



IMP Series

運動控制函式庫

整合測試環境使用手冊

版本：V.3.00

日期：2019.09

<http://www.epcio.com.tw>



目 錄

1. 運動控制函式庫測試軟體簡介	2
2. 啟動測試軟體.....	4
3. 運動屬性設定.....	6
4. 原點復歸運動屬性設定.....	10
5. 執行運動命令.....	11
6. JOG 運動	13
7. 原點復歸運動.....	14
8. 運動狀態與訊息顯示.....	15
9. Asynchronous Remote I/O 測試	18



1. 運動控制函式庫測試軟體簡介

運動控制函式庫測試軟體可用於測試單張 IMP Series 運動控制平台，並使用單一 Group (有關 Group 的說明請參考”IMP Series 運動控制函式庫使用手冊”)，Group 參數設定如下：

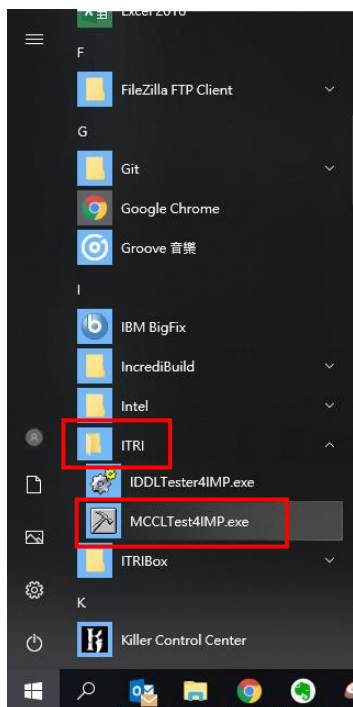
```
m_nGroupIndex = MCC_CreateGroup(  
    0, // X 軸規劃結果由 Channel 0 輸出  
    1, // Y 軸規劃結果由 Channel 1 輸出  
    2, // Z 軸規劃結果由 Channel 2 輸出  
    3, // U 軸規劃結果由 Channel 3 輸出  
    4, // V 軸規劃結果由 Channel 4 輸出  
    5, // W 軸規劃結果由 Channel 5 輸出  
    6, // A 軸規劃結果由 Channel 6 輸出  
    7, // B 軸規劃結果由 Channel 7 輸出  
    0); // 此 Group 對應之運動控制平台編號
```

因此程式中所使用的函式如需輸入 Group 編號，則一律給定

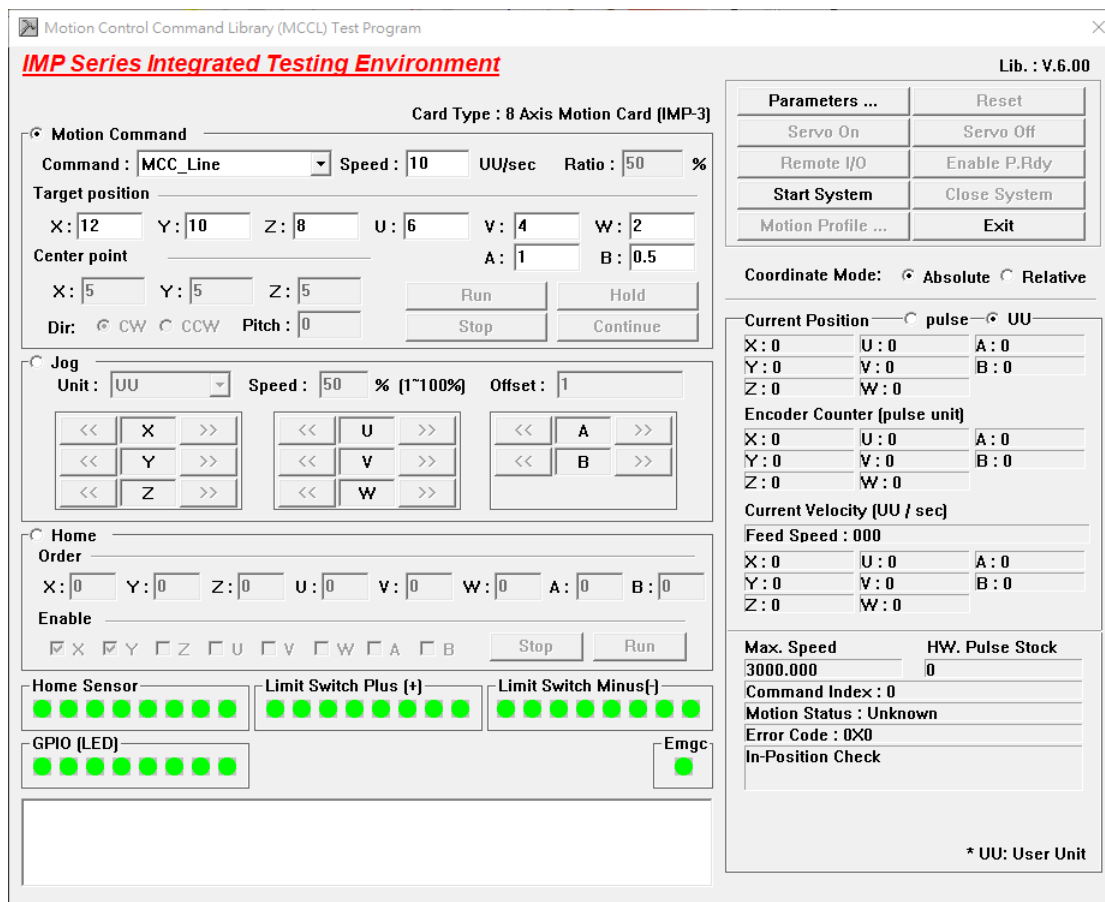
***m_nGroupIndex**。*

測試軟體僅使用到 MCCL 所提供的基本功能函式，其他函式的用法請參考”IMP Seires 運動控制函式庫參考手冊”的說明，而如果對於函式的用法仍有疑義請參考”IMP Seires 運動控制函式庫範例手冊”的說明。下面將配合圖示說明使用到的基本功能函式，並簡單說明測試軟體的操作方法。

進入運動控制函式庫測試軟體(MCCLTest4IMP)的操作方式：



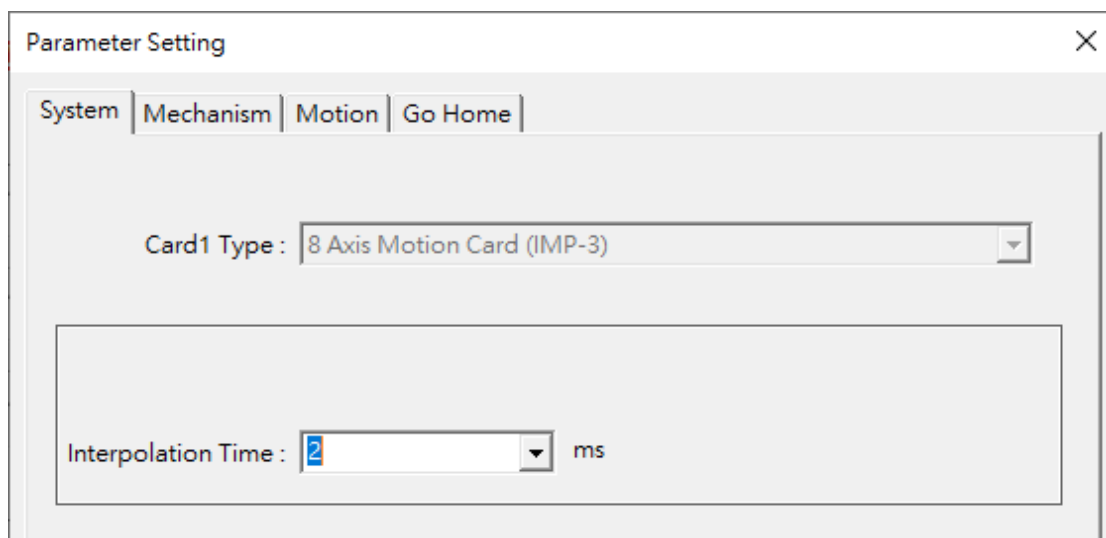
運動控制函式庫測試軟體(MCCLTest4IMP)主畫面：



2. 啟動測試軟體

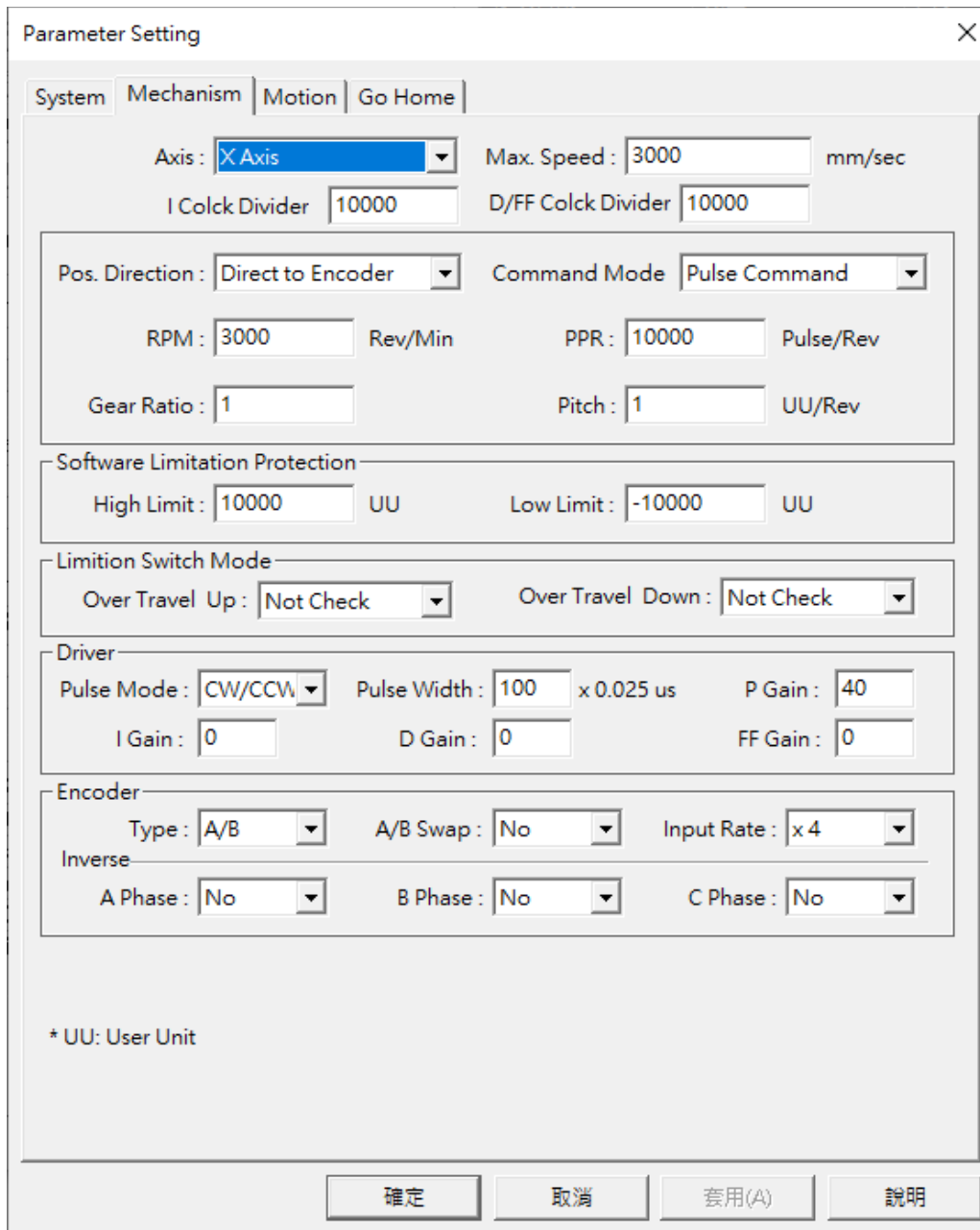
欲測試 MCCL 的功能需先啟動 MCCL，設定參數步驟如下：

1. 按下主畫面中 **Parameters ...** 按鍵，可設定”System”（如下圖）的 Interpolation Time(插值時間)，建議值為 2ms。



2. 檢查機構參數的設定是否正確，可使用”Mechanism(機構參數設定)”(如下圖)的各項設定。各項參數的意義請參考”IMP Series 運動控制函式庫使用手冊

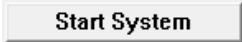
2.4.1 機構參數”的說明。



The image shows a 'Parameter Setting' dialog box with the 'Mechanism' tab selected. The dialog is organized into several sections:

- System:** Includes tabs for 'System', 'Mechanism', 'Motion', and 'Go Home'.
- Axis:** A dropdown menu is set to 'X Axis'.
- Max. Speed:** A text input field containing '3000' with the unit 'mm/sec'.
- I Colck Divider:** A text input field containing '10000'.
- D/FF Colck Divider:** A text input field containing '10000'.
- Pos. Direction:** A dropdown menu set to 'Direct to Encoder'.
- Command Mode:** A dropdown menu set to 'Pulse Command'.
- RPM:** A text input field containing '3000' with the unit 'Rev/Min'.
- PPR:** A text input field containing '10000' with the unit 'Pulse/Rev'.
- Gear Ratio:** A text input field containing '1'.
- Pitch:** A text input field containing '1' with the unit 'UU/Rev'.
- Software Limitation Protection:** Includes 'High Limit' (10000 UU) and 'Low Limit' (-10000 UU).
- Limitation Switch Mode:** Includes 'Over Travel Up' (Not Check) and 'Over Travel Down' (Not Check).
- Driver:** Includes 'Pulse Mode' (CW/CCW), 'Pulse Width' (100 x 0.025 us), 'P Gain' (40), 'I Gain' (0), 'D Gain' (0), and 'FF Gain' (0).
- Encoder:** Includes 'Type' (A/B), 'A/B Swap' (No), 'Input Rate' (x4), and 'Inverse' section with 'A Phase' (No), 'B Phase' (No), and 'C Phase' (No).

At the bottom, there is a note: '* UU: User Unit'. Below the dialog are four buttons: '確定' (OK), '取消' (Cancel), '套用(A)' (Apply), and '說明' (Help).

3. 按下主畫面中  按鍵，此動作除了將設定系統參數外，並將呼叫 MCC_InitSystem()。

3. 運動屬性設定

說明在變動”Motion(運動屬性設定)”(如下圖)的設定時，測試軟體需使用的 MCCL 函式，使用者可依照運動軌跡的變動，瞭解 MCCL 的使用方式。

Parameter Setting
✕

System |
 Mechanism |
 Motion |
 Go Home

Acceleration Curve: Trapezoid ▾

Acceleration Time: 300 ms

Deceleration Curve: Trapezoid ▾

Deceleration Time: 300 ms

Path Blending: Disable ▾

In Position

In Position: Disable ▾ Mode: IPM_ONETIME_BLOCK ▾

Max. Check Time: 100 ms Settle Time: 100 ms

Tolerance: 0.1 UU

Tracking Error

Enable: X Y Z U V W A B

Error Limit: 10 10 10 10 10 10 10 10

Software Over Travel Check

X Axis Y Axis Z Axis U Axis V Axis W Axis A Axis B Axis

Hardware Limit Switch Check

Check Limit Switch Limit Mode: No Error Bi-Direction ▾

* UU: User Unit

確定
取消
套用(A)
說明



下面將說明各選項對應的函式呼叫：

Acceleration Curve

Acceleration Curve :

設定在進行直線、圓弧、圓、螺線運動時 X, Y, Z, U, V, W, A, B 軸的加速型式，可設定梯形曲線與 S 形曲線。

當選擇 Trapezoid 時，將呼叫 MCC_SetAccType('T')，表示使用梯形加速曲線；

當選擇 S 時，則呼叫 MCC_SetAccType('S')，表示使用 S 形加速曲線。

Deceleration Curve

Deceleration Curve :

設定在進行直線、圓弧、圓、螺線運動時 X, Y, Z, U, V, W, A, B 軸的減速型式，可設定梯形曲線與 S 形曲線。

當選擇 Trapezoid 時，將呼叫 MCC_SetDecType('T')，表示使用梯形減速曲線；

當選擇 S 時，則呼叫 MCC_SetAccType('S')，表示使用 S 形減速曲線。

Acceleration Time

Acceleration Time : ms

設定加速時間，單位為 ms。設定的加速時間必須大於 0。假設要求加速時間為 dfAccTime，則可呼叫 MCC_SetAccTime(dfAccTime)。

Deceleration Time

Deceleration Time : ms

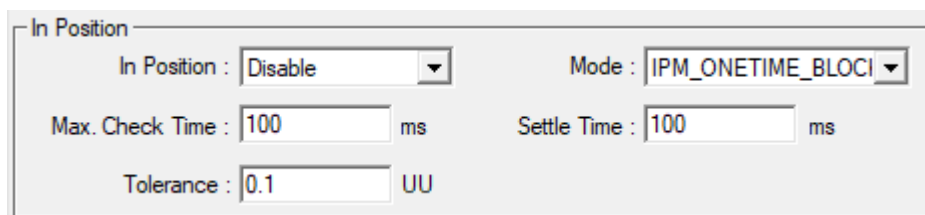
設定減速時間，單位為 ms。設定的減速時間必須大於 0。假設要求減速時間為 dfDecTime，則可呼叫 MCC_SetDecTime(dfDecTime)。

Path Blending

Path Blending :

選擇是否開啟平滑運動功能。當選擇"Disable"時將關閉平滑運動功能，此時呼叫 MCC_DisableBlend()；當選擇"Enable"時將開啟平滑運動功能，此時呼叫 MCC_EnableBlend()。

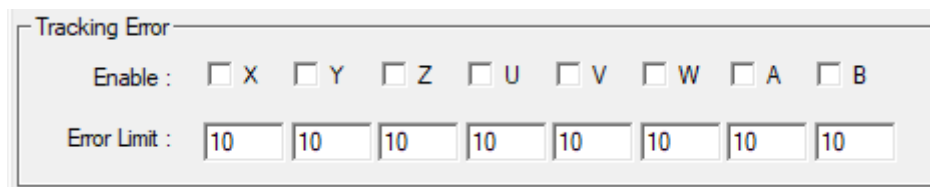
In Position



選擇是否開啟定位確認功能及設定其參數(有關定位確認功能請參考” *IMP Seires 運動控制函式庫使用手冊 2.7.2 定位確認* ”)。

In Position : <input type="text" value="Disable"/>	MCC_EnableInPos() / MCC_DisableInPos()
Mode : <input type="text" value="IPM_ONETIME_BLOCK"/>	MCC_SetInPosMode()
Max. Check Time : <input type="text" value="100"/> ms	MCC_SetInPosMaxCheckTime()
Settle Time : <input type="text" value="100"/> ms	MCC_SetInPosSettleTime()
Tolerance : <input type="text" value="0.1"/> UU	MCC_SetInPosToleranceEx()

Tracking Error

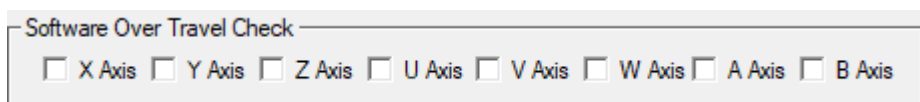


選擇是否開啟跟隨誤差功能及設定其參數(有關跟隨誤差功能請參考” *IMP Seires 運動控制函式庫使用手冊 2.7.3 跟隨誤差偵測* ”)。

開啟/關閉跟隨誤差功能：MCC_EnableTrackError()/MCC_DisableTrackError()

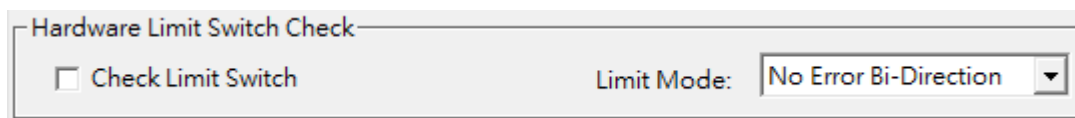
設定跟隨誤差容許範圍：MCC_SetTrackErrorLimit()

Software Over Travel Check

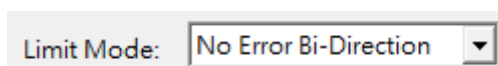


使用 MCC_SetOverTravelCheck() 設定是否開啟各軸的軟體過行程檢查功能，此項功能可使移動位置限制在工作區間內。

Hardware Limit Switch Check



狀態若為 Check Limit Switch 則將呼叫 `MCC_EnableLimitSwitchCheck()` 開啟極限開關檢查功能，否則將會呼叫 `MCC_DisableLimitSwitchCheck()` 關閉極限開關檢查功能。使用者也能呼叫 `MCC_GetLimitSwitchStatus()` 檢查目前是否已碰觸到極限開關。使用這些函式必須正確設定機構參數中的 `wOverTravelUpSensorMode` 與 `wOverTravelDownSensorMode` (必須為 Normal Open 或 Normal Close)。



可選擇碰觸到硬體極限開關之模式，可參考 `MCC_EnableLimitSwitchCheck()` 之詳細說明。

No Error Bi-Direction 模式：只要碰觸到極限開關，皆會停止輸出該軸之運動命令。

No Error Single Direction 模式：只有在碰觸到該軸運動方向的極限開關時，才會停止輸出該軸之運動命令。

Error Msg Bi-Direction 模式：只要碰觸到極限開關，皆會停止輸出該軸之運動命令；並且會產生錯誤記錄。

Error Msg Single Direction 模式：只有在碰觸到該軸運動方向的極限開關時，才會停止輸出該軸之運動命令；並且會產生錯誤記錄。

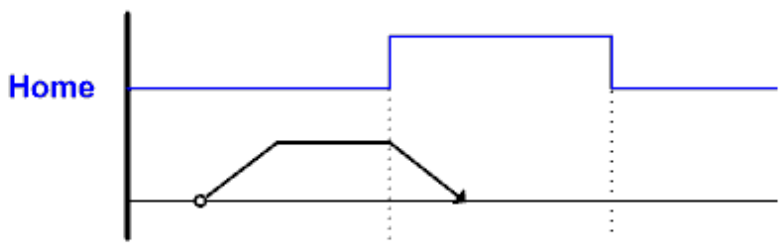
4. 原點復歸運動屬性設定

Parameter Setting

System | Mechanism | Motion | **Go Home**

Channel : 0

Go-Home :	Mode 3	Direction :	Negative		
Sensor Mode :	Normal Open	Index(Z) Count :	0		
High Speed :	10	UU/sec	Low Speed :	2	UU/sec
Acc. Time :	500	ms	Dec. Time :	500	ms
Home Offset :	0	UU			



* UU: User Unit

確定 取消 套用(A) 說明

上圖顯示“Go Home(原點復歸運動屬性設定)”，這些設定值將直接反映在原點復歸參數，詳細的內容請參閱”IMP Seires 運動控制函式庫使用手冊之 2.8 原點復歸”。

5. 執行運動命令

下圖為“Motion Command(運動命令參數設定)”區，下面將分別說明一般運動操作相關內容：

Motion Command

Command: Speed: UU/sec Ratio: %

Target position

X: Y: Z: U: V: W:

Center point

X: Y: Z: A: B:

Dir: CW CCW Pitch:

Run Hold

Stop Continue

運動命令選項： Command:

由此選項可挑選運動命令的類型，選項內容與函式名稱相同，包括點對點、直線、圓、圓弧與螺線運動。

速度設定：

Speed: UU/sec 用來設定進給速度，單位為 UU/sec，此項輸入值將作為 MCC_SetFeedSpeed() 的呼叫參數，其值不得小於或等於 0。

Ratio: % 設定點對點的速度比例，此項輸入值的範圍為 1~100，將作為 MCC_SetPtPSpeed() 的呼叫參數。

參數內容：

Target position

X: Y: Z: U: V: W:

Center point

X: Y: Z: A: B:

Dir: CW CCW Pitch:

Run Hold

Stop Continue

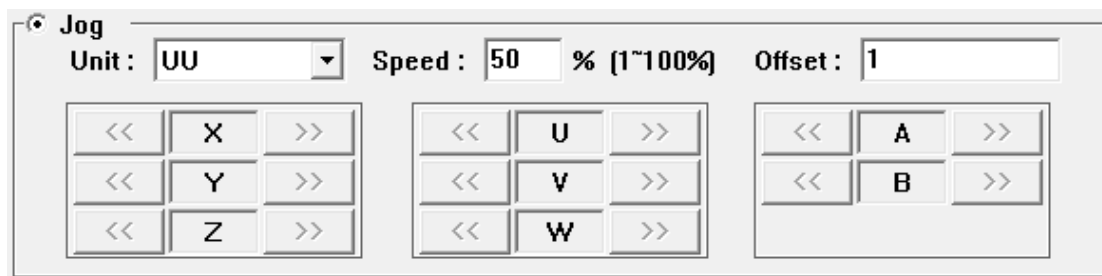
上圖中的 Target position 與 Center point 是呼叫” Command(運動命令選項)”所列出函式時所需要傳入的參數，詳細內容請參考”IMP Series 運動控制函式庫參考手冊”。



當各項參數設定無誤後，可按下 **Run** 按鍵將運動命令送至運動命令緩衝區。按下 **Hold** 按鍵可呼叫 `MCC_HoldMotion()` 暫停運動；按下 **Continue** 按鍵可呼叫 `MCC_ContiMotion()` 繼續執行被暫停的運動命令；按下 **Stop** 按鍵可呼叫 `MCC_AbortMotionEx()` 停止目前運動並清除運動命令緩衝區中之庫存命令。

6. JOG 運動

下圖為”JOG 運動參數設定”區，下面將說明與 JOG 運動相關選項的內容：



位移量的單位：**Unit**：UU

位移量的單位如果選擇”UU(User Unit)”選項，則 JOG 運動將使用 User Unit 作為位移量單位，並在使用 JOG 運動控制鍵時(例如按下 **>>**)，依照指定之增量位移值 (**Offset**：1)及進給速度比例(**Speed**：50%)帶動指定軸，此時將呼叫 MCC_JogSpace()。

位移量的單位如果選擇”Pulse”選項，則 JOG 運動將使用 pulse 作為位移量單位，此時系統如果處於運動停止狀態，且使用 JOG 運動控制鍵，將依照指定之脈衝位移量與方向帶動指定軸。脈衝之位移量設定不宜過大(不能超過 2048 pulses)，此時將呼叫 MCC_JogPulse()。

詳細函式使用請參考”IMP Seires 運動控制函式庫使用手冊 2.6.2 基本軌跡規劃”的說明。

7. 原點復歸運動

下圖為”Home(原點復歸)”區，下面將說明與原點復歸運動操作相關選項的內容：



與原點復歸運動有關的函式宣告如下：

```
MCC_Home( int nOrder0, int nOrder1, int nOrder2,
          int nOrder3, int nOrder4, int nOrder5,
          int nOrder6, int nOrder7, WORD wCardIndex);
```

```
MCC_GetGoHomeStatus();
```

其中 MCC_Home()可使機器完成原點復歸動作，使用此函數時，可配合呼叫函數 MCC_GetGoHomeStatus()，以檢查原點復歸是否完成。nOrder0~ nOrder7 分別表示 X, Y, Z, U, V, W, A, B 各軸執行原點復歸動作的順序，各軸執行原點復歸動作的順序可設定為 0~7，設定值可重複。這些參數可由”Home(原點復歸)”區獲得。

原點復歸動作的順序設定值如為 255(0xff)，表示不對該運動軸執行原點復歸的動作， 未勾選的運動軸復歸順序將被設為 0xff。

當各項參數設定無誤後，按下  按鍵將呼叫 MCC_Home()執行原點復歸運動。在執行原點復歸運動時，按下  按鍵將呼叫 MCC_AbortGoHome()停止原點復歸運動。

8. 運動狀態與訊息顯示

下圖為”運動狀態訊息顯示”區，下面將分別說明各項資訊的獲得方式：

Coordinate Mode: <input checked="" type="radio"/> Absolute <input type="radio"/> Relative		
Current Position <input type="radio"/> pulse <input checked="" type="radio"/> UU		
X: 0	U: 0	A: 0
Y: 0	V: 0	B: 0
Z: 0	W: 0	
Encoder Counter (pulse unit)		
X: 0	U: 0	A: 0
Y: 0	V: 0	B: 0
Z: 0	W: 0	
Current Velocity (UU / sec)		
Feed Speed : 000		
X: 0	U: 0	A: 0
Y: 0	V: 0	B: 0
Z: 0	W: 0	
Max. Speed	HW. Pulse Stock	
3000.000	0	
Command Index : 0		
Motion Status : Unknown		
Error Code : 0X0		
In-Position Check		
* UU: User Unit		

Coordinate Mode

Coordinate Mode: Absolute Relative

設定以絕對座標型態或以增量型態表示各軸座標位置。當選擇”Absolute”時，將呼叫 `MCC_SetAbsolute()`；當選擇”Relative”時，將呼叫 `MCC_SetIncrease()`。

各軸目前位置命令之直角座標值：

Current Position <input type="radio"/> pulse <input checked="" type="radio"/> UU		
X: 0	U: 0	A: 0
Y: 0	V: 0	B: 0
Z: 0	W: 0	

可使用 `MCC_GetCurPos()`讀取各軸目前位置之直角座標值。

Encoder Counter (pulse unit)		
X : 0	U : 0	A : 0
Y : 0	V : 0	B : 0
Z : 0	W : 0	

各軸目前位置的編碼器計數值：

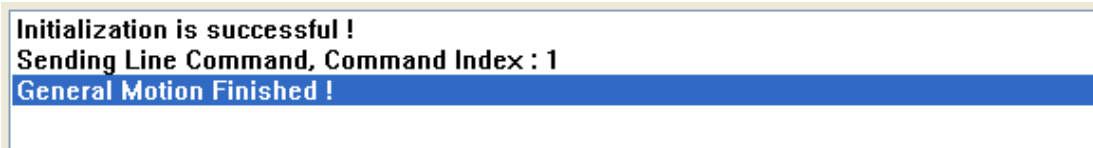
如果系統有安裝編碼器可使用 MCC_GetENCValue()讀取各軸目前位置的編碼器計數值。

Current Velocity (UU / sec)		
Feed Speed : 000		
X : 0	U : 0	A : 0
Y : 0	V : 0	B : 0
Z : 0	W : 0	

目前實際進給速度與各軸速度：

呼叫 MCC_GetCurFeedSpeed()與 MCC_GetSpeed()可獲得一般運動(不包括點對點運動)目前的進給速度與各軸速度。

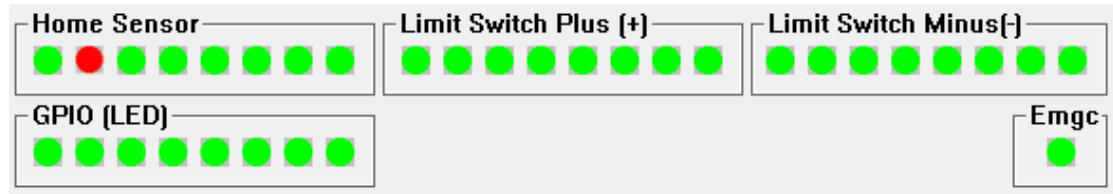
訊息視窗



”訊息視窗”的內容顯示目前的運動狀態；此外，也顯示在按下”Motion Command(運動命令參數設定)”區中的 按鍵時，被送至運動命令緩衝區的運動命令之編碼，編碼可由函式的傳回值獲得(例如呼叫 MCC_Line()的函式傳回值)。目前正在執行的運動命令之相關資訊可以利用 MCC_GetCurCommand()獲得，這些資訊包括運動命令編碼。執行中的運動命令編碼顯示如下：

Max. Speed	HW. Pulse Stock
3000.000	30
Command Index : 3	
Motion Status : Unknown	
Error Code : 0X0	
In-Position Check	

”Home Sensor & Limit Switch Sensor & GPIO(LED) & Emergency Stop 狀態”顯示區，用來顯示這些輸入點的狀態。



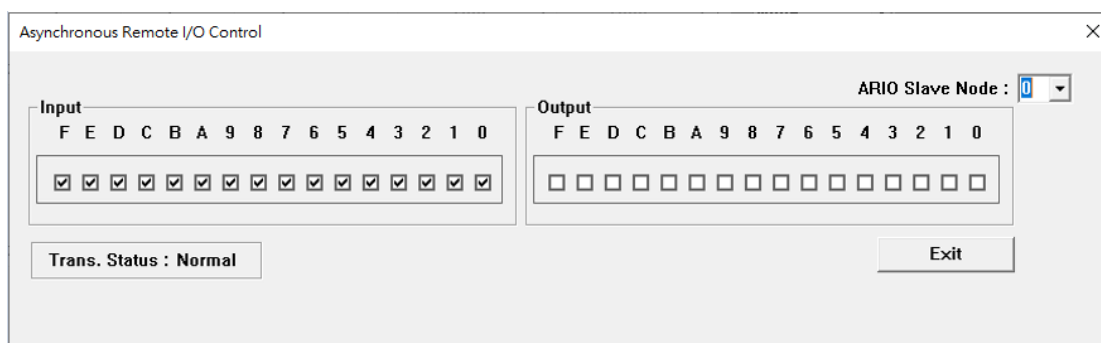
可以使用 `MCC_GetLimitSwitchStatus()`、`MCC_GetGoHomeStatus()`、`MCC_LIO_GetLedLightStatus()` 與 `MCC_GetEmgcStopStatus()` 讀取 Home Sensor、Limit Switch Sensor、GPIO(LED)與 Emergency Stop 的狀態。

9. ASYNCHRONOUS REMOTE I/O 測試

如系統有安裝 Asynchronous Remote I/O 控制子版(ARIO)，在初始化系統成功即可按下 **Remote I/O** 按鍵獲得 ARIO 控制視窗。注意系統在使用 MCC_InitSystem 成功初始化系統後，尚需呼叫下列函式才能正常使用 ARIO 的功能，這些函式包括：

```
MCC_EnableARIOSetControl();  
MCC_EnableARIOSlaveControl();
```

下圖為 ARIO 控制視窗



分別使用 MCC_GetARIOInputValue()與 MCC_SetARIOOutputValue()讀取與設定 ARIO 的訊號狀態。