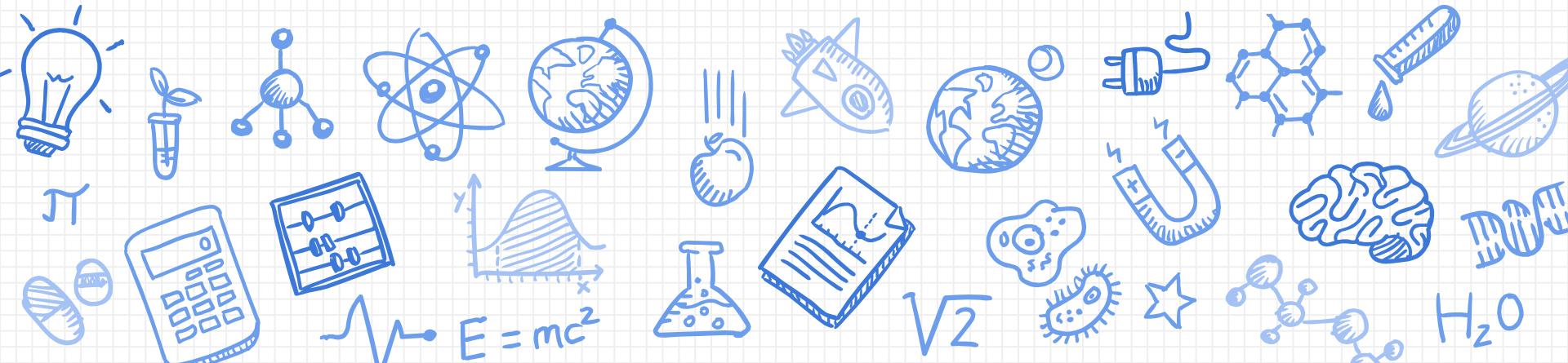


Python 程式設計與 Raspberry Pi GPIO 控制



Demo 1 - 關閉2.4G網卡

1. 找到2.4G網卡名稱：

```
$ iwconfig
```

2. Edit interfaces file

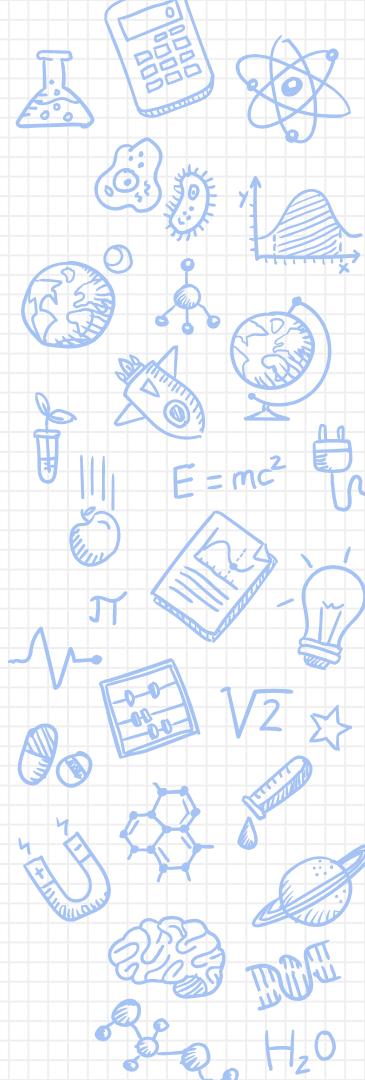
```
$ vim /etc/network/interfaces
```

將所有該網卡的設定以 “#” comment

3. Reboot

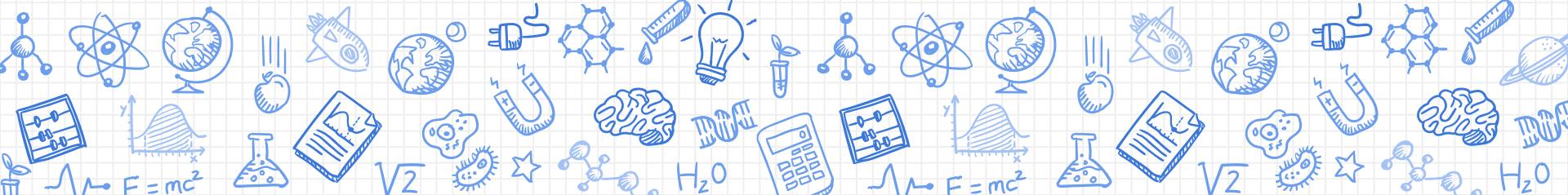
檢查網卡是否已被關閉

```
$ iwconfig
```

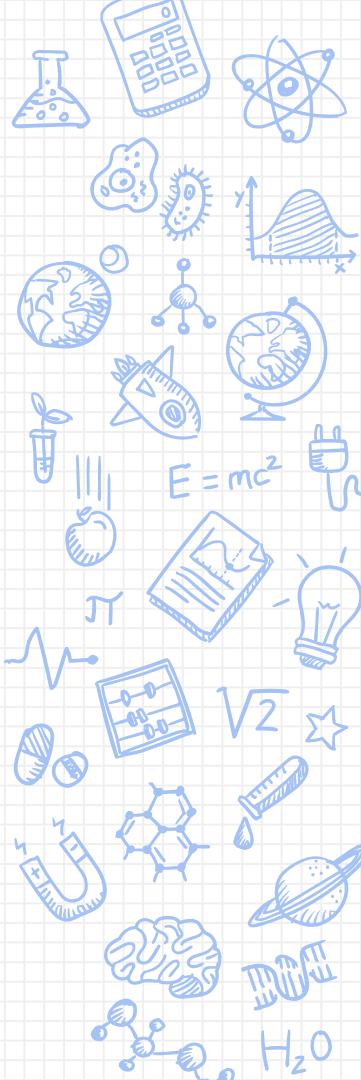
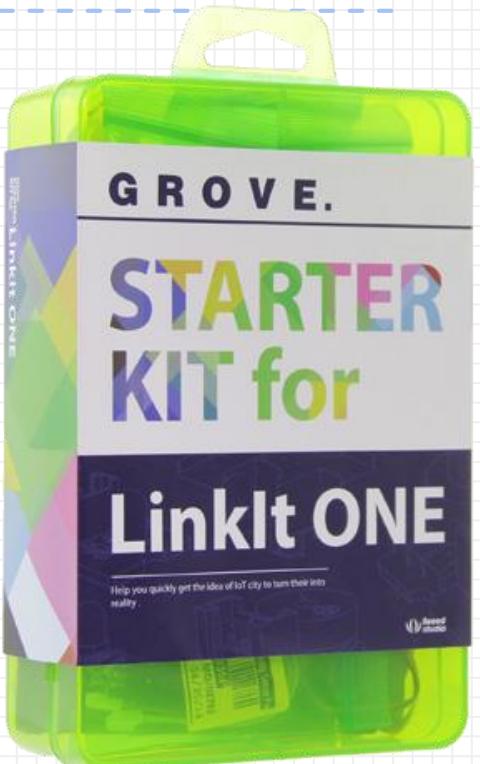
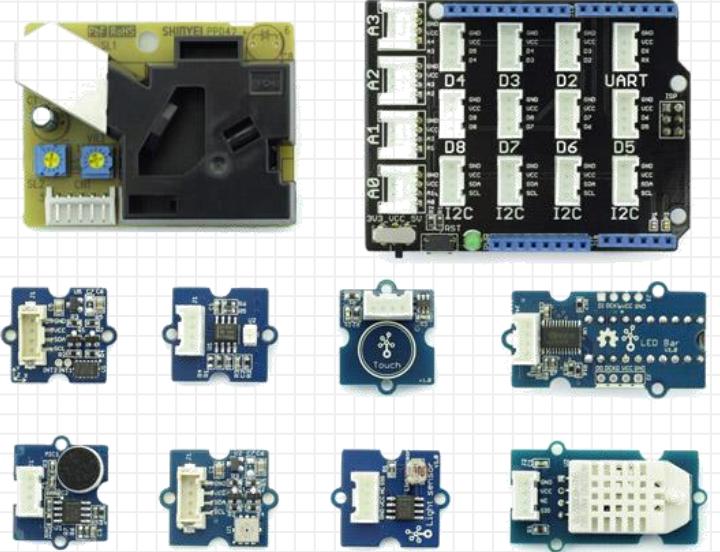


Grove Starter Kit for LinkIt ONE for Raspberry Pi

開發套件實戰 GPIO



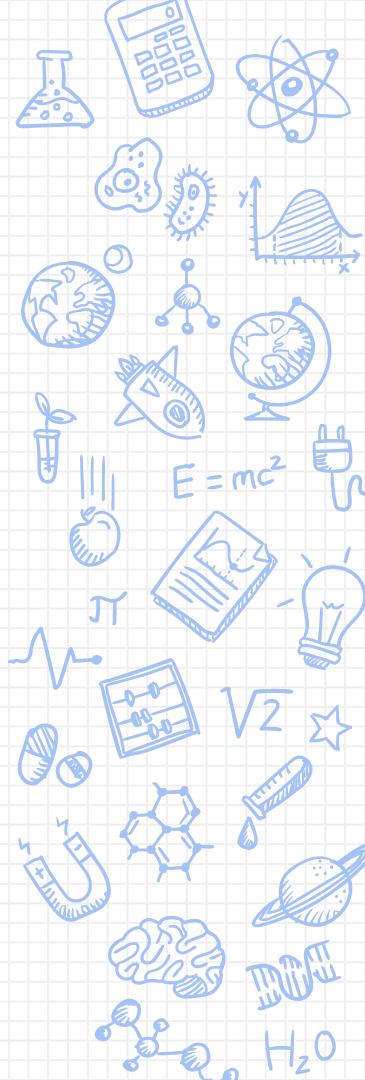
Grove Starter Kit for LinkIt ONE



<http://www.seeedstudio.com/depot/Grove-Starter-Kit-for-LinkIt-ONE-p-2028.html>

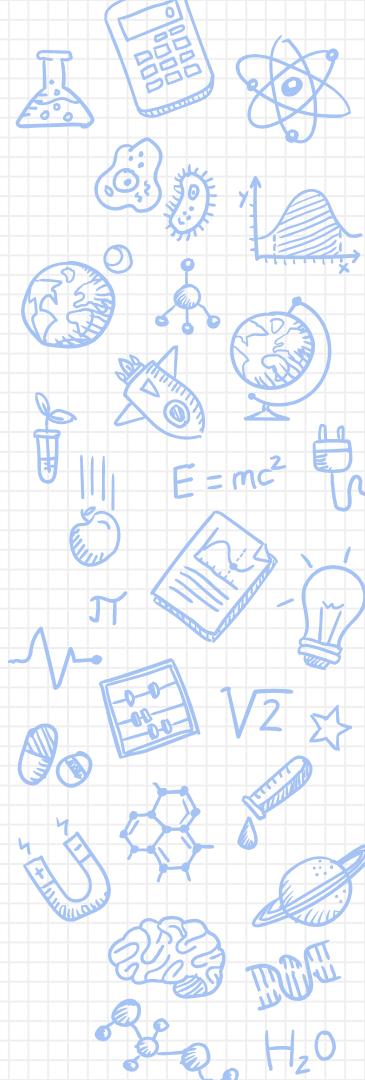
Grove Starter Kit for LinkIt ONE

- 粉塵感測器
- 溫濕度感測器
- 聲音感測器
- 紫外線感測器
- 氣壓感測器
- 光感測器
- 三軸加速度感測器
- LED 顯示條
- 伺服馬達
- 觸碰感測器
- 轉接板
- 說明書



Grove Starter Kit for LinkIt ONE

- 粉塵感測器
- 溫濕度感測器
- 聲音感測器
- 紫外線感測器
- 氣壓感測器
- 光感測器
- 三軸加速度感測器
- LED 顯示條
- 伺服馬達
- 觸碰感測器
- 轉接板
- 說明書



IEILab Grove Starter Kit 函式庫

- 函式庫名稱
`IEILab.GroveStartKit`
- 函式庫包含以下類別

DHT

- 溫溼度感測

LEDBar

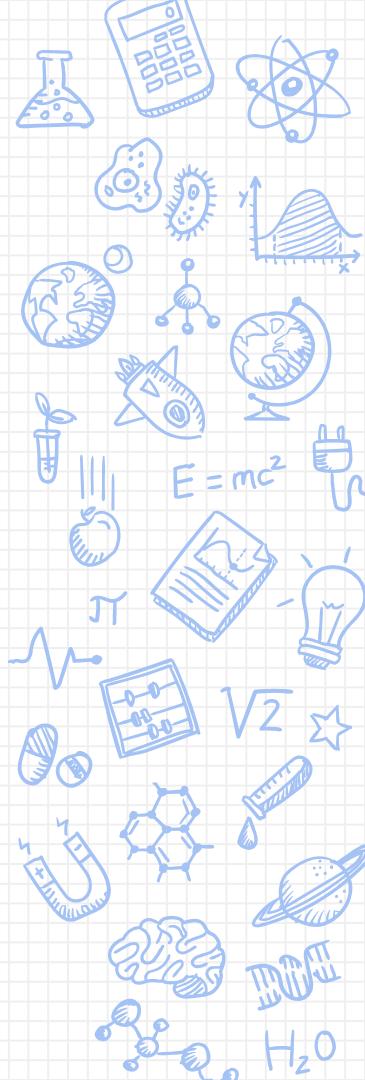
- LED 顯示條

Servo

- 同服馬達

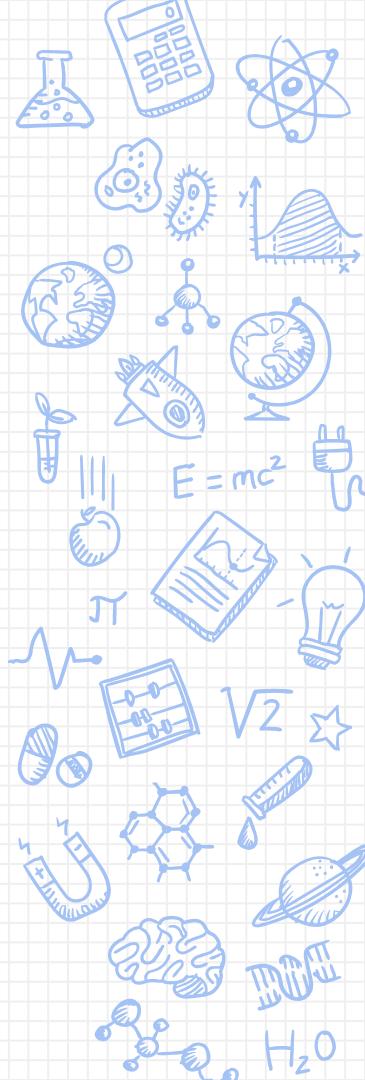
TouchSensor

- 觸碰感測器



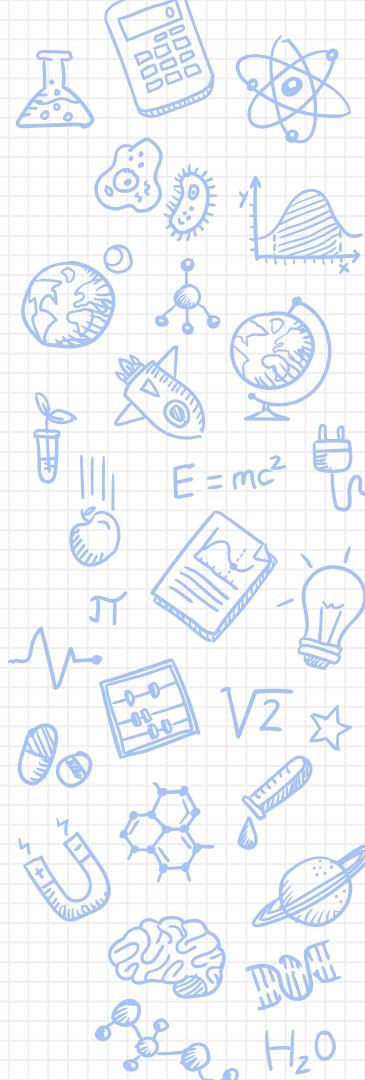
函式庫參考

- class DHT
 begin()
 readHT()
- class LEDBar
 setLevel(level)
 setLevelReverse(level)
 indexBit(index_bits)
 singleLed(num, state)



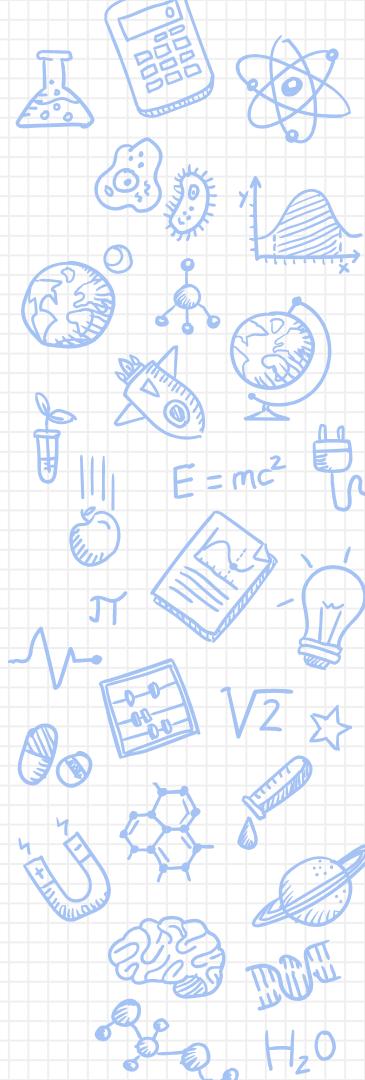
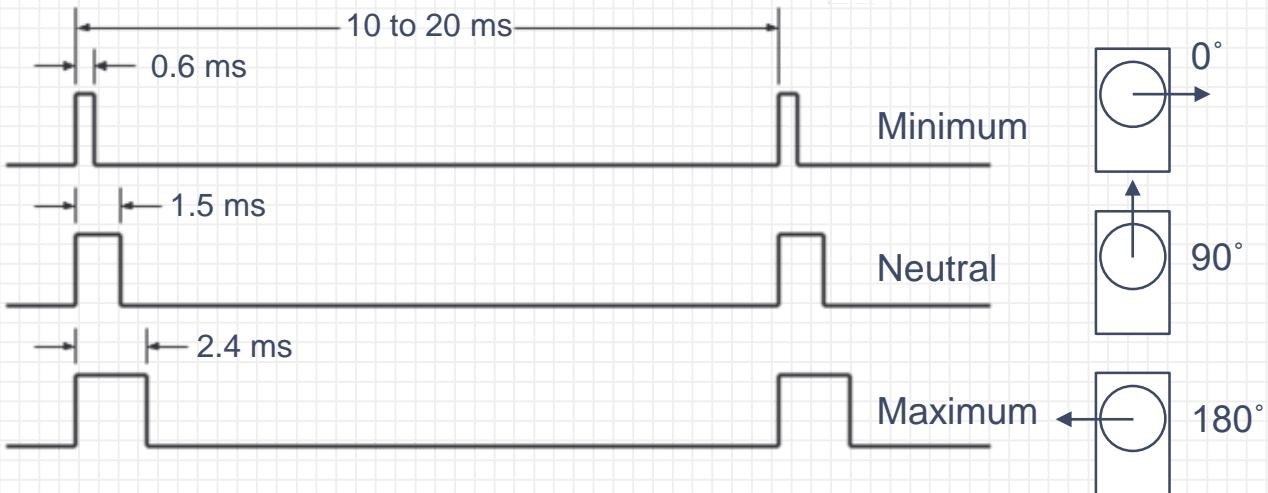
函式庫參考

- class Servo
 attach(pin)
 write(degree_angle)
 read()
- class TouchSensor
 attach(pin)
 isTouched()



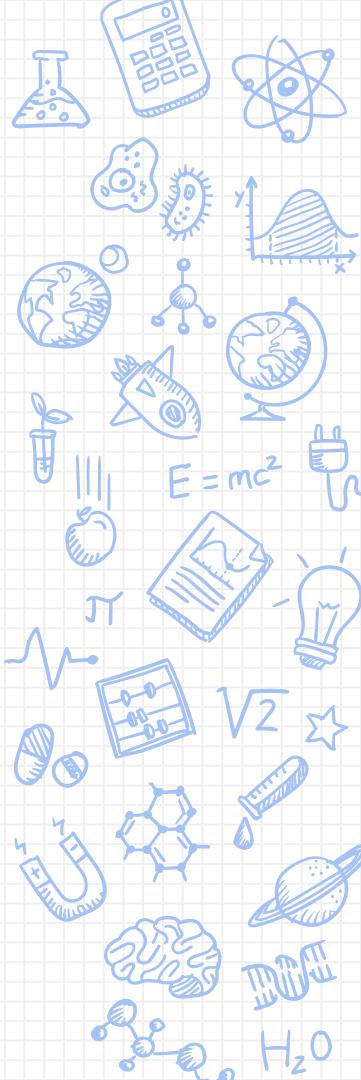
伺服馬達

- 用 pulse 控制轉動角度
- 機械結構限制只能轉 180 度

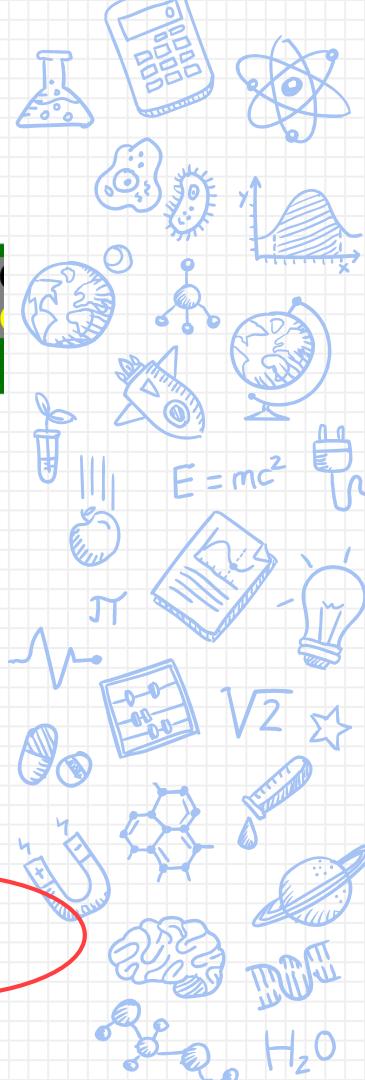
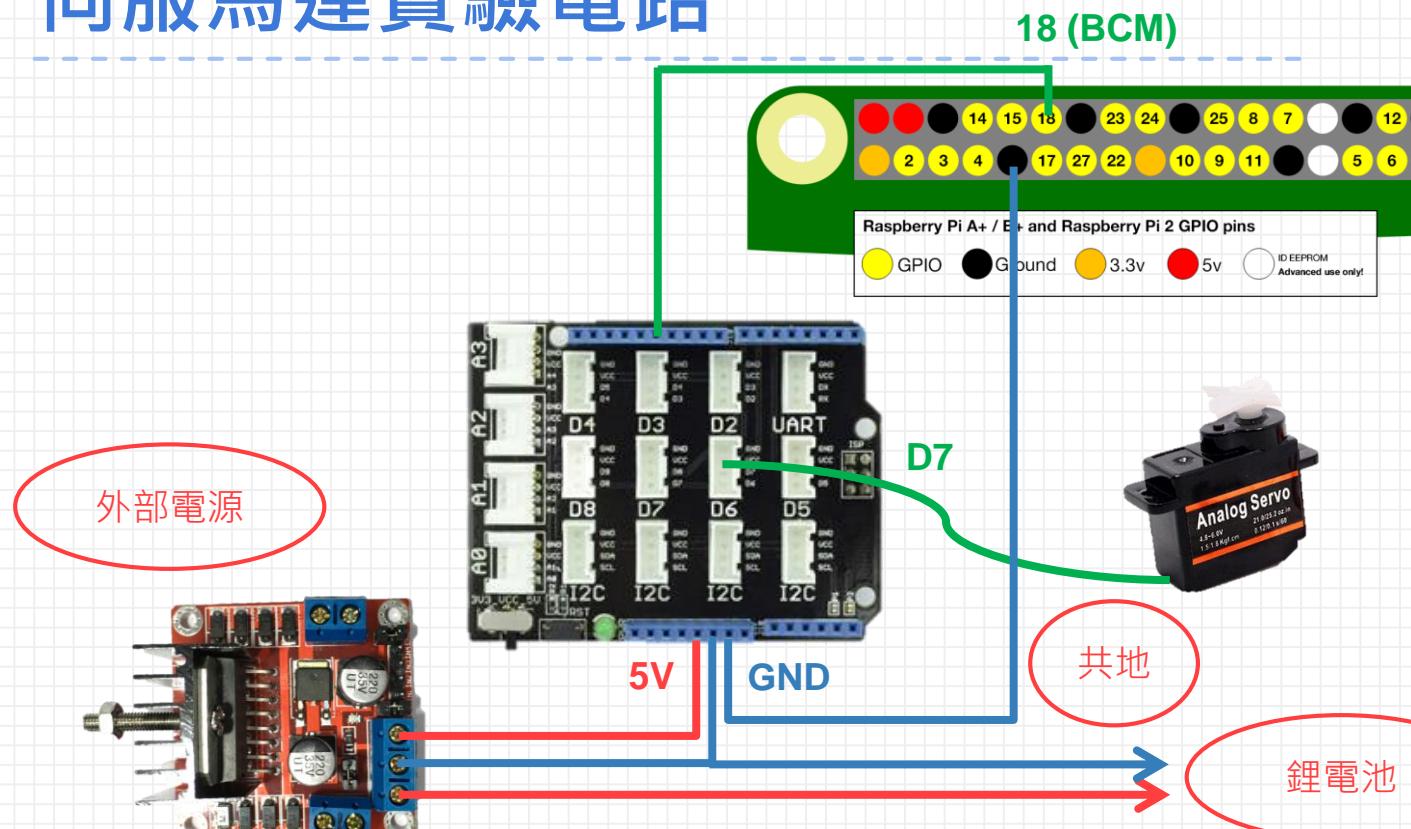


伺服馬達使用時注意事項

- 由於伺服馬達需要的電流量較大，因此我們必須採用**獨立電源**，才能供應伺服馬達足夠的電流量
- 電力微弱或未共地的情況下，伺服馬達會亂跳動，可能損壞馬達，請確認供電方式正確後再使用



伺服馬達實驗電路



伺服馬達 Python 程式

servo.py - Python

```
0 import RPi.GPIO as GPIO  
1 from IEILab.GroveStarterKit import Servo  
2 from time import sleep  
3  
4 GPIO.setmode(GPIO.BCM)  
5  
6 servo = Servo()  
7 servo.attach(18)      # PCM_CLK pin in BCM mode  
8  
9 servo.write(0)  
10 sleep(1)  
11 servo.write(180)  
12 sleep(1)
```

引入 GPIO 函式庫

引入 IEILab.GroveStarterKit 函式庫

選擇腳位編號方式

建立 servo 物件

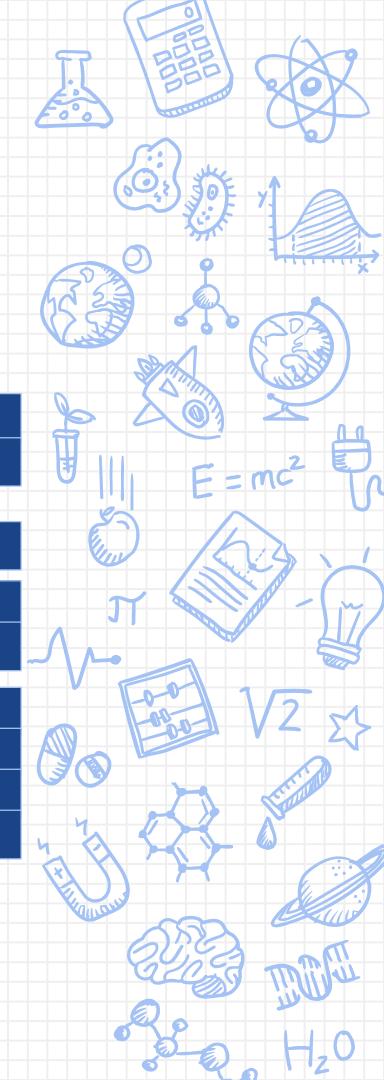
指定 18 號腳位

轉動到角度 0 的位置

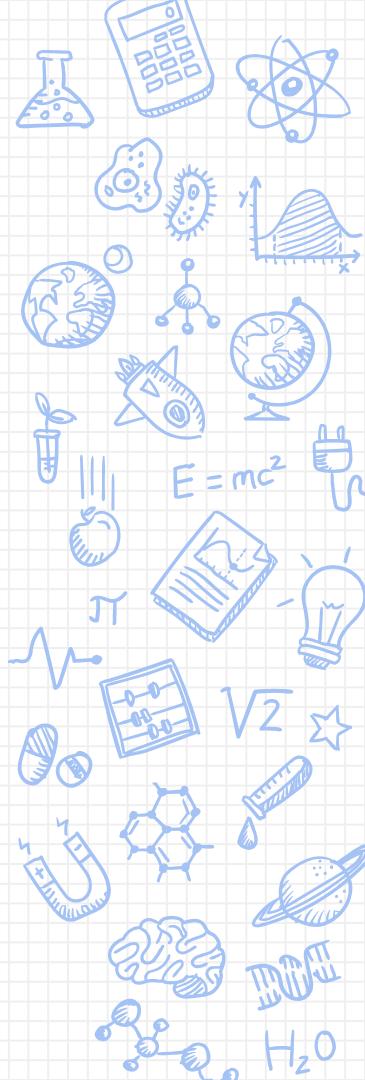
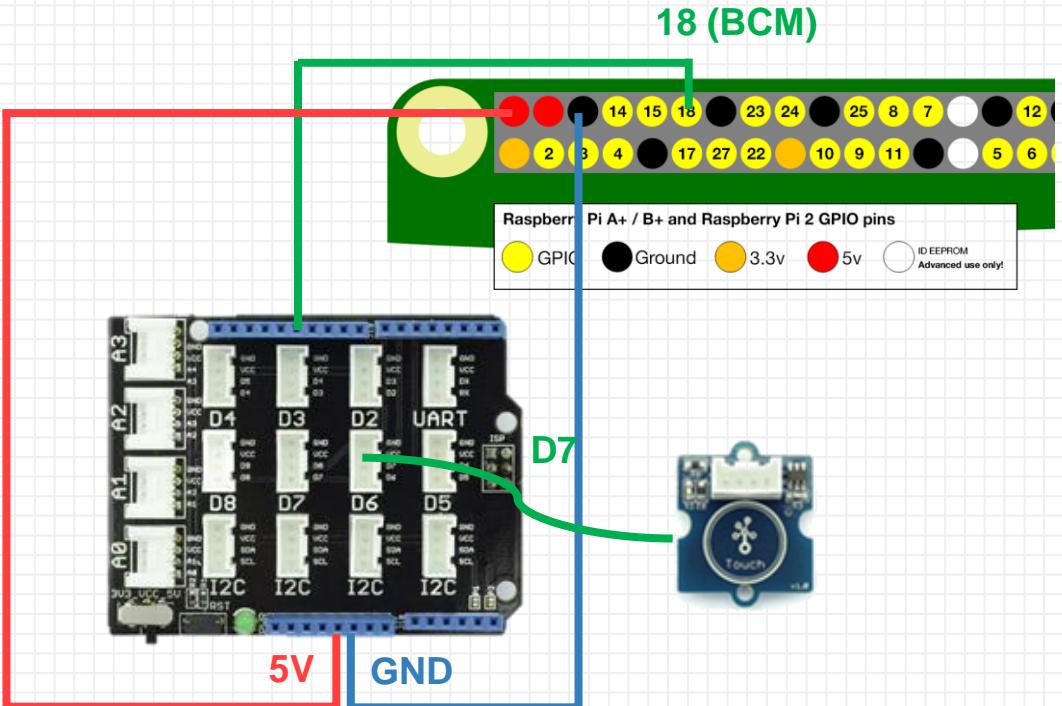
停 1 秒

轉動到角度 180 的位置

停 1 秒



觸碰感測器實驗電路



觸碰感測器Python 程式

touch.py - Python

```
0 import RPi.GPIO as GPIO
1 from IEILab.GroveStarterKit import TouchSensor
2 from time import sleep
3
4 PIN = 18
5
6 GPIO.setmode(GPIO.BCM)
7 touch = TouchSensor()
8 touch.attach(PIN)
9
10 try:
11     while True:
12         print 'Touched' if touch.isTouched() else ''
13         sleep(0.1)
14
15 except KeyboardInterrupt:
16     pass
17
18 GPIO.cleanup()
19
20
```

建立 `touch` 物件

指定 18 號腳位

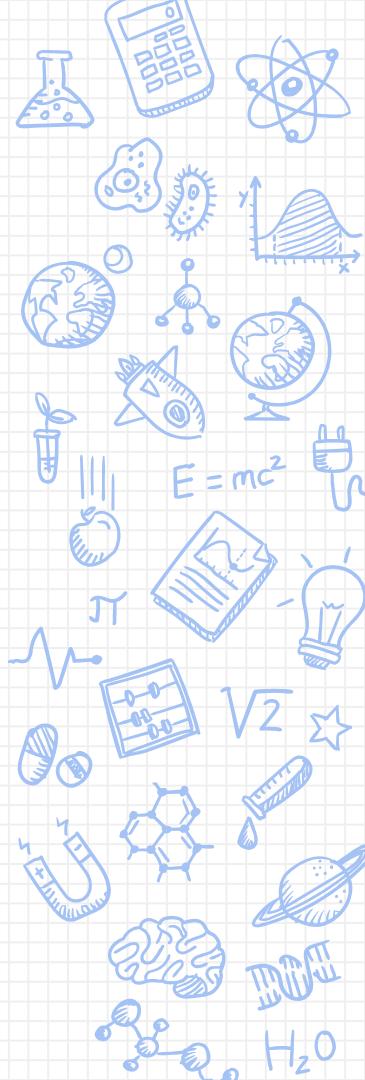
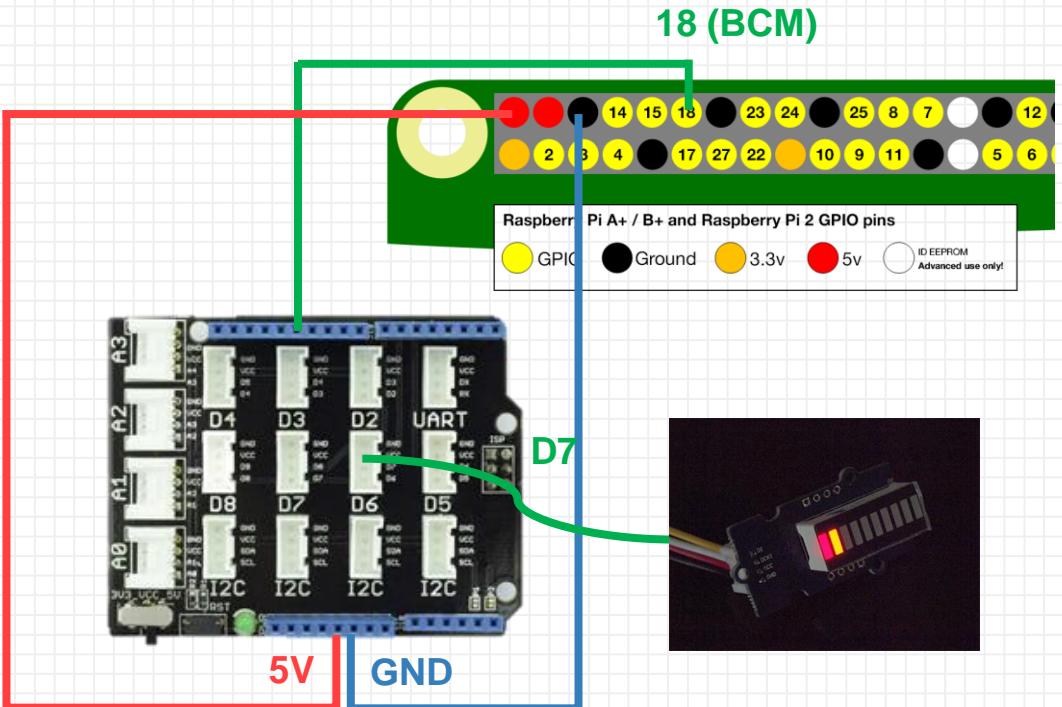
無限迴圈

有觸碰就顯示 `touched` 反之顯示空行

停 0.1 秒



LED 顯示條實驗電路



LED 顯示條 Python 程式

led.py – Python

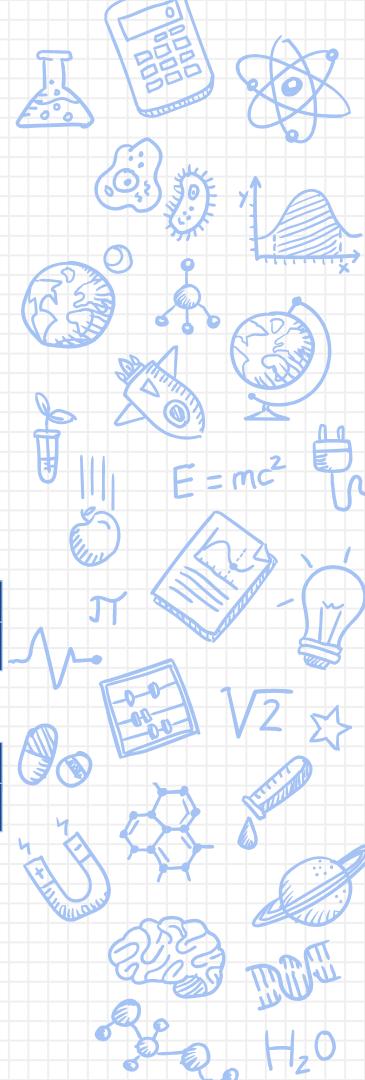
```
0 import RPi.GPIO as GPIO
1 from IEILab.GroveStarterKit import LEDBar
2 from time import sleep
3
4 clk = 23
5 dta = 18
6
7 GPIO.setmode(GPIO.BCM)
8 ledBar = LEDBar(clk, dta)
9
10 for level in range(11):
11     ledBar.setLevel(level)
12     sleep(0.1)
13
14 GPIO.cleanup()
```

建立 servo 物件

指定 18 號為 clock 腳位

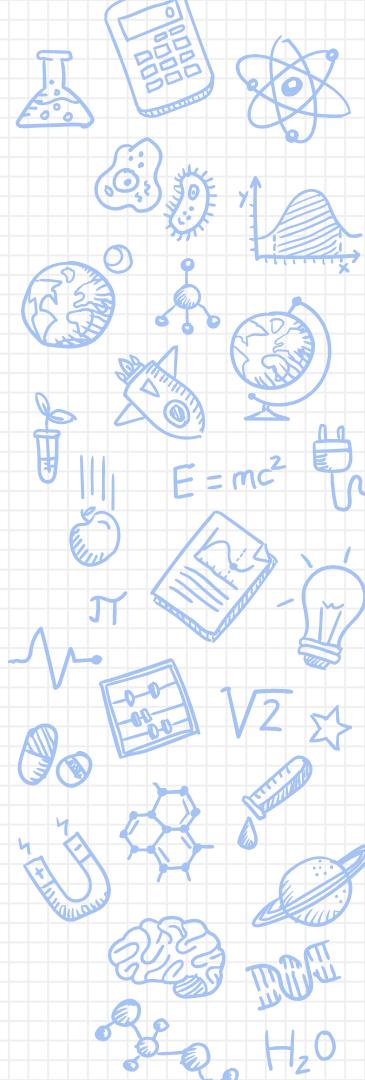
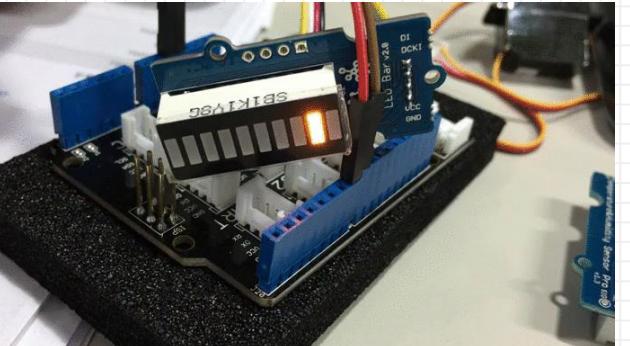
level 掃描 0 ~ 10

停 1 秒



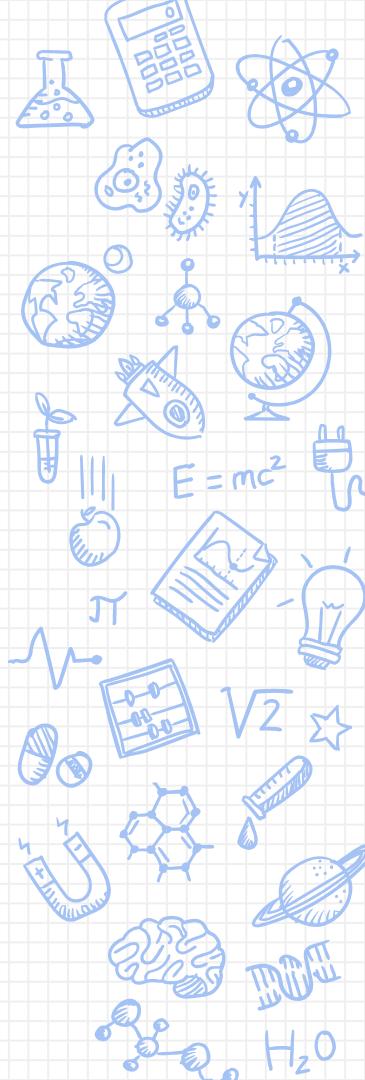
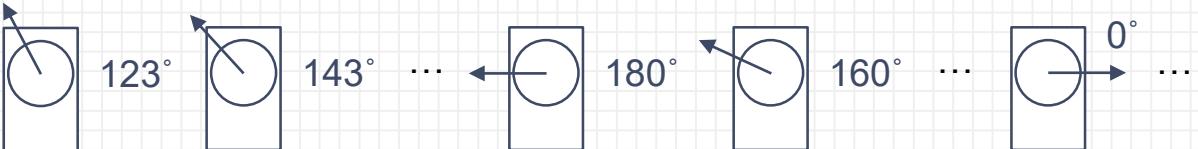
Demo 2

- 讓 LED 顯示條只顯示一顆的方式來回掃描，提示：
 1. 用 `ledBar.singleLed()`
 2. 開現在的燈並關上一個燈



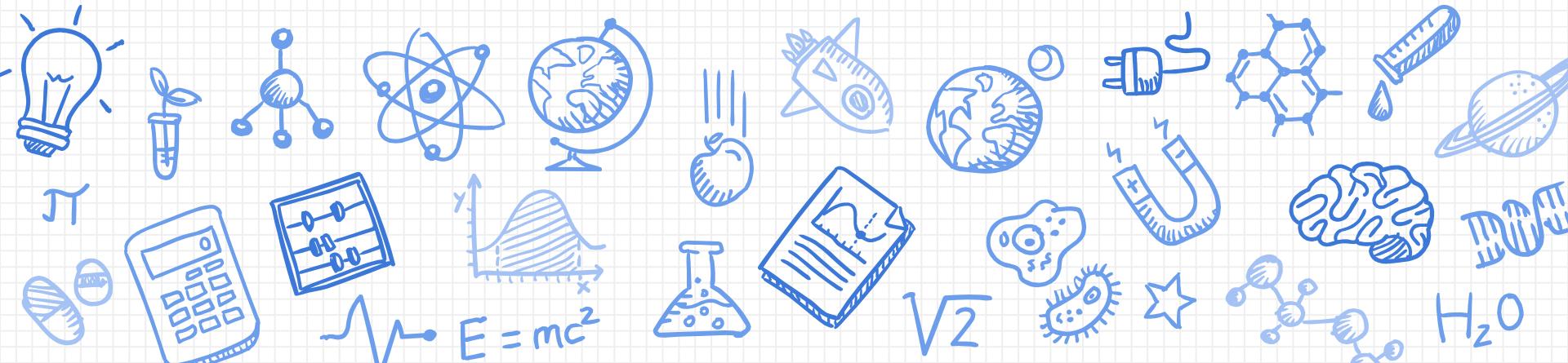
Demo 3

- 一開始伺服馬在任意狀態
- 每按下觸碰感測器，伺服馬達逆時針轉 20°
- 直到 180° (最左邊)，每當按下觸碰感測器順時針轉 10° 直到 0° (最右邊)，再變成逆時針，以此類推



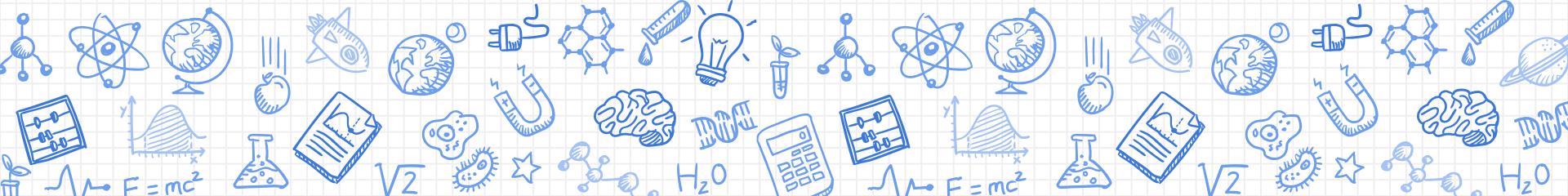
Raspberry Pi 物聯網

影像監控專案開發實驗



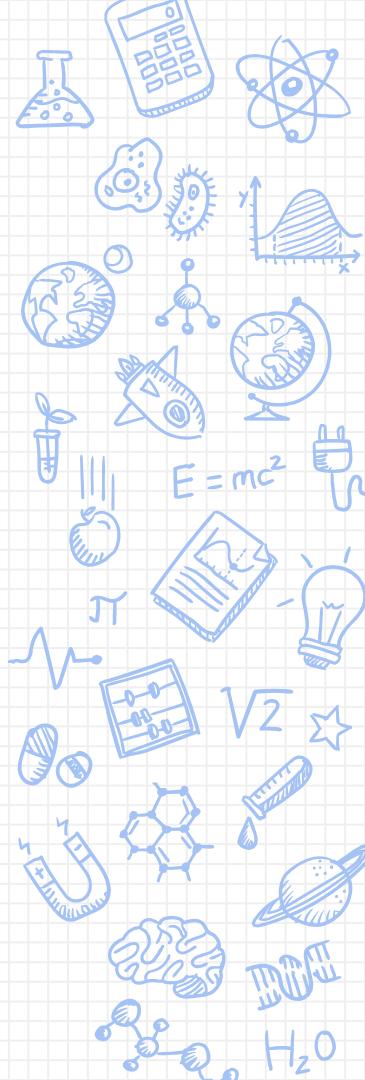
物聯網影像監控介紹

樹梅派很會欸



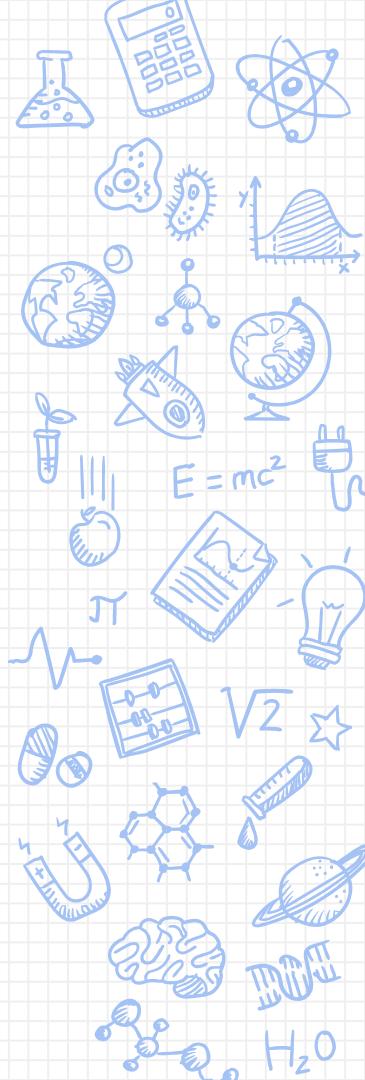
物聯網影像監控應用

- 綠色農業
- 公共安全
- 工業監控
- 城市管理
- 遠距醫療
- 智慧家庭
- 智慧交通
- 環境監控

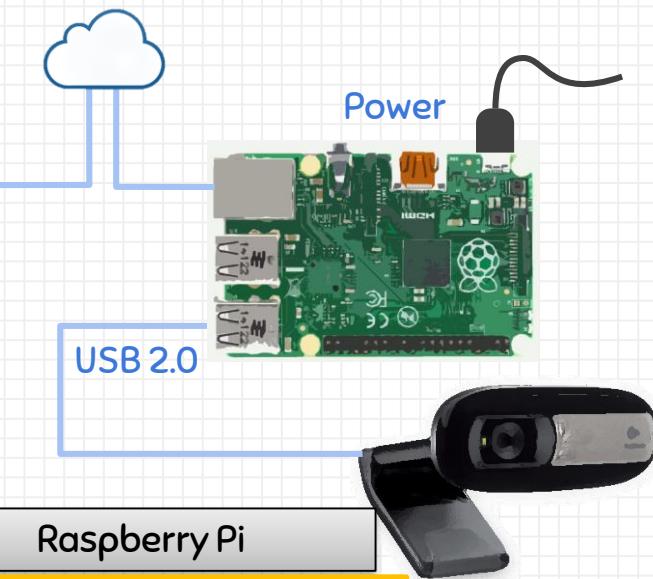


物聯網影像監控應用—智慧家庭

- 綠色農業
 - 公共安全
 - 工業監控
 - 城市管理
 - 遠距醫療
 - 智慧家庭
 - 智慧交通
 - 環境監控
- 居家安全
 - 老人照護
 - 嬰兒監視器
 - 寵物餵食



物聯網影像監控系統



影像輸出、警報等

影像串流

影像處理

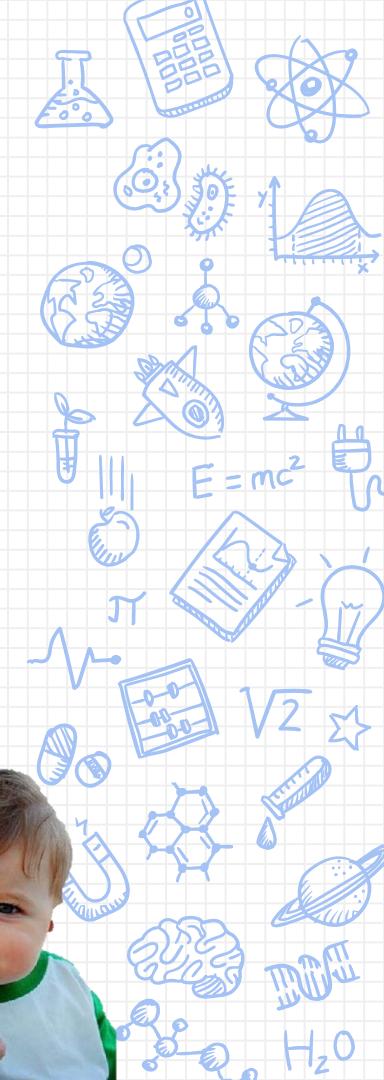
攝影機

影像輸出

影像處理

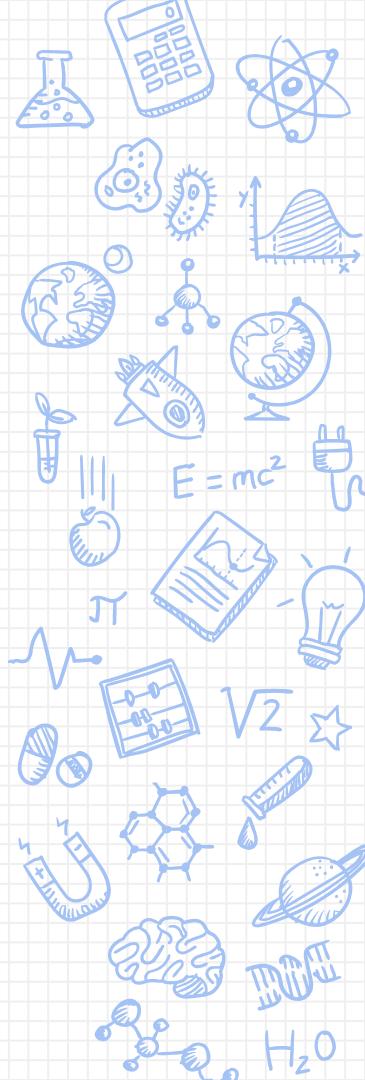
影像串流

攝影機



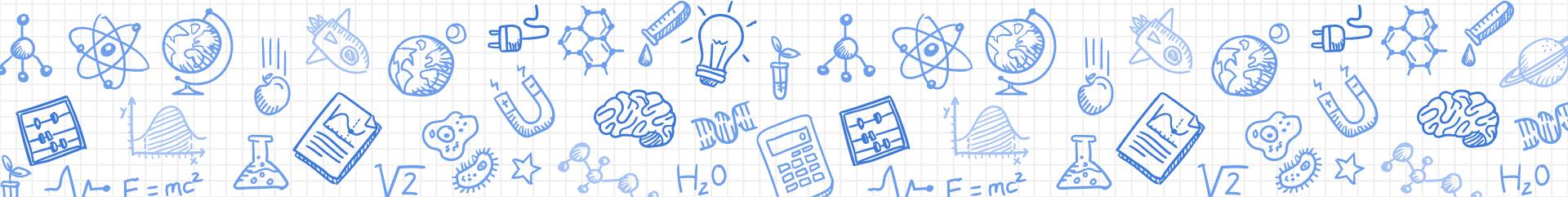
物聯網影像監控系統

- 網路攝影機串流
- 動作偵測
- 物件偵測
- 人臉偵測
- 自走車



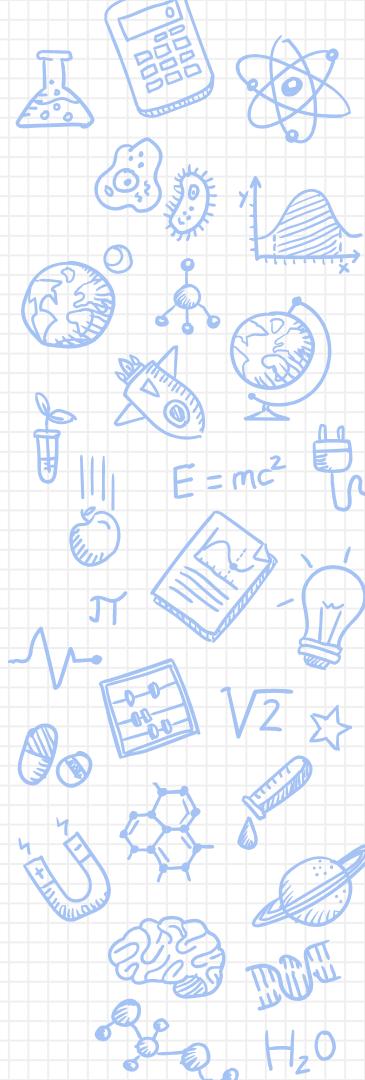
Raspberry Pi 網路攝影機串流

樹梅派看得見



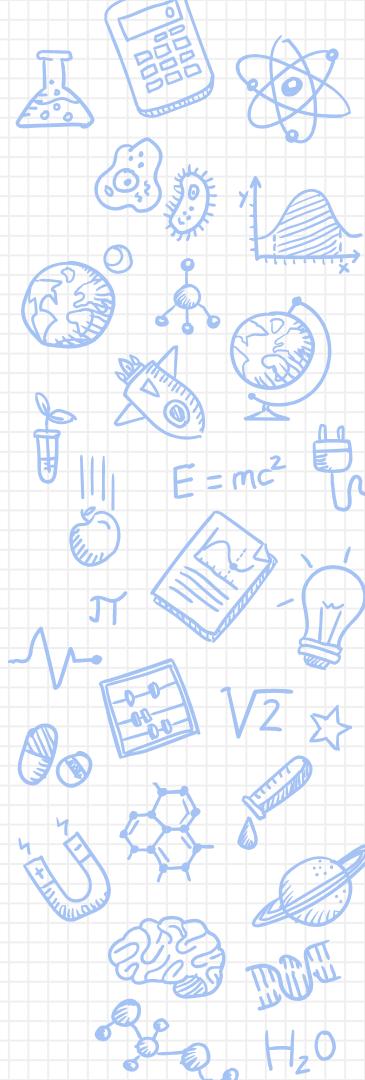
Raspberry Pi 網路攝影機串流

- Motion
- VLC
- OpenCV



Motion 串流 (1)

- 安裝 motion
`sudo apt-get install -y motion`
- motion 常駐程式
`sudo nano /etc/default/motion`
`start_motion_daemon=yes`

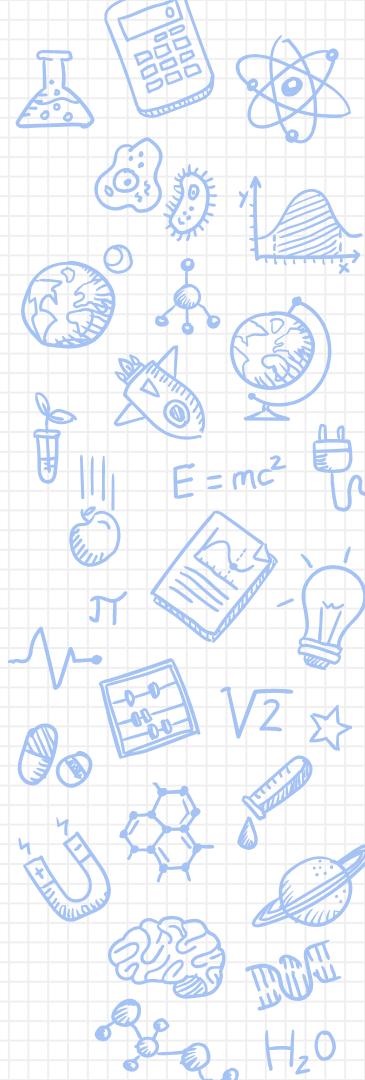


Motion 串流 (2)

- 更改 motion 設定檔
`sudo nano`

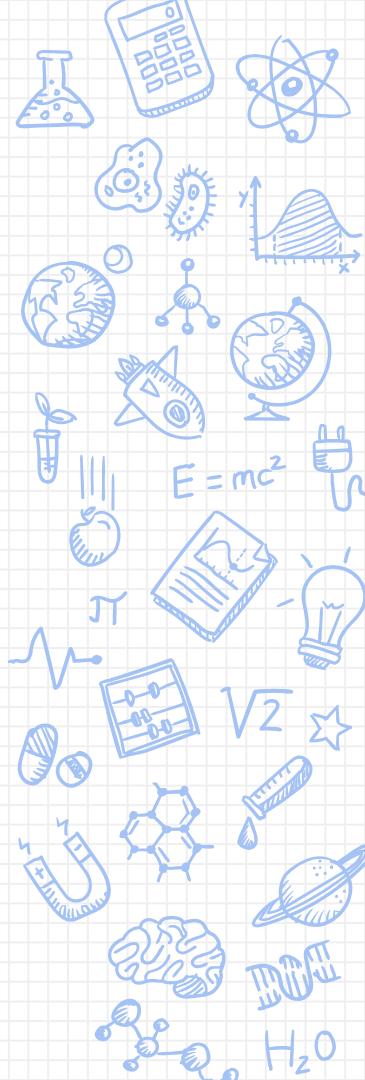
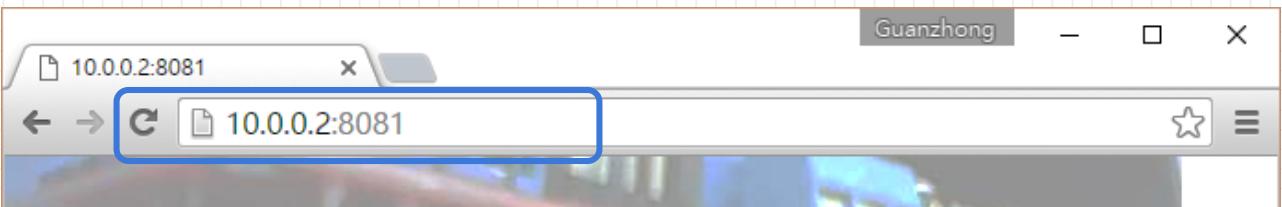
motion.conf - Raspberry Pi

```
Daemon on                                # 常駐
LogFile /tmp/motion.log                  # 錯誤記錄 (拿掉前方註解)
Width 640                                 # 長 640
Height 480                                # 寬 480
framerate 30                             # 放寬擷取畫格數限制
output_picture off                      # 停用動作偵測拍照
ffmpeg_output_movies off                 # 停用動作偵測錄影
stream_maxrate 30                         # 放寬串流畫格數限制
stream_localhost off                     # 停用只串流影像到 Raspberry Pi 本身
```



Motion 串流 (3)

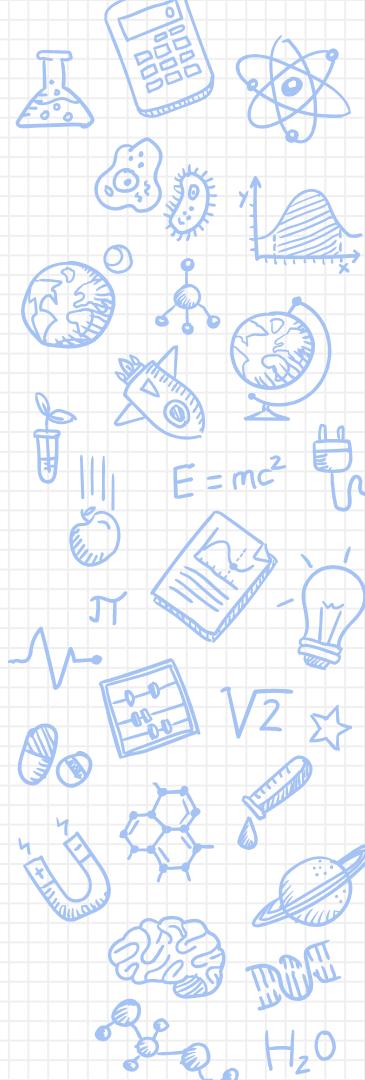
- 重新啟動 Raspberry Pi
`sudo reboot`
- 在個人電腦上的瀏覽器網址列上輸入
`http://10.0.0.2:8081/`
(假設燒錄本課程的映像檔，且照著連線步驟設定)



Motion 串流 (4)

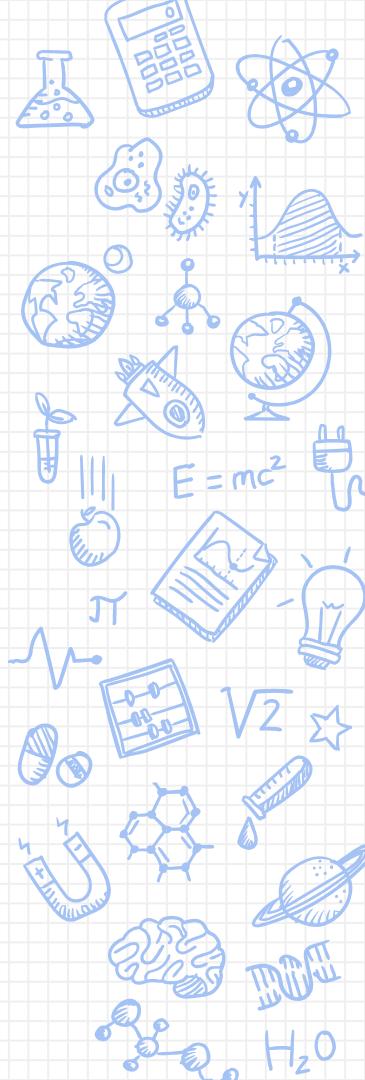
- 因為 motion 會一直佔用網路攝影機，讓其他程式沒辦法使用，請輸入以下指令停用 motion 再做後續的動作

`sudo service motion stop`



Raspberry Pi 網路攝影機串流

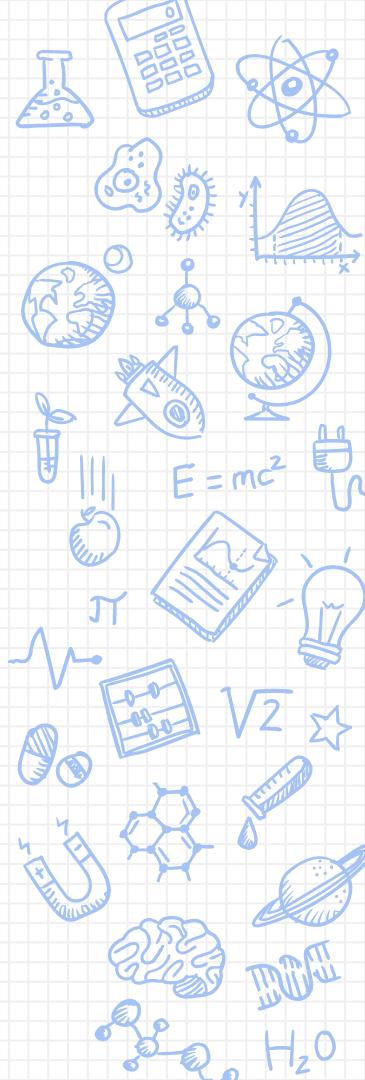
- Motion
- VLC
- OpenCV



OpenCV 串流 (1)

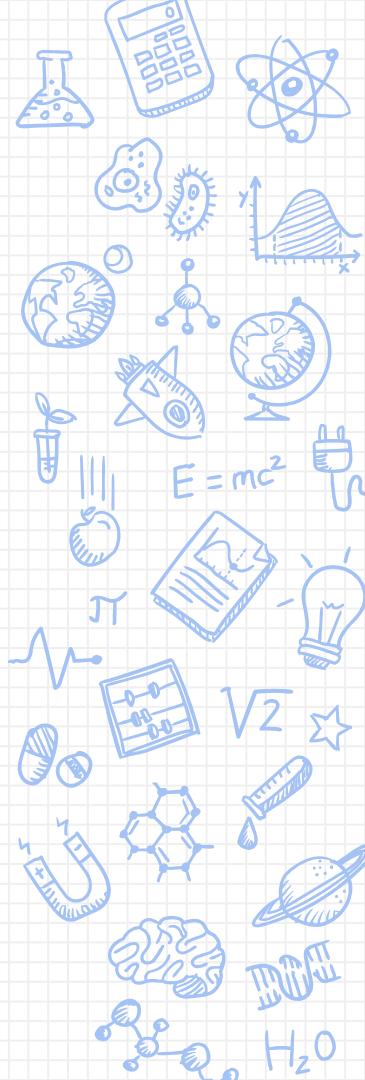
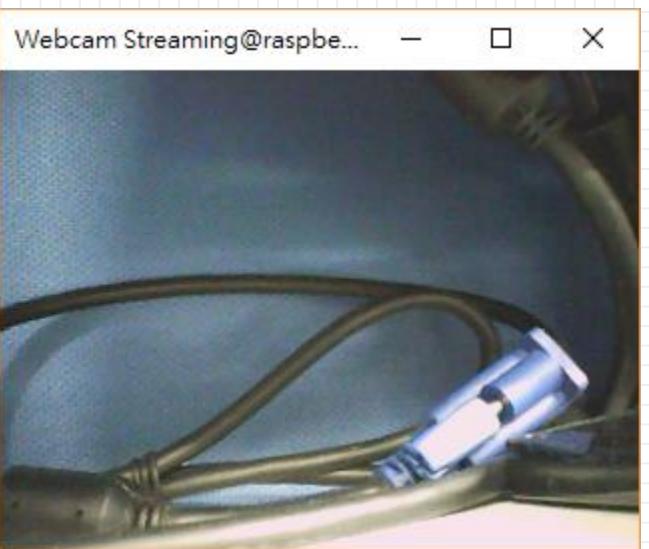
streaming.py - Python

```
0 import cv2
1
2 camera = cv2.VideoCapture(0)
3 camera.set(3, 320)          # width (max for c170: 640)
4 camera.set(4, 240)          # height (max for c170: 480)
5
6 while True:
7     # 摄取畫格
8     grabbed, frame = camera.read()
9
10    # 你可以在此做影像處理
11
12    # 顯示在視窗上
13    cv2.imshow('Webcam Streaming', frame)
14    if cv2.waitKey(1) & 0xFF == ord('q'):
15        break
16
17 camera.release()
18 cv2.destroyAllWindows()
```



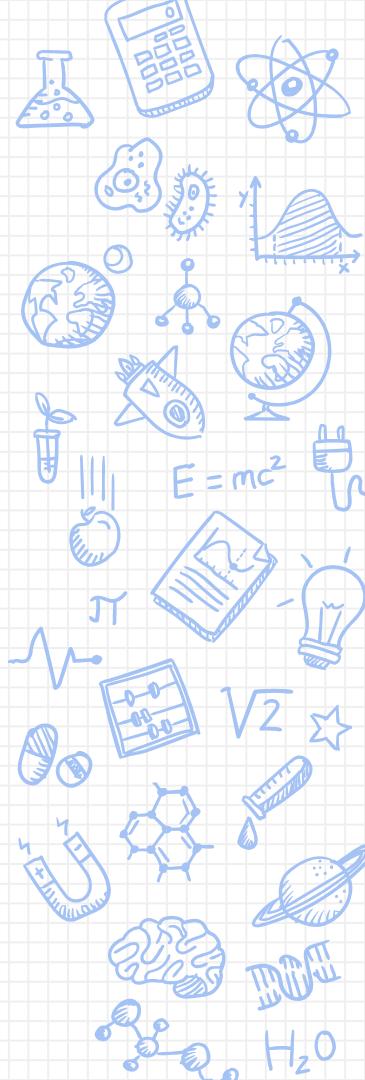
OpenCV 串流 (2)

- 執行 Python 程式
`python streaming.py`



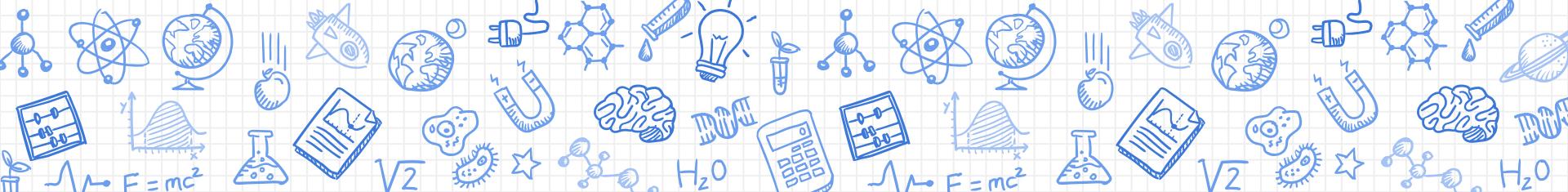
Demo 1

- 上網找找看，把 OpenCV 串流的影像顯示2個視窗
一個視窗顯示正常畫面
另一個視窗則左右翻轉



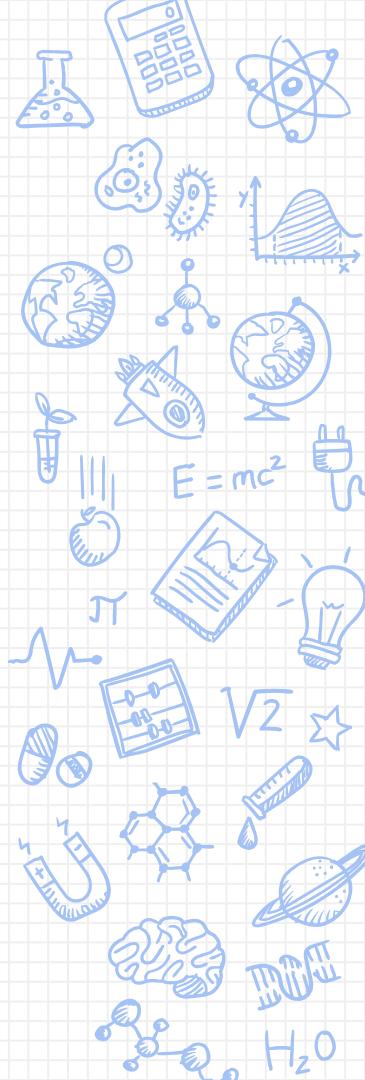
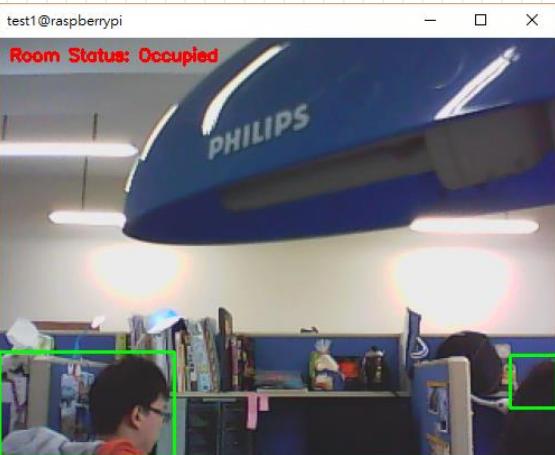
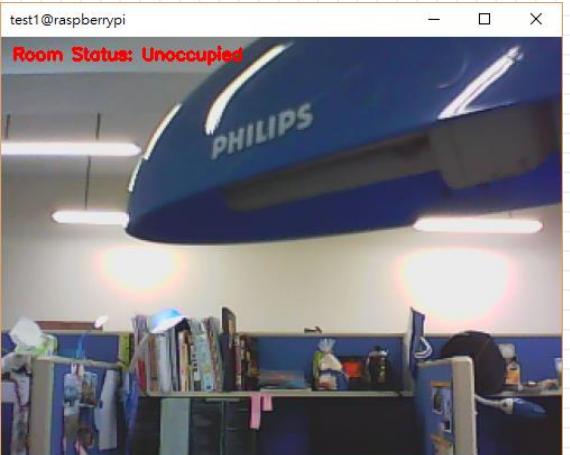
Raspberry Pi 動作偵測

樹梅派喜歡活潑的你

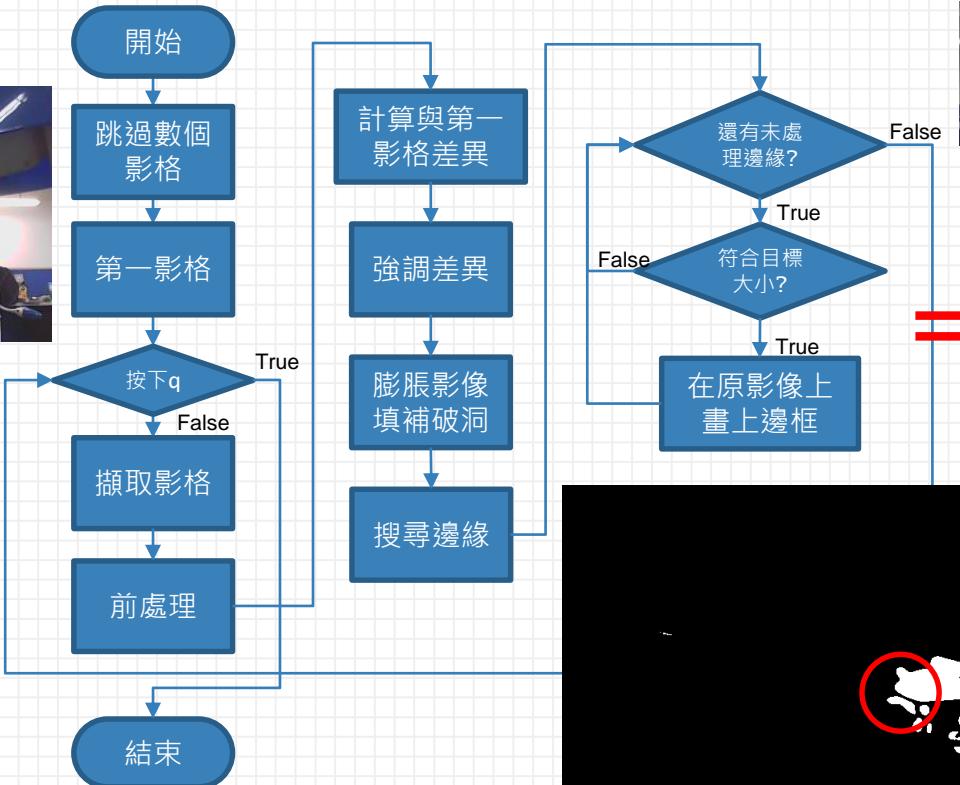


動作偵測 (1)

- 偵測畫面中的變化
 偵測入侵
 減少錄影檔案大小



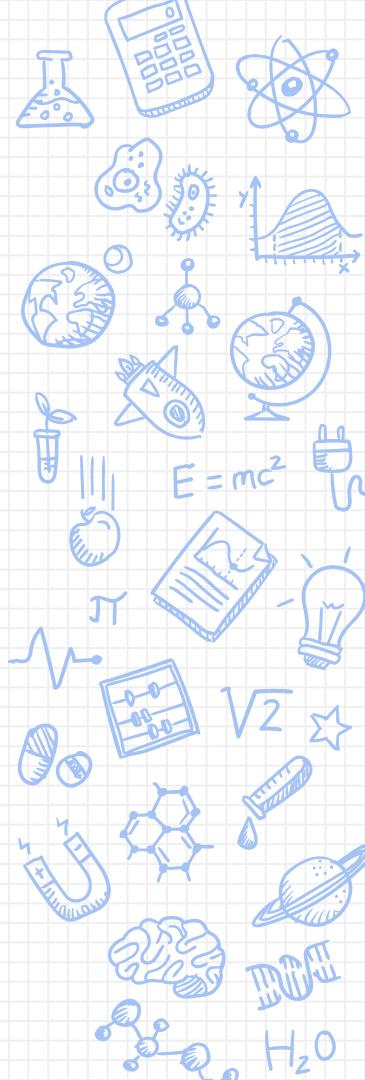
動作偵測 (2)



動作偵測 (3)

play_video.py (1) - Python

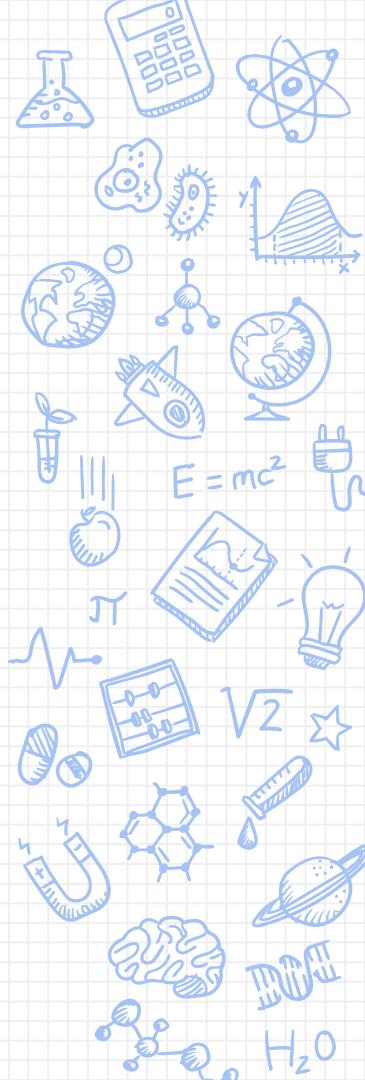
```
0 import time  
1 import cv2  
2  
3 camera = cv2.VideoCapture(0)  
4 camera.set(3, 320)  
5 camera.set(4, 240)  
6  
7 firstFrame = None  
8 skipFrame = 0  
9 while True:  
10     (grabbed, frame) = camera.read()  
11     text = "Unoccupied"  
12  
13     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)  
14     gray = cv2.GaussianBlur(gray, (21, 21), 0)
```



動作偵測 (3)

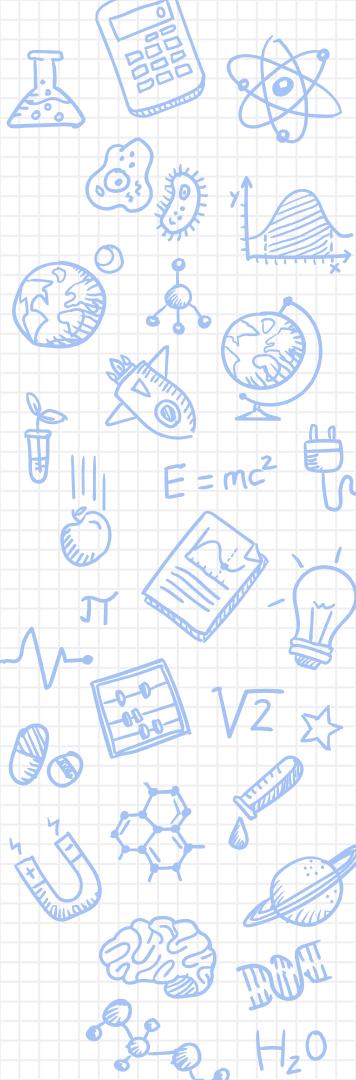
play_video.py (2) - Python

```
15
16     if firstFrame is None:
17         if skipFrame > 15:
18             firstFrame = gray
19
20     text = "Waiting"
21     cv2.putText(frame, "Room Status: {}".format(text), (10, 20),
22                 cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
23     cv2.imshow('test1', frame)
24     if cv2.waitKey(1) & 0xFF == ord('q'):
25         break
26     skipFrame += 1
27     continue
28
```

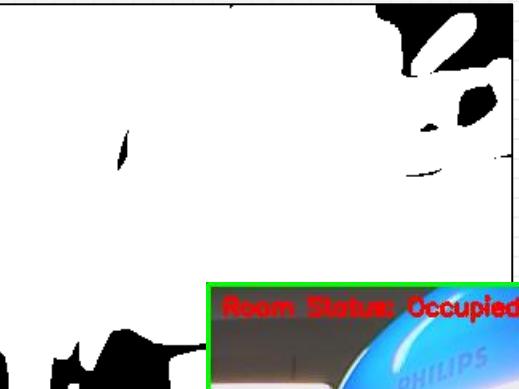
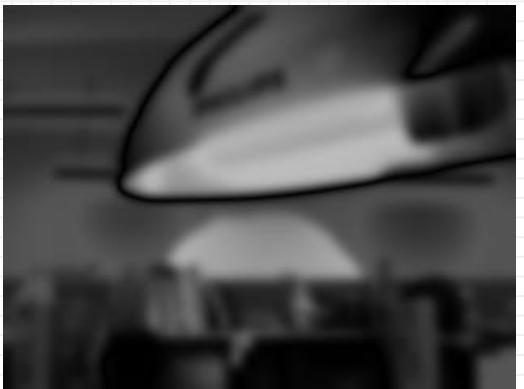


play_video.py (3) – Python

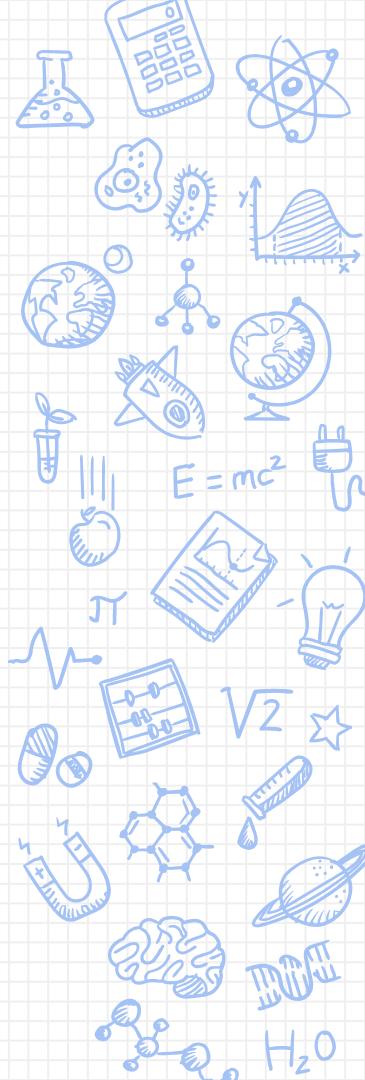
```
29     frameDelta = cv2.absdiff(firstFrame, gray)
30     thresh = cv2.threshold(frameDelta, 25, 255, cv2.THRESH_BINARY)[1]
31     cv2.imshow('test4', thresh)
32
33     thresh = cv2.dilate(thresh, None, iterations=2)
34     (contours, _) = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
35                                     cv2.CHAIN_APPROX_SIMPLE)
36
37     for c in contours:
38         if cv2.contourArea(c) < 500:
39             continue
40         (x, y, w, h) = cv2.boundingRect(c)
41         cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
42         text = "Occupied"
43
44         cv2.putText(frame, "Room Status: {}".format(text), (10, 20),
45                     cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
46
47         cv2.imshow('test1', frame)
48         if cv2.waitKey(1) & 0xFF == ord('q'):
49             break
50
51 camera.release()
52 cv2.destroyAllWindows()
```



動作偵測(4)——問題

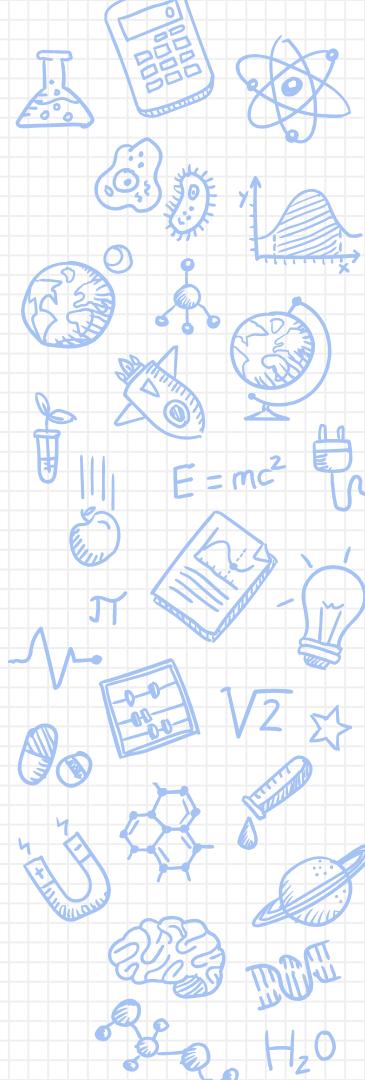


- 簡單的演算法沒辦法處理光源變化



Demo 2

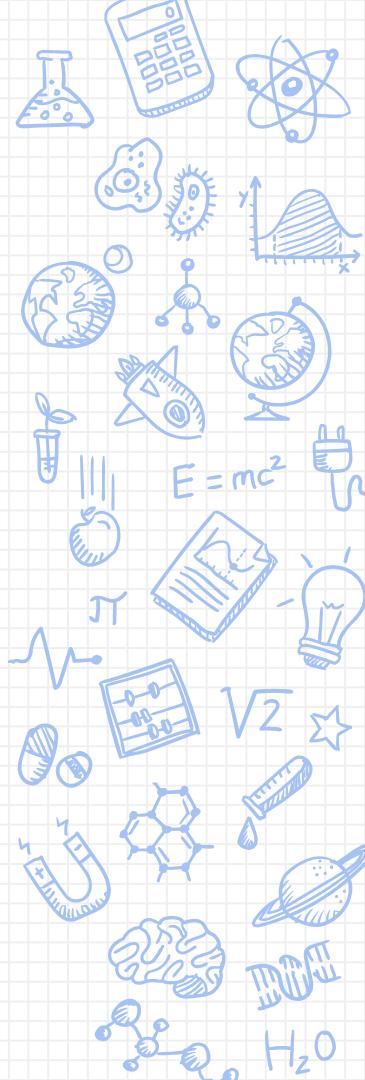
- 動作偵測 + 錄影功能
偵測到畫面有變化自動錄影
- 用 OpenCV 播放影片
- 提示：後2頁投影片程式碼教你如何用 OpenCV
錄影及播放影片



錄影

record_video.py - Python

```
0 import cv2
1
2 camera = cv2.VideoCapture(0)
3
4 # Define the codec and create VideoWriter object
5 fourcc = cv2.cv.CV_FOURCC(*'XVID')
6 out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))
7
8 while(camera.isOpened()):
9     ret, frame = camera.read()
10    if ret == True:
11        # write the frame
12        out.write(frame)
13        cv2.imshow('Record Video',frame)
14        if cv2.waitKey(1) & 0xFF == ord('q'):
15            break
16    else:
17        break
18
19 camera.release()
20 out.release()
21 cv2.destroyAllWindows()
```



播放影片

play_video.py - Python

```
0 import cv2
1
2 video = cv2.VideoCapture('output.avi')
3
4 while video.isOpened():
5     ret, frame = video.read()
6
7     if ret == False:
8         break
9
10    cv2.imshow('Play Video', frame)
11    if cv2.waitKey(1) & 0xFF == ord('q'):
12        break
13
14 cap.release()
15 cv2.destroyAllWindows()
```

