

# **Raspberry Pi3**

## **Linux Driver II 實驗模組建置**

實驗模組名稱：Linux Driver (II)

開發學生：溫方志、陳鎮國

開發教師：陳鵬升

學校系所：中正大學資訊工程學系

聯絡電話：05-2720411 ext.33117

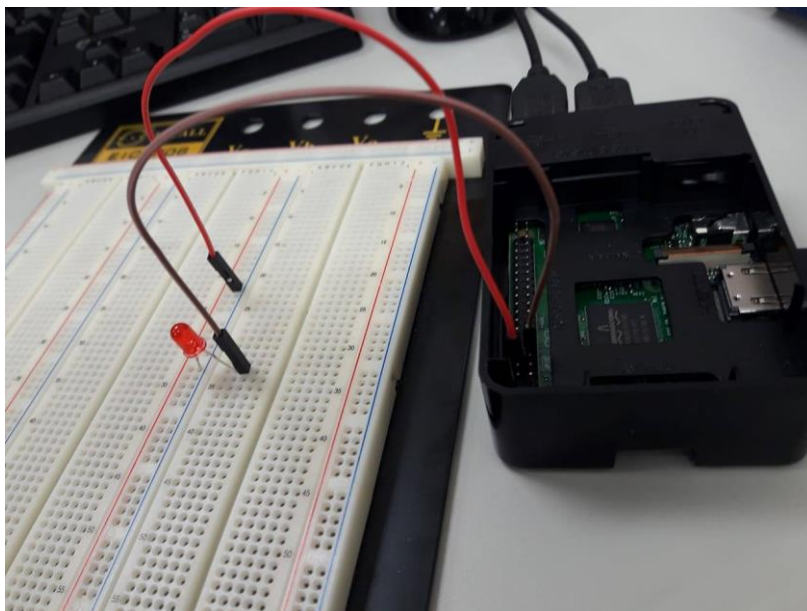
聯絡地址：62102 嘉義縣民雄鄉大學路 168 號資  
訊工程學系

實驗平台：Ubuntu

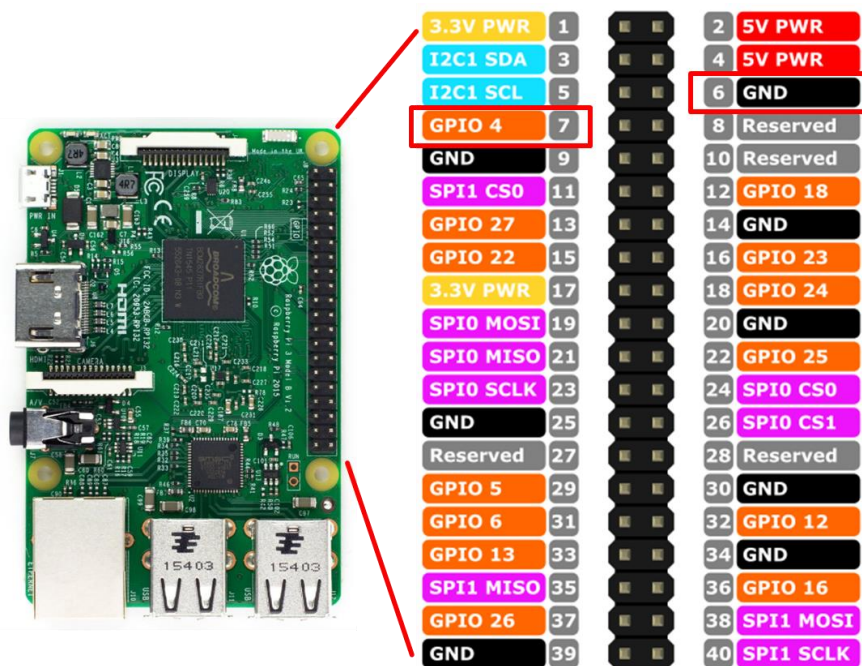
實驗內容關鍵字：

## Part 1

設定硬體，把 LED 和電線插好如下圖所示：



實驗使用的 GPIO4 與 GND 腳位。



## Part2

如前次實驗所述，Linux 把 device 當成檔案使用，所以一般的 file operation 也是行的通的，以下範例是藉由 shell 提供的指令操作 RaspberryPi 的 GPIO。

Write the following commands in your terminal:

Change to root

/\* 配置 GPIO，產生了一個新的目錄“gpio4”，裡面包含了該 I/O 的輸出輸入設置等配置文件 \*/

```
$ cd /sys/class/gpio
```

```
$ echo 4 > /sys/class/gpio/export
```

/\* 設置 GPIO 的輸出/輸入方向 \*/

```
$ echo out > /sys/class/gpio/gpio4/direction
```

/\* 設置 GPIO 的輸出電壓 \*/

```
$ echo "1" > /sys/class/gpio/gpio4/value
```

/\* 關閉 GPIO，可以看到 gpio4 已經被刪除 \*/

```
$ echo 4 > /sys/class/gpio/unexport
```

## Part 3

既然 shell 的指令行的通，那 C 語言的 `fopen` 呢？答案也是行的通的，藉由 `fopen()` 開檔和 `fwrite()` 寫檔也可以達到控制 GPIO 的需求。

**DEMO:** 請使用下列的函式，寫一個 C program，讓 LED 燈可以持續閃爍。

(1)

//需由 root 身份執行程式

//配置 GPIO

```
p = fopen("/sys/class/gpio/export", "w");  
fprintf(p, "%d", 4);  
fclose(p);
```

產生了一個新的目錄 “gpio4”，裡面包含了該 I/O 的輸出輸入設置等配置文件

(2)

設置 GPIO 的輸出/輸入方向

```
p = fopen("/sys/class/gpio/gpio4/direction", "w");  
fprintf(p, "out");  
fclose(p);
```

(3)

設置 GPIO 的輸出電壓

```
p = fopen("/sys/class/gpio/gpio4/value", "w");  
fprintf(p, "%d", 1);  
fclose(p);
```

(4)

關閉 GPIO

```
p = fopen("/sys/class/gpio/unexport", "w");  
fprintf(p, "%d", 4);  
fclose(p);
```

可以看到 gpio4 已經被刪除