

Servo Runner A

十六組伺服機輸出控制模組

版本: V2.0



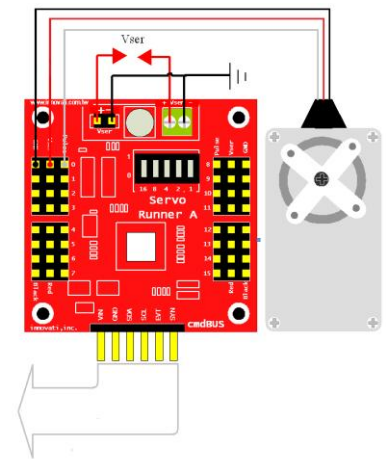
產品介紹: 利基 Servo Runner A 模組可以一次控制十六個伺服機，並且提供整合好的指令，讓使用者可以直接使用速度或時間，決定伺服機的移動模式。設有多達 250 組記憶體可以儲存伺服機目標位置與移動方式(速度或時間)，讓各種動作輕易組合完成。

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:

- 十六組伺服機輸出介面，可同時控制十六組伺服機。
- 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。
- 內建 250 組伺服機記憶空間，每組可以儲存目前設定好的十六個伺服機目標位置，與速度或時間參數，在需要時直接呼叫，可以免去重覆設定的動作，也可以快速組合出多樣化的效果。
- 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。
- 可透過 cmdBUS 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 Arminno 上對應的腳位，就可透過 Arminno 執行操作。在模組上有以三個腳位為一組，共十六組伺服機連接座，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:

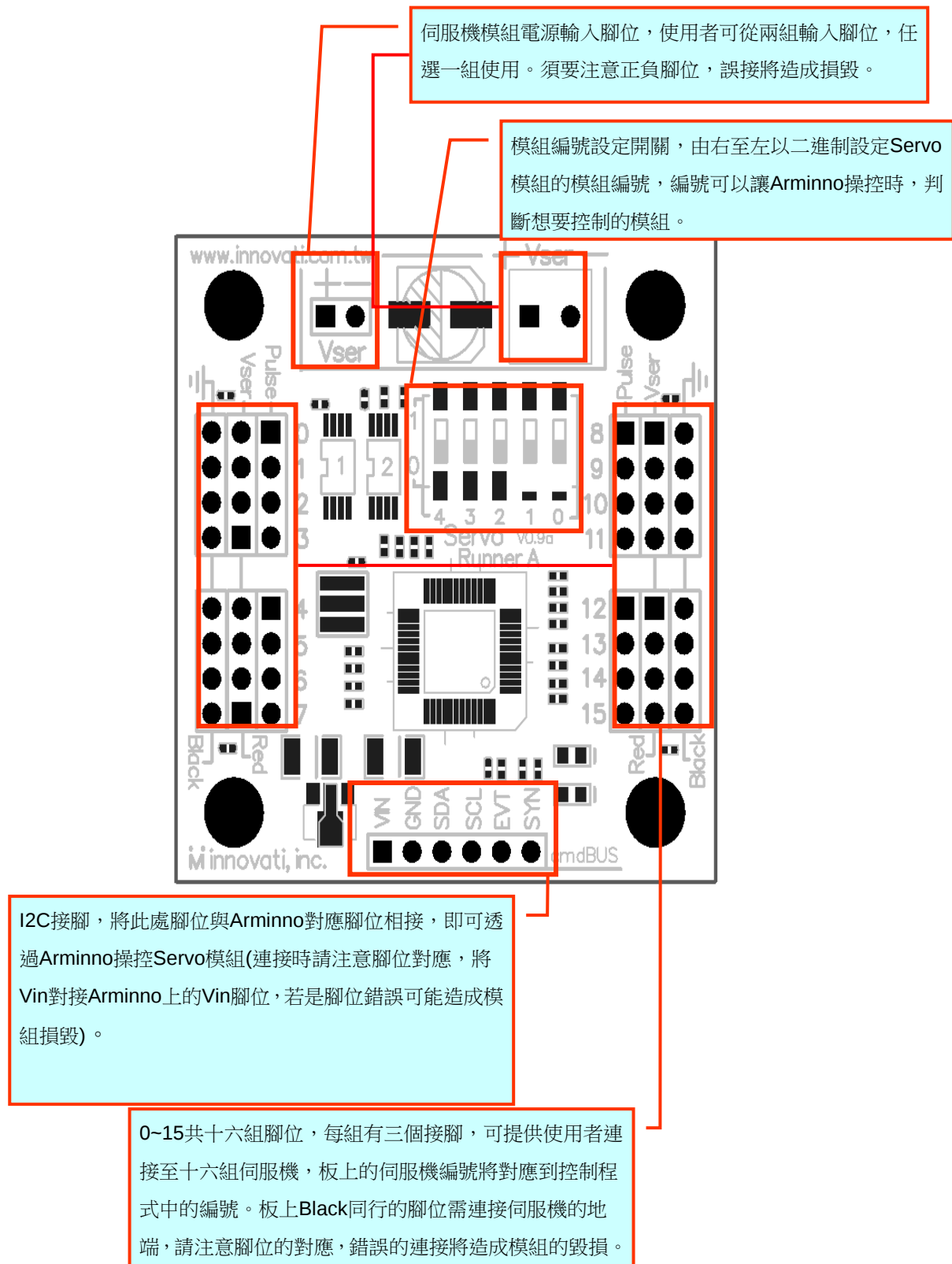


圖 1: 模組腳位與開關介紹

使用電流量: 7 mA (ServoRunnerA 模組未接上伺服機於 I2C 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°C)

模組操作溫度 0 °C ~ 70 °C (伺服機之操作溫度請於伺服機規格確認)

模組儲存溫度 -50 °C ~ 125 °C

指令格式	指令功能
伺服機位置設定相關指令	
SetPos (uint8_t SerID, uint16_t Pos)	以 SerID 指定所要操作的伺服機，請輸入 0~15 之間的整數值，設定該伺服機所要到達的位置為 Pos ，若設定 Run 則立刻開始動作 (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(uint8_t SerID, uint16_t Pos)	
SetPosSpd(uint8_t SerID, uint16_t Pos, uint16_t Spd)	以 SerID 指定所要操作的伺服機，請輸入 0~15 之間的整數值，設定伺服機所要到達的位置為 Pos ， Pos 請輸入 499~2500 之間的整數值，並且設定伺服機以 Spd 移動伺服機，若設定 Run 則立刻開始動作(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(uint8_t SerID, uint16_t Pos, uint16_t Spd)	
SetPosTime(uint8_t SerID, uint16_t Pos, uint16_t Time)	以 SerID 指定所要操作的伺服機，設定伺服機所要到達的位置為 Pos ， Pos 請輸入 499~2500 之間的整數值，並且設定伺服機需要以固定速度，花 Time 所設定的時間到達指定位置，若設定 Time 則立刻開始動作(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)
SetPosTimeAndRun(uint8_t SerID, uint16_t Pos, uint16_t Time)	
伺服機啟動相關指令	
Run1Servo(uint8_t SerID)	根據所設定的 SerID ，請輸入 0~15 之間

Run2Servo(uint8_t SerID, uint8_t SerID1)	的整數值，啟動各伺服機執行所設定的動作，若起動伺服機時，只進行位置的設定，而未設定速度或時間，伺服機會用最快速度動作
Run3Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2)	
Run4Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2, uint8_t SerID3)	
Run5Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2, uint8_t SerID3, uint8_t SerID4)	
Run6Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2, uint8_t SerID3, uint8_t SerID4, uint8_t SerID5)	
Run7Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2, uint8_t SerID3, uint8_t SerID4, uint8_t SerID5, uint8_t SerID6)	
Run8Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2, uint8_t SerID3, uint8_t SerID4, uint8_t SerID5, uint8_t SerID6, uint8_t SerID7)	
Run9Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2, uint8_t SerID3, uint8_t SerID4, uint8_t SerID5, uint8_t SerID6, uint8_t SerID7, uint8_t SerID8)	
Run10Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2, uint8_t SerID3, uint8_t SerID4, uint8_t SerID5, uint8_t SerID6, uint8_t SerID7, uint8_t SerID8, uint8_t SerID9)	
Run11Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2, uint8_t SerID3, uint8_t SerID4, uint8_t SerID5, uint8_t SerID6, uint8_t SerID7, uint8_t SerID8, uint8_t SerID9, uint8_t SerID10)	
Run12Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2, uint8_t SerID3, uint8_t SerID4, uint8_t SerID5, uint8_t SerID6)	

, uint8_t <i>SerID7</i>, uint8_t <i>SerID8</i>, uint8_t <i>SerID9</i>, uint8_t <i>SerID10</i>) Run13Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t <i>SerID5</i>, uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i>, uint8_t <i>SerID8</i>, uint8_t <i>SerID9</i>, uint8_t <i>SerID10</i> uint8_t <i>SerID11</i>, uint8_t <i>SerID12</i>) Run14Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t <i>SerID5</i>, uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i>, uint8_t <i>SerID8</i>, uint8_t <i>SerID9</i>, uint8_t <i>SerID10</i> uint8_t <i>SerID11</i>, uint8_t <i>SerID12</i>, uint8_t <i>SerID13</i>) Run15Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t <i>SerID5</i>, uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i>, uint8_t <i>SerID8</i>, uint8_t <i>SerID9</i>, uint8_t <i>SerID10</i> uint8_t <i>SerID11</i>, uint8_t <i>SerID12</i>, uint8_t <i>SerID13</i>, uint8_t <i>SerID14</i>) RunAllServo(void)	
伺服機停止相關指令	
Pause1Servo(uint8_t <i>SerID</i>) Pause2Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>) Pause3Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i>) Pause4Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i>, uint8_t <i>SerID3</i>) Pause5Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> , uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>) Pause6Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t <i>SerID5</i>)	根據所設定的 <i>SerID</i>，請輸入 0~15 之間的整數值，暫停各伺服機正在執行的動作

Pause7Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t <i>SerID5</i>, uint8_t <i>SerID6</i>)	
Pause8Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t <i>SerID5</i>, uint8_t <i>SerID6</i>, uint8_t <i>SerID7</i>)	
Pause9Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t <i>SerID5</i>, uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i>, uint8_t <i>SerID8</i>)	
Pause10Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t <i>SerID5</i>, uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i>, uint8_t <i>SerID8</i>, uint8_t <i>SerID9</i>)	
Pause11Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t <i>SerID5</i>, uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i>, uint8_t <i>SerID8</i>, uint8_t <i>SerID9</i>, uint8_t <i>SerID10</i>)	
Pause12Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t <i>SerID5</i>, uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i>, uint8_t <i>SerID8</i>, uint8_t <i>SerID9</i>, uint8_t <i>SerID10</i>)	
Pause13Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t <i>SerID5</i>, uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i>, uint8_t <i>SerID8</i>, uint8_t <i>SerID9</i>, uint8_t <i>SerID10</i> uint8_t <i>SerID11</i>, uint8_t <i>SerID12</i>)	
Pause14Servo(uint8_t <i>SerID</i>, uint8_t <i>SerID1</i>, uint8_t <i>SerID2</i> uint8_t <i>SerID3</i>, uint8_t <i>SerID4</i>, uint8_t	

<i>SerID5, uint8_t SerID6 , uint8_t SerID7, uint8_t SerID8, uint8_t SerID9, uint8_t SerID10 uint8_t SerID11, uint8_t SerID12, uint8_t SerID13)</i>	
<i>Pause15Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2 uint8_t SerID3, uint8_t SerID4, uint8_t SerID5, uint8_t SerID6 , uint8_t SerID7, uint8_t SerID8, uint8_t SerID9, uint8_t SerID10 uint8_t SerID11, uint8_t SerID12, uint8_t SerID13, uint8_t SerID14)</i>	
<i>PauseAllServo(void)</i>	
<i>Stop1Servo(uint8_t SerID)</i>	根據所設定的 <i>SerID</i>，請輸入 0~15 之間的整數值，停止各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
<i>Stop2Servo(uint8_t SerID, uint8_t SerID1)</i>	
<i>Stop3Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2)</i>	
<i>Stop4Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2, uint8_t SerID3)</i>	
<i>Stop5Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2 , uint8_t SerID3, uint8_t SerID4)</i>	
<i>Stop6Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2 uint8_t SerID3, uint8_t SerID4, uint8_t SerID5)</i>	
<i>Stop7Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2 uint8_t SerID3, uint8_t SerID4, uint8_t SerID5, uint8_t SerID6)</i>	
<i>Stop8Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2 uint8_t SerID3, uint8_t SerID4, uint8_t SerID5, uint8_t SerID6, uint8_t SerID7)</i>	
<i>Stop9Servo(uint8_t SerID, uint8_t SerID1, uint8_t SerID2 uint8_t SerID3, uint8_t SerID4, uint8_t SerID5, uint8_t SerID6)</i>	

, uint8_t <i>SerID7</i> , uint8_t <i>SerID8</i>)	
Stop10Servo(uint8_t <i>SerID</i> , uint8_t <i>SerID1</i> , uint8_t <i>SerID2</i> uint8_t <i>SerID3</i> , uint8_t <i>SerID4</i> , uint8_t <i>SerID5</i> , uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i> , uint8_t <i>SerID8</i> , uint8_t <i>SerID9</i>)	
Stop11Servo(uint8_t <i>SerID</i> , uint8_t <i>SerID1</i> , uint8_t <i>SerID2</i> uint8_t <i>SerID3</i> , uint8_t <i>SerID4</i> , uint8_t <i>SerID5</i> , uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i> , uint8_t <i>SerID8</i> , uint8_t <i>SerID9</i> , uint8_t <i>SerID10</i>)	
Stop12Servo(uint8_t <i>SerID</i> , uint8_t <i>SerID1</i> , uint8_t <i>SerID2</i> uint8_t <i>SerID3</i> , uint8_t <i>SerID4</i> , uint8_t <i>SerID5</i> , uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i> , uint8_t <i>SerID8</i> , uint8_t <i>SerID9</i> , uint8_t <i>SerID10</i>)	
Stop13Servo(uint8_t <i>SerID</i> , uint8_t <i>SerID1</i> , uint8_t <i>SerID2</i> uint8_t <i>SerID3</i> , uint8_t <i>SerID4</i> , uint8_t <i>SerID5</i> , uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i> , uint8_t <i>SerID8</i> , uint8_t <i>SerID9</i> , uint8_t <i>SerID10</i> uint8_t <i>SerID11</i> , uint8_t <i>SerID12</i>)	
Stop14Servo(uint8_t <i>SerID</i> , uint8_t <i>SerID1</i> , uint8_t <i>SerID2</i> uint8_t <i>SerID3</i> , uint8_t <i>SerID4</i> , uint8_t <i>SerID5</i> , uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i> , uint8_t <i>SerID8</i> , uint8_t <i>SerID9</i> , uint8_t <i>SerID10</i> uint8_t <i>SerID11</i> , uint8_t <i>SerID12</i> , uint8_t <i>SerID13</i>)	
Stop15Servo(uint8_t <i>SerID</i> , uint8_t <i>SerID1</i> , uint8_t <i>SerID2</i> uint8_t <i>SerID3</i> , uint8_t <i>SerID4</i> , uint8_t <i>SerID5</i> , uint8_t <i>SerID6</i> , uint8_t <i>SerID7</i> , uint8_t <i>SerID8</i> , uint8_t <i>SerID9</i> , uint8_t <i>SerID10</i> uint8_t <i>SerID11</i> , uint8_t <i>SerID12</i> , uint8_t	

<i>SerID13, uint8_t SerID14)</i>	
StopAllServo(void)	
伺服機狀態與記憶相關指令	
LoadFrame(<i>FrameID</i>)	讀取 <i>FrameID</i> 所指定記憶位址的紀錄值，作為現在伺服機的目標位置與移動方式， <i>FrameID</i> 請輸入 0~249 之間的整數值
LoadOffset()	將 EEPROM 中儲存的 Offset 值，取代現有的設定值
SaveFrame(<i>FrameID</i>)	將目前所設定的各伺服機位置，儲存於 <i>FrameID</i> 所指定的記憶體中， <i>FrameID</i> 請輸入 0~249 之間的整數值
SaveOffset()	將現在設定的 Offset 值，儲存到 EEPROM 中
SetPosOffset(<i>SerID, Offset</i>)	設定 <i>SerID</i> ，請輸入 0~15 之間的整數值，所指定的伺服機微調值為 <i>Offset</i> ， <i>Offset</i> 請輸入 -128~127 之間的整數值

範例程式:

```
#include "arminno.h"

// 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

ServoRunnerA mySer(0);           // 設定模組編號為 0

uint8_t i;                        // 儲存迴圈的判斷值
uint8_t SerStatus;                // 儲存 Servo 的 Status 值

int main(void)
{
    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(10000);                 // 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.SaveFrame(0);            // 將目前設定的伺服機動作存到 Frame0 中
    mySer.Run1Servo(0);            // 讓 Servo0 開始動作
    Pause (5000);

    mySer.SetPosSpdAndRun(0, 700, 1000); // 啟動 Servo0 以速度 1000 移動到 700 的位置
    Pause (20000);
}
```

```

mySer.SetPosTimeAndRun(0, 2200, 1000);           // 啟動 Servo0 花一秒的時間移動到 2200 的位置
Pause (10000);

// 下面的迴圈反覆讀取 Frame0 與 Frame1 的設定值，再啟動 Servo0 執行
// Frame0 儲存的位置為 2200，Frame1 儲存的位置為 700
// Servo0 將在這兩個位置間來回移動四次
for( i=0 ; i < 4; i++)
{
    mySer.LoadFrame(1);                           // 讀取儲存於 Frame1 的設定值
    mySer.Run1Servo(0);
    Pause (10000);
    mySer.LoadFrame(0);                           // 讀取儲存於 Frame0 的設定值
    mySer.Run1Servo(0);
    Pause (10000);
}
mySer.SetPosAndRun(0, 1500);
Pause (20000);

}

```

附錄

1. 已知問題:

- V1.0 版本用 **GetPos** 取得目標位置，須在設定位置後，相隔 1 毫秒以上，否則可能取得錯誤位置。
- V1.0 版本使用 **SaveFrame** 儲存位置時，所設定的位置需為偶數，若為奇數則取回值將出現錯誤。
- V1.0 版本連續執行 **SaveFrame** 作儲存動作時，各儲存動作間須有 5 毫秒以上之間隔，否則可能無法正確完成所有的儲存命令。
- V1.1 之前版本僅能儲存 120 組動作。
- V1.1 之前版本沒有提供 **SaveOffset** 與 **LoadOffset** 指令。
- V1.1 之前版本不會將 **Offset** 值存在 **EEPROM** 中。
- V1.1 之前版本在讀取儲存的動作值時，會有讀取到錯誤值的現象。

※版本標示於模組上的雷射貼紙

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