

Raspberry Pi 3 Setup

for

GPIO control

目錄

安裝 GPIO 函式庫 Wiring Pi.....	3
Cross Compiler in Eclipse 環境設置(VM)	4
周邊元件接線說明.....	5
WiringPi 程式說明	8
參考資料.....	9

安裝 GPIO 函式庫 Wiring Pi

套件軟體更新

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

安裝 WiringPi

```
$ sudo apt-get install git-core
```

```
$ git clone git://git.drogon.net/wiringPi
```

取得目前最新版本

```
$ cd wiringPi
```

```
$ git pull origin
```

若已是最新版本則執行以下指令

```
$ cd wiringPi
```

```
$ sudo ./build
```

測試 WiringPi 是否安裝成功

```
$ gpio -v
```

```
$ gpio readall
```

若安裝成功，則會出現當前所使用板子的腳位圖

```
This Raspberry Pi is a revision 2 board.
root@raspberrypi:/home/pi/wiringPi# gpio readall
+-----+--Rev2--+-----+-----+-----+
| wiringPi | GPIO | Phys | Name   | Mode | Value |
+-----+-----+-----+-----+-----+-----+
| 0 | 17 | 11 | GPIO 0 | IN   | Low   |
| 1 | 18 | 12 | GPIO 1 | IN   | High  |
| 2 | 27 | 13 | GPIO 2 | IN   | Low   |
| 3 | 22 | 15 | GPIO 3 | IN   | Low   |
| 4 | 23 | 16 | GPIO 4 | IN   | High  |
| 5 | 24 | 18 | GPIO 5 | IN   | High  |
| 6 | 25 | 22 | GPIO 6 | IN   | High  |
| 7 | 4  | 7  | GPIO 7 | IN   | Low   |
| 8 | 2  | 3  | SDA    | IN   | High  |
| 9 | 3  | 5  | SCL    | IN   | High  |
| 10 | 8  | 24 | CE0    | IN   | Low   |
| 11 | 7  | 26 | CE1    | IN   | High  |
| 12 | 10 | 19 | MOSI   | IN   | Low   |
| 13 | 9  | 21 | MISO   | IN   | Low   |
| 14 | 11 | 23 | SCLK   | IN   | Low   |
| 15 | 14 | 8  | TxD    | ALT0 | High  |
| 16 | 15 | 10 | RxD    | ALT0 | High  |
| 17 | 28 | 3  | GPIO 8 | ALT2 | Low   |
| 18 | 29 | 4  | GPIO 9 | ALT2 | Low   |
| 19 | 30 | 5  | GPIO10 | ALT2 | Low   |
| 20 | 31 | 6  | GPIO11 | ALT2 | Low   |
+-----+-----+-----+-----+-----+-----+
```

Cross Compiler in Eclipse 環境設置(VM)

將 Pi 底下的 libwiringPi 檔 copy 至 VM 中，並重新命名為「libwiringPi.so」
/usr/local/lib/libwiringPi.so.1.0

在 VM 中安裝 WiringPi

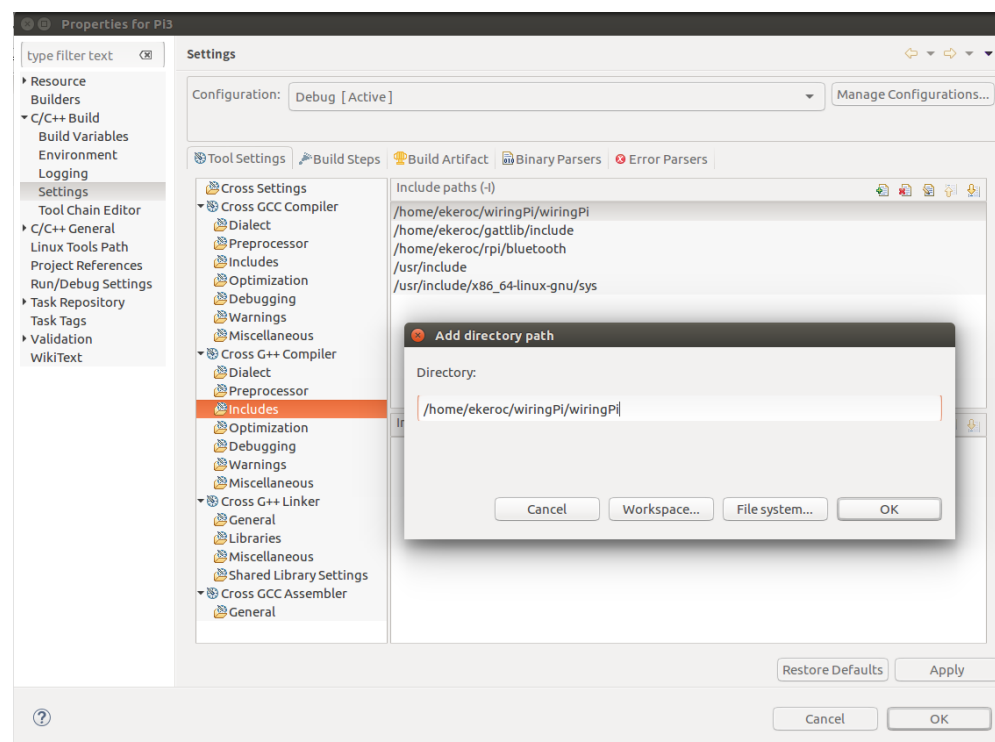
```
git clone git://git.drogon.net/wiringPi
```

```
cd wiringPi
```

```
git pull origin
```

執行 Eclipse

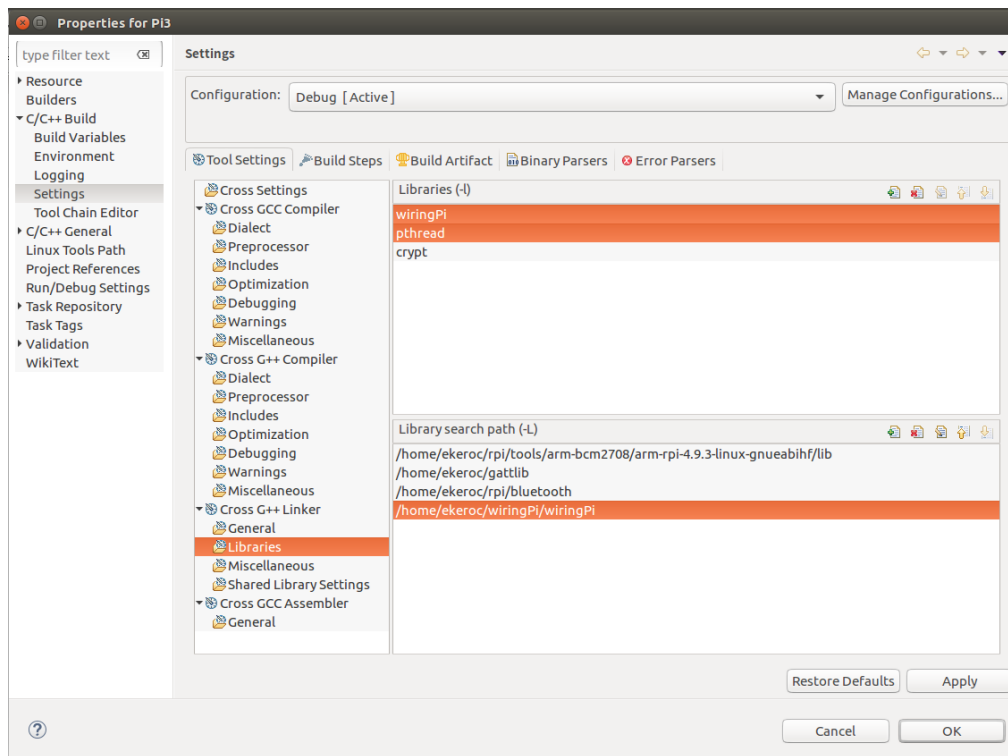
至 Project Properties -> GCC C++ Compiler -> Includes -> Include path 增加.h 路徑



至 Project Properties -> GCC C++ Linker -> Libraries

-> Libraries 增加 WiringPi 及 pthread

-> Libraries search path TAB 增加「libwiringPi.so」的路徑位置



周邊元件接線說明

所需元件：

硬體控制板(Arduino or Raspberry Pi)

步進馬達 x1

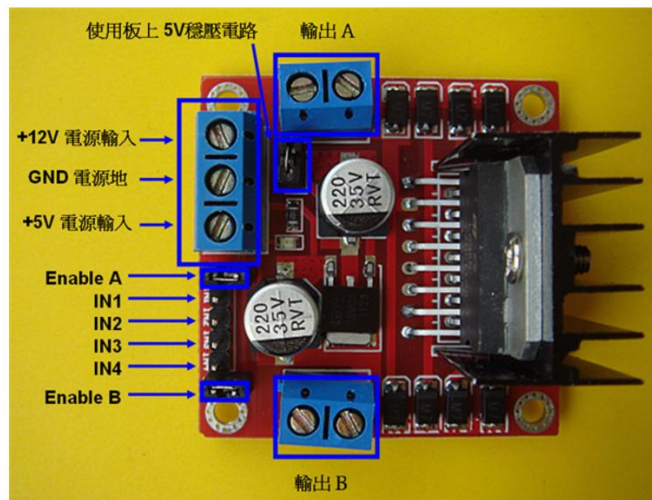
伺服馬達 x1

L298N 驅動板

18650 電池 x2

元件說明

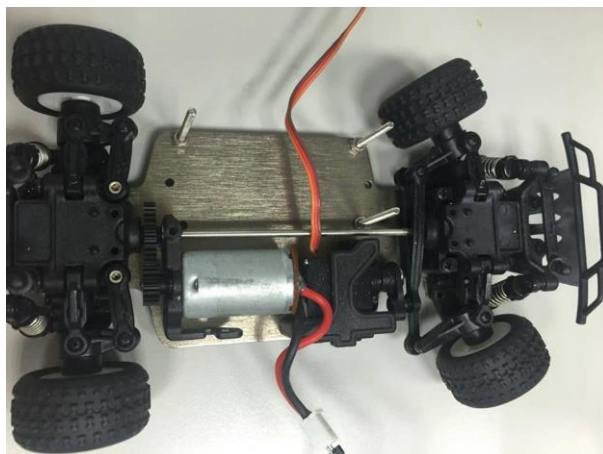
L298N 驅動板



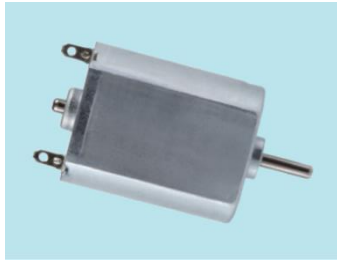
Motor A 控制 IN1 及 IN2 控制邏輯，Motor B 控制 IN3 及 IN4 方法如下圖相同

ENA	IN1	IN2	功能
HIGH	HIGH	LOW	馬達正轉
HIGH	LOW	HIGH	馬達反轉
HIGH	IN1=IN2	IN1=IN2	馬達快速停止
LOW	ingored	ignored	馬達慢速停止

自走車使用 18650 電池 x2 透過 L298 輸出 5v 電壓來驅動馬達(FK-130SH)
(此馬達當電壓超過 7V 時會因保護機制而切斷電流，所以用 L298N 輸出 5V 電壓來驅動)



步進馬達 FK-130SH – 控制前進後退

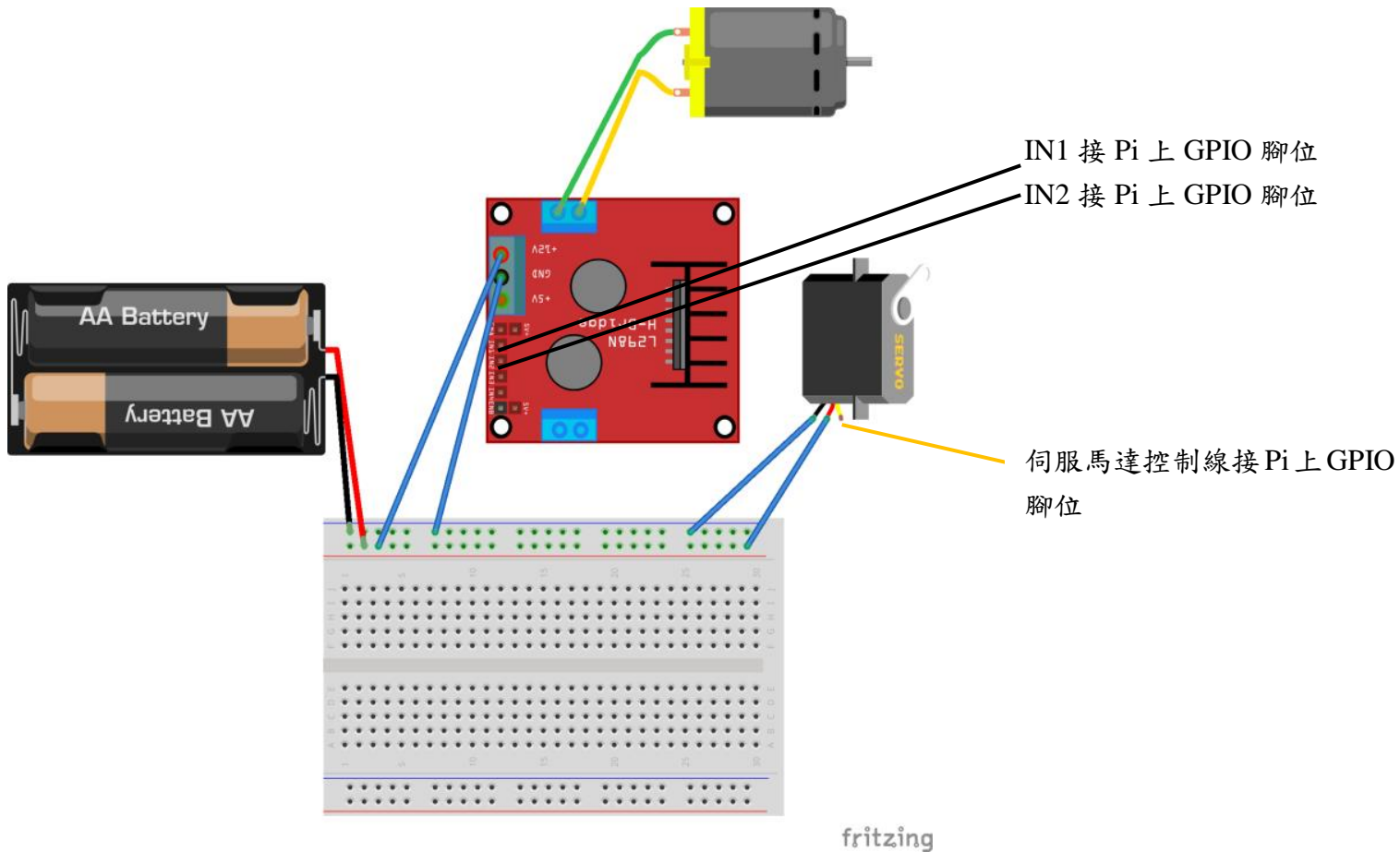


伺服馬達 – 控制左轉右轉



紅色為 VCC 黑色為 GND
橙色為訊號控制線

接線示意圖



WiringPi 程式說明

控制前後左右動作 example

```
13 #include <iostream>
14 #include <wiringPi.h>
15 #include <softPwm.h>
16 #include <stdio.h>
17 #include <stdlib.h>
18
19 #define INITIAL_VALUE 0
20 #define RANGE 500
21
22 using namespace std;
23
24 void forward();
25 void backward();
26 void right();
27 void left();
28 void stop();
29
30 int main() {
31     wiringPiSetup();
32
33     softPwmCreate(2, INITIAL_VALUE, RANGE); //create the softPwm Pin and set the Range for Max Speed
34     softPwmCreate(3, INITIAL_VALUE, RANGE);
35
36     softPwmCreate(12, INITIAL_VALUE, RANGE);
37     forward();
38     backward();
39     right();
40     left();
41     stop();
42 }
43
44 }
```

13~14 行:

將<wiringPi.h>及<softPwm.h> include 進來

若所使用之 GPIO 腳位有提供 Pwm 輸出可不需使用 softPwm library

32 行:

wiringPi library 初始化設定

34-37 行:

透過 softPwmCreate 設定指定的 GPIO 腳位可使用 Pwm 控制


```

46 void forward() {
47     for (int i = 100; i <= 500; i++) { //begin low speed to high speed
48         softPwmWrite(3, i);
49         softPwmWrite(2, 0);
50         delay(10);
51     }
52 }
53
54 void backward() {
55     for (int i = 50; i <= 500; i++) {
56         softPwmWrite(3, 0);
57         softPwmWrite(2, i);
58         delay(10);
59     }
60 }
--

```

給予 GPIO 3(接至 IN1)高電位，GPIO2(接至 IN2)低電位，則可以使車子向前，後退則反之

可自行設定 i 值即可控制馬達快慢

```

void right(){
    for (int i = 0; i <= 500; i++) {
        softPwmWrite(12, i);
        delay(50);
    }
}

void left(){
    for (int i = 500; i <= 0; i--) {
        softPwmWrite(12, i);
        delay(50);
    }
}

```

將 GPIO12 腳位接至伺服馬達控制線

同上，可透過 softPwmWrite 中的 i 值來控制伺服馬達的角度

若 GPIO 腳位有支援 Pwm 輸出

可將

softPwmCreate(Pin, INITIAL_VALUE, RANGE)替換成 pinMode (Pin, PWM_OUTPUT);
 softPwmWrite(12, i);替換成 pwmWrite (1, i);

參考資料

<http://wiringpi.com/>