

i-8092 運動控制模組使用手冊

(Version 2.4)

應用程式函式庫
WinCon-8000、I-8000 系列控制器



ICP DAS CO., LTD.
泓格科技股份有限公司

Warranty

All products manufactured by ICPDAS Inc. are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICPDAS Inc. assumes no liability for damages consequent to the use of this product. ICPDAS Inc. reserves the right to change this manual at any time without notice. The information furnished by ICPDAS Inc. is believed to be accurate and reliable. However, no responsibility is assumed by ICPDAS Inc. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright 1997-2009 by ICPDAS Inc., LTD. All rights reserved worldwide.

Trademark

The names used for identification only maybe registered trademarks of their respective companies.

License

The user can use, modify and backup this software on a single machine. The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

目錄

1 前言	6
1.1 手冊的使用	6
1.2 基本函式和巨集函式	6
1.3 函式基本結構說明	6
1.3 巨集函式類別	7
1.4 基本函式類別	7
2 基本設定功能	8
2.1 各軸定義的代碼	8
2.2 註冊軸卡及版本讀取	8
2.3 軸卡重置	10
2.4 軸輸出PULSE模式設定	10
2.5 設定軸速度輸出最大範圍	11
2.6 設定軸前後極限的輸入觸發邏輯	11
2.7 設定碰觸前後極限的處理模式	12
2.8 設定軸近原點輸入觸發邏輯	12
2.9 設定軸原點輸入觸發邏輯	13
2.10 設定軸前後軟體極限,參考來源及取消	14
2.11 設定編碼器輸入參數	15
2.12 伺服開關(Servo_ON/OFF)	16
2.13 設定伺服馬達異常ALARM輸入參數	17
2.14 設定伺服馬達定位完成輸入參數	17
2.15 設定數位輸入雜訊濾波功能	18
2.16 指定軸為圓形運動軸(環狀計數器)	19
2.17 三角形速度曲線的預防	20
2.18 外部輸入驅動	21
2.18.1 手輪脈波驅動	21
2.18.2 固定脈波驅動	22
2.18.3 連續脈波驅動	22
2.18.4 外部輸入關閉	23
2.19 載入設定檔並完成相關的硬體設定	24
3 狀態讀取及設定功能	25

3.1 設定及讀取指令邏輯位置	25
3.2 設定及讀取ENCODER位置	26
3.3 讀取目前速度	27
3.4 讀取目前加速度	27
3.5 讀取目前DI狀態	28
3.6 讀取目前ERROR狀態	30
3.7 一般DO 輸出	31
4 FRNET功能(I8092F專用函式)	32
4.1 FRnet DI讀取	32
4.2 FRnet DO寫入	32
5 軸自動歸零	33
5.1 軟體自動歸零功能	34
5.2 軟體單步驟歸零功能	37
5.2.1 尋找近原點開關	37
5.2.2 尋找原點開關	38
5.2.3 尋找伺服馬達Z相信號	39
6 軸控功能	40
6.1 各軸獨立運動	40
6.1.1 設定單軸加減速模式	40
6.1.2 設定軸初始速度	42
6.1.3 設定軸定速度	42
6.1.4 設定軸加速度	43
6.1.5 設定軸減速度	43
6.1.6 設定軸加速度變化率	44
6.1.7 設定軸減速(保留脈波數)	44
6.1.8 固定脈波數輸出	45
6.1.9 連續脈波輸出	45
6.2 補間運動	46
6.2.1 設定補間加減速模式	46
6.2.2 設定軸向量初始速度	49
6.2.3 設定軸向量定速度	49
6.2.4 設定軸向量加速度	50
6.2.5 設定軸向量減速度	50
6.2.6 設定軸向量加速度變化率	51
6.2.7 設定軸向量減速(保留脈波數)	51
6.2.8 二軸直線補間	52
6.2.9 二軸圓弧補間	53

6.2.10 二軸圓形補間.....	55
6.3 連續補間運動.....	56
6.3.1 二軸矩形連續補間.....	56
6.3.2 二軸直線連續補間.....	57
6.3.3 二軸多段連續補間.....	58
6.3.4 二軸比例運動.....	60
6.3.5 二軸混合連續補間.....	62
6.4 中斷條件因子設定.....	64
6.4.1 中斷條件設定.....	64
6.4.2 中斷條件關閉.....	67
6.4.3 讀取中斷狀態.....	67
6.5 其他功能.....	68
6.5.1 設定軸暫停.....	68
6.5.2 設定軸啓動.....	68
6.5.3 等待完成軸運動.....	69
6.5.4 設定軸停止.....	70
6.5.5 清除停止狀態.....	73
6.5.6 補間動作結束(單軸運動或改變座標系).....	73
6.5.7 設定COMPARE值.....	74

1 前言

1.1 手冊的使用

- 使用 i8092 運動控制模組，去設計你的自動化設備時，本手冊提供了完整且詳細的說明，幫助你很快的找到你要的運動控制函式，並配合簡單的範例，迅速開發你的應用程式。
- 手冊分為六大章，本章是手冊的前言，2、3、4、5、6 五章為巨集函式(MF)的說明。
- 本手冊需搭配泓格公司所提供的應用程式函式庫(DLL)，它支援各類軟體平台(eVCpp、VB.net、C#.net)及作業系統(MiniOS7 / WinCE / Linux)。

1.2 基本函式和巨集函式

- 巨集函式替使用者架構了簡單易用的程式撰寫環境，降低了運動控制高難度的門檻。直覺式的參數設計、客製化巨集的運動函式、連續補間及減速點的自動運算.....，已經符合絕大多數使用者的需求，希望這能提供使用者一個更好的選擇。

1.3 函式基本結構說明

- 函式名稱(參數一, 參數二,))

功能: 函式基本功能說明。

參數: 參數的定義及使用方法。

回應: 函式的回傳值。

範例: 簡單的示範參考程式。(手冊中的範例皆以 C++ code 撰寫)

備註: 備忘註解。

1.3 巨集函式類別

泓格 ICPDAS 針對 eVC++ 提供一 CI8092MF 的類別，可以引導及幫助程式撰寫巨集函式，詳細請參考，快速上手第四章。

1.4 基本函式類別

泓格 ICPDAS 針對 eVC++ 提供一 CI8092 的類別，可以引導及幫助程式撰寫基本函式，詳細請參考，快速上手第四章。


```

if (Found == 0)
{
    //找不到軸卡，異常處理。
    return;
}
//===== I-8000 =====
//設定各槽(slot0~slot7)，對應的卡號為 1~8。
BYTE cardNo;
BYTE slot;
int Found = 0;
for (slot = 0; slot < 8; slot++)
{
    cardNo = slot + 1;
    if (i8092MF_REGISTRATION(cardNo, slot) == YES)
    {
        //找到軸卡，註冊。
        i8092MF_RESET_CARD(cardNo);
        Found++;
    }
}
if (Found == 0)
{
    //找不到軸卡，異常處理。
    return;
}

```

● WORD i8092MF_GET_VERSION(void)

功能： 讀取 i8092 運動函式庫之版本。

參數： 無

回應： WORD 版本號碼 0x0000 ~ 0x9999

範例： WORD VER_No;
VER_No = i8092MF_GET_VERSION();
//讀取 i8092.dll 版本號碼。

備註： 以下為讀到的版本release資訊(2006年7月)
i8092MF_GET_VERSION: 0x0607
06 函式庫版本年流水序
07 函式庫版本月流水序

2.3 軸卡重置

- **void i8092MF_RESET_CARD(BYTE cardNo)**

功能： I-8092 重設成電源開啟狀態。

參數： **cardNo:** 指定卡號

回應： 無

範例： **i8092MF_RESET_CARD (1);**
//重置第 1 卡。

2.4 軸輸出 PULSE 模式設定

- **void i8092MF_SET_PULSE_MODE(BYTE cardNo, WORD axis, BYTE nMode)**

功能： 設定軸之輸出模式，包含 CW/CCW 或 PULSE/DIR，及正方向定義。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼(參考表 2-1)
nMode: 指定模式(參考表 2-2)

回應： 無

範例： **i8092MF_SET_PULSE_MODE(1, AXIS_XY, 2);**
//指定第 1 卡 XY 軸，脈波輸出模式皆為 2。

脈波輸出模式表(2-2)

形式	模式	脈波信號輸出	
		nPP	nPM
CW / CCW	0	CW(正緣觸發)	CCW(正緣觸發)
	1	CW(負緣觸發)	CCW(負緣觸發)
PULSE / DIR	2	PULSE(正緣觸發)	DIR (LOW:正轉/HIGH:反轉)
	3	PULSE(負緣觸發)	DIR (LOW:正轉/HIGH:反轉)
	4	PULSE(正緣觸發)	DIR (HIGH:正轉/LOW:反轉)
	5	PULSE(負緣觸發)	DIR (HIGH:正轉/LOW:反轉)

2.5 設定軸速度輸出最大範圍

- **void i8092MF_SET_MAX_V**(BYTE *cardNo*, WORD *axis*, DWORD *data*)

功能: 設定軸之輸出最高速度 PPS 限制，影響: 最高速度越小，速度解析度越高，反之越大 (速度總共有 8000 段)。

參數: **cardNo:** 指定卡號
axis: 指定軸號碼(參考表 2-1)
data: 指定最高速度，單軸(8,000~4,000,000 PPS)
補間最高速度，Y 軸(8,000~2,828,854 PPS)

回應: 無

範例: **i8092MF_SET_MAX_V(1, AXIS_XY, 200000L);**
//設定第 1 卡 XY 軸，最高速為 200K PPS，每段速度為 $200000 / 8000 = 25$ PPS。

2.6 設定軸前後極限的輸入觸發邏輯

- **void i8092MF_SET_HLMT**(BYTE *cardNo*, WORD *axis*, BYTE *nFLEdge*, BYTE *nRLEdge*)

功能: 設定軸之"前後極限"開關觸發邏輯。

參數: **cardNo:** 指定卡號
axis: 指定軸號碼(參考表 2-1)
nFLEdge: 前極限觸發邏輯: 0=低準位觸發, 1=高準位觸發
nRLEdge: 後極限觸發邏輯: 0=低準位觸發, 1=高準位觸發

回應: 無

範例: **i8092MF_SET_HLMT(1, AXIS_XY, 0, 0);**
//設定第1卡 XY 軸，其"前後極限"觸發邏輯，全部為低準位觸發。

2.7 設定碰觸前後極限的處理模式

- **void i8092MF_LIMITSTOP_MODE** (BYTE *cardNo*, WORD *axis*, BYTE *nMode*)

功能： 設定碰觸"前後極限"處理模式。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼(參考表 2-1)
nMode: 設定處理方法: 0=立即停止,1=減速後停止

回應： 無

範例： **i8092MF_LIMITSTOP_MODE(1, AXIS_X, 0);**
//設定第 1 卡 X 軸，碰觸前後極限後立即停止。

2.8 設定軸近原點輸入觸發邏輯

- **void i8092MF_SET_NHOME**(BYTE *cardNo*, WORD *axis*, BYTE *nNHEdge*)

功能： 設定軸之"近原點"開關觸發邏輯。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼(參考表 2-1)
nNHEdge: "近原點"開關觸發邏輯: 0=低準位觸發, 1=高準位觸發

回應： 無

範例： **i8092MF_SET_NHOME(1, AXIS_XY, 0);**
//設定第 1 卡 XY 軸，其"近原點"開關，觸發邏輯全部為低準位觸發。

2.9 設定軸原點輸入觸發邏輯

● **void i8092MF_SET_HOME_EDGE**(BYTE *cardNo*, WORD *axis*, BYTE *nHEdge*)

功能： 設定軸之"原點"開關觸發邏輯。

參數：
cardNo: 指定卡號
axis: 指定軸號碼(參考表 2-1)
nHEdge: "原點"開關觸發邏輯: 0=低準位觸發, 1=高準位觸發

回應： 無

範例：
`i8092MF_SET_HOME_EDGE(1, AXIS_XY, 1);`
//設定第 1 卡 XY 軸，其"原點"開關，觸發邏輯全部為高準位觸發。

2.10 設定軸前後軟體極限,參考來源及取消

- **void i8092MF_SET_SLMT**(BYTE *cardNo*, WORD *axis*, long *dwFL*, long *dwRL*, BYTE *nType*)

功能: 設定軸之"前後軟體極限"功能。

參數: **cardNo:** 指定卡號
axis: 指定軸號碼(參考表 2-1)
dwFL: 前軟體極限值(-2,147,483,648 ~ +2,147,483,647)
dwRL: 後軟體極限值(-2,147,483,648 ~ +2,147,483,647)
nType: 比較對象: 0=指令輸出位置,1=實際編碼器回饋位置

回應: 無

範例: `i8092MF_SET_SLMT(1, AXIS_XY, 20000, -3000, 0);`
`//設定第 1 卡 XY 軸，以指令輸出位置做比較，前軟體極限=20000，後軟體極限=-3000。`

- **void i8092MF_CLEAR_SLMT**(BYTE *cardNo*, WORD *axis*)

功能: 取消軸之"前後軟體極限"功能。

參數: **cardNo:** 指定卡號
axis: 指定軸號碼(參考表 2-1)

回應: 無

範例: `i8092MF_CLEAR_SLMT(1, AXIS_XY);`
`//取消第 1 卡 XY 軸，前後軟體極限功能。`

2.11 設定編碼器輸入參數

- **void i8092MF_SET_ENCODER**(BYTE *cardNo*, WORD *axis*, BYTE *nMode*, BYTE *nDevision*, BYTE *nZEdge*)

功能： 設定軸之編碼器輸入參數。

參數：
cardNo: 指定卡號
axis: 指定軸號碼(參考表 2-1)
nMode: 編碼器輸入模式: 0=AB 相輸入,1=上下計數輸入
nDevision: 模式為 AB 相輸入時,指定除頻: 0=1:1, 1=1:2, 2=1:4
nZEdge: 設定伺服 Z 輸入信號觸發邏輯: 0=低準位觸發, 1=高準位觸發

回應： 無

範例：
`i8092MF_SET_ENCODER(1, AXIS_XY, 0, 0, 0);`
//設定第 1 卡 XY 軸，編碼器輸入為 AB 相，不除頻，Z 輸入信號低準位觸發。

- **void i8092MF_SET_EN_DIR**(BYTE *cardNo*, WORD *axis*, BYTE *nDir*)

功能： 設定軸之編碼器輸入正反向。

參數：
cardNo: 指定卡號
axis: 指定軸號碼(參考表 2-1)
nDire: 編碼器輸入方向: 0=正向輸入,1=反向輸入

回應： 無

範例：
`i8092MF_SET_EN_DIR (1, AXIS_XY, 1);`
//設定第 1 卡 XY 軸，編碼器輸入為反向輸入。

2.12 伺服開關(Servo_ON/OFF)

- **void i8092MF_SERVO_ON**(BYTE *cardNo*, WORD *axis*)

功能： 設定軸驅動器伺服啟動。

參數： *cardNo*: 指定卡號
axis: 指定軸號碼(參考表 2-1)

回應： 無

範例： `i8092MF_SERVO_ON(1, AXIS_XY);`
//設定第 1 卡 XY 軸，啟動驅動器伺服。

- **void i8092MF_SERVO_OFF**(BYTE *cardNo*, WORD *axis*)

功能： 設定軸驅動器伺服關閉。

參數： *cardNo*: 指定卡號
axis: 指定軸號碼(參考表 2-1)

回應： 無

範例： `i8092MF_SERVO_OFF(1, AXIS_XY);`
//設定第 1 卡 XY 軸，關閉驅動器伺服。

2.13 設定伺服馬達異常 ALARM 輸入參數

● **void i8092MF_SET_ALARM**(BYTE *cardNo*, WORD *axis*, BYTE *nMode*, BYTE *nAEdge*)

功能： 設定軸之驅動器異常(ALARM)輸入參數。

參數：
cardNo: 指定卡號
axis: 指定軸號碼(參考表 2-1)
nMode: 模式: 0=關閉,1=開啟
nAEdge: 設定異常(ALARM)輸入信號觸發邏輯: 0=低準位觸發, 1=高準位觸發

回應： 無

範例：
`i8092MF_SET_ALARM(1, AXIS_XY, 1, 0);`
//設定第 1 卡 X Y 軸，異常(ALARM)輸入為開啟，輸入信號觸發邏輯為低準位觸發。

2.14 設定伺服馬達定位完成輸入參數

● **void i8092MF_SET_INPOS**(BYTE *cardNo*, WORD *axis*, BYTE *nMode*, BYTE *nIEdge*)

功能： 設定軸之驅動器定位完成輸入參數。

參數：
cardNo: 指定卡號
axis: 指定軸號碼(參考表 2-1)
nMode: 模式: 0=關閉,1=開啟
nIEdge: 設定定位完成輸入信號觸發邏輯: 0=低準位觸發, 1=高準位觸發

回應： 無

範例：
`i8092MF_SET_INPOS(1, AXIS_X, 1, 0);`
//設定第 1 卡 X 軸，定位完成輸入為開啟，輸入信號觸發邏輯為低準位觸發。

備註： 請配合硬體接線使用,參考(Fig. 2.12 一般 DI 輸入接線範例)

2.15 設定數位輸入雜訊濾波功能

● **void i8092MF_SET_FILTER**(BYTE *cardNo*, WORD *axis*, WORD *FEn*, WORD *FLn*)

功能： 設定軸之輸入數位濾波項目及濾波時間參數。

參數：
cardNo: 指定卡號
axis: 指定軸號碼(參考表 2-1)
FEn: 濾波項目:欲開啟項目代號加總值(0~31)如下表:

代號	開啟項目
1	緊急停止,前後極限, 近原點, 原點
2	編碼器 Z-相輸入
4	定位完成,伺服 ALARM
8	nEXPP, nEXPM, EXPLSN
16	輸入信號(IN3)

FLn: 設定濾波時間參數(0~7) 如下表:

代號	可移除最大雜訊寬(width)	輸入延遲時間
0	1.75 μ SEC	2 μ SEC
1	224 μ SEC	256 μ SEC
2	448 μ SEC	512 μ SEC
3	896 μ SEC	1.024mSEC
4	1.792mSEC	2.048mSEC
5	3.584mSEC	4.096mSEC
6	7.168mSEC	8.192mSEC
7	14.336mSEC	16.384mSEC

回應： 無

範例：**i8092MF_SET_FILTER(1, AXIS_XY, 21, 3);**
//設定第 1 卡 X Y 軸，(21=1+4+16) 1→緊急停止、前後極限、近原點、原點，4→定位完成、伺服 ALARM，16→輸入信號(IN3)輸入濾波為開啟，濾波時間常數=1.024mSEC。

2.16 指定軸為圓形運動軸(環狀計數器)

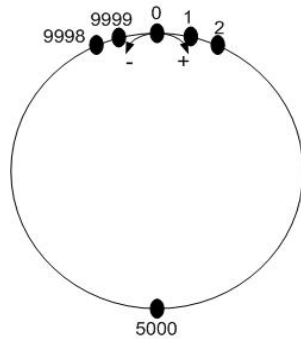
● void i8092MF_VRING_ENABLE(BYTE cardNo, WORD axis, DWORD nVRing)

功能： 指定軸啟動為環狀計數器 (如附圖)。

參數：
cardNo: 指定卡號
axis: 指定軸號碼(參考表 2-1)
nVRing: 環狀計數器值(-2,147,483,648 ~ +2,147,483,647)

回應： 無

範例：
i8092MF_VRING_ENABLE(1, AXIS_X, 9999);
//設定第 1 卡 X 軸，指定為環狀計數器，一圈為 10000 Pulse。



環狀計數器=9999

例如：

我們設計轉一圈為 10000 Pulse, 環狀計數器值設為 9999 正轉到 9999 後下一 Pulse 歸為 0,1..重新計算起

備註：
1.此功能,會同時使指令位置計數器及實際位置計數器同時有效,不能分別設定
2.此功能啟動後,軟體極限功能將不能使用

● void i8092MF_VRING_DISABLE(BYTE cardNo, WORD axis)

功能： 指定軸關閉環狀計數器功能。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)

回應： 無

範例：
i8092MF_VRING_DISABLE(1, AXIS_X);
//設定第 1 卡 X 軸，關閉環狀計數器功能。

2.17 三角形速度曲線的預防

- **void i8092MF_AVTRI_ENABLE(BYTE cardNo, WORD axis)**

功能： 致能預防三角形速度曲線的產生。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 (參考表 2-1)

回應： 無

範例： **i8092MF_AVTRI_ENABLE(1, AXIS_X);**
//設定第 1 卡 X 軸，致能預防三角形速度的產生。

- **void i8092MF_AVTRI_DISABLE(BYTE cardNo, WORD axis)**

功能： 除能預防三角形速度的產生。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 (參考表 2-1)

回應： 無

範例： **i8092MF_AVTRI_DISABLE(1, AXIS_X);**
//設定第 1 卡 X 軸，除能預防三角形速度的產生。

2.18 外部輸入驅動

2.18.1 手輪脈波驅動

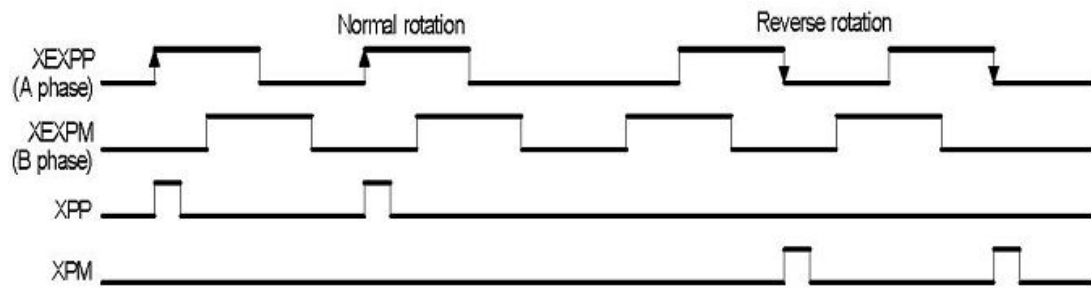
● **void i8092MF_EXD_MP(BYTE cardNo, WORD axis, long data)**

功能： 執行手輪輸入驅動，輸出固定步數。

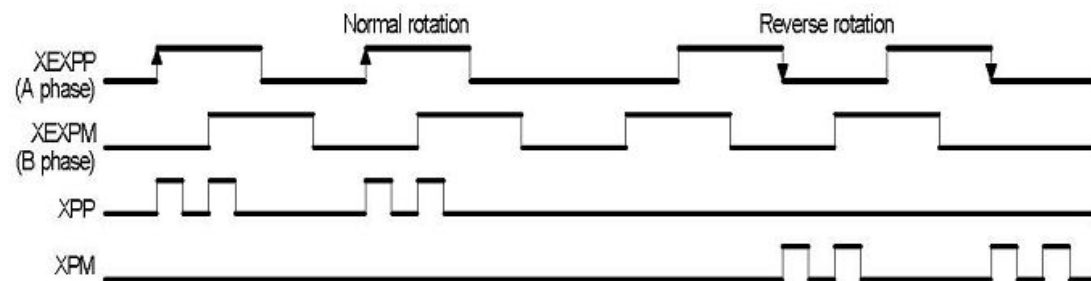
參數：
cardNo: 指定卡號
axis: 指定軸號碼 X 或 Y (1 or 2)
data: 指定步數

回應： 無

範例：
i8092MF_EXD_MP(1, AXIS_X, 1);
//第1卡 X 軸，手輪觸發移動1步(Pulse)。



i8092MF_EXD_MP(1, AXIS_X, 2);
//第1卡 X 軸，手輪觸發移動2步(Pulse)。



2.18.2 固定脈波驅動

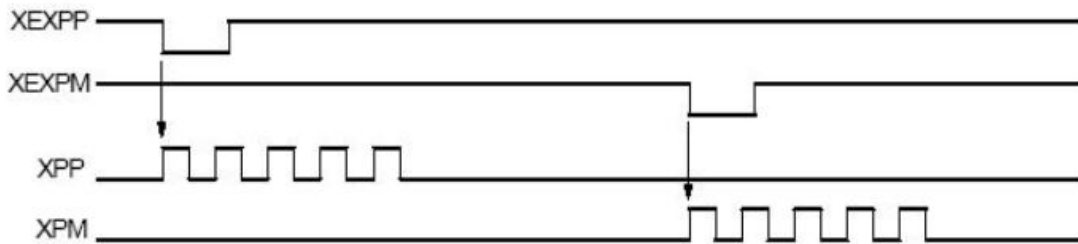
● **void** i8092MF_EXD_FP(BYTE cardNo, WORD axis, long data)

功能： 執行外部輸入驅動，輸出固定步數。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 X 或 Y (1 or 2)
data: 指定步數

回應： 無

範例：
i8092MF_EXD_FP(1, AXIS_X, 5);
//第 1 卡 X 軸，外部觸發移動 5 步(Pulse)。



2.18.3 連續脈波驅動

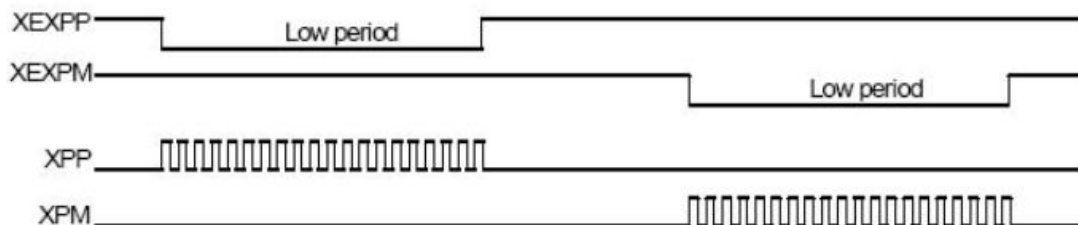
● **void** i8092MF_EXD_CP(BYTE cardNo, WORD axis, long data)

功能： 執行外部輸入驅動固定步數輸出。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 X 或 Y (1 or 2)
data: 設定速度

回應： 無

範例：
i8092MF_EXD_CP(1, AXIS_X, 20);
//第 1 卡 X 軸，觸發速度 20 PPS 移動。



2.18.4 外部輸入關閉

● **void i8092MF_EXD_DISABLE**(BYTE *cardNo*, WORD *axis*)

功能： 關閉外部輸入驅動功能。

參數： *cardNo*: 指定卡號
axis: 指定軸號碼 X 或 Y(1 or 2)

回應： 無

範例： **i8092MF_EXD_DISABLE(1, AXIS_X);**
//關閉第1卡 X 軸，外部輸入驅動功能。

2.19 載入設定檔並完成相關的硬體設定

- **short** `i8092MF_LOAD_CONFIG` (**BYTE** `cardNo`)

功能： 載入由 PACEzGo 所儲存的設定檔並完成相關的硬體設定。

參數： `cardNo`: 指定卡號

回應： `0`：正常
`-1`：無法開啟設定檔

範例： `i8092MF_LOAD_CONFIG(1);`
`//載入設定檔並設定第 1 卡。`

3 狀態讀取及設定功能

3.1 設定及讀取指令邏輯位置

- **void** i8092MF_SET_LP(BYTE cardNo, WORD axis, long wdata)

功能： 設定軸之目前指令位置。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)
wdata: 指令位置(-2,147,483,648 ~ +2,147,483,647)

回應： 無

範例：
i8092MF_SET_LP(1, AXIS_XY, 0);
//設定第 1 卡 XY 軸，目前指令位置皆為 0。

- **long** i8092MF_GET_LP(BYTE cardNo, WORD axis)

功能： 讀取軸目前之指令位置。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 X 或 Y(1 or 2)

回應： 目前指令位置 (-2,147,483,648 ~ +2,147,483,648)

範例：
long X_LP;
X_LP = i8092MF_GET_LP(1, AXIS_X);
//讀取第 1 卡 X 軸，目前指令位置值。

3.2 設定及讀取 ENCODER 位置

- **void** i8092MF_SET_EP(BYTE cardNo, WORD axis, long wdata)

功能： 設定軸之目前 ENCODER 回授位置。

參數：
cardNo: 指定卡號
axis: 指定軸號碼(參考表 2-1)
wdata: 指令位置(-2,147,483,648 ~ +2,147,483,647)

回應： 無

範例：
i8092MF_SET_EP(1, AXIS_XY, 0);
//設定第 1 卡 XY 軸，目前 ENCODER 回授位置皆為 0。

- **long** i8092MF_GET_EP(BYTE cardNo, WORD axis)

功能： 讀取軸目前之 ENCODER 回授位置。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 X 或 Y(1 or 2)

回應： 目前指令位置 (-2,147,483,648 ~ +2,147,483,648)

範例：
long X_EP;
X_EP = i8092MF_GET_EP(1, AXIS_X);
//讀取第 1 卡 X 軸，目前 ENCODER 回授位置值。

3.3 讀取目前速度

- **DWORD i8092MF_GET_CV(BYTE cardNo, WORD axis)**

功能: 讀取軸目前之運動速度。

參數: **cardNo:** 指定卡號
axis: 指定軸號碼 X 或 Y(1 or 2)

回應: 目前速度(PPS)

範例: **DWORD** dwdata;
dwdata = i8092MF_GET_CV(1, AXIS_X);
//讀取第 1 卡 X 軸，目前之運動速度。

3.4 讀取目前加速度

- **DWORD i8092MF_GET_CA(BYTE cardNo, WORD axis)**

功能: 讀取軸目前之運動加速度 PPS/Sec。

參數: **cardNo:** 指定卡號
axis: 指定軸號碼 X 或 Y(1 or 2)

回應: 目前加速度(PPS/Sec)

範例: **DWORD** dwdata;
dwdata = i8092MF_GET_CA(1, AXIS_X);
//讀取第 1 卡 X 軸，目前之運動加速度。

3.5 讀取目前 DI 狀態

● BYTE i8092MF_GET_DI(BYTE cardNo, WORD axis, WORD nType)

功能： 讀取軸之輸入點狀態。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 X 或 Y(1 or 2)
nType: 0 → DRIVING (檢查 i8092 有沒有輸出脈波)
1 → LIMIT+ (檢查有沒有碰觸前極限)
2 → LIMIT- (檢查有沒有碰觸後極限)
3 → EMERGENCY (檢查緊急停止信號)
4 → ALARM (檢查警報信號)
5 → HOME (檢查原點信號)
6 → NEAR HOME (檢查近原點信號)
7 → IN3 (檢查 IN3 信號)
8 → INPOS (檢查 INPOS 信號)
9 → Z PHASE (檢查 INPOS 信號)

回應： YES: on
NO: off

範例：

```
if (i8092MF_GET_DI(1, AXIS_X, 1) == YES)
{
    //讀取第 1 卡 X 軸，前極限信號處理。
}
```

● **WORD i8092MF_GET_DI_ALL(BYTE cardNo, WORD axis)**

功能: 讀取軸所有之輸入點狀態。

參數: **cardNo:** 指定卡號
axis: 指定軸號碼 X 或 Y(1 or 2)

回應: — 16 bits 值 (0=Low,1=High)

bit 0	NHOME 信號
bit 1	HOME 信號
bit 2	Z-PHASE 信號
bit 3	EMG 信號(Only AXIS_X)
bit 4	EXP+ 信號
bit 5	EXP- 信號
bit 6	READY(INPOS) 信號
bit 7	ALARM 信號
bit 8	N/A
bit 9	N/A
bit 10	N/A
bit 11	IN3 信號
bit 12	N/A
bit 13	N/A
bit 14	LMT+ 信號
bit 15	LMT- 信號

範例: **WORD DI_Flag=i8092MF_GET_DI_ALL(1, AXIS_X);**
//讀取第 1 卡 X 軸 DI 信號。

3.6 讀取目前 ERROR 狀態

● **BYTE i8092MF_GET_ERROR(BYTE cardNo)**

功能： 讀取軸運動有無錯誤發生。

參數： **cardNo:** 指定卡號

回應： **YES:** 有錯誤發生(欲讀錯誤碼請搭配使用 **i8092MF_GET_ERROR_CODE**)
NO: 沒有錯誤

範例： `if (i8092MF_GET_ERROR(1) == YES)`
`{`
`//讀取第 1 卡，錯誤停止處理。`
`WORD ErrorCode_X = i8092MF_GET_ERROR_CODE(1, AXIS_X);`
`WORD ErrorCode_Y = i8092MF_GET_ERROR_CODE(1, AXIS_Y);`
`if ((ErrorCode_X & ErrorCode_Y) == 256)`
`{`
`//表示使用了 6.5.4 功能，使軸運動停止，請故障排除後，清除停止狀態。`
`i8092MF_CLEAR_STOP(1);`
`}`
`}`

● **WORD i8092MF_GET_ERROR_CODE(BYTE cardNo, WORD axis)**

功能： 讀取各軸之錯誤碼。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 X 或 Y(1 or 2)

回應： **0:** 沒有任何錯誤
 非零值請參考下表,如同時有多個錯誤,會回傳其所有錯誤之錯誤碼總和。

錯誤碼	原因	說明
1	SOFT LIMIT+	碰觸軟體前極限
2	SOFT LIMIT-	碰觸軟體後極限
4	LIMIT+	碰觸前極限
8	LIMIT-	碰觸後極限
16	ALARM	伺服警報
32	EMERGENCY	緊急停止
64	N/A	N/A
128	N/A	N/A
256	請參考 6.5.4	軟體緊急停止

例如 48 表示"伺服警報"及"緊急停止" 同時發生

範例: `if (i8092MF_GET_ERROR_CODE(1, AXIS_X) == 10)`
`{`
`//讀取第 1 卡 X 軸，碰觸軟體後極限(2)及碰觸後極限(8) 的錯誤處理。`
`//軟體後極限(SOFT LIMIT-)及後極限(LIMIT-)設同一點。`
`}`

3.7 一般 DO 輸出

● `void i8092MF_SET_OUT0(BYTE cardNo, WORD axis, WORD nLevel)`

功能: 設定軸之驅動器異常(ALARM)輸入參數。

參數: `cardNo:` 指定卡號
`axis:` 指定軸號碼(參考表 2-1)
`nLevel:` DO 輸出: 0=OFF,1=ON

回應: 無

範例: `i8092MF_SET_OUT0 (1, AXIS_XY, 1);`
`//設定第 1 卡 X Y 軸，DO 輸出為 ON。`

4 FRnet 功能(i8092F 專用函式)

4.1 FRnet DI 讀取

● **WORD i8092MF_FRNET_IN**(BYTE *cardNo*, WORD *wRA*)

功能： 讀取 FRnet 的數位輸入資料。

參數： **cardNo:** 指定卡號
wRA: 群組範圍(8~15)RA8~RA15

回應： **WORD** 16-位元輸入資料

範例： **WORD IN_Data;**
IN_Data = i8092MF_FRNET_IN(1, 8);
//設定第 1 卡，RA 群組 = 8。

4.2 FRnet DO 寫入

● **void i8092MF_FRNET_OUT**(BYTE *cardNo*, WORD *wSA*, WORD *data*)

功能： 寫入 FRnet 的數位輸出資料。

參數： **cardNo:** 指定卡號
wSA: 群組範圍(0~7)SA0~SA7
dara: 16-位元資料

回應： 無

範例： **i8092MF_FRNET_OUT(1, 0, 0xffff);**
//設定第 1 卡，SA 群組 = 0，16 位元資料為 0xffff。

5 軸自動歸零

i8092 提供軟體自動歸零功能，只要經適當設定後，即可下指令自動執行，主要步驟如下：

- 以高速尋找近原點開關
- 以低速尋找原點開關
- 以低速尋找伺服馬達 Z 相信號
- 以高速運動到補正值(Offset)位置(程式原點)

設定時，其中步驟可以選擇不執行，以符合客戶實際需求動作，執行時完全自動執行，簡化程式設計。

i8092 亦提供軟體單步驟歸零功能

- 以指定速度尋找近原點開關
- 以指定速度原點開關
- 以指定速度伺服馬達 Z 相信號

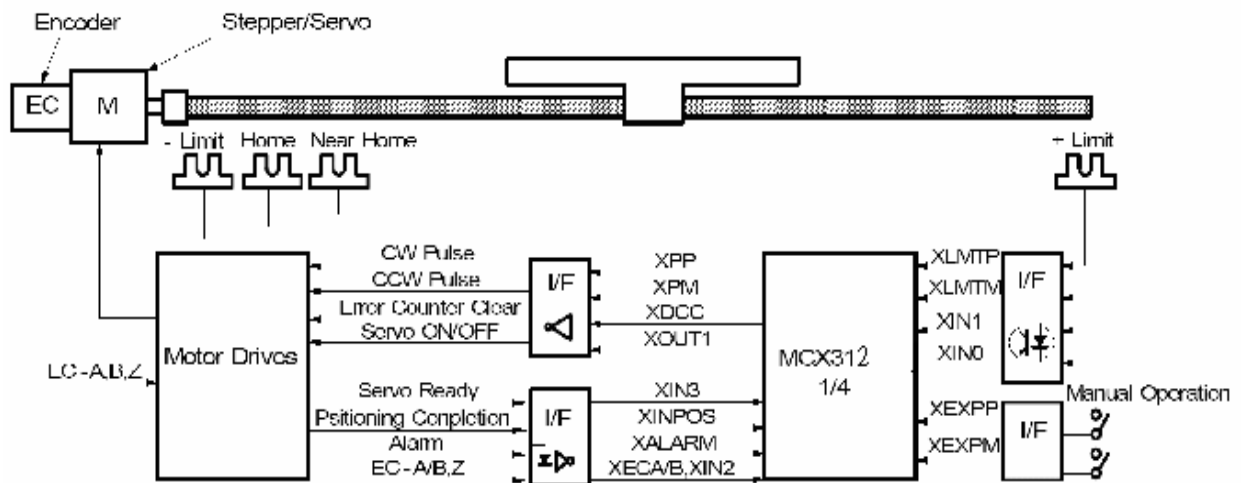
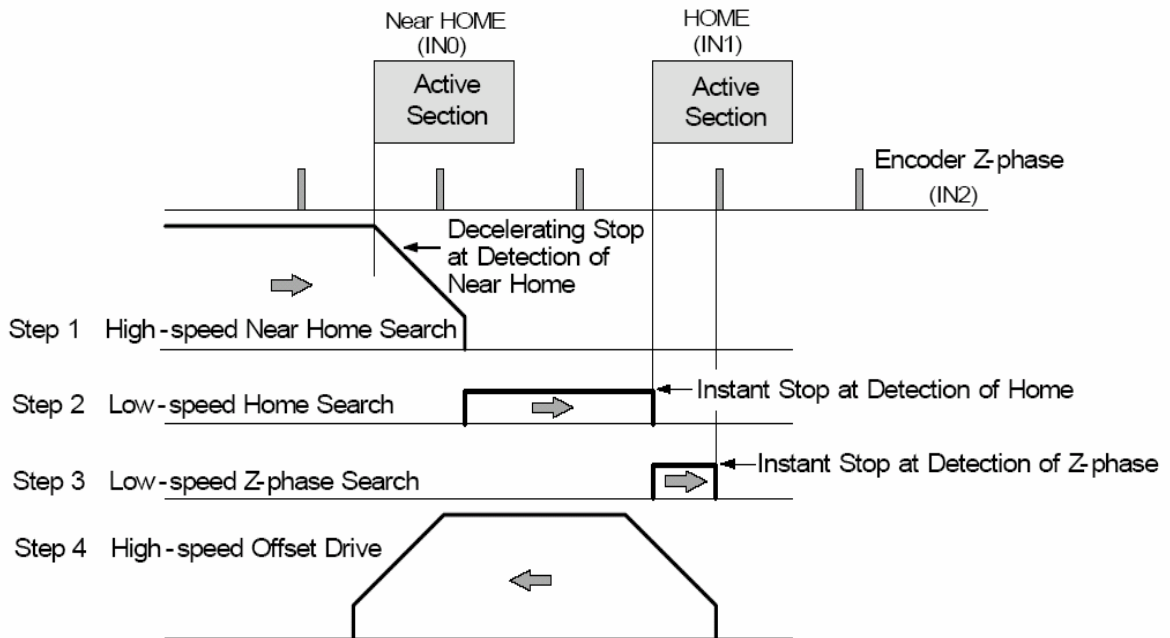
請參考其他相關指令：

- 2.6 設定軸前後極限的輸入觸發邏輯
- 2.8 設定軸近原點(NHOME)輸入觸發邏輯
- 2.9 設定軸原點(HOME)輸入觸發邏輯
- 2.10 設定軸前後軟體極限,參考來源及取消

5.1 軟體自動歸零功能

歸原點常常使用，當機器開機時，或系統發生警報，或信號錯誤時。上述情況使用者都能使用歸原點，讓機器回到原先的工作點。

i8092 提供了一連串自動歸原點的功能，例如高速尋找近原點 → 低速歸原點 → 編碼器 Z-相尋找 → 不是 CPU 插入的補正驅動，使用者處理的狀況應該和下圖所示類似。這個範例是單軸驅動系統，兩軸也能以相同方法處理。



● BYTE i8092MF_AUTO_HOME(BYTE cardNo, WORD axis, WORD nStep1, WORD nStep2, WORD nStep3, WORD nStep4 , long IOffset,DWORD dwV,DWORD dwHV,long IBack)

功能： 軟體自動歸零功能。(軟體功能的巨集函式，會耗用系統資源)

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 AXIS_X,AXIS_Y (參考表 2-1)
nStep1: 設定 0=不執行,1=朝正向尋找,2=朝負向尋找
nStep2: 設定 0=不執行,1=朝正向尋找,2=朝負向尋找
nStep3: 設定 0=不執行,1=朝正向尋找,2=朝負向尋找
nStep4: 設定 0=不執行,1=執行
IOffset: 補正值(- 2,147,483,647~+ 2,147,483,647)
dwV: 驅動速度
dwHV: 歸零速度
IBack: 異常反向移出 Pulse 數(正好在 Senser 上時或先碰到極限開關後返回碰觸 Senser 後)

自動歸零步驟(Homing Step)

步驟	動作	運動速度	開關
步驟 1	以高速尋找近原點開關	驅動速度 (dwV)	近原點(ST0)
步驟 2	以歸零速度尋找原點開關	歸 零 速 度 (dwHV)	原點(ST1)
步驟 3	以歸零速度尋找伺服馬達 Z 相信號	歸 零 速 度 (dwHV)	Z 相信號(ST2)
步驟 4	以高速運動到位移值	驅動速度 (dwV)	

回應： 0: 正常結束。
 4: 伺服 ALARM 停止。
 8: 緊急停止。
 253: 步驟 3 如果壓在 Z-PHASE(與 HOME 同時 ON)錯誤。
 254: 找不到 Senser(NHOME,HOME)。
 255: 指定軸錯誤。

範例： BYTE m_Axis=AXIS_X;
 i8092MF_SET_MAX_V(cardNo, m_Axis, 16000);
 i8092MF_SET_A(cardNo, m_Axis,50000); //set Acc =50000 PPS/S
 i8092MF_SET_SV(cardNo, m_Axis, 1000);
 BYTE RR=i8092MF_AUTO_HOME(cardNo, m_Axis, 2, 2, 0, 1,

```

10000,10000,2000,6000);
Sleep(100);
if (RR==0)
{
    i8092MF_SET_LP(cardNo,m_Axis,0);
    i8092MF_SET_EP(cardNo,m_Axis,0);
}
//設定第 1 卡 AXIS_X 以下表為執行範例:

```

	輸入信號	尋找方向	尋找速度
步驟 1	近原點 (ST0) Low active	—	10000 (PPS) (V)
步驟 2	原點 (ST1) Low active	—	2000 (PPS) (HV)
步驟 3	Z相信號 (ST2) High active	+	2000 (PPS) (HV)
步驟 4	+10000 pulse 補正(offset)		10000 (PPS) (V)

5.2 軟體單步驟歸零功能

5.2.1 尋找近原點開關

● BYTE i8092MF_SEARCH_NHOME(BYTE cardNo, WORD axis, WORD nDir, DWORD v,long IBack)

功能: 以指定速度尋找近原點開關。
(軟體功能的巨集函式，會耗用系統資源)

參數: **cardNo:** 指定卡號
axis: 指定軸號碼 AXIS_X,AXIS_Y (參考表 2-1)
nDir: 設定 0=朝正向尋找,1=朝負向尋找
v: 驅動速度
dwHV: 歸零速度
IBack: 異常反向移出 Pulse 數
(正好在 Senser 上時或先碰到極限開關後返回碰觸 Senser 後)

回應: 0: 正常結束。
4: 伺服 ALARM 停止。
8: 緊急停止。
254: 找不到 Senser(NHOME,HOME)。
255: 指定軸錯誤。

範例:

```
BYTE m_Axis=AXIS_X;
i8092MF_SET_MAX_V(cardNo, m_Axis, 16000);
i8092MF_SET_A(cardNo, m_Axis,50000); //set Acc =50000 PPS/S
i8092MF_SET_SV(cardNo, m_Axis, 1000);
BYTE RR=i8092MF_SEARCH_NHOME(cardNo, m_Axis, 1, 2000,5000);
Sleep(100);
if (RR==0)
{
    i8092MF_SET_LP(cardNo,m_Axis,0);
    i8092MF_SET_EP(cardNo,m_Axis,0);
}
//設定第 1 卡 AXIS_X 執行範例:
```

5.2.2 尋找原點開關

● **BYTE i8092MF_SEARCH_HOME(BYTE cardNo, WORD axis, WORD nDir, DWORD v,long IBack)**

功能: 以指定速度尋找原點開關。
(軟體功能的巨集函式，會耗用系統資源)

參數:

cardNo:	指定卡號
axis:	指定軸號碼 AXIS_X,AXIS_Y (參考表 2-1)
nDir:	設定 0=朝正向尋找,1=朝負向尋找
v:	驅動速度
dwHV:	歸零速度
IBack:	異常反向移出 Pulse 數 (正好在 Senser 上時或先碰到極限開關後返回碰觸 Senser 後)

回應:

0:	正常結束。
4:	伺服 ALARM 停止。
8:	緊急停止。
254:	找不到 Senser(NHOME,HOME) 。
255:	指定軸錯誤。

範例:

```
BYTE m_Axis=AXIS_X;
i8092MF_SET_MAX_V(cardNo, m_Axis, 16000);
i8092MF_SET_A(cardNo, m_Axis,50000); //set Acc =50000 PPS/S
i8092MF_SET_SV(cardNo, m_Axis, 1000);
BYTE RR=i8092MF_SEARCH_HOME(cardNo, m_Axis, 1, 2000,5000);
Sleep(100);
if (RR==0)
{
    i8092MF_SET_LP(cardNo,m_Axis,0);
    i8092MF_SET_EP(cardNo,m_Axis,0);
}
//設定第 1 卡 AXIS_X 執行範例:
```

5.2.3 尋找伺服馬達 Z 相信號

● BYTE i8092MF_SEARCH_ZPHASE(BYTE cardNo, WORD axis, WORD nDir, DWORD v)

功能： 指定速度尋找伺服馬達 Z 相信號。
(軟體功能的巨集函式，會耗用系統資源)

參數：
cardNo: 指定卡號
axis: 指定軸號碼 AXIS_X,AXIS_Y (參考表 2-1)
nDir: 設定 0=朝正向尋找,1=朝負向尋找
v: 驅動速度

回應：
0: 正常結束。
4: 伺服 ALARM 停止。
8: 緊急停止。
253: 如果開始時已壓在 Z-PHASE 上。
255: 指定軸錯誤。

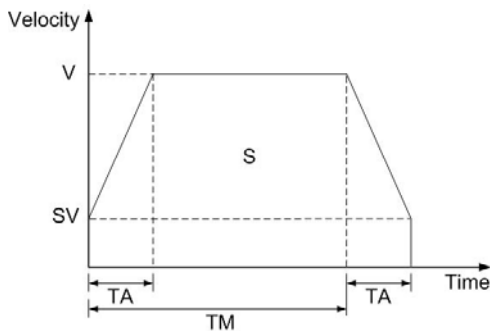
範例：
BYTE m_Axis=AXIS_X;
i8092MF_SET_MAX_V(cardNo, m_Axis, 16000);
i8092MF_SET_A(cardNo, m_Axis,50000); //set Acc =50000 PPS/S
i8092MF_SET_SV(cardNo, m_Axis, 1000);
BYTE RR=i8092MF_SEARCH_ZPHASE(cardNo, m_Axis, 1, 1000);
Sleep(100);
if (RR==0)
{
 i8092MF_SET_LP(cardNo,m_Axis,0);
 i8092MF_SET_EP(cardNo,m_Axis,0);
}
//設定第 1 卡 AXIS_X 執行範例:

6 軸控功能

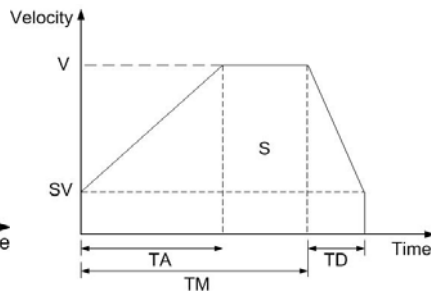
6.1 各軸獨立運動

- 單軸運動中，各軸可在任一時間同時運動。
- 各軸下完指令後，完全獨立運作不會互相干擾。
- 可單獨對每一軸下獨立指令，多工運動(各軸不補間)。
- 在運動執行中，我們可以動態改變參數值，包含位移脈波數、速度....等等。
- 也可以中途令其減速停止或立即停止...，以順應我們對運動控制不同的需求。

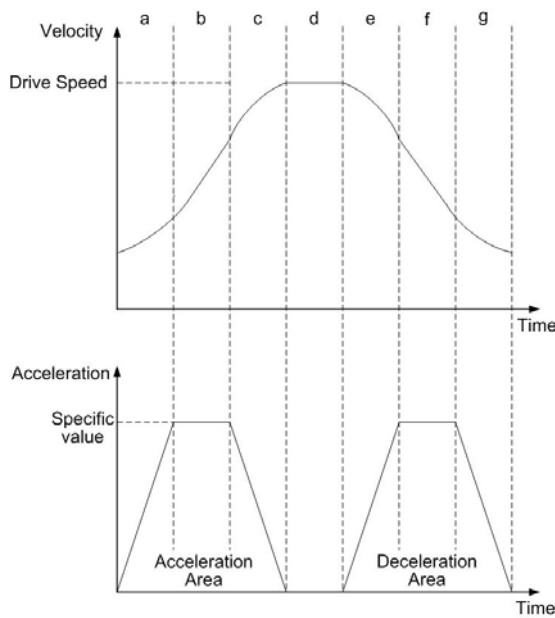
6.1.1 設定單軸加減速模式



對稱 T-曲線加減速



對稱 T-曲線加減速 (A<D)



S-曲線加減速驅動

● **void i8092MF_NORMAL_SPEED(BYTE cardNo, WORD axis , WORD nMode)**

功能： 設定速度模式。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 (參考表 2-1)
nMode: 0 → 對稱 T 曲線 (SV、V、A、AO)
1 → 對稱 S 曲線 (SV、V、K、AO)
2 → 非對稱 T 曲線 (SV、V、A、D、AO)

回應： 無

範例： **BYTE cardNo=1; //設定第 1 號卡。**
i8092MF_SET_MAX_V(cardNo, AXIS_XY, 20000); //設定 XY 軸最高速 20K PPS。

```
//=====
i8092MF_NORMAL_SPEED(cardNo, AXIS_XY, 0); //設定 XY 軸為對稱 T 曲線。
i8092MF_SET_V(cardNo, AXIS_XY, 2000); //設定 XY 軸速度=2000 PPS。
i8092MF_SET_A(cardNo, AXIS_XY,1000); //設定 XY 軸加速度=1000 PPS/S。
i8092MF_SET_SV(cardNo, AXIS_XY, 2000); //設定 XY 軸初始速度=2000 PPS。
i8092MF_SET_AO(cardNo, AXIS_XY, 9); //設定 XY 軸減速(保留脈波數)= 9 PPS。
i8092MF_FIXED_MOVE(cardNo, AXIS_XY, 10000); //XY 軸移動 10000 Pulse。
```

```
//=====
i8092MF_NORMAL_SPEED(cardNo, AXIS_XY,1); //設定 XY 軸為對稱 S 曲線。
i8092MF_SET_V(cardNo, AXIS_XY, 2000); //設定 XY 軸速度=2000 PPS。
i8092MF_SET_K(cardNo, AXIS_XY, 3000); // XY 軸加速度變化率=3000 PPS/S。
i8092MF_SET_SV(cardNo, AXIS_XY, 200); //設定 XY 軸初始速度=200 PPS。
i8092MF_SET_AO(cardNo, AXIS_XY, 9); //設定 XY 軸減速(保留脈波數)= 9 PPS。
i8092MF_FIXED_MOVE(cardNo, AXIS_XY, -10000); //XY 軸移動-10000 Pulse。
```

```
//=====
i8092MF_NORMAL_SPEED(cardNo, AXIS_XY,2); //設定 XY 軸為非對稱 T 曲線。
i8092MF_SET_V(cardNo, AXIS_XY, 2000); //設定 XY 軸速度=2000 PPS。
i8092MF_SET_A(cardNo, AXIS_XY,1000 ); //設定 XY 軸加速度=1000 PPS/S。
i8092MF_SET_D(cardNo, AXIS_XY, 500); //設定 XY 軸減速度=500 PPS/S。
i8092MF_SET_SV(cardNo, AXIS_XY, 200); //設定 XY 軸初始速度=200 PPS。
i8092MF_SET_AO(cardNo, AXIS_XY, 9); //設定 XY 軸減速(保留脈波數)= 9 PPS。
i8092MF_FIXED_MOVE(cardNo, axis, 10000); //執行 XY 軸移動 10000 Pulse。
```

備註： 請搭配設定相關速度參數.....。

6.1.2 設定軸初始速度

- **void i8092MF_SET_SV**(BYTE *cardNo*, WORD *axis*, DWORD *data*)

功能： 設定軸之初始速度。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)
data: 設定速度值 (最大值請參考 3.1.3) PPS

回應： 無

範例：
`i8092MF_SET_SV(1, AXIS_X, 1000);`
`//設定第 1 卡 X 軸，初始速度為 1000 PPS。`

6.1.3 設定軸定速度

- **void i8092MF_SET_V**(BYTE *cardNo*, WORD *axis*, DWORD *data*)

功能： 設定軸之定速度。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)
data: 設定速度值 (最大值請參考 3.1.3) PPS

回應： 無

範例：
`i8092MF_SET_V(1, AXIS_X, 120000L);`
`//設定第 1 卡 X 軸，定速度為 120000 PPS。`

6.1.4 設定軸加速度

● **void i8092MF_SET_A**(BYTE *cardNo*, WORD *axis*, DWORD *data*)

功能： 設定軸之加速度。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)
data: 設定加速度值 (PPS/Sec)

回應： 無

範例：
i8092MF_SET_A (1, AXIS_X, 100000L);
//設定第 1 卡 X 軸，加速度為 100K PPS/Sec。

6.1.5 設定軸減速度

● **void i8092MF_SET_D**(BYTE *cardNo*, WORD *axis*, DWORD *data*)

功能： 設定軸之減速度。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)
data: 設定減速度值 (PPS/Sec)

回應： 無

範例：
i8092MF_SET_D(1, AXIS_X, 100000L);
//設定第 1 卡 X 軸，減速度為 100K PPS/Sec。

6.1.6 設定軸加速度變化率

- **void i8092MF_SET_K**(BYTE *cardNo*, WORD *axis*, DWORD *data*)

功能： 設定軸之輸出加速度變化率。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)
data: 設定加速度變化率值 (Jerk PPS/ Sec²)

回應： 無

範例：
`i8092MF_SET_K(1, AXIS_X, 500000L);`
//設定第 1 卡 X 軸，加速度變化率為 500K PPS/ Sec²。

6.1.7 設定軸減速(保留脈波數)

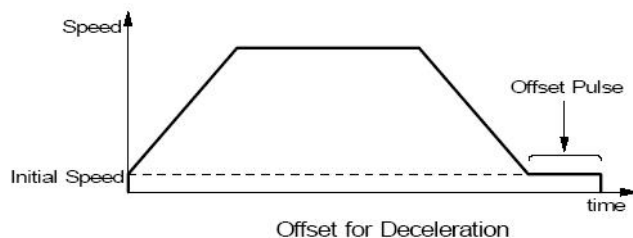
- **void i8092MF_SET_AO**(BYTE *cardNo*, WORD *axis*, short int *data*)

功能： 於固定脈波數運動控制時，至目標前保留低速輸出 Offset Pulse 數，如圖所示 Offset Pulse 位置。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)
data: 設定 Offset Pulse 值 (-32,768 ~ +32,767)

回應： 無

範例：
`i8092MF_SET_AO(1, AXIS_X, 200);`
//設定第 1 卡 X 軸，Offset Pulse 為 200 Pulses。



6.1.8 固定脈波數輸出

● BYTE i8092MF_FIXED_MOVE(BYTE *cardNo*, WORD *axis*, long *data*)

功能： 執行單軸固定步數輸出。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)
data: 指定步數(-268,435,455 ~ + 268,435,455)

回應： YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8092MF_GET_ERROR_CODE)
NO: 沒有錯誤

範例：
BYTE *cardNo*=1; //設定第 1 號卡。
i8092MF_SET_MAX_V(*cardNo*, AXIS_XY, 20000); //設定 XY 軸最高速 20K PPS。
i8092MF_NORMAL_SPEED(*cardNo*, AXIS_XY, 0); //設定 XY 軸為對稱 T 曲線。
i8092MF_SET_V(*cardNo*, AXIS_XY, 2000); //設定 XY 軸速度=2000 PPS。
i8092MF_SET_A(*cardNo*, AXIS_XY,1000); //設定 XY 軸加速度=1000 PPS/S。
i8092MF_SET_SV(*cardNo*, AXIS_XY, 2000); //設定 XY 軸初始速度=2000 PPS。
i8092MF_SET_AO(*cardNo*, AXIS_XY, 9); //設定 XY 軸減速(保留脈波數)= 9 PPS。
i8092MF_FIXED_MOVE(*cardNo*, AXIS_XY, 10000); //XY 軸移動 10000 Pulse。

6.1.9 連續脈波輸出

● BYTE i8092MF_CONTIUNE_MOVE(BYTE *cardNo*, WORD *axis*, long *data*)

功能： 執行單軸連續脈波輸出。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)
data: 指定速度: -V(CCW) ~ +V(CW) PPS, V=Vmin~Vmax

回應： YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8092MF_GET_ERROR_CODE)
NO: 沒有錯誤

範例：
BYTE *cardNo*=1; //設定第 1 號卡。
i8092MF_SET_MAX_V(*cardNo*, AXIS_XY, 20000); //設定 XY 軸最高速 20K PPS。
i8092MF_NORMAL_SPEED(*cardNo*, AXIS_XY, 0); //設定 XY 軸為對稱 T 曲線。
i8092MF_SET_V(*cardNo*, AXIS_XY, 2000); //設定 XY 軸速度=2000 PPS。
i8092MF_SET_A(*cardNo*, AXIS_XY,1000); //設定 XY 軸加速度=1000 PPS/S。
i8092MF_SET_SV(*cardNo*, AXIS_XY, 2000); //設定 XY 軸初始速度=2000 PPS。
i8092MF_CONTIUNE_MOVE(*cardNo*, AXIS_XY, 1000); //1000 PPS 速度連續移動。

6.2 補間運動

由於 i8092/F 為兩軸模組,因此其補間第一軸恆為 AXIS_X 軸,而第二軸就是 AXIS_Y 軸因此其位置參數,固定為 X(FP1,CP1),後 Y(FP2,CP2)請您在使用時請注意。

6.2.1 設定補間加減速模式

● void i8092MF_VECTOR_SPEED(BYTE cardNo, WORD nMode)

功能: 設定向量加減速模式。

參數: **cardNo:** 指定卡號
nMode: 0 → 二軸(直線&弧&圓)固定向量速度 (VV)
1 → 二軸直線對稱 T 曲線 (VSV、VV、VA、VAO)
2 → 二軸直線對稱 S 曲線 (VSV、VV、VK、VAO)
3 → 二軸直線非對稱 T 曲線 (VSV、VV、VA、VD、VAO)
4 → N/A
5 → 二軸(弧&圓)對稱 T 曲線 (VSV、VV、VA、VAO)
6 → 二軸(弧&圓)非對稱 T 曲線 (VSV、VV、VA、VD、VAO)

回應: 無

範例: **BYTE cardNo=1; //設定第 1 號卡。**
i8092MF_SET_MAX_V(cardNo, AXIS_XY, 20000);
//設定 XY 軸最高速 20K PPS
//=====
i8092MF_VECTOR_SPEED(cardNo, 0);
//二軸(直線&弧&圓)固定向量速度 VSV=VV
i8092MF_SET_VSV(cardNo, 1000); //設定第 1 卡, 向量初始速度為 1000 PPS。
i8092MF_SET_VV(cardNo, 1000); //設定第 1 卡, 向量定速度為 1000 PPS。
i8092MF_LINE_2D(1, 12000, 10000); //執行向量 2D 補間。

//=====
BYTE axis=AXIS_XY; //for AXIS_XY
i8092MF_SET_MAX_V(cardNo, axis, 20000);
i8092MF_VECTOR_SPEED(cardNo, 1);
//二軸直線對稱 T 曲線 (VSV、VV、VA、VAO)。
i8092MF_SET_VSV(cardNo, 500); //設定向量初始速度為 500 PPS。
i8092MF_SET_VV(cardNo, 2000); //設定向量速度為 2000 PPS。
i8092MF_SET_VA(cardNo, 1000); //設定向量加速速度為 1000 PPS。
i8092MF_LINE_2D(cardNo, 20000, 10000); //執行向量 2D 補間。
//=====
BYTE axis=AXIS_XY; //for AXIS_XY
i8092MF_SET_MAX_V(cardNo, axis, 30000);
i8092MF_VECTOR_SPEED(cardNo, 2);

```

//二軸直線對稱 S 曲線 (VSV、VV、VA、VK、AO)。
i8092MF_SET_VSV(cardNo, 200); //設定向量初始速度為 500 PPS。
i8092MF_SET_VV(cardNo, 2000); //設定向量速度為 2000 PPS。
i8092MF_SET_VK(cardNo, 12500); //設定向量加速度變化率為 3000 PPS/s^2。
i8092MF_SET_VAO(cardNo, 20); //設定向量減速(保留脈波數)20 Pulse。
i8092MF_LINE_2D(cardNo, 10000, 10000); //執行向量 2D 補間。

```

```

//=====
BYTE axis=AXIS_XY; //for AXIS_XY
i8092MF_SET_MAX_V(cardNo, axis, 20000);
i8092MF_VECTOR_SPEED(cardNo, 3);
//二軸直線非對稱 T 曲線 (VSV、VV、VA、VD、VAO)。
i8092MF_SET_VSV(cardNo, 100); //設定向量初始速度為 500 PPS。
i8092MF_SET_VV(cardNo, 2000); //設定向量速度為 2000 PPS。
i8092MF_SET_VA(cardNo, 1000); //設定向量加速度為 1000 PPS/s。
i8092MF_SET_VD(cardNo, 500); //設定向量減速度為 500 PPS/s。
i8092MF_SET_VAO(cardNo, 20); //設定向量減速(保留脈波數)20 Pulse。
i8092MF_LINE_2D(cardNo, 10000, 5000); //執行向量 2D 補間。

```

```

//=====
long fp1=11000;
long fp2=9000;
long c1=10000;
long c2=0;
int sv=100;
int v=3000;
int a=5000;
int d=5000;
i8092_SET_MAX_V(cardNo, AXIS_XY, 8000);
i8092_VECTOR_SPEED(cardNo, 5);
//二軸(弧&圓)對稱 T 曲線 (VSV、VV、VA、VAO)。
i8092_SET_VSV(cardNo, sv); //設定向量初始速度為 sv PPS。
i8092_SET_VV(cardNo, v); //設定向量速度為 v PPS。
i8092_SET_VA(cardNo, a); //設定向量加速度為 a PPS/s。
i8092_SET_VAO(cardNo, 0); //設定向量減速(保留脈波數)0 Pulse。
i8092_ARC_CW(cardNo, c1,c2, fp1, fp2); //執行二軸順時針圓弧補間。

```

```

//=====
long c1=300;
long c2=0;
int sv=100;
int v=3000;
int a=125;
int d=12;
i8092_SET_MAX_V(cardNo, AXIS_XY, 8000);

```

```
i8092_VECTOR_SPEED(cardNo, 6);  
//二軸(弧&圓)對稱T曲線 (VSV、VV、VA、VAO)。  
i8092_SET_VSV(cardNo, sv); //設定向量初始速度為 sv PPS。  
i8092_SET_VV(cardNo, v); //設定向量速度為 v PPS。  
i8092_SET_VA(cardNo, a); //設定向量加速度為 a PPS/s。  
i8092_SET_VD(cardNo, d); //設定向量減速度為 d PPS/s。  
i8092_SET_VAO(cardNo, 0); //設定向量減速(保留脈波數)0 Pulse。  
i8092_CIRCLE_CW(cardNo, c1, c2); //執行二軸順時針圓形補間。
```

備註: 請搭配設定相關向量速度參數.....。

6.2.2 設定軸向量初始速度

- **void i8092MF_SET_VSV(BYTE cardNo, DWORD data)**

功能： 設定軸之向量初始速度。

參數： **cardNo:** 指定卡號
data: 設定向量速度值 (最大值請參考 3.1.3) PPS

回應： 無

範例： **i8092MF_SET_VSV(1, 1000);**
//設定第 1 卡，向量初始速度為 1000 PPS。

6.2.3 設定軸向量定速度

- **void i8092MF_SET_VV(BYTE cardNo, DWORD data)**

功能： 設定軸之向量定速度。

參數： **cardNo:** 指定卡號
data: 設定向量速度值 (最大值請參考 3.1.3) PPS

回應： 無

範例： **i8092MF_SET_VV(1, 120000L);**
//設定第 1 卡，向量定速度為 120000 PPS。

6.2.4 設定軸向量加速度

- **void i8092MF_SET_VA**(BYTE *cardNo*, DWORD *data*)

功能： 設定軸之向量加速度。

參數： **cardNo:** 指定卡號
data: 設定向量加速度值 (PPS/Sec)

回應： 無

範例： **i8092MF_SET_VA** (1, 100000L);
//設定第 1 卡，向量加速度為 100K PPS/Sec。

6.2.5 設定軸向量減速度

- **void i8092MF_SET_VD**(BYTE *cardNo*, DWORD *data*)

功能： 設定軸之向量減速度。

參數： **cardNo:** 指定卡號
data: 設定向量減速度值 (PPS/Sec)

回應： 無

範例： **i8092MF_SET_VD**(1, 100000L);
//設定第 1 卡，向量減速度為 100K PPS/Sec。

6.2.6 設定軸向量加速度變化率

- **void i8092MF_SET_VK(BYTE cardNo, DWORD data)**

功能： 設定軸之輸出向量加速度變化率。

參數： **cardNo:** 指定卡號
data: 設定向量加速度變化率值 (Jerk PPS/ Sec²)

回應： 無

範例： **i8092MF_SET_VK(1, 500000L);**
//設定第 1 卡，向量加速度變化率為 500K PPS/ Sec²。

6.2.7 設定軸向量減速(保留脈波數)

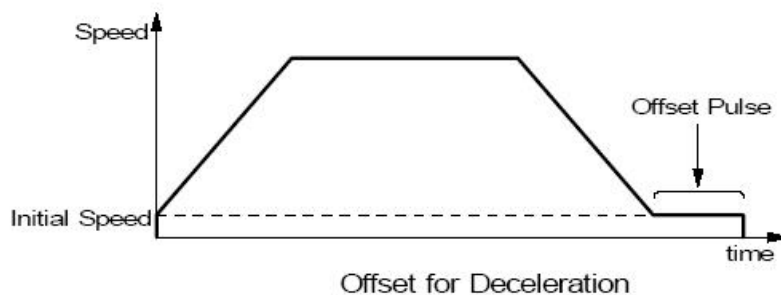
- **void i8092MF_SET_VAO(BYTE cardNo, short int data)**

功能： 於固定脈波數運動控制時，至目標前保留低速輸出 Offset Pulse 數，如圖所示 Offset Pulse 位置。

參數： **cardNo:** 指定卡號
data: 設定 Offset Pulse 值 (-32,768 ~ +32,767)

回應： 無

範例： **i8092MF_SET_VAO(1, 200);**
//設定第 1 卡補間軸，Offset Pulse 為 200 Pulses。



6.2.8 二軸直線補間

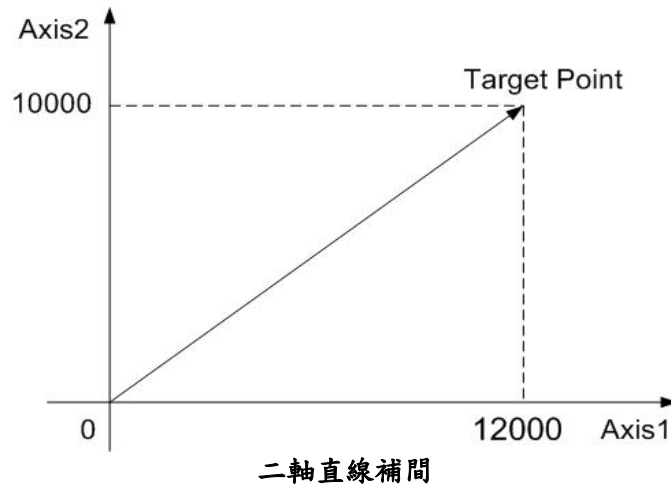
- **BYTE i8092MF_LINE_2D**(BYTE *cardNo*, long *fp1*, long *fp2*)

功能： 執行二軸直線補間。

參數：
cardNo: 指定卡號
fp1: 指定 X 軸 Pulse 數(-8,388,607 ~ +8,388,607)
fp2: 指定 Y 軸 Pulse 數(-8,388,607 ~ +8,388,607)

回應：
YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8092MF_GET_ERROR_CODE)
NO: 沒有錯誤

範例：
`i8092MF_LINE_2D(1, 12000, 10000);`
//設定第 1 卡，執行二軸直線補間。



6.2.9 二軸圓弧補間

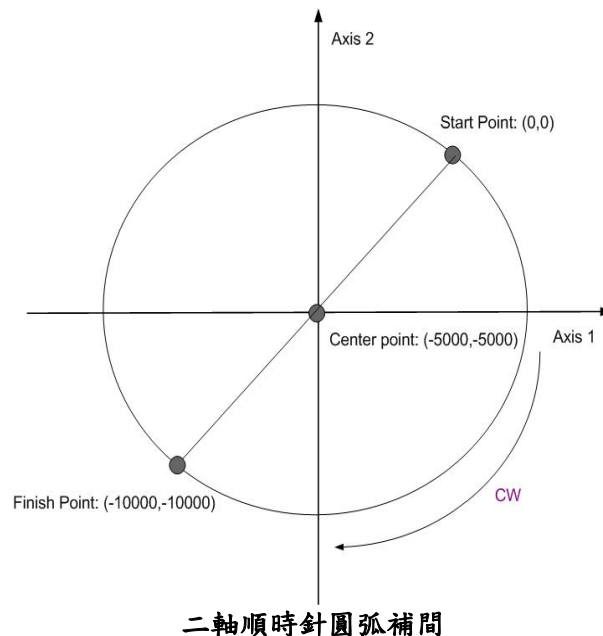
● BYTE i8092MF_ARC_CW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2)

功能： 執行二軸順時針圓弧補間。

參數：
cardNo: 指定卡號
cp1: 指定 X 軸圓弧中心相對位置
(-8,388,607 ~ +8,388,607)
cp2: 指定 Y 軸圓弧中心相對位置
(-8,388,607 ~ +8,388,607)
fp1: 指定 X 軸圓弧終點相對位置
(-8,388,607 ~ +8,388,607)
fp2: 指定 Y 軸圓弧終點相對位置
(-8,388,607 ~ +8,388,607)

回應：
YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8092MF_GET_ERROR_CODE)
NO: 沒有錯誤

範例：
i8092MF_ARC_CW(1, -5000, -5000, -10000, -10000);
//設定第 1 卡，執行二軸順時針圓弧補間。



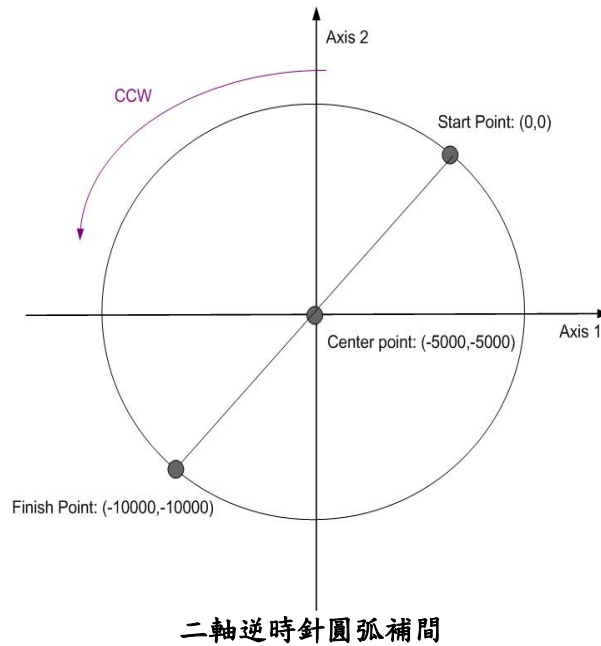
● **BYTE i8092MF_ARC_CCW**(BYTE *cardNo*, long *cp1*, long *cp2*, long *fp1*, long *fp2*)

功能： 執行二軸逆時針圓弧補間。

參數：
cardNo: 指定卡號
cp1: 指定 X 軸圓弧中心相對位置
(-8,388,607 ~ +8,388,607)
cp2: 指定 Y 軸圓弧中心相對位置
(-8,388,607 ~ +8,388,607)
fp1: 指定 X 軸圓弧終點相對位置
(-8,388,607 ~ +8,388,607)
fp2: 指定 Y 軸圓弧終點相對位置
(-8,388,607 ~ +8,388,607)

回應：
YES: 有錯誤發生(欲讀錯誤碼請搭配使用 **i8092MF_GET_ERROR_CODE**)
NO: 沒有錯誤

範例：
i8092MF_ARC_CCW(1, -5000, -5000, -10000, -10000);
//設定第 1 卡，執行二軸逆時針圓弧補間。



6.2.10 二軸圓形補間

- **BYTE i8092MF_CIRCLE_CW**(BYTE *cardNo*, long *cp1*, long *cp2*)

功能： 執行二軸順時針圓形補間。

參數：
cardNo: 指定卡號
cp1: 指定 X 軸圓弧中心相對位置
(-8,388,607 ~ +8,388,607)
cp2: 指定 Y 軸圓弧中心相對位置
(-8,388,607 ~ +8,388,607)

回應： YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8092MF_GET_ERROR_CODE)
NO: 沒有錯誤

範例： i8092MF_CIRCLE_CW(1, 0, 10000);
//設定第 1 卡，執行二軸順時針圓形補間。

- **BYTE i8092MF_CIRCLE_CCW**(BYTE *cardNo*, long *cp1*, long *cp2*)

功能： 執行二軸逆時針圓形補間。

參數：
cardNo: 指定卡號
cp1: 指定 X 軸圓弧中心相對位置
(-8,388,6078 ~ +8,388,6078)
cp2: 指定 Y 軸圓弧中心相對位置
(-8,388,6078 ~ +8,388,6078)

回應： YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8092MF_GET_ERROR_CODE)
NO: 沒有錯誤

範例： i8092MF_CIRCLE_CCW(1, 0, 10000);
//設定第 1 卡，執行二軸逆時針圓形補間。

6.3 連續補間運動

連續補間運動如被異常中斷而停止，請參考 6.5.5 排除！

6.3.1 二軸矩形連續補間

- BYTE i8092MF_RECTANGLE(

BYTE cardNo, WORD nAcc, WORD Sp, WORD nDir, long Lp, long Wp, long Rp, DWORD RSV, DWORD RV, DWORD RA, DWORD RD)

功能： 執行二軸矩形補間。

(軟體功能的巨集函式，會耗用系統資源)

參數： cardNo: 指定卡號

nAcc: 0 → 定速度補間

1 → 對稱 T 曲線加減速補間

Sp: 設定起點 0 ~ 7 (Sp0 ~ Sp7 如下圖所示)

nDir: 設定方向 0、1 (CCW or CW)

Lp: 設定長度 Pulse 數(1 ~ 8,388,607)

Wp: 設定寬度 Pulse 數(1 ~ 8,388,607)

Rp: 設定圓弧半徑 Pulse 數(1 ~ 8,388,607)

RSV: 設定矩形補間向量起始速度(PPS)

RV: 設定矩形補間向量速度(PPS)

RA: 設定矩形補間向量加速度(PPS/Sec)

RD: 設定矩形末段補間向量減速度(PPS/Sec)

回應： YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8092MF_GET_ERROR_CODE)
NO: 沒有錯誤

範例： BYTE cardNo=1; //設定第 1 號卡。

int sv=1000; //設定向量初始速度為 1000 PPS。

int v=10000; //設定向量速度為 10000 PPS。

int a=5000; //設定向量加速度為 5000 PPS/s。

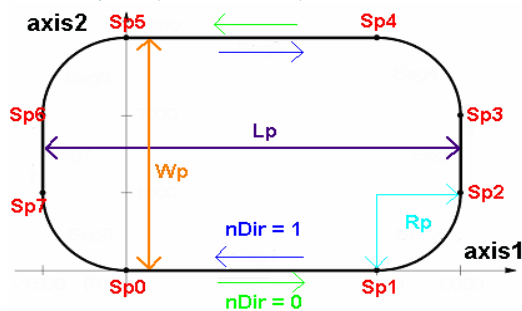
int d=5000; //設定向量減速度為 5000 PPS/s。

i8092MF_SET_MAX_V(cardNo, AXIS_XY, 16000); //最高速度為 16000 PPS。

i8092MF_RECTANGLE(

cardNo, 1, 0, 0, 20000, 10000, 1000, sv, v, a, d);

//設定第 1 卡，執行二軸矩形連續補間，減速點會自動運算。



6.3.2 二軸直線連續補間

- **void i8092MF_LINE_2D_INITIAL**(BYTE cardNo, DWORD VSV, DWORD VV, DWORD VA)

功能： 二軸直線連續補間初始設定(對稱 T 曲線加減速)。

參數：
cardNo: 指定卡號
VSV: 設定向量初始速度(PPS)
VV: 設定向量速度(PPS)
VA: 設定加速度(PPS/Sec)

回應： 無

範例： 和 **void i8092MF_LINE_2D_CONTINUE**(BYTE cardNo, WORD nType, long fp1, float fp2) 並用。

- **BYTE i8092MF_LINE_2D_CONTINUE**(BYTE cardNo, WORD nType, long fp1, long fp2)

功能： 執行二軸直線連續補間。
(軟體功能的巨集函式，會耗用系統資源)

參數：
cardNo: 指定卡號
nType: 0 → 二軸直線連續補間
1 → 二軸直線連續補間結束
fp1: 指定 X 軸 Pulse 數(-8,388,607 ~ +8,388,607)
fp2: 指定 Y 軸 Pulse 數(-8,388,607 ~ +8,388,607)

回應：
YES: 有錯誤發生(欲讀錯誤碼請搭配使用 **i8092MF_GET_ERROR_CODE**)
NO: 沒有錯誤

範例：
BYTE cardNo=1; //設定第 1 號卡。
int sv=300; //設定向量初始速度為 PPS。
int v=18000; //設定向量速度為 PPS。
long a=500000; //設定向量加速度為 PPS/s。
int loop1;
i8092MF_SET_MAX_V(cardNo, AXIS_XY,160000L);
i8092MF_LINE_2D_INITIAL(cardNo, AXIS_X, AXIS_Y, sv, v, a);
for (loop1 = 0; loop1 < 10000; loop1++)
{
i8092MF_LINE_2D_CONTINUE (cardNo, 0, 100, 100);
i8092MF_LINE_2D_CONTINUE (cardNo, 0, -100, -100);
}
i8092MF_LINE_2D_CONTINUE (cardNo, 1, 100, 100);
//設定第 1 卡，執行 X、Y 兩軸直線連續補間運動。

6.3.3 二軸多段連續補間

● BYTE i8092MF_CONTINUE_INTP(BYTE cardNo, WORD nAcc, DWORD VSV, DWORD VV, DWORD VA, DWORD VD, BYTE nType[], long cp1[], long cp2[], long fp1[], long fp2[])

功能: 執行多點連續補間(對稱 T 曲線)。
(軟體功能的巨集函式，會耗用系統資源)

參數: **cardNo:** 指定卡號
nAcc: 0 → 定速度補間 (VV)
1 → 對稱 T 曲線加減速補間 (VSV、VV、VA、VD)

VSV: 設定補間向量起始速度(PPS)

VV: 設定補間向量速度(PPS)

VA: 設定補間向量加速度(PPS/Sec)

VD: 設定末段補間向量減速度(PPS/Sec)

nType[]: 連續補間點最大: 1024 點(0 ~ 1023)
1 → i8092MF_LINE_2D(BYTE cardNo, long fp1, long fp2)
2 → i8092MF_ARC_CW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2)
3 → i8092MF_ARC_CCW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2)
4 → i8092MF_CIRCLE_CW(BYTE cardNo, long cp1, long cp2)
5 → i8092MF_CIRCLE_CCW(BYTE cardNo, long cp1, long cp2)
7 → 連續補間結束

cp1[]: 指定 X 軸圓、弧中心相對位置(-8,388,607 ~ +8,388,607)

cp2[]: 指定 Y 軸圓、弧中心相對位置(--8,388,607 ~ +8,388,607)

fp1[]: 指定 X 軸 Pulse 數(-8,388,607 ~ +8,388,607)
指定 X 軸圓弧終點相對位置

fp2[]: 指定 Y 軸 Pulse 數(-8,388,607 ~ +8,388,607)
指定 Y 軸圓弧終點相對位置

回應: YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8092MF_GET_ERROR_CODE)
NO: 沒有錯誤

範例: BYTE cardNo=1; //設定第 1 號卡。
int sv=100; //設定向量初始速度為 100 PPS。
int v=3000; //設定向量速度為 3000 PPS。
int a=2000; //設定向量加速度為 2000 PPS/s。

```

int d=2000; //設定向量減速度為 2000 PPS/s。
i8092MF_SET_MAX_V(cardNo, AXIS_XY, 20000); //設定各軸最高速度為 20K PPS。
BYTE nType[10]= { 1, 2, 1, 2, 1,7,0,0,0,0};
long cp1[10]= { 0, 10000, 0, 0, 0,0,0,0,0,0};
long cp2[10]= { 0, 0, 0,-10000, 0,0,0,0,0,0};
long fp1[10]= { 10000, 10000, 1000, 10000,-31000,0,0,0,0,0};
long fp2[10]= { 10000, 10000, 0,-10000,-10000,0,0,0,0,0};
i8092MF_CONTINUE_INTP(
cardNo, 1, sv, v, a, d, nType,cp1, cp2, fp1, fp2,fp3);
//設定第 1 卡，執行多點連續補間而減速點會自動運算。
//此範例主要以兩軸補間，直線搭配圓弧的運動，起點運動後最終將回到起點位置。

```

6.3.4 二軸比例運動

- **void i8092MF_RATIO_INITIAL**(BYTE *cardNo*, DWORD *SV*, DWORD *V*, DWORD *A*, float *ratio*)

功能： 比例運動初始設定(對稱 T 曲線加減速)。

參數：
cardNo: 指定卡號
SV: 設定比例運動初始速度(PPS)
V: 設定比例運動速度(PPS)
A: 設定比例運動加速度(PPS/Sec)
ratio: 設定兩軸的比例

回應： 無

範例： 和 **void i8092MF_RATIO_2D**(BYTE *cardNo*, WORD *nType*, long *data*, WORD *nDir*) 並用。

- **BYTE i8092MF_RATIO_2D**(BYTE *cardNo*, WORD *nType*, long *data*, WORD *nDir*)

功能： 執行比例連續運動。
(軟體功能的巨集函式，會耗用系統資源)

參數：
cardNo: 指定卡號
nType: 0 → 比例連續運動
1 → 比例運動結束
data: 比例運動 X 軸 Pulse 數(-8,388,607 ~ +8,388,607)
nDir: 比例運動 Y 軸方向：
0 → 正轉 CW
1 → 反轉 CCW

回應： YES: 有錯誤發生(欲讀錯誤碼請搭配使用 **i8092MF_GET_ERROR_CODE**)
NO: 沒有錯誤

範例：
BYTE cardNo=1; //設定第 1 號卡。
int sv=300; //設定初始速度為 PPS。
int v=18000; //設定速度為 PPS。
long a=500000; //設定加速度為 PPS/s。
int loop1;
int loop2;
i8092MF_SET_MAX_V(cardNo, AXIS_XY,160000L);
i8092MF_RATIO_INITIAL(cardNo, AXIS_U, AXIS_X, sv, v, a, 0.36f);

```
for (loop2 = 0; loop2 < 5; loop2++)
{
    for (loop1 = 0; loop1 < 5; loop1++)
    {
        i8092MF_RATIO_2D(cardNo, 0, 3600, 0);
        i8092MF_RATIO_2D(cardNo, 0, 3600, 1);
    }
    i8092MF_RATIO_2D(cardNo, 0, 7200, 0);
    i8092MF_RATIO_2D(cardNo, 0, 3600, 1);
}
i8092MF_RATIO_2D(cardNo, 1, 7200, 0);
//設定第 1 卡，執行 X、Y 兩軸比例運動。
```

6.3.5 二軸混合連續補間

- **void i8092MF_MIX_2D_INITIAL**(BYTE *cardNo*, WORD *nAcc*, DWORD *VSV*, DWORD *VV*, DWORD *VA*)

功能： 二軸直線和圓弧連續補間初始設定。

參數：
cardNo: 指定卡號
nAcc: 0 → 定速度補間 (vv)
 1 → 對稱 T 曲線加減速補間 (vsv、vv、va)
VSV: 設定向量初始速度(PPS)
VV: 設定向量速度(PPS)
VA: 設定加速度(PPS/Sec)

回應： 無

範例： 和 **void i8092MF_MIX_2D_CONTINUE**(
BYTE *cardNo*, WORD *nAcc*, WORD *nType*, long *cp1*, long *cp2*, long *fp1*, long *fp2*)並用。

- **BYTE i8092MF_MIX_2D_CONTINUE**(BYTE *cardNo*, WORD *nAcc*, WORD *nType*, long *cp1*, long *cp2*, long *fp1*, long *fp2*)

功能： 執行二軸直線和圓弧連續補間(可連續，無段數限制)。

參數：
cardNo: 指定卡號
nAcc: 0 → 連續補間
 1 → 結束連續補間減速停止(定速度不需減速)
nType: 1 → i8092MF_LINE_2D(BYTE *cardNo*, long *fp1*, long *fp2*)
 2 → i8092MF_ARC_CW(BYTE *cardNo*, long *cp1*, long *cp2*, long *fp1*, long *fp2*)
 3 → i8092MF_ARC_CCW(BYTE *cardNo*, long *cp1*, long *cp2*, long *fp1*, long *fp2*)
 4 → i8092MF_CIRCLE_CW(BYTE *cardNo*, long *cp1*, long *cp2*)
 5 → i8092MF_CIRCLE_CCW(BYTE *cardNo*, long *cp1*, long *cp2*)
cp1: 指定 X 軸圓、弧中心相對位置(-8,388,607 ~ +8,388,607)
cp2: 指定 Y 軸圓、弧中心相對位置(-8,388,607 ~ +8,388,607)
fp1: 指定 X 軸 Pulse 數(-8,388,607 ~ +8,388,607)
fp2: 指定 Y 軸 Pulse 數(-8,388,607 ~ +8,388,607)

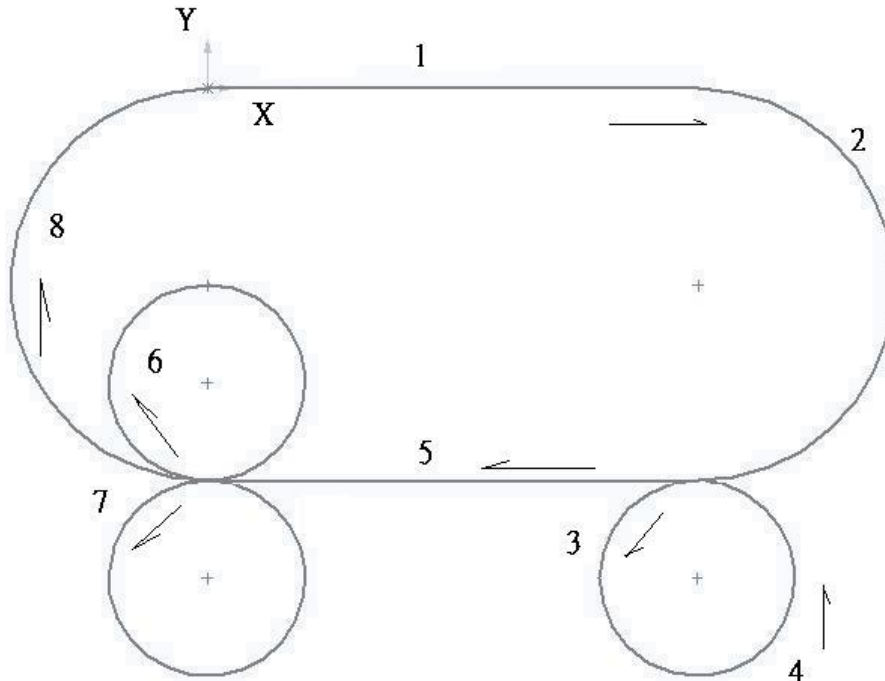
回應： YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8092MF_GET_ERROR_CODE)
 NO: 沒有錯誤

範例： BYTE *cardNo*=1; //設定第 1 號卡。
 unsigned short *sv*=300; //設定向量初始速度為 PPS。
 unsigned short *v*=8000; //設定向量速度為 PPS。

```

unsigned long a=50000; //設定向量加速度為 PPS/s。
unsigned short loop1;
i8092MF_SET_MAX_V(cardNo, AXIS_XY,16000L);
i8092MF_MIX_2D_INITIAL(cardNo, 1, sv, v, a);
i8092MF_MIX_2D_CONTINUE(cardNo, 0, 1, 0,    0, 50000,    0);
//1 segment
i8092MF_MIX_2D_CONTINUE(cardNo, 0, 2, 0,-20000,    0,-40000);
//2 segment
i8092MF_MIX_2D_CONTINUE(cardNo, 0, 3, 0,-10000,    0,-20000);
//3 segment
i8092MF_MIX_2D_CONTINUE(cardNo, 0, 3, 0, 10000,    0, 20000);
//4 segment
i8092MF_MIX_2D_CONTINUE(cardNo, 0, 1, 0,    0,-50000,    0);
//5 segment
i8092MF_MIX_2D_CONTINUE(cardNo, 0, 4, 0, 10000,    0,    0);
//6 segment
i8092MF_MIX_2D_CONTINUE(cardNo, 0, 5, 0,-10000,    0,    0);
//7 segment
i8092MF_MIX_2D_CONTINUE(cardNo, 1, 2, 0, 20000,    0, 40000);
//8 segment and end the Interpolation
//設定第 1 卡，執行 X、Y 兩軸連續補間運動。

```



6.4 中斷條件因子設定

6.4.1 中斷條件設定

● **void i8092MF_INTFACTOR_ENABLE(BYTE cardNo, WORD axis, WORD nINT)**

功能： 設定中斷條件因子

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)
nINT 中斷條件因子，如下表說明

號碼	代號	說明
0	PULSE	不支援
1	P>=C-	中斷發生於位置計數器大於等於負方向比較暫存器的設定值必須和 i8092MF_SET_COMPARE 並用 (6.5.7)
2	P<C-	中斷發生於位置計數器小於負方向比較暫存器的設定值必須和 i8092MF_SET_COMPARE 並用 (6.5.7)
3	P<C+	中斷發生於位置計數器小於正方向比較暫存器的設定值必須和 i8092MF_SET_COMPARE 並用 (6.5.7)
4	P>=C+	中斷發生於位置計數器大於等於正方向比較暫存器的設定值必須和 i8092MF_SET_COMPARE 並用 (6.5.7)
5	C-END	中斷發生於等速段和補正段結束兩個位置
6	C-STA	中斷發生於等速段和補正段開始兩個位置
7	D-END	中斷發生於驅動結束時

回應： 無

範例：

```
HANDLE hINT; //中斷事件變數宣告
HANDLE i8092_hThread; //中斷執行緒變數宣告
DWORD WINAPI i8092_ThreadFunction(LPVOID IParm); //中斷執行緒函數宣告
BYTE CardNo=1;
BYTE Slot1=1;

//按鈕事件程式:建立執行緒函數與開啟中斷功能
void CI8092QCdIlg::OnTestint()
{
    DWORD dwThreadId = 0;
    HWND hWnd = NULL;
    //建立 i8092_ThreadFunction 執行緒函數
    i8092_hThread = CreateThread(NULL, 0, i8092_ThreadFunction, hWnd, 0,
    &dwThreadId);
    //設置 2 軸(XY)
    BYTE axis=AXIS_XY;
```



```

i8092MF_SET_MAX_V(CardNo, axis, 20000);
i8092MF_NORMAL_SPEED(CardNo, axis, 0);
i8092MF_SET_V(CardNo, axis, 20000);
i8092MF_SET_A(CardNo, axis, 100000);
i8092MF_SET_SV(CardNo, axis, 20000);
i8092MF_SET_AO(CardNo, axis, 0);
//中斷功能初使化
hINT=Slot_Register_Interrupt(Slot1);
//設置中斷條件: D-END
i8092MF_INTFACTOR_ENABLE(CardNo, AXIS_X, 7);
// 2 軸定量驅動
i8092MF_FIXED_MOVE(CardNo, AXIS_XY, 10000);

    while (i8092MF_STOP_WAIT(CardNo, AXIS_XY) == NO)
    { // 等待運動軸停止
        DoEvents();
        Sleep(1);
    }
}

// 中斷執行緒函數
DWORD WINAPI i8092_ThreadFunction(LPVOID IPParam)
{
    DWORD dwEvent;
    WORD RR3_X;
    if(hINT != NULL)
    {
        // 等待中斷事件發生
        dwEvent = WaitForSingleObject(hINTP, INFINITE);
        switch(dwEvent)
        {
        case WAIT_OBJECT_0:
            // 成功取得中斷事件物件，以下為使用者的中斷執行緒程式
            // 當驅動結束後清除 X 軸位置計數器為 0
            i8092MF_SET_LP(CardNo, AXIS_X, 0)
            // ...
            // 其他中斷副程式...
            // ...
            // 中斷結束
            Slot_Interrupt_Done(Slot1);
            //讀取並清除 X 軸中斷條件狀態
            RR3_X = i8092_GET_RR3(CardNo, AXIS_X);
            //關閉中斷條件因子
            i8092MF_INTFACTOR_DISABLE(CardNo, AXIS_X);
            //關閉中斷功能
            Slot_Interrupt_Close(Slot1);

```

```
        break;
    case WAIT_TIMEOUT:
        break;
    case WAIT_FAILED:
        break;
    }
}

return 1;
}
```

備註： **Slot_Register_Interrupt(BYTE Slot)**，**Slot_Interrupt_Done(BYTE Slot)**，**Slot_Interrupt_Close(BYTE Slot)**此三各函數定義在新版 WinConSDK 中，i8092 模組的運動中斷功能需搭配此三各函數使用。

6.4.2 中斷條件關閉

- **void** i8092MF_INTFACTOR_DISABLE(**BYTE** cardNo, **WORD** axis)

功能： 關閉中斷發生條件

參數： **cardNo**: 指定卡號
axis: 指定軸號碼 (參考表 2-1)

回應： 無

範例： 請參考 6.5.1

6.4.3 讀取中斷狀態

- **WORD** i8092MF_GET_RR3(**BYTE** cardNo, **WORD** axis)

功能： 讀回主要狀態暫存器(RR3)。

參數： **cardNo**: 指定卡號
axis: 指定軸號碼(參考表 2-1)

回應： 主要狀態暫存器(RR3)的內容值。

內容值		說明
0x02	P>=C-	中斷發生於位置計數器大於等於負方向比較暫存器的設定值
0x04	P<C-	中斷發生於位置計數器小於負方向比較暫存器的設定值
0x08	P<C+	中斷發生於位置計數器小於正方向比較暫存器的設定值
0x10	P>=C+	中斷發生於位置計數器大於等於正方向比較暫存器的設定值
0x20	C-END	中斷發生於等速段和補正段結束兩個位置
0x40	C-STA	中斷發生於等速段和補正段開始兩個位置
0x80	D-END	中斷發生於驅動結束時

範例： i8092MF_GET_RR3(cardNo, AXIS_X);
//取回 X 軸的主要狀態暫存器(RR3)的值。

6.5 其他功能

6.5.1 設定軸暫停

- **void i8092MF_DRV_HOLD(BYTE cardNo, WORD axis)**

功能： 指定軸運動暫停。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 (參考表 2-1)

回應： 無

範例： 請參考 6.5.2

6.5.2 設定軸啟動

- **void i8092MF_DRV_START(BYTE cardNo, WORD axis)**

功能： 指定軸開始動作。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 (參考表 2-1)

回應： 無

範例：

```
BYTE cardNo=1; //設定第 1 號卡。
```

```
i8092MF_DRV_HOLD(cardNo, AXIS_XY); //設定 XY 兩軸暫停移動。
```

```
i8092MF_SET_MAX_V(cardNo, AXIS_XY, 8000); //設定 XY 軸最高速 8K PPS。
```

```
i8092MF_NORMAL_SPEED(cardNo, AXIS_XY, 0); //設定 XY 軸對稱 T 曲線。
```

```
i8092MF_SET_V(cardNo, AXIS_X, 2000); //設定 X 軸速度=2,000 PPS。
```

```
i8092MF_SET_A(cardNo, AXIS_X, 10000); //設定 X 軸加速度 10,000 PPS/S。
```

```
i8092MF_SET_SV(cardNo, AXIS_X, 1000); //設定 X 初始速度 1,000 PPS。
```

```
i8092MF_SET_V(cardNo, AXIS_Y, 4000); //設定 Y 軸速度=4,000 PPS。
```

```
i8092MF_SET_A(cardNo, AXIS_Y, 10000); //設定 Y 軸加速度 10,000 PPS/S。
```

```
i8092MF_SET_SV(cardNo, AXIS_Y, 500); //設定 Y 初始速度 500 PPS。
```

```
i8092MF_FIXED_MOVE(cardNo, AXIS_X, 5000); //X 移動 5,000 Pulse。
```

```
i8092MF_FIXED_MOVE(cardNo, AXIS_Y, 8000); //Y 移動 8,000 Pulse。
```

```
i8092MF_DRV_START(cardNo, AXIS_XY); //開始 XY 兩軸同時移動。
```

6.5.3 等待完成軸運動

- **BYTE i8092MF_STOP_WAIT(BYTE cardNo, WORD axis)**

功能： 等待軸完成停止。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 (參考表 2-1)

回應： YES 完成
NO 未完

範例： **BYTE cardNo=1; //設定第 1 號卡。**

```
i8092MF_SET_MAX_V(cardNo, AXIS_XY, 20000); //設定 XY 軸最高速 20K PPS。  
i8092MF_NORMAL_SPEED(cardNo, AXIS_XY, 0); //設定 XY 軸為對稱 T 曲線。  
i8092MF_SET_V(cardNo, AXIS_XY, 2000); //設定 XY 軸速度=2000 PPS。  
i8092MF_SET_A(cardNo, AXIS_XY,1000); //設定 XY 軸加速度=1000 PPS/S。  
i8092MF_SET_SV(cardNo, AXIS_XY, 2000); //設定 XY 軸初始速度=2000 PPS。  
i8092MF_SET_AO(cardNo, AXIS_XY, 9); //設定 XY 軸減速(保留脈波數)= 9 PPS。  
i8092MF_FIXED_MOVE(cardNo, AXIS_XY, 10000); //XY 軸移動 10000 Pulse。
```

```
if (i8092MF_STOP_WAIT(cardNo, AXIS_X) == NO)  
{  
    //第 cardNo 卡 X 軸運動尚未停止，處理程序。  
}
```

6.5.4 設定軸停止

- **void i8092MF_STOP_SLOWLY**(BYTE *cardNo*, WORD *axis*)

功能： 指定軸之輸出減速停止。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 (參考表 2-1)

回應： 無

範例： **i8092MF_STOP_SLOWLY(1, AXIS_XY);**
//設定第 1 卡 XY 軸，減速停止。

- **void i8092MF_STOP_SUDDENLY**(BYTE *cardNo*, WORD *axis*)

功能： 指定軸之輸出立即(緊急)停止。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 (參考表 2-1)

回應： 無

範例： **i8092MF_STOP_SUDDENLY(1, AXIS_XY);**
//設定第 1 卡 XY 軸，立即(緊急)停止。

- **void i8092MF_VSTOP_SLOWLY**(BYTE *cardNo*)

功能： 指定補間軸之輸出減速停止。

參數： **cardNo:** 指定卡號

回應： 無

範例： **i8092MF_VSTOP_SLOWLY(1);**
//設定第 1 卡補間軸，減速停止。

● **void** i8092MF_VSTOP_SUDDENLY(BYTE cardNo)

功能： 指定補間軸之輸出立即(緊急)停止。

參數： **cardNo:** 指定卡號

回應： 無

範例： i8092MF_VSTOP_SUDDENLY(1);
//設定第 1 卡補間軸，立即(緊急)停止。

● **void** i8092MF_SSTOP_SLOWLY(BYTE cardNo, WORD axis)

功能： 指定軸之輸出減速停止。

這與 i8092MF_STOP_SLOWLY()的差別在於 i8092MF_SSTOP_SLOWLY()
僅停止軸之輸出，不涉及軸卡狀況的管控(不會產生 ERROR_CODE 256)。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 (參考表 2-1)

回應： 無

範例： i8092MF_SSTOP_SLOWLY(1, AXIS_XY);
//設定第 1 卡 XY 軸，減速停止。

● **void** i8092MF_SSTOP_SUDDENLY(BYTE cardNo, WORD axis)

功能： 指定軸之輸出立即(緊急)停止。

這與 i8092MF_STOP_SUDDENLY()的差別在於 i8092MF_SSTOP_ SUDDENLY()僅停止軸之輸出，不涉及軸卡狀況的管控(不會產生 ERROR_CODE 256)。

參數： **cardNo:** 指定卡號
axis: 指定軸號碼 (參考表 2-1)

回應： 無

範例： i8092MF_SSTOP_SUDDENLY(1, AXIS_XY);
//設定第 1 卡 XY 軸，立即(緊急)停止。

● **void i8092MF_SVSTOP_SLOWLY(BYTE cardNo)**

功能: 指定補間軸之輸出減速停止。

這與 **i8092MF_VSTOP_SLOWLY()**的差別在於 **i8092MF_SVSTOP_SLOWLY()**僅停止軸之輸出，不涉及軸卡狀況的管控(不會產生 **ERROR_CODE 256**)。

參數: **cardNo:** 指定卡號

回應: 無

範例: **i8092MF_SVSTOP_SLOWLY(1);**
//設定第 1 卡補間軸，減速停止。

● **void i8092MF_SVSTOP_SUDDENLY(BYTE cardNo)**

功能: 指定補間軸之輸出立即(緊急)停止。

這與 **i8092MF_SVSTOP_SUDDENLY ()**的差別在於 **i8092MF_SVSTOP_SUDDENLY ()**僅停止軸之輸出，不涉及軸卡狀況的管控(不會產生 **ERROR_CODE 256**)。

參數: **cardNo:** 指定卡號

回應: 無

範例: **i8092MF_SVSTOP_SUDDENLY(1);**
//設定第 1 卡補間軸，立即(緊急)停止。

6.5.5 清除停止狀態

- **void i8092MF_CLEAR_STOP(BYTE cardNo)**

功能： 軸運動發生錯誤時，或使用(i8092MF_STOP_SLOWLY，i8092MF_STOP_SUDDENLY)，使軸運動停止，請故障排除後，清除錯誤狀態。

參數： **cardNo:** 指定卡號

回應： 無

範例： **i8092MF_CLEAR_STOP(1);**
//清除第 1 卡錯誤狀態。

6.5.6 補間動作結束(單軸運動或改變座標系)

- **void i8092MF_INTP_END(BYTE cardNo)**

功能： 補間動作結束。

參數： **cardNo:** 指定卡號

回應： 無

範例： **i8092MF_INTP_END(1);**
//設定第 1 卡補間，動作結束。

6.5.7 設定 COMPARE 值

- `void i8092MF_SET_COMPARE(BYTE cardNo, WORD axis, WORD nSELECT, WORD nTYPE, long data)`

功能： 設定位置比較器的值，將會使軟體極限功能失效。

參數：
cardNo: 指定卡號
axis: 指定軸號碼(參考表 2-1)
nSELECT: 0 → C+
 1 → C-
nTYPE: 0 → Position(P) = LP
 1 → Position(P) = EP
data: 設定 COMPARE 值: -2,147,483,648 ~ +2,147,483,647

回應： 無

範例：
`i8092MF_SET_COMPARE(cardNo, AXIS_X, 0, 1, 5000);`
`//設定 COMP+ 的值=5,000，來源參考 X 軸 EP。`