

# Chapter 3：樹莓派 × OpenCV × YOLOv5

## 本章重點

安裝與設定樹莓派，並在樹莓派的 Linux OS 中實作 OpenCV 與 YOLOv5。

## 準備材料



可上網的電腦



樹莓派 (含相機模組與外殼)

## 學習目標

1. 樹莓派介紹與基本操作。
2. 網路設定。
3. 安裝樹莓派的相機模組。
4. 在樹莓派上編輯 Python。
5. 執行 OpenCV 專案。
6. 執行 YOLOv5 專案。

## 3-1. 樹莓派介紹與基本操作

樹莓派 (Raspberry Pi, 簡稱 Pi) 有以下特點：

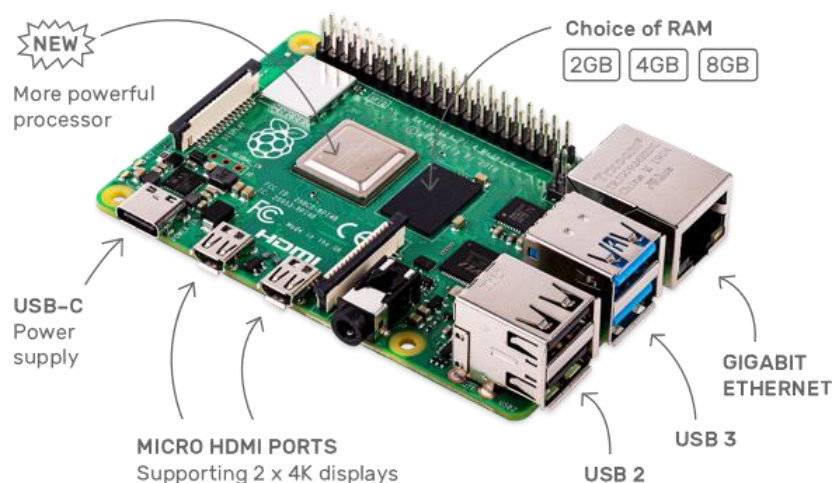
- 支援 SD 卡啟動：樹莓派沒有配置 FLASH，支援 SD 卡啟動作業系統。
- 支援多種作業系統：樹莓派由於其開源軟體的特性，支援很多的系統，Raspbian、Arch Linux ARM、Firefox OS、Google Chrome OS、Raspberry Pi Fedora Remix、WebOS、RISC OS、FreeBSD...等。
- 便宜、耗電低：使用 USB (micro USB 或 USB-C) 供電。
- 開發用途廣：可作為控制模組，開發多種電子、軟體、機械和機器人。
  - ✓ 樹莓派電腦：接上鍵盤、滑鼠與螢幕，安裝上 Windows 或 Linux 系統，就能當作平價的桌上型電腦使用。
  - ✓ 物聯網節點：樹莓派的網路介面完整，尤其是無線網路 (WiFi)，適合架構各種物聯網的應用。

### 硬體介紹

#### 規格

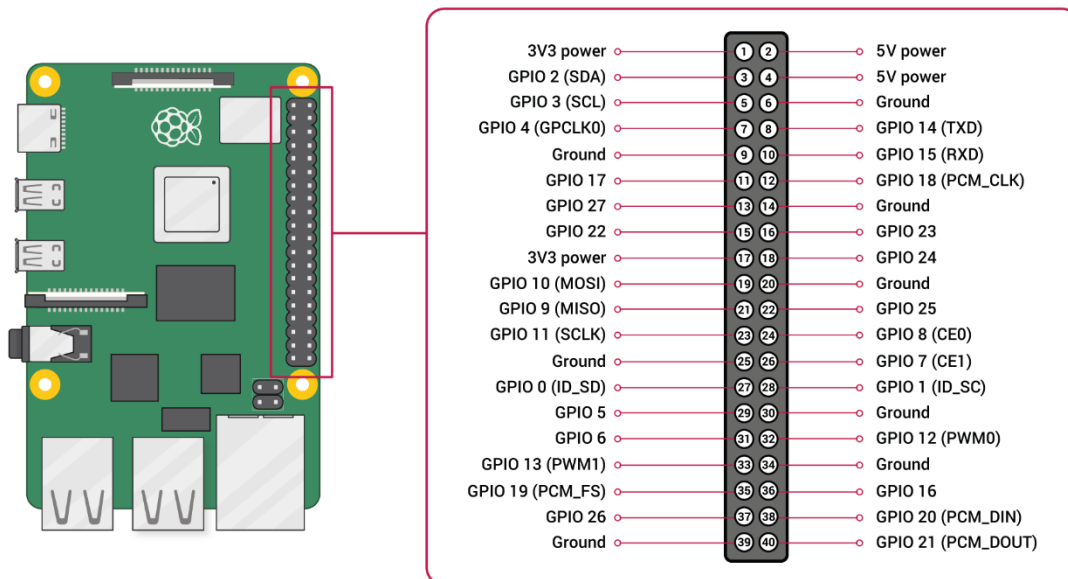
目前最新的是 Raspberry Pi 5，以下是 Pi 4 Model B 的介紹：

- CPU：ARM 7。
- RAM：有 2G/4G/8G 三種記憶體可選配。
- 使用 Type-C USB 供電。
- 支援雙螢幕輸出 (Micro HDMI)。
- 內建 WiFi/Gigabit Ethernet 兩種網路卡。



## 輸出/輸入腳位：General-Purpose Input/Output (GPIO)

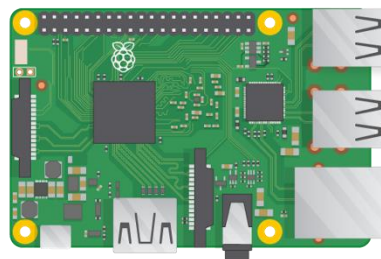
樹莓派有 40 個腳位可外接感測器或控制電路。注意單晶片的實際腳位與 PCB 板上的編號不同，寫程式時是控制實際腳位的，例如：當使用『12』，實際上是『GPIO 18』。



## 安裝作業系統

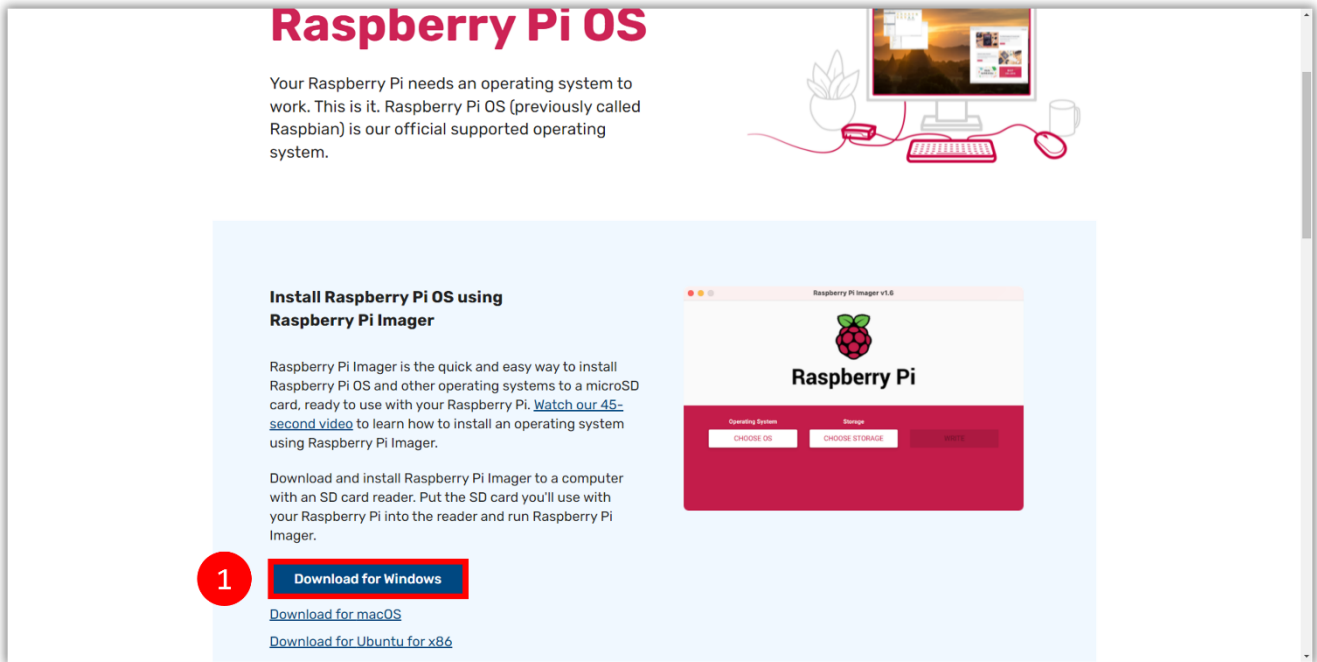
安裝樹莓派系統的三個步驟：

1. 下載樹莓派映象檔（在一台電腦上操作）。
2. 燒錄映象檔到 SD 卡（在一台電腦上操作）。
3. 安裝 SD 卡到樹莓派。



## Step 1 下載作業系統的燒錄程式。

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



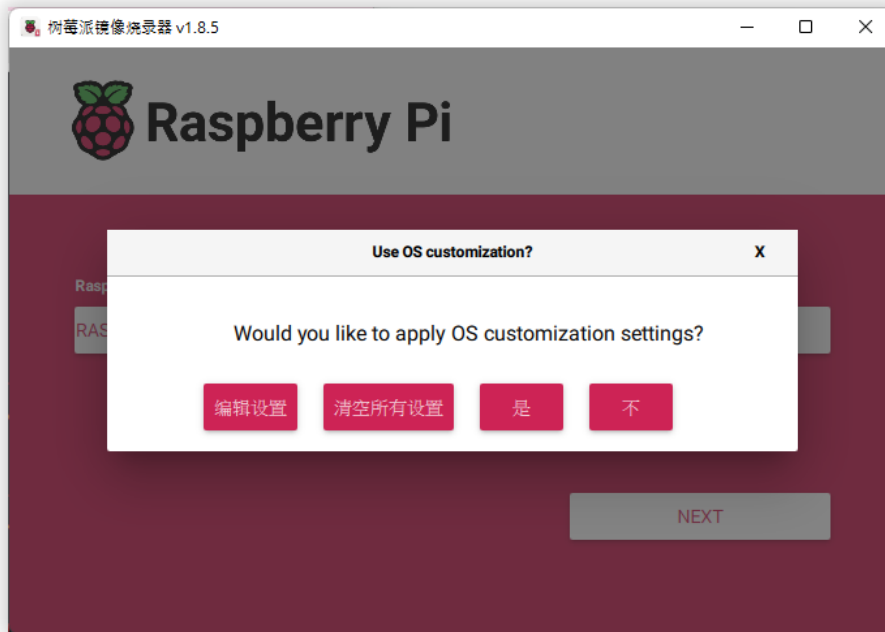
## Step 2 基本設定。

安裝 Raspberry Pi Image，執行後並燒錄合適的作業系統。以樹莓派安裝 **RASPBERRY PI OS (LEGACY, 32-BIT)** 為例：

1. 選擇裝置 『 RASPBERRY PI 4 』。
2. 作業系統選擇 『 RASPBERRY PI OS (LEGACY, 32-BIT) 』。
3. 選擇你的 SD 卡，按下 『 NEXT 』，進入設定。



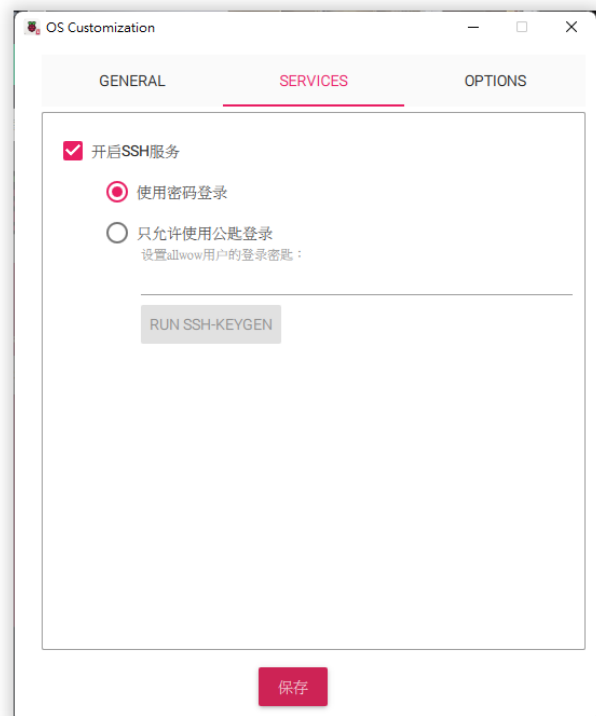
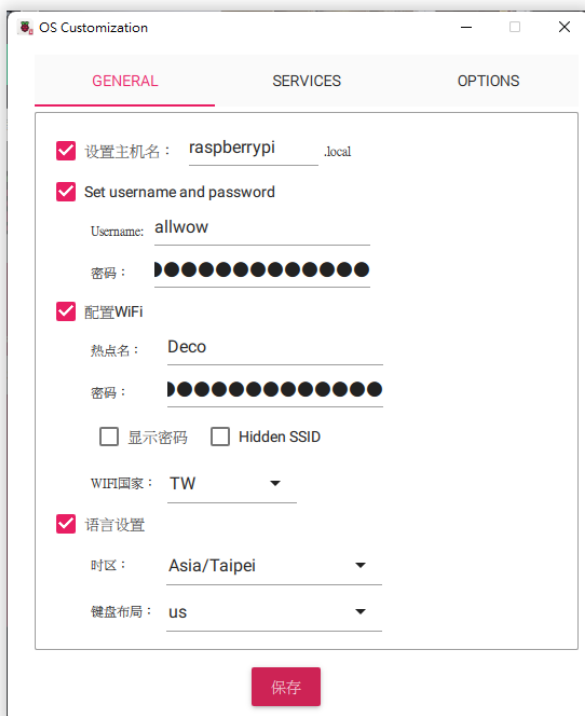
4. 按下編輯設置：



5. 在 GENERAL，設定『 Username 』與『 密碼 』。

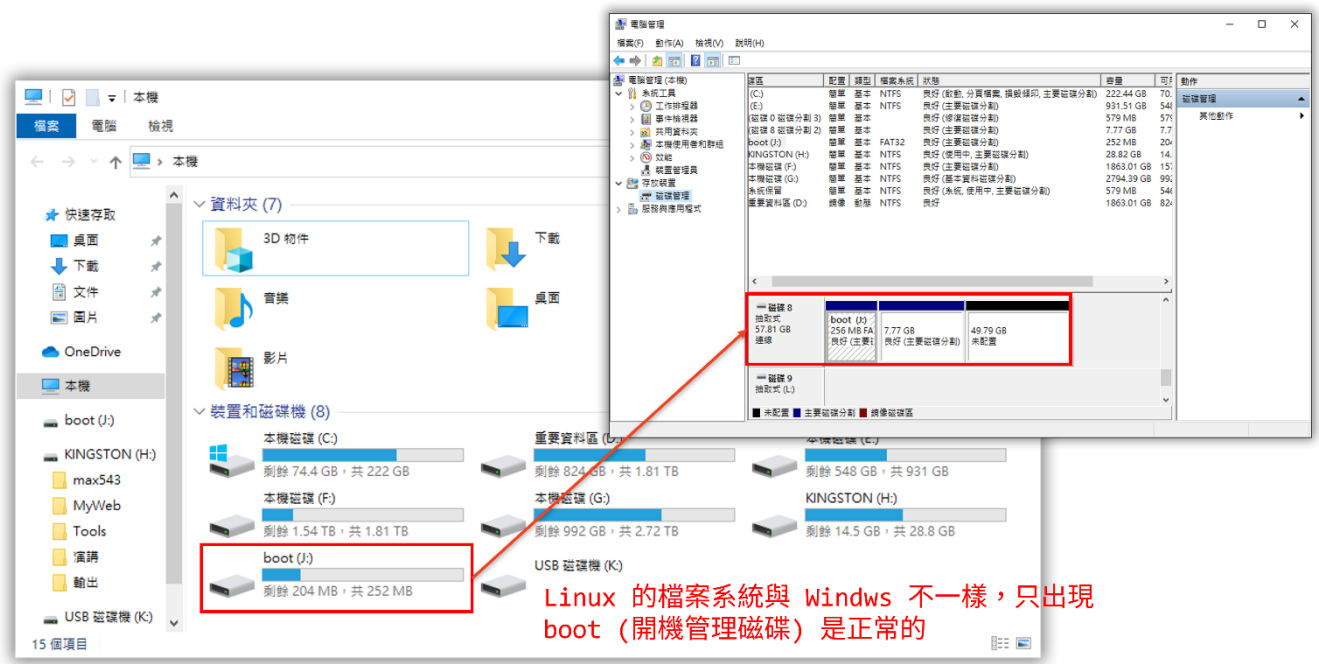
6. 設定 WiFi 的帳號與密碼。

7. 在 SERVICES，設定 SSH。按下『 保存 』，回到上一張圖後，按下『 是 』。



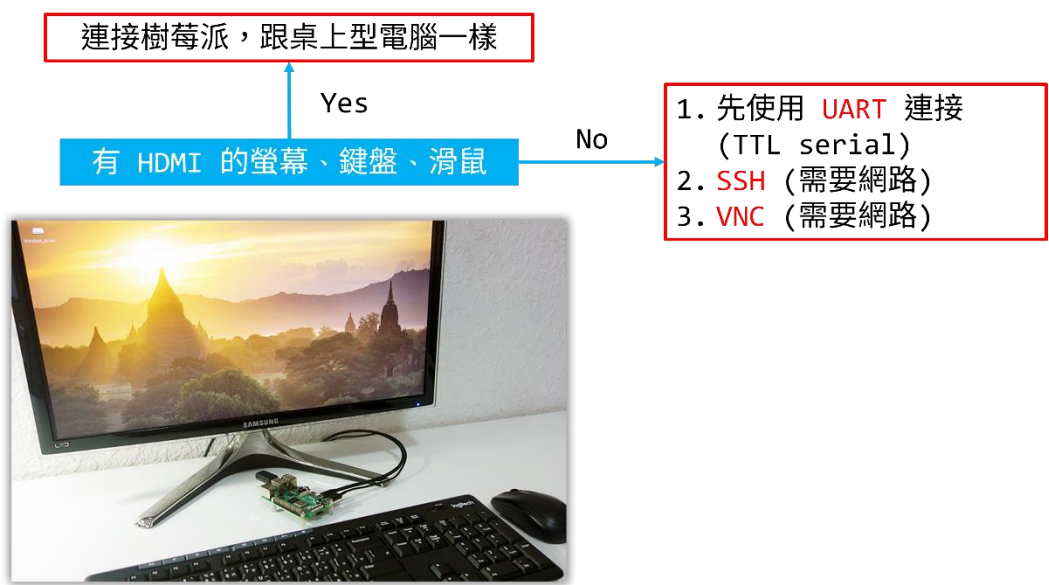
**Step 3** 燒錄完成會自動卸除 SD 卡，請再次插入電腦中。

只會看到 boot : Linux 的開機磁區。



## 控制原理

1. 可使用 micro HDMI to HDMI，連接一組專用的螢幕、鍵盤與滑鼠。(不推薦)
2. 使用一台電腦透過 UART (TTL serial) 設定好相關設定，再使用 SSH 或 VNC 遠端登入樹莓派。



## 需要準備的工作

使用一台電腦透過 UART Serial，先設定樹莓派的 WiFi。

- 本次實驗，請使用你的手機基地台分享一個區網 IP 給樹莓派，或登入的電腦與樹莓派同屬於同一個網段。
- 並熟悉設定 IP 的技巧。

在同一區網內，使用相同區網的電腦連上樹莓派。

- 有筆電，用你的 WiFi。
- 只有桌電，用一個 WiFi 網卡。

## 串列控制 (UART/TTL Serial)

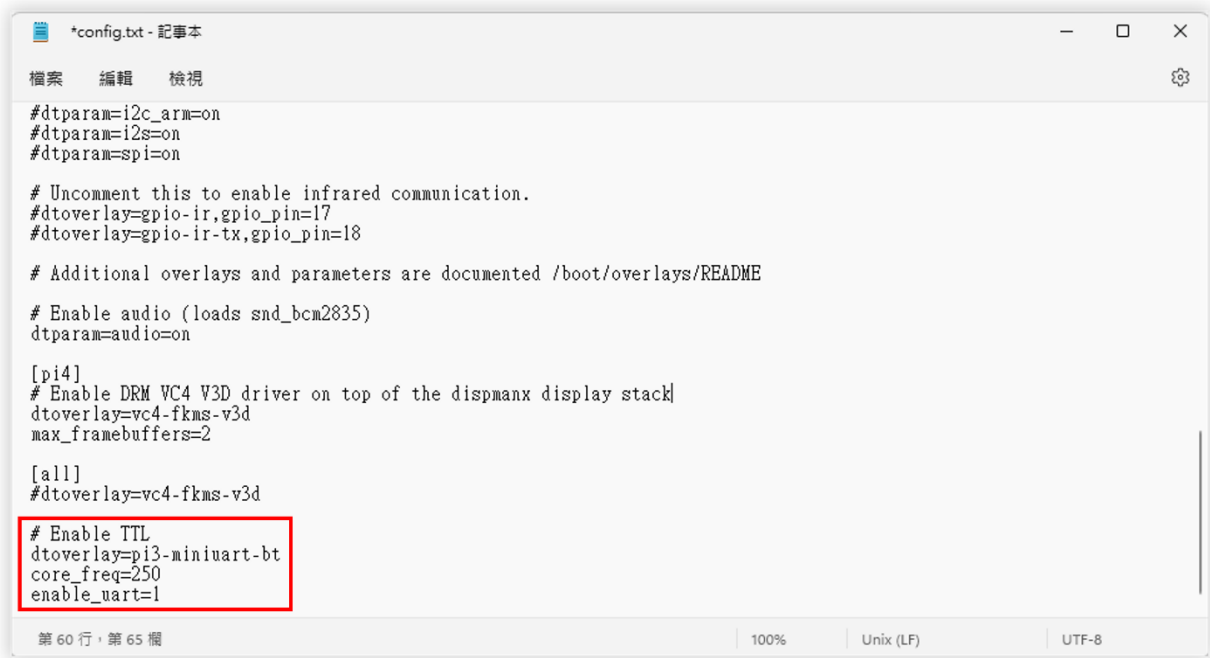
在一開始樹莓派沒有無線網路連接，所以也沒辦法使用 SSH。所以可以使用 UART port，連接 TTL 線來設定網路連線，並且 Enable SSH。我們使用 PL2303HXD 的 TTL 線。



## Step 1 修改樹莓派的 `/boot/config.txt`。

SD 卡插入讀卡機，使用一台電腦的編輯器（例如：記事本），修改 `/boot/config.txt`，加入下列三行到最後。

```
dtoverlay=pi3-miniuart-bt
core_freq=250
enable_uart=1
```



```
*config.txt - 記事本
檔案 編輯 檢視
#dtparam=i2c_arm=on
#dtparam=i2s=on
#dtparam=spi=on

# Uncomment this to enable infrared communication.
#dtoverlay=gpio-ir,gpio_pin=17
#dtoverlay=gpio-ir-tx,gpio_pin=18

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

[pi4]
# Enable DRM VC4 V3D driver on top of the dispmanx display stack
dtoverlay=vc4-fkms-v3d
max_framebuffers=2

[all]
#dtoverlay=vc4-fkms-v3d

# Enable TTL
dtoverlay=pi3-miniuart-bt
core_freq=250
enable_uart=1

第 60 行，第 65 欄 | 100% | Unix (LF) | UTF-8
```

## Step 2 刪除樹莓派的 `/boot/cmdline.txt` 中的 `quiet`，並存檔。



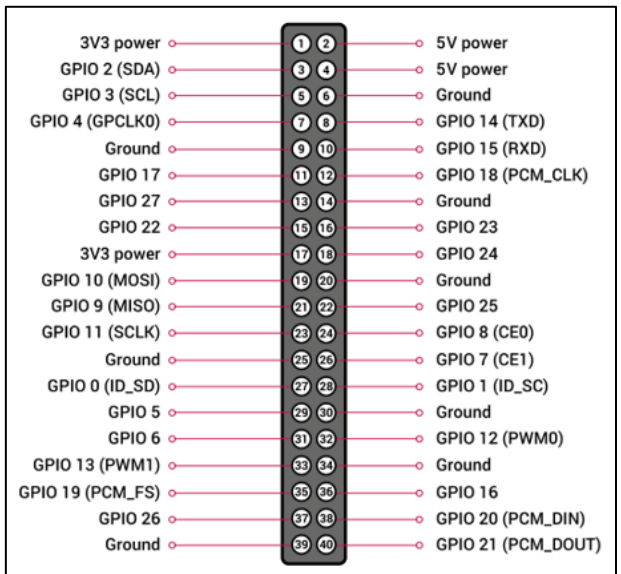
```
cmdline.txt - 記事本
檔案 編輯 檢視

console=serial0,115200 console=tty1 root=PARTUUID=8da8f3bf-02 rootfstype=ext4 fsck.repair=yes rootwait quiet init=/usr

第 2 行，第 1 欄 | 100% | Unix (LF) | UTF-8
```

**Step 3**

將兩個檔案修改好之後，SD 卡插入樹莓派，並連接好 TTL 線。



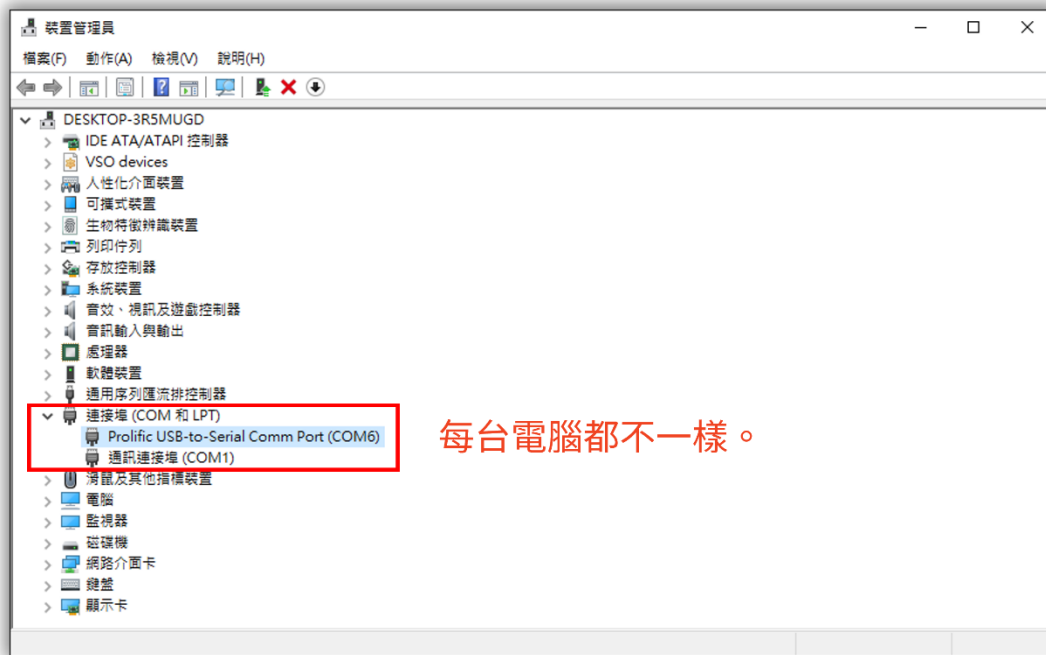
按接線，連接樹莓派與 TTL 線。



Pin06：黑線  
Pin08：白線  
Pin10：綠線

Step 4 在電腦上安裝 PL2303 的驅動程式 (Windows)。

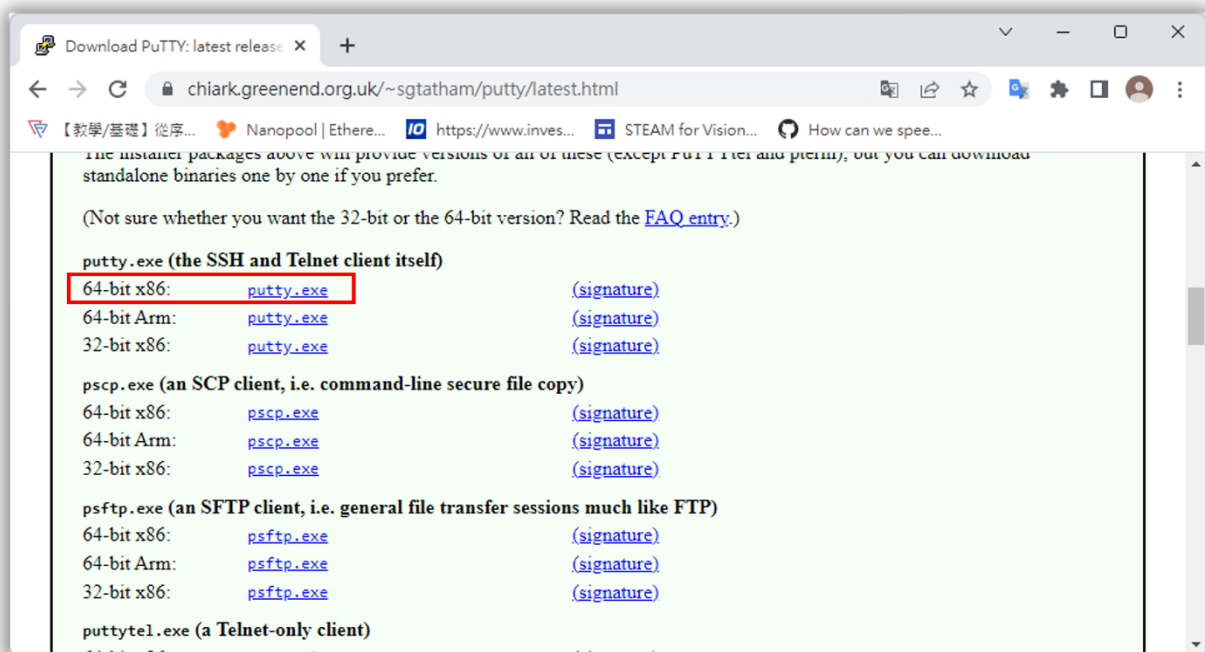
官網 [http://avrtx.cn/download/USB%20driver/PL2303/pl2303\\_332102\\_64.zip](http://avrtx.cn/download/USB%20driver/PL2303/pl2303_332102_64.zip)



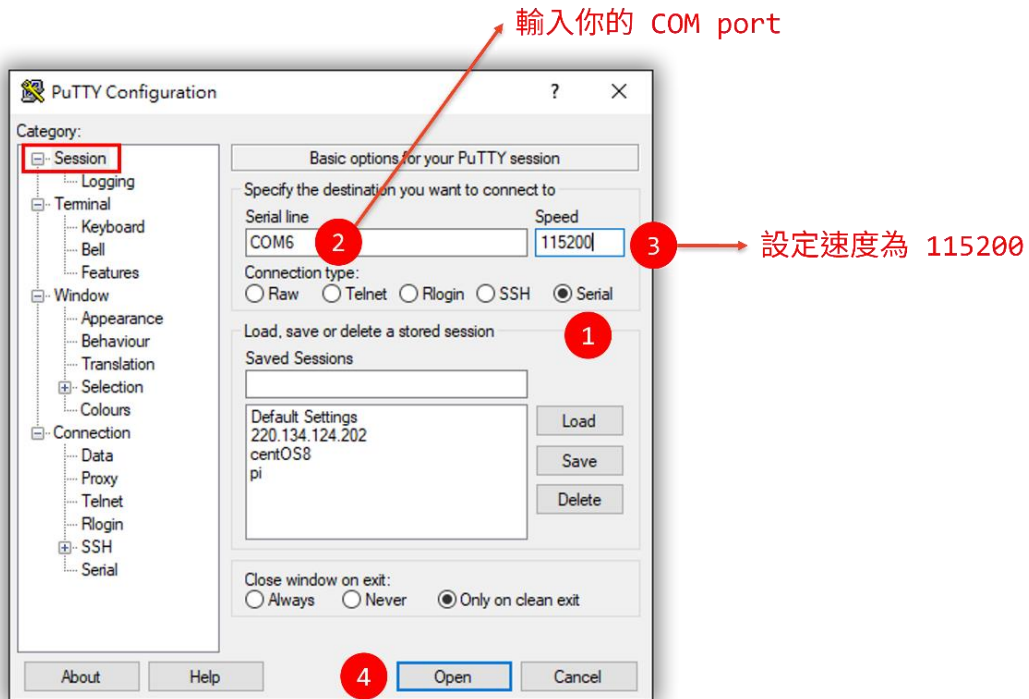
※ 記住你的 COM Port 號碼，例如：上面所示的 COM6。

Step 5 安裝 Putty，並使用 UART (TTL serial) 遠端連線。

官網 <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>



在電腦上執行 Putty，設定好如下圖的參數，並按下『 Open 』。



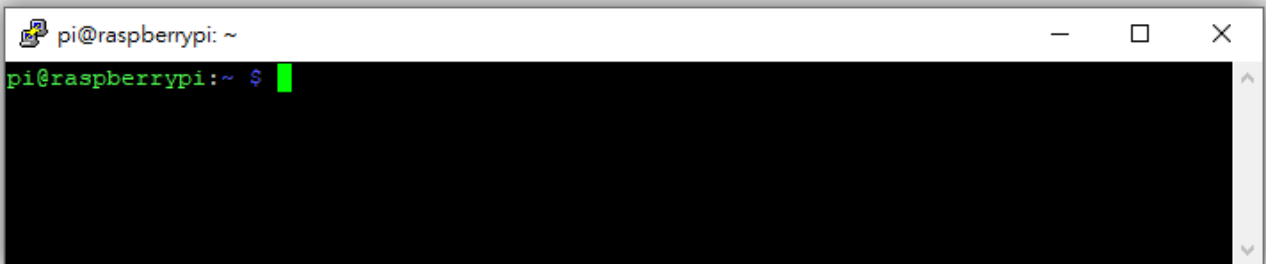
**Step 6** 若連接成功，輸入『 RASPBERRY PI OS (LEGACY) 』預設的帳號/密碼。

連接成功後，會出現『 raspberrypi login: 』的提示文字。(只有文字介面可操作)



## 基本操作

### 登入畫面的符號說明



```
pi@raspberrypi: ~  
pi@raspberrypi:~ $
```

- pi：目前登入的使用者。
- @：表示 "在"。
- raspberrypi：是預設的主機名稱。
- ~：表示在家目錄 (home directory)，預設是在 `home/pi`。
- \$：表示該使用者所使用的 shell (一種文字工具介面)。

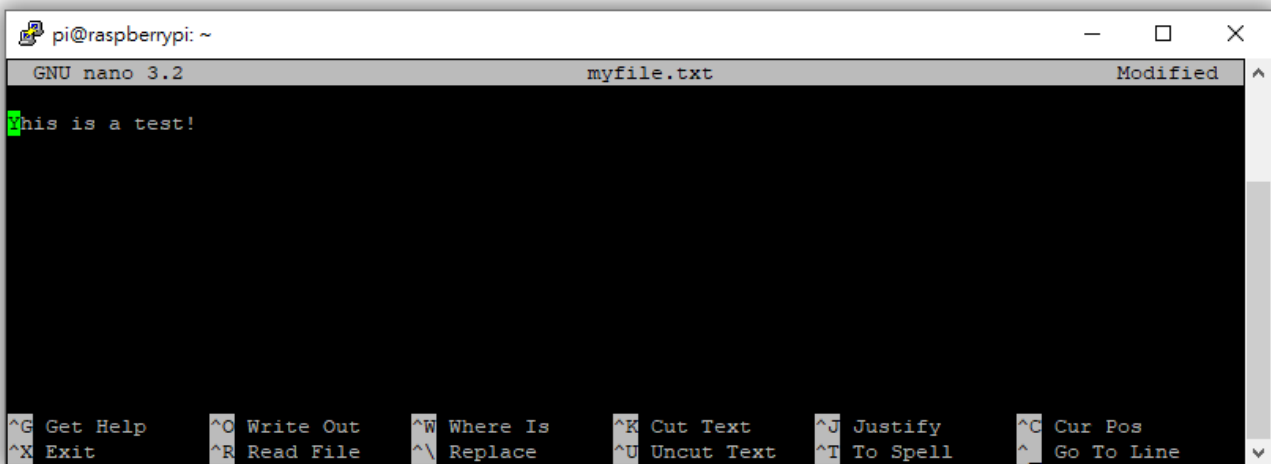
### 編輯檔案：使用 nano

在家目錄下，輸入：

```
~$ nano myfile.txt
```

- 表示用 nano 編輯器，在家目錄下開啟舊檔或新增 `myfile.txt` 檔案。

在 nano 中，輸入文字：



```
GNU nano 3.2 myfile.txt Modified  
this is a test!  
^G Get Help    ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos  
^X Exit        ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

- 不存檔離開：Ctrl + x
- 儲存檔案：Ctrl + o (若離開需要再次 Ctrl + x)

## Linux 常用的指令

### 基本指令

- ~\$ `ls`：列出目前所有的檔案/目錄。 ( ~\$ `ls -al`：為列出檔案的詳細資訊)
- ~\$ `pwd`：顯示當前所在的目錄位置 (絕對路徑)。
- ~\$ `mkdir 目錄名稱`：新增一個目錄。 (例如：~\$ `mkdir ch3`)
- ~\$ `rm -rf 目錄名稱`：刪除一個目錄。 (例如：~\$ `rm -rf ch3`)
- ~\$ `rm 檔案名稱`：刪除一個檔案 (含副檔名)。 (例如：~\$ `rm myfile.txt`)
- ~\$ `cd 目錄路徑`：變更目錄到指定位置。 (例如：~\$ `cd ch3`)
- ~\$ `cd ..`：回到到上一層。
- ~\$ `cd ~`：直接到家目錄的位置。 (預設的家目錄在 `/home/pi`)
- ~\$ `sudo shutdown now`：關機。 ( `sudo` 為取得 `root` 權限)
- ~\$ `sudo reboot`：重新啟動。

### 好用的快速鍵

- 『 ↑ 』 鍵：顯示曾經使用過的指令。
- 『 Tab 』 鍵：自動填指令。

### Linux 常用的系統指令

Linux 系統常用 `apt` (高級軟體工具, `Advanced Package Tool`)，來安裝軟體。前提是您的 Linux 要連上網際網路，以及取得 `root` 權限，以下 `sudo` 指令就是用來取得 `root` 權限的！

- ~\$ `sudo apt-get update` 軟體資料庫同步，升級前都先做。
- ~\$ `sudo apt-get upgrade` 升級已安裝的軟體。
- ~\$ `sudo apt-get install <軟體名稱>` 安裝某個軟體。
- ~\$ `sudo apt-get remove <軟體名稱>` 刪除某個軟體。

## 3-2. 網路設定

檢查目前的網路，輸入：

```
~$ ifconfig wlan0
```

- 若加上 wlan0，只針對無線網卡。若沒加，則列出所有的網路卡。



```
COM4 - PuTTY
pi@raspberrypi:~$ ifconfig wlan0
wlan0: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether dc:a6:32:6c:ac:0b txqueuelen 1000  (Ethernet)
    RX packets 32 bytes 4042 (3.9 KiB)
    RX errors 0 dropped 1 overruns 0 frame 0
    TX packets 57 bytes 8753 (8.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$
```

沒有 IP，無線網卡沒有連上網路



```
pi@raspberrypi:~$ ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.68.112 netmask 255.255.255.0 broadcast 192.168.68.255
    inet6 fe80::c885:6d69:604b:5bbb prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:6c:ac:0b txqueuelen 1000  (Ethernet)
    RX packets 14433 bytes 2200185 (2.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1338 bytes 205608 (200.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$
```

有 IP，無線網卡連上網路

### 設定 WiFi

未來，物聯網的神話只會有 WiFi。樹梅派有兩種方法設定 WiFi：

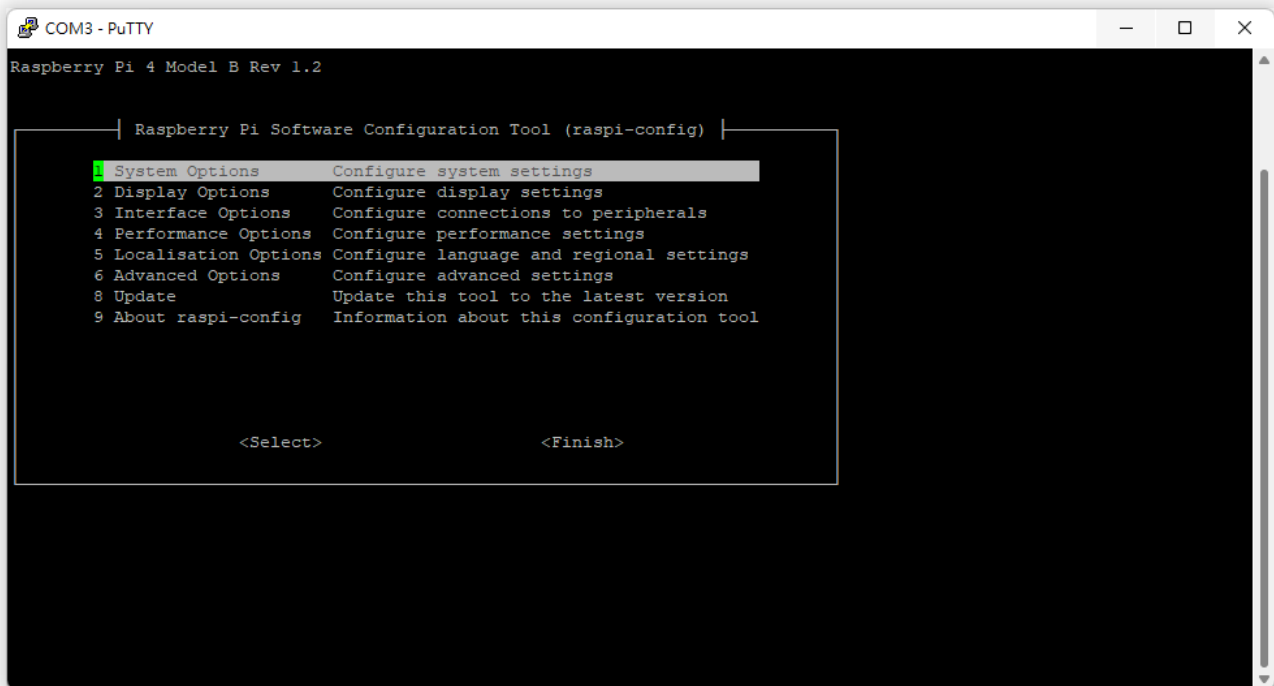
1. 使用指令 `raspi-config`，利用交談視窗設定。
2. 手動修改 `wpa-supPLICANT configuration file`。

#### 設定 WiFi 方法一：使用 `raspi-config`

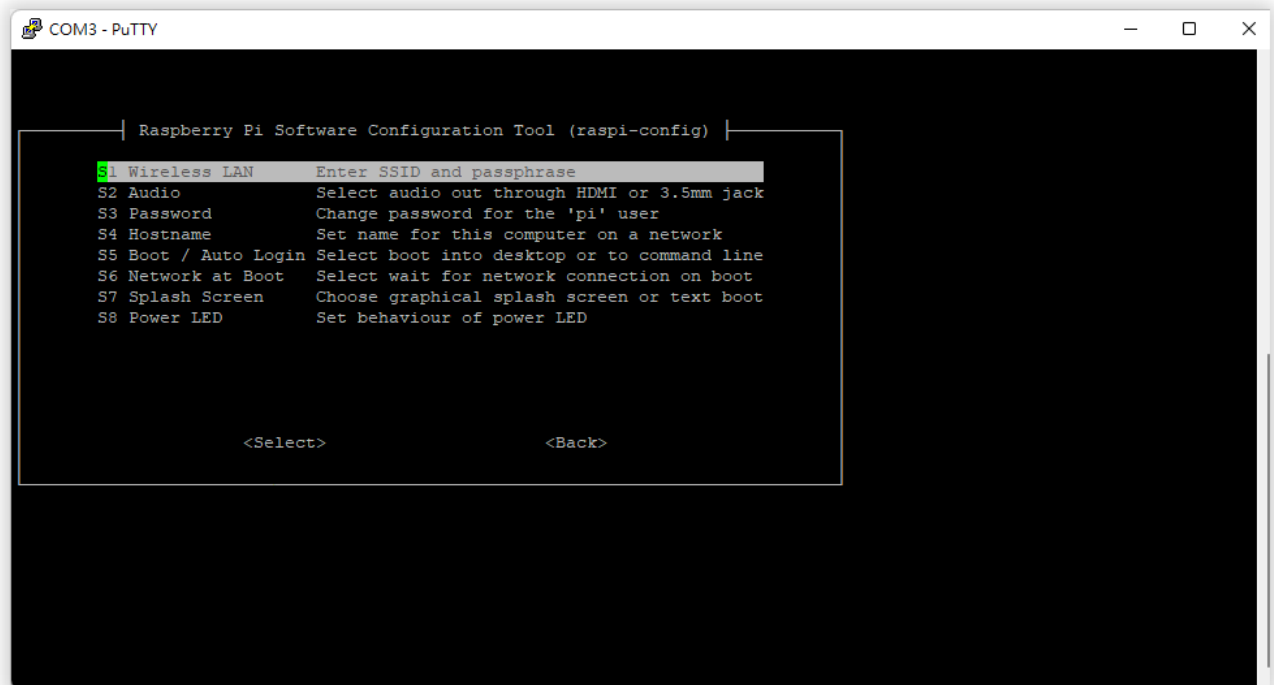
**Step 1** 輸入指令，進入設定視窗。

```
~$ sudo raspi-config
```

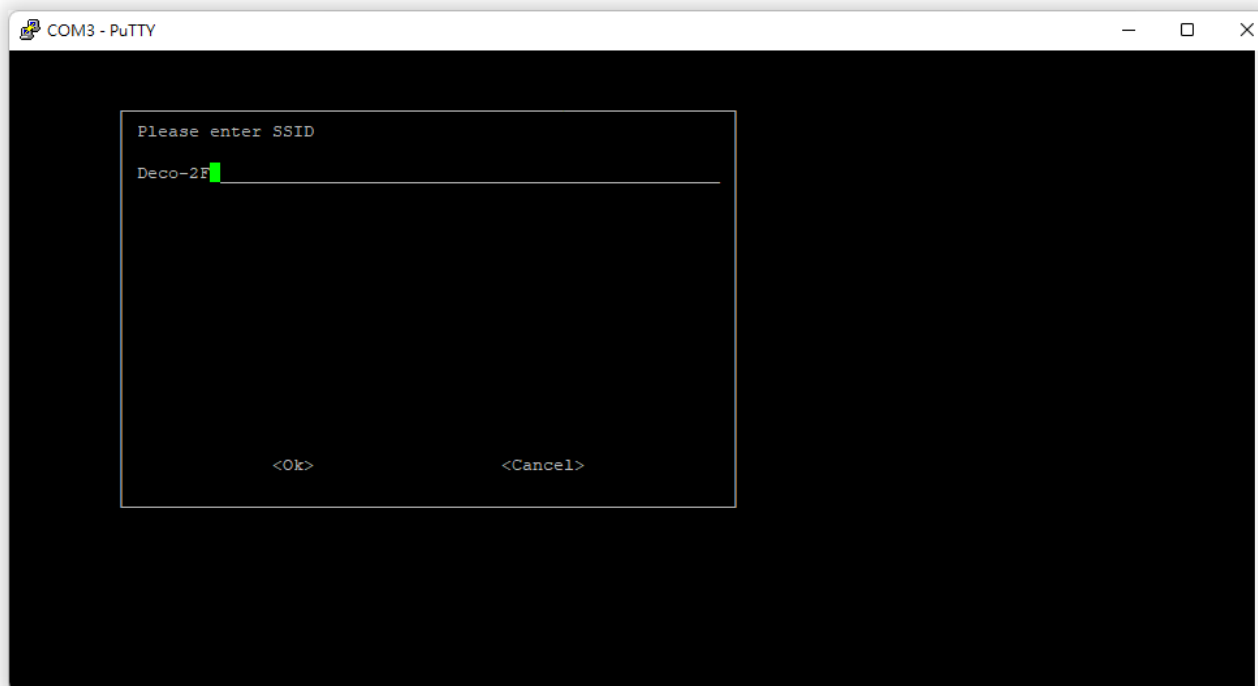
Step 2 選擇『 1 System Options 』。



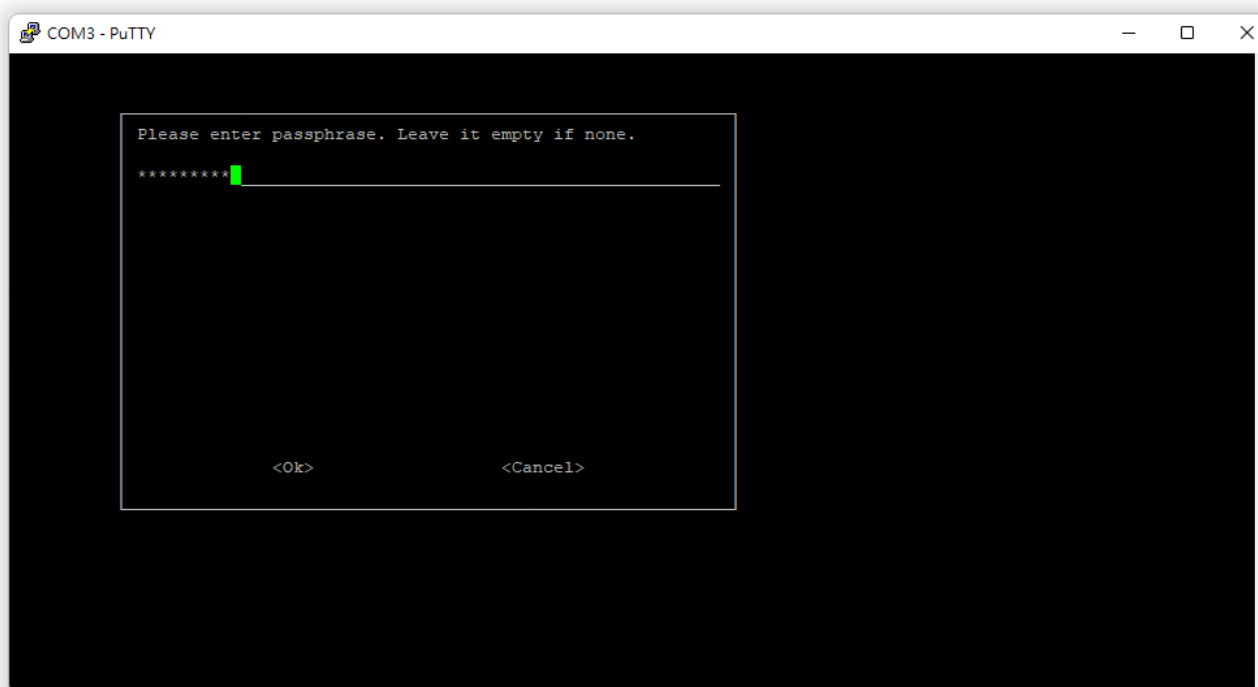
Step 3 選擇『 S1 Wireless LAN 』。



**Step 4** 輸入你的 WiFi AP 的 SSID (無線基地台的帳號)。



**Step 5** 輸入 SSID 的密碼。

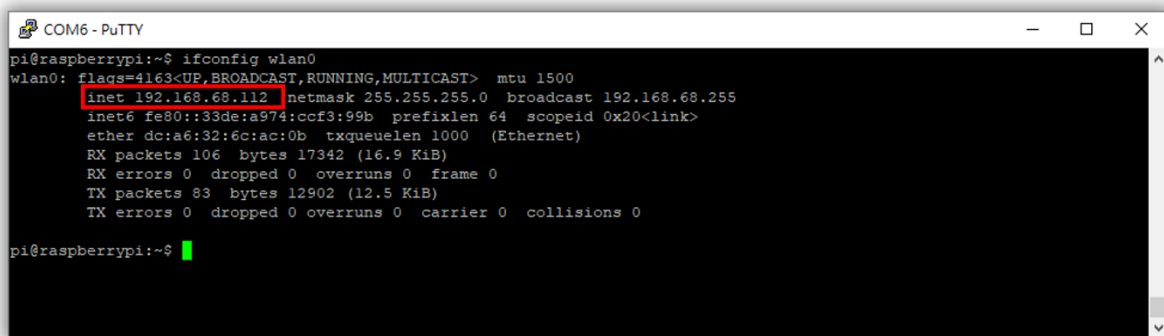


設定完成後，按下『ESC』，離開交談視窗。

## Step 6 檢查目前的網路。

再次輸入：

```
~$ ifconfig wlan0
```



```
COM6 - PuTTY
pi@raspberrypi:~$ ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.68.112 netmask 255.255.255.0 broadcast 192.168.68.255
    inet6 fe80::33de:a974:ccf3:99b prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:6c:ac:0b txqueuelen 1000 (Ethernet)
    RX packets 106 bytes 17342 (16.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 83 bytes 12902 (12.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$
```

記住你的網路位址 (IP address)，區網一般都是『192.168.x.x』。

## 設定 WiFi 方法二：手動修改 wpa-supPLICANT configuration file

### Step 1 輸入指令，修改 wpa\_supplicant.conf。

```
~$ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

設定或檢查 ssid (帳號) 與 psk (密碼) 是否正確，設定好後重新啟動樹莓派。

```
network={
    ssid="Deco-2F"
    psk="*****"
}
network={
    ssid="max543"
    psk="*****"
}
```

啟動 wlan0：

```
~$ sudo ifconfig wlan0 up
```

停止 wlan0：

```
~$ sudo ifconfig wlan0 down
```

## 遠端控制（使用 SSH/VNC）

設定 WiFi 後，可以使用 SSH 或 VNC 取代 UART/TTL Serial。

- SSH：純文字終端機介面。
- VNC：遠端桌面視窗。

建議使用 SSH。

- 純文字介面比漂亮視窗速度快很多，只是要學好命令式操作。
- 微型系統：Linux + 文字介面是王道，少人用 Windows + 視窗介面。
- SSH Server：樹莓派。
- SSH client：Windows 用戶需要安裝，例如：Putty。Mac/Linux：直接使用終端機 (terminal)。

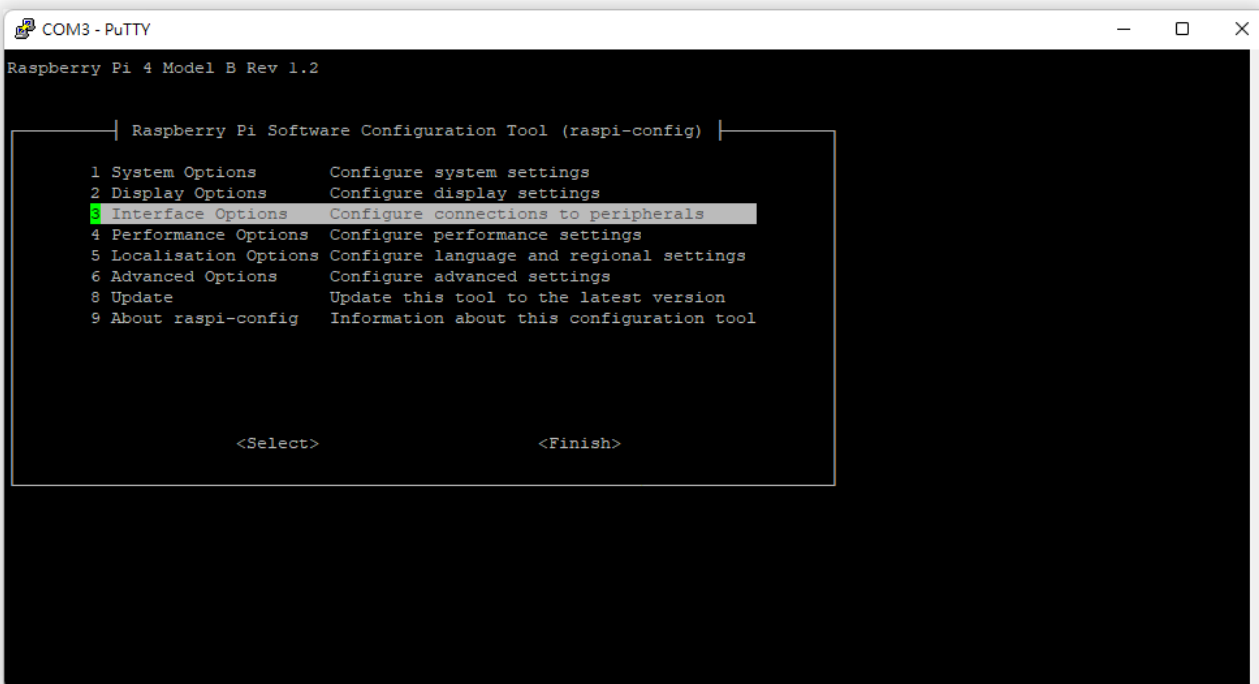
### 啟動 SSH/VNC Server

使用 SSH 連接樹莓派之前，需要先 enable SSH，這裡一併也啟動 VNC。

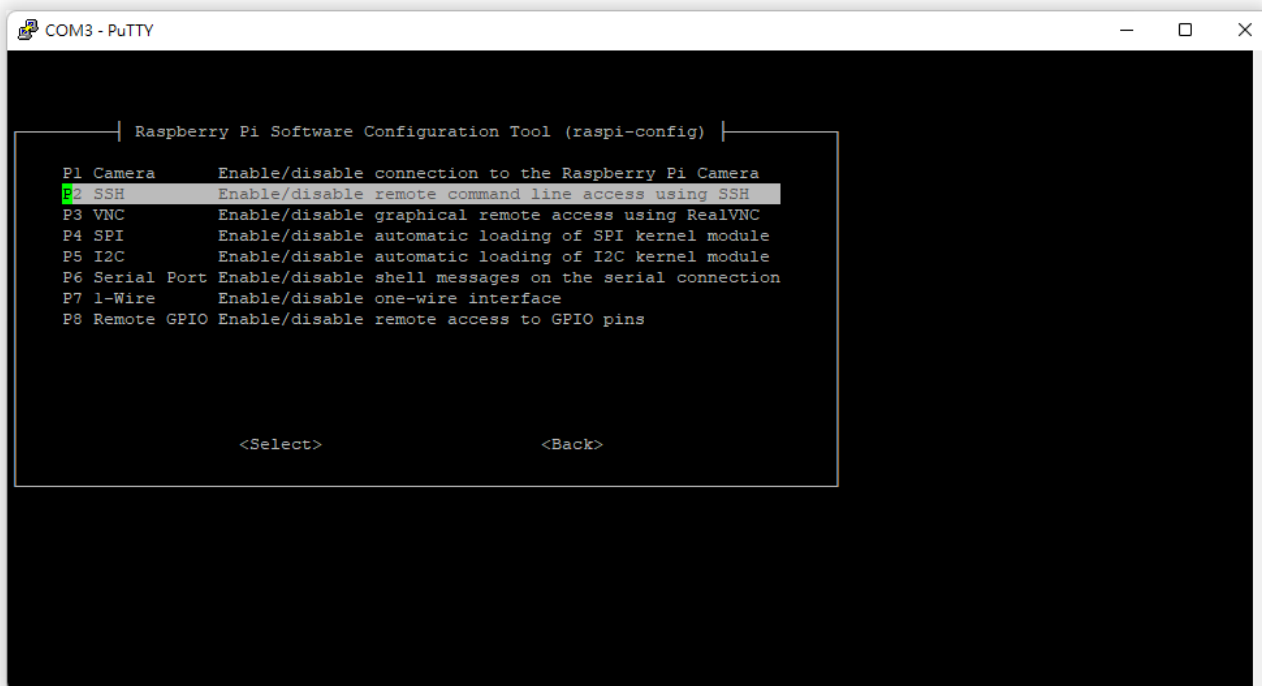
**Step 1** 輸入指令，進入設定視窗。

```
~$ sudo raspi-config
```

**Step 2** 選擇『 3 Interface Options 』。



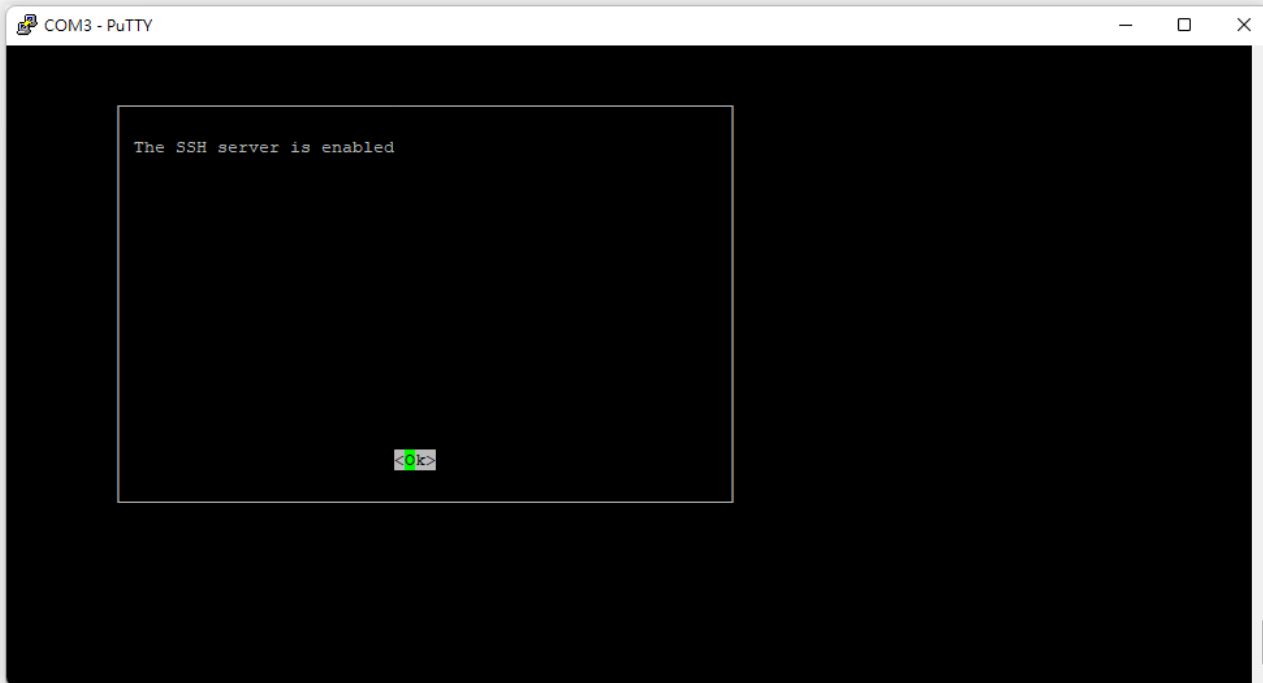
Step 3 選擇『 P2 SSH 』。



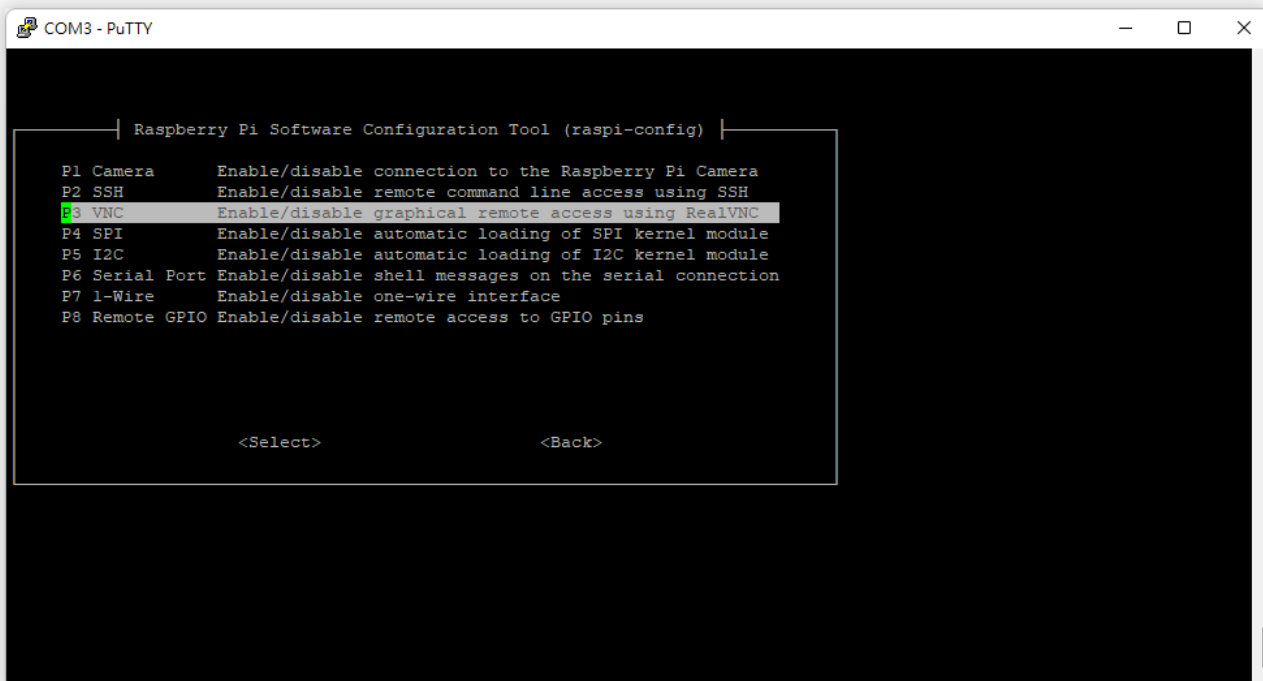
Step 4 啟動 SSH，選擇『 <Yes> 』。



Step 5 確認 SSH enabled，選擇『<Ok>』。



Step 6 與上面相同步驟，選擇『P2 VNC』啟動 VNC。(若不須遠端桌面，可不做)

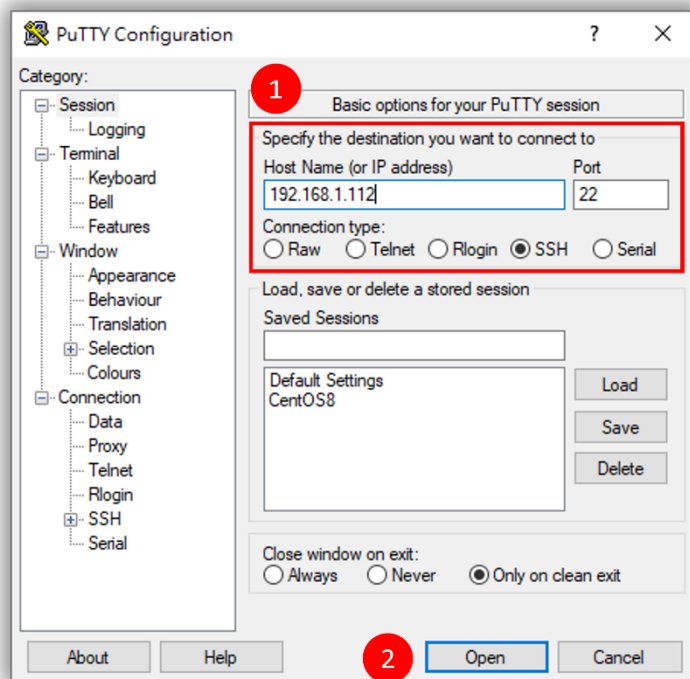


# SSH 遠端控制

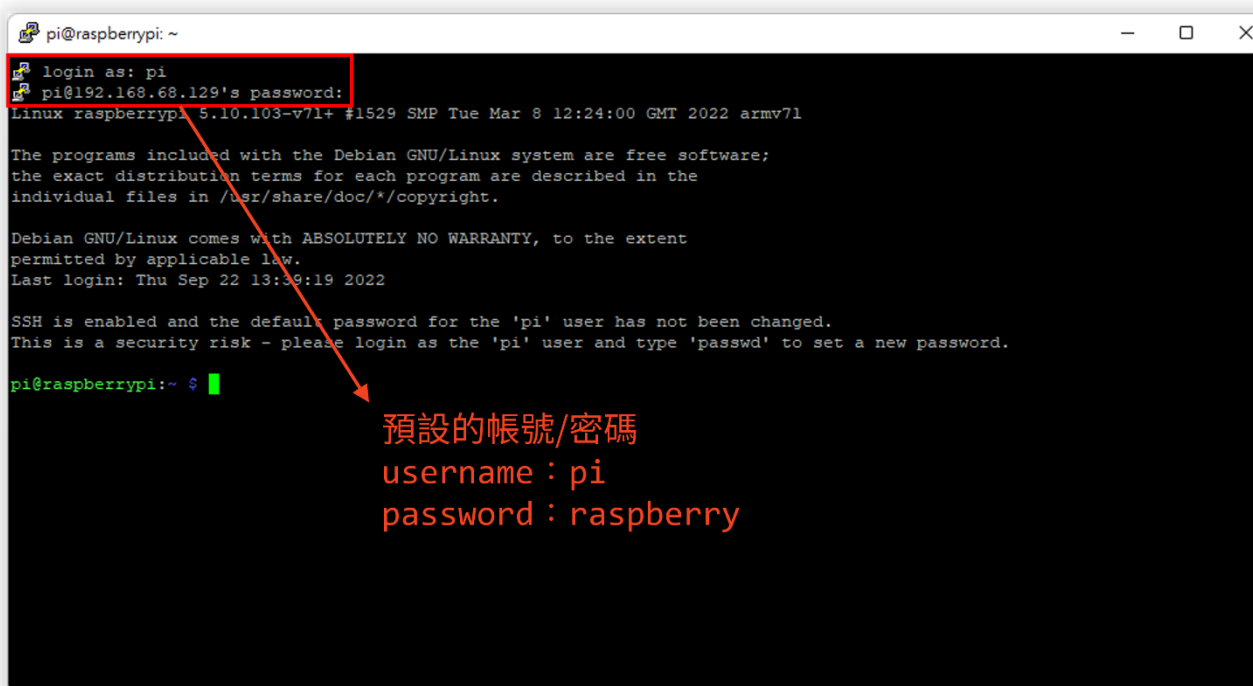
## SSH 遠端控制樹莓派 (Windows)

**Step 1** 開啟 Putty 使用 SSH 連線。

- Host Name (or IP address) : 你的樹莓派 IP 。
- Port : 預設 SSH 為 『 22 』。



**Step 2** 若連接成功，輸入預設的帳號：pi，密碼：raspberrypi。

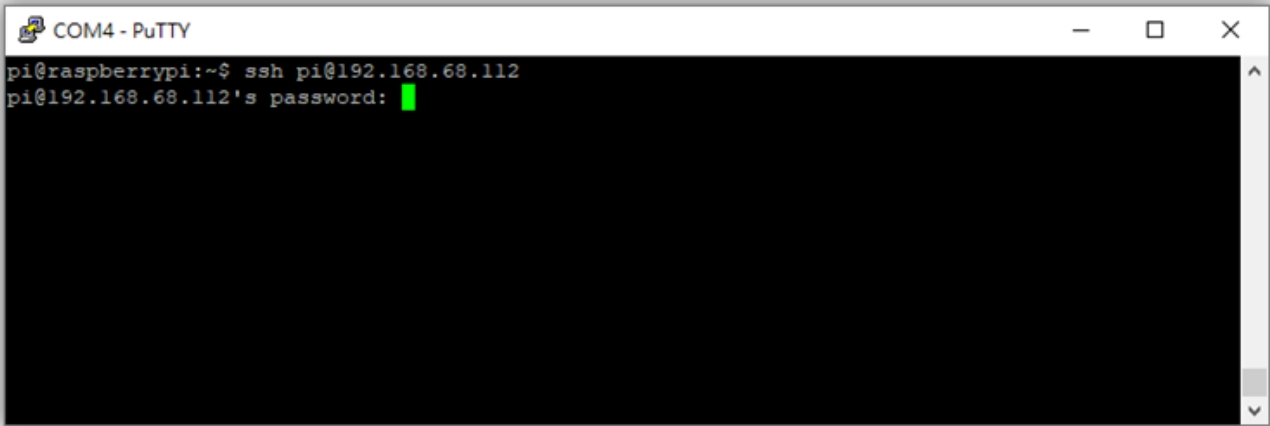


## SSH 遠端控制樹莓派 (Mac/Linux)

若你使用 Unix-like 系統 (Mac/Linux)，開啟終端機 (terminal)：

~\$ `ssh pi@'樹莓派的網路位址'` (例如：~\$ `ssh pi@192.168.1.112`)

~\$ `exit` (離開 SSH 連線)



## Serial 連線和 SSH 連線有什麼不同？

- Serial 以實體線路相連，純文字、是獨占式的連線。
- SSH 是 TCP/IP 通訊協定，透過 Ethernet 或 WiFi 連線，允許多人連線。

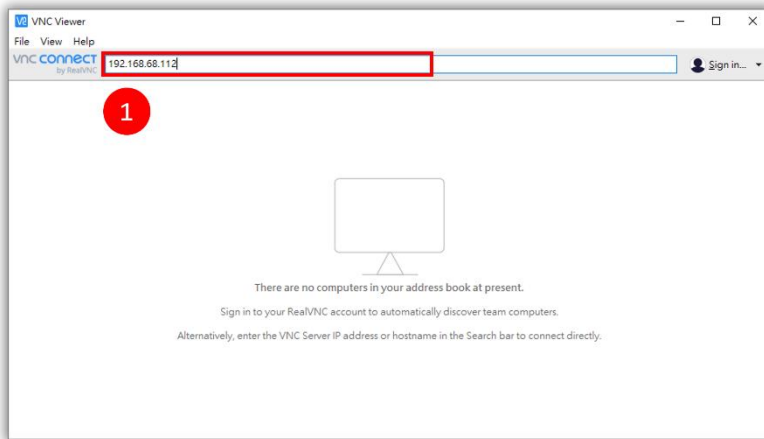
## VNC 遠端桌面

請先確認有啟動樹莓派的 VNC Server，然後在電腦上操作下列步驟：

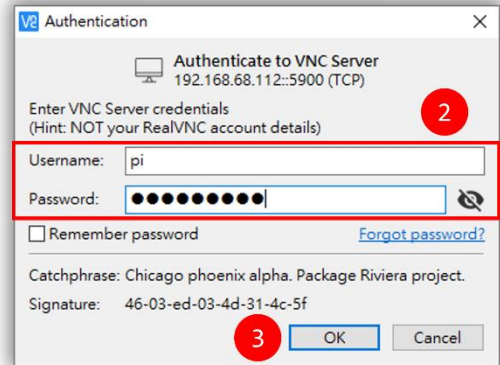
**Step 1** 下載與安裝 VNC Viewer。

**官網** <https://www.realvnc.com/en/connect/download/viewer/>

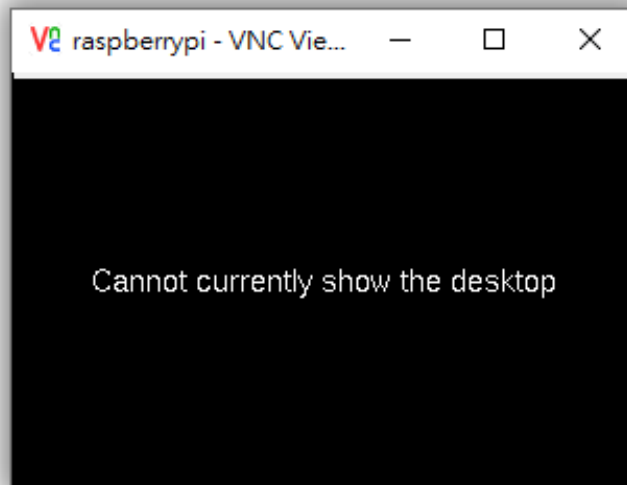
**Step 2** 執行 VNC Viewer，輸入樹莓派的登入資訊。



username : pi  
password : raspberry



**常見問題** 當出現這個錯誤時，需要調整解析度。



**Step 1** 更改更高的分辨率即可：修改樹莓派的 `/boot/config.txt`。

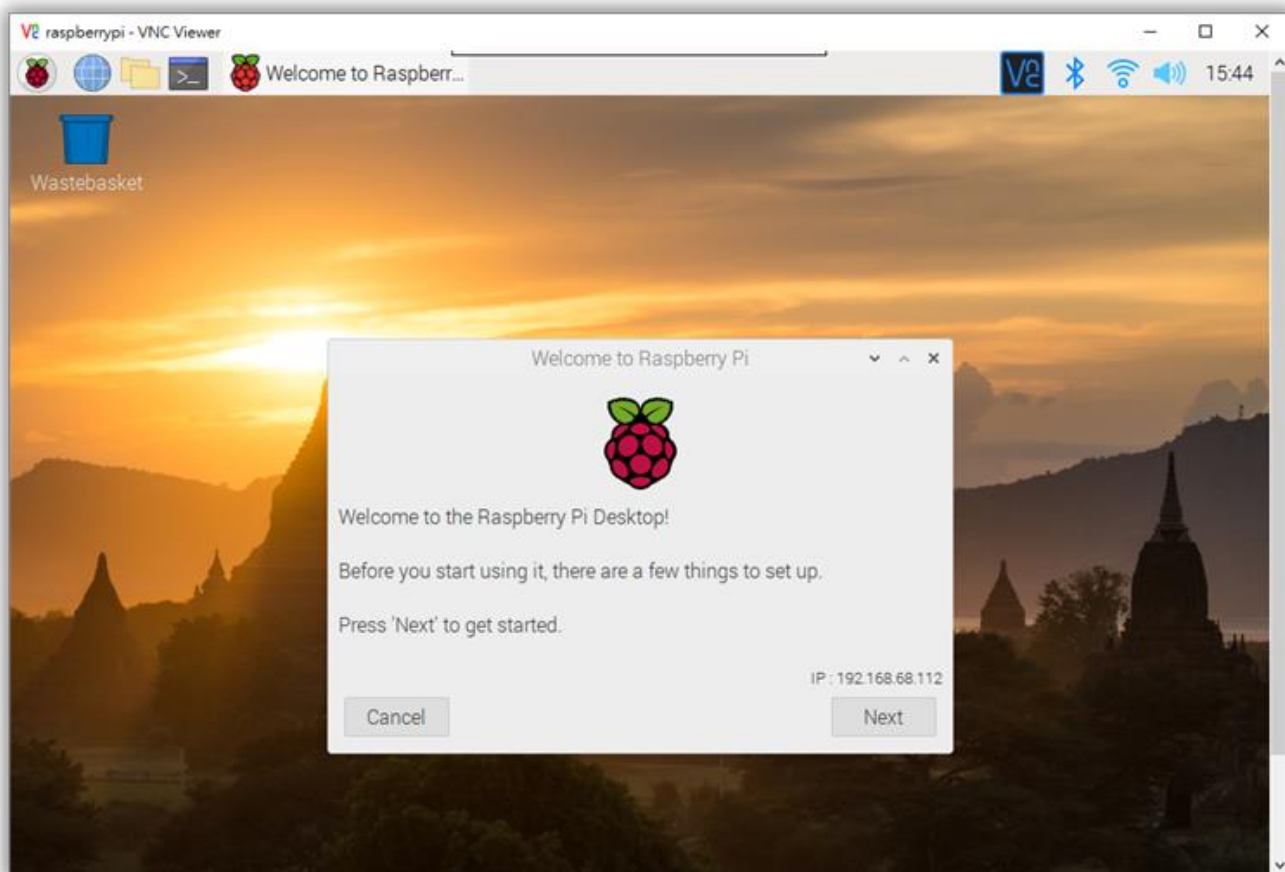
```
# uncomment if hdmi display is not detected and composite is being output
#hdmi_force_hotplug=1
# uncomment to force a specific HDMI mode (this will force VGA)
#hdmi_group=1  拿掉註解
#hdmi_mode=1
```



```
# uncomment if hdmi display is not detected and composite is being output
hdmi_force_hotplug=1
# uncomment to force a specific HDMI mode (this will force VGA)
hdmi_group=2  更改參數
hdmi_mode=16
```

**Step 2** 然後重開機。( `~$ sudo reboot` )

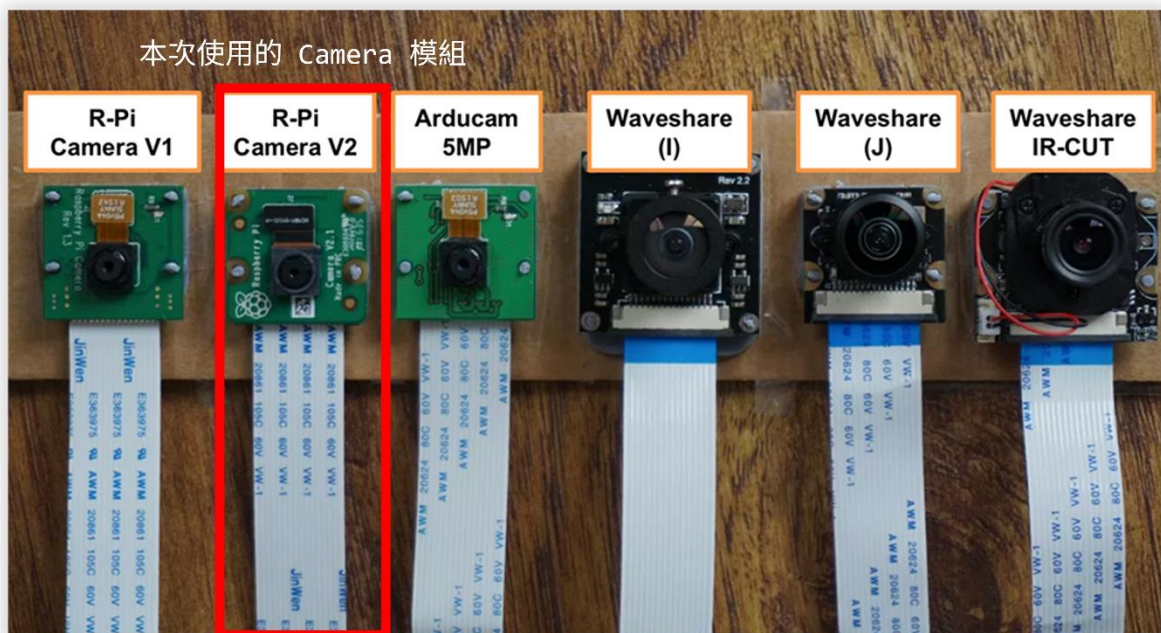
## 登入成功畫面



## 3-3. 樹莓派的相機模組

### 安裝相機模組

支援樹莓派的相機模組



### 安裝相機模組

安裝時樹莓派需要關機，輸入關機指令：

```
~$ sudo shutdown now
```

連接相機，注意金屬面要與樹莓派的金手指相連接：(確認斷電，最好拔除 USB 供電)



## 相機組態設定

### Step 1 組態設定。

進入組態設定的交談視窗：

```
~$ sudo raspi-config
```

啟動相機：

- 『 3 Interfacing Options 』 → 『 P1 Camera 』。

調整記憶體（預設 128M，可跳過）

- 『 4 Performance Options 』 → 『 P2 GPU Memory 』 → 128M

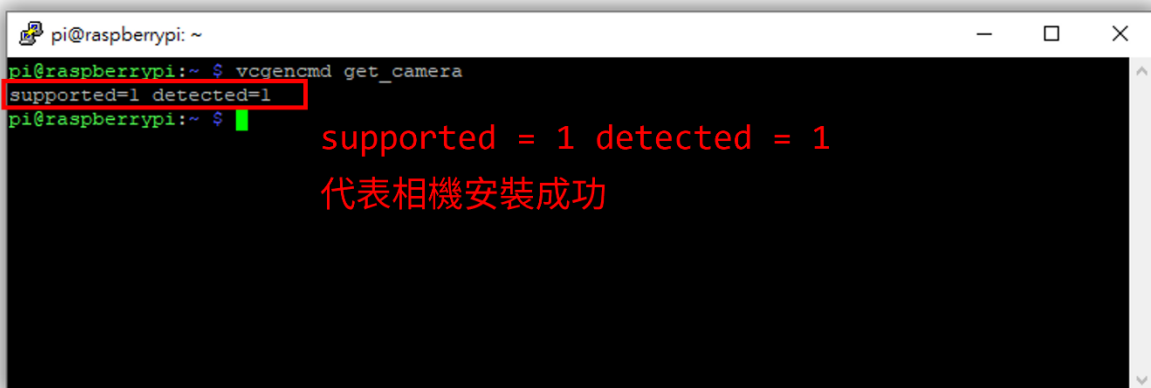
重開機：

```
~$ sudo reboot
```

### Step 2 檢查相機狀態。

輸入：

```
~$ vccgencmd get_camera
```



A terminal window on a Raspberry Pi showing the command `vccgencmd get_camera` being executed. The output is `supported=1 detected=1`, which is highlighted with a red box. Below the terminal output, there is a red text overlay that reads: `supported = 1 detected = 1` and `代表相機安裝成功` (representing successful camera installation).

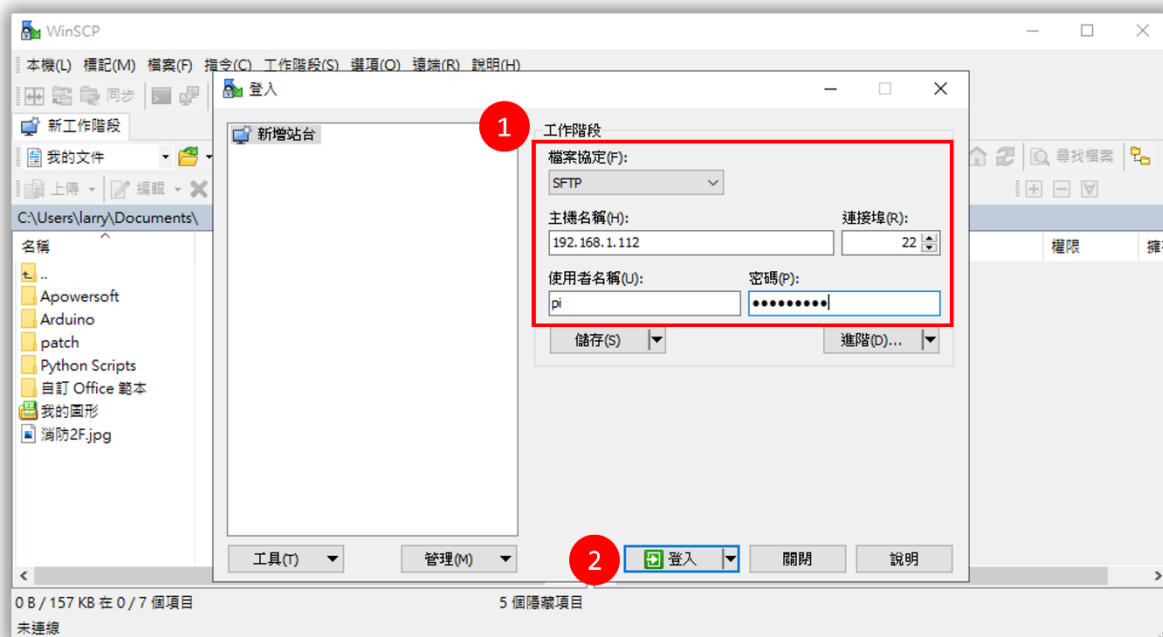
## 如何開啟影像檔

### 方法一：將樹莓派的檔案傳回本機端（遠端檔案管理）

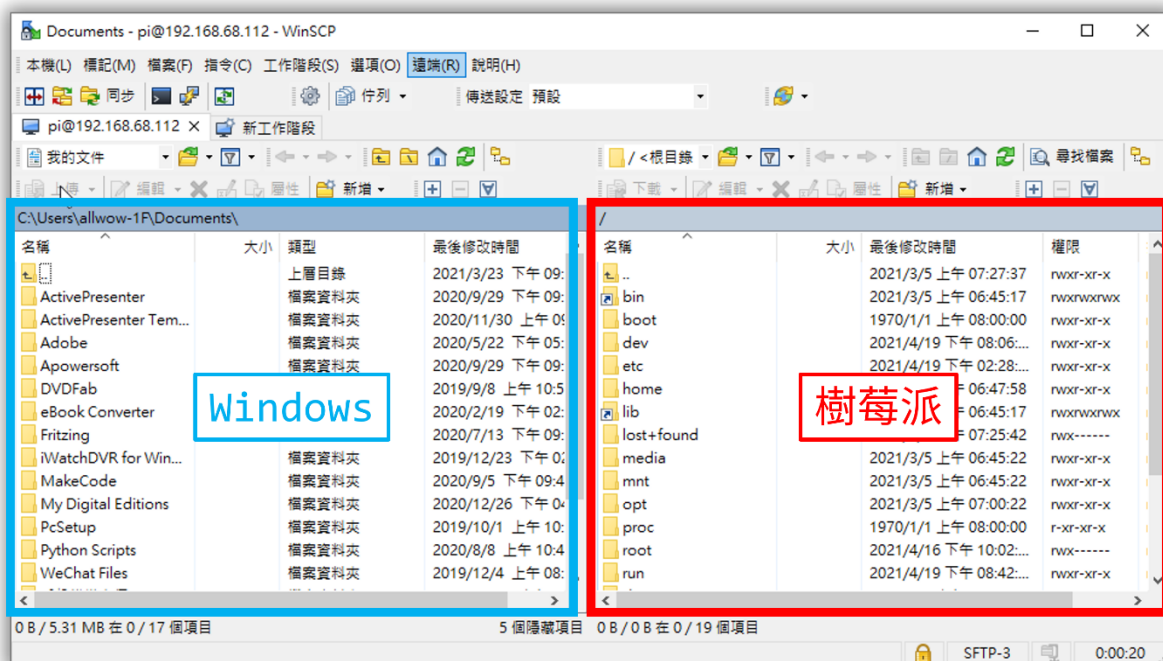
上傳/下載可使用 WinSCP，參數設定如下圖所示：

官網

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



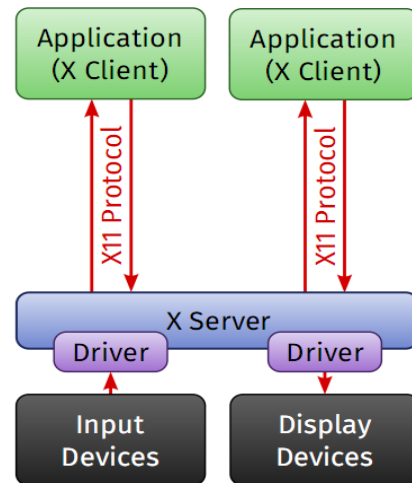
登入後，右邊為電腦磁碟區，左邊為樹莓派的磁碟區，請自行操作。



## 方法二：使用 X11 Forwarding

使用 X Window System：

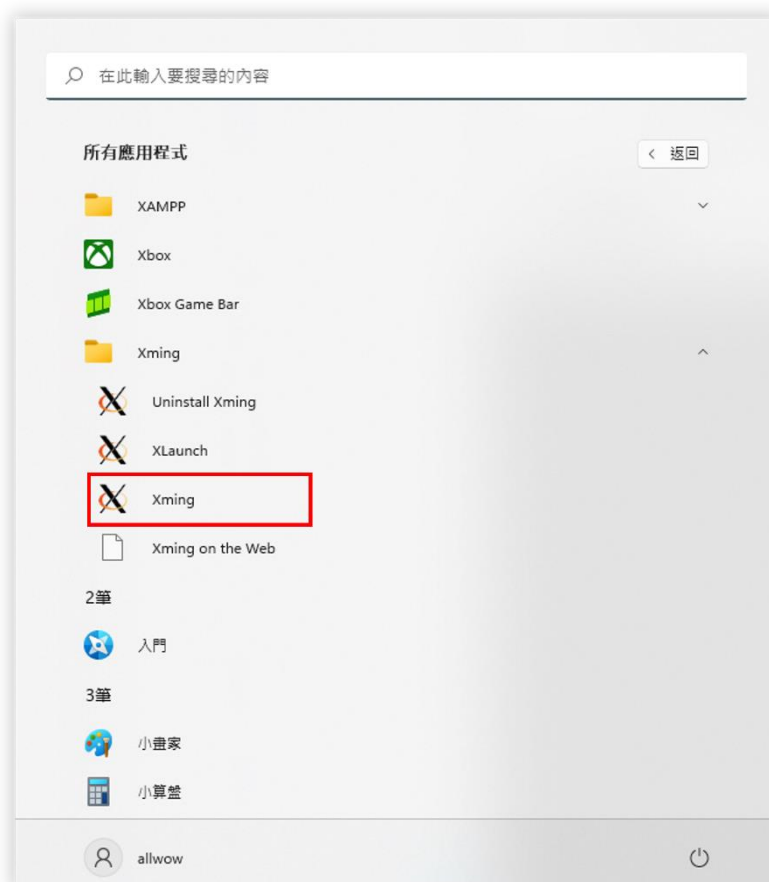
- 是一種圖形應用標準。
- Client/Server 架構。
  - ✓ X Client：應用程式。
  - ✓ X Server：管理硬體輸入/輸出。
- 可透過網路傳輸。
  - ✓ TCP/IP 或是 Unix Domain Socket
- X11 是通訊協定名稱。



**Step 1** 安裝 Xming。

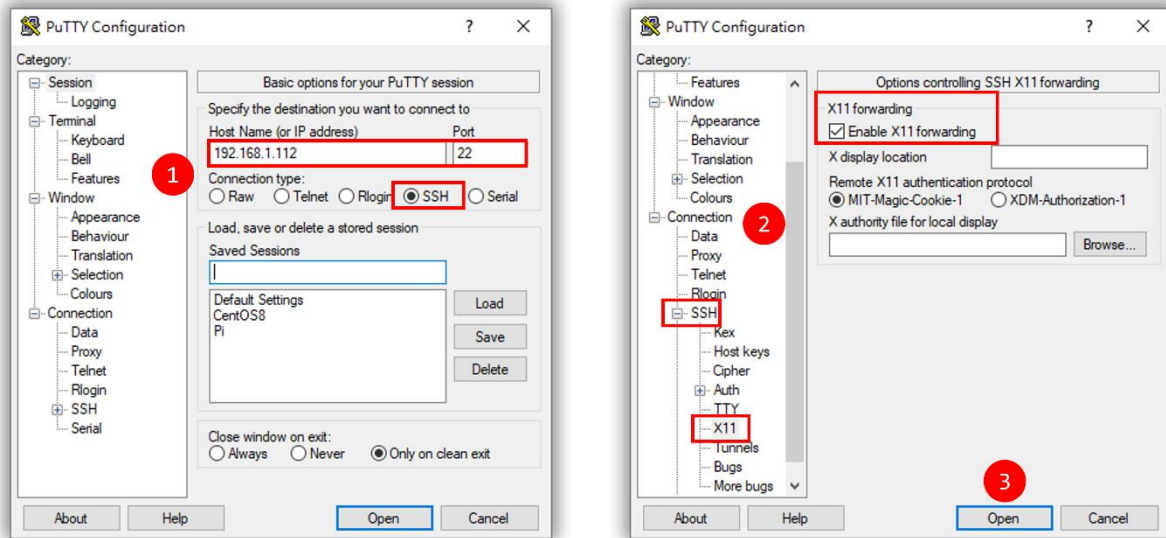
**官網** <https://sourceforge.net/projects/xming/>

**Step 2** 在電腦上啟動 Xming，當作 X Server。



### Step 3 在 Windows 的 SSH Client 上，設定 X11 Forwarding。

- 選擇『SSH』→『X11』→『Enable X11 forwarding』後，再按下 Open。



如果是 Mac/Linux，新增 `-X` 的參數：

```
~$ ssh -X pi@'樹莓派的網路位址'
```

### Step 4 設定好 SSH 後，執行 Putty 登入。

下載一張照片檔：

```
~$ wget https://fchart.github.io/img/koala.png
```

執行 `gpicview`：

```
~$ gpicview koala.png &
```

- `&` 參數：允許該命令在背景執行。

**常見問題** 無法開啟：檢查樹莓派的 `/etc/ssh/sshd_config`。

```
~$ sudo nano /etc/ssh/sshd_config
```

- 修改 `# X11Forwarding no` 這行。
- 把 `no` 改成 `yes`，並且把註解 `#` 拿掉 → `X11Forwarding yes`。(預設已修改，請確認)

**常見問題** 無法開啟：檢查電腦本機上的 `Xming` (X Server)，有沒有正在執行。

## 常見問題

MAC 無法開啟："Can not open display" for MAC

檢查樹莓派的 `/etc/ssh/sshd_config`：與上述的方式相同。

在 MAC 上，下載安裝 Xquartz 並重開機。

## 官網

<http://xquartz.macosforge.org/landing/>

## 3-4. 在樹莓派上編輯 Python

### 樹莓派的 Python

Raspberry Pi OS 預設安裝 Python 3 的直譯器。

查詢 CPU 的版本：

```
~$ uname -m  
aarch64 # aarch64 為 64 位元 ARM 架構，armv7l 為 32 位元
```

- aarch64 表示 64 位元，OpenCV 專案可執行，Yolov5 要安裝 armv7l 的 torch。

查詢 OS 的版本：

```
~$ file /bin/bash  
/bin/bash: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV),  
dynamically linked, interpreter  
/lib/ld-linux-armhf.so.3, BuildID[sha1]=f12e6d40fb262ad0037b6  
ec43162208b76d4da71, for GNU/Linux 3.2.0, stripped
```

- ELF 64-bit：表示 OS 是 64 位元。
- ELF 32-bit：表示 OS 是 32 位元。(請確認你裝的是 32 位元，不然 torch 裝不了)

查詢 Python 的版本：

```
~$ python -V  
Python 3.9.2 # 顯示目前的 Python 3 版本
```

進入 Python 的互動模式：

```
~$ python
```

進入後，會出現 `>>>` 代表互動模式。使用 `exit()` 離開該模式：

```
>>> exit()
```

## 編輯 Python 程式

### 方法一：使用 nano

在家目錄下建立 ch3 資料夾，並進入該資料夾：

```
~$ mkdir ~/ch3 && cd ~/ch3
```

新增一個 Python 的程式檔：

```
~/ch03$ nano 3-1.py
```

**3-1** 暫停 5 秒後，印出 'Hello World!'。

```
1. # 暫停 5 秒後，印出 Hello World!
2. from time import sleep
3.
4. sleep(5)
5. print('Hello World!')
```

透過 nano 撰寫 3-1 並儲存後，用 python 執行。

```
~/ch3$ python 3-1.py
```

### 方法二：使用 mousepad 編輯器

**Step 1** 安裝 mousepad。

更新軟體資料庫：

```
~$ sudo apt-get update
```

更新已安裝的軟體：

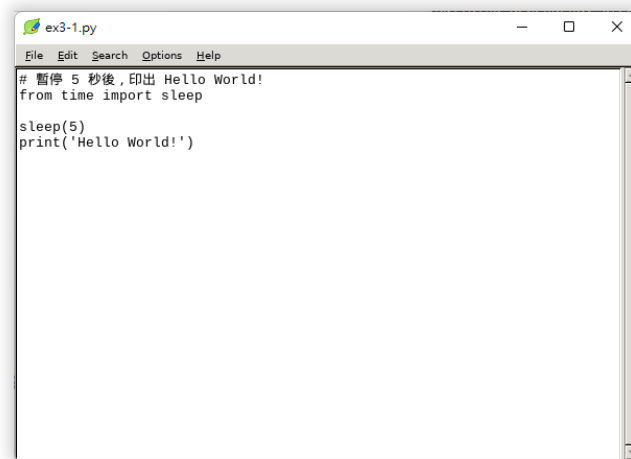
```
~$ sudo apt-get upgrade
```

安裝 mousepad：

```
~$ sudo apt-get install mousepad
```

**Step 2** 執行 mousepad，編寫程式並儲存。

```
~$ mousepad &
```



```
ex3-1.py
File Edit Search Options Help
# 暫停 5 秒後，印出 Hello World!
from time import sleep

sleep(5)
print('Hello World!')
```

同樣的，撰寫 **ex3-1** 的程式碼並儲存後，執行 `python ex3-1.py`。

### 方法三：使用 thonny 編輯器

**Step 1** 安裝 thonny。(預設 Raspberry Pi OS 已安裝)

安裝 thonny：

```
~$ sudo apt-get install thonny
```

**Step 2** 執行 thonny，編寫程式並儲存。

執行 thonny 後，與前面方法類似，請自行操作：

```
~$ thonny &
```



## 3-4. 執行 OpenCV 專案

### 建立與管理 Python 虛擬環境

Python 虛擬環境可以針對不同 Python 專案建立專屬的開發環境。

- Raspberry Pi OS 可以使用 `venv` 或 `virtualenv` 來建立、啟動、刪除與管理 Python 虛擬環境。
- 本次是使用 `virtualenv` + `virtualenvwrapper` 套件。

### 安裝 `virtualenv`

目前的 Raspberry Pi OS 安裝 Python 3.9.2。

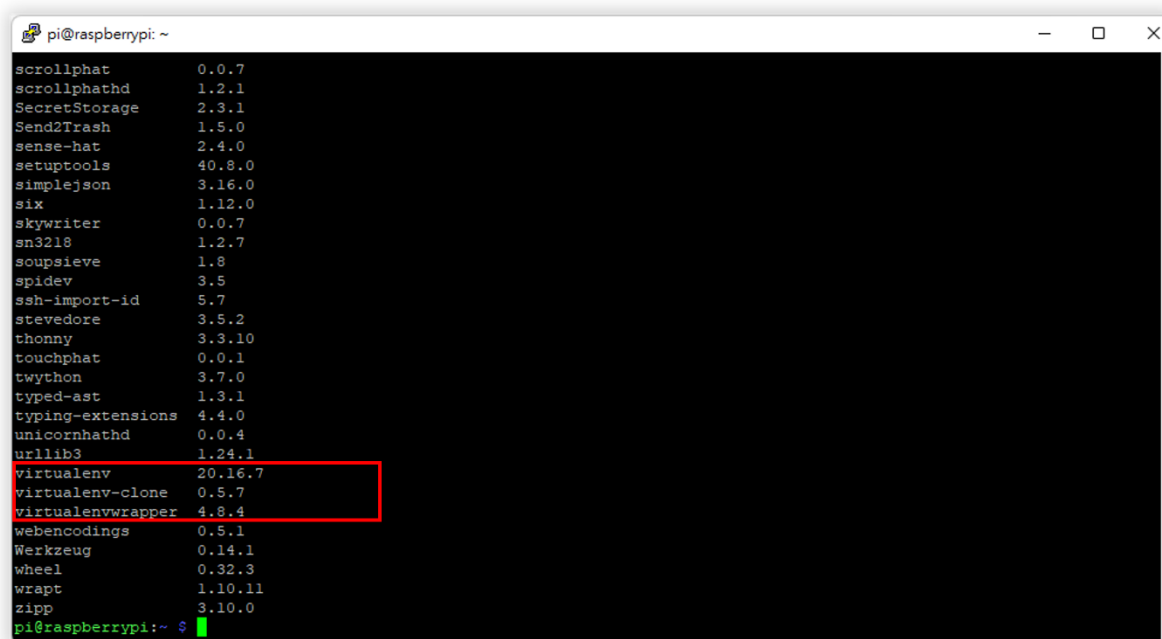
**Step 1** 以 `virtualenv` 安裝為例。(需要加 `sudo`，因為會動到系統檔案)

```
~$ sudo pip install virtualenv
```

```
~$ sudo pip install virtualenvwrapper
```

確認是否安裝成功：

```
~$ pip3 list
```



```
pi@raspberrypi: ~  
scrollphat 0.0.7  
scrollphatd 1.2.1  
SecretStorage 2.3.1  
Send2Trash 1.5.0  
sense-hat 2.4.0  
setuptools 40.8.0  
simplejson 3.16.0  
six 1.12.0  
skywriter 0.0.7  
sn3218 1.2.7  
soupsieve 1.8  
spidev 3.5  
ssh-import-id 5.7  
stevedore 3.5.2  
thonny 3.3.10  
touchphat 0.0.1  
twython 3.7.0  
typed-ast 1.3.1  
typing-extensions 4.4.0  
unicornhathd 0.0.4  
urllib3 1.24.1  
virtualenv 20.16.7  
virtualenv-clone 0.5.7  
virtualenvwrapper 4.8.4  
webencodings 0.5.1  
Werkzeug 0.14.1  
wheel 0.32.3  
wrapit 1.10.11  
zipp 3.10.0  
pi@raspberrypi:~$
```

**Step 2** 編輯 `.bashrc` 檔，新增路徑設定，以便可以執行 `virtualenvwrapper`。

輸入：

```
~$ sudo nano ~/.bashrc
```

加入下列三行到 `.bashrc`：

```
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
source /usr/local/bin/virtualenvwrapper.sh
```

重新載入更新的路徑設定：

```
~$ source ~/.bashrc
```

## 建立與管理 Python 3 的 OpenCV 虛擬環境

完成相關套件的安裝和設定後，準備新增名為 `opencv` 的 Python 虛擬環境。

```
~$ mkvirtualenv opencv -p python3
```

- 參數 `-p` 指定 Python 版本是 Python 3。

關閉目前虛擬環境：

```
~$ deactivate
```

顯示已建立的虛擬環境：

```
~$ lsvirtualenv
```

啟動虛擬環境：當成功新增 `opencv` 虛擬環境後，使用 `workon` 指令來啟動虛擬環境：

```
~$ workon opencv
```

移除虛擬環境：

```
~$ rmvirtualenv opencv
```

複製虛擬環境 (opencv 為來源，opencv\_new 為目的)：

```
~$ cpvirtualenv opencv opencv_new
```

## 安裝 OpenCV 所需模組

**Step 1** 更新與升級作業系統。

軟體資料庫同步，升級前都先做：

```
~$ sudo apt-get update
```

升級已安裝的軟體：

```
~$ sudo apt-get update
```

**Step 2** 安裝 OpenCV 相關的 OS 系統套件。

- 非 Python 模組，需要 sudo 的權限。

安裝 HDF (Hierarchical Data Format version 5) 資料集的套件，這是支援大型、複雜和異質資料的開源檔案格式，可以用來儲存大量的圖片資料：

```
~$ sudo apt-get install -y libhdf5-dev
```

安裝矩陣運算和 JPEG-2000 圖片函式庫：

```
~$ sudo apt-get install -y libatlas-base-dev
```

```
~$ sudo apt-get install -y libopenjp2-7-dev
```

安裝 QT GUI 圖形介面的套件：

```
~$ sudo apt-get install -y libqt5gui5
```

```
~$ sudo apt-get install -y libqt5test5
```

```
~$ sudo apt-get install -y python3-pyqt5
```

### Step 3 建立 Opencv 的虛擬環境，若沒有的話。

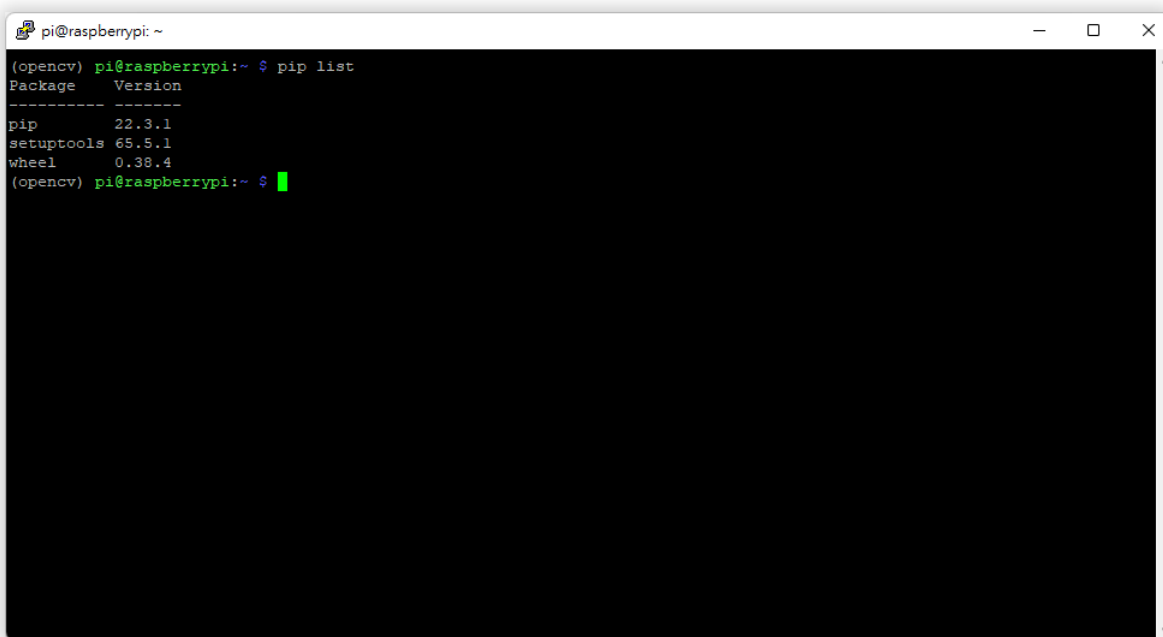
新增名為 opencv 的 Python 3 虛擬環境：

```
~$ mkvirtualenv opencv -p python3
```

檢視目前虛擬環境的模組：

```
(opencv) ~$ pip list
```

- 虛擬環境指定的是 Python 3，只要使用 pip 就可以了。



```
pi@raspberrypi: ~  
(opencv) pi@raspberrypi:~$ pip list  
Package      Version  
-----  
pip          22.3.1  
setuptools   65.5.1  
wheel        0.38.4  
(opencv) pi@raspberrypi:~$
```

### Step 4 在虛擬環境安裝 OpenCV 需要的相依套件。

在虛擬環境安裝支援的 Python 套件，numpy 為陣列容器、matplotlib 為 Python 及其數值計算庫 numpy 的繪圖庫、imutils 可以讓我們更容易使用 OpenCV 圖片處理：

```
(opencv) ~$ pip install numpy==1.21.1
```

- 不指定 numpy 的版本，會安裝最新的版本，若不支援 OpenCV 則會發生錯誤。

```
(opencv) ~$ pip install matplotlib
```

```
(opencv) ~$ pip install imutils
```

## Step 5 在 Python 虛擬環境安裝 OpenCV。

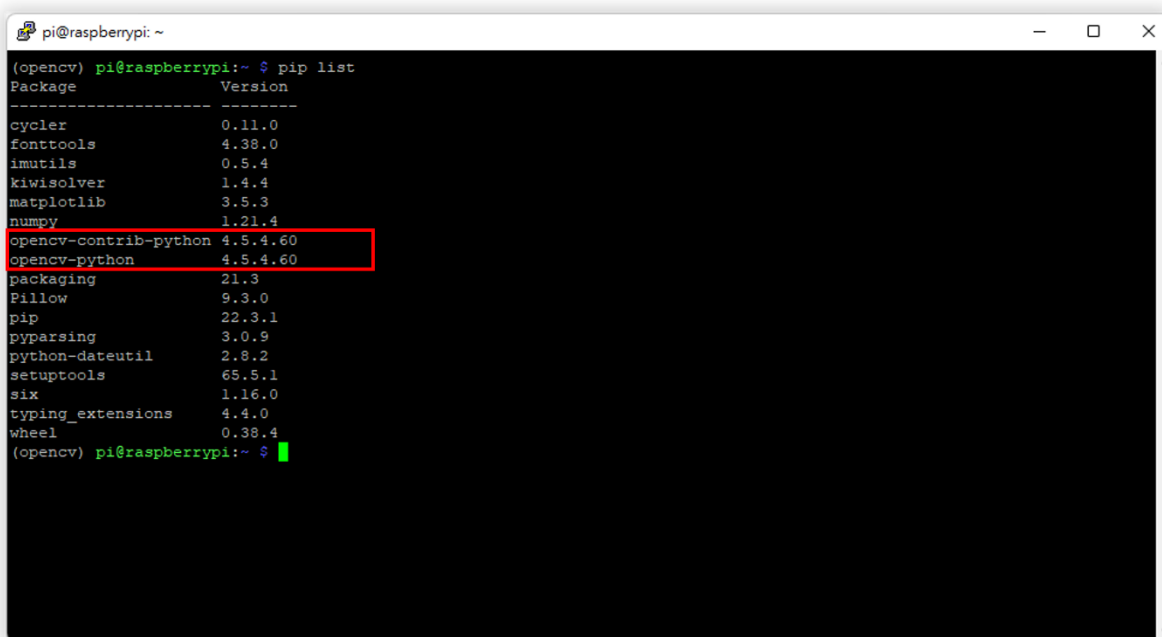
使用安裝 OpenCV，同樣需要版本：

```
(opencv) ~$ pip install opencv-python==4.5.1.48
```

```
(opencv) ~$ pip install opencv-contrib-python==4.5.1.48
```

檢視目前虛擬環境的模組：

```
(opencv) ~$ pip list
```



```
pi@raspberrypi: ~  
(opencv) pi@raspberrypi:~$ pip list  
Package            Version  
-----  
cycler              0.11.0  
fonttools           4.38.0  
imutils             0.5.4  
kiwisolver          1.4.4  
matplotlib          3.5.3  
numpy               1.21.4  
opencv-contrib-python 4.5.4.60  
opencv-python       4.5.4.60  
packaging           21.3  
Pillow              9.3.0  
pip                 22.3.1  
pyparsing           3.0.9  
python-dateutil     2.8.2  
setuptools          65.5.1  
six                 1.16.0  
typing_extensions  4.4.0  
wheel               0.38.4  
(opencv) pi@raspberrypi:~$
```

## Step 6 檢查 Python 3 是否安裝好 OpenCV。

進入 Python 的互動模式：

```
(opencv) ~$ python
```

出現 >>> 後，就可輸入 Python 的程式碼：

```
>>> import cv2  
>>> cv2.__version__  
'4.6.0.66'
```

- cv2：為 OpenCV 的模組名稱。
- cv2.\_\_version\_\_：使用屬性 \_\_version\_\_ 讀取 OpenCV 的版本。

## 執行 OpenCV 的專案

在樹莓派裡直接下載 Chapter 1 的專案 (ch1.zip)：

```
(opencv) ~$ wget http://max543.com/debugger/class/python02/人臉辨識與物體  
辨識/code/ch1.zip
```

解壓縮 ch1.zip：

```
(opencv) ~$ unzip ch1.zip
```

進入 ch1 資料夾：

```
(opencv) ~$ cd ~/ch1
```

執行 Python 程式碼：(例如：1-7.py)

```
(opencv) ~ch1$ python 1-7.py
```

- 上面為編譯並執行 1-7.py 的程式碼。

**常見問題** 若相機的影像是反方向的。

可在程式碼中自行加入旋轉的程式：

```
frame = cv2.rotate(frame, rotateCode = 1)
```

## 3-5. 執行 YOLOv5 專案

### 建立與管理 YOLOv5 的虛擬環境

離開目前的虛擬環境，例如：opencv。

```
(opencv) ~$ deactivate
```

建立 yolov5 的虛擬環境，這裡同樣會用到與目前的 OpenCV 相同的模組，所以從已建立的虛擬環境 opencv，複製出一個虛擬環境 yolov5，再進行後續安裝：

```
~$ cpvirtualenv opencv yolov5
```

### 安裝 YOLOv5 所需模組

**Step 1** 下載 YOLOv5 原始碼。

先切換至到家目錄：

```
(yolov5) ~$ cd ~
```

下載 YOLOv5 原始碼：

```
(yolov5) ~$ git clone https://github.com/ultralytics/yolov5
```

進入 yolov5 資料夾：

```
(yolov5) ~$ cd yolov5
```

**Step 2** 手動安裝所需模組。

- 在 Chapter 2 中，我們是在電腦中的 Python 3.9 環境中，直接運行 `pip install -r requirements.txt`，但現在是樹莓派 CPU 為 Arm71 的 Linux 環境。因此略有不同，所以我們改成手動安裝模組，請自行對照 requirements.txt 的內容。

安裝 ipython 互動模式：

```
(yolov5) ~/yolov5$ pip install ipython
```

安裝 matplotlib：(虛擬環境 opencv 已安裝)

```
(yolov5) ~/yolov5$ pip install matplotlib
```

安裝 numpy：(虛擬環境 opencv 已安裝)

```
(yolov5) ~/yolov5$ pip install numpy==1.26.0
```

安裝 OpenCV：(需降級，虛擬環境 opencv 已安裝)

```
(yolov5) ~/yolov5$ pip install opencv-python==4.6.0.66
```

安裝 Pillow：

```
(yolov5) ~/yolov5$ pip install Pillow
```

安裝 psutil：

```
(yolov5) ~/yolov5$ pip install psutil
```

安裝 PyYAML：

```
(yolov5) ~/yolov5$ pip install PyYAML
```

安裝 requests：

```
(yolov5) ~/yolov5$ pip install requests
```

安裝 scipy：

```
(yolov5) ~/yolov5$ pip install scipy
```

安裝 torch：

```
(yolov5) ~/yolov5$ pip install torch
```

- pip install torch 與 torchvision 皆無效，請改以手動下載 .whl 檔來安裝。

## 手動安裝下載的 torch 與 torchvision

下載樹莓派 4B 專用的 32 位元 ARM 架構的 torch：

```
(yolov5) ~/yolov5$ wget https://github.com/KumaTea/pytorch-arm/releases/download/v1.8.1/torch-1.8.1-cp39-cp39-linux_armv7l.whl
```

安裝 **torch-1.8.1**：

```
(yolov5) ~/yolov5$ pip install torch-1.8.1-cp39-cp39-linux_armv7l.whl
```

下載樹莓派 4B 專用的 32 位元 ARM 架構的 torchvision：

```
(yolov5) ~/yolov5$ wget https://github.com/KumaTea/pytorch-arm/releases/download/v1.8.1/torchvision-0.9.1-cp39-cp39-linux_armv7l.whl
```

安裝 **torchvision-0.9.1**：

```
(yolov5) ~/yolov5$ pip install torchvision-0.9.1-cp39-cp39-linux_armv7l.whl
```

執行 torch，樹莓派作業系統的套件需要再安裝 **libopenblas-dev**：

```
(yolov5) ~/yolov5$ sudo apt-get install libopenblas-dev
```

## 繼續安裝剩餘的模組

安裝 ultralytics，需要指定支援 Python 3.9 的版本：

```
(yolov5) ~/yolov5$ pip install ultralytics==8.0.232
```

安裝 thop：

```
(yolov5) ~/yolov5$ pip install thop
```

安裝 tqdm：

```
(yolov5) ~/yolov5$ pip install tqdm
```

安裝 tensorboard :

```
(yolov5) ~/yolov5$ pip install tensorboard
```

安裝 pandas :

```
(yolov5) ~/yolov5$ pip install pandas
```

安裝 seaborn :

```
(yolov5) ~/yolov5$ pip install seaborn
```

## 執行 YOLOv5 的專案

進行攝影機偵測，第一台攝影機參數為 `--source 0`，第二台為 `--source 1`，依此類推：

```
(yolov5) ~/yolov5$ python detect.py --img-size 40 --iou-thres 0.3  
--source 0
```

- 樹莓派效能不如一般的電腦，可加上 `--img-size` 降低解析度。

樹莓派可再使用 `yo1ov5n.pt` (最小模型) 降低辨識延遲：

```
(yolov5) ~/yolov5$ python detect.py --img-size 40 --iou-thres 0.3  
--weight yo1ov5n.pt --source 0
```

在樹莓派裡直接下載 Chapter 2 的已經訓練好的專案 (ch2.zip)。

```
(yolov5) ~$ wget http://max543.com/debugger/class/python02/人臉辨識與物體  
辨識/code/ch2.zip
```

解壓縮 ch2.zip，且進入 ch2 資料夾中：

```
(yolov5) ~$ unzip ch2.zip && cd ./ch2
```

進入 yolov5-master 資料夾中：

```
(yolov5) ~$ cd ./yolov5-master
```

以自行訓練的 YOLO 模型偵測攝影機物件：

```
(yolov5) ~/ch2/yolov5-master$ python detect.py --weight  
runs/train/exp/weights/best.pt --img 40 --source 0
```

以自行訓練的 YOLO 模型偵測 Youtube 的影片 (路口的直播，可執行的)：

```
(yolov5) ~/ch2/yolov5-master$ python detect.py --weight  
runs/train/exp/weights/best.pt --img 40 --source  
https://www.youtube.com/watch?v=1EiC9bvVGnk
```

### 練習 1

1. 請將 2-4.py 修改成適合樹莓派的辨識速度。