



BUN LAB

Broadband Ubiquitous Networking Lab

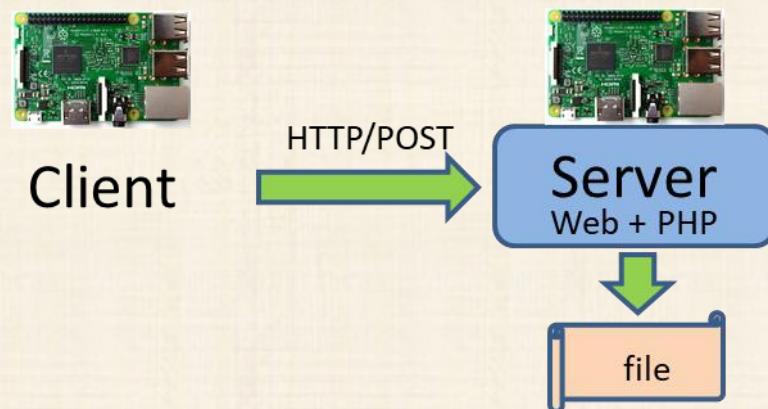
# 通訊網路實驗

IoT應用  
RaspPI Networking

Dept. of Electrical and Computer Engineering (ECE)  
**National Yang Ming 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 :有兩個sensor會傳送資料，server將資料存成不同檔案。
- 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**藍牙**傳送資料給Server。

# Introduction

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



# Introduction

## □ Web Server 組成:

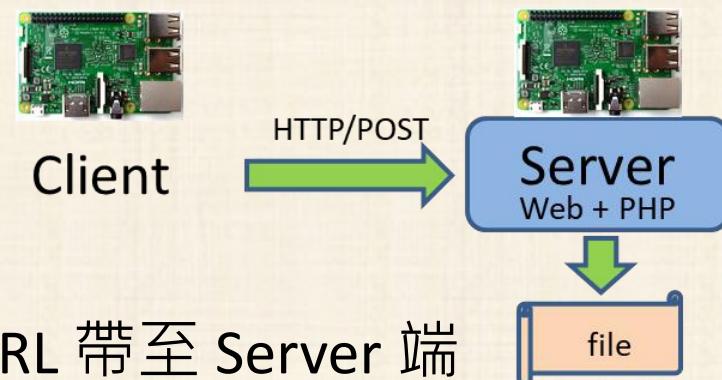
### □ 軟體(LAMP)

- Linux – Raspbian
- Apache – webserver (http) software
- MySQL – database
- PHP or Perl or Python – Web Programming Language



# Web Server & HTTP/POST

- Client & Server 的 Request / Response
- PHP範例, 利用HTTP/POST傳送資料並存檔
- GET、HEAD、POST、PUT、DELETE、TRACE、OPTIONS、CONNECT



- GET: 將所要求的資料引數由 URL 帶至 Server 端
  - <https://example.com/image/test.png>
- POST: 向指定資源提交data，Server做後續處理
  - data放在request body 中傳遞

# HTTP Request Method

Verb	SAFE	IDEMPOTENT	Cacheable	動作	語意
GET	O	O	O	讀取	請求所需要的資源。
HEAD	O	O	O	讀取	與GET相同，但伺服器只傳輸HTML的狀態與表頭。
POST	X	X	O	新增	在請求中攜帶負載，並執行特定資源的處理。
PATCH	X	X	X	部分更新	請求更新一筆資源的部分內容，必須是存在的資源。
PUT	X	O	X	完整更新	請求更新一筆資源的所有內容，必須是存在的資源，資源傳遞必須完整，否則為空。
DELETE	X	O	X	刪除	請求移除資源。

**安全方法 (Safe Methods):**在請求的過程中，是否有任何資源的改動(新增、修改、移除)。

**幂等方法 (Idempotent Methods):**求就算重複操作，也會產生相同的結果。

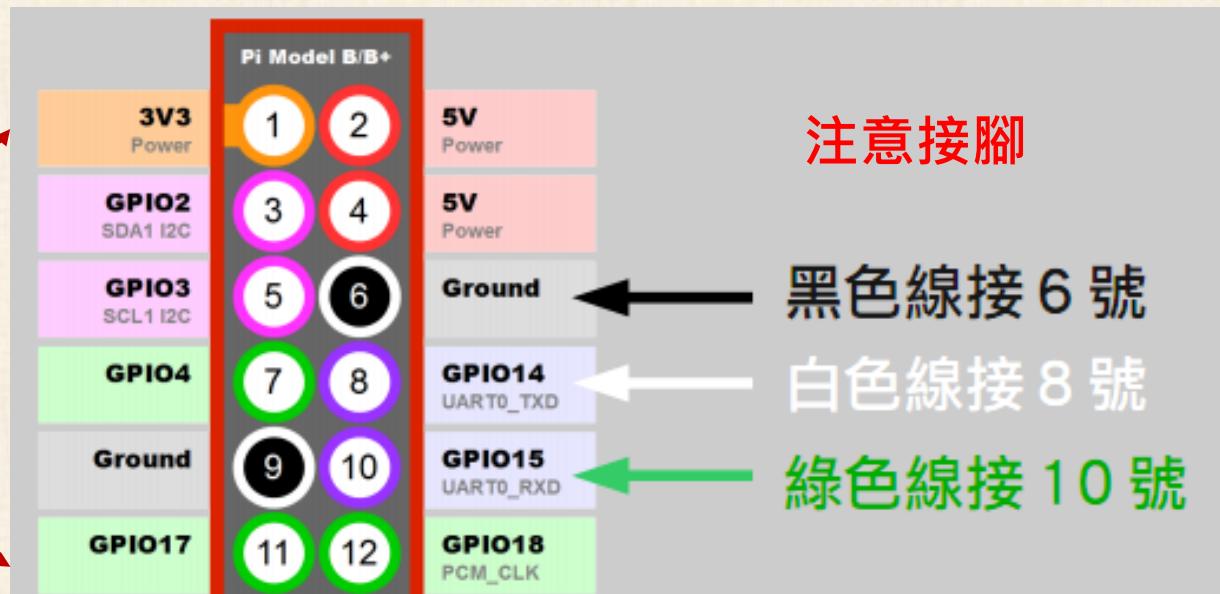
**可快取方法 (Cacheable Methods):**允許客戶端將內容保存下來。

<https://hackmd.io/@monkenWu/H1yxwXyNN>

# 溫馨小提醒

## □ 預設登入帳密

- ID: pi
- PW: raspberry



# 建立 Web Server 網站

## □ 安裝網頁伺服器apache2:

- \$ sudo apt update
- \$ sudo apt-get install apache2

**Apache**  
HTTP SERVER



```
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
```

# 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ex1.php，內容如下頁
  - \$ sudo nano ex1.php

# Web Server & HTTP/POST

- 在網頁根目錄放置一個 ex1.php 的檔案, 內容如下
  - 網頁根目錄在 /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];           送來的Temp與Sensor值,
    存到Temperature與SensorID
    echo 'Temperature: '.$Temperature. "\n";

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

當SensorID為1  
將Temperature數值存成該檔案

不要直接複製code，因為是不同字元，會造成程式錯誤

# Web Server & HTTP/POST

## □ Curl指令

□ \$ curl -d “POST內容” 目標網址

## □ 自己傳送post資料給自己

檔案名稱

□ \$ curl -d "sensor=1&Temp=25" http://**yourIP**/ex1.php

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

- 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：

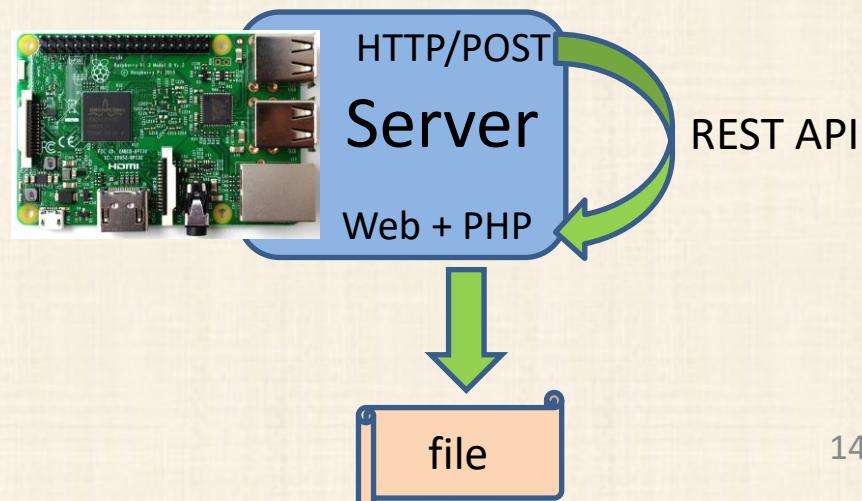
在 var/www/html 下

```
$ sudo rm index.php
```

```
$ sudo rm index.html
```

重建一個新的

```
$ sudo nano index.php
```



# Q1

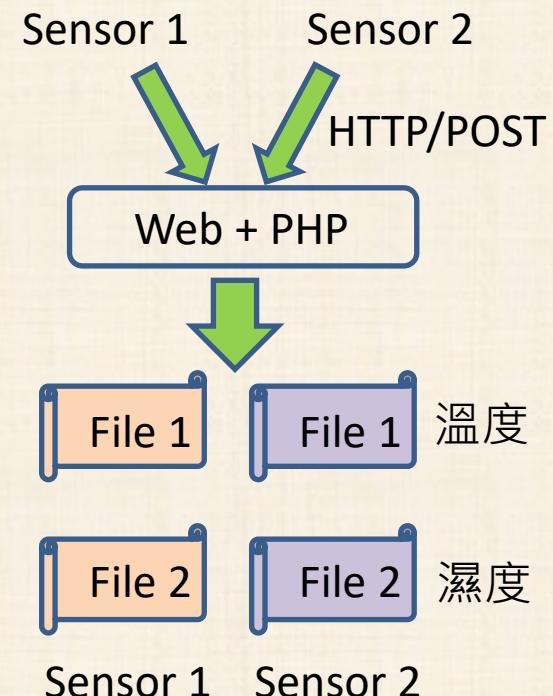
- 有兩個sensor會傳送2筆資料，修改範例php，使server將資料存成不同檔案。

Ex: sensor 1 & 2分別傳送溫濕度資訊

- 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



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

- Ctrl + C 中斷重新輸入即可

檔案名稱可自取(Ex : demo1.php)

```
<?php
    header ("Content-Type:text/html;charset=utf-8");
    /*Temp->server*/
    //ex:curl -d "sensor=1&Temp=28&Humi=75" http:IP_addr
    $Temp=[REDACTED];
    $Humi=[REDACTED];          建立參數
    $SensorID=[REDACTED];

    echo ' Sensor:'.$SensorID."\n";
    echo 'Temperature:'.$Temp."\n";
    echo 'Humidity:'.$Humi."\n";

$fp = fopen ('/home/pi/www-data/temp_'.$SensorID.'.txt','w');
fwrite($fp,$Temp);
fclose($fp);

$fp = fopen ('/home/pi/www-data/humi_'.$SensorID.'.txt','w');

[REDACTED]          寫入檔案
?>
```

# 成功範例

## □ 傳送成功

```
pi@raspberrypi:/var/www/html $ curl -d "s=1&T=28&H=78" http://192.168.50.225/demo1.php
Sensor:1
Temperature:28
Humidity:78
pi@raspberrypi:/var/www/html $ curl -d "s=2&T=25&H=58" http://192.168.50.225/demo1.php
Sensor:2
Temperature:25
Humidity:58
```

## □ cd到 www-data資料夾中, ls可查看目錄中檔案

```
pi@raspberrypi:~/www-data $ ls
humid_1.txt  humid_2.txt  temp_1.txt  temp_2.txt  temp.txt
```

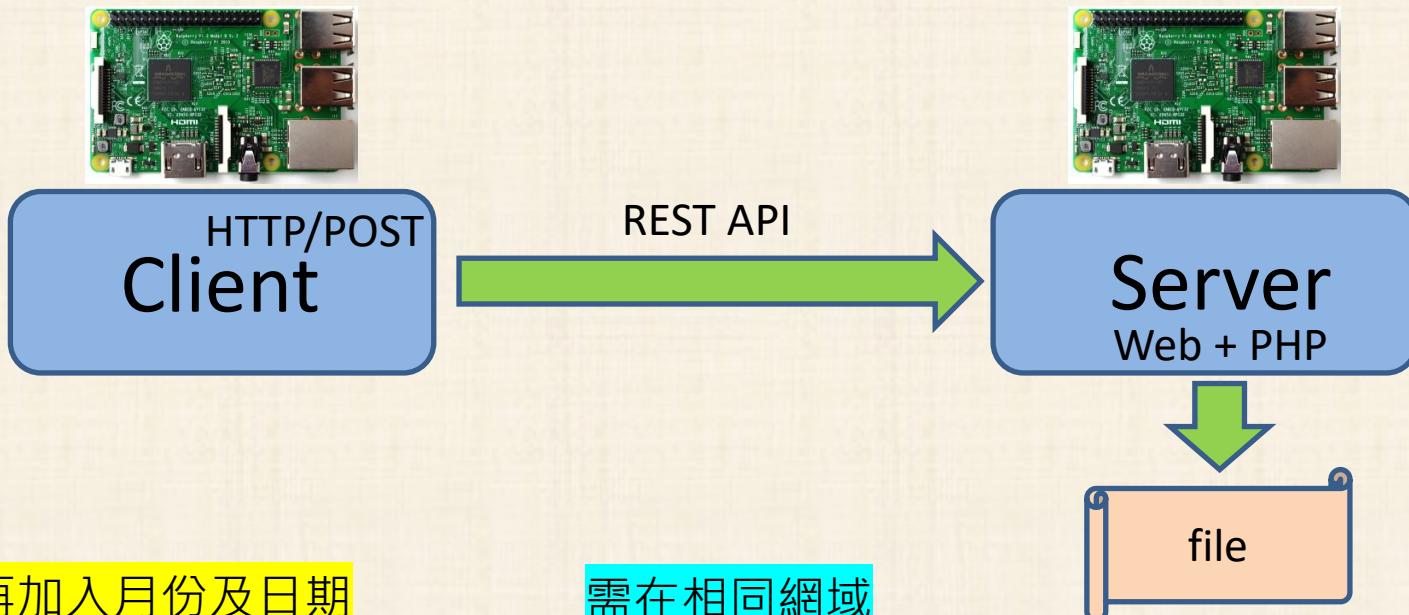
## □ 輸入more + 檔名.副檔名 可查看檔案內容

```
pi@raspberrypi:~/www-data $ more humid_1.txt
78
pi@raspberrypi:~/www-data $ more temp_1.txt
28
pi@raspberrypi:~/www-data $ more humid_2.txt
58
pi@raspberrypi:~/www-data $ more temp_2.txt
25
```

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，輪流操作、demo
  - Server需開啟檔案確認有成功接收到資料(參考範例)
  - 加入月份及日期，檔案命名為 month.txt、date.txt
  - 新創一個php檔案(Ex:demo2.php)，檔名不可相同



```
$ curl -d "POST內容" http://server_IP_addr/demo2.php
```

# 成功範例

## □ Client

```
pi@raspberrypi: ~
pi@raspberrypi:~ $ curl -d "s=1&M=03&D=26&T=28&H=58" http://192.168.50.225
Minth:03
Date:26
Sensor:1
Temperature:28
Humidity:58
pi@raspberrypi:~ $
```

# 成功範例

## □ Server

```
pi@raspberrypi:~/www-data
pi@raspberrypi:~/www-data $ ls -l
total 28
-rw-r--r-- 1 www-data www-data 2 Mar 26 12:36 date.txt
-rw-r--r-- 1 www-data www-data 2 Mar 26 12:36 humi_1.txt
-rw-r--r-- 1 www-data www-data 2 Mar 26 11:53 humi_2.txt
-rw-r--r-- 1 www-data www-data 2 Mar 26 12:36 month.txt
-rw-r--r-- 1 www-data www-data 2 Mar 26 12:36 temp_1.txt
-rw-r--r-- 1 www-data www-data 2 Mar 26 11:53 temp_2.txt
-rw-r--r-- 1 www-data www-data 5 Mar 26 11:04 temp.txt
pi@raspberrypi:~/www-data $ more month.txt
03
pi@raspberrypi:~/www-data $ more date.txt
26
pi@raspberrypi:~/www-data $ more temp_1.txt
28
pi@raspberrypi:~/www-data $ more humi_1.txt
58
```

此次建立的檔案

# 藍牙 Bluetooth

- 目的
  - 讓裝置在短距離間交換資料，以形成個人區域網路（PAN）
- 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: IEEE802.15(停止維護)
- Bluetooth 4.0 Low Energy
- Bluetooth 5.0 Faster, Further, for IoT



# 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版是否有內建藍牙  
優先使用內建的

```
pi@raspberrypi:~$ hciconfig
hci0:  Type: Primary  Bus: USB → USB 藍牙接收器
        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

[red box] hci1: → 內建
        Type: Primary  Bus: UART → 內建
        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
```

# Python Bluetooth

- 使裝置可被搜尋且也可以搜尋別人
  - \$ sudo hciconfig hci0 piscan
- 更改裝置藍牙名稱
  - \$ 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

```
import bluetooth as bt  
  
nearby_device = bt.discover_devices()  
  
for bdaddr in nearby_device:  
    print(bdaddr)  
    print(bt.lookup_name(bdaddr))
```

記得先建立新檔案  
\$ nano + 檔名.py

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

# Q3

- 開啟一個Bluetooth的server和client，  
client傳送 client\_學號 & server\_學號 紿server

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

注意：檔名不能有 bluetooth 字眼

Ex. BT\_client.py, BT\_server.py

# Bluetooth Server範例

```
import bluetooth

server_sock=bluetooth.BluetoothSocket (bluetooth.RFCOMM)
port=2

server_sock.bind ((" Server BD address ", port))
server_sock.listen(1) #可以接受多少連線

client_sock,address = server_sock.accept()
print "Accepted connection from ",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        " # server 端的 addr

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://speakerdeck.com/piepie\\_tw/introduction-to-raspberry-pi-iot-with-bluetooth?slide=42](https://speakerdeck.com/piepie_tw/introduction-to-raspberry-pi-iot-with-bluetooth?slide=42)