

# i-8094 運動控制模組使用手冊

(Version 2.5)

應用程式函式庫

WinCon-8000、I-8000、PAC 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 .....	7
1.1 手冊的使用 .....	7
1.2 巨集函式.....	7
1.3 函式基本結構說明 .....	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.....	11
2.4 軸輸出PULSE模式設定 .....	11
2.5 設定軸速度輸出最大範圍 .....	12
2.6 設定軸前後極限的輸入觸發邏輯 .....	12
2.7 設定碰觸前後極限的處理模式 .....	13
2.8 設定軸近原點輸入觸發邏輯 .....	13
2.9 設定軸原點輸入觸發邏輯 .....	14
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).....	15
2.13 設定伺服馬達異常ALARM輸入參數 .....	16

2.14	設定伺服馬達定位完成輸入參數 .....	16
2.15	設定數位輸入雜訊濾波功能 .....	17
2.16	指定軸為圓形運動軸(環狀計數器).....	18
2.17	三角形速度曲線的預防 .....	19
2.18	外部輸入驅動 .....	20
2.18.1	手輪脈波驅動.....	20
2.18.2	固定脈波驅動.....	21
2.18.3	連續脈波驅動.....	21
2.18.4	外部輸入關閉.....	22
2.19	載入設定檔並完成相關的硬體設定 .....	23
<b>3</b>	<b>狀態讀取及設定功能.....</b>	<b>24</b>
3.1	設定及讀取指令邏輯位置 .....	24
3.2	設定及讀取ENCODER位置 .....	25
3.3	讀取目前速度 .....	26
3.4	讀取目前加速度 .....	26
3.5	讀取目前DI狀態 .....	27
3.6	讀取目前ERROR狀態 .....	29
<b>4</b>	<b>FRNET功能(I8094F專用函式) .....</b>	<b>31</b>
4.1	FRnet DI讀取 .....	31
4.2	FRnet DO寫入 .....	31
<b>5</b>	<b>軸自動歸零.....</b>	<b>32</b>

5.1 設定軸歸零速度 .....	34
5.2 設定以極限當原點 .....	34
5.3 設定歸零模式 .....	35
5.3.1 步驟一高速尋找近原點 .....	35
5.3.2 步驟二低速尋找原點 .....	36
5.3.3 步驟三低速尋找Z-相 .....	37
5.3.4 步驟四高速補正驅動 .....	37
5.4 啟動軸歸零 .....	39
5.5 等待完成歸零動作 .....	39
<b>6 軸控功能 .....</b>	<b>40</b>
6.1 各軸獨立運動 .....	40
6.1.1 設定加減速模式 .....	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 設定軸減速度變化率 .....	44
6.1.8 設定軸減速(保留脈波數) .....	45
6.1.9 固定脈波數輸出 .....	46
6.1.10 連續脈波輸出 .....	46
6.2 補間運動 .....	47
6.2.1 設定補間軸 .....	47
6.2.2 設定補間加減速模式 .....	48
6.2.3 設定軸向量初始速度 .....	52
6.2.4 設定軸向量定速度 .....	52
6.2.5 設定軸向量加速度 .....	53
6.2.6 設定軸向量減速度 .....	53
6.2.7 設定軸向量加速度變化率 .....	54
6.2.8 設定軸向量減速度變化率 .....	54
6.2.9 設定軸向量減速(保留脈波數) .....	55

6.2.10	二軸直線補間.....	56
6.2.11	三軸直線補間.....	57
6.2.12	二軸圓弧補間.....	58
6.2.13	二軸圓形補間.....	60
<b>6.3</b>	<b>同步運動.....</b>	<b>61</b>
6.3.1	設定同步運動條件.....	62
6.3.2	設定COMPARE值 .....	65
6.3.3	讀取LATCH值.....	65
6.3.4	設定PRESET資料.....	66
<b>6.4</b>	<b>連續補間運動 .....</b>	<b>67</b>
6.4.1	二軸矩形連續補間.....	68
6.4.2	二軸直線連續補間.....	69
6.4.3	三軸直線連續補間.....	71
6.4.4	二軸混合連續補間.....	73
6.4.5	多點連續補間(陣列資料).....	75
6.4.6	三軸螺旋運動.....	77
6.4.7	二軸比例運動.....	78
<b>6.5</b>	<b>中斷條件因子設定 .....</b>	<b>80</b>
6.5.1	中斷條件設定.....	80
6.5.2	中斷條件關閉.....	83
6.5.3	讀取中斷狀態.....	83
<b>6.6</b>	<b>其他功能.....</b>	<b>84</b>
6.6.1	設定軸暫停 .....	84
6.6.2	設定軸啓動 .....	84
6.6.3	等待完成軸運動 .....	85
6.6.4	設定(補間)軸停止.....	85
6.6.5	清除停止狀態.....	88
6.6.6	補間動作結束.....	88

---

# 1 前言

---

## 1.1 手冊的使用

- 使用 i8094 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(eVC++、VB.net、C#.net)及作業系統(MiniOS7 / WinCE / Linux)。

## 1.2 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。

## 2 基本設定功能

### 2.1 各軸定義的代碼

所有功能中有關軸參數，是以 X=1、Y=2、Z=4、U=8 作為代碼，假設我們要指定 XY=3，就是 1+2=3，又如 YZ=0x6(2+4=6)，以此類推，XYZU=0xf(1+2+4+8)，因此同一功能，可以一次做單軸設定，也可以一次設多軸相同設定，所有功能中有關軸參數代碼(WORD axis)與意義如下：

對照表(2-1)

軸	X	Y	Z	U	XY	XZ	XU	YZ
代碼	0x1	0x2	0x4	0x8	0x3	0x5	0x9	0x6
變數	AXIS_X	AXIS_Y	AXIS_Z	AXIS_U	AXIS_XY	AXIS_XZ	AXIS_XU	AXIS_YZ
軸	YU	ZU	XYZ	XYU	XZU	YZU	XYZU	
代碼	0xa	0xc	0x7	0xb	0xd	0xe	0xf	
變數	AXIS_YU	AXIS_ZU	AXIS_XYZ	AXIS_XYU	AXIS_XZU	AXIS_YZU	AXIS_XYZU	

### 2.2 註冊軸卡及版本讀取

#### ● BYTE i8094MF\_REGISTRATION(BYTE cardNo, BYTE slot)

功能： 註冊軸卡，指定插槽及卡號，使用 i8094 所有功能前，都必須做此註冊。

參數： **cardNo:** 指定卡號 → I-8000 : 1~8  
→ WinCon-8000 : 1~7  
**slot:** 插槽號碼 → I-8000 : 0~7  
→ WinCon-8000 : 1~7

回應： YES: 正常  
NO: 異常



```

範例: //===== WinCon-8000 =====
//設定各槽(slot1~slot7)，對應的卡號為 1~7。
BYTE cardNo;
BYTE slot;
short int Found = 0;
for (slot = 1; slot < 8; slot++)
{
    cardNo = slot;
    if (i8094MF_REGISTRATION(cardNo, slot) == YES)
    {
        //找到軸卡，註冊。
        i8094MF_RESET_CARD(cardNo);
        Found++;
    }
}
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;
short int Found = 0;
for (slot = 0; slot < 8; slot++)
{
    cardNo = slot + 1;
    if (i8094MF_REGISTRATION(cardNo, slot) == YES)
    {
        //找到軸卡，註冊。
        i8094MF_RESET_CARD(cardNo);
        Found++;
    }
}
if (Found == 0)
{
    //找不到軸卡，異常處理。
    return;
}

```

● **WORD i8094MF\_GET\_VERSION(void)**

功能: 讀取 i8094 運動函式庫之版本。

參數: **cardNo:** 指定卡號

回應: 版本號碼: 西元年月 0x0000 ~ 0x9999

範例: WORD VER\_No;  
VER\_No = i8094MF\_GET\_VERSION();  
//讀取 i8094.dll 版本號碼。

備註: 以下為讀到的版本release資訊(2006年7月)

i8094MF\_GET\_VERSION: 0x0607

0,6 → 函式庫版本年流水序

0,7 → 函式庫版本月流水序

## 2.3 軸卡重置

- **void i8094MF\_RESET\_CARD(BYTE cardNo)**

功能： I-8094 重設成電源開啟狀態。

參數： **cardNo:** 指定卡號

回應： 無

範例： **i8094MF\_RESET\_CARD (1);**  
*//重置第 1 卡。*

## 2.4 軸輸出 PULSE 模式設定

- **void i8094MF\_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： **i8094MF\_SET\_PULSE\_MODE(1, AXIS\_XYZ, 2);**  
**i8094MF\_SET\_PULSE\_MODE(1, AXIS\_U, 3);**  
*//指定第 1 卡 XYZ 軸，脈波輸出模式皆為 2。*  
*//指定第 1 卡 U 軸，脈波輸出模式為 3。*

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 i8094MF\_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, 第二軸(8,000~2,828,854 PPS)  
補間最高速度, 第三軸(8,000~2,309,468 PPS)

回應: 無

範例: `i8094MF_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 i8094MF\_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: `i8094MF_SET_HLMT(1, AXIS_XYZU, 0, 0);`  
`//設定第1卡 XYZU 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 i8094MF\_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：  
`i8094MF_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 i8094MF\_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：  
`i8094MF_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 i8094MF\_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：  
`i8094MF_SET_HOME_EDGE(1, AXIS_XYZU, 1);`  
//設定第 1 卡 X Y Z U 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 i8094MF\_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：  
`i8094MF_SET_SLMT(1, AXIS_XYZU, 20000, -3000, 0);`  
//設定第 1 卡 X Y Z U 軸，以指令輸出位置做比較，前軟體極限=20000，後軟體極限=-3000。

- **void i8094MF\_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：  
`i8094MF_CLEAR_SLMT(1, AXIS_XYZU);`  
//取消第 1 卡 X Y Z U 軸，前後軟體極限功能。

## 2.11 設定編碼器輸入參數

- **void i8094MF\_SET\_ENCODER(BYTE cardNo, WORD axis, BYTE nMode, BYTE nDivision, 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  
**nDivision:** 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：  
`i8094MF_SET_ENCODER(1, AXIS_XYZU, 0, 0, 0);`  
`//設定第 1 卡 X Y Z U 軸，編碼器輸入為 AB 相，不除頻，Z 輸入信號低準位觸發。`

## 2.12 伺服開關(Servo\_ON/OFF)

- **void i8094MF\_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：  
`i8094MF_SERVO_ON(1, AXIS_XYZU);`  
`//設定第 1 卡 X Y Z U 軸，啟動驅動器伺服。`

- **void i8094MF\_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：  
`i8094MF_SERVO_OFF(1, AXIS_XYZU);`  
`//設定第 1 卡 X Y Z U 軸，關閉驅動器伺服。`

## 2.13 設定伺服馬達異常 ALARM 輸入參數

- **void i8094MF\_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：  
`i8094MF_SET_ALARM(1, AXIS_ZU, 1, 0);`  
*//設定第 1 卡 Z U 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 i8094MF\_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：  
`i8094MF_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(i8094\_Getting\_Started\_1.6tc.pdf)手冊，一般 DI 輸入接線範例。



## 2.15 設定數位輸入雜訊濾波功能

- `void i8094MF_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 $\mu$ SEC	2 $\mu$ SEC
1	224 $\mu$ SEC	256 $\mu$ SEC
2	448 $\mu$ SEC	512 $\mu$ SEC
3	896 $\mu$ 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：  
`i8094MF_SET_FILTER(1, AXIS_XYZU, 21, 3);`  
 //設定第 1 卡 X Y Z U 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)

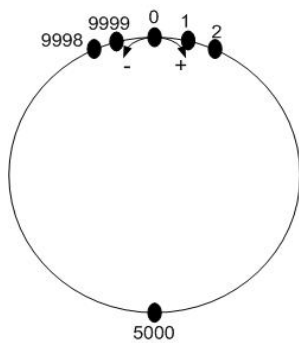
### ● void i8094MF\_VRING\_ENABLE(BYTE cardNo, WORD axis, DWORD nVRing)

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(1 ~ 0xffffffff)

回應： 無

範例：  
`i8094MF_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。`



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

環狀計數器=9999

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 i8094MF\_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：  
`i8094MF_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 i8094MF\_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：  
`i8094MF_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 i8094MF\_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：  
`i8094MF_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

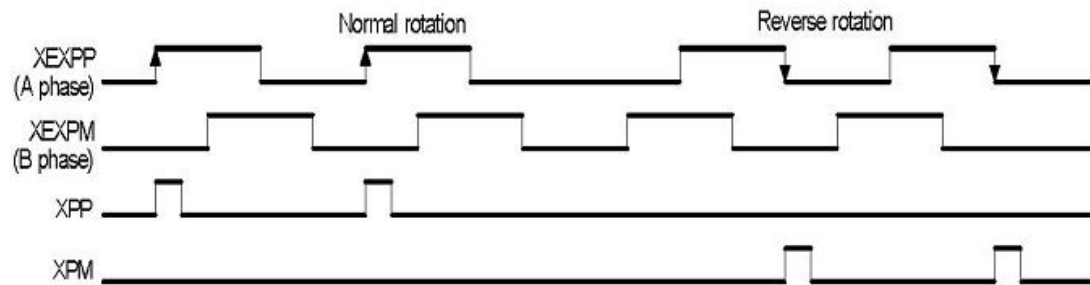
● **void i8094MF\_EXD\_MP(BYTE cardNo, WORD axis, long data)**

功能： 執行手輪輸入驅動，輸出固定步數。

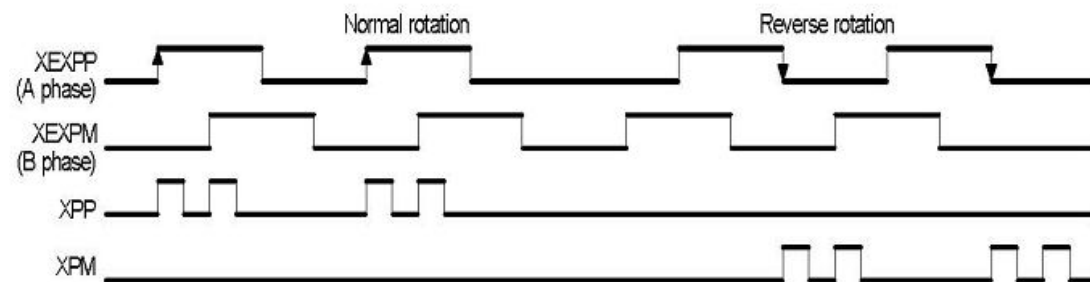
參數：  
**cardNo:** 指定卡號  
**axis:** 指定軸號碼 X 或 Y 或 Z 或 U (1 or 2 or 4 or 8)  
**data:** 指定步數

回應： 無

範例：  
`i8094MF_EXD_MP(1, AXIS_X, 1);`  
//第1卡 X 軸，手輪觸發移動1步(Pulse)。



`i8094MF_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

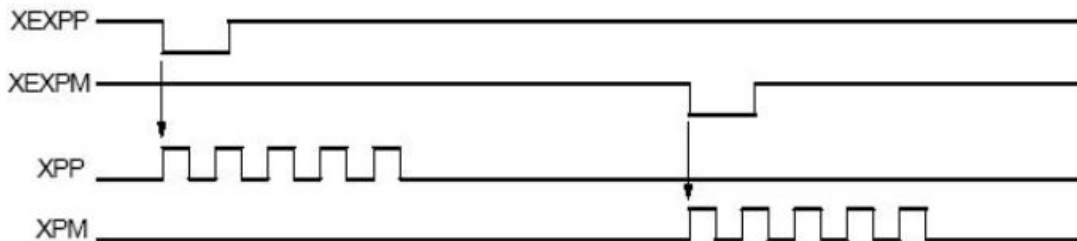
● **void i8094MF\_EXD\_FP**(**BYTE cardNo**, **WORD axis**, **long data**)

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 或 Z 或 U (1 or 2 or 4 or 8)  
**data**: 指定步數

回應： 無

範例：  
`i8094MF_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

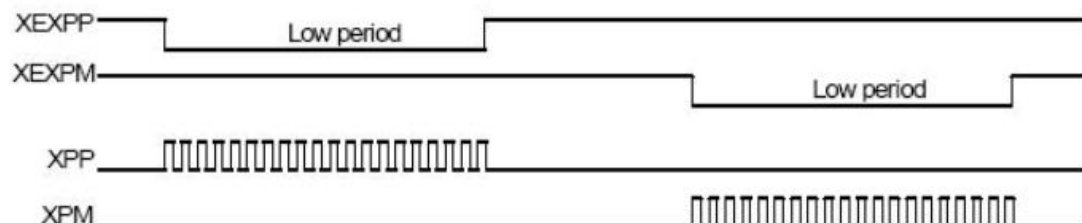
● **void i8094MF\_EXD\_CP**(**BYTE cardNo**, **WORD axis**, **long data**)

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 或 Z 或 U (1 or 2 or 4 or 8)  
**data**: 設定速度

回應： 無

範例：  
`i8094MF_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** i8094MF\_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 或 Z 或 U (1 or 2 or 4 or 8)

回應： 無

範例：  
i8094MF\_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** `i8094MF_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： `i8094MF_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** i8094MF\_SET\_LP(**BYTE** cardNo, **WORD** axis, **long** wdata)

功能： 設定軸之目前指令邏輯位置。

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：  
i8094MF\_SET\_LP(1, AXIS\_XYZU, 0);  
//設定第 1 卡 X Y Z U 軸，目前指令位置皆為 0。

- **long** i8094MF\_GET\_LP(**BYTE** cardNo, **WORD** axis)

功能： 讀取軸目前之指令邏輯位置。

參數：  
**cardNo:** 指定卡號  
**axis:** 指定軸號碼 X 或 Y 或 Z 或 U (1 or 2 or 4 or 8)

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

範例：  
**long** X\_LP;  
X\_LP = i8094MF\_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 i8094MF\_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： `i8094MF_SET_EP(1, AXIS_XYZU, 0);`  
`//設定第 1 卡 X Y Z U 軸，目前 ENCODER 回授位置皆為 0。`

### ● long i8094MF\_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 或 Z 或 U (1 or 2 or 4 or 8)

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

範例： `long X_EP;`  
`X_EP = i8094MF_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** `i8094MF_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 或 Z 或 U (1 or 2 or 4 or 8)

回應: 目前速度(PPS)

範例: `DWORD dwdata;`  
`dwdata = i8094MF_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** `i8094MF_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 或 Z 或 U (1 or 2 or 4 or 8)

回應: 目前加速度(PPS/Sec)

範例: `DWORD dwdata;`  
`dwdata = i8094MF_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** i8094MF\_GET\_DI(**BYTE** cardNo, **WORD** axis, **WORD** nType)

功能： 讀取軸之輸入點狀態。

參數：

<b>cardNo:</b>	指定卡號
<b>axis:</b>	指定軸號碼 X 或 Y 或 Z 或 U (1 or 2 or 4 or 8)
<b>nType:</b>	0 → DRIVING (檢查 i8094 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 IN1 信號)
	6 → NEAR HOME (檢查近原點 IN0 信號)
	7 → IN3 (檢查 IN3 信號)
	8 → INPOS (檢查 INPOS 信號)
	9 → INDEX (檢查編碼器 Z 相信號)

回應： YES: on  
NO: off

範例：

```
if (i8094MF_GET_DI(1, AXIS_X, 1) == YES)
{
    //讀取第 1 卡 X 軸，前極限信號處理。
}
```

● **WORD** i8094MF\_GET\_DI\_ALL(**BYTE** cardNo, **WORD** axis)

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 或 Z 或 U (1 or 2 or 4 or 8)

回應： 為下列值的組合

0x001 → DRIVING (檢查 i8094 有沒有輸出脈波)  
0x002 → LIMIT+ (檢查有沒有碰觸前極限)  
0x004 → LIMIT- (檢查有沒有碰觸後極限)  
0x008 → EMERGENCY (檢查緊急停止信號)  
0x010 → ALARM (檢查警報信號)  
0x020 → HOME (檢查原點信號為 HIGH)  
0x040 → NEAR HOME (檢查近原點信號為 HIGH)  
0x080 → IN3 (檢查 IN3 信號為 HIGH)  
0x100 → INPOS (INPOS 信號為 HIGH)  
0x200 → Z-Phase (Z 相 IN2 信號為 HIGH)

範例： **WORD** wStatus;  
wStatus =i8094H\_GET\_DI\_ALL(1, AXIS\_X);

```
if( (wStatus & 0x002) == 0x002 )  
{  
    //讀取第 1 卡 X 軸，前極限信號處理。  
}
```

## 3.6 讀取目前 ERROR 狀態

### ● BYTE i8094MF\_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 i8094MF\_GET\_ERROR\_CODE)  
如果 GET\_ERROR\_CODE = 256 → 表示有使用"設定(補間)軸停止"  
"，請參考 6.5.5 及如下範例排除 ERROR。

NO: 沒有錯誤

範例： **if (i8094MF\_GET\_ERROR(1) == YES)**  
{

*//讀取第 1 卡，錯誤停止處理。*

**WORD ErrorCode\_X = i8094MF\_GET\_ERROR\_CODE(1, AXIS\_X);**

**WORD ErrorCode\_Y = i8094MF\_GET\_ERROR\_CODE(1, AXIS\_Y);**

**WORD ErrorCode\_Z = i8094MF\_GET\_ERROR\_CODE(1, AXIS\_Z);**

**WORD ErrorCode\_U = i8094MF\_GET\_ERROR\_CODE(1, AXIS\_U);**

**if ((ErrorCode\_X || ErrorCode\_Y || ErrorCode\_Z || ErrorCode\_U) == 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。*

**i8094MF\_CLEAR\_STOP(1);**

}

}

● **WORD i8094MF\_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 或 Z 或 U (1 or 2 or 4 or 8)

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，會傳回所有錯誤碼總和。

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	Reserved	保留
128	HOME	Z 相和 HOME 同時 on
256	請參考 6.5.4	軟體緊急停止

例: 48 表示(16+32)，"伺服警報"及"緊急停止"同時發生

---

## 4 FRnet 功能(i8094F 專用函式)

---

### 4.1 FRnet DI 讀取

- **WORD** i8094MF\_FRNET\_IN(**BYTE** cardNo, **WORD** wRA)

功能: 讀取 FRnet 的數位輸入資料，**RA** 的意義是 **Receiving Address**，一個 FRnet 合法的群組號碼是由 16 位元資料組成，如此在 FRnet 的界面共能定義 128 DI。

參數: **cardNo:** 指定卡號  
**wRA:** 群組範圍 RA8~RA15

回應: **WORD** 16-位元輸入資料

範例: **WORD** IN\_Data;  
**IN\_Data = i8094MF\_FRNET\_IN(1, 8);**  
*//設定第 1 卡，RA 群組 = 8。*

### 4.2 FRnet DO 寫入

- **void** i8094MF\_FRNET\_OUT(**BYTE** cardNo, **WORD** wSA, **WORD** data)

功能: 寫入 FRnet 的數位輸出資料，**SA** 的意義是 **Sending Address**，一個 FRnet 合法的群組號碼是由 16 位元資料組成，如此在 FRnet 的界面共能定義 128 DO。

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

回應: 無

範例: **i8094MF\_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

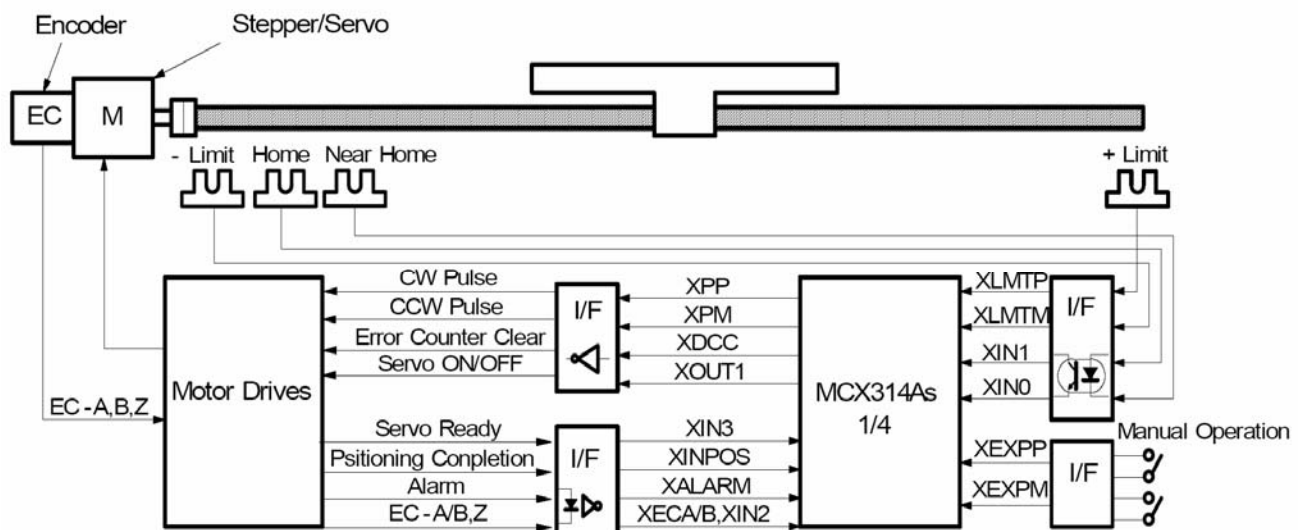
I-8094 提供自動歸零功能，只要經適當設定後，即可下指令自動執行，主要步驟如下：

- 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，節省 CPU 資源，及程式設計。

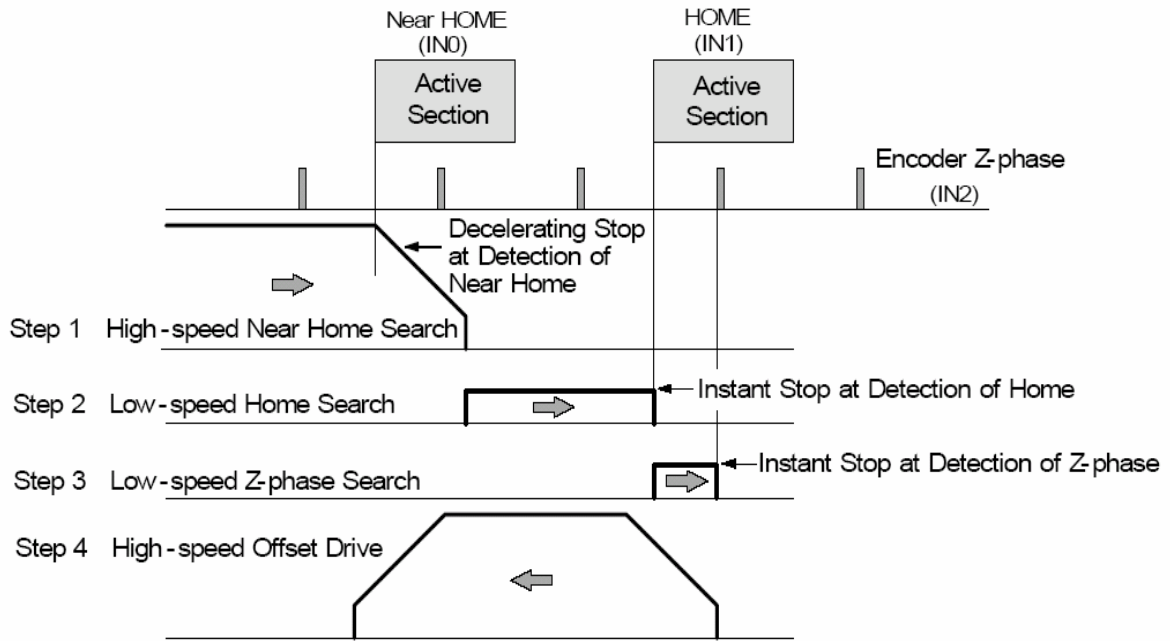
歸原點常常使用，當機器開機時，或系統發生警報，或信號錯誤時。上述情況使用者都能使用歸原點，讓機器回到原先的工作點。

i8094 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，四軸也能以相同方法處理。



X-軸硬體信號狀況





自動歸原點的樣板

**Note:** 如果你沒有硬體的近原點信號，你可以把近原點和原點信號接在同一 Pin，並且把 step1 除能。

## 5.1 設定軸歸零速度

- **void i8094MF\_SET\_HV**(**BYTE** *cardNo*, **WORD** *axis*, **DWORD** *data*)

功能： 設定軸之歸零速度。

參數：  
*cardNo*: 指定卡號  
*axis*: 指定軸號碼 (參考表 2-1)  
*data*: 設定速度值 (Vmin~Vmax PPS)

回應： 無

範例：  
`i8094MF_SET_HV(1, AXIS_X, 500);`  
`//設定第 1 卡 X 軸，歸零速度為 500 PPS。`

## 5.2 設定以極限當原點

- **void i8094MF\_HOME\_LIMIT**(**BYTE** *cardNo*, **WORD** *axis*, **WORD** *nType*)

功能： 設定軸之 Limit 開關當原點開關。

參數：  
*cardNo*: 指定卡號  
*axis*: 指定軸號碼 (參考表 2-1)  
*nType*: 設定 0=取消,1=啟用

回應： 無

範例：  
`i8094MF_HOME_LIMIT(1, AXIS_X, 0);`  
`//設定第 1 卡 X 軸，取消 Limit 開關當原點。`

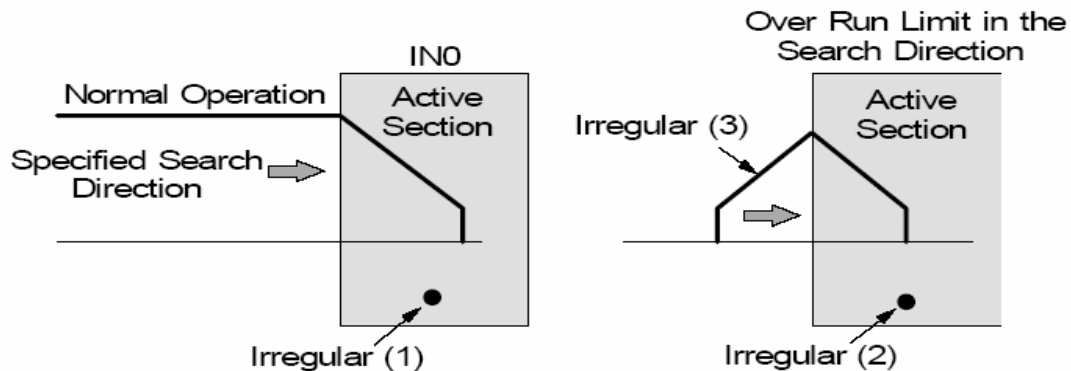
## 5.3 設定歸零模式

### ■ 每一步的操作

在每一步都有詳細的說明，在模式設定可以選擇是否執行或設定正負方向的尋找。如果選擇不執行，他將跳向下一步。

### 5.3.1 步驟一 高速尋找近原點

驅動的脈波是指定的方向，速度是設定的驅動速度(V)，一直到近原點信號(nIN0)改變，減速停止。執行高速尋找作業，先設定一個較高值的加減速驅動。



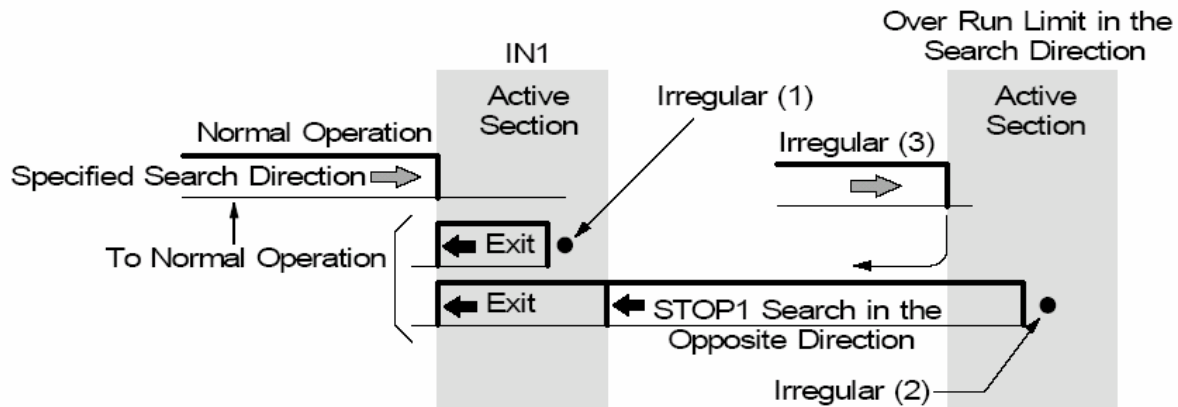
歸原點第一步

非正規的作業：

- (1) 一開始已經在近原點(nIN0)上了。  
→ 繼續進行第二步
- (2) 一開始已經發生碰觸到極限信號。  
→ 繼續進行第二步
- (3) 極限信號是在執行期間發生。  
→ 停止驅動並且繼續進行第二步

### 5.3.2 步驟二低速尋找原點

驅動的脈波以指定的方向輸出，偵測原點的速度是設定(HV)，一直到原點信號(nIN1)改變。執行低速尋找作業，先設定一個比初始速度(SV)還要低的值，做為偵測原點的速度(HV)，一個定速驅動模式適用於當偵測到原點(nIN1)能立即停止。



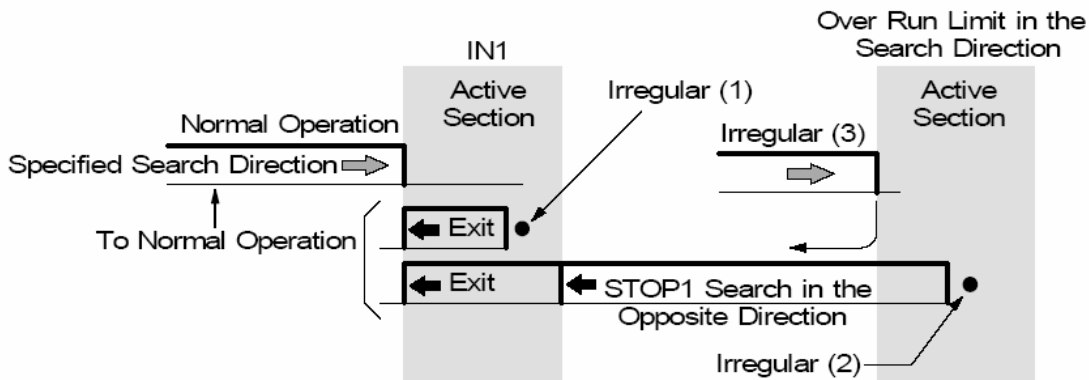
歸原點第二步

非正規的作業：

- (1) 一開始已經在原點(nIN1)上了。  
→ 這馬達驅動的軸將以定義的方向，反方向用歸原點的速度(HV)離開原點(nIN1)，在執行一次步驟二重新尋找原點
- (2) 一開始已經發生碰觸到極限信號。  
→ 這馬達驅動的軸將以定義的方向，反方向用歸原點的速度(HV)尋找原點(nIN1)。找到原點後再以馬達驅動的軸將以定義的方向，反方向用歸原點的速度(HV)離開原點(nIN1)，在執行一次步驟二重新尋找原點
- (3) 極限信號是在執行期間發生。  
→ 停止驅動並且和(2)執行步驟相同

### 5.3.3 步驟三低速尋找 Z-相

驅動的脈波是指定的方向，偵測Z-相的速度是設定(HV)，一直到編碼器Z-相信號(nIN2)改變。執行低速尋找作業，先設定一個比初始速度(SV)還要低的值，做為偵測編碼器Z-相信號(nIN2)的速度(HV)，一個定速驅動模式適用於當偵測到編碼器Z-相信號(nIN2)能立即停止。編碼器Z-相信號(nIN2)和原點信號(nIN1)的條件都能適用。



歸原點第三步

非正規的作業：

- (1) 一開始已經在編碼器Z-相(nIN2)上了。  
→ 當這個錯誤發生，自動歸原點結束。調整機械的系統，讓他一開始不會在編碼器Z-相(nIN2)上
- (2) 一開始已經發生碰觸到極限信號。  
→ 當這個錯誤發生，自動歸原點結束。
- (3) 極限信號是在執行期間發生。  
→ 尋找作業將中斷，自動歸原點結束。

### 5.3.4 步驟四高速補正驅動

這個功能輸出你所設定的脈波數(P)，用設定的驅動速度(V)，和定義的方向驅動。使用這個步驟讓軸從機械原點移動到作業原點的位置。透過模式的設定，邏輯位置計數器和真實位置計數器，能在移動後被清除。如果在執行期間或開始之前，發生碰觸極限信號，作業將中斷，自動歸原點結束。

- **void i8094MF\_SET\_HOME\_MODE**(**BYTE cardNo**, **WORD axis**, **WORD nStep1**, **WORD nStep2**, **WORD nStep3**, **WORD nStep4** , **long data**)

功能： 設定軸歸零方法及參數。

參數：  
**cardNo**: 指定卡號  
**axis**: 指定軸號碼 (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=正向補正,2=負向補正  
**data**: 補正值(0 ~ 2,147,483,647)

#### 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 (V)	近原點(IN0)
步驟 2	以低速尋找原點開關	歸零速度 (HV)	原點(IN1)
步驟 3	以低速尋找伺服馬達 Z 相信號	歸零速度 (HV)	Z 相信號(IN2)
步驟 4	以高速運動到位移值	驅動速度 (V)	

回應： 無

範例：  
i8094MF\_SET\_V(1, 0x1, 20000);  
i8094MF\_SET\_HV(1, 0x1, 500);  
i8094MF\_SET\_HOME\_MODE(1, 0x1, 2, 2, 1, 1, 3500);  
i8094MF\_HOME\_START(1, 0x1);  
i8094MF\_WAIT\_HOME(1, 0x1);  
//設定第 1 卡 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 (IN0) Low active	—	20000 (PPS) (V)
步驟 2	原點 (IN1) Low active	—	500 (PPS) (HV)
步驟 3	Z相信號 (IN2) High active	+	500 (PPS) (HV)
步驟 4	3500 pulse 補正(offset)	+	20000 (PPS) (V)

## 5.4 啟動軸歸零

- **void i8094MF\_HOME\_START(BYTE cardNo, WORD axis)**

功能： 設定軸開始執行軸歸零。

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

回應： 無

範例： **i8094MF\_HOME\_START(1, AXIS\_X);**  
*//設定第 1 卡 X 軸，開始執行軸歸零。*

## 5.5 等待完成歸零動作

- **BYTE i8094MF\_HOME\_WAIT(BYTE cardNo, WORD axis)**

功能： 等待軸歸零執行完成。

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

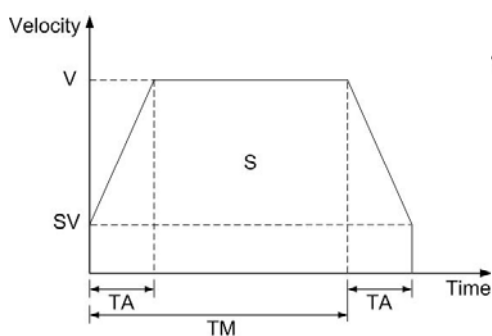
範例： **if (i8094MF\_HOME\_WAIT(1, AXIS\_X) == NO)**  
**{**  
*//第 1 卡 X 軸，歸零執行未完處理。*  
**}**

## 6 軸控功能

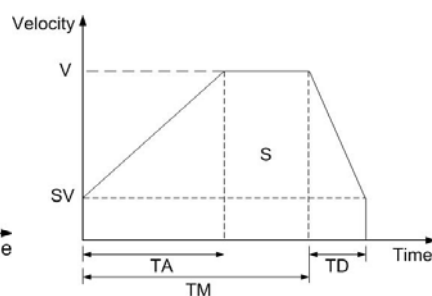
### 6.1 各軸獨立運動

- 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。
- 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。
- 也可以搭配補間運動或同步運動，做更複雜及多樣化的運動控制。

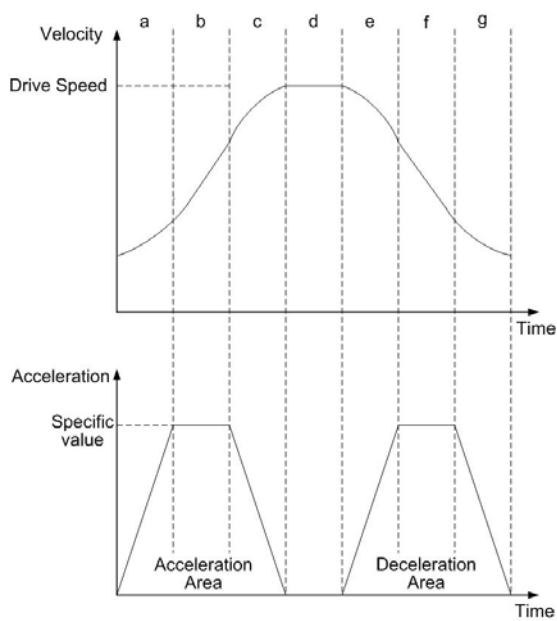
#### 6.1.1 設定加減速模式



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 i8094MF\_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)  
3 → 非對稱 S 曲線 (SV、V、K、L、AO)

回應： 無

範例： BYTE cardNo=1; //設定第 1 號卡。  
i8094MF\_SET\_MAX\_V(cardNo, AXIS\_XYZU, 20000); //設定軸最高速 20K PPS。

```
//=====
i8094MF_NORMAL_SPEED(cardNo, AXIS_XYZU, 0); //設定 XYZU 為對稱 T 曲線。
i8094MF_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094MF_SET_A(cardNo, AXIS_XYZU, 1000); //設定 XYZU 軸加速度 1000 PPS/S。
i8094MF_SET_SV(cardNo, AXIS_XYZU, 2000); //設定 XYZU 初始速度 2000 PPS。
i8094MF_SET_AO(cardNo, AXIS_XYZU, 9); //XYZU 軸減速(保留脈波數)= 9 PPS。
i8094MF_FIXED_MOVE(cardNo, AXIS_XYZU, 10000); //XYZU 移動 10000 Pulse。
```

```
//=====
i8094MF_NORMAL_SPEED(cardNo, AXIS_XYZU, 1); //設定 XYZU 軸對稱 S 曲線。
i8094MF_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094MF_SET_K(cardNo, AXIS_XYZU, 500); // XYZU 軸 K=500 PPS/S^2。
i8094MF_SET_SV(cardNo, AXIS_XYZU, 200); //設定 XYZU 軸初始速度=200 PPS。
i8094MF_SET_AO(cardNo, AXIS_XYZU, 9); //XYZU 軸減速(保留脈波數)= 9 PPS。
i8094MF_FIXED_MOVE(cardNo, AXIS_XYZU, -10000); //XYZU 移-10000 Pulse。
```

```
//=====
i8094MF_NORMAL_SPEED(cardNo, AXIS_XYZU, 2); //設定 XYZU 非對稱 T 曲線。
i8094MF_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094MF_SET_A(cardNo, AXIS_XYZU, 1000); //設定 XYZU 軸加速度 1000 PPS/S。
i8094MF_SET_D(cardNo, AXIS_XYZU, 500); //設定 XYZU 軸減速度=500 PPS/S。
i8094MF_SET_SV(cardNo, AXIS_XYZU, 200); //設定 XYZU 軸初始速度=200 PPS。
i8094MF_SET_AO(cardNo, AXIS_XYZU, 9); // XYZU 軸減速(保留脈波數)= 9 PPS。
i8094MF_FIXED_MOVE(cardNo, AXIS_XYZU, 10000); //XYZU 軸移動 10000
```

Pulse。

```
//=====
i8094MF_NORMAL_SPEED(cardNo, AXIS_XYZU, 3); // XYZU 軸為非對稱 S 曲線。
i8094MF_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094MF_SET_K(cardNo, AXIS_XYZU, 500); // XYZU 軸 K=500 PPS/S^2。
i8094MF_SET_L(cardNo, AXIS_XYZU, 300); // XYZU 軸 L=300 PPS/S^2。
i8094MF_SET_SV(cardNo, AXIS_XYZU, 200); //設定 XYZU 軸初始速度=200 PPS。
i8094MF_SET_AO(cardNo, AXIS_XYZU, 9); // XYZU 軸減速(保留脈波數)= 9 PPS。
i8094MF_FIXED_MOVE(cardNo, AXIS_XYZU, 10000); //XYZU 軸移 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 i8094MF\_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 2.5) PPS

回應： 無

範例：  
`i8094MF_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 i8094MF\_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 2.5) PPS

回應： 無

範例：  
`i8094MF_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 i8094MF_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)  
參考 2.5 所設定，`i8094MF_SET_MAX_V` → 最大速度值  
最小加速度單位值: 最大速度值 ÷ 64  
最大加速度值: 最大速度值 × 125

回應： 無

範例：  
`i8094MF_SET_MAX_V(1, AXIS_X, 20000);`  
//最小加速度單位值:  $20,000 \div 64 = 312.5 \times n \approx 313...625...938...。$   
//最大加速度值:  $20,000 \times 125 = 2,500,000。$   
`i8094MF_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 i8094MF_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)  
參考 2.5 所設定，`i8094MF_SET_MAX_V` → 最大速度值  
最小減速度單位值: 最大速度值 ÷ 64  
最大減速度值: 最大速度值 × 125

回應： 無

範例：  
`i8094MF_SET_MAX_V(1, AXIS_X, 20000);`  
//最小減速度單位值:  $20,000 \div 64 = 312.5 \times n \approx 313...625...938...。$   
//最大減速度值:  $20,000 \times 125 = 2,500,000。$   
`i8094MF_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 i8094MF\_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/ Sec<sup>2</sup>)  
參考 2.5 所設定，i8094MF\_SET\_MAX\_V → 最大速度值  
最小加速度變化率單位值: 最大速度值 × 0.0119211  
最大加速度變化率值: 4,294,967,295

回應： 無

範例：  
i8094MF\_SET\_MAX\_V(1, AXIS\_X, 20000);  
//最小加速度變化率單位值: 20,000 × 0.0119211 = 238.422 × n ≐ 238...476...。  
i8094MF\_SET\_K(1, AXIS\_X, 10000);  
//設定第 1 卡 X 軸，加速度變化率為 10,000 PPS/ Sec<sup>2</sup>。

## 6.1.7 設定軸減速度變化率

- **void i8094MF\_SET\_L(BYTE cardNo, WORD axis, DWORD data)**

功能： 設定軸之輸出減速度變化率。

參數：  
**cardNo:** 指定卡號  
**axis:** 指定軸號碼 (參考表 2-1)  
**data:** 設定減速度變化率值 (Jerk PPS/ Sec<sup>2</sup>)  
參考 2.5 所設定，i8094MF\_SET\_MAX\_V → 最大速度值  
最小減速度變化率單位值: 最大速度值 × 0.0119211  
最大減速度變化率值: 4,294,967,295

回應： 無

範例：  
i8094MF\_SET\_MAX\_V(1, AXIS\_X, 20000);  
//最小減速度變化率單位值: 20,000 × 0.0119211 = 238.422 × n ≐ 238...476...。  
i8094MF\_SET\_L(1, AXIS\_X, 10000);  
//設定第 1 卡 X 軸，減速度變化率為 10,000 PPS/ Sec<sup>2</sup>。

## 6.1.8 設定軸減速(保留脈波數)

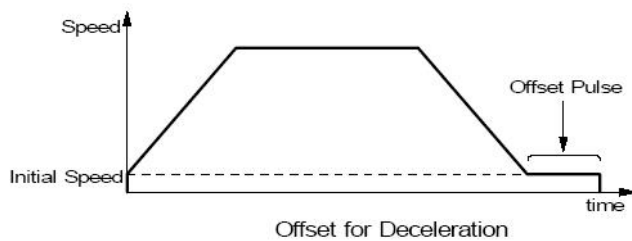
● **void i8094MF\_SET\_AO**(**BYTE cardNo**, **WORD axis**, **short int data**)

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: **i8094MF\_SET\_AO(1, AXIS\_X, 200);**  
**//設定第 1 卡 X 軸, Offset Pulse 為 200 Pulses。**



## 6.1.9 固定脈波數輸出

### ● BYTE i8094MF\_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(-2,147,483,648 ~ +2,147,483,647)

回應： YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8094MF\_GET\_ERROR\_CODE)  
NO: 沒有錯誤

範例： **BYTE cardNo=1; //設定第 1 號卡。**  
**i8094MF\_SET\_MAX\_V(cardNo, AXIS\_XYZU, 20000); //設定最高速 20K PPS。**  
**i8094MF\_NORMAL\_SPEED(cardNo, AXIS\_XYZU, 0); //設定 XYZU 軸對稱 T 曲線。**  
**i8094MF\_SET\_V(cardNo, AXIS\_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。**  
**i8094MF\_SET\_A(cardNo, AXIS\_XYZU,1000); //設定 XYZU 軸加速度 1000 PPS/S。**  
**i8094MF\_SET\_SV(cardNo, AXIS\_XYZU, 2000); //設定 XYZU 初始速度 2000 PPS。**  
**i8094MF\_SET\_AO(cardNo, AXIS\_XYZU, 9); //XYZU 軸減速(保留脈波數)= 9 PPS。**  
**i8094MF\_FIXED\_MOVE(cardNo, AXIS\_XYZU, 10000); //XYZU 移動 10000 Pulse。**

## 6.1.10 連續脈波輸出

### ● BYTE i8094MF\_CONTINUE\_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 i8094MF\_GET\_ERROR\_CODE)  
NO: 沒有錯誤

範例： **BYTE cardNo=1; //設定第 1 號卡。**  
**i8094MF\_SET\_MAX\_V(cardNo, AXIS\_XYZU, 20000); //設定最高速 20K PPS。**  
**i8094MF\_NORMAL\_SPEED(cardNo, AXIS\_XYZU, 0); //設定 XYZU 軸對稱 T 曲線。**  
**i8094MF\_SET\_V(cardNo, AXIS\_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。**  
**i8094MF\_SET\_A(cardNo, AXIS\_XYZU,1000); //設定 XYZU 軸加速度 1000 PPS/S。**  
**i8094MF\_SET\_SV(cardNo, AXIS\_XYZU, 2000); //設定 XYZU 初始速度 2000 PPS。**  
**i8094MF\_CONTINUE\_MOVE(cardNo, AXIS\_XYZU, 1000); //1K PPS 連續移動。**

## 6.2 補間運動

### 6.2.1 設定補間軸

- **void i8094MF\_AXIS\_ASSIGN(BYTE cardNo, WORD axis1, WORD axis2, WORD axis3)**

功能： 設定補間軸對象。

參數：  
**cardNo:** 指定卡號  
**axis1:** 指定第一軸號碼: X、Y、Z、U (1、2、4、8)  
**axis2:** 指定第二軸號碼: X、Y、Z、U (1、2、4、8)  
**axis3:** 指定第三軸號碼: 沒有(0) 或 X、Y、Z、U (1、2、4、8)

回應： 無

範例：  
**i8094MF\_AXIS\_ASSIGN(1, AXIS\_X, AXIS\_Y, 0);**  
*//設定第 1 卡 X 軸為第一軸，Y 軸為第二軸，做兩軸補間設定。*

## 6.2.2 設定補間加減速模式

### ● void i8094MF\_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 → 二軸直線非對稱 S 曲線 (VSV、VV、VK、VL、VAO)
- 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)
- 7 → 三軸直線固定向量速度 (VV)
- 8 → 三軸直線對稱 T 曲線 (VSV、VV、VA、VAO)
- 9 → 三軸直線對稱 S 曲線 (VSV、VV、VK、VAO)
- 10 → 三軸直線非對稱 T 曲線 (VSV、VV、VA、VD、VAO)
- 11 → 三軸直線非對稱 S 曲線 (VSV、VV、VK、VL、VAO)

回應： 無

範例： BYTE cardNo=1; //設定第 1 號卡。  
i8094MF\_SET\_MAX\_V(cardNo, AXIS\_XYZU, 20000); //設 XYZU 最高速 20K PPS

```
//=====
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, 0);
//設定第 1 卡 X 軸為第一軸，Y 軸為第二軸，做兩軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 0);
//二軸(直線&弧&圓)固定向量速度 VSV=VV，設 VV 即可。
i8094MF_SET_VV(cardNo, 1000); //設定第 1 卡，向量定速度為 1000 PPS。
i8094MF_LINE_2D(cardNo, 12000, 10000); //執行向量 2D 補間。
```

```
//=====
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, 0);
//設定 X 軸為第一軸，Y 軸為第二軸，做兩軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 1);
//二軸直線對稱 T 曲線 (VSV、VV、VA、VAO)。
i8094MF_SET_VSV(cardNo, 500); //設定向量初始速度為 500 PPS。
i8094MF_SET_VV(cardNo, 2000); //設定向量速度為 2000 PPS。
i8094MF_SET_VA(cardNo, 1000); //設定向量加速速度為 1000 PPS。
i8094MF_LINE_2D(cardNo, 20000, 10000); //執行向量 2D 補間。
```

```
//=====
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, 0);
//設定 X 軸為第一軸，Y 軸為第二軸，做兩軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 2);
//二軸直線對稱 S 曲線(VSV、VV、VA、VK、AO)。
i8094MF_SET_VSV(cardNo, 200); //設定向量初始速度為 500 PPS。
i8094MF_SET_VV(cardNo, 2000); //設定向量速度為 2000 PPS。
i8094MF_SET_VK(cardNo, 500); //設定 VK=500 PPS/S^2。
```



i8094MF\_SET\_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。  
i8094MF\_LINE\_2D(cardNo, 10000, 10000); //執行向量 2D 補間。

```
//=====
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, 0);
//設定 X 軸為第一軸，Y 軸為第二軸，做兩軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 3);
//二軸直線非對稱 T 曲線 (VSV、VV、VA、VD、VAO)。
i8094MF_SET_VSV(cardNo, 100); //設定向量初始速度為 500 PPS。
i8094MF_SET_VV(cardNo, 2000); //設定向量速度為 2000 PPS。
i8094MF_SET_VA(cardNo, 1000); //設定向量加速度為 1000 PPS/s。
i8094MF_SET_VD(cardNo, 500); //設定向量減速度為 500 PPS/s。
i8094MF_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094MF_LINE_2D(cardNo, 10000, 5000); //執行向量 2D 補間。
```

```
//=====
long fp1=4000;
long fp2=10000;
unsigned short sv=200;
unsigned short v=2000;
i8094MF_SET_MAX_V(cardNo, AXIS_XYZU, 8000);
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, 0);
//設定 X 軸為第一軸，Y 軸為第二軸，做兩軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 4);
//二軸直線非對稱 S 曲線 (VSV、VV、VK、VL、VAO)。
i8094MF_SET_VSV(cardNo, sv); //設定向量初始速度為 sv PPS。
i8094MF_SET_VV(cardNo, v); //設定向量速度為 v PPS。
i8094MF_SET_VK(cardNo, 500); //設定 VK=500 PPS/S^2。
i8094MF_SET_VL(cardNo, 300); //設定 VL=300 PPS/S^2。
i8094MF_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094MF_LINE_2D(cardNo, fp1, fp2); //執行向量 2D 補間。
```

```
//=====
long fp1=11000;
long fp2=9000;
long c1=10000;
long c2=0;
unsigned short sv=100;
unsigned short v=3000;
unsigned long a=5000;
unsigned long d=5000;
i8094MF_SET_MAX_V(cardNo, AXIS_XYZU, 8000);
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, 0);
//設定 X 軸為第一軸，Y 軸為第二軸，做兩軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 5);
//二軸(弧&圓)對稱 T 曲線 (VSV、VV、VA、VAO)。
i8094MF_SET_VSV(cardNo, sv); //設定向量初始速度為 sv PPS。
i8094MF_SET_VV(cardNo, v); //設定向量速度為 v PPS。
i8094MF_SET_VA(cardNo, a); //設定向量加速度為 a PPS/s。
i8094MF_SET_VAO(cardNo, 0); //設定軸向量減速(保留脈波數)0 Pulse。
i8094MF_ARC_CW(cardNo, c1, c2, fp1, fp2); //執行二軸順時針圓弧補間。
```

```

//=====
long c1=300;
long c2=0;
unsigned short sv=100;
unsigned short v=3000;
unsigned long a=125;
unsigned long d=12;
i8094MF_SET_MAX_V(cardNo, AXIS_XYZU, 8000);
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, 0);
//設定 X 軸為第一軸，Y 軸為第二軸，做兩軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 6);
//二軸(弧&圓)對稱 T 曲線 (VSV、VV、VA、VAO)。
i8094MF_SET_VSV(cardNo, sv); //設定向量初始速度為 sv PPS。
i8094MF_SET_VV(cardNo, v); //設定向量速度為 v PPS。
i8094MF_SET_VA(cardNo, a); //設定向量加速度為 a PPS/s。
i8094MF_SET_VD(cardNo, d); //設定向量減速度為 d PPS/s。
i8094MF_SET_VAO(cardNo, 0); //設定軸向量減速(保留脈波數)0 Pulse。
i8094MF_CIRCLE_CW(cardNo, c1, c2); //執行二軸順時針圓形補間。

//=====
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, AXIS_Z);
//設定第 1 卡 X 軸為第一軸，Y 軸為第二軸，Z 軸為第三軸，做三軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 7);
//三軸直線固定向量速度 (VSV=VV)。
i8094MF_SET_VSV(cardNo, 1000); //設定向量初始速度為 1000 PPS。
i8094MF_SET_VV(cardNo, 1000); //設定向量速度為 1000 PPS。
i8094MF_LINE_3D(cardNo, 10000, 10000,10000); //執行向量 3D 補間。

//=====
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, AXIS_Z);
//設定第 1 卡 X 軸為第一軸，Y 軸為第二軸，Z 軸為第三軸，做三軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 8);
//三軸直線對稱 T 曲線 (VSV、VV、VA、VAO)。
i8094MF_SET_VSV(cardNo, 100); //設定向量初始速度為 1000 PPS。
i8094MF_SET_VV(cardNo, 3000); //設定向量速度為 3000 PPS。
i8094MF_SET_VA(cardNo, 500); //設定向量加速度為 500 PPS/s。
i8094MF_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094MF_LINE_3D(cardNo, 10000, 1000,20000); //執行向量 3D 補間。

//=====
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, AXIS_Z);
//設定第 1 卡 X 軸為第一軸，Y 軸為第二軸，Z 軸為第三軸，做三軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 9);
//三軸直線對稱 S 曲線 (VSV、VV、VK、VAO)
i8094MF_SET_VSV(cardNo, 100); //設定向量初始速度為 1000 PPS。
i8094MF_SET_VV(cardNo, 3000); //設定向量速度為 3000 PPS。
i8094MF_SET_VK(cardNo, 500); //設定 VK=500 PPS/S^2。
i8094MF_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094MF_LINE_3D(cardNo, 10000, 1000,1000); //執行向量 3D 補間。

```

```
//=====
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, AXIS_Z);
//設定第 1 卡 X 軸為第一軸，Y 軸為第二軸，Z 軸為第三軸，做三軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 10);
//三軸直線非對稱 T 曲線 (VSV、VV、VA、VD、VAO)。
i8094MF_SET_VSV(cardNo, 100); //設定向量初始速度為 1000 PPS。
i8094MF_SET_VV(cardNo, 2000); //設定向量速度為 3000 PPS。
i8094MF_SET_VA(cardNo, 1000); //設定向量加速度為 1000 PPS/s。
i8094MF_SET_VD(cardNo, 500); //設定向量減速度為 500 PPS/s。
i8094MF_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094MF_LINE_3D(cardNo, 10000, 1000,1000); //執行向量 3D 補間。
```

```
//=====
long fp1=4000;
long fp2=10000;
long fp3=20000;
unsigned short sv=200;
unsigned short v=2000;
i8094MF_SET_MAX_V(cardNo, AXIS_XYZU, 8000);
i8094MF_AXIS_ASSIGN(cardNo, AXIS_X, AXIS_Y, AXIS_Z);
//設定第 1 卡 X 軸為第一軸，Y 軸為第二軸，Z 軸為第三軸，做三軸補間設定。
i8094MF_VECTOR_SPEED(cardNo, 11);
//三軸直線非對稱 S 曲線 (VSV、VV、VK、VL、VAO)。
i8094MF_SET_VSV(cardNo, sv); //設定向量初始速度為 sv PPS。
i8094MF_SET_VV(cardNo, v); //設定向量速度為 v PPS。
i8094MF_SET_VK(cardNo, 500); //設定 VK=500 PPS/S^2。
i8094MF_SET_VL(cardNo, 300); //設定 VL=300 PPS/S^2。
i8094MF_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094MF_LINE_3D(cardNo, fp1, fp2,fp3); //執行向量 3D 補間。
```

備註： 請搭配設定相關向量速度參數.....。

## 6.2.3 設定軸向量初始速度

● **void i8094MF\_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 2.5) PPS

回應： 無

範例： **i8094MF\_SET\_VSV(1, 1000);**  
**//設定第 1 卡，向量初始速度為 1000 PPS。**

## 6.2.4 設定軸向量定速度

● **void i8094MF\_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 2.5) PPS

回應： 無

範例： **i8094MF\_SET\_VV(1, 120000L);**  
**//設定第 1 卡，向量定速度為 120000 PPS。**

## 6.2.5 設定軸向量加速度

- `void i8094MF_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)  
參考 2.5 所設定，`i8094MF_SET_MAX_V` → 最大速度值  
最小向量加速度單位值: 最大速度值 ÷ 64  
最大向量加速度值: 最大速度值 × 125

回應： 無

範例：  
`i8094MF_SET_MAX_V(1, AXIS_X, 20000);`  
//最小向量加速度單位值:  $20,000 \div 64 = 312.5 \times n \doteq 313...625...938...。$   
//最大向量加速度值:  $20,000 \times 125 = 2,500,000。$   
`i8094MF_SET_VA(1, 100000L);`  
//設定第 1 卡 X 軸，向量加速度為 100K PPS/Sec。

## 6.2.6 設定軸向量減速度

- `void i8094MF_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)  
參考 2.5 所設定，`i8094MF_SET_MAX_V` → 最大速度值  
最小向量減速度單位值: 最大速度值 ÷ 64  
最大向量減速度值: 最大速度值 × 125

回應： 無

範例：  
`i8094MF_SET_MAX_V(1, AXIS_X, 20000);`  
//最小向量減速度單位值:  $20,000 \div 64 = 312.5 \times n \doteq 313...625...938...。$   
//最大向量減速度值:  $20,000 \times 125 = 2,500,000。$   
`i8094MF_SET_VD(1, 100000L);`  
//設定第 1 卡 X 軸，向量減速度為 100K PPS/Sec。

## 6.2.7 設定軸向量加速度變化率

- **void i8094MF\_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/ Sec<sup>2</sup>)  
參考 2.5 所設定，i8094MF\_SET\_MAX\_V → 最大速度值  
最小向量加速度變化率單位值: 最大速度值 × 0.0119211  
最大向量加速度變化率值: 4,294,967,295

回應： 無

範例： i8094MF\_SET\_MAX\_V(1, AXIS\_X, 20000);  
//最小向量加速度變化率單位值: 20,000×0.0119211=238.422 × n ≐238...476...。  
i8094MF\_SET\_VK(1, 10000);  
//設定第 1 卡 X 軸，向量加速度變化率為 10,000 PPS/ Sec<sup>2</sup>。

## 6.2.8 設定軸向量減速度變化率

- **void i8094MF\_SET\_VL(BYTE cardNo, DWORD data)**

功能： 設定軸之輸出向量減速度變化率。

參數： **cardNo:** 指定卡號  
**data:** 設定向量減速度變化率值 (Jerk PPS/ Sec<sup>2</sup>)  
參考 2.5 所設定，i8094MF\_SET\_MAX\_V → 最大速度值  
最小向量減速度變化率單位值: 最大速度值 × 0.0119211  
最大向量減速度變化率值: 4,294,967,295

回應： 無

範例： i8094MF\_SET\_MAX\_V(1, AXIS\_X, 20000);  
//最小向量減速度變化率單位值: 20,000×0.0119211=238.422 × n ≐238...476...。  
i8094MF\_SET\_VL(1, 10000);  
//設定第 1 卡 X 軸，向量減速度變化率為 10,000 PPS/ Sec<sup>2</sup>。

## 6.2.9 設定軸向量減速(保留脈波數)

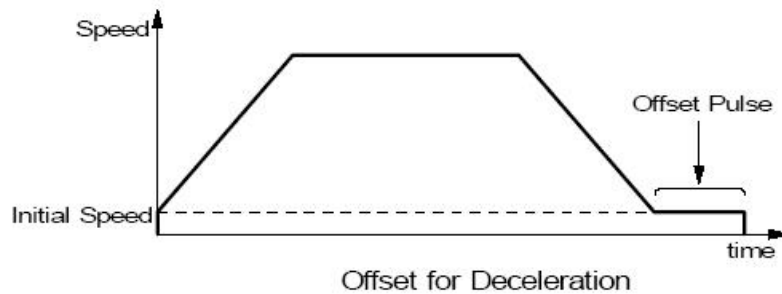
- **void i8094MF\_SET\_VAO(BYTE cardNo, short int data)**

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: **i8094MF\_SET\_VAO(1, 200);**  
**//設定第 1 卡補間軸, Offset Pulse 為 200 Pulses。**



## 6.2.10 二軸直線補間

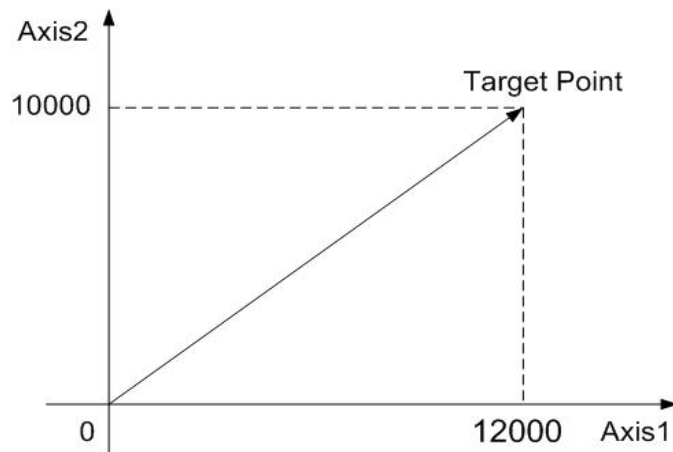
- **BYTE** `i8094MF_LINE_2D`(**BYTE** `cardNo`, **long** `fp1`, **long** `fp2`)

功能： 執行二軸直線補間。

參數：  
**cardNo**: 指定卡號  
**fp1**: 指定第一軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)  
**fp2**: 指定第二軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)

回應：  
**YES**: 有錯誤發生(欲讀錯誤碼請搭配使用 `i8094MF_GET_ERROR_CODE`)  
**NO**: 沒有錯誤

範例：  
`i8094MF_LINE_2D(1, 12000, 10000);`  
*//設定第 1 卡，執行二軸直線補間。*



二軸直線補間



## 6.2.11 三軸直線補間

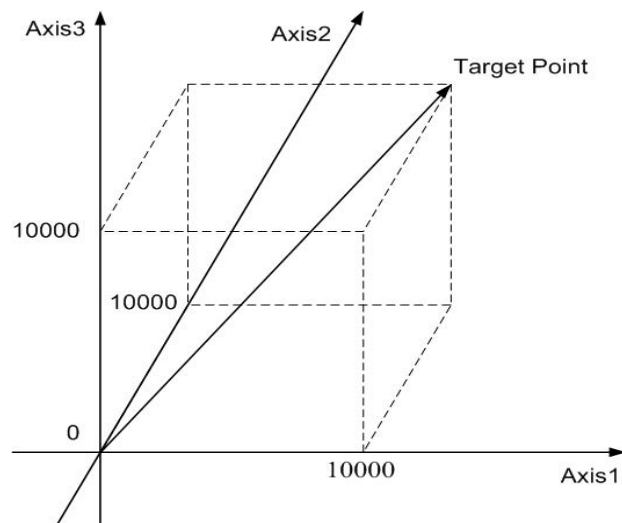
### ● BYTE i8094MF\_LINE\_3D(BYTE cardNo, long fp1, long fp2, long fp3)

功能： 執行三軸直線補間。

參數：  
**cardNo:** 指定卡號  
**fp1:** 指定第一軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)  
**fp2:** 指定第二軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)  
**fp3:** 指定第三軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)

回應：  
**YES:** 有錯誤發生(欲讀錯誤碼請搭配使用 i8094MF\_GET\_ERROR\_CODE)  
**NO:** 沒有錯誤

範例：  
`i8094MF_LINE_3D(1, 10000, 10000, 10000);`  
`//設定第 1 卡，執行三軸直線補間。`



三軸直線補間

## 6.2.12 二軸圓弧補間

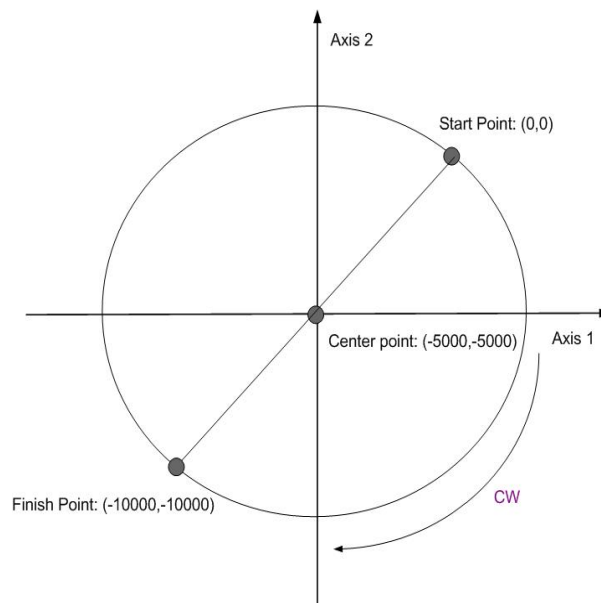
● **BYTE** `i8094MF_ARC_CW`(**BYTE** `cardNo`, **long** `cp1`, **long** `cp2`, **long** `fp1`, **long** `fp2`)

功能： 執行二軸順時針圓弧補間。

參數：  
**cardNo**: 指定卡號  
**cp1**: 指定第一軸圓弧中心相對位置(-2,147,483,648 ~ +2,147,483,647)  
**cp2**: 指定第二軸圓弧中心相對位置(-2,147,483,648 ~ +2,147,483,647)  
**fp1**: 指定第一軸圓弧終點相對位置(-2,147,483,648 ~ +2,147,483,647)  
**fp2**: 指定第二軸圓弧終點相對位置(-2,147,483,648 ~ +2,147,483,647)

回應： **YES**: 有錯誤發生(欲讀錯誤碼請搭配使用 `i8094MF_GET_ERROR_CODE`)  
**NO**: 沒有錯誤

範例：  
`i8094MF_ARC_CW(1, -5000, -5000, -10000, -10000);`  
//設定第 1 卡，執行二軸順時針圓弧補間。



二軸順時針圓弧補間

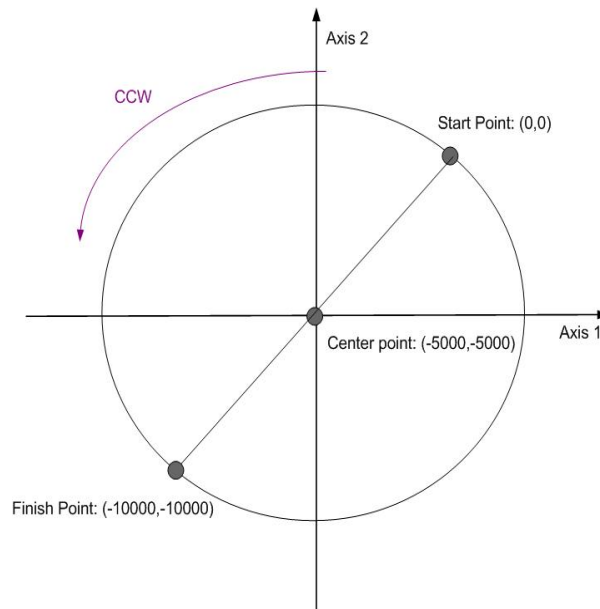
● **BYTE i8094MF\_ARC\_CCW**(**BYTE cardNo**, **long cp1**, **long cp2**,  
**long fp1**, **long fp2**)

功能： 執行二軸逆時針圓弧補間。

參數：  
**cardNo:** 指定卡號  
**cp1:** 指定第一軸圓弧中心相對位置  
**cp2:** 指定第二軸圓弧中心相對位置  
**fp1:** 指定第一軸圓弧終點相對位置  
**fp2:** 指定第二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 **i8094MF\_GET\_ERROR\_CODE**)  
**NO:** 沒有錯誤

範例：  
**i8094MF\_ARC\_CCW(1, -5000, -5000, -10000, -10000);**  
*//設定第 1 卡，執行二軸逆時針圓弧補間。*



二軸逆時針圓弧補間

### 6.2.13 二軸圓形補間

- **BYTE** `i8094MF_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*: 指定第一軸圓弧中心相對位置(-2,147,483,648 ~ +2,147,483,647)  
*cp2*: 指定第二軸圓弧中心相對位置(-2,147,483,648 ~ +2,147,483,647)

回應： YES: 有錯誤發生(欲讀錯誤碼請搭配使用 `i8094MF_GET_ERROR_CODE`)  
NO: 沒有錯誤

範例： `i8094MF_CIRCLE_CW(1, 0, 10000);`  
`//設定第 1 卡，執行二軸順時針圓形補間。`

- **BYTE** `i8094MF_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*: 指定第一軸圓弧中心相對位置  
*cp2*: 指定第二軸圓弧中心相對位置

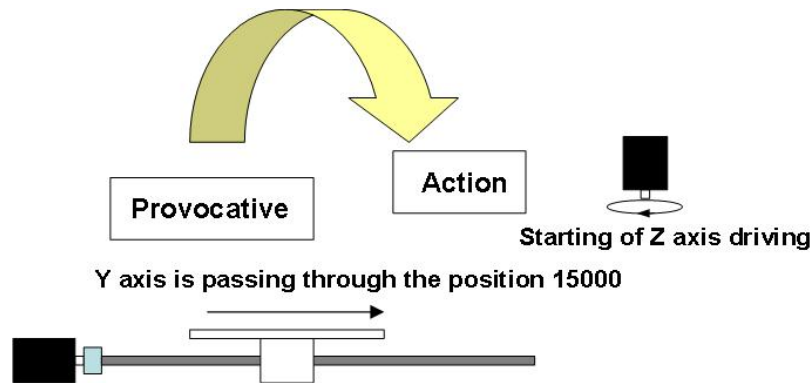
回應： YES: 有錯誤發生(欲讀錯誤碼請搭配使用 `i8094MF_GET_ERROR_CODE`)  
NO: 沒有錯誤

範例： `i8094MF_CIRCLE_CCW(1, 0, 10000);`  
`//設定第 1 卡，執行二軸逆時針圓形補間。`

## 6.3 同步運動

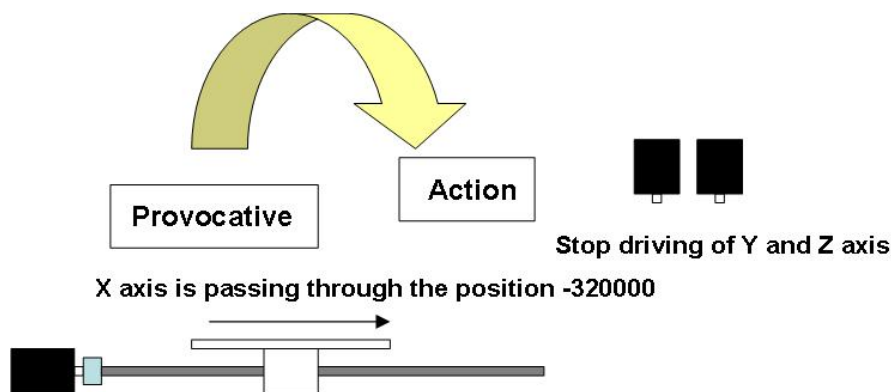
同步運動這是IC執行的運動，像是每一軸和軸、IC和IC之間、或外部裝置，都能由內部去操控啟動和停止。如下範例的動作都能被執行。

範例 1: 在Y軸通過15000這個位置後，Z軸跟著啟動。



同步運動範例 1

範例 2: 在X軸通過-320000這個位置後，Y軸和Z軸驅動停止。



同步運動範例 2

正常寫一個程式在 CPU 這邊都能執行同步運動，不論如何這功能是有幫助的，當 CPU 軟體在執行時，時間不能延遲是必要的。同步運動是 IC 內部一個執行插入的功能，它能實現高精度的同步。

### 6.3.1 設定同步運動條件

- `void i8094MF_SYNC_ACTION(BYTE cardNo, WORD axis1, WORD axis2, WORD nSYNC, WORD nDRV, WORD nLATCH, WORD nPRESET)`

功能： 同步運動條件的設定。

參數： **cardNo:** 指定卡號  
**axis1:** 指定主軸號碼 X 或 Y 或 Z 或 U (1 or 2 or 4 or 8)

**axis2:** 指定同步運動軸號碼，如下表說明

<b>axis1</b> <b>axis2</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	<b>U</b>
<b>0</b>	無	無	無	無
<b>1</b>	Y	Z	U	X
<b>2</b>	Z	U	X	Y
<b>3</b>	YZ	ZU	UX	XY
<b>4</b>	U	X	Y	Z
<b>5</b>	YU	ZX	UY	XZ
<b>6</b>	ZU	UX	XY	YZ
<b>7</b>	YZU	ZUX	UXY	XYZ

**nSYNC:** 同步運動條件因子，可複選，如下表說明

號碼	代號	說明
0x0000		除能同步運動條件因子
0x0001	P ≥ C+	同步運動發生於邏輯或真實位置計數器的值超過 COMP+ 暫存器的值 <b>必須和 i8094MF_SET_COMPARE 並用</b>
0x0002	P < C+	同步運動發生於邏輯或真實位置計數器的值小於 COMP+ 暫存器的值 <b>必須和 i8094MF_SET_COMPARE 並用</b>
0x0004	P < C-	同步運動發生於邏輯或真實位置計數器的值小於 COMP- 暫存器的值 <b>必須和 i8094MF_SET_COMPARE 並用</b>
0x0008	P ≥ C-	同步運動發生於邏輯或真實位置計數器的值超過 COMP- 暫存器的值 <b>必須和 i8094MF_SET_COMPARE 並用</b>
0x0010	D-STA	同步運動發生於驅動開始時
0x0020	D-END	同步運動發生於驅動結束時
0x0040	IN3 ↑	同步運動發生於 nIN3 信號正邊緣觸發從低到高準位
0x0080	IN3 ↓	同步運動發生於 nIN3 信號負邊緣觸發從高到低準位

例：選 P ≥ C+ 和 IN3 ↑ (0x0001 + 0x0040 = 0x0041)

**nDRV:** 同步驅動項目，如下表說明

號碼	代號	說明
0		取消同步驅動
1	FDRV+	正方向固定脈波驅動，對 <b>移動中</b> 之軸設定無效 步數設定需搭配 <b>OPSET</b> 設定新的位置
2	FDRV-	負方向固定脈波驅動，對 <b>移動中</b> 之軸設定無效 步數設定需搭配 <b>OPSET</b> 設定新的位置
3	CDRV+	正方向連續脈波驅動，對 <b>移動中</b> 之軸設定無效
4	CDRV-	負方向連續脈波驅動，對 <b>移動中</b> 之軸設定無效
5	SSTOP	減速停止
6	ISTOP	直接停止

**nLATCH:** 同步驅動項目，如下表說明

號碼	代號	說明
0		取消同步位置門鎖
1	LPSAV	儲存目前邏輯位置計數器(LP)，[LP → LATCH]
2	EPSAV	儲存目前真實位置計數器(EP)，[EP → LATCH]

必須和 **i8094MF\_GET\_LATCH** 並用。

**nPRESET:** 同步資料設定項目，如下表說明

號碼	代號	說明
0		取消同步資料設定
1	LPSET	設定新的邏輯位置計數器(LP)，[LP ← PRESET]
2	EPSET	設定新的真實位置計數器(EP)，[EP ← PRESET]
3	OPSET	設定新的位置(P)，[P ← PRESET] <b>連續運動指令 CONTINUE_MOVE</b> 之軸無法設定新的位置
4	VLSET	設定新的速度(V)，[V ← PRESET]

必須和 **i8094MF\_SET\_PRESET** 並用。

回應: 無

範例: //範例 1. 當 U 軸 IN3 收到正邊緣觸發信號，便改變移動速度及 LATCH encoder 的值。

```
i8094MF_SYNC_ACTION(cardNo, AXIS_U, 0, 0X0040, 0, 2, 4);  
i8094MF_SET_MAX_V(cardNo, AXIS_U, 5000); //設定 U 軸最高速 5K PPS。  
i8094MF_NORMAL_SPEED(cardNo, AXIS_U, 0); //設定 U 軸為對稱 T 曲線。  
i8094MF_SET_V(cardNo, AXIS_U, 2000); //設定 U 軸速度=2,000 PPS。  
i8094MF_SET_A(cardNo, AXIS_U, 100000); //設定 U 軸加速度=100K PPS/S。  
i8094MF_SET_SV(cardNo, AXIS_U, 100); //設定 U 軸初始速度=100 PPS。  
i8094MF_FIXED_MOVE(cardNo, AXIS_U, 10000); //設定 U 軸移動 10,000 Pulse。  
i8094MF_SET_PRESET(cardNo, AXIS_U, 100); //設定 U 軸新的速度值=100 PPS。
```

```
while (i8094MF_STOP_WAIT(cardNo, AXIS_U) == NO)  
{  
    //第 cardNo 卡 U 軸運動尚未停止，處理程序。  
    DoEvents();  
    Sleep(1);  
};  
long Vsb = i8094MF_GET_LATCH(cardNo, AXIS_U);
```

```
//範例 2. 當 U 軸 EP 的值超過 COMP+(5,000)的值，便啟動 Y 軸移動 2,000 PPS。  
i8094MF_SYNC_ACTION(cardNo, AXIS_U, 2, 0X0001, 1, 0, 3);  
i8094MF_SET_COMPARE(cardNo, AXIS_U, 0, 1, 5000);  
//設定 COMP+的值=5,000，來源參考 U 軸 EP。  
i8094MF_SET_MAX_V(cardNo, AXIS_YU, 9000); //設定 YU 軸最高速 9K PPS。  
i8094MF_NORMAL_SPEED(cardNo, AXIS_YU, 0); //設定 YU 軸為對稱 T 曲線。  
i8094MF_SET_V(cardNo, AXIS_YU, 3000); //設定 YU 軸速度=3,000 PPS。  
i8094MF_SET_A(cardNo, AXIS_YU, 200000); //設定 YU 軸加速度=200K PPS/S。  
i8094MF_SET_SV(cardNo, AXIS_YU, 200); //設定 YU 軸初始速度=200 PPS。  
i8094MF_FIXED_MOVE(cardNo, AXIS_U, 10000); //設定 U 軸移動 10,000 Pulse。  
i8094MF_SET_PRESET(cardNo, AXIS_Y, 2000); //設定 Y 軸移動 2,000 PPS。
```



## 6.3.2 設定 COMPARE 值

- **void** i8094MF\_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：  
i8094MF\_SET\_COMPARE(cardNo, AXIS\_U, 0, 1, 5000);  
//設定 COMP+的值=5,000，來源參考 U 軸 EP。

## 6.3.3 讀取 LATCH 值

- **long** i8094MF\_GET\_LATCH(**BYTE** cardNo, **WORD** axis)

功能： 讀取同步位置門鎖值。

參數：  
**cardNo:** 指定卡號  
**axis:** 指定軸號碼 X 或 Y 或 Z 或 U (1 or 2 or 4 or 8)

回應： 位置門鎖值 -2,147,483,648 ~ +2,147,483,647

範例：  
**long** data = i8094MF\_GET\_LATCH(1, AXIS\_Y);  
//讀取第 1 卡 Y 軸，同步位置門鎖值。

## 6.3.4 設定 PRESET 資料

● **void i8094MF\_SET\_PRESET(BYTE cardNo, WORD axis, long data)**

功能： 選擇同步資料設定方式。

參數：  
**cardNo:** 指定卡號  
**axis:** 指定同步資料設定軸號碼(參考表 2-1)  
**data:** LP: -2,147,483,648 ~ +2,147,483,647  
EP: -2,147,483,648 ~ +2,147,483,647  
P : -2,147,483,648 ~ +2,147,483,647  
V : 最大值請參考 2.5

回應： 無

範例： 請參考 6.3.1 之範例

## 6.4 連續補間運動

連續補間運動功,有些是軟體功能的巨集函式,會耗用系統資源,執行完成才會繼續執行下一行指令,在即時系統中使用時請務必小心。

連續補間運動如被異常中斷而停止,請參考 6.5.5 排除!

## 6.4.1 二軸矩形連續補間

- **BYTE** i8094MF\_RECTANGLE(  
**BYTE** cardNo, **WORD** axis1, **WORD** axis2,  
**WORD** nAcc, **WORD** Sp, **WORD** nDir, **long** Lp, **long** Wp, **long** Rp,  
**DWORD** RSV, **DWORD** RV, **DWORD** RA, **DWORD** RD)

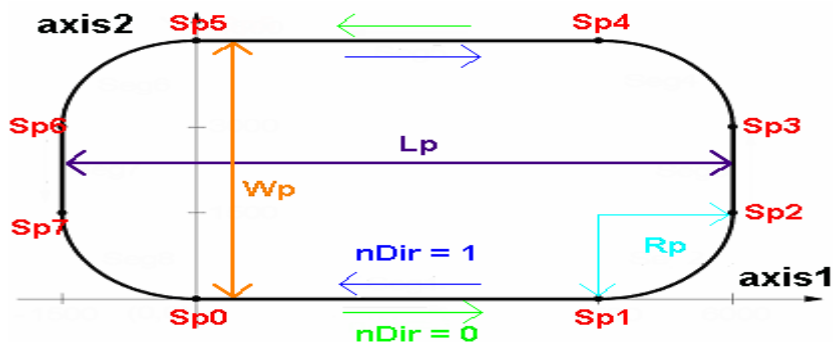
功能： 執行二軸矩形補間。  
 (軟體功能的巨集函式，會耗用系統資源)

參數：

<b>cardNo:</b>	指定卡號
<b>axis1:</b>	指定第一軸號碼: X、Y、Z、U (1、2、4、8)
<b>axis2:</b>	指定第二軸號碼: X、Y、Z、U (1、2、4、8)
<b>nAcc:</b>	0 → 定速度補間 1 → 對稱 T 曲線加減速補間
<b>Sp:</b>	設定起點 0 ~ 7 (Sp0 ~ Sp7 如下圖所示)
<b>nDir:</b>	設定方向 0、1 (CCW or CW)
<b>Lp:</b>	設定長度 Pulse 數(1 ~ 2,147,483,647)
<b>Wp:</b>	設定寬度 Pulse 數(1 ~ 2,147,483,647)
<b>Rp:</b>	設定圓弧半徑 Pulse 數(1 ~ 2,147,483,647)
<b>RSV:</b>	設定矩形補間向量起始速度(PPS)
<b>RV:</b>	設定矩形補間向量速度(PPS)
<b>RA:</b>	設定矩形補間向量加速度(PPS/Sec)
<b>RD:</b>	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 i8094MF\_GET\_ERROR\_CODE)  
 NO: 沒有錯誤

範例：  
**unsigned short** sv=1000; //設定向量初始速度為 1000 PPS。  
**unsigned short** v=10000; //設定向量速度為 10000 PPS。  
**unsigned long** a=5000; //設定向量加速度為 5000 PPS/s。  
**unsigned long** d=5000; //設定向量減速度為 5000 PPS/s。  
 i8094MF\_SET\_MAX\_V(1, AXIS\_XYZU, 16000); //最高速度為 16000 PPS。  
 i8094MF\_RECTANGLE(1, AXIS\_X, AXIS\_Y, 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。



## 6.4.2 二軸直線連續補間

- **void i8094MF\_LINE\_2D\_INITIAL**(**BYTE cardNo**, **WORD axis1**, **WORD axis2**, **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：

<b>cardNo:</b>	指定卡號
<b>axis1:</b>	指定第一軸號碼: X、Y、Z、U (1、2、4、8)
<b>axis2:</b>	指定第二軸號碼: X、Y、Z、U (1、2、4、8)
<b>VSV:</b>	設定向量初始速度(PPS)
<b>VV:</b>	設定向量速度(PPS)
<b>VA:</b>	設定加速度(PPS/Sec)

回應： 無

範例： 和 **void i8094MF\_LINE\_2D\_CONTINUE**(**BYTE cardNo**, **WORD nType**, **long fp1**, **long fp2**) 並用。

- **BYTE** i8094MF\_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:** 指定第一軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)  
**fp2:** 指定第二軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)

回應: **YES:** 有錯誤發生(欲讀錯誤碼請搭配使用 i8094MF\_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=18000; //設定向量速度為 PPS。  
**unsigned long** a=500000; //設定向量加速度為 PPS/s。  
**unsigned short** loop1;  
i8094MF\_SET\_MAX\_V(cardNo, AXIS\_XYZU,160000L);  
i8094MF\_LINE\_2D\_INITIAL(cardNo, AXIS\_X, AXIS\_Y, sv, v, a);  
**for** (loop1 = 0; loop1 < 10000; loop1++)  
{  
    i8094MF\_LINE\_2D\_CONTINUE (cardNo, 0, 100, 100);  
    i8094MF\_LINE\_2D\_CONTINUE (cardNo, 0, -100, -100);  
}  
i8094MF\_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。

### 6.4.3 三軸直線連續補間

- **void i8094MF\_LINE\_3D\_INITIAL**(**BYTE cardNo**, **WORD axis1**, **WORD axis2**, **WORD axis3**, **DWORD VSV**, **DWORD VV**, **DWORD VA**)

功能： 三軸直線連續補間初始設定(對稱 T 曲線加減速)。

參數：

<b>cardNo:</b>	指定卡號
<b>axis1:</b>	指定第一軸號碼: X、Y、Z、U (1、2、4、8)
<b>axis2:</b>	指定第二軸號碼: X、Y、Z、U (1、2、4、8)
<b>axis3:</b>	指定第三軸號碼: X、Y、Z、U (1、2、4、8)
<b>VSV:</b>	設定向量初始速度(PPS)
<b>VV:</b>	設定向量速度(PPS)
<b>VA:</b>	設定加速度(PPS/Sec)

回應： 無

範例： 和 **void i8094MF\_LINE\_3D\_CONTINUE**(**BYTE cardNo**, **WORD nType**, **long fp1**, **float fp2**, **float fp3**) 並用。

● **BYTE i8094MF\_LINE\_3D\_CONTINUE**(**BYTE cardNo**, **WORD nType**, **long fp1**,  
**long fp2**, **long fp3**)

功能: 執行三軸直線連續補間。  
(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 → 三軸直線連續補間  
1 → 三軸直線連續補間結束  
**fp1**: 指定第一軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)  
**fp2**: 指定第二軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)  
**fp3**: 指定第三軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)

回應: **YES**: 有錯誤發生(欲讀錯誤碼請搭配使用 **i8094MF\_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=18000**; //設定向量速度為 PPS。  
**unsigned long a=500000**; //設定向量加速度為 PPS/s。  
**unsigned short loop1**;  
**i8094MF\_SET\_MAX\_V(cardNo, AXIS\_XYZU,160000L)**;  
**i8094MF\_LINE\_3D\_INITIAL(cardNo, AXIS\_X, AXIS\_Y, AXIS\_Z, sv, v, a)**;  
**for** (**loop1 = 0**; **loop1 < 10000**; **loop1++**)  
{  
    **i8094MF\_LINE\_3D\_CONTINUE(cardNo, 0, 100, 100, 100)**;  
    **i8094MF\_LINE\_3D\_CONTINUE(cardNo, 0, -100, -100, -100)**;  
}  
**i8094MF\_LINE\_3D\_CONTINUE(cardNo, 1, 100, 100, 100)**;  
//設定第 1 卡，執行 X、Y、Z 三軸直線連續補間運動。



## 6.4.4 二軸混合連續補間

- **void i8094MF\_MIX\_2D\_INITIAL**(**BYTE cardNo**, **WORD axis1**, **WORD axis2**, **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：

<b>cardNo:</b>	指定卡號
<b>axis1:</b>	指定第一軸號碼: X、Y、Z、U (1、2、4、8)
<b>axis2:</b>	指定第二軸號碼: X、Y、Z、U (1、2、4、8)
<b>nAcc:</b>	0 → 定速度補間 (VV) 1 → 對稱 T 曲線加減速補間 (VSV、VV、VA)
<b>VSV:</b>	設定向量初始速度(PPS)
<b>VV:</b>	設定向量速度(PPS)
<b>VA:</b>	設定加速度(PPS/Sec)

回應： 無

範例： 和 **void i8094MF\_MIX\_2D\_CONTINUE**(**BYTE cardNo**, **WORD nAcc**, **WORD nType**, **long cp1**, **long cp2**, **long fp1**, **long fp2**)並  
用。

● **BYTE i8094MF\_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。  
(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 → 連續補間  
 1 → 結束連續補間減速停止(定速度不需減速)  
**nType**: 1 → i8094MF\_LINE\_2D(BYTE cardNo, long fp1, long fp2)  
 2 → i8094MF\_ARC\_CW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2)  
 3 → i8094MF\_ARC\_CCW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2)  
 4 → i8094MF\_CIRCLE\_CW(BYTE cardNo, long cp1, long cp2)  
 5 → i8094MF\_CIRCLE\_CCW(BYTE cardNo, long cp1, long cp2)  
**cp1**: 指定第一軸圓、弧中心相對位置(-2,147,483,648 ~ +2,147,483,647)  
**cp2**: 指定第二軸圓、弧中心相對位置(-2,147,483,648 ~ +2,147,483,647)  
**fp1**: 指定第一軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)  
**fp2**: 指定第二軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)

回應: YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8094MF\_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=18000**; //設定向量速度為 PPS。  
**unsigned long a=500000**; //設定向量加速度為 PPS/s。  
**unsigned short loop1**;  
**i8094MF\_SET\_MAX\_V**(cardNo, AXIS\_XYZU,160000L);  
**i8094MF\_MIX\_2D\_INITIAL**(cardNo, AXIS\_X, AXIS\_Y, 1, sv, v, a);  
**for** (loop1 = 0; loop1 < 10000; loop1++)  
 {  
     **i8094MF\_MIX\_2D\_CONTINUE** (cardNo, 0, 1, 0, 0, 100, 100);  
     **i8094MF\_MIX\_2D\_CONTINUE** (cardNo, 0, 2, 100, 0, 100, 100);  
 }  
**i8094MF\_MIX\_2D\_CONTINUE** (cardNo, 1, 4, 100, 100, 0, 0);  
 //設定第 1 卡，執行 X、Y 兩軸連續補間運動。

## 6.4.5 多點連續補間(陣列資料)

- **BYTE** i8094MF\_CONTINUE\_INTP(  
**BYTE** cardNo, **WORD** axis1, **WORD** axis2, **WORD** axis3,  
**WORD** nAcc, **DWORD** VSV, **DWORD** VV, **DWORD** VA, **DWORD** VD,  
**BYTE** nType[ ], **long** cp1[ ], **long** cp2[ ], **long** fp1[ ], **long** fp2[ ], **long**  
fp3[ ])

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
axis1:	指定第一軸號碼: X、Y、Z、U (1、2、4、8)
axis2:	指定第二軸號碼: X、Y、Z、U (1、2、4、8)
axis3:	指定第三軸號碼: X、Y、Z、U (1、2、4、8)
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 → i8094MF_LINE_2D(BYTE cardNo, long fp1, long fp2) 2 → i8094MF_ARC_CW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2) 3 → i8094MF_ARC_CCW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2) 4 → i8094MF_CIRCLE_CW(BYTE cardNo, long cp1, long cp2) 5 → i8094MF_CIRCLE_CCW(BYTE cardNo, long cp1, long cp2) 6 → i8094MF_LINE_3D(BYTE cardNo, long fp1, long fp2, long fp3) 7 → 連續補間結束
cp1[ ]:	指定第一軸圓、弧中心相對位置(-2,147,483,648 ~ +2,147,483,647)
cp2[ ]:	指定第二軸圓、弧中心相對位置(-2,147,483,648 ~ +2,147,483,647)
fp1[ ]:	指定第一軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647) 指定第一軸圓弧終點相對位置
fp2[ ]:	指定第二軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647) 指定第二軸圓弧終點相對位置
fp3[ ]:	指定第三軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647) (二軸和三軸無法混合使用, 所有未使用數值請填 0)

回應: YES: 有錯誤發生(欲讀錯誤碼請搭配使用 i8094MF\_GET\_ERROR\_CODE)  
NO: 沒有錯誤

```

範例:  BYTE cardNo=1; //設定第 1 號卡。
        unsigned short sv=100; //設定向量初始速度為 100 PPS。
        unsigned short v=3000; //設定向量速度為 3000 PPS。
        unsigned long a=2000; //設定向量加速度為 2000 PPS/s。
        unsigned long d=2000; //設定向量減速度為 2000 PPS/s。
        i8094MF_SET_MAX_V(cardNo, AXIS_XYZU, 20000); //設各軸最高速度 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};
        long fp3[10]= { 0, 0, 0, 0, 0,0,0,0,0,0};
        i8094MF_CONTINUE_INTP(
        cardNo, AXIS_X, AXIS_Y, 0, 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。

```

## 6.4.6 三軸螺旋運動

- **BYTE** i8094MF\_HELIX\_3D(  
**BYTE** cardNo, **WORD** axis1, **WORD** axis2, **WORD** axis3, **WORD** nDir,  
**DWORD** VV, **long** cp1, **long** cp2, **long** cycle, **long** pitch)

功能: 執行螺旋運動(定速)。  
(軟體功能的巨集函式, 會耗用系統資源)

參數: **cardNo**: 指定卡號  
**axis1**: 指定圓形運動第一軸號碼: X、Y、Z、U (1、2、4、8)  
**axis2**: 指定圓形運動第二軸號碼: X、Y、Z、U (1、2、4、8)  
**axis3**: 指定同動單軸號碼: X、Y、Z、U (1、2、4、8)  
**nDir**: 0 → 圓形運動 CW  
1 → 圓形運動 CCW  
**VV**: 設定螺旋向量速度(PPS)  
**cp1**: 指定第一軸圓中心相對位置(-2,147,483,648 ~ +2,147,483,647)  
**cp2**: 指定第二軸圓中心相對位置(-2,147,483,648 ~ +2,147,483,647)  
**cycle**: 圓形運動循環次數  
**pitch**: 單軸節距(-2,147,483,648 ~ +2,147,483,647)

回應: **YES**: 有錯誤發生(欲讀錯誤碼請搭配使用 i8094MF\_GET\_ERROR\_CODE)  
**NO**: 沒有錯誤

範例: **BYTE** cardNo=1; //設定第 1 號卡。

```
//=====
i8094MF_SET_MAX_V(cardNo, AXIS_XYZU,160000L); //設最高速度為 16K PPS。
long v=50000; //設定向量速度為 PPS。
i8094MF_HELIX_3D(cardNo, AXIS_Y, AXIS_Z, AXIS_X, 1, v, 0, 1000, 5, -2000);
//設定第 1 卡, 執行 Y、Z 兩軸圓形運動補間, X 軸同動跟隨。
```

```
//=====
i8094MF_SET_MAX_V(cardNo, AXIS_XYZU,160000L); //設最高速度為 16K PPS。
long v=100000; //設定向量速度為 PPS。
i8094MF_HELIX_3D(cardNo, AXIS_Y, AXIS_Z, AXIS_U, 1, v, 0, 25000, 50, 3600);
//設定第 1 卡, 執行 Y、Z 兩軸圓形運動補間, U 軸同動跟隨。
```

## 6.4.7 二軸比例運動

- **void i8094MF\_RATIO\_INITIAL**(**BYTE** *cardNo*, **WORD** *axis1*, **WORD** *axis2*, **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  
**axis1**: 指定比例運動第一軸號碼: X、Y、Z、U (1、2、4、8)  
**axis2**: 指定比例運動第二軸號碼: X、Y、Z、U (1、2、4、8)  
**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 i8094MF\_RATIO\_2D**(**BYTE** *cardNo*, **WORD** *nType*, **long** *data*, **WORD** *nDir*) 並用。

- **BYTE** i8094MF\_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:** 比例運動第一軸 Pulse 數(-2,147,483,648 ~ +2,147,483,647)  
**nDir:** 比例運動第二軸方向:  
 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 i8094MF\_GET\_ERROR\_CODE)  
**NO:** 沒有錯誤

範例: **BYTE** cardNo=1; //設定第 1 號卡。  
**unsigned short** sv=300; //設定初始速度為 PPS。  
**unsigned short** v=18000; //設定速度為 PPS。  
**unsigned long** a=500000; //設定加速度為 PPS/s。  
**unsigned short** loop1;  
**unsigned short** loop2;  
i8094MF\_SET\_MAX\_V(cardNo, AXIS\_XYZU,160000L);  
i8094MF\_RATIO\_INITIAL(cardNo, AXIS\_U, AXIS\_X, sv, v, a, 0.36f);  
**for** (loop2 = 0; loop2 < 5; loop2++)  
{  
  **for** (loop1 = 0; loop1 < 5; loop1++)  
  {  
    i8094MF\_RATIO\_2D(cardNo, 0, 3600, 0);  
    i8094MF\_RATIO\_2D(cardNo, 0, 3600, 1);  
  }  
  i8094MF\_RATIO\_2D(cardNo, 0, 7200, 0);  
  i8094MF\_RATIO\_2D(cardNo, 0, 3600, 1);  
}  
i8094MF\_RATIO\_2D(cardNo, 1, 7200, 0);  
//設定第 1 卡，執行 U、X 兩軸比例運動。

## 6.5 中斷條件因子設定

### 6.5.1 中斷條件設定

● **void i8094MF\_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	中斷發生於每一個脈波正緣產生時
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 <b>i8094MF_SET_COMPARE</b> 並用 (6.3.2)
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 <b>i8094MF_SET_COMPARE</b> 並用(6.3.2)
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 <b>i8094MF_SET_COMPARE</b> 並用(6.3.2)
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 <b>i8094MF_SET_COMPARE</b> 並用(6.3.2)
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 i8094_hThread; //中斷執行緒變數宣告
DWORD WINAPI i8094_ThreadFunction(LPVOID IParam); //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 CI8094QCDlg::OnTestint()
{
    DWORD dwThreadID = 0;
    HWND hWnd = NULL;
    //建立 i8094_ThreadFunction 執行緒函數
    i8094_hThread = CreateThread(NULL, 0, i8094_ThreadFunction, hWnd, 0,
    &dwThreadID);
    //設置 4 軸(XYZU)
    BYTE axis=AXIS_XYZU;
    i8094MF_SET_MAX_V(CardNo, axis, 20000);
    i8094MF_NORMAL_SPEED(CardNo, axis, 0);
    i8094MF_SET_V(CardNo, axis, 20000);
    i8094MF_SET_A(CardNo, axis, 100000);
}
```



```

i8094MF_SET_SV(CardNo, axis, 20000);
i8094MF_SET_AO(CardNo, axis, 0);
//中斷功能初使化
hINTP=Slot_Register_Interrupt(Slot1);
//設置中斷條件: D-END
i8094MF_INTFACTOR_ENABLE(CardNo, AXIS_X, 7);
// 4 軸定量驅動
i8094MF_FIXED_MOVE(CardNo, AXIS_XYZU, 10000);

    while (i8094MF_STOP_WAIT(CardNo, 0xf) == NO)
    { // 等待運動軸停止
        DoEvents();
        Sleep(1);
    }
}

// 中斷執行緒函數
DWORD WINAPI i8094_ThreadFunction(LPVOID IParam)
{
    DWORD dwEvent;
    WORD RR3_X;
    if(hINTP != 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
                i8094MF_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 = i8094_GET_RR3(CardNo, AXIS_X);
                //關閉中斷條件因子
                i8094MF_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 , i8094  
**模組** 的運動中斷功能需搭配此三各函數使用。

## 6.5.2 中斷條件關閉

- **void i8094MF\_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

## 6.5.3 讀取中斷狀態

- **WORD i8094MF\_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
0x001	PULSE	中斷發生於每一個脈波正緣產生時 (設定為正緣觸發時, 參考表 2.2)
0x002	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
0x004	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
0x008	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
0x010	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
0x020	C-END	中斷發生於等速段和補正段結束兩個位置
0x040	C-STA	中斷發生於等速段和補正段開始兩個位置
0x080	D-END	中斷發生於驅動結束時

範例： **i8094MF\_GET\_RR3(cardNo, AXIS\_X);**  
**//取回 X 軸的主要狀態暫存器(RR3)的值。**

## 6.6 其他功能

### 6.6.1 設定軸暫停

- **void i8094MF\_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

### 6.6.2 設定軸啟動

- **void i8094MF\_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。**

**i8094MF\_DRV\_HOLD(cardNo, AXIS\_XYU); //設定 XYU 三軸暫停移動。**

**i8094MF\_SET\_MAX\_V(cardNo, AXIS\_U, 10000); //設定 U 軸最高速 10K PPS。**

**i8094MF\_NORMAL\_SPEED(cardNo, AXIS\_U, 0); //設定 U 軸對稱 T 曲線。**

**i8094MF\_SET\_V(cardNo, AXIS\_U, 2000); //設定 U 軸速度=2,000 PPS。**

**i8094MF\_SET\_A(cardNo, AXIS\_U, 1000); //設定 U 軸加速度 1,000 PPS/S。**

**i8094MF\_SET\_SV(cardNo, AXIS\_U, 2000); //設定 U 初始速度 2,000 PPS。**

**i8094MF\_SET\_AO(cardNo, AXIS\_U, 9); // U 軸減速(保留脈波數)= 9 PPS。**

**i8094MF\_SET\_MAX\_V(cardNo, AXIS\_XY, 20000); //設定 XY 最高速 20K PPS。**

**i8094MF\_AXIS\_ASSIGN(cardNo, AXIS\_X, AXIS\_Y, 0);**

**//設定第 1 卡 X 軸為第一軸，Y 軸為第二軸，做兩軸補間設定。**

**i8094MF\_VECTOR\_SPEED(cardNo, 0); //二軸直線固定速度 VSV=VV，設 VV 即可。**

**i8094MF\_SET\_VV(cardNo, 5000); //設定第 1 卡，向量定速度為 5,000 PPS。**

**i8094MF\_FIXED\_MOVE(cardNo, AXIS\_U, 5000); //U 移動 5,000 Pulse。**

**i8094MF\_LINE\_2D(cardNo, 12000, 10000); //執行直線補間移動。**

**i8094MF\_DRV\_START(cardNo, AXIS\_XYU); //開始 XYU 三軸同時移動。**

### 6.6.3 等待完成軸運動

- **BYTE** i8094MF\_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。  
i8094MF\_SET\_MAX\_V(cardNo, AXIS\_XYZU, 20000); //設定最高速 20K PPS。  
i8094MF\_NORMAL\_SPEED(cardNo, AXIS\_XYZU, 0); //設定 XYZU 軸對稱 T 曲線。  
i8094MF\_SET\_V(cardNo, AXIS\_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。  
i8094MF\_SET\_A(cardNo, AXIS\_XYZU, 1000); //設定 XYZU 軸加速度 1000 PPS/S。  
i8094MF\_SET\_SV(cardNo, AXIS\_XYZU, 2000); //設定 XYZU 初始速度 2000 PPS。  
i8094MF\_SET\_AO(cardNo, AXIS\_XYZU, 9); //XYZU 軸減速(保留脈波數)= 9 PPS。  
i8094MF\_FIXED\_MOVE(cardNo, AXIS\_XYZU, 10000); //XYZU 移動 10000 Pulse。

```
if (i8094MF_STOP_WAIT(cardNo, AXIS_X) == NO)
{
    //第 cardNo 卡 X 軸運動尚未停止，處理程序。
}
```

### 6.6.4 設定(補間)軸停止

- **void** i8094MF\_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： i8094MF\_STOP\_SLOWLY(1, AXIS\_XY);  
//設定第 1 卡 XY 軸，減速停止。

- **void** i8094MF\_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： i8094MF\_STOP\_SUDDENLY(1, AXIS\_ZU);  
//設定第 1 卡 ZU 軸，立即(緊急)停止。

● **void i8094MF\_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： **i8094MF\_VSTOP\_SLOWLY(1);**  
*//設定第 1 卡補間軸，減速停止。*

● **void i8094MF\_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： **i8094MF\_VSTOP\_SUDDENLY(1);**  
*//設定第 1 卡補間軸，立即(緊急)停止。*

● **void i8094MF\_SSTOP\_SLOWLY(BYTE cardNo, WORD axis)**

功能： 指定軸之輸出減速停止。  
這與 **i8094MF\_STOP\_SLOWLY( )** 的差別在於 **i8094MF\_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： **i8094MF\_SSTOP\_SLOWLY(1, AXIS\_XY);**  
*//設定第 1 卡 XY 軸，減速停止。*

● **void i8094MF\_SSTOP\_SUDDENLY**(**BYTE cardNo**, **WORD axis**)

功能: 指定軸之輸出立即(緊急)停止。  
這與 **i8094MF\_STOP\_SUDDENLY ( )**的差別在於 **i8094MF\_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: **i8094MF\_SSTOP\_SUDDENLY(1, AXIS\_ZU);**  
**//設定第 1 卡 Z U 軸，立即(緊急)停止。**

● **void i8094MF\_SVSTOP\_SLOWLY**(**BYTE cardNo**)

功能: 指定補間軸之輸出減速停止。  
這與 **i8094MF\_VSTOP\_SLOWLY ( )**的差別在於 **i8094MF\_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: **i8094MF\_SVSTOP\_SLOWLY(1);**  
**//設定第 1 卡補間軸，減速停止。**

● **void i8094MF\_SVSTOP\_SUDDENLY**(**BYTE cardNo**)

功能: 指定補間軸之輸出立即(緊急)停止。  
這與 **i8094MF\_VSTOP\_SUDDENLY ( )**的差別在於 **i8094MF\_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: **i8094MF\_SVSTOP\_SUDDENLY(1);**  
**//設定第 1 卡補間軸，立即(緊急)停止。**

## 6.6.5 清除停止狀態

### ● void i8094MF\_CLEAR\_STOP(BYTE cardNo)

功能: 使用 6.5.4 功能，使軸運動停止，請故障排除後，清除停止狀態。

參數: **cardNo:** 指定卡號

回應: 無

範例: `i8094MF_VSTOP_SUDDENLY(1);`  
`i8094MF_CLEAR_STOP(1);`  
`//清除第 1 卡錯誤狀態。`

## 6.6.6 補間動作結束

### ● void i8094MF\_INTP\_END(BYTE cardNo, WORD type)

功能: 1. 改做單軸運動或改變補間運動座標系，請於動作前下此指令。  
2. 也可所有軸之 **MAX\_V** 重新定義，就不用執行 INTP\_END。

參數: **cardNo:** 指定卡號  
**type:** 0 → 二軸補間  
1 → 三軸補間

回應: 無

範例: `i8094MF_INTP_END(1, 0);`  
`//設定第 1 卡二軸補間，動作結束。`