



EPCIO Series

運動控制函式庫

參考手冊

版本：V.6.00

日期：2020.06

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



目 錄

I. MCCL 功能列表	2
II. MCCL 函式庫	11
A. 系統功能.....	11
B. 近端輸出入接點(Local I/O)控制.....	20
C. 座標系統.....	24
D. 過行程保護.....	29
E. 直線、圓弧、圓與螺線運動(一般運動).....	32
F. 點對點運動.....	50
G. JOG 運動.....	56
H. 運動狀態檢視.....	58
I. 原點復歸.....	62
J. 定位控制.....	64
K. 進階軌跡規劃.....	75
L. 編碼器控制.....	80
M. 計時器與 Watch Dog 控制.....	86
N. Remote I/O 控制.....	90
O. D/A Converter 控制.....	100
P. A/D Converter 控制.....	104
III. 錯誤訊息代碼	113
IV. 函式傳回值	114
V. 運動控制函式庫初始設定	115
VI. MCCL 函式庫的改變	117
A. 被刪除的函式.....	117
B. 被列為過時(Obsolete)的函式.....	118
C. 行為與早期版本不同的函式.....	119
VII. 附錄	121
Revision History.....	121



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_CloseGroup()	關閉指定運動群組
4	MCC_CloseAllGroups()	關閉所有運動群組
5	MCC_SetMacParam()	設定機構參數
6	MCC_GetMacParam()	讀取機構參數
7	MCC_SetEncoderConfig()	設定編碼器參數
8	MCC_GetEncoderConfig()	讀取編碼器參數
9	MCC_SetHomeConfig()	設定原點復歸參數
10	MCC_GetHomeConfig()	讀取原點復歸參數
11	MCC_UpdateParam()	使系統反應更新後的參數
12	MCC_SetCmdQueueSize()	設定運動命令佇列大小
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_CloseSystem()	結束運動控制函式庫
16	MCC_InitSimulation	啟動模擬運動控制函式庫
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_SetUnit	設定系統座標單位
24	MCC_GetUnit	讀取系統座標單位
25	MCC_TimeDelay	使系統延遲特定時間

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_SetLIORoutineEx()	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型態
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

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()	設定目前的系統座標值

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



E. 直線、圓弧、圓與螺線運動(一般運動)

編號	函式名稱	說明
1	MCC_SetAccDecMode()	設定在進行一般運動時的加減速模式
2	MCC_GetAccDecMode()	讀取在進行一般運動時的加減速模式
3	MCC_SetAccType()	設定加速型式
4	MCC_GetAccType()	讀取使用的加速型式
5	MCC_SetDecType()	設定減速型式
6	MCC_GetDecType()	讀取使用的減速型式
7	MCC_SetAccTime()	設定加速時間
8	MCC_GetAccTime()	讀取加速時間
9	MCC_SetDecTime()	設定減速時間
10	MCC_GetDecTime()	讀取減速時間
11	MCC_SetFeedSpeed()	設定進給速度
12	MCC_GetFeedSpeed()	讀取使用的進給速度
13	MCC_GetCurFeedSpeed()	讀取機台目前的進給速度
14	MCC_GetSpeed()	讀取目前各軸的速度
15	MCC_Line()	六軸同動直線運動
16	MCC_ArcXYZ()	X-Y-Z 空間任三點圓弧運動
17	MCC_ArcXYZUVW()	X-Y-Z 空間任三點圓弧運動與輔助軸直線運動
18	MCC_ArcXY()	X-Y 平面圓弧運動
19	MCC_ArcYZ()	Y-Z 平面圓弧運動
20	MCC_ArcZX()	Z-X 平面圓弧運動
21	MCC_ArcXYUVW()	X-Y 平面圓弧與輔助軸直線運動
22	MCC_ArcYZUVW()	Y-Z 平面圓弧與輔助軸直線運動
23	MCC_ArcZXUVW()	Z-X 平面圓弧與輔助軸直線運動
24	MCC_ArcThetaXY()	X-Y 平面圓弧運動(以旋轉角度為參數)
25	MCC_ArcThetaYZ()	Y-Z 平面圓弧運動(以旋轉角度為參數)
26	MCC_ArcThetaZX()	Z-X 平面圓弧運動(以旋轉角度為參數)
27	MCC_CircleXY()	X-Y 平面全圓運動
28	MCC_CircleYZ()	Y-Z 平面全圓運動
29	MCC_CircleZX()	Z-X 平面全圓運動
30	MCC_CircleXYUVW()	X-Y 平面全圓與輔助軸直線運動
31	MCC_CircleYZUVW()	Y-Z 平面全圓與輔助軸直線運動



32	MCC_CircleZXUVW()	Z-X 平面全圓與輔助軸直線運動
33	MCC_HelicaXY_Z()	在 X-Y 平面進行圓周運動的螺線運動
34	MCC_HelicaYZ_X()	在 Y-Z 平面進行圓周運動的螺線運動
35	MCC_HelicaZX_Y()	在 Z-X 平面進行圓周運動的螺線運動

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_SetPtPAccType()	設定點對點運動的加速型式
5	MCC_GetPtPAccType()	讀取點對點運動使用的加速型式
6	MCC_SetPtPDecType()	設定點對點運動的減速型式
7	MCC_GetPtPDecType()	讀取點對點運動使用的減速型式
8	MCC_SetPtPAccTime()	設定點對點運動的加速時間
9	MCC_GetPtPAccTime()	讀取點對點運動使用的加速時間
10	MCC_SetPtPDecTime()	設定點對點運動的減速時間
11	MCC_GetPtPDecTime()	讀取點對點運動使用的減速時間

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

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_SetMaxPulseSpeed()	設定各軸最大 Pulse 速度的上限
6	MCC_GetMaxPulseSpeed()	讀取各軸最大 Pulse 速度的上限
7	MCC_SetMaxPulseAcc()	設定各軸最大 Pulse 加速度的上限
8	MCC_GetMaxPulseAcc()	讀取各軸最大 Pulse 加速度的上限
9	MCC_SetInPosMode()	設定定位確認使用模式
10	MCC_SetInPosMaxCheckTime()	設定定位檢查時間
11	MCC_SetInPosSettleTime()	設定定位持續時間
12	MCC_EnableInPos()	開啟定位確認功能
13	MCC_DisableInPos()	關閉定位確認功能
14	MCC_SetInPosToleranceEx()	設定定位誤差容許範圍
15	MCC_GetInPosToleranceEx()	讀取使用的定位誤差容許範圍
16	MCC_GetInPosStatus()	確認實際位置是否已滿足定位確認條件
17	MCC_EnableTrackError()	開啟軌跡誤差檢查功能
18	MCC_DisableTrackError()	關閉軌跡誤差檢查功能



19	MCC_SetTrackErrorLimit()	設定軌跡誤差容許範圍
20	MCC_GetTrackErrorLimit()	讀取軌跡誤差容許範圍
21	MCC_SetPCLRoutine()	串接自訂的位置控制閉迴路(PCL)中斷服務函式(ISR)

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_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
4	MCC_EnableBlend()	開啟平滑運動功能
5	MCC_DisableBlend()	關閉平滑運動功能
6	MCC_CheckBlend()	檢查是否開啟平滑運動功能
7	MCC_DelayMotion()	設定運動延遲時間
8	MCC_OverrideSpeed()	設定一般運動的速度比例
9	MCC_GetOverrideRate()	讀取使用的一般運動速度比例
10	MCC_OverridePtPSpeed()	設定點對點運動的速度比例
11	MCC_GetPtPOVERRIDERate()	讀取使用的點對點運動速度比例

L. 編碼器控制

編號	函式名稱	說明
1	MCC_SetENCRoutineEx()	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

M. 計時器與 Watch Dog 控制

編號	函式名稱	說明
1	MCC_SetTimer()	設定計時器的計時時間
2	MCC_EnableTimer()	開啟計時器計時功能
3	MCC_DisableTimer()	關閉計時器計時功能
4	MCC_EnableTimerTrigger()	開啟計時器中斷功能
5	MCC_DisableTimerTrigger()	關閉計時器中斷功能
6	MCC_SetWatchDogTimer()	設定 Watch Dog 的計時時間
7	MCC_SetWatchDogResetPeriod()	設定 Watch Dog 的 Reset 訊號持續時間
8	MCC_EnableWatchDogTimer()	開啟 Watch Dog 功能
9	MCC_DisableWatchDogTimer()	關閉 Watch Dog 功能
10	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_SetRIORoutineEx()	串接自訂的 Remote I/O 中斷服務函式 (ISR)
2	MCC_EnableRIOSetControl()	開啟指定的 Remote I/O Set 之控制功能
3	MCC_DisableRIOSetControl()	關閉指定的 Remote I/O Set 之控制功能
4	MCC_EnableRIOSlaveControl()	開啟指定的 Remote I/O Slave 資料傳輸功能
5	MCC_DisableRIOSlaveControl()	關閉指定的 Remote I/O Slave 資料傳輸功能
6	MCC_GetRIOTransStatus()	讀取目前 Remote I/O 的資料傳輸狀態
7	MCC_GetRIOMasterStatus()	讀取目前 Remote I/O Master 端傳送資料到 Slave 的狀態
8	MCC_GetRIOSlaveStatus()	讀取目前 Remote I/O Slave 端接收 Master 資料的狀態
9	MCC_GetRIOInputValue()	讀取指定 Set、Port 的 16 點輸入訊號狀態值



10	MCC_SetRIOOutputValue()	設定指定 Set、Port 的 16 點輸出訊號狀態值
11	MCC_EnquRIOOutputValue()	設定指定 Set、Port 的 16 點輸出訊號狀態值
12	MCC_SetRIOTransError()	設定 Remote I/O 資料傳輸錯誤時的重送次數
13	MCC_SetRIOTriggerType()	設定 Remote I/O 輸入接點訊號觸發中斷服務函式的型態
14	MCC_EnableRIOInputTrigger()	開啟 Remote I/O 輸入接點訊號觸發中斷服務函式的功能
15	MCC_DisableRIOInputTrigger()	關閉 Remote I/O 輸入接點訊號觸發中斷服務函式的功能
16	MCC_EnableRIOTransTrigger()	開啟 Remote I/O 「傳輸錯誤觸發」觸發中斷服務函式的功能
17	MCC_DisableRIOTransTrigger()	關閉 Remote I/O 「傳輸錯誤觸發」中斷功能

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()	串接自訂的 ADC 中斷服務函式(ISR)
2	MCC_SetADCCConvType()	設定電壓轉換型式
3	MCC_GetADCCConvType()	讀取使用的電壓轉換型式
4	MCC_SetADCCConvMode()	設定電壓轉換模式
5	MCC_GetADCInput()	讀取直流電壓輸入值
6	MCC_SetADCSingleChannel()	設定電壓轉換單一 Channel
7	MCC_GetADCWorkStatus()	讀取單一轉換 Channel 的工作狀態
8	MCC_EnableADCCConvTrigger()	開啟任一 Channel 電壓轉換完成時觸



		發的功能
9	MCC_DisableADCConvTrigger()	關閉任一 Channel 電壓轉換完成時觸發的功能
10	MCC_SetADCTagChannel()	設定電壓轉換標籤 Channel
11	MCC_EnableADCTagTrigger()	開啟標籤 Channel 電壓轉換完成時觸發的功能
12	MCC_DisableADCTagTrigger()	關閉標籤 Channel 電壓轉換完成時觸發的功能
13	MCC_SetADCCompMask()	設定進行電壓轉換時的遮蔽位元
14	MCC_SetADCCompType()	設定電壓轉換時的比較條件
15	MCC_SetADCCompValue()	設定電壓轉換時的電壓比較值
16	MCC_GetADCCompValue()	讀取使用的電壓比較值
17	MCC_EnableADCCompTrigger()	開啟滿足電壓比較條件時觸發的功能
18	MCC_DisableADCCompTrigger()	關閉滿足電壓比較條件時觸發的功能
19	MCC_EnableADCConvChannel()	開啟選定的 Channel 電壓轉換功能
20	MCC_DisableADCConvChannel()	關閉選定的 Channel 電壓轉換功能
21	MCC_StartADCConv()	啟動電壓功能
22	MCC_StopADCConv()	停止電壓功能



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	<i>strVersion</i>	指向一記憶體緩衝區，用來接收函式庫版本
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_CreateGroup(int *xMapToCh*, int *yMapToCh*, int *zMapToCh*, int *uMapToCh*, int *vMapToCh*, int *wMapToCh*, 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，必需先利用此函式建立群組，並得到新建立群組之編號，作為其傳入參數 (之一)。	
Description	此函式必需在初始化 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 ~ 5)
Parameters	<i>yMapToCh</i>	指定此 Group 中 Y 軸所對應之實體輸出 Channel(0 ~ 5)



<i>zMapToCh</i>	指定此 Group 中 Z 軸所對應之實體輸出 Channel(0 ~ 5)
<i>uMapToCh</i>	指定此 Group 中 U 軸所對應之實體輸出 Channel(0 ~ 5)
<i>vMapToCh</i>	指定此 Group 中 V 軸所對應之實體輸出 Channel(0 ~ 5)
<i>wMapToCh</i>	指定此 Group 中 W 軸所對應之實體輸出 Channel(0 ~ 5)
<i>nCardIndex</i>	指定此群組所對應之運動控制卡編號(0 ~ 11)

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_CloseGroup(int *nGroupIndex*)

Description	關閉指定的群組。	
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

4. int MCC_CloseAllGroups()

Description	關閉系統中所有群組。在第一次呼叫 MCC_CreateGroup()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

5. 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，內含欲設定之機構



		參數
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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_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，用來接收欲讀取之機構參數內容
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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_SetEncoderConfig(
    SYS_ENCODER_CONFIG* pstEncoderConfig,
    WORD wChannel,
    WORD wCardIndex
)
```

Description	設定編碼器參數的內容。	
Parameters	<i>pstEncoderConfig</i>	指向一 SYS_ENCODER_CONFIG 結構，內含欲設定之編碼器參數
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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_GetEncoderConfig**(
 SYS_ENCODER_CONFIG* *pstEncoderConfig*,
 WORD *wChannel*,
 WORD *wCardIndex*
)

Description 讀取指定軸之機構參數內容。

Parameters *pstEncoderConfig* 指向一 **SYS_ENCODER_CONFIG** 結構，用來接收欲讀
取之機構參數內容

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_SetHomeConfig**(
 SYS_HOME_CONFIG* *pstHomeConfig*,
 WORD *wChannel*,
 WORD *wCardIndex*
)

Description 設定原點復歸參數的內容。

Parameters *pstHomeConfig* 指向一 **SYS_HOME_CONFIG** 結構，內含欲設定之原
點復歸參數

wChannel 運動控制卡的輸出 Channel(0 ~ 5)

wCardIndex 運動控制卡的編號(0 ~ 11)

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_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
	<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_UpdateParam()

Description	反應更新後的機構、編碼器、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，則需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

12. int MCC_SetCmdQueueSize(

int *nSize*,

WORD *wGroupIndex*

)

Description	設定運動命令緩衝區的大小。	
Parameters	<i>nSize</i>	運動命令緩衝區的大小(單位為運動命令)
	<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_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	<p>啟動運動控制函式庫。</p> <p>除了 MCC_CreateGroup()、MCC_SetMacParam()、MCC_SetEncoder Config()、MCC_SetHomeConfig()、MCC_SetCompParam()之外，在使用 MCCL 其他函式之前，必須先呼叫此函式。此函式僅需呼叫一次即可。</p>	
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，但需視系統能否負荷而定。一般系統使用 5ms 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”EPCIO 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 ~ 12)
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_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

**16. int MCC_InitSimulation(
 int *nInterpolateTime*,
 SYS_CARD_CONFIG **pstCardConfig*,
 WORD *wCardNo***

)

Description	啟動模擬運動控制函式庫功能。	
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，但需視系統能否負荷而定。一般系統使用 5ms 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” <i>EPCIO Series 運動控制函式庫使用手冊</i> ”
	<i>wCardNo</i>	運動控制卡的使用張數(1 ~ 12)
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 MCC_InitSystem 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，此時可利用 MCC_GetCurPos 與	
-------------	--	--



MCC_GetPulsePos 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、圓與螺線)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、圓與螺線)進給速度的上限，使用單位為 User Unit/sec*。(註)

Return Value 進給速度的上限

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 ~ 5)

wCardIndex 運動控制卡編號(0 ~ 11)

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 ~ 5)

wCardIndex 運動控制卡編號(0 ~ 11)

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 ~ 11)

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 ~ 11)

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，以 EPCIO-4000/4005 運動控制卡為例，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 ~ 11)

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_SetLIORoutineEx(

LIOISR_EX *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”EPCIO 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 ~ 11)
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	<i>wTriggerType</i>	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
	<i>wPoint</i>	輸入接點編號，編號範圍 LIO_LDI0 ~ LIO_LDI6 (0 ~ 6)，輸入訊號的意義請參考”EPCIO Series 運動控制函式庫使用手冊”。
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_LDI0 ~ LIO_LDI6 (0 ~ 6)，輸入訊號的意義請參考”EPCIO Series 運動控
-------------------	---------------	---



制函式庫使用手冊’。

	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_LDI0 ~ LIO_LDI6 (0 ~ 6)，輸入訊號的意義請參考”EPCIO Series 運動控制函式庫使用手冊’
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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 採用絕對座標模式。

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*,
WORD *wGroupIndex*

)

Description	讀取目前各軸位置的直角座標值(不含補償)。	
Parameters	<i>pdfX ~ pdfW</i>	指向一 double 值，用來存放 X ~ W 各軸目前位置之直角座標值 (不含補償)
	<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

5. int MCC_GetCurPos(

double* *pdfX*,
double* *pdfY*,
double* *pdfZ*,
double* *pdfU*,
double* *pdfV*,
double* *pdfW*,
WORD *wGroupIndex*

)

Description	讀取目前各軸位置的直角座標值(含補償)。	
Parameters	<i>pdfX ~ pdfW</i>	指向一 double 值，用來存放 X ~ W 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*,
 WORD *wGroupIndex*

)

Description 讀取目前各軸位置的馬達座標值(或稱為 *pulse* 座標值，含補償)。

Parameters *plX* ~ *plW* 指向一 long 值，用來存放 X ~ W 各軸目前位置的馬達座標值(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

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，使用此函式需在指定的 Group 為運動停止狀態。

Parameters *wAxis* 指定的運動軸編號 0 ~ 5 分別代表 X ~ W 軸

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. 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	<i>wGroupIndex</i>	Group 編號
	<i>dwAxisMask</i>	指定欲發生作用的軸，指定參數可為 EPCIO_AXIS_X X 軸 EPCIO_AXIS_Y Y 軸 EPCIO_AXIS_Z Z 軸 EPCIO_AXIS_U U 軸 EPCIO_AXIS_V V 軸 EPCIO_AXIS_W W 軸 EPCIO_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： (EPCIO_AXIS_X EPCIO_AXIS_Z EPCIO_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 (
 double *dfCart* ,
 WORD *wAxis* ,
 WORD *wGroupIndex*
)

Description 設定目前的系統座標值。
使用者可重新設定目前位置座標值。在呼叫此函式成功後，此時系統座標值將更新為新設定的位置。

Parameters	<i>dfCart</i>	設定目前位置(mm)
	<i>wAxis</i>	指定欲發生作用的軸，指定參數可為
	0	X 軸
	1	Y 軸
	2	Z 軸



		3	U 軸
		4	V 軸
		5	W 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	

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* 需設定為 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(並產生錯誤紀錄)。

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紀錄(代碼 0xF701 ~ 0xF706 分別代表 X ~ W 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()清除錯誤紀錄，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；並且會產生錯誤紀錄
- 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；並且會產生錯誤紀錄
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*,
WORD *wGroupIndex*

)

Description	設定軟體過行程保護功能。	
Parameters	<i>nOTCheck0</i> ~ <i>nOTCheck5</i>	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
	<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_GetOverTravelCheck(

int* *pnOTCheck0*,
int* *pnOTCheck1*,
int* *pnOTCheck2*,
int* *pnOTCheck3*,
int* *pnOTCheck4*,



```
int*   pnOTCheck5,
WORD   wGroupIndex
```

)

Description	讀取軟體過行程保護的設定情形。	
Parameters	<i>pnOTCheck0</i> ~ <i>pnOTCheck5</i> 指向一 int 值，用來存放 X ~ W 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，使用此函式前需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 ~ 5)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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、圓與螺線運動(一般運動)

1. **int** MCC_SetAccDecMode(
 char *cAccDecMode*,
 WORD *wGroupIndex*
)

Description	設定在進行一般運動時的加減速模式。	
Parameters	<i>cAccDecMode</i>	各軸的加減速型式，可設定為： 'A' 使用後加減速模式 'B' 使用前加減速模式
	<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

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

Description	讀取在進行一般運動時的加減速模式。	
Parameters	<i>wGroupIndex</i>	Group 編號
Return Value	0	目前使用後加減速模式
	1	目前使用前加減速模式

3. **int** MCC_SetAccType(
 char *cAccType*,
 WORD *wGroupIndex*
)

Description	設定在進行一般運動時的加速型式。	
Parameters	<i>cAccType</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



	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_GetAccType(
 WORD *wGroupIndex*
)**

Description	讀取在進行一般運動時使用的加速型式。	
Parameters	<i>wGroupIndex</i>	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

**5. int MCC_SetDecType(
 char *cDecType*,
 WORD *wGroupIndex*
)**

Description	設定在進行一般運動時的減速型式。	
Parameters	<i>cDecType</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

**6. 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

7. 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 *dfAccTime* 要求的加速時間，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

8. 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 *wGroupIndex* 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

9. 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

10. 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	<i>wGroupIndex</i> 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

11. 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。但一般運動實際操作時的進給速度（可使用 MCC_GetCurFeedSpeed() 獲得）需參考是否曾使用 MCC_OverrideSpeed() 設定進給速度倍率，例如最後一次設定進給速度倍率為使用 MCC_OverrideSpeed(150)，假使此時呼叫 MCC_SetFeedSpeed(10)，則一般運動實際使用的進給速度為 $10 \times 150\% = 15$ 。
Parameters	<i>dfFeedSpeed</i> 要求的進給速度 <i>wGroupIndex</i> Group 編號
Return Value	實際設定的進給速度

12. 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，此部份請參考對 MCC_SetFeedSpeed() 的說明。
Parameters	<i>wGroupIndex</i> Group 編號
Return Value	目前設定的進給速度



13. 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

14. int MCC_GetSpeed(

double* *pdfVel0*,

double* *pdfVel1*,

double* *pdfVel2*,

double* *pdfVel3*,

double* *pdfVel4*,

double* *pdfVel5*,

WORD *wGroupIndex*

)

Description 讀取各軸目前的進給速度值。

Parameters *pdfVel0* ~ *pdfVel5* 指向一 double 值，用來存放各軸目前的進
給速度值

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

15. int MCC_Line(

double *dfX0*,

double *dfX1*,

double *dfX2*,

double *dfX3*,

double *dfX4*,

double *dfX5*,

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 ~ dfX5</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
	EPCIO_AXIS_X	X 軸
	EPCIO_AXIS_Y	Y 軸
	EPCIO_AXIS_Z	Z 軸
	EPCIO_AXIS_U	U 軸
	EPCIO_AXIS_V	V 軸
	EPCIO_AXIS_W	W 軸
	EPCIO_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：
(EPCIO_AXIS_X | EPCIO_AXIS_Z | EPCIO_AXIS_V)

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

16. 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 ~ 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>wGroupIndex</i>	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

17. int MCC_ArcXYZUVW(double *dfRX0*, double *dfRX1*, double *dfRX2*, double *dfX0*, double *dfX1*, double *dfX2*, double *dfX3*, double *dfX4*, double *dfX5*, 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 軸以同動方式進行直線運。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 ~ dfX5</i>	目的點的 U-V-W 軸座標值
	<i>wGroupIndex</i>	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

18. 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。呼叫此函式成功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 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

19. 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。呼叫此函式成功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 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

20. 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 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

21. int MCC_ArcXYUVW(

double *dfRX0,*
double *dfRX1,*
double *dfX0,*
double *dfX1,*
double *dfX3,*
double *dfX4,*
double *dfX5,*
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 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 ~ dfX5</i>	目的點的 U-V-W 軸座標值
	<i>wGroupIndex</i>	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

22. int MCC_ArcYZUVW(

double *dfRX1*,
double *dfRX2*,
double *dfX1*,
double *dfX2*,
double *dfX3*,
double *dfX4*,
double *dfX5*,
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 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 ~ dfX5</i>	目的點的 U-V-W 軸座標值
	<i>wGroupIndex</i>	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

23. int MCC_ArcZXUVW(

double *dfRX2*,
double *dfRX0*,
double *dfX2*,
double *dfX0*,
double *dfX3*,
double *dfX4*,
double *dfX5*,
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 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 ~ dfX5</i>	目的點的 U-V-W 軸座標值
	<i>wGroupIndex</i>	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

24. int MCC_ArcThetaXY(

double *dfX0*,
double *dfX1*,
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>dfX0, dfX1</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 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

25. int MCC_ArcThetaYZ(

double *dfX1*,
double *dfX2*,
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>dfX1, dfX2</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 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

26. int MCC_ArcThetaZX(

double *dfX2*,
double *dfX0*,
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>dfX2, dfX0</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 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

27. int MCC_CircleXY(

double *dfCX0*,
double *dfCX1*,
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>dfCX0, dfCX1</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 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

28. int MCC_CircleYZ(

double *dfCX1*,
double *dfCX2*,
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>dfCX1, dfCX2</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 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

29. int MCC_CircleZX(

double *dfCX2*,
double *dfCX0*,
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>dfCX2, dfCX0</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 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

30. int MCC_CircleXYUVW(

double *dfCX0*,
double *dfCX1*,
double *dfU*,
double *dfV*,
double *dfW*,
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 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>dfCX0, dfCX1</i>	圓心的 X-Y 軸座標值
<i>dfU, dfV, dfW</i>	目的點的 U-V-W 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 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

31. int MCC_CircleYZUVW(

double *dfCX1*,
double *dfCX2*,
double *dfU*,
double *dfV*,
double *dfW*,
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 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>dfCX1, dfCX2</i>	圓心的 Y-Z 軸座標值
	<i>dfU, dfV, dfW</i>	目的點的 U-V-W 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 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

32. int MCC_CircleZXUVW(

double *dfCX2*,
double *dfCX0*,
double *dfU*,
double *dfV*,
double *dfW*,
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 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>dfCX2, dfCX0</i>	圓心的 Z-X 軸座標值
	<i>dfU, dfV, dfW</i>	目的點的 U-V-W 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 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

33. int MCC_HelicaXY_Z(



double *dfCX*,
double *dfCY*,
double *dfZ*,
double *dfPitch*,
BYTE *byCirDir*,
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，並須指定目的點的 Z 軸座標值，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>dfZ</i>	目的點的 Z 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>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 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

34. int MCC_HelicaYZ_X(

double *dfCY*,
double *dfCZ*,
double *dfX*,
double *dfPitch*,
BYTE *byCirDir*,
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，並須指定目的點的 X 軸座標值，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>dfX</i>	目的點的 X 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>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 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_HelicaZX_Y(

double *dfCZ,*
double *dfCX,*
double *dfY,*
double *dfPitch,*
BYTE *byCirDir,*
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，並須指定目的點的 Y 軸座標值，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>dfY</i>	目的點的 Y 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>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 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

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(mm/s)等於 $((\text{馬達最大轉速}/60) \times \text{pitch 數} / \text{齒輪減速比}) \times (\text{速度比例} / 100)$ 其中馬達最大轉速(*wRPM*)、pitch 數(*dfPitch*)、齒輪減速比(*dfGearRatio*)皆定義在機構參數中。因此速度比例的計算可以利用： $(\text{要求的速度}) / ((\text{馬達最大轉速能帶動機台的最大速度}) \times \text{百分比})$ ，再乘上 100 而獲得。但點對點運動實際操作時的進給速度需參考是否曾使用 `MCC_OverrideSpeed()` 設定運動進給速度倍率。

Parameters *dfRatio* 速度比例，其值必須大於 0 且小於等於 100
wGroupIndex 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 *wGroupIndex* Group 編號

Return Value 大於或等於 0 進行點對點運動時使用的速度比例
小於 0 失敗，傳回值意義請參考 IV.函式傳回值

3. int MCC_PtP(

double *dfX*,
double *dfY*,
double *dfZ*,
double *dfU*,



double *dfV*,
double *dfW*,
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>dfX, dfY, dfZ</i>	目的點的 X-Y-Z 軸座標值
	<i>dfU, dfV, dfW</i>	目的點的 U-V-W 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
	EPCIO_AXIS_X	X 軸
	EPCIO_AXIS_Y	Y 軸
	EPCIO_AXIS_Z	Z 軸
	EPCIO_AXIS_U	U 軸
	EPCIO_AXIS_V	V 軸
	EPCIO_AXIS_W	W 軸
	EPCIO_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：	
	(EPCIO_AXIS_X EPCIO_AXIS_Z EPCIO_AXIS_V)	
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

4. int MCC_SetPtPAccType(

char *cAccType0*,
char *cAccType1*,
char *cAccType2*,
char *cAccType3*,
char *cAccType4*,
char *cAccType5*,
WORD *wGroupIndex*

)

Description	設定在進行點對點運動時各軸使用的加速型式。	
Parameters	<i>cType0 ~ cType5</i>	各軸的加速型式，可設定為： ‘T’ 使用梯型加速曲線 ‘S’ 使用 S 型加速曲線
	<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

5. **int** MCC_GetPtPAccType(
 char **pcAccType0*,
 char **pcAccType1*,
 char **pcAccType2*,
 char **pcAccType3*,
 char **pcAccType4*,
 char **pcAccType5*,
 WORD *wGroupIndex*

)

Description	讀取在進行點對點運動時各軸使用的加速型式。	
Parameters	<i>*pcAccType0 ~ *pcAccType5</i>	各軸的加速型式，0 表示使用梯型加速曲線，1 表示使用 S 型加速曲線
	<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

6. **int** MCC_SetPtPDecType(
 char *cDecType0*,
 char *cDecType1*,
 char *cDecType2*,
 char *cDecType3*,
 char *cDecType4*,



```

char  cDecType5,
WORD  wGroupIndex
)

```

Description	設定在進行點對點運動時各軸使用的減速型式。	
Parameters	<i>cDecType0</i> ~ <i>cDecType5</i>	各軸的減速型式，可設定為： 'T' 使用梯型減速曲線 'S' 使用 S 型減速曲線
	<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

7. int MCC_GetPtPDecType(

```

char  *pcDecType0,
char  *pcDecType1,
char  *pcDecType2,
char  *pcDecType3,
char  *pcDecType4,
char  *pcDecType5,
WORD  wGroupIndex
)

```

Description	讀取在進行點對點運動時各軸使用的減速型式。	
Parameters	<i>*pcDecType0</i> ~ <i>*pcDecType5</i>	各軸的減速型式，0 表示使用梯形減速曲線， 1 表示使用 S 型減速曲線
	<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

8. int MCC_SetPtPAccTime(

```

double dfAccTime0,
double dfAccTime1,
double dfAccTime2,

```



```
double dfAccTime3,  
double dfAccTime4,  
double dfAccTime5,  
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* ~ *dfAccTime5* 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

9. int MCC_GetPtPAccTime(

```
double *pdfAccTime0,  
double *pdfAccTime1,  
double *pdfAccTime2,  
double *pdfAccTime3,  
double *pdfAccTime4,  
double *pdfAccTime5,  
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 *pdfAccTime0* ~ *pdfAccTime5* 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

10. int MCC_SetPtPDecTime(

```
double dfTime0,  
double dfTime1,
```



```
double dfTime2,  
double dfTime3,  
double dfTime4,  
double dfTime5,  
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	<i>dfAccTime0</i> ~ <i>dfAccTime5</i> 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

11. int MCC_GetPtPDecTime(

```
double *pdfDecTime0,  
double *pdfDecTime1,  
double *pdfDecTime2,  
double *pdfDecTime3,  
double *pdfDecTime4,  
double *pdfDecTime5,  
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	<i>pdfDecTime0</i> ~ <i>pdfDecTime5</i> 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

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 *nPulse* 位移量,單位為 pulse,範圍:-2048~2048

cAxis 要求進行脈衝運動的運動軸編號
(0~5 代表 X~W 軸)

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_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 ~ 5 代表 X ~ W 軸)
	<i>wGroupIndex</i>	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

```

3. int MCC_JogConti(
    int nDir,
    int nRatio,
    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>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 ~ 5 代表 X ~ W 軸)
	<i>wGroupIndex</i>	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

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	GMS_RUNNING 處於運動狀態，尚有運動命令未執行完成 GMS_STOP 處於停止狀態，已無庫存運動命令 GMS_HOLD 處於暫停狀態(因使用者呼叫 MCC_HoldMotion) GMS_DELAYING 處於延遲狀態(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];  
} 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：進給速度

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

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”*EPCIO 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	<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_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	<i>pwStockCount</i>	指向一 WORD 值，用來存放 Pulse 命令庫存數目
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)

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>	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 硬體 FIFO 的最大使用個數

8. int MCC_GetErrorCode(WORD *wGroupIndex*)

Description 讀取目前錯誤紀錄，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 100 ms)呼叫此函式確認系統目前工作正常，若發現錯誤紀錄，則須採取相對應之錯誤回復(Error Recovery)處理。

Parameters *wGroupIndex* Group 編號

Return Value 0 無錯誤
其它 錯誤代碼，請參考 III. 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紀錄，否則系統仍無法正常運作。

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

I. 原點復歸

1. int MCC_Home(

```
int nOrder0,  
int nOrder1,  
int nOrder2,  
int nOrder3,  
int nOrder4,  
int nOrder5,  
WORD wCardIndex
```

)

Description 執行原點復歸運動，關於其設定請參考 MCC_SetHomeConfig()函式。使用此函式時，可配合呼叫 MCC_GetGoHomeStatus()，來檢查動作是否完成。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* ~ *nOrder5* 各軸進行原點復歸的順序。可設定值為 0 ~ 5，數字越小越早執行復歸動；不執行原點復歸動作的運動軸，需將復歸順序設定為 255。需特別注意，這些參數所對應的是編號 *wCardIndex* 控制卡上的 0 ~ 5 號輸出軸，而不是 Group 中之運動軸。更詳細的說明請參考”EPCIO 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 ~ 11)

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_GetHomeSensorStatuss(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 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 ~ 5)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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”EPCIO Series 運動控制函式庫使用手冊 2.7.5 齒輪齒隙、背隙補償”。

Parameters *pstCompParam* 指向一 SYS_COMP_PARAM 結構，用來描述補償參數
wChannel 運動控制卡的輸出 Channel(0 ~ 5)
wCardIndex 運動控制卡編號(0 ~ 11)

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 *wCardIndex*

)

Description 設定位置閉迴路控制使用的比例增益(P Gain)。

Parameters *wGain0* ~ *wGain5* 各軸使用的比例增益，設定範圍為 1 ~ 16256

wCardIndex 運動控制卡編號(0 ~ 11)

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 *wCardIndex*

)

Description 讀取位置閉迴路控制使用的比例增益(P Gain)。

Parameters *pwGain0* ~ *pwGain5* 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 ~ 11)

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_SetMaxPulseSpeed(

int *nPulse0*,

int *nPulse1*,

int *nPulse2*,

int *nPulse3*,

int *nPulse4*,

int *nPulse5*,

**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” <i>EPCIO 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</i> ”。	
Parameters	<i>nPulse0 ~ nPulse5</i>	各軸最大 pulse 速度的上限 設定範圍為 1~32767，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 ~ 11)
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_GetMaxPulseSpeed(**int*** *pnSpeed0,***int*** *pnSpeed1,***int*** *pnSpeed2,***int*** *pnSpeed3,***int*** *pnSpeed4,***int*** *pnSpeed5,***WORD** *wCardIndex*

)

Description	讀取各軸最大 pulse 速度的上限。	
Parameters	<i>pnSpeed0 ~ pnSpeed5</i>	指向一 int 值，用來存放各軸最大 pulse 速度的上限
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_SetMaxPulseAcc(



```

int   nPulseAcc0,
int   nPulseAcc1,
int   nPulseAcc2,
int   nPulseAcc3,
int   nPulseAcc4,
int   nPulseAcc5,
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”EPCIO 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>nPulseAcc0</i> ~ <i>nPulseAcc5</i> 各軸最大 pulse 加速度的上限，設定範圍為 1~32767，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 ~ 11)
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_GetMaxPulseAcc(

```

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

```

)

Description	讀取各軸最大 pulse 加速度的上限。
Parameters	<i>pnPulseAcc0</i> ~ <i>pnPulseAcc5</i> 指向一 int 值，用來存放各軸最大 pulse 加速度的上限



	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_SetInPosMode(

WORD *wMode*,

WORD *wGroupIndex*

)

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

Parameters *wMode* 定位確認模式

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

10. 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”*EPCIO 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

11. 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”*EPCIO Series 運動控制函*”

式庫使用手冊 2.7.2 定位確認”。

Parameters	<i>wSettleTime</i>	定位確認持續時間，單位 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

12. 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
	EPCIO_AXIS_X	X 軸
	EPCIO_AXIS_Y	Y 軸
	EPCIO_AXIS_Z	Z 軸
	EPCIO_AXIS_U	U 軸
	EPCIO_AXIS_V	V 軸
	EPCIO_AXIS_W	W 軸
	EPCIO_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：	
	(EPCIO_AXIS_X EPCIO_AXIS_Z EPCIO_AXIS_V)	
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

13. 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 EPCIO_AXIS_X X 軸 EPCIO_AXIS_Y Y 軸 EPCIO_AXIS_Z Z 軸 EPCIO_AXIS_U U 軸 EPCIO_AXIS_V V 軸 EPCIO_AXIS_W W 軸 EPCIO_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： (EPCIO_AXIS_X EPCIO_AXIS_Z EPCIO_AXIS_V)
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

14. int MCC_SetInPosToleranceEx(

double *dfTolerance0*,
double *dfTolerance1*,
double *dfTolerance2*,
double *dfTolerance3*,
double *dfTolerance4*,
double *dfTolerance5*,
WORD *wGroupIndex*

)

Description	設定各軸定位誤差容許範圍。	
Parameters	<i>dfTolerance0</i> ~ <i>dfTolerance5</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

15. int MCC_GetInPosToleranceEx(

double* *pdfTolerance0*,



double* *pdfTolerance1*,
double* *pdfTolerance2*,
double* *pdfTolerance3*,
double* *pdfTolerance4*,
double* *pdfTolerance5*,
WORD *wGroupIndex*

)

Description	讀取各軸定位誤差容許範圍。	
Parameters	<i>pdfTolerance0</i> ~ <i>pdfTolerance5</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

16. int MCC_GetInPosStatus(

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

)

Description	讀取各軸定位確認狀況。	
Parameters	<i>pbyInPos0</i> ~ <i>pbyInPos5</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

17. 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”*EPCIO 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

EPCIO_AXIS_X X 軸

EPCIO_AXIS_Y Y 軸

EPCIO_AXIS_Z Z 軸

EPCIO_AXIS_U U 軸

EPCIO_AXIS_V V 軸

EPCIO_AXIS_W W 軸

EPCIO_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：

(EPCIO_AXIS_X | EPCIO_AXIS_Z | EPCIO_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

18. 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

19. 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(0~5 代表 X~W 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

20. 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(0~5 代表 X~W 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

21. int MCC_SetPCLRoutine(

PCLISR *pfnENCRoutine*,
WORD *wCardIndex*

)

Description	串接自訂的位置控制閉迴路(PCL)中斷處理函式(ISR)，當位置閉迴路控制失效時系統將自動呼叫此中斷處理函式。詳細說明請參考“EPCIO 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 ~ 11)
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_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

4. 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 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

5. int MCC_DisableBlend(WORD *wGroupIndex*)

Description	關閉連續運動功能。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
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

6. int MCC_CheckBlend(WORD *wGroupIndex*)

Description	檢查是否開啟連續運動功能。	
Parameters	<i>wGroupIndex</i>	Group 編號
Return Value	0	已開啟連續運動功能
	1	未開啟連續運動功能



其他

失敗，傳回值的意義請參考 IV.函式傳回值

7. 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 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

8. double MCC_OverrideSpeed(**double *dfRate*,****WORD *wGroupIndex*****)**

Description 設定一般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 *dfRate* *dfRate* 為變更進給速度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

dfRate 的設定值必須大於 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此設定值

wGroupIndex Group 編號

Return Value 大於 0 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

9. 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 0	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

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

)

Description	設定點對點運動的速度強制比例，使用此項函式將即時變動各軸的速度。	
Parameters	<i>dfRate</i>	<i>dfRate</i> 為變更各軸的速度比例為原來速度比例的多少百分比再乘以 100，也就是說點對點各軸新的速度比例將等於(原來速度比例 × <i>dfRate</i> / 100) <i>dfRate</i> 的設定範圍為大於或等於 1 的整數，若 <i>dfRate</i> 小於 1，則自動將 <i>dfRate</i> 設定 1；若更新後各軸的速度超過 <code>MCC_SetSysMaxSpeed()</code> 的設定值，則新的進給速度將等於此設定值
	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於 1	實際設定的速度強制比例
	其他	失敗，傳回值的意義請參考 V.函式傳回值

11. double MCC_GetPtPOVERRIDERate(**WORD *wGroupIndex***

)

Description	讀取目前點對點運動使用的速度強制比例。	
Parameters	<i>wGroupIndex</i>	Group 編號
Return Value	大於或等於 1	點對點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**

L. 編碼器控制

此節主要講述內容為 EPCIO 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”EPCIO 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_SetENCRoutineEx(

ENCISR_EX *pfnENCRoutine*,

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 ~ 11)
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 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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* *pIValue*,
 WORD *wChannel*,
 WORD *wCardIndex*
)

Description	讀取編碼器的計數值。	
Parameters	<i>pIValue</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 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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，可設定值有：
	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閘鎖新的計數值，次數不限
	<i>wChannel</i>	運動軸之編號(0 ~ 5)

	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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(

WORD *wSource*,

WORD *wChannel*,

WORD *wCardIndex*

)

Description 設定觸發門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 Switch 時，皆會將編碼器計數值記錄在第 0 張卡的第 0 個 Channel 的門鎖暫存器內。

Parameters	<i>wSource</i>	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_LIO0	發生近端輸入接點 DI 0 的中斷
	ENC_TRIG_LIO1	發生近端輸入接點的 DI 1 中斷
	ENC_TRIG_RDI0	發生遠端輸入接點 Set 0 的 DI 0 中斷
	ENC_TRIG_RDI1	發生遠端輸入接點 Set 0 的 DI 1 中斷
	ENC_TRIG_ADC0	發生 ADC Channel 0 的比較條件成立
	ENC_TRIG_ADC1	發生 ADC Channel 1 的比較條件成立
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)



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* *pIValue*,
WORD *wChannel*,
WORD *wCardIndex*

)

Description	讀取 門鎖暫存器內的紀錄值。	
Parameters	<i>pIValue</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 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)

	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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	<i>pwStatus</i>	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
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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 *lValue*, WORD *wChannel*, WORD *wCardIndex*)

Description	設定編碼器的比較值。	
Parameters	<i>lValue</i>	編碼器的比較值
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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 ~ 5)
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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 EPCIO 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”EPCIO Series 運動控制函式庫使用手冊 2.13 計時器與 Watch Dog 控制”一節共同閱讀。

1. 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 25ns，設定範圍為 1~16777215
wCardIndex 運動控制卡的編號(0~11)

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_EnableTimer(WORD *wCardIndex*)

Description 開啟計時器計時功能。

Parameters *wCardIndex* 運動控制卡的編號(0~11)

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_DisableTimer(WORD *wCardIndex*)

Description 關閉計時器計時功能。

Parameters *wCardIndex* 運動控制卡的編號(0~11)



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_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 *wCardIndex* 運動控制卡的編號(0 ~ 11)

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_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 *wCardIndex* 運動控制卡的編號(0 ~ 11)

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_SetWatchDogTimer(

WORD *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 *dwValue* Watch Dog 的倒數計時時間。單位為以 MCC_SetTimer()

		所設定之計時器計時週期，設定範圍為 1 ~ 65535
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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_SetWatchDogResetPeriod(

WORD *wValue*,

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>wValue</i>	硬體 Reset 訊號持續時間，單位為 25ns
	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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_EnableWatchDogTimer(

WORD *wCardIndex*

)

Description	開啟 Watch Ddog 功能。	
Parameters	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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_DisableWatchDogTimer(

WORD *wCardIndex*

)

Description	關閉 Watch Dog 功能。	
Parameters	<i>wCardIndex</i>	運動控制卡的編號(0 ~ 11)
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_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 *wCardIndex* 運動控制卡的編號(0 ~ 11)

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 EPCIO 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”EPCIO 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_SetRIORoutineEx(**
 RIOISR_EX *pfnRIORoutine,*
 WORD *wCardIndex*
)

Description	串接自訂的 Remote I/O 中斷服務函式。		
Parameters	<i>pfnRIORoutine</i>	自訂 Remote I/O 中斷服務函式的函式指標	
	<i>wSet</i>	Remote I/O Set 編號	
	<i>RIO_SET0</i>	Remote I/O Set 0	
	<i>RIO_SET1</i>	Remote I/O Set 1	
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)	
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_EnableRIOSetControl(**
 WORD *wSet,*
 WORD *wCardIndex*
)

Description	開啟指定之 Remote I/O Set 的資料傳輸功能，而該 Set 的 Slave 之資料傳輸功能須再呼叫 MCC_EnableRIOSlaveControl() 才會開啟。		
Parameters	<i>wSet</i>	Remote I/O Set 編號	
	<i>RIO_SET0</i>	Remote I/O Set 0	
	<i>RIO_SET1</i>	Remote I/O Set 1	
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)	
	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_DisableRIOSetControl(

WORD *wSet*,

WORD *wCardIndex*

)

Description 關閉指定之 Remote I/O Set 的資料傳輸功能，該 Set 的 Slave 之資料傳輸功能此時也將一併關閉。

Parameters *wSet* Remote I/O Set 編號
RIO_SET0 Remote I/O Set 0
RIO_SET1 Remote I/O Set 1
wCardIndex 運動控制卡編號(0 ~ 11)

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_EnableRIOSlaveControl(

WORD *wSet*,

WORD *wCardIndex*

)

Description 開啟指定的 Remote I/O Slave 的資料傳輸功能。Slave 的資料傳輸功能開啟後，必須再呼叫 MCC_EnableRIOSetControl()開啟 Set 的資料傳輸功能，Remote I/O 模組才開始進行傳送與接收。

Parameters *wSet* Remote I/O Set 編號
RIO_SET0 Remote I/O Set 0
RIO_SET1 Remote I/O Set 1
wCardIndex 運動控制卡編號(0 ~ 11)

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_DisableRIOSlaveControl(

WORD *wSet*,



WORD *wCardIndex*

)

Description	關閉指定 Remote I/O Set 的 Slave 之資料傳輸功能。	
Parameters	<i>wSet</i>	Remote I/O Set 編號
	<i>RIO_SET0</i>	Remote I/O Set 0
	<i>RIO_SET1</i>	Remote I/O Set 1
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_GetRIOTransStatus(

WORD* *pwStatus,*

WORD *wSet,*

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_GetRIOMasterStatus()及 MCC_GetRIOSlaveStatus()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>wSet</i>	Remote I/O Set 編號
	<i>RIO_SET0</i>	Remote I/O Set 0
	<i>RIO_SET1</i>	Remote I/O Set 1
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_GetRIOMasterStatus(

WORD* *pwStatus,*

WORD *wSet,*

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>wSet</i>	Remote I/O Set 編號 <i>RIO_SET0</i> Remote I/O Set 0 <i>RIO_SET1</i> Remote I/O Set 1
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_GetRIOSlaveStatus(

WORD* *pwStatus*,
WORD *wSet*,
WORD *wCardIndex*

)

Description	讀取目前 Remote I/O Slave 端接收 Master 資料的狀態。	
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 Slave 端接收訊號正常，0 則否
	<i>wSet</i>	Remote I/O Set 編號 <i>RIO_SET0</i> Remote I/O Set 0 <i>RIO_SET1</i> Remote I/O Set 1
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_GetRIOInputValue(

WORD* *pwValue*,
WORD *wSet*,



WORD *wPort*,
WORD *wCardIndex*

)

Description	讀取指定 Set、Port 的 16 點輸入訊號狀態值。	
Parameters	<i>pwValue</i>	指向一 WORD 值，用來存放指定位置 (Set、Port) 中 16 點輸入訊號的輸入訊號狀態 (bit 0 ~ bit 15 分別代表此 Port 中第 0 點到第 15 點的狀態)
	<i>wSet</i>	Remote I/O Set 編號 <i>RIO_SET0</i> Remote I/O Set 0 <i>RIO_SET1</i> Remote I/O Set 1
	<i>wPort</i>	輸入訊號的 Port 編號 <i>RIO_PORT0</i> Remote I/O 輸入點 DI 0 ~ DI 15 <i>RIO_PORT1</i> Remote I/O 輸入點 DI 16 ~ DI 31 <i>RIO_PORT2</i> Remote I/O 輸入點 DI 32 ~ DI 47 <i>RIO_PORT3</i> Remote I/O 輸入點 DI 48 ~ DI 63
	<i>wCardIndex</i>	運動控制卡編號 (0 ~ 11)
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_SetRIOOutputValue(

WORD *wValue*,
WORD *wSet*,
WORD *wPort*,
WORD *wCardIndex*

)

Description	設定指定 Set、Port 的 16 點輸出訊號狀態值	
Parameters	<i>wValue</i>	指定位置 (Set、Port) 中 16 點輸出訊號的輸出訊號狀態 (bit 0 ~ bit 15 分別代表此 Port 中第 0 點到第 15 點的狀態)
	<i>wSet</i>	Remote I/O Set 編號



		<i>RIO_SET0</i>	Remote I/O Set 0
		<i>RIO_SET1</i>	Remote I/O Set 1
	<i>wPort</i>	輸出訊號的 Port 編號	
		<i>RIO_PORT0</i>	Remote I/O 輸出點 DO 0 ~ DO 15
		<i>RIO_PORT1</i>	Remote I/O 輸出點 DO 16 ~ DO 31
		<i>RIO_PORT2</i>	Remote I/O 輸出點 DO 32 ~ DO 47
		<i>RIO_PORT3</i>	Remote I/O 輸出點 DO 48 ~ DO 63
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)	
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_EnqRIOOutputValue(

WORD *wValue*,
WORD *wMask*,
WORD *wSet*,
WORD *wPort*,
WORD *wCardIndex*

)

Description	設定指定 Set、Port 的 16 點輸出訊號狀態值。		
Parameters	<i>wValue</i>	指定位置(Set、Port)中 16 點輸出訊號的輸出訊號狀態(bit 0 ~ bit 15 分別代表此 Port 中第 0 點到第 15 點的狀態)	
	<i>wMask</i>	欲輸出之遮蔽位置(若無輸出之遮蔽位置，將保留最後一次 MCC_SetRIOOutputValu()之狀態值)	
	<i>wSet</i>	Remote I/O Set 編號	
		<i>RIO_SET0</i>	Remote I/O Set 0
		<i>RIO_SET1</i>	Remote I/O Set 1
	<i>wPort</i>	輸出訊號的 Port 編號	
		<i>RIO_PORT0</i>	Remote I/O 輸出點 DO 0 ~ DO 15



		<i>RIO_PORT1</i>	Remote I/O 輸出點 DO 16 ~ DO 31
		<i>RIO_PORT2</i>	Remote I/O 輸出點 DO 32 ~ DO 47
		<i>RIO_PORT3</i>	Remote I/O 輸出點 DO 48 ~ DO 63
	<i>wCardIndex</i>		運動控制卡編號(0 ~ 11)
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: MCC_SetRIOOutputValue() 為即時反應。

MCC_EnqRIOOutputValue() 會進入運動命令暫存器。

12. int MCC_SetRIOTransError(

WORD *wTime*,
WORD *wSet*,
WORD *wCardIndex*

)

Description	設定 Remote I/O 傳輸時最大重送次數，重送次數預設值為 16。當資料無法正確傳輸時，EPCIO Series 運動控制卡將重新傳輸資料，當重送次數到達此設定值，仍無法正確傳輸資料時，將產生資料傳輸錯誤(此時呼叫 MCC_GetRIOTransStatus() 將得到資料傳輸狀態異常的結果)。		
Parameters	<i>wTime</i>		資料傳輸錯誤重送次數(0 ~ 16)
	<i>wSet</i>		Remote I/O Set 編號
		<i>RIO_SET0</i>	Remote I/O Set 0
		<i>RIO_SET1</i>	Remote I/O Set 1
	<i>wCardIndex</i>		運動控制卡編號(0 ~ 11)
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_SetRIOTriggerType(

WORD *wType*,
WORD *wSet*,
WORD *wDigitalInput*,

WORD *wCardIndex*

)

Description	設定 Remote I/O 輸入訊號觸發中斷服務函式的型態為「上緣觸發」、「下緣觸發」或「轉態觸發」，每一組 Slave 的前四點輸入接點(RIO_DI0、RIO_DI1、RIO_DI2、RIO_DI3)可觸發使用者自訂的中斷服務函式。本函式設定完成後，必須呼叫 MCC_EnableRIOInputTrigger()函式，開啟中斷功能。		
Parameters	wType	Remote I/O 輸入訊號觸發中斷的型態 <i>RIO_INT_RISE</i> 上緣觸發 <i>RIO_INT_FALL</i> 下緣觸發 <i>RIO_INT_LEVEL</i> 轉態觸發	
	wSet	Remote I/O Set 編號 <i>RIO_SET0</i> Remote I/O Set 0 <i>RIO_SET1</i> Remote I/O Set 1	
	wDigitalInput	Remote I/O Slave 輸入點 <i>RIO_DI0</i> Remote I/O Slave 輸入點 0(DI 0) <i>RIO_DI1</i> Remote I/O Slave 輸入點 1(DI 1) <i>RIO_DI2</i> Remote I/O Slave 輸入點 2(DI 2) <i>RIO_DI3</i> Remote I/O Slave 輸入點 3(DI 3)	
	wCardIndex	運動控制卡編號(0 ~ 11)	
	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_EnableRIOInputTrigger(

WORD *wSet,*

WORD *wDigitalInput,*

WORD *wCardIndex*

)

Description	每一組 Slave 的前四點輸入接點 (RIO_DI0、RIO_DI1、RIO_DI2、RIO_DI3) 可觸發使用者自訂的中斷服務函式。本函式用來開啟 RIO_DI0 ~ RIO_DI3
-------------	--



中斷功能。

Parameters	<i>wSet</i>	Remote I/O Set 編號
		<i>RIO_SET0</i> Remote I/O Set 0
	<i>RIO_SET1</i> Remote I/O Set 1	
	<i>wDigitalInput</i>	Remote I/O Slave 輸入點
		<i>RIO_DI0</i> Remote I/O Slave 輸入點 0(DI 0)
		<i>RIO_DI1</i> Remote I/O Slave 輸入點 1(DI 1)
		<i>RIO_DI2</i> Remote I/O Slave 輸入點 2(DI 2)
		<i>RIO_DI3</i> Remote I/O Slave 輸入點 3(DI 3)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
	Return Value	0
非零		失敗，傳回值的意義請參考 IV.函式傳回值

15. int MCC_DisableRIOInputTrigger(

WORD *wSet*,

WORD *wDigitalInput*,

WORD *wCardIndex*

)

Description 每一組 Slave 的前四點輸入接點(DI0、DI1、DI2、DI3)可觸發使用者自訂的中斷服務函式。本函式用來關閉 DI0 ~ DI3 中斷功能。

Parameters	<i>wSet</i>	Remote I/O Set 編號
		<i>RIO_SET0</i> Remote I/O Set 0
	<i>RIO_SET1</i> Remote I/O Set 1	
	<i>wDigitalInput</i>	Remote I/O Slave 輸入點
		<i>RIO_DI0</i> Remote I/O Slave 輸入點 0(DI 0)
		<i>RIO_DI1</i> Remote I/O Slave 輸入點 1(DI 1)
		<i>RIO_DI2</i> Remote I/O Slave 輸入點 2(DI 2)
		<i>RIO_DI3</i> Remote I/O Slave 輸入點 3(DI 3)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
	Return Value	0

非零

失敗，傳回值的意義請參考 IV.函式傳回值

16. int MCC_EnableRIOTransTrigger(

WORD *wSet*,

WORD *wCardIndex*

)

Description 開啟 Remote I/O 「資料傳輸錯誤」中斷功能。

Parameters *wSet*

Remote I/O Set 編號

RIO_SET0

Remote I/O Set 0

RIO_SET1

Remote I/O Set 1

wCardIndex

運動控制卡編號(0 ~ 11)

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_DisableRIOTransTrigger(

WORD *wSet*,

WORD *wCardIndex*

)

Description 關閉 Remote I/O 「資料傳輸錯誤」中斷功能。

Parameters *wSet*

Remote I/O Set 編號

RIO_SET0

Remote I/O Set 0

RIO_SET1

Remote I/O Set 1

wCardIndex

運動控制卡編號(0 ~ 11)

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 EPCIO 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”EPCIO 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(-10 ~ 10 V)
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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 ~ 5)不使用 V Command 操作模式時(也就是機構參數中的 *wCommandMode* 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 ~ 5)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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 *dwSource* 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_ADC0	ADC 0 特定電壓輸入值
DAC_TRIG_ADC1	ADC 1 特定電壓輸入值
DAC_TRIG_ADC2	ADC 2 特定電壓輸入值
DAC_TRIG_ADC3	ADC 3 特定電壓輸入值
DAC_TRIG_ADC4	ADC 4 特定電壓輸入值
DAC_TRIG_ADC5	ADC 5 特定電壓輸入值
DAC_TRIG_ADC6	ADC 6 特定電壓輸入值
DAC_TRIG_ADC7	ADC 7 特定電壓輸入值
DAC_TRIG_LDI0	近端輸入接點 DI 0 訊號輸入(OT0+)
DAC_TRIG_LDI1	近端輸入接點 DI 1 訊號輸入(OT1+)
DAC_TRIG_LDI2	近端輸入接點 DI 2 訊號輸入(OT2+)
DAC_TRIG_LDI3	近端輸入接點 DI 3 訊號輸入(OT3+)
DAC_TRIG_R0DI0	遠端輸入接點 Set 0 的 DI 0 訊號輸入

	DAC_TRIG_R0DI1	遠端輸入接點 Set 0 的 DI 1 訊號輸入
	DAC_TRIG_R0DI2	遠端輸入接點 Set 0 的 DI 2 訊號輸入
	DAC_TRIG_R0DI3	遠端輸入接點 Set 0 的 DI 3 訊號輸入
	DAC_TRIG_R1DI0	遠端輸入接點 Set 1 的 DI 0 訊號輸入
	DAC_TRIG_R1DI1	遠端輸入接點 Set 1 的 DI 1 訊號輸入
	DAC_TRIG_R1DI2	遠端輸入接點 Set 0 的 DI 2 訊號輸入
	DAC_TRIG_R1DI3	遠端輸入接點 Set 0 的 DI 3 訊號輸入
	<i>wChannel</i>	運動控制卡的輸出 Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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 *wChannel* 運動控制卡的輸出 Channel(0 ~ 5)
wCardIndex 運動控制卡編號(0 ~ 11)

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 *wChannel* 運動控制卡的輸出 Channel(0 ~ 5)



	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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 <i>wCardIndex</i> 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 ~ 11)
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 <i>wCardIndex</i> 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	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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 EPCIO 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”EPCIO 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	<i>pfnADCRoutine</i>	自訂 ADC 中斷服務函式的函式指標
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_SetADCCnvType(

WORD *wConvType*,

WORD *wChannel*,

WORD *wCardIndex*

)

Description	設定 ADC 電壓轉換型式為雙極性或單極性轉換型式。	
Parameters	<i>wConvType</i>	電壓轉換型式設定 ADC_TYPE_BIP 雙極性轉換型式(Bipolar)，電壓範圍：-5 ~ 5 V ADC_TYPE_UNI 單極性轉換型式(Unipolar)，電壓範圍為 0 ~ 10 V
	<i>wChannel</i>	ADC Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
	Return Value	0

非零

失敗，傳回值的意義請參考 IV.函式傳回值

3. int MCC_GetADCConvType(

WORD* *pwConvType*,**WORD** *wChannel*,**WORD** *wCardIndex*

)

Description 讀取 ADC 電壓轉換型式。

Parameters *pwConvType* 指向一 WORD 值，用來存放電壓轉換型式，可能值如下：

ADC_TYPE_BIP 雙極性轉換型式(Bipolar)，電壓範圍：-5 ~ 5 V

ADC_TYPE_UNI 單極性轉換型式(Unipolar)，電壓範圍為 0 ~ 10 V

wChannel ADC Channel(0 ~ 5)

wCardIndex 運動控制卡編號(0 ~ 11)

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_SetADCConvMode(

WORD *wConvMode*,**WORD** *wCardIndex*

)

Description 設定 ADC 電壓轉換型態為單一電壓轉換(Single Run)或連續電壓轉換(Free Run)型態。

Parameters *wConvMode* 電壓轉換模式，有以下模式可供設定：

ADC_MODE_SINGLE 單一電壓轉換(Single Run)

ADC_MODE_FREE 連續電壓轉換(Free Run)

wCardIndex 運動控制卡編號(0 ~ 11)

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_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 (Unipolar)轉換型式」，輸入電壓的有效值為 0 ~ 10 V。若 ADC 設定為「雙極性轉換型式(Bipolar)」則輸入電壓的有效值為-5 ~ 5 V。	
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 ~ 5)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_SetADCSingleChannel(

WORD *wChannel*,
WORD *wCardIndex*

)

Description	設定某一 ADC Channel 為「Single Channel」，配合 MCC_SetADCConvMode()將轉換模式設定為單一電壓轉換模式，則在呼叫 MCC_StartADCConv()後，此選定的 Channel 會將電壓值轉換一次。轉換完成後即不再進行轉換，使用者需再次呼叫 MCC_StartADCConv()才會進行下一單次的轉換。轉換期間可經由 MCC_GetADCWorkStatus()確認轉換過程是否完成。	
Parameters	<i>wChannel</i>	ADC Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_GetADCWorkStatus(WORD* *pwStatus*, WORD *wCardIndex*)

Description 讀取目前 ADC 的工作狀態。

Parameters *pwStatus* 指向一 WORD 值，用來存放 ADC 的工作狀態；1 代表正在轉換，0 則否
wCardIndex 運動控制卡編號(0 ~ 11)

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_EnableADCCnvTrigger(WORD *wCardIndex*)

Description 開啟任一 ADC Channel 電壓轉換完成時觸發使用者自訂中斷服務函式的功能。

Parameters *wCardIndex* 運動控制卡編號(0 ~ 11)

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_DisableADCCnvTrigger(WORD *wCardIndex*)

Description 關閉任一 ADC Channel 電壓轉換完成時觸發使用者自訂中斷服務函式的功能。

Parameters *wCardIndex* 運動控制卡編號(0 ~ 11)

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_SetADCTagChannel(

WORD *wChannel*,

WORD *wCardIndex*

)

Description 設定某一 ADC 為標籤 Channel，配合 MCC_EnableADCTagTrigger() 函式，當標籤 Channel 電壓轉換完成時可觸發使用者自訂的中斷服務函式。

Parameters *wChannel* ADC Channel(0 ~ 5)
wCardIndex 運動控制卡編號(0 ~ 11)

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_EnableADCTagTrigger(

WORD *wCardIndex*

)

Description 開啟 ADC 標籤 Channel 電壓轉換完成時觸發使用者自訂的中斷服務函式之功能。

Parameters *wCardIndex* 運動控制卡編號(0 ~ 11)

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_DisableADCTagTrigger(

WORD *wCardIndex*

)

Description 關閉 ADC 標籤 Channel 電壓轉換完成時觸發使用者自訂的中斷服務函式之功能。

Parameters *wCardIndex* 運動控制卡編號(0 ~ 11)

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_SetADCCompMask(

WORD *wMask*,

WORD *wCardIndex*

)

Description 當輸入電壓與設定之比較值比較時，可遮蔽最小幾個 bit 不作比較動作，如此可降低比較器的靈敏度，避免因輸入電壓跳動造成中斷持續發生。

Parameters

<i>wMask</i>	電壓遮蔽位元，可設定位元旗標有：
<i>ADC_MASK_NO</i>	不使用電壓遮蔽位元
<i>ADC_MASK_BIT1</i>	使用 1 個電壓遮蔽位元
<i>ADC_MASK_BIT2</i>	使用 2 個電壓遮蔽位元
<i>ADC_MASK_BIT3</i>	使用 3 個電壓遮蔽位元
<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)

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_SetADCCompType(

WORD *wCompType*,

WORD *wChannel*,

WORD *wCardIndex*

)

Description 設定 ADC 電壓比較型式，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 觸發訊號亦可用來閃鎖編碼器的計數值。

Parameters

<i>wCompType</i>	電壓比較型式，可設定值有：
<i>ADC_COMP_RISE</i>	輸入電壓由小到大通過比較值
<i>ADC_COMP_FALL</i>	輸入電壓由大到小通過比較值
<i>ADC_COMP_LEVEL</i>	輸入電壓值改變並通過比較值
<i>wChannel</i>	ADC Channel(0 ~ 5)

	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_SetADCCompValue(

float *fValue*,
WORD *wChannel*,
WORD *wCardIndex*

)

Description 設定 ADC Channel 在雙極性(Bipolar)轉換模式下的輸入電壓比較值，本函式不提供單極性(Unipolar)轉換模式下電壓比較功能。設定本函式後必須再使用 MCC_Set ADCCompType()及 MCC_EnableADCCompTrigger()函式，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(-5 ~ 5 V)
	<i>wChannel</i>	ADC Channel(0 ~ 5)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_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 ~ 5)
	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_EnableADCCompTrigger(

WORD *wChannel*,

WORD *wCardIndex*

)

Description 開啟滿足 ADC 電壓比較條件時觸發使用者自訂的中斷服務函式之功能。

Parameters *wChannel* ADC Channel(0 ~ 5)
wCardIndex 運動控制卡編號(0 ~ 11)

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_DisableADCCompTrigger(

WORD *wChannel*,

WORD *wCardIndex*

)

Description 關閉滿足 ADC 電壓比較條件時觸發使用者自訂的中斷服務函式之功能。

Parameters *wChannel* ADC Channel(0 ~ 5)
wCardIndex 運動控制卡編號(0 ~ 11)

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_EnableADCConvChannel(

WORD *wChannel*,

WORD *wCardIndex*

)

Description 開啟 ADC Channel 輸入電壓類比轉數位功能。本函式設定的轉換 Channel 必須在連續電壓轉換(Free Run)模式下才有效，設定完成後必須再呼叫 MCC_StartADCConv()函式，啟動 ADC 轉換功能。

Parameters *wChannel* ADC Channel(0 ~ 5)

	<i>wCardIndex</i>	運動控制卡編號(0 ~ 11)
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_DisableADCCConvChannel(WORD *wChannel*, WORD *wCardIndex*)

Description 關閉 ADC Channel 輸入電壓類比轉數位功能。

Parameters *wChannel* ADC Channel(0 ~ 5)
wCardIndex 運動控制卡編號(0 ~ 11)

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_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()函式使用。

Parameters *wCardIndex* 運動控制卡編號(0 ~ 11)

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_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 ~ 11)

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
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
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
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

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()函式
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了解錯誤的內容
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。

初使設定內容	初 始 設 定	相 關 函 式
命令緩衝區大小	10000 筆命令	MCC_SetCmdQueueSize() 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、圓與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、圓與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、圓與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()
連續運動功能	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()

VI. MCCL 函式庫的改變

本節列出 MCCL 5.0 版(或以上版本)與前一版本之差異處。若是初次接觸 MCCL 的使用者，可直接略過本節；但若使用者原先使用早期版本之 MCCL，請仔細閱讀本節說明。

A. 被刪除的函式

函式名稱	原因及處理方式
MCC_RedefineCoord	本版 MCCL 不允許有不同運動軸對應到同一實體 Channel 的情況，故不再需要此函式
MCC_SetInterpolateTime	插值時間在系統正常運作時不應被動態修改，故為維持系統穩定，本版 MCCL 不再支援此函式；若使用者需動態改變插值時間，請執行 MCC_CloseSystem() 再重新呼叫 MCC_InitSystem()
MCC_GetInterpolateTime	插值時間為 MCC_InitSystem() 的參數之一，若使用者在程式中其它地方須要此值，請使用者自行保存其值
MCC_GetErrorCount	Error Counter 計數值為 MCCL 內部使用所需，對使用者而言並無意義，故本版 MCCL 不再支援此函式
MCC_GetInPosStableTime	本版 MCCL 新增三種(共四種)定位確認模式，不再需要此函式，故本版 MCCL 不再支援此函式
MCC_ChangeFeedSpeed	此函式與 MCC_OverrideSpeed() 功能相似，為避免使用者混淆，故本版 MCCL 不再支援此函式，請使用 MCC_OverrideSpeed()
MCC_ChangePtPSpeed	此函式與 MCC_OverrideSpeed() 功能相似，為避免使用者混淆，故本版 MCCL 不再支援此函式，請使用 MCC_OverrideSpeed()
MCC_SetCycleInterruptRoutine	為避免使用者自訂函式與 MCCL 內部插值補間動作互相影響，故本版 MCCL 不再支援此函式；請改用 MCCL 所提供之計時器功能
MCC_SetAccStep	請參考 MCC_SetAccTime()
MCC_GetAccStep	MCC_GetAccTime()
MCC_SetDecStep	MCC_SetDecTime()
MCC_GetDecStep	MCC_GetDecTime()
MCC_SetPtPAccStep	請參考 MCC_SetAccTime()



MCC_GetPtPAccStep	MCC_GetAccTime()
MCC_SetPtPDecStep	MCC_SetDecTime()
MCC_GetPtPDecStep	MCC_GetDecTime()
MCC_SetGoHomeAccTime	請參考 MCC_SetHomeConfig()
MCC_GetGoHomeAccTime	
MCC_SetGoHomeDecTime	
MCC_GetGoHomeDecTime	
MCC_SetGoHomeAccStep	
MCC_GetGoHomeAccStep	
MCC_SetGoHomeDecStep	
MCC_GetGoHomeDecStep	
MCC_SetLeaveHomeSensorSpeed	

B. 被列為過時(Obsolete)的函式

下列函式在此版 MCCL 中被列為過時函式，其存在只因為要相容於早期版本。雖然在此版 MCCL 中仍然可以正常使用這些函式，但使用者應盡可能避免，因其可能在未來版本的 MCCL 中被刪除。

函式名稱	替代函式
MCC_SetGroupConfig	MCC_CreateGroup() MCC_CloseGroup() MCC_CloseAllGroups()
MCC_SetInPosCheckTime	MCC_SetInPosMaxCheckTime()
MCC_SetInPosTolerance	MCC_SetInPosToleranceEx()
MCC_GetInPosTolerance	MCC_GetInPosToleranceEx()
MCC_AbortMotion	MCC_AbortMotionEx()
MCC_SetDACClockDivider	N/A (不需要)
MCC_SetADCClockDivider	
MCC_SetRIOClockDivider	
MCC_SetMachParam	MCC_SetMacParam()
MCC_GetMachParam	MCC_GetMacParam()



MCC_UpdateMachParam	MCC_SetEncoderConfig() MCC_SetHomeConfig() MCC_UpdateParam()
MCC_GoHome	MCC_SetHomeConfig() MCC_Home()
MCC_LineX MCC_LineY MCC_LineZ MCC_LineU MCC_LineV MCC_LineW	MCC_Line()
MCC_PtPX MCC_PtPY MCC_PtPZ MCC_PtPU MCC_PtPV MCC_PtPW	MCC_PtP()

C. 行為與早期版本不同的函式

函式名稱	行為差異
MCC_EnableLimitSwitchCheck	<ol style="list-style-type: none"> 早期版本中若碰觸極限開關時，系統僅停止輸出命令，但不產生錯誤紀錄(可以立即使用 Jog 函式退出極限開關區域)，使用者必須自行呼叫 MCC_GetLimitSwitchStatus() 始可得知是否發生此錯誤；而新版中改為將產生錯誤紀錄，故使用者只要呼叫 MCC_GetErrorCode() 就可知道目前是否發生此錯誤，待錯誤排除後必須呼叫 MCC_ClearError() 始可以 Jog 方式退出極限開關 在新版本中若成功呼叫此函式，則無論傳入參數為何，都只會在碰觸之極限開關與運動方向吻合時，才會產生錯誤紀錄並停止運動
MCC_Home MCC_GoHome	早期版本中完成原點復歸動作後，會產生與 MCC_ResetMotion() 相似結果，使系統回復到初始狀



	態；而新版本中僅會 Reset 執行原點復歸的運動軸
MCC_AbortGoHome	早期版本中呼叫此函式後，除停止原點復歸外，亦產生與 MCC_ResetMotion()相似結果；而新版本中僅會停止原點復歸動作
MCC_DelayMotion	計時單位由插值時間改為 ms
MCC_AbortMotionEx	請參考其函式說明



VII. 附錄

Revision History

日期	版本	修改內容
2020/06/22	6.00	更新函式與參數說明、排版、語意與錯字修正、新增各手冊間的對應章節。
2010/02/26	5.10	增加 4 個函式 MCC_DefinePos ()、MCC_SetAccDecMode()、MCC_GetAccDecMode()、MCC_EnquRIOOutputValue()。 功能列表修改，P.3、P.8。函式庫修改，P.25-26、P.30、P.91-92。