Raspberry Pi

互動科技藝術與感測設計

劉士達

課程大綱

- **1　電子材料清單介紹、認識Raspberry Pi 3 硬體、安裝Raspbian系統、GPIO控制(LED元件、按鈕元件)**
- 2　感測器模組與Raspberry Pi連接(繼電器、溫度、超音波距離感測器)
- 3　攝影機模組、SimpleCV影像處理、MCP3008+可變電阻
- 4　網路連線、伺服器架設、遠距馬達控制
- 5　物聯網、文字轉語音、Arduino應用開發
- 6　蘑菇總動員 - 範例製作

課前環境設定

- 無線AP帳號/密碼：
  - SSID = RaspberryPi_AP
  - 密碼 = raspberry
- 預設的樹梅派帳號/密碼：
  - 帳號：pi
  - 密碼：raspberry
- IP位置查找：
  - Advanced IP Scanner
- 有線網路連接：
  - 拿出2M長的網路線連接至Hub(用此方法比較穩定)

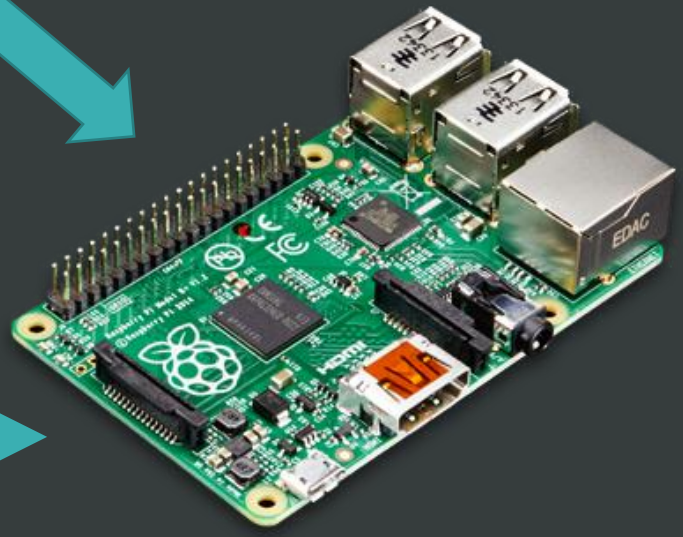
網路基本傳輸方式實作

- Socket Server / Client 傳送與接受
- Processing + Rpi Socket Server/Client
- Processing + Rpi Camera with Socket Server
- Processing + GPIO with Socket Server

Client – Server 連線方式

發送資料

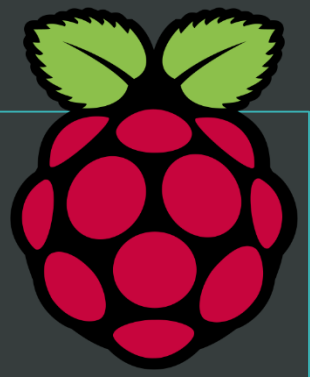
等待接收



Client端

TCP連線

Socket – Server端 Step 1

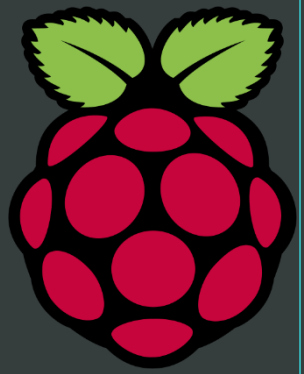


```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

host = ‘192.168.3.4’
port = 1688
```

```
#匯入Socket
#建立socket物件
AF_INET = 使用IPv4        SOCK_STREAM = TCP模式
AF_INET6 = 使用IPv6       SOCKET_DGRAM = UDP模式
#host = 要建立的Server IP 自己的pi
#port = 要建立的Server port 可以自訂
```

Socket – Server端 Step 2

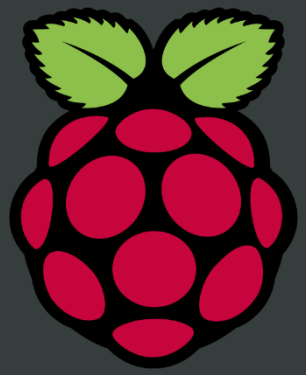
```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

host = ‘192.168.3.4’
port = 1688

s.bind((host,port))
s.listen(5)

conn,addr = s.accept()
```



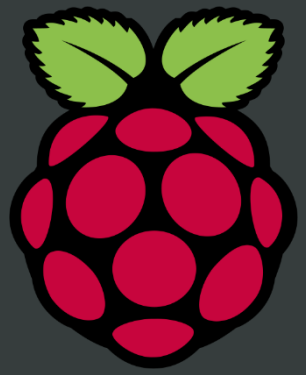
```
#bind((host,port))  建立起Server
#listen(5) 等待Client連線，最大連線數為5

#conn, addr 兩個變數接收Socket client連線，用accept()連接
```

Socket – Server端 Step 3

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

host = ‘192.168.3.4’
port = 1688

s.bind((host,port))
s.listen(5)

conn,addr = s.accept()

while True:
    data = conn.recv(1024)
    if data:
        print(“Receive Data: %s” % str(data.decode(‘utf-8’)))
    else:
        break
conn.close()
```



```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
print("Socket Created")
host = '192.168.3.4'
port = 1688

s.bind((host,port))
s.listen(5)

conn,addr = s.accept()
print("Client Address: %s" % str(addr))
while True:
    data = conn.recv(1024)
    if data:
        print("Receive Data: %s" % str(data.decode('utf-8')))
    else:
        break
conn.close()
```

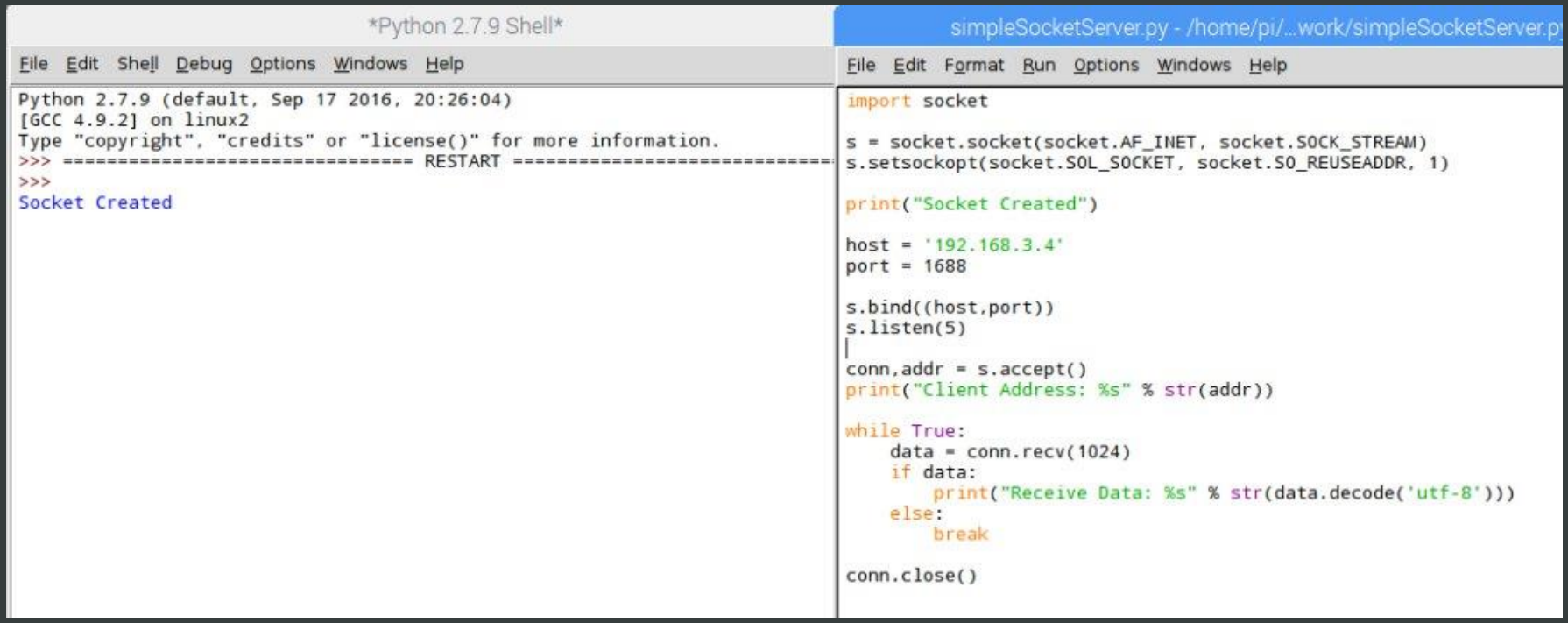


simpleSocketServer.py

simpleSocketServer.py說明

```
#s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

setsockopt(level, optname, value)
設置socket的額外選項

SOL_SOCKET = 使用SOCKET特殊選項
SO_REUSEADDR = 設置重複使用該address，若沒設定此選項，當關閉server端時會無法重新建立需等待幾分鐘時間，若設定後再斷開Server端時立即釋放該連接埠
1 = 設定該optname的value為1 -> True
```

建立Socket Server後等待client

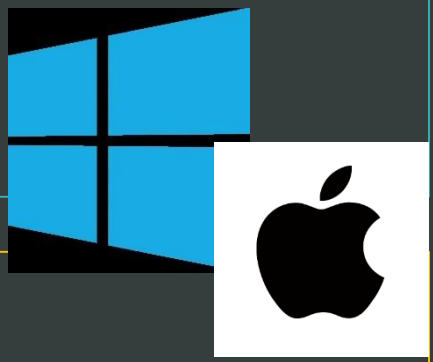


建立Client端可以使用Win/Mac

- Windows需下載安裝Python2.x版或3.x並安裝
- Mac內建Python
  - 打開一個記事本另存檔名為simpleSocketClient.py
  - 在MAC命令列執行`$ python simpleSocketClient.py`

Socket – Client端 Step 1

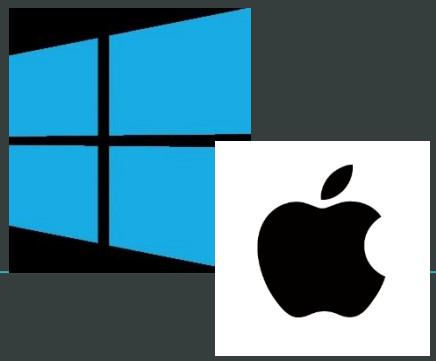
```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

host = ‘192.168.3.4’
port = 1688
```



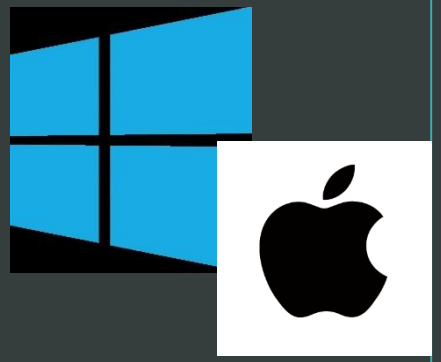
```
#匯入Socket
#建立socket物件
AF_INET = 使用IPv4          SOCK_STREAM = TCP模式
AF_INET6 = 使用IPv6         SOCKET_DGRAM = UDP模式
#host = 要連線的 IP
#port = 要連線的目標Server port
```

Socket – Server端 Step 2

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

host = ‘192.168.3.4’
port = 1688

s.connect((host,port))

while True:
    data = “Hello Pi:”
    s.sendall(data.encode(‘utf-8’))

s.close()
```

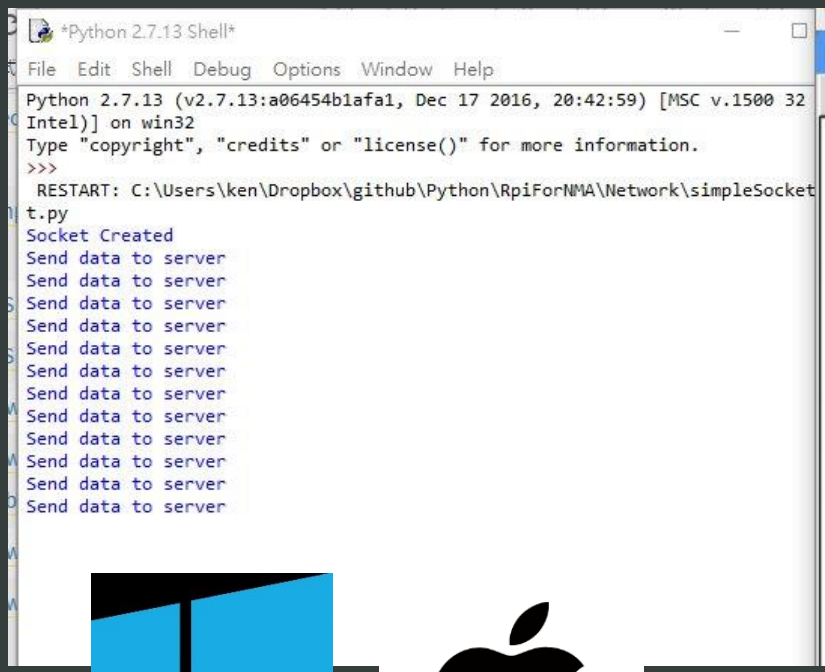


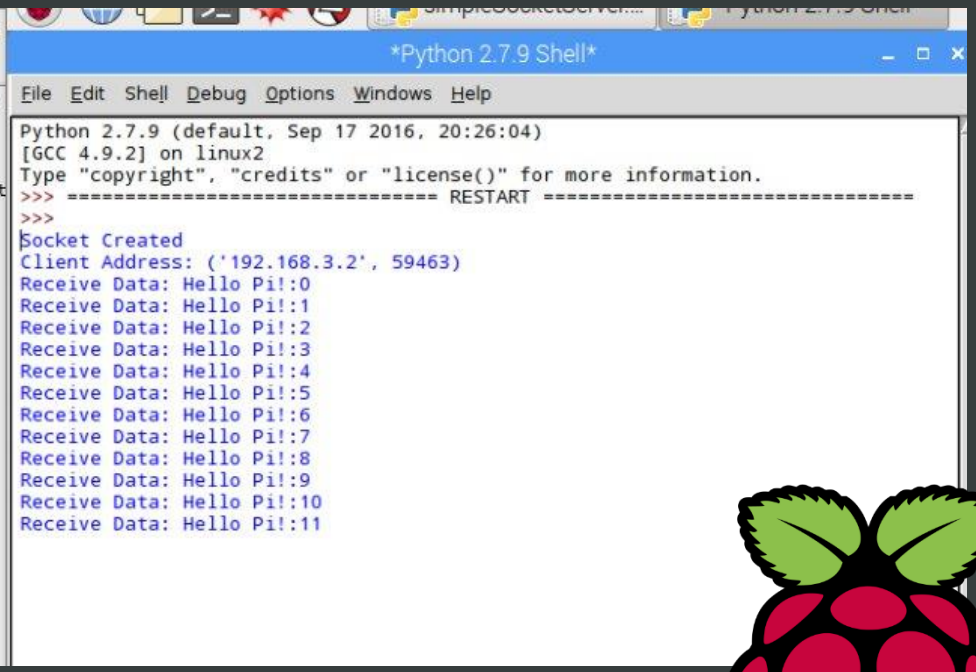
```
import socket
import time

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

print("Socket Created")
host = '192.168.3.4'
port = 1688
s.connect((host,port))

counter = 0
while True:
    data = "Hello Pi!:" + str(counter)
    s.sendall(data.encode('utf-8'))
    print("Send data to server")
    time.sleep(1)
    counter += 1

s.close()
```

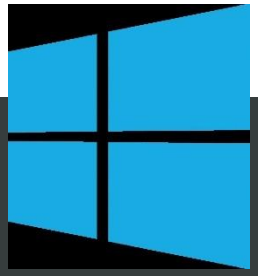


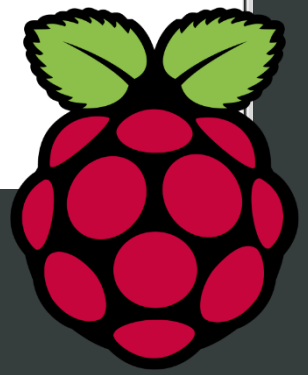
simpleSocketClient.py

Win -> Client Rasperry Pi -> Server









Server – Client 連線方式

發送資料

等待接收

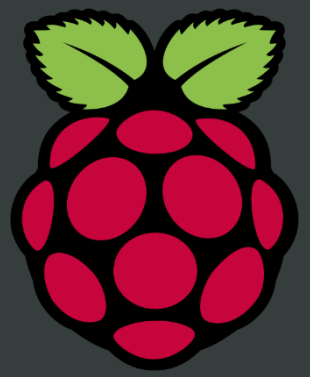
Client端

TCP連線

Server端

Server端回傳data

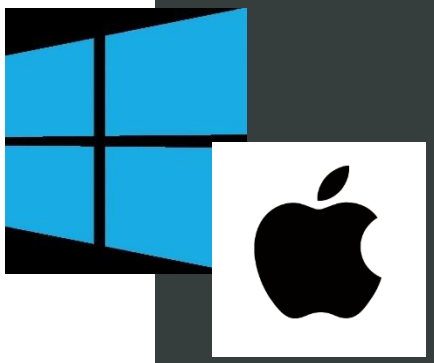
```
import socket
import time

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

print("Socket Created")

host = '192.168.3.4'
port = 1688

s.bind((host,port))
s.listen(5)

conn,addr = s.accept()
print("Client Address: %s" % str(addr))
counter = 0
while True:
    data = "I'm Rpi:" + str(counter)
    conn.send(data.encode('utf-8'))
    time.sleep(1)
    counter += 1

conn.close()
```



simpleSocketServer_sendData.py

Client端接收data

```
import socket
import time

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

print("Socket Created")

host = '192.168.3.4'
port = 1688
s.connect((host,port))

counter = 0

while True:
    data = s.recv(1024)
    print(data.decode('utf-8'))

s.close()
```



simpleSocketClient_recvData.py

Raspberry -> 傳送　　Win -> 接收

Python 2.7.9 Shell

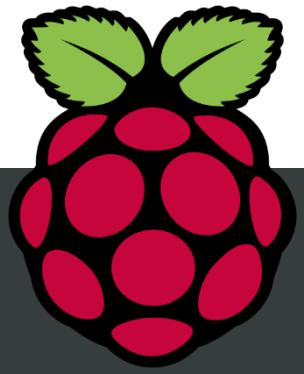
File Edit Shell Debug Options Windows Help

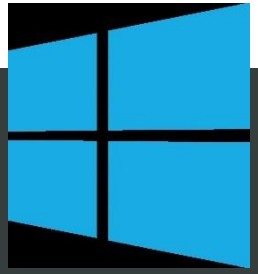
```
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more info
>>> ================================ RESTART ================================
>>>
Socket Created
Client Address: ('192.168.3.2', 60706)
```

Python 2.7.13 Shell

File Edit Shell Debug Options Window Help

```
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:/Users/ken/Dropbox/github/Python/RpiForNMA/Network/simpleSocketClien
t_recvData.py
Socket Created
I'm Rpi:0
I'm Rpi:1
I'm Rpi:2
I'm Rpi:3
I'm Rpi:4
I'm Rpi:5
I'm Rpi:6
I'm Rpi:7
I'm Rpi:8
I'm Rpi:9
I'm Rpi:10
I'm Rpi:11
I'm Rpi:12
```





用Processing -> Raspberry 連線方式

等待接收

發送資料

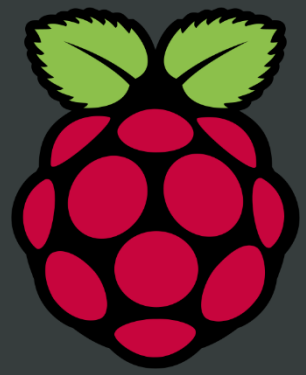
TCP連線

Client端

Server端

使用剛剛的simpleSocketServer_sendData.py

```
import socket
import time

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

print("Socket Created")

host = '192.168.3.4'
port = 1688

s.bind((host,port))
s.listen(5)

conn,addr = s.accept()
print("Client Address: %s" % str(addr))
counter = 0
while True:
    data = "I'm Rpi:" + str(counter)
    conn.send(data.encode('utf-8'))
    time.sleep(1)
    counter += 1

conn.close()
```



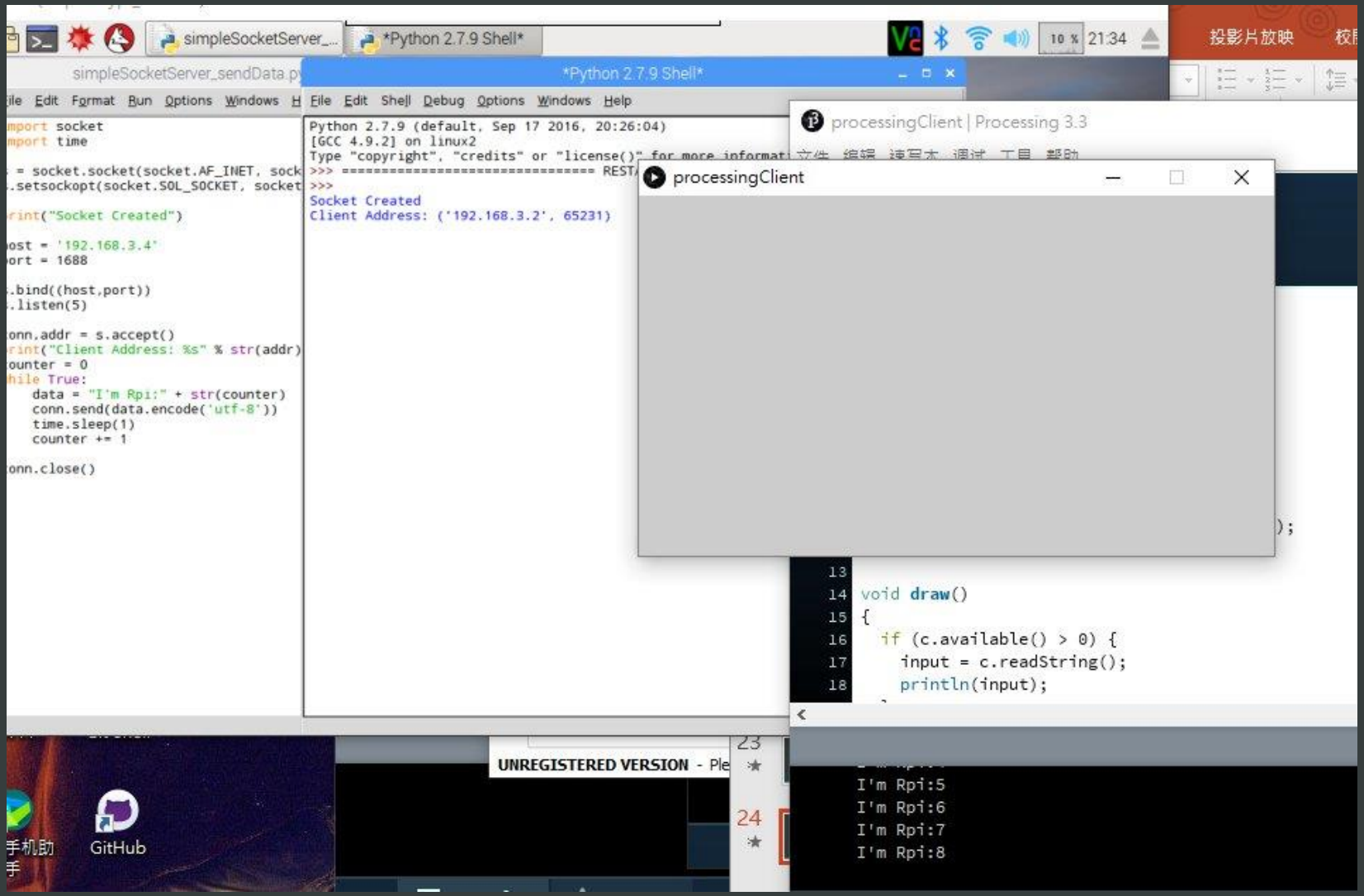
simpleSocketServer_sendData.py

```
import processing.net.*;

Client c;
String input;
int data[];

void setup()
{
  size(450, 255);
  background(204);
  c = new Client(this, "192.168.3.4", 1688);
}

void draw()
{
  if (c.available() > 0) {
    input = c.readString();
    println(input);
  }
}
```

processingClient.pde

Raspberry + Processing 的 Socket Client Receive Data

Raspberry Pi -> sendData -> Processing



用Processing -> Raspberry 連線方式

發送資料

等待接收

TCP連線

Client端

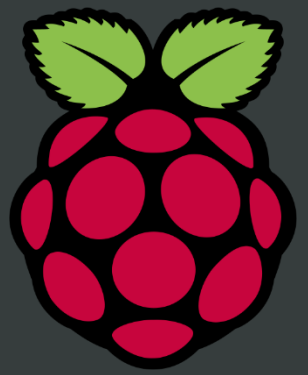
Server端

```
6  boolean on = false;
7
8  void setup()
9  {
10   size(450, 255);
11   background(204);
12   c = new Client(this, "192.168.3.4", 1688);
13 }
14
15 void draw()
16 {
17   if (c.available() > 0) {
18     input = c.readString();
19     println(input);
20   }
21 }
22
23 void mouseClicked(){
24   if(on){
25     on = !on;
26     c.write("ON");
27     background(204);
28   }else{
29     on = !on;
30     c.write("OFF");
31     background(0);
32   }
33 }
```

processingClient_withMouseClick.pde

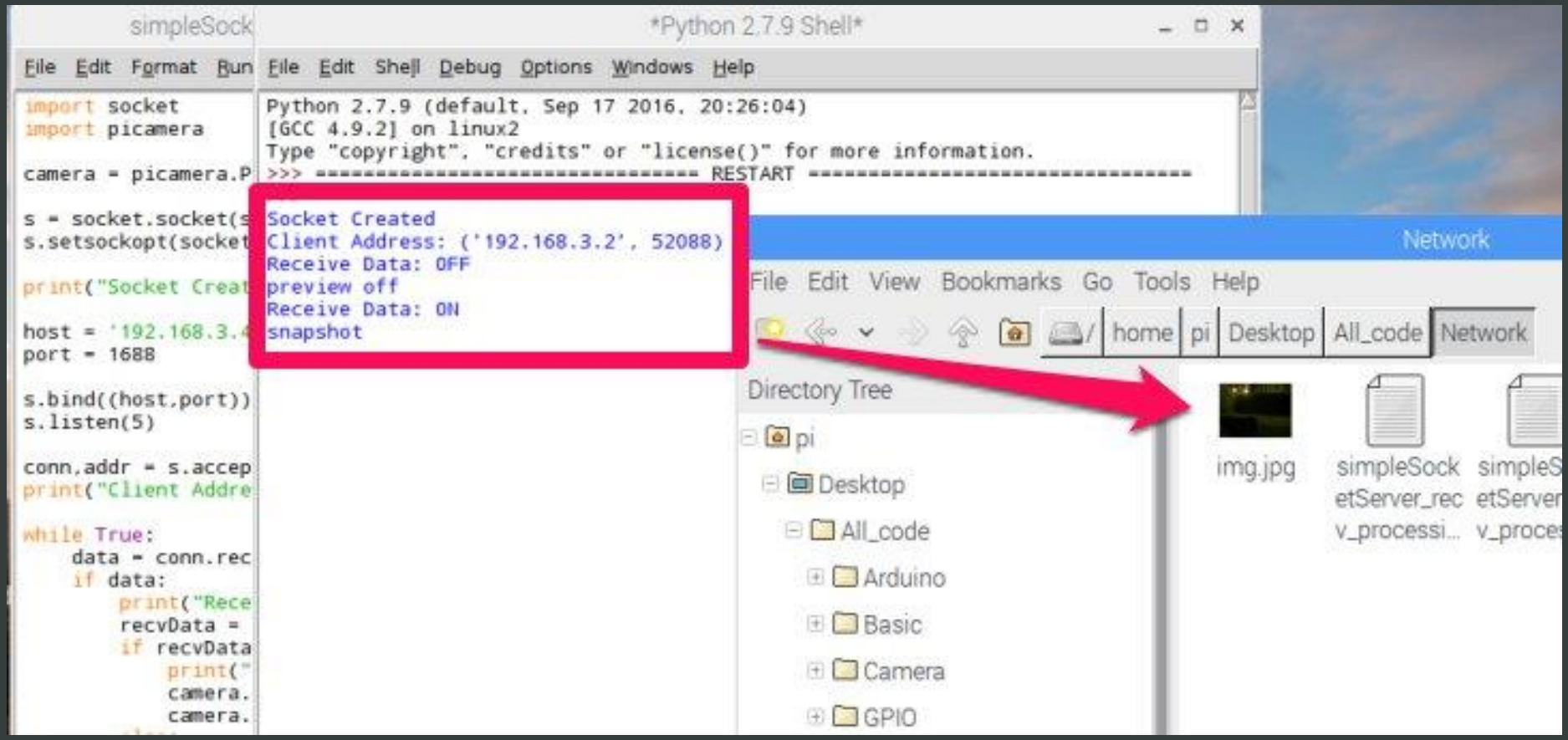
Raspberry + Processing 的 Socket Client Send Data

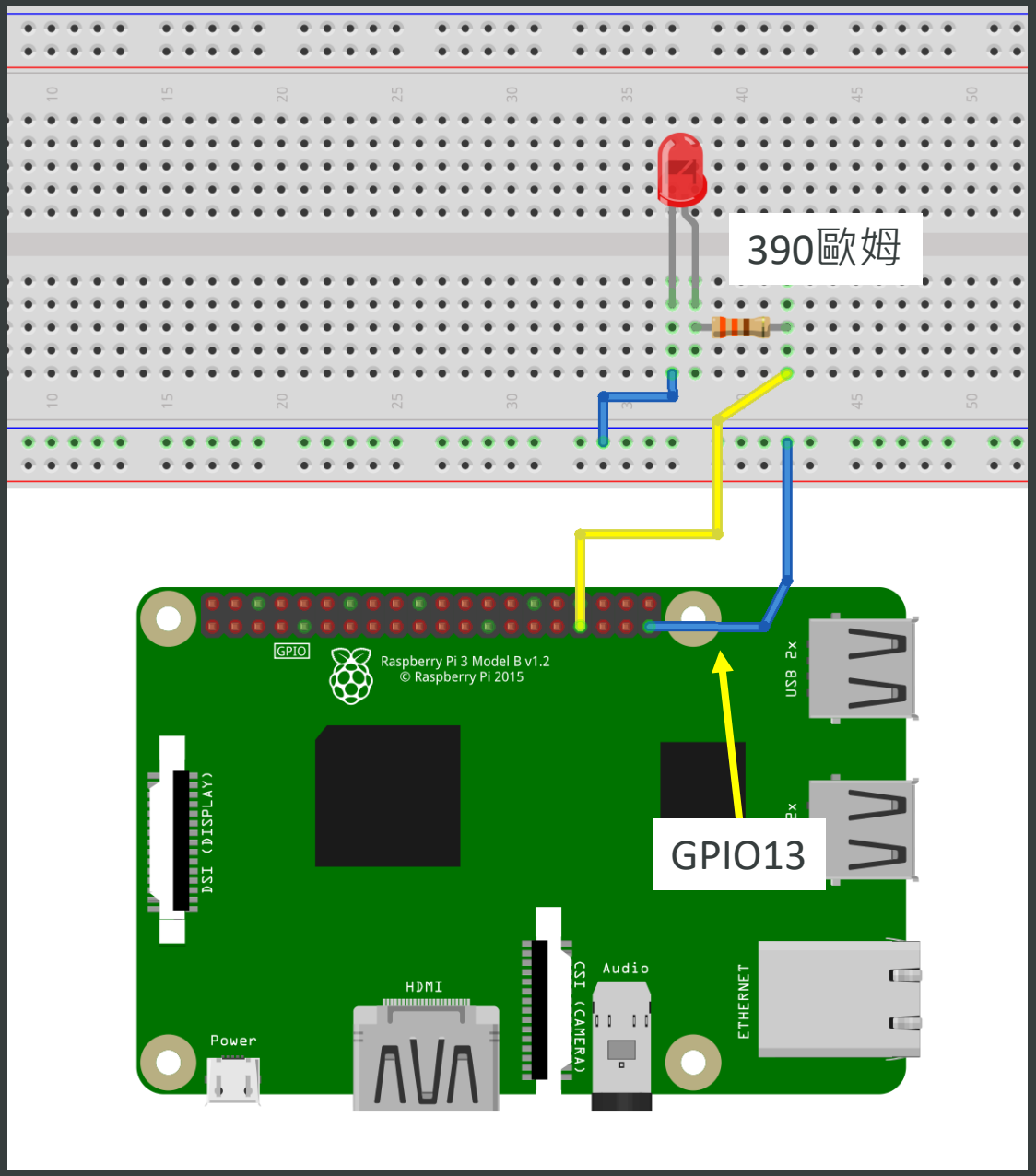
```python
import socket
import picamera

camera = picamera.PiCamera()

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

print("Socket Created")

host = '192.168.3.4'
port = 1688

s.bind((host,port))
s.listen(5)

conn,addr = s.accept()
print("Client Address: %s" % str(addr))

while True:
    data = conn.recv(1024)
    if data:
        print("Receive Data: %s" % str(data.decode('utf-8')))
        recvData = str(data.decode('utf-8'))
        if recvData == "ON":
            print("snapshot")
            camera.capture('img.jpg')
            camera.start_preview()
        else:
            print("preview off")
            camera.stop_preview()
    else:
        break

conn.close()
```



simpleSocketServer_recv_processing_snapshot.py

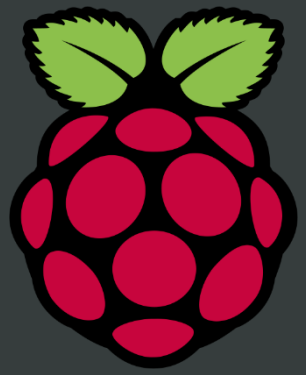
從Processing按下按鈕讓RpiCamera拍攝





	Pin No.		
3.3V	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
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

```python
import socket
import picamera
import RPi.GPIO as GPIO
#setting gpio13
led = 13
GPIO.setmode(GPIO.BCM)
GPIO.setup(led, GPIO.OUT)
#setting camera
camera = picamera.PiCamera()
#setting socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

print("Socket Created")

host = '192.168.3.4'
port = 1688

s.bind((host,port))
s.listen(5)

conn,addr = s.accept()
print("Client Address: %s" % str(addr))

while True:
    data = conn.recv(1024)
    if data:
        print("Receive Data: %s" % str(data.decode('utf-8')))
        recvData = str(data.decode('utf-8'))
        if recvData == "ON":
            print("snapshot")
            camera.capture('img.jpg')
            camera.start_preview()
            GPIO.output(led, True)
        else:
            print("preview off")
            camera.stop_preview()
            GPIO.output(led,False)
    else:
        break

conn.close()
```



simpleSocketServer_recv_processing_gpio.py

Servo

Servo 直接驅動

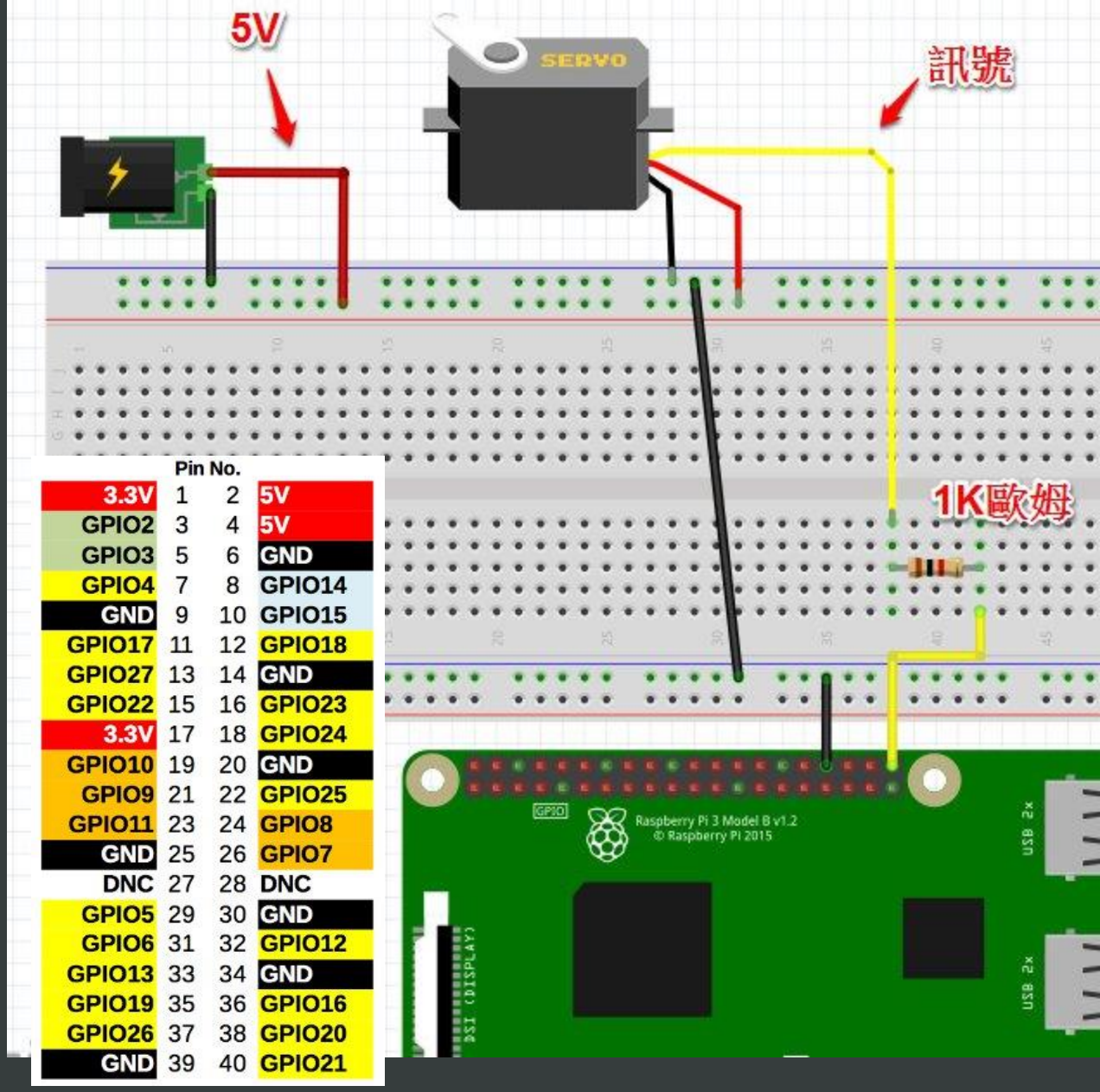
黃色 = 訊號 (橘色,白色,藍色)

紅色 = 5V

黑色 = GND (棕色)

訊號要串接一個1K歐姆電阻

5V可從外部變壓器電源或是接到raspberry上



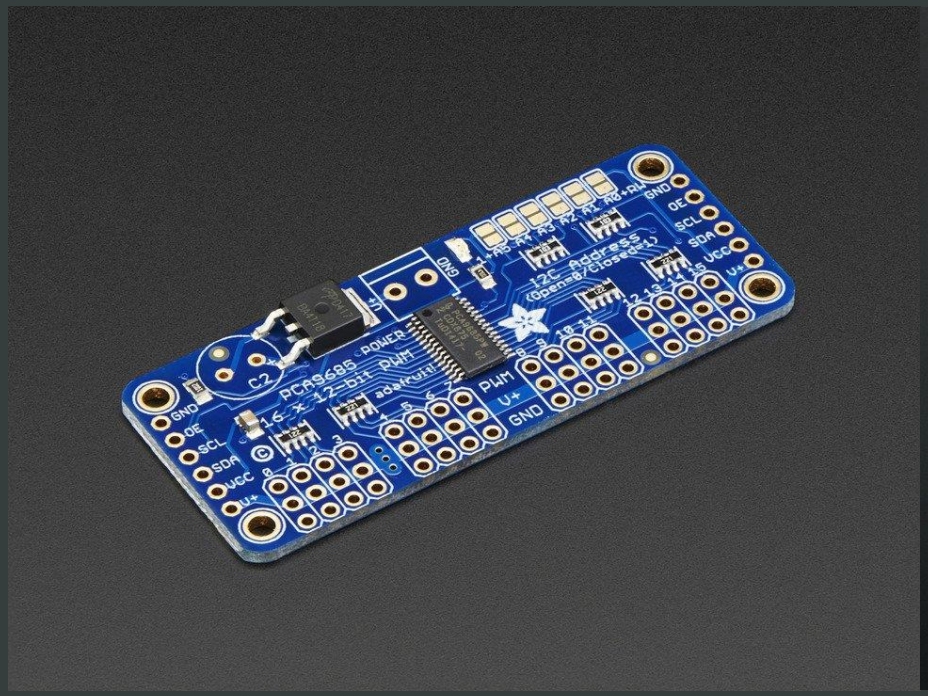
	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

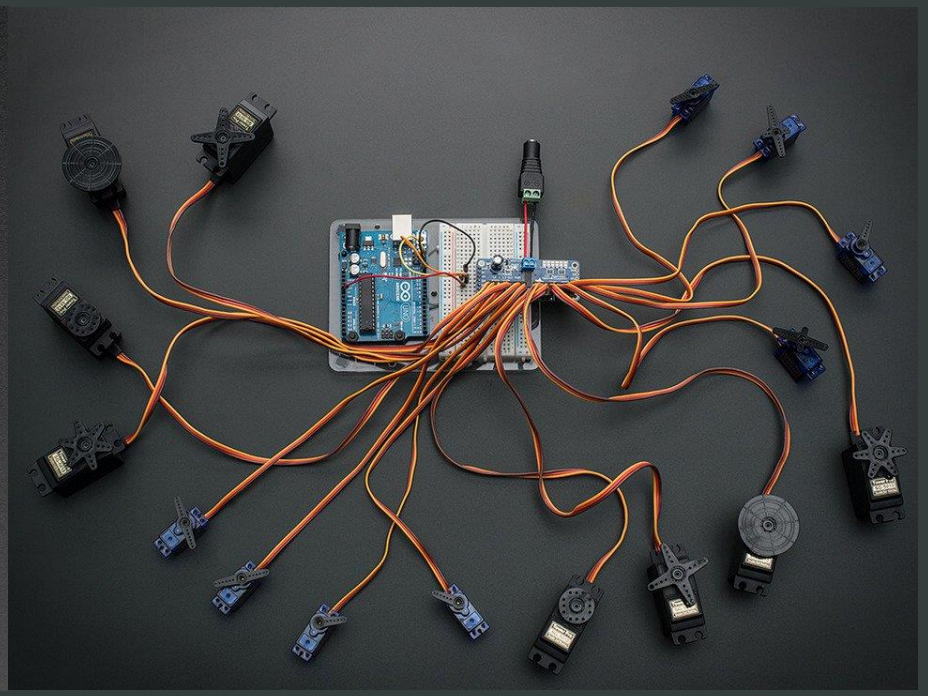
```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

servoPin = 21

GPIO.setup(servoPin, GPIO.OUT)
servo1 = GPIO.PWM(servoPin, 50)
servo1.start(2.5) #0

while True:
    servo1.ChangeDutyCycle(12.5)  #180
    time.sleep(2)
    servo1.ChangeDutyCycle(2.5)   #0
    time.sleep(2)
    servo1.ChangeDutyCycle(7.5)   #90
    time.sleep(2)
```

```
#匯入GPIO模塊
#匯入time模塊


#關閉警告訊息

#宣告要輸出的腳位=21


#設置PWM = 50Hz = 20ms
#設置起始位置 0度


#轉到180度位置

#轉到0度位置

#轉到90度位置
```

Servo_pi.py

PCA9685

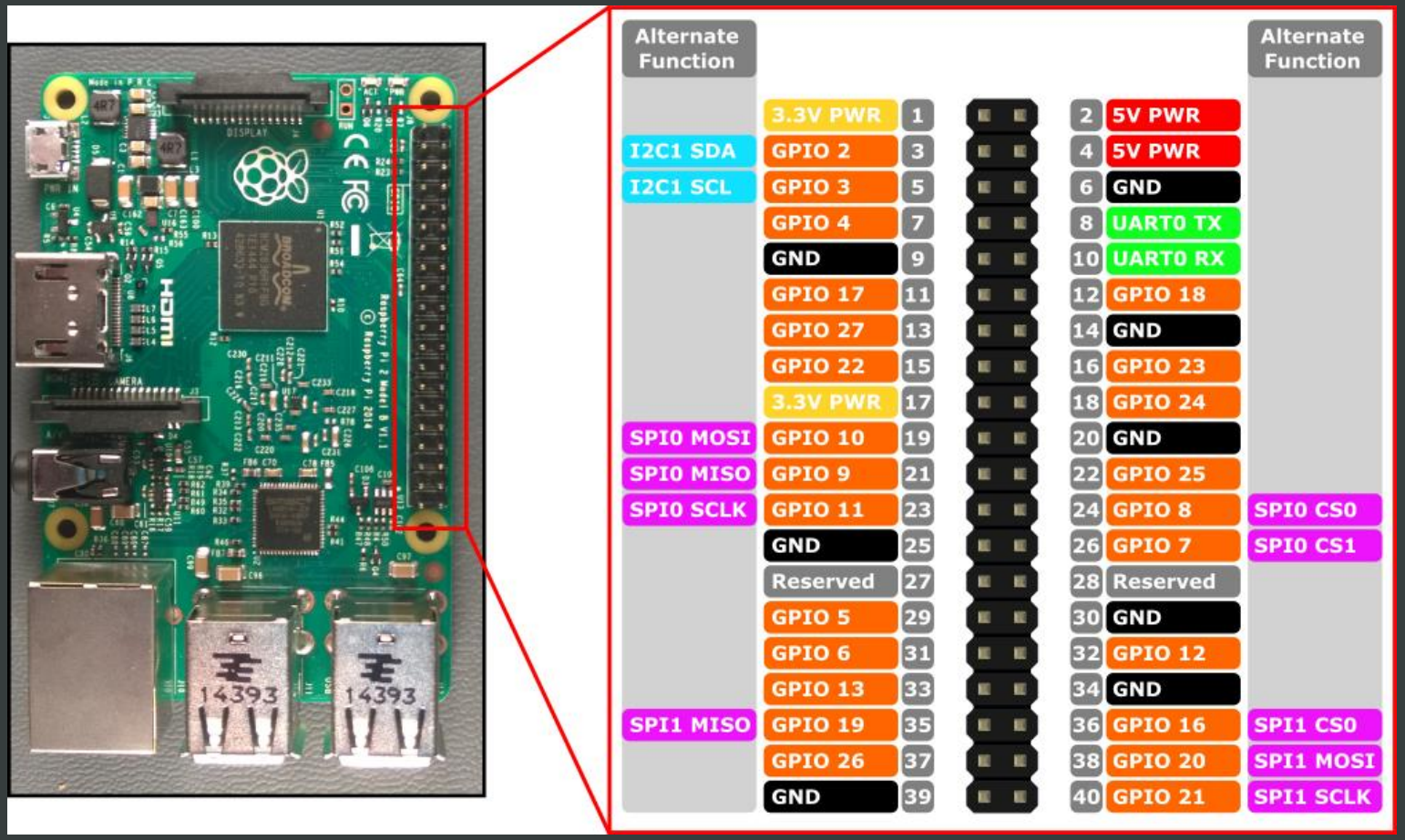




PCA9685

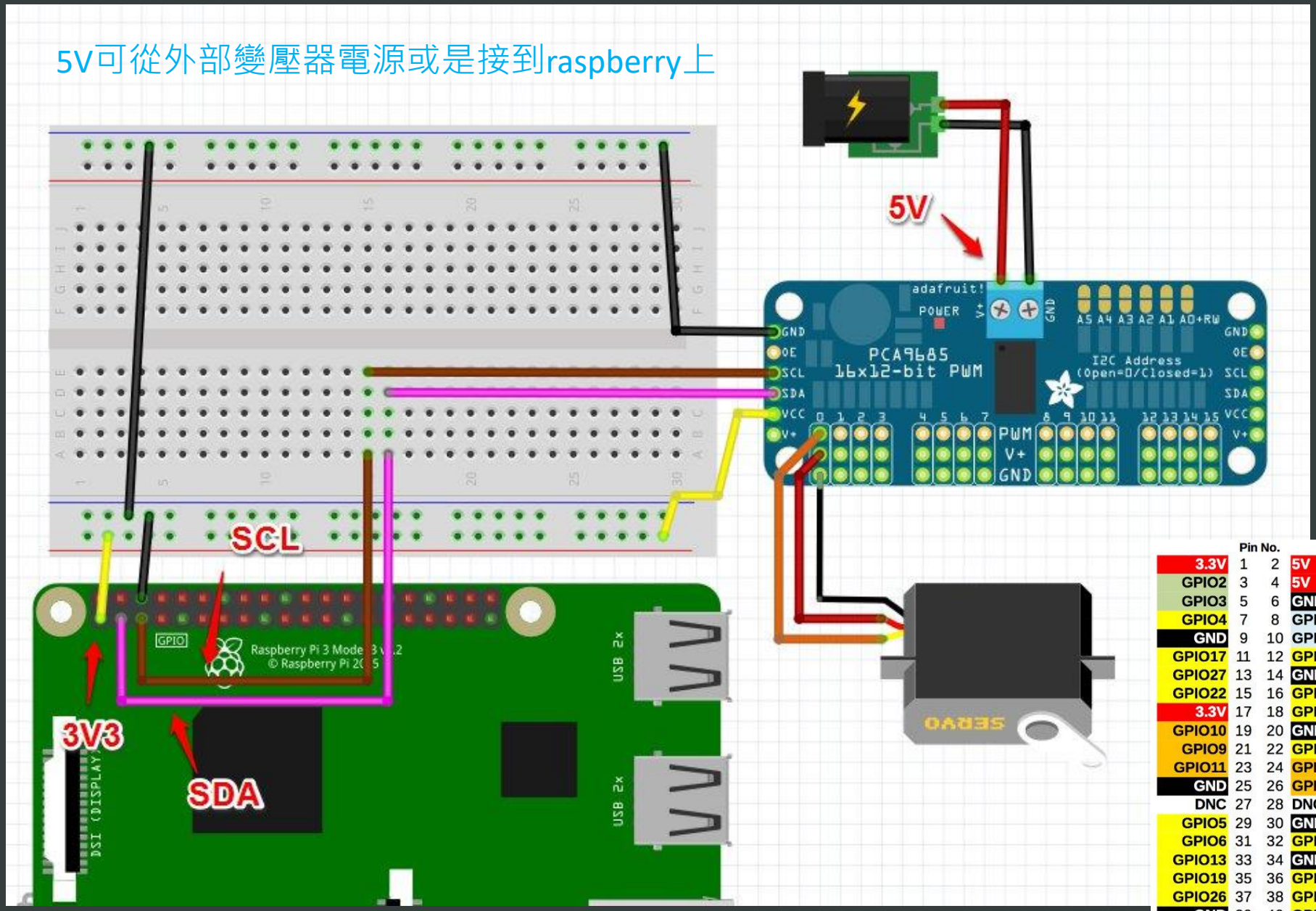
- 12bit PWM Driver = 0~4095 (4096階)
- 16路輸出
- 輸入電壓3-5V
- 輸出邏輯電壓3-5V
- 可程式化PWM頻率40~1KHz
- 額外電壓電流輸入端子
- I2C通訊界面

連接i2C訊號線路



I2C SDA = GPIO 2
I2C SCL = GPIO 3

5V可從外部變壓器電源或是接到raspberry上



	Pin No.		
3.3V	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
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

安裝PCA9685模組程式庫

Step1 – 下載Adafruit_Python_PCA9685程式庫

```
$ cd ~
$ git clone https://github.com/adafruit/Adafruit_Python_PCA9685.git
```

Step2 – 安裝

```
$ cd Adafruit_Python_PCA9685
$ sudo python setup.py install
```

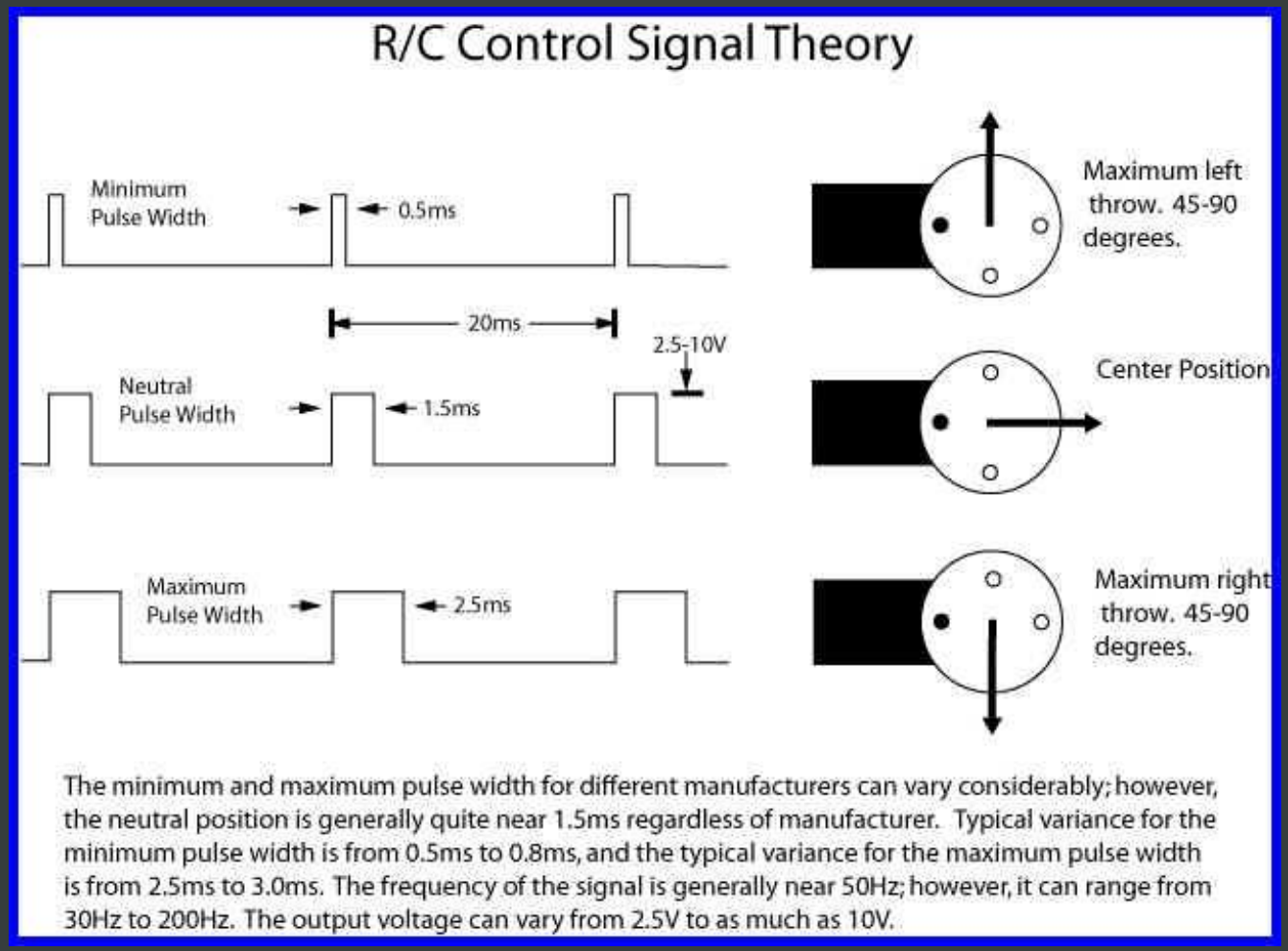
```
import Adafruit_PCA9685
import time

pwm = Adafruit_PCA9685.PCA9685()

pwm.set_pwm_freq(50)

while True:
    #Channel 0~15
    pwm.set_pwm(0, 0, 104)
    time.sleep(1)
    pwm.set_pwm(0, 0, 520)
    time.sleep(1)
```

```
#匯入Adafruit_PCA9685模塊
#匯入time模塊

#建立pwm物件

#設定PWM頻率為50Hz=20毫秒
 (範圍40~1K)


#設置Ch0, +90度

#設置Ch0，-90度
```

SimplePCA9685.py

Adafruit Python PC9685

- set_pwm_freq(frequency)
  - 範圍 40Hz ~ 1000Hz
  - 設置PWM頻率
- set_pwm(channel, on, off)
  - Channel = 要驅動的通道 0 ~ 15
  - On = 由Low到High的訊號 0~4095
  - Off = 由High到Low的訊號 0~4095
- PWM範例
  - 假設頻率為50Hz => 每1Hz 花費20毫秒 (1000毫秒 / 50Hz)
  - 每一階的PWM時間為 20 / 4096 = 0.0048毫秒
  - 若要控制1000階的一個PWM = 4.8毫秒
  - 若要2.5毫秒的PWM需 2.5/0.0048 = 520 同理0.5毫秒的PWM需 0.5/0.0048 = 104



當週期為50Hz = 20ms時
0.5ms 的脈波寬度 = +90度
1.5ms 的脈波寬度 = 0度
2.5ms 的脈波寬度 = -90度

0.5ms = 104
1.5ms = 312
2.5ms = 520

1度角 = 2.88階PWM寬度

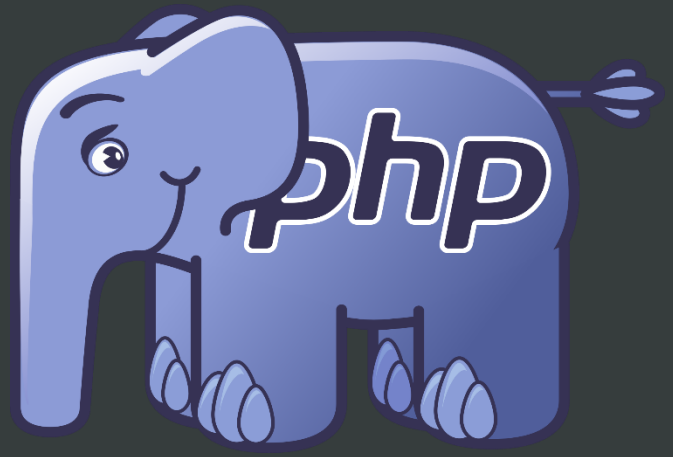
```
import Adafruit_PCA9685
import time

pwm =  Adafruit_PCA9685.PCA9685()
pwm.set_pwm_freq(50)

def set_servo_angle(channel, angle):
    data = angle * 3
    pwm.set_pwm(channel, 0, data)

while True:
    #Ch0
    for i in range(0, 181):
        set_servo_angle(0, i)
        time.sleep(0.025)

    for i in range(181, 0, -1):
        set_servo_angle(0, i)
        time.sleep(0.025)
```

```
#匯入Adafruit_PCA9685模塊
#匯入time模塊

#建立pwm物件
#設定PWM頻率為50Hz=20毫秒

#自訂set_servo_angle函式
#傳入的角度*2.88 四捨五入*3
#設定channel與pwm



#循環角度從0~180旋轉



#反向從0~180旋轉
```

SimplePCA9685_angleToPWM.py

LAMP(Linux, Apache, MySQL, PHP)

- Linux系統 -> Raspbian 已經安裝完畢
- Apache -> 網頁伺服器(WebServer)，負責處理來自瀏覽器的http網頁請求
- MySQL -> 資料庫(Database)，負責新增、刪除、修改、變更資料內容
- PHP -> 動態網頁程式，動態顯示資料庫內的內容到網頁上



安裝Apache

出現無法安裝請
sudo apt-get update

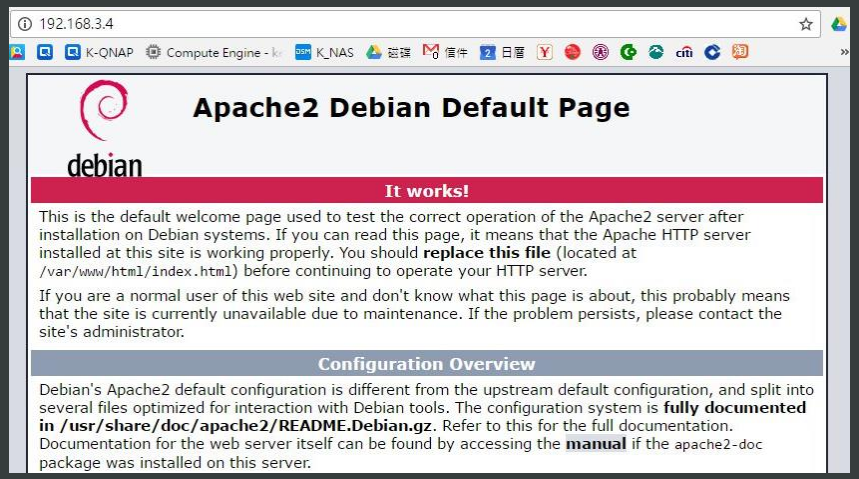
Step1 – 安裝相關套件

```
$ sudo apt-get install apache2
```

中途碰到[Y/n]都輸入Y

```
pi@raspberrypi_testbed:~ $ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom
  openssl-blacklist
```

Step2 – 測試Web Server
打開瀏覽器輸入http://192.168.3.x



安裝PHP

Step3 – 安裝PHP5

```
$ sudo apt-get install php5 libapache2-mod-php5
```

中途碰到[Y/n]都輸入Y

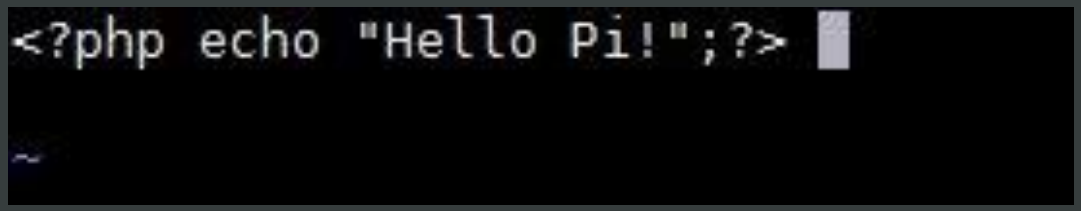
Step4 – 新增index.php

```
$ sudo vim /var/www/html/index.php
```

Step5 – 在index.php內輸入以下程式

```
<?php echo “Hello Pi !”;?>
```

儲存好之後離開



測試PHP

Step6 – 打開瀏覽器輸入以下網址

http://192.168.3.x/index.php

安裝MySQL

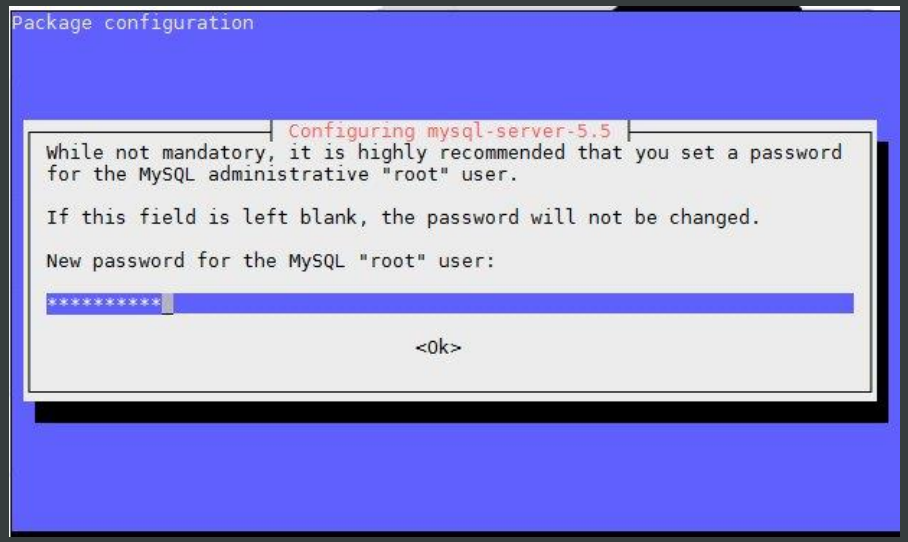
Step7 – 安裝MySQL

```
$ sudo apt-get install mysql-server php5-mysql
```

中途碰到[Y/n]都輸入Y

Step8 – 修改MySQL的root使用者密碼

輸入自己定義的密碼，例如：1234，然後按下Enter之後再次重複輸入密碼



安裝PhpMyAdmin

Step9 – 安裝phpmyadmin

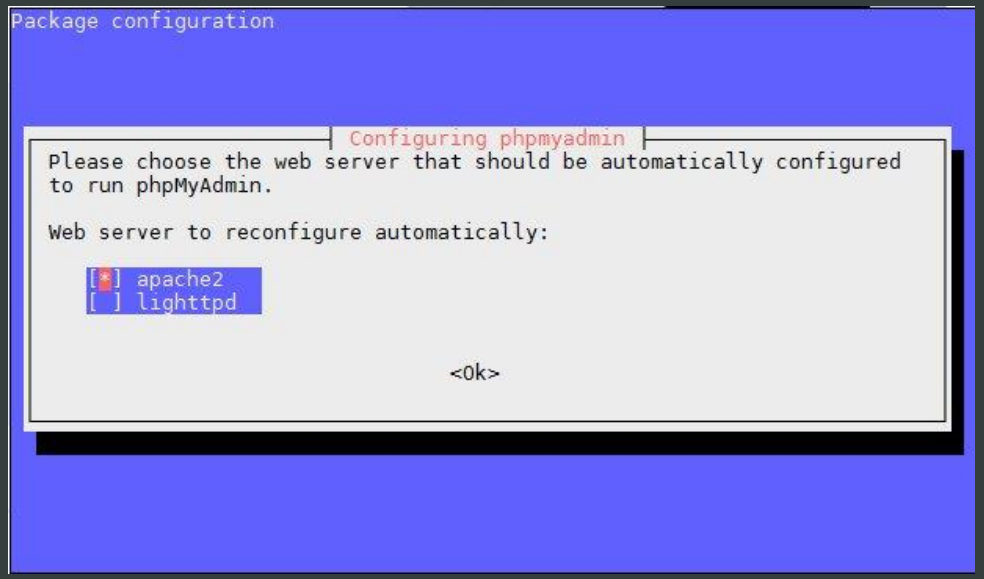
```
$ sudo apt-get install phpmyadmin
```

中途碰到[Y/n]都輸入Y

Ste10 – 設定phpmyadmin

選擇apache2按下空白鍵後會在該選項出現*號

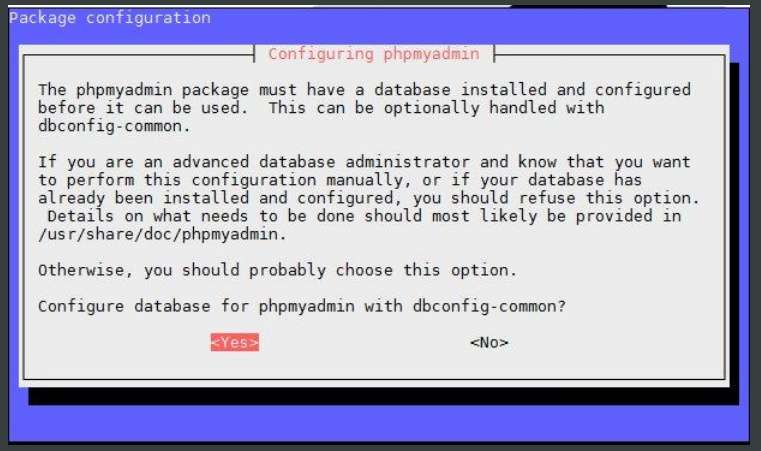
之後按下Enter



設定PhpMyAdmin

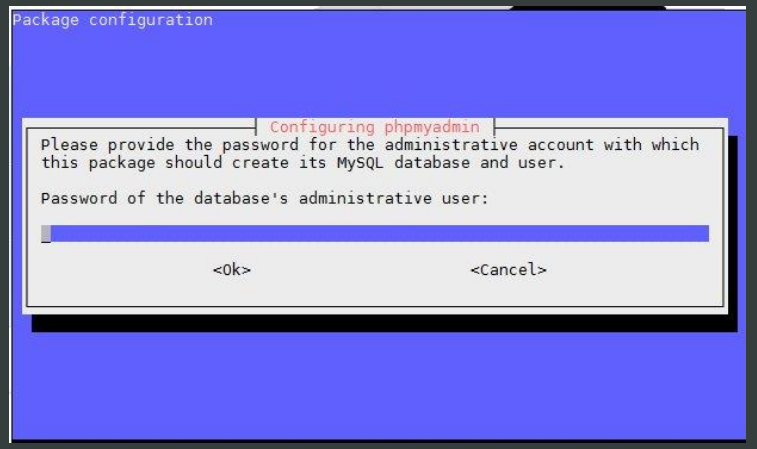
Step11 – 設定phpmyadmin

選擇<YES>按下Enter鍵繼續設定



Ste12 – 預設的root使用者密碼

輸入剛剛設定的root使用者密碼作為登入Phpmyadmin的帳號密碼，總共會輸入三次



測試MySQL使用PhpMyAdmin

Step13 – 打開瀏覽器測試

打開瀏覽器輸入http://192.168.3.x/phpmyadmin

Step14 – 輸入帳號密碼

使用者名稱: root

密碼: 剛剛輸入的root密碼

回家作業

- 利用Socket方式，透過電腦端控制樹梅派上的servo進行轉動，電腦端可用processing, python, C, C++, Java語言等等。並上傳照片+影片+程式碼到iLMS
- 在樹梅派上安裝wordpress，並讓樹梅派可透過**遠端存取，發布一篇文章內容寫上組員名字跟組別，以及改變佈景主題，將遠端存取的網址跟畫面擷取下來繳交到iLMS。

**遠端存取方式需固定IP，若無固定IP請利用校內網路或實驗室網路，有開啟80埠的即可遠端存取。