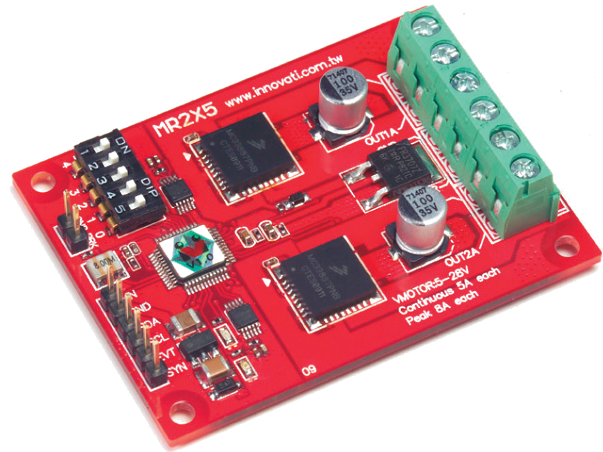


MR 2x5

雙直流馬達控制模組

版本: V2.0

產品介紹: 利基 MR 2x5 模組可以達到透過簡易的指令設定，同時自由操控兩顆直流馬達的需求。可以隨時動態的更改馬達轉速，並取得馬達現在的狀態包含轉速或是方向，也可以透過指令取得電流值。



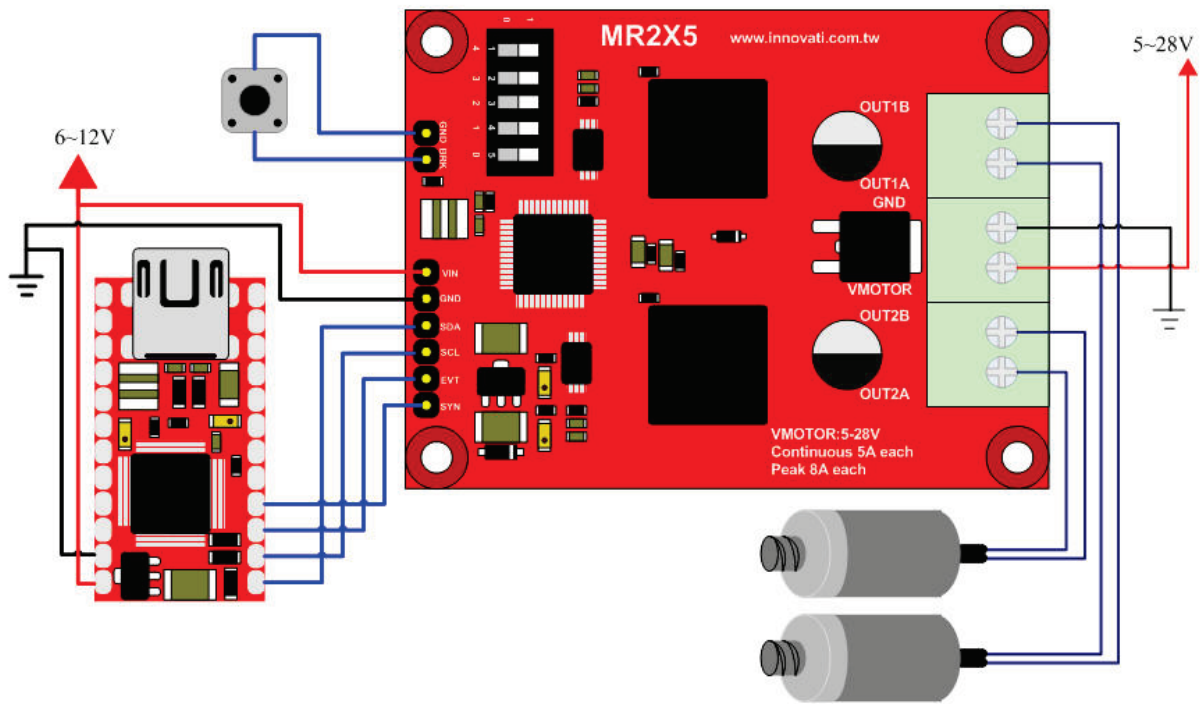
應用方向:

- 控制馬達趨動，設定模型車的前進與後退，經由速度差設定，能做出轉向變化。
- 分別控制不同向的馬達，操作物體能向前後與左右移動，如控制機械手臂。
- 可以直接加上小風扇，並操作風量強度。
- 監控馬達電流，設定過電流中斷馬達動作。

產品特色:

- 可以同時對兩個馬達模組設定命令的執行。
- 可提供範圍 $\pm 5A$ 的連續輸出電流。(Peak 可達 $\pm 7.8A$)
- 正常工作輸入電壓最高可接受至 28 V。(降低效能狀態可以承受至 40V)
- 內部最高 PWM 電流控制頻率可達 8k Hz。
- 提供過熱自動斷電保護($\sim 175^{\circ}C$)。
- 提供過載電流保護。
- 透過 Brake 指令可以快速停止馬達的動作。
- 可提供達 256 階轉速變換。
- 執行指令能讓兩個馬達同時做出不同向不同速的操作。
- 透過指令可輕鬆取得現在馬達轉速或轉向的設定。
- 提供電流讀取指令，並能設定電流過大時的馬達停止機制。
- 可透過 I2C 方式，下達指令。

連接方式: 直接將 ID 開關撥至欲設定的編號，再將 cmdBUS 連接至 BASIC Commander 上對應的腳位，就可透過 BASIC Commander 執行操作。想要操作的直流馬達請如下圖腳位，連接至對應的 OUT1AB 與 OUT2AB 輸入接腳，並將下圖模組右方 VMOTOR 與 GND 腳位，分別連接至馬達所需要的電源，連接時須注意電壓的方向性。圖左方有一個停止按鈕連接示意，如果需要增加外部的停止按鈕，就可如圖連接一個按鈕，一端接到按鈕停止輸入點，另一端接到地端。



產品規格:

停止按鈕連接腳位。將BRK腳位外接一個按鈕，並將按鈕另一端與地端連接，當按鈕按下，就會停止馬達的動作。

模組編號設定開關，由右至左以二進制設定模組的模組編號，編號可以讓BASIC Commander操控時，判斷想要控制的模組。

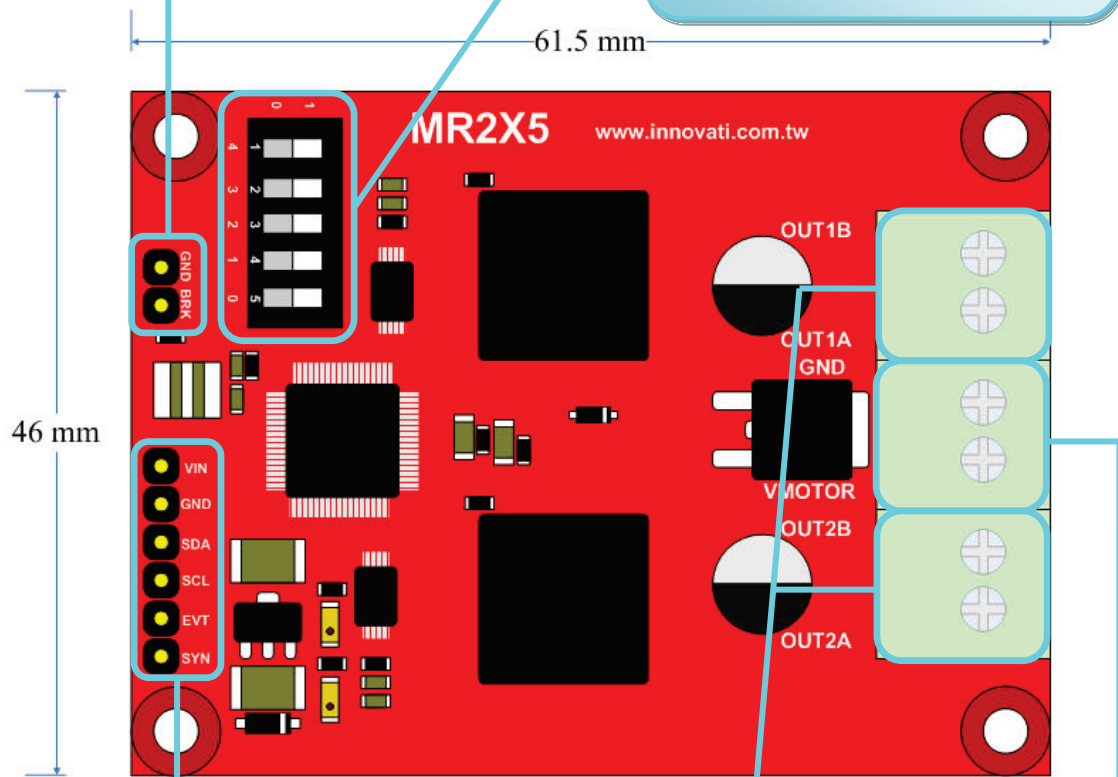


圖 1: 模組腳位與開關介紹

cmdBUS接腳，將此處腳位與BASIC Commander對應腳位相接，即可透過BASIC Commander操控Motor模組(連接時請注意腳位對應，將VIN對接BASIC Commander上的VIN腳位，若是腳位錯誤可能造成模組損毀)

馬達連接孔位，OUT1A與OUT1B請與要控制的第一組馬達連接，OUT2A與OUT2B，則與第二組要控制的馬達相接。

馬達模組電源連接孔位。請將提供馬達電源的接線接到此處。

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{DD}	Conditions				
I _{DD}	Operating Current	7.5	No Load	—	14.7	—	mA
f _{pwm}	PWM Output frequency	—	—	—	8K	—	Hz

表 1: 工作電流特性 (於 25 °C 之環境)

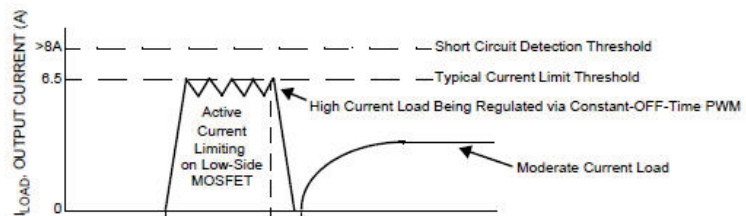
Test Condition: $T_A=25^{\circ}\text{C}$, $V_M=30\text{V}$

Characteristic	Symbol	Limits			
		Min.	Typ.	Max.	Units
Operating Voltage Range	V_+	5	-	28	V
Output ON-Resistance $5.0\text{ V} \leq V_+ \leq 28\text{ V}, T_J = 25^{\circ}\text{C}$ $8.0\text{ V} \leq V_+ \leq 28\text{ V}, T_J = 150^{\circ}\text{C}$ $5.0\text{ V} \leq V_+ \leq 8\text{ V}, T_J = 150^{\circ}\text{C}$	$R_{DS(ON)}$	-	120	-	$\text{m}\Omega$
		-	-	225	
		-	-	300	
Active Current Limiting Threshold (via Internal Constant OFF-Time PWM) on Low-Side MOSFETs	I_{LIM}	5.2	6.5	7.8	A
High-Side Short Circuit Detection Threshold	I_{SCH}	11	-	-	A
Low-Side Short Circuit Detection Threshold	I_{SCL}	8.0	-	-	A
Leaking Current $V_{OUT} = V_+$ $V_{OUT} = \text{Ground}$	$I_{OUT(LEAK)}$	-	100	200	μA
		-	30	60	
Output MOSFET Body Diode Forward Voltage Drop $I_{OUT} = 3.0\text{ A}$	V_F	-	-	2.0	V
Thermal Shutdown Temp.	T_{LIM}	175	-	225	$^{\circ}\text{C}$
Thermal Shutdown Hysteresis.	T_{HYS}	10	-	30	$^{\circ}\text{C}$

表 2: 馬達相關電氣特性

過熱保護動作: 過熱保護電路在感測到驅動 IC 內部溫度到達 175°C 時, 將自動斷路, 此時馬達即停止動作, 當溫度下降約 15°C 後, 保護電路自動回復導通, 馬達就繼續先前的動作。

電流限流保護動作: 請參照下圖, 在 H-bridge 開始輸出時, 電流隨著馬達轉動增加, 當電流值超過 6.5A (如下圖 Typical Current Limit Threshold), 就會停止 H-bridge 的輸出, 直到內部震盪器下一個時脈傳送, 又會開始電流的傳送, 如此反覆, 電流會被固定在如圖的範圍內。



操作注意事項:

馬達模組提供兩組馬達連接腳位, 請確認所連接的馬達為直流馬達。

模組操作溫度 $0^{\circ}\text{C} \sim 70^{\circ}\text{C}$ (馬達之操作溫度請另行確認)

模組儲存溫度 $-50^{\circ}\text{C} \sim 125^{\circ}\text{C}$

模組下達指令的方式可分為兩種：**cmdBUS**、**I2C** 控制方式

cmdBUS 指令表:

下面的指令表是專供控制 MR 2x5 模組的各種指令，必要輸入的指令名稱與參數，以粗底或粗斜體表示，粗體的文字在輸入時請不要更改，粗斜體的文字請自行定義適當格式的參數填入。輸入時請注意 innoBASIC Workshop 大寫與小寫會視為相同字。

在執行 MR 2x5 指令前，請先於程式開頭定義對應參數與編號，例:

Peripheral *ModuleName* As MR2x5 @ *ModuleID*

I2C 通訊協議(Protocol):

為了使更廣泛的使用者能控制模組，提供了部份指令的通訊協議讓使用者應用。透過通訊規格，使用者可使用 I2C 通訊協議為模組下達命令。

通訊協議常見的封包如下：

MID：模組 ID 編號，空間大小為 Byte 的變數。對應於硬體的指播開關。

CID：命令 ID 編號，空間大小為 Byte 的變數。依不同命令而改變。

Checksum1：驗證位元_1，空間大小為 Byte 的變數。

定義方式： $255 - (MID * 2) - CID$

Checksum2：驗證位元_2，空間大小為 Byte 的變數。

定義方式： $255 - (Checksum1 \sim Checksum2 \text{ 之間的變數總和})$

Checksum3：驗證位元_3，空間大小為 Byte 的變數。

定義方式： $255 - MID - (MID \sim Checksum3 \text{ 之間的變數總和})$

Dummy：虛設位元，可為任意變數。空間大小為 Byte 的變數。

於通訊規格**每筆資料空間大小階為 Byte**，若資料空間大小超過一個 Byte 時，需將資料拆開，並由 Low Byte 開始傳送。

Ex: 傳送資料 Temp 為一筆空間大小為 Word 的資料，則需將 Temp 拆開，分為 Temp_L、Temp_H，並且**先傳送 Temp_L**。

Ex1 模組編號為 2，命令編號為 153，傳送參數 Byte 為 100，通訊協議為

MID+CID+Checksum1+Byte+Checksum2+Dummy 則：

MID = 2

CID = 153

Checksum1 = $255 - (2 * 2) - 153 = 98$

Byte = 100

Checksum2 = $255 - 100$

Dummy = 0~255 之間的任意數

Ex2 模組編號為 2，命令編號為 153，傳送參數 Temp 為 511，通訊協議為

MID+CID+Checksum1+Temp_L+Temp_H+Checksum2+Dummy 則：

MID = 2

CID = 153

Checksum1 = $255 - (2 * 2) - 153 = 98$

Temp_L = 255，Temp_H = 1

Checksum2 = $255 - Temp_L - Temp_H = 255$

Dummy = 0~255 之間的任意數

指令格式		指令功能
馬達加速相關指令		
CmdBUS： BackwardA(<i>DutyCycle</i>) I2C： MID+92+Checksum1 +DutyCycle+Checksum2+Dummy	命令 A，B 或 A 與 B 馬達進行向後轉的動作，並且根據 <i>DutyCycle</i> 所給的值，決定馬達的轉速，請輸入 0~255 之間的整數值(<i>DutyCycle</i> 值越高，轉速越快)	
BackwardAB(<i>DutyCycleA</i>, <i>DutyCycleB</i>)		
CmdBUS： BackwardB(<i>DutyCycle</i>) I2C： MID+93+Checksum1 +DutyCycle+Checksum2+Dummy		
BackwardDual(<i>DutyCycleAll</i>)		
CmdBUS： ForwardA(<i>DutyCycle</i>) I2C： MID+88+Checksum1 +DutyCycle+Checksum2+Dummy	命令 A，B 或 A 與 B 馬達進行向前轉的動作，並且根據 <i>DutyCycle</i> 所給的值，決定馬達的轉速，請輸入 0~255 之間的整數值(<i>DutyCycle</i> 值越高，轉速越快)	
ForwardAB(<i>DutyCycleA</i>, <i>DutyCycleB</i>)		
CmdBUS： ForwardB(<i>DutyCycle</i>) I2C： MID+89+Checksum1 +DutyCycle+Checksum2+Dummy		
ForwardDual(<i>DutyCycleAll</i>)		
馬達停止相關指令		
CmdBUS： BrakeA () I2C： MID+99+Checksum1+Dummy	快速停止 A，B 或 A 與 B 馬達模組的動作	
CmdBUS： BrakeB () I2C： MID+100+Checksum1+Dummy		

CmdBUS : BrakeDual()	
I2C : MID+101+Checksum1+Dummy	
CmdBUS : StopA()	
I2C : MID+96+Checksum1+Dummy	
CmdBUS : StopB()	
I2C : MID+97+Checksum1+Dummy	停止 A，B 或 A 與 B 馬達模組的動作
CmdBUS : StopDual()	
I2C : MID+98+Checksum1+Dummy	
設定與狀態相關指令	
GetDCA(DutyCycle)	取得 A 或 B 的馬達轉速，存放於 DutyCycle 參數中 (DutyCycle 值越高，轉速越快)， DutyCycle 會回傳 0~255 之間的整數值
GetDCB(DutyCycle)	
GetDCAB(DutyCycleA, DutyCycleB)	
GetDirA(Dir)	取得 A 或 B 的馬達轉向，存放於 Dir (Dir 為 0 表示向前，為 1 表示向後)
GetDirB(Dir)	
GetDirAB(DirA, DirB)	
SetDCA(DutyCycle)	以 DutyCycle 設定 A，B 或 A 與 B 馬達模組的轉速，請輸入 0~255 之間的整數值 (DutyCycle 值越高，轉速越快)
SetDCAB(DutyCycleA, DutyCycleB)	
SetDCB(DutyCycle)	
SetDCDual(DutyCycleAll)	
SetDirA(Dir)	以 Dir 設定 A，B 或 A 與 B 馬達模組的轉向 (Dir 為 0 表示向前，為 1 表示向後)
CmdBUS : SetDirAB(DirA, DirB)	
I2C : MID+104+Checksum1+DirA+DirB+Checksum2+Dummy	
SetDirB(Dir)	
SetDirDual(DirAll)	

GetVelA(DutyCycle)	取得 A 或 B 的馬達轉速，並以正負表示方向，存放於 DutyCycle 參數中， DutyCycle 絕對值越高，轉速越快，正值代表向前，負值代表向後， DutyCycle 會回傳-255~255 之間的整數值
GetVelB(DutyCycle)	
GetVelAB(DutyCycleA, DutyCycleB)	
SetVelA(DutyCycleA)	以 DutyCycle 設定 A, B 或 A 與 B 馬達模組的轉速，請輸入-255~255 之間的整數值， DutyCycle 絕對值越高，轉速越快，輸入正值代表向前轉動，負值代表向後轉動
SetVelB(DutyCycleB)	
CmdBUS : SetVelAB(DutyCycleA, DutyCycleB)	
I2C : MID+118+Checksum1 +DutyCycleA_L+DutyCycleA_H +DutyCycleB_L+DutyCycleB_H +Checksum2+Dummy	
SetVelDual(DutyCycle)	
CmdBUS : GetCurr(CurrA, CurrB)	
I2C : Out : MID+132+Checksum1+Dummy In : MID+CurrA_L+CurrA_H +CurrB_L+CurrB_H+Checksum3	
CmdBUS : SetCurrMode(Mode, Time, Curr)	以 Mode 設定電流超過時的中斷模式，以 Time 設定啟動該模式，過大電流要維持的時間，以 Curr 設定電流的限制值 有下列四種模式可供設定 0: 關閉電流控制 1: 啟動電流監控，並在超過設定電流值時，回傳 Event 2: 啟動電流監控，並在超過設定電流值時，回傳 Event，並自動對所有馬達執行 Stop 動作 3: 啟動電流監控，並在超過設定電流值時，回傳 Event，並自動對所有馬達執行 Brake 動作 Time 可以輸入 0~255 間的整數，以 100 ms 為單位 Curr 可以輸入 0~1023 間的整數 Mode 預設為 0
I2C : MID+133+Checksum1 +Mode+Time +Curr_L+Curr_H +Checksum2+Dummy	
GetCurrMode(Mode, Time, Curr)	回傳電流監控的模式儲存於 Mode ，設定的時間於 Time ，以及設定的電流值於 Curr ， Mode 會回傳 0~3 之間的整數， Time 會回傳 0~255 間的整數， Curr 會回傳 0~1023 間的整數

停止按鈕相關指令 *1	
CmdBUS : ClrBrakeButStatus() I2C : MID+124+Checksum1+Dummy	將停止按鈕的狀態值清除為 0，在按下停止按鈕後，如果沒有執行清除，就無法重新啟動馬達動作
CmdBUS : bStatus = GetBrakeButStatus() I2C : Out : MID+123+Checksum1+Dummy In : MID+bStatus+Checksum3	回傳停止按鈕現在的狀態 bStatus 0: 沒有按下停止按鈕 1: 停止按鈕被按下，還未執行清除的動作
錯誤狀態檢測相關指令 *2	
CmdBUS : EnFaultStop() I2C : MID+125+Checksum1+Dummy	啟動錯誤停止機制，當錯誤狀態產生，就會自動停止馬達轉動，並且在執行清除錯誤狀態指令前，無法執行啟動馬達等相關指令，預設為不啟動
CmdBUS : DisFaultStop() I2C : MID+126+Checksum1+Dummy	關閉錯誤停止機制，當錯誤狀態產生，系統會自動進行錯誤狀態排除，並繼續馬達的轉動，但若有啟動錯誤事件提醒，仍會回傳錯誤提醒事件，預設為關閉錯誤停止機制
CmdBUS : bStatus = GetFaultStatus() I2C : Out : MID+127+Checksum1+Dummy In : MID+bStatus+Checksum3	取得錯誤狀態儲存於 bStatus 中，在關閉錯誤停止機制下，錯誤狀態會被自動清除為 0，並繼續原先設定的馬達轉動，但若是啟動錯誤停止機制下，一旦有錯誤產生，回傳的錯誤狀態在沒有執行清除動作前，都會維持為 1，此時無法執行轉動馬達的動作
CmdBUS : ClrFaultStatus() I2C : MID+128+Checksum1+Dummy	清除錯誤停止機制的停止狀態，在啟動錯誤停止機制下，一旦發生錯誤狀態，就必須執行此指令，才能重新啟動馬達的轉動
EnFaultEvent()	啟動錯誤狀態提醒事件，預設為啟動
DisFaultEvent()	關閉錯誤狀態提醒事件，預設為關閉
CmdBUS : RestoreStatus() I2C : MID+131+Checksum1+Dummy	回復錯誤狀態發生時的各個設定值

- *1 停止按鈕相關指令，需要在外部加裝停止按鈕後，才能產生效果
- *2 錯誤狀態只有在馬達轉速設定超過 2 時，才會產生相關的錯誤提醒動作
- *3 回傳值是馬達電流分流轉換出的數位轉換電壓值，分流為總電流的 1/375，再經過 220 Ω 電阻值產生的電壓

模組提供應用事件:

事件名稱 (Event)	啟動條件
FaultEvent	當偵測到過熱等狀態造成馬達啟動防護機制，並且在啟動錯誤事件提醒狀態，就會產生此事件
OverCurrAEvent	當設定電流監控模式為 1~3 之間任一模式，馬達 A 的電流值超過設定監控值並維持超過設定時間，就會產生此事件
OverCurrBEvent	當設定電流監控模式為 1~3 之間任一模式，馬達 B 的電流值超過設定監控值並維持超過設定時間，就會產生此事件

範例程式:

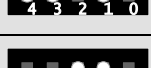
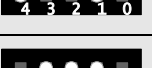
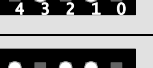
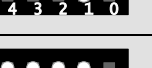
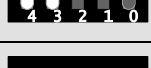
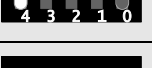
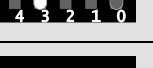
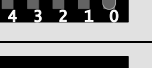
```
Peripheral myMotor As MR2x5 @ 0 ' 設定模組編號為 0

Sub Main()
    Debug CLS
    myMotor.ForwardDual(200) ' 讓兩個操縱馬達同時正向轉動
    Pause 3000
    myMotor.StopDual() ' 停止兩個馬達
    Pause 3000
    myMotor.BackwardDual(200) ' 讓兩個操縱馬達同時反向轉動
    Pause 3000
    myMotor.SetDirDual(0) ' 設定兩個馬達轉向，改為正向
    Pause 3000
    myMotor.SetDCDual(150) ' 將馬達轉速改為 150
    Pause 3000
    myMotor.BrakeDual() ' 快速停止兩個馬達
    Pause 3000
End Sub
```

附錄

1. 已知問題:

2. 模組編號開關對應編號表:

	0		8		16		24
	1		9		17		25
	2		10		18		26
	3		11		19		27
	4		12		20		28
	5		13		21		29
	6		14		22		30
	7		15		23		31