

發設備~

- Raspberry PI 主體
- USB電源供應器
- Micro SD卡
- USB-to-TTL傳輸線
- 麵包版
- 感應器
 - 溫濕度, 超音波, IMU...等
- 相機模組, 麥克風
- LED燈, 電阻, 杜邦線





課程WiFi (2.4GHz)

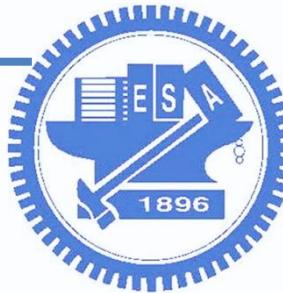
- class1 / 12345678
- class2 / 12345678
- class3 / 12345678
- class4 / 12345678



嵌入式系統設計概論與實作

曾煜棋、吳昆儒

National Yang Ming Chiao Tung University



Course Overview

Embedded devices can be used to control, monitor or assist in the operation of equipment, machinery or plant.

嵌入式系統設計概論與實作

應用1: 倒車雷達

應用2: 智慧手環

應用3: 網路攝影機

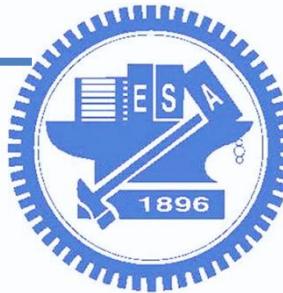
延伸應用3: 機器學習影像辨識

延伸應用3: 影像辨識

應用4: 語音助理

應用5: 網路功能(beacon)

應用X.....



嵌入式系統設計概論與實作



嵌入式應用與原理

電子學/電路學

電阻, 電壓, 電流

網路攝影機

影像傳輸, 影像辨識, AI模型

閱讀規格書

記憶體位置, 設定參數

網路通訊協定

訊號處理

藍牙推播廣告

語音識別

物理, 地球科學

加速度, 角速度, 磁北極, 大氣壓力

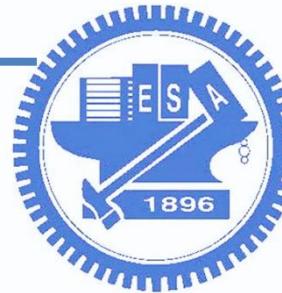
微積分, 濾波器

嵌入式開發

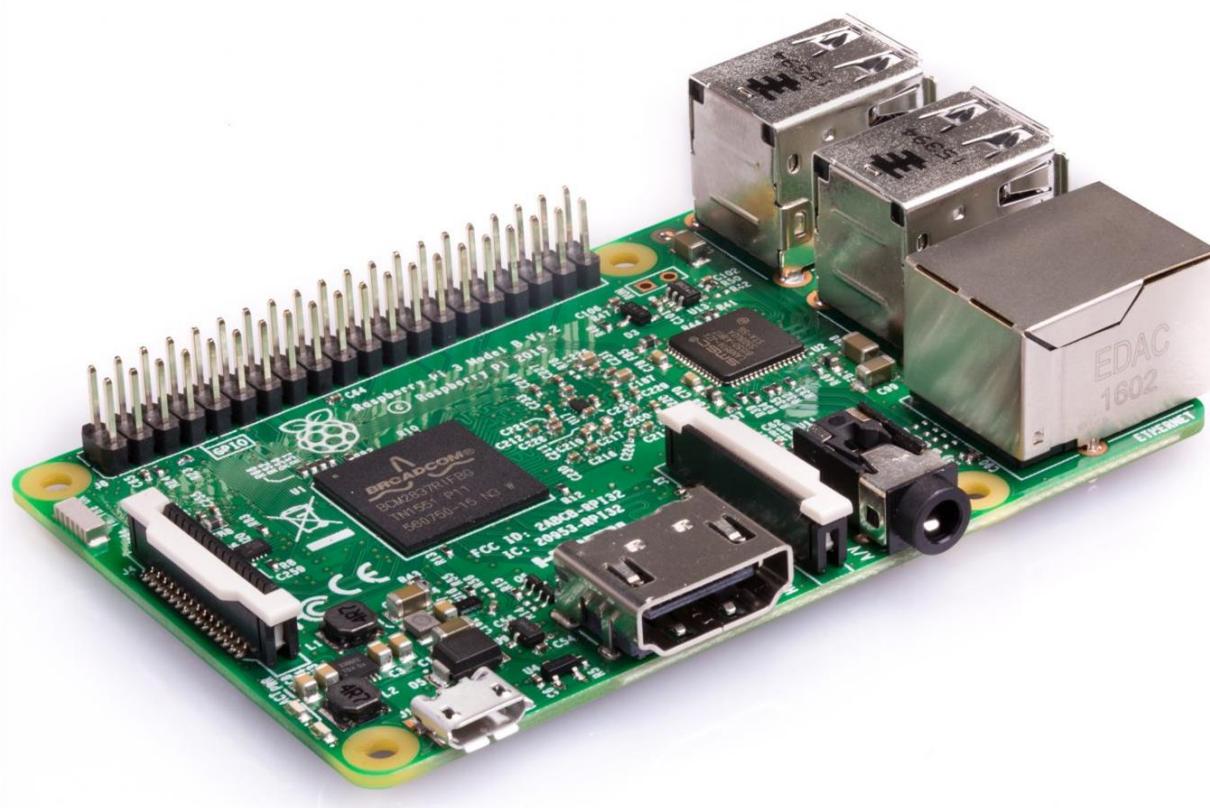
(其他...)

Cross-compile

Build Kernel



Raspberry Pi





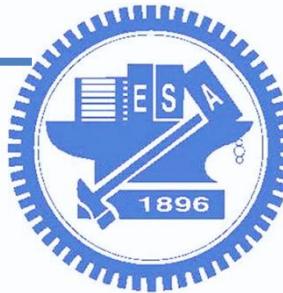
1. Raspberry PI Introduction Installation

National Chiao Tung University



Outline

- **1. 安裝OS (Raspbian)**
- **2. PI的環境設定**
 - A. 系統工具: raspi-config (擴充SD卡空間, 開啟interface...等)
 - B. 使用apt-get安裝程式
- **3. 設定遠端桌面連線**
 - A. 內建的realvnc
 - B. 設定有線網路對接
- **4. GPIO + Python + LED**
 - A. GPIO introduction
 - B. Python example
 - C. 電子零件與控制LED燈
- **5. 傳輸檔案到PI**

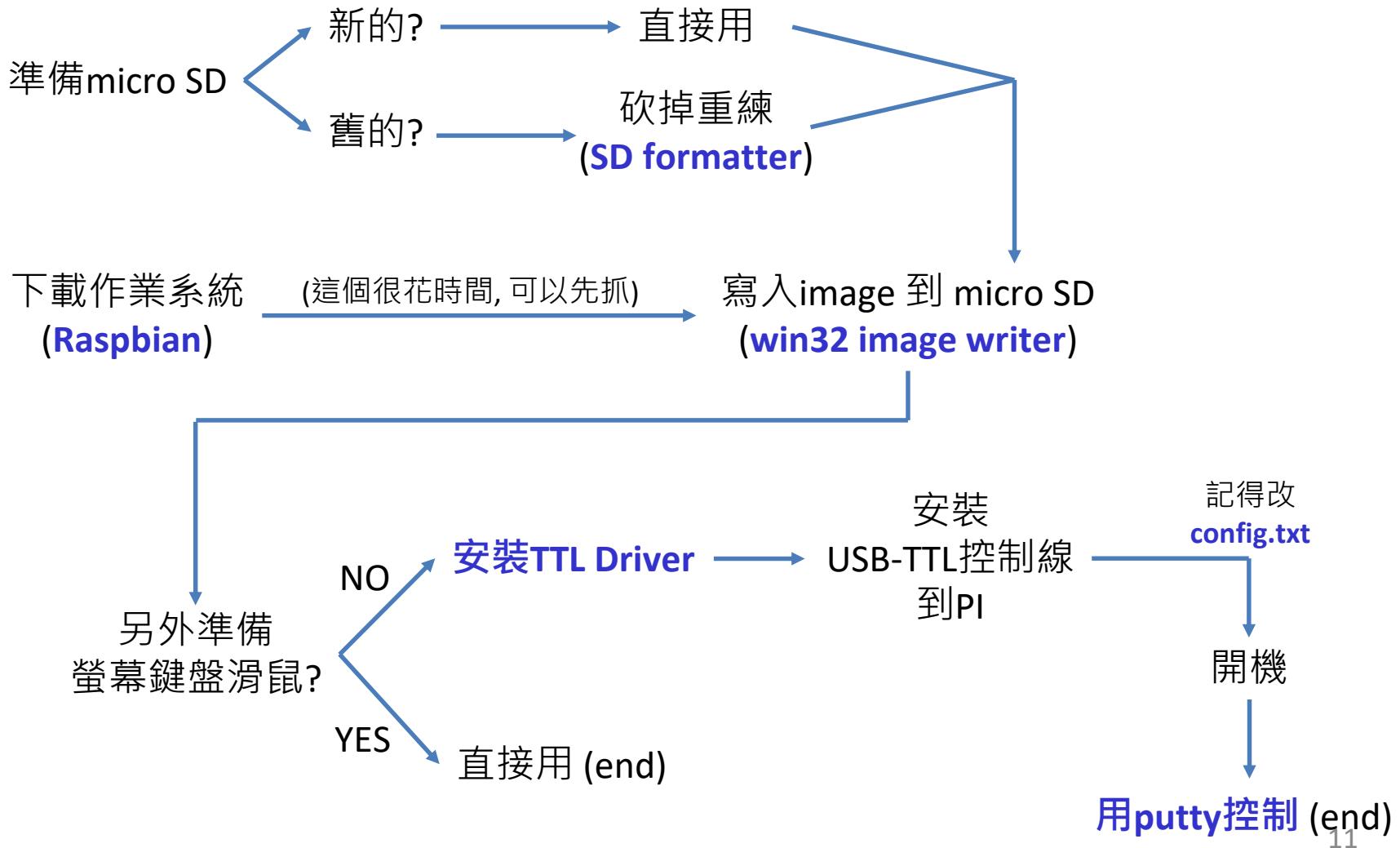


Outline

- 1. 安裝OS (Raspbian)
- 2. PI的環境設定
 - A. 系統工具: raspi-config (擴充SD卡空間, 開啟interface...等)
 - B. 使用apt-get安裝程式
- 3. 設定遠端桌面連線
 - A. 內建的realvnc
 - B. 設定有線網路對接
- 4. GPIO + Python + LED
 - A. GPIO introduction
 - B. Python example
 - C. 電子零件與控制LED燈
- 5. 傳輸檔案到PI



準備流程





準備工具

- 下載作業系統(Raspbian)
 - 下載zip之後, 要先解壓縮, 將*.img寫入到SD卡裡
 - 官方連結 (速度很慢)
 - https://downloads.raspberrypi.org/raspbian_latest
 - Google Drive 分流 (2021-01-11-raspios-buster-armhf-full)
 - Raspberry Pi OS with desktop and recommended software
 - <https://drive.google.com/file/d/1MjoibeAdc8bGL-lozVxT1rklnL5laX24/view?usp=sharing>



準備工具

- SD Formatter (清除SD卡舊有的檔案系統)
 - https://www.sdcard.org/cht/downloads/formatter_4/
- Etcher (寫入映像檔到SD卡)
 - <https://www.balena.io/etcher/>
- Notepad ++ (編輯開機設定檔)
 - <https://notepad-plus-plus.org/download/>
- USB TTL driver (TTL控制線驅動程式)
 - http://www.prolific.com.tw/US>ShowProduct.aspx?p_id=225&pcid=41
- Putty (終端機程式)
 - <https://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>



安裝Raspbian

- 步驟1：下載映像檔
 - Raspbian
- 步驟2：將映像檔燒錄至SD卡
 - SD formatter(格式化), Etcher(寫入資料)
- 步驟3：調整開機設定檔
 - config.txt, cmdline.txt
- 步驟4：電腦端準備 USB 轉 TTL 序列傳輸線
 - 安裝TTL Driver
- 步驟5：將SD卡插到Raspberry PI並開機
 - 用putty控制 (by USB-TTL), SSH連線, VNC遠端桌面



步驟1：下載映像檔

- **官方下載網頁**
 - <http://www.raspberrypi.org/downloads>
- **選擇映像檔(image)**
 - **Raspbian(推薦)** ↗

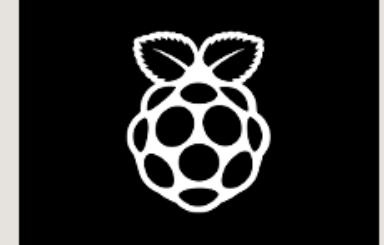
Third Party Operating System Images

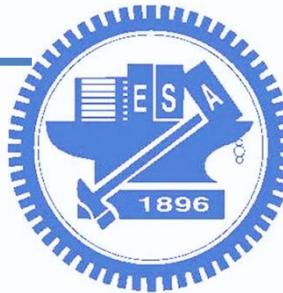
Third-party operating system images for Raspberry Pi are also available:

 Ubuntu MATE	 Snappy Ubuntu Core
 Windows 10 IoT Core	 OSMC

Downloads

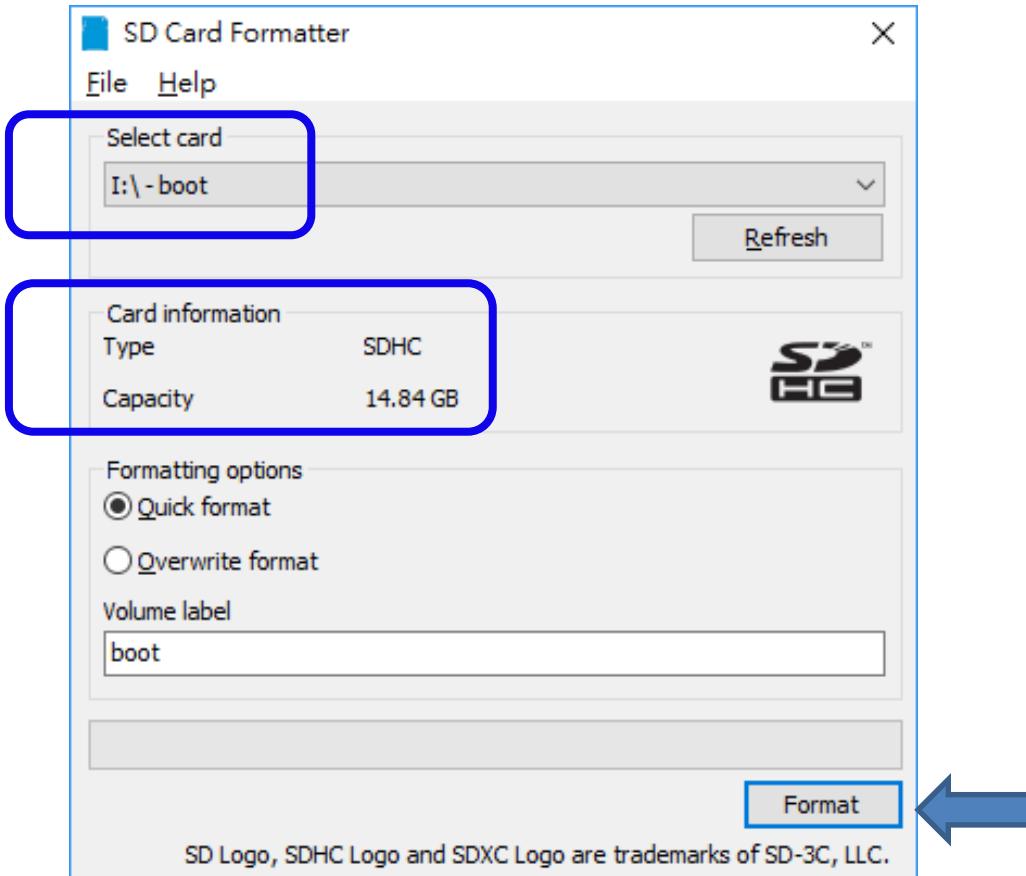
Raspbian is our official operating system for **all** models of the Raspberry Pi. Download it here, or use **NOOBS**, our easy installer for Raspbian and more.

 NOOBS	 Raspbian
--	--

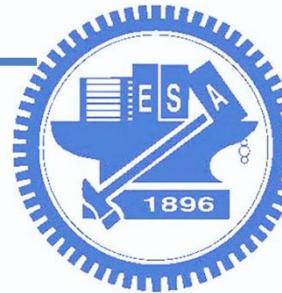


步驟2：將映像檔燒錄至SD卡之前

- If the micro SD is used before, use **SD Card Formatter** to erase it!!



Check Device ID and capacity



步驟2：將映像檔燒錄至SD卡

□ 下載燒錄軟體

□ balenaEtcher

■ <https://www.balena.io/etcher>

■ 可以寫入至SD卡，並顯示該SD卡的大小。

The screenshot shows the official website for balenaEtcher. At the top, it says "An open source project by balena | More products". The main heading is "balenaEtcher" with a green cube icon. To the right are links for "Forums", "Mailing list", "Changelog", and "Etcher Pro". Below the heading is a large white text "Flash. Flawless.". Underneath it, a subtext reads "Flash OS images to SD cards & USB drives, safely and easily.". A central call-to-action button has three steps: "Select image" (with a plus sign icon), "Select drive" (with a smartphone icon), and "Flash!" (with a lightning bolt icon). At the bottom, there's a green button for "Download for Windows (x86|x64)" and the text "v1.5.78 See what's new".



步驟2：將映像檔燒錄至SD卡

□ (Alternative) 燒錄軟體

□ Win32 Disk Imager

- <http://sourceforge.net/projects/win32diskimager/>
- 可以寫入/讀取SD卡的內容. 寫入時要注意磁碟代號

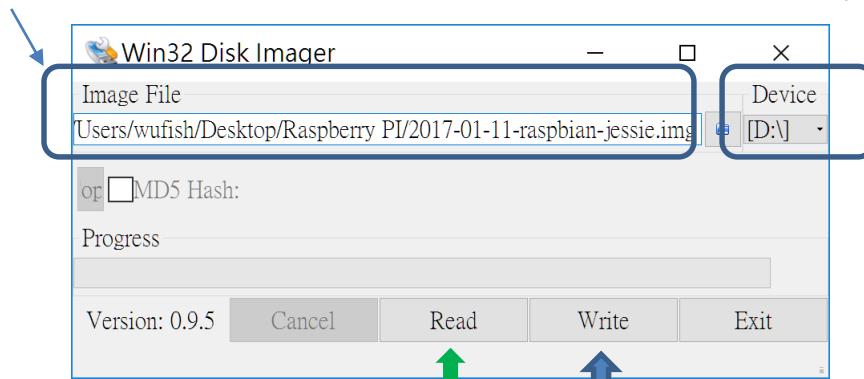


□ 燒錄映像檔

□ 開啟Win32DiskImager

□ 選取映像檔及欲燒入SD卡路徑並開始燒錄

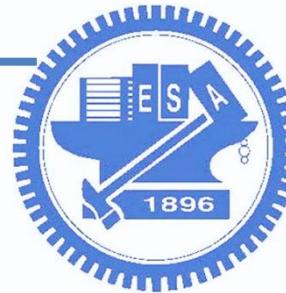
選擇image位置



選擇SD卡位置

Read: 將SD卡儲存至image路徑, 可備份整個系統

Write: 將image寫入至SD卡



步驟2：將映像檔燒錄至SD卡 (MAC user)

For MAC OS

- Use “`diskutil list`” to check the device number of your SD card
 - Ex: `/dev/disk2`
- After insert SD card, remember to unmount it
- `diskutil unmountDisk /dev/diskX` (change **diskX** to above ID)
- Write image to SD card
 - `sudo dd bs=1m if=2017-01-11-raspbian-jessie.img of=/dev/diskX`



Raspbian image path



Your SD card ID

- You can use **Ctrl+T** to check the progress (It would take a long time)



步驟3：調整開機設定檔

- 拿到 Pi3 or Pi4, 想使用序列埠連線, 會發現亂碼, 該怎麼辦?
(使用 SSH 或 螢幕鍵盤滑鼠 可以忽略這問題)

因為原本 *Pi 3* 內建的硬體 *UART* 被 *BCM2837 SoC* 拿去給 *Bluetooth* 晶片組使用，而原本的 *UART* 輸出腳位(*GPIOs 14 & 15*)改成用 *mini-uart port*。意思是原本硬體 *UART* 有獨立的 *clock divisor*，因此 *baud rate* 可以維持在 *115200*，可是 *mini-uart* 使用系統核心時脈，實際只能跑到 *72000* 左右的 *baud rate*，因此當使用 *115200* 的 *baud rate* 連線就會出現亂碼



步驟3：調整開機設定檔 (PI3)

- 用notepad++修改`/boot/config.txt`,新增三行

- `enable_uart=1`
 - `core_freq=250`
 - `dtoverlay=minuart-bt`

新增 {

```
# Enable audio (loads snd_bcm2835)
dtoparam=audio=on
dtoverlay=pi3-minuart-bt
core_freq=250
enable_uart=1
```

- 修改`/boot/cmdline.txt`,將quiet移除

```
dwc_otg.lpm_enable=0 console=serial0,115200
console=tty1 root=PARTUUID=6b3a87b5-02 rootfstype=ext4
elevator=deadline fsck.repair=yes rootwait quiet
init=/usr/lib/raspi-config/init_resize.sh splash
plymouth.ignore-serial-consoles
```

SD卡插入windows電腦後, 會出現一個磁碟機. 修改磁碟機裡面的檔案即可!



步驟3：調整開機設定檔 (PI4)

- 用 **notepad++** 修改 **/boot/config.txt**, 新增

- `enable_uart=1`

新增 

```
[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_uart=1
```

- 修改 **/boot/cmdline.txt**, 將 `quiet splash` 的 `quiet` 移除

```
console=serial0,115200 console=tty1 root=PARTUUID=fba96bfa-02
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait quiet
plymouth.ignore-serial-consoles
```

SD卡插入windows電腦後，會出現一個磁碟機。修改磁碟機裡面的檔案即可！



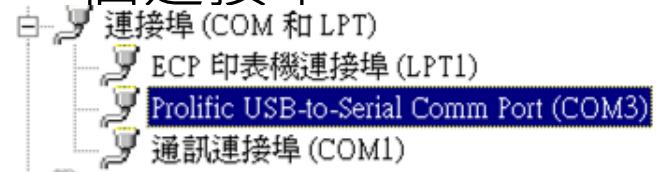
步驟4：電腦端準備 USB 轉 TTL 序列傳輸線

- 從序列埠登入到 Raspberry Pi
 - 透過 USB 轉 TTL 序列傳輸線，就可以在不需要螢幕和鍵盤滑鼠的情況下登入 Raspberry Pi
- 晶片組PL2303HXD:支援Windows 8/10, MAC
 - Windows Driver:
 - http://www.prolific.com.tw/US>ShowProduct.aspx?p_id=225&pcid=41
 - MAC driver:
 - http://www.prolific.com.tw/US>ShowProduct.aspx?p_id=229&pcid=41
 - <https://www.ftdichip.com/Drivers/VCP.htm>

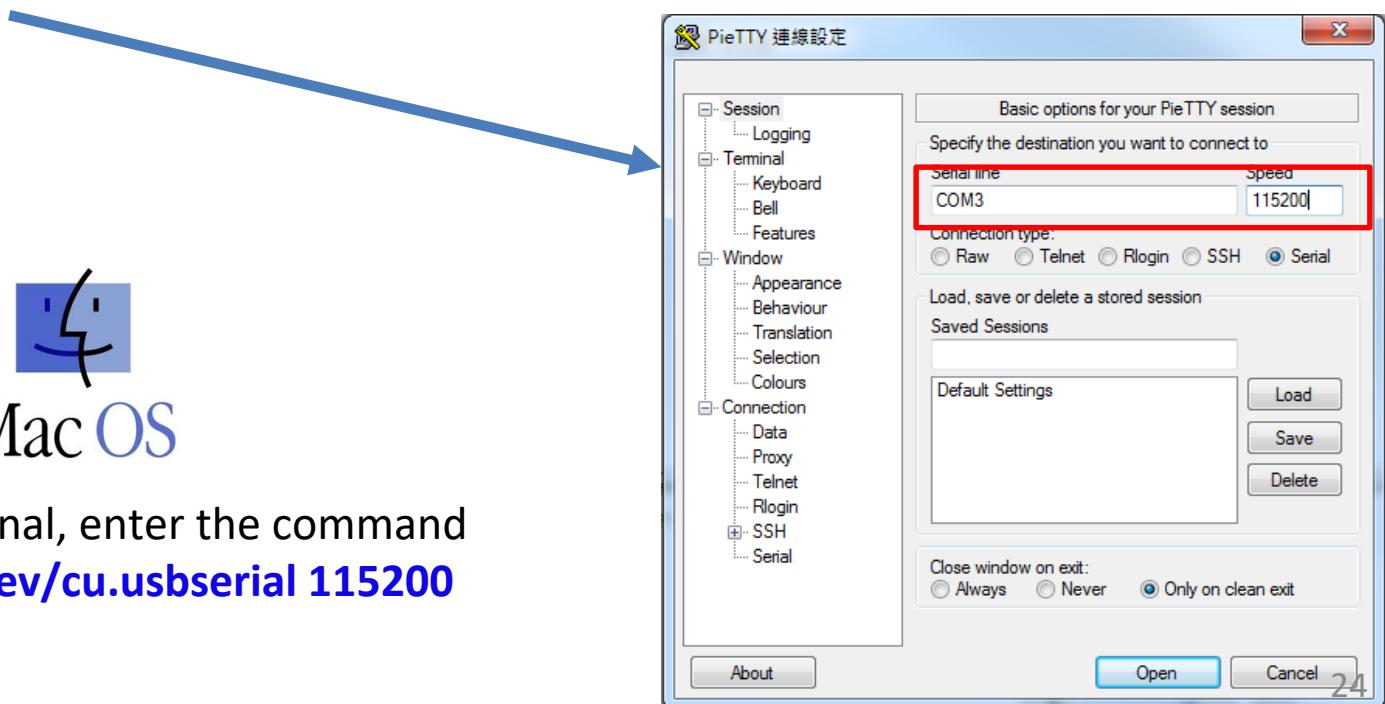


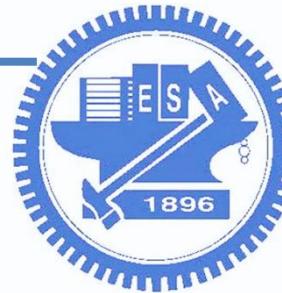
步驟4：電腦端準備 USB 轉 TTL 序列傳輸線

- 安裝完驅動，在裝置管理員可發現多一個連接埠
(下圖為 **COM3**)
 - 我的電腦 -> 右鍵-> 內容-> 裝置管理員
- 使用putty連線，設定Serial port (**COM3**) 與 Speed(**115200**)



- For MAC OS
 - Use terminal, enter the command
 - **screen /dev/cu.usbserial 115200**





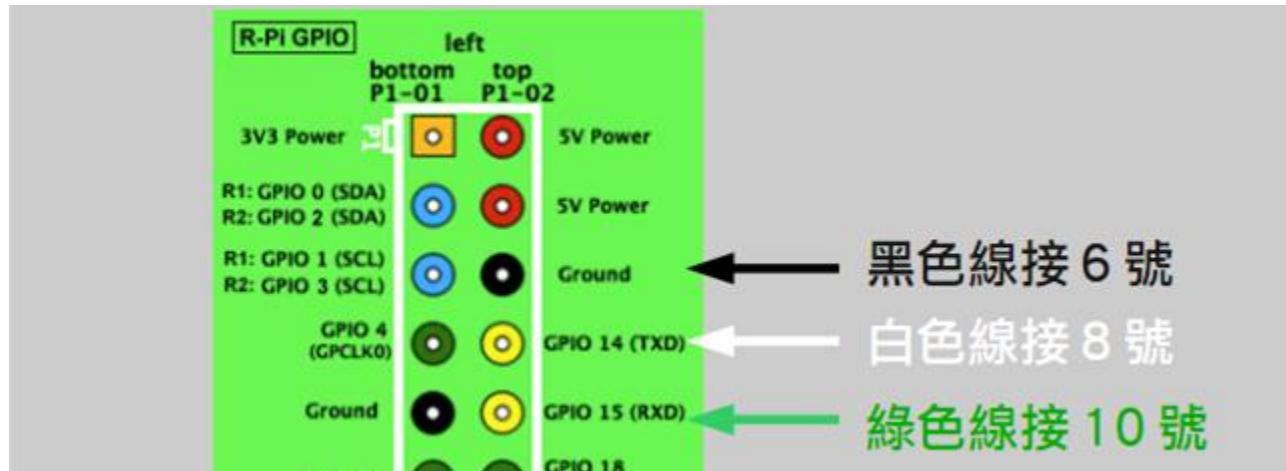
步驟5：將SD卡插到Raspberry Pi並開機

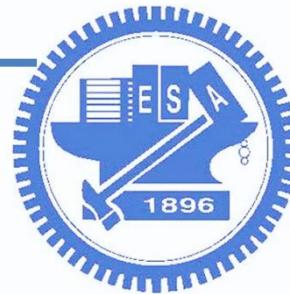
□ 從序列埠登入到 Raspberry Pi

- 透過 USB 轉 TTL 序列傳輸線，就可以在不需要螢幕和鍵盤滑鼠的情況下登入 Raspberry Pi

□ 預設登入帳密

- ID: pi
- PW: raspberry



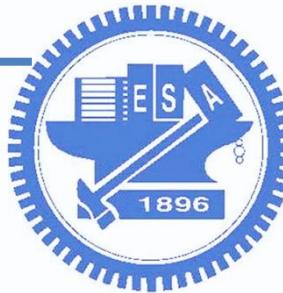


步驟5：將SD卡插到Raspberry PI並開機

COM3 - PuTTY

```
Raspbian GNU/Linux 8 raspberrypi ttyAMA0
raspberrypi login: █
```

還是沒有畫面? -> 按一下Enter鍵 or 電腦重開機試試



Outline

- 1. 安裝OS (Raspbian)
- 2. PI的環境設定
 - A. 系統工具: raspi-config (擴充SD卡空間, 開啟interface...等)
 - B. 使用apt-get安裝程式
- 3. 設定遠端桌面連線
 - A. 內建的realvnc
 - B. 設定有線網路對接
- 4. GPIO + Python + LED
 - A. GPIO introduction
 - B. Python example
 - C. 電子零件與控制LED燈
- 5. 傳輸檔案到PI



2. PI的環境設定

□ 系統工具: raspi-config

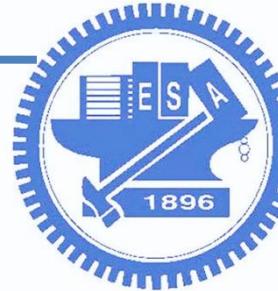
COM6 - PuTTY

Raspberry Pi 4 Model B Rev 1.4

```
| Raspberry Pi Software Configuration Tool (raspi-config) |
```

1 System Options	Configure system settings
2 Display Options	Configure display settings
3 Interface Options	Configure connections to peripherals
4 Performance Options	Configure performance settings
5 Localisation Options	Configure language and regional settings
6 Advanced Options	Configure advanced settings
8 Update	Update this tool to the latest version
9 About raspi-config	Information about this configuration tool

<Select> <Finish>



A. PI的環境設定

□ 1 System Options

- S1 Wireless LAN
 - Select the country in which the Pi is to be used
- S2 Audio

以前要修改系統設定檔, 現在
可以直接輸入SSID與密碼了!

□ 2 Display Options

- D1 Resolution
 - DMT Mode 16 1024x768 60Hz 4:3

設定聲音輸出選項
(HDMI 與 3.5mm Headphone)

設定螢幕解析度, VNC會用到

□ 3 Interface Options

- P1 Camera
- P2 SSH
- P3 VNC
- P5 I2C

啟用camera interface

啟用SSH server, 可以透過網路遠端操控(文字介面)

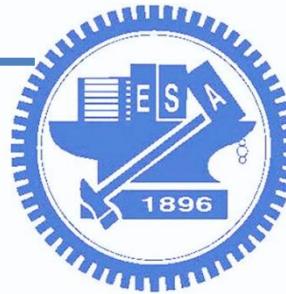
啟用vnc server, 可以透過網路遠端操控(圖形介面)

啟用I2C interface, 可以接I2C sensor

□ 6 Advanced Options

- A1 Expand Filesystem

擴充系統到整張SD卡的容量



S1 Wireless LAN

```
COM6 - PuTTY
Please enter SSID


COM6 - PuTTY
Please enter passphrase. Leave it empty if none.
*****
<Ok> <Cancel>
```



確認網路狀態

□ 是否已連到網路?

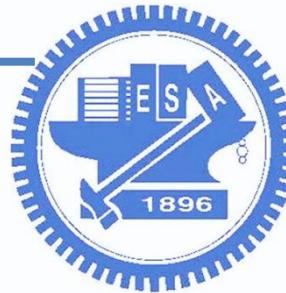
□ 指令: ifconfig 或 iwconfig

■ ifconfig 執行結果

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.20 netmask 255.255.255.0 broadcast 192.168.1.255
      inet6 fe80::bdbe:cb97:93ed:4516 prefixlen 64 scopeid 0x20<link>
        ether b8:27:eb:ea:da:f8 txqueuelen 1000 (Ethernet)
          RX packets 10 bytes 1575 (1.5 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 34 bytes 5841 (5.7 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

■ iwconfig 執行結果

```
wlan0      IEEE 802.11  ESSID:"hscc_wu56"
          Mode:Managed  Frequency:2.437 GHz  Access Point: AC:9E:17:8E:9A:AC
          Bit Rate=65 Mb/s   Tx-Power=31 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=70/70  Signal level=-33 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```



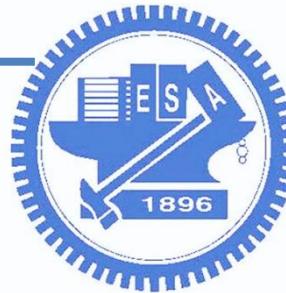
S2 Audio

COM6 - PuTTY

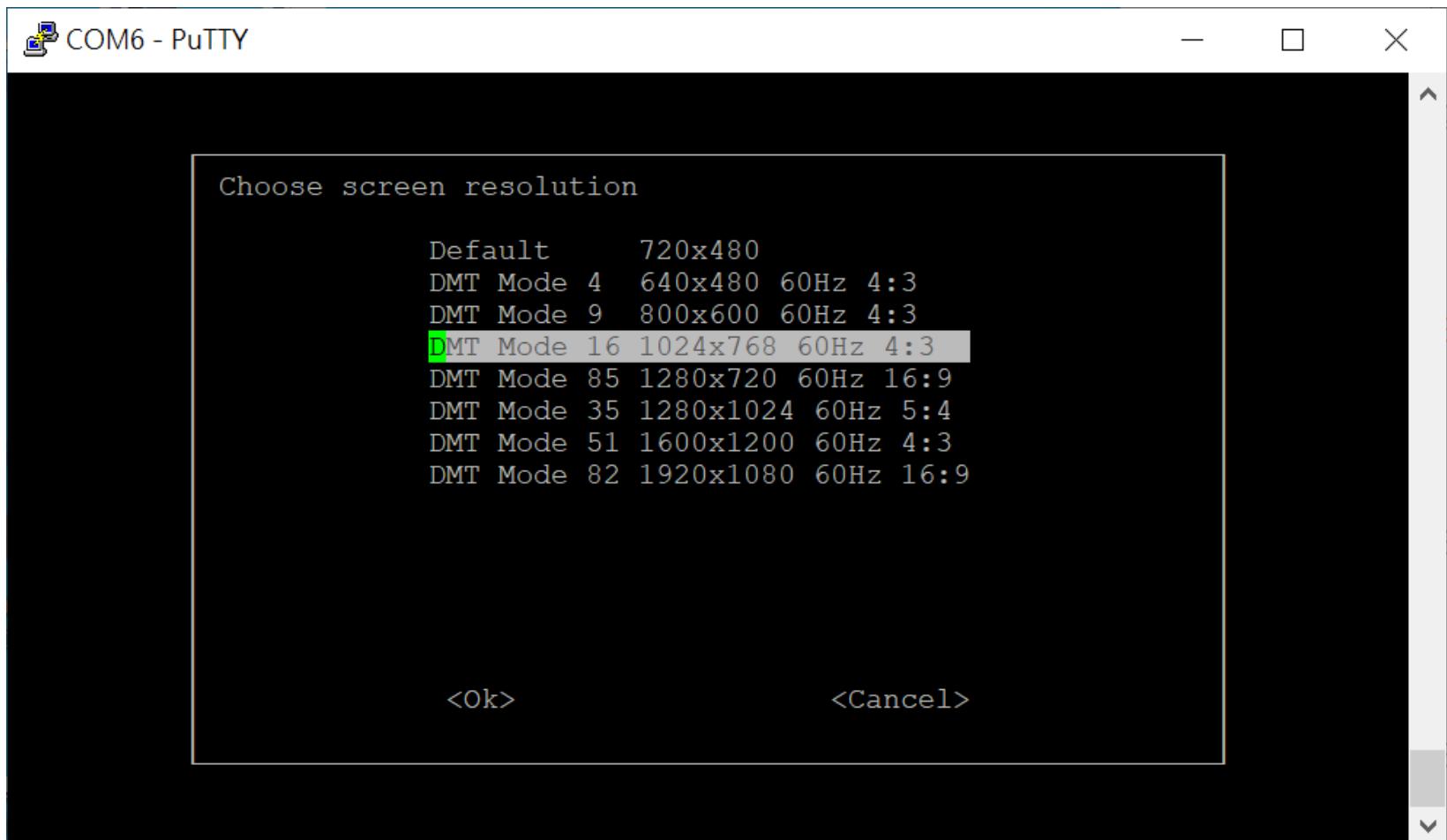
Choose the audio output

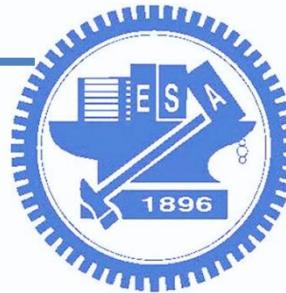
- 0 USB Audio
- 1 HDMI 1
- 2 Headphones

<Ok> <Cancel>

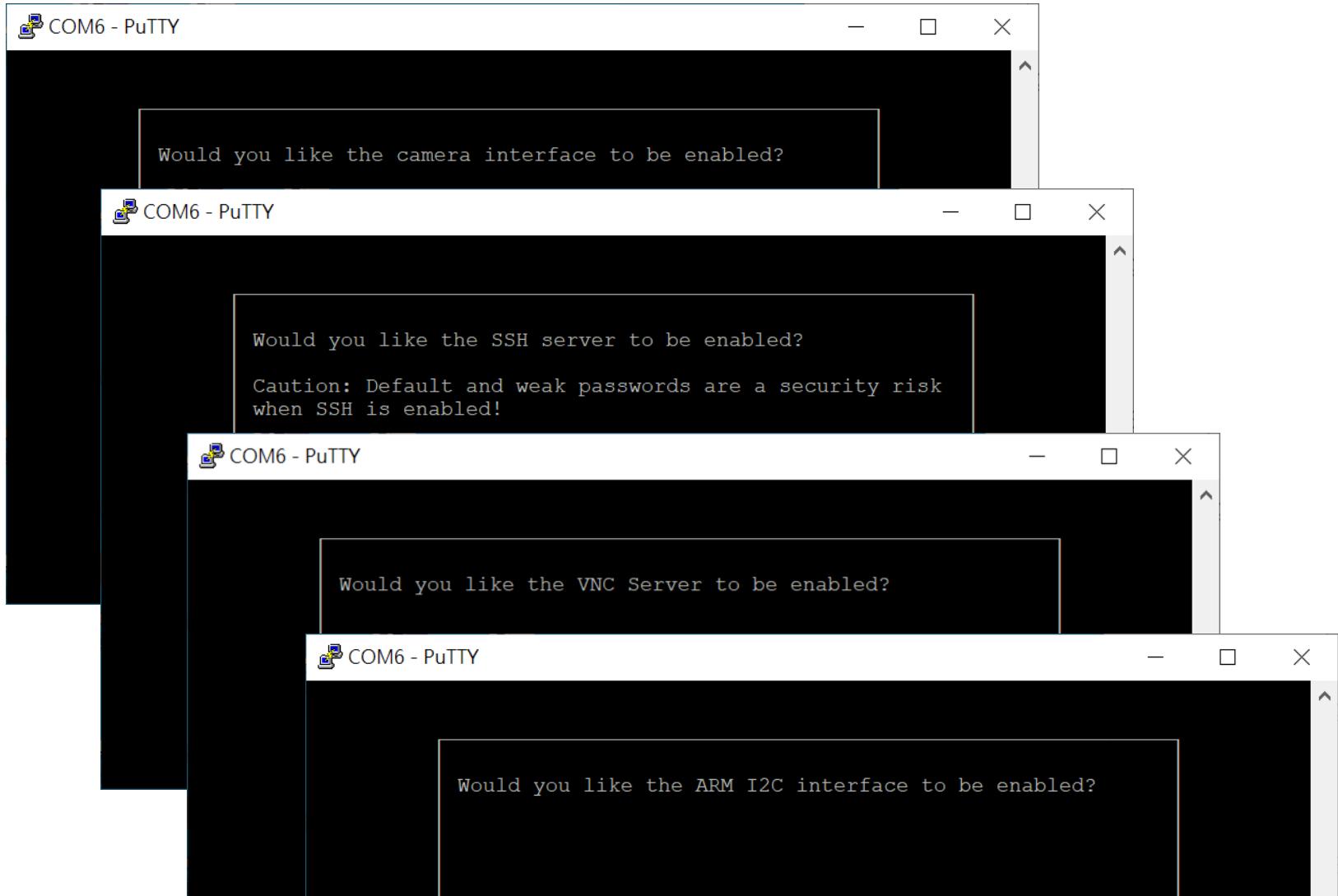


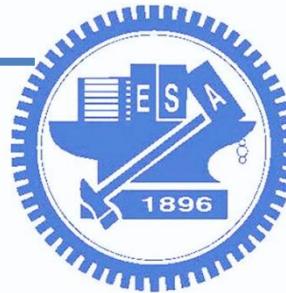
D1 Resolution





3 Interface





A1 Expand Filesystem

COM6 - PuTTY

```
Root partition has been resized.  
The filesystem will be enlarged upon the next reboot
```

- 預設只有使用約2G的空間
-> 可是SD卡有16G
-> 浪費了10G的空間
- Reboot後系統可使用整張SD卡的空間

<Ok>



(參考用) 設定Wi-Fi的方法

□ 舊版的設定Wi-Fi方法

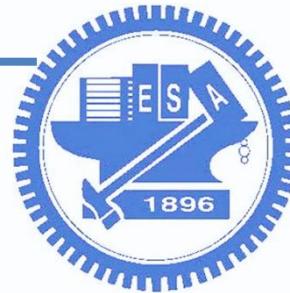
- 使用螢幕鍵盤滑鼠, 可以直接點選Wi-Fi
-> 在純文字介面要怎麼設定Wi-Fi?

□ 用文字編輯器修改設定檔

1. sudo nano /etc/wpa_supplicant/wpa_supplicant.conf (編輯設定檔)
2. 填寫SSID與密碼
3. 重開機
 - sudo reboot
4. 使用 ifconfig 與 iwconfig 檢查連線狀態

```
network={  
    ssid="your_ap"  
    key_mgmt=NONE  
} // open system
```

```
network={  
    ssid="your_ap"  
    psk="your_passwd"  
} // with password
```



(參考用) 設定Wi-Fi的方法

□ [設定Wi-Fi] 修改設定檔

□ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

```
country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

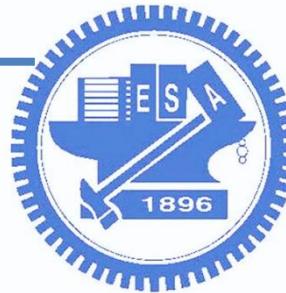
network={
    ssid="████████"
    psk="████████"
}

[ Read 8 lines ]
^G Get Help      ^O WriteOut     ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit          ^J Justify      ^W Where Is   ^V Next Page  ^U Uncut Text ^T To Spell
```

The screenshot shows the configuration file with two network entries. The first entry is for a secured network with SSID and PSK fields redacted. The second entry is for an open network with SSID and key_mgmt fields redacted.

```
network={
    ssid="SSID"
    psk="your_pw"
}

network={
    ssid="SSID_open"
    key_mgmt=NONE
}
```



(參考用) 設定Wi-Fi的方法

□ [設定Wi-Fi] 文字編輯器 nano

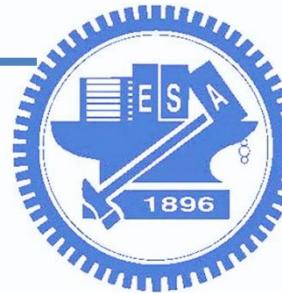
□ 編輯結束按 **ctrl + x** 離開

■ 若有變動, 會問你是否存檔, 輸入 **Y** 即可

```
country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="XXXXXXXXXX"
    psk="XXXXXXXXXX"
}
```

Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ? Y Yes N No ^C Cancel



(參考用) 設定Wi-Fi的方法

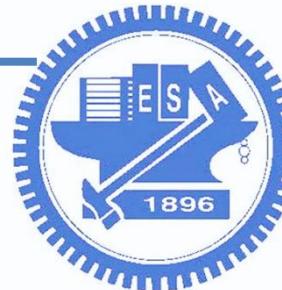
- [設定Wi-Fi] 重新啟動
 - 在終端機打 sudo reboot

```
#         psk="12345678"
#}

network={

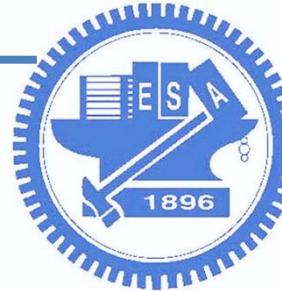
}

pi@raspberrypi:~$ reboot
Failed to set wall message, ignoring: Interactive authentication required.
Failed to reboot system via logind: Interactive authentication required.
Failed to open /dev/initctl: Permission denied
Failed to talk to init daemon.
pi@raspberrypi:~$ sudo reboot
[71643.595503] reboot: Restarting system
```



B. PI的環境設定

- 網路設定好之後, 可以開始建立想要的服務 or 下載程式
- 建立服務: 使用apt安裝程式
 - APT = Advanced Packaging Tools
 - 連上網路自動下載程式來安裝
 - 已編譯好, 不須從source code重新編譯
 - 類似Appstore與Google play線上商店的概念
- Ex: sudo apt install git
 - 安裝Git工具

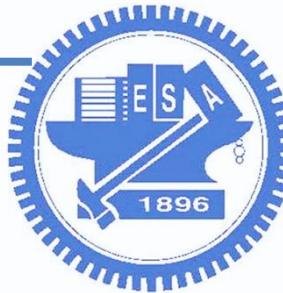


B. 使用apt-get安裝程式

<https://www.raspbian.org/RaspbianMirrors>

- (Optional) 修改apt站台成國網中心, 速度會較快:
 - sudo nano /etc/apt/sources.list
 - 原本: deb http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi
 - 修改: deb http://free.nchc.org.tw/raspbian/raspbian/ buster main contrib non-free rpi
- **更新套件資訊: sudo apt-get update**
- **安裝套件: sudo apt-get install <pkg_name>**
- **升級套件: sudo apt-get upgrade**
- 彩蛋: sudo apt-get moo

```
pi@raspberrypi:~$ sudo apt-get moo
(_____
 (oo)
 /-----\ \
 / |     || \
 *  /\---/ \
 ~~      ~~
... "Have you mooed today?"...
```



Outline

- 1. 安裝OS (Raspbian)
- 2. PI的環境設定
 - A. 系統工具: raspi-config (擴充SD卡空間, 開啟interface...等)
 - B. 使用apt-get安裝程式
- 3. 設定遠端桌面連線
 - A. 內建的realvnc
 - B. 設定有線網路對接
- 4. GPIO + Python + LED
 - A. GPIO introduction
 - B. Python example
 - C. 電子零件與控制LED燈
- 5. 傳輸檔案到PI



A. 內建的realvnc

- 電腦端執行vncviewer
 - 輸入樹莓派的IP (可在PI執行 ifconfig 查詢)

VNC Viewer

File View Help

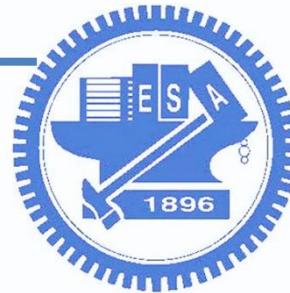
192.168.1.20

Sign in...

192.168.1.48:0 192.168.1.48:1

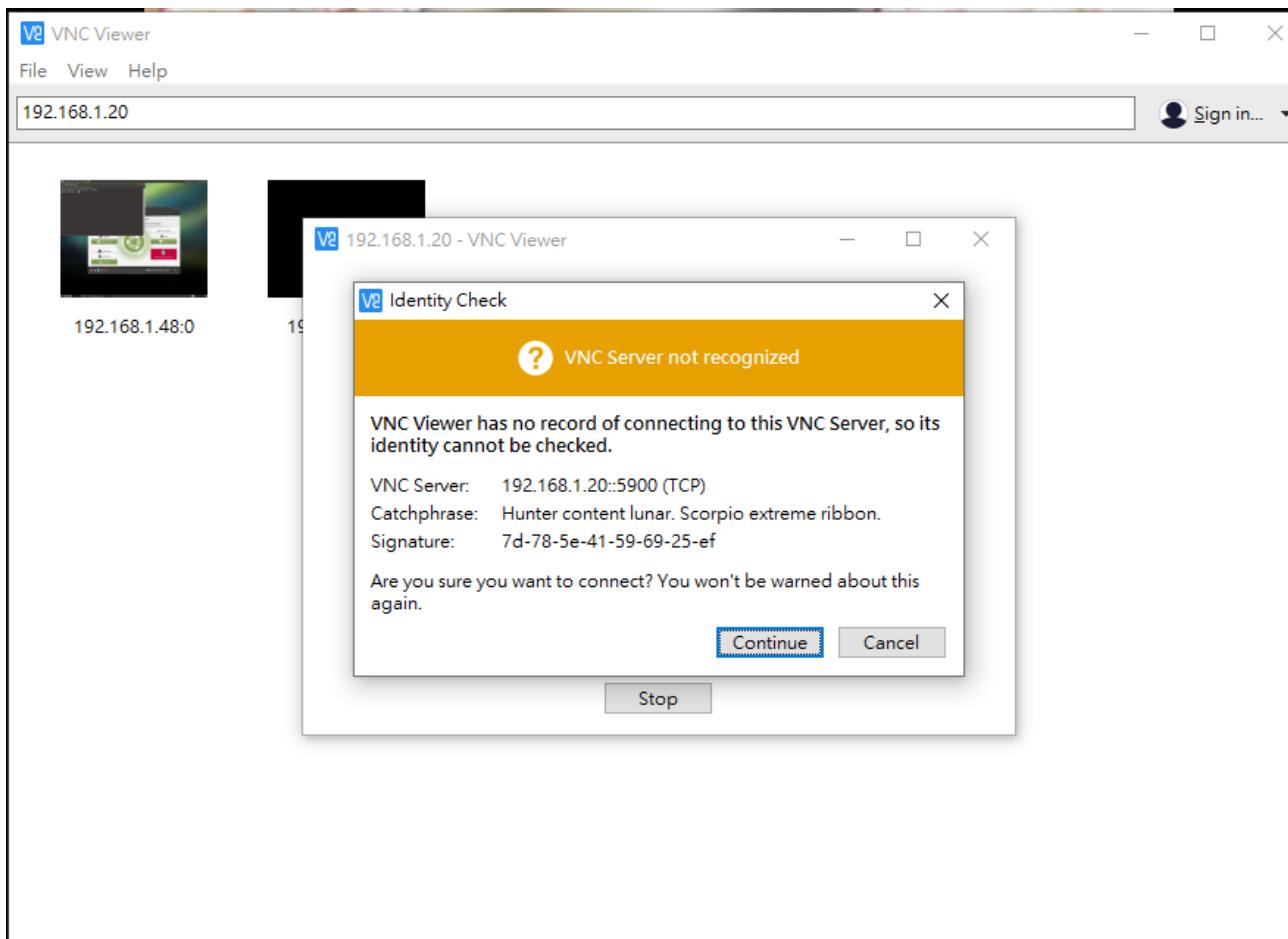
```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.20 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::bdbe:cb97:93ed:4516 prefixlen 64 scopeid 0x20<link>
          ether b8:27:eb:ea:da:f8 txqueuelen 1000  (Ethernet)
            RX packets 10 bytes 1575 (1.5 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 34 bytes 5841 (5.7 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

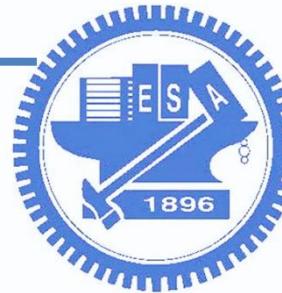
45



A. 內建的realvnc

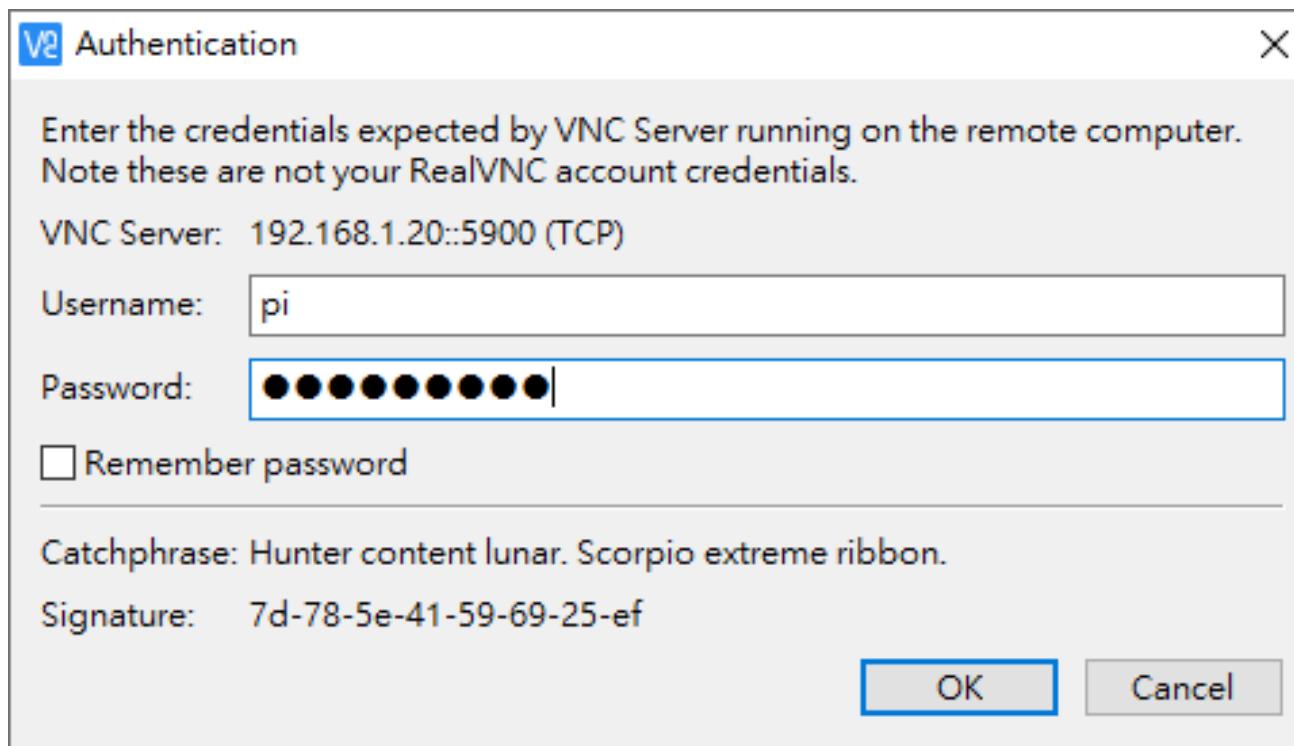
□ 電腦端執行vncviewer

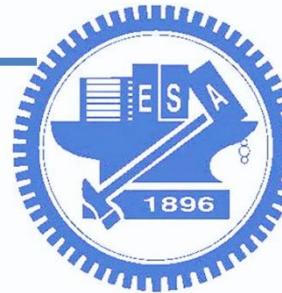




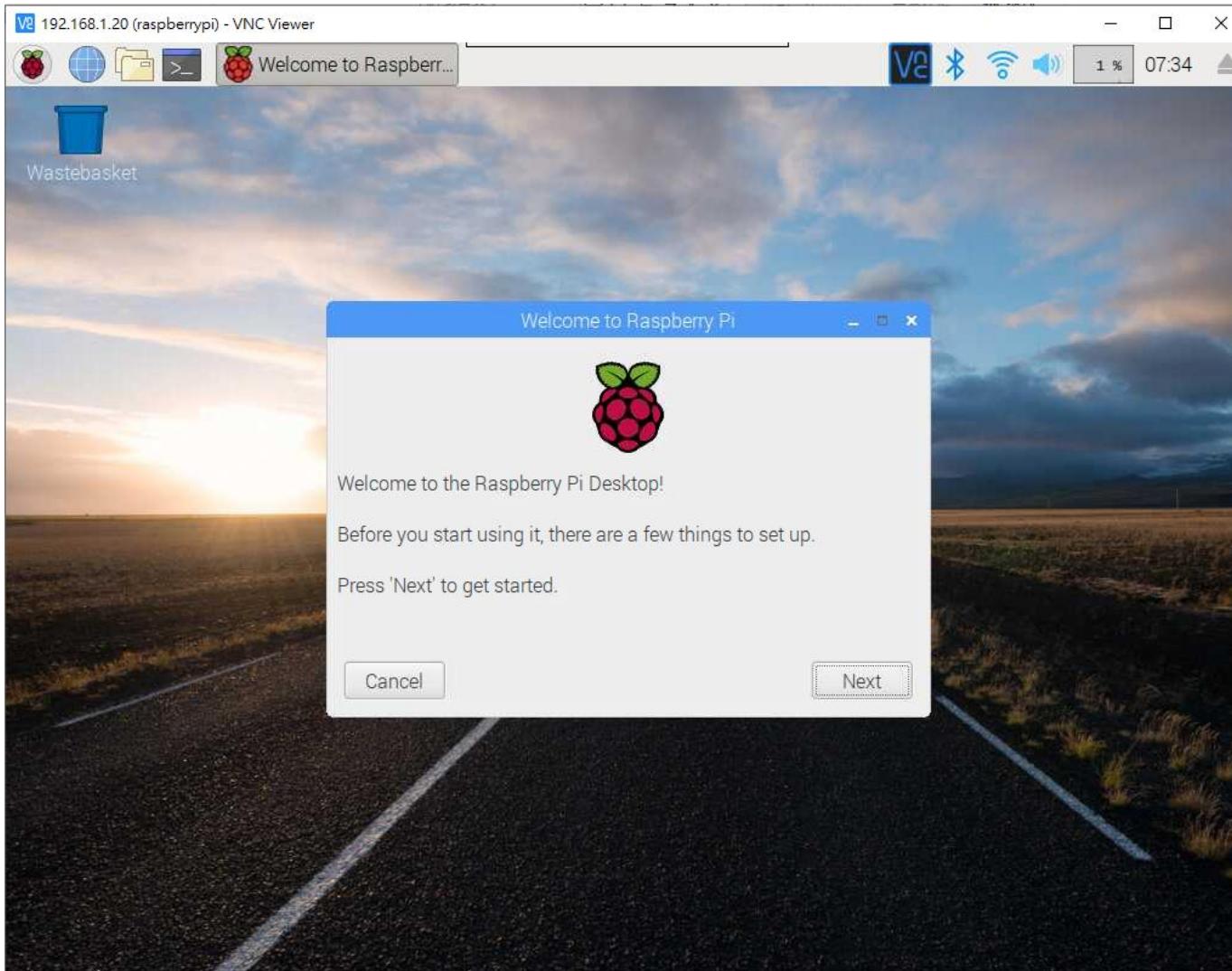
A. 內建的realvnc

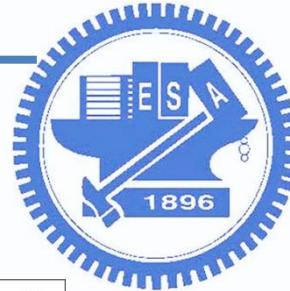
- 電腦端執行vncviewer



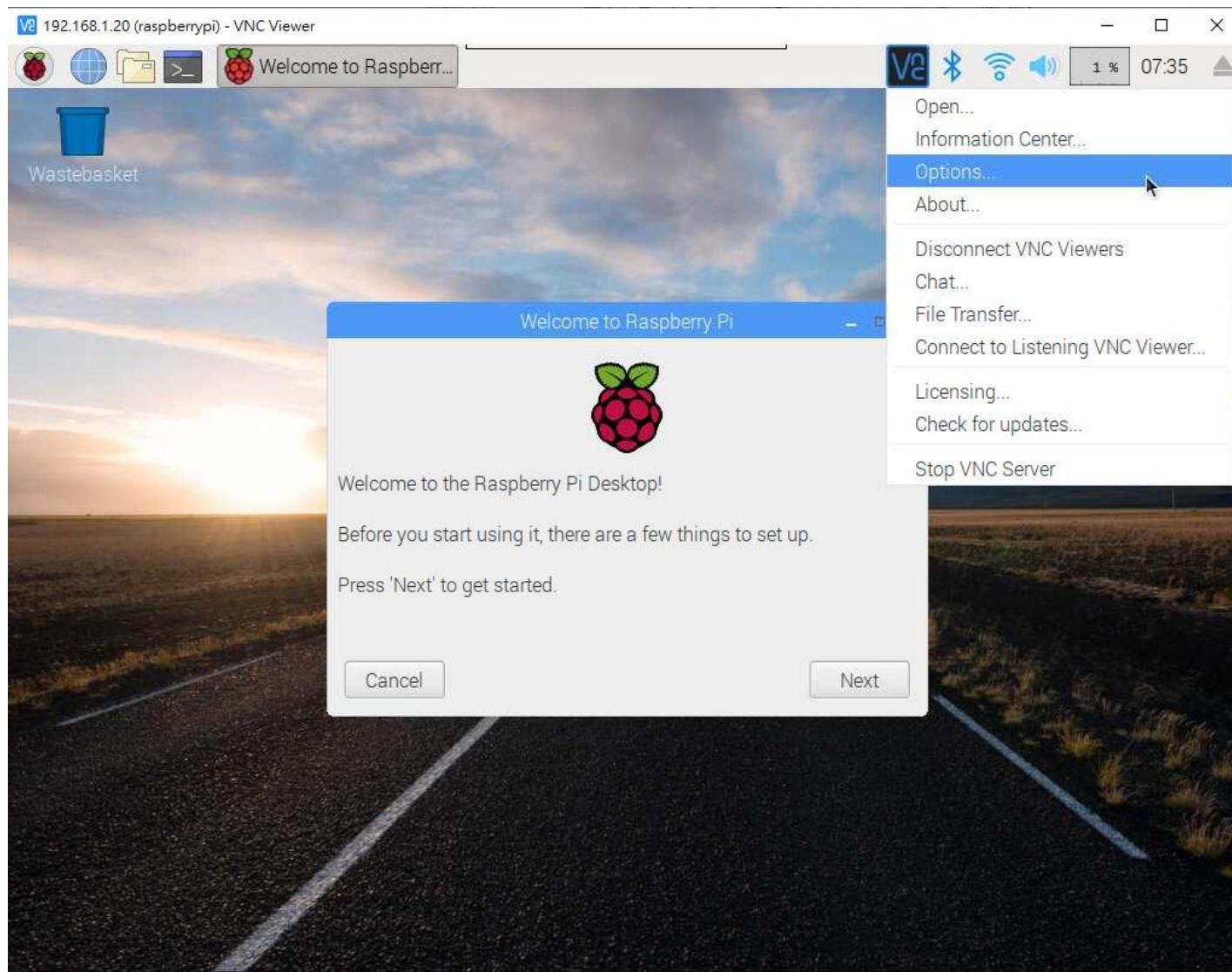


A. 內建的realvnc



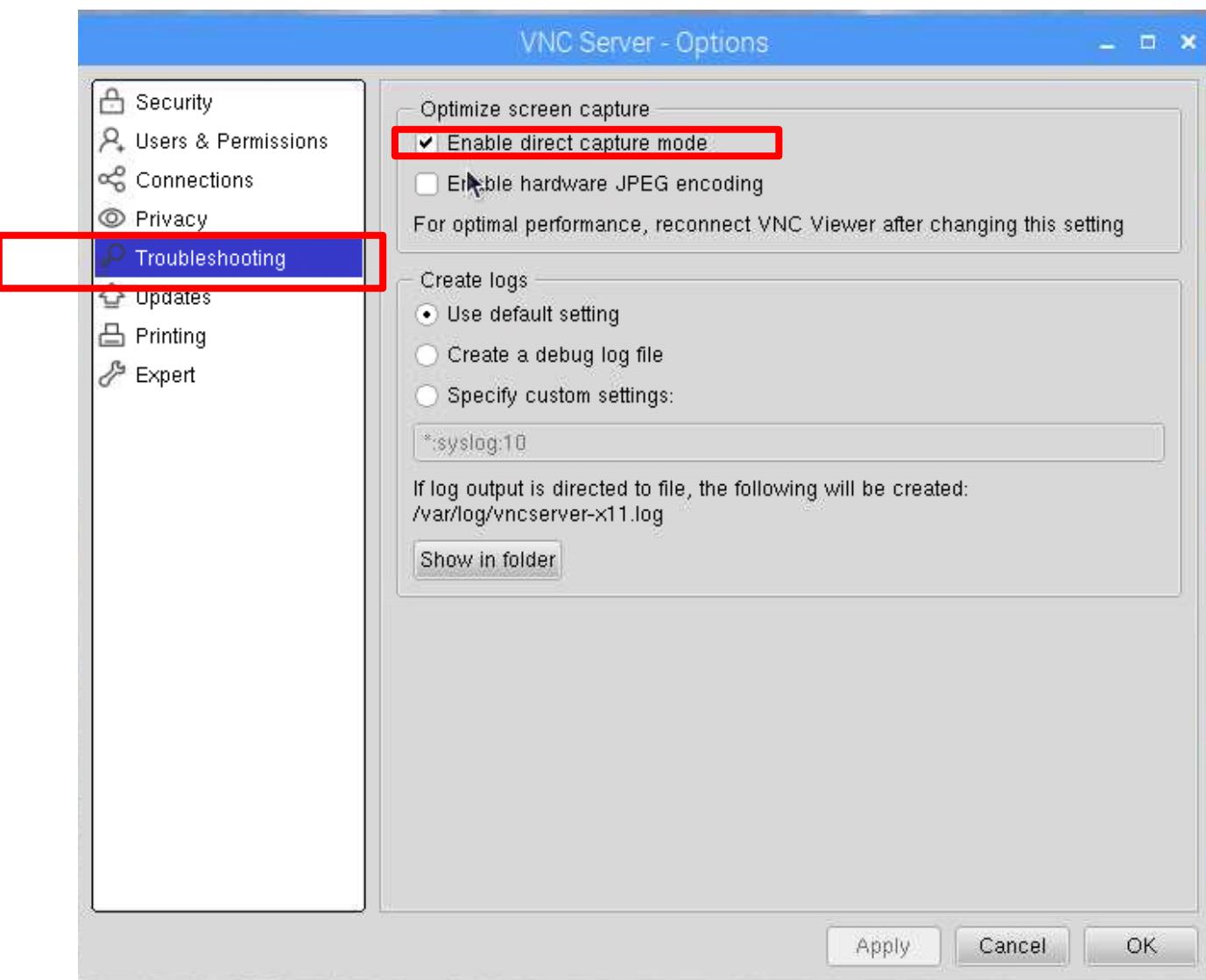


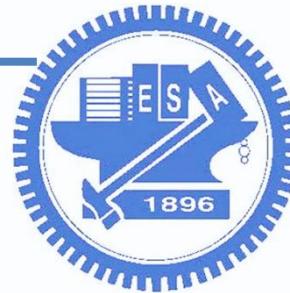
A. VNC額外設定





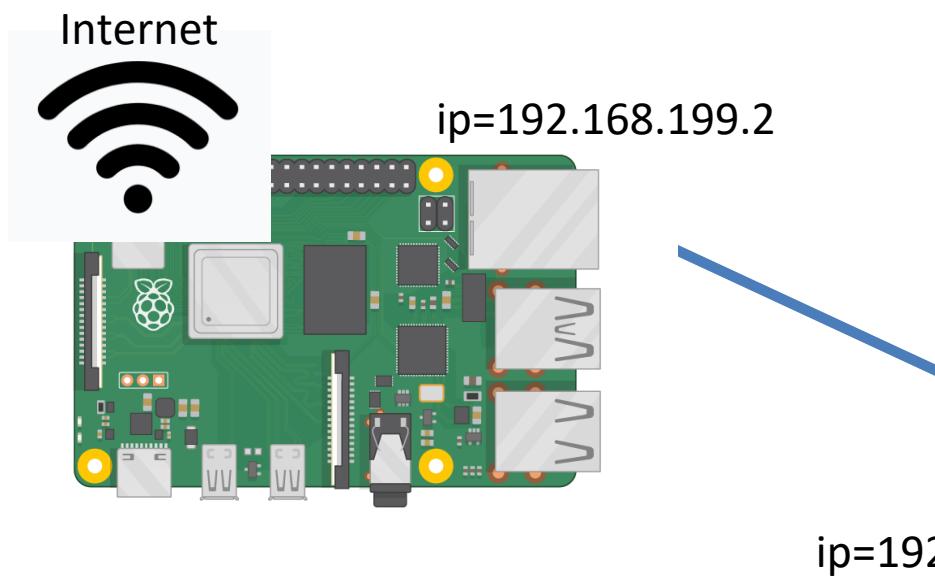
A. VNC額外設定





B. 設定有線網路對接

- 設定static IP address (PI與電腦互聯)





B. 設定有線網路對接

□ PI端:

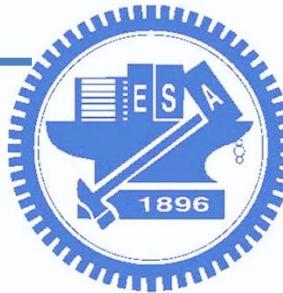
- sudo nano /boot/cmdline.txt
- 最後加上 ip=192.168.199.2

```
console=serial0,115200 console=tty1 root=PARTUUID=fba96bfa-02
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait
plymouth.ignore-serial-consoles ip=192.168.199.2
```

□ 電腦端:

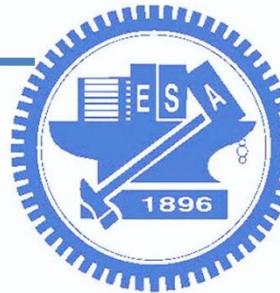
- 有線網路設定:
192.168.199.1/255.255.255.0





Outline

- 1. 安裝OS (Raspbian)
- 2. PI的環境設定
 - A. 系統工具: raspi-config (擴充SD卡空間, 開啟interface...等)
 - B. 使用apt-get安裝程式
- 3. 設定遠端桌面連線
 - A. 內建的realvnc
 - B. 設定有線網路對接
- 4. GPIO + Python + LED
 - A. GPIO introduction
 - B. Python example
 - C. 電子零件與控制LED燈
- 5. 傳輸檔案到PI

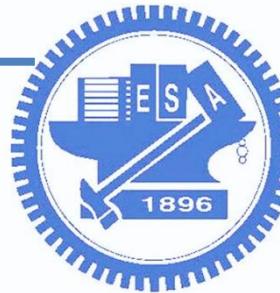


4. GPIO introduction

- General-purpose input/output (GPIO)
 - You can set PIN as **Input** or **Output** or **both**(Input and output)
 - Input: write a value on PIN
 - Output: Read the value on PIN



<https://www.raspberrypi.org/documentation/usage/gpio/README.md>



4. GPIO introduction

- Pin number != GPIO number
 - Physical numbering vs. GPIO numbering

Raspberry Pi Pinout

3v3 Power	1	5v Power
GPIO 2 (I2C1 SDA)	3	5v Power
GPIO 3 (I2C1 SCL)	5	Ground
GPIO 4 (GPCLK0)	7	GPIO 14 (UART TX)
Ground	9	GPIO 15 (UART RX)
GPIO 17	11	GPIO 18 (PCM CLK)
GPIO 27	13	Ground
GPIO 22	15	GPIO 23
3v3 Power	17	GPIO 24
GPIO 10 (SPI0 MOSI)	19	Ground
GPIO 9 (SPI0 MISO)	21	GPIO 25
GPIO 11 (SPI0 SCLK)	23	GPIO 8 (SPI0 CE0)
Ground	25	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27	GPIO 1 (EEPROM SCL)
GPIO 5	29	Ground
GPIO 6	31	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	Ground
GPIO 19 (PCM FS)	35	GPIO 16
GPIO 26	37	GPIO 20 (PCM DIN)
Ground	39	GPIO 21 (PCM DOUT)

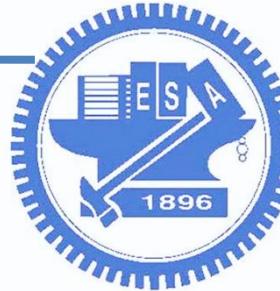
5v Power	SDIO	JTAG	3v3 Power	UART	DPI	PCM	1-WIRE	WiringPi
GPCLK	Ground	I2C	PWM	SPI				

Browse pinouts for HATs, pHATs and add-ons »

GPIO 4

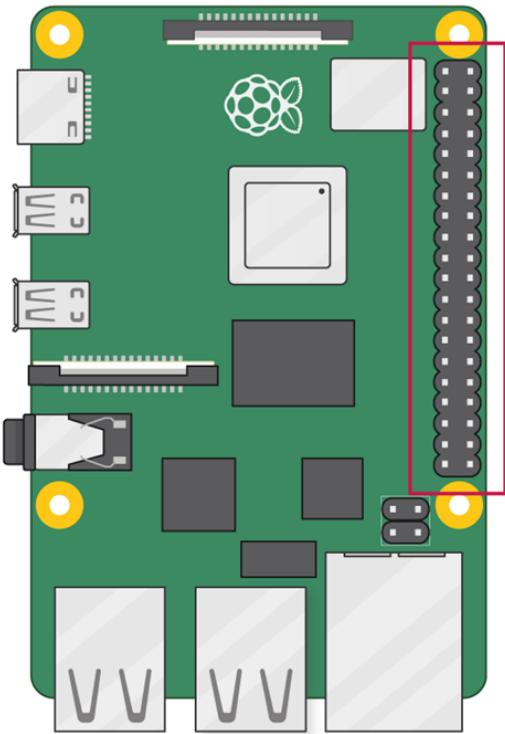
Alt0	Alt1	Alt2	Alt3	Alt4	Alt5
GPCLK0	SMI SA1	DPI D0	AVEOUT VID0	AVEIN VID0	JTAG TDI

- Physical/Board pin 7
- GPIO/BCM pin 4
- Wiring Pi pin 7

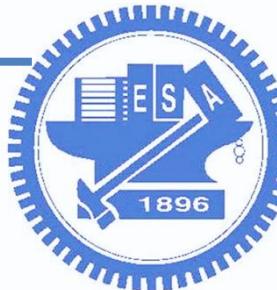


4. GPIO introduction

- The PIN (Physical) numbering is in Z-shape

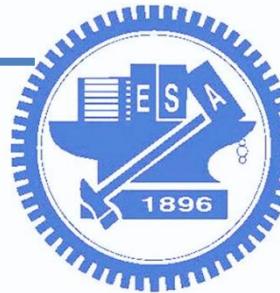


3V3 power	1	2	5V power
GPIO 2 (SDA)	3	4	5V power
GPIO 3 (SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)
Ground	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18 (PCM_CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3V3 power	17	18	GPIO 24
GPIO 10 (MOSI)	19	20	Ground
GPIO 9 (MISO)	21	22	GPIO 25
GPIO 11 (SCLK)	23	24	GPIO 8 (CE0)
Ground	25	26	GPIO 7 (CE1)
GPIO 0 (ID_SD)	27	28	GPIO 1 (ID_SC)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM_FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM_DIN)
Ground	39	40	GPIO 21 (PCM_DOUT)



4. GPIO introduction

- Raspbian有內建gpio查詢指令
 1. pinout
 2. gpio readall
- 如果gpio readall遇到錯誤訊息:
 - Oops - unable to determine board type... model: 17
- 解決方法:
 - wget <https://project-downloads.drogon.net/wiringpi-latest.deb>
 - sudo dpkg -i wiringpi-latest.deb

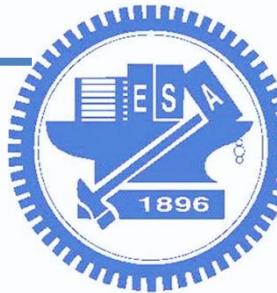


4.1 pinout結果

```
pi@raspberrypi:~$ pinout
,--.
| oooooooooooooooo J8 +====+
| 1ooooooooooooooo PoE | Net
| Wi | Pi Model 4B V1.4 oo +====+
| Fi | | SoC | | | |
| | D | | | | |
| | S | | | | |
| | I | | | | |
| | | | | C | | USB3
| | | | | S | | +====+
| | | | | I | | A | | USB2
| | | | | MI | | MI | | V | | -----
`-| |---|HD|---|HD|---|I|---|A|---|V|-----'
Revision : d03114
SoC : BCM2711
RAM : NoneMb
Storage : MicroSD
USB ports : 4 (excluding power)
Ethernet ports : 1
Wi-fi : True
Bluetooth : True
Camera ports (CSI) : 1
Display ports (DSI) : 1

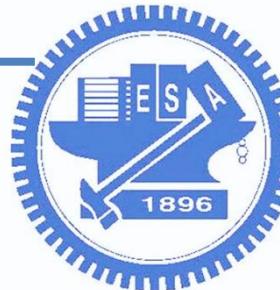
J8:
      3V3  (1) (2)  5V
      GPIO2 (3) (4)  5V
      GPIO3 (5) (6) GND
      GPIO4 (7) (8) GPIO14
      GND   (9) (10) GPIO15
      GPIO17 (11) (12) GPIO18
      GPIO27 (13) (14) GND
      GPIO22 (15) (16) GPIO23
      3V3  (17) (18) GPIO24
      GPIO10 (19) (20) GND
      GPIO9  (21) (22) GPIO25
      GPIO11 (23) (24) GPIO8
      GND   (25) (26) GPIO7
      GPIO0  (27) (28) GPIO1
      GPIO5  (29) (30) GND
      GPIO6  (31) (32) GPIO12
      GPIO13 (33) (34) GND
      GPIO19 (35) (36) GPIO16
      GPIO26 (37) (38) GPIO20
      GND   (39) (40) GPIO21

For further information, please refer to https://pinout.xyz/
pi@raspberrypi:~$
```



4.2 gpio readall 結果

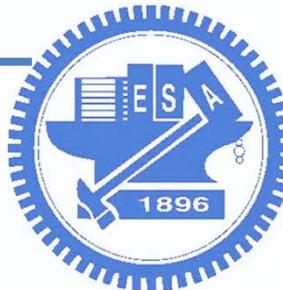
```
pi@raspberrypi:~$ gpio readall
+-----+-----+-----+-----+-----+Pi 4B-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     | 3.3v |      |   |         |   |      | 5v  |      |   |
| 2   | 8   | SDA.1 | ALTO | 1 | 3       | 4 |      | 5v  |      |   |
| 3   | 9   | SCL.1 | ALTO | 1 | 5       | 6 |      | 0v  |      |   |
| 4   | 7   | GPIO. 7 | IN   | 1 | 7       | 8 | ALT5  | TxD | 15  | 14 |
|     |     | 0v    |      |   | 9       | 10| ALT5  | RxD | 16  | 15 |
| 17  | 0   | GPIO. 0 | IN   | 0 | 11      | 12| 0    | IN  | 1   | 18 |
| 27  | 2   | GPIO. 2 | IN   | 0 | 13      | 14|      | 0v  |      |   |
| 22  | 3   | GPIO. 3 | IN   | 0 | 15      | 16| 0    | IN  | 4   | 23 |
|     |     | 3.3v |      |   | 17      | 18| 0    | IN  | 5   | 24 |
| 10  | 12  | MOSI  | IN   | 0 | 19      | 20|      | 0v  |      |   |
| 9   | 13  | MISO  | IN   | 0 | 21      | 22| 0    | IN  | 6   | 25 |
| 11  | 14  | SCLK  | IN   | 0 | 23      | 24| 1    | IN  | 10  | 8  |
|     |     | 0v    |      |   | 25      | 26| 1    | IN  | 11  | 7  |
| 0   | 30  | SDA.0 | IN   | 1 | 27      | 28| 1    | IN  | 31  | 1  |
| 5   | 21  | GPIO.21 | IN   | 1 | 29      | 30|      | 0v  |      |   |
| 6   | 22  | GPIO.22 | IN   | 1 | 31      | 32| 0    | IN  | 26  | 12 |
| 13  | 23  | GPIO.23 | IN   | 0 | 33      | 34|      | 0v  |      |   |
| 19  | 24  | GPIO.24 | IN   | 0 | 35      | 36| 0    | IN  | 27  | 16 |
| 26  | 25  | GPIO.25 | IN   | 0 | 37      | 38| 0    | IN  | 28  | 20 |
|     |     | 0v    |      |   | 39      | 40| 0    | IN  | 29  | 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+Pi 4B-----+-----+-----+-----+
```



4. GPIO Limitations

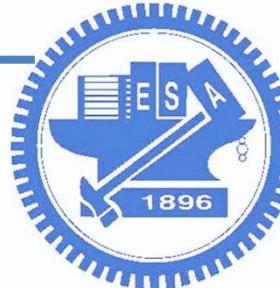
- Do not put more than 3.3V on any GPIO pin being used as an input.
- Do not draw more than 16mA per output and keep the total for all outputs below 50mA in total for an older 26-pin Raspberry PI, and below 100mA on a 40-pin Raspberry PI.
- When using LEDs, 3mA is enough to light a red LED reasonably brightly with a 470 Ohm series resistor.
- Do not poke at the GPIO connector with a screwdriver or any metal object when the PI is powered up.
- Do not power the PI with more than 5V.
- Do not draw more than a total of 250mA from the 5V supply pins.

From: Raspberry Pi Cookbook: Software and Hardware Problems and Solutions
<https://books.google.com.tw/books?id=0skvDAAAQBAJ&pg=PT270&lpg=PT270#v=onepage&q&f=false>



4. GPIO Limitations

- 在GPIO 輸入電壓不可以超過3.3V
- 不要拿金屬物體接觸GPIO PIN (會短路)
 - 使用杜邦線, 針腳不要碰到板子
- 使用GPIO PIN啟動PI時, 電壓不可以超過5V
- GPIO PIN的輸出電流有上限
 - 早期資料
 - 3.3V的供電腳位不可以超過50mA
 - 5V的供電腳位不可以超過250mA
 - 實驗資訊
 - 3.3V的供電腳位大約可以支援500mA



4. GPIO Limitations

4 Electrical Specification

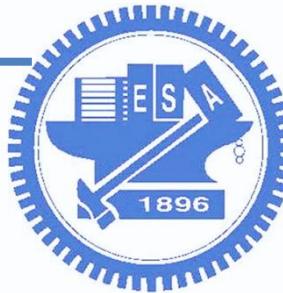
Caution! Stresses above those listed in Table 2 may cause permanent damage to the device. This is a stress rating only; functional operation of the device under these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Symbol	Parameter	Minimum	Maximum	Unit
VIN	5V Input Voltage	-0.5	6.0	V

Table 2: Absolute Maximum Ratings

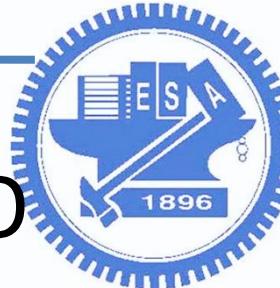
4.1 Power Requirements

The Pi4B requires a good quality USB-C power supply capable of delivering 5V at 3A. If attached downstream USB devices consume less than 500mA, a 5V, 2.5A supply may be used.



Outline

- 1. 安裝OS (Raspbian)
- 2. PI的環境設定
 - A. 系統工具: raspi-config (擴充SD卡空間, 開啟interface...等)
 - B. 使用apt-get安裝程式
- 3. 設定遠端桌面連線
 - A. 內建的realvnc
 - B. 設定有線網路對接
- 4. GPIO + Python + LED
 - A. GPIO introduction
 - B. Python example
 - C. 電子零件與控制LED燈
- 5. 傳輸檔案到PI



4. Programming language for GPIO

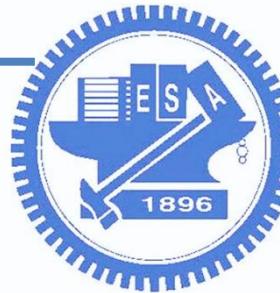
- C
- C + wiringPi
- C#
- Ruby
- Perl
- **Python**
- Scratch
- Java Pi4J Library
- Shell script



We use Python3

Sunsetting Python 2

We are volunteers who make and take care of the Python programming language. We have decided that January 1, 2020, was the day that we sunset Python 2. That means that we will not improve it anymore after that day, even if someone finds a security problem in it. You should upgrade to Python 3 as soon as you can.



4. Python example

```
# pound key is annotation  
i = 3                                # set i to 3  
print i  
i = [1, 2, 3, 4, 5]                      # set i to an array  
print i[2]                             # print number 3 (start from 0)  
i = "abcde"                           # set i to a string  
print i[0]                            # print a
```

```
# import MODULE  
import RPi.GPIO  
  
# import MODULE as ALIAS  
import RPi.GPIO as GPIO
```

```
GPIO.setmode(GPIO.BOARD)  
GPIO.setup(12, GPIO.OUT)
```

```
for i in xrange(start, stop[, step]) :  
    process
```

**It does not use {} in the code.
It uses 4 space in for-loop, if-else...**

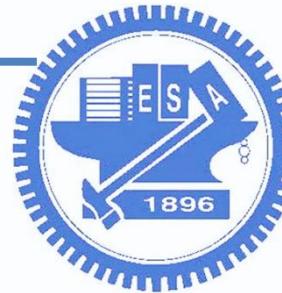


4 space here



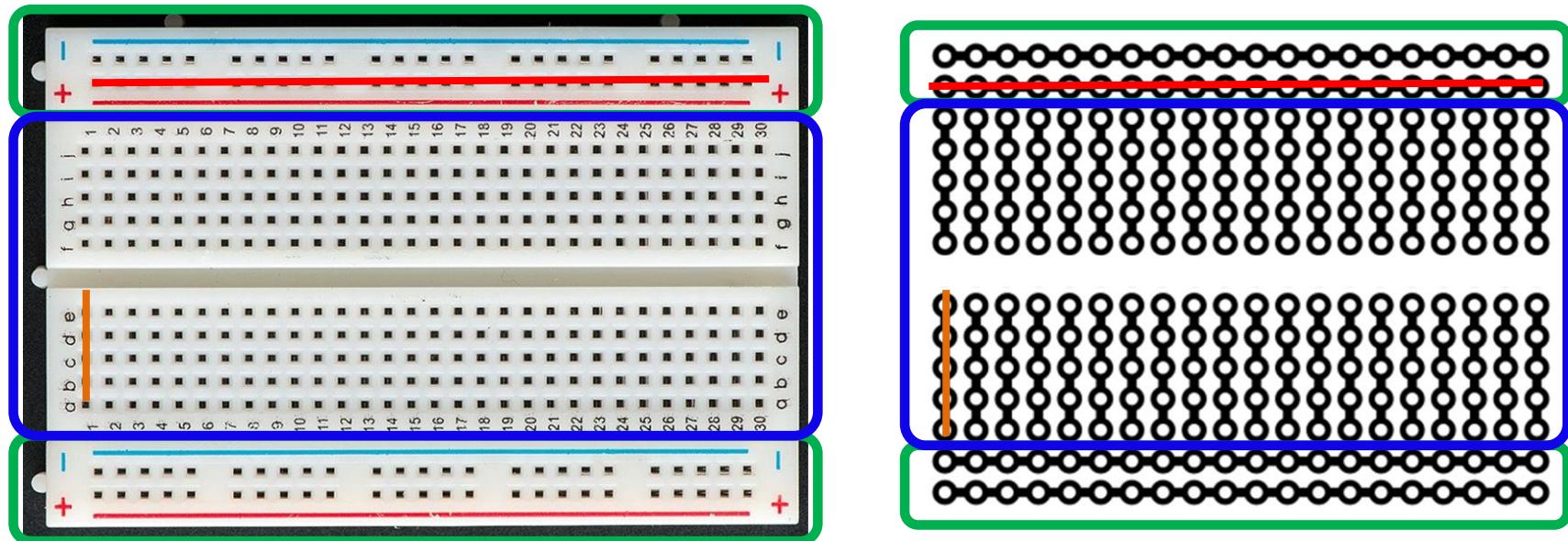
Outline

- 1. 安裝OS (Raspbian)
- 2. PI的環境設定
 - A. 系統工具: raspi-config (擴充SD卡空間, 開啟interface...等)
 - B. 使用apt-get安裝程式
- 3. 設定遠端桌面連線
 - A. 內建的realvnc
 - B. 設定有線網路對接
- 4. GPIO + Python + LED
 - A. GPIO introduction
 - B. Python example
 - C. 電子零件與控制LED燈
- 5. 傳輸檔案到PI

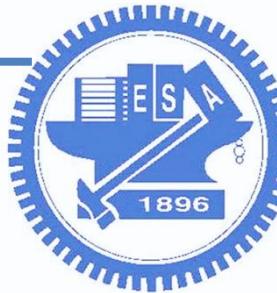


Bread Board

Bus strips: one for ground (-) and one for a supply voltage (+)

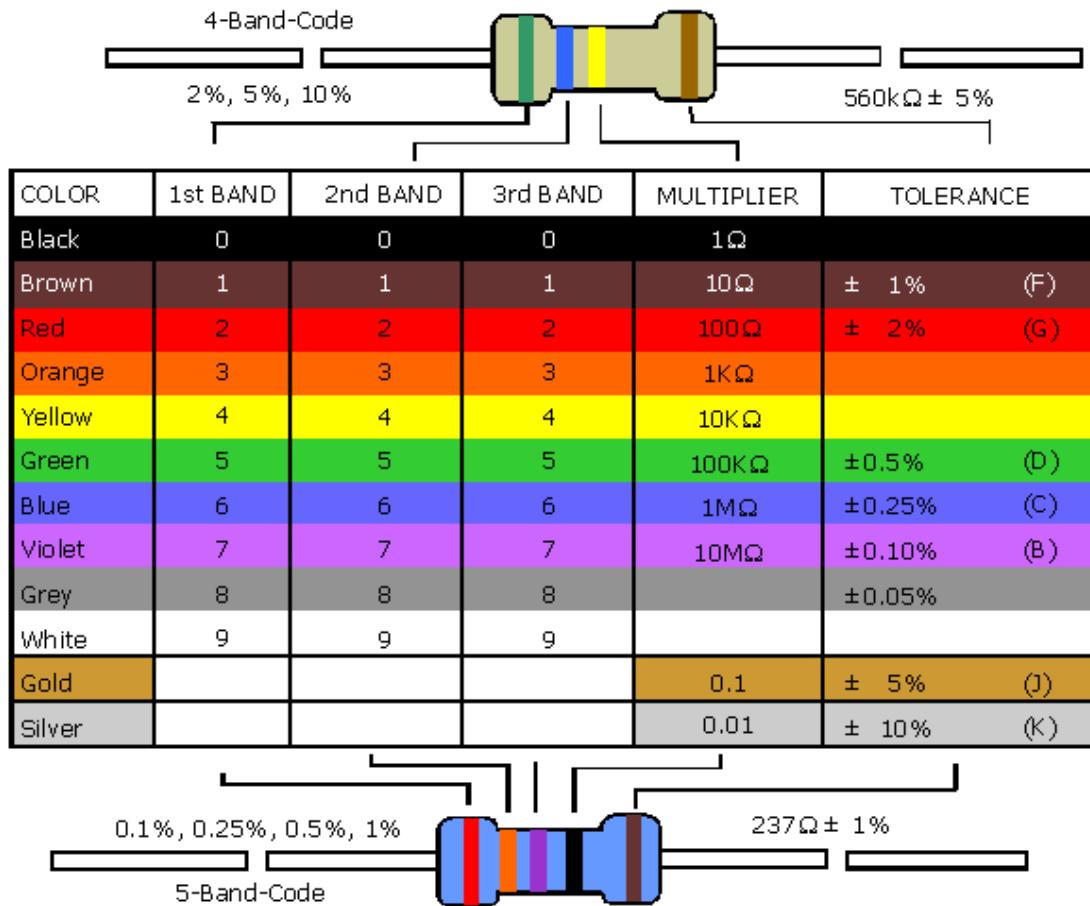


Terminal strips (ex: a1, b1, c1, d1 and e1 are connected)



Resistor

use color table to determine value

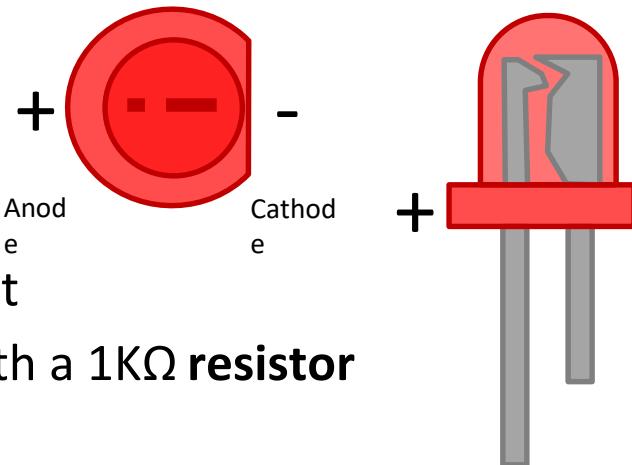


Ex: $20K\Omega$ = Red, Black, Orange = $20 * 1000$ (4-band)
= Red, Black, Black, Red = $200 * 100$ (5-band)



LED

- The **LED (Light Emitting Diode)** is a simple, digital **actuator**
- LEDs have a **short leg (-)** and a **long leg (+)** and it matters how they are oriented in a circuit
- To prevent damage, LEDs are used together with a **1KΩ resistor** (or anything from 300Ω to $2K\Omega$)



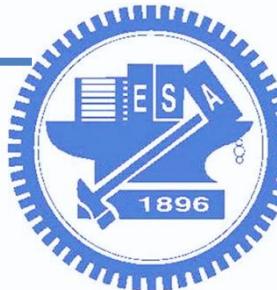
Electrical / Optical Characteristics at TA=25°C

Symbol	Parameter	Device	Typ.	Max.	Units	Test Conditions
λ_{peak}	Peak Wavelength	Super Bright Red	660		nm	$I_f=20mA$
λ_D [1]	Dominant Wavelength	Super Bright Red	640		nm	$I_f=20mA$
$\Delta\lambda_{1/2}$	Spectral Line Half-width	Super Bright Red	20		nm	$I_f=20mA$
C	Capacitance	Super Bright Red	45		pF	$V_f=0V; f=1MHz$
V_F [2]	Forward Voltage	Super Bright Red	1.85	2.5	V	$I_f=20mA$
I_R	Reverse Current	Super Bright Red		10	uA	$V_R = 5V$

Notes:

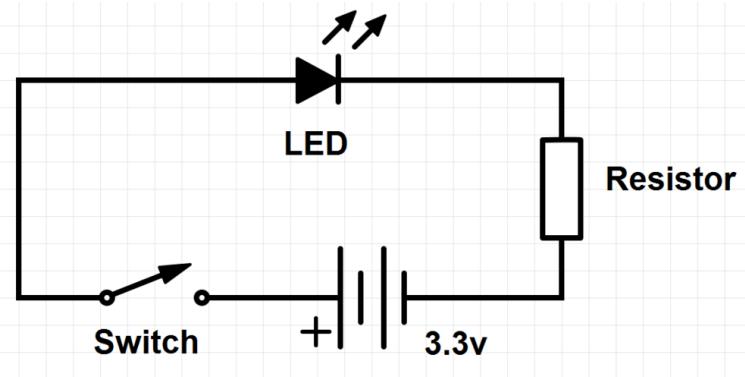
1. Wavelength: +/-1nm.

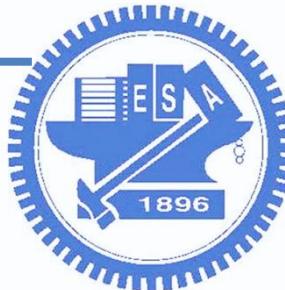
2. Forward Voltage: +/-0.1V.



LED

- 根據前面的參數
 - PI的GPIO腳位可以提供3.3V
 - LED的順向電壓是1.85V
 - LED的電流需要20mA
- 電阻公式: $R=V/I$
 - $R = (3.3 - 1.85) / 0.02 = 72.5$ 歐姆
 - 最少須接 72.5 歐姆的電阻，才可避免 LED 燒毀

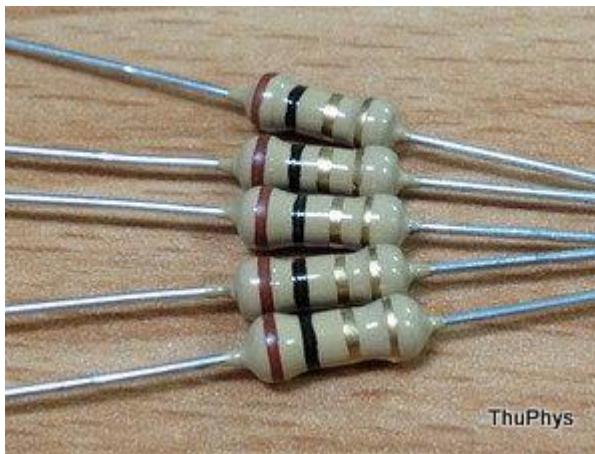




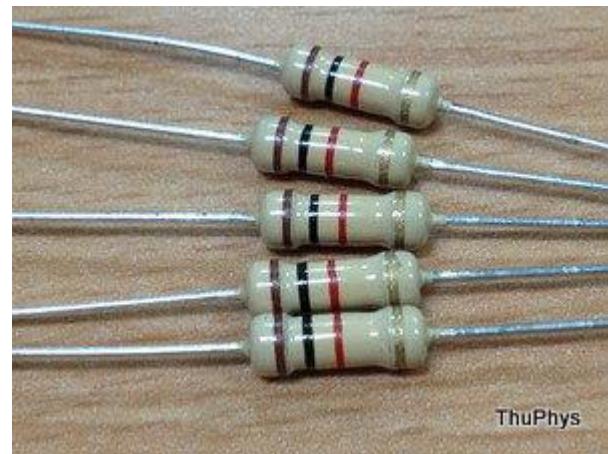
Discussion 1

- Identify the resistors. (Ω)
 - Write down the color which you see
 - Identify the corresponding resistor value
 - How to calculate the value?

A



B



C

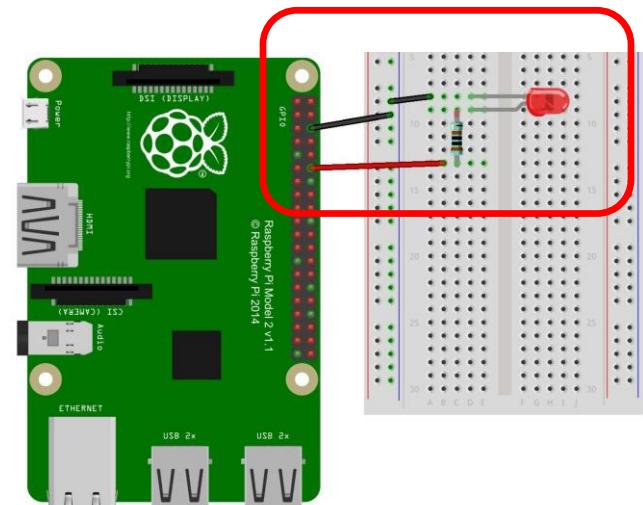
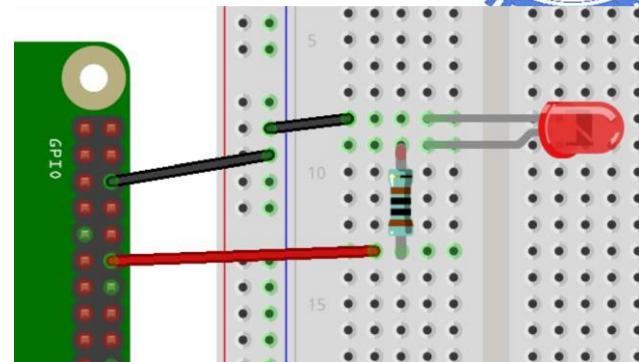
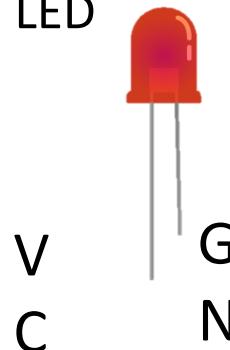




Control LED

- **Goal:** create a simplest thing to control GPIO and see physical output
- **Output:** it blinks an LED.
- **Hardware Required**

- Raspberry PI
- LED





Control LED by Python

```
import RPi.GPIO as GPIO  
import time
```

Load GPIO library

```
LED_PIN = 12  
GPIO.setmode(GPIO.BOARD)  
GPIO.setup(LED_PIN, GPIO.OUT)
```

LED is on pin 12 by pin numbering (z-shape)

```
try:  
    while True:  
        print("LED is on")  
        GPIO.output(LED_PIN, GPIO.HIGH)  
        time.sleep(1)  
        print("LED is off")  
        GPIO.output(LED_PIN, GPIO.LOW)  
        time.sleep(1)
```

The **try** clause (the statement(s) between the try and except keywords) is executed.

```
except KeyboardInterrupt:  
    print "Exception: KeyboardInterrupt"
```

A user-generated **interruption** is signaled (ctrl + c)

```
finally:  
    GPIO.output(LED_PIN, GPIO.LOW)  
    GPIO.cleanup()
```

A **finally** clause is always executed before leaving the try statement, whether an exception has occurred or not.



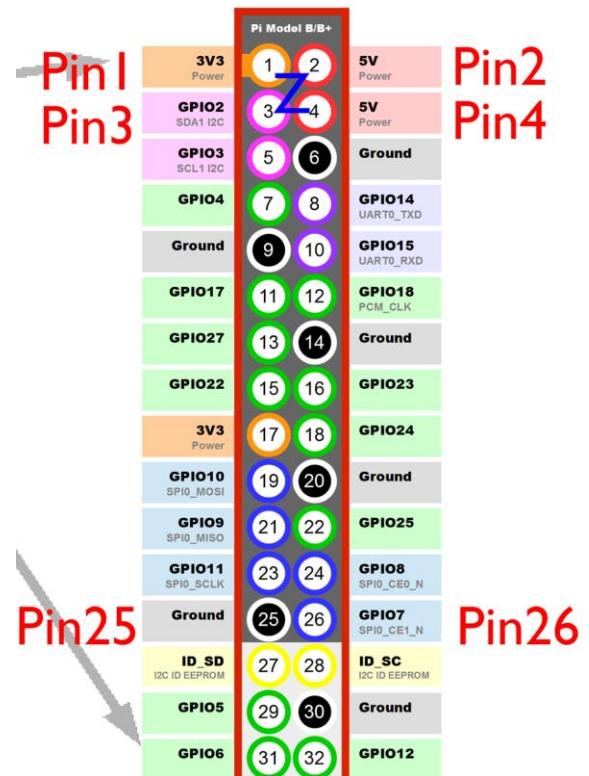
Syntax - GPIO

- `GPIO.setmode(GPIO.BCM)` // two parameters
 - `GPIO.BCM`: Define the pins by the number of the pin plug (z-shape)
 - `GPIO.BCM`: Define the pins by the "Broadcom SOC channel" number

- `GPIO.setup(LED_PIN, GPIO.OUT)`
 - Set `LED_PIN` to output mode

- `GPIO.cleanup()`
 - clean up all the ports you've used

[Sad story] when you have a port set HIGH as an output and you accidentally connect it to GND (LOW), which would short-circuit the port and possibly fry it.

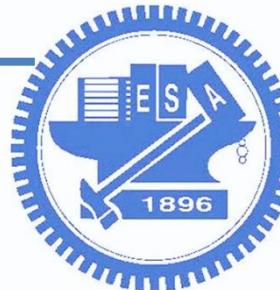




Syntax – Python error statement

- The **try** statement works as follows.
 - First, the **try** clause (the statement(s) between the try and except keywords) is executed.
 - If no exception occurs, the **except** clause is skipped and execution of the try statement is finished.
 - If an exception occurs during execution of the try clause, the rest of the clause is skipped. Then if its type matches the exception named after the except keyword, the except clause is executed, and then execution continues after the try statement.
 - If an exception occurs which does not match the exception named in the except clause, it is passed on to outer try statements; if no handler is found, it is an unhandled exception and execution stops with a message as shown above.

```
try:  
    while True:  
        print("LED is on")  
        GPIO.output(LED_PIN, GPIO.HIGH)  
        time.sleep(1)  
        print("LED is off")  
        GPIO.output(LED_PIN, GPIO.LOW)  
        time.sleep(1)  
  
    except KeyboardInterrupt:  
        print "Exception: KeyboardInterrupt"  
  
    finally:  
        GPIO.cleanup()
```



How to write Python code on PI?

- Use text editor (ex: nano, vim ... etc)
 - Ex: **nano blink.py**
- Write code, then save it
 - In nano, press **ctrl + x** to exit
- Execution (use **ctrl + c** to stop)
 - **sudo python3 blink.py**

```
COM8 [80x24]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)
pi@raspberrypi:~$ python blink_led2.py
LED is on
LED is off
LED is on
LED is off
[green LED icon]
(use ctrl + c to stop)
```



Outline

- 1. 安裝OS (Raspbian)
- 2. PI的環境設定
 - A. 系統工具: raspi-config (擴充SD卡空間, 開啟interface...等)
 - B. 使用apt-get安裝程式
- 3. 設定遠端桌面連線
 - A. 內建的realvnc
 - B. 設定有線網路對接
- 4. GPIO + Python + LED
 - A. GPIO introduction
 - B. Python example
 - C. 電子零件與控制LED燈
- 5. 傳輸檔案到PI



5. 傳輸檔案到PI

- 從網路上下載檔案, ex: 溫溼度library
 - https://github.com/adafruit/Adafruit_Python_DHT
- 在PC寫程式, 再複製到開發板
- A. 直接在開發板下載檔案
 - wget, git clone
- B. 透過網路傳輸檔案
 - Winscp, Python web server
- C. 在電腦下載, 用USB隨身碟複製過去
 - Linux操作指令, mount



A. 直接用PI下載檔案

[adafruit / Adafruit_Python_DHT](#)

Watch 97 ★ Star 765 Fork 607

Code Issues 9 Pull requests 10 Projects 0 Insights

Join GitHub today
GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#) Dismiss

Python library to read the DHT series of humidity and temperature sensors on a Raspberry Pi or Beaglebone Black.

71 commits 4 branches 3 releases 16 contributors MIT

Branch: master New pull request Find file Clone or download ▾

Brennen Bearnes version bump to 1.4.0 in setup.py
.github Add GitHub pull request template
Adafruit_DHT add Pi 3b+ support
examples Improve error message
source Issue #49 - fix long cable read delay on DHT sensor with RPI 1
.gitignore Update version after merging pulls.
LICENSE Initial commit
MANIFEST.in Prepared for installation with pip.
README.md updated install instructions per @juliogonzalez suggestions
setup.py version bump to 1.4.0 in setup.py

Clone with HTTPS
Use Git or checkout with SVN using the web URL.
https://github.com/adafruit/Adafruit_Pytl

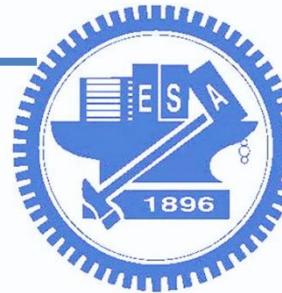
Open in Desktop Download ZIP

複製下載連結

在新分頁中開啟連結(T)
在新視窗中開啟連結(W)
在無痕式視窗中開啟連結(G)
另存為(S...)
複製連結網址(E)

檢查(N)

2 months ago



A. 直接用PI下載檔案

- 在終端機執行 wget 指令

- Ex: wget

```
https://github.com/adafruit/Adafruit_Python_DHT/archive/master.zip
  (COM8) [80x24]
  -  □  ×
  連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)
pi@raspberrypi:~$ wget https://github.com/adafruit/Adafruit_Python_DHT/archive/m
aster.zip
--2019-01-14 07:57:02--  https://github.com/adafruit/Adafruit_Python_DHT/archive/
/master.zip
Resolving github.com (github.com)... 192.30.253.112, 192.30.253.113
Connecting to github.com (github.com)|192.30.253.112|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/adafruit/Adafruit_Python_DHT/zip/master [f
ollowing]
--2019-01-14 07:57:04--  https://codeload.github.com/adafruit/Adafruit_Python_DH
T/zip/master
Resolving codeload.github.com (codeload.github.com)... 192.30.253.121, 192.30.25
3.120
Connecting to codeload.github.com (codeload.github.com)|192.30.253.121|:443... c
onnected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: ? 黃aster.zip? ?

master.zip          [    =>                      ]  54.30K  125KB/s   in 0.4s

2019-01-14 07:57:05 (125 KB/s) - ? 黃aster.zip? ? saved [55607]
pi@raspberrypi:~$
```



A. 直接用PI下載檔案

□ 下載完畢

```
(COM8) [80x24]
 連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)
Connecting to codeload.github.com (codeload.github.com) |192.30.253.121|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: ? 黃aster.zip? ?

master.zip          [ => ] 54.30K  125KB/s   in 0.4s

2019-01-14 07:57:05 (125 KB/s) - ? 黃aster.zip? ? saved [55607]

pi@raspberrypi:~$ ls -l
total 96
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Desktop
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Documents
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Downloads
drwxr-xr-x 2 pi pi  4096 Nov 13 13:45 MagPi
-rw-r--r-- 1 pi pi 55607 Jan 14 07:57 master.zip
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Music
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Pictures
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Public
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Templates
drwxr-xr-t 2 pi pi  4096 Jan 10 11:00 thinclient_drives
drwxr-xr-x 2 pi pi  4096 Nov 13 14:25 Videos
pi@raspberrypi:~$
```



B. 透過網路傳輸檔案

□ 網路架構:

- Case 1: PI 有 Public IP address (通常不可能)
- Case 2: 電腦與PI在同一個子網路 (較常見)
- Case 3: 設定Wi-Fi熱點在電腦 or PI (win10筆電有熱點功能)
- Case 4: PI與筆電用網路線對接





(a) Create Wi-Fi hotspot on Laptop

- Create your own Wi-Fi hotspot on windows 10





(a) Create Wi-Fi hotspot on Laptop

- Create your own Wi-Fi hotspot on windows 10

設定

首頁

尋找設定

網路和網際網路

狀態

Wi-Fi

撥號

VPN

飛航模式*

行動熱點

數據使用量

Proxy

行動熱點

開啟

從下列來源共用我的網際網路連線

Wi-Fi

透過下列來源共用我的網際網路連線

Wi-Fi

藍牙

網路名稱: DESKTOP-N7N29HF 2820

網路密碼: wufish5566

網路頻帶: 2.4 GHz

編輯

裝置已連接: 0 個 (共 8 個)

相關設定

變更介面卡選項

網路和共用中心

Windows 防火牆

有任何疑問嗎?

取得協助

編輯網路資訊

變更其他人使用您分享的連線時所需的網路名稱與密碼。

網路名稱

DESKTOP-N7N29HF 2820

網路密碼 (至少 8 個字元)

網路頻帶

2.4 GHz

儲存 取消



B. 透過網路傳輸檔案

- 1. python有內建simple http server
 - 指令: python -m SimpleHTTPServer 8000
 - 電腦端可以直接透過網頁下載PI裡面的檔案

(COM8) [80x24]

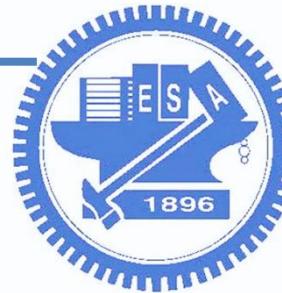
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

```
pi@raspberrypi:~$ python -m SimpleHTTPServer 8000
Serving HTTP on 0.0.0.0 port 8000 ...
192.168.1.80 - - [14/Jan/2019 08:01:15] "GET / HTTP/1.1" 200 -
192.168.1.80 - - [14/Jan/2019 08:01:16] code 404, message File not found
192.168.1.80 - - [14/Jan/2019 08:01:16] "GET /favicon.ico HTTP/1.1" 404 -
192.168.1.80 - - [14/Jan/2019 08:01:26] "GET /Desktop/ HTTP/1.1" 200 -
192.168.1.80 - - [14/Jan/2019 08:01:26] "GET /Desktop/ HTTP/1.1" 200 -
-----
Exception happened during processing of request from ('192.168.1.80', 64505)
Traceback (most recent call last):
  File "/usr/lib/python2.7/SocketServer.py", line 290, in _handle_request_noblock
    self.process_request(request, client_address)
  File "/usr/lib/python2.7/SocketServer.py", line 318, in process_request
    self.finish_request(request, client_address)
  File "/usr/lib/python2.7/SocketServer.py", line 331, in finish_request
    self.RequestHandlerClass(request, client_address, self)
  File "/usr/lib/python2.7/SocketServer.py", line 654, in __init__
    self.finish()
  File "/usr/lib/python2.7/SocketServer.py", line 713, in finish
    self.wfile.close()
  File "/usr/lib/python2.7/socket.py", line 283, in close
    self.flush()
  File "/usr/lib/python2.7/socket.py", line 307, in flush
```

← → C ⌂ ⓘ 不安全 | 192.168.1.48:8000

Directory listing for /

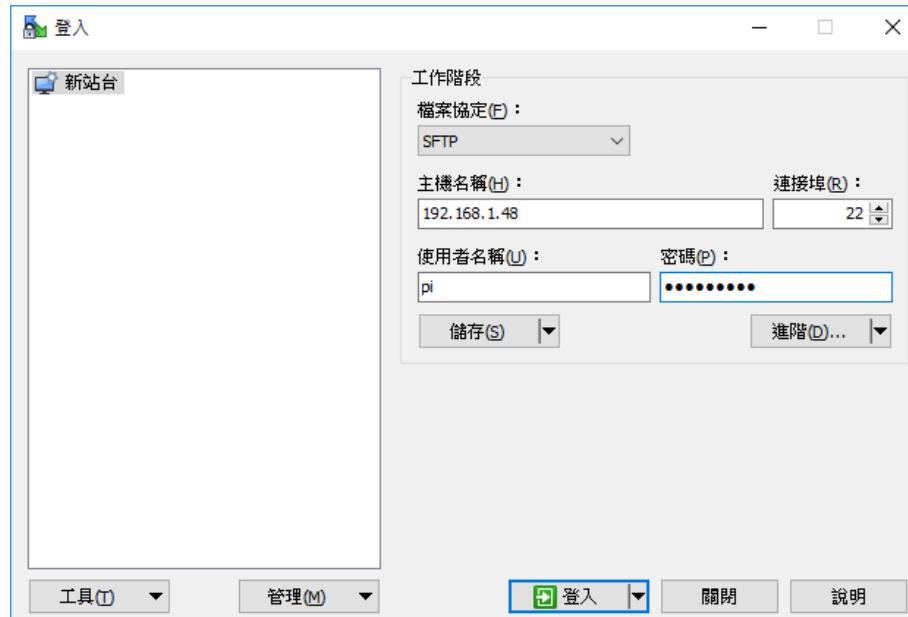
- [.bash_history](#)
- [.bash_logout](#)
- [.bashrc](#)
- [.cache/](#)
- [.config/](#)
- [.gnupg/](#)
- [.local/](#)
- [.nano/](#)
- [.profile](#)
- [.wget-hsts](#)
- [.Xauthority](#)
- [.xsession-errors](#)
- [.xsession-errors.old](#)
- [.Desktop/](#)
- [.Documents/](#)
- [.Downloads/](#)
- [.MagPi/](#)
- [.master.zip](#)
- [.Music/](#)
- [.Pictures/](#)
- [.Public/](#)
- [.Templates/](#)
- [.thinclient_drives/](#)
- [.Videos/](#)



B. 透過網路傳輸檔案

□ 2. winscp:

- 在電腦上安裝winscp, 輸入IP的ip address與帳密, 即可連線傳輸資料(如同ftp介面)
- 需先啟用ssh login功能 (sudo raspi-config)





B. 透過網路傳輸檔案

□ 2. winscp 使用畫面

The screenshot shows the WinSCP interface with two panes. The left pane shows the local directory structure under 'C:\Users\wufish\Documents\' on the left. The right pane shows the remote directory structure under '/home/pi/' on the right. A file named 'master.zip' is selected in the right pane. The bottom status bar indicates '0 B / 0 B 在 0 / 9 個項目' (0 B / 0 B in 0 / 9 items), '54.3 KB / 54.3 KB 在 1 / 11 個項目' (54.3 KB / 54.3 KB in 1 / 11 items), and '14 個隱藏項目' (14 hidden items). The bottom navigation bar includes icons for lock, SFTP-3, and timestamp (0:32:51).

名稱	大小	最後修改時間	權限	擁有者
..		2018/11/13 下午 09:09...	rwxr-xr-x	root
Any Video Converter		2018/11/13 下午 10:25...	rwxr-xr-x	pi
Arduino		2018/11/13 下午 10:25...	rwxr-xr-x	pi
DisplayFusion Backu...		2018/11/13 下午 10:25...	rwxr-xr-x	pi
Goodreader		2018/11/13 下午 10:25...	rwxr-xr-x	pi
RasPi		2018/11/13 下午 09:45...	rwxr-xr-x	pi
VBox_share		2018/11/13 下午 10:25...	rwxr-xr-x	pi
Visual Studio 2015		2018/11/13 下午 10:25...	rwxr-xr-x	pi
自訂 Office 範本		2018/11/13 下午 10:25...	rwxr-xr-x	pi
我的圖形		2019/1/10 下午 07:00...	rwxr-xr-t	pi
Desktop		2018/11/13 下午 10:25...	rwxr-xr-x	pi
Documents		2018/11/13 下午 10:25...	rwxr-xr-x	pi
Downloads		2018/11/13 下午 10:25...	rwxr-xr-x	pi
MagPi		2018/11/13 下午 10:25...	rwxr-xr-x	pi
Music		2018/11/13 下午 10:25...	rwxr-xr-x	pi
Pictures		2018/11/13 下午 10:25...	rwxr-xr-x	pi
Public		2018/11/13 下午 10:25...	rwxr-xr-x	pi
Templates		2018/11/13 下午 10:25...	rwxr-xr-x	pi
thinclient_drives		2018/11/13 下午 10:25...	rwxr-xr-x	pi
Videos		2018/11/13 下午 10:25...	rwxr-xr-x	pi
master.zip	55 KB	2019/1/14 下午 03:57:...	rw-r--r--	pi

winscp 使用畫面



pi - pi@192.168.1.48 - WinSCP

本機(L) 標記(M) 檔案(F) 指令(C) 工作階段(S) 選項(O) 遠端(R) 說明(H)

同步 行列 傳送設定 預設

pi@192.168.1.48 x 新工作階段

我的文件 上傳 下載 檔案屬性 新增

C:\Users\wufish\Documents\ pi\home/pi/

名稱	大小	類型	最後修改
..		上層目錄	2020/2/1
Fax		檔案資料夾	2020/1/1
G雲端硬碟		檔案資料夾	2019/12
Scanned Documents		檔案資料夾	2020/2/1
自訂 Office 範本		檔案資料夾	2019/12
我的圖形		檔案資料夾	2019/12
Desktop			2020/2/13 下午 11:55:52
Documents			2020/2/20 下午 02:38:06
Downloads			2020/2/20 下午 02:38:07
MagPi			2020/2/20 下午 02:38:07
Music			2020/2/20 下午 02:38:07
Pictures			2020/2/20 下午 02:38:07
Public			2020/2/20 下午 02:38:07
Templates			2020/2/20 下午 02:38:07
Videos			2020/2/20 下午 02:38:07

檔案 常用 共用 檢視

← → ↑ ↓ Sample code > w5+6_GY801 搜尋 w5+6_GY801

下載 文件 圖片 Google 雲端硬碟 Google Drive File Stream OneDrive 本機 3D 物件 下載 文件 音樂 桌面

10 個項目 選取 1 個項目 4.38 KB

datasheet.github sample sol - code 暫存資料 1accQ.py 2gyroQ.py 3calibrate-hmc5883l.py 3magQ.py 4barQ.py i2cutils.py

修改日期 類型

名稱	修改日期	類型
datasheet	2020/2/12 上午 10:39	檔案資料夾
github sample	2020/2/4 下午 01:31	檔案資料夾
sol - code	2020/2/17 下午 03:04	檔案資料夾
暫存資料	2020/2/17 下午 02:47	檔案資料夾
1accQ.py	2020/2/7 下午 05:36	PY 檔案
2gyroQ.py	2020/2/10 下午 05:41	PY 檔案
3calibrate-hmc5883l.py	2020/2/17 下午 02:45	PY 檔案
3magQ.py	2020/2/17 下午 02:54	PY 檔案
4barQ.py	2020/2/17 下午 02:51	PY 檔案
i2cutils.py	2020/2/6 上午 11:38	PY 檔案

COM6 - Putty

```
Setting up libglx-dev:armhf (1.3.0-7-bpo10+1) ...
Setting up libxi-dev:armhf (2:1.7.9-1) ...
Setting up libxrender-dev:armhf (1:0.9.10-1) ...
Setting up libgl1-dev:armhf (1.3.0-7-bpo10+1) ...
Setting up libxf86-dev:armhf (2:3.2-2) ...
Setting up libxtst-dev:armhf (2:1.2.3-1) ...
Setting up libxdamage-dev:armhf (1:1.1.4-3+b3) ...
Setting up libatspi2.0-dev:armhf (2.30.0-7) ...
Setting up libegl-dev:armhf (1.3.0-7-bpo10+1) ...
Setting up libcomposite-dev:armhf (1:0.4.4-2) ...
Setting up libxcursor-dev:armhf (1:1.1.15-2) ...
Setting up libatk-bridge2.0-dev:armhf (2.30.0-5) ...
Setting up libxrandr-dev:armhf (2:1.5.1-1) ...
Setting up libxinerama-dev:armhf (2:1.1.4-2) ...
Setting up libglib1-mesa-dev:armhf (19.3.2-1-bpo10+1-rpt1) ...
Setting up libcairo2-dev:armhf (1.16.0-4+rpt1) ...
Setting up libegl1-mesa-dev:armhf (19.3.2-1-bpo10+1-rpt1) ...
Setting up libepoxy-dev:armhf (1.5.3-0.1) ...
Setting up libpango1.0-dev:armhf (1.42.4-7-deb10u1) ...
Setting up libgtk-3-dev:armhf (3.24.5-1+rpt2) ...
pi@raspberrypi:~ ls
desktop Downloads Music Public Videos
Documents MagPi Pictures Templates
pi@raspberrypi:~
```

E:\我的雲端硬碟\NCTU (Course data)\2020(春)-嵌入式應用\Sample code\w5+6_GY801\1accQ.py - Notepad++

檔案(F) 檔案(E) 檢視(V) 語言(N) 設定(I) 工具(O) 巨集(M) 執行(R) 外掛(P) 視窗(W) ?

dashboard-tilteddetector.json 1accQ.py

```
1 import smbus
2 import time
3 from math import *
4
5 bus = smbus.SMBus(1); # 0 for R-Pi Rev. 1, 1 for Rev. 2
6
7 EARTH_GRAVITY_MS2 = 9.80665 # m/s2
8
9 ADXL345_ADDRESS = 0x53
10
11 ADXL345_BW_RATE = 0x2C
12 ADXL345_POWER_CTL = 0x2D
13 ADXL345_DATA_FORMAT = 0x31
14 ADXL345_DATA_X0 = 0x32
15 ADXL345_DATA_Y0 = 0x34
16 ADXL345_DATA_Z0 = 0x36
17 ADXL345_SCALE_MULTIPLIER = ??? # G/LSP
18 ADXL345_BW_RATE_100HZ = 0x??
19 ADXL345_MEASURE = 0x??
20
21 class IMU(object):
22
23     def write_byte(self,addr,value):
24         bus.write_byte_data(self.ADDRESS, addr, value)
25
26     def read_byte(self,addr):
27         return bus.read_byte_data(self.ADDRESS, addr)
28
29     def read_word(self,addr,rf=1):
30         # rf=1 Little Endian Format, rf=0 Big Endian Format
31         if (rf == 1):
32             low = self.read_byte(addr)
33             high = self.read_byte(addr+1)
34         else:
35             high = self.read_byte(addr)
36             low = self.read_byte(addr+1)
37         val = (high << 8) + low
38         return val
39
40     def read_word_2c(self,addr,rf=1):
41         val = self.read_word(addr,rf)
42         if(val & (1 << 16 - 1)):
43             return val - (1<<16)
44         else:
45             return val
46
47 class gy801(object):
48     def __init__(self):
49         self.accel = ADXL345()
50
51 class ADXL345(IMU):
52
53     ADDRESS = ADXL345_ADDRESS
54
55     def __init__(self):
56         #Class Properties
57         self.Xoffset = 0.0
58         self.Yoffset = 0.0
59         self.Zoffset = 0.0
60         self.Xraw = 0.0
61         self.Yraw = 0.0
62         self.Zraw = 0.0
63         self.Xg = 0.0
64         self.Vg = 0.0
65
66 Python file length:4,489 lines:157 Ln:1 Col:1 Sel:0 | 0 Unix (LF) UTF-8 IN
```



C. 在電腦下載, 用USB隨身碟複製

□ 指令: mount

- 較複雜, 適用於有linux操作經驗的使用者
- 優點: 沒有網路也可複製

```
(COM8) [80x24]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

Disk /dev/mmcblk0: 14.9 GiB, 15931539456 bytes, 31116288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xf292df1d

Device      Boot Start      End Sectors  Size Id Type
/dev/mmcblk0p1        8192    98045   89854 43.9M  c W95 FAT32 (LBA)
/dev/mmcblk0p2     98304 31116287 31017984 14.8G 83 Linux

Disk /dev/sda: 3.6 GiB, 3880452096 bytes, 7579008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc3072e18

Device      Boot Start      End Sectors  Size Id Type
/dev/sdal    *    8064 7579007 7570944  3.6G  c W95 FAT32 (LBA)
pi@raspberrypi:~$
```



C. 在電腦下載, 用USB隨身碟複製

- 掛載: sudo mount /dev/sda1 /mnt
- 卸載: sudo umount /mnt

```
(COM8) [80x24]
連線(C) 編輯(E) 檢視(V) 視窗(W) 選項(O) 說明(H)

Device      Boot Start      End Sectors  Size Id Type
/dev/mmcblk0p1        8192    98045   89854 43.9M  c W95 FAT32 (LBA)
/dev/mmcblk0p2     98304 31116287 31017984 14.8G 83 Linux

Disk /dev/sda: 3.6 GiB, 3880452096 bytes, 7579008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc3072e18

Device      Boot Start      End Sectors  Size Id Type
/dev/sdal  *     8064 7579007 7570944 3.6G  c W95 FAT32 (LBA)
pi@raspberrypi:~$ sudo mount /dev/sdal /mnt/
pi@raspberrypi:~$ ls /mnt/
autorun.inf isolinux  preseed          ubnkern
boot       ldlinux.sys README.diskdefines ubnpath1.txt
casper     md5sum.txt  syslinux.cfg      wubi.exe
dists      menu.c32   System Volume Information
EFI         pics      ubnfile1.txt
install    pool      ubninit
pi@raspberrypi:~$
```



Summary

- **Labs**
 - 1. 根據安裝步驟, 使用TTL控制樹莓派, 完成環境設定
 - 2. 建立VNC遠端桌面, 需開啟 “direct capture mode”
 - 3. 使用Python + GPIO控制LED燈
 - 4. 練習傳輸檔案的方式
- Write down the answer for discussion, upload to e-campus.
Deadline: AM11:59, 3/12(Fri.)
 - Discussion 1: Identify the resistors. (Ω)
- **No quiz this week.**
When you finish all Labs, tell TAs to record your attendance.