

Topic#3 Lab2 實驗結報

姓名：蔡佩蓉 學號：109511286

一、實驗目的

Q1: 架設 Web Server，利用 HTTP/POST 送資料 (sensor、溫度與濕度) 至 server 端，server 將資料存成不同的檔案。

Q2: 使用兩個 Raspberry Pi，一個作為 Client (傳送)，另一個作為 Server (接收)，在彼此間傳送資料 (sensor、月份、日期、溫度與濕度)。

Q3: 使用兩個 Raspberry Pi，建立藍牙 Server 和 Client，並透過藍牙傳送資料 (client_學號 和 server_學號)。

二、實驗過程 (Code + 說明)

Q1: 有兩個 sensor 會傳送資料，server 將資料存成不同檔案。

```
<?php
# 設置響應的內容類型和字符編碼 utf-8
header("Content-Type:text/html; charset=utf-8");

# 從 client 端接收 POST 數據後分別存到各自參數裡
$Temp=$_POST['T'];
$Humid=$_POST['H'];
$SensorID=$_POST['S'];

# 輸出接收到的數據
echo 'Sensor:'. $SensorID. "\n";
echo 'Temperature:'. $Temp. "\n";
echo 'Humidity:'. $Humid. "\n";

# 將數據寫入文件
$fp = fopen('/home/pi/www-data/temp_'.$SensorID.'.txt',
'w'); # 文件路徑
fwrite($fp, $Temp);
fclose($fp);
$fp=fopen('/home/pi/www-data/humid_'.$SensorID.'.txt',
'w'); # 文件路徑
fwrite($fp, $Humid);
fclose($fp);
?>
```

說明：

1. Apache 和 PHP 之間的關係。

Apache 是一種常見的開源網頁伺服器軟體，用於提供 Web 服務。它能夠接收客戶端的 HTTP 請求並回應這些請求，以提供網頁、應用程式或其他內容。Apache 具有高度可配置性和擴展性，支援多種作業系統，並且在 Web 伺服器市場上佔有相當大的份額。

PHP 是一種流行的伺服器端腳本語言，專門用於 Web 開發。它能夠與 HTML 整合，以動態生成網頁內容。PHP 可以在伺服器端運行，處理請求並生成動態的 HTML 頁面，也能與資料庫交互，處理表單數據，以及許多其他的伺服器端任務。

Apache 伺服器與 PHP 的結合常被使用於建立 Web 伺服器環境。當客戶端（如瀏覽器）發送 HTTP 請求到 Apache 伺服器時，如果請求的文件是 PHP 檔案，Apache 會將該請求交給 PHP 處理。PHP 處理器解讀並執行 PHP 代碼，然後將結果返回給 Apache，最後將其發送給客戶端。

Request → Apache → PHP → Output → HTML → Apache → Response

2. Client 傳送值的方式：

```
curl -d "S=1&T=29&H=50" http://172.20.10.4/demo1.php
```

172.20.10.4 為自己的 IP

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

```
<?php
# 設置響應的內容類型和字符編碼 utf-8
header("Content-Type:text/html; charset=utf-8");

# 從 client 端接收 POST 數據後分別存到各自 variable 裡面
$Temp=$_POST['T'];
$Humid=$_POST['H'];
$SensorID=$_POST['S'];
$Month=$_POST['M'];
$Date=$_POST['D'];

# 輸出接收到的數據
echo "Sensor:". $SensorID. "\n";
echo "Temperature:". $Temp. "\n";
echo "Humidity:". $Humid. "\n";
echo "Month:". $Month. "\n";
```

```

    echo 'Date:' . $Date . "\n";

    # 將數據寫入文件
    $fp = fopen('/home/pi/www-data/temp_' . $SensorID . '.txt',
'w');
    fwrite($fp, $Temp);
    fclose($fp);
    $fp=fopen('/home/pi/www-data/humid_' . $SensorID . '.txt',
'w');
    fwrite($fp, $Humid);
    fclose($fp);
    $fp=fopen('/home/pi/www-data/month_' . $SensorID . '.txt',
'w');
    fwrite($fp, $Month);
    fclose($fp);
    $fp=fopen('/home/pi/www-data/date_' . $SensorID . '.txt',
'w');
    fwrite($fp, $Date);
    fclose($fp);
?>

```

綠色 Bold 字為比 Q1 多新增的部分

說明：

1. Client 傳送值的方式：

curl -d "S=1&T=29&H=50&M=12&D=31" http://172.20.10.2/ex2.php
 172.20.10.2 為組員的 IP (server 端)；ex2.php 為組員的 PHP 檔案
 (server 端)

Q3: 兩人為一組，Client 透過藍芽傳送資料給 Server

```

# Server
import bluetooth

# 建立一個 RFCOMM 藍牙 socket
server_sock=bluetooth.BluetoothSocket(bluetooth.RFCOMM)

port=2

# 綁定藍牙位址(Server BD address)和通道號
server_sock.bind(("B8:27:EB:74:0E:EF", port))
server_sock.listen(1) #可接受多少連線

# 接受來自客戶端的連接請求(會一直到客戶端連上才返回)
client_sock,address = server_sock.accept()
print "Accepted connection from ",address

# 接收來自客戶端的資料(最多接收 1024 bytes)
data = client_sock.recv(1024)
print "received [%s]" % data

```

```
# 關閉與客戶端的連接
client_sock.close()
# 關閉伺服器 socket
server_sock.close()
```

```
# Client
import bluetooth

# 欲連線的藍牙裝置的地址 (Server BD address)
bd_addr="DC:A6:32:98:11:A1"

port=2

# 建立一個 RFCOMM 藍牙 socket
sock=bluetooth.BluetoothSocket(bluetooth.RFCOMM)
# 連接到指定的藍牙設備
sock.connect((bd_addr, port))

# 向藍牙設備發送資料
sock.send("client_109511286, server_109511041")

# 關閉藍牙連接
sock.close()
```

三、問題及解法

問題 1: 如何区分不同传感器的数据以创建不同的文件。

解法：

- 通过在传感器数据中添加唯一标识符 (SensorID)，將数据分离并保存到相应的文件中。
- 用 if-else statement 辨別 SensorID 為 1 或 2，以此來將檔案分類。

四、心得

Q1: 這個實驗讓我們熟悉了 PHP 中 (server 端) 處理 POST 資料的方法，並學會了如何將提取到的資料寫入對應檔案中。

Q2: 這個實驗與 Q1 相似，不過多了兩個變數 Month 和 Date (多了兩個檔案的寫入操作，分別將月份和日期資訊寫入檔案中)。這個實驗讓我們了解了如何在不同裝置間透過 PHP 和網路通訊來傳送資料。Client 端負責向

server 端發送 HTTP POST 請求，並傳送資料給 server 端。Server 端則會接收來自 client 端的 HTTP POST 請求，提取資料，然後將資料寫入到檔案中。

Q3: 與 Topic 2: Sensor Networks Technologies 中用過的 BLE 不同，這次用的是 Bluetooth Classic。這個實驗讓我們學會了如何使用藍牙在不同裝置之間傳送資料，並學到了藍牙通訊的基本原理和連線設定。(建立 RFCOMM 藍牙連接並在該連接上發送訊息至特定的藍牙設備)

在 demo 的時候就照著講義打 code，也不太清楚訊息是怎麼傳來傳去的，寫結報時認真研究才了解它們如何運作，覺得非常之神奇！