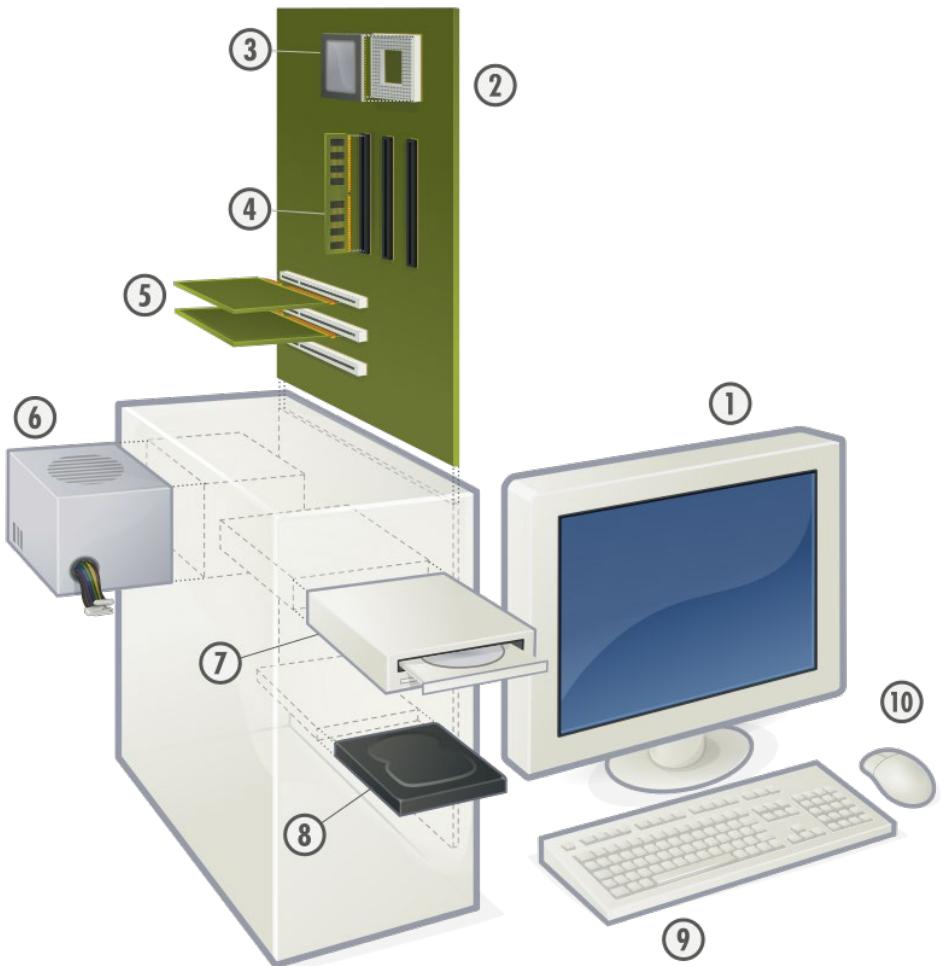


# 單板機應用入門

---

CH1. 單晶片與單板機

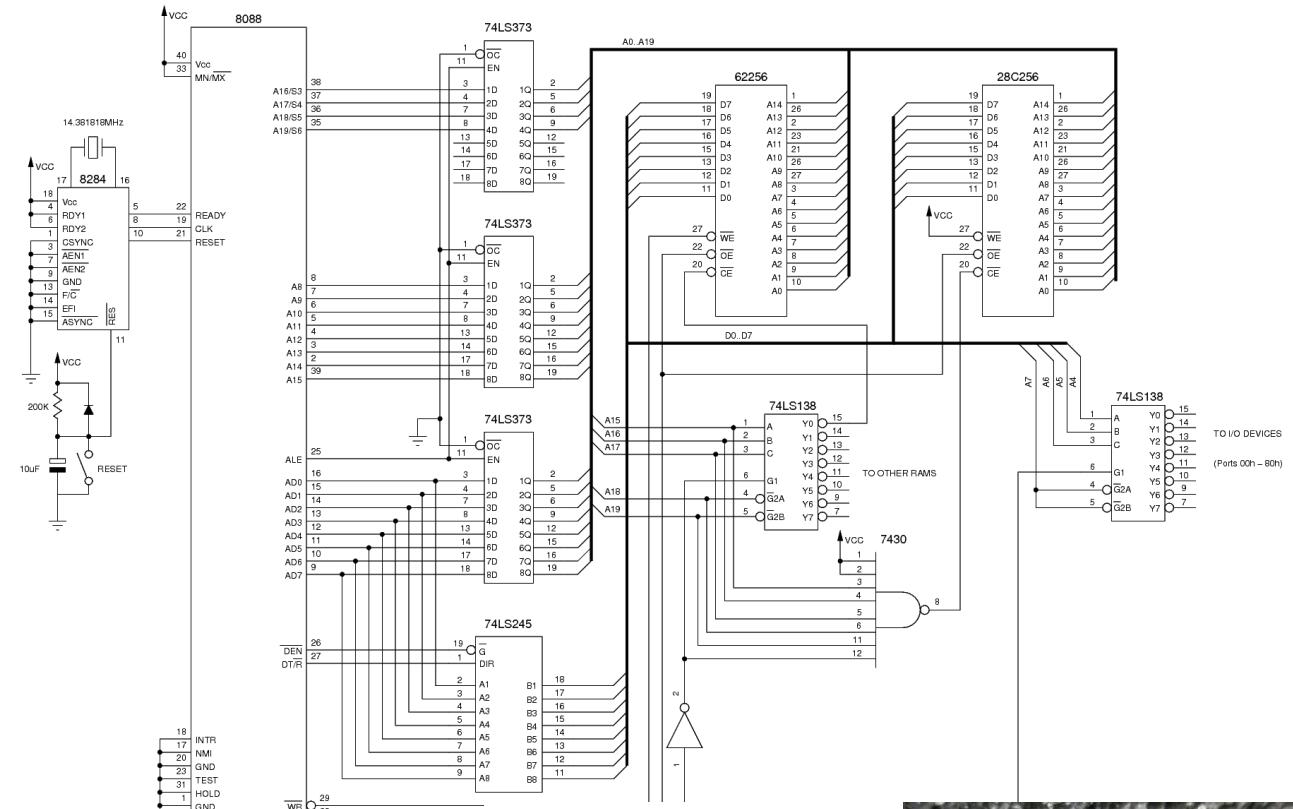
# 個人電腦



- 螢幕
- 主機板
- 中央處理器（微處理器）
- 記憶體
- 介面卡（如音效卡、網路卡）
- 電源供應器
- 軟碟機 / 光碟機
- 硬碟
- 鍵盤
- 滑鼠

# 微電腦的基本組成

- CPU - 8088
- ROM - 27C256(EEPROM)
- RAM - M5M5256
- I/O - 8255



# 單晶片



單晶片微電腦(Single-Chip Microcomputer)，又稱為控制器(Microcontroller)。

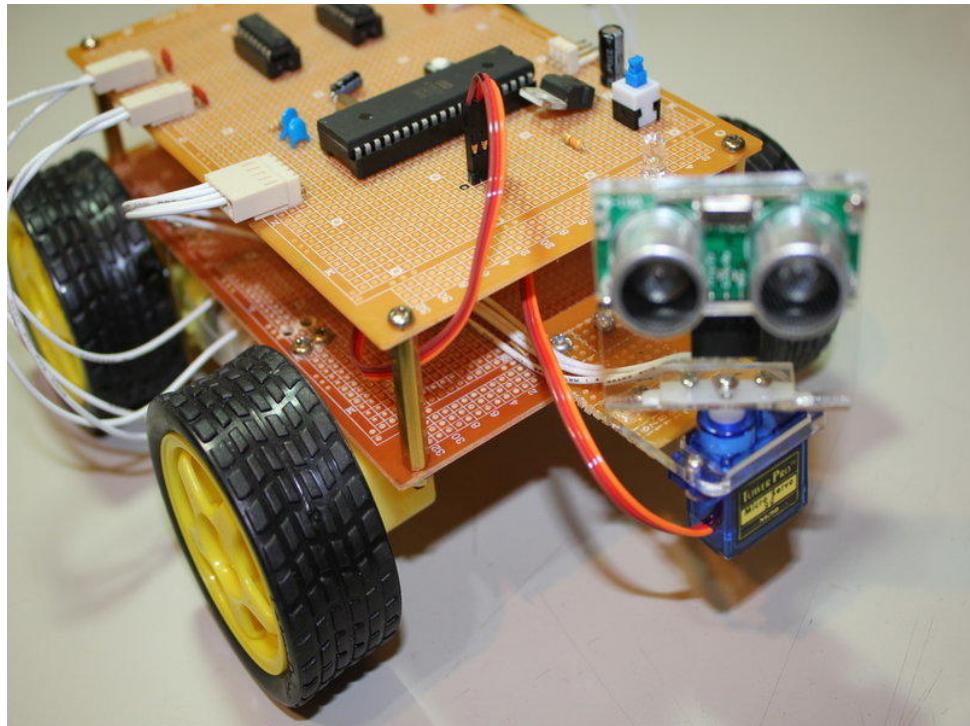
是將中央處理器、記憶體、定時/計數器、各種輸入輸出介面，整合在一塊積體電路上。

# 單晶片應用專題



- 電子存錢筒
- 電子保險箱
- 智慧電控鎖

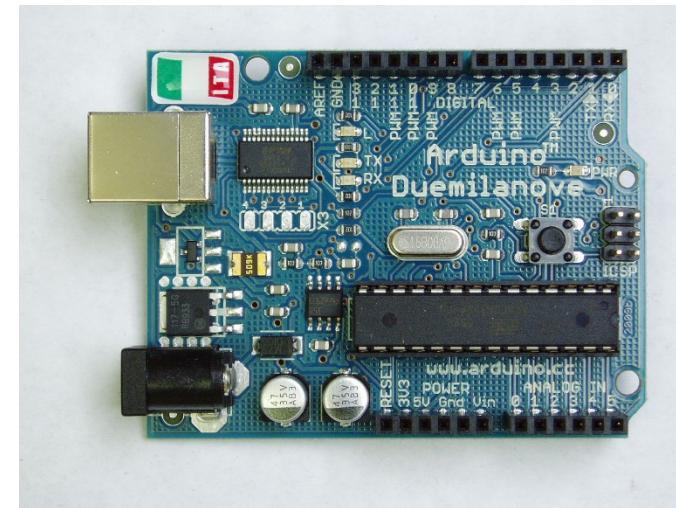
# 單晶片自走車專題



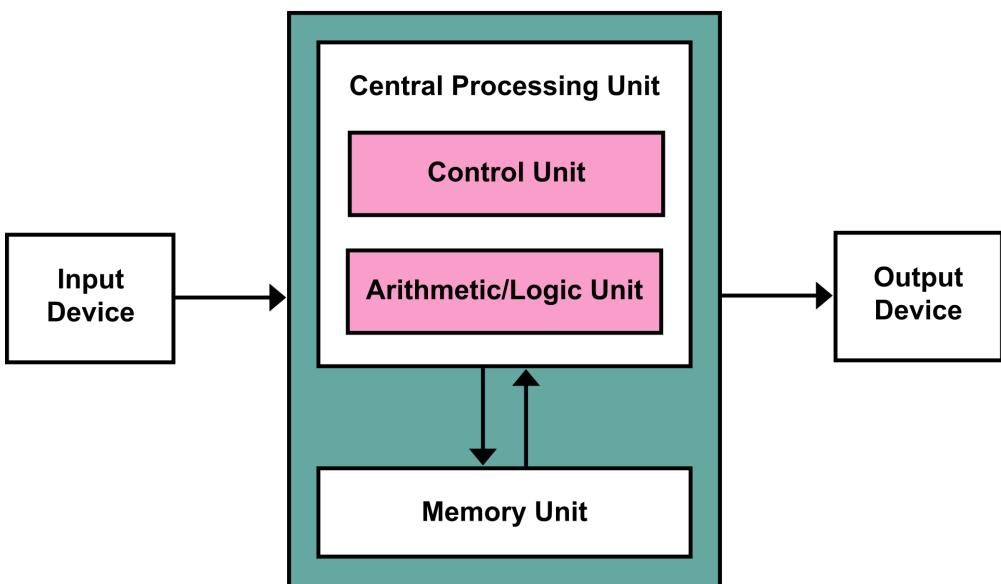
- 超音波避障自走車

# 近代單晶片

- Arduino
  - 開放原始碼的單晶片微控制器。
  - 開機時電壓於重置接腳上，使程式計數器的值為000H，開始接續後面的執行。
- 各式分類(無線/藍芽)

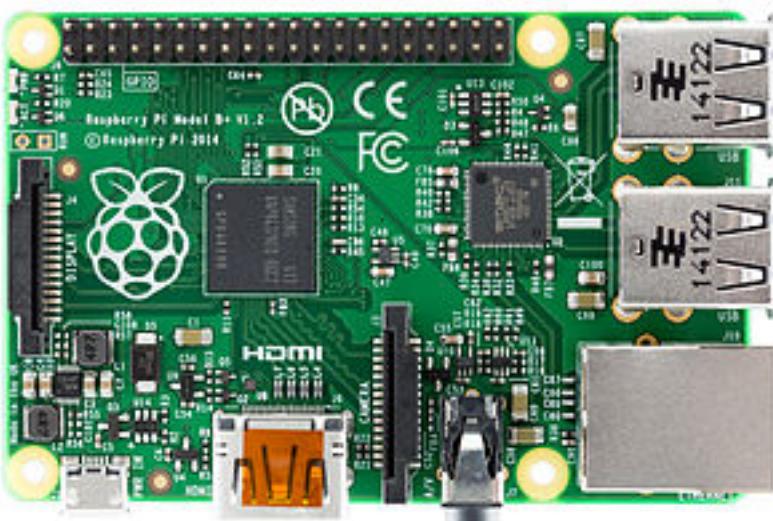


# 單板機



- 單板電腦(Single Board Computer, SBC)，將計算機的各個部分，組裝在一塊電路板上。
- 馮紐曼架構
  - 輸入單元：接收外部資料
  - 輸出單元：送出運算結果
  - 控制單元：負責聯繫各個基本單元
  - 算術邏輯運算單元：進行算術與邏輯運算
  - 記憶單元：儲存輸入單元的程式與資料

# 近代單板機



- 樹莓派(Raspberry Pi)
  - 基於Linux的單板機電腦。
  - 至功能執行前，需等候作業系統完成開機。
- 水果派系列
  - 香蕉派(Banana Pi)
  - 橘子派(Orange Pi)
  - 蓮霧派(Roseapple Pi)

# 樹莓派介紹

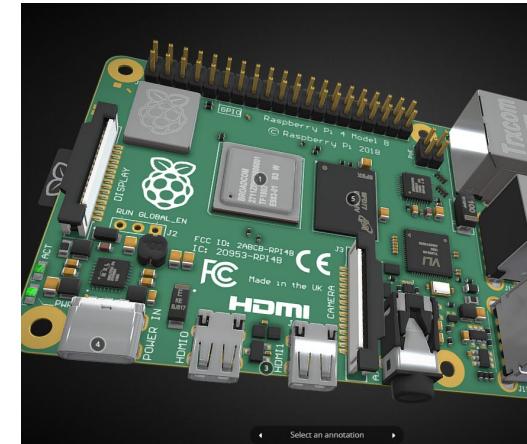
- Raspberry Pi是基於Linux的單晶片電腦。
- 是以低價硬體及自由軟體，促進學校基本電腦教育為目的。
- Raspberry源於對微型電腦以水果為基礎的命名方式。
- Pi代表為Python，是第一個移植到樹莓派上執行的程式。

# 樹莓派規格

|         | A型   | B型             | B型（三代）           | B型（四代）           |
|---------|--|----------------|------------------|------------------|
| 系統晶片    | Broadcom BCM2835   |                | Broadcom BCM2837 | Broadcom BCM2711 |
| CPU     | ARM 11系列   |                | ARM v8系列         |                  |
| GPU     | Broadcom Video Core IV<br>OpenGL ES 2.0<br>1080p 30 h.264 / MPEG-4 AVC高畫質解碼器 |                |                  | H.265<br>H.264   |
| 記憶體     | 256MB  | 256MB<br>512MB | 1024MB           | 1G/2G/4G/8G      |
| USB 2.0 | 1  | 2              | 4                | 2                |
| USB 3.0 |  | 0              |                  | 2                |

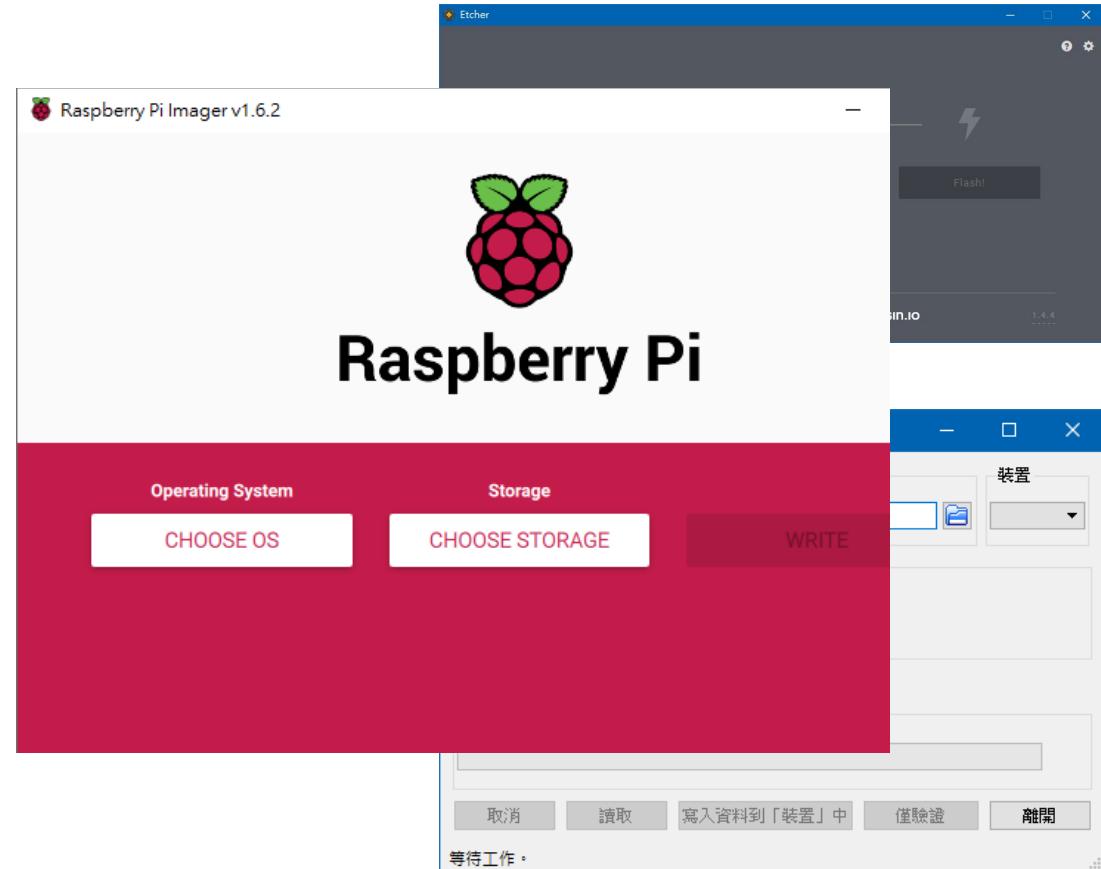
# 線上檢視

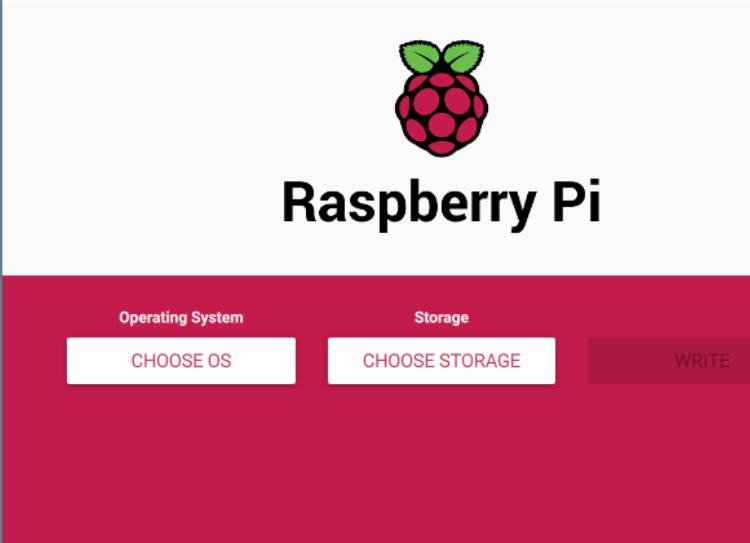
<https://sketchfab.com/3d-models/raspberry-pi-4-model-b-37d36f67edaa4048b90858aa0af347e8>



# 燒錄映像檔

- SD卡上的作業系統資料，可由映像檔來還原。
  - Win32DiskImager (還原及備份)
    - <https://sourceforge.net/projects/win32diskimager/>
  - Etcher (僅還原)
    - <https://etcher.io/>
  - Raspberry Pi Imager (官方)
    - <https://www.raspberrypi.com/software/>





Raspberry Pi Imager v1.6.2

| Operating System |  |
|------------------|--|
|                  | Raspberry Pi OS (other)<br>Other Raspberry Pi OS based images                  |
|                  | Other general purpose OS<br>Other general purpose Operating Systems            |
|                  | Media player - Kodi OS<br>Kodi based Media player operating systems            |
|                  | Emulation and game OS<br>Emulators for running retro-computing platforms       |
|                  | Other specific purpose OS<br>Thin clients, digital signages and 3D printers OS |

# Raspberry Pi Imager

1. Choose OS 選系統並下載
2. Choose Storage 選磁碟(SD)
3. Write 開始寫入

# How to use Raspberry Pi Imager

---

OS  
TO  
SD



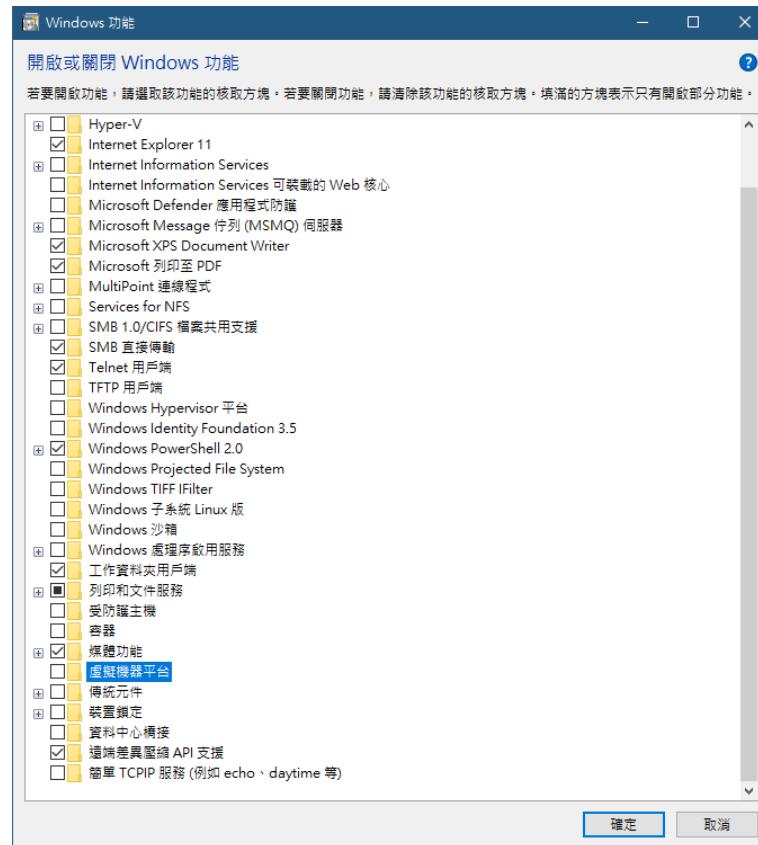
# 單板機應用入門

---

CH2. 單板機作業系統

# 虛擬機器

- Microsoft Windows 10
  - 虛擬機器平台
- VMware
  - 提供雲端運算與硬體虛擬化的軟體及服務。
  - Workstation、Fusion、ESXi
- VirtualBox
- Parallels Desktop



# 作業系統介紹

- Microsoft Windows
  - Windows XP、Windows 8、Windows 10
- Linux
  - Debian、Fedora、Ubuntu、CentOS
- Mac OS
- Android
- iPad OS / iPhone

# Virtual Box 介紹

- VirtualBox原是由德國InnoTek軟體公司出品，目前由甲骨文(Oracle)發展，是甲骨文公司xVM虛擬化平台技術的一部分。
- 提供使用者在32位元、64位元的Windows、Solaris、Linux作業系統上，虛擬其它x86或x64的作業系統。



# Virtual Box 特色

- 支援32位元主機，使用64位元客戶端作業系統。
- 提供虛擬硬碟快照技術。
- 能在主機端與客戶端，共享剪貼簿內容。
- 可在主機端與客戶端間，建立分享資料夾。
- 支援VMware磁碟格式，及Microsoft VHD磁碟格式。
- 3D虛擬化技術支援。
- 支援VT-x與AMD-V硬體虛擬化技術。
- USB支援USB2.0及3.0。



# VirtualBox 取得

- <https://www.virtualbox.org/wiki/Downloads>



The screenshot shows the official VirtualBox download page. At the top left is the Oracle VirtualBox logo. The main title "VirtualBox" is in large blue letters. On the right side, there's a search bar and links for "登入" (Login), "偏好設定" (Preferences), "起始頁" (Home), "索引" (Index), and "歷史紀錄" (History). The left sidebar has links for "About", "Screenshots", "Downloads", "Documentation", "End-user docs", "Technical docs", "Contribute", and "Community". The main content area is titled "Download VirtualBox". It says "Here you will find links to VirtualBox binaries and its source code." Below this is a section titled "VirtualBox binaries" with a note about accepting license terms. It mentions that version 6.1 will remain supported until December 2023. A section titled "virtualBox 7.0.10 platform packages" is shown, with a red circle highlighting the "Windows hosts" link. Below this, it says the binaries are released under the GPL version 3. There's a "changelog" link, a note about checksums, and a "Note" about upgrading guest additions. At the bottom, there's another section titled "VirtualBox 7.0.10 Oracle VM virtualBox Extension Pack" with a red circle highlighting the "All supported platforms" link. A note at the very bottom explains the extension pack's purpose.

search...

登入 | 偏好設定  
起始頁 | 索引 | 歷史紀錄

About  
Screenshots  
Downloads  
Documentation  
End-user docs  
Technical docs  
Contribute  
Community

# VirtualBox

## Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

### VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.1 packages, see [VirtualBox 6.1 builds](#). Version 6.1 will remain supported until December 2023.

#### virtualBox 7.0.10 platform packages

- Windows hosts
- macOS / Intel hosts
- Linux distributions
- Solaris hosts
- Solaris 11 IPS hosts

The binaries are released under the terms of the GPL version 3.

See the [changelog](#) for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*

- [SHA256 checksums](#), [MD5 checksums](#)

**Note:** After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

#### VirtualBox 7.0.10 Oracle VM virtualBox Extension Pack

- All supported platforms

Support VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See [this chapter from the User Manual](#) for an introduction to this Extension Pack. The Extension Pack binaries are released under the [VirtualBox Personal Use and Evaluation License \(PUEL\)](#). Please install the same version extension pack as your installed version of VirtualBox.

檔案(F) 機器(M) 說明(H)



新增(N)

設定值(S)  
捨棄  
啟動(T)

# 操作虛擬機器

- 新增並設定虛擬機器環境
- 安裝作業系統 & 客體驅動程式
- 關閉虛擬機器 & 設定磁碟快照
- 網路介面設定



機器工具(M)



全域工具(G)

## 歡迎使用 VirtualBox!

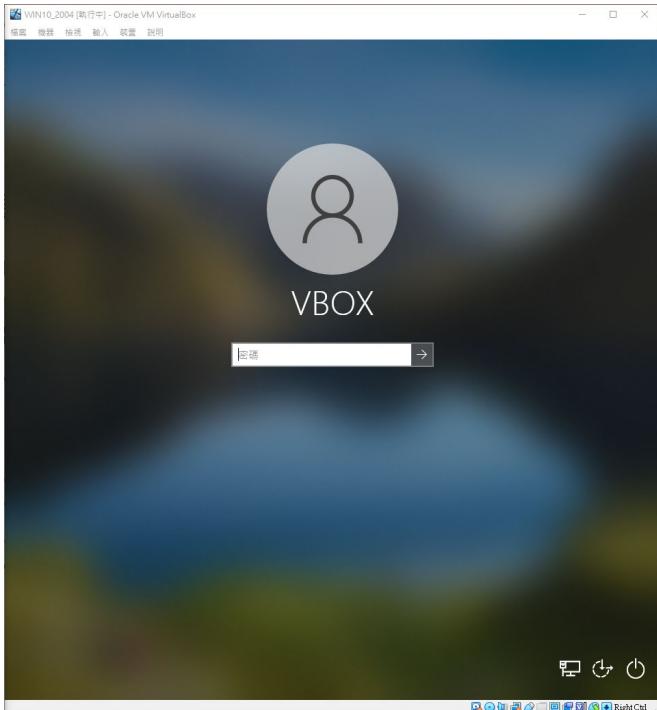
這個視窗的左側是電腦中所有虛擬機器和虛擬機器群組的清單。 清單目前是空的因為尚未建立任何虛擬機器。

為了新建虛擬機器，按下位於視窗頂端的主工具列的 [新增] 按鈕。

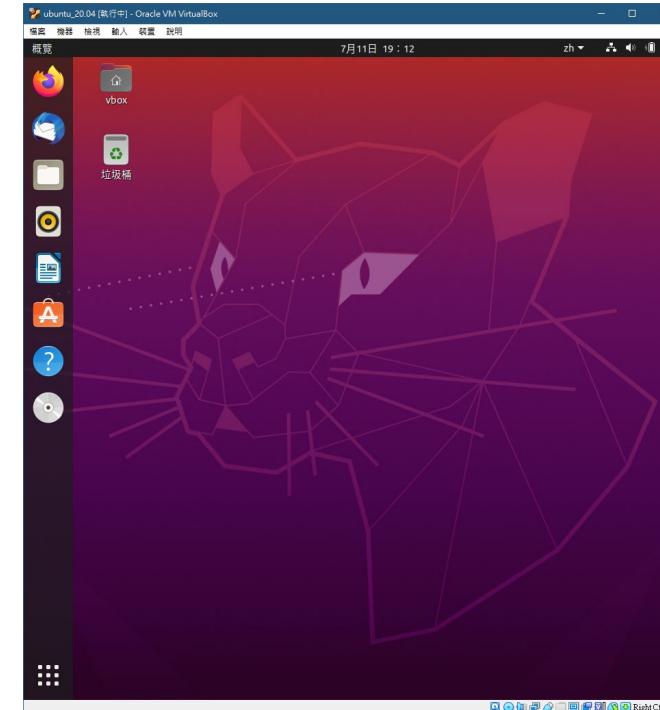
您可以按下 F1 鍵取得即時說明，或訪問 [www.virtualbox.org](http://www.virtualbox.org) 取得更多資訊和最新新聞。

# 作業系統安裝

Windows 11



Ubuntu 20.10

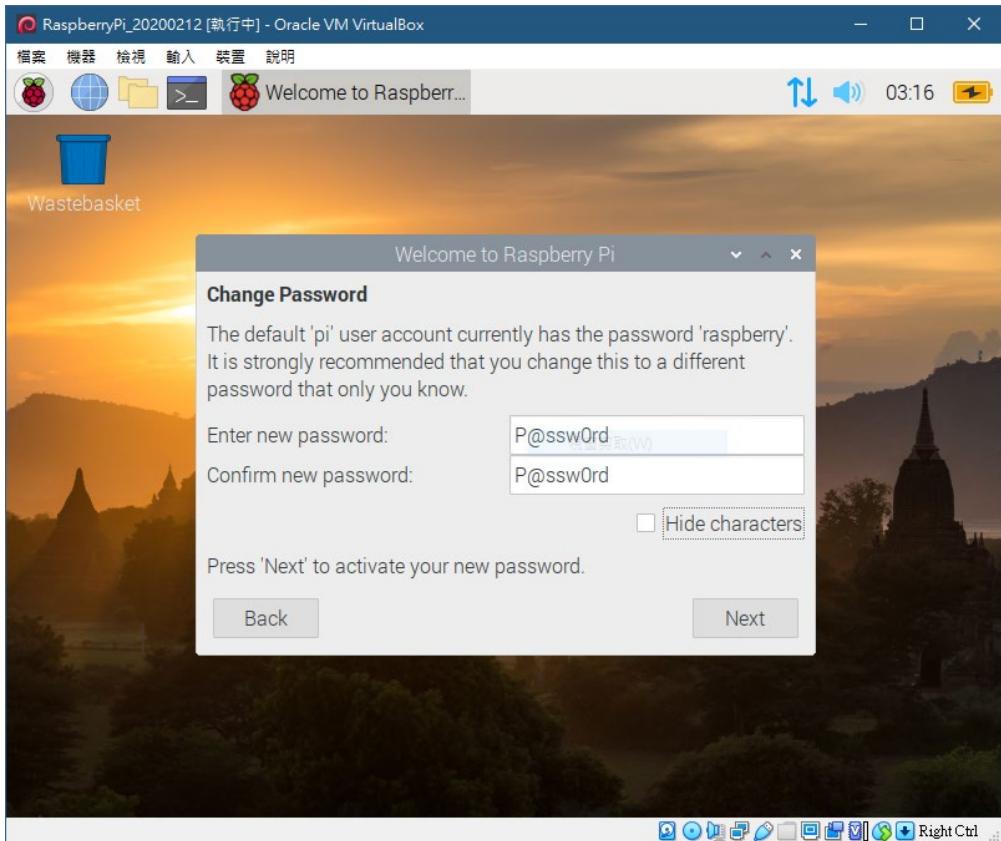


# 關閉虛擬機器

- 儲存機器狀態
  - 設備未關機，僅進行狀態儲存，下次啟動則立即恢復。
- 傳送關機訊號
  - 如同按下機殼面板上的電源紐，作業系統將根據預設設定，進行關機動作。
- 關閉機器
  - 若虛擬機器如同一般電腦班當機(死機)，需要拔除機器電源，則可選此。



# 作業



- 使用虛擬機，完成Raspberry Pi OS 作業系統安裝。

# 終端機常用Linux指令

## 文字編輯器

- 內建文字編輯器
  - vi
- 上色文字編輯器
  - vim
- 友善文字編輯器
  - nano

## 軟體管理工具(APT)

- 更新軟體包清單
  - sudo apt update
- 安裝指定軟體
  - sudo apt install vim
- 升級軟體版本
  - sudo apt upgrade

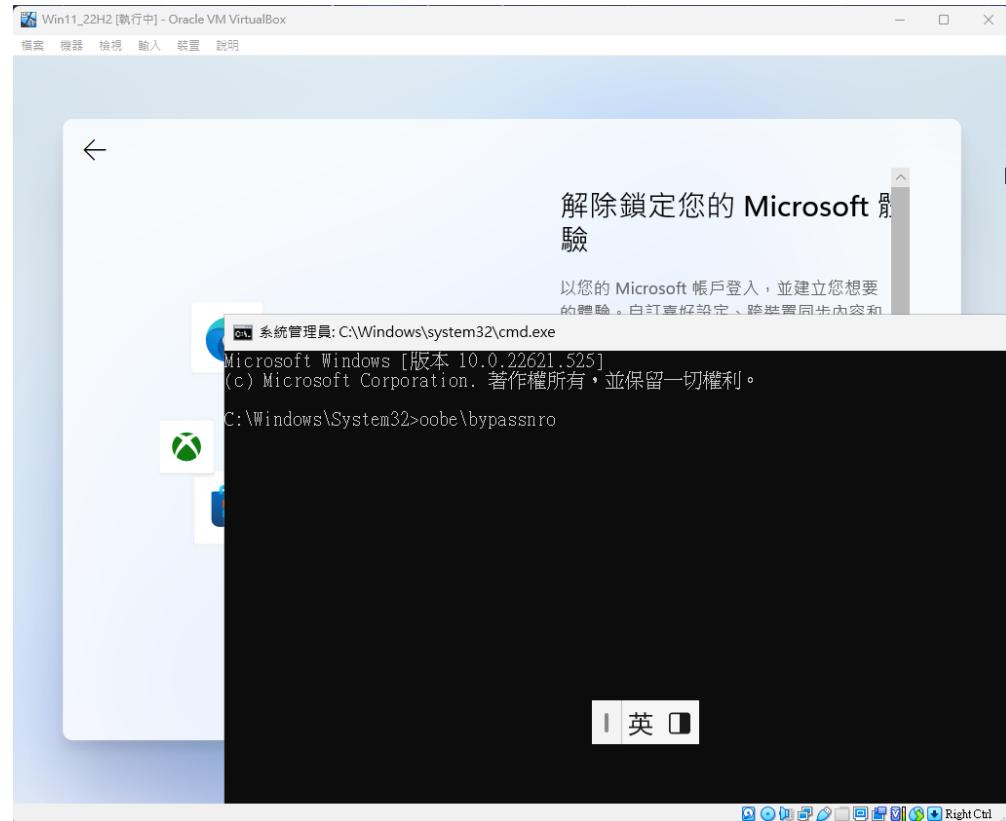
## 安裝遠端桌面工具 (XRDP)

- sudo apt update
- sudo apt install xrdp
- sudo apt -fix-broken install

# Windows11 bypass

- 光碟影像檔位置
  - C:\Program Files\Oracle\VirtualBox\VBoxGuestAdditions.iso
- 檔案名稱
  - windows11-bypass.reg
- 虛擬機器安裝時，可以多新增一個虛擬光碟，並導入此光碟影像檔案備用。
- 實體機器安裝時，可以預先將光碟檔案中的登錄檔文件複製到安裝光碟。

# 啟用無網路模式



- 開啟終端機
  - [SHIFT] + [F10]
- 終端機模式中輸入
  - oobe\bypassnro

# 單板機應用入門

---

CH3. 單板機開發環境

# 網路工具安裝(for Ubuntu)

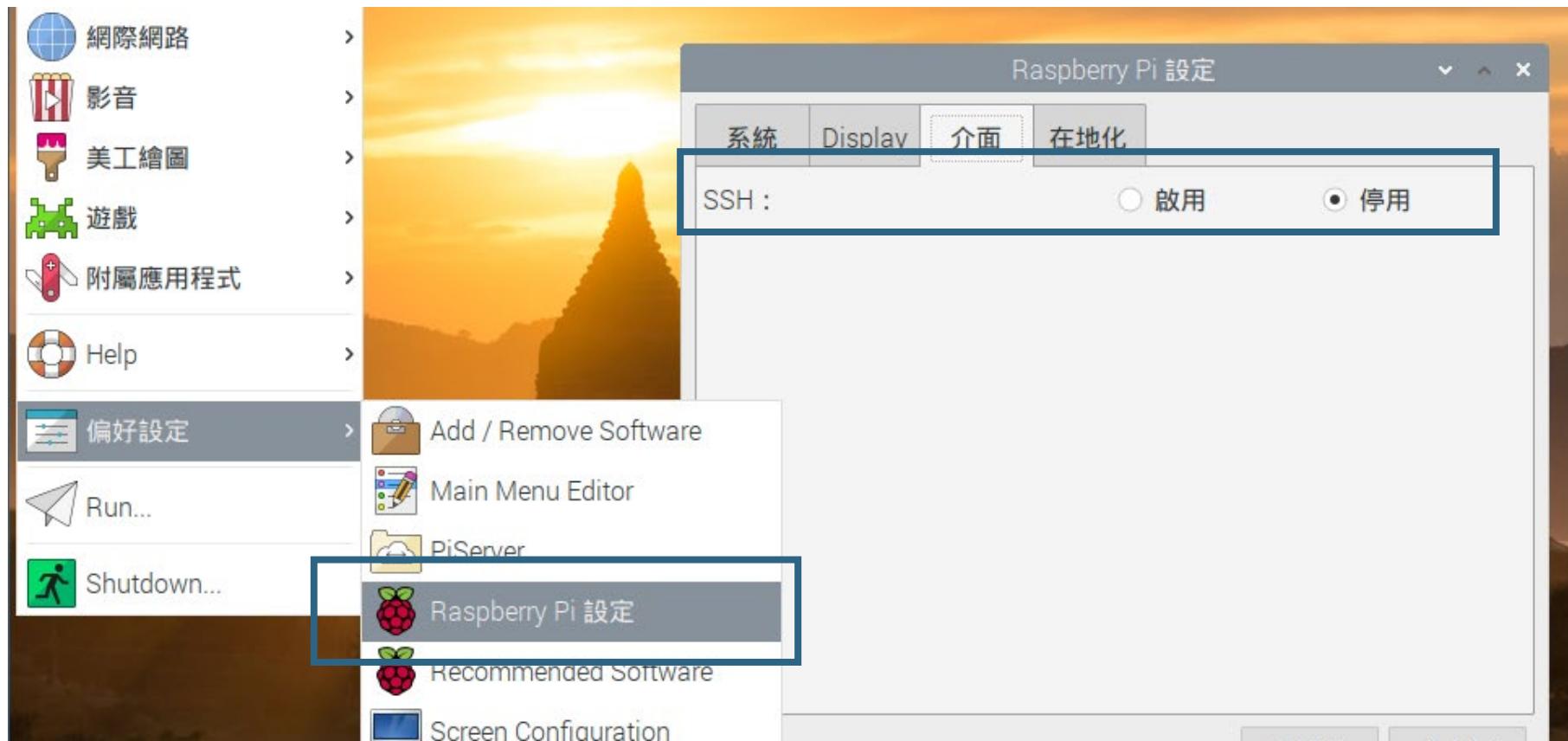
## SSH遠端連線工具

- `sudo apt install openssh-server`
- Raspberry Pi OS 已安裝，僅需手動啟用。

## 基礎網路工具組

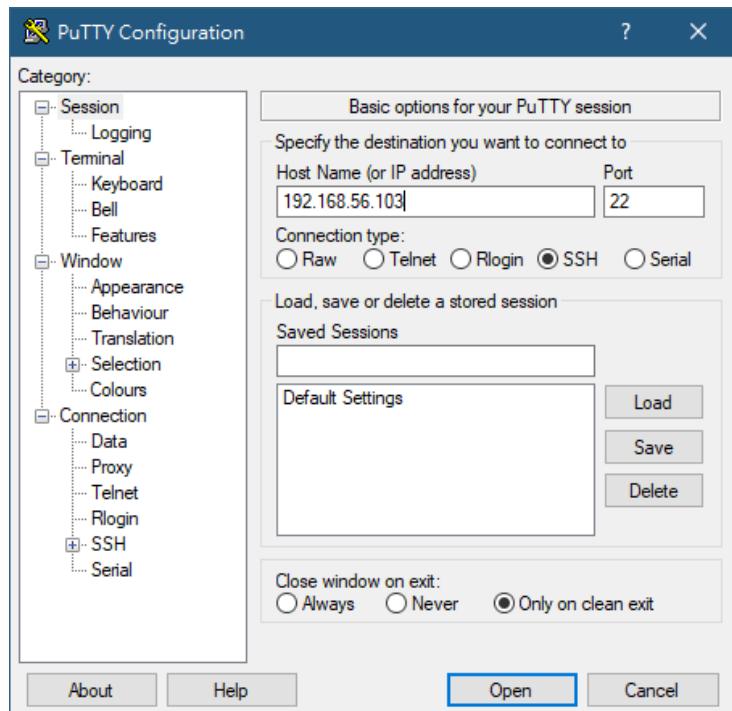
- `sudo apt install net-tools`
- Raspberry Pi OS 已含有，基礎網路工具指令。

# 啟用SSH介面

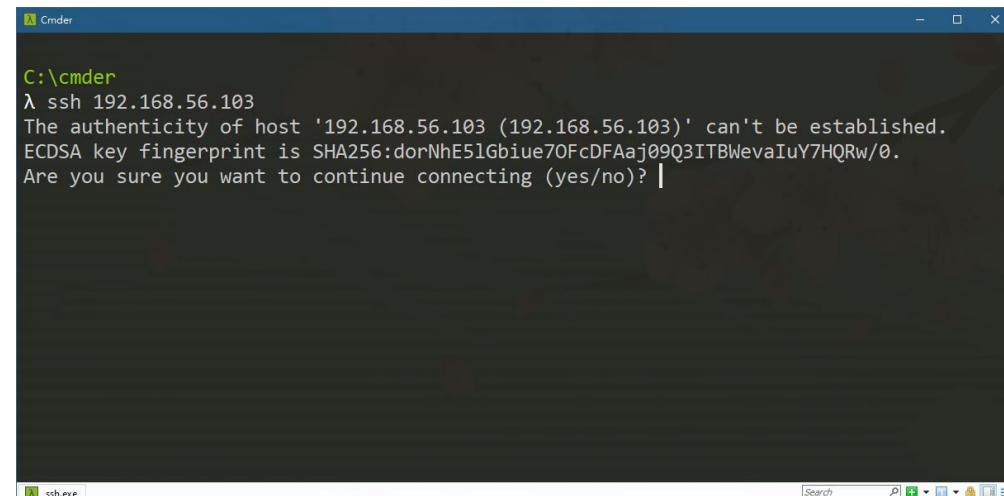


# 使用遠端連入

## PuTTY



## CMD/PowerShell/Cmder



# 開發工具安裝

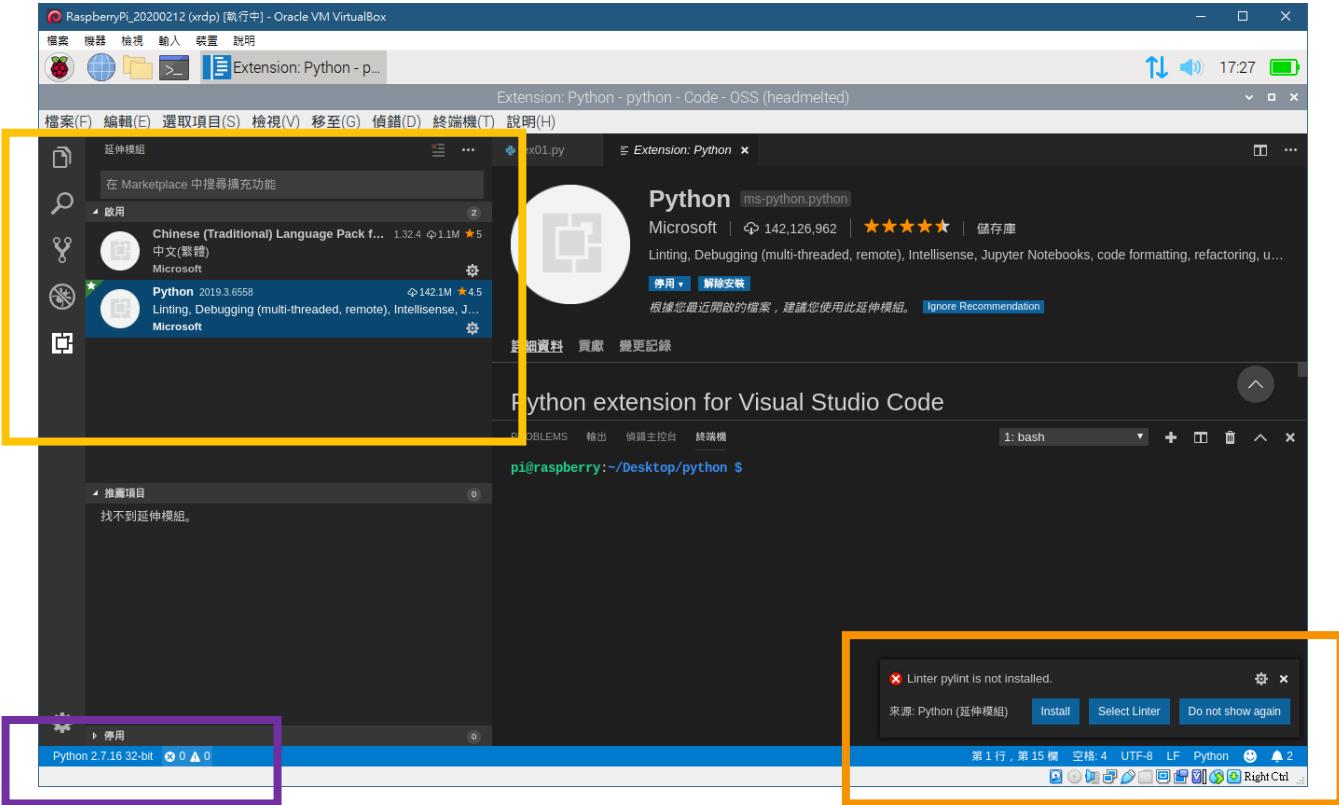
## python

- <https://www.python.org/>
- Raspberry Pi OS已內建Python 2.7
- `sudo apt install python`

## Visual Studio Code

- <https://code.visualstudio.com/>

# VS Code 擴充套件



- 擴充套件可自行安裝。
- 可檢視提示建議安裝。(右下)
- 可切換不同設定環境。(左下)

# Hello World

A screenshot of the VS Code interface on a Raspberry Pi. The title bar says "RaspberryPi\_20200212 (xrdp) [執行中] - Oracle VM VirtualBox". The code editor shows a single line of Python code: "print('hello')". The terminal below shows the output of running the script: "pi@raspberry:~/Desktop/python \$ cd /home/pi/Desktop/python && env PYTHONIOENCODING=UTF-8 PYTHONUNBUFFERED=1 /usr/bin/python /home/pi/.vscode-oss/extensions/ms-python.python-2019.3.6558/pythonFiles/ptvsd\_launcher.py --default --client --host localhost --port 34439 /home/pi/Desktop/python/ex01.py hello pi@raspberry:~/Desktop/python \$".

- Print("HELLO PI")

# Python

- Python是一種廣泛使用的直譯式語言，也是一種通用型程式語言。
  - 直譯式語言：是指不需要經過編譯，可將程式碼一行一行直接執行。
  - 通用型程式語言：是指可在各種領域中使用的語言，不含有特定領域所設計的結構。
- Python直譯器本身，幾乎可以在所有已知的作業系統中執行，在跨平台執行語言中具有優勢。

# Python 縮排

- Python使用縮排，來表示語句塊的開始與結束。
- 根據PEP8的規定，必須使用4個空格來表示，每一級的縮排。
- 可透過IDE設定，將Tab轉換為4個空白。

# Python 常用語句

- if 語句
  - 當條件成立時，執行語句塊區段。
- for 語句
  - 遍歷列表、字串、字典、集合等內容，依次處理內容中的每個元素。
- while 語句
  - 當條件為True時，循環執行語句塊區段。

# Python 輸出系統時間

- 了解如何取得系統時間
- 了解如何印出指定字串
- 了解如何使用循環語句
- 了解如何使用縮排區段

The screenshot shows a terminal window with the following content:

```
ex01.py - python - Code - OSS (headmelte...  
ex01.py  x  
1 import datetime as systime  
2 import time  
3  
4 while True:  
5     print(systime.datetime.now())  
6     time.sleep(1)  
  
PROBLEMS  輸出  備錯主控台  終端機  
  
File "/usr/lib/python2.7/runpy.py", line 251, in run_path...  
  code = _get_code_from_file(path_name)  
File "/usr/lib/python2.7/runpy.py", line 232, in _get_code...  
  code = compile(f.read(), fname, 'exec')  
File "/home/pi/Desktop/python/ex01.py", line 4  
  while:  
      ^  
SyntaxError: invalid syntax  
pi@raspberry:~/Desktop/python $ cd /home/pi/Desktop/python  
n /home/pi/.vscode-oss/extensions/ms-python.python-2019.3.6  
alhost --port 46349 /home/pi/Desktop/python/ex01.py  
2020-07-13 17:57:05.770636  
2020-07-13 17:57:06.781086  
2020-07-13 17:57:07.781452  
2020-07-13 17:57:08.782094  
2020-07-13 17:57:09.783215  
2020-07-13 17:57:10.784700  
終止  
pi@raspberry:~/Desktop/python $
```

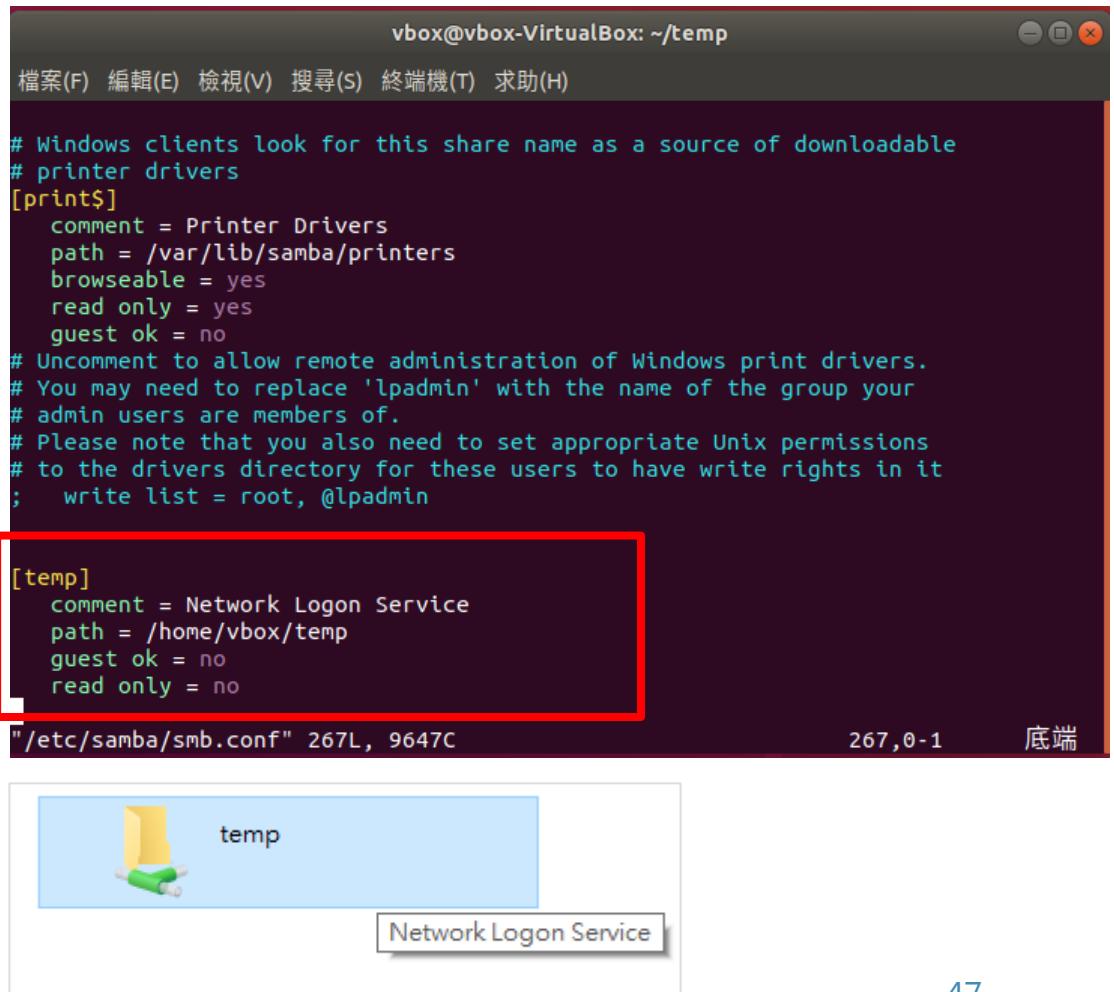
Integrated Terminal (python)

# 完善開發工作環境

- Linux環境的操作，遠端使用「終端機模式」，通常是以純指令的方式進行作業，與遠端使用Windows系統「視窗介面模式」有差異。
- 可透過架設「網路共享資料夾」，使Linux環境中的檔案，在遠端進行編輯。
- Samba是用來讓UNIX系統與Windows系統的SMB/CIFS  
網路協定做連結，可用來存取及分享SMB的資料夾與印表機。

# 共享資料夾設定文件(smb.conf)

- 設定檔路徑
  - /etc/samba/smb.conf
- 設定內容參考
  - [temp]
    - comment = Network Logon Service
    - path = /home/vbox/temp
    - guest ok = no
    - read only = no

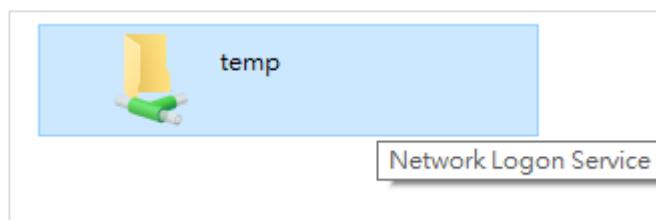


```
vbox@vbox-VirtualBox: ~/temp
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
;   write list = root, @lpadmin

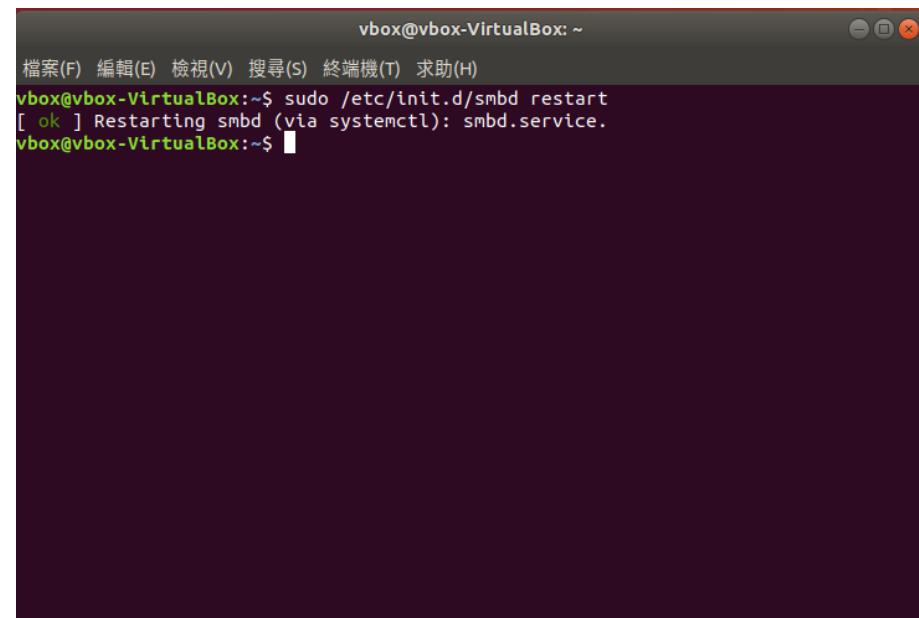
[temp]
comment = Network Logon Service
path = /home/vbox/temp
guest ok = no
read only = no

"/etc/samba/smb.conf" 267L, 9647C          267,0-1      底端
```



# 共享資料夾設定權限

- 設定服務可使用的帳號密碼。
  - `sudo smbpasswd -a {user id}`
- 進行過設定，需將服務重新啟動。
  - `sudo /etc/init.d/smbd restart`



```
vbox@vbox-VirtualBox:~$ sudo /etc/init.d/smbd restart
[ ok ] Restarting smbd (via systemctl): smbd.service.
vbox@vbox-VirtualBox:~$ █
```

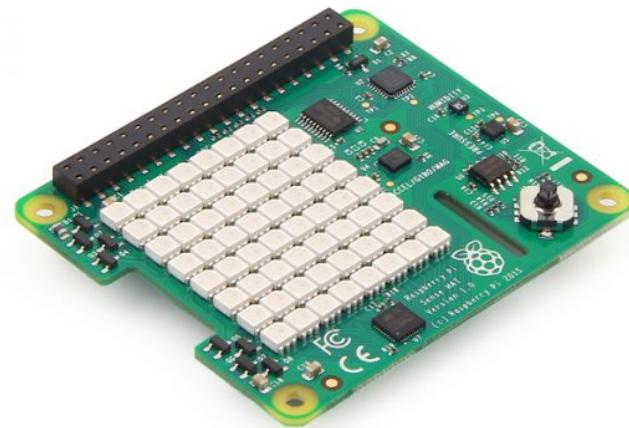
# 單板機應用入門

---

CH4. 單板機感測應用

# Sense HAT 規格

- 8x8 矩陣彩色LED
- 溫度感測器(Temperature)
- 相對濕度感測器(Humidity)
- 大氣壓力感測器(Barometric Pressure)
- 加速計(Accelerometer)
- 電子羅盤磁力儀(Magnetometer)
- 陀螺儀(Gyroscope)
- 可進行上、下、左、右、及按下的搖桿(Joystick)



# Sense HAT與樹莓派連接

- 與樹莓派的連接，共使用了6個PIN腳。
- 感測器資料主要以I<sup>2</sup>C進行通訊。
  - [https://pinout.xyz/pinout/sense\\_hat](https://pinout.xyz/pinout/sense_hat)



|                |    |                           |    |
|----------------|----|---------------------------|----|
| 3v3 Power      | 1  | 5v Power                  | 2  |
| BCM 2 (SDA)    | 3  | 5v Power                  | 4  |
| BCM 3 (SCL)    | 5  | Ground                    | 6  |
| BCM 4 (GPCLK0) | 7  | BCM 14 (TXD)              | 8  |
| Ground         | 9  | BCM 15 (RXD)              | 10 |
| BCM 17         | 11 | BCM 18 (PWM0)             | 12 |
| BCM 27         | 13 | Ground                    | 14 |
| BCM 22         | 15 | BCM 23 (IMU Interrupt)    | 16 |
| 3v3 Power      | 17 | BCM 24 (IMU Interrupt)    | 18 |
| BCM 10 (MOSI)  | 19 | Ground                    | 20 |
| BCM 9 (MISO)   | 21 | BCM 25 (Atmel Prog Reset) | 22 |
| BCM 11 (SCLK)  | 23 | BCM 8 (Atmel Chip Select) | 24 |
| Ground         | 25 | BCM 7 (CE1)               | 26 |
| BCM 0 (ID_SD)  | 27 | BCM 1 (ID_SC)             | 28 |
| BCM 5          | 29 | Ground                    | 30 |
| BCM 6          | 31 | BCM 12 (PWM0)             | 32 |
| BCM 13 (PWM1)  | 33 | Ground                    | 34 |
| BCM 19 (MISO)  | 35 | BCM 16                    | 36 |
| BCM 26         | 37 | BCM 20 (MOSI)             | 38 |
| Ground         | 39 | BCM 21 (SCLK)             | 40 |

# 顯示文字訊息

- Sense HAT提供8x8 RGB LED Matrix。
- 顯示文字訊息，使用show\_message()。

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.show_message("Hello")
```

# 跑馬燈的速度

- 跑馬燈預設速度為10ms，可由參數調整文字的顯示速度。
- 參數以秒數為單位。

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.show_message("Hello", 0.5)
```

# 跑馬燈的文字顏色

- 跑馬燈預設的文字顏色為白色，可藉由參數調整為其他顏色。
- 參數RGB設定0~255。

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.show_message("Hello", 0.5, (255,0,0))
```

# 跑馬燈的背景顏色

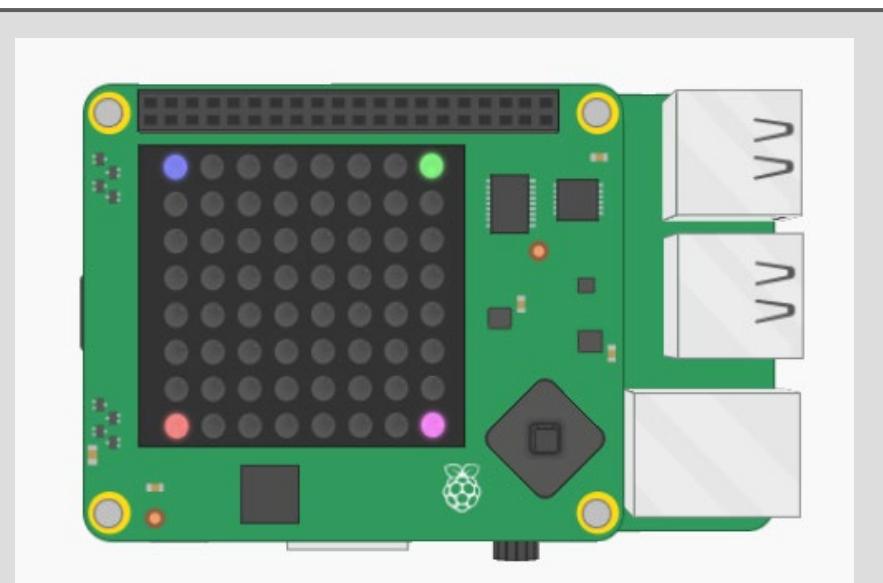
- 跑馬燈預設的背景顏色為黑色，可藉由參數調整為其他顏色。
- 參數RGB設定0~255。

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.show_message("Hello", 0.5, (255,0,0), (255,255,0))
```

# LED點座標

- Sense HAT提供的矩陣顯示，可使用點座標來點亮矩陣上的LED，呼叫set\_pixel()使用。

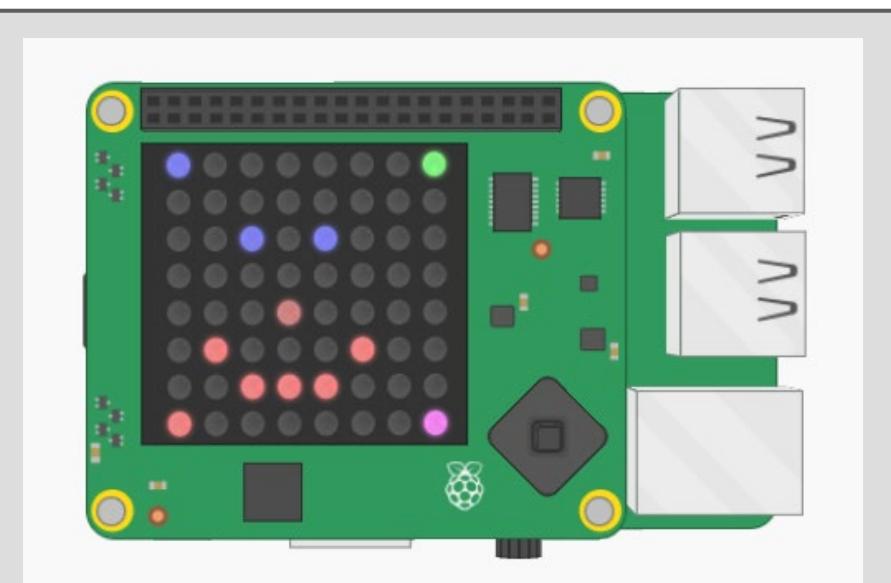
```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.set_pixel(0, 0, (0, 0, 255))  
sense.set_pixel(7, 0, (0, 255, 0))  
sense.set_pixel(0, 7, (255, 0, 0))  
sense.set_pixel(7, 7, (255, 0, 255))
```



# 圖示製作

- Sense HAT提供的矩陣顯示，可使用點座標來點亮矩陣上的LED，呼叫set\_pixel()使用。

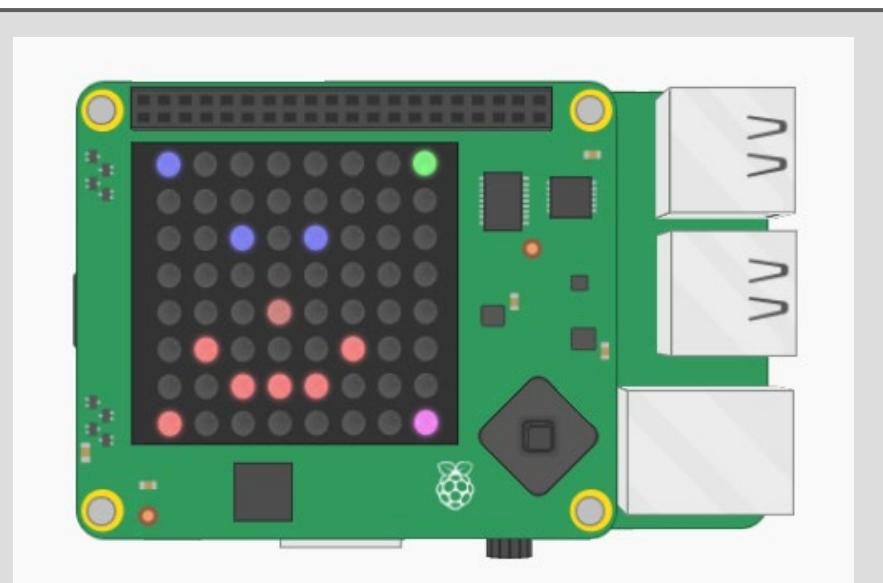
```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.set_pixel(2, 2, (0, 0, 255))  
sense.set_pixel(4, 2, (0, 0, 255))  
sense.set_pixel(3, 4, (100, 0, 0))  
sense.set_pixel(1, 5, (255, 0, 0))  
sense.set_pixel(2, 6, (255, 0, 0))  
sense.set_pixel(3, 6, (255, 0, 0))  
sense.set_pixel(4, 6, (255, 0, 0))  
sense.set_pixel(5, 5, (255, 0, 0))
```



# 清除圖示

- 使用set\_pixel()點亮每一個LED，執行不會移除其他LED的狀態。使用clear()清除LED的狀態。

```
from sense_hat import SenseHat  
sense = SenseHat()  
Sense.clear()  
sense.set_pixel(2, 2, (0, 0, 255))  
sense.set_pixel(4, 2, (0, 0, 255))  
sense.set_pixel(3, 4, (100, 0, 0))  
sense.set_pixel(1, 5, (255, 0, 0))  
sense.set_pixel(2, 6, (255, 0, 0))  
sense.set_pixel(3, 6, (255, 0, 0))  
sense.set_pixel(4, 6, (255, 0, 0))  
sense.set_pixel(5, 5, (255, 0, 0))
```

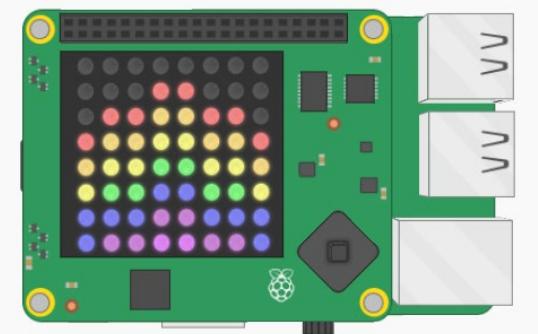


# 繪製圖示

- 使用Sense HAT矩陣顯示器，以點座標的方式顯示點陣圖示，呼叫set\_pixels()使用。

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.clear()  
  
r = (255, 0, 0)  
o = (255, 127, 0)  
y = (255, 255, 0)  
g = (0, 255, 0)  
b = (0, 0, 255)  
i = (75, 0, 130)  
v = (159, 0, 255)  
e = (0, 0, 0)
```

```
image = [  
    e,e,e,e,e,e,e,e,  
    e,e,e,r,r,e,e,e,  
    e,r,r,o,o,r,r,e,  
    r,o,o,y,y,o,o,r,  
    o,y,y,g,g,y,y,o,  
    y,g,g,b,b,g,g,y,  
    b,b,b,i,i,b,b,b,  
    b,i,i,v,v,i,i,b]  
  
sense.set_pixels(image)
```



# Temperature

- 取得溫度感測器(Temperature)的數值。

```
from sense_hat import SenseHat
sense = SenseHat()
while True:
    t = sense.get_temperature()
    t = round(t, 1)

    msg = "Temperature = {}".format(t)
    print msg
```

# Barometric Pressure

- 取得大氣壓力感測器(Barometric Pressure)的數值。

```
from sense_hat import SenseHat
sense = SenseHat()
while True:
    h = sense.get_pressure()
    h = round(h, 1)

    msg = "Pressure = {}".format(h)
    print msg
```

# Humidity

- 取得相對濕度感測器(Humidity)的數值。

```
from sense_hat import SenseHat
sense = SenseHat()
while True:
    h = sense.get_humidity()
    h = round(h, 1)

    msg = "Humidity = {0}".format(h)
    print msg
```