



IMP Series

運動控制函式庫

參考手冊

版本：V.3.00

日期：2019.09

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



目 錄

I.	MCCL 功能列表.....	2
II.	MCCL 函式庫.....	12
A.	系統功能.....	12
B.	近端輸出入接點(Local I/O)控制.....	23
C.	座標系統.....	29
D.	過行程保護.....	34
E.	直線、圓弧、圓、螺線運動(一般運動).....	39
F.	點對點運動.....	59
G.	JOG 運動.....	69
H.	運動狀態檢視.....	71
I.	原點復歸.....	76
J.	定位控制.....	78
K.	進階軌跡規劃.....	94
L.	編碼器控制.....	100
M.	計時器與 Watch Dog 控制.....	112
N.	Remote I/O 控制.....	116
O.	D/A Converter 控制.....	121
P.	A/D Converter 控制.....	125
III.	錯誤訊息代碼.....	130
IV.	函式傳回值.....	132
V.	運動控制函式庫初始設定.....	133



I. MCCL 功能列表

A. 系統功能

編號	函式名稱	說明
1	MCC_GetVersion()	讀取函式庫版本
2	MCC_CreateGroup()	新增運動群組
3	MCC_CreateGroupEx()	新增運動群組與設定運動命令佇列大小
4	MCC_CloseGroup()	關閉指定運動群組
5	MCC_CloseAllGroups()	關閉所有運動群組
6	MCC_SetMacParam()	設定機構參數
7	MCC_GetMacParam()	讀取機構參數
8	MCC_SetEncoderConfig()	設定編碼器參數
9	MCC_GetEncoderConfig()	讀取編碼器參數
10	MCC_SetHomeConfig()	設定原點復歸參數
11	MCC_GetHomeConfig()	讀取原點復歸參數
12	MCC_UpdateParam()	使系統反應更新後的參數
13	MCC_GetCmdQueueSize()	讀取運動命令佇列大小
14	MCC_InitSystem()	啟動運動控制函式庫
15	MCC_InitSystemEx()	啟動運動控制函式庫
16	MCC_CloseSystem()	結束運動控制函式庫
17	MCC_ResetMotion()	重置運動控制函式庫
18	MCC_EnableDryRun()	開啟運動空跑功能
19	MCC_DisableDryRun()	關閉運動空跑功能
20	MCC_CheckDryRun()	檢查運動空跑功能是否開啟
21	MCC_SetSysMaxSpeed()	設定一般運動進給速度的上限
22	MCC_GetSysMaxSpeed()	讀取一般運動進給速度的上限
23	MCC_GetConvFactor()	讀取硬體參數($PPR * Gear Ratio / Pitch$)
24	MCC_InitSimulation()	啟動運動控制函式庫模擬功能



B. 輸出入接點(Local I/O)控制

編號	函式名稱	說明
1	MCC_SetServoOn()	伺服啟動
2	MCC_SetServoOff()	伺服停止
3	MCC_EnablePosReady()	輸出 Position Ready 訊號
4	MCC_DisablePosReady()	停止輸出 Position Ready 訊號
5	MCC_GetEmgcStopStatus()	讀取緊急停止開關的輸入狀態
6	MCC_SetLIORoutine()	串接自訂的近端輸出、入接點(LIO)中斷服務函式(ISR)
7	MCC_SetLIOTriggerType()	設定近端輸入接點中斷觸發類型
8	MCC_EnableLIOTrigger()	開啟近端輸入接點訊號觸發 ISR 功能
9	MCC_DisableLIOTrigger()	關閉近端輸入接點訊號觸發 ISR 功能
10	MCC_SetLedLightOn()	開啟 LED 燈輸出
11	MCC_SetLedLightOff()	關閉 LED 燈輸出
12	MCC_GetLedLightStatus()	讀取目前 LED 燈輸出狀態
13	MCC_QueueLIOOutputValue()	將指定近端輸出接點之訊號狀態值存放至運動命令佇列中

C. 座標系統

編號	函式名稱	說明
1	MCC_SetAbsolute()	使用絕對座標型態
2	MCC_SetIncrease()	使用增量座標型態
3	MCC_GetCoordType()	讀取使用的座標型態
4	MCC_GetCurRefPos()	讀取各軸位置之直角座標值(不含補償)
5	MCC_GetCurPos()	讀取各軸位置之直角座標值(含補償)
6	MCC_GetPulsePos()	讀取各軸位置之馬達座標值(含補償)
7	MCC_DefineOrigin()	將目前位置視為原點
8	MCC_DefinePosHere()	將目前座標位置與實際機台位置對齊
9	MCC_DefinePos()	設定目前的座標位置

D. 過行程保護

編號	函式名稱	說明
1	MCC_EnableLimitSwitchCheck()	開啟硬體極限開關保護功能
2	MCC_DisableLimitSwitchCheck()	關閉硬體極限開關保護功能
3	MCC_SetOverTravelCheck()	設定軟體過行程保護功能
4	MCC_GetOverTravelCheck()	讀取軟體過行程保護的設定情形
5	MCC_GetLimitSwitchStatus()	讀取硬體極限開關的狀態
6	MCC_GetLatchedLimitSwitchStatus()	讀取硬體極限開關的門鎖狀態
7	MCC_ClearLatchedLimitSwitchStatus()	清除硬體極限開關的門鎖狀態

E. 直線、圓弧、圓、螺線運動(一般運動)

編號	函式名稱	說明
1	MCC_SetAccType()	設定加速型式
2	MCC_GetAccType()	讀取使用的加速型式
3	MCC_SetDecType()	設定減速型式
4	MCC_GetDecType()	讀取使用的減速型式
5	MCC_SetAccTime()	設定加速時間
6	MCC_GetAccTime()	讀取加速時間
7	MCC_SetDecTime()	設定減速時間
8	MCC_GetDecTime()	讀取減速時間
9	MCC_SetFeedSpeed()	設定進給速度
10	MCC_GetFeedSpeed()	讀取使用的進給速度
11	MCC_GetCurFeedSpeed()	讀取機台目前的進給速度
12	MCC_GetSpeed()	讀取目前各軸的速度
13	MCC_Line()	八軸同動直線運動
14	MCC_ArcXYZ()	X-Y-Z 空間任三點圓弧運動
15	MCC_ArcXY()	X-Y 平面圓弧運動
16	MCC_ArcYZ()	Y-Z 平面圓弧運動
17	MCC_ArcZX()	Z-X 平面圓弧運動
18	MCC_ArcXYZ_Aux()	X-Y-Z 空間任三點圓弧運動與輔助軸直線運動
19	MCC_ArcXY_Aux()	X-Y 平面圓弧與輔助軸直線運動
20	MCC_ArcYZ_Aux()	Y-Z 平面圓弧與輔助軸直線運動



21	MCC_ArcZX_Aux()	Z-X 平面圓弧與輔助軸直線運動
22	MCC_ArcThetaXY()	X-Y 平面圓弧運動(以旋轉角度為參數)
23	MCC_ArcThetaYZ()	Y-Z 平面圓弧運動(以旋轉角度為參數)
24	MCC_ArcThetaZX()	Z-X 平面圓弧運動(以旋轉角度為參數)
25	MCC_CircleXY()	X-Y 平面全圓運動
26	MCC_CircleYZ()	Y-Z 平面全圓運動
27	MCC_CircleZX()	Z-X 平面全圓運動
28	MCC_CircleXY_Aux()	X-Y 平面全圓與輔助軸直線運動
29	MCC_CircleYZ_Aux()	Y-Z 平面全圓與輔助軸直線運動
30	MCC_CircleZX_Aux()	Z-X 平面全圓與輔助軸直線運動
31	MCC_HelicalXY_Z()	在 X-Y 平面進行圓周運動的螺線運動
32	MCC_HelicalYZ_X()	在 Y-Z 平面進行圓周運動的螺線運動
33	MCC_HelicalZX_Y()	在 Z-X 平面進行圓周運動的螺線運動
34	MCC_HelicalXY_Z_Aux()	在 X-Y 平面進行圓周運動的螺線運動 與輔助軸直線運動
35	MCC_HelicalYZ_X_Aux()	在 Y-Z 平面進行圓周運動的螺線運動 與輔助軸直線運動
36	MCC_HelicalZX_Y_Aux()	在 Z-X 平面進行圓周運動的螺線運動 與輔助軸直線運動

F. 點對點運動

編號	函式名稱	說明
1	MCC_SetPtPSpeed()	設定速度比例
2	MCC_GetPtPSpeed()	讀取使用的速度比例
3	MCC_PtP()	點對點運動
4	MCC_PtPX()	X 軸點對點運動
5	MCC_PtPY()	Y 軸點對點運動
6	MCC_PtPZ()	Z 軸點對點運動
7	MCC_PtPU()	U 軸點對點運動
8	MCC_PtPV()	V 軸點對點運動
9	MCC_PtPW()	W 軸點對點運動
10	MCC_PtPA()	A 軸點對點運動
11	MCC_PtPB()	B 軸點對點運動
12	MCC_SetPtPAccType()	設定點對點運動的各軸加速型式
13	MCC_GetPtPAccType()	讀取點對點運動的各軸加速型式



14	MCC_SetPtPDecType()	設定點對點運動的各軸減速型式
15	MCC_GetPtPDecType()	讀取點對點運動的各軸減速型式
16	MCC_SetPtPAccTime()	設定點對點運動的各軸加速時間
17	MCC_GetPtPAccTime()	讀取點對點運動的各軸加速時間
18	MCC_SetPtPDecTime()	設定點對點運動的各軸減速時間
19	MCC_GetPtPDecTime()	讀取點對點運動的各軸減速時間
20	MCC_SyncPtP()	同動點對點運動

G. JOG 運動

編號	函式名稱	說明
1	MCC_JogPulse()	微動(脈衝運動)
2	MCC_JogSpace()	吋動(單步運動)
3	MCC_JogConti()	連續吋動

H. 運動狀態檢視

編號	函式名稱	說明
1	MCC_GetMotionStatus()	讀取目前的運動狀態
2	MCC_GetCurCommand()	讀取執行中的運動命令之相關資訊
3	MCC_GetCommandCount()	讀取運動命令的庫存數目
4	MCC_ResetCommandIndex()	使運動命令編碼值歸零
5	MCC_GetCurPulseStockCount()	讀取目前硬體上的 Pulse 命令庫存數目
6	MCC_SetMaxPulseStockNum()	設定最大使用的 Pulse 命令庫存數目
7	MCC_GetMaxPulseStockNum()	讀取最大使用的 Pulse 命令庫存數目
8	MCC_GetErrorCode()	讀取現存錯誤的代碼
9	MCC_ClearError()	清除現存的錯誤狀態
10	MCC_SetCommandIndex()	設定運動命令編碼值
11	MCC_GetCommandIndex()	讀取運動命令編碼值

I. 原點復歸

編號	函式名稱	說明
1	MCC_Home()	要求原點復歸運動
2	MCC_GetGoHomeStatus()	確認原點復歸運動是否完成
3	MCC_AbortGoHome()	停止原點復歸運動
4	MCC_GetHomeSensorStatus()	讀取 Home Sensor 接點的狀態

J. 定位控制

編號	函式名稱	說明
1	MCC_SetCompParam()	設定齒輪齒隙、間隙補償參數
2	MCC_UpdateCompParam()	反應更新後的齒輪齒隙、間隙補償參數
3	MCC_SetPGain()	設定位置閉迴路控制使用的比例增益
4	MCC_GetPGain()	讀取位置閉迴路控制使用的比例增益
5	MCC_SetIGain()	設定閉迴路控制使用的積分增益
6	MCC_GetIGain()	讀取閉迴路控制使用的積分增益
7	MCC_SetDGain()	設定閉迴路控制使用的微分增益
8	MCC_GetDGain()	讀取閉迴路控制使用的微分增益
9	MCC_SetFGain()	設定閉迴路控制使用的前饋增益
10	MCC_GetFGain()	讀取閉迴路控制使用的前饋增益
11	MCC_SetMaxPulseSpeed()	設定各軸最大 Pulse 速度的上限
12	MCC_GetMaxPulseSpeed()	讀取各軸最大 Pulse 速度的上限
13	MCC_SetMaxPulseAcc()	設定各軸最大 Pulse 加速度的上限
14	MCC_GetMaxPulseAcc()	讀取各軸最大 Pulse 加速度的上限
15	MCC_SetInPosMode()	設定定位確認使用模式
16	MCC_SetInPosMaxCheckTime()	設定定位檢查時間
17	MCC_SetInPosSettleTime()	設定定位持續時間
18	MCC_EnableInPos()	開啟定位確認功能
19	MCC_DisableInPos()	關閉定位確認功能
20	MCC_SetInPosToleranceEx()	設定定位誤差容許範圍
21	MCC_GetInPosToleranceEx()	讀取使用的定位誤差容許範圍
22	MCC_GetInPosStatus()	確認實際位置是否已滿足定位確認條件
23	MCC_EnableTrackError()	開啟軌跡誤差檢查功能
24	MCC_DisableTrackError()	關閉軌跡誤差檢查功能
25	MCC_SetTrackErrorLimit()	設定軌跡誤差容許範圍



26	MCC_GetTrackErrorLimit()	讀取軌跡誤差容許範圍
27	MCC_SetPCLRoutine()	串接自訂的位置控制閉迴路(PCL)中斷服務函式(ISR)
28	MCC_SetErrorCountThreshold()	設定位置脈波誤差容許範圍
29	MCC_GetErrorCount()	讀取各軸位置脈波誤差大小
30	MCC_SetIGainClockDivider()	設定閉迴路控制使用的積分取樣時間
31	MCC_SetDGainClockDivider()	設定閉迴路控制使用的微分取樣時間

K. 進階軌跡規劃

編號	函式名稱	說明
1	MCC_HoldMotion()	暫停運動
2	MCC_ContiMotion()	繼續運動
3	MCC_AbortMotion()	緊急停止，並放棄所有未執行之運動命令
4	MCC_AbortMotionEx()	以設定的減速時間減速至停止，並放棄所有未執行之運動命令
5	MCC_EnableBlend()	開啟平滑運動功能
6	MCC_DisableBlend()	關閉平滑運動功能
7	MCC_CheckBlend()	檢查是否開啟平滑運動功能
8	MCC_DelayMotion()	設定運動延遲時間
9	MCC_CheckDelay()	檢查是否進入運動延遲狀態
10	MCC_OverrideSpeed()	設定一般運動與點對點運動的速度比例
11	MCC_OverrideSpeedEx()	設定一般運動與點對點運動的速度比例
12	MCC_GetOverrideRate()	讀取使用的一般運動與點對點運動速度比例
13	MCC_TimeDelay()	設定運動延遲時間

L. 編碼器控制

編號	函式名稱	說明
1	MCC_SetENCRoutine()	串接自訂的編碼器(ENC)中斷服務函式(ISR)
2	MCC_SetENCInputRate()	設定編碼器的回授倍率
3	MCC_ClearENCCounter()	使編碼器的計數值歸零
4	MCC_GetENCValue()	讀取編碼器的計數值
5	MCC_SetENCLatchType()	設定觸發門鎖(Latch)編碼器計數值動作的方式
6	MCC_SetENCLatchSource()	設定觸發門鎖編碼器計數值動作的訊號來源
7	MCC_GetENCLatchValue()	讀取門鎖暫存器內的紀錄值
8	MCC_EnableENCIndexTrigger()	開啟編碼器 INDEX 中斷功能
9	MCC_DisableENCIndexTrigger()	關閉編碼器 INDEX 中斷功能
10	MCC_GetENCIndexStatus()	讀取目前編碼器 INDEX 訊號狀態
11	MCC_SetENCCompValue()	設定編碼器的比較值
12	MCC_EnableENCCompTrigger()	開啟編碼器比較中斷功能
13	MCC_DisableENCCompTrigger()	關閉編碼器比較中斷功能
14	MCC_Enable3DCompareChannel()	開啟 3D 編碼器計數值比較功能
15	MCC_Set3DCompareTolerance()	設定 3D 編碼器的比較值之誤差容許範圍
16	MCC_SetENCCompValueTab()	設定相同間隔編碼器計數值比較值
17	MCC_SetLIOTrigOutEnable()	設定編碼器計數值比較成功時，是否要啟用近端輸出接點，及欲輸出的接點編號
18	MCC_SetLIOTrigOutValue()	設定編碼器計數值比較成功時，近端輸出接點是否輸出及欲輸出的接點編號
19	MCC_SetLIOTrigOutValueEx()	設定編碼器計數值比較成功時，多組近端輸出接點是否輸出及欲輸出的接點編號
20	MCC_SetLIOTrigOutSource()	設定欲比較編碼器計數值之實體軸與對應的近端輸出接點編號
21	MCC_SetLIOTrigOutPeriod()	設定編碼器計數值比較成功時，近端輸出接點輸出時間
22	MCC_GetLIOCompStockCount()	取得目前所有編碼器計數比較值觸發 LIO 之庫存筆數
23	MCC_EraseLIOCompValue()	清除所有編碼器計數比較值觸發 LIO 之庫存筆數



24	MCC_SetENCGearRate()	設定軟體的 Gear Rate 值
25	MCC_GetENCCompValue()	讀取正要被編碼器計數比較值的數值
26	MCC_GetENCCompStockCount()	讀取目前所有編碼器計數比較值之筆數
27	MCC_EraseENCCompValue()	清除所有編碼器計數比較值之筆數

M. 計時器與 Watch Dog 控制

編號	函式名稱	說明
1	MCC_SetTMRRoutine()	串接自訂的計時觸發中斷服務函式 (ISR)
2	MCC_SetTimer()	設定計時器的計時時間
3	MCC_EnableTimer()	開啟計時器計時功能
4	MCC_DisableTimer()	關閉計時器計時功能
5	MCC_EnableTimerTrigger()	開啟計時器中斷功能
6	MCC_DisableTimerTrigger()	關閉計時器中斷功能
7	MCC_SetWatchDogTimer()	設定 Watch Dog 的計時時間
8	MCC_SetWatchDogResetPeriod()	設定 Watch Dog 的 Reset 訊號持續時間
9	MCC_EnableWatchDogTimer()	開啟 Watch Dog 功能
10	MCC_DisableWatchDogTimer()	關閉 Watch Dog 功能
11	MCC_RefreshWatchDogTimer()	重置 Watch Dog 計時時間

N. Remote I/O 控制

編號	函式名稱	說明
1	MCC_EnableARIOSlaveControl()	開啟指定的 Remote I/O Slave 資料傳輸功能
2	MCC_DisableARIOSlaveControl()	關閉指定的 Remote I/O Slave 資料傳輸功能
3	MCC_GetARIOTransStatus()	讀取目前 Remote I/O 的資料傳輸狀態
4	MCC_GetARIOMasterStatus()	讀取目前 Remote I/O Master 端傳送資料到 Slave 的狀態
5	MCC_GetARIOSlaveStatus()	讀取目前 Remote I/O Slave 端接收 Master 資料的狀態
6	MCC_GetARIOInputValue()	讀取指定 Slave 的 16 點輸入訊號狀態值
7	MCC_SetARIOOutputValue()	設定指定 Slave 的 16 點輸出訊號狀態值
8	MCC_SetARIOClockDivider()	設定 Remote I/O 傳輸資料的時脈頻率
9	MCC_GetARIOSlaveFail()	讀取 Remote I/O Slave 端狀態是否正常



10	MCC_GetARIOOutputValue()	讀取指定 Slave 的 16 點輸出訊號狀態值
11	MCC_QueueARIOOutputValue()	將指定 Slave 的 16 點輸出訊號狀態值存放至運動命令佇列中

O. D/A Converter 控制

編號	函式名稱	說明
1	MCC_SetDACOutput()	輸出指定的電壓值
2	MCC_SetDACTriggerOutput()	設定預先規劃的電壓輸出值
3	MCC_SetDACTriggerSource()	設定輸出觸發預先規劃電壓的來源
4	MCC_EnableDACTriggerMode()	開啟觸發輸出預先規劃電壓的功能
5	MCC_DisableDACTriggerMode()	關閉觸發輸出預先規劃電壓的功能
6	MCC_StartDACConv()	開啟電壓輸出功能
7	MCC_StopDACConv()	關閉電壓輸出功能

P. A/D Converter 控制

編號	函式名稱	說明
1	MCC_SetADCRoutine()	串接使用者自訂的 ADC 中斷服務函式 (ISR)
2	MCC_SetADCConvType()	設定電壓轉換形式為單極性或雙極性
3	MCC_GetADCInput()	讀取直流電壓輸入值
4	MCC_SetADCCompType()	設定電壓轉換時的比較條件
5	MCC_SetADCCompValue()	設定電壓轉換時的電壓比較值
6	MCC_GetADCCompValue()	讀取使用的電壓比較值
7	MCC_EnableADCConvChannel()	開啟選定的 Channel 電壓轉換功能
8	MCC_DisableADCConvChannel()	關閉選定的 Channel 電壓轉換功能
9	MCC_StartADCConv()	啟動電壓功能
10	MCC_StopADCConv()	停止電壓功能
11	MCC_GetADCConvType()	取得 ADC 電壓轉換模式

II. MCCL 函式庫

A. 系統功能

1. void MCC_GetVersion(
 char* *strVersion*
)

Description 讀取函式庫版本。

Parameters *strVersion* 指向一 char 值，用來存放函式庫版本

2. int MCC_CreateGroup(
 int *xMapToCh*,
 int *yMapToCh*,
 int *zMapToCh*,
 int *uMapToCh*,
 int *vMapToCh*,
 int *wMapToCh*,
 int *aMapToCh*,
 int *bMapToCh*,
 int *nCardIndex*
)

Description 此函式用來建立一個新的運動群組。

在呼叫 MCCL 中與運動群組有關的函式 (Ex: MCC_Line) 之前，必須先利用此函式或 **MCC_CreateGroupEx** 建立群組，並得到新建立群組之編號，作為其傳入參數 (之一)。

此函式必須在初始化 MCCL (MCC_InitSystem) 前始可呼叫；又，在第一次呼叫此函式前請先呼叫 MCC_CloseAllGroups。注意，任兩運動軸，不可對應到同一實體輸出 Channel。

Parameters *xMapToCh* 指定此 Group 中 X 軸所對應之實體輸出 Channel(0~7)



<i>yMapToCh</i>	指定此 Group 中 Y 軸所對應之實體輸出 Channel(0~7)
<i>zMapToCh</i>	指定此 Group 中 Z 軸所對應之實體輸出 Channel(0~7)
<i>uMapToCh</i>	指定此 Group 中 U 軸所對應之實體輸出 Channel(0~7)
<i>vMapToCh</i>	指定此 Group 中 V 軸所對應之實體輸出 Channel(0~7)
<i>wMapToCh</i>	指定此 Group 中 W 軸所對應之實體輸出 Channel(0~7)
<i>aMapToCh</i>	指定此 Group 中 A 軸所對應之實體輸出 Channel(0~7)
<i>bMapToCh</i>	指定此 Group 中 B 軸所對應之實體輸出 Channel(0~7)
<i>nCardIndex</i>	指定此群組所對應之運動控制卡編號(0 ~ 5)

若運動軸並不對應到實體軸，則須傳入 *AXIS_INVALID*

Return Value	大於等於零	新建立之群組編號
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

3. int MCC_CreateGroupEx(

```

int xMapToCh,
int yMapToCh,
int zMapToCh,
int uMapToCh,
int vMapToCh,
int wMapToCh,
int aMapToCh,
int bMapToCh,
int nCardIndex,
int nMotionQueneSize

```

)

Description 此函式用來建立一個新的運動群組及設定運動命令佇列大小。

在呼叫 MCCL 中與運動群組有關的函式 (Ex: MCC_Line) 之前，必須先利用此函式或 **MCC_CreateGroup** 建立群組，並得到新建立群組之編號，作為其傳入參數 (之一)。

此函式必須在初始化 MCCL (MCC_InitSystem) 前始可呼叫；又，在第一次呼叫此函式前請先呼叫 MCC_CloseAllGroups。注意，任兩運動軸，

不可對應到同一實體輸出 Channel。

Parameters	<i>xMapToCh</i>	指定此 Group 中 X 軸所對應之實體輸出 Channel(0~7)
	<i>yMapToCh</i>	指定此 Group 中 Y 軸所對應之實體輸出 Channel(0~7)
	<i>zMapToCh</i>	指定此 Group 中 Z 軸所對應之實體輸出 Channel(0~7)
	<i>uMapToCh</i>	指定此 Group 中 U 軸所對應之實體輸出 Channel(0~7)
	<i>vMapToCh</i>	指定此 Group 中 V 軸所對應之實體輸出 Channel(0~7)
	<i>wMapToCh</i>	指定此 Group 中 W 軸所對應之實體輸出 Channel(0~7)
	<i>aMapToCh</i>	指定此 Group 中 A 軸所對應之實體輸出 Channel(0~7)
	<i>bMapToCh</i>	指定此 Group 中 B 軸所對應之實體輸出 Channel(0~7)
	<i>nCardIndex</i>	指定此群組所對應之運動控制卡編號(0 ~ 5)
	<i>nMotionQueneSize</i>	自訂運動命令緩衝區大小
		若運動軸並不對應到實體軸，則須傳入 <i>AXIS_INVALID</i>
Return Value	大於等於零	新建立之群組編號
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_CloseGroup(int *nGroupIndex*)

Description	關閉指定的群組。	
Parameters	<i>nGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_CloseAllGroups()

Description	關閉系統中所有群組。在第一次呼叫 MCC_CreateGroup 或 MCC_CreateGroupEx 之前，請先呼叫此函式。	
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

```
6. int MCC_SetMacParam(  
    SYS_MAC_PARAM* pstMacParam,  
    WORD wChannel,  
    WORD wCardIndex,  
)
```

Description 設定各軸之機構參數。

Parameters	<i>pstMacParam</i>	指向一 SYS_MAC_PARAM 結構，用來存放欲設定之機構參數
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

```
7. int MCC_GetMacParam(  
    SYS_MAC_PARAM* pstMacParam,  
    WORD wChannel,  
    WORD wCardIndex,  
)
```

Description 讀取指定軸之機構參數內容。

Parameters	<i>pstMacParam</i>	指向一 SYS_MAC_PARAM 結構，用來存放欲讀取之機構參數內容
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

```
8. int MCC_SetEncoderConfig(  
    SYS_ENCODER_CONFIG* pstEncoderConfig,  
    WORD wChannel,  
    WORD wCardIndex,  
)
```

Description	設定各軸之編碼器參數。	
Parameters	<i>pstEncoderConfig</i>	指向一 SYS_ENCODER_CONFIG 結構，用來存放欲設定之編碼器參數
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

9. int MCC_GetEncoderConfig(
 SYS_ENCODER_CONFIG* *pstEncoderConfig*,
 WORD *wChannel*,
 WORD *wCardIndex*
)

Description	讀取指定軸之編碼器參數內容。	
Parameters	<i>pstEncoderConfig</i>	指向一 SYS_ENCODER_CONFIG 結構，用來存放欲讀取之編碼器參數內容
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

10. int MCC_SetHomeConfig(
 SYS_HOME_CONFIG* *pstHomeConfig*,
 WORD *wChannel*,
 WORD *wCardIndex*
)

Description	設定原點復歸參數。	
Parameters	<i>pstHomeConfig</i>	指向一 SYS_HOME_CONFIG 結構，用來存放欲設定之原點復歸參數
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)



	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

11. int MCC_GetHomeConfig(
SYS_HOME_CONFIG* *pstHomeConfig*,
WORD *wChannel*,
WORD *wCardIndex*
)

Description	讀取原點復歸參數的內容。	
Parameters	<i>pstHomeConfig</i>	指向一 SYS_HOME_CONFIG 結構，用來存放原點復歸參數內容
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

12. int MCC_UpdateParam()

Description	反應更新後的機構、編碼器與原點復歸參數。在呼叫過 MCC_InitSystem 後，如果再次使用 MCC_SetMacParam、MCC_SetEncoderConfig 變更相關的參數，則須使用此函式系統才會反應更新後的設定值；但須注意，呼叫此函式之結果與 MCC_ResetMotion 相似，系統將回復到初始狀態。	
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

13. int MCC_GetCmdQueueSize(
WORD *wGroupIndex*
)

Description	讀取運動命令緩衝區的大小。	
Parameters	<i>wGroupIndex</i>	Group 編號



Return Value	大於或等於零	運動命令緩衝區的大小(單位為運動命令)
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

14. int MCC_InitSystem(

int *nInterpolateTime*,
SYS_CARD_CONFIG* *pstCardConfig*,
WORD *wCardNo*

)

Description 啟動運動控制函式庫。
除了 MCC_CreateGroup、MCC_CreateGroupEx、MCC_SetMacParam、MCC_SetEncoderConfig、MCC_SetHomeConfig 與 MCC_SetCompParam 之外，在使用 MCCL 其他函式之前，必須先呼叫此函式或 **MCC_InitSystemEx**。此函式僅須呼叫一次即可。

Parameters *nInterpolateTime* 插值時間，單位為 ms，設定範圍為 1ms ~ 50ms，較小的插值時間會使 MCCL 擁有較佳的運作性能，但須視系統能否負荷而定。一般系統使用 2ms 的設定值即可。

pstCardConfig 運動控制卡硬體參數，對於硬體參數的詳細說明請參考”IMP Series 運動控制函式庫使用手冊”。

wCardNo 運動控制卡的使用張數(1 ~ 6)

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

15. int MCC_InitSystemEx(

double *nInterpolateTime*,
SYS_CARD_CONFIG* *pstCardConfig*,
WORD *wCardNo*

)

Description 啟動運動控制函式庫。

除了 `MCC_CreateGroup`、`MCC_CreateGroupEx`、`MCC_SetMacParam`、`MCC_SetEncoderConfig`、`MCC_SetHomeConfig` 與 `MCC_SetCompParam` 之外，在使用 MCCL 其他函式之前，必須先呼叫此函式或 `MCC_InitSystem`。此函式僅須呼叫一次即可。

Parameters	<i>nInterpolateTime</i>	插值時間，單位為 ms，設定範圍為 1ms ~ 50ms，較小的插值時間會使 MCCL 擁有較佳的運作性能，但須視系統能否負荷而定。一般系統使用 2ms 的設定值即可。
	<i>pstCardConfig</i>	運動控制卡硬體參數，對於硬體參數的詳細說明請參考”IMP Series 運動控制函式庫使用手冊”。
	<i>wCardNo</i>	運動控制卡的使用張數(1 ~ 6)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

16. int MCC_CloseSystem()

Description	關閉運動控制函式庫。	
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

17. int MCC_ResetMotion()

Description	重置運動控制函式庫。在使用此函式後，將清除錯誤狀態，並使直角座標值與馬達座標值歸零，最後系統會回復到如同呼叫 <code>MCC_InitSystem</code> 過後的初始狀態。	
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

18. int MCC_EnableDryRun()

Description	開啟運動空跑功能。開啟此項功能後，仍會進行運動命令的計算，但並不輸出計算後的結果，可利用 <code>MCC_GetCurPos</code> 與 <code>MCC_GetPulsePos</code>	
-------------	--	--

讀取所需的座標值進行分析或繪圖。

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

19. int MCC_DisableDryRun()

Description 關閉運動空跑功能。

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

20. int MCC_CheckDryRun()

Description 檢查運動空跑功能設定狀態。

Return Value	0	運動空跑功能已開啟
	1	運動空跑功能關閉中
	其他	失敗，傳回值的意義請參考 IV.函式傳回值

21. int MCC_SetSysMaxSpeed(double *dfMaxSpeed*)

Description 設定一般運動(直線、圓弧、圓、螺線)進給速度的上限，避免在使用 MCC_SetFeedSpeed 時所設定的進給速度超出系統的工作範圍，使用單位為 User Unit/sec(註)。

Parameters *dfMaxSpeed* 進給速度的上限

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

22. double MCC_GetSysMaxSpeed()

Description 讀取一般運動(直線、圓弧、圓、螺線)進給速度的上限，使用單位為 User Unit/sec。

Return Value 進給速度的上限

23. duoble MCC_GetConvFactor(**WORD** *wChannel,***WORD** *wCardIndex*

)

Description 讀取硬體參數(PPR * Gear Ratio / Pitch)。

Parameters *wChannel* 運動控制卡的輸出 Channel(0 ~ 7)*wCardIndex* 運動控制卡的編號(0 ~ 5)

Return Value 硬體參數(PPR * Gear Ratio / Pitch)值

24. int MCC_InitSimulation(**double** *nInterpolateTime,***SYS_CARD_CONFIG*** *pstCardConfig,***WORD** *wCardNo*

)

Description 啟動運動控制函式庫模擬功能。

除了 MCC_CreateGroup、MCC_CreateGroupEx、MCC_SetMacParam、MCC_SetEncoderConfig、MCC_SetHomeConfig 與 MCC_SetCompParam 之外，在使用 MCCL 其他函式之前，必須先呼叫此函式，此函式僅具有軟體模擬功能。此函式僅須呼叫一次即可。

Parameters *nInterpolateTime* 插值時間，單位為 ms，設定範圍為 1ms ~ 50ms，較小的插值時間會使 MCCL 擁有較佳的運作性能，但須視系統能否負荷而定。一般系統使用 2ms 的設定值即可。*pstCardConfig* 運動控制卡硬體參數，對於硬體參數的詳細說明請參考”IMP Series 運動控制函式庫使用手冊”。*wCardNo* 運動控制卡的使用張數(1 ~ 6)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值



*註：User Unit 即使用者在設定機構參數時，自行選定之長(角)度單位(例：dfPitch、dfHighLimit、dfLowLimit)。一經選定，MCCL 內部亦全部使用同樣長(角)度單位；以下以 UU 代表之。

B. 近端輸出入接點 (Local I/O) 控制

1. **int** MCC_SetServoOn(
 WORD *wChannel*,
 WORD *wCardIndex*,
)

Description 開啟伺服系統。

Parameters *wChannel* 運動控制卡的輸出 Channel(0 ~ 7)

wCardIndex 運動控制卡編號(0 ~ 5)

Return Value 0 成功

 非零 失敗，傳回值的意義請參考 IV.函式傳回值

2. **int** MCC_SetServoOff(
 WORD *wChannel*,
 WORD *wCardIndex*,
)

Description 關閉伺服系統。

Parameters *wChannel* 運動控制卡的輸出 Channel(0 ~ 7)

wCardIndex 運動控制卡編號(0 ~ 5)

Return Value 0 成功

 非零 失敗，傳回值的意義請參考 IV.函式傳回值

3. **int** MCC_EnablePosReady(
 WORD *wCardIndex*,
)

Description 由運動控制卡上的 Position Ready 輸出接點輸出訊號。

Parameters *wCardIndex* 運動控制卡編號(0 ~ 5)

Return Value 0 成功

 非零 失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_DisablePosReady(

WORD *wCardIndex*

)

Description 停止由運動控制卡上的 Position Ready 輸出接點輸出訊號。

Parameters *wCardIndex* 運動控制卡編號(0 ~ 5)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_GetEmgcStopStatus(

WORD* *pwStatus*,

WORD *wCardIndex*

)

Description 讀取緊急停止開關的輸入狀態。要啟動此項功能請參考相關硬體使用手冊對 Emergency Stop 的設定方式，IMP 在出廠時將 JP1 短路，當系統配接好 Emergency Stop 電路時，必須將 JP1 開路，以免 Emergency Stop 不會有動作。

Parameters *pwStatus* 指向一 WORD 值，內容代表緊急停止開關輸入狀態值

0 無觸發

1 觸發緊急停止開關

wCardIndex 運動控制卡編號(0 ~ 5)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_SetLIORoutine(

LIOISR *pfnLIORoutine*,

WORD *wCardIndex*

)

Description 串接自訂的近端輸入接點(LIO)中斷服務函式(ISR)，詳細的說明請參考”IMP Series 運動控制函式庫使用手冊”。

Parameters *pfnLIORoutine* 自訂近端輸入接點中斷服務函式的函式指標



	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

7. int MCC_SetLIOTriggerType(

WORD *wTriggerType*,

WORD *wPoint*,

WORD *wCardIndex*

)

Description 設定近端輸入接點的訊號以何種型態觸發自訂的中斷服務函式。觸發型態可設定為上緣(Rising Edge)觸發、下緣(Falling Edge)觸發或是轉態(Level Change)觸發。

Parameters

wTriggerType 觸發型態，可設定為

LIO_INT_RISE	上緣觸發
LIO_INT_FALL	下緣觸發
LIO_INT_LEVEL	轉態觸發

wPoint 輸入接點編號，編號範圍 LIO_LDI_OTP0 ~ LIO_LDI_OTP7、LIO_LDI_OTN0 ~ LIO_LDI_OTN7、LIO_LDI_HOME0 ~ LIO_LDI_HOME7，輸入訊號的意義請參考”IMP Series 運動控制函式庫使用手冊”。

	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

8. int MCC_EnableLIOTrigger(

WORD *wPoint*,

WORD *wCardIndex*

)

Description 開啟近端輸入接點訊號觸發自訂中斷服務函式的功能。

Parameters	<i>wPoint</i>	近端輸入接點編號，編號範圍 LIO_LDI_OTP0 ~ LIO_LDI_OTP7、LIO_LDI_OTN0 ~ LIO_LDI_OTN7、LIO_LDI_HOME0 ~ LIO_LDI_HOME7，輸入訊號的意義請參考”IMP Series 運動控制函式庫使用手冊”。
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

9. int MCC_DisableLIOTrigger(WORD *wPoint*, WORD *wCardIndex*)

Description 關閉近端輸入接點訊號觸發自訂中斷服務函式的功能。

Parameters	<i>wPoint</i>	近端輸入接點編號，編號範圍 LIO_LDI_OTP0 ~ LIO_LDI_OTP7、LIO_LDI_OTN0 ~ LIO_LDI_OTN7、LIO_LDI_HOME0 ~ LIO_LDI_HOME7，輸入訊號的意義請參考”IMP Series 運動控制函式庫使用手冊”。
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

10. int MCC_SetLedLightOn(WORD *wChannel*, WORD *wCardIndex*)

Description 開啟 LED 燈輸出。

Parameters	<i>wChannel</i>	運動控制卡的 LED 輸出 Channel (0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

11. int MCC_SetLedLightOff(

WORD *wChannel*,

WORD *wCardIndex*

)

Description 關閉 LED 燈輸出。

Parameters *wChannel* 運動控制卡的 LED 輸出 Channel (0 ~ 7)

wCardIndex 運動控制卡編號(0 ~ 5)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

12. int MCC_GetLedLightStatus(

WORD *wChannel*,

WORD* *Status*,

WORD *wCardIndex*

)

Description 讀取目前 LED 燈輸出狀態。

Parameters *wChannel* 運動控制卡的 LED 輸出 Channel(0 ~ 7)

Status 指向一 WORD 值，內容代表 LED 燈輸入狀態值

0 無觸發

1 觸發 LED 燈

wCardIndex 運動控制卡編號(0 ~ 5)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

13. int MCC_QueueLIOOutputValue(

WORD *wBitMask*,

WORD *wValue*,

WORD *wGroupIndex*

)

Description 將指定近端輸出接點之訊號狀態值存放至運動命令佇列(Queue)中。成功呼叫此函式將增加運動命令的庫存數目



Parameters	<i>wBitMask</i>	欲輸出訊號之近端輸出點編號，可設定多組接點同時輸出
	<i>wValue</i>	設定每個接點的輸出狀態
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

C. 座標系統

1. **int MCC_SetAbsolute(
 WORD *wGroupIndex*
)**

Description 使用絕對座標模式。

Parameters *wGroupIndex* Group 編號

Return Value 0 成功
 非零 失敗，傳回值的意義請參考 IV.函式傳回值

2. **int MCC_SetIncrease(
 WORD *wGroupIndex*
)**

Description 使用增量座標模式。

Parameters *wGroupIndex* Group 編號

Return Value 0 成功
 非零 失敗，傳回值的意義請參考 IV.函式傳回值

3. **int MCC_GetCoordType(
 WORD *wGroupIndex*
)**

Description 讀取使用的座標模式。

Parameters *wGroupIndex* Group 編號

Return Value 0 使用增量座標
 1 使用絕對座標
 其他 失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_GetCurRefPos(

```
double* pdfX,  
double* pdfY,  
double* pdfZ,  
double* pdfU,  
double* pdfV,  
double* pdfW,  
double* pdfA,  
double* pdfB,  
WORD wGroupIndex
```

)

Description 讀取目前各軸位置的直角座標值(不含補償)。

Parameters pdfX ~ pdfB 指向一 double 值，用來存放 X ~ B 各軸目前位置之直角座標值 (不含補償)

wGroupIndex Group 編號

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_GetCurPos(

```
double* pdfX,  
double* pdfY,  
double* pdfZ,  
double* pdfU,  
double* pdfV,  
double* pdfW,  
double* pdfA,  
double* pdfB,  
WORD wGroupIndex
```

)

Description 讀取目前各軸位置的直角座標值(含補償)。

Parameters pdfX ~ pdfB 指向一 double 值，用來存放 X ~ B 各軸目前位置之直角座標值 (含補償)

	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_GetPulsePos(

long* *plX*,
long* *plY*,
long* *plZ*,
long* *plU*,
long* *plV*,
long* *plW*,
long* *plA*,
long* *plB*,
WORD *wGroupIndex*

)

Description	讀取目前各軸位置的馬達座標值(或稱為 <i>Pulse</i> 座標值，含補償)。	
Parameters	<i>plX</i> ~ <i>plB</i>	指向一 long 值，用來存放 X ~ B 各軸目前位置的馬達座標值(含補償)
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

7. int MCC_DefineOrigin(

WORD *wAxis*,
WORD *wGroupIndex*

)

Description	使特定 Group 指定的運動軸之座標值歸零，使用此函式須在指定的 Group 為運動停止狀態。	
Parameters	<i>wAxis</i>	指定的運動軸編號 0 ~ 7 分別代表 X ~ B 軸
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功



非零

失敗，傳回值的意義請參考 IV.函式傳回值

8. **int** MCC_DefinePosHere(
 WORD *wGroupIndex* ,
 DWORD *dwAxisMask*
)

Description

修正目前的系統座標值使得與機台實際的位置相符合。

在某些情況下有可能使用手動方式移動機台，這時候機台實際的位置與運動控制函式庫中的系統座標值將產生差異，假使系統有安裝編碼器，則在呼叫此函式成功後將利用編碼器的計數值修正系統座標值，此時系統座標值將反應機台真實的位置。

Parameters

wGroupIndex Group 編號

dwAxisMask 指定欲發生作用的軸，指定參數可為

IMP_AXIS_X X 軸

IMP_AXIS_Y Y 軸

IMP_AXIS_Z Z 軸

IMP_AXIS_U U 軸

IMP_AXIS_V V 軸

IMP_AXIS_W W 軸

IMP_AXIS_A A 軸

IMP_AXIS_B B 軸

IMP_AXIS_ALL 全部運動軸

以上參數可自由組合，以作用在 X、Z 與 V 軸上為例：

`(IMP_AXIS_X | IMP_AXIS_Z | IMP_AXIS_V)`

Return Value

0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

9. **int** MCC_DefinePos(
 WORD *wAxis* ,
 DOUBLE *dfPos* ,



WORD *wGroupIndex*

)

Description

設定目前的系統座標值。

在某些情況下有可能機台需要重新開機，這時候控制卡上的編碼器位置將被歸零，當運動控制函式庫重新初始化後，此時機台實際的位置與運動控制函式庫中的系統座標值將產生差異，假使系統**有安裝絕對值編碼器**，則在呼叫此函式成功後將利用絕對值編碼器的計數值修正系統座標值，此時系統座標值將反應機台真實的位置。

Parameters

wAxis 指定的運動軸編號 0 ~ 7 分別代表 X ~ B 軸
dfPos 設定的系統座標值
wGroupIndex Group 編號

Return Value

0 成功
非零 失敗，傳回值的意義請參考 **IV.函式傳回值**

D. 過行程保護

1. int MCC_EnableLimitSwitchCheck(

int *nMode*

)

Description

開啟極限開關保護功能，機構參數中的 *wOverTravelUpSensorMode* 與 *wOverTravelDownSensorMode* 須設定為 0(Normal Open)或 1 (Normal Close)，如此才能正確執行此項功能。

當功能開啟時，在碰觸到該軸運動方向的極限開關時(例如往正方向移動且觸到正向極限開關，或往負方向移動且碰觸到負向極限開關)，將會停止輸出 Group 的運動命令(並產生錯誤記錄)。

MCC_EnableLimitSwitchCheck() 通常會與 MCC_GetErrorCode() 搭配使用，利用不斷呼叫 MCC_GetErrorCode() 可獲知系統是否因碰觸到極限開關而產生錯誤記錄(代碼 0xF701~0xF708 分別代表 X~B 軸碰觸極限開關)；當發現碰觸極限開關之錯誤時，一般作法可能是：在螢幕上顯示訊息告知操作員，然後在程式中呼叫 MCC_ClearError() 清除錯誤記錄，則此時系統可再往反方向退出極限開關。

Parameters

nMode

硬體極限開關保護模式，可設定為：

- 0 只要碰觸到極限開關，皆會停止輸出該軸之運動命令。
- 1 只有在碰觸到該軸運動方向的極限開關時(例如往正方向移動且觸到正向極限開關，或往負方向移動且碰觸到負向極限開關)，才會停止輸出該軸之運動命令。
- 2 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。
- 3 只有在碰觸到該軸運動方向的極限開關時(例如往正方向移動且觸到正向極限開關，或往負方向移

動且碰觸到負向極限開關)，才會停止輸出該軸之運動命令；並且會產生錯誤記錄。

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

2. int MCC_DisableLimitSwitchCheck()

Description 關閉極限開關保護功能。

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

3. int MCC_SetOverTravelCheck(

int *nOTCheck0*,
int *nOTCheck1*,
int *nOTCheck2*,
int *nOTCheck3*,
int *nOTCheck4*,
int *nOTCheck5*,
int *nOTCheck6*,
int *nOTCheck7*,
WORD *wGroupIndex*

)

Description 設定軟體過行程保護功能。

Parameters *nCheckOT0*, *nCheckOT1*, *nCheckOT2*, *nCheckOT3*, *nCheckOT4*, *nCheckOT5*, *nCheckOT6*, *nCheckOT7* 為設定參數，1 表示要開啟此軸軟體過行程保護的功能，0 則表示要關閉此項功能。

wGroupIndex Group 編號

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_GetOverTravelCheck(

```
int*   pnOTChk0,  
int*   pnOTChk1,  
int*   pnOTChk2,  
int*   pnOTChk3,  
int*   pnOTChk4,  
int*   pnOTChk5,  
int*   pnOTChk6,  
int*   pnOTChk7,  
WORD   wGroupIndex
```

)

Description	讀取軟體過行程保護的設定情形。	
Parameters	<i>pnOTChk0</i> ~ <i>pnOTChk7</i> 指向一 int 值，用來存放 X~B 各軸目前軟體過行程保護的設定情形，1 表示開啟，0 則表示關閉	
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_GetLimitSwitchStatus(

```
WORD*   pwStatus,  
WORD    wUpDown,  
WORD    wChannel,  
WORD    wCardIndex
```

)

Description	讀取極限開關的狀態，使用此函式前須先正確定義極限開關的配線方式，配線方式被定義在機構參數中的 <i>wOverTravelUpSensorMode</i> 與 <i>wOverTravelDownSensorMode</i> 。	
Parameters	<i>pwStatus</i>	指向一 WORD 值，用來存放欲讀取之極限開關的狀態，1 表示目前碰觸到極限開關，0 則否
	<i>wUpDown</i>	0 表示讀取負向極限開關的狀態，1 表示讀取正向極限開關的狀態

	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_GetLatchedLimitSwitchStatus(

WORD* *pwStatus*,
WORD *wUpDown*,
WORD *wChannel*,
WORD *wCardIndex*

)

Description 讀取極限開關的閘鎖狀態，當極限開關被碰觸，極限狀態將被閘鎖，此函式可讀取極限開關的閘鎖狀態。使用此函式前須先正確定義極限開關的配線方式，配線方式被定義在機構參數中的 *wOverTravelUpSensorMode* 與 *wOverTravelDownSensorMode*。

Parameters

<i>pwStatus</i>	指向一 WORD 值，用來存放欲讀取之極限開關的狀態，1 表示曾碰觸到極限開關，0 則否
<i>wUpDown</i>	0 表示讀取負向極限開關的狀態，1 表示讀取正向極限開關的狀態
<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

7. int MCC_ClearLatchedLimitSwitchStatus(

WORD *wUpDown*,
WORD *wChannel*,
WORD *wCardIndex*

)

Description 清除極限開關的閘鎖狀態，當極限開關被碰觸，極限狀態將被



閃鎖，此函式可清除極限開關的閃鎖狀態。

Parameters	<i>wUpDown</i>	0 表示清除負向極限開關的閃鎖狀態,1 表示清除正向極限開關的閃鎖狀態
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

E. 直線、圓弧、圓、螺線運動(一般運動)

```
1. int MCC_SetAccType(  
    char  cAccType,  
    WORD  wGroupIndex  
)
```

Description 設定在進行一般運動時的加速型式。

Parameters *cAccType* 各軸的加速型式，可設定為：

'T' 使用梯形加速曲線

'S' 使用 S 形加速曲線

wGroupIndex Group 編號

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

```
2. int MCC_GetAccType(  
    WORD wGroupIndex  
)
```

Description 讀取在進行一般運動時使用的加速型式。

Parameters *wGroupIndex* Group 編號

Return Value 0 目前使用梯形加速曲線

1 目前使用 S 形加速曲線

```
3. int MCC_SetDecType(  
    char  cDecType,  
    WORD  wGroupIndex  
)
```

Description 設定在進行一般運動時的減速型式。

Parameters *cDecType* 各軸的減速型式，可設定為：

'T' 使用梯形減速曲線

		'S'	使用 S 形減速曲線
	<i>wGroupIndex</i>	Group 編號	
Return Value	0	成功	
	非零	失敗，傳回值的意義請參考 IV.函式傳回值	

4. int MCC_GetDecType(WORD *wGroupIndex*)

Description	讀取在進行一般運動時使用的減速型式。		
Parameters	<i>wGroupIndex</i>	Group 編號	
Return Value	0	目前使用梯形減速曲線	
	1	目前使用 S 形減速曲線	

5. int MCC_SetAccTime(double *dfAccTime*, WORD *wGroupIndex*)

Description	設定在進行一般運動時加速到穩定速度所需的時間。		
Parameters	<i>dfAccTime</i>	要求的加速時間，必須大於 0，單位為 ms。	
	<i>wGroupIndex</i>	Group 編號	
Return Value	0	成功	
	非零	失敗，傳回值的意義請參考 IV.函式傳回值	

6. double MCC_GetAccTime(WORD *wGroupIndex*)

Description	讀取在進行一般運動時加速到穩定速度所需的時間。		
Parameters	<i>wGroupIndex</i>	Group 編號	
Return Value	進行一般運動時加速到穩定速度所需的時間，單位為 ms。		

7. int MCC_SetDecTime(
 double *dfDecTime*,
 WORD *wGroupIndex*
)

Description 設定在進行一般運動時由穩定速度減速至停止運動所需的時間。

Parameters *dfDecTime* 要求的減速時間，必須大於0，單位為 ms。

wGroupIndex Group 編號

Return Value 0 成功

 非零 失敗，傳回值的意義請參考 **IV.函式傳回值**

8. double MCC_GetDecTime(
 WORD *wGroupIndex*
)

Description 讀取在進行一般運動時由穩定速度減速至停止運動所需的時間。

Parameters *wGroupIndex* Group 編號

Return Value 進行一般運動時由穩定速度減速至停止運動所需的時間，單位為 ms。

9. double MCC_SetFeedSpeed(
 double *dfFeedSpeed*,
 WORD *wGroupIndex*
)

Description 設定在進行一般運動時的進給速度，單位為 UU/sec；此值不可為 0。但一般運動實際的進給速度(可使用 `MCC_GetCurFeedSpeed()` 獲得)須參考是否曾使用 `MCC_OverrideSpeed()` 設定進給速度強制比例，例如最後一次設定進給速度倍強制比例使用 `MCC_OverrideSpeed(150)`，此時呼叫 `MCC_SetFeedSpeed(10)`，則一般運動實際使用的進給速度為 $10 \times 150\% = 15$ 。

Parameters *dfFeedSpeed* 要求的進給速度

wGroupIndex Group 編號

Return Value 實際設定的進給速度



10. double MCC_GetFeedSpeed(

WORD *wGroupIndex*

)

Description 讀取在進行一般運動時所設定的進給速度。利用此函式所獲得的進給速度並未加入使用 MCC_OverrideSpeed()後對實際進給速度的影響，而只是單純傳回 MCC_SetFeedSpeed()時所使用的設定值。

Parameters *wGroupIndex* Group 編號

Return Value 目前設定的進給速度

11. double MCC_GetCurFeedSpeed(

WORD *wGroupIndex*

)

Description 讀取機台目前實際的進給速度。

Parameters *wGroupIndex* Group 編號

Return Value 機台目前實際的進給速度

12. int MCC_GetSpeed(

double* *pdfVel0*,

double* *pdfVel1*,

double* *pdfVel2*,

double* *pdfVel3*,

double* *pdfVel4*,

double* *pdfVel5*,

double* *pdfVel6*,

double* *pdfVel7*,

WORD *wGroupIndex*

)

Description 讀取各軸目前的進給速度值。

Parameters *pdfVel0* ~ *pdfVel7* 指向一 double 值，用來存放欲讀取之各軸目前的進給速度值

	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

13. int MCC_Line(

double *dfX0*,
double *dfX1*,
double *dfX2*,
double *dfX3*,
double *dfX4*,
double *dfX5*,
double *dfX6*,
double *dfX7*,
WORD *wGroupIndex* ,
DWORD *dwAxisMask*

)

Description 從目前位置以直線運動方式移動到指定的目的點。成功呼叫此函式將增加運動命令的庫存數目。

Parameters	<i>dfX0 ~ dfX7</i>	目的點的座標值
	<i>wGroupIndex</i>	Group 編號
	<i>dwAxisMask</i>	指定欲發生作用的軸，指定參數可為
	IMP_AXIS_X	X 軸
	IMP_AXIS_Y	Y 軸
	IMP_AXIS_Z	Z 軸
	IMP_AXIS_U	U 軸
	IMP_AXIS_V	V 軸
	IMP_AXIS_W	W 軸
	IMP_AXIS_A	A 軸
	IMP_AXIS_B	B 軸
	IMP_AXIS_ALL	全部運動軸

以上參數可自由組合，以作用在 X、Z 與 V 軸上為例：

(IMP_AXIS_X | IMP_AXIS_Z | IMP_AXIS_V)

Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

14. int MCC_ArcXYZ(

double *dfRX0*,
double *dfRX1*,
double *dfRX2*,
double *dfX0*,
double *dfX1*,
double *dfX2*,
WORD *wGroupIndex*

)

Description	在 X-Y-Z 軸所構成的空間上，以圓弧運動的方式從目前位置經過指定的參考點到目的點。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfRX0</i> ~ <i>dfRX2</i>	參考點的 X-Y-Z 軸座標值
	<i>dfX0</i> ~ <i>dfX2</i>	目的點的 X-Y-Z 軸座標值
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

15. int MCC_ArcXY(

double *dfRX0*,
double *dfRX1*,
double *dfX0*,
double *dfX1*,
WORD *wGroupIndex*

)

Description	在 X-Y 軸所構成的平面上，以圓弧運動的方式從目前位置經過指定的參考點到目的點。成功呼叫此函式將增加運動命令的庫存數目。	
-------------	---	--



Parameters	<i>dfRX0, dfRX1</i>	參考點的 X-Y 軸座標值
	<i>dfX0, dfX1</i>	目的點的 X-Y 軸座標值
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

16. int MCC_ArcYZ(

double *dfRX1*,
double *dfRX2*,
double *dfX1*,
double *dfX2*,
WORD *wGroupIndex*

)

Description	在 Y-Z 軸所構成的平面上，以圓弧運動的方式從目前位置經過指定的參考點到目的點。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfRX1, dfRX2</i>	參考點的 Y-Z 軸座標值
	<i>dfX1, dfX2</i>	目的點的 Y-Z 軸座標值
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

17. int MCC_ArcZX(

double *dfRX2*,
double *dfRX0*,
double *dfX2*,
double *dfX0*,
WORD *wGroupIndex*

)

Description	在 Z-X 軸所構成的平面上，以圓弧運動的方式從目前位置經過指定的參考點到目的點。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfRX2, dfRX0</i>	參考點的 Z-X 軸座標值

	<i>dfX2, dfX0</i>	目的點的 Z-X 軸座標值
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

18. int MCC_ArcXYZ_Aux(

double *dfRX0*,
double *dfRX1*,
double *dfRX2*,
double *dfX0*,
double *dfX1*,
double *dfX2*,
double *dfX3*,
double *dfX4*,
double *dfX5*,
double *dfX6*,
double *dfX7*,
WORD *wGroupIndex*

)

Description	在 X-Y-Z 軸所構成的空間上，以圓弧運動的方式從目前位置經過指定的參考點到目的點，且 U、V、W、A、B 軸以同動方式進行直線運動。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfRX0 ~dfRX2</i>	參考點的 X-Y-Z 軸座標值
	<i>dfX0~dfX2</i>	目的點的 X-Y-Z 軸座標值
	<i>dfX3~dfX7</i>	目的點的 U-V-W-A-B 軸座標值
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值



19. int MCC_ArcXY_Aux(

```

double dfRX0,
double dfRX1,
double dfX0,
double dfX1,
double dfX3,
double dfX4,
double dfX5,
double dfX6,
double dfX7,
WORD wGroupIndex

```

)

Description	在 X-Y 軸所構成的平面上，以圓弧運動的方式從目前位置經過指定的參考點到目的點，且 U、V、W、A、B 軸以同動方式進行直線運動。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfRX0, dfRX1</i>	參考點的 X-Y 軸座標值
	<i>dfX0, dfX1</i>	目的點的 X-Y 軸座標值
	<i>dfX3~dfX7</i>	目的點的 U-V-W-A-B 軸座標值
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

20. int MCC_ArcYZ_Aux(

```

double dfRX1,
double dfRX2,
double dfX1,
double dfX2,
double dfX3,
double dfX4,
double dfX5,
double dfX6,
double dfX7,

```



WORD *wGroupIndex*

)

Description	在 Y-Z 軸所構成的平面上，以圓弧運動的方式從目前位置經過指定的參考點到目的點，且 U、V、W、A、B 軸以同動方式進行直線運動。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfRX1, dfRX2</i>	參考點的 Y-Z 軸座標值
	<i>dfX1, dfX2</i>	目的點的 Y-Z 軸座標值
	<i>dfX3~dfX7</i>	目的點的 U-V-W-A-B 軸座標值
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

21. int MCC_ArcZX_Aux(

double *dfRX2,*

double *dfRX0,*

double *dfX2,*

double *dfX0,*

double *dfX3,*

double *dfX4,*

double *dfX5,*

double *dfX6,*

double *dfX7,*

WORD *wGroupIndex*

)

Description	在 Z-X 軸所構成的平面上，以圓弧運動的方式從目前位置經過指定的參考點到目的點，且 U、V、W、A、B 軸以同動方式進行直線運動。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfRX2, dfRX0</i>	參考點的 Z-X 軸座標值
	<i>dfX2, dfX0</i>	目的點的 Z-X 軸座標值
	<i>dfX3~dfX7</i>	目的點的 U-V-W-A-B 軸座標值
	<i>wGroupIndex</i>	Group 編號

Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

22. int MCC_ArcThetaXY(double *dfCX*, double *dfCY*, double *dfTheta*, WORD *wGroupIndex*)

Description 在 X-Y 軸所構成的平面上，以指定的圓心與移動角度進行圓弧運動。移動角度為負則進行順時針運動，移動角度為正則進行逆時針運動。成功呼叫此函式將增加運動命令的庫存數目。

Parameters

<i>dfCX, dfCY</i>	指定的圓心座標值
<i>dfTheta</i>	移動角度
<i>wGroupIndex</i>	Group 編號

Return Value

大於或等於零	MCCL 給予此運動命令的編碼
小於零	失敗，傳回值的意義請參考 IV.函式傳回值

23. int MCC_ArcThetaYZ(double *dfCY*, double *dfCZ*, double *dfTheta*, WORD *wGroupIndex*)

Description 在 Y-Z 軸所構成的平面上，以指定的圓心與移動角度進行圓弧運動。移動角度為負則進行順時針運動，移動角度為正則進行逆時針運動。成功呼叫此函式將增加運動命令的庫存數目。

Parameters

<i>dfCY, dfCZ</i>	指定的圓心座標值
<i>dfTheta</i>	移動角度
<i>wGroupIndex</i>	Group 編號

Return Value

大於或等於零	MCCL 給予此運動命令的編碼
--------	-----------------

小於零

失敗，傳回值的意義請參考 IV.函式傳回值

24. int MCC_ArcThetaZX(

double *dfCZ*,
double *dfCX*,
doubled *dfTheta*,
WORD *wGroupIndex*

)

Description	在 Z-X 軸所構成的平面上，以指定的圓心與移動角度進行圓弧運動。移動角度為負則進行順時針運動，移動角度為正則進行逆時針運動。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfCZ, dfCX</i>	指定的圓心座標值
	<i>dfTheta</i>	移動角度
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

25. int MCC_CircleXY(

double *dfCX*,
double *dfCY*,
BYTE *byCirDir*,
WORD *wGroupIndex*

)

Description	在 X-Y 軸所構成的平面上，由目前位置執行一個以指定點為圓心座標的全圓軌跡運動。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfCX, dfCY</i>	圓心的 X-Y 軸座標值
	<i>byCirDir</i>	運動方向，0 為順時針運動，1 為逆時針運動
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

26. int MCC_CircleYZ(

double *dfCY*,
double *dfCZ*,
BYTE *byCirDir*,
WORD *wGroupIndex*

)

Description	在 Y-Z 軸所構成的平面上，由目前位置執行一個以指定點為圓心座標的全圓軌跡運動。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfCY, dfCZ</i>	圓心的 Y-Z 軸座標值
	<i>byCirDir</i>	運動方向，0 為順時針運動，1 為逆時針運動
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

27. int MCC_CircleZX(

double *dfCZ*,
double *dfCX*,
BYTE *byCirDir*,
WORD *wGroupIndex*

)

Description	在 Z-X 軸所構成的平面上，由目前位置執行一個以指定點為圓心座標的全圓軌跡運動。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfCZ, dfCX</i>	圓心的 Z-X 軸座標值
	<i>byCirDir</i>	運動方向，0 為順時針運動，1 為逆時針運動
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

28. int MCC_CircleXY_Aux(

double *dfCX*,
double *dfCY*,

double *dfU*,
double *dfV*,
double *dfW*,
double *dfA*,
double *dfB*,
BYTE *byCirDir*,
WORD *wGroupIndex*

)

Description	在 X-Y 軸所構成的平面上，由目前位置執行一個以指定點為圓心座標的全圓軌跡運動，且 U、V、W、A、B 軸以同動方式進行直線運動。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfCX, dfCY</i>	圓心的 X-Y 軸座標值
	<i>dfU, dfV, dfW, dfA, dfB</i>	目的點的 U-V-W-A-B 軸座標值
	<i>byCirDir</i>	運動方向，0 為順時針運動，1 為逆時針運動
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

29. int MCC_CircleYZ_Aux(

double *dfCY*,
double *dfCZ*,
double *dfU*,
double *dfV*,
double *dfW*,
double *dfA*,
double *dfB*,
BYTE *byCirDir*,
WORD *wGroupIndex*

)

Description	在 Y-Z 軸所構成的平面上，由目前位置執行一個以指定點為圓心座標的全圓軌跡運動，且 U、V、W、A、B 軸以同動方式進行直線運動。成功	
-------------	--	--



呼叫此函式將增加運動命令的庫存數目。

Parameters	<i>dfCY, dfCZ</i>	圓心的 Y-Z 軸座標值
	<i>dfU, dfV, dfW, dfA, dfB</i>	目的點的 U-V-W-A-B 軸座標值
	<i>byCirDir</i>	運動方向，0 為順時針運動，1 為逆時針運動
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

30. int MCC_CircleZX_Aux(

```

double dfCZ,
double dfCX,
double dfU,
double dfV,
double dfW,
double dfA,
double dfB,
BYTE byCirDir,
WORD wGroupIndex

```

)

Description 在 Z-X 軸所構成的平面上，由目前位置執行一個以指定點為圓心座標的全圓軌跡運動，且 U、V、W、A、B 軸以同動方式進行直線運動。成功呼叫此函式將增加運動命令的庫存數目。

Parameters	<i>dfCZ, dfCX</i>	圓心的 Z-X 軸座標值
	<i>dfU, dfV, dfW, dfA, dfB</i>	目的點的 U-V-W-A-B 軸座標值
	<i>byCirDir</i>	運動方向，0 為順時針運動，1 為逆時針運動
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

31. int MCC_HelicalXY_Z()

double *dfCX*,
double *dfCY*,
double *dfPitch*,
double *dfTheta*,
WORD *wGroupIndex*

Description

由目前位置執行螺線運動，此運動在 X-Y 平面進行圓周運動，可利用 MCC_SetFeedSpeed() 設定此圓周運動的速度。使用此函式必須指定 X-Y 平面上圓周運動的圓心座標值，圓周半徑由目前位置與圓心座標值決定，並須規劃圓周運動移動的角度值，成功呼叫此函式將增加運動命令的庫存數目。

Parameters

<i>dfCX, dfCY</i>	圓周運動圓心的 X-Y 軸座標值
<i>dfPitch</i>	在 X-Y 平面進行一個整圓運動後，Z 軸所移動的距離，此值必須大於 0
<i>dfTheta</i>	圓周運動所移動的角度與運動方向，正值為順時針運動，負值為逆時針運動
<i>wGroupIndex</i>	Group 編號

Return Value

大於或等於零	MCCL 給予此運動命令的編碼
小於零	失敗，傳回值的意義請參考 IV. 函式傳回值

32. int MCC_HelicalYZ_X()

double *dfCY*,
double *dfCZ*,
double *dfPitch*,
double *dfTheta*,
WORD *wGroupIndex*

Description

由目前位置執行螺線運動，此運動在 Y-Z 平面進行圓周運動，可利用 MCC_SetFeedSpeed() 設定此圓周運動的速度。使用此函式必須指定 Y-Z 平面上圓周運動的圓心座標值，圓周半徑由目前位置與圓心座標值決

定，並須規劃圓周運動移動的角度值。成功呼叫此函式將增加運動命令的庫存數目。

Parameters	<i>dfCY, dfCZ</i>	圓周運動圓心的 Y-Z 軸座標值
	<i>dfPitch</i>	在 Y-Z 平面進行一個整圓運動後，X 軸所移動的距離，此值必須大於 0
	<i>dfTheta</i>	圓周運動所移動的角度與運動方向，正值為順時針運動，負值為逆時針運動
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

33. int MCC_HelicalZX_Y(

double *dfCZ*,
double *dfCX*,
double *dfPitch*,
double *dfTheta*,
WORD *wGroupIndex*

)

Description 由目前位置執行螺線運動，此運動在 Z-X 平面進行圓周運動，可利用 MCC_SetFeedSpeed() 設定此圓周運動的速度。使用此函式必須指定 Z-X 平面上圓周運動的圓心座標值，圓周半徑由目前位置與圓心座標值決定，並須規劃圓周運動移動的角度值。成功呼叫此函式將增加運動命令的庫存數目。

Parameters	<i>dfCZ, dfCX</i>	圓周運動圓心的 Z-X 軸座標值
	<i>dfPitch</i>	在 Z-X 平面進行一個整圓運動後，Y 軸所移動的距離，此值必須大於 0
	<i>dfTheta</i>	圓周運動所移動的角度與運動方向，正值為順時針運動，負值為逆時針運動
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼



小於零

失敗，傳回值的意義請參考 IV.函式傳回值

34. int MCC_HelicalXY_Z_Aux(

double *dfCX*,
double *dfCY*,
double *dfPitch*,
double *dfU*,
double *dfV*,
double *dfW*,
double *dfA*,
double *dfB*,
double *dfTheta*,
WORD *wGroupIndex*

)

Description

由目前位置執行螺線運動，此運動在 X-Y 平面進行圓周運動，且 U、V、W、A、B 軸以同動方式進行直線運動，可利用 MCC_SetFeedSpeed() 設定此圓周運動的速度。使用此函式必須指定 X-Y 平面上圓周運動的圓心座標值，圓周半徑由目前位置與圓心座標值決定，並須規劃圓周運動移動的角度值。成功呼叫此函式將增加運動命令的庫存數目。

Parameters

dfCX, dfCY 圓周運動圓心的 X-Y 軸座標值
dfPitch 在 X-Y 平面進行一個整圓運動後，Z 軸所移動的距離，此值必須大於 0
dfU, dfV, dfW, dfA, dfB 目的點的 U-V-W-A-B 軸座標值
dfTheta 圓周運動所移動的角度與運動方向，正值為順時針運動，負值為逆時針運動
wGroupIndex Group 編號

Return Value

大於或等於 0 MCCL 給予此運動命令的編碼
小於 0 失敗，傳回值的意義請參考 IV.函式傳回值

35. int MCC_HelicalYZ_X_Aux(

double *dfCY*,
double *dfCZ*,
double *dfPitch*,
double *dfU*,
double *dfV*,
double *dfW*,
double *dfA*,
double *dfB*,
double *dfTheta*,
WORD *wGroupIndex*

)

Description 由目前位置執行螺線運動，此運動在 Y-Z 平面進行圓周運動，且 U、V、W、A、B 軸以同動方式進行直線運動，可利用 MCC_SetFeedSpeed() 設定此圓周運動的速度。使用此函式必須指定 Y-Z 平面上圓周運動的圓心座標值，圓周半徑由目前位置與圓心座標值決定，並須規劃圓周運動移動的角度值。成功呼叫此函式將增加運動命令的庫存數目。

Parameters

<i>dfCY, dfCZ</i>	圓周運動圓心的 Y-Z 軸座標值
<i>dfPitch</i>	在 Y-Z 平面進行一個整圓運動後，X 軸所移動的距離，此值必須大於 0
<i>dfU, dfV, dfW, dfA, dfB</i>	目的點的 U-V-W-A-B 軸座標值
<i>dfTheta</i>	圓周運動所移動的角度與運動方向，正值為順時針運動，負值為逆時針運動
<i>wGroupIndex</i>	Group 編號

Return Value

大於或等於零	MCCL 給予此運動命令的編碼
小於零	失敗，傳回值的意義請參考 IV.函式傳回值

36. int MCC_HelicalZX_Y_Aux(

double *dfCZ*,
double *dfCX*,
double *dfPitch*,

double *dfU*,
double *dfV*,
double *dfW*,
double *dfA*,
double *dfB*,
double *dfTheta*,
WORD *wGroupIndex*

)

Description	由目前位置執行螺線運動，此運動在 Z-X 平面進行圓周運動，且 U、V、W、A、B 軸以同動方式進行直線運動，可利用 MCC_SetFeedSpeed() 設定此圓周運動的速度。使用此函式必須指定 Z-X 平面上圓周運動的圓心座標值，圓周半徑由目前位置與圓心座標值決定，並須規劃圓周運動移動的角度值。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfCZ, dfCX</i>	圓周運動圓心的 Z-X 軸座標值
	<i>dfPitch</i>	在 Z-X 平面進行一個整圓運動後，Y 軸所移動的距離，此值必須大於 0
	<i>dfU, dfV, dfW, dfA, dfB</i>	目的點的 U-V-W-A-B 軸座標值
	<i>dfTheta</i>	圓周運動所移動的角度與運動方向，正值為順時針運動，負值為逆時針運動
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV. 函式傳回值

F. 點對點運動

1. double MCC_SetPtPSpeed(

double *dfRatio*,
WORD *wGroupIndex*

)

Description	設定進行點對點運動時的速度比例。 進行點對點運動時各軸的速度(UU/sec) 等於 $((\text{馬達最大轉速}/60) \times \text{導螺桿間隙值} / \text{齒輪箱減速比}) \times (\text{速度比例} / 100)$ 其中馬達最大轉速(<i>wRPM</i>)、齒輪箱減速比(<i>dfPitch</i>)、齒輪減速比(<i>dfGearRatio</i>)皆定義在機構參數中。因此速度比例的計算可以利用： $(\text{要求的速度}) / (\text{馬達最大轉速能帶動機台的最大安全速度})$ 之百分比 $\times 100$ 。 但點對點運動實際操作時的進給速度需參考是否曾使用 <i>MCC_OverrideSpeed()</i> 設定運動進給速度倍率。	
Parameters	<i>dfRatio</i>	速度比例，其值必須大於 0 且小於等於 100
	<i>wGroupIndex</i>	Group 編號
Return Value	實際設定的速度比例	

2. double MCC_GetPtPSpeed(

WORD *wGroupIndex*

)

Description	讀取進行點對點運動時使用的速度比例。	
Parameters	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	進行點對點運動時使用的速度比例
	小於零	失敗，傳回值意義請參考 IV.函式傳回值

3. int MCC_PtP(

double *dfX0*,
double *dfX1*,



double *dfX2*,
double *dfX3*,
double *dfX4*,
double *dfX5*,
double *dfX6*,
double *dfX7*,
WORD *wGroupIndex*,
DWORD *dwAxisMask*

)

Description	使用點對點運動的方式，從目前位置以設定的進給速度比例移動至指定的目的點。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfX0, dfX1, dfX2</i>	目的點的 X-Y-Z 軸座標值
	<i>dfX3, dfX4, dfX5</i>	目的點的 U-V-W 軸座標值
	<i>dfX6, dfX7</i>	目的點的 A-B 軸座標值
	<i>wGroupIndex</i>	Group 編號
	<i>dwAxisMask</i>	指定欲發生作用的軸，指定參數可為
	IMP_AXIS_X	X 軸
	IMP_AXIS_Y	Y 軸
	IMP_AXIS_Z	Z 軸
	IMP_AXIS_U	U 軸
	IMP_AXIS_V	V 軸
	IMP_AXIS_W	W 軸
	IMP_AXIS_A	A 軸
	IMP_AXIS_B	B 軸
	IMP_AXIS_ALL	全部運動軸
	以上參數可自由組合，以作用在 X、Z 與 V 軸上為例：	
	<code>(IMP_AXIS_X IMP_AXIS_Z IMP_AXIS_V)</code>	
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值



4. int MCC_PtPX(

double *dfX*,
WORD *wGroupIndex*

)

Description 使用點對點運動的方式，從目前位置以設定的進給速度比例移動至指定的目的點。成功呼叫此函式將增加運動命令的庫存數目。

Parameters *dfX* 目的點的 X 軸座標值
wGroupIndex Group 編號

Return Value 大於或等於零 MCCL 給予此運動命令的編碼
小於零 失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_PtPY(

double *dfY*,
WORD *wGroupIndex*

)

Description 使用點對點運動的方式，從目前位置以設定的進給速度比例移動至指定的目的點。成功呼叫此函式將增加運動命令的庫存數目。

Parameters *dfY* 目的點的 Y 軸座標值
wGroupIndex Group 編號

Return Value 大於或等於零 MCCL 給予此運動命令的編碼
小於零 失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_PtPZ(

double *dfZ*,
WORD *wGroupIndex*

)

Description 使用點對點運動的方式，從目前位置以設定的進給速度比例移動至指定的目的點。成功呼叫此函式將增加運動命令的庫存數目。

Parameters *dfZ* 目的點的 Z 軸座標值
wGroupIndex Group 編號

Return Value 大於或等於零 MCCL 給予此運動命令的編碼

小於零

失敗，傳回值的意義請參考 IV.函式傳回值

7. int MCC_PtPU(

double *dfU*,

WORD *wGroupIndex*

)

Description 使用點對點運動的方式，從目前位置以設定的進給速度比例移動至指定的目的點。成功呼叫此函式將增加運動命令的庫存數目。

Parameters *dfU* 目的點的 U 軸座標值

wGroupIndex Group 編號

Return Value 大於或等於零 MCCL 給予此運動命令的編碼

小於零 失敗，傳回值的意義請參考 IV.函式傳回值

8. int MCC_PtPV(

double *dfV*,

WORD *wGroupIndex*

)

Description 使用點對點運動的方式，從目前位置以設定的進給速度比例移動至指定的目的點。成功呼叫此函式將增加運動命令的庫存數目。

Parameters *dfV* 目的點的 V 軸座標值

wGroupIndex Group 編號

Return Value 大於或等於零 MCCL 給予此運動命令的編碼

小於零 失敗，傳回值的意義請參考 IV.函式傳回值

9. int MCC_PtPW(

double *dfW*,

WORD *wGroupIndex*

)

Description 使用點對點運動的方式，從目前位置以設定的進給速度比例移動至指定的目的點。成功呼叫此函式將增加運動命令的庫存數目。

Parameters *dfW* 目的點的 W 軸座標值



	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

10. int MCC_PtPA(

double *dfA*,
WORD *wGroupIndex*

)

Description	使用點對點運動的方式，從目前位置以設定的進給速度比例移動至指定的目的點。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfA</i>	目的點的 A 軸座標值
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

11. int MCC_PtPB(

double *dfB*,
WORD *wGroupIndex*

)

Description	使用點對點運動的方式，從目前位置以設定的進給速度比例移動至指定的目的點。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>dfB</i>	目的點的 B 軸座標值
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

12. int MCC_SetPtPAccType(char *cAccType0*, char *cAccType1*, char *cAccType2*, char *cAccType3*, char *cAccType4*, char *cAccType5*, char *cAccType6*, char *cAccType7*, WORD *wGroupIndex*

)



Description	設定在進行點對點運動時使用的加速型式，各軸使用各自獨立的加速型式。	
Parameters	<i>cAccType0</i> ~ <i>cAccType7</i> 各軸的加速型式，可設定為：	
	’T’	使用梯形加速曲線
	’S’	使用 S 形加速曲線
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

**13. int MCC_GetPtPAccType(char* *pcAccType0*, char* *pcAccType1*,
char* *pcAccType2*, char* *pcAccType3*,
char* *pcAccType4*, char* *pcAccType5*,
char* *pcAccType6*, char* *pcAccType7*,
WORD *wGroupIndex*)**

Description	讀取在進行點對點運動時使用的加速型式。	
Parameters	<i>pcAccType0</i> ~ <i>pcAccType7</i> 指向一 char 值，用來存放各軸的加速型式：	
	0	使用梯形加速曲線
	1	使用 S 形加速曲線
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

**14. int MCC_SetPtPDecType(char *cDecType0*, char *cDecType1*, char *cDecType2*,
char *cDecType3*, char *cDecType4*, char *cDecType5*,
char *cDecType6*, char *cDecType7*,
WORD *wGroupIndex*)**

Description	設定在進行點對點運動時使用的減速型式，各軸使用各自獨立的減速型式。	
Parameters	<i>cDecType0</i> ~ <i>cDecType7</i> 各軸的減速型式，可設定為：	
	’T’	使用梯形減速曲線

'S' 使用 S 形減速曲線

	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

**15. int MCC_GetPtPDecType(char* *pcDecType0*, char* *pcDecType1*,
char* *pcDecType2*, char* *pcDecType3*,
char* *pcDecType4*, char* *pcDecType5*,
char* *pcDecType6*, char* *pcDecType7*,
WORD *wGroupIndex*)**

Description 讀取在進行點對點運動時使用的減速型式。

Parameters *pcDecType0* ~ *pcDecType7* 指向一 char 值，用來存放各軸的減速型式

0 使用梯形減速曲線

1 使用 S 形減速曲線

	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

**16. int MCC_SetPtPAccTime(double *dfAccTime0*, double *dfAccTime1*,
double *dfAccTime2*, double *dfAccTime3*,
double *dfAccTime4*, double *dfAccTime5*,
double *dfAccTime6*, double *dfAccTime7*,
WORD *wGroupIndex*)**

Description 設定在進行點對點運動時加速到穩定速度所需的時間，各軸使用各自獨立的加速時間。

Parameters *dfAccTime0* ~ *dfAccTime7* 各軸的加速時間，必須大於 0，單位為 ms

wGroupIndex Group 編號

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

**17. int MCC_GetPtPAccTime(double* *pdfAccTime0*, double* *pdfAccTime1*,
double* *pdfAccTime2*, double* *pdfAccTime3*,
double* *pdfAccTime4*, double* *pdfAccTime5*,
double* *pdfAccTime6*, double* *pdfAccTime7*,
WORD *wGroupIndex*)**

Description 讀取在進行點對點運動時加速到穩定速度所需的時間，各軸使用各自獨立的加速時間。

Parameters *pdfAccTime0* ~ *pdfAccTime7* 指向一 double 值，用來存放各軸的加速時間，單位為 ms

wGroupIndex Group 編號

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

**18. int MCC_SetPtPDecTime(double *dfDecTime0*, double *dfDecTime1*,
double *dfDecTime2*, double *dfDecTime3*,
double *dfDecTime4*, double *dfDecTime5*,
double *dfDecTime6*,double *dfDecTime7*,
WORD *wGroupIndex*)**

Description 設定在進行點對點運動時由穩定速度減速到停止運動所需的時間，各軸使用各自獨立的減速時間。

Parameters *dfDecTime0* ~ *dfDecTime7* 各軸的減速時間，必須大於 0，單位為 ms

wGroupIndex Group 編號

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

**19. int MCC_GetPtPDecTime(double* *pdfDecTime0*, double* *pdfDecTime1*,
double* *pdfDecTime2*, double* *pdfDecTime3*,
double* *pdfDecTime4*, double* *pdfDecTime5*,
double* *pdfDecTime6*, double* *pdfDecTime7*,
WORD *wGroupIndex*)**

Description 讀取在進行點對點運動時由穩定速度減速到停止運動所需的時間，各軸



使用各自獨立的減速時間。

Parameters	<i>pdfDecTime0 ~ pdfDecTime7</i>	指向一 double 值，用來存放各軸的減速時間，單位為 ms
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

20. int MCC_SyncPtP(

double *dfX0*,
 double *dfX1*,
 double *dfX2*,
 double *dfX3*,
 double *dfX4*,
 double *dfX5*,
 double *dfX6*,
 double *dfX7*,
 WORD *wGroupIndex*,
 DWORD *dwAxisMask*

)

Description 使用同動點對點運動的方式，從目前位置以設定的進給速度及所需移動時間最長的座標軸比例移動至指定的目的點。成功呼叫此函式將增加運動命令的庫存數目。

Parameters	<i>dfX0, dfX1, dfX2</i>	目的點的 X-Y-Z 軸座標值
	<i>dfX3, dfX4, dfX5</i>	目的點的 U-V-W 軸座標值
	<i>dfX6, dfX7</i>	目的點的 A-B 軸座標值
	<i>wGroupIndex</i>	Group 編號
	<i>dwAxisMask</i>	指定欲發生作用的軸，指定參數可為
	IMP_AXIS_X	X 軸
	IMP_AXIS_Y	Y 軸
	IMP_AXIS_Z	Z 軸



IMP_AXIS_U	U 軸
IMP_AXIS_V	V 軸
IMP_AXIS_W	W 軸
IMP_AXIS_A	A 軸
IMP_AXIS_B	B 軸
IMP_AXIS_ALL	全部運動軸

以上參數可自由組合，以作用在 X、Z 與 V 軸上為例：

`(IMP_AXIS_X | IMP_AXIS_Z | IMP_AXIS_V)`

Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

G.JOG運動

1. int MCC_JogPulse(

int *nPulse*,
char *cAxis*,
WORD *wGroupIndex*

)

Description 微動(脈衝運動)。在其他運動命令皆已執行完成後(此時呼叫 MCC_GetMotionStatus()所獲得的傳回值應為 GMS_STOP),依照指定的位移量(Pulse 數)及方向帶動特定軸。

此函式為手動程式的細調模式且需處在運動停止狀態時呼叫才有效。脈衝運動並沒有包含加減速的動作,因此給定的位移量不宜過大,避免機台過度震動。

Parameters

<i>nPulse</i>	位移量,單位為 pulse,可給定範圍為-2048 ~ 2048
<i>cAxis</i>	要求進行脈衝運動的運動軸編號 (0 ~ 7 代表 X ~ B 軸)
<i>wGroupIndex</i>	Group 編號

Return Value

0	成功
非零	失敗,傳回值的意義請參考 IV.函式傳回值

2. int MCC_JogSpace(

double *dfOffset*,
int *nRatio*,
char *cAxis*,
WORD *wGroupIndex*

)

Description 吋動(單步運動)。在其他運動命令皆已執行完成後(此時呼叫 MCC_GetMotionStatus()所獲得的傳回值應為 GMS_STOP),依照指定的位移量(增量)及速度比例(與點對點運動的速度比例意義相同)帶動特定軸。

成功呼叫此函式將增加運動命令的庫存數目。

Parameters	<i>dfOffset</i>	位移量，單位為 UU
	<i>nRatio</i>	速度比例，其值必須大於 0 且小於等於 100
	<i>cAxis</i>	要求進行單步運動的運動軸之編號 (0 ~ 7 代表 X ~ B 軸)
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

3. int MCC_JogConti(

```
int   nDir,  
double dfRatio,  
char  cAxis,  
WORD  wGroupIndex
```

)

Description 連續吋動。在其他運動命令皆已執行完成後(此時呼叫 MCC_GetMotionStatus()所獲得的傳回值應為 GMS_STOP)，依照指定的方向及速度比例(與點對點運動的速度比例意義相同)帶動特定軸，並移動到有效工作區間的邊界才停止(機構參數定義了有效工作區間的範圍)。成功呼叫此函式將增加運動命令的庫存數目。

Parameters	<i>nDir</i>	連續吋動的方向，可設定為： 1 往正方向運動 -1 往負方向運動
	<i>dfRatio</i>	速度比例，其值必須大於 0 且小於等於 100
	<i>cAxis</i>	要求進行單步運動的運動軸之編號 (0 ~ 7 代表 X ~ B 軸)
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

H. 運動狀態檢視

1. **int** MCC_GetMotionStatus(
 WORD *wGroupIndex*
)

Description	檢視系統目前的運動狀態。	
Parameters	<i>wGroupIndex</i>	Group 編號
Return Value	0	處於運動狀態(GMS_RUNNING),代表尚有運動命令未執行完成
	1	處於停止狀態(GMS_STOP),已無庫存運動命令
	2	處於暫停狀態(GMS_HOLD),可能因使用者呼叫MCC_HoldMotion
	3	處於延遲狀態(GMS_DELAYING),可能因使用者呼叫MCC_DelayMotion
	其他	失敗,傳回值的意義請參考 IV.函式傳回值

2. **int** MCC_GetCurCommand(
 COMMAND_INFO* *pstCurCmdInfo*,
 WORD *wGroupIndex*
)

Description	讀取執行中的運動命令之相關資訊,包括運動命令之類型、運動命令編碼、要求的進給速度與目的點位置等。	
Parameters	<i>pstCurCmdInfo</i>	指向一 COMMAND_INFO 結構,用來存放執行中之運動命令內容,定義如下:

```
typedef struct _COMMAND_INFO  
{  
    int      nType;  
    int      nCommandIndex;  
    double   dfFeedSpeed;  
    double   dfPos[MAX_AXIS_NUM];  
    double   dfAcctime;  
    double   dfDectime;  
} COMMAND_INFO;
```



nType : 運動命令類型

0	點對點運動
1	直線運動
2	順時針圓弧、圓運動
3	逆時針圓弧、圓運動
4	順時針螺線運動
5	逆時針螺線運動
6	運動延遲命令
7	開啟平滑運動
8	關閉平滑運動
9	開啟定位確認
10	關閉定位確認

nCommandIndex : 運動命令編碼

dfFeedSpeed :

一般運動	規劃的進給速度
點對點運動	規劃的速度比例
運動延遲	目前剩餘的延遲時間(單位:ms)

dfPos[] : 目的點的絕對位置座標

dfAcctime : 加速時間, 單位為 ms

dfDectime : 減速時間, 單位為 ms

wGroupIndex Group 編號

Return Value	0	成功
	非零	失敗, 傳回值的意義請參考 IV.函式傳回值

3. int MCC_GetCommandCount(

int* *pnCmdCount*,
WORD *wGroupIndex*

)

Description 讀取運動命令緩衝區中尚未執行的運動命令之庫存數目。
關於哪些函式呼叫後會增加庫存命令, 請參考”**IMP Series 運動控制函式庫使用手冊 2.3 函式庫操作特性**”。

Parameters *pnCmdCount* 指向一 int 值, 用來存放運動命令庫存數目
wGroupIndex Group 編號

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_ResetCommandIndex(WORD *wGroupIndex*)

Description 使運動命令編碼值歸零。運動命令編碼相當於 MCCL 給於每一筆運動命令的識別資料，利用此函式可以運動命令編碼值從 0 開始計數。

Parameters *wGroupIndex* Group 編號

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_GetCurPulseStockCount(WORD* *pwStockCount*, WORD *wChannel*, WORD *wCardIndex*)

Description 讀取目前硬體上的 pulse 命令庫存數目。在運動過程中，利用此函式所獲得的 pulse 命令庫存數目不應小於 60，如此才能保有穩定的運動控制性能；如不能達到此要求，請增加插值時間(重新呼叫 MCC_InitSystem)。

Parameters *pwStockCount* 指向一 WORD 值，用來存放 Pulse 命令庫存數目
wChannel 運動控制卡的輸出 Channel(0 ~ 7)
wCardIndex 運動控制卡編號(0 ~ 5)

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_SetMaxPulseStockNum(int *nMaxStockNum*,)



Description	設定硬體 FIFO 的使用個數。硬體 FIFO 的使用個數應該搭配作業系統的即時性能一起評估，設定越少的使用個數，作業系統要具備的即時性能越強，設定越多的使用個數，作業系統對即時性能的容忍性越大，對運動控制性能的穩定性越高；如不能達到運動控制性能的穩定性的要求時，請設定較多的使用個數或增加插值時間。	
Parameters	<i>nMaxStockNum</i>	指向一 int 值，用來存放 FIFO 最大使用個數
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

7. int MCC_GetMaxPulseStockNum()

Description	讀取硬體 FIFO 的最大使用個數。	
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

8. int MCC_GetErrorCode(WORD *wGroupIndex*)

Description	讀取目前錯誤記錄，用來檢查系統運作時是否發生錯誤。 系統運作中應隨時(例如每 100ms)呼叫此函式確認系統目前工作正常，若發現錯誤記錄產生，則須採取相對應之錯誤回復(Error Recovery)處理。	
Parameters	<i>wGroupIndex</i>	Group 編號
Return Value	0	無錯誤
	其它	錯誤代碼(請參考 IV.錯誤訊息代碼)

9. int MCC_ClearError(WORD *wGroupIndex*)

Description	在系統運作發生錯誤後，若已排除這些錯誤，必須使用此函式清除系統內的錯誤記錄，否則系統仍無法正常運作。	
Parameters	<i>wGroupIndex</i>	Group 編號



Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

10. int MCC_SetCommandIndex(

WORD *wIndex*,
BOOL *bManualIndex*,
WORD *wGroupIndex*

)

Description	設定運動命令編碼值。	
Parameters	<i>wIndex</i>	編碼值數值
	<i>bManualIndex</i>	設定是否自動增加編碼值數值 true，編碼值為 <i>wIndex</i> false，編碼值為 <i>wIndex</i> +1
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

11. int MCC_GetCommandIndex(

WORD* *wIndex*,
WORD *wGroupIndex*

)

Description	取得運動命令編碼值。	
Parameters	<i>wIndex</i>	指向一 WORD 值用來存放編碼值
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

I. 原點復歸

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

Description 執行原點復歸運動，關於其設定請參考 `MCC_SetHomeConfig()`。使用此函式時，可配合呼叫 `MCC_GetGoHomeStatus()`，來檢查原點復歸是否完成。當完成原點復歸的運動後各軸的直角座標值將被設定為零。

Parameters `nOrder0 ~ nOrder7` 各軸進行原點復歸的順序。可設定值為 0~7，數字越小越早執行復歸運動；不執行原點復歸動作的運動軸，須將復歸順序設定為 255。須特別注意，這些參數所對應的是編號 `wCardIndex` 控制卡上的 0 ~ 7 號輸出軸，而不是 Group 中之運動軸。更詳細的說明請參考”*IMP Series 運動控制函式庫使用手冊 2.8 原點復歸*”

`wCardIndex` 運動控制卡編號(0 ~ 5)

Return Value 0 成功
非零 失敗，傳回值的意義請參考 **V.函式**

2. int MCC_GetGoHomeStatus()

Description 在呼叫 `MCC_Home()`後，使用此函式可檢視是否已完成原點復歸動作。



Return Value	0	尚未完成原點復歸動作
	1	完成原點復歸動作
	其它	失敗，傳回值的意義請參考 IV.函式傳回值

3. int MCC_AbortGoHome()

Description 當呼叫 MCC_Home()後，使用此函式可停止原點復歸的動作。

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_GetHomeSensorStatus(

WORD* *pwStatus*,
WORD *wChannel*,
WORD *wCardIndex*

)

Description 讀取 Home Sensor 的狀態，使用此函式前需先正確定義 Home Sensor 的配線方式(Normal Open 或 Normal Close)，配線方式被定義在原點復歸參數中。

Parameters	<i>pwStatus</i>	指向一 WORD 值，用來存放欲讀取 Home Sensor 的狀態，1 表示目前碰觸到 Home Sensor，0 則否。
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

J. 定位控制

```
1. int MCC_SetCompParam(  
    SYS_COMP_PARAM* pstCompParam,  
    WORD wChannel,  
    WORD wCardIndex  
)
```

Description 設定齒輪齒隙、間隙補償參數，使用者可先設定補償參數的內容，再利用此函式將補償參數傳入，最後呼叫 `MCC_UpdateCompParam()`，補償參數的內容必須涵蓋機台全部的工作行程，以避免產生不正常的動作。更詳細的說明請參考”*IMP Series 運動控制函式庫使用手冊 2.7.5 齒輪齒隙、背隙補償*”。

Parameters

<i>pstCompParam</i>	指向一 SYS_COMP_PARAM 結構，內含欲設定之補償參數
<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)

Return Value

0	成功
非零	失敗，傳回值的意義請參考 IV.函式傳回值

```
2. int MCC_UpdateCompParam()
```

Description 反應更新後的齒輪齒隙、間隙補償參數。在呼叫過 `MCC_SetCompParam()`後，須執行此函式系統才會反應新的設定值。

Return Value

0	成功
非零	失敗，傳回值的意義請參考 IV.函式傳回值

```
3. int MCC_SetPGain(  
    WORD wGain0,  
    WORD wGain1,  
    WORD wGain2,
```



WORD *wGain3*,
 WORD *wGain4*,
 WORD *wGain5*,
 WORD *wGain6*,
 WORD *wGain7*,
 WORD *wCardIndex*

)

Description 設定位置閉迴路控制使用的比例增益(P Gain)。

Parameters *wGain0~wGain7* 各軸使用的比例增益，設定範圍為 1 ~ 127
wCardIndex 運動控制卡編號(0 ~ 5)

Return Value 0 成功
 非零 失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_GetPGain(

WORD* *pwGain0*,
 WORD* *pwGain1*,
 WORD* *pwGain2*,
 WORD* *pwGain3*,
 WORD* *pwGain4*,
 WORD* *pwGain5*,
 WORD* *pwGain6*,
 WORD* *pwGain7*,
 WORD *wCardIndex*

)

Description 讀取位置閉迴路控制使用的比例增益(P Gain)。

Parameters *pwGain0~pwGain7* 指向一 WORD 值，用來存放各軸使用的比例增益
wCardIndex 運動控制卡編號(0 ~ 5)

Return Value 0 成功
 非零 失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_SetIGain(

WORD *wGain0*,
WORD *wGain1*,
WORD *wGain2*,
WORD *wGain3*,
WORD *wGain4*,
WORD *wGain5*,
WORD *wGain6*,
WORD *wGain7*,
WORD *wCardIndex*

)

Description	設定位置閉迴路控制使用的積分增益(I Gain)。	
Parameters	<i>wGain0~wGain7</i>	各軸使用的積分增益，設定範圍為 1 ~ 127
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_GetIGain(

WORD* *pwGain0*,
WORD* *pwGain1*,
WORD* *pwGain2*,
WORD* *pwGain3*,
WORD* *pwGain4*,
WORD* *pwGain5*,
WORD* *pwGain6*,
WORD* *pwGain7*,
WORD *wCardIndex*

)

Description	讀取位置閉迴路控制使用的積分增益(I Gain)。	
Parameters	<i>pwGain0~pwGain7</i>	指向一 WORD 值，用來存放各軸使用的積分增益
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功



非零

失敗，傳回值的意義請參考 IV.函式傳回值

7. int MCC_SetDGain(

WORD *wGain0*,
WORD *wGain1*,
WORD *wGain2*,
WORD *wGain3*,
WORD *wGain4*,
WORD *wGain5*,
WORD *wGain6*,
WORD *wGain7*,
WORD *wCardIndex*

)

Description 設定位置閉迴路控制使用的微分增益(D Gain)。

Parameters *wGain0~wGain7* 各軸使用的微分增益，設定範圍為 1 ~ 127

wCardIndex 運動控制卡編號(0 ~ 5)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

8. int MCC_GetDGain(

WORD* *pwGain0*,
WORD* *pwGain1*,
WORD* *pwGain2*,
WORD* *pwGain3*,
WORD* *pwGain4*,
WORD* *pwGain5*,
WORD* *pwGain6*,
WORD* *pwGain7*,
WORD *wCardIndex*

)

Description 讀取位置閉迴路控制使用的微分增益(D Gain)。

Parameters *pwGain0~pwGain7* 指向一 WORD 值，用來存放各軸使用的微分增益



	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

9. int MCC_SetFGain(

WORD *wGain0*,
WORD *wGain1*,
WORD *wGain2*,
WORD *wGain3*,
WORD *wGain4*,
WORD *wGain5*,
WORD *wGain6*,
WORD *wGain7*,
WORD *wCardIndex*

)

Description	設定位置閉迴路控制使用的前饋增益(Feedforward Gain)。	
Parameters	<i>wGain0~wGain7</i>	各軸使用的前饋增益，設定範圍為 1 ~ 127
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

10. int MCC_GetFGain(

WORD* *pwGain0*,
WORD* *pwGain1*,
WORD* *pwGain2*,
WORD* *pwGain3*,
WORD* *pwGain4*,
WORD* *pwGain5*,
WORD* *pwGain6*,
WORD* *pwGain7*,
WORD *wCardIndex*

)



Description	讀取位置閉迴路控制使用的前饋增益(Feedforward Gain)。	
Parameters	<i>pwGain0~pwGain7</i>	指向一 WORD 值，用來存放各軸使用的前饋增益
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

11. int MCC_SetMaxPulseSpeed(

```

int  nPulse0,
int  nPulse1,
int  nPulse2,
int  nPulse3,
int  nPulse4,
int  nPulse5,
int  nPulse6,
int  nPulse7,
WORD wCardIndex

```

)

Description	設定各軸最大 Pulse 速度的上限。最大 Pulse 速度的上限用來限制一插值單位時間內，各軸能送出的最大 Pulse 數，避免機台的速度超出工作範圍。更詳細的說明請參考”IMP Series 運動控制函式庫使用手冊 2.6.4 插值時間與加減速時間”。	
Parameters	<i>nPulse0~nPulse7</i>	各軸最大 Pulse 速度的上限 設定範圍為 1~1000000，適當值需視機台特性與插值時間而定
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

12. int MCC_GetMaxPulseSpeed(

```

int*  pnSpeed0,
int*  pnSpeed1,

```



```

int*   pnSpeed2,
int*   pnSpeed3,
int*   pnSpeed4,
int*   pnSpeed5,
int*   pnSpeed6,
int*   pnSpeed7,
WORD   wCardIndex
    )
    
```

Description	讀取各軸最大 Pulse 速度的上限。	
Parameters	<i>pnSpeed0~pnSpeed7</i> 指向一 int 值，各軸最大 Pulse 速度的上限	
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

13. int MCC_SetMaxPulseAcc(

```

int   nPulseAcc0,
int   nPulseAcc1,
int   nPulseAcc2,
int   nPulseAcc3,
int   nPulseAcc4,
int   nPulseAcc5,
int   nPulseAcc6,
int   nPulseAcc7,
WORD   wCardIndex
    )
    
```

Description 設定各軸最大 Pulse 加速度的上限。最大 Pulse 加速度的上限用來限制任連續兩插值時間中，各軸所送出 Pulse 數最大的差異量，可避免機台的加(減)速度超出工作範圍。更詳細的說明請參考”IMP Series 運動控制函式庫使用手冊 2.6.4 插值時間與加減速時間”。

Parameters *nPulseAcc0~nPulseAcc7* 各軸最大 Pulse 加速度的上限。



設定範圍為 1~1000000，適當值需視機台特性與插值時間而定

	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

14. int MCC_GetMaxPulseAcc(

```

int*  pnPulseAcc0,
int*  pnPulseAcc1,
int*  pnPulseAcc2,
int*  pnPulseAcc3,
int*  pnPulseAcc4,
int*  pnPulseAcc5,
int*  pnPulseAcc6,
int*  pnPulseAcc7,
WORD  wCardIndex

```

)

Description	讀取各軸最大 Pulse 加速度的上限。	
Parameters	<i>pnPulseAcc0~pnPulseAcc7</i>	指向一 int 值，存放各軸最大 Pulse 加速度的上限
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

15. int MCC_SetInPosMode(

```

WORD  wMode,
WORD  wGroupIndex

```

)

Description	設定定位確認使用模式。詳細說明請參考”IMP Series 運動控制函式庫使用手冊 2.7.2 定位確認”。	
-------------	--	--



Parameters	<i>wMode</i>	定位確認模式
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

16. int MCC_SetInPosMaxCheckTime(

WORD *wMaxCheckTime*,

WORD *wGroupIndex*

)

Description 設定定位確認最大檢查時間。詳細說明請參考”*IMP Series 運動控制函式庫使用手冊 2.7.2 定位確認*”。

Parameters *wMaxCheckTime* 定位確認最大檢查時間，單位 ms
wGroupIndex Group 編號

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

17. int MCC_SetInPosSettleTime(

WORD *wSettleTime*,

WORD *wGroupIndex*

)

Description 設定定位確認持續時間。詳細說明請參考”*IMP Series 運動控制函式庫使用手冊 2.7.2 定位確認*”。

Parameters *wSettleTime* 定位確認持續時間，單位 ms
wGroupIndex Group 編號

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

18. int MCC_EnableInPos(

WORD *wGroupIndex*,

DWORD *dwAxisMask*

)



Description	開啟定位確認功能。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>wGroupIndex</i>	Group 編號
	<i>dwAxisMask</i>	指定欲發生作用的軸，指定參數可為
	IMP_AXIS_X	X 軸
	IMP_AXIS_Y	Y 軸
	IMP_AXIS_Z	Z 軸
	IMP_AXIS_U	U 軸
	IMP_AXIS_V	V 軸
	IMP_AXIS_W	W 軸
	IMP_AXIS_A	A 軸
	IMP_AXIS_B	B 軸
	IMP_AXIS_ALL	全部運動軸
	以上參數可自由組合，以作用在 X、Z 與 V 軸上為例：	
	<code>(IMP_AXIS_X IMP_AXIS_Z IMP_AXIS_V)</code>	
Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

19. int MCC_DisableInPos(
 WORD *wGroupIndex*,
 DWORD *dwAxisMask*
)

Description	關閉定位確認功能。呼叫此函式成功將增加運動命令的庫存數目。	
Parameters	<i>wGroupIndex</i>	Group 編號
	<i>dwAxisMask</i>	指定欲發生作用的軸，指定參數可為
	IMP_AXIS_X	X 軸
	IMP_AXIS_Y	Y 軸
	IMP_AXIS_Z	Z 軸
	IMP_AXIS_U	U 軸
	IMP_AXIS_V	V 軸
	IMP_AXIS_W	W 軸



IMP_AXIS_A A 軸

IMP_AXIS_B B 軸

IMP_AXIS_ALL 全部運動軸

以上參數可自由組合，以作用在 X、Z 與 V 軸上為例：

(IMP_AXIS_X | IMP_AXIS_Z | IMP_AXIS_V)

Return Value	大於或等於零	MCCL 給予此運動命令的編碼
	小於零	失敗，傳回值的意義請參考 IV.函式傳回值

20. int MCC_SetInPosToleranceEx(

```

double dfTolerance0,
double dfTolerance1,
double dfTolerance2,
double dfTolerance3,
double dfTolerance4,
double dfTolerance5,
double dfTolerance6,
double dfTolerance7,
WORD wGroupIndex

```

)

Description	設定各軸定位誤差容許範圍。	
Parameters	<i>dfTolerance0</i> ~ <i>dfTolerance7</i>	各軸定位誤差容許範圍，單位為 UU
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

21. int MCC_GetInPosToleranceEx(

```

double* pdfTolerance0,
double* pdfTolerance1,
double* pdfTolerance2,
double* pdfTolerance3,
double* pdfTolerance4,

```



double* *pdfTolerance5*,
double* *pdfTolerance6*,
double* *pdfTolerance7*,
WORD *wGroupIndex*

)

Description	讀取各軸定位誤差容許範圍。	
Parameters	<i>pdfTolerance0~pdfTolerance7</i>	指向一 double 值，用來存放各軸定位誤差容許範圍，單位為 UU
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

22. int MCC_GetInPosStatus(

BYTE* *pbyInPos0*,
BYTE* *pbyInPos1*,
BYTE* *pbyInPos2*,
BYTE* *pbyInPos3*,
BYTE* *pbyInPos4*,
BYTE* *pbyInPos5*,
BYTE* *pbyInPos6*,
BYTE* *pbyInPos7*,
WORD *wGroupIndex*

)

Description	讀取各軸定位確認狀況。	
Parameters	<i>pbyInPos0~pbyInPos7</i>	指向一 BYTE 值，用來存放各軸的定位確認狀態，0xff(255)表示已滿足定位確認條件，0則表示尚未滿足定位確認條件
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

23. int MCC_EnableTrackError(**WORD** *wGroupIndex*,**DWORD** *dwAxisMask*

)

Description 開啟跟隨誤差偵測功能。詳細說明請參考”*IMP Series 運動控制函式庫使用手冊 2.7.3 跟隨錯誤偵測*”。

Parameters *wGroupIndex* Group 編號
dwAxisMask 指定欲發生作用的軸，指定參數可為

IMP_AXIS_X X 軸

IMP_AXIS_Y Y 軸

IMP_AXIS_Z Z 軸

IMP_AXIS_U U 軸

IMP_AXIS_V V 軸

IMP_AXIS_W W 軸

IMP_AXIS_A A 軸

IMP_AXIS_B B 軸

IMP_AXIS_ALL 全部運動軸

以上參數可自由組合，以作用在 X、Z 與 V 軸上為例：

(IMP_AXIS_X | IMP_AXIS_Z | IMP_AXIS_V)

Return Value 0 成功
非零 失敗，傳回值的意義請參考 **IV.函式傳回值**

24. int MCC_DisableTrackError(**WORD** *wGroupIndex*

)

Description 關閉檢查軌跡誤差功能。

Parameters *wGroupIndex* Group 編號

Return Value 0 成功
非零 失敗，傳回值的意義請參考 **IV.函式傳回值**

25. int MCC_SetTrackErrorLimit(

double *dfLimit*,
char *cAxis*,
WORD *wGroupIndex*

)

Description	設定跟隨誤差容許範圍。	
Parameters	<i>dfLimit</i>	跟隨誤差容許範圍，單位為 UU
	<i>cAxis</i>	運動軸編號(0 ~ 7 代表 X ~ B 軸)
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

26. int MCC_GetTrackErrorLimit(

double* *pdfLimit*,
char *cAxis*,
WORD *wGroupIndex*

)

Description	讀取跟隨誤差容許範圍。	
Parameters	<i>pdfLimit</i>	指向一 double 值，用來存放跟隨誤差容許範圍，單位為 UU
	<i>cAxis</i>	運動軸編號(0 ~ 7 代表 X ~ B 軸)
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

27. int MCC_SetPCLRoutine(

PCLISR *pfnPCLRoutine*,
WORD *wCardIndex*

)

Description	串接自訂的位置控制閉迴路(PCL)中斷服務函式(ISR)，當位置閉迴路控制失效時系統將自動呼叫此中斷服務函式。詳細說明請參	
-------------	---	--

考”IMP Series 運動控制函式庫使用手冊 2.7.4 位置閉迴路控制失效處理”。

Parameters	<i>pfnPCLRoutine</i>	自訂位置控制閉迴路中斷服務函式的函式指標
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

28. int MCC_SetErrorCountThreshold (

WORD *wChannel*,
WORD *wPlusThreshold*,
WORD *wMinusThreshold*,
WORD *wCardIndex*

)

Description	設定位置脈波誤差(Error Counter)容許範圍。	
Parameters	<i>wChannel</i>	運動軸編號(0 ~ 7 代表 X ~ B 軸)
	<i>wPlusThreshold</i>	設定正值位置脈波誤差容許範圍，單位為 pulse
	<i>wMinusThreshold</i>	設定負值位置脈波誤差容許範圍，單位為 pulse
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

29. int MCC_GetErrorCount (

int **pnErrCount0*,
int **pnErrCount1*,
int **pnErrCount2*,
int **pnErrCount3*,
int **pnErrCount4*,
int **pnErrCount5*,
int **pnErrCount6*,
int **pnErrCount7*,
WORD *wCardIndex*

)

Description	讀取各軸位置脈波誤差(Error Counter)大小。	
Parameters	<i>pnErrCount0~ pnErrCount7</i>	指向一 int 值，用來存放各軸的位置脈波誤差大小，單位為 pulse
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

30. int MCC_SetIGainClockDivider(

DWORD *dwClockDivider*,

WORD *wCardIndex*

)

Description	設定位置閉迴路控制使用的積分增益(I Gain)之積分取樣時間。設定範圍 $2^{16} \sim 2^{32}$ 個系統時脈時間。	
Parameters	<i>dwClockDivider</i>	PCL 閉迴路 I Gain 積分取樣時間($2^{16} \sim 2^{32}$)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

31. int MCC_SetDGainClockDivider(

DWORD *dwClockDivider*,

WORD *wCardIndex*

)

Description	設定位置閉迴路控制使用的微分增益(D Gain)之微分取樣時間。設定範圍 $0 \sim 2^{32}$ 個系統時脈時間。	
Parameters	<i>dwClockDivider</i>	PCL 閉迴路 D Gain 微分取樣時間 ($0 \sim 2^{32}$)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

K. 進階軌跡規劃

1. int MCC_HoldMotion(WORD *wGroupIndex*)

Description 暫時停止運動，但必須在運動進行中使用此函式才有意義。呼叫此函式後將減速至停止運動，在減速至停止前，若呼叫 MCC_GetMotionStatus() 所獲得的傳回值仍為 GMS_RUNNING，必須等到運動完全停止後，才會得到 GMS_HOLD 的傳回值。

Parameters *wGroupIndex* Group 編號

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

2. int MCC_ContiMotion(WORD *wGroupIndex*)

Description 繼續執行未完成的運動命令，但必須在運動暫停狀態使用此函式才有意義。

Parameters *wGroupIndex* Group 編號

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

3. int MCC_AbortMotion(WORD *wGroupIndex*, BOOL *bBlockEnd*)

Description 緊急停止，並放棄所有未執行之運動命令。注意，**在使用此函式之後，必須等到系統進入 GMS_STOP 狀態，始可下達後續之運動命令**，否則將得到 ABORT_NOT_FINISH_ERR(-15)的回傳值。

Parameters	<i>wGroupIndex</i>	Group 編號
	<i>bBlockEnd</i>	true，會執行完目前的運動命令，在減速停止並放棄所有未執行之運動命令 false，立即停止目前的運動命令，放棄所有未執行之運動命令
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_AbortMotionEx(double *dfDecTime*, WORD *wGroupIndex*)

Description 以設定的減速時間，減速至停止並拋棄後續全部運動命令。呼叫此函式後將減速至停止運動，在減速至停止前，若呼叫 MCC_GetMotionStatus()所獲得的傳回值仍為 GMS_RUNNING，必須等到運動完全停止後，才會得到 GMS_STOP 的傳回值。注意，**在使用此函式之後，必須等到系統進入 GMS_STOP 狀態，始可下達後續之運動命令**，否則將得到 ABORT_NOT_FINISH_ERR(-15)的回傳值。

Parameters	<i>dfDecTime</i>	要求減速的時間
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_EnableBlend(WORD *wGroupIndex*)

Description 開啟平滑運動功能。呼叫此函式後，以連續路徑方式進行軌跡規劃。成功呼叫此函式將增加運動命令的庫存數目。

Parameters	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於零	MCCL 給予此運動命令的編碼

小於零

失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_DisableBlend(**WORD *wGroupIndex***

)

Description 關閉平滑運動功能。成功呼叫此函式將增加運動命令的庫存數目。

Parameters *wGroupIndex* Group 編號

Return Value 大於或等於零 MCCL 給予此運動命令的編碼

小於零 失敗，傳回值的意義請參考 IV.函式傳回值

7. int MCC_CheckBlend(**WORD *wGroupIndex***

)

Description 檢查是否開啟平滑運動功能。

Parameters *wGroupIndex* Group 編號

Return Value 0 已開啟平滑運動功能

1 未開啟平滑運動功能

其他 失敗，傳回值的意義請參考 IV.函式傳回值

8. int MCC_DelayMotion(**DWORD *dwTime*,****WORD *wGroupIndex***

)

Description 設定運動延遲時間，強迫延遲執行下一個運動命令。成功呼叫此函式將增加運動命令的庫存數目。

Parameters *dwTime* 延遲時間，單位為 ms*wGroupIndex* Group 編號

Return Value 大於或等於零 MCCL 給予此運動命令的編碼

小於零 失敗，傳回值的意義請參考 IV.函式傳回值

9. int MCC_CheckDelay(**WORD** *wGroupIndex*

)

Description 檢查目前是否進入運動延遲狀態（此時若呼叫 MCC_GetMotionStatus() 將得到 GMS_DELAYING 的回傳值）。

Parameters *wGroupIndex* Group 編號

Return Value 0 未處於運動延遲狀態

1 處於運動延遲狀態

其他 失敗，傳回值的意義請參考 IV.函式傳回值

10. double MCC_OverrideSpeed(**double** *dfRate*,**WORD** *wGroupIndex*

)

Description 設定一般運動與點對點運動的速度強制比例，使用此項函式將即時變動一般運動與點對點運動的進給速度。

Parameters *dfRate* 速度強制比例，為原來進給速度的多少百分比再乘以 100，也就是說一般運動新的進給速度將等於 $(dfFeedSpeed \times dfRate / 100)$ ，*dfFeedSpeed* 為原來使用 MCC_SetFeedSpeed() 所設定的進給速度，而點對點運動速度等於 $(速度比例 \times dfRate / 100)$ 。

設定值必須大於 0；若更新後的進給速度超過 MCC_SetSysMaxSpeed() 的設定值，則新的進給速度將等於 MCC_SetSysMaxSpeed() 的設定值。

wGroupIndex Group 編號

Return Value 大於零 實際設定的速度強制比例

其他 失敗，傳回值的意義請參考 IV.函式傳回值

11. double MCC_OverrideSpeedEx(

double *dfRate*,
BOOL *bInstant*,
WORD *wGroupIndex*

)

Description	設定一般運動與點對點運動的速度強制比例，使用此項函式將即時變動一般運動的進給速度。	
Parameters	<i>dfRate</i>	速度強制比例，為原來進給速度的多少百分比再乘以 100，也就是說一般運動新的進給速度將等於 $(dfFeedSpeed \times dfRate / 100)$ ， <i>dfFeedSpeed</i> 為原來使用 <code>MCC_SetFeedSpeed()</code> 所設定的進給速度。 設定值必須大於 0；若更新後的進給速度超過 <code>MCC_SetSysMaxSpeed()</code> 的設定值，則新的進給速度將等於 <code>MCC_SetSysMaxSpeed()</code> 的設定值。
	<i>bInstant</i>	更新速度時機，true 代表立即更新速度，false 代表下一筆命令才更新速度。
	<i>wGroupIndex</i>	Group 編號
Return Value	大於零	實際設定的速度強制比例
	其他	失敗，傳回值的意義請參考 IV.函式傳回值

12. double MCC_GetOverrideRate(

WORD *wGroupIndex*

)

Description	讀取一般運動目前使用的速度強制比例。	
Parameters	<i>wGroupIndex</i>	Group 編號
Return Value	大於零	一般運動目前使用的速度強制比例
	其他	失敗，傳回值的意義請參考 IV.函式傳回值



**13. int MCC_TimeDelay(
 DWORD *dwTime*
)**

Description	設定運動延遲時間，強迫延遲執行下一個運動命令。	
Parameters	<i>dwTime</i>	延遲時間，單位為 ms
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

L. 編碼器控制

此節主要講述內容為 IMP Series 控制卡中編碼器模組所提供之功能及其使用方式，使用者應搭配”IMP Series 運動控制函式庫使用手冊 2.10 編碼器控制”一節共同閱讀。

1. int MCC_SetENCRoutine(

ENCISR *pfnEncoderRoutine*,

WORD *wCardIndex*

)

Description	串接自訂的編碼器中斷服務函式。	
Parameters	<i>pfnENCRoutine</i>	自訂編碼器中斷服務函式的函式指標
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

2. int MCC_SetENCInputRate(

WORD *wInputRate*,

WORD *wChannel*,

WORD *wCardIndex*

)

Description	設定編碼器的回授倍率，使用此函數與機構參數中 <i>wInputRate</i> 對回授倍率的影響相同。	
Parameters	<i>wInputRate</i>	編碼器的回授倍率，可設定值為 1、2、4
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

3. int MCC_ClearENCCounter(

WORD *wChannel*,

WORD *wCardIndex*

)

Description	使編碼器的計數值歸零。	
Parameters	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_GetENCValue(

long* *plValue*,

WORD *wChannel*,

WORD *wCardIndex*

)

Description	讀取編碼器的計數值。	
Parameters	<i>plValue</i>	指向一 long 值，用來存放編碼器的計數值
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_SetENCLatchType(

WORD *wType*,

WORD *wChannel*,

WORD *wCardIndex*

)

Description	設定觸發閘鎖(Latch)編碼器計數值動作的方式。	
Parameters	<i>wType</i>	觸發閘鎖編碼器計數值動作的方式，可設定為： ENC_TRIG_FIRST 第一次滿足觸發條件時，計數 值即被閘鎖並不再變動

	ENC_TRIG_LAST	當觸發條件滿足時即閃鎖新的計數值，次數不限
	<i>wChannel</i>	運動軸之編號(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_SetENCLatchSource(

DWORD *wSource*,
WORD *wChannel*,
WORD *wCardIndex*

)

Description 設定觸發閃鎖編碼器計數值動作的訊號來源，可同時設定多個訊號來源的聯集，例如可以使用
MCC_SetENCLatchSource(ENC_TRIG_INDEX0 | ENC_TRIG_LIO0, 0, 0)
表示在輸入編碼器 Channel 0 的 INDEX 訊號與碰觸到 Channel 0 的正方向 Limit 時，皆會將編碼器計數值紀錄在第 0 張卡的第 0 個 Channel 的閃鎖暫存器內。

Parameters *wSource* 訊號來源，可設定為：

ENC_TRIG_NO	沒有選擇任何觸發訊號源
ENC_TRIG_INDEX0	編碼器 Channel 0 的 INDEX 訊號
ENC_TRIG_INDEX1	編碼器 Channel 1 的 INDEX 訊號
ENC_TRIG_INDEX2	編碼器 Channel 2 的 INDEX 訊號
ENC_TRIG_INDEX3	編碼器 Channel 3 的 INDEX 訊號
ENC_TRIG_INDEX4	編碼器 Channel 4 的 INDEX 訊號
ENC_TRIG_INDEX5	編碼器 Channel 5 的 INDEX 訊號
ENC_TRIG_INDEX6	編碼器 Channel 6 的 INDEX 訊號
ENC_TRIG_INDEX7	編碼器 Channel 7 的 INDEX 訊號
ENC_TRIG_OTP0	Channel 0 的正極限訊號
ENC_TRIG_OTP1	Channel 1 的正極限訊號



	ENC_TRIG_OTP2	Channel 2 的正極限訊號
	ENC_TRIG_OTP3	Channel 3 的正極限訊號
	ENC_TRIG_OTP4	Channel 4 的正極限訊號
	ENC_TRIG_OTP5	Channel 5 的正極限訊號
	ENC_TRIG_OTP6	Channel 6 的正極限訊號
	ENC_TRIG_OTP7	Channel 7 的正極限訊號
	ENC_TRIG_OTN0	Channel 0 的負極限訊號
	ENC_TRIG_OTN1	Channel 1 的負極限訊號
	ENC_TRIG_OTN2	Channel 2 的負極限訊號
	ENC_TRIG_OTN3	Channel 3 的負極限訊號
	ENC_TRIG_OTN4	Channel 4 的負極限訊號
	ENC_TRIG_OTN5	Channel 5 的負極限訊號
	ENC_TRIG_OTN6	Channel 6 的負極限訊號
	ENC_TRIG_OTN7	Channel 7 的負極限訊號
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

7. int MCC_GetENCLatchValue(

long* *plLatchValue*,
WORD *wChannel*,
WORD *wCardIndex*

)

Description	讀取門鎖暫存器內的紀錄值。	
Parameters	<i>plLatchValue</i>	指向一 long 值，用來存放門鎖暫存器的紀錄值
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

8. int MCC_EnableENCIndexTrigger(

WORD *wChannel*,

WORD *wCardIndex*

)

Description 開啟編碼器 INDEX 訊號觸發編碼器中斷服務函式的功能。

Parameters *wChannel* 運動控制卡的輸出 Channel(0 ~ 7)

wCardIndex 運動控制卡的編號(0 ~ 5)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

9. int MCC_DisableENCIndexTrigger(

WORD *wChannel*,

WORD *wCardIndex*

)

Description 關閉編碼器 INDEX 訊號觸發編碼器中斷服務函式的功能。

Parameters *wChannel* 運動控制卡的輸出 Channel(0 ~ 7)

wCardIndex 運動控制卡的編號(0 ~ 5)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

10. int MCC_GetENCIndexStatus(

WORD* *pwStatus*,

WORD *wChannel*,

WORD *wCardIndex*

)

Description 確認目前位置是否位於 INDEX 訊號的輸入點上。

Parameters *pwStatus* 指向一 WORD 值，用來存放 INDEX 訊號的輸入狀態；1 代表目前位置在 INDEX 上，0 則否

wChannel 運動控制卡的輸出 Channel(0 ~ 7)

wCardIndex 運動控制卡的編號(0 ~ 5)

Return Value 0 成功

非零

失敗，傳回值的意義請參考 IV.函式傳回值

11. int MCC_SetENCCompValue(

long *IValue*,

WORD *wChannel*,

WORD *wCardIndex*

)

Description 設定編碼器的比較值。

Parameters *IValue* 編碼器的比較值
wChannel 運動控制卡的輸出 Channel(0 ~ 7)
wCardIndex 運動控制卡的編號(0 ~ 5)

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

12. int MCC_EnableENCCompTrigger(

WORD *wChannel*,

WORD *wCardIndex*

)

Description 開啟編碼器的計數值等於比較值時觸發編碼器中斷服務函式的功能。

Parameters *wChannel* 運動控制卡的輸出 Channel(0 ~ 7)
wCardIndex 運動控制卡的編號(0 ~ 5)

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

13. int MCC_DisableENCCompTrigger(

WORD *wChannel*,

WORD *wCardIndex*

)

Description 關閉編碼器的計數值等於比較值時觸發編碼器中斷服務函式的功能。



Parameters	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

14. int MCC_Enable3DCompareChannel(

WORD *wCompareAxisMask*,

WORD *wCardIndex*

)

Description	啟用 3D 編碼器計數值比較功能。	
Parameters	<i>wCompareAxisMask</i>	欲啟用之實體軸
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

15. int MCC_Set3DCompareTolerance(

WORD *wTolerance*,

WORD *wCardIndex*

)

Description	設定 3D 編碼器的比較值的誤差容許範圍。	
Parameters	<i>wTolerance</i>	比較值的誤差容許範圍
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

16. int MCC_SetENCCompValueTab(

long *lStartValue*,

long *lEndValue*,

long *lDeltaValue*,

WORD *wChannel*,

WORD *wCardIndex*



)

Description	設定相同間隔編碼器計數值比較值。	
Parameters	<i>lStartValue</i>	起始計數值
	<i>lEndValue</i>	結束計數值
	<i>lDeltaValue</i>	間距值
	<i>wChannel</i>	運動控制卡的輸出 Channel (0~7)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

17. int MCC_SetLIOTrigOutEnable(

WORD *wEnable*,
WORD *wPoint*,
WORD *wCardIndex*

)

Description	設定編碼器計數值比較成功時，是否要啟用近端輸出接點，及欲輸出的接點編號。	
Parameters	<i>wEnable</i>	是否啟用近端輸出接點功能，1表示啟用，0表示關閉
	<i>wPoint</i>	近端輸出點之接點編號
	<i>wCardIndex</i>	運動控制卡編號(0~5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考IV.函式傳回值

18. int MCC_SetLIOTrigOutValue(

WORD *wValue*,
WORD *wPoint*,
WORD *wCardIndex*

)

Description	設定編碼器計數值比較成功時，設定近端輸出接點的狀態及欲輸出的接點編號。	
-------------	-------------------------------------	--



Parameters	<i>wValue</i>	輸出接點的狀態
	<i>wPoint</i>	欲輸出的接點編號近端輸出點之接點編號
	<i>wCardIndex</i>	運動控制卡編號(0~5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

19. int MCC_SetLIOTrigOutValueEx(

DWORD *dwBitMask*,
DWORD *dwBitValue*,
WORD *wCardIndex*

)

Description	設定編碼器計數值比較成功時，多組近端輸出接點是否輸出及欲輸出的接點編號。	
Parameters	<i>dwBitMask</i>	欲輸出近端輸出點之接點編號，可設定多組接點同時輸出
	<i>dwBitValue</i>	設定每個接點的輸出狀態
	<i>wCardIndex</i>	運動控制卡編號(0~5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

20. int MCC_SetLIOTrigOutSource(

WORD *wSource*,
WORD *wPoint*,
WORD *wCardIndex*

)

Description	設定欲比較編碼器計數值之實體軸與對應的近端輸出接點編號。	
Parameters	<i>wSource</i>	欲比較編碼計數值之實體軸編號
	<i>wPoint</i>	對應之近端輸出接點編號
	<i>wCardIndex</i>	運動控制卡編號(0~5)
Return Value	0	成功



非零 失敗，傳回值的意義請參考 IV.函式傳回值

21. int MCC_SetLIOTrigOutPeriod(

DWORD *dwPeriod*,
WORD *wCardIndex*

)

Description 設定編碼器計數值比較成功時，近端輸出點輸出時間。

Parameters *dwPeriod* 輸出時間，單位 ns
wCardIndex 運動控制卡編號(0~5)

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

22. int MCC_GetLIOCompStockCount(

WORD* *wCount*,
WORD *wCardIndex*

)

Description 取得目前所有編碼器計數比較值觸發 LIO 之庫存筆數。

Parameters *wCount* 指向一 WORD 值，用來存放編碼器比較數值觸發 LIO
之庫存筆數
wCardIndex 運動控制卡編號(0~5)

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

23. int MCC_EraseLIOCompValue(

WORD *wCardIndex*

)

Description 清除所有編碼器計數比較值觸發 LIO 之庫存筆數。

Parameters *wCardIndex* 運動控制卡編號(0~5)

Return Value 0 成功
非零 失敗，傳回值的意義請參考 IV.函式傳回值

24. int MCC_SetENCGearRate(

double *dfGearRate*,
WORD *wChannel*,
WORD *wCardIndex*

)

Description	設定軟體的 Gear Rate 值。	
Parameters	<i>dfGearRate</i>	Gear Rate 數值
	<i>wChannel</i>	運動控制卡的輸出 Channel (0~7)
	<i>wCardIndex</i>	運動控制卡編號(0~5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

25. int MCC_GetENCCompValue(

long* *lValue*,
WORD *wChannel*,
WORD *wCardIndex*

)

Description	讀取正要被編碼器計數值比較的數值。	
Parameters	<i>lValue</i>	指向一 long 值，用來存放編碼器計數值比較的數值
	<i>wChannel</i>	運動控制卡的輸出 Channel (0~7)
	<i>wCardIndex</i>	運動控制卡編號(0~5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

26. int MCC_GetENCCompStockCount(

WORD* *wCount*,
WORD *wChannel*,
WORD *wCardIndex*

)

Description	讀取目前所有編碼器計數比較值之筆數。	
Parameters	<i>wCount</i>	指向一 WORD 值，用來存放編碼器比較數值之筆數



	<i>wChannel</i>	運動控制卡的輸出 Channel (0~7)
	<i>wCardIndex</i>	運動控制卡編號(0~5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

27. MCC_EraseENCCompValue(

WORD *wChannel*,

WORD *wCardIndex*

)

Description	清除所有編碼器計數比較值之筆數。	
Parameters	<i>wChannel</i>	運動控制卡的輸出 Channel (0~7)
	<i>wCardIndex</i>	運動控制卡編號(0~5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

M. 計時器與 Watch Dog 控制

此節主要講述內容為 IMP Series 控制卡中計時器與 Watch Dog 所提供之功能及其使用方式，使用者應搭配”*IMP Series 運動控制函式庫使用手冊 2.13.2 計時器與 Watch Dog 控制*”一節共同閱讀。

1. **int MCC_SetTMRRoutine(**
 TMRISR *pfnLIORoutine,*
 WORD *wCardIndex*
)

Description 串接自訂的計時觸發中斷服務函式 (ISR)，詳細的說明請參考”*IMP Series 運動控制函式庫使用手冊 2.13.1 計時器(Timer)與 Watch Dog 控制*”。

Parameters *pfnLIORoutine* 自訂計時觸發中斷服務函式的函式指標
wCardIndex 運動控制卡的編號(0 ~ 5)

Return Value 0 成功
非零 失敗，傳回值的意義請參考 **IV.函式傳回值**

2. **int MCC_SetTimer(**
 DWORD *dwValue,*
 WORD *wCardIndex*
)

Description 設定計時器的計時週期，在每個計時週期可觸發自訂的近端輸入接點中斷服務函式。

Parameters *dwValue* 計時週期，單位為 1us，設定範圍為 $1 \sim 2^{32}-1$
wCardIndex 運動控制卡的編號(0 ~ 5)

Return Value 0 成功
非零 失敗，傳回值的意義請參考 **IV.函式傳回值**

3. int MCC_EnableTimer(WORD *wCardIndex*)

Description	開啟計時器計時功能。	
Parameters	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_DisableTimer(WORD *wCardIndex*)

Description	關閉計時器計時功能。	
Parameters	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_EnableTimerTrigger(WORD *wCardIndex*)

Description	開啟計時器在每個計時週期時觸發近端輸入接點中斷服務函式的功能。	
Parameters	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_DisableTimerTrigger(WORD *wCardIndex*)

Description	關閉計時器在每個計時週期時觸發近端輸入接點中斷服務函式的功能。	
Parameters	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

7. int MCC_SetWatchDogTimer(DWORD *wValue*, WORD *wCardIndex*)

Description 設定 Watch Dog 的倒數計時時間。一旦 Watch Dog 計時終了時會產生硬體 Reset 的訊號，如不想產生 Reset 訊號，則在計時終了前可利用 MCC_RefreshWatchDogTimer()使 Watch Dog 計時器重新計數。

Parameters

<i>wValue</i>	Watch Dog 的倒數計時時間，單位為 1us，設定範圍為 $1 \sim 2^{32}-1$
<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

8. int MCC_SetWatchDogResetPeriod(DWORD *dwValue*, WORD *wCardIndex*)

Description 設定在 Watch Dog 計時終了時所產生硬體 Reset 訊號的持續時間。

Parameters

<i>dwValue</i>	硬體 Reset 訊號持續時間，單位為 10ns
<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值



9. int MCC_EnableWatchDogTimer(

WORD *wCardIndex*

)

Description	開啟 Watch Dog 功能。	
Parameters	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

10. int MCC_DisableWatchDogTimer(

WORD *wCardIndex*

)

Description	關閉 Watch Dog 功能。	
Parameters	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

11. int MCC_RefreshWatchDogTimer(

WORD *wCardIndex*

)

Description	重置 Watch Dog 的計時時間，避免 Watch Dog 計時終了產生硬體 Reset 訊號。	
Parameters	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

N. Remote I/O 控制

此節主要講述內容為 IMP Series 控制卡中 Remote I/O 模組所提供之功能及其使用方式，使用者應搭配”IMP Series 運動控制函式庫使用手冊 2.14 Remote I/O 控制”一節共同閱讀。

1. int MCC_EnableARIOSlaveControl(

WORD *wSlave*,
WORD *wCardIndex*
)

Description	開啟指定的 Remote I/O 的 Slave 資料傳輸功能。	
Parameters	<i>wSlave</i>	Slave 的編號 (Slave 0 ~ 31)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

2. int MCC_DisableARIOSlaveControl(

WORD *wSlave*,
WORD *wCardIndex*
)

Description	關閉指定 Remote I/O 的 Slave 之資料傳輸功能。	
Parameters	<i>wSlave</i>	Slave 的編號 (Slave 0 ~ 31)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

3. int MCC_GetARIOTransStatus(

WORD* *pwStatus*,
WORD *wCardIndex*



)

Description	讀取目前 Remote I/O 資料傳輸狀態。若傳輸已停止，則可再呼叫 MCC_GetARIOMasterStatus()及 MCC_GetARIOSlaveStatus()分辨產生錯誤的為 Master 端或 Slave 端。	
Parameters	<i>pwStatus</i>	指向一 WORD 值，用來存放 Remote I/O 資料傳輸狀態；1 代表 Remote I/O Set 正常工作中，0 則否
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_GetARIOMasterStatus(

WORD* *pwStatus*,

WORD *wCardIndex*

)

Description	讀取目前 Remote I/O Master 端傳送資料到 Slave 的狀態。	
Parameters	<i>pwStatus</i>	指向一 WORD 值，用來存放 Remote I/O 資料傳輸狀態；1 代表 Remote I/O Master 端接收訊號正常，0 則否
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_GetARIOSlaveStatus(

DWORD* *pdwStatus*,

WORD *wCardIndex*

)

Description	讀取目前 Remote I/O Master 端是否收到來自 Slave 資料傳輸的狀態。	
Parameters	<i>pdwStatus</i>	指向一 DWORD 值，用來存放 Remote I/O 資料傳輸狀態；0 代表 Remote I/O Master 端有接收到來自 Slave 端的訊號正常，1 則否(bit 0 ~ bit 31 分別代表 Slave 0

		到 Slave 31 的狀態)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_GetARIOInputValue(

WORD* *pwValue*,
WORD *wSlave*,
WORD *wCardIndex*
)

Description	讀取指定 Slave 的 16 點輸入訊號狀態值。	
Parameters	<i>pwValue</i>	指向一 WORD 值，用來存放指定 Slave 中 16 點的輸入訊號狀態(bit 0 ~ bit 15 分別代表第 0 點到第 15 點的狀態)
	<i>wSlave</i>	Slave 編號 (Slave 0 ~ 31)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

7. int MCC_SetARIOOutputValue(

WORD *wValue*,
WORD *wSlave*,
WORD *wCardIndex*
)

Description	設定指定 Slave 的 16 點輸出訊號狀態值。	
Parameters	<i>wValue</i>	指定 Slave 中 16 點的輸出訊號狀態(bit 0 ~ bit 15 分別代表第 0 點到第 15 點的狀態)
	<i>wSlave</i>	Slave 編號 (Slave 0 ~ 31)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功

非零

失敗，傳回值的意義請參考 IV.函式傳回值

8. int MCC_SetARIOClockDivider(

WORD *wDivider*,

WORD *wCardIndex*

)

Description 設定 Remote I/O 傳輸資料的時脈頻率。換算後之傳輸頻率為 System Clock 除以(2*divider +1)，內定設定值為 255。

Parameters *wDivider* Remote I/O Clock Divider(0 ~ 255)

wCardIndex 運動控制卡編號(0 ~ 5)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

9. int MCC_GetARIOSlaveFail (

DWORD* *pdwStatus*,

WORD *wCardIndex*

)

Description 讀取 Remote I/O Slave 端狀態是否正常。

Parameters *pdwStatus* 指向一 DWORD 值，用來存放 Remote I/O 資料傳輸狀態；0 代表 Remote I/O Master 端有接收到來自 Slave 端的訊號正常，1 則否(bit 0 ~ bit 31 分別代表 Slave 0 到 Slave 31 的狀態)

wCardIndex 運動控制卡編號(0 ~ 5)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

10. int MCC_GetARIOOutputValue(

WORD* *pwValue*,

WORD *wSlave*,

WORD *wCardIndex*

)

Description	讀取指定 Slave 的 16 點輸出訊號狀態值。	
Parameters	<i>pwValue</i>	指向一 WORD 值，用來存放指定 Slave 中 16 點的輸出訊號狀態(bit 0 ~ bit 15 分別代表第 0 點到第 15 點的狀態)
	<i>wSlave</i>	Slave 的編號 (Slave 0 ~ 31)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

11. int MCC_QueueARIOOutputValue(

WORD *wValue*,
WORD *wSlave*,
WORD *wGroupIndex*

)

Description	將指定 Slave 的 16 點輸出訊號狀態值存放至運動命令佇列(Queue)中。成功呼叫此函式將增加運動命令的庫存數目。	
Parameters	<i>wValue</i>	指定 Slave 的 16 點輸出訊號狀態(bit 0 ~ bit 15 分別代表此 Slave 中第 0 點到第 15 點的狀態)
	<i>wSlave</i>	Slave 的編號 (Slave 0 ~ 31)
	<i>wGroupIndex</i>	Group 編號
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

O.D/A Converter 控制

此節主要講述內容為 IMP Series 控制卡中 D/A Converter(DAC)模組所提供之功能及其使用方式，使用者應搭配”IMP Series 運動控制函式庫使用手冊 2.11 類比電壓輸出控制”一節共同閱讀。

```
1. int MCC_SetDACOutput(  
    float fVoltage,  
    WORD wChannel,  
    WORD wCardIndex  
)
```

Description	輸出指定的電壓值。	
Parameters	<i>fVoltage</i>	輸出電壓值(-10V ~ 10 V)
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

```
2. int MCC_SetDACTriggerOutput(  
    float fVoltage,  
    WORD wChannel,  
    WORD wCardIndex  
)
```

Description	當運動卡的輸出軸(0 ~ 7)不使用 V Command 操作模式時(也就是機構參數中的 <i>wCommandMode</i> 設定為 OCM_PULSE),可預先規劃一個電壓值在 DAC 模組內，當滿足觸發條件時硬體可立即輸出預先規劃的電壓。	
Parameters	<i>fVoltage</i>	預先規劃的輸出電壓值(-10V ~ 10 V)
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)



Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

3. int MCC_SetDACTriggerSource(

DWORD *dwSource*,

WORD *wChannel*,

WORD *wCardIndex*

)

Description 當運動軸不使用 V Command 操作模式時，可使用此函式設定觸發輸出預先規劃電壓的來源。每一 DAC Channel 可設定多種觸發來源。設定完本函式後必須再使用 MCC_EnableDACTriggerMode() 開啟觸發模式。

Parameters	<i>dwSource</i>	DAC 觸發來源，可為：
		DAC_TRIG_ENC0 編碼器 Channel 0 特定計數值
		DAC_TRIG_ENC1 編碼器 Channel 1 特定計數值
		DAC_TRIG_ENC2 編碼器 Channel 2 特定計數值
		DAC_TRIG_ENC3 編碼器 Channel 3 特定計數值
		DAC_TRIG_ENC4 編碼器 Channel 4 特定計數值
		DAC_TRIG_ENC5 編碼器 Channel 5 特定計數值
		DAC_TRIG_ENC6 編碼器 Channel 6 特定計數值
		DAC_TRIG_ENC7 編碼器 Channel 7 特定計數值
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

4. int MCC_EnableDACTriggerMode(

WORD *wChannel*,

WORD *wCardIndex*

)

Description 當運動軸不使用 V Command 操作模式時，可使用此函式開啟觸發輸出

預先規劃電壓的功能，開啟觸發模式前請先設定觸發來源。

Parameters	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_DisableDACTriggerMode(

WORD *wChannel*,

WORD *wCardIndex*

)

Description 當運動軸不使用 V Command 操作模式時，可使用此函式關閉觸發輸出預先規劃電壓的功能。

Parameters	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_StartDACConv(

WORD *wCardIndex*

)

Description 當 *wCardIndex* 所指定之控制卡中沒有任何輸出 Channel 使用 V Command 操作模式時，可使用此函式開啟電壓輸出功能。

Parameters	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值



7. **int MCC_StopDACConv(**
 WORD *wCardIndex*
)

Description 當 *wCardIndex* 所指定之控制卡中沒有任何輸出 Channel 使用 V Command 操作模式時，可使用此函式關閉電壓輸出功能。

Parameters *wCardIndex* 運動控制卡編號(0 ~ 5)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 **IV.函式傳回值**

P. A/D Converter 控制

此節主要講述內容為 IMP Series 控制卡中 A/D Converter(ADC)模組所提供之功能及其使用方式，使用者應搭配”IMP Series 運動控制函式庫使用手冊 2.12 類比電壓輸入控制”一節共同閱讀。

1. int MCC_SetADCRoutine(

ADCISR *pfnADCRoutine*,

WORD *wCardIndex*

)

Description 串接自訂的 ADC 中斷服務函式。

Parameters *pfnADCRoutine* 自訂 ADC 中斷服務函式的函式指標

wCardIndex 運動控制卡編號(0 ~ 5)

Return Value 0 成功

非零 失敗，傳回值的意義請參考 IV.函式傳回值

2. int MCC_SetADCConvType(

WORD *wConvType*,

WORD *wChannel*,

WORD *wCardIndex*

)

Description 設定 ADC 電壓轉換模式為雙極性或單極性。

Parameters *wConvType* 電壓轉換模式設定

ADC_TYPE_BIP_DIFF 輸入源為差動訊號
(Differential)使用雙極性轉換型式(Bipolar Converter Type)，可讀取的電壓範圍為-5V ~ 5V

ADC_TYPE_BIP_SING 輸入源為單點輸入(Single Ended)使用雙極性轉換型式(Bipolar Converter Type)，可讀取的電壓範圍為-5V ~ 5V

	ADC_TYPE_UNI_DIFF	輸入源為差動訊號 (Differential)使用單極性轉換型式(Unipolar Converter Type)，可讀取的電壓範圍為 0V ~ 10V
	ADC_TYPE_UNI_SING	輸入源為單點輸入(Single Ended)使用單極性轉換型式 (Unipolar Converter Type)，可讀取的電壓範圍為 0V ~ 10V
	<i>wChannel</i>	ADC Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

3. **int** MCC_GetADCInput(
 float* *pfInput*,
 WORD *wChannel*,
 WORD *wCardIndex*
)

Description	讀取指定的 ADC Channel 輸入之直流電壓值，若 ADC 指定為單極性轉換型式(Unipolar)則 IMP Series 輸入電壓的有效值為 0V ~ 10V。若 ADC 設定為雙極性轉換型式(Bipolar)則 IMP Series 輸入電壓的有效值為-5V ~ +5V。	
Parameters	<i>pfInput</i>	指向一 float 值，用來存放 ADC Channel 直流電壓輸入值
	<i>wChannel</i>	ADC Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

4. **int** MCC_SetADCCompType(
 WORD *wCompType*,
 WORD *wChannel*,

WORD *wCardIndex*

)

Description	設定 ADC 電壓比較模式，呼叫本函式再呼叫 MCC_EnableADCCompTrigger()，則當比較條件成立便會產生 ADC 硬體中斷訊號。此觸發訊號除了可觸發使用者自訂的中斷服務函式外，亦可用來觸發 DAC 模組輸出預先規劃的電壓值，同時前兩組 ADC Channel 觸發訊號亦可用來門鎖編碼器的計數值。	
Parameters	<i>wCompType</i>	電壓比較模式，可設定值有： ADC_COMP_RISE 輸入電壓由小到大通過比較值 ADC_COMP_FALL 輸入電壓由大到小通過比較值 ADC_COMP_LEVEL 輸入電壓值改變並通過比較值
	<i>wChannel</i>	ADC Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

5. int MCC_SetADCCompValue(

float *fValue*,

WORD *wChannel*,

WORD *wCardIndex*

)

Description	設定 ADC Channel 在雙極性(Bipolar)型式下的輸入電壓比較值，本函式不提供單極性(Unipolar)型式下電壓比較功能。設定本函式後必須再使用 MCC_SetADCCompType()與 MCC_EnableADCCompTrigger()，此時在該 ADC Channel 輸入電壓滿足比較條件後，可產生 ADC 硬體中斷訊號。	
Parameters	<i>fValue</i>	輸入電壓比較值(-5V ~ 5V)
	<i>wChannel</i>	ADC Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

6. int MCC_GetADCCompValue(

float* *pfValue*,
WORD *wChannel*,
WORD *wCardIndex*
)

Description	讀取使用的電壓比較值。	
Parameters	<i>pfValue</i>	指向一 float 值，用來存放輸入電壓比較值
	<i>wChannel</i>	ADC Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

7. int MCC_EnableADCCConvChannel(

WORD *wChannel*,
WORD *wCardIndex*
)

Description	開啟 ADC Channel 輸入電壓類比轉數位功能。設定完成後必須再呼叫 MCC_StartADCCConv()，啟動 ADC 轉換功能。	
Parameters	<i>wChannel</i>	ADC Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)
Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

8. int MCC_DisableADCCConvChannel(

WORD *wChannel*,
WORD *wCardIndex*
)

Description	關閉 ADC Channel 輸入電壓類比轉數位功能。	
Parameters	<i>wChannel</i>	ADC Channel(0 ~ 7)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 5)

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

9. int MCC_StartADCCConv(
 WORD *wCardIndex*
)

Description 開始進行 ADC Channel 類比電壓值轉換，本函式必須配合 MCC_EnableADCCConvChannel()使用。

Parameters *wCardIndex* 運動控制卡編號(0 ~ 5)

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

10. int MCC_StopADCCConv(
 WORD *wCardIndex*
)

Description 停止進行所有 ADC Channel 的類比電壓值轉換功能。

Parameters *wCardIndex* 運動控制卡編號(0 ~ 5)

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

11. int MCC_GetADCCConvType(
 WORD* *pwConvType*,
 WORD *wChannel*,
 WORD *wCardIndex*
)

Description 取得 ADC 電壓轉換模式。

Parameters *pwConvType* 指向一 WORD 值，用來存放電壓轉換型式
wChannel ADC Channel(0 ~ 7)
wCardIndex 運動控制卡編號(0 ~ 5)

Return Value	0	成功
	非零	失敗，傳回值的意義請參考 IV.函式傳回值

III. 錯誤訊息代碼

錯誤代碼	說明
0xF101	尚未初使化運動控制函式庫
0xF102	緊急停止
0xF104	在使用圓弧運動命令時給定的參數不合理
0xF203	進給速度太快，超過每個插值時間內允許輸出的 Pulse 數
0xF204	進給加速度太快，超過每個插值時間內允許輸出的 Pulse 增量數
0xF301	X 軸座標值超出機構參數設定的工作範圍
0xF302	Y 軸座標值超出機構參數設定的工作範圍
0xF303	Z 軸座標值超出機構參數設定的工作範圍
0xF304	U 軸座標值超出機構參數設定的工作範圍
0xF305	V 軸座標值超出機構參數設定的工作範圍
0xF306	W 軸座標值超出機構參數設定的工作範圍
0xF307	A 軸座標值超出機構參數設定的工作範圍
0xF308	B 軸座標值超出機構參數設定的工作範圍
0xF401	運動命令在執行間發生錯誤
0xF501	定位確認錯誤
0xF701	X 軸碰觸硬體極限開關
0xF702	Y 軸碰觸硬體極限開關
0xF703	Z 軸碰觸硬體極限開關
0xF704	U 軸碰觸硬體極限開關
0xF705	V 軸碰觸硬體極限開關
0xF706	W 軸碰觸硬體極限開關
0xF707	A 軸碰觸硬體極限開關
0xF708	B 軸碰觸硬體極限開關
0xF801	X 軸跟隨誤差超出設定容許範圍
0xF802	Y 軸跟隨誤差超出設定容許範圍



0xF803	Z 軸跟隨誤差超出設定容許範圍
0xF804	U 軸跟隨誤差超出設定容許範圍
0xF805	V 軸跟隨誤差超出設定容許範圍
0xF806	W 軸跟隨誤差超出設定容許範圍
0xF807	A 軸跟隨誤差超出設定容許範圍
0xF808	B 軸跟隨誤差超出設定容許範圍

IV. 函式傳回值

傳 回 值 定 義	數 值	說 明
NO_ERR	0	函式呼叫成功
INITIAL_MOTION_ERR	-1	系統尚未啟動，請再次呼叫 MCC_InitSystem()
COMMAND_BUFFER_FULL_ERR	-2	運動命令緩衝區已滿，此時無法接受此筆命令。
COMMAND_NOTACCEPTED_ERR	-3	系統處於忙碌狀態，此時無法接受此筆命令。
COMMAND_NOTFINISHED_ERR	-4	執行中的運動命令尚未完成，此時無法接受此筆命令。
PARAMETER_ERR	-5	呼叫函式時所傳入的參數格式錯誤
GROUP_PARAMETER_ERR	-6	Group 參數給定錯誤，所指定為無效的 Group
FEED_RATE_ERR	-7	進給速度未設定或設定錯誤，請重新呼叫 MCC_SetFeedSpeed()
VOLTAGE_COMMAND_NOTCALLED_ERR	-9	因系統或此運動軸使用 V Command 操作模式，限制使用此函式
HOME_COMMAND_NOTCALLED_ERR	-10	目前並不在原點復歸模式下
HOLD_ILLEGAL_ERR	-11	不適當時機發出暫停(Hold)命令
CONTI_ILLEGAL_ERR	-12	不適當時機發出繼續(Continue)命令
ABORT_ILLEGAL_ERR	-13	不適當時機使用棄置(Abort)命令
RUN_TIME_ERR	-14	執行時期產生錯誤，利用呼叫 MCC_GetErrorcode() 所獲得的錯誤訊息代碼可瞭解錯誤的內容
ABORT_NOT_FINISH_ERR	-15	命令棄置動作尚未完成
GROUP_RAN_OUT_ERR	-16	已無多餘 Group 可使用

V. 運動控制函式庫初始設定

下表所列出的為呼叫 MCC_InitSystem()後，MCCL 的初始設定，初始設定若無法滿足使用者的需要，可呼叫相關函式加以更改。

初始設定內容	初始設定	相關函式
命令緩衝區大小	10000 筆命令	MCC_CreateGroupEx() MCC_GetCmdQueueSize()
運動空跑功能	未開啟	MCC_EnableDryRun() MCC_DisableDryRun()
機台允許的的最大進給速度	100	MCC_SetSysMaxSpeed() MCC_GetSysMaxSpeed()
系統座標型態	絕對座標	MCC_SetAbsolute() MCC_SetIncrease() MCC_GetCoordType()
各軸允許的最大 Pulse 加速度	32767	MCC_SetMaxPulseAcc() MCC_GetMaxPulseAcc()
各軸允許的最大 Pulse 速度	32767	MCC_SetMaxPulseSpeed() MCC_GetMaxPulseSpeed()
軟體過行程檢查	未開啟	MCC_SetOverTravelCheck() MCC_GetOverTravelCheck
硬體極限開關檢查	未開啟	MCC_EnableLimitSwitchCheck() MCC_DisableLimitSwitchCheck()
位置控制閉迴路使用的的比例增益	64	MCC_SetPGain() MCC_GetPGain()
進行直線、圓弧、圓、螺線運動時各軸使用的加、減速型式	S 形曲線	MCC_SetAccType() MCC_GetAccType() MCC_SetDecType() MCC_GetDecType()
進行直線、圓弧、圓、螺線運動時各軸使用的加、減速時間	300 ms	MCC_SetAccTime() MCC_GetAccTime() MCC_SetDecTime() MCC_GetDecTime()
進行直線、圓弧、圓、螺線運動時使用的進給速度	1	MCC_SetFeedSpeed() MCC_GetFeedSpeed()
進行點對點運動時各軸使用的速度比例	1	MCC_SetPtPSpeed() MCC_GetPtPSpeed()
定位確認最大檢查時間	1000 ms	MCC_SetInPosMaxCheckTime()
定位確認持續時間	100 ms	MCC_SetInPosSettleTime()
定位確認容許誤差範圍	∞	MCC_SetInPosToleranceEx() MCC_GetInPosToleranceEx()
定位確認功能	未開啟	MCC_EnableInPos() MCC_DiableInPos()
平滑運動功能	未開啟	MCC_EnableBlend()



		MCC_DisnableBlend()
跟隨誤差偵測功能	未開啟	MCC_EnableTrackError() MCC_DisnableTrackError()
跟隨誤差容許範圍	∞	MCC_SetTrackErrorLimit() MCC_GetTrackErrorLimit()