

主要特點 (Main Features):

- USB-Serial (PL2303) 模組 — 直接與 Raspberry Pi B+ UART 連線
- UART 外接腳座
- CC2500 RF 模組延伸腳座
- I2C 延伸腳座
- PWM 延伸腳座
- 字元型液晶模組延伸腳座 (LCM Module) 連接座
- 紅外線訊號接收器
- 繼電器
- 蜂鳴器
- I2C 介面序列型 EEPROM
- SPI 介面 FLASH 記憶體
- GPIO 按鈕 x 2
- GPIO 發光二極體 x 2
- Raspbian Linux 驅動程式與範例程式碼支援

目錄

主要特點 (Main Features):.....	0
目錄.....	錯誤! 尚未定義書籤。
1. 硬體規格 (Technical Specifications)	2
2. 技術設計圖樣 (Technical Drawing).....	3
3. 安裝說明 (First Time Installation)	3
3.1 適用作業系統.....	3
3.2 硬體安裝:	4
3.3 使用 USB-Serial 模組登入 Raspbian Linux 作業系統	5
3.4 軟體安裝設定:	6
4. 使用 Rasp Pi I/O 子板 (Using your IO board).....	7
4.1 使用 USB-Serial 連接 Raspberry Pi B+ UART	7
4.2 SPI 介面操作 FLASH 記憶體測試	8
4.3 使用 i2c-tools 讀寫 EEPROM	9
4.4 字元型液晶模組延伸腳座(LCM Module) 連接座	9
4.5 紅外線訊號接收器測試	10
4.6 繼電器	11
4.7 蜂鳴器	11
4.8 GPIO 按鈕	12
4.9 發光二極體	12
4.10 PWM 輸出	12
5. 其他連結 (Additional Links)	13

1. 硬體規格 (Technical Specifications)

Size: 65 x 56 mm

USB-Serial (PL2303)

- USB 轉 TTL 連接 Pi 板序列埠。
- 預設鮑率 115200

UART 延伸腳座

- UART SWITCH 切換 Pi 板序列埠連接 USB-Serial 或 UART 延伸腳座

CC2500 RF 模組 SPI 介面延伸腳座

- SPI Channel Select : 1 (Pin 26)

字元型液晶模組延伸腳座 (LCM Module)

- 可外接字元型液晶模組或作為一般 GPIO 延伸腳座使用。

紅外線訊號接收器 (IR Signal Receiver)

- 接收一般紅外線遙控器訊號。

繼電器 (Relay)

- 10A 240VDC 繼電器

蜂鳴器 (Buzzer)

- 自激式 DC 直流蜂鳴器

I2C 介面序列型 EEPROM

- Slave Address: 0x50
- 型號: 24AA16

SPI 介面快閃記憶體 (FLASH memory)

- SPI Channel Select : 0 (Pin 24)
- 型號: MX25L4005A

GPIO 按鈕 x 2

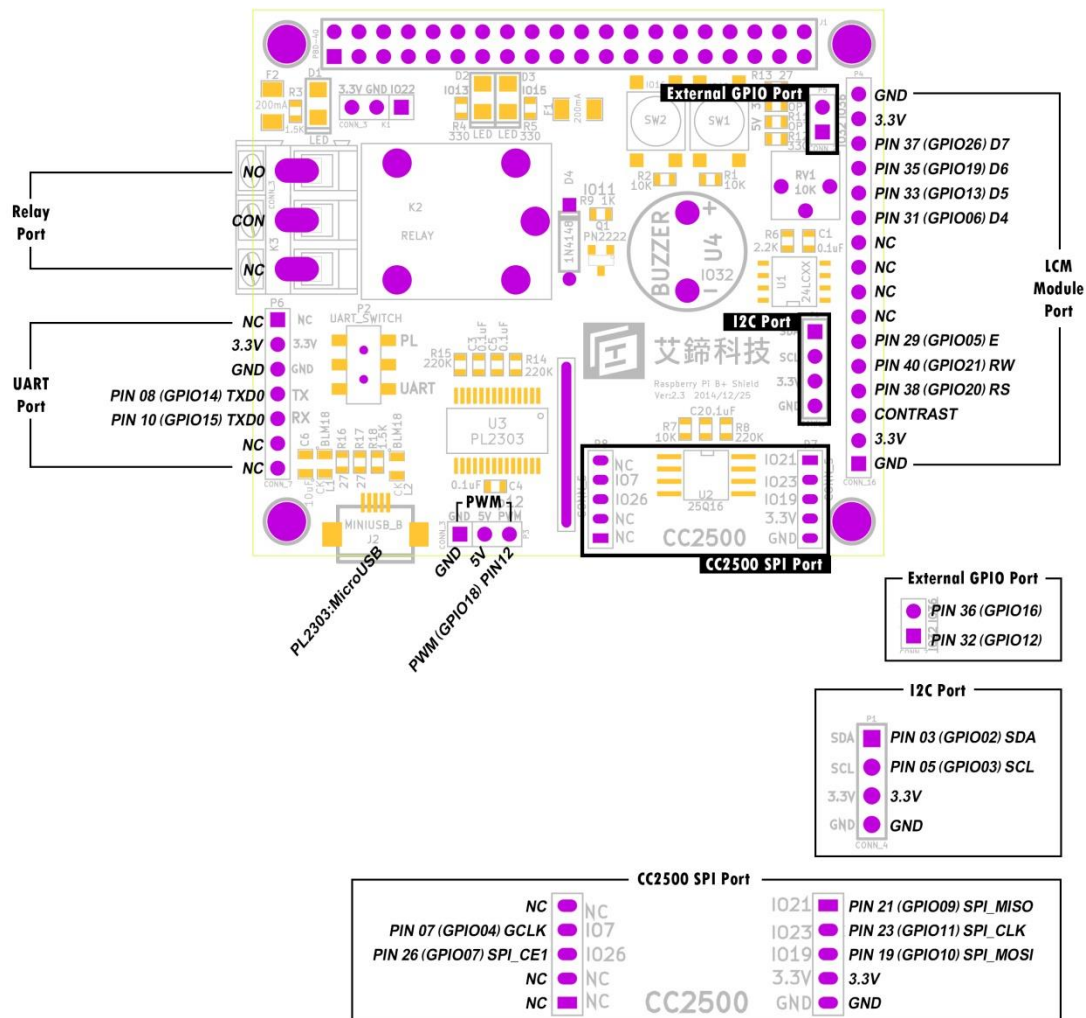
GPIO 發光二極體 x 2

I2C 延伸腳座

PWM 延伸腳座

Linux (Raspbian) 驅動程式支援

2. 技術設計圖樣 (Technical Drawing)



3. 安裝說明 (First Time Installation)

3.1 適用作業系統

請使用可支援 I²C 與 SPI 傳輸介面的作業系統:

1. **Raspbian Linux** 作業系統映像檔請至 Raspberry Pi 官方網頁下載最新版
本 <http://www.raspberrypi.org/downloads/> 並參照網頁教學安裝燒錄至 microSD 卡上。

2. 插入該 micro SD 卡至 Raspberry Pi B+ 板子上進行開機。
3. 第一次登入作業系統請執行系統設定程式，並參考 3.4.1 章節開啟 I2C 和 SPI 介面功能：

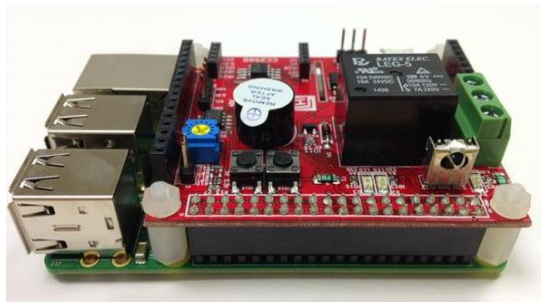
```
$ sudo raspi-config
```

3.2 硬體安裝:

適用 **Raspberry Pi B+** 版本外觀如下:



- 如下圖所示連接 Rasp Pi I/O 子板於 Raspberry Pi B+ 板上:



- 如下圖所示連接 LCM 模組於板子之上字元型液晶模組延伸腳座:



3.3 使用 USB-Serial 模組登入 Raspbian Linux 作業系統

- 使用 USB-micro 連接線，連接電腦和 Rasp Pi I/O 子板上的 USB-Serial 轉接埠。



- 如下圖箭頭所示，切換 SWITCH 朝向“PL”標示，連接 USB-Serial 模組與 Raspberry Pi UART：



- 登入 Raspbian Linux 系統：

此時從個人電腦端可以透過終端機連線軟體登入 Raspbian Linux 作業系統。

個人電腦端作業系統為 Windows，需至 [Prolific 官方網站](#) 下載驅動程式。

安裝完成後可至裝置管理員中查詢與 PL2303 晶片相對應的 COM Port 號碼。個人電腦端作業系統為 Linux 或 MAC OS X，可使用 minicom 通訊軟體開啟裝置檔 “/dev/ttyUSB0” (Linux) 或 “/dev/tty.usbserial” (MAC OS X)。

終端機連線設定為 8-N-1 連線模式，鮑率 (Baud Rate) 值為 115200:

鮑率 (Baud Rate): 115200

資料位元 (Data Btis): 8

同位檢查 (Parity Check): None

停止位 (Stop Bit): 1

- Raspbian Linux 系統預設 (帳號/密碼) 為 (pi/raspberry)。

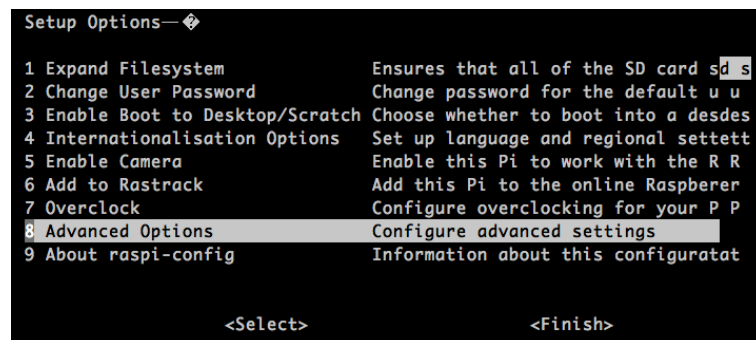
3.4 軟體安裝設定:

Raspbian Linux 作業系統的軟體安裝需要上網，請確認 Raspberry Pi B+ 板子上的網路接孔與網路連接。作業系統預設使用 DHCP 方式自動分配 IP。

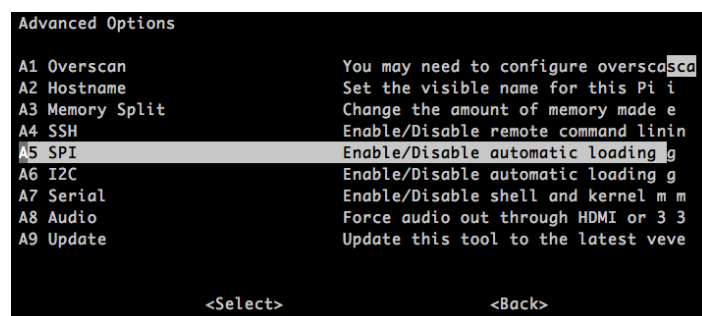
- 設定 SPI 與 I2C 模組為系統開機後自動載入

登入系統後執行指令進入類似以下系統設定畫面。

```
$ sudo raspi-config
```



如圖所示，進入選項 **8 Advanced Options** → 分別進入選項 **A5 SPI** 與 **A6 I2C** 之後選擇 <Yes> 打開 SPI 與 I2C 功能設定。



之後退回一開始的畫面選擇 <Finish> 離開設定程式後，重新開機。

```
$ sudo reboot
```

- 下載並安裝 **bcm2835 C 語言函式庫**

請至網頁 <http://www.airspayce.com/mikem/bcm2835/> 找到最新版 bcm2835 C 函式庫壓縮檔 (bcm2835-1.xx.tar.gz) 連結，照下列指示安裝:

```
$ wget
http://www.airspayce.com/mikem/bcm2835/bcm2835-1.xx.tar.gz
$ tar zxvf bcm2835-1.xx.tar.gz
$ cd bcm2835-1.xx
$ ./configure
$ make
$ sudo make check
$ sudo make install
```

- 下載並安裝 **wiringPi C 語言函示庫**

```
$ sudo apt-get install git-core
$ git clone git://git.drogon.net/wiringPi
$ cd wiringPi
$ ./build
```

- 安裝 **I2C 工具**

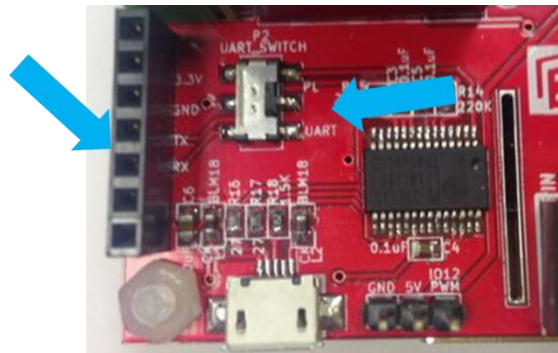
```
$ sudo apt-get install i2c-tools
```

4. 使用 **Rasp Pi I/O 子板 (Using your IO board)**

4.1 使用 **USB-Serial 連接 Raspberry Pi B+ UART**

如前文 3.3 章節所示，將所示 UART SWITCH 切換到標示 “PL” 方向，Rasp Pi I/O 子板上的 USB-Serial 模組即與 Raspberry Pi B+ 板上的 UART 端 (Tx/Rx: Pin 08/Pin 10) 連接。如下圖箭頭所示將 UART SWITCH 切換到 “UART” 標示方向，Raspberry

Pi B+ 板 UART 的 Tx/Rx (Pin 08:Pin 10) 會被分別連接至側邊 UART port 針座的 TX/RX 端，切斷與 USB-Serial 模組的連繫。



4.2 SPI 介面操作 FLASH 記憶體測試

Rasp Pi I/O 子板上的 FLASH 記憶體被連接至 SPI 介面的 Channel 0。下列範例 C 程式碼為利用 bcm2835 C 函式庫所寫的測試程式：

```
#include<stdio.h>
#include<stdlib.h>
#include<bcm2835.h>
int main(int argc,char *argv[]){
    uint8_t buff[]={0x90, 0x00,0x00,0x00,0x00,0x00 };
    int i = 0;
    bcm2835_init();
    bcm2835_spi_begin();
    bcm2835_spi_setBitOrder(BCM2835_SPI_BIT_ORDER_MSBFIRST);
    bcm2835_spi_setDataMode(BCM2835_SPI_MODE0);
    bcm2835_spi_setClockDivider(BCM2835_SPI_CLOCK_DIVIDER_65536);
    bcm2835_spi_chipSelect(BCM2835_SPI_CS0);
    bcm2835_spi_setChipSelectPolarity(BCM2835_SPI_CS0, LOW);

    bcm2835_spi_transfer(buff,6);
    printf("Manufacturer ID: %02X\n", buff[4]);
    printf("Device ID: %02X\n",buff[5]);
    return 0;
}
```

將其存檔為 test.c 並用 gcc 指令將其編譯為可執行檔 test。如下圖所示，執行 test 之後螢幕上會顯示從 FLASH 記憶體中讀取到的廠商代碼(C2)和裝置代碼(12)：

```
pi@raspberrypi ~ $ gcc test.c -o test -lbcm2835
pi@raspberrypi ~ $ sudo ./test
Manufacturer ID: C2
Device ID: 12
```

4.3 使用 i2c-tools 讀寫 EEPROM

Rasp Pi I/O 子板上的 I2C 上已安裝好一個 EEPROM (24AA16) 其 Slave Address 為 0x50。在使用 i2c-tools 工具套件測試 EEPROM 之前必須先使用 modprobe 指令載入 i2c-dev 驅動模，載入成功之後使用 lsmod 指令可以看到 i2c_dev 模組已經掛載在 Linux 系統之中了：

```
pi@raspberrypi ~ $ sudo modprobe i2c-dev
pi@raspberrypi ~ $ lsmod
Module                  Size  Used by
i2c_dev                 5769  0
snd_bcm2835             19496  0
snd_soc_tas5713          5573  0
```

之後便可使用 i2c-tools 工具套件的指令 i2cdetect 來偵測 I2C 介面上所有元件的 slave address。

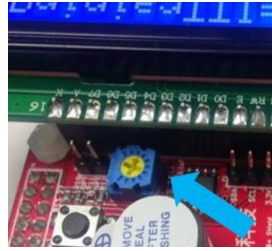
```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- UU -- -- --
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  50 -- -- -- -- -- -- -- -- -- -- -- -- --
60:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
70:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

可看到有一元件占用了 slave address: 0x50，此元件即為 Rasp Pi I/O 子板上的 EEPROM。知道 slave address 之後便可使用 i2c-tools 模組所提供的 i2cset 和 i2cget 存取 EEPROM 中指定暫存器的值。下圖示範儲存數值 0x64 至 EEPROM 暫存器 0x01，並讀取出來驗證其結果：

```
pi@raspberrypi ~ $ sudo i2cset -f -y 1 0x50 0x13 0x64
pi@raspberrypi ~ $ sudo i2cget -f -y 1 0x50 0x13
0x64
pi@raspberrypi ~ $
```

4.4 字元型液晶模組延伸腳座(LCM Module) 連接座

依照 3.2 節安裝 LCM 模組至 Rasp Pi I/O 子板上，調整 LCM 顯示對比度可透過下圖箭頭所示的可變電阻進行。



LCM 模組需使用 4bit 操作模式，腳位對照如下：

LCM	RS	RW	E	D4	D5	D6	D7
Pi Pin #	Pin 38	Pin 40	Pin 29	Pin 31	Pin 33	Pin 35	Pin 37
GPIO #	GPIO 20	GPIO 21	GPIO 05	GPIO 06	GPIO 13	GPIO 19	GPIO 26

因操作方式較為複雜，詳細的操作方式請參考教學網頁：

<http://blog.ittraining.com.tw/2014/12/raspberry-pi-b-python-lcd-16x2-hd44780.html>

4.5 紅外線訊號接收器測試

線外線接收器訊號腳位與 Raspberry Pi B+板上的 Pin: 22 接腳連接。下列範例 C 程

式碼為利用 bcm2835 C 函式庫所寫的測試程式：

```
#include<stdio.h>
#include<stdlib.h>
#include<stdint.h>
#include<sys/time.h>
#include<bcm2835.h>
#define IR_PIN RPI_BPLUS_GPIO_J8_22
struct timeval pre_t;
struct timeval now_t;
uint8_t now_l,pre_l;
uint16_t d_t;
int main(int argc,char *argv[]){
    bcm2835_init();
    bcm2835_gpio_fsel( IR_PIN, BCM2835_GPIO_FSEL_INPT);
    now_l = bcm2835_gpio_lev(IR_PIN);
    pre_l = now_l;
    while(1){
        now_l = bcm2835_gpio_lev(IR_PIN);
        if( now_l != pre_l ){
            gettimeofday(&now_t, NULL);
            d_t = 1000000 * (now_t.tv_sec - pre_t.tv_sec);
            d_t += (now_t.tv_usec - pre_t.tv_usec);
            printf("%ld\n", d_t);
            pre_l = now_l;
            pre_t.tv_sec = now_t.tv_sec;
            pre_t.tv_usec = now_t.tv_usec;
        }
    }
    return 0;
}
```

將其存檔為 `test.c` 並用 `gcc` 指令將其編譯為可執行檔 `test`。

```
pi@raspberrypi ~ $ gcc test.c -o test -lbcm2835
pi@raspberrypi ~ $ sudo ./test
9274
8895
4494
580
607
```

執行後可用一般常見紅外線遙控器朝向 Rasp Pi I/O 子板上的紅外線接收器發射訊號，螢幕上會顯示接收到的紅外線訊號變化時間間隔(單位為 `micro second`)。

可按 `Ctrl + C` 結束程式。

4.6 繼電器

繼電器開關腳位與 Raspberry Pi B+板上的 Pin: 11(GPIO17)接腳連接。可使用下列指令設定該 pin:11 (GPIO17)為輸出來操作繼電器:

```
$ echo 17 > /sys/class/gpio/export
$ echo out > /sys/class/gpio/gpio17/direction
```

打開繼電器:

```
$ echo 1 > /sys/class/gpio/gpio17/value
```

關掉繼電器:

```
$ echo 0 > /sys/class/gpio/gpio17/value
```

4.7 蜂鳴器

該蜂鳴器為自激式 DC 直流蜂鳴器，和 Raspberry Pi 板上的 Pin 32 連結。可使用下列指令設定該 pin:32 (GPIO12)為輸出來操作蜂鳴器:

```
$ echo 12 > /sys/class/gpio/export
$ echo out > /sys/class/gpio/gpio12/direction
```

打開蜂鳴器:

```
$ echo 0 > /sys/class/gpio/gpio12/value
```

關掉蜂鳴器:

```
$ echo in > /sys/class/gpio/gpio12/direction
```

4.8 GPIO 按鈕

Rasp Pi I/O 子板上的兩顆按鈕 SW01、SW02 分別對應 Raspberry Pi B+ 板上的 Pin 16 (GPIO23) 和 Pin 18 (GPIO24)。下列指令示範使用藉由讀取 pin:16 (GPIO23)的輸入電位來得知 SW01 的狀態:

```
$ echo 23 > /sys/class/gpio/export  
$ echo in > /sys/class/gpio/gpio23/direction
```

讀取按鈕狀態:

```
$ cat /sys/class/gpio/gpio23/value
```

若按鈕未按下回傳值為 1 (High Level)，若按鈕按下則回傳值為 0 (Low Level)

4.9 發光二極體

Rasp Pi I/O 子板上的兩顆發光二極體 LED01、LED02 分別對應 Raspberry Pi B+ 板上的 Pin 13 (GPIO 27)、Pin 15 (GPIO 22)。下列指令使用 Pin 13 (GPIO27)來操作發光二極體 LED01:

```
$ echo 27 > /sys/class/gpio/export  
$ echo out > /sys/class/gpio/gpio27/direction
```

打開發光二極體 LED01:

```
$ echo 1 > /sys/class/gpio/gpio27/value
```

關掉發光二極體 LED01:

```
$ echo 0 > /sys/class/gpio/gpio27/value
```

4.10 PWM 輸出

Rasp Pi I/O 子板上的 PWM 延伸腳座對應 Raspberry Pi B+板上的 Pin 12 (GPIO18)。

該 PWM 延伸腳座為 Raspberry Pi B+ MCU 晶片系統內建硬體 PWM 輸出。下面使用 wiringPi 套件的 gpio 指令控制該硬體 PWM (GPIO18) Pin 12 訊號輸出。

設定 Pin 12 (GPIO18) 為 PWM 輸出:

```
$ gpio mode 1 pwm
```

設定 Pin 12 (GPIO18) PWM 輸出訊號:

```
$ gpio pwm 1 500
```

關掉 Pin 12 (GPIO18) PWM 模式。

```
$ gpio mode 1 input
```

5. 其他連結 (Additional Links)

<http://blog.ittraining.com.tw/>

<http://www.raspberrypi.org/>

<http://www.airspayce.com/mikem/bcm2835/>

<http://wiringpi.com/>