

樹莓派 EX2 基礎輸入輸出應用

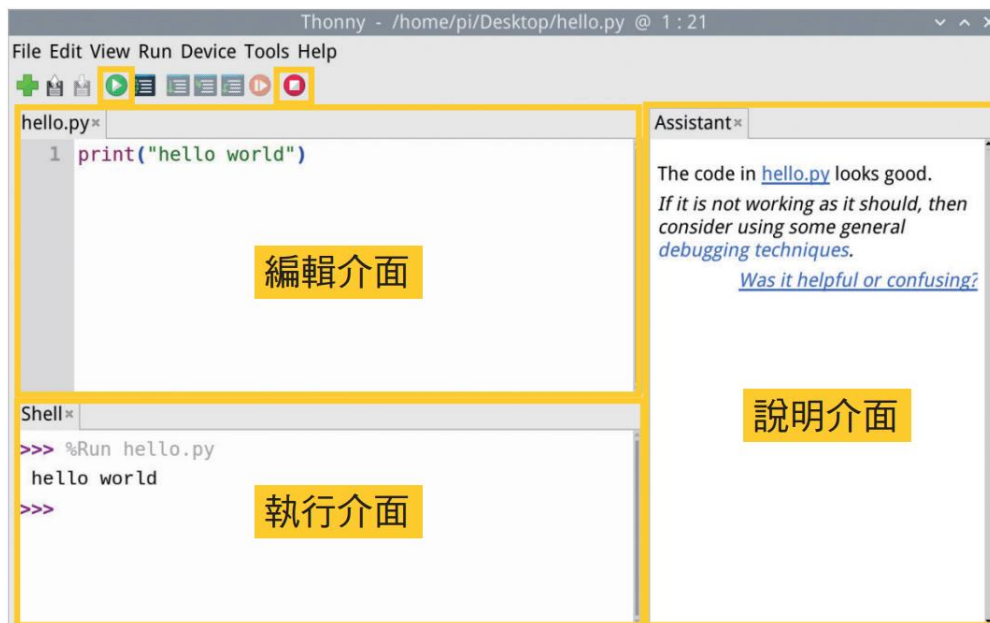


一、在樹莓派上寫控制程式的 3 種方法

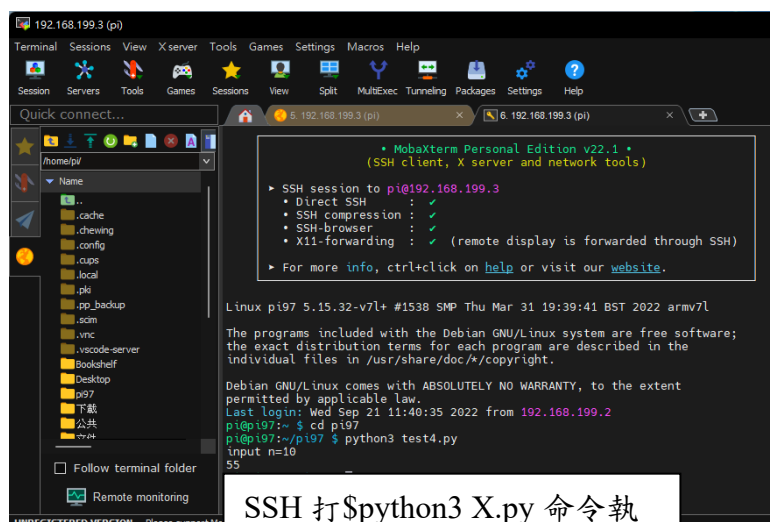
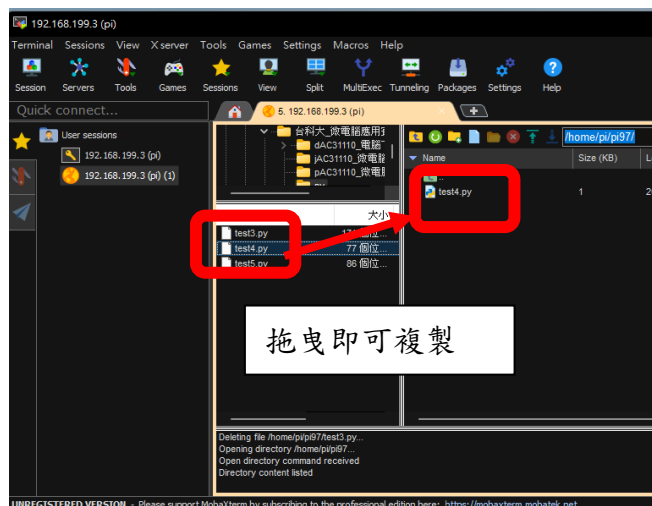
1.用 VNC Viewer，遠端登入樹莓派

(類遠端桌面模式，簡單、穩定性及效率差)

Menu(樹莓派圖) → 軟體開發 → 開啟 Thonny Python IDE



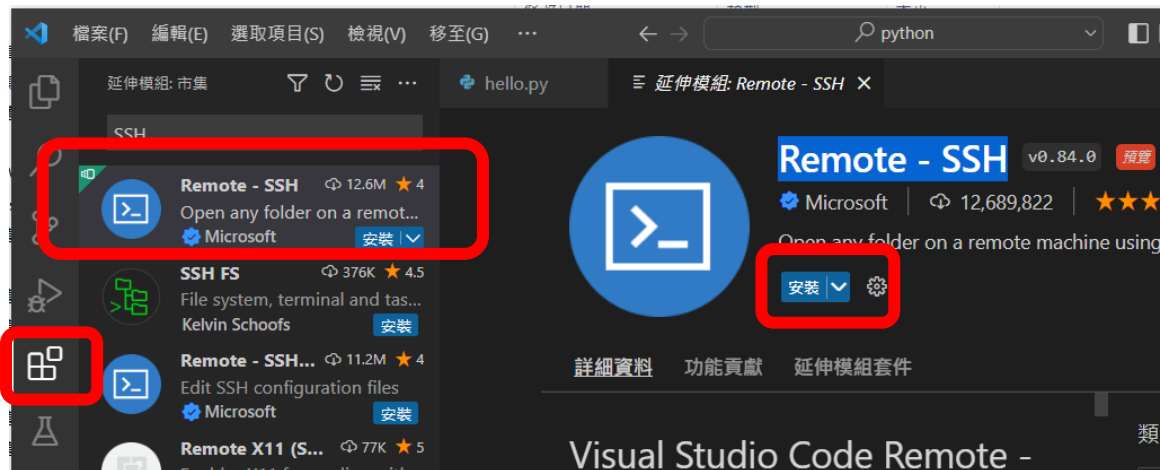
2.用 VS Code 打程式，用 SFTP 傳至樹莓派，SSH 打\$python3 X.py 命令執行 (穩定性及效率佳，但要用 3 種介面，較麻煩)



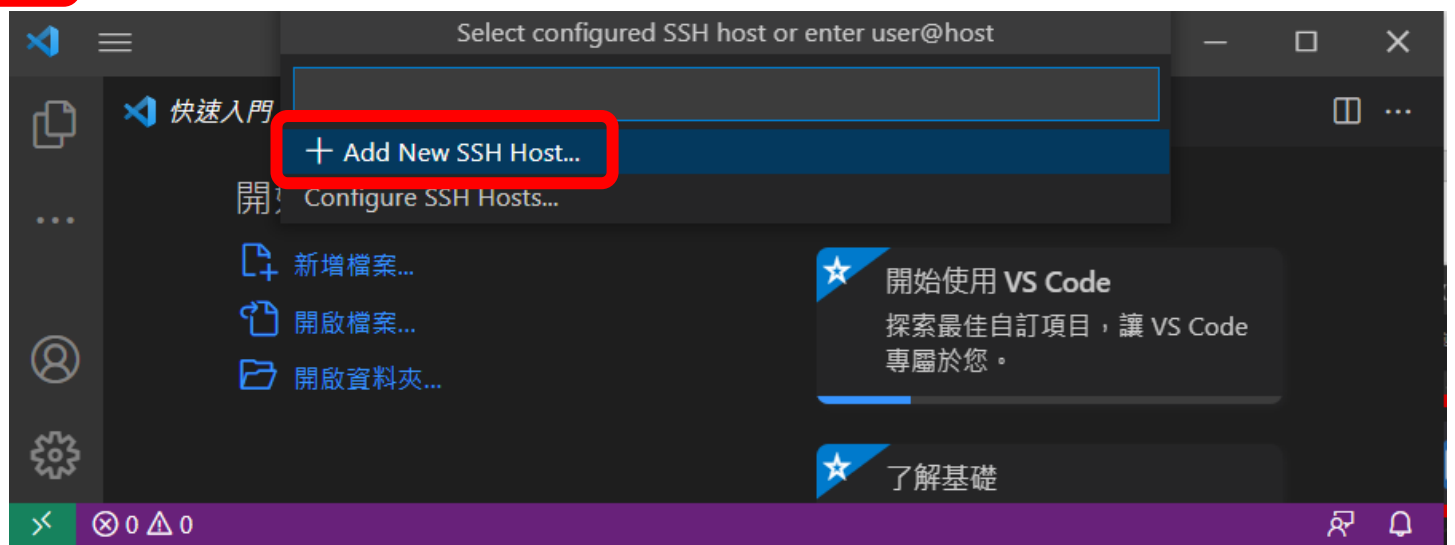
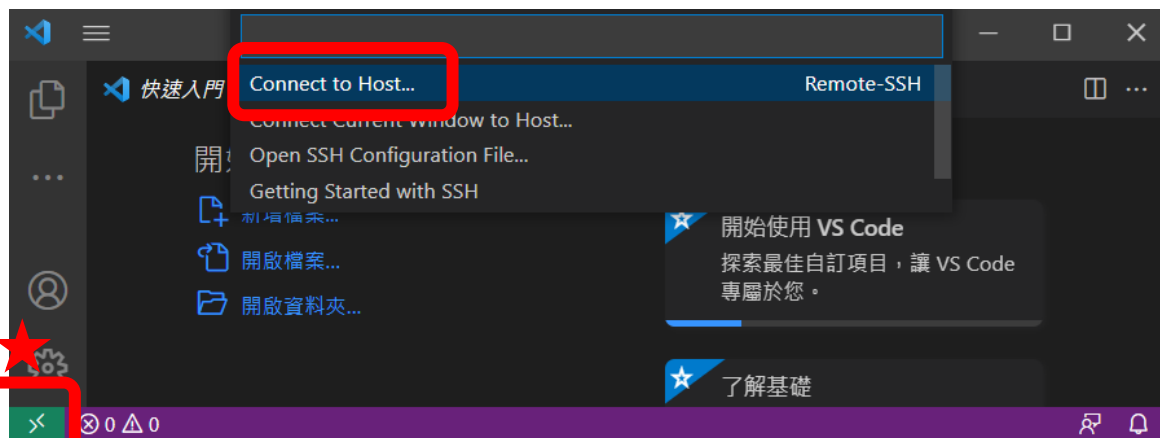
3 用安裝 VS Code 之 Remote-SSH 模組，打完程式直接用 SSH 傳至樹莓派執行

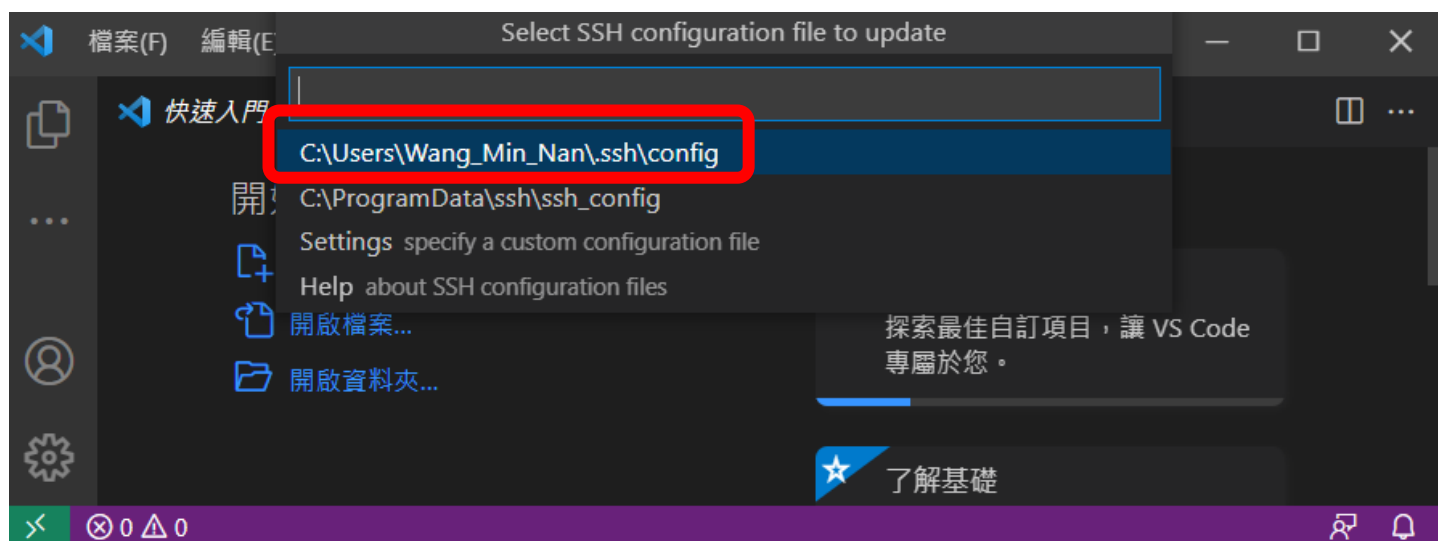
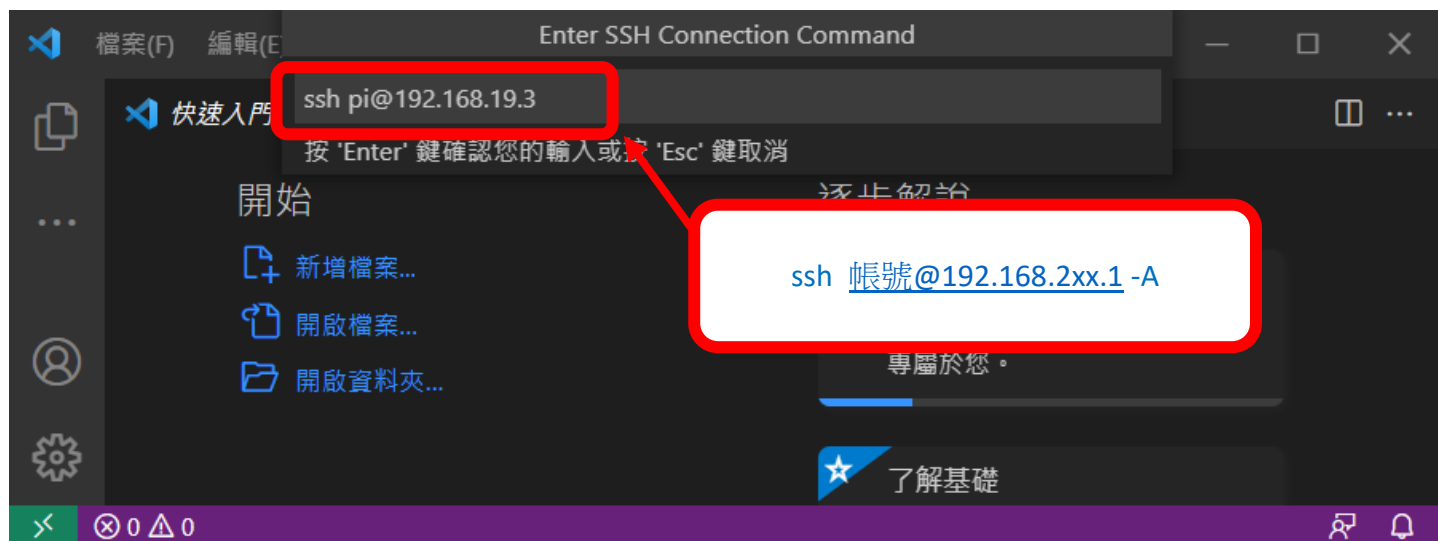
(穩定性及效率中，只要用 1 種介面，但初始設定很麻煩)

參考 <https://www.wongwonggoods.com/draft notes/vscode-ssh/>



連線

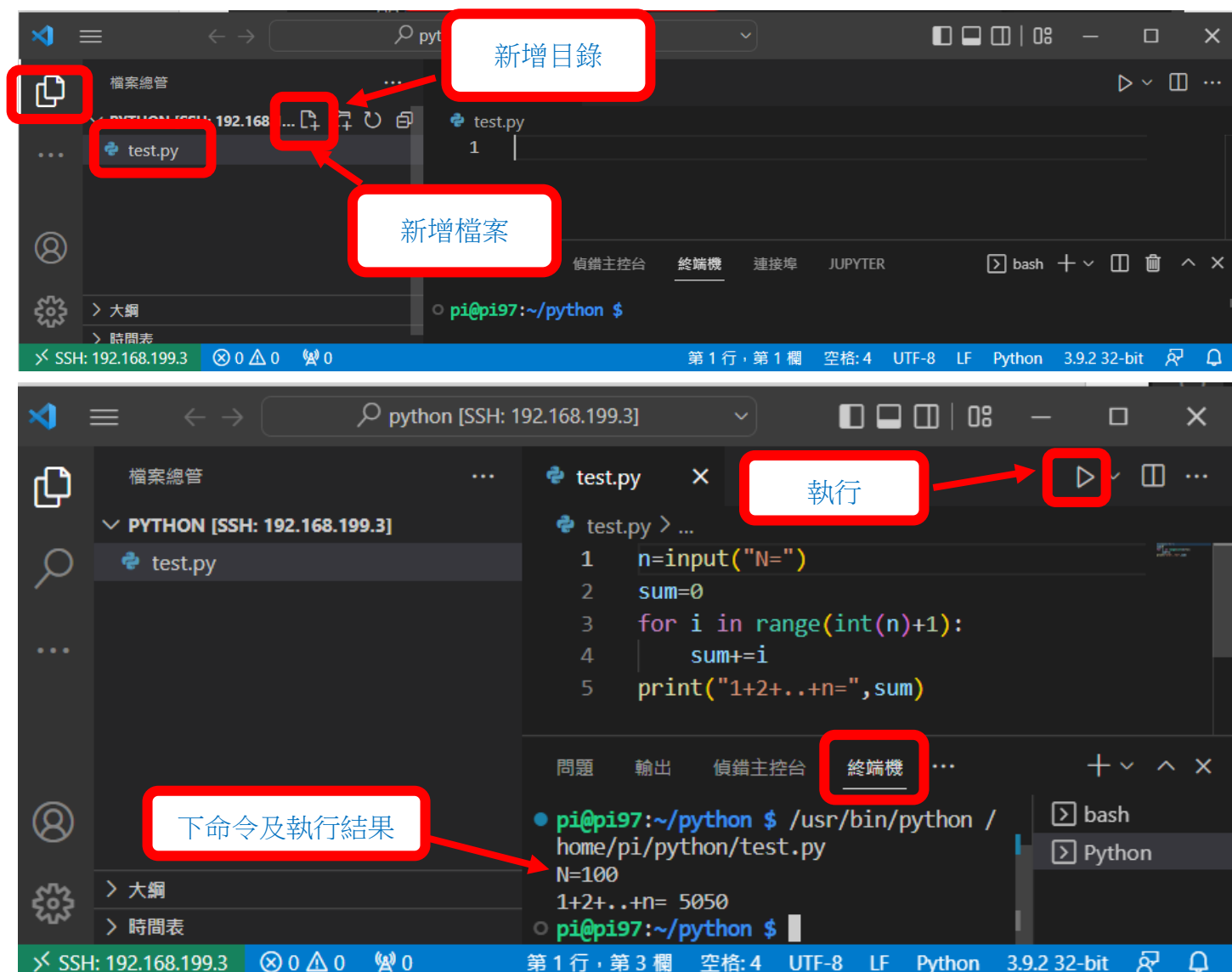




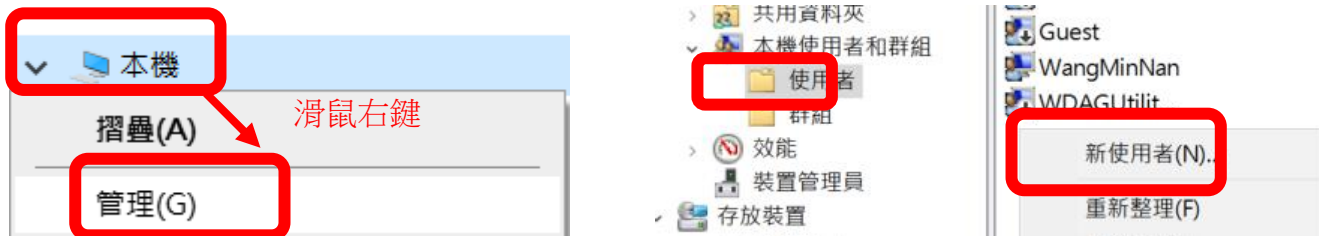
開啟檔案(首次需先 VNC 到樹莓派中建立 Python 資料夾)



新增檔案的檔名時，檔名後面須加上.py，才能轉成 python 檔



*若一直無法連線成功，有可能是其他使用者已寫入加密金鑰在電腦中，建議在 windows 中再建立一個自己的本機帳號，方法如下：



新使用者

使用者名稱(U): TESTpi

全名(F):

描述(D):

密碼(P):

確認密碼(C):

☐ 使用者必須在下次登入時變更密碼(M)

☐ 使用者不能變更密碼(S)

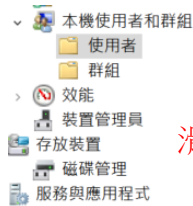
☐ 密碼永久有效(W)

☐ 帳戶已停用(B)

說明(H)

建立(E)

關閉(O)



滑鼠右鍵



TESTpi

一般 成員群組 設定檔

成員群組(M):

Users

新增(D)...

移除(R)

對使用者群組成員資格的變更需等到使用者下次登入時才會生效。

選取群組

選取這個物件類型(S):

群組

物件類型(Q)...

從這個位置(F):

WANGMINNAN

位置(L)...

輸入物件名稱來選取 (範例)(E):

檢查名稱(C)

進階(A)...

確定

取消

選取群組

選取這個物件類型(S):

群組

物件類型(Q)...

從這個位置(F):

WANGMINNAN

位置(L)...

公用查詢

名稱(A): 開頭含有

描述(D): 開頭含有

☐ 已停用的帳戶(B)

☐ 密碼不會到期(X)

上次登入至今的天數(D):

立即尋找(N)

停止(I)

搜尋結果(U):

名稱

在資料夾

Administrators

WANGMINNAN

WANGMINNAN

確定

取消

選取群組

選取這個物件類型(S):

群組

物件類型(Q)...

從這個位置(F):

WANGMINNAN

位置(L)...

輸入物件名稱來選取 (範例)(E):

WANGMINNAN\Administrators

檢查名稱(C)

進階(A)...

確定

取消

TESTpi - 內容

一般 成員群組 設定檔

成員群組(M):

Administrators

Users

新增(D)...

移除(R)

對使用者群組成員資格的變更需等到使用者下次登入時才會生效。

確定

取消

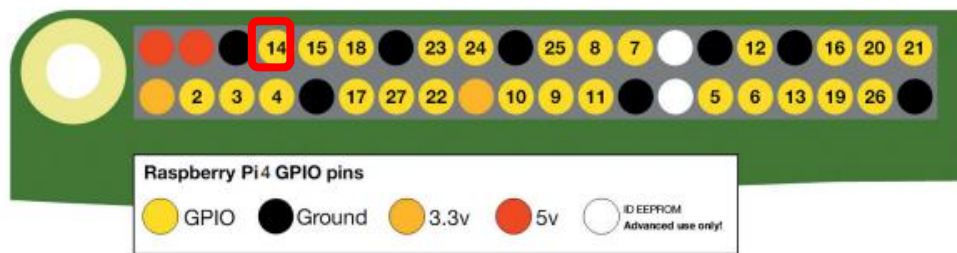
套用(A)

說明

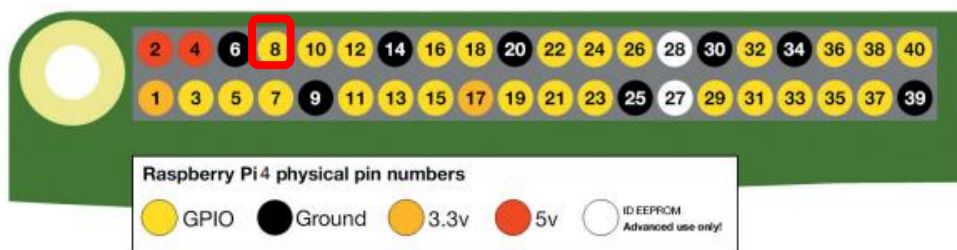
登出再登入新使用者，重新設定

二、基本輸入/輸出(I/O)練習

參考課本 ch2_P22，接腳的名稱有 2 種，下圖中 GPIO14(較常用)又名 Pin 8。



• 圖 2-4 GPIO 腳位編號由 2 ~ 27，其餘為電源或接地接腳



• 圖 2-5 Pin 腳位編號由 1 ~ 40

〔註〕在 Python 程式中，使用 GPIO 腳位定義為 (GPIO.BCM)
使用 Pin 腳位定義為 (GPIO.BOARD)

實驗擴充板編號

D 是 GPIO 接腳名

+ 是 3.3V(VCC)

- 是 0V(GND)



1. 參考 CH6_P124-P127, 完成 LED 閃爍控制練習, 貼上 P127 練習程式

```
1  import RPi.GPIO as GPIO
2  import time
3
4  # 使用BCM模式, 即GPIO編號模式
5  GPIO.setmode(GPIO.BCM)
6
7  # 設置GPIO引腳編號, 例如GPIO 17
8  LED_PIN = 13
9  GPIO.setup(LED_PIN, GPIO.OUT)
10
11 try:
12     while True:
13         # 開啟LED
14         GPIO.output(LED_PIN, GPIO.HIGH)
15         print("LED is ON")
16         time.sleep(1)
17
18         # 關閉LED
19         GPIO.output(LED_PIN, GPIO.LOW)
20         print("LED is OFF")
21         time.sleep(1)
22
23 except KeyboardInterrupt:
24     # 捕獲Ctrl+C終止程序
25     print("Program terminated")
26
27 finally:
28     # 清理GPIO設置
29     GPIO.cleanup()
```

影片連結:

<https://youtube.com/shorts/Y0Ol0l1T4VA?feature=share>

2. 參考 CH6_P129-P133, 完成按鈕開關控制練習, 貼上 P133 練習程式

```
1  import RPi.GPIO as GPIO
2  import time
3
4  # 禁用GPIO的警告信息
5  GPIO.setwarnings(False)
6
7  # 设置GPIO为BCM编号模式
8  GPIO.setmode(GPIO.BCM)
9
10 # 定义LED和按钮引脚
11 LED_PIN = 13
12 btn1_pin = 14
13 btn2_pin = 15
14
15 # 设置LED引脚为输出模式, 按钮引脚为输入模式
16 GPIO.setup(LED_PIN, GPIO.OUT)
17 GPIO.setup(btn1_pin, GPIO.IN)
18 GPIO.setup(btn2_pin, GPIO.IN)
19
20 try:
21     while True:
22         v1 = 1 - GPIO.input(btn1_pin) # 检测按钮1状态
23         v2 = 1 - GPIO.input(btn2_pin) # 检测按钮2状态
24
25         if v1 == 1:
26             print("btn 1 pressed")
27         if v2 == 1:
28             print("btn 2 pressed")
29
30         time.sleep(0.1)
31
32 except KeyboardInterrupt:
33     # 捕获Ctrl+C终止程序
34     print("Program terminated")
35
36 finally:
37     # 清理GPIO设置
38     GPIO.cleanup()
```

影片連結:

<https://youtube.com/shorts/1dKe9Pyd6yY?feature=share>

3. 參考 CH6_P137-P143, 完成 PWM 控制練習, 貼上 P143 練習程式

```
1  import RPi.GPIO as GPIO
2  import time
3
4  # 使用BCM模式, 即GPIO编号模式
5  GPIO.setmode(GPIO.BCM)
6
7  # 设置GPIO引脚编号, 例如GPIO 17
8  LED_PIN = 12
9  GPIO.setup(LED_PIN, GPIO.OUT)
10 PWM = GPIO.PWM(LED_PIN, 50)
11 PWM.start(50)
12
13 try:
14     while True:
15         print("Do")
16         PWM.ChangeFrequency(523)
17         time.sleep(1)
18
19         print("Re")
20         PWM.ChangeFrequency(587)
21         time.sleep(1)
22
23         print("Mi")
24         PWM.ChangeFrequency(659)
25         time.sleep(1)
26
27 except KeyboardInterrupt:
28     # 捕获Ctrl+C终止程序
29     print("Program terminated")
30
31 finally:
32     # 清理GPIO设置
33     GPIO.cleanup()
```

影片連結:

<https://youtube.com/shorts/BIZtEalsUNQ?feature=share>

4. 參考 CH6_P144-P149，完成邏輯控制練習，貼上 P149 練習程式

```
1  import RPi.GPIO as GPIO
2  import time
3
4  # 禁用GPIO的警告信息
5  GPIO.setwarnings(False)
6
7  # 设置GPIO为BCM编号模式
8  GPIO.setmode(GPIO.BCM)
9
10 # 定义LED和按钮引脚
11 LED_PIN = 13
12 btn1_pin = 14
13 btn2_pin = 15
14
15 # 设置LED引脚为输出模式，按钮引脚为输入模式
16 GPIO.setup(LED_PIN, GPIO.OUT)
17 GPIO.setup(btn1_pin, GPIO.IN)
18 GPIO.setup(btn2_pin, GPIO.IN)
19
20 try:
21     while True:
22         v1 = 1 - GPIO.input(btn1_pin) # 检测按钮1状态
23         v2 = 1 - GPIO.input(btn2_pin) # 检测按钮2状态
24
25         if v1 ^ v2:
26             GPIO.output(LED_PIN, GPIO.HIGH)
27         else:
28             GPIO.output(LED_PIN, GPIO.LOW)
29
30         time.sleep(0.1)
31
32 except KeyboardInterrupt:
33     # 捕获Ctrl+C终止程序
34     print("Program terminated")
35
36 finally:
37     # 清理GPIO设置
38     GPIO.cleanup()
```

影片連結:

<https://youtube.com/shorts/fztHuRPQhy0?feature=share>

5.將以上實習結果貼在報告上

三、完成練習後，將樹莓派機依正常程序關機，2 種方式

1.在 SSH 的終端機下立即關閉命令 `$sudo shutdown -h now`

2.用 VNC viewer 登入，選[登出]-> [Shutdown](關機)，

等電源綠燈熄滅後才可關電源，取 SD 卡，以免系統檔毀損。

將實驗盒、樹莓派、HDMI 線、豆腐頭及電源線，交回至前方桌面。

四、心得(200 字以上，請自己寫不准直接貼 ChatGPT)

我之前在社團課程中學過 GPIO 的相關語法，雖然有些部分已經記不太清楚了，但在 AI 的幫助下，順利完成了所有程式碼的撰寫。其實，這部分並不是今天遇到的最大問題。真正讓我感到困擾的是實驗中的一個設備——蜂鳴器。它發出的聲音實在是太大聲了，甚至讓我的耳朵感到有些疼痛。除了這個小問題，整堂課的進行過程還算順利，並沒有遇到太多麻煩。然而，我大部分的時間都花在幫助其他同學解決他們遇到的問題上。也正因如此，我差點就沒能按時完成所有的實習項目。