

Servo Runner 8

八組伺服機輸出控制模組

版本: V1.0

產品介紹: 利基 Servo Runner 8 模組可以一次控制八個伺服機，並且提供整合好的指令，讓使用者可以直接使用速度或時間，決定伺服機的移動效率。



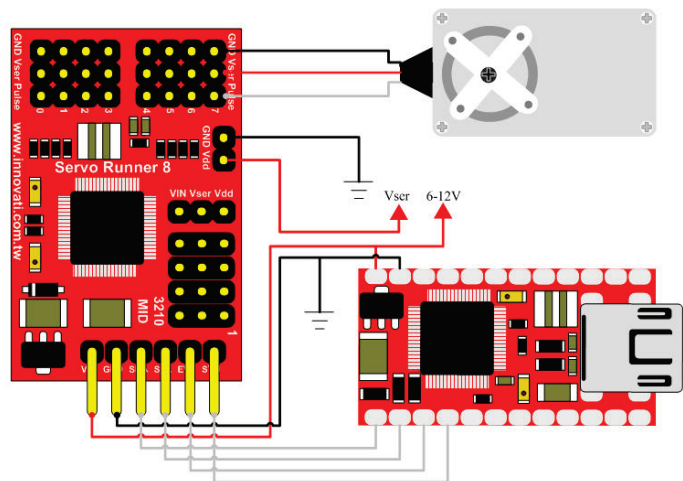
應用方向:

- 各種伺服機的操作與應用，包括機械手臂，機器人關節。
-

產品特色:

- 八組伺服機輸出介面，可同時控制八組伺服機。
- 可控制伺服機位置由 0.5 ms 至 2.5 ms。
- 軟體微調指令，不用機械拆裝，僅由軟體設定，就可以達到微調各個伺服機轉向角度的目的，可設定-128~127 μ S。
- 程式可以設定伺服機轉向速度，使用者可根據需求設定多段的伺服機轉向速度。
- 使用者可以設定一個共同時間，讓各個伺服機在同時間達到不同的轉向角度。
- 設有四組事件提醒，讓使用者可以在判斷動作完成後，自動進行下一項操作。各事件可以設定任意 1~8 個伺服機作為判斷依據。
- 各種狀態取得指令，使用者可以隨時確認伺服機是否動作完成，取得現在位置，目標位置，微調參數，或是設定的時間與速度值。
- 解析度可達 2 μ S。

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。在模組上有以三個腳位為一組，共八組伺服機連接座，提供各伺服機的控制訊號與電源，根據伺服機的對應腳位連接就可以動作(如右圖)，另外提供給伺服機的電源請在圖示中的電源輸入位置外加，需確定伺服機所需的電流與電壓，以免伺服機產生不正常動作，造成伺服機損毀。



產品規格:

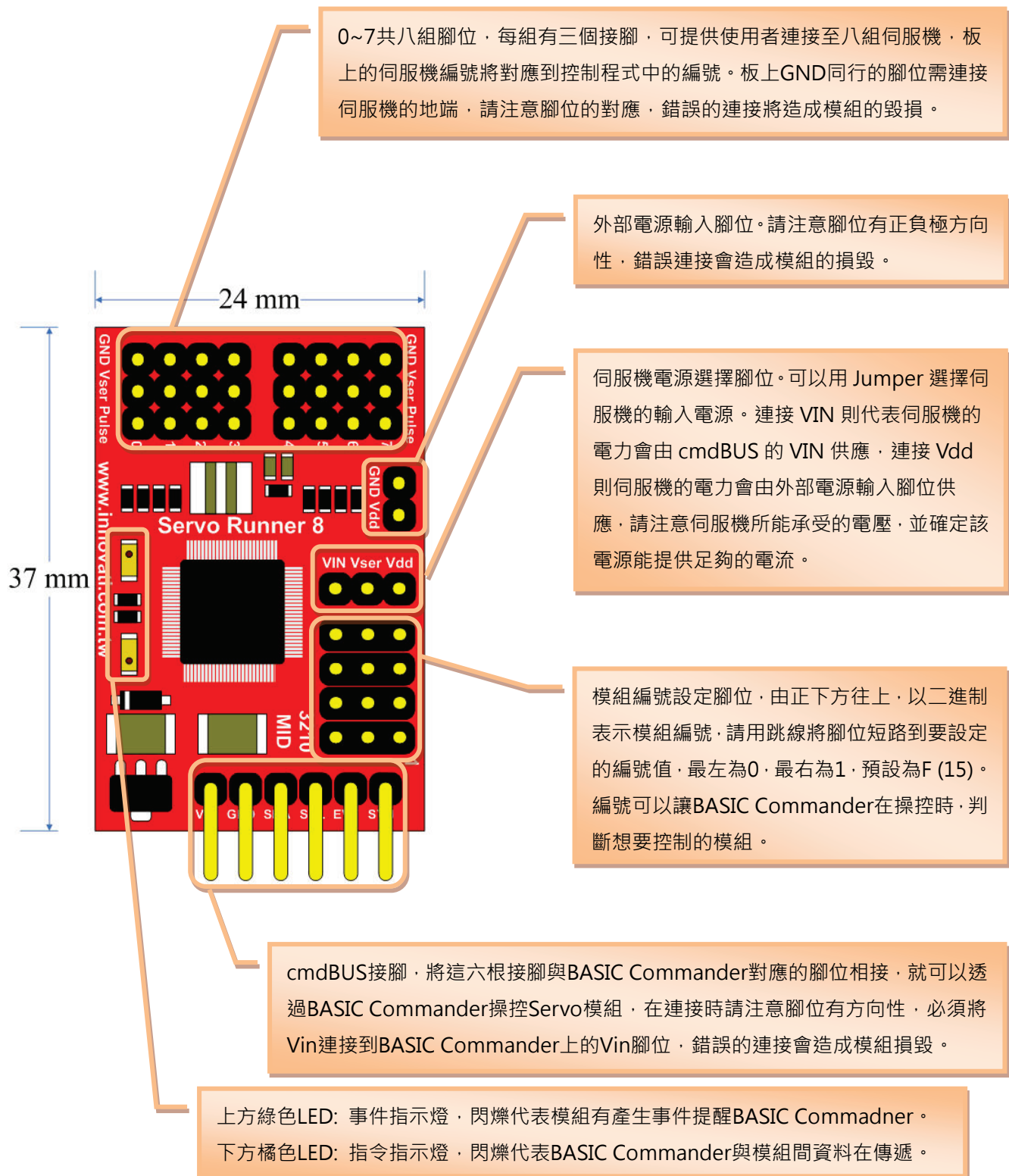


圖 1: 模組腳位與開關介紹

使用電流量: 7 mA (Servo Runner 8 模組未接上伺服機於 cmdBUS 之耗電量)

操作注意事項: 請確認所連接之伺服機所需之電壓範圍，與所需電流大小，選擇合適之電源，由 Vser 連接正確之電源。

伺服機的 Pulse 腳位與模組連接須符合表 1 規範→(此為模組所能動作之範圍)

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{IN} =7.5V	Conditions				
V _{OH}	I/O Port output high voltage	-	No loading	-	5	-	V
V _{OL}	I/O Port output low voltage	-	No loading	-	0	-	V
I _{OL}	I/O Port Sink Current	-	V _{load} =0.1V _{OH}	10	20	-	mA
I _{OH}	I/O Port Source Current	-	V _{load} =0.9V _{OH}	-5	-10	-	mA

表 1: ServoRunnerA 模組電流限制 (Test Temperature=25℃)

模組操作溫度 0℃~70℃ (伺服機之操作溫度請於伺服機規格確認)
 模組儲存溫度 -50℃~125℃

指令表:

下面的指令表是專供控制 Servo Runner 8 模組的各種指令，必要輸入的指令名稱與參數，以粗體或粗斜體表示，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 Servo Runner 8 指令前，請先於程式開頭定義對應參數與編號，例:

Peripheral ModuleName As ServoRunner8 @ 0

指令格式	指令功能
伺服機位置設定相關指令	
SetPos (SerID, Pos)	以 SerID 指定所要操作的伺服機，設定該伺服機要到達的位置為 Pos ，若設定 Run 則立刻開始動作， SerID 請輸入 0~7 之間的整數值， Pos 請輸入 499~2500 之間的整數值，單位為 μS，若設定超過此範圍的值，這次設定的命令將不被執行
SetPosAndRun(SerID, Pos)	
SetPosSpd(SerID, Pos, Spd)	以 SerID 指定所要操作的伺服機，設定伺服機要到達的位置為 Pos ，同時需要以 Spd 為速度值移動伺服機，若設定 Run 則立刻開始動作， SerID 請輸入 0~7 之間的整數值， Pos 請輸入 499~2500 之間的整數值， Spd 請輸入 0~65535 之間的整數值，0 代表全速，值越大移動越快，單位為 μS/S
SetPosSpdAndRun(SerID, Pos, Spd)	
SetPosTime(SerID, Pos, Time)	以 SerID 指定所要操作的伺服機，設定伺服機要到達的位置為 Pos ，同時伺服機需要以固定速度，花 Time 所設定的時間到達指定位置， SerID
SetPosTimeAndRun(SerID, Pos, Time)	

	請輸入 0~7 之間的整數值， Pos 請輸入 499~2500 之間的整數值， Time 請輸入 0~65535 之間的整數值， Time 若設為 0 則會以全速移動。 Time 的單位為毫秒[ms]，如果設定時間太短，伺服機就以全速移動
伺服機啟動相關指令	
Run1Servo(SerID)	根據設定的 SerID ，啟動該伺服機執行設定的動作，如果啟動的伺服機，只設定位置，而沒有設定速度或時間，伺服機會用最快的速度動作。 SerID 請輸入 0~7 之間的整數值
...	
Run7Servo(SerID1 , ..., SerID7)	
RunAllServo()	
Run1ServoWithEventA(SerID)	根據設定的 SerID ，啟動該伺服機執行設定的動作，並且在動作完成後，產生事件 A。 SerID 請輸入 0~7 之間的整數值
...	
Run7ServoWithEventA(SerID1 , ..., SerID7)	
RunAllServoWithEventA()	
Run1ServoWithEventB(SerID)	根據設定的 SerID ，啟動該伺服機執行設定的動作，並且在動作完成後，產生事件 B。 SerID 請輸入 0~7 之間的整數值
...	
Run7ServoWithEventB(SerID1 , ..., SerID7)	
RunAllServoWithEventB()	
Run1ServoWithEventC(SerID)	根據設定的 SerID ，啟動該伺服機執行設定的動作，並且在動作完成後，產生事件 C。 SerID 請輸入 0~7 之間的整數值
...	
Run7ServoWithEventC(SerID1 , ..., SerID7)	
RunAllServoWithEventC()	
Run1ServoWithEventD(SerID)	根據設定的 SerID ，啟動該伺服機執行設定的動作，並且在動作完成後，產生事件 D。 SerID 請輸入 0~7 之間的整數值
...	
Run7ServoWithEventD(SerID1 , ..., SerID7)	
RunAllServoWithEventD()	
伺服機停止相關指令	
Pause1Servo(SerID)	根據設定的 SerID ，暫停該伺服機正在執行的動作， SerID 請輸入 0~7 之間的整數值
...	
Pause7Servo(SerID1 , ..., SerID7)	
PauseAllServo()	
Stop1Servo (SerID)	根據設定的 SerID ，停止該伺服機正在執行的動作，同時停止供應給伺服機電流，此時伺服機若受外力移動，將改變位置， SerID 請輸入 0~7 之間的整數值
...	
Stop7Servo (SerID1 , ..., SerID7)	
StopAllServo ()	
伺服機狀態與記憶相關指令	

Get1ServoReadyStatus(<i>SerID</i>, <i>Status</i>)	取得 <i>SerID</i> 指定伺服機的狀態，並將狀態值存放於 <i>Status</i> 中，如果所指定的伺服機中，有任一個還沒有到達指定位置，所傳回的狀態將會是 0，當所有伺服機皆到達指定位置，傳回狀態則為 1， <i>SerID</i> 請輸入 0~7 之間的整數值
...	
Get7ServoReadyStatus(<i>SerID1</i>, ..., <i>SerID7</i>, <i>Status</i>)	
GetAllServoReadyStatus(<i>Status</i>)	
GetNowPos (<i>SerID</i>, <i>Pos</i>)	取得 <i>SerID</i> 指定伺服機的現在位置，存放於 <i>Pos</i> 中， <i>SerID</i> 請輸入 0~7 之間的整數值， <i>Pos</i> 會回傳 499~2500 之間的整數值
GetPos(<i>SerID</i>, <i>Pos</i>)	取得 <i>SerID</i> 指定伺服機的目標位置，存放於 <i>Pos</i> 中， <i>SerID</i> 請輸入 0~7 之間的整數值， <i>Pos</i> 會回傳 499~2500 之間的整數值
GetPosOffset(<i>SerID</i>, <i>Offset</i>)	取得 <i>SerID</i> 指定伺服機的微調值，存放於 <i>Offset</i> 中， <i>SerID</i> 請輸入 0~7 之間的整數值， <i>Offset</i> 單位為微秒[μS]，範圍可從-128 到 127
GetSpdAndTime(<i>SerID</i>, <i>Type</i>, <i>Value</i>)	取得 <i>SerID</i> 指定伺服機設定的移動方式 <i>Type</i> ，與設定值 <i>Value</i> ， <i>SerID</i> 請輸入 0~7 之間的整數值，如果設定的移動方式為速度，則 <i>Type</i> 回傳值為 1，若是時間則為 0， <i>Value</i> 會回傳 0~65535 之間的整數值
SetPosOffset(<i>SerID</i>, <i>Offset</i>)	設定 <i>SerID</i> 指定伺服機的微調值為 <i>Offset</i> ， <i>SerID</i> 請輸入 0~7 之間的整數值， <i>Offset</i> 請輸入-128~127 之間的整數值

模組提供應用事件:

事件名稱 (Event)	啟動條件
ServoPosReadyEventA	執行 Run1ServoWithEventA 類別的指令(1 可為 1~7 或 All)，當所有指定伺服機都到達指定位置，就會啟動此事件
ServoPosReadyEventB	執行 Run1ServoWithEventB 類別的指令(1 可為 1~7 或 All)，當所有指定伺服機都到達指定位置，就會啟動此事件
ServoPosReadyEventC	執行 Run1ServoWithEventC 類別的指令(1 可為 1~7 或 All)，當所有指定伺服機都到達指定位置，就會啟動此事件
ServoPosReadyEventD	執行 Run1ServoWithEventD 類別的指令(1 可為 1~7 或 All)，當所有指定伺服機都到達指定位置，就會啟動此事件

範例程式:

```
'      範例程式中的伺服機位置是以多數伺服機的範圍設定，
'      請根據所使用的伺服機可設定的位置做調整，以免造成伺服機毀損

Peripheral mySer As ServoRunner8 @ 15      '      設定模組編號為 15

Dim EventEnd As Byte      '      儲存事件處理完成的判斷參數
Dim i As Byte      '      儲存迴圈的判斷值
Dim SerStatus As Byte      '      儲存 Servo 的 Status 值

Sub Main()      '      主程式
    mySer.SetPosOffset(0, 0)      '      設定 Servo0 的微調值為 0
    mySer.SetPosAndRun(0, 1500)      '      啟動 Servo0 移動到 1500 的位置
    Pause 1000      '      暫停一段時間讓伺服機移動到指定位置

    mySer.SetPos(0, 2200)      '      指定 Servo0 目標位置為 2200
    mySer.Run1Servo(0)      '      讓 Servo0 開始動作
    Pause 500

    mySer.SetPosSpdAndRun(0, 700, 1000)      '      啟動 Servo0 以速度 1000 移動到 700 的位置
    Pause 2000
    mySer.SetPosTimeAndRun(0, 2200, 1000)      '      啟動 Servo0 花一秒的時間移動到 2200 的位置
    Pause 1000

    EventEnd=0
    mySer.SetPosTime(0, 700, 1000)      '      設定 Servo0 花一秒的時間移動到 700 的位置
    mySer.Run1ServoWithEventA(0)      '      啟動 Servo0 並於動作結束產生 EventA

    Do
        Pause 1
    Loop Until EventEnd=1

    mySer.SetPosAndRun(0, 1500)
'      下面的迴圈反覆執行讀取 Status 的動作，
'      在確定動作結束後，才會跳出迴圈
    Do
        mySer.Get1ServoReadyStatus(0, SerStatus)      '      讀取 Servo0 的狀態存於 SerStatus 中
    Loop Until SerStatus>0

End Sub

Event mySer.ServoPosReadyEventA()
    mySer.SetPosAndRun(0, 2200)
    Pause 1000
    EventEnd=1
End Event
```

附錄

1. 已知問題:

※版本標示於模組上的雷射貼紙