通訊網路實驗

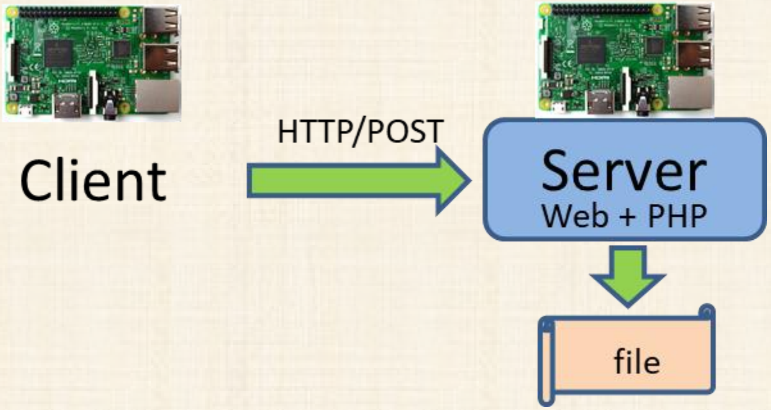
IoT應用
RaspPI Networking

Dept. of Electrical and Computer Engineering (ECE)

National Chiao Tung University

課程大綱

- 1.架設web server
  - 提供php範例, 利用HTTP/POST送資料至server端



- 2.透過藍牙傳送資料

Demo項目

- Q1 : Sensor 1 & 2同時傳兩筆資料,修改php 使server分別存成不同檔案

- Q2： 兩人為一組，Client透過HTTP/POST傳送資料給Server

- Q3： 兩人為一組，Client透過藍芽傳送資料給Server

Introduction

- Web Server:
  - 軟體(LAMP)
    - Linux – Raspbian
    - Apache – webserver (http) software
    - MySQL – database server
    - PHP or Perl or Python – Web Programming

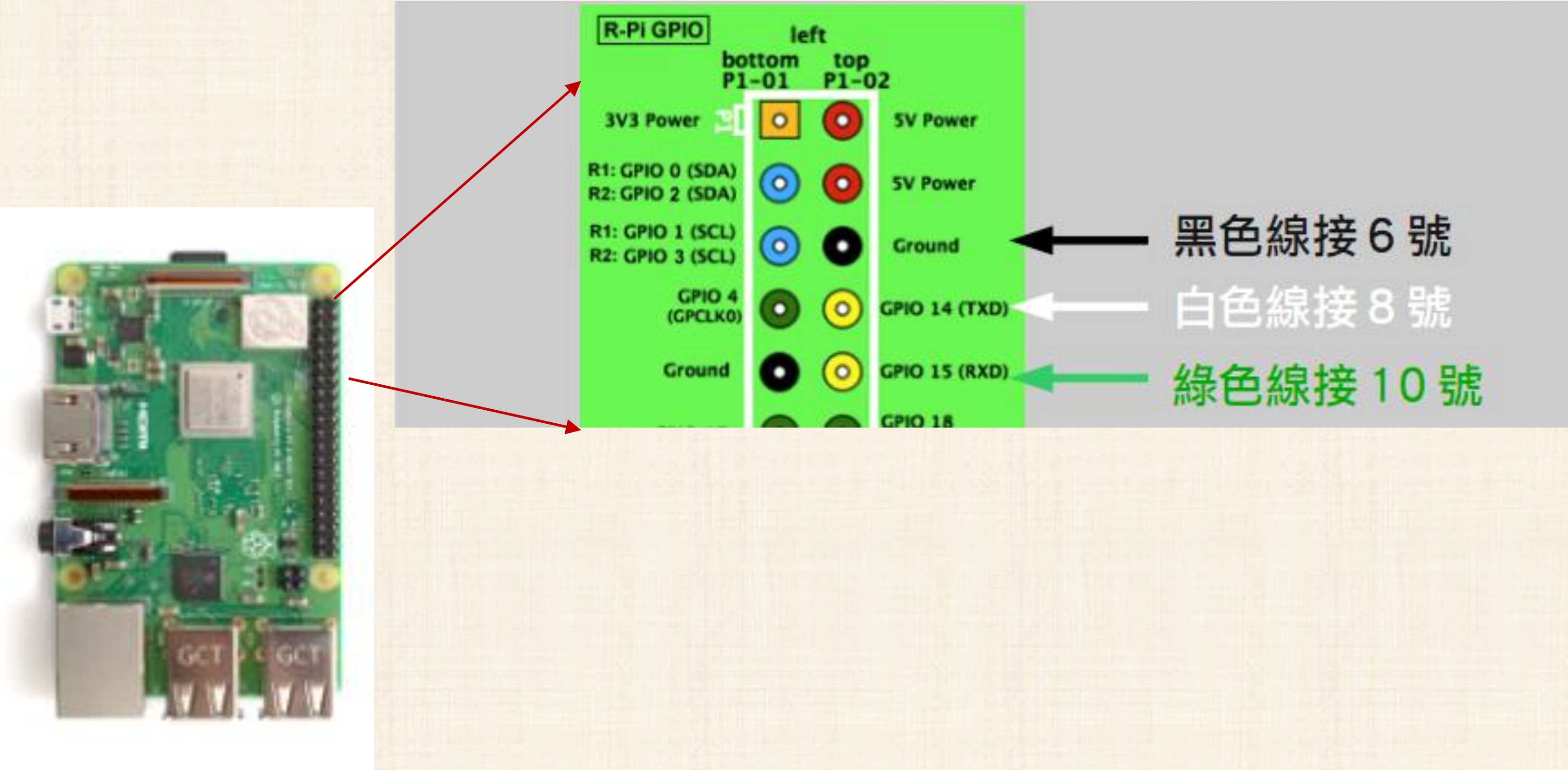
Introduction

- Web server Software (http服務):
  - Apache
  - Nginx (發音同engine x)
  - Microsoft - Internet Information Server (IIS)
  - AWS - Amazon Web Services (AWS)

溫馨小提醒

- 預設登入帳密
  - ID: pi
  - PW: raspberry

注意接角



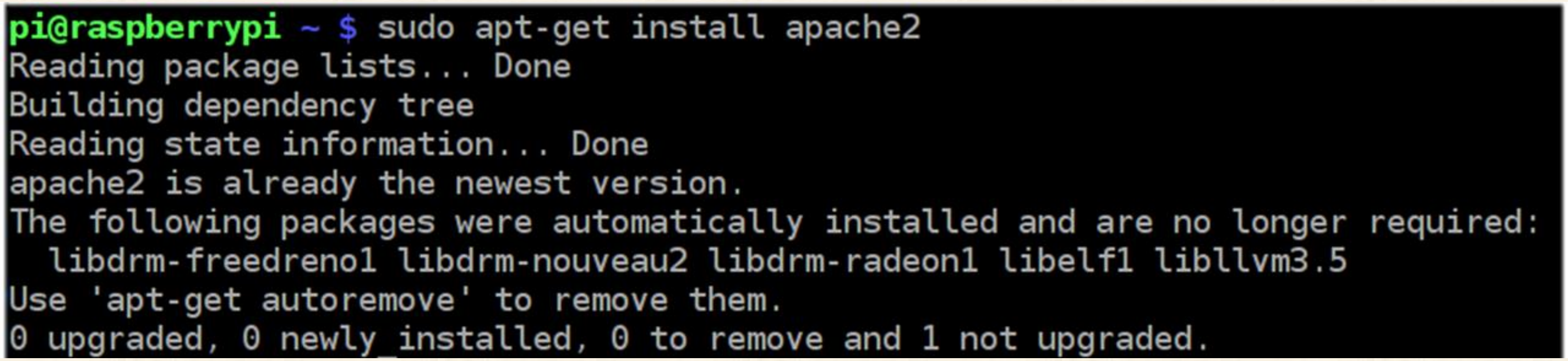
黑色線接 6 號

白色線接 8 號

綠色線接 10 號

建立 Web Server 網站

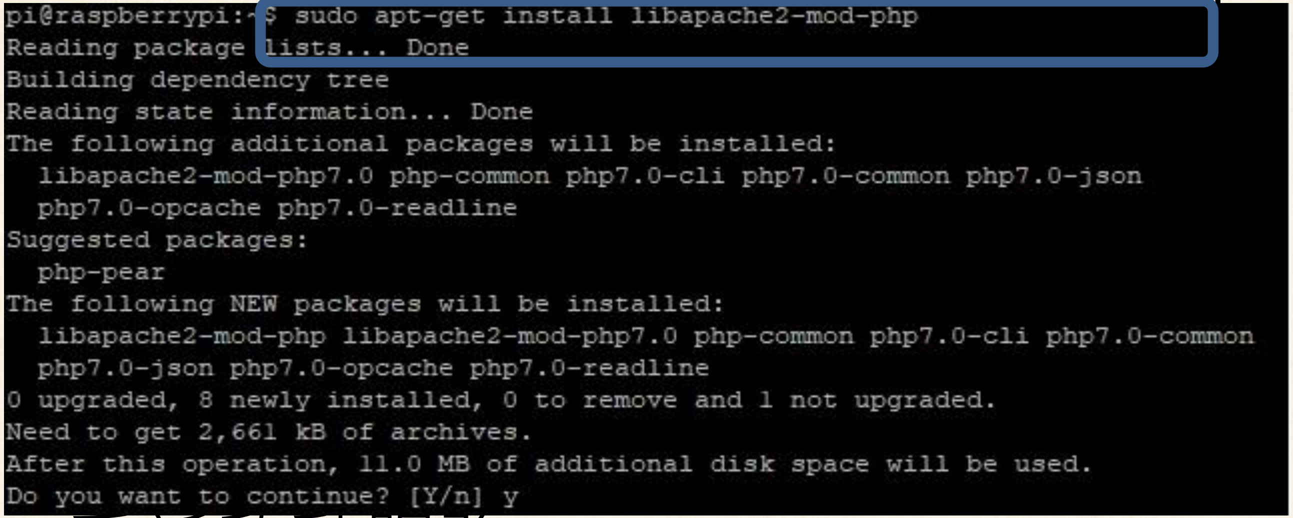
- 安裝網頁伺服器apache2:
  - $ sudo apt-get install apache2



```
pi@raspberrypi ~ $ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version.
The following packages were automatically installed and are no longer required:
  libdrm-freedreno1 libdrm-nouveau2 libdrm-radeon1 libelf1 libllvm3.5
Use 'apt-get autoremove' to remove them.
0 upgraded, 0 newly_installed, 0 to remove and 1 not upgraded.
```

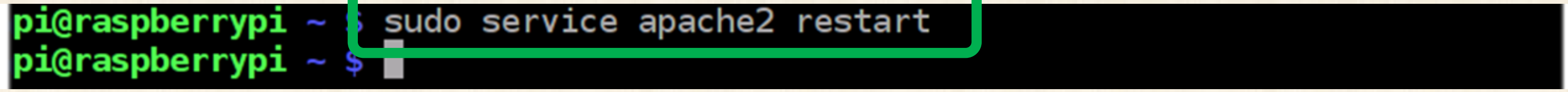
建立 Web Server 網站

- 安裝PHP程式語言:
  - $ sudo apt-get install libapache2-mod-php



```
pi@raspberrypi:~$ sudo apt-get install libapache2-mod-php
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.0 php-common php7.0-cli php7.0-common php7.0-json
  php7.0-opcache php7.0-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php7.0 php-common php7.0-cli php7.0-common
  php7.0-json php7.0-opcache php7.0-readline
0 upgraded, 8 newly installed, 0 to remove and 1 not upgraded.
Need to get 2,661 kB of archives.
After this operation, 11.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

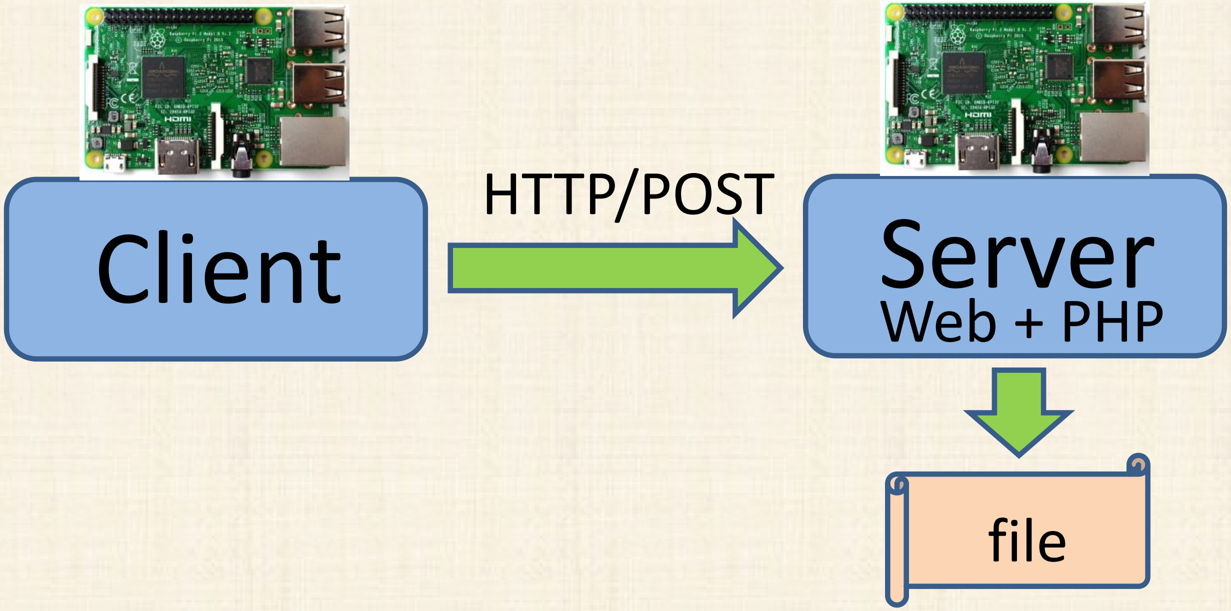
- 重新啟動服務 (Web Server):
  - $ sudo service apache2 restart



```
pi@raspberrypi ~ $ sudo service apache2 restart
pi@raspberrypi ~ $ 
```

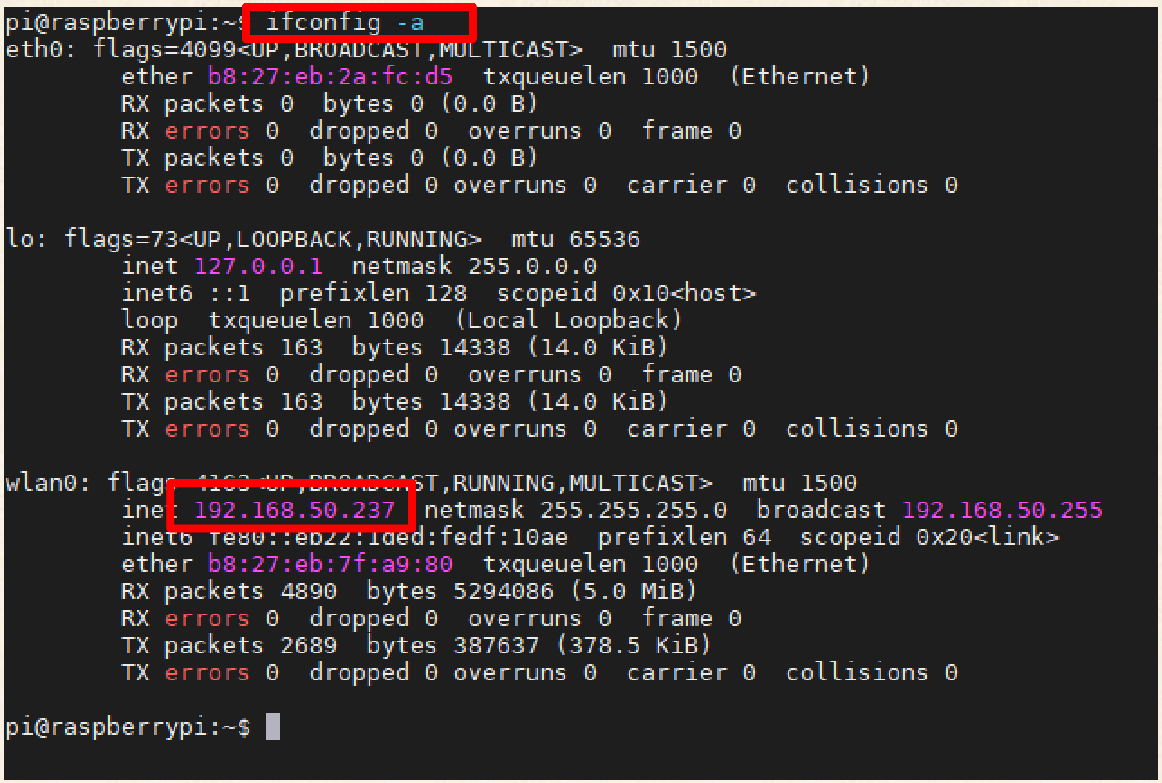
Web Server & HTTP/POST

- php範例, 利用HTTP/POST傳送資料並存檔



Web Server & HTTP/POST

- 查看自己的IP是多少，並記住
  - ifconfig –a



```
pi@raspberrypi:~$ ifconfig -a
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:2a:fc:d5  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 163  bytes 14338 (14.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 163  bytes 14338 (14.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.50.237  netmask 255.255.255.0  broadcast 192.168.50.255
        inet6 fe80::eb22:1ded:fedf:10ae  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:7f:a9:80  txqueuelen 1000  (Ethernet)
        RX packets 4890  bytes 5294086 (5.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2689  bytes 387637 (378.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

pi@raspberrypi:~$ 
```

Web Server & HTTP/POST

- 1.先在根目錄建立資料夾 (與PHP裡面的路徑一致)
  - cd ~
  - mkdir www-data
- 2.設定該資料夾權限,使web server具有存取能力
  - sudo chown www-data:pi www-data/
- 3.進入網頁根目錄
  - cd /var/www/html
- 4.新建index.php ，內容如下頁
  - sudo nano index.php

Web Server & HTTP/POST

- 在網頁根目錄放置一個 index.php 的檔案, 內容如下
  - 網頁根目錄 /var/www/html/

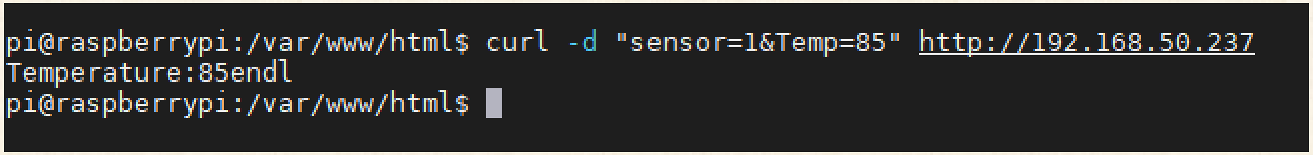
```
<?php
            header("Content-Type:text/html; charset=utf-8");
            /* Temp -> server */
            // ex: curl -d "sensor=1&Temp28.9" http://IP_addr
            $Temperature=$_POST[Temp];
            $SensorID=$_POST[sensor];

            echo 'Temperature:'.$Temperature. endl;

            if ($SensorID==1) {
            $fp = fopen('/home/pi/www-data/temp.txt', 'w');
            fwrite($fp, $Temperature);
            fclose($fp);
}
?>
```

送來的Temp與Sensor值,
存到Temperature與SensorID

當SensorID為1
將Temperature數值存成該檔案

Web Server & HTTP/POST

- 自己傳送post資料給自己
  - curl -d "sensor=1&Temp=28.9" http://**yourIP**



```
pi@raspberrypi:/var/www/html$ curl -d "sensor=1&Temp=85" http://192.168.50.237
Temperature:85endl
pi@raspberrypi:/var/www/html$ 
```

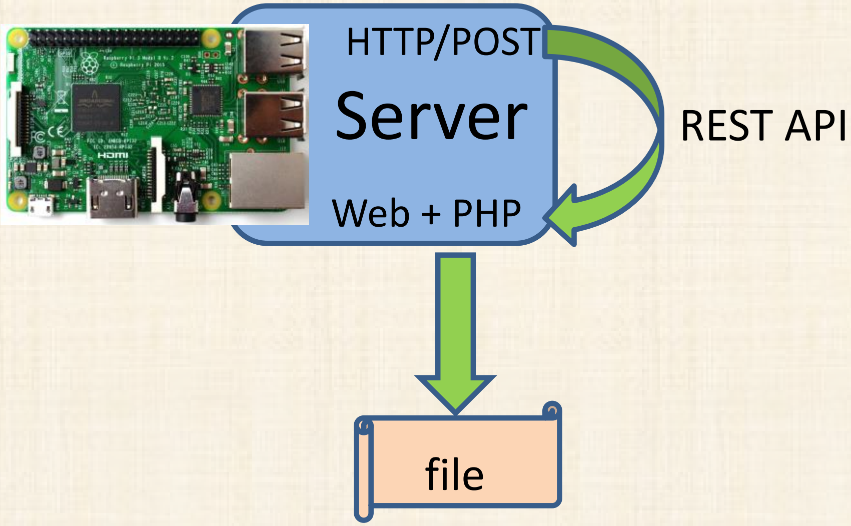
如果沒印出Temperature:85endl
先檢查code有沒有打錯

如果是出現一大堆字
在 var/www/html 下

- sudo rm index.php
- sudo rm index.html

重建一個新的

- sudo nano index.php



HTTP/POST

Server

Web + PHP

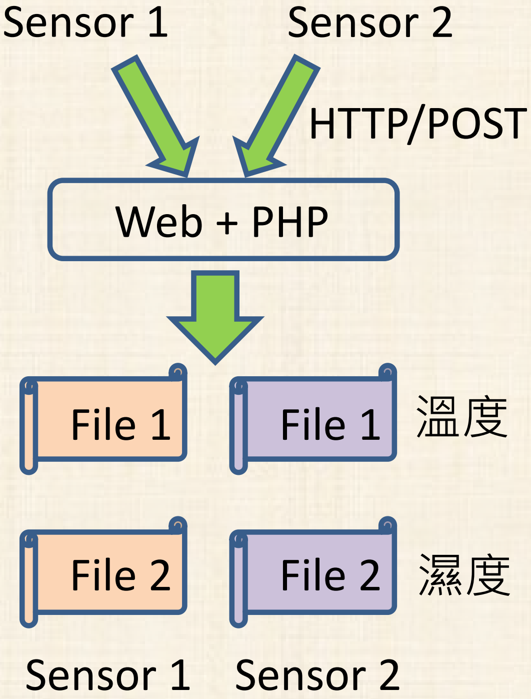
REST API

file

Q1

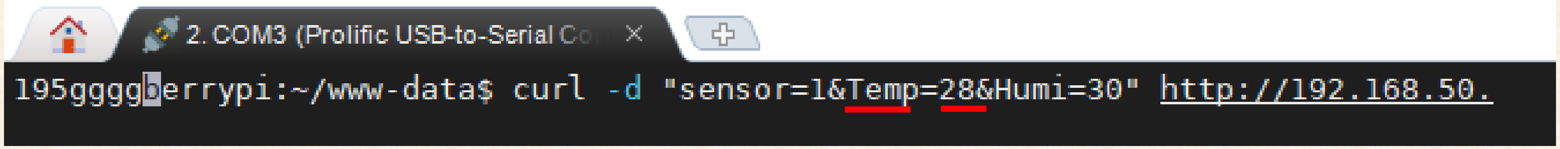
- □ 同時有兩個sensor會傳送2筆資料, 修改php使server將資料存成不同檔案
  - ◘ Ex: sensor 1 & 2同時傳送溫濕度資訊

- □ 需有4個檔案



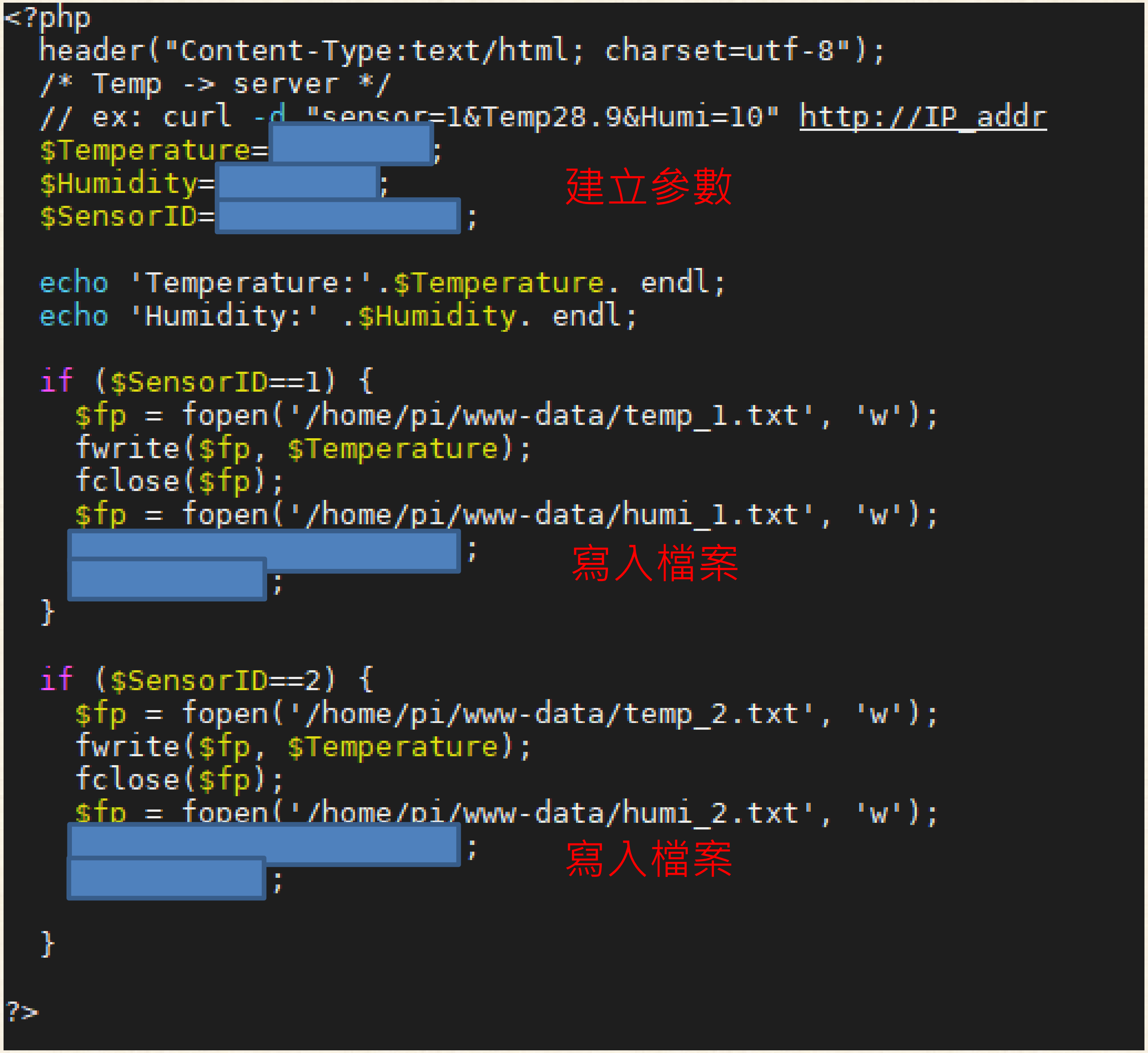
!!注意:在輸入curl指令時，如果指令太長
可能會出現錯誤

可改變參數命名及數值，建議用一個字母即可



```
195ggggberrypi:~/www-data$ curl -d "sensor=1&Temp=28&Humi=30" http://192.168.50.
```

- □ Ctrl + C 中斷重新輸入即可



```php
<?php
  header("Content-Type:text/html; charset=utf-8");
  /* Temp -> server */
  // ex: curl -d "sensor=1&Temp28.9&Humi=10" http://IP_addr
  $Temperature=[redacted];
  $Humidity=[redacted];
  $SensorID=[redacted];

  echo 'Temperature:'.$Temperature. endl;
  echo 'Humidity:' .$Humidity. endl;

  if ($SensorID==1) {
    $fp = fopen('/home/pi/www-data/temp_1.txt', 'w');
    fwrite($fp, $Temperature);
    fclose($fp);
    $fp = fopen('/home/pi/www-data/humi_1.txt', 'w');
    [redacted];
    [redacted];
  }

  if ($SensorID==2) {
    $fp = fopen('/home/pi/www-data/temp_2.txt', 'w');
    fwrite($fp, $Temperature);
    fclose($fp);
    $fp = fopen('/home/pi/www-data/humi_2.txt', 'w');
    [redacted];
    [redacted];

  }

?>
```

建立參數

寫入檔案

寫入檔案

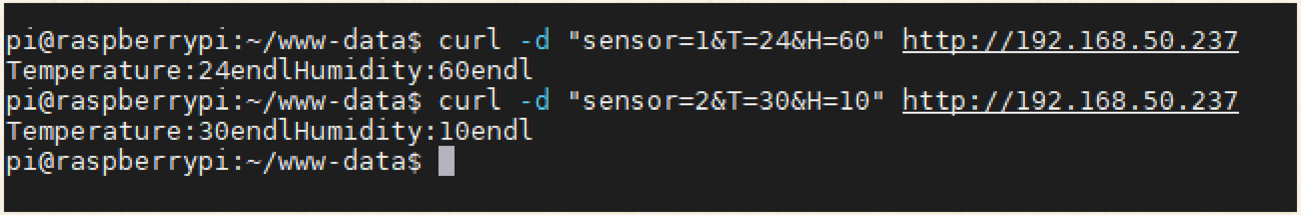
下頁有文字可以複製

```
<?php
 header("Content-Type:text/html; charset=utf-8");
 /* Temp -> server */
 // ex: curl -d "sensor=1&Temp28.9&Humi=10" http://IP_addr
 $Temperature=    ;
 $Humidity=    ;
 $SensorID=   ;

 echo 'Temperature:'.$Temperature. endl;
 echo 'Humidity:' .$Humidity. endl;

 if ($SensorID==1) {
  $fp = fopen('/home/pi/www-data/temp_1.txt', 'w');
  fwrite($fp, $Temperature);
  fclose($fp);
  $fp = fopen('/home/pi/www-data/humi_1.txt', 'w');
  ;
  ;
 }

 if ($SensorID==2) {
  $fp = fopen('/home/pi/www-data/temp_2.txt', 'w');
  fwrite($fp, $Temperature);
  fclose($fp);
  $fp = fopen('/home/pi/www-data/humi_2.txt', 'w');
  ;
  ;
 }

?>
```

空白處
記得修改

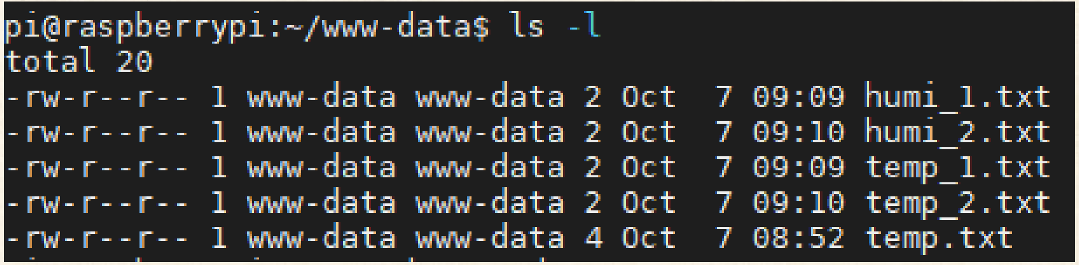
成功範例

- 傳送成功



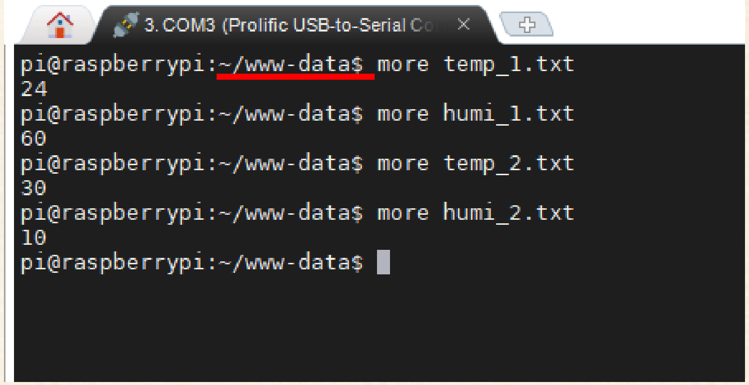
```
pi@raspberrypi:~/www-data$ curl -d "sensor=1&T=24&H=60" http://192.168.50.237
Temperature:24endlHumidity:60endl
pi@raspberrypi:~/www-data$ curl -d "sensor=2&T=30&H=10" http://192.168.50.237
Temperature:30endlHumidity:10endl
pi@raspberrypi:~/www-data$ 
```

- 輸入 ls –l 可查看傳送紀錄



```
pi@raspberrypi:~/www-data$ ls -l
total 20
-rw-r--r-- 1 www-data www-data 2 Oct  7 09:09 humi_1.txt
-rw-r--r-- 1 www-data www-data 2 Oct  7 09:10 humi_2.txt
-rw-r--r-- 1 www-data www-data 2 Oct  7 09:09 temp_1.txt
-rw-r--r-- 1 www-data www-data 2 Oct  7 09:10 temp_2.txt
-rw-r--r-- 1 www-data www-data 4 Oct  7 08:52 temp.txt
```

- 輸入more + 檔名.副檔名 可查看檔案內容

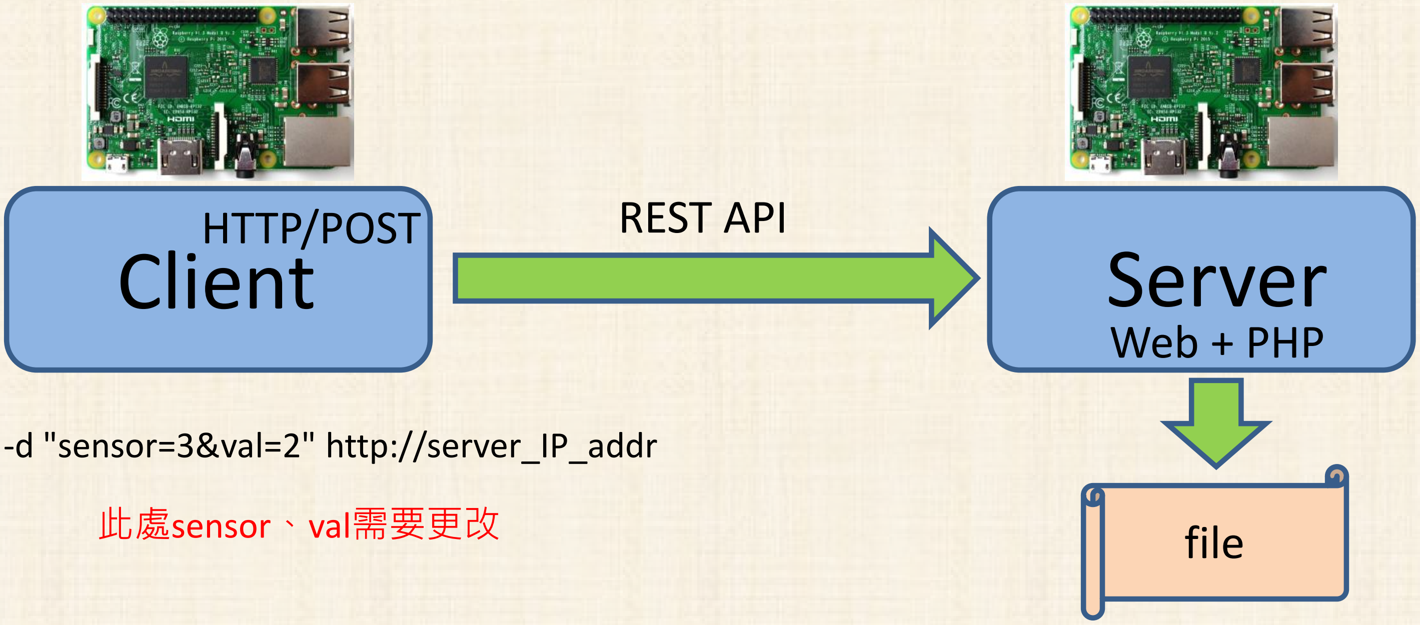


```
pi@raspberrypi:~/www-data$ more temp_1.txt
24
pi@raspberrypi:~/www-data$ more humi_1.txt
60
pi@raspberrypi:~/www-data$ more temp_2.txt
30
pi@raspberrypi:~/www-data$ more humi_2.txt
10
pi@raspberrypi:~/www-data$ 
```

要在fopen打的資料夾底下
才有檔案 (cd ~/www-data)

Q2

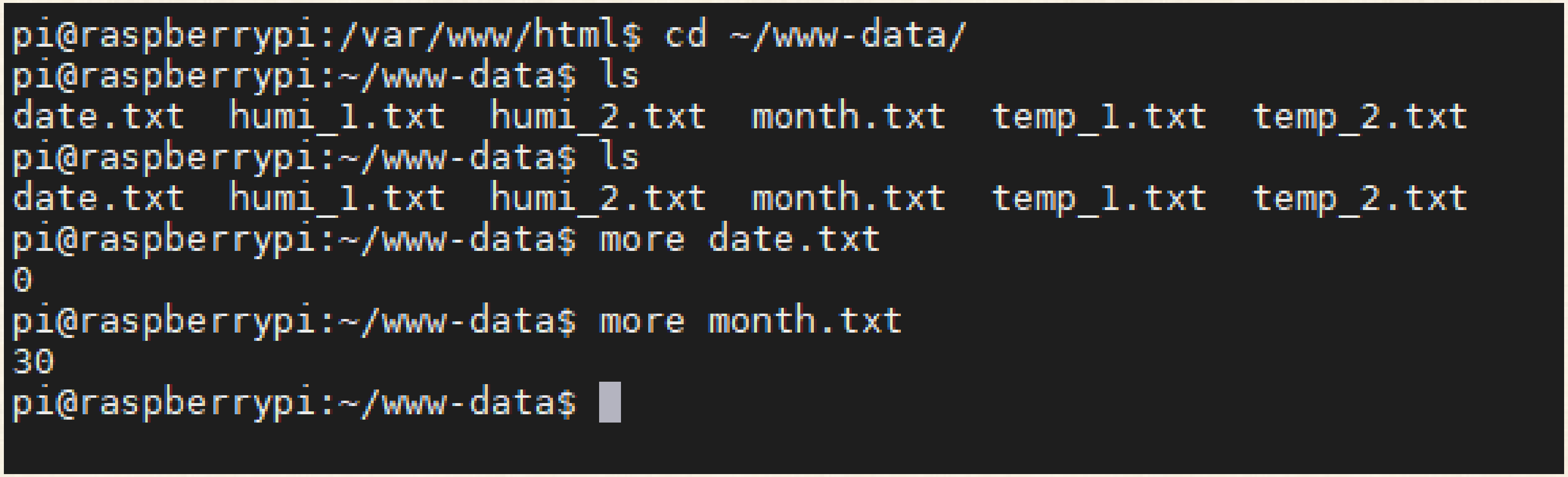
- 使用兩個PI, 其中 Client 傳送資料 至 Server
  - 一人當client, 一人當server
  - Server需開啟檔案確認有成功接收到2筆資料 (月份,日期)
  - 檔案命名為 month.txt 、date.txt



```
curl -d "sensor=3&val=2" http://server_IP_addr
```

此處sensor、val需要更改

成功範例



```
pi@raspberrypi:/var/www/html$ cd ~/www-data/
pi@raspberrypi:~/www-data$ ls
date.txt  humi_1.txt  humi_2.txt  month.txt  temp_1.txt  temp_2.txt
pi@raspberrypi:~/www-data$ ls
date.txt  humi_1.txt  humi_2.txt  month.txt  temp_1.txt  temp_2.txt
pi@raspberrypi:~/www-data$ more date.txt
0
pi@raspberrypi:~/www-data$ more month.txt
30
pi@raspberrypi:~/www-data$ 
```

藍牙 Bluetooth

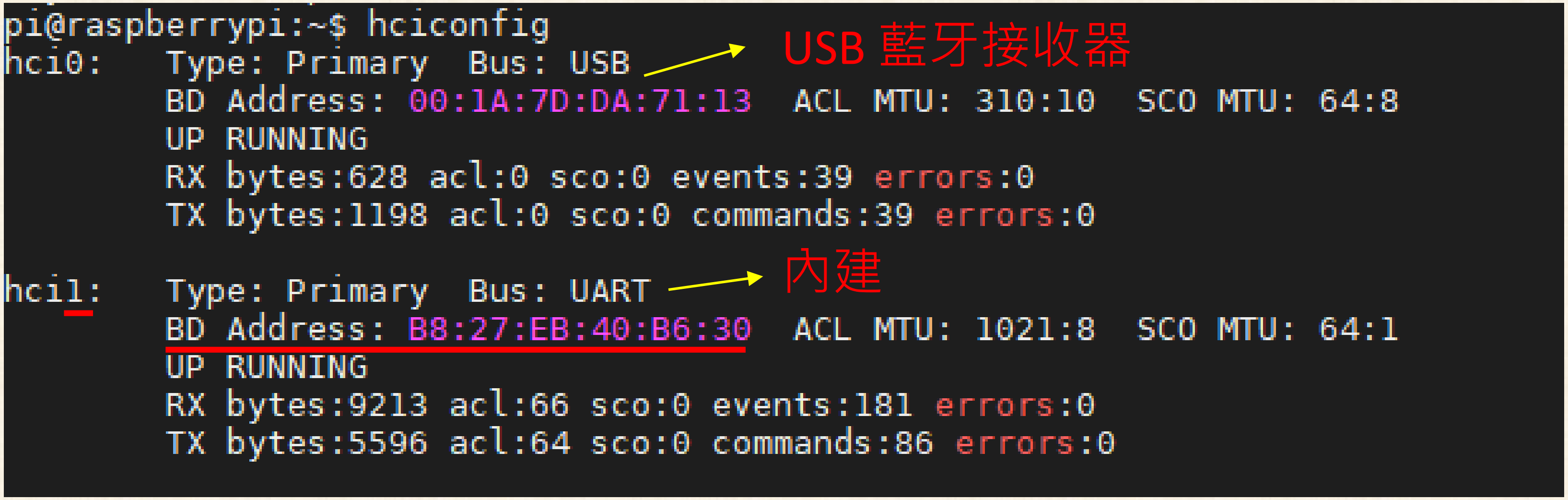
- 目的
  - 為了解決電腦與電器設備之間的傳輸問題
- 特色
  - 短距離無線技術 (10 - 100m)
  - 使用 2.4 至 2.485 GHz 的 ISM 頻段
- Bluetooth Classic: 802.15
- Bluetooth 4.0 Low Energy (BLE): 802.15.1
- Bluetooth 5.0: Faster, Further, for IoT

https://zh.wikipedia.org/zh-tw/%E8%97%8D%E7%89%99

Python Bluetooth

- 先安裝Bluetooth的套件
  - cd ~
  - sudo apt-get install python-bluez

輸入hciconfig看自己拿到的pi版使否有內建藍牙
優先使用內建的



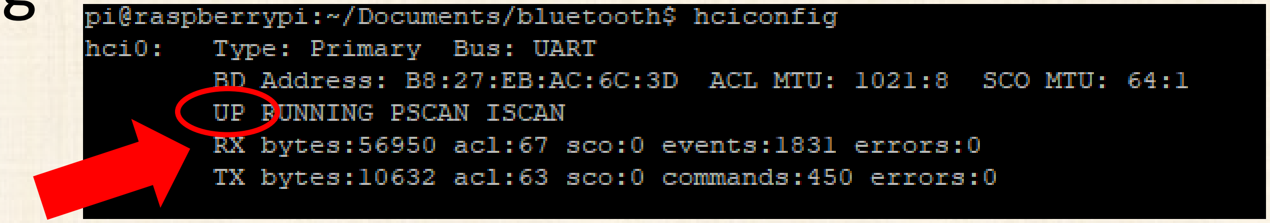
```
pi@raspberrypi:~$ hciconfig
hci0:   Type: Primary  Bus: USB
        BD Address: 00:1A:7D:DA:71:13  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING
        RX bytes:628 acl:0 sco:0 events:39 errors:0
        TX bytes:1198 acl:0 sco:0 commands:39 errors:0

hci1:   Type: Primary  Bus: UART
        BD Address: B8:27:EB:40:B6:30  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING
        RX bytes:9213 acl:66 sco:0 events:181 errors:0
        TX bytes:5596 acl:64 sco:0 commands:86 errors:0
```

USB 藍牙接收器

內建

Python Bluetooth

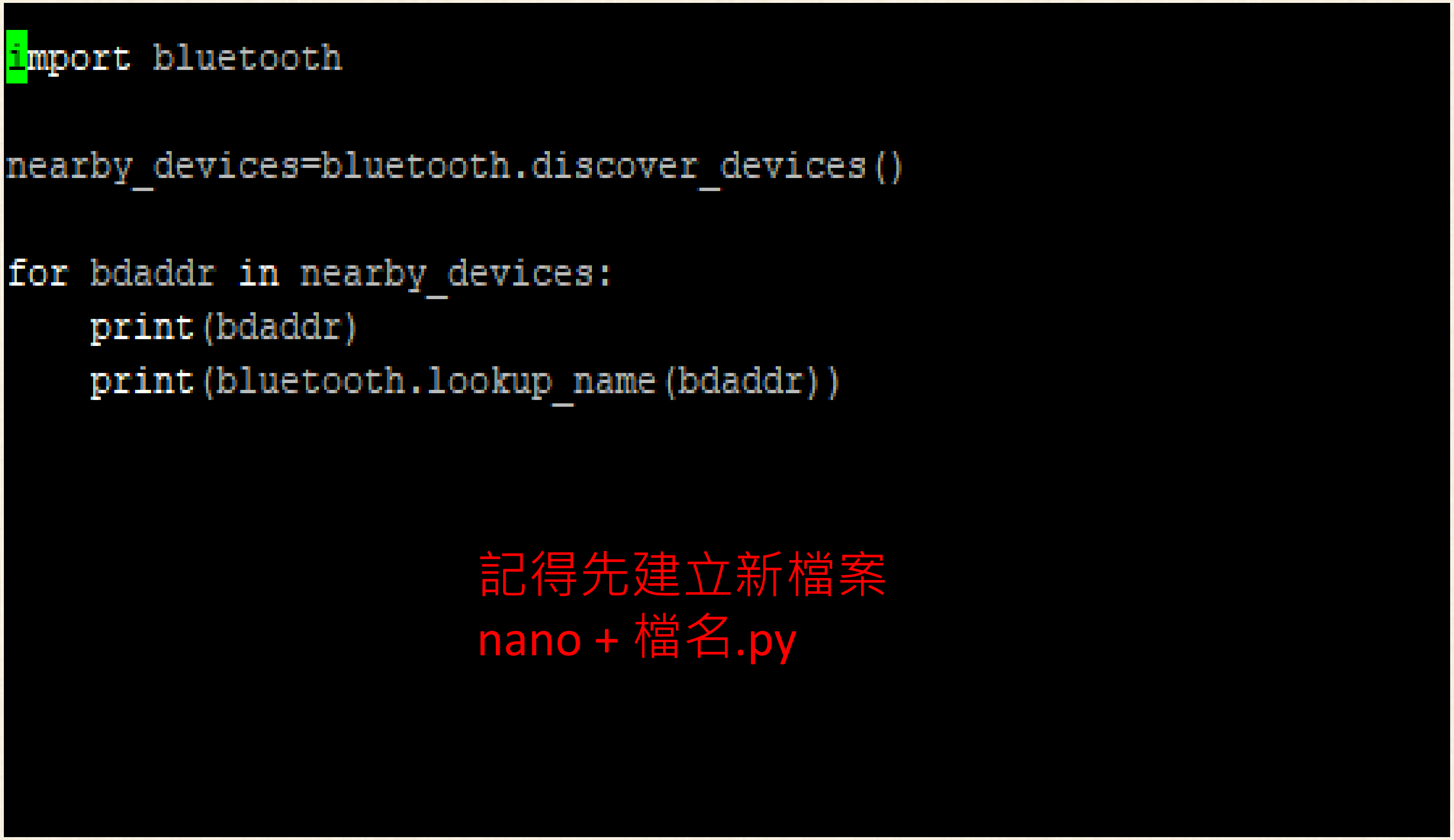
- 使裝置可被搜尋
  - sudo hciconfig hci0 piscan
- 更改裝置名稱
  - sudo hciconfig hci0 name “ your name ”
- 確認藍牙裝置接收器已被啟動
  - hciconfig



```
pi@raspberrypi:~/Documents/bluetooth$ hciconfig
hci0:   Type: Primary  Bus: UART
        BD Address: B8:27:EB:AC:6C:3D  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING PSCAN ISCAN
        RX bytes:56950 acl:67 sco:0 events:1831 errors:0
        TX bytes:10632 acl:63 sco:0 commands:450 errors:0
```

搜尋附近藍牙裝置

參考；運行程式後可開啟手機藍牙功能看附近的裝置是否相同

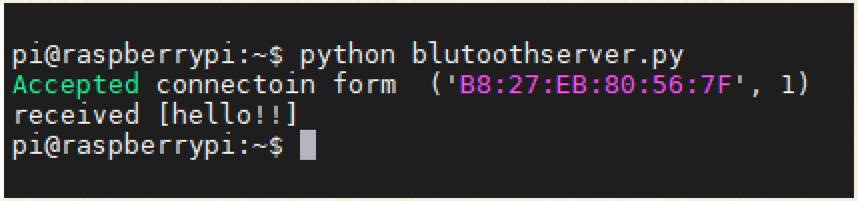


```
import bluetooth

nearby_devices=bluetooth.discover_devices()

for bdaddr in nearby_devices:
    print(bdaddr)
    print(bluetooth.lookup_name(bdaddr))
```

記得先建立新檔案
nano + 檔名.py

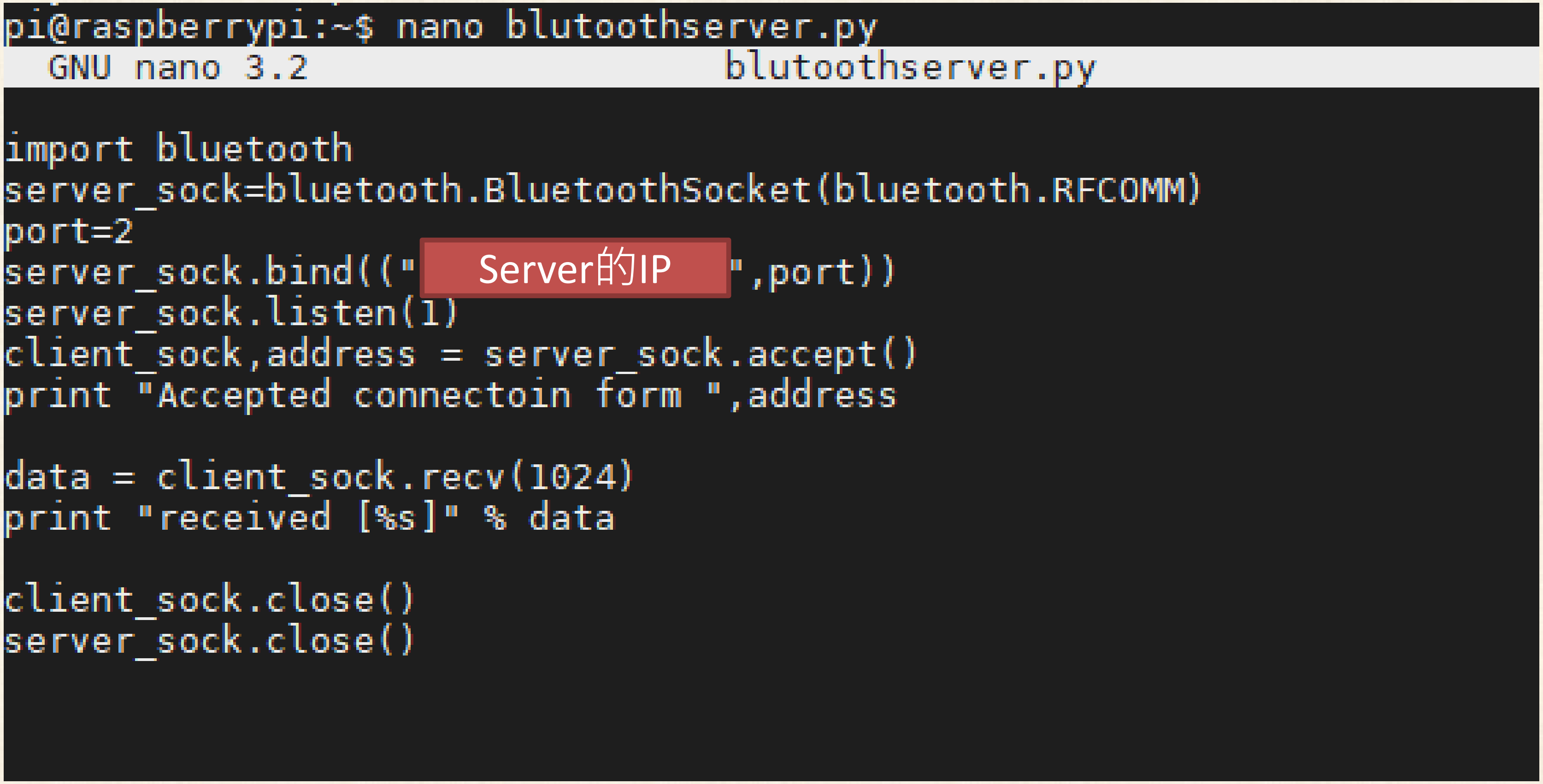
Q3

- □ 開啟一個bluetooth的server和client，client傳送 client_學號 & server_學號 給server



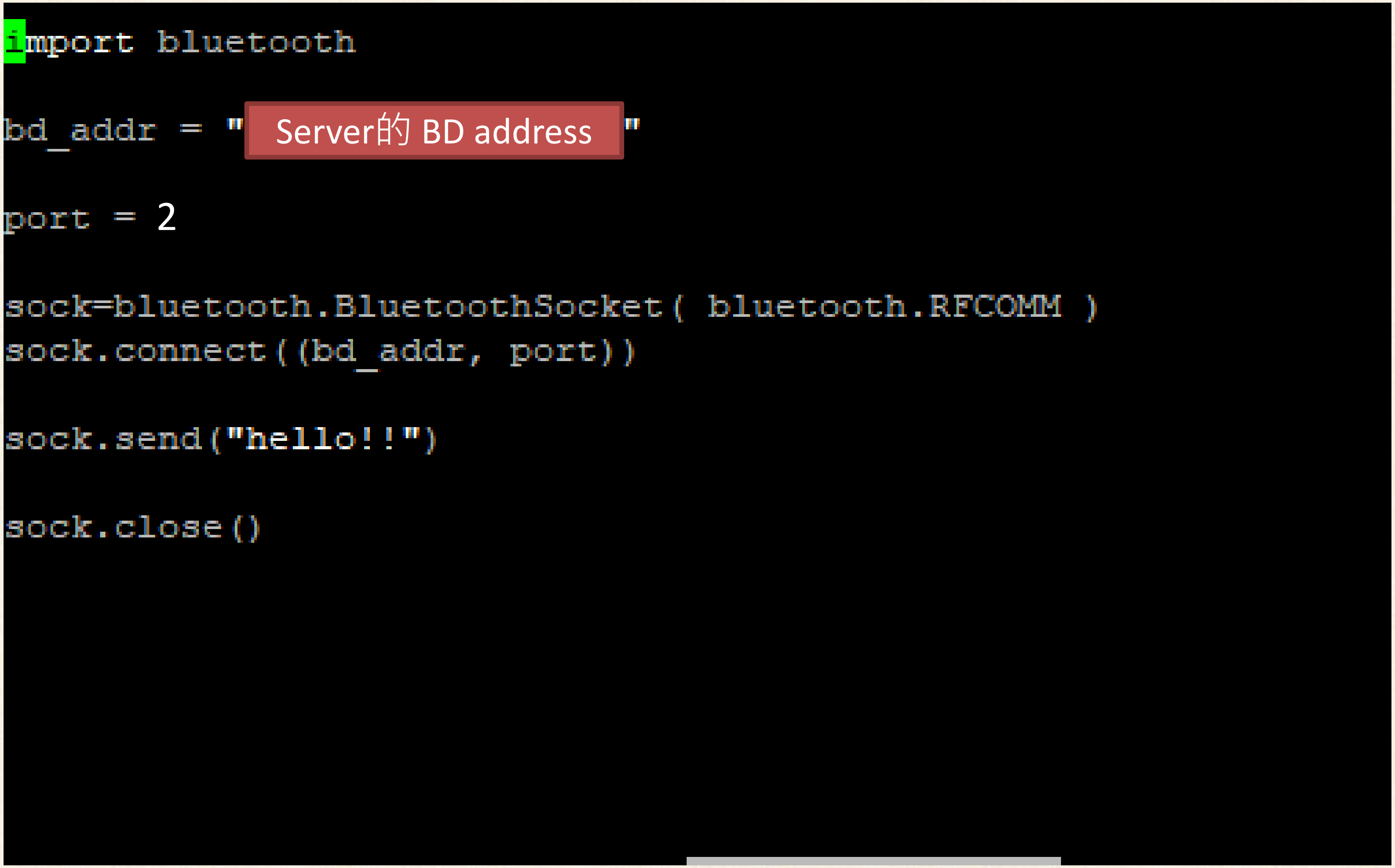
```
pi@raspberrypi:~$ python blutoothserver.py
Accepted connectoin form  ('B8:27:EB:80:56:7F', 1)
received [hello!!]
pi@raspberrypi:~$ 
```

Bluetooth Server 範例



```
pi@raspberrypi:~$ nano blutoothserver.py
  GNU nano 3.2                     blutoothserver.py

import bluetooth
server_sock=bluetooth.BluetoothSocket(bluetooth.RFCOMM)
port=2
server_sock.bind((" Server的IP ",port))
server_sock.listen(1)
client_sock,address = server_sock.accept()
print "Accepted connectoin form ",address

data = client_sock.recv(1024)
print "received [%s]" % data

client_sock.close()
server_sock.close()
```

Bluetooth Client 範例



```
import bluetooth

bd_addr = " Server的 BD address "

port = 2

sock=bluetooth.BluetoothSocket( bluetooth.RFCOMM )
sock.connect((bd_addr, port))

sock.send("hello!!")

sock.close()
```

Reference

- □ Raspberry Pi IoT無線傳輸技術介紹 – Bluetooth
  - https://www.slideshare.net/raspberrypi-tw/raspberry-pi-iot-bluetooth
- □ Eddystone
  - https://github.com/google/eddystone
- □ Eddystone Protocol Specification
  - https://github.com/google/eddystone/blob/master/protocol-specification.md
- □ Eddystone-URL Beacon Implementations
  - https://github.com/google/eddystone/tree/master/eddystone-url/implementations/