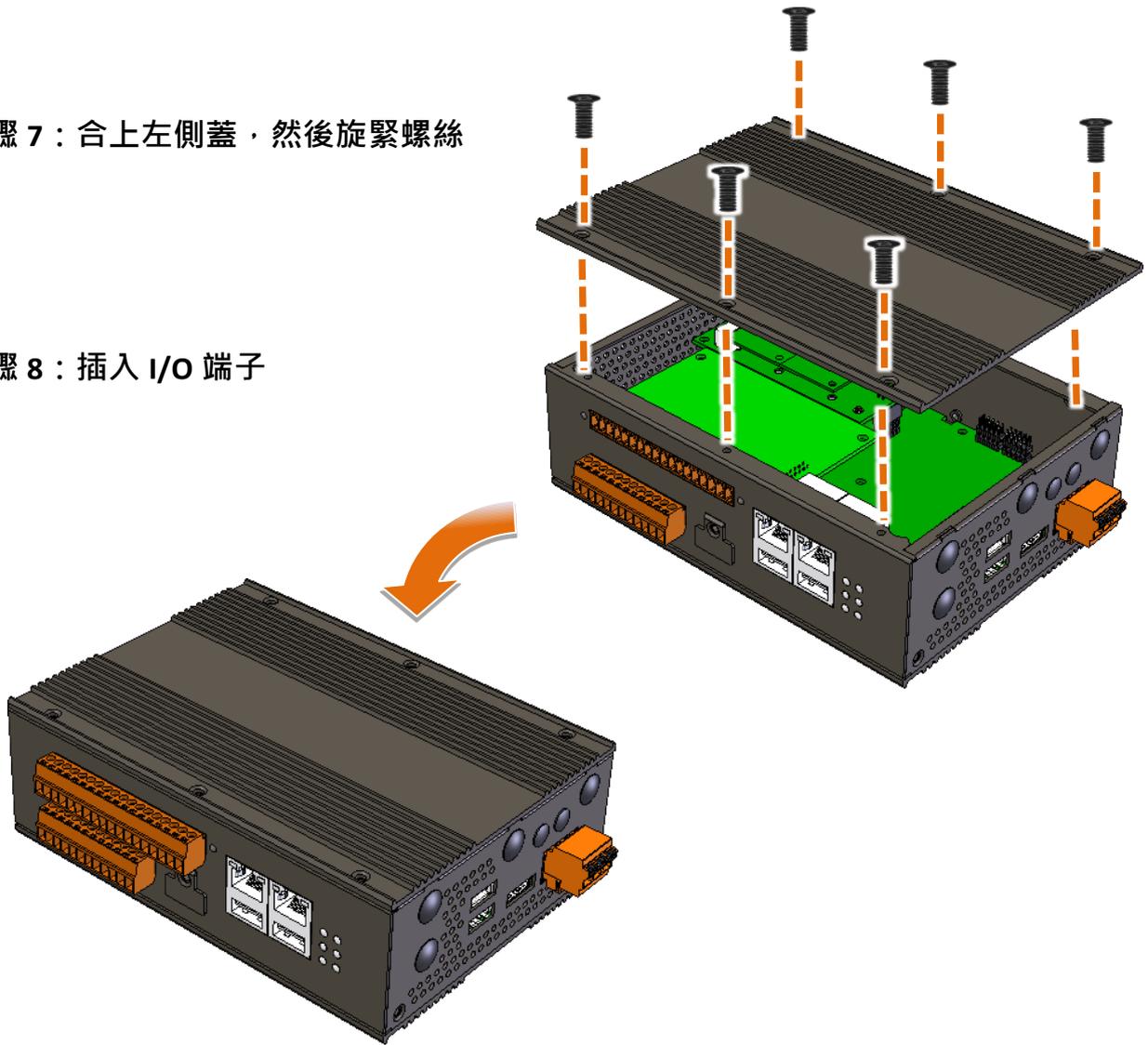
#### 提示&警告



安裝 XV-board/XB-board 擴充模組時，需確認 XV-board/XB-board 擴充模組上的排針已對齊 AIB-2941M PCB 上的排針座，再將其推入。

步驟 7：合上左側蓋，然後旋緊螺絲

步驟 8：插入 I/O 端子

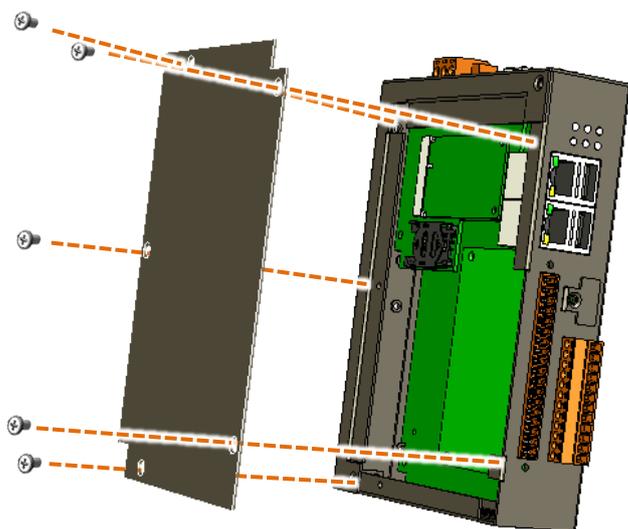


### 2.2.3. 插入 SIM 卡

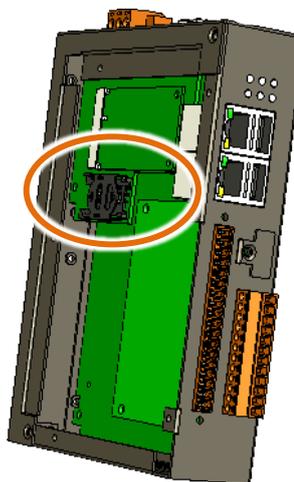
(僅適用已安裝行動網路擴充卡)

SIM 卡插槽位於模組側邊。

步驟 1：使用螺絲起子將側邊的螺絲移除，然後移除側蓋



步驟 2：將 SIM 卡插入 SIM 卡插槽



#### 指導與提示

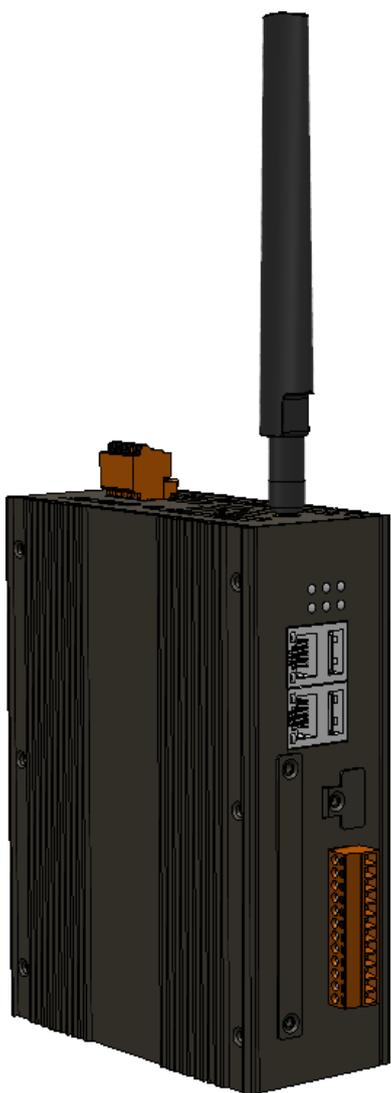


安裝或退出 SIM 卡，須確認 AIB-2941M 已關機。注意不要彎曲或劃傷 SIM 卡。

## 2.2.4. 安裝天線

**(僅適用已安裝行動網路擴充卡)**

行動網路模組，有 1 個可用於連接 3G/4G 天線的天線連接器。安裝天線時，只需將天線緊緊插入連接器，並將天線放在適當的地點即可。

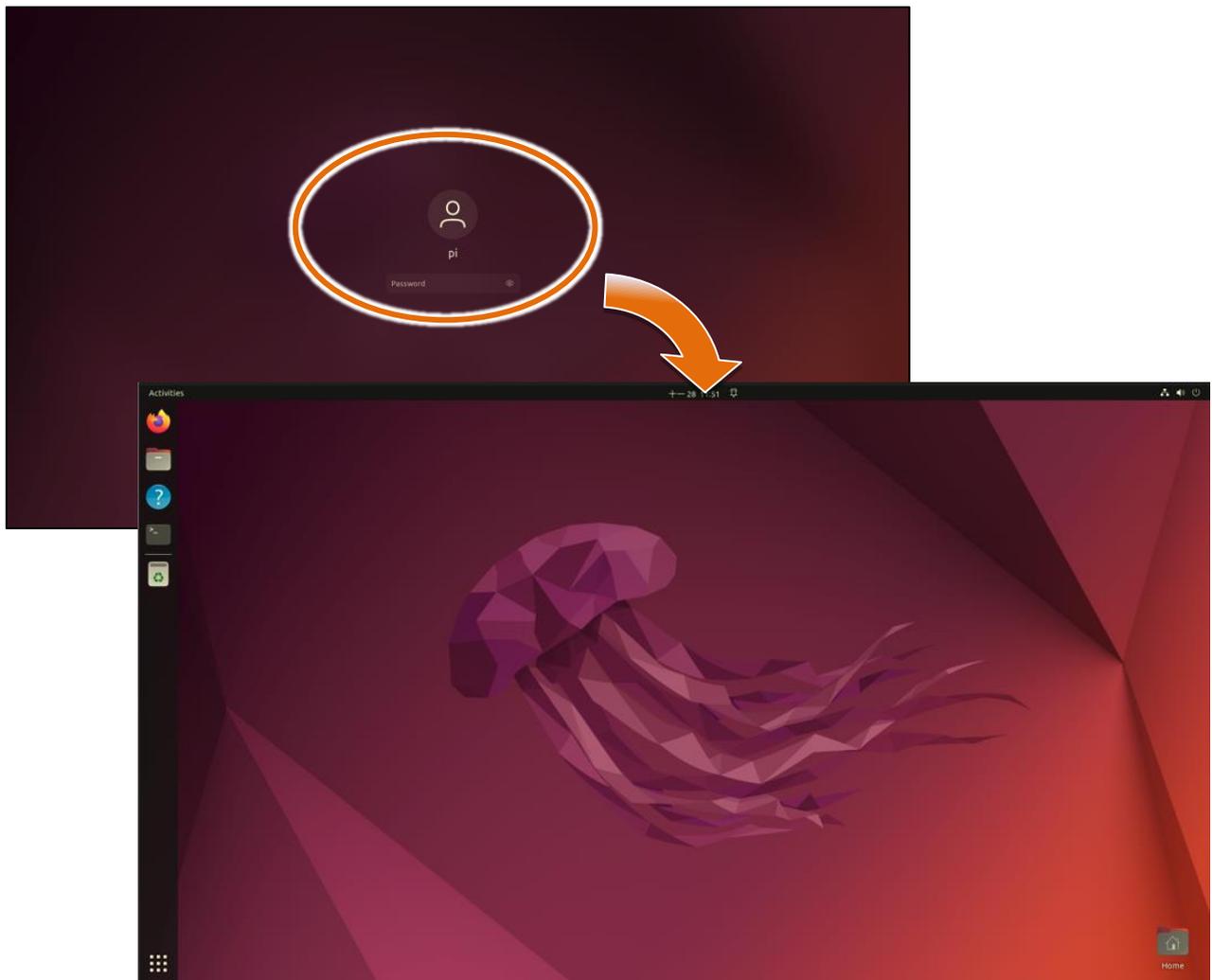
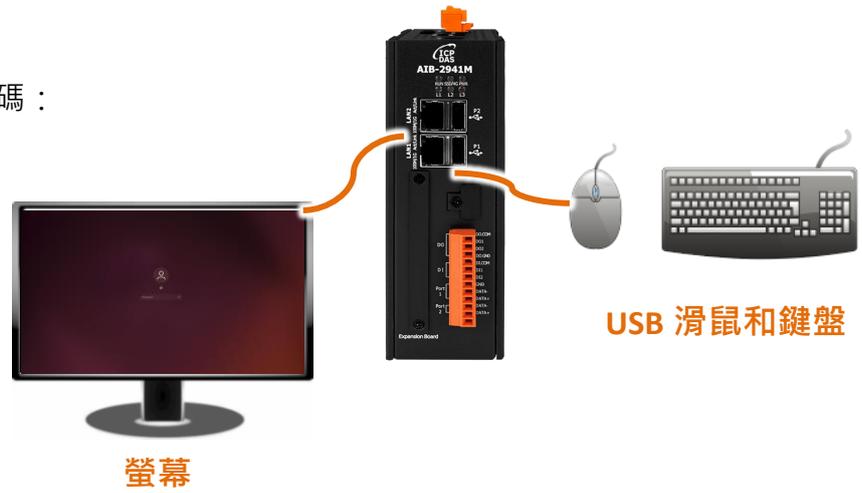


## 2.3. 管理者登入

安裝完成後，啟動 AIB-2941M 即進入管理者登入畫面。

下表為預設的管理者帳戶和密碼：

項目	預設值
用戶名	pi
密碼	icpdas

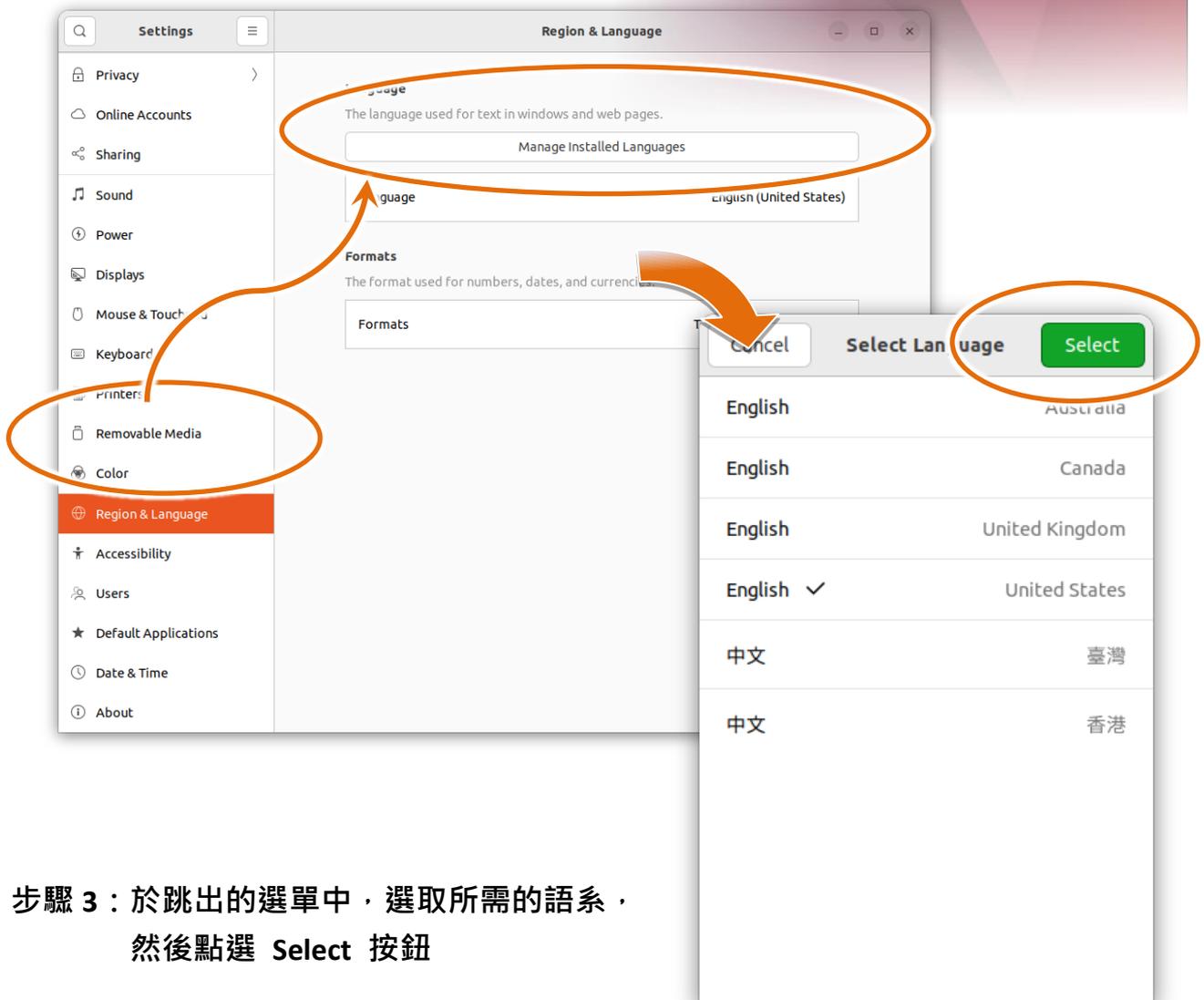
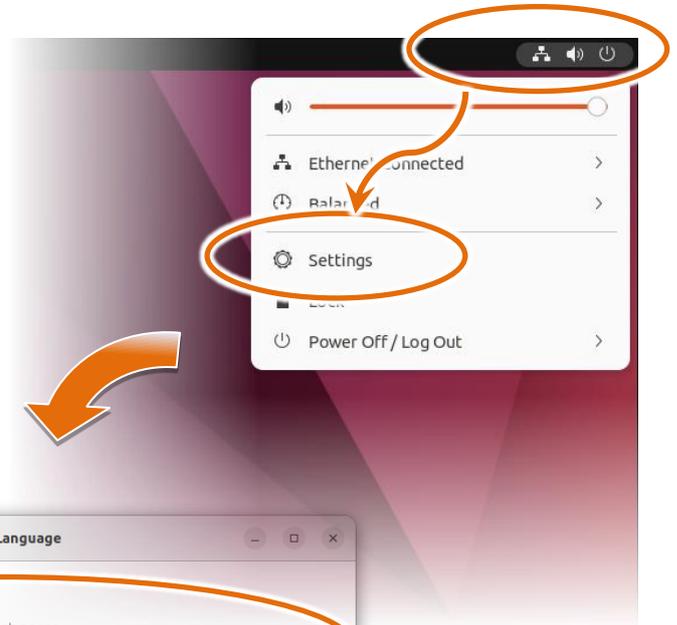


## 2.4. 使用者介面語言變更

Regional Settings 是 Ubuntu 內建功能，它可使使用者輕易改變 AIB-2941M 的使用者介面語系。請依照下列步驟變更語言介面。

**步驟 1：**點選桌面右上角的圖示，於跳出的選單中然點選 **Settings**

**步驟 2：**點選 **Region & Language**，再點選 **Language** 項目

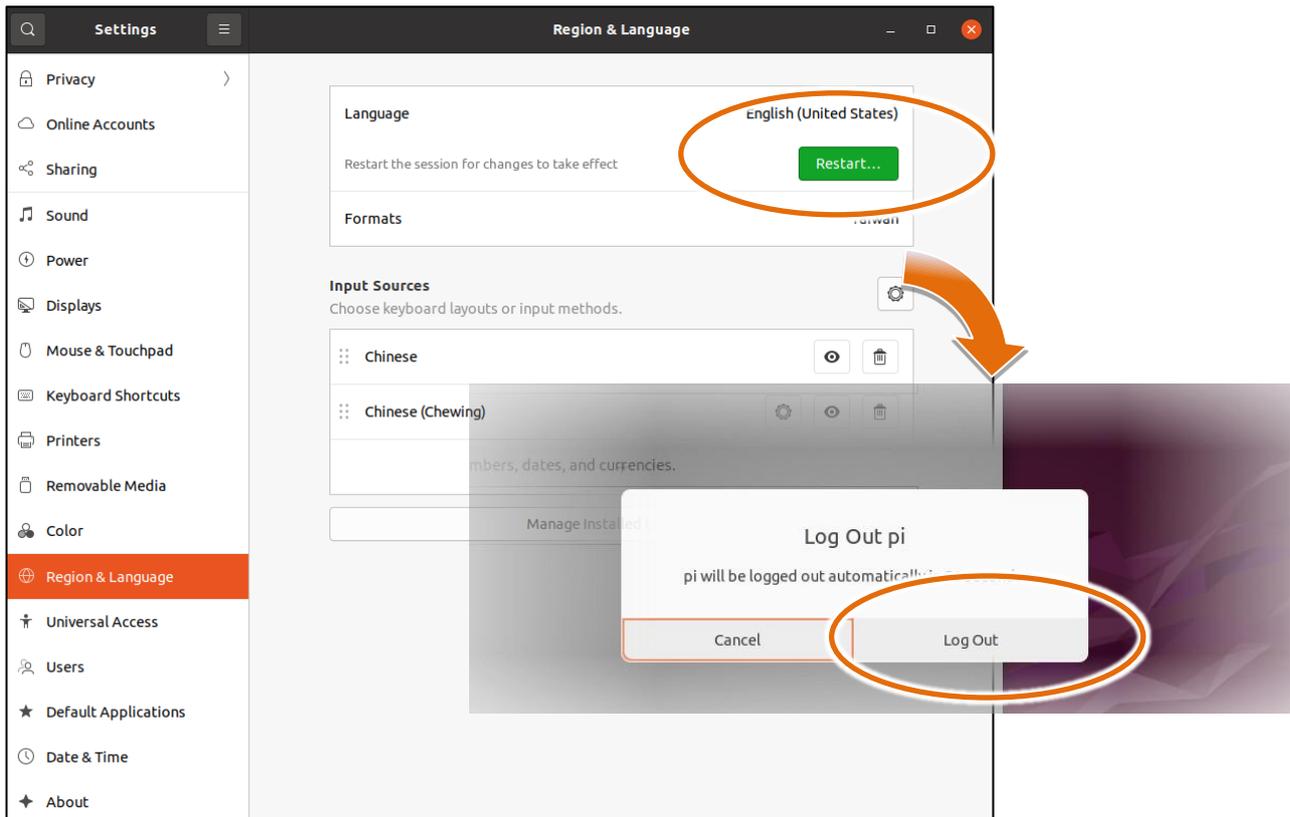


**步驟 3：**於跳出的選單中，選取所需的語系，然後點選 **Select** 按鈕

#### 步驟 4：點選 Restart 按鈕來重新啟動使設定生效

回到 Region&Language 對話視窗，在選取不同語系的狀況下，Language 項目會出現 Restart 按鈕

#### 步驟 5：點選 Log Out 確認重新啟動流程

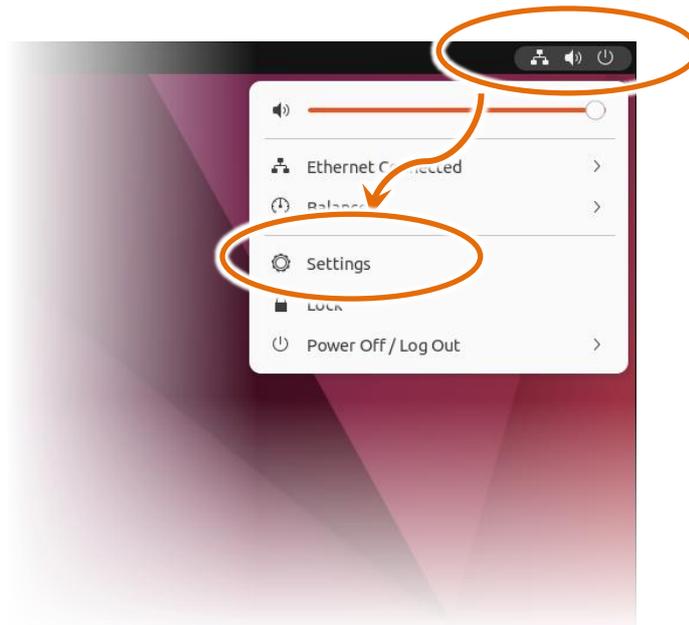


#### 步驟 6：於管理者登入畫面中輸入密碼重新登入，完成系統語系變更程序

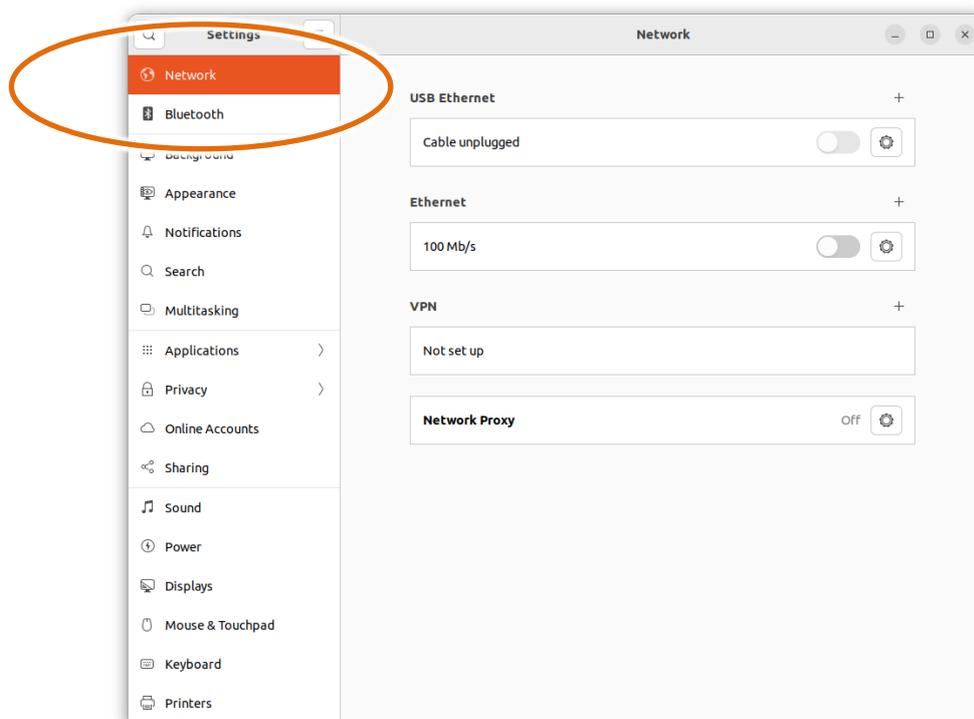
## 2.5. IP 設定

Ubuntu 內建 IP 設定的功能，請依照下列步驟設定 IP。

**步驟 1：**點選桌面右上角的圖示，於跳出的選單中然點選 **Settings**



**步驟 2：**點選 **Network**



### 步驟 3：設定 IP

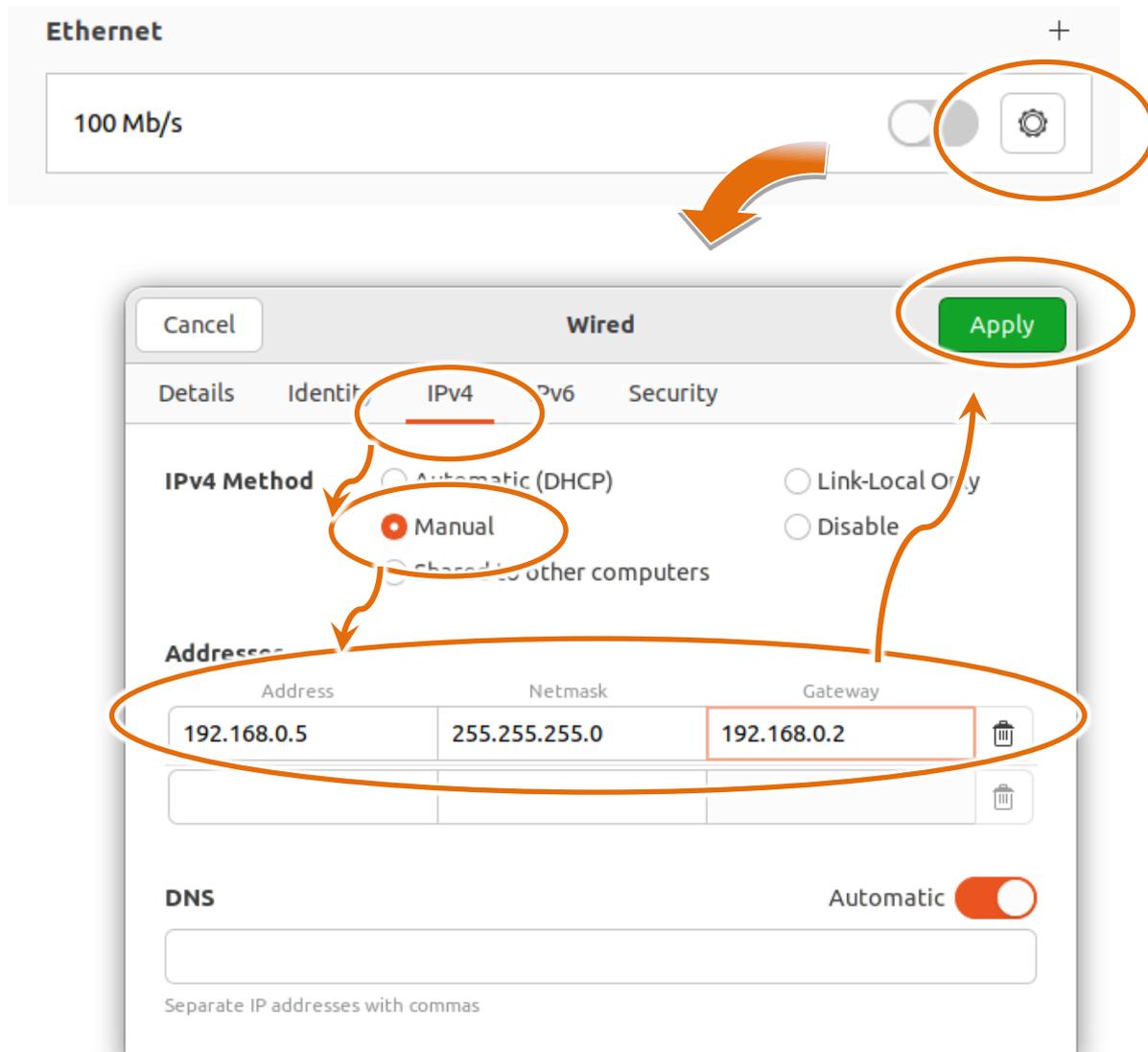
#### 使用 DHCP 指派 IP 位址

直接點選開關按鈕啟用即可



#### 手動指派 IP 位址

點選設定  按鈕，於跳出的對話視窗點選 IPv4 頁籤後，選取 Manual 選項，設定 IP 位址，完成後點選 Apply 按鈕



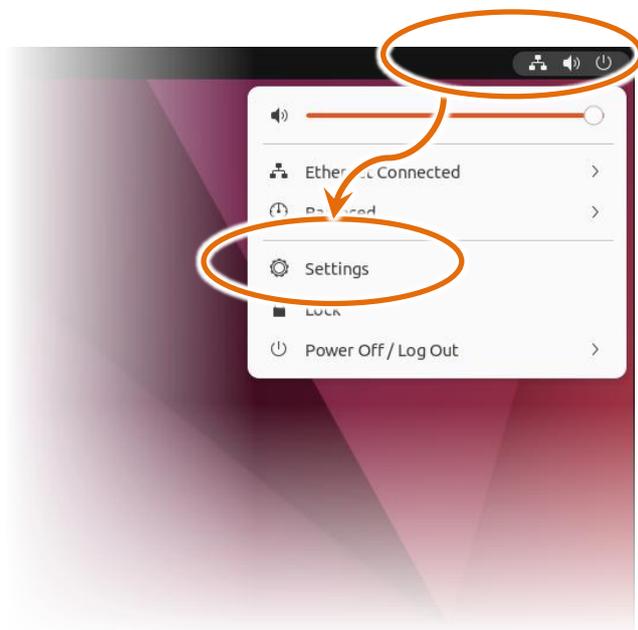
### 3. 工具與任務

本章簡單介紹 AIB-2941M 常用的系統、軟體工具及其優點。

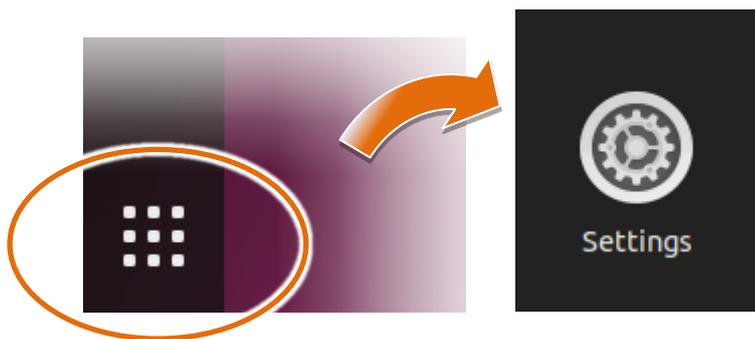
AIB-2941M 除了 Ubuntu 內建的軟體工具外，還設計了一些專為 AIB-2941M 設計的軟體工具。

#### 3.1. 系統工具

系統工具可經由點選桌面上右上角圖示，然後點選 Settings 進入做設定。



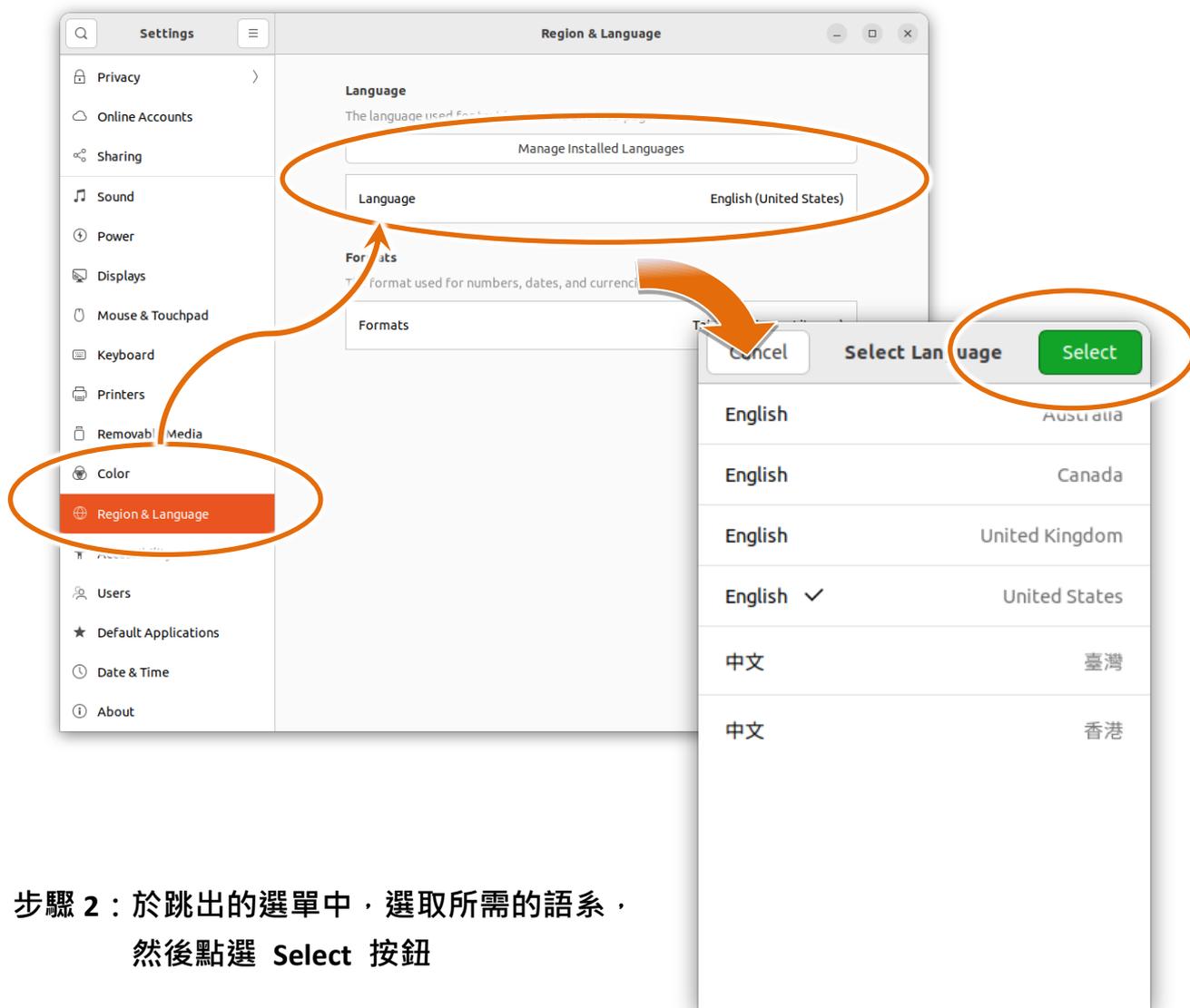
亦可點選左下角  圖示，然後點選 Settings 進入做設定。



### 3.1.1. 使用者介面語言變更

Regional Settings 是 Ubuntu 內建功能，它可使使用者輕易改變 AIB-2941M 的使用者介面語系。請依照下列步驟變更語言介面。

**步驟 1：**點選 **Region & Language**，再點選 **Language** 項目

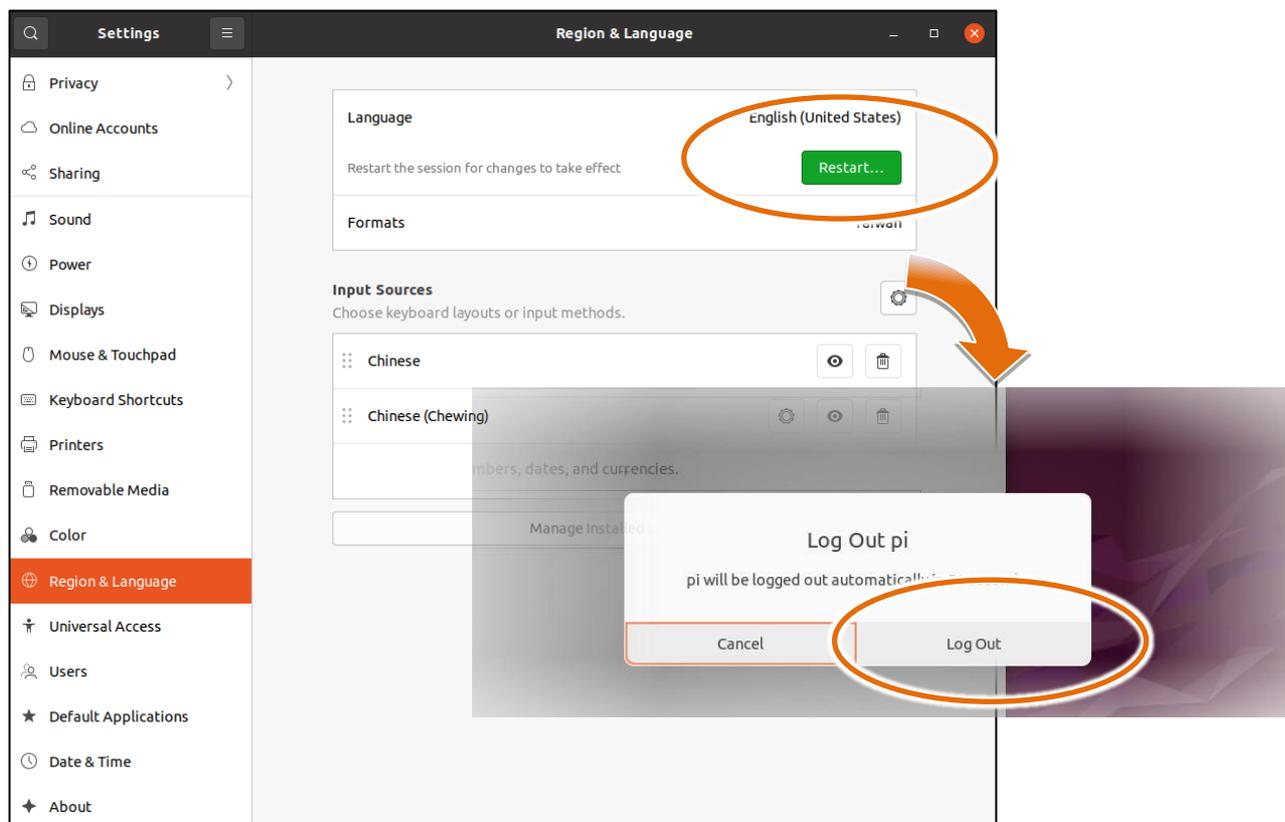


**步驟 2：**於跳出的選單中，選取所需的語系，然後點選 **Select** 按鈕

**步驟 3：**點選 **Restart** 按鈕來重新啟動使設定生效

回到 Region&Language 對話視窗，在選取不同語系的狀況下，Language 項目會出現 Restart 按鈕

#### 步驟 4：點選 Log Out 確認重新啟動流程

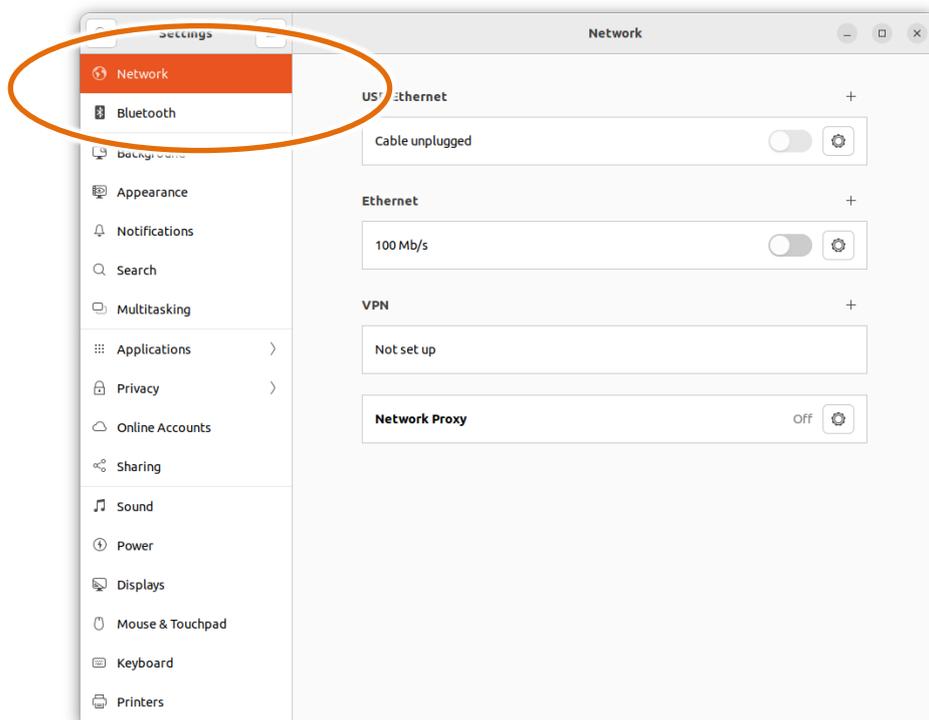


#### 步驟 5：於管理者登入畫面中輸入密碼重新登入，完成系統語系變更程序

### 3.1.2. IP 設定

Ubuntu 內建 IP 設定的功能，請依照下列步驟設定 IP。

#### 步驟 1：點選 Network



## 步驟 2：設定 IP

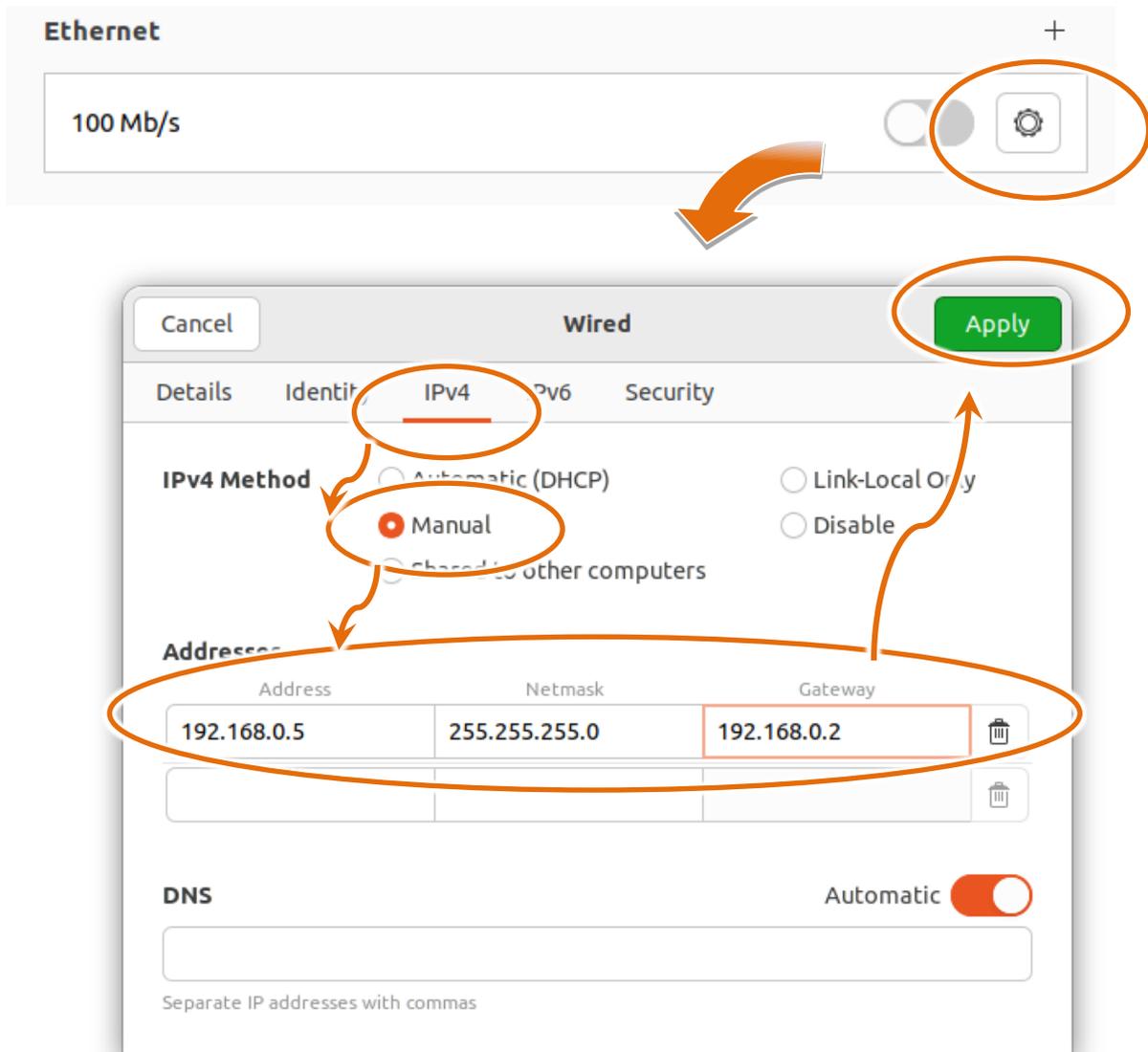
### 使用 DHCP 指派 IP 位址

直接點選開關按鈕啟用即可



### 手動指派 IP 位址

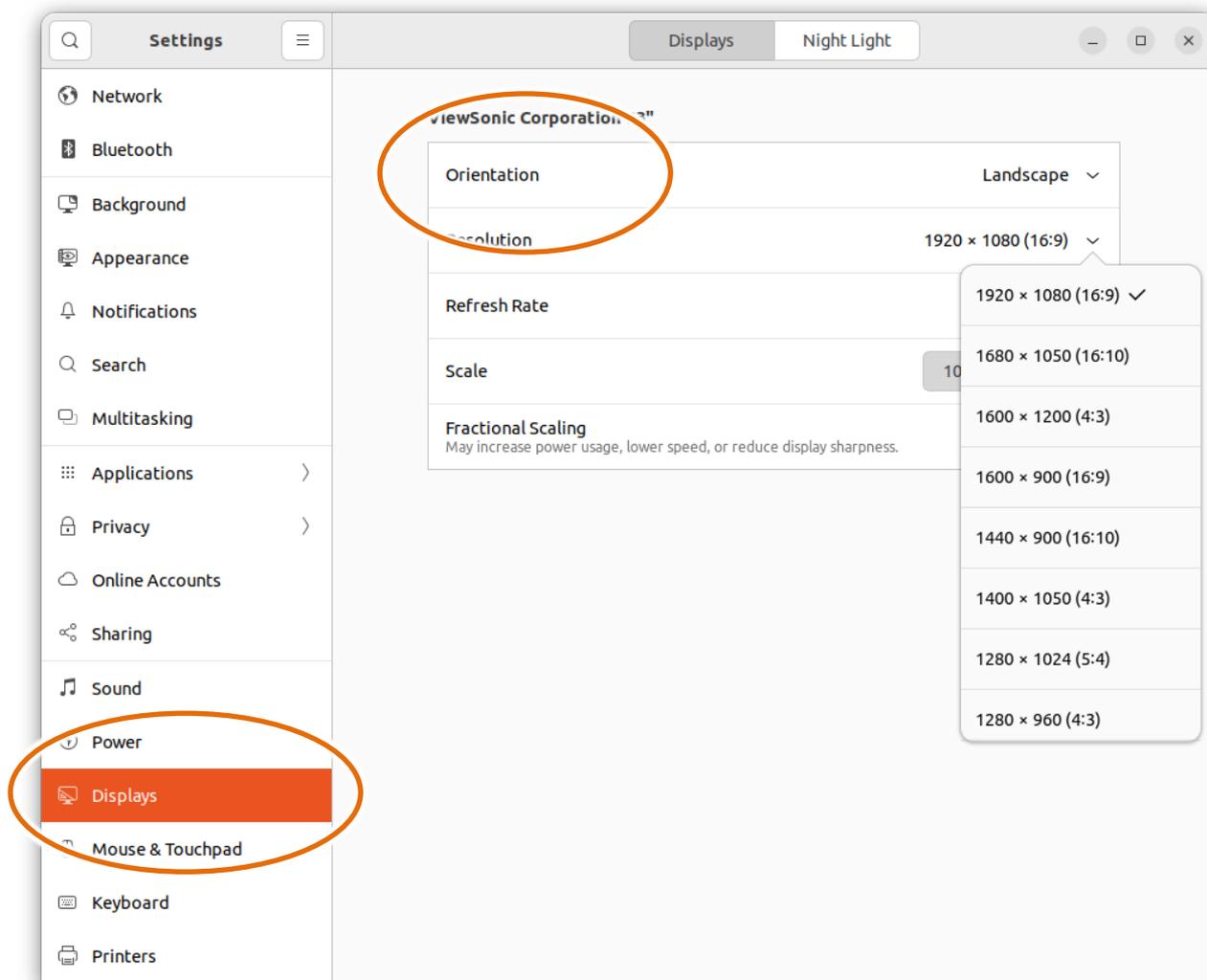
點選設定  按鈕，於跳出的對話視窗點選 IPv4 頁籤後，選取 Manual 選項，設定 IP 位址，完成後點選 Apply 按鈕



### 3.1.3. 螢幕解析度設定

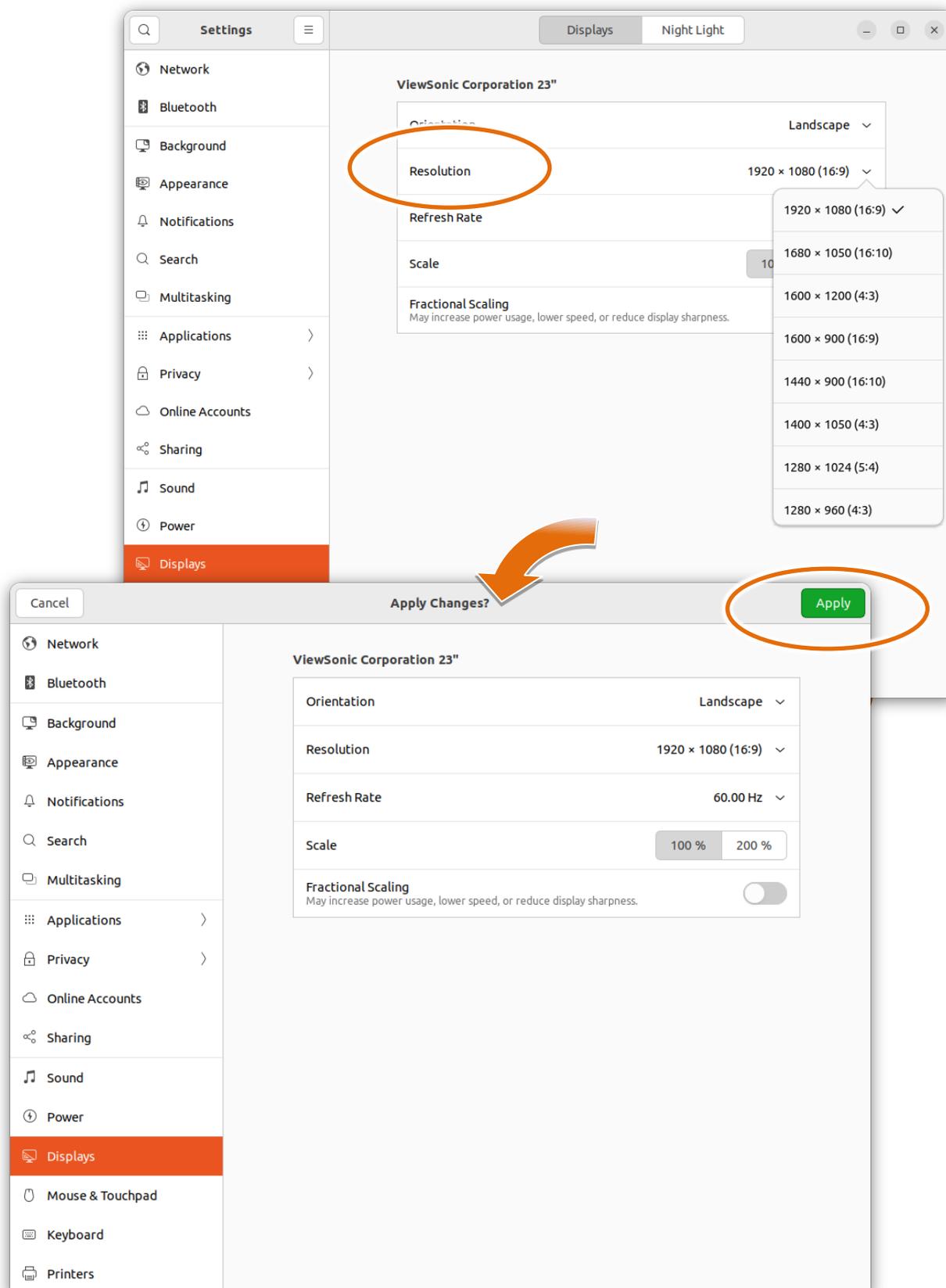
Ubuntu 內建有解析度設定的功能，解析度可設定範圍從 800\*600~3410\*2160(4K)，請依照下列步驟設定顯示器的最大解析度。

#### 步驟 1：點選 Display 頁籤

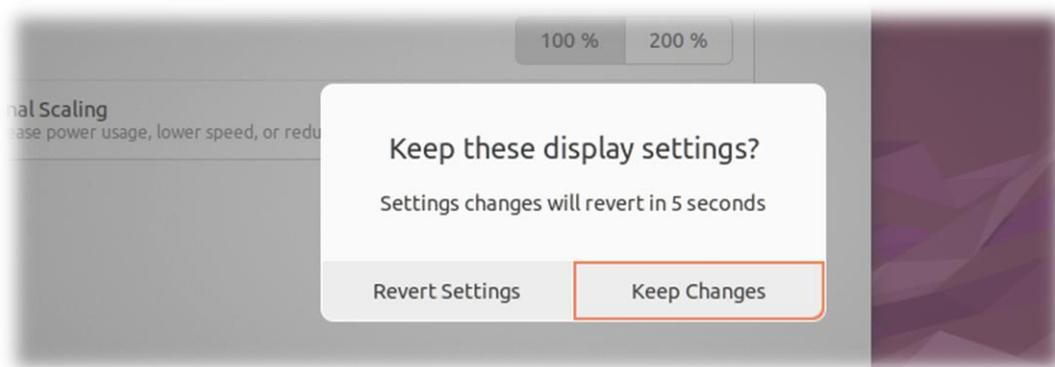


步驟 2：點選 Resolution 項目，設定解析度

步驟 3：設定完成後點選 Apply 按鈕



步驟 4：選擇的解析度如果能正常顯示，於跳出的對話訊息點選 **Keep Changes**，如果無法正常則點選 **Revert Settings** 或等待 20 秒後會回到先前的設定值

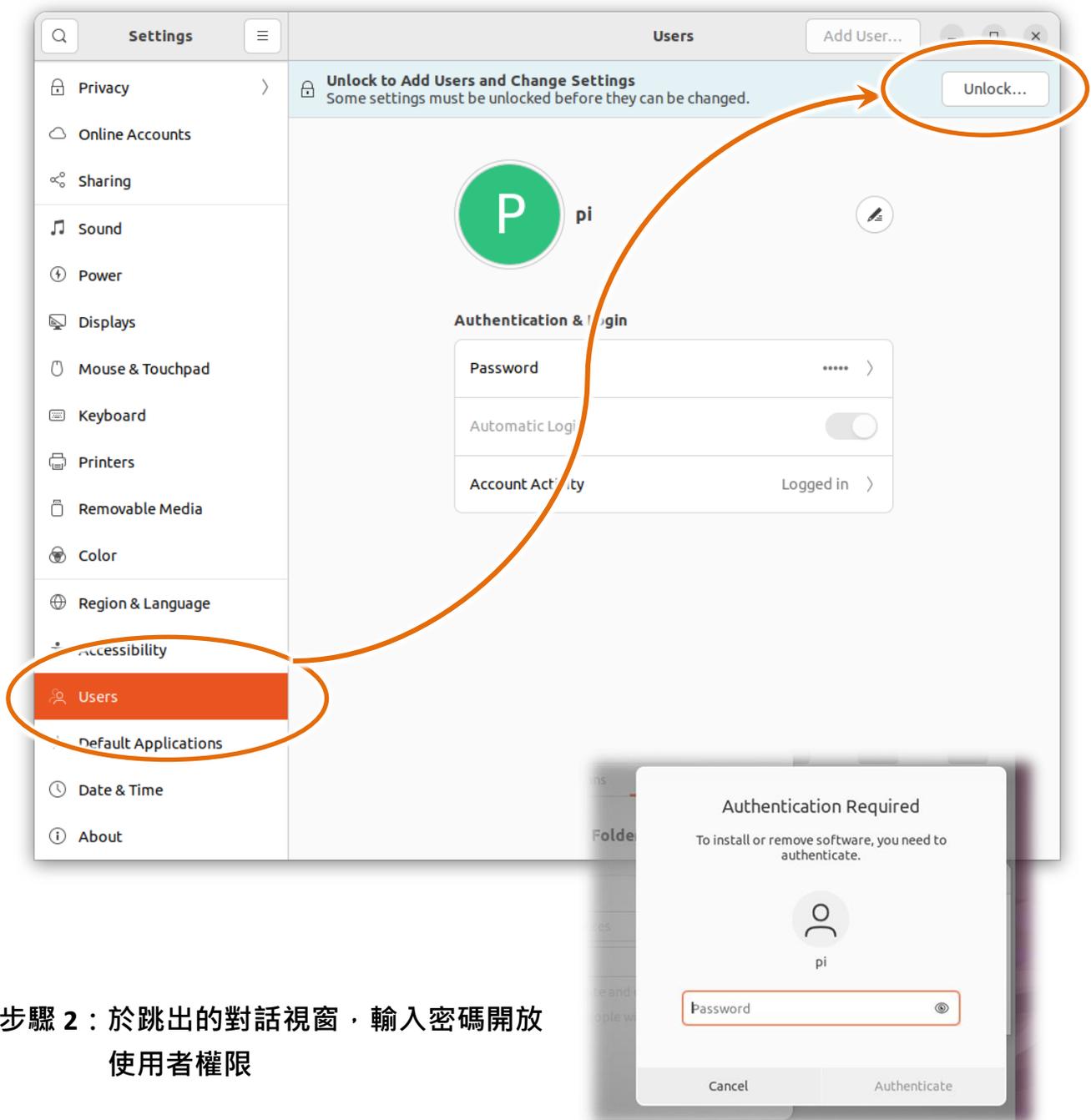


### 3.1.4. 使用者帳戶管理

使用者帳戶管理是一安全功能，有助於防止對 AIB-2941M 進行未經授權的系統變更。

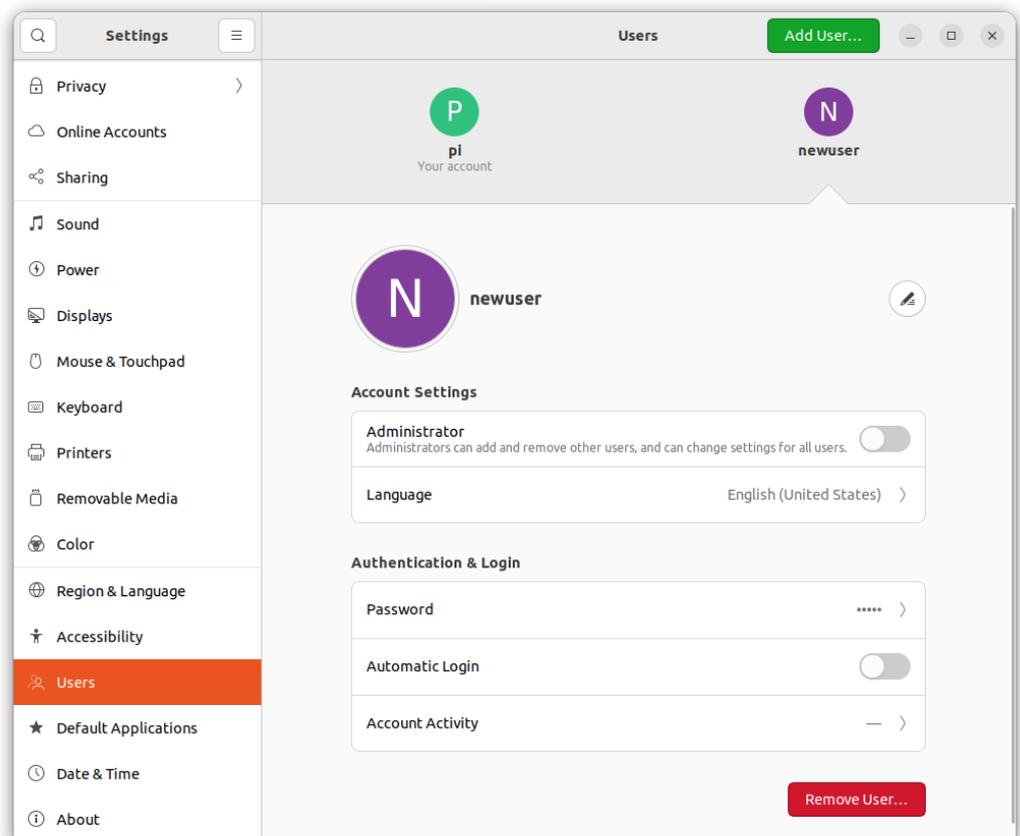
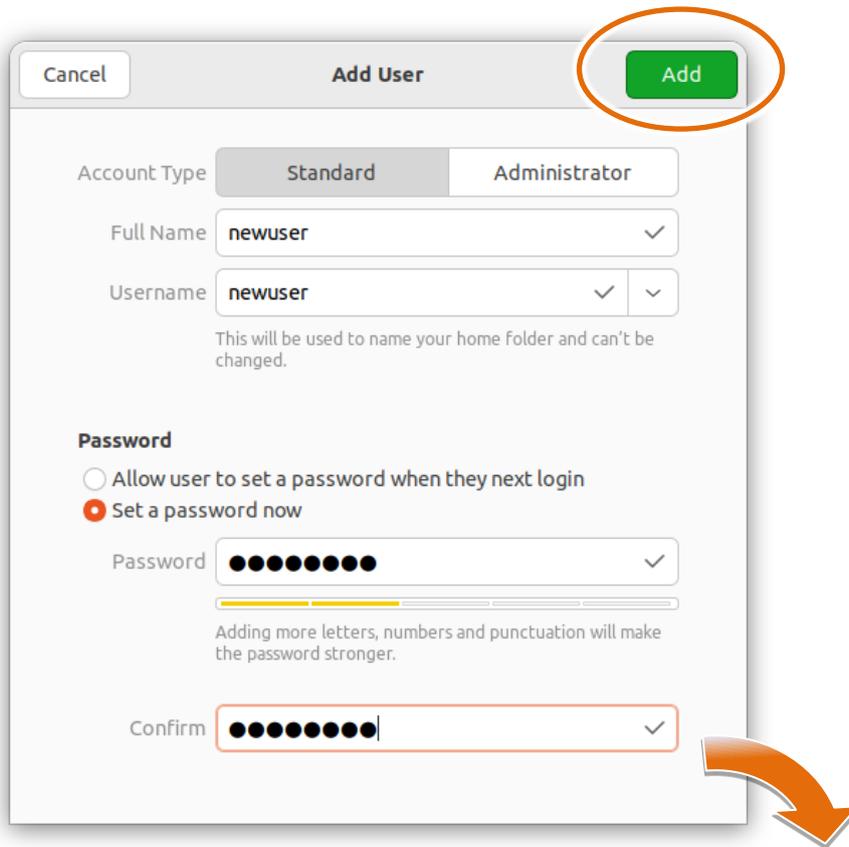
#### 新增使用者帳戶

步驟 1：點選 Users，再點選 Unlock... 按鈕



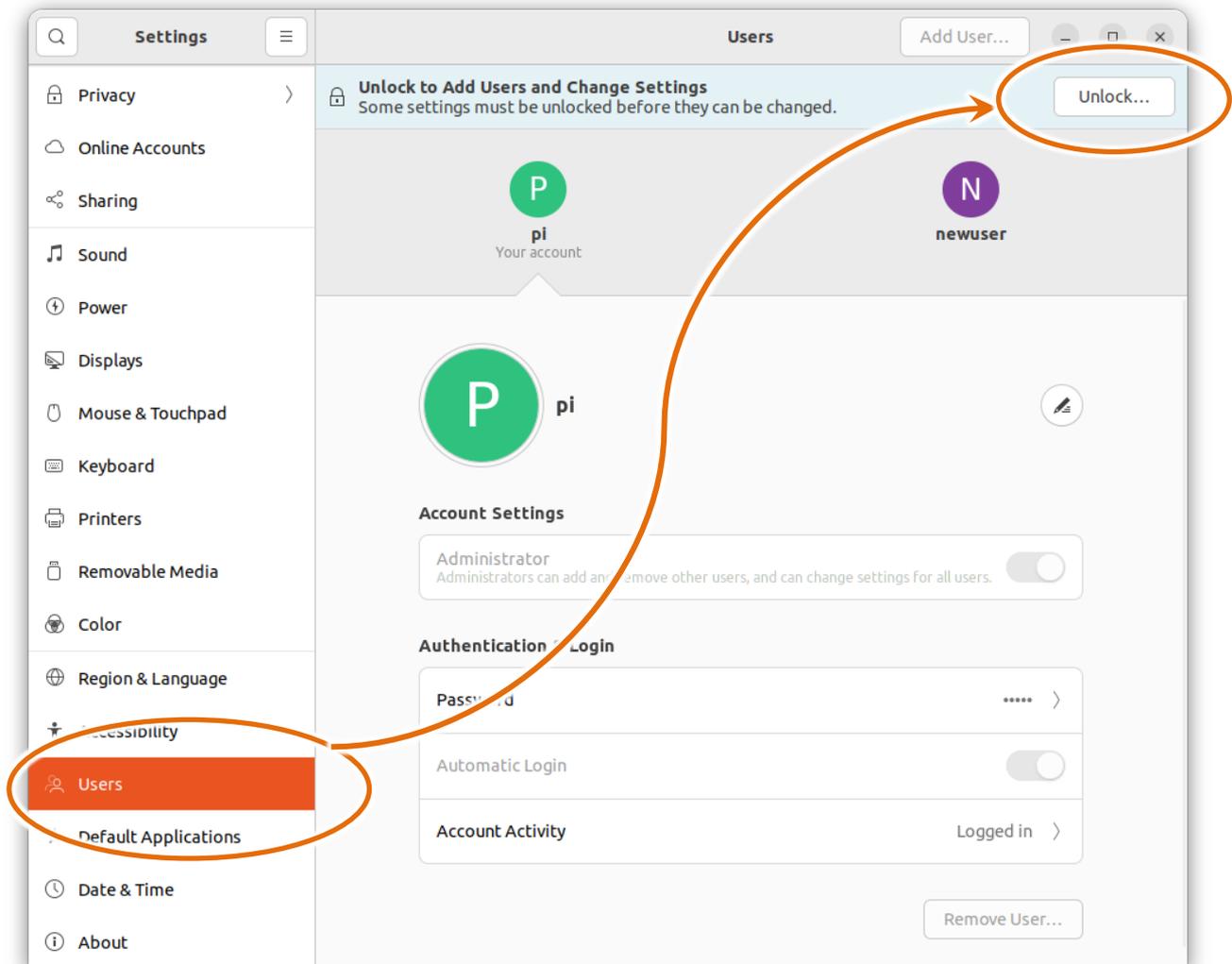
步驟 2：於跳出的對話視窗，輸入密碼開放使用者權限

步驟 3：於 Add User 對話視窗中，輸入使用者帳戶資料後，點選 Add 按鈕完成使用者建立

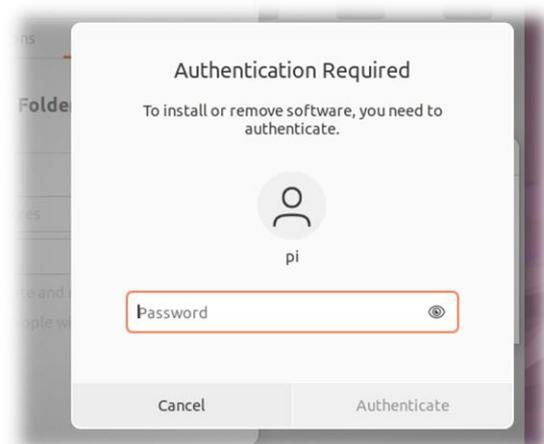


## 刪除管理者帳戶

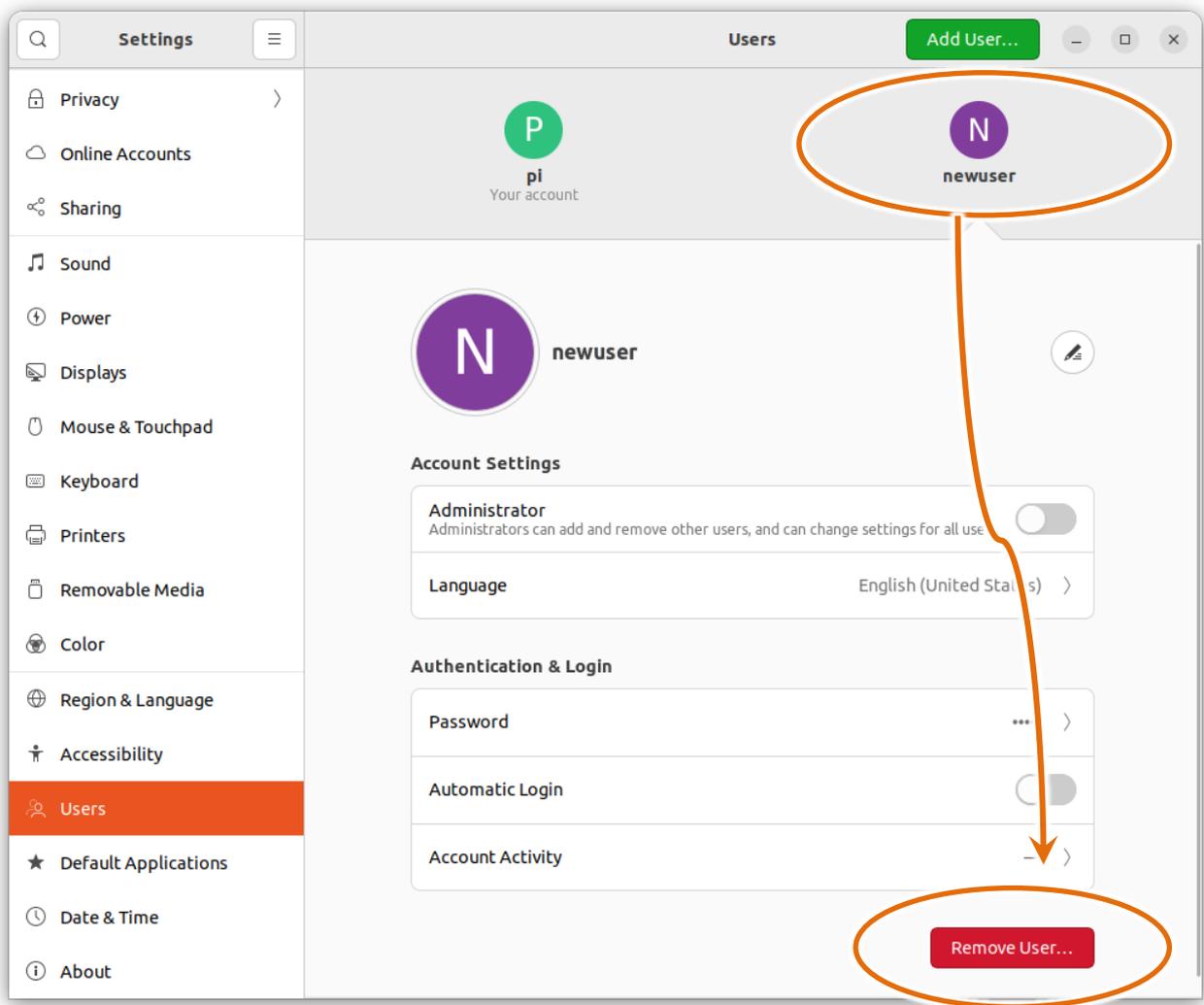
步驟 1：點選 Users，再點選 Unlock... 按鈕



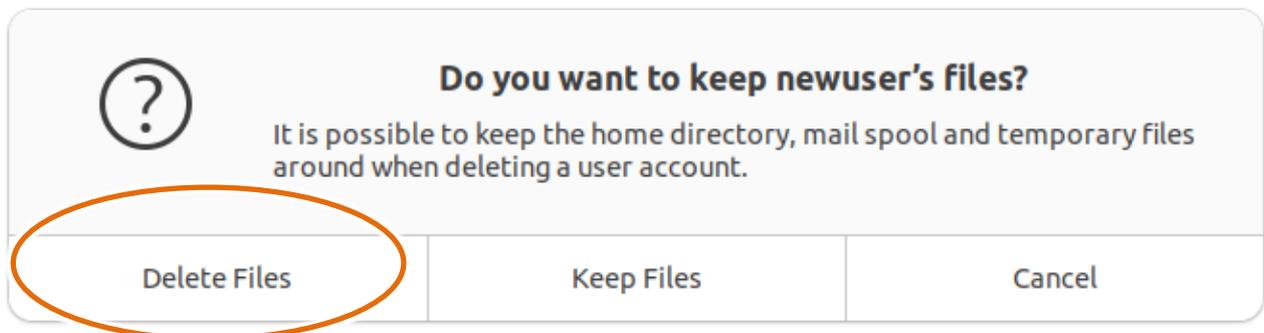
步驟 2：於跳出的對話視窗，輸入密碼開放使用者權限



步驟 3：點選欲刪除的使用者帳戶，然後點選 **Remove User...**



步驟 4：於跳出的警告對話視窗中，點選 **Delete Files** 按鈕確認刪除使用者



## 3.2. 軟體工具

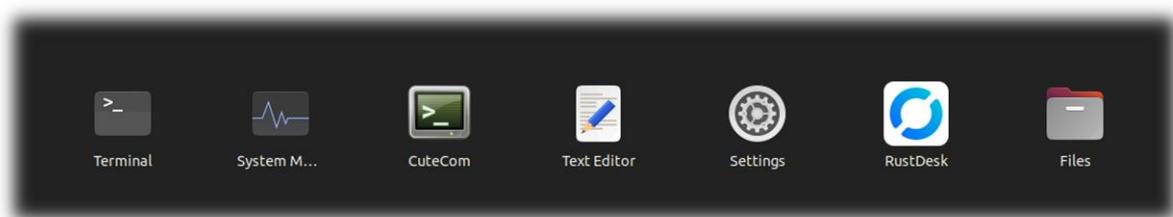
軟體工具可透過點選左下角  圖示進入做檢視。本節介紹些常用的軟體工具。



### 3.2.1. Utility – 檢視系統資訊

Utility 整合了數個應用程式於一體，方便管理與控制 AIB-2941M 中的系統與功能。

#### 點選 Utility



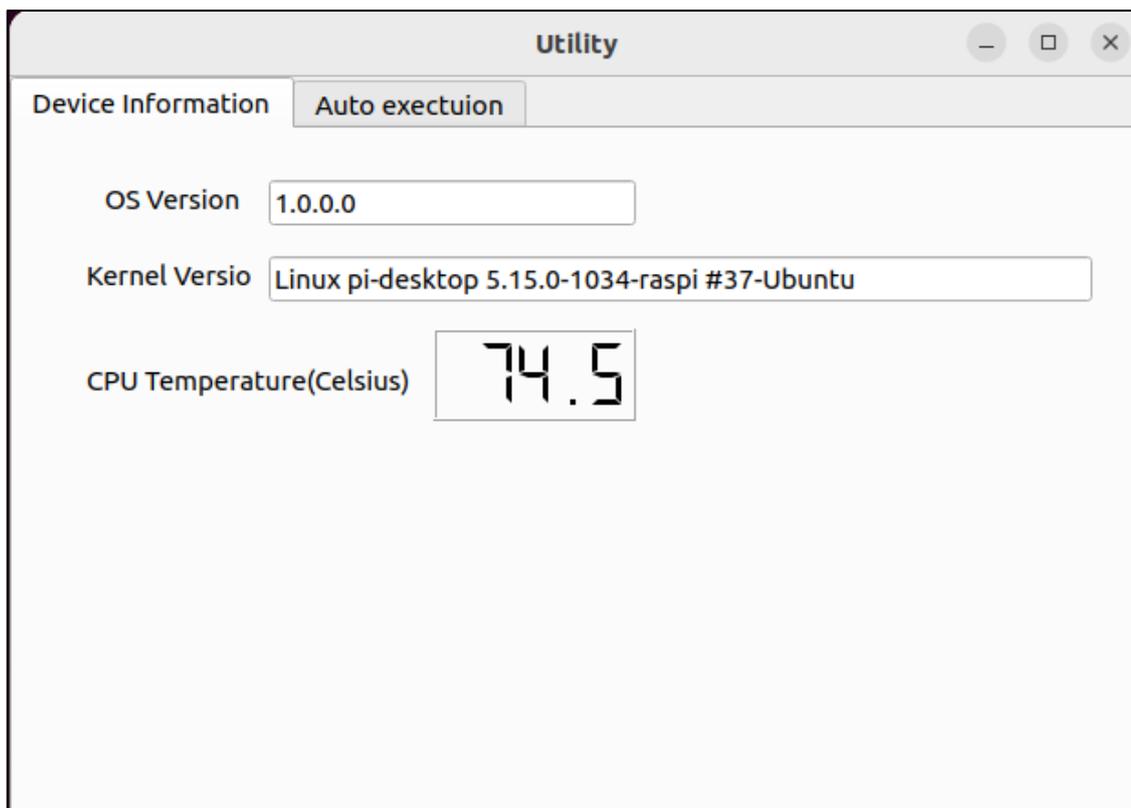
下列為 Utility 系統頁籤及其功能說明。

頁籤	功能說明
Device Information	檢視 OS 版本、kernel 版本和 CPU 溫度
Auto Execution	設定開機啟動程式

## 系統版本、CPU 溫度檢視

Device Information 頁籤提供 AIB-2941M OS 版本、CPU 溫度...等系統資訊。

Utility 所提供的系統資訊如下圖所示。



## 管理開機啟動程式

Auto Execution 頁籤提供管理開機後自動執行程式的功能，自動執行的程式最多 7 個。

### 指導與提示



自動執行的程式附檔名為 .py 和可執行檔案，執行的順序從 program 1 到 program 7。

路徑指定完成後、點選 Apply 按鈕，即可設定完成。

Label	Input Field	Button
Pogram 1	<input type="text"/>	Browse
Pogram 2	<input type="text"/>	Browse
Pogram 3	<input type="text"/>	Browse
Pogram 4	<input type="text"/>	Browse
Pogram 5	<input type="text"/>	Browse
Pogram 6	<input type="text"/>	Browse
Pogram 7	<input type="text"/>	Browse

Apply

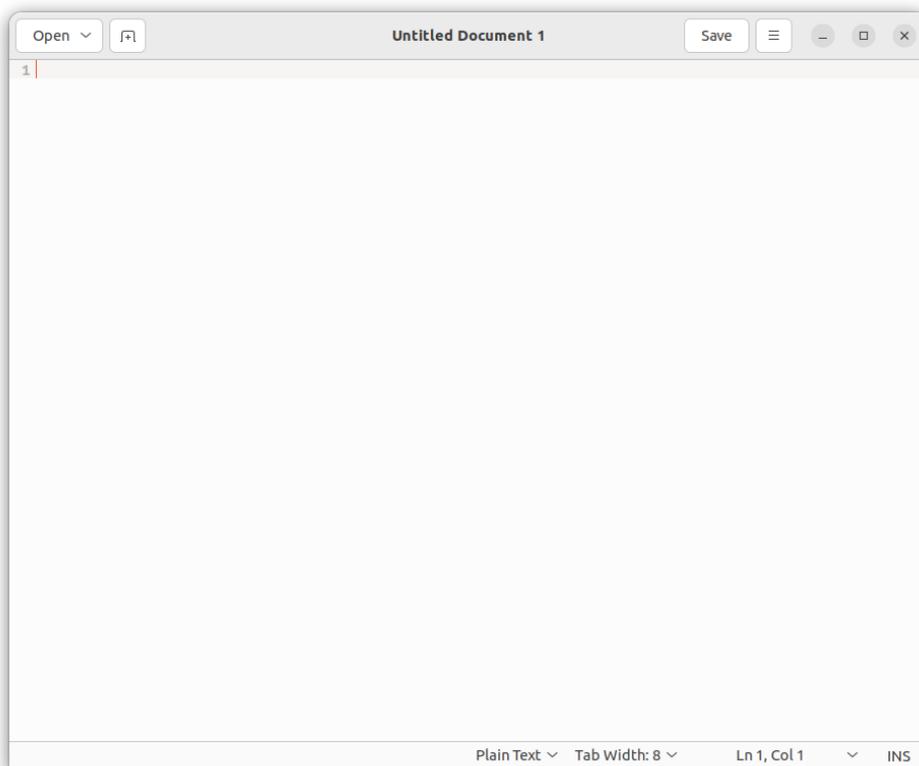
### 3.2.2. Text Edit 文字編輯器

Text Edit 是一個簡單的文字編輯器，檔案內容沒有任何格式標籤或風格。請在程式列中選擇 Text Edit 並執行。

#### 點選 Text Editor



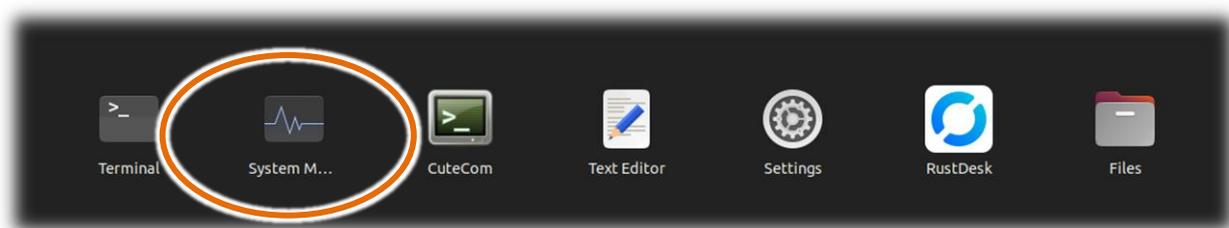
Text Edit 開啟後執行畫面如下



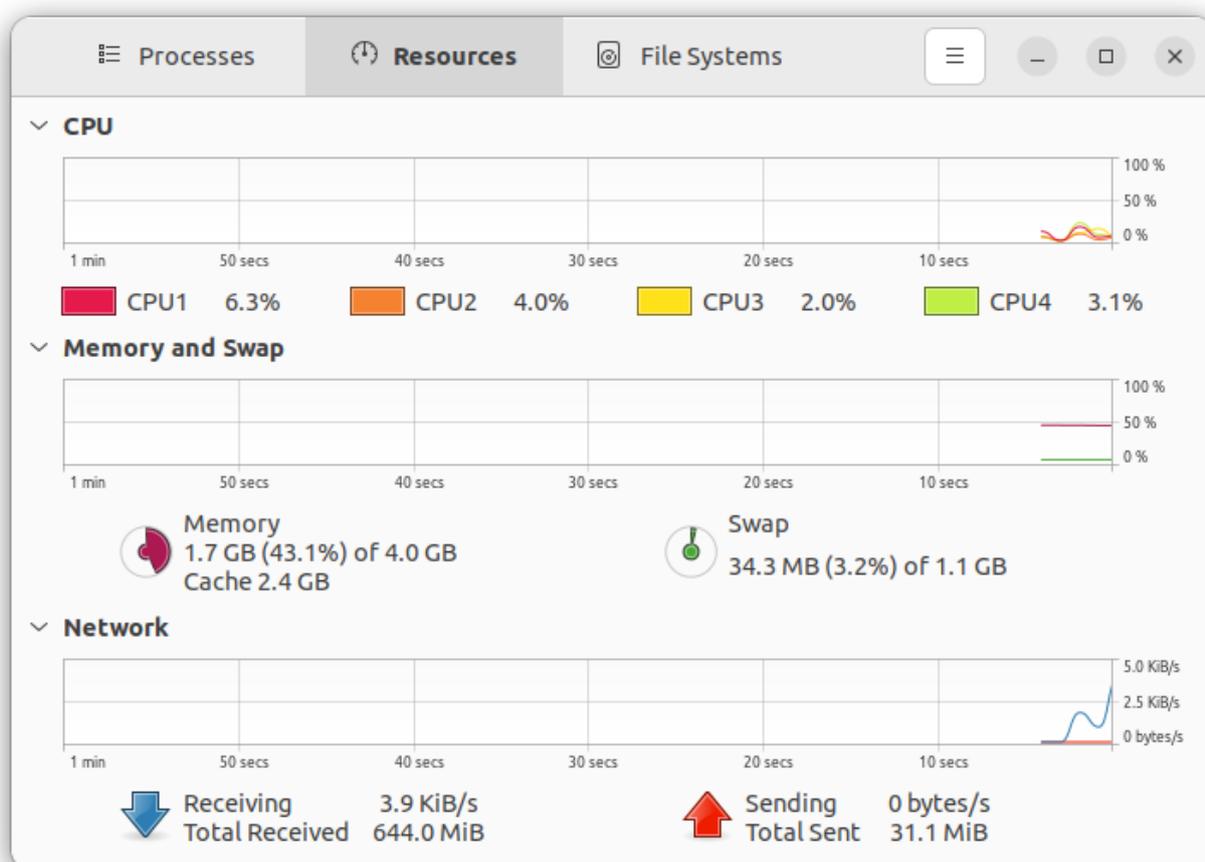
### 3.2.3. 效能監控管理

System Monitor 提供了所有程序、系統執行序的即時訊息，類似 Windows 的工作管理員，Ubuntu 已內建此軟體，開啟步驟如下。

#### 點選 System Monitor



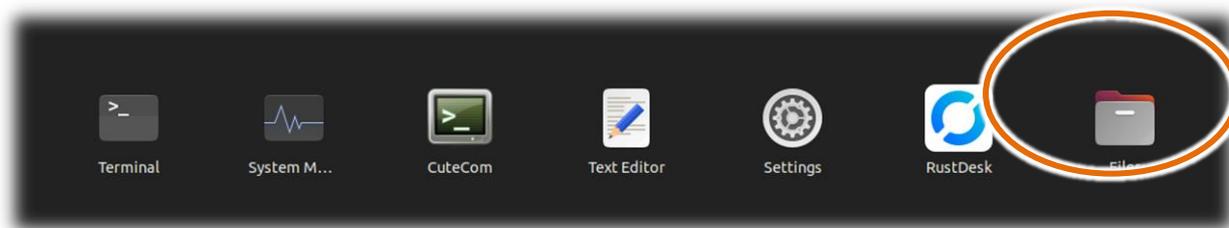
#### System Monitor 開啟後執行畫面如下



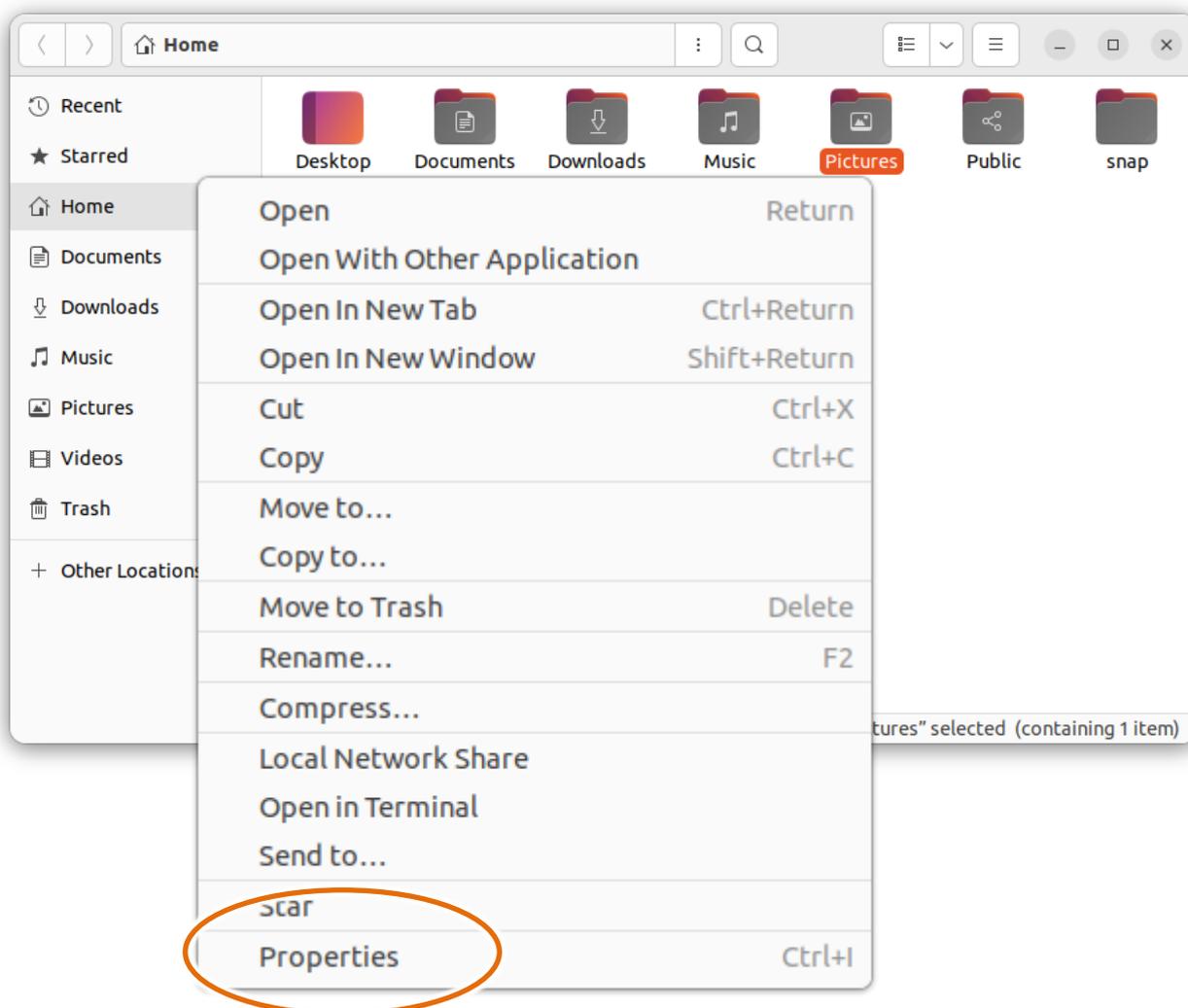
### 3.2.4. 網路芬鄰設定

AIB-2941M 也有支援網路芬鄰功能，請依照以下步驟設定。

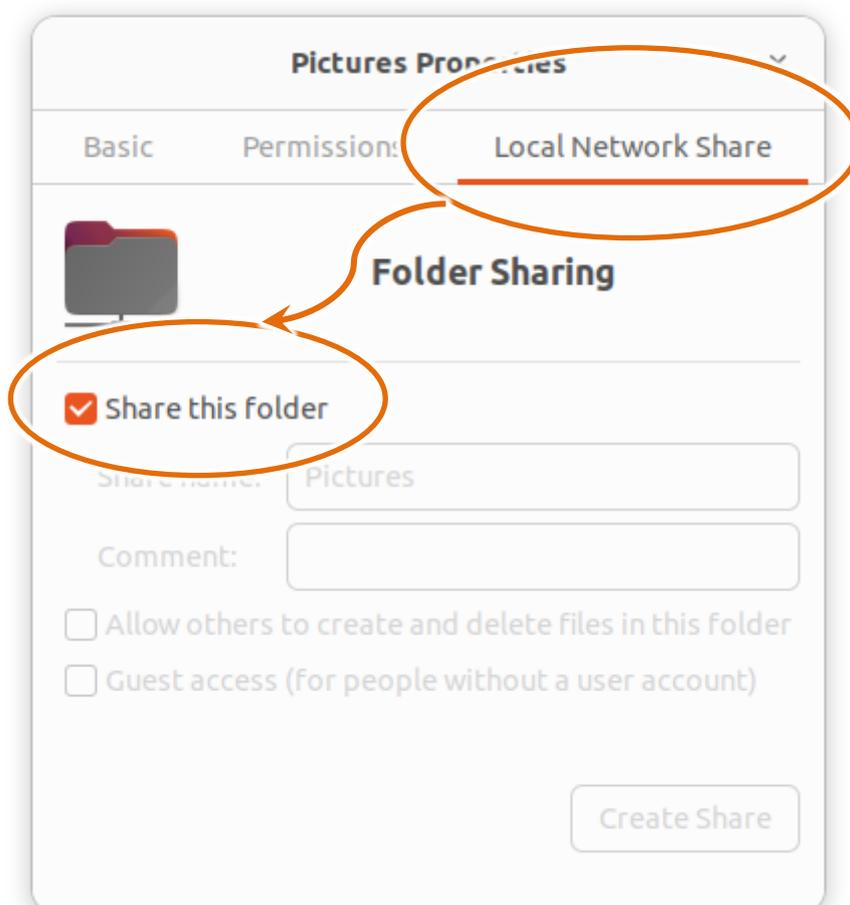
#### 步驟 1：點選 Files



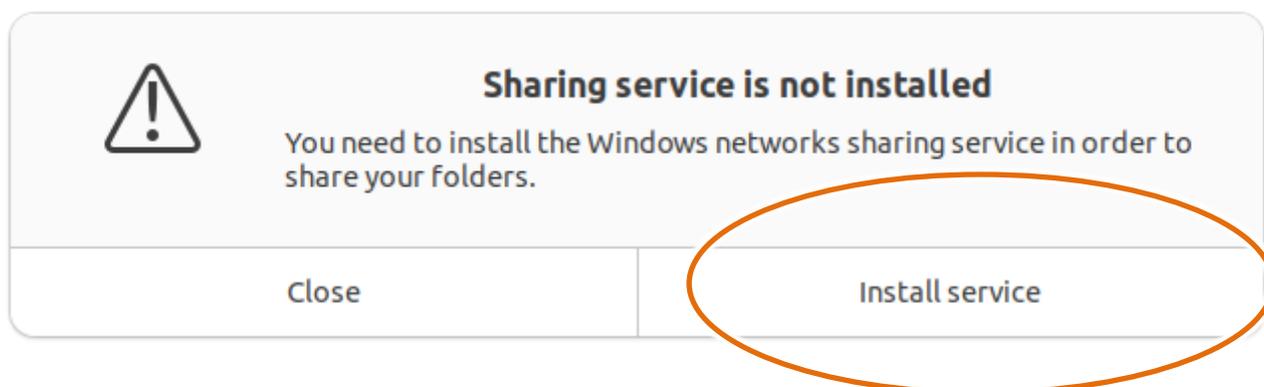
#### 步驟 2：在要分享的目錄點選滑鼠右鍵，然後點選 “Properties”



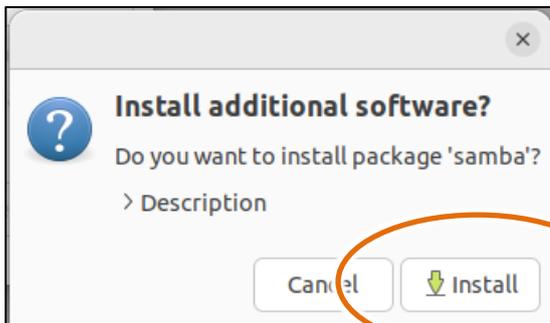
步驟 3：於跳出的 Properties 對話視窗中，點選 Local Network Share 頁籤，然後點選 Share this folder



步驟 4：於跳出的警告對話視窗，點選 Install service 安裝 Windows 網路芳鄰服務

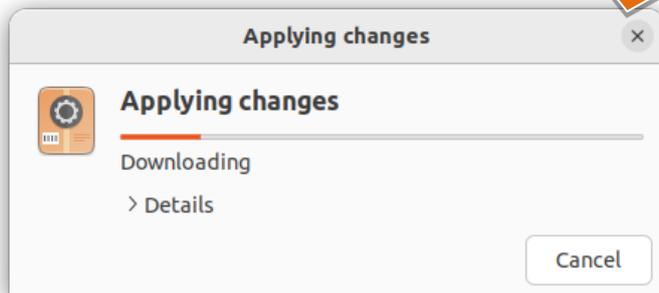
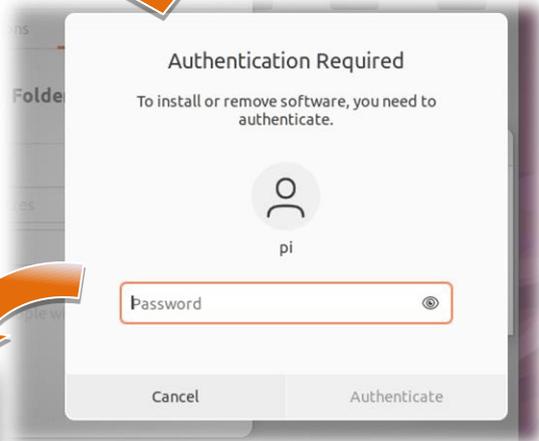


步驟 5：於跳出的對話視窗，點選 Install 進行安裝

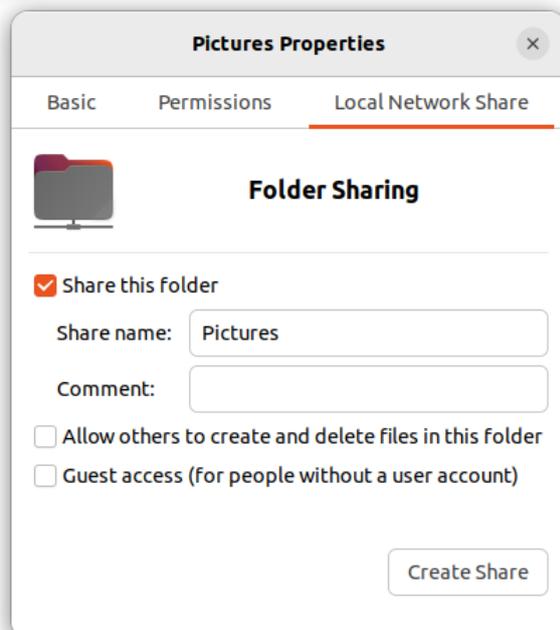


步驟 6：於跳出的對話視窗，輸入密碼開放使用者權限

安裝約 10 秒結束。



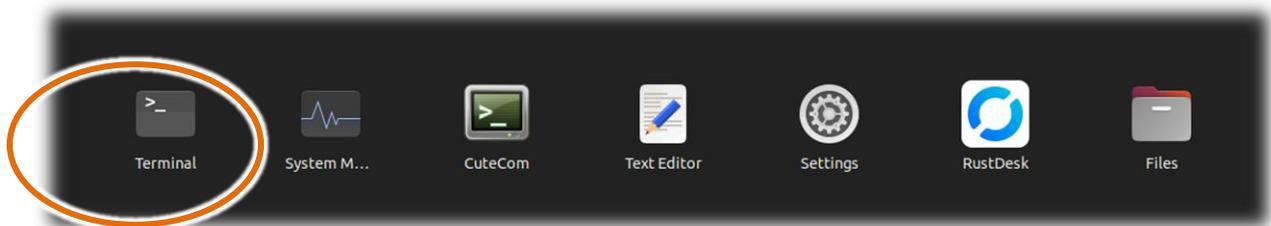
步驟 7：安裝完成後，網路芳鄰功能已啟用，也可自行設定



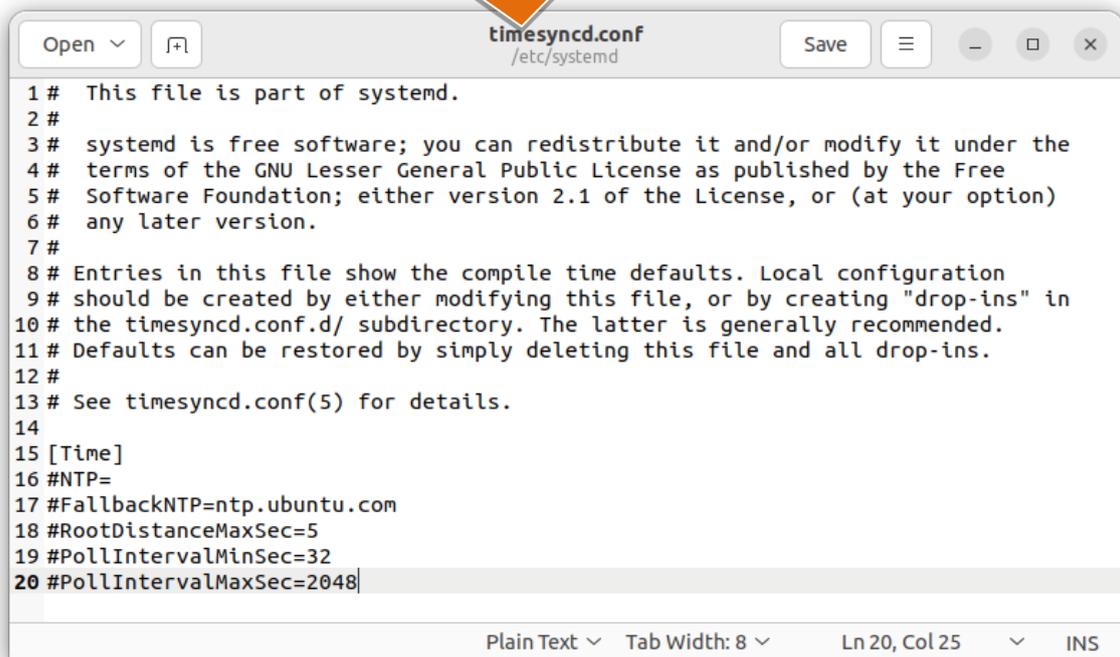
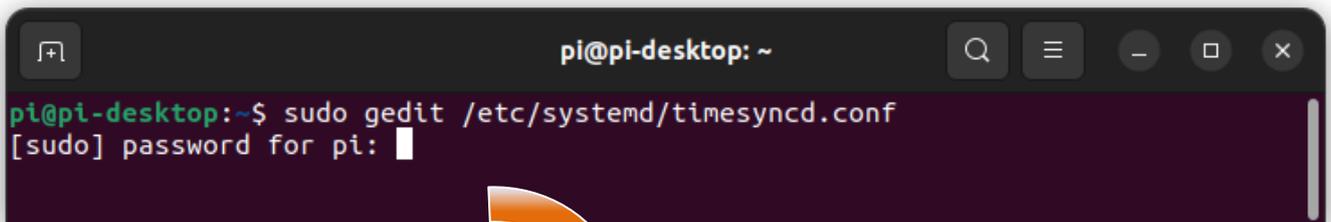
### 3.2.5. 自動網路校時

Ubuntu 已內建有時間校正功能，請依照下列步驟進行設定。

#### 步驟 1：點選 Terminal

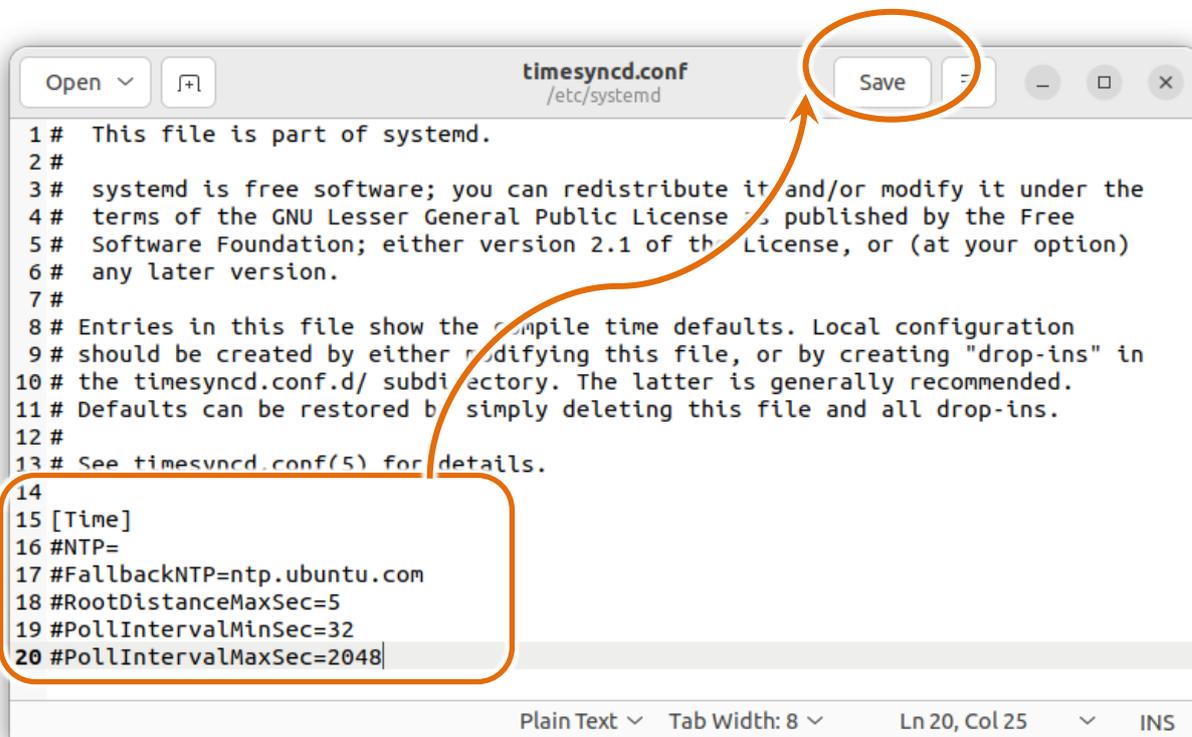


#### 步驟 2：輸入指令 `sudo gedit /etc/systemd/timesyncd.conf` 後，輸入密碼



步驟 3：設定 timesyncd.conf 檔中的 [time] 參數

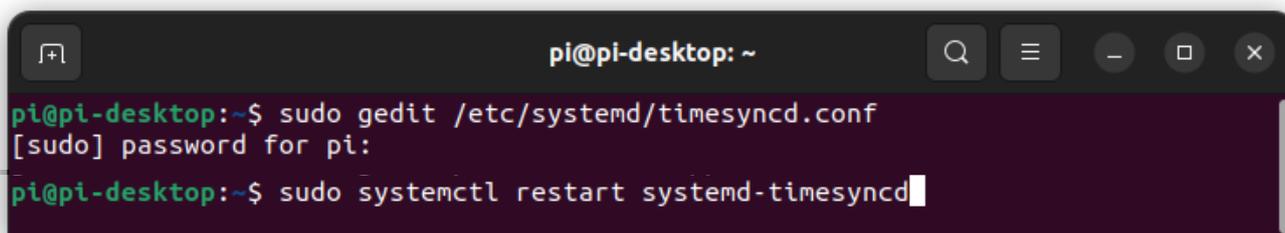
步驟 4：設定完成後，點選 Save 儲存設定



各參數設定如下表

欄位	說明
NTP	主要校時伺服器
FallbackNTP	備用校時伺服器
RootDistanceMaxSec	與伺服器的時間差距最多為多少(單位為秒)
PollIntervalMinSec	最少幾秒同步一次
PollIntervalMaxSec	最多幾秒同步一次

步驟 5：於 Terminal 中輸入 `sudo systemctl restart systemd-timesyncd`，將時間校正服務重新啟動載入新設定



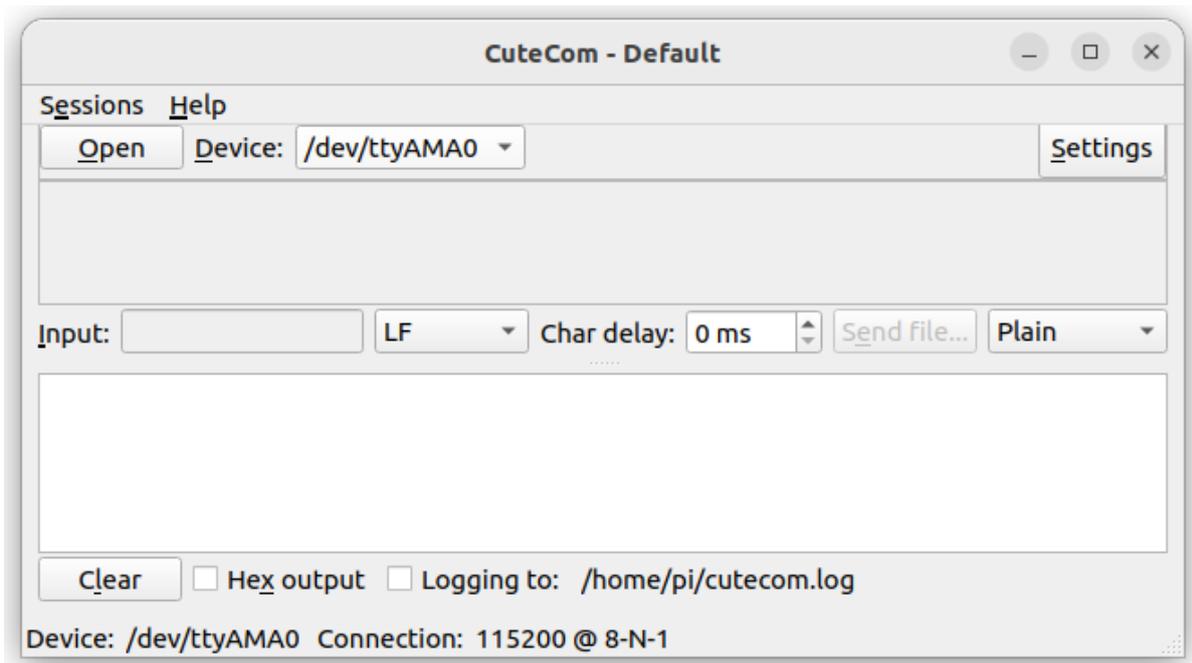
### 3.2.6. CuteCOM

CuteCom 是 Ubuntu 的圖形介面序列埠通訊軟體，可以傳送命令到外接模組，也可以從序列埠接收外接模組傳回的數據資料。

#### 點選 CuteCom



#### CuteCom 開啟後執行畫面如下



關於更多外接模組的通訊指令，請參閱：

[http://www.icpdas.com/root/product/solutions/remote\\_io/rs-485/i-8k\\_i-87k/i-8k\\_i-87k\\_selection.html#b](http://www.icpdas.com/root/product/solutions/remote_io/rs-485/i-8k_i-87k/i-8k_i-87k_selection.html#b)

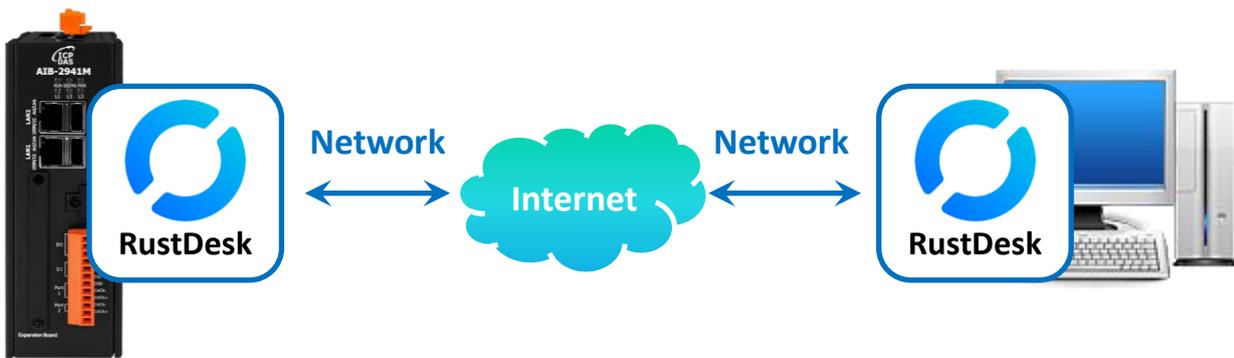
因開啟序列埠有權限問題，請在 Terminal 執行 “sudo CuteCom” 用 root 權限開啟 CuteCom，否則 open com port 時會發生錯誤。

### 3.3. 其他支援工具

#### 3.3.1. 遠端連線軟體

遠端桌面軟體有安全性的問題，AIB-2941M 是沒預先安裝的。故如果有遠端桌面工具的需求，RustDesk 經測試可正常運作。

RustDesk 和 AIB2941M 架構如下圖



步驟 1：於 PC 端進入 Rustdesk 官網

<https://rustdesk.com/>

步驟 2：於下載頁面中，點選 Ubuntu – Arm64 下載版本

1.3.0 Latest Compare

rustdesk released this Aug 19 · 85 commits to master since this release · 1.3.0 · 2a0fd55

**WARNING: YOU MAY BE BEING SCAMMED!**

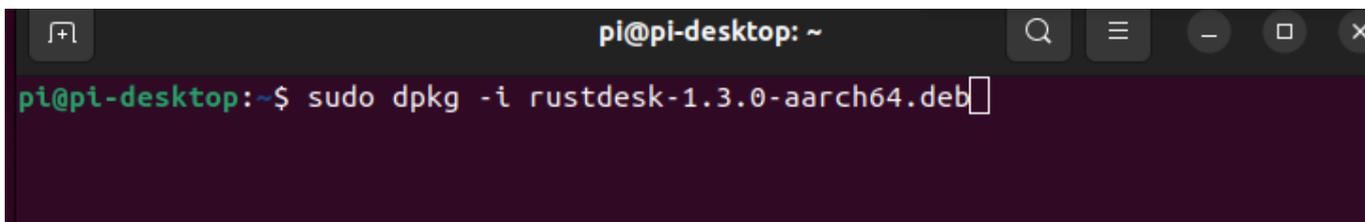
If you are on the phone with someone you DON'T know AND TRUST who has asked you to install RustDesk, do not install and hang up immediately.

They are likely a scammer trying to steal your money or other private information.

Architecture	Windows	Ubuntu	Mac	Android	Flatpak	AppImage	iOS
x86-64 (64-bit)	<a href="#">EXE</a> <a href="#">MSI</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Universal</a>	<a href="#">Download</a>	<a href="#">Download</a>	
AArch64 (ARM64)		<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">TestFlight</a>
ARMv7 (32-bit)		<a href="#">Download</a>		<a href="#">Download</a>			
x86-32 (32-bit)	<a href="#">Download</a>						

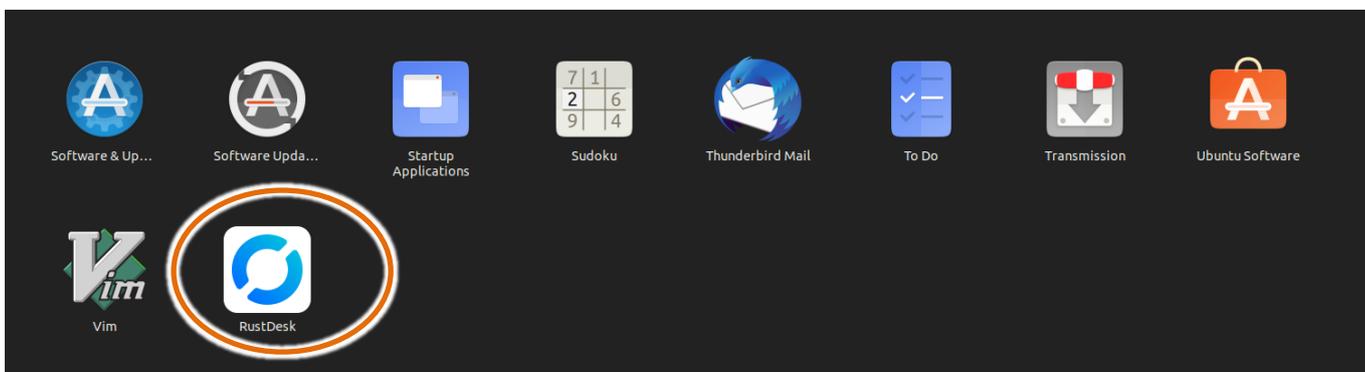
步驟 3：下載完成後，將下載的 rustdesk-1.3.0-aarch64.deb 檔案 (本例為 1.3.0 版)上傳至 AIB-2941M

步驟 4：在 AIB-2941M 啟動 Terminal，輸入 “sudo dpkg -i rustdesk-1.3.0-aarch64.deb” (本例為 1.3.0 版)，進行安裝

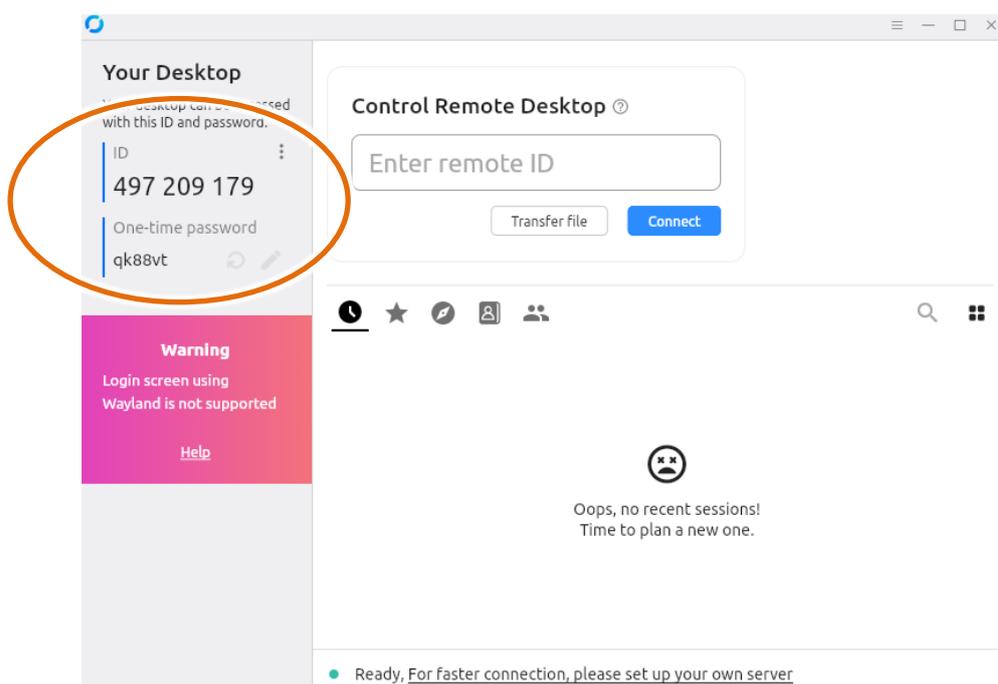


```
pi@pi-desktop: ~  
pi@pi-desktop:~$ sudo dpkg -i rustdesk-1.3.0-aarch64.deb
```

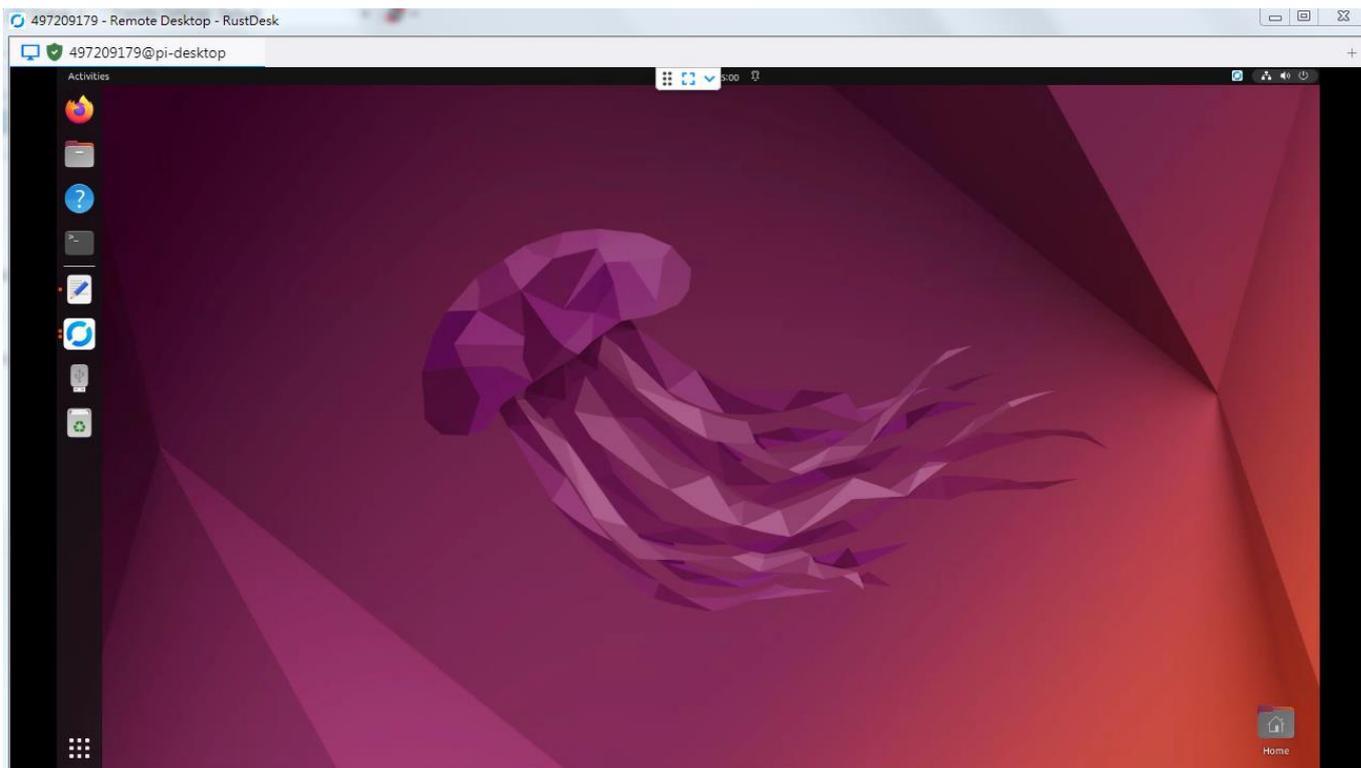
步驟 6：安裝完成後在程式列就能看到 RustDesk



步驟 7：RustDesk 操作方法和 TeamView 一樣，在 AIB-2941M 執行 RustDesk，將下圖中的 ID 和一次性密碼輸入 PC 端的 RustDesk 客戶端程式。



步驟 8：PC 端的 RustDesk 客戶端程式連線成功後會如下圖顯示 AIB-2941M 桌面



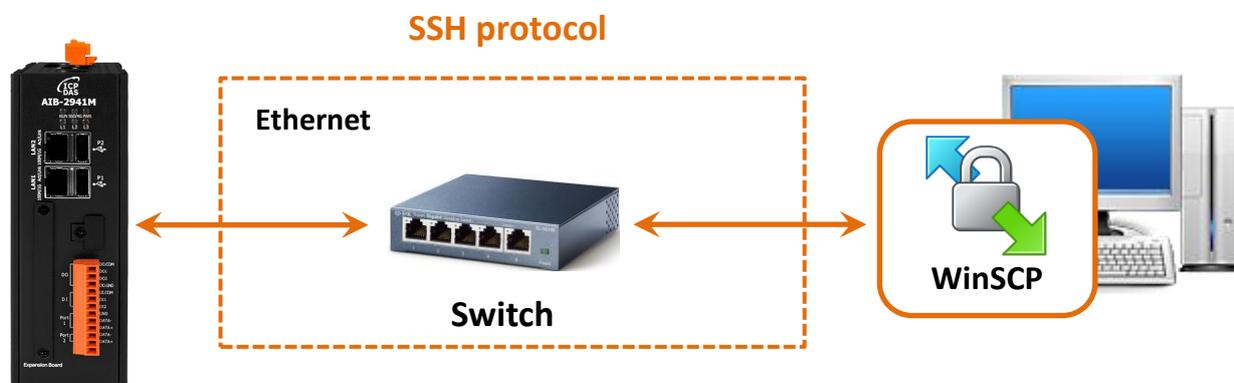
### 3.3.2. SSH 檔案傳輸

AIB-2941M 預設已開啟 ssh 連線，要遠端下載 AIB-2941M 的檔案，需要有支援 ssh 的軟體才可連線下載。本範例使用軟體 WinSCP 來示範如何遠端連線下載 AIB-2941M 的檔案。

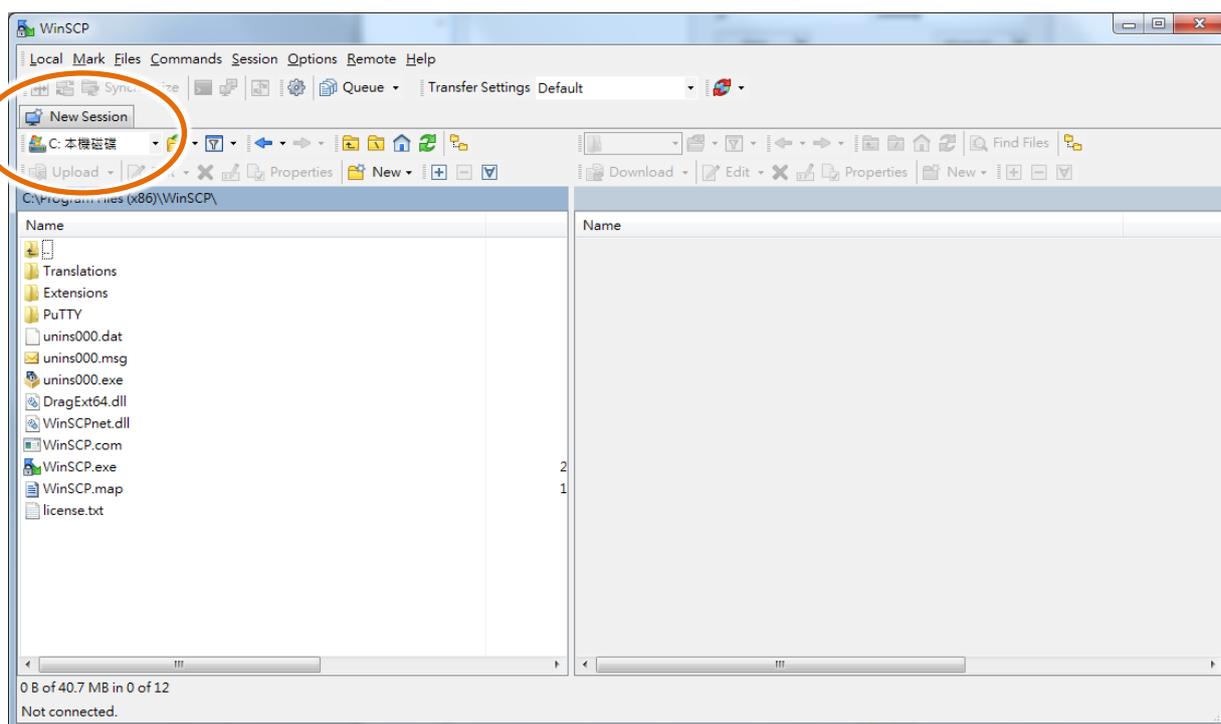
最新版的 WinSCP 可至下方路徑取得：

<https://winscp.net/eng/download.php>

WinSCP 和 AIB-2941M 架構如下圖



步驟 1：啟動 WinSCP，點選“New Session”



步驟 2：於 Login 對話視窗中，輸入下方資料後，點選 Login 按鈕

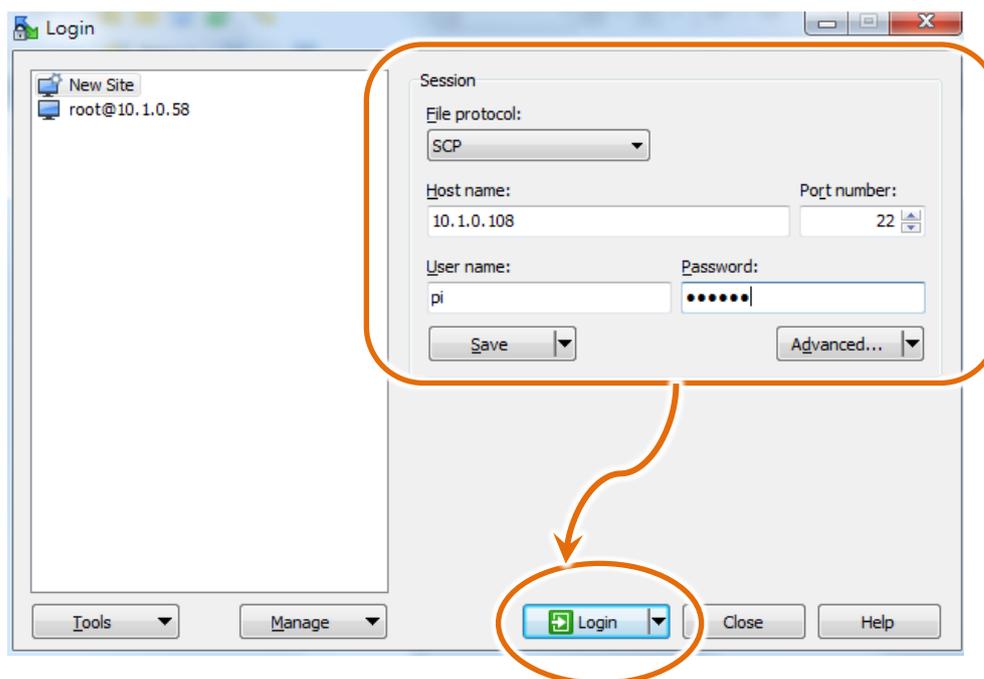
File protocol: SCP

Host name: 輸入 AIB-2941M 的 IP

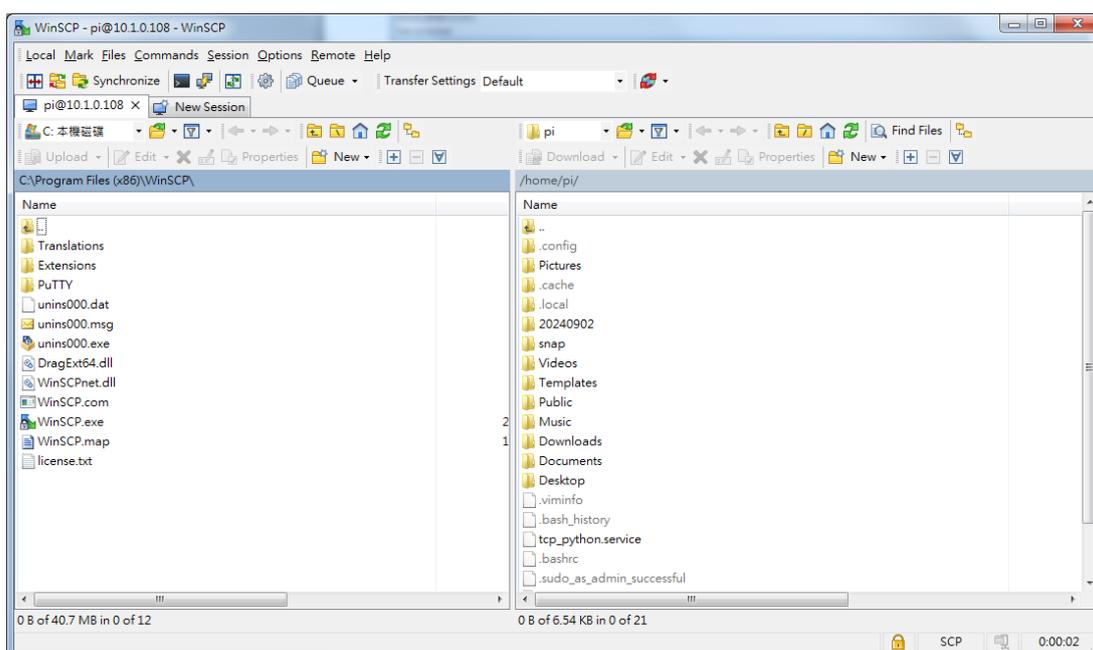
Port number: 22

User name: pi

Password: icpdas



步驟 3：連線成功後，即可進行檔案傳輸



## 4. AIB-2941M 程式開發

本章節提供的指引，包含說明如何建立一個開發環境的步驟、下載、安裝、配置用戶程式在 AIB-2941M 系列模組上。

### 開發工具

AIB-2941M 系列是以 Ubuntu 為基礎的成品，Ubuntu 是一個可以快速發展的成熟 Linux 操作系統。

下列表格中列出支援開發 AIB-2941M 應用程式的開發環境及開發語言。

開發語言 開發工具	Python	C/C++
Qt5	√	√

## 4.1. 開發工具準備

AIB-2941M 是以 Ubuntu 為基礎的成品，Ubuntu 是一個可以快速發展的成熟 Linux 操作系統。標準的開發工具是一個高度整合的工具，並全面支援以 Ubuntu 開發的 AIB-2941M 的應用程式。

### Qt 5



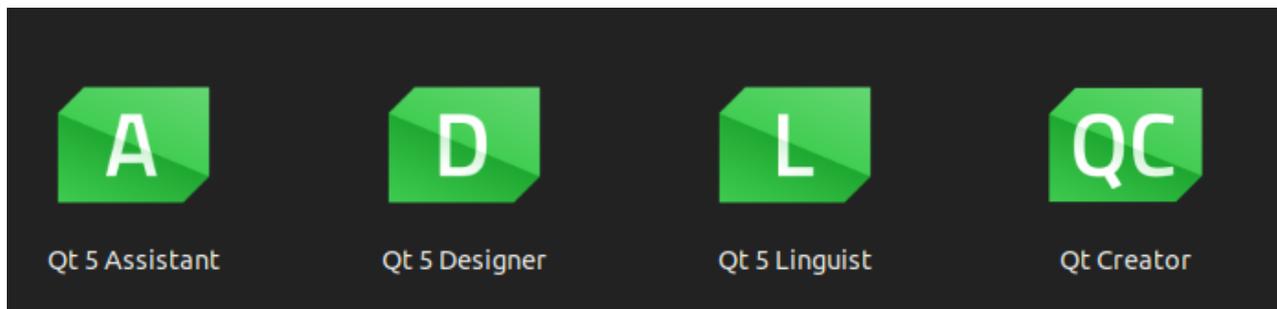
AIB-2941M 本身已安裝 Python 3，版本號為 3.10，Qt5 是 Ubuntu 上開發圖形介面的工具軟體，能充分利用 Python 的優點，透過公共互聯網標準整合新的和現有的應用程式使其能運行於任何平台。

### 安裝步驟

#### 1. Qt5

請在 Terminal 介面輸入 `sudo apt-get install qtcreator`

將會下載 Qt5 安裝軟體，Qt5 安裝完成後，會如下圖有 4 個新的 Qt 軟體



這四個軟體的功用如下

- Qt 5 Assistant：Qt5 說明手冊
- Qt 5 Designer：Qt5 圖形介面設定軟體
- Qt 5 Linguist：Qt 5 的翻譯工具
- Qt Creator：Qt 5 的主要開發工具

## 2. PyQt

請在 Terminal 介面輸入 `sudo apt-get install python3-pyqt5 qttools5-dev-tools qttools5-dev`

PyQt 是 Qt 的 python 圖形介面開發工具，安裝後就能在 Qt 開發 python 的圖形介面程式

## 3. Pyside

請在 Terminal 介面輸入 `sudo apt-get install python3-pyside2.qt3dcore python3-pyside2.qt3dinput python3-pyside2.qt3dlogic python3-pyside2.qt3drender python3-pyside2.qtcharts python3-pyside2.qtconcurrent python3-pyside2.qtcore python3-pyside2.qgui python3-pyside2.qthelp python3-pyside2.qtlocation python3-pyside2.qtmultimedia python3-pyside2.qtmultimediawidgets python3-pyside2.qtnetwork python3-pyside2.qtopengl python3-pyside2.qtpositioning python3-pyside2.qtprintsupport python3-pyside2.qtqml python3-pyside2.qtquick python3-pyside2.qtquickwidgets python3-pyside2.qtscript python3-pyside2.qtscripttools python3-pyside2.qtsensors python3-pyside2.qtsql python3-pyside2.qtsvg python3-pyside2.qtest python3-pyside2.qtexttospeech python3-pyside2.qtuio python3-pyside2.qtwebchannel python3-pyside2.qtwebsockets python3-pyside2.qtwidgets python3-pyside2.qtx11extras python3-pyside2.qtxml python3-pyside2.qtxmlpatterns`

PySide 是 Qt 的 python 圖形介面開發工具，安裝後就能在 Qt 開發 python 的圖形介面程式

## 4.2. 用 Python 建立第一個 AIB-2941M 應用程式

學習設計 AIB-2941M 上的程式的最好方法，就是直接實際建立一個 AIB-2941M 程式。

下面的例子示範如何利用 python 在 AIB-2941M 上建立一個展示程式。

要建立一個 python 的程式需要以下主要步驟：

1. 建立一個新專案
2. 增加控制選項到視窗中
3. 增加控制選項的事件處理
4. 執行編譯好的程式

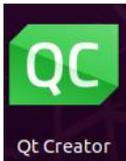
所有主要步驟都會在子章節詳細介紹。

在教程中，我們會假定您已經在 AIB-2941M 上安裝 Qt 5、PyQt 和 Pyside。

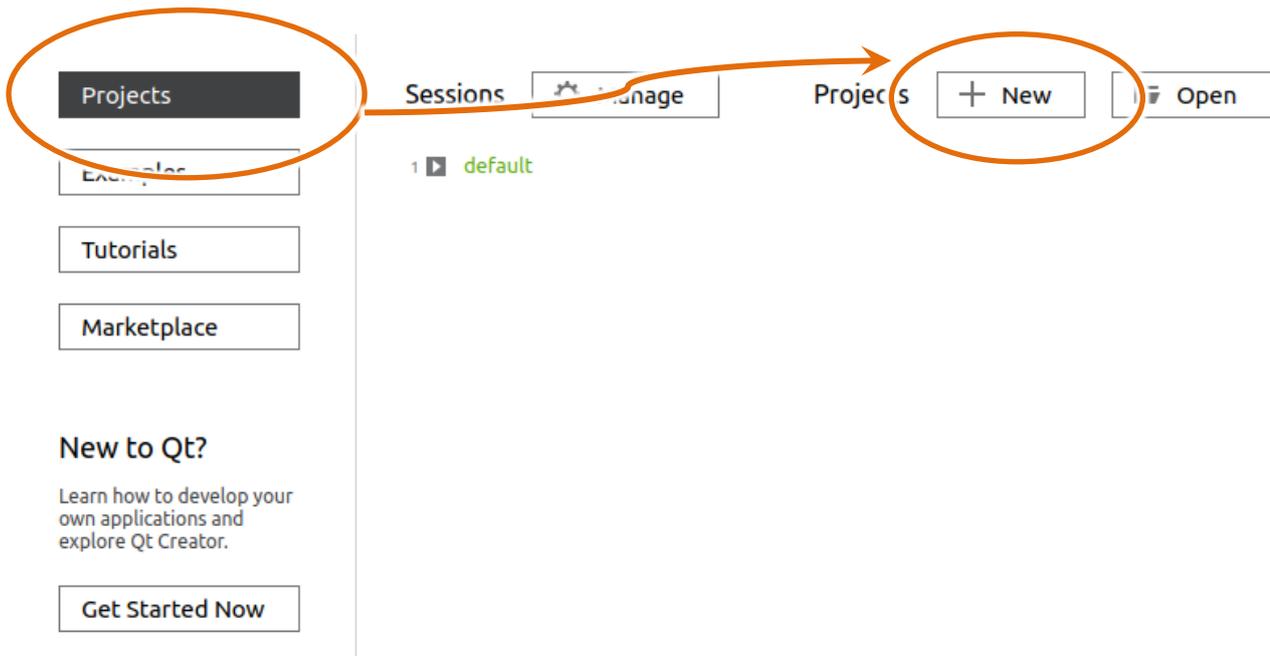
### 4.2.1. 建立一個新專案

此範例，用來創建一個，新的複合控制項專案，與使用者控制項範例程式。

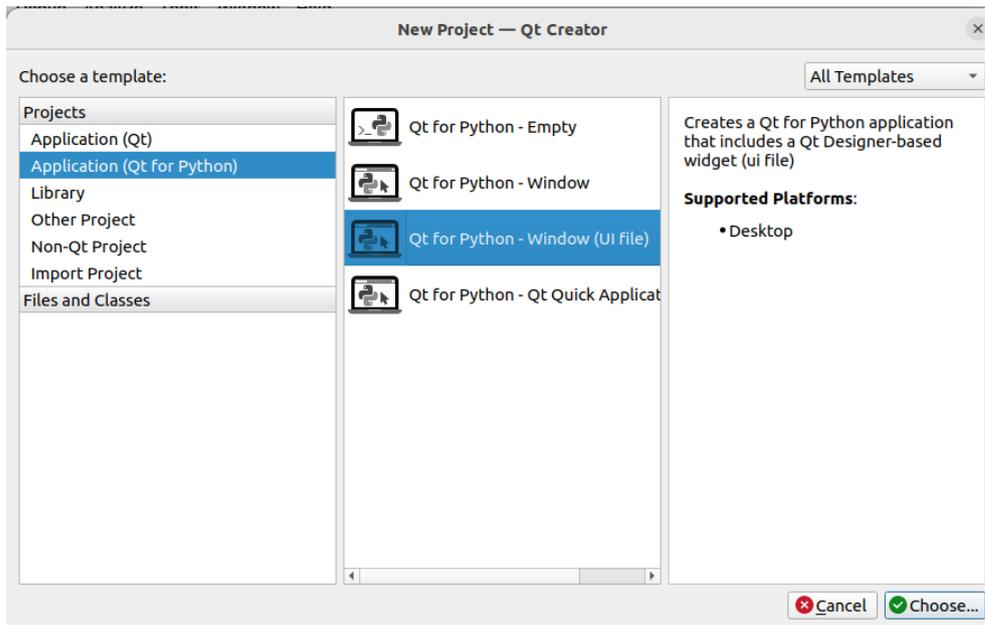
#### 步驟 1：執行 Qt Creator



#### 步驟 2：在 Project 項目點選 NEW

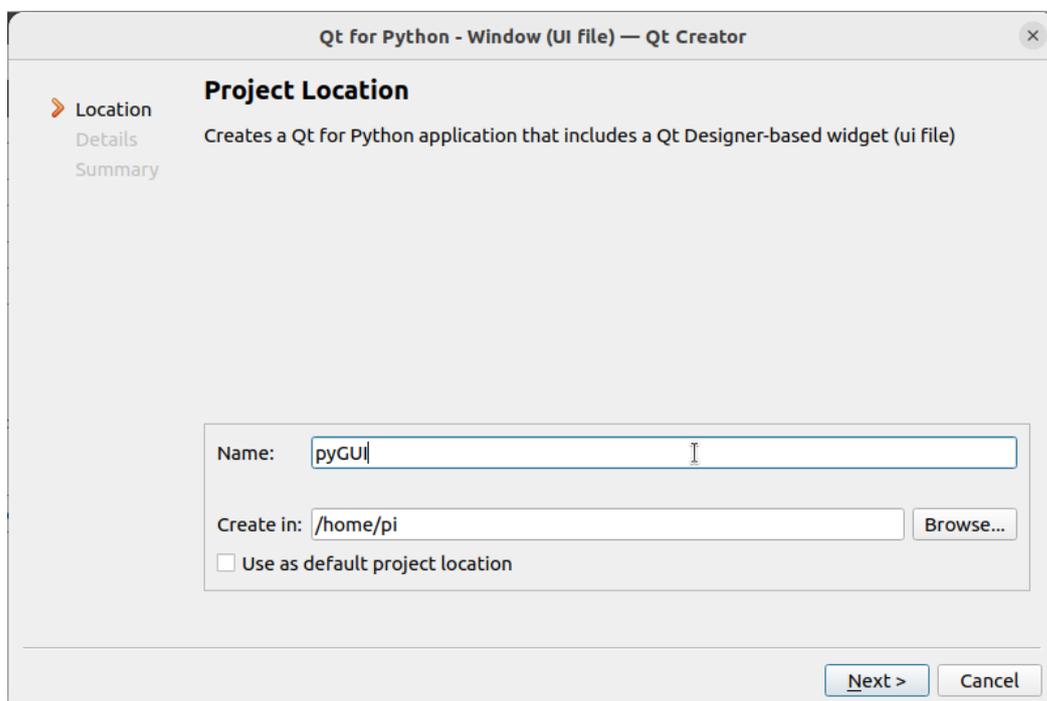


步驟 3：在 Project 中下選擇 “Application (Qt for Python)” 選項，然後點擊 Qt for Python – windows (UI file)

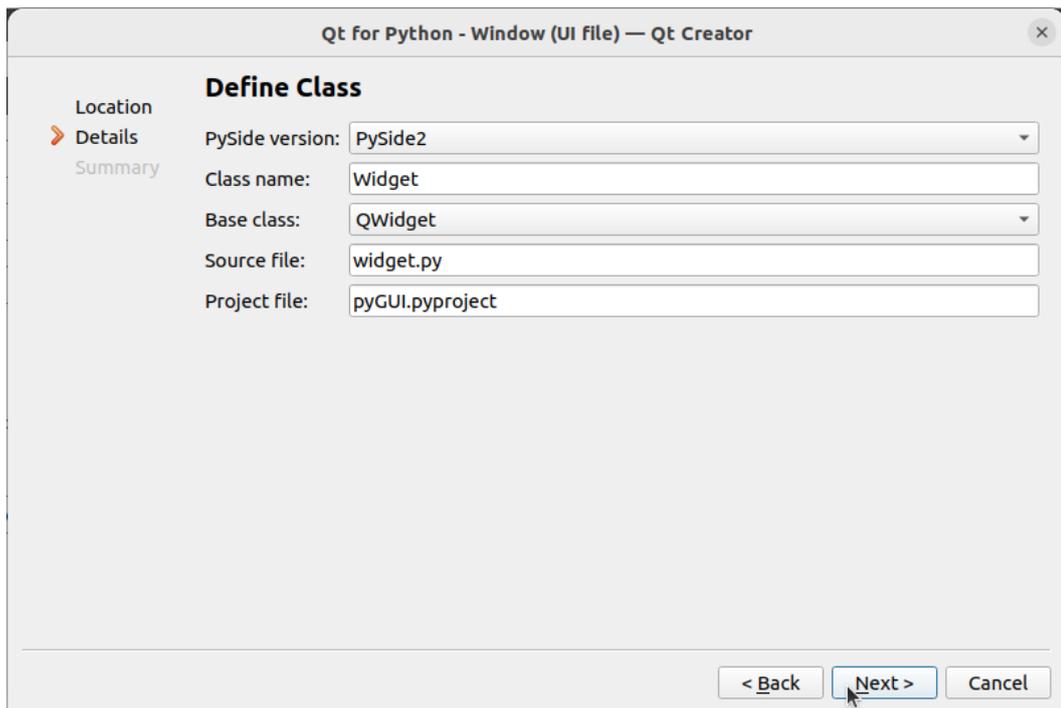


步驟 4：在名稱欄填入名稱，點擊 OK 按鈕

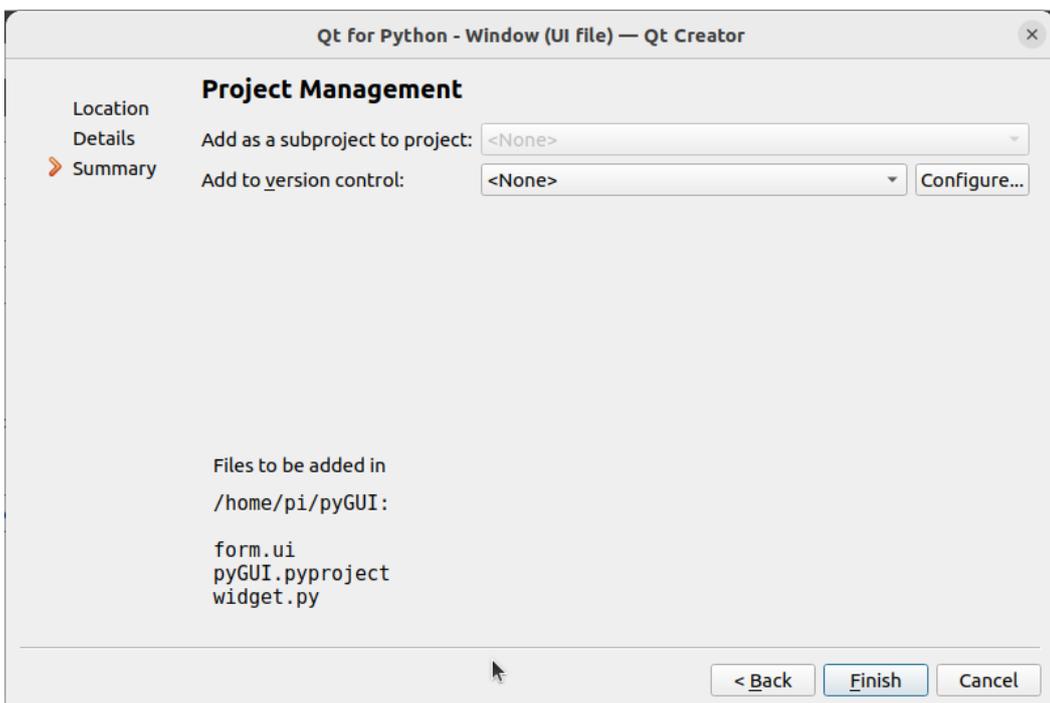
在此我們輸入 pyGUI，你也可以選擇按下“Browse”按鈕選擇不同的 Project 放置位置



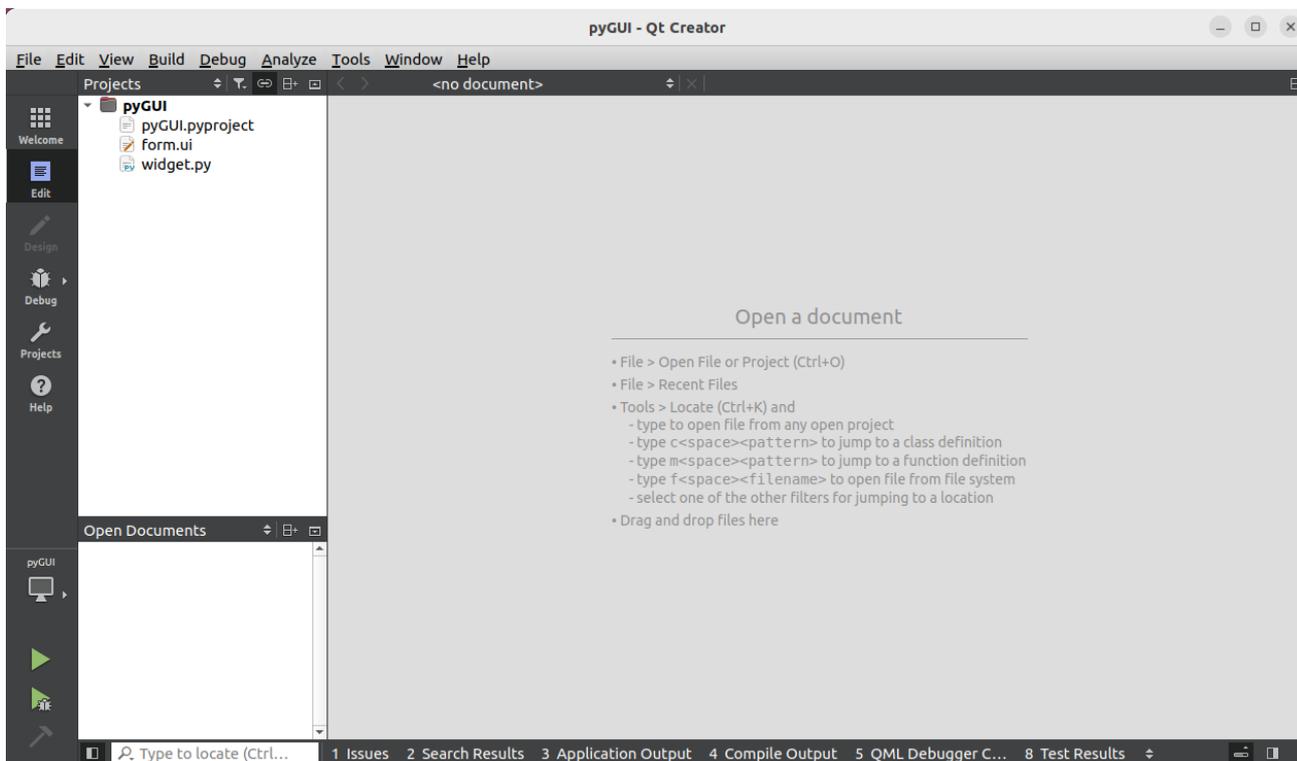
步驟 5：在 PySide version 欄位，選擇 PySide 版本，此例選擇 PySide2



步驟 6：在 Add to version control 欄位，如果你有需要進行版本管理，預設可選擇 git，此例選擇 “None”，按下 “Finish”完成建立 Project



步驟 7：建立好的 Project 會有三個檔案，form.ui、widget.py 和 <project name>.pyproject，顯示如下



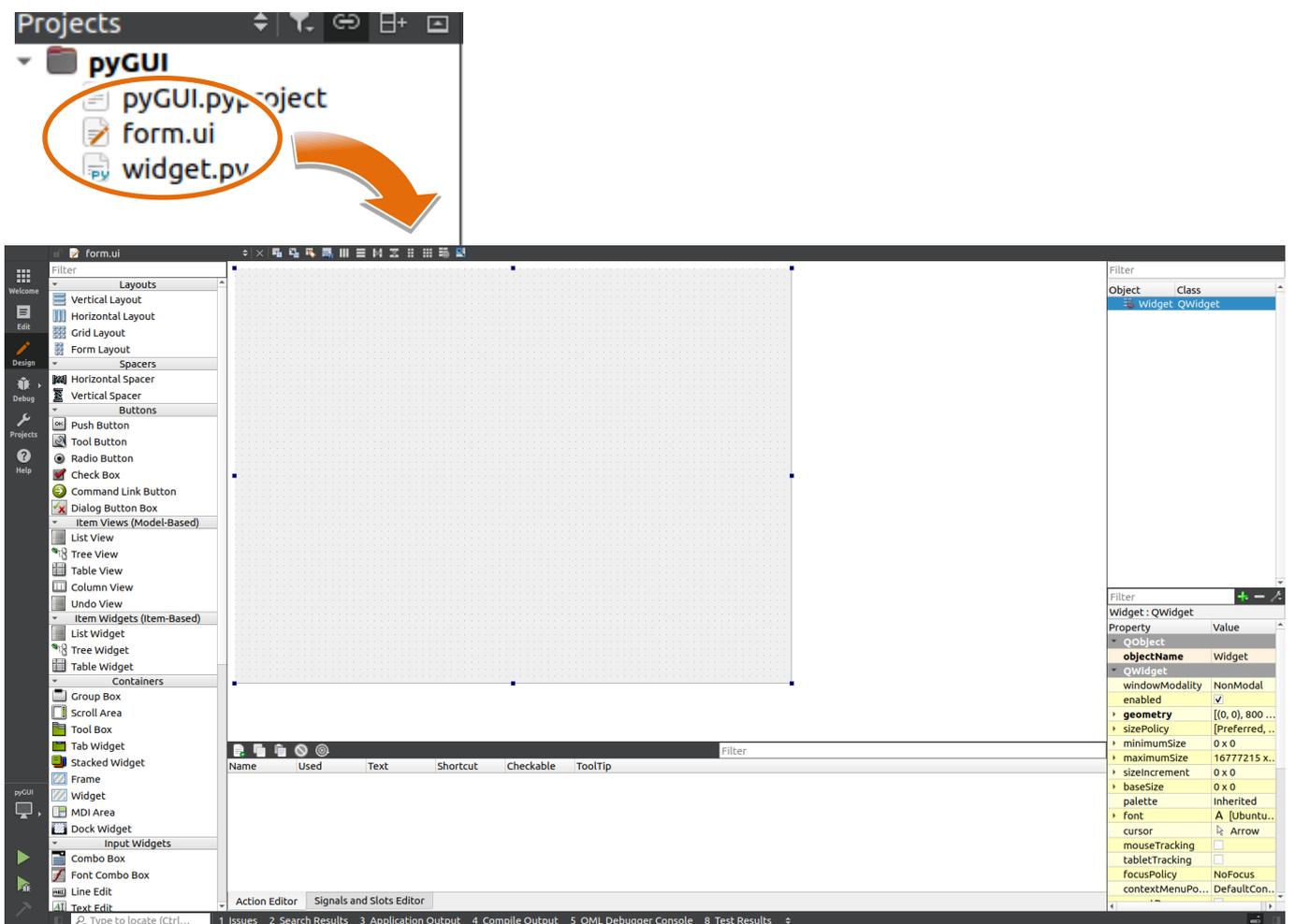
## 4.2.2. 在視窗上加入控制選項

您可以在表單中隨意新增 Toolbox 中的控制選項，並在表單中用拖曳的方式調整想要的位置。

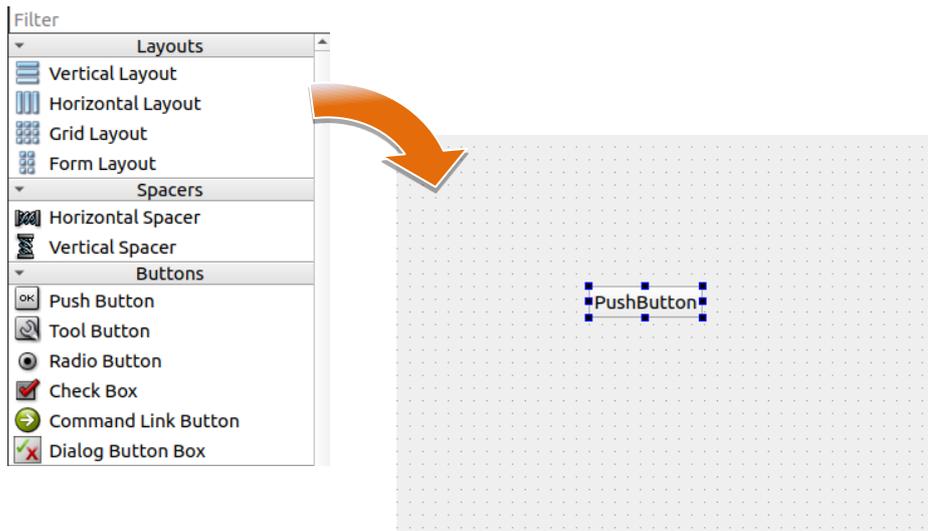
增加控制項目到視窗後，可以從 Properties 視窗調整對應屬性，例如背景顏色或預設顯示文字。

這些在屬性視窗設定的屬性，將在程式運行時以初始化值的形式設定在對應欄位上。

### 步驟 1：開啟 Project 表單頁面，開啟 form.ui 檔案



步驟 2：在 Toolbox 中，拖曳 Push Button 控制元件到視窗上



步驟 3：在屬性視窗中，在 ObjectName 屬性欄輸入 btnClick，在 Text 屬性欄輸入 Show Message

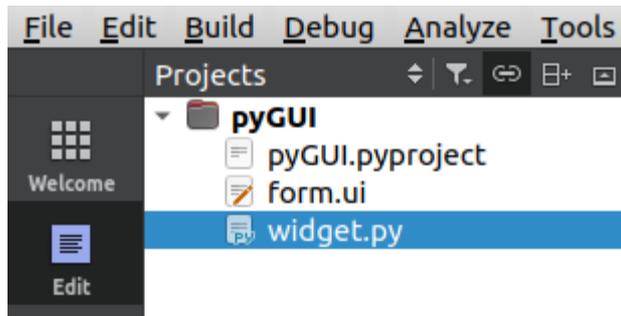
Property	Value
QObject	
objectName	btnClick
QWidget	
enabled	<input checked="" type="checkbox"/>
geometry	[(150, 120), ...]
sizePolicy	[Minimum, ...]
minimumSize	0 x 0
maximumSize	16777215 x..
sizeIncrement	0 x 0
baseSize	0 x 0
palette	Inherited
font	A [Ubuntu..
cursor	Arrow
mouseTracking	<input type="checkbox"/>
tabletTracking	<input type="checkbox"/>
focusPolicy	StrongFocus
contextMenuPo...	DefaultCon..
acceptDrops	<input type="checkbox"/>

QAbstractButton	
text	Show Message
icon	
iconSize	16 x 16
shortcut	
checkable	<input type="checkbox"/>
checked	<input type="checkbox"/>
autoRepeat	<input type="checkbox"/>
autoExclusive	<input type="checkbox"/>
autoRepeatDelay	300
autoRepeatInte...	100

### 4.2.3. 在控制項目中加入事件

您已經完成應用程式視窗的設計，可以開始為應用程式撰寫功能程式碼。

#### 步驟 1：選擇 Edit mode，打開 widget.py



會顯示如下圖的程式碼

```
< > widget.py
1 # This Python file uses the following encoding: utf-8
2 import os
3 from pathlib import Path
4 import sys
5
6 from PySide2.QtWidgets import QApplication, QWidget
7 from PySide2.QtCore import QFile
8 from PySide2.QtUiTools import QUiLoader
9
10
11 class Widget(QWidget):
12     def __init__(self):
13         super(Widget, self).__init__()
14         self.load_ui()
15
16     def load_ui(self):
17         loader = QUiLoader()
18         path = os.fspath(Path(__file__).resolve().parent / "form.ui")
19         ui_file = QFile(path)
20         ui_file.open(QFile.ReadOnly)
21         loader.load(ui_file, self)
22         ui_file.close()
23
24
25 if __name__ == "__main__":
26     app = QApplication([])
27     widget = Widget()
28     widget.show()
29     sys.exit(app.exec_())
30
```

## 步驟 2：在 widget.py 加入下列程式碼

```
from PyQt5 import QtWidgets

def __init__(self):
    super(Widget, self).__init__()
    self.load_ui()
    self.load_event()

def load_event(self):
    self.loader.btnClick.clicked.connect(self.Message)

def Message(self):
    Form = QtWidgets.QWidget()
    mbox = QtWidgets.QMessageBox(Form)
    mbox.information(Form, 'info', 'first program')
```

完成的程式碼如下圖



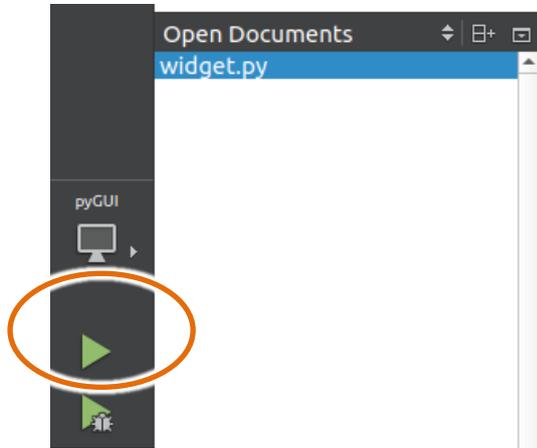
```
< > widget.py*
1 # This Python file uses the following encoding: utf-8
2 import os
3 from pathlib import Path
4 import sys
5
6 from PyQt5 import QtWidgets
7 from PySide2.QtWidgets import QApplication, QWidget
8 from PySide2.QtCore import QFile
9 from PySide2.QtUiTools import QUiLoader
10
11
12 class Widget(QWidget):
13     def __init__(self):
14         super(Widget, self).__init__()
15         self.load_ui()
16         self.load_event()
17
18     def load_ui(self):
19         loader = QUiLoader()
20         path = os.fspath(Path(__file__).resolve().parent / "form.ui")
21         ui_file = QFile(path)
22         ui_file.open(QFile.ReadOnly)
23         self.window = loader.load(ui_file, self)
24         ui_file.close()
25
26     def load_event(self):
27         self.window.btnOSVer.clicked.connect(self.Message)
28
29     def Message(self):
30         Form = QtWidgets.QWidget()
31         mbox = QtWidgets.QMessageBox(Form)
32         mbox.information(Form, 'info', 'first program')
33
34
35
36
37 if __name__ == "__main__":
38     app = QApplication([])
39     widget = Widget()
40     widget.show()
41     sys.exit(app.exec_())
42
```

#### 4.2.4. 執行編譯好的應用程式

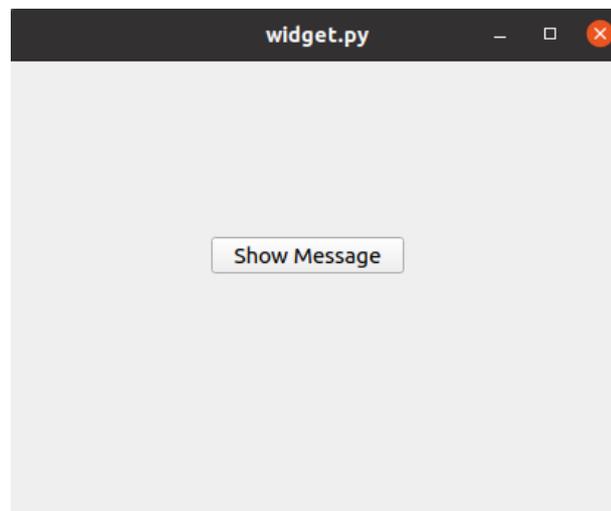
在 Qt 右下角，點擊 `widget.py`，表示接下來要執行 `widget.py`，點擊



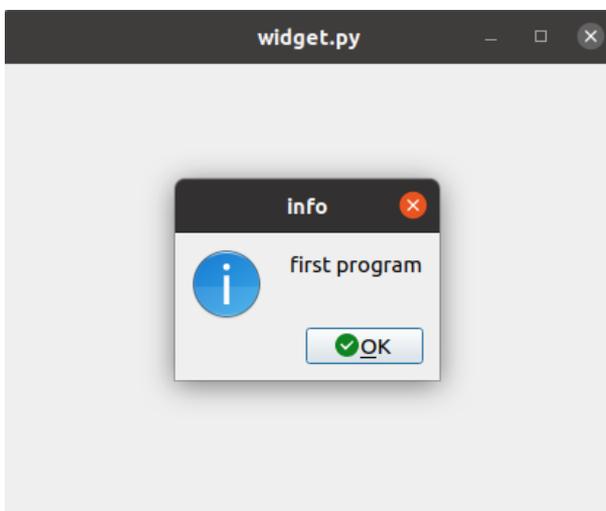
圖示開始執行程式



程式運作如右圖



按下 “Show Message” 按鈕將會跳出如下 MessageBox 訊息



## 4.2.5. Python 範例

在 Python 語言環境中使用的範例。可由 ICP DAS 網站下載最新的版本。

可從下列的位址獲得 Python 的範例程式：

<https://www.icpdas.com/tw/download/index.php?model=AIB-2941M>

範例程式有以下四個

di.py：讀取 AIB-2941M 上的 DI Port 狀態

do.py：對 AIB-2941M 上的 DO Port 輸出 ON/OFF

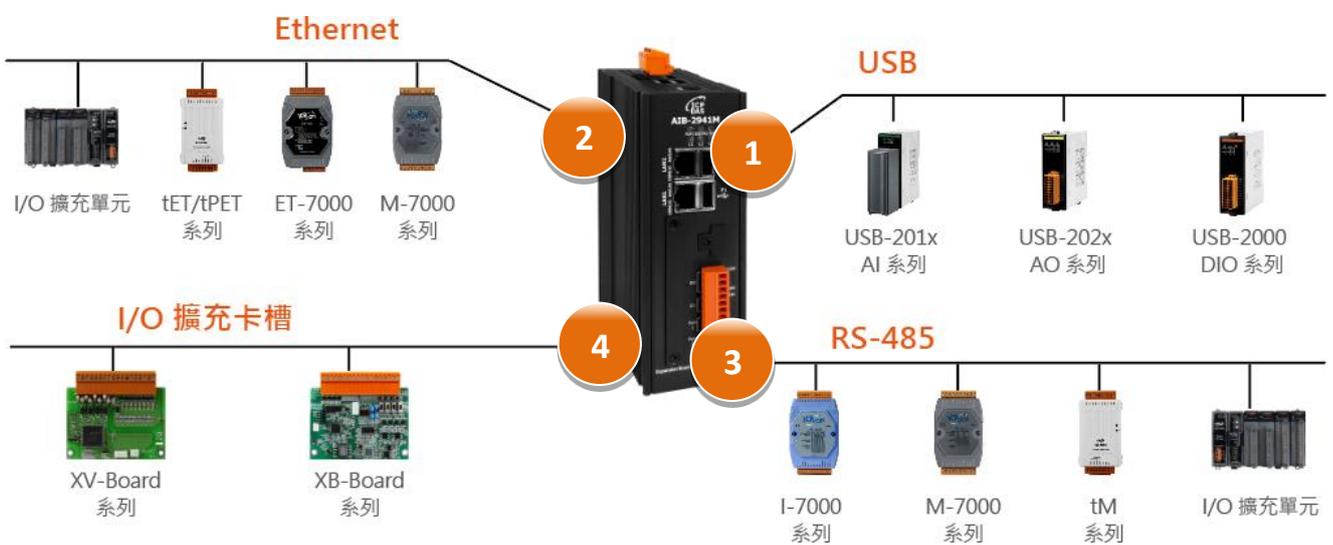
led.py：對 AIB-2941M 上的 LED1~3 輸出 ON/OFF

ser.py：對 AIB-2941M 上的 Port1~2 進行讀寫

## 5. I/O 擴充模組與 SDK 的選擇

本章將介紹如何選擇一個合適的 I/O 擴充模組及其對應的 SDK，讓使用者能夠更靈活的使用 AIB-2941M，更全面的開發 AIB-2941M 程式。

AIB-2941M 提供以下 I/O 擴充的配置



### ① USB (USB-2000 系列及 USB-87Pn 系列)

AIB-2941M 可透過 USB 來連接 USB-2000 和 USB-87Pn 搭配高卡 I-87k 系列 I/O 模組。

模組	原生 SDK	Python 範例
USB-2000 系列	Python 範例	Python 範例
USB-87Pn 系列	Python 範例	Python 範例

## ② Ethernet (ET-7000 系列及 I-8KE4/8-MTCP)

AIB-2941M 可透過 Ethernet 來連接 ET-7000 和 I-8KE4/8-MTCP 搭配高卡 I-87k 系列 I/O 模組，並支援 Modbus/TCP 通訊協議。

模組	原生 SDK	Python 範例
ET-7000	Modbus 範例	Modbus 範例
I-8KE4/8-MTCP	Modbus 範例	Modbus 範例

## ③ RS-485 (I-7000 系列及 M-7000 系列)

AIB-2941M 可透過 RS-485 來連接 I-7000、M-7000 和 RU-87Pn 搭配高卡 I-87k 系列 I/O 模組。RS-485 網路採用雙絞作為傳輸介質，傳輸速率最高可達 115.2 Kbps。

### ➤ I-7000系列I/O模組

模組	原生 SDK	Python 範例
I-7000 系列	libAIB.so	libAIB.py
I-7000 系列之 I-7088 (D)	libAIB.so	libAIB.py

更多有關 I-7000 系列 I/O 模組與範例說明，請參閱：

[https://www.icpdas.com/tw/product/guide+Remote I O Module and Unit+RS-485 I O Modules+I-7000](https://www.icpdas.com/tw/product/guide+Remote+I+O+Module+and+Unit+RS-485+I+O+Modules+I-7000)

### ➤ M-7000系列I/O模組

模組	原生 SDK	Python 範例
M-7000 系列	Modbus 範例	Modbus 範例

更多有關 M-7000 系列 I/O 模組與範例說明，請參閱：

[https://www.icpdas.com/tw/product/guide+Remote I O Module and Unit+RS-485 I O Modules+I-7000](https://www.icpdas.com/tw/product/guide+Remote+I+O+Module+and+Unit+RS-485+I+O+Modules+I-7000)

➤ **RU-87Pn + I-87K系列I/O模組**

模組	原生 SDK	Python 範例
RU-87Pn + I-87K	libAIB.so	libAIB.py

➤ **其他指定 I/O**

模組	原生 SDK	Python 範例
I-7088W	libAIB.so	libAIB.py

**4** **本機 I/O 擴充模組 (XV-Board)**

AIB-2941M 可支援－ XV-Board/XB-Board。

➤ **XV-Board**

模組	原生 SDK	Python 範例
XV-Board	Modbus Demo	Modbus Demo

更多的支援模組訊息，請參閱：

[https://www.icpdas.com/tw/product/guide+Remote I O Module and Unit+PAC I O Modules+XV-board](https://www.icpdas.com/tw/product/guide+Remote+I+O+Module+and+Unit+PAC+I+O+Modules+XV-board)

➤ **XB-Board**

模組	原生 SDK	Python 範例
XB-Board	Modbus Demo	Modbus Demo

更多的支援模組訊息，請參閱：

[https://www.icpdas.com/tw/product/guide+Remote I O Module and Unit+PAC I O Modules+XB-board](https://www.icpdas.com/tw/product/guide+Remote+I+O+Module+and+Unit+PAC+I+O+Modules+XB-board)

## 6. AIB-2941M 系統更新與升級

本章節提供了如何更新 AIB-2941M OS 所需要的步驟。

ICP DAS 將繼續在未來添加額外的功能於 AIB-2941M OS，因此我們建議您定期檢查 ICP DAS 網站的最新資料。

AIB-2941M 是透過 Raspberry Pi Imager 進行燒錄來更新 OS。Raspberry Pi Imager 須要在 PC 上進行安裝，更新 OS 前，請先在 PC 安裝 Raspberry Pi Imager。

最新版的 OS 更新安裝檔可至 ICP DAS 的網站下載。

<https://www.icpdas.com/tw/download/index.php?model=AIB-2941M>

### 步驟 1：安裝 Raspberry Pi Imager & rpiboot

最新版本的 Raspberry Pi Imager 可至下列連結下載

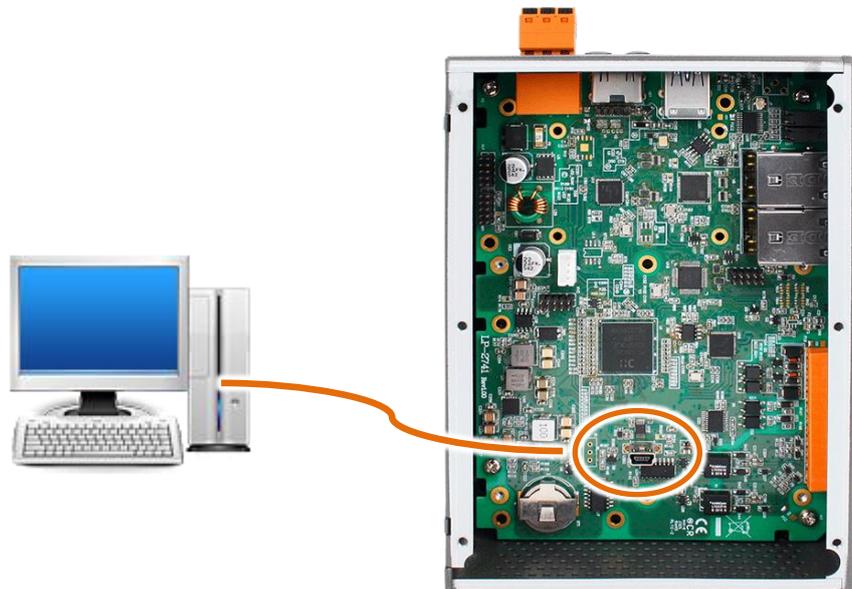
<https://www.raspberrypi.com/software/>

最新版本的 rpiboot 可至下列連結下載

[https://github.com/raspberrypi/usbboot/raw/master/win32/rpiboot\\_setup.exe](https://github.com/raspberrypi/usbboot/raw/master/win32/rpiboot_setup.exe)

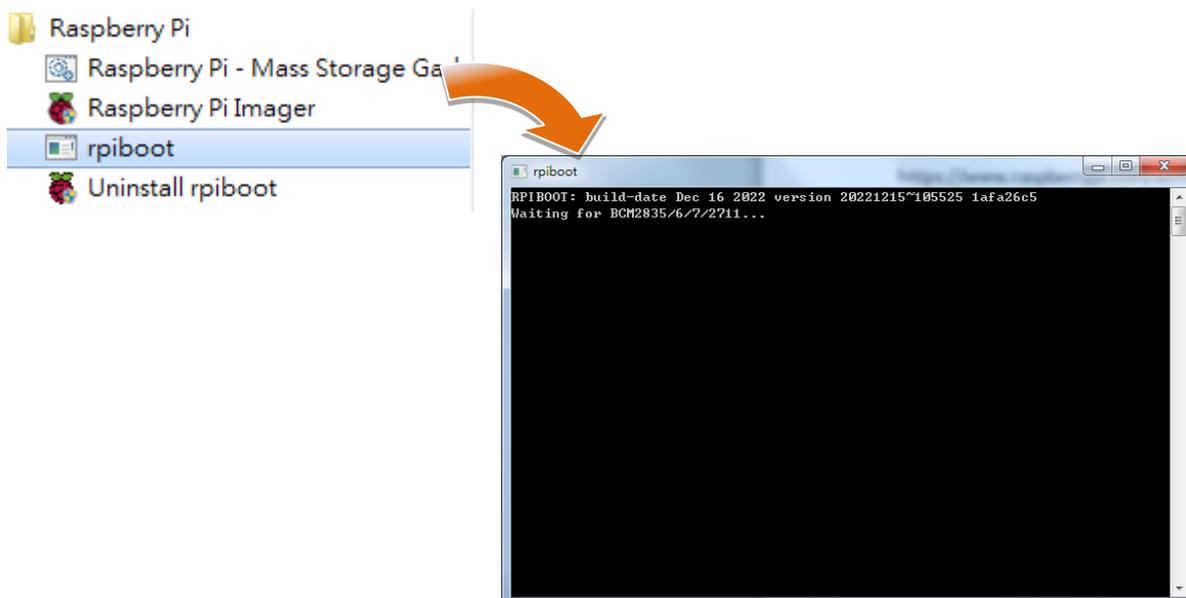
如果已安裝 Raspberry Pi Imager & rpiboot 可跳過此步驟。

## 步驟 2：將 AIB-2941M 的 mini USB 連接到 PC



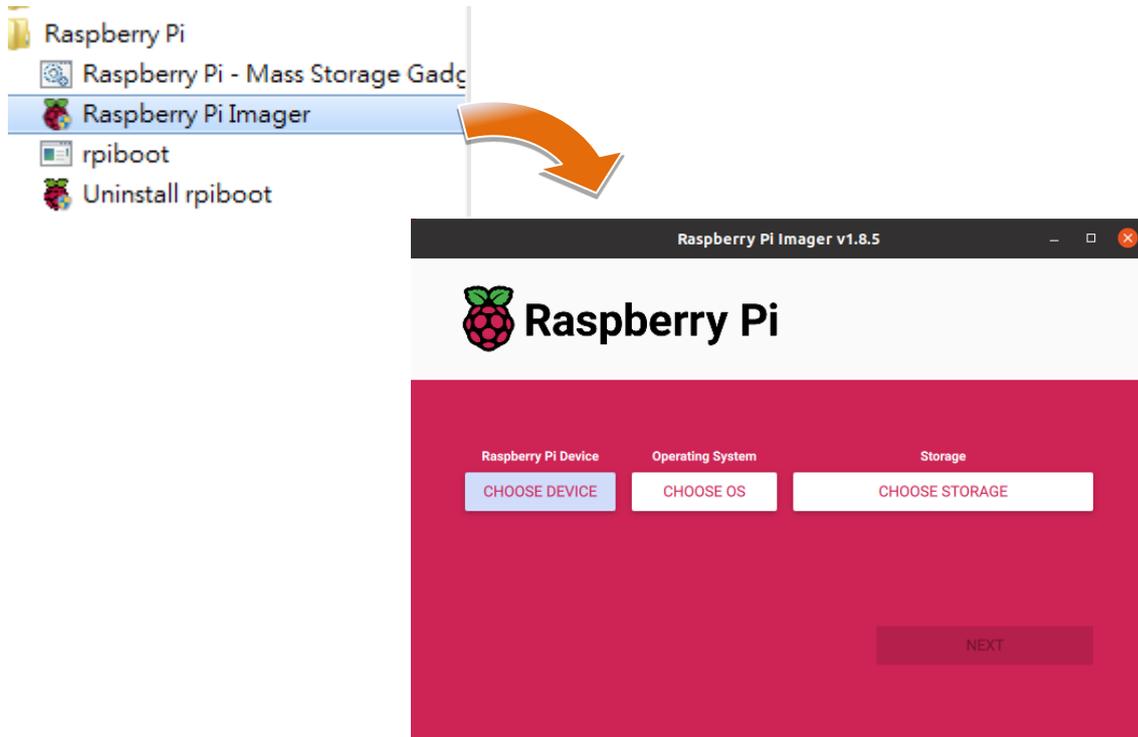
## 步驟 3：啟動 PC 的 rpiboot

等待 rpiboot 將 eMMC 掛載，掛載完成後啟動 Raspberry Pi Imager 進行燒錄 OS

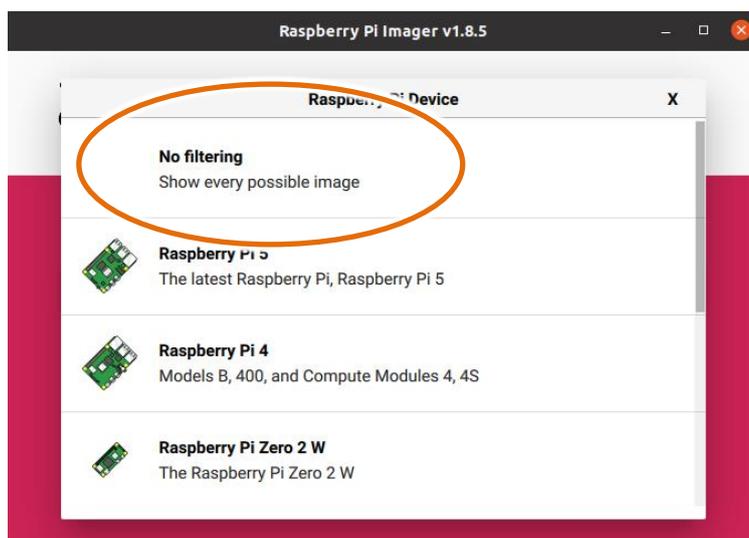


## 步驟 4：啟動 Raspberry Pi Imager

Raspberry Pi Imager 有三個項目需要進行設定，請依下列步驟進行



## 步驟 5：Raspberry Pi Device 項目選擇 “No Filtering”

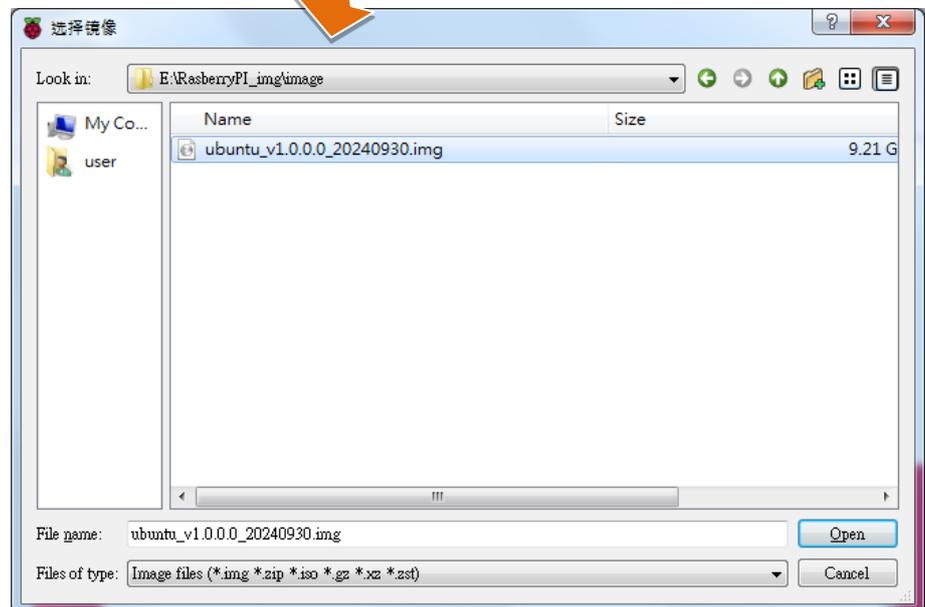
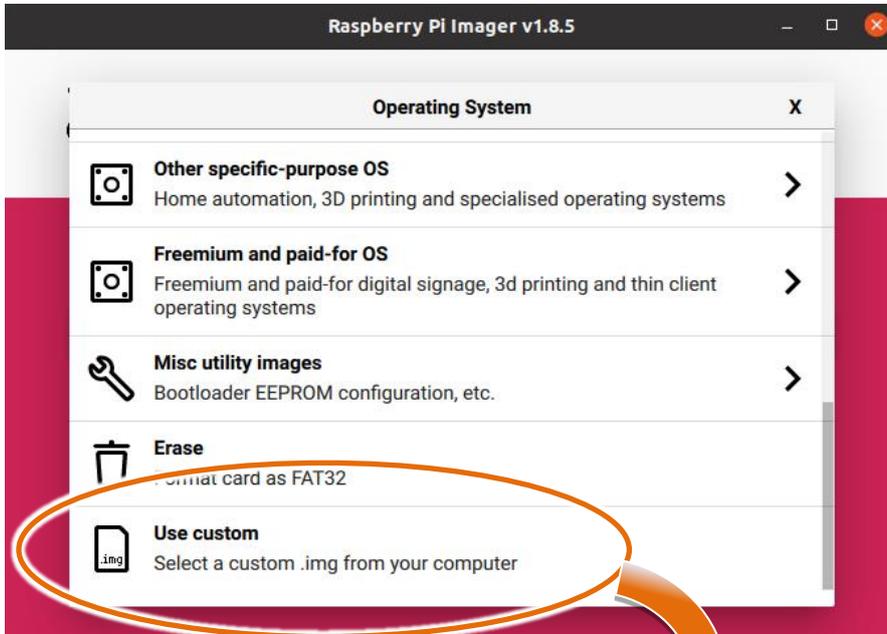


## 步驟 6：Operating system 選擇 “Use custom”

選擇從

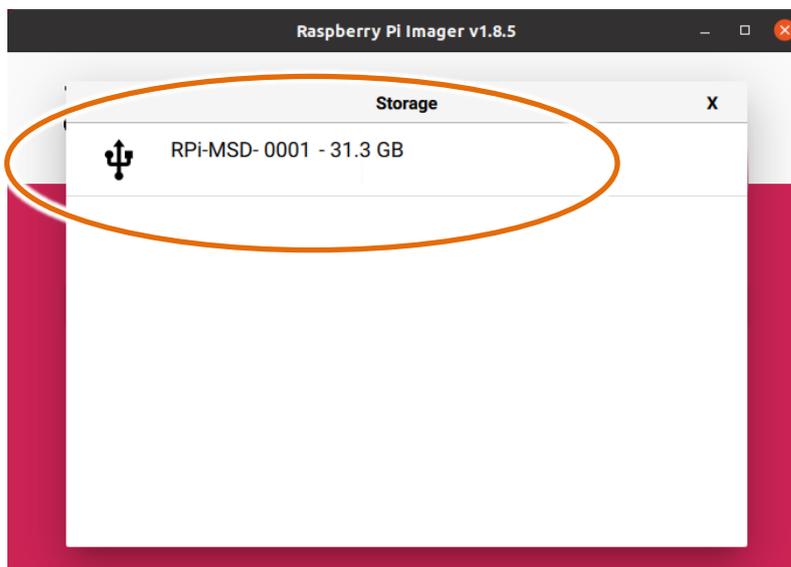
<https://www.icpdas.com/tw/download/show.php?num=5555&model=AIB-2941M> 下

載的 OS image，本例為初版 OS image “ubuntu\_v1.0.0.0\_20240930.img”

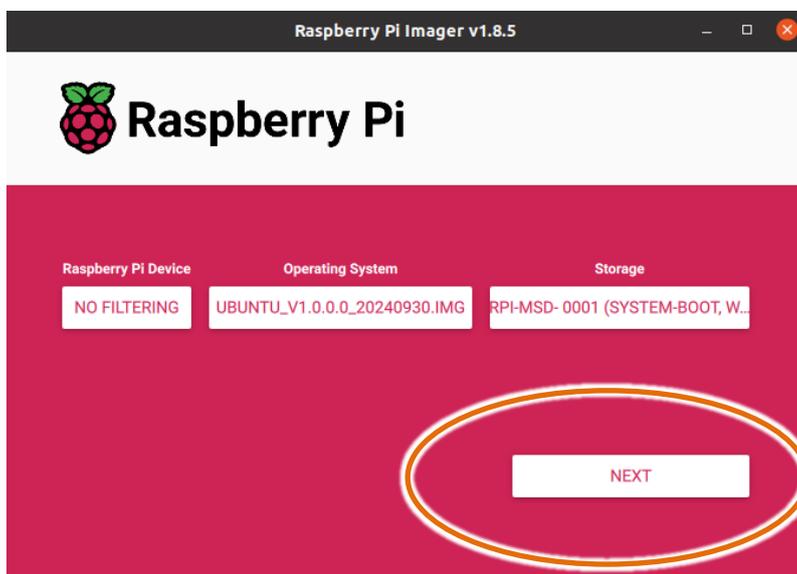


### 步驟 7：Storage 選擇 “RPI-MSD-0001”

“RPI-MSD-0001”是 rpiboot 掛載 eMMC 的名稱，

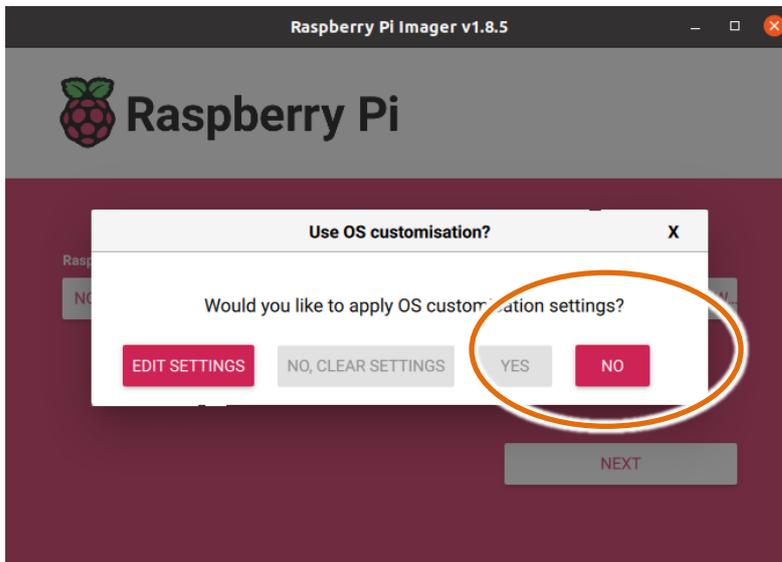


### 步驟 8：按下 “Next”開始進行下一步

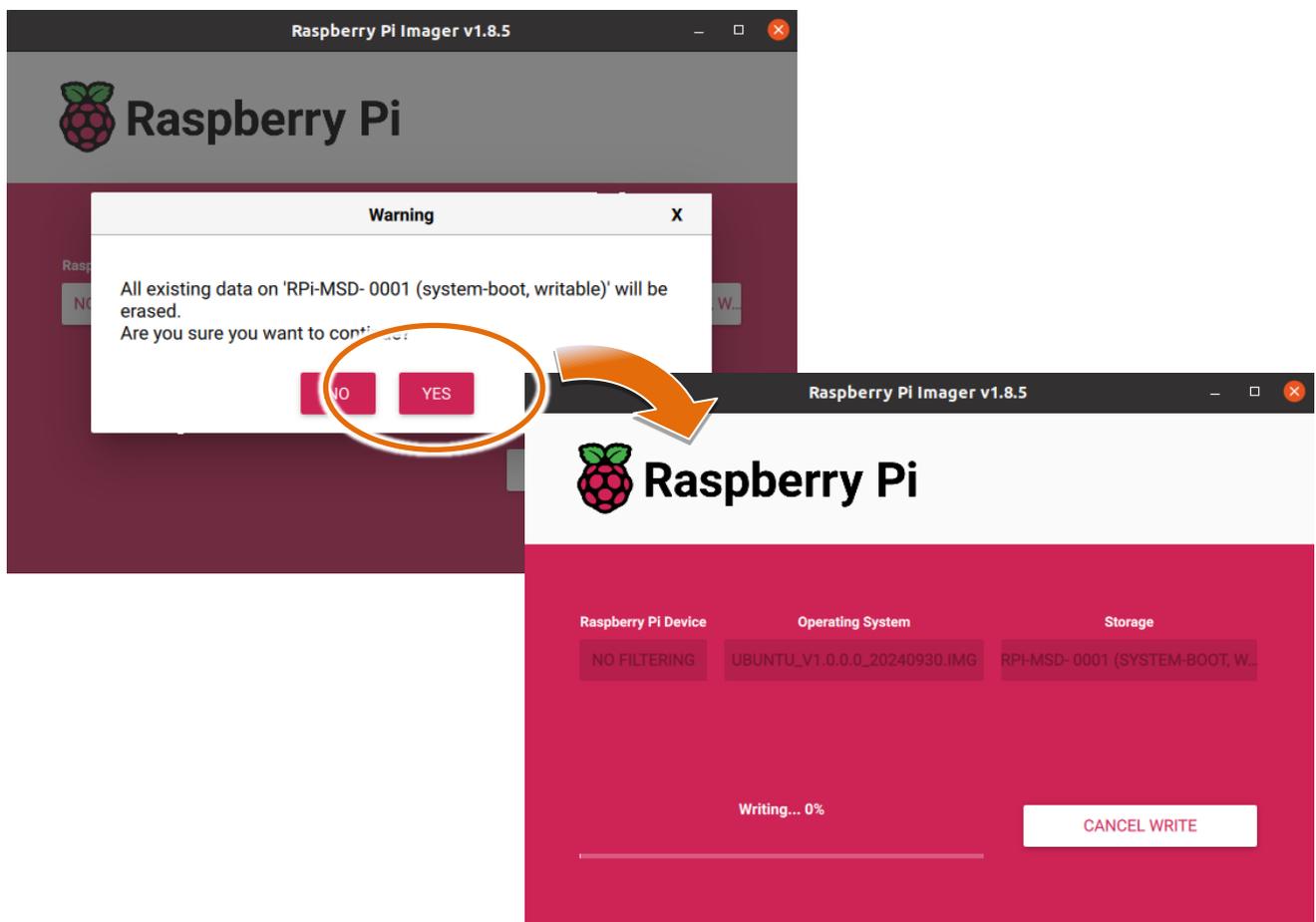


## 步驟 9：燒錄前最後設定

此次選項是詢問是否有其他 OS 客製化設定，請選擇 “NO” 進行下一步

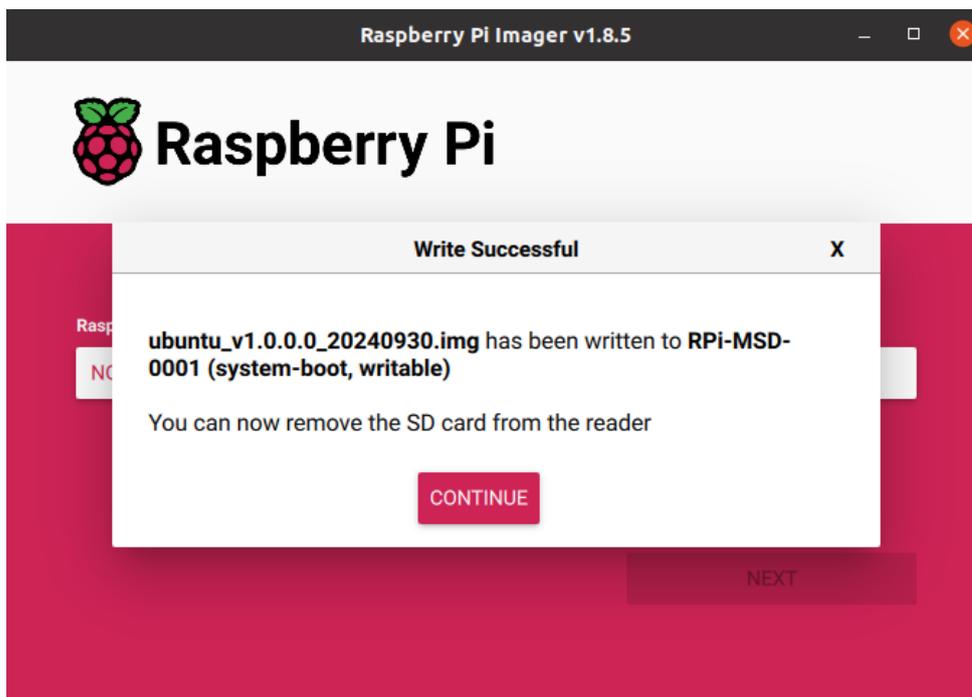


步驟 10：警告磁碟將會被清除，如果已備份資料請選擇 “YES”，將會開始燒錄 OS  
燒錄 OS 後約等待 30~40 分鐘



## 步驟 10：OS 燒錄完成

燒錄 OS 完成後的圖片如下



## 步驟 11：將 AIB-2941M 電源關閉，並將接在 AIB-2941M 上 mini USB 線移除

## 步驟 12：檢查 OS 版本

燒錄完成後重新上電，並執行 Utility，選取 Device Information 頁面檢查 OS 版本。



## 附錄 A. Tips –實用技巧

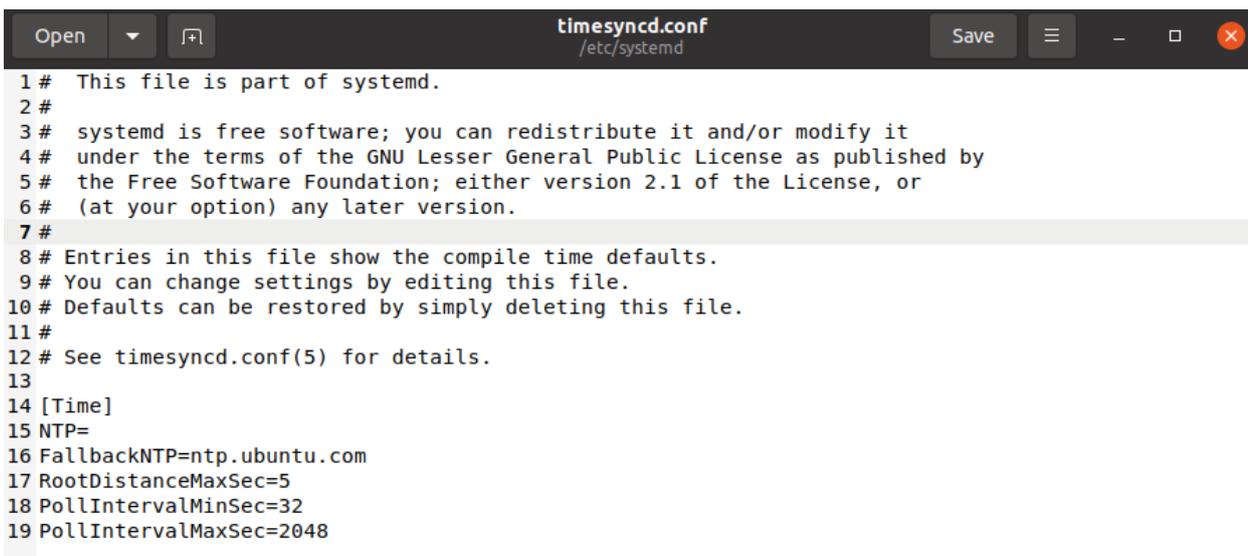
本章提供了一些對於使用和維護 AIB-2941M 的提示。

### A.1. 如何設定 AIB-2941M 自動網路校時

你可以透過時間服務器，同步 AIB-2941M 的系統時間。

如果同步啟用，AIB-2941M 的系統時間將與網路時間服務器同步。

**步驟 1：**開啟 Terminal，並輸入 `sudo gedit /etc/systemd/timesyncd.conf`，開啟設定如下圖



```
1 # This file is part of systemd.
2 #
3 # systemd is free software; you can redistribute it and/or modify it
4 # under the terms of the GNU Lesser General Public License as published by
5 # the Free Software Foundation; either version 2.1 of the License, or
6 # (at your option) any later version.
7 #
8 # Entries in this file show the compile time defaults.
9 # You can change settings by editing this file.
10 # Defaults can be restored by simply deleting this file.
11 #
12 # See timesyncd.conf(5) for details.
13
14 [Time]
15 NTP=
16 FallbackNTP=ntp.ubuntu.com
17 RootDistanceMaxSec=5
18 PollIntervalMinSec=32
19 PollIntervalMaxSec=2048
```

各參數設定如下表

欄位	說明
NTP	主要校時伺服器
FallbackNTP	備用校時伺服器
RootDistanceMaxSec	與伺服器的時間差距最多為多少(單位為秒)
PollIntervalMinSec	最少幾秒同步一次
PollIntervalMaxSec	最多幾秒同步一次

步驟 2：設定完成，按下 **SAVE** 儲存設定

步驟 3：在 **Terminal** 內輸入 `sudo systemctl restart systemd-timesyncd`，將時間校正服務重新啟動載入新設定

## A.2. 如何更換 RTC 電池

RTC 晶片使用鋰電池，可以提供持續 10 年的電力。下圖顯示了安裝在 AIB-2941M PCB 板上的電池的位置。



### 更換電池

1. 關閉 AIB-2941M 設備的電源。
2. 卸下 CPU 板。
3. 從 CPU 板上的電池盒中取出電量不足的電池。
4. 插入新電池。
5. 設置 RTC 時間。

### 訂購資訊

電池類型：BR1225 ( ICP DAS 的型號為 2LB010 )

有關更多詳細信息，請聯繫您當地的銷售辦事處或分銷商。

## 附錄 B. Hailo-8L 開發及應用

Hailo-8L 的 AI 學習模型的應用開發流程如下圖：



### 環境設置

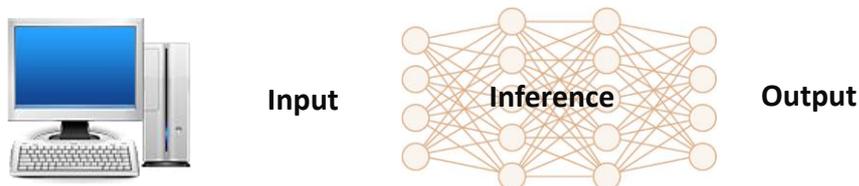
- AIB-2941M runtime 環境設置: HailoRT 及 Hailo PCIe driver 安裝 (參閱章節 4.1.1)
- 訓練 AI 模型用的 PC 環境設置: CUDA 及深度學習框架安裝 (參閱章節 4.1.2)



### AI 學習模型訓練和效能評估

現行 AI 模型需要使用者提供大量資料給 AI 進行訓練，才能讓 AI 提取出目標物件的特徵進行辨識或預測，訓練完的 AI 模型也需要進行測試評估其推理效能。

更多有關於 AI 學習模型導入及訓練資訊，請參閱章節 4.2。



## 模型轉換

訓練好的 AI 模型，需要經過 Hailo 的轉換工具，將其轉換成 Hailo 8L 能使用的檔案格式，模型轉換工具安裝及轉換流程 (參閱章節 4.3.2)



## 使用及開發 AI 模型程式

轉換完的 AI 模型在 AIB-2941M 上布署，需要程式將 AI 模型以及資料輸入至 Hailo 8，並由程式接收 AI 模型在 Hailo 8 運算的辨識結果

更多有關於 AI 學習模型導入及訓練資訊，請參閱章節 4.4。

後續將逐章說明安裝和操作流程順序，本次操作的 PC 使用的 OS 為 Ubuntu 20.04，所用 GPU 為 RTX 2060。

## B.1. 環境設置

### B.1.1. AIB-2941M runtime 環境設置: HailoRT 及 Hailo PCIe driver 安裝

AIB-2941M 如果要運作 Hailo 的 AI 加速卡，需要安裝 HailoRT 和 Hailo PCIe driver，以下為安裝 HailoRT 和 Hailo PCIe driver 的步驟，請使用者先至下列 Hailo 網站下載 HailoRT 和 Hailo PCIe driver。

<https://hailo.ai/developer-zone/software-downloads/>

以下過程為安裝 4.15 版 HailoRT 和 PCIe driver

#### 指導與提示



HailoRT 和 PCIe driver 版本一定要一致，版本不一致，HailoRT 運作過程可能會發生問題。

**步驟 1：**開啟 Terminal，輸入指令 `sudo apt-get install build-essential` 後，輸入密碼

**步驟 2：**開啟 Terminal，切換至 HailoRT 安裝檔案路徑，並輸入 `sudo dpkg -i hailort_4.15.0_arm64.deb` (此例為 4.15 版 HailoRT)進行檔案安裝，

出現下列畫面表示已安裝完成。

```
pi@pi-desktop:~$ sudo dpkg -i hailort_4.15.0_arm64.deb
(Reading database ... 168833 files and directories currently installed.)
Preparing to unpack hailort_4.15.0_arm64.deb ...
Unpacking hailort (4.15.0) over (4.15.0) ...
Setting up hailort (4.15.0) ...
Do you wish to activate hailort service? (required for most pyHailoRT use cases) [y/N]:
Starting hailort.service
Created symlink /etc/systemd/system/multi-user.target.wants/hailort.service → /lib/systemd/system/hailort.service.
```

**步驟 3：**開啟 Terminal，切換至 Hailo PCIe Driver 安裝檔案路徑，輸入 `sudo dpkg -i hailort-pcie-driver_4.15.0_all.deb` (此例為 4.15 版 Hailo PCIe driver)執行安裝  
出現下列畫面表示已安裝完成。

```
pi@pi-desktop:~$ sudo dpkg -i hailort-pcie-driver_4.15.0_all.deb
Selecting previously unselected package hailort-pcie-driver.
(Reading database ... 168833 files and directories currently installed.)
Preparing to unpack hailort-pcie-driver_4.15.0_all.deb ...
Could not test for SecureBoot, assuming SecureBoot is disabled on this machine.
Unpacking hailort-pcie-driver (4.15.0) ...
Setting up hailort-pcie-driver (4.15.0) ...

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

build-essential/jammy,now 12.9ubuntu3 arm64 [installed]
Do you wish to use DKMS? [Y/n]:
Please reboot your computer for the installation to take effect.
```

**步驟 4：**開啟 Terminal，輸入 `hailortcli scan` 確認是否有偵測到 Hailo 硬體

有偵測到 Hailo 硬體，會列出 Device ID。如下圖偵測到的 Hailo 硬體即表示 HailoRT 和 Hailo PCI Driver 完裝成功。

```
pi@pi-desktop:~$ hailortcli scan
Hailo Devices:
[-] Device: 0000:04:00.0
```

### 指導與提示



OS 有分成已安裝 Hailo 相關軟體(HailoRT 和 Hailo PCIe driver)和未安裝相關軟體版本，請使用者可自行至下列網址下載 OS Image  
<https://www.icpdas.com/en/download/show.php?num=9388&model=AIB-2941M>

## B.1.2. PC 環境設置: CUDA 及深度學習框架安裝

PC 環境需要安裝 CUDA 和 Deep Learning Framework，因目前主流的 GPU 都是 Nvidia 所設計，Nvidia 提供的 GPU API 是 CUDA，大部份 Deep learning architectures 都是建置在 CUDA 上，因此 CUDA 是必須安裝的。

以下將說明在 Ubuntu 環境下，如何安裝 CUDA 和 Deep learning architectures，以及如何訓練 AI model。

### CUDA 安裝

**步驟 1：在 ubuntu 環境下，請在 terminal 下輸入 ubuntu-drivers devices**

將會列出 OS 內可以安裝的 Nvidia GPU driver，建議安裝有 “recommended” 提示的 driver，此版 driver 有經過 Nvidia 驗證，較不容易出現問題。

```
test@test-MS-7D24:~$ ubuntu-drivers devices
== /sys/devices/pci0000:00/0000:00:01.0/0000:01:00.0 ==
modalias : pci:v000010DEd00001F03sv0000196Esd00001394bc03sc00i00
vendor    : NVIDIA Corporation
driver    : nvidia-driver-520 - third-party non-free
driver    : nvidia-driver-555 - third-party non-free
driver    : nvidia-driver-555-open - third-party non-free recommended
driver    : nvidia-driver-545 - third-party non-free
driver    : nvidia-driver-535-server - distro non-free
driver    : nvidia-driver-535 - third-party non-free
driver    : nvidia-driver-550 - third-party non-free
driver    : nvidia-driver-470-server - distro non-free
driver    : nvidia-driver-525 - third-party non-free
driver    : nvidia-driver-535-server-open - distro non-free
driver    : nvidia-driver-470 - third-party non-free
driver    : nvidia-driver-515 - third-party non-free
driver    : nvidia-driver-535-open - distro non-free
driver    : nvidia-driver-550-open - third-party non-free
driver    : xserver-xorg-video-nouveau - distro free builtin
```

## 步驟 2：安裝完 GPU driver 後，請在 terminal 下輸入 nvidia-smi

右上角的“CUDA Version”就是你後續步驟要安裝的 CUDA toolkit 版本，請安裝對應的版本，不然可能會出現異常。

```
test@test-MS-7D24:~$ nvidia-smi
Wed Nov 13 09:10:54 2024
+-----+
| NVIDIA-SMI 535.183.01          Driver Version: 535.183.01    CUDA Version: 12.0   |
+-----+-----+-----+-----+-----+-----+
| GPU  Name                   Persistence-M | Bus-Id        Disp.A    Volatile Uncorr. ECC |
| Fan  Temp   Perf              Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+=====+=====+
|  0  NVIDIA GeForce RTX 2060     Off          | 00000000:01:00:0  On          N/A           | |
| 45%  32C   P8               16W / 184W   | 182MiB / 12288MiB |    0%      Default |
|                                          |                    |            MIG M. |
+-----+-----+-----+-----+-----+-----+
| Processes: |
| GPU   GI    CI          PID    Type   Process name          GPU Memory |
|=====+=====+=====+=====+=====+=====+
|  0   N/A  N/A         1176     G   /usr/lib/xorg/Xorg           23MiB |
|  0   N/A  N/A         1685     G   /usr/lib/xorg/Xorg           38MiB |
|  0   N/A  N/A         1814     G   /usr/bin/gnome-shell        111MiB |
+-----+-----+-----+-----+-----+-----+

```

## 步驟 3：請至下列 Nvidia 網站下載對應的 CUDA toolkit 安裝檔案

<https://developer.nvidia.com/cuda-toolkit-archive>

網站如下圖，本範例是對應 12.0 版本

# CUDA Toolkit Archive

Previous releases of the CUDA Toolkit, GPU Computing SDK, documentation and developer drivers can be found using the links below. Please select the release you want from the list below, and be sure to check [www.nvidia.com/drivers](http://www.nvidia.com/drivers) for more recent production drivers appropriate for your hardware configuration.

[Download Latest CUDA Toolkit](#) [Learn More about CUDA Toolkit](#)

**Latest Release**  
[CUDA Toolkit 12.6.2 \(October 2024\), Versioned Online Documentation](#)

### Archived Releases

- [CUDA Toolkit 12.6.1 \(August 2024\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.6.0 \(August 2024\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.5.1 \(July 2024\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.5.0 \(May 2024\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.4.1 \(April 2024\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.4.0 \(March 2024\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.3.2 \(January 2024\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.3.1 \(November 2023\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.3.0 \(October 2023\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.2.2 \(August 2023\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.2.1 \(July 2023\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.2.0 \(June 2023\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.1.1 \(April 2023\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.1.0 \(February 2023\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.0.1 \(January 2023\), Versioned Online Documentation](#)
- [CUDA Toolkit 12.0.0 \(December 2022\), Versioned Online Documentation](#)
- [CUDA Toolkit 11.8.0 \(October 2022\), Versioned Online Documentation](#)

#### 步驟 4：請按照模組所使用軟體環境做選擇

依序選擇 OS、硬體架構...等。本範例及模組所使用硬體為 x86 PC，所用的 OS 為 Ubuntu 20.04，安裝種類上我們是採用 deb 檔案安裝，因此按下圖選擇

Operating System	Linux	Windows							
Architecture	x86_64	ppc64le	arm64-sbsa	aarch64-jetson					
Distribution	CentOS	Debian	Fedora	KylinOS	OpenSUSE	RHEL	Rocky	SLES	Ubuntu
Version	20.04	22.04							
Installer Type	deb (local)	deb (network)	runfile (local)						

#### 步驟 5：按照提示操作，將指令逐步在 terminal 上執行即可完成安裝步驟

### Download Installer for Linux Ubuntu 20.04 x86\_64

The base installer is available for download below.

> Base Installer

Installation Instructions:

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin
$ sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600
$ wget https://developer.download.nvidia.com/compute/cuda/12.2.0/local_installers/cuda-repo-ubuntu2004-12-2-local_12.2.0-535.54.03-1_amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu2004-12-2-local_12.2.0-535.54.03-1_amd64.deb
$ sudo cp /var/cuda-repo-ubuntu2004-12-2-local/cuda-*-keyring.gpg /usr/share/keyrings/
$ sudo apt-get update
$ sudo apt-get -y install cuda
```

#### 步驟 6：安裝完成 CUDA 後，請至 terminal 上輸入 `nvcc --version`，確認安裝的 CUDA 版本是否正確

本次安裝的是 12.0 版本，輸出畫面如下

```
test@test-MS-7D24:~$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Mon Oct 24 11:12:58 PDT 2022
Cuda compilation toolchain release 12.0, V12.0.76
Build cuda_12.0.r12.0/compiler.31968024_0
```

## 深度學習框架 (Deep Learning Architectures) 安裝

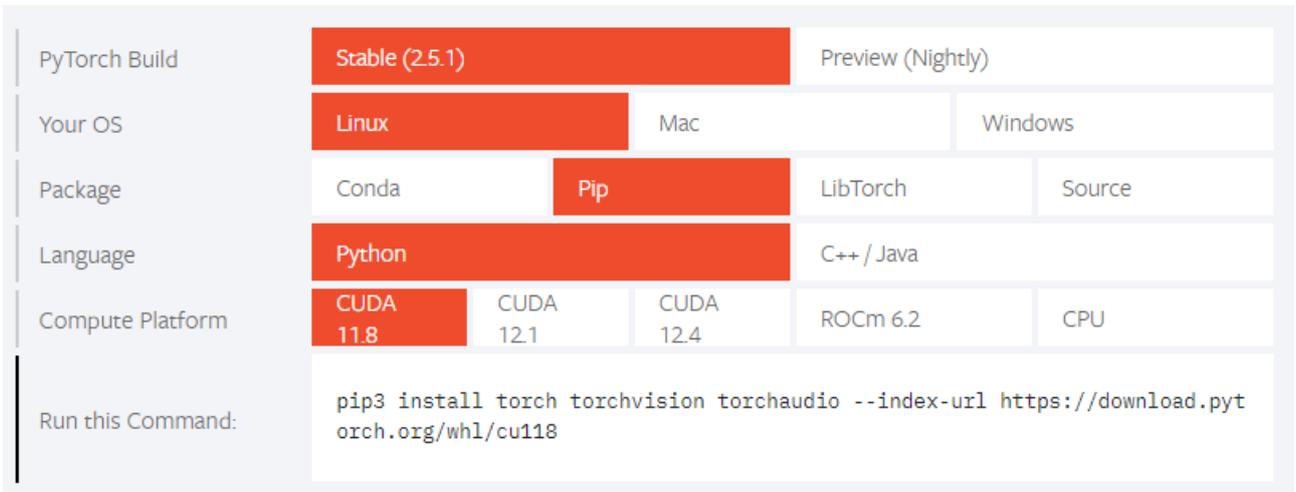
Deep learning architectures 有多種例如 Pytorch、tensorflow...etc，主要是對應你所要用的 AI Model，不同的 AI Model 會對應不同的 Deep learning architectures，因此請先確認你要用的 AI Model 來安裝相對應的 Deep learning architectures，本例所用 AI model 是 yolov5，對應的 Deep learning architectures 是 Pytorch，因此本章節會以 Pytorch 安裝為主。

### 步驟 1：請至 Pytorch 網站 <https://pytorch.org/> 進行安裝流程

如下圖依自己 PC 環境來依序點選

本例是在 Ubuntu 下用 pip 進行安裝，CUDA 版本的選擇上，CUDA 可以向下相容，請選擇較你當前 CUDA 版本舊或相等版本，例如前一章節安裝 CUDA 版本為 12.0，因此選擇 11.8。

因本次用 pip 進行安裝，因此請在 terminal 輸入 “Run this Command” 顯示的安裝命令。



PyTorch Build	Stable (2.5.1)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 11.8	CUDA 12.1	CUDA 12.4	ROCm 6.2 CPU
Run this Command:	pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118			

步驟 2：安裝完成後，可在 pip 查詢，查詢結果如下，表示 Pytorch 已經安裝完成。

```
test@test-MS-7D24:~$ pip list | grep torch
torch                2.0.1+cu118
torchaudio           2.0.2+cu118
torchvision          0.15.2+cu118
```

## B.2. AI 學習模型訓練和效能評估

AI Model 有多種，使用者請依自己所需選擇自己所要用的 AI model，本例以 yolov5 為例，yolov5 為 ultralytics 基於 yolov4 發展的物件辨識 AI Model，較 yolov4 有輕量化的特點，以下流程為 yolov5 的 AI Model 訓練流程和效能評估。

### B.2.1. 模型訓練

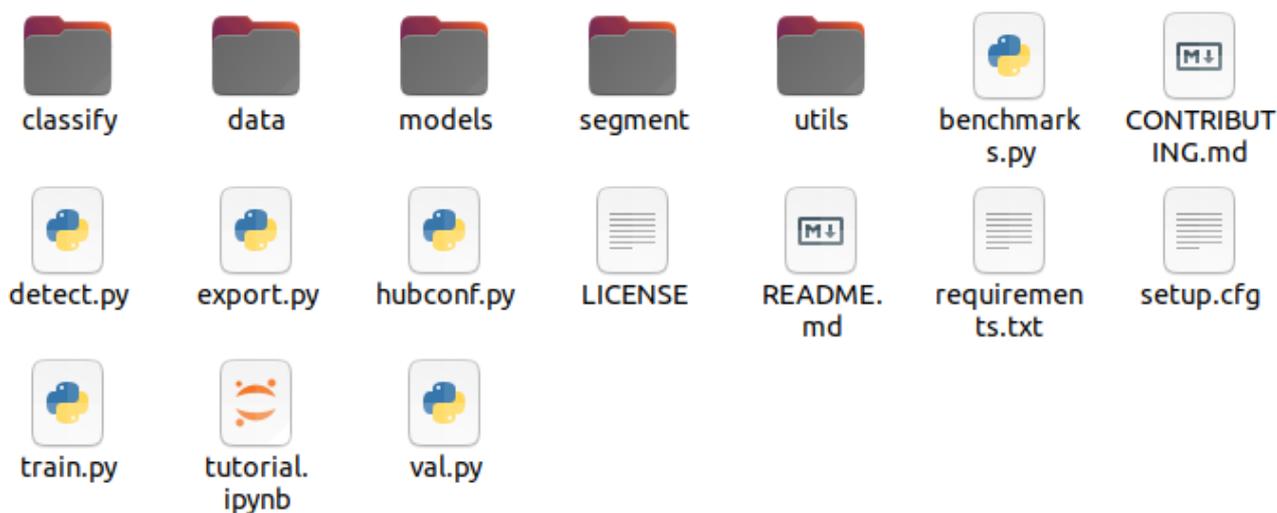
#### 步驟 1：從 ultralytics 的 github 下載 yolov5

最新版的 yolov5 可至以下連結下載：

<https://github.com/ultralytics/yolov5>

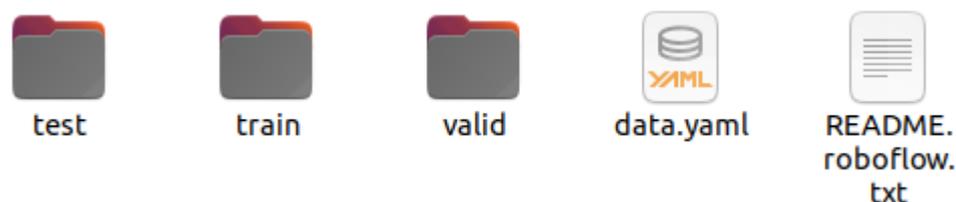
yolov5 目前最新版已到 v7.0，以下將以 v7.0 為例

#### 步驟 2：下載的 yolov5 解壓縮後如下圖



#### 步驟 3：加入訓練用資料集，訓練用資料

目前網路上有多個網站可免費分享訓練圖片例如 coco、roboflow...etc.，本次訓練用資料從 roboflow 下載進行訓練，本次訓練目標是針對汽車和車牌進行辨識，下圖的即為本次訓練用資料集



#### 步驟 4：開啟 terminal，切換至 yolov5 的資料夾，並輸入如下訓練指令

```
python3 train.py --img-size 640 --batch 16 --epochs 300
--data ./car_license/data.yaml --weights yolov5s.pt
```

##### 以下介紹幾項參數

--data 為資料集中的 yaml 檔路徑

--epochs 為訓練周期，如果你的資料集內有 100 張圖片，對這 100 張圖片進行一輪運算即為一個周期

--batch 為每次輸入多少資料給 GPU 進行訓練，數值越大所用 GPU 記憶體越大

其他參數，請輸入 `python3 train -help` 獲得詳細說明

在 terminal 輸入上述指令後，即如下圖開始進行訓練

```
AutoAnchor: 3.98 anchors/target, 1.000 Best Possible Recall (BPR). Current anchors are a good fit to dataset ✓
Plotting labels to runs/train/exp/labels.jpg...
Image sizes 640 train, 640 val
Using 8 dataloader workers
Logging results to runs/train/exp
Starting training for 300 epochs...

Epoch   GPU_mem  box_loss  obj_loss  cls_loss  Instances  Size
0/299   3.46G    0.1238   0.03906   0.02909     90        640: 50% | ██████████ | 8/16 00:02
```

#### 步驟 5：訓練完成後如下圖顯示資訊

訓練完的 AI model 會在 `/runs/train/exp<number>/weights` 內，number 是你已運行過幾次訓練，本例為第一次訓練，因此會是 `exp`，`weights` 資料夾內會有兩個模型，`best.pt` 是所有周期中訓練最好的，`last.pt` 是最後一個周期訓練出來的模型。下圖訓練完成的資訊所顯示之意義，在下一個章節將進行說明。

```
Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs
Class      Images  Instances  P      R      mAP50  mAP50-95: 100% | ██████████ | 3/3 00:00
  all       70      218        0.839  0.834  0.873   0.676
  license-plate'  70      84         0.888  0.893  0.941   0.705
  vehicle   70      134        0.79   0.776  0.805   0.646
Results saved to runs/train/exp
```

#### 指導與提示



本次訓練用資料集約 300 張圖片，訓練用 GPU 為 RTX 2060，batch 值設定為 16，訓練用時間約 2~3 個小時

## B.2.2. 模型效能評估

在 “B.2. AI 學習模型訓練和效能評估”，訓練完成後顯示如下圖

```
Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs
   Class      Images  Instances      P      R      mAP50  mAP50-95: 100% | ██████████ | 3/3 00:00
   all         70       218      0.839   0.834   0.873   0.676
   license-plate 70        84      0.888   0.893   0.941   0.705
   vehicle     70       134      0.79    0.776   0.805   0.646
Results saved to runs/train/exp/weights/best.pt
```

其中 mAP50-95 的數值是用來評估模型對影像中的物件識別率和正確框選出來的比率，數值越高表示模型對物件的辨識能力越強

各項參數意義可參考 Ultralytics 的說明

<https://docs.ultralytics.com/guides/yolo-performance-metrics/#introduction>

訓練完的 AI model 可以用 detect.py 進行驗證，驗證資料集以外的圖像是否能有效辨識，輸入  
python3 detect.py --source test.jpg --weight best.pt  
下圖為訓練好的 AI model 辨識汽車和車牌的結果

--source 為你要偵測的圖片或影像

--weight 為你要用於偵測的 AI model



## B.3. 模型轉換

### B.3.1. 模型轉換工具安裝及模型供 Hailo-8L 使用

B.2.3 章節已說明如何訓練 AI model，但訓練出來的 AI model 不能馬上用於 AIB-2941M 上，因為 HailoRT 只能運作 Hailo 格式的 AI model，因此我們還需要進行 AI model 的轉換。

AI model 的轉換，除了需要安裝 Data flow compiler，還需要另外編寫程式進行，因轉換的程式會隨著使用者在訓練過程中所使用的參數跟著修改，且轉換後的處理結果可能會有所偏差需要進行修正，因此沒有通用的程式版本，這部份需要尋求 Hailo 的支援，可以將你們的 AI model 轉換問題提交給我們，由我們幫你們聯絡 Hailo 進行支援，本章節只說明 Data flow compiler 的安裝和轉換 AI model 的前置工作。

## 模型轉換工具 Data Flow Compiler 安裝

步驟 1：請使用者先至下列 Hailo 網站下載 Data flow compiler

<https://hailo.ai/developer-zone/software-downloads/>

### 指導與提示



HailoRT 和 Data flow compiler 有相對應關係，對應表如下

HailoRT	Data flow compiler
4.13	3.23
4.14	3.24
4.15	3.25

HailoRT 4.15 以後的版本依此表類推。

1. HailoRT 版本可以較 Data flow compiler 新，但 Data flow compiler 不能比 HailoRT 新，例如 HailoRT 為 4.15，Data flow compiler 可以安裝 3.23~3.25，如果 HailoRT 為 4.13 則只能安 3.23 以下版本。
2. 安裝版本如果不相容，轉換好的 AI model 會無法在 AIB-2941M 上的 runtime 環境運作。
3. OS 有分成已安裝 Hailo 相關軟體(HailoRT 和 Hailo PCIe driver)和未安裝相關軟體版本，已安裝 Hailo 相關軟體的 OS 會預先安裝 4.19 版 HailoRT 和 Hailo PCIe Driver，請安裝 4.29 版或較舊版本的 Data flow compiler

因在“4.1.1. AIB-2941M 的 runtime 環境建立”章節已安裝 4.15 版 HailoRT，以下說明為安裝 3.25 版本的 Data flow compiler。

## 步驟 2：輸入 `pip install hailo_dataflow_compiler-3.25.0-py3-none-linux_x86_64.whl`

因 data flow compiler 還需要多種套件下載和安裝，安裝過程較花時間，安裝完成顯示如下資訊

```
Successfully installed hailo-dataflow-compiler-3.25.0 keras-2.12.0 onnx-1.14.0 onnxsim-0.4.17 protobuf-3.20.3 pydantic-1.10.8 setuptools-68.0.0 tensorboard-2.12.3 tensorflow-2.12.0 tensorflow-estimator-2.12.0 tensorflow-probability-0.20.1 tfllite-2.10.0
```

在 `pip list` 中可以查詢我們安裝的 Data flow compiler 版本

```
test@test-VirtualBox:~$ pip list | grep dataflow
hailo-dataflow-compiler 3.25.0
```

## 模型轉換供 Hailo-8L 使用

在開始轉換前，我們要將先前章節“B.2.3. AI model 訓練”訓練出來的“best.pt”轉換成 onnx 檔案，data flow compiler 轉換的檔案只支援 onnx 和 ckpt 格式檔案，因此“best.pt”需要進行轉換。

Yolov5 內的 `export.py` 可支援轉換 onnx 格式，請開啟 terminal，切換至 yolov5 的資料夾，並輸入如下指令

```
python3 export.py --weights best.pt --include torchscript onnx
```

轉換成 onnx 檔案格式成功的畫面如下

```
ONNX: export success □ 0.4s, saved as best.onnx (27.2 MB)

Export complete (1.3s)
Results saved to /home/test/yolov5-7.0
Detect:      python detect.py --weights best.onnx
Validate:   python val.py --weights best.onnx
PyTorch Hub: model = torch.hub.load('ultralytics/yolov5', 'custom', 'best.onnx')
Visualize:  https://netron.app
```

轉換成 Hailo 支援 HEF 格式所需要進行的三個步驟 Parsing、Quantization 和 Compilation，每一個步驟都需要另外編寫轉換程式進行轉換，我們在此步驟無法提供協助，需要 Hailo 專業支援，可聯絡我們將你的問題轉交給 Hailo 處理。

## B.4. 使用及開發 AI 模型程式

在 “B.3. 將訓練完的 AI model 轉換成 Hailo-8L 能運作的檔案格式” 章節中，已將你的 AI model 轉換成 HEF 格式檔案，後續需要開發應用程式在 AIB-2941M 上運作。

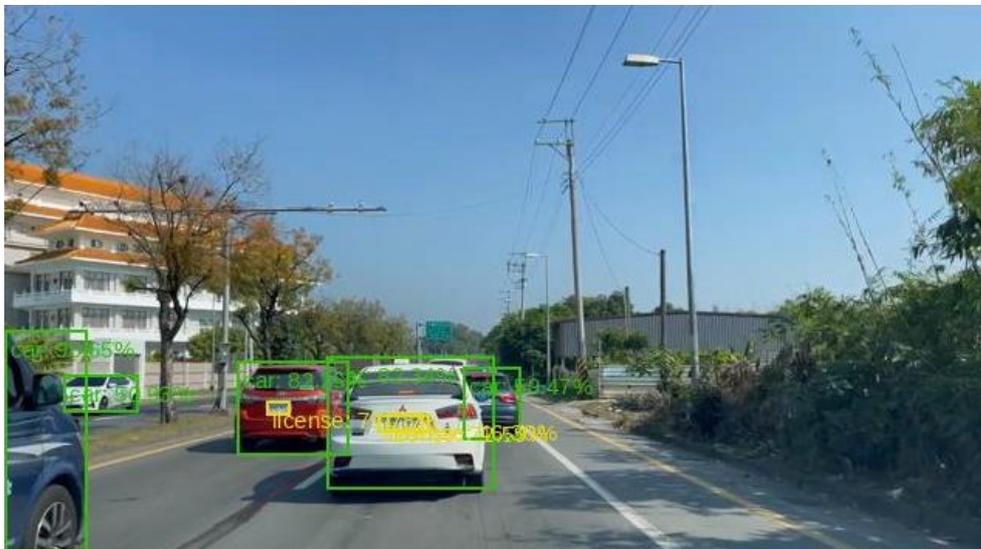
Hailo 的 GitHub 已提供應用程式的相關範例，請至下列的 GitHub 網址下載

<https://github.com/hailo-ai/Hailo-Application-Code-Examples>

hailo 已提供 python、C++和 C#...etc.範例給使用者參考，以 python 為例，hailo 提供下列種類的 demo 程式

APP	Description
depth_estimation	Depth estimation with StereoNet
detection_with_tracker	Object detection with tracking using ByteTracker and Supervision
hailo_onnxruntime	Inference with a Hailo device and postprocessing with ONNXRuntime
instance_segmentation	Instance segmentation with yolov5_seg/yolov8_seg
lane_detection	Lane detection with yolo, ssd or centernet
object_detection	Object detection with yolo, ssd or centernet
pose_estimation	Pose estimation with yolov8
streaming	Object detection on a streaming input from a camera using OpenCV
super_resolution	Super resolution with espcnx4 or srgan

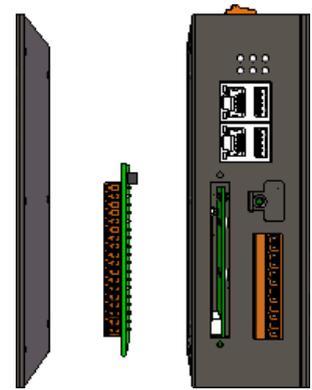
先前章節訓練出來的汽車和車牌辨識 AI model，可在 object\_detection 上應用，下圖為我們修改 demo 後在 AIB-2941M 上運作的範例



## 附錄 C. XV-Board 和 XB-Board 系列 I/O 擴充卡

XV-Board 和 XB-Board 系列 I/O 擴充卡適用於 AIB-2941M 系列控制器。每一台控制器皆可搭配一塊 XV-Board 或 XB-Board 系列 I/O 擴充卡來做擴充。

下表為各個 XV-Board 擴充模組的規格說明。



### DIO 擴充卡

模組	DI			DO	
	通道數	類型	Sink/Source	通道數	Sink/Source
XV107	8	Wet	Source	8	Sink
XV107A			Sink		Source
XV110	16	Dry/Wet	Sink/Source	-	-
XV111	-			16	Sink
XV111A					Source

### 繼電器輸出擴充卡

模組	DI			繼電器輸出	
	通道數	類型	Sink/Source	通道數	類型
XV116	5	Wet	Sink/Source	2	Signal Relay
				4	Power Relay

### 混合型 I/O 擴充模組

模組	AI	AO	DI			DO	
	通道數		類型	Sink/Source	通道數	Sink/Source	
XV308	8	-	DI+DO=8	Dry/Wet	Source	DI+DO=8	Sink
XV310	4	5			4		Sink

更多有關於 XV-Board 擴充模組的詳細說明，請參閱：

[http://www.icpdas.com/root/product/solutions/hmi\\_touch\\_monitor/touchpad/xv-board\\_selection.html](http://www.icpdas.com/root/product/solutions/hmi_touch_monitor/touchpad/xv-board_selection.html)

下表為各個 XB-Board 擴充模組的規格說明

### 混合型 I/O 擴充模組

模組	AI	AO	DI			DO	
	通道數		類型	Sink/Source	通道數	Sink/Source	
XB310	4	4	4	Dry/Contact	Source	4	Sink

更多有關於 XB-Board 擴充模組的詳細說明，請參閱：

[https://www.icpdas.com/en/product/guide+Remote I O Module and Unit+PAC I O Modules+XB-board](https://www.icpdas.com/en/product/guide+Remote+I+O+Module+and+Unit+PAC+I+O+Modules+XB-board)

## 附錄 D. 手冊修訂記錄

本章提供此使用手冊的修訂記錄。

下表提供此文件每次修訂的日期與說明。

版本	發行日	說明
1.0.1	Sep.2024	首次發行