

i-8094H/i-8094A 運動控制模組使用

手冊

(Version 1.3)

應用程式函式庫
WinCon-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 1997-2007 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	8
1.1 手冊的使用	8
1.2 函式基本結構說明	8
1.3 函式在使用上的分類說明	9
2 基本設定功能	14
2.1 各軸定義的代碼	14
2.2 註冊軸卡及版本讀取	15
2.3 軸卡重置	16
2.4 軸輸出PULSE模式設定	17
2.5 設定軸速度輸出最大範圍	18
2.6 設定軸前後極限的輸入觸發邏輯	18
2.7 設定碰觸前後極限的處理模式	19
2.8 設定軸近原點輸入觸發邏輯	19
2.9 設定軸原點輸入觸發邏輯	20
2.10 設定軸前後軟體極限,參考來源及取消	20
2.11 設定編碼器輸入參數	21
2.12 伺服開關(Servo_ON/OFF)	21
2.13 設定伺服馬達異常ALARM輸入參數	22

2.14	設定伺服馬達定位完成輸入參數	22
2.15	設定數mm位輸入雜訊濾波功能.....	23
2.16	指定軸為圓形運動軸(環狀計數器).....	24
2.17	三角形速度曲線的預防	25
2.18	外部輸入驅動	26
2.18.1	手輪脈波驅動.....	26
2.18.2	固定脈波驅動.....	27
2.18.3	連續脈波驅動.....	27
2.18.4	外部輸入關閉.....	28
2.19	讀寫自定義變數(VAR).....	28
2.20	讀寫斷電保持資料(MD)	30
3	狀態讀取及設定功能.....	31
3.1	設定及讀取指令邏輯位置	31
3.2	設定及讀取ENCODER位置	32
3.3	讀取目前速度	33
3.4	讀取目前加速度	33
3.5	讀取目前DI狀態	34
3.6	讀取目前ERROR狀態	36
3.7	讀取目前RTC狀態	38
4	FRNET功能	39
4.1	FRnet DI讀取	39

4.2 FRnet DO寫入	39
5 軸自動歸零	40
5.1 設定軸歸零速度	40
5.2 設定以極限當原點	40
5.3 設定歸零模式	41
5.4 啟動軸歸零	42
5.5 等待完成歸零動作	42
6 軸控功能	43
6.1 各軸獨立運動	43
6.1.1 設定加減速模式	43
6.1.2 設定軸初始速度	44
6.1.3 設定軸定速度	45
6.1.4 設定軸加速度	45
6.1.5 設定軸減速度	46
6.1.6 設定軸加速度變化率	46
6.1.7 設定軸減速度變化率	47
6.1.8 設定軸減速(保留脈波數)	47
6.1.9 固定脈波數輸出	48
6.1.10 連續脈波輸出	49
6.2 補間運動	50
6.2.1 設定補間軸	50
6.2.2 設定補間加減速模式	50
6.2.3 設定軸向量初始速度	54
6.2.4 設定軸向量定速度	55
6.2.5 設定軸向量加速度	55
6.2.6 設定軸向量減速度	56
6.2.7 設定軸向量加速度變化率	56
6.2.8 設定軸向量減速度變化率	57

6.2.9 設定軸向量減速(保留脈波數).....	57
6.2.10 二軸直線補間.....	58
6.2.11 三軸直線補間.....	59
6.2.12 二軸圓弧補間.....	60
6.2.13 二軸圓形補間.....	62
6.3 同步運動.....	63
6.3.1 設定同步運動條件.....	63
6.3.2 設定COMPARE值	66
6.3.3 讀取LATCH值.....	66
6.3.4 設定PRESET資料.....	67
6.3.5 設定OUT資料	67
6.3.6 控制器對軸卡(i8094H)中斷功能	68
6.3.7 軸卡(i8094H)對控制器中斷功能	71
6.4 客製特殊應用	73
6.4.1 二軸矩形連續補間.....	73
6.4.2 二軸直線連續補間.....	74
6.4.3 三軸直線連續補間.....	76
6.4.4 二軸混合連續補間.....	77
6.4.5 多點連續補間(陣列資料).....	79
6.4.6 三軸螺旋運動.....	81
6.4.7 二軸比例運動.....	82
6.4.8 同步輸出Line Scan運動.....	83
6.4.9 三軸圓弧補間.....	86
6.4.10 三軸混合連續補間.....	87
6.5 其他功能.....	89
6.5.1 設定軸暫停	89
6.5.2 設定軸啓動	89
6.5.3 等待完成軸運動	90
6.5.4 設定(補間)軸停止.....	91
6.5.5 清除停止狀態.....	92
6.5.6 補間動作結束.....	92
7 I8094H專用特殊功能	93
7.1 初始化參數表	95

7.2 巨集程式的創建	96
7.2.1 建立巨集程式碼	96
7.2.2 執行巨集程式.....	97
7.2.3 自定義變數(Variable)	98
7.2.4 程序循環(FOR~NEXT)	100
7.2.5 程序條件(IF~ELSE).....	101
7.2.6 延時器(TIMER).....	103
7.2.7 等待軸運動停止(MP專用)	103
7.2.8 使用者定義RINT	104

1 前言

1.1 手冊的使用

- i8094H 是 i8094F 功能進階版的運動控制模組，在硬體部份：1.內建 CPU 去處理更複雜的運動控制動作。2.內建 Dual Port Ram 提供控制命令 30 層深度的緩衝。在軟體部份亦提供多樣化兼具彈性的設計環境，本手冊將完整且詳細的說明，幫助你很快的找到所需運動控制函式，並配合簡單的範例，迅速開發你的應用程式。
- 手冊分為七大章，本章是手冊的前言，2、3、4、5、6 五章為 i8094H 函式的說明(這部份和 i8094F 函式的定義大致雷同)，第七章內容為 i8094H 專用的特殊巨集函式。
- 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 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 函式在使用上的分類說明

RTC (Real Time Command)：能直接下給 i8094H 執行的函式。

MP (Macro Program)：在 MP_CREATE 後，能使用的函式。

ISR (Interrupt Service Routine)：在 MP_CREATE 後，能使用於 ISR 的函式。

IT (Initial Table)：可以使用在下載參數表中的函式。

ISR1 ~ ISR20 和 MP1 ~ MP157 所能容納的 Function Line

ISR(6)	ISR1	ISR2	ISR3	ISR4	ISR5	ISR6			
<i>Total:</i>	8	8	8	8	8	8			
ISR(9)	ISR7	ISR8	ISR9	ISR10	ISR11	ISR12	ISR13	ISR14	ISR15
<i>Total:</i>	16	16	16	16	16	16	16	16	16
ISR(3)	ISR16	ISR17	ISR18						
<i>Total:</i>	32	32	32						
ISR(2)	ISR19	ISR20							
<i>Total:</i>	64	64							
MP(40)	MP1	~	MP40						
<i>Total:</i>	8		8						
MP(50)	MP41	~	MP90						
<i>Total:</i>	16		16						
MP(40)	MP91	~	MP130						
<i>Total:</i>	32		32						
MP(20)	MP131	~	MP150						
<i>Total:</i>	64		64						
MP(5)	MP151	MP152	MP153	MP154	MP155				
<i>Total:</i>	128	128	128	128	128				
MP(2)	MP156	MP157							
<i>Total:</i>	512	512							

如下 Function 的列表，第 2、3、4、5、6 五章大部份的函式，都能使用在 i8094H_MP_CREATE(請參考 7.2.1)後，所有的值都能以變數取代(用於 MP 或 ISR 時):

bvarNo: 自定義變數: bVAR0 ~ bVAR127 (所有 BYTE 資料型別)

varNo: 自定義變數: VAR0 ~ VAR511 (所有 long 或 DWORD 資料型別)

注意: 在下列章節將以 * 標示可用於 MP 的函式。

在下列章節將以 * 標示可用於 ISR 的函式。

章節	函式	RTC	MP	ISR	IT
版本讀取					
2.2	i8094H_GET_FW_VERSION	⊙			
基本設定功能					
2.3	i8094H_RESET_CARD	⊙			
2.3	i8094H_CLEAR_CARD_BUFFER	⊙			
2.4	i8094H_SET_PULSE_MODE	⊙	⊙		⊙
2.5	i8094H_SET_MAX_V	⊙	⊙		⊙
2.6	i8094H_SET_HLMT	⊙	⊙		⊙
2.7	i8094H_LIMITSTOP_MODE	⊙	⊙		⊙
2.8	i8094H_SET_NHOME	⊙	⊙		⊙
2.9	i8094H_SET_HOME_EDGE	⊙	⊙		⊙
2.10	i8094H_SET_SLMT	⊙	⊙		⊙
2.10	i8094H_CLEAR_SLMT	⊙	⊙		⊙
2.11	i8094H_SET_ENCODER	⊙	⊙		⊙
2.12	i8094H_SERVO_ON	⊙	⊙		⊙
2.12	i8094H_SERVO_OFF	⊙	⊙		⊙
2.13	i8094H_SET_ALARM	⊙	⊙		⊙
2.14	i8094H_SET_INPOS	⊙	⊙		⊙
2.15	i8094H_SET_FILTER	⊙	⊙		⊙
2.16	i8094H_VRING_ENABLE	⊙	⊙		⊙
2.16	i8094H_VRING_DISABLE	⊙	⊙		⊙
2.17	i8094H_AVTRI_ENABLE	⊙	⊙		⊙
2.17	i8094H_AVTRI_DISABLE	⊙	⊙		⊙
2.18	i8094H_EXD_MP	⊙			
2.18	i8094H_EXD_FP	⊙			
2.18	i8094H_EXD_CP	⊙			
2.18	i8094H_EXD_DISABLE	⊙			
2.19	i8094H_READ_bVAR	⊙			
2.19	i8094H_WRITE_bVAR	⊙			
2.19	i8094H_READ_VAR	⊙			

2.19	i8094H_WRITE_VAR	⊙			
2.20	i8094H_READ_MD	⊙			
2.20	i8094H_WRITE_MD	⊙			
狀態讀取及設定功能					
3.1	i8094H_SET_LP	⊙	⊙	⊙	
3.1	i8094H_GET_LP	⊙	⊙	⊙	
3.2	i8094H_SET_EP	⊙	⊙	⊙	
3.2	i8094H_GET_EP	⊙	⊙	⊙	
3.3	i8094H_GET_CV	⊙			
章節	函式	RTC	MP	ISR	IT
3.4	i8094H_GET_CA	⊙			
3.5	i8094H_GET_DI	⊙	⊙	⊙	
3.5	i8094H_GET_DI_ALL	⊙	⊙	⊙	
3.6	i8094H_GET_ERROR	⊙	⊙	⊙	
3.6	i8094H_GET_ERROR_CODE	⊙	⊙	⊙	
FRnet 功能					
4.1	i8094H_FRNET_IN	⊙	⊙	⊙	
4.2	i8094H_FRNET_OUT	⊙	⊙	⊙	
軸自動歸零					
5.1	i8094H_SET_HV	⊙	⊙		
5.2	i8094H_HOME_LIMIT	⊙	⊙		
5.3	i8094H_SET_HOME_MODE	⊙	⊙		
5.4	i8094H_HOME_START	⊙	⊙		
各軸獨立運動					
6.1.1	i8094H_NORMAL_SPEED	⊙	⊙	⊙	
6.1.2	i8094H_SET_SV	⊙	⊙	⊙	
6.1.3	i8094H_SET_V	⊙	⊙	⊙	
6.1.4	i8094H_SET_A	⊙	⊙	⊙	
6.1.5	i8094H_SET_D	⊙	⊙	⊙	
6.1.6	i8094H_SET_K	⊙	⊙	⊙	
6.1.7	i8094H_SET_L	⊙	⊙	⊙	
6.1.8	i8094H_SET_AO	⊙	⊙	⊙	
6.1.9	i8094H_FIXED_MOVE	⊙	⊙	⊙	
6.1.9	i8094H_SET_PULSE	⊙	⊙	⊙	
6.1.10	i8094H_CONTINUE_MOVE	⊙	⊙	⊙	
補間運動					
6.2.1	i8094H_AXIS_ASSIGN	⊙	⊙	⊙	
6.2.2	i8094H_VECTOR_SPEED	⊙	⊙	⊙	

6.2.3	i8094H_SET_VSV	⊙	⊙	⊙	
6.2.4	i8094H_SET_VV	⊙	⊙	⊙	
6.2.5	i8094H_SET_VA	⊙	⊙	⊙	
6.2.6	i8094H_SET_VD	⊙	⊙	⊙	
6.2.7	i8094H_SET_VK	⊙	⊙	⊙	
6.2.8	i8094H_SET_VL	⊙	⊙	⊙	
6.2.9	i8094H_SET_VAO	⊙	⊙	⊙	
6.2.10	i8094H_LINE_2D	⊙	⊙	⊙	
6.2.11	i8094H_LINE_3D	⊙	⊙	⊙	
6.2.12	i8094H_ARC_CW	⊙x2	⊙x2	⊙x2	
章節	函式	RTC	MP	ISR	IT
6.2.12	i8094H_ARC_CCW	⊙x2	⊙x2	⊙x2	
6.2.13	i8094H_CIRCLE_CW	⊙	⊙	⊙	
6.2.13	i8094H_CIRCLE_CCW	⊙	⊙	⊙	
同步運動					
6.3.1	i8094H_SYNC_ACTION	⊙x2	⊙x2	⊙x2	
6.3.2	i8094H_SET_COMPARE	⊙	⊙	⊙	
6.3.3	i8094H_GET_LATCH	⊙	⊙	⊙	
6.3.4	i8094H_SET_PRESET	⊙	⊙	⊙	
6.3.5	i8094H_SET_OUT	⊙	⊙	⊙	
連續補間運動					
6.4.1	i8094H_RECTANGLE	⊙x4	⊙x4		
6.4.2	i8094H_LINE_2D_INITIAL	⊙x2	⊙x2		
6.4.2	i8094H_LINE_2D_CONTINUE	⊙	⊙		
6.4.3	i8094H_LINE_3D_INITIAL	⊙x2	⊙x2		
6.4.3	i8094H_LINE_3D_CONTINUE	⊙	⊙		
6.4.4	i8094H_MIX_2D_INITIAL	⊙x2	⊙x2		
6.4.4	i8094H_MIX_2D_CONTINUE	⊙x2	⊙x2		
6.4.5	i8094H_CONTINUE_INTP	⊙			
6.4.5	i8094H_CONTINUE_INTP_ARRAY	⊙			
6.4.6	i8094H_HELIX_3D	⊙x3	⊙x3		
6.4.7	i8094H_RATIO_INITIAL	⊙x2	⊙x2		
6.4.7	i8094H_RATIO_2D	⊙	⊙		
6.4.8	i8094H_LINE_SCAN	⊙			
6.4.8	i8094H_LINE_SCAN_START	⊙			
6.4.8	i8094H_LINE_SCAN_OFFSET	⊙			
致能/除能中斷功能					
6.3.6	i8094H_ENABLE_INT	⊙	⊙		

⊙x2 表示佔用
MP 兩個
Function Line

6.3.6	i8094H_DISABLE_INT	⊙	⊙		
6.3.6	i8094H_INTFACTOR_ENABLE	⊙	⊙	⊙	
6.3.6	i8094H_INTFACTOR_DISABLE	⊙	⊙	⊙	
6.3.7	i8094H_ENABLE_RINT	⊙			
6.3.7	i8094H_DISABLE_RINT	⊙			
6.3.7	i8094H_RINT_WAIT	⊙			
其他功能					
6.5.1	i8094H_DRV_HOLD	⊙	⊙	⊙	
6.5.2	i8094H_DRV_START	⊙	⊙	⊙	
6.5.3	i8094H_STOP_WAIT	⊙			
章節	函式	RTC	MP	ISR	IT
6.5.4	i8094H_STOP_SLOWLY	⊙	⊙		
6.5.4	i8094H_STOP_SUDDENLY	⊙	⊙		
6.5.4	i8094H_VSTOP_SLOWLY	⊙	⊙		
6.5.4	i8094H_VSTOP_SUDDENLY	⊙	⊙		
6.5.5	i8094H_CLEAR_STOP	⊙	⊙		
6.5.6	i8094H_INTP_END	⊙	⊙		
i8094H 專用特殊功能					
7.1	i8094H_LOAD_INITIAL	⊙			
7.2.1	i8094H_MP_CREATE	⊙			
7.2.1	i8094H_MP_CLOSE		⊙		
7.2.2	i8094H_MP_CALL	⊙	⊙		
7.2.3	i8094H_MP_SET_VAR		⊙	⊙	
7.2.3	i8094H_MP_SET_RVAR		⊙	⊙	
7.2.3	i8094H_MP_VAR_CALCULATE		⊙	⊙	
7.2.4	i8094H_MP_FOR		⊙		
7.2.4	i8094H_MP_NEXT		⊙		
7.2.5	i8094H_MP_IF		⊙	⊙	
7.2.5	i8094H_MP_ELSE		⊙	⊙	
7.2.5	i8094H_MP_IF_END		⊙	⊙	
7.2.6	i8094H_MP_TIMER		⊙		
7.2.7	i8094H_MP_STOP_WAIT		⊙		
7.2.8	i8094H_MP_SET_RINT		⊙	⊙	

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(BYTE 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	

只將設定的值寫入參數表 **IT**，而不改變目前設定(請參考 7.1)，定義如下：

對照表(2-1a)

軸	X	Y	Z	U	XY	XZ	XU	YZ
代碼	0x11	0x12	0x14	0x18	0x13	0x15	0x19	0x16
變數	INITIAL_X	INITIAL_Y	INITIAL_Z	INITIAL_U	INITIAL_XY	INITIAL_XZ	INITIAL_XU	INITIAL_YZ
軸	YU	ZU	XYZ	XYU	XZU	YZU	XYZU	
代碼	0x1a	0x1c	0x17	0x1b	0x1d	0x1e	0x1f	
變數	INITIAL_YU	INITIAL_ZU	INITIAL_XYZ	INITIAL_XYU	INITIAL_XZU	INITIAL_YZU	INITIAL_XYZU	

不可使用在巨集程式(MP)內。

2.2 註冊軸卡及版本讀取

● **BYTE** i8094H_REGISTRATION(**BYTE** cardNo, **BYTE** slot)

功能： 註冊軸卡，指定插槽及卡號，使用 i8094H 所有功能前，都必須做此註冊。

參數： **cardNo**: 指定卡號 → WinCon-8000 : 0~7

slot: 插槽號碼 → 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 (i8094H_REGISTRATION(cardNo, slot) == YES)
    { //找到軸卡，註冊。
        i8094H_RESET_CARD(cardNo);
        Found++;
    }
}
if (Found == 0) { return;} //找不到軸卡，異常處理。
```

● **WORD** i8094H_GET_VERSION(**void**)

功能： 讀取 i8094H 運動函式庫之版本。

參數： 無

回應： 版本號碼： 西元年月 0x0000 ~ 0x9999

範例：

```
WORD VER_No;
VER_No = i8094H_GET_VERSION();
//讀取 i8094Hce.dll 版本號碼。
```

備註： 以下為讀到的版本release資訊(2007年1月)

i8094H_GET_VERSION: 0x0701

0,7 → 函式庫版本年流水序

0,1 → 函式庫版本月流水序

● **DWORD** i8094H_GET_FW_VERSION(**BYTE** cardNo)

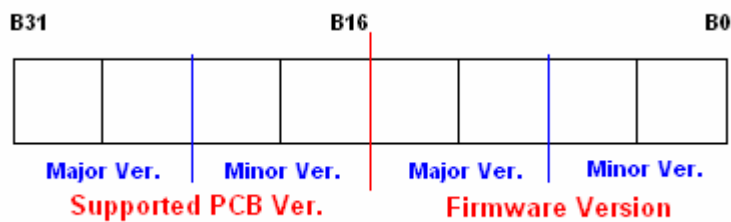
功能： 讀取 i8094H 的韌體版本與相容的 PCB 版次。

參數： **cardNo**: 指定卡號

回應： 版本號碼： 0x00000000 ~ 0x99999999

範例：
DWORD dwVER_No;
WORD wFirmward_Ver, wPCB_Ver;
dwVER_No = i8094H_GET_FW_VERSION(1);
wPCB_Ver = (WORD)(dwVER_No >> 4) & 0xFFFF);
wFirmware_Ver = (WORD)(dwVER_No & 0xFFFF);
//讀取第一張卡的 **Firmware/PCB** 版本號碼。

備註： i8094H_GET_FW_VERSION()回傳值可分為兩部份



例如回傳值為 **0x02210111**代表

Firmware Ver.: **1.11**

支援的PCB: **2.21**以上版次的PCB

2.3 軸卡重置

● **void** i8094H_RESET_CARD(**BYTE** cardNo)

功能： i-8094H 重設成電源開啟狀態，請參考**第7章**重置後的初值設定。

參數： **cardNo**: 指定卡號

回應： 無

範例： i8094H_RESET_CARD (1);
//重置第 1 卡。

● **void** i8094H_CLEAR_CARD_BUFFER(**BYTE** cardNo)

功能： 清除 i-8094H 命令緩衝區所有資料。

參數： **cardNo**: 指定卡號

回應: 無

範例: `i8094H_CLEAR_CARD_BUFFER(1);`
`//清空第1卡命令緩衝區資料。`

2.4 軸輸出PULSE模式設定

● `*void i8094H_SET_PULSE_MODE(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)
`nMode:` 指定模式(參考表 2-2)

回應: 無

範例: `i8094H_SET_PULSE_MODE(1, AXIS_XYZ, 2);`
`i8094H_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。`
`i8094H_SET_PULSE_MODE(1, INITIAL_XYZU, 0);`
`//將第1卡 XYZU 軸，脈波輸出模式皆設為 0，寫入參數表。`

脈波輸出模式表(2-2)

形式	模式	方向	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)
	3		PULSE(負緣觸發)	DIR(LOW)
	4	-	PULSE(正緣觸發)	DIR(LOW)
	5		PULSE(負緣觸發)	DIR(LOW)

2.5 設定軸速度輸出最大範圍

- ***void i8094H_SET_MAX_V(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)
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: `i8094H_SET_MAX_V(1, AXIS_XY, 200000L);`
`//設定第 1 卡 X Y 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 i8094H_SET_HLMT(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)
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: `i8094H_SET_HLMT(1, AXIS_XYZU, 0, 0);`
`//設定第1卡 X Y Z U 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 i8094H_LIMITSTOP_MODE(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)
nMode: 設定處理方法: 0=立即停止,1=減速後停止

回應： 無

範例：
`i8094H_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 i8094H_SET_NHOME(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)
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：
`i8094H_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 i8094H_SET_HOME_EDGE(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)
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：
`i8094H_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 i8094H_SET_SLMT(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)
dwFL: 前軟體極限值(-2,000,000,000 ~ +2,000,000,000)
dwRL: 後軟體極限值(-2,000,000,000 ~ +2,000,000,000)
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：
`i8094H_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 i8094H_CLEAR_SLMT(BYTE cardNo, BYTE 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：
`i8094H_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 i8094H_SET_ENCODER(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)
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：
`i8094H_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 i8094H_SERVO_ON(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)

回應： 無

範例：
`i8094H_SERVO_ON(1, AXIS_XYZU);`
//設定第 1 卡 X Y Z U 軸，啟動驅動器伺服。

- ***void i8094H_SERVO_OFF(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)

回應： 無

範例：
`i8094H_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 i8094H_SET_ALARM(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)
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：
`i8094H_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 i8094H_SET_INPOS(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)
OnMode: 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：
`i8094H_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(i8094H_Getting_Started_1.6tc.pdf)手冊，一般 DI 輸入接線範例。

2.15 設定數mm位輸入雜訊濾波功能

- ***void i8094H_SET_FILTER(BYTE cardNo, BYTE axis, BYTE FEn, BYTE 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)
 只將設定的值寫入參數表(參考表 2-1a)
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: `i8094H_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)

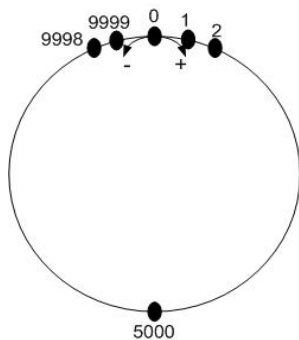
- ***void i8094H_VRING_ENABLE(BYTE cardNo, BYTE axis, DWORD nVRing)**

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)
只將設定的值寫入參數表(參考表 2-1a)
nVRing: 環狀計數器值(0 ~ +2,000,000,000)

回應： 無

範例：
i8094H_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 i8094H_VRING_DISABLE(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)

回應： 無

範例：
i8094H_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 i8094H_AVTRI_ENABLE(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)

回應： 無

範例：
`i8094H_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 i8094H_AVTRI_DISABLE(BYTE cardNo, BYTE 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)
只將設定的值寫入參數表(參考表 2-1a)

回應： 無

範例：
`i8094H_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

無法將外部輸入驅動的設定值寫入參數表。

2.18.1 手輪脈波驅動

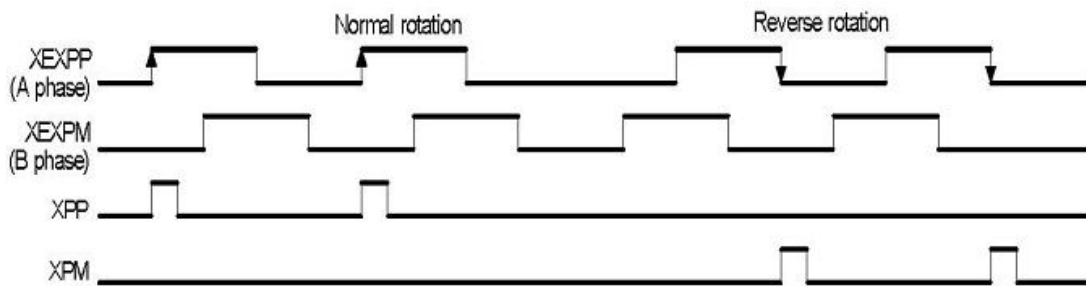
● **void i8094H_EXD_MP**(**BYTE cardNo**, **BYTE axis**, **DWORD data**)

功能： 執行手輪輸入驅動，輸出固定步數。

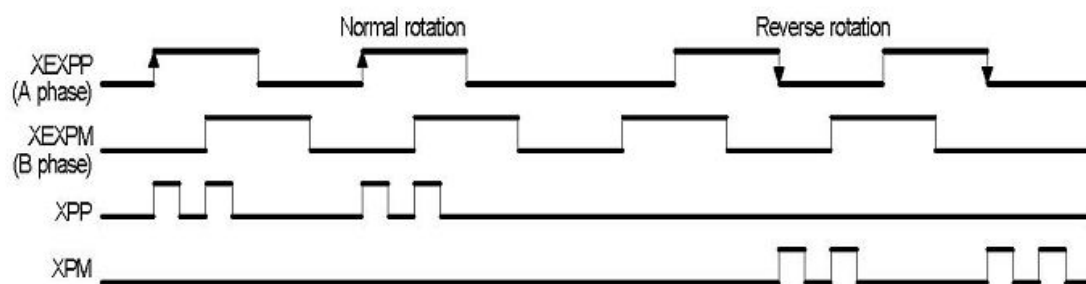
參數：
cardNo: 指定卡號
axis: 指定軸號碼 X 或 Y 或 Z 或 U (1 or 2 or 4 or 8)
data: 指定步數 (0 ~ +2,000,000,000)

回應： 無

範例：
i8094H_EXD_MP(1, AXIS_X, 1);
//第1卡 X 軸，手輪觸發移動1步(Pulse)。



i8094H_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

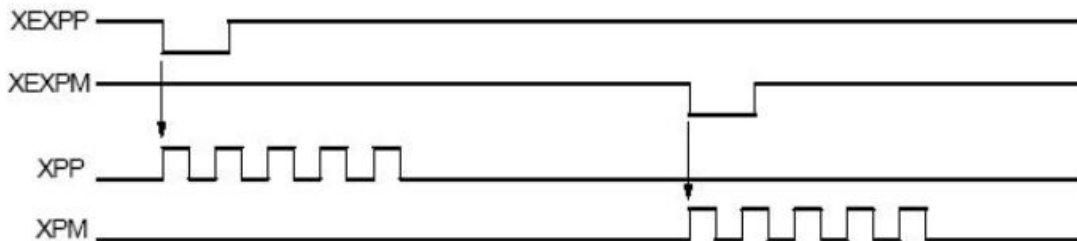
● **void i8094H_EXD_FP(BYTE cardNo, BYTE axis, DWORD data)**

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(0 ~ +2,000,000,000)

回應： 無

範例：
`i8094H_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

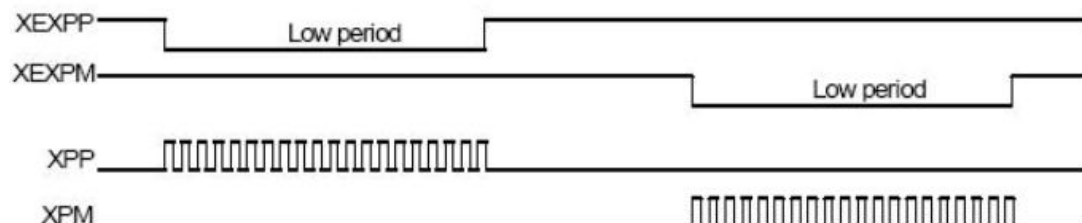
● **void i8094H_EXD_CP(BYTE cardNo, BYTE axis, DWORD data)**

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(0 ~ +2,000,000,000)

回應： 無

範例：
`i8094H_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 i8094H_EXD_DISABLE(BYTE cardNo, BYTE 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： **i8094H_EXD_DISABLE(1, AXIS_X);**
//關閉第1卡 X 軸，外部輸入驅動功能。

2.19 讀寫自定義變數(VAR)

- **BYTE i8094H_READ_bVAR(BYTE cardNo, BYTE bvarNo)**

功能： 讀取變數值 bVARn，可直接引入巨集程式(MP)中使用，請參考第7章。

參數： **cardNo:** 指定卡號
bvarNo: 自定義變數: bVAR0 ~ bVAR127

回應： 目前變數值 (0 ~ +255)

範例： **BYTE bdata;**
bdata = i8094H_READ_bVAR(1, bVAR100);
//讀取第1卡 VAR100 變數值。

- **void i8094H_WRITE_bVAR(BYTE cardNo, BYTE bvarNo, BYTE bVar)**

功能： 寫入變數值 bVARn，可直接引入巨集程式(MP)中使用，請參考第7章。

參數： **cardNo:** 指定卡號
bvarNo: 自定義變數: bVAR0 ~ bVAR127
bVar: 變數值 (0 ~ +255)

回應： 無

範例： **i8094H_WRITE_bVAR(1, bVAR100, 100);**
//寫入第1卡 bVAR100 變數值 = 100。

● **long** i8094H_READ_VAR(**BYTE** cardNo, **long** varNo)

功能: 讀取變數值 VARn，可直接引入巨集程式(MP)中使用，請參考第 7 章。

參數: **cardNo:** 指定卡號
varNo: 自定義變數: VAR0 ~ VAR511

回應: 目前變數值 (-2,000,000,000 ~ +2,000,000,000)

範例: **long** ldata;
ldata = i8094H_READ_VAR(1, VAR100);
//讀取第 1 卡 VAR100 變數值。

● **void** i8094H_WRITE_VAR(**BYTE** cardNo, **long** varNo, **long** IVar)

功能: 寫入變數值 VARn，可直接引入巨集程式(MP)中使用，請參考第 7 章。

參數: **cardNo:** 指定卡號
varNo: 自定義變數: VAR0 ~ VAR511
IVar: 變數值 (-2,000,000,000 ~ +2,000,000,000)

回應: 無

範例: i8094H_WRITE_VAR(1, VAR100, 10000);
//寫入第 1 卡 VAR100 變數值 = 10,000。

2.20 讀寫斷電保持資料(MD)

● void i8094H_READ_MD(BYTE cardNo, long mdNo, long* ldata, float* fdata)

功能： 讀取機器資料值(Machine Data)。

參數： **cardNo**: 指定卡號
mdNo: 機器資料值(**long**): MD0 ~ MD1023
機器資料值(**float**): MD1024 ~ MD2047
&ldata: 讀取 MD **long** (-2,147,483,648 ~ +2,147,483,647)
&fdata: 讀取 MD **float** (整數位加小數位共 6 位數)

回應： 無

範例：
long ldata;
float fdata;
i8094H_WRITE_MD(CardNo, MD100, -99999, 0);
i8094H_READ_MD(CardNo, MD100, &ldata, 0);
//讀取第 1 卡 MD100 long data 值。
i8094H_WRITE_MD(CardNo, MD1500, 0, -990.999);
i8094H_READ_MD(CardNo, MD1500, 0, &fdata);
//讀取第 1 卡 MD1500 float data 值。

● void i8094H_WRITE_MD(BYTE cardNo, long mdNo, long ldata, float fdata)

功能： 寫入機器資料值(Machine Data)。

參數： **cardNo**: 指定卡號
mdNo: 機器資料值(**long**): MD0 ~ MD1023
機器資料值(**float**): MD1024 ~ MD2047
ldata: 寫入 MD **long** (-2,147,483,648 ~ +2,147,483,647)
fdata: 寫入 MD **float** (整數位加小數位共 6 位數)

回應： 無

範例： 同上

3 狀態讀取及設定功能

3.1 設定及讀取指令邏輯位置

- ****void i8094H_SET_LP(BYTE cardNo, BYTE 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(-2,000,000,000 ~ +2,000,000,000)

回應： 無

範例：
`i8094H_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 i8094H_GET_LP(BYTE cardNo, BYTE 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 (-2,000,000,000 ~ +2,000,000,000)

範例：
`long X_LP;`
`X_LP = i8094H_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 i8094H_SET_EP(BYTE cardNo, BYTE 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(-2,000,000,000 ~ +2,000,000,000)

回應： 無

範例：
`i8094H_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 i8094H_GET_EP(BYTE cardNo, BYTE 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 (-2,000,000,000 ~ +2,000,000,000)

範例：
`long X_EP;`
`X_EP = i8094H_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** `i8094H_GET_CV`(**BYTE** *cardNo*, **BYTE** *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 = i8094H_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** `i8094H_GET_CA`(**BYTE** *cardNo*, **BYTE** *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 = i8094H_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 i8094H_GET_DI(BYTE cardNo, BYTE axis, BYTE 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 或 Z 或 U (1 or 2 or 4 or 8)
nType:	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 Z 相 IN2 信號)

回應： YES: on
NO: off

範例：

```
if (i8094H_GET_DI(1, AXIS_X, 1) == YES)
{
    //讀取第 1 卡 X 軸，前極限信號處理。
}
```

● ****WORD i8094H_GET_DI_ALL(BYTE cardNo, BYTE 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;**
i8094H_GET_DI_ALL(1, AXIS_X, &wStatus);

```
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 i8094H_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 **i8094H_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 (i8094H_GET_ERROR(1) == YES)**

```
{  
    //讀取第 1 卡，錯誤停止處理。  
    WORD ErrorCode_X = i8094H_GET_ERROR_CODE(1, AXIS_X);  
    WORD ErrorCode_Y = i8094H_GET_ERROR_CODE(1, AXIS_Y);  
    WORD ErrorCode_Z = i8094H_GET_ERROR_CODE(1, AXIS_Z);  
    WORD ErrorCode_U = i8094H_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。  
        i8094H_CLEAR_STOP(1);  
    }  
}
```

● ****WORD i8094H_GET_ERROR_CODE(BYTE cardNo, BYTE 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

3.7 讀取目前RTC狀態

- **BYTE** `i8094H_CHECK_RTC`(**BYTE** `cardNo`)

功能： 讀取軸卡 RTC 命令之緩衝區狀態。

參數： `cardNo`: 指定卡號

回應： YES: 表示緩衝區已滿
NO: 表示緩衝區未滿

範例：

```
if (i8094H_CHECK_RTC(1) == YES)
{
    //緩衝區已滿之例外處理。
}
```

4 FRnet功能

4.1 FRnet DI讀取

- ****WORD** i8094H_FRNET_IN(**BYTE** cardNo, **BYTE** wSA)

功能: 讀取 FRnet 的數位輸入資料，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
wSA: 外接群組範圍 SA8~SA15

回應: **WORD** 16-位元輸入資料 (0x0000 ~ 0xffff)

範例: **WORD** IN_Data;
IN_Data = i8094H_FRNET_IN(1, 8);
//設定第 1 卡，外接群組 SA = 8。

4.2 FRnet DO寫入

- ****void** i8094H_FRNET_OUT(**BYTE** cardNo, **BYTE** wRA, **DWORD** data)

功能: 寫入 FRnet 的數位輸出資料，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
wRA: 外接群組範圍 RA0~RA7
dara: 0x00000000 ~ 0x0000ffff

回應: 無

範例: **i8094H_FRNET_OUT(1, 0, 0x0000ffff);**
//設定第 1 卡，外接群組 RA = 0，輸出資料為 0x0000ffff。

5 軸自動歸零

I-8094H 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。

5.1 設定軸歸零速度

- ***void i8094H_SET_HV(BYTE cardNo, BYTE 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： `i8094H_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 i8094H_HOME_LIMIT(BYTE cardNo, BYTE axis, BYTE 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： `i8094H_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

- ***void i8094H_SET_HOME_MODE(BYTE cardNo, BYTE axis, BYTE nStep1, BYTE nStep2, BYTE nStep3, BYTE nStep4, DWORD 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 ~ +2,000,000,000)

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：
i8094H_SET_V(1, 0x1, 20000);
i8094H_SET_HV(1, 0x1, 500);
i8094H_SET_HOME_MODE(1, 0x1, 2, 2, 1, 1, 3500);
i8094H_HOME_START(1, 0x1);
i8094H_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 i8094H_HOME_START(BYTE cardNo, BYTE 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：
`i8094H_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

- 請參考 6.5.3 或 7.2.7 ，與 **STOP_WAIT function** 共用。

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

- ****void i8094H_NORMAL_SPEED(BYTE cardNo, BYTE axis, BYTE 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。
i8094H_SET_MAX_V(cardNo, AXIS_XYZU, 20000); //設定軸最高速 20K PPS。

```
//=====
i8094H_NORMAL_SPEED(cardNo, AXIS_XYZU, 0); //設定 XYZU 為對稱 T 曲線。
i8094H_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094H_SET_A(cardNo, AXIS_XYZU, 1000); //設定 XYZU 軸加速度 1000 PPS/S。
i8094H_SET_SV(cardNo, AXIS_XYZU, 2000); //設定 XYZU 初始速度 2000 PPS。
i8094H_SET_AO(cardNo, AXIS_XYZU, 9); //XYZU 軸減速(保留脈波數)= 9 PPS。
i8094H_FIXED_MOVE(cardNo, AXIS_XYZU, 10000); //XYZU 移動 10000 Pulse。
```

```
//=====
i8094H_NORMAL_SPEED(cardNo, AXIS_XYZU, 1); //設定 XYZU 軸對稱 S 曲線。
i8094H_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094H_SET_K(cardNo, AXIS_XYZU, 500); // XYZU 軸 K=500 PPS/S^2。
i8094H_SET_SV(cardNo, AXIS_XYZU, 200); //設定 XYZU 軸初始速度=200 PPS。
i8094H_SET_AO(cardNo, AXIS_XYZU, 9); //XYZU 軸減速(保留脈波數)= 9 PPS。
i8094H_FIXED_MOVE(cardNo, AXIS_XYZU, -10000); //XYZU 移-10000 Pulse。
```

```
//=====
i8094H_NORMAL_SPEED(cardNo, AXIS_XYZU,2); //設定 XYZU 非對稱 T 曲線。
i8094H_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094H_SET_A(cardNo, AXIS_XYZU, 1000); //設定 XYZU 軸加速度 1000 PPS/S。
i8094H_SET_D(cardNo, AXIS_XYZU, 500); //設定 XYZU 軸減速度=500 PPS/S。
i8094H_SET_SV(cardNo, AXIS_XYZU, 200); //設定 XYZU 軸初始速度=200 PPS。
i8094H_SET_AO(cardNo, AXIS_XYZU, 9); // XYZU 軸減速(保留脈波數)= 9 PPS。
i8094H_FIXED_MOVE(cardNo, axis, 10000); //執行 XYZU 軸移動 10000 Pulse。
```

```
//=====
i8094H_NORMAL_SPEED(cardNo, AXIS_XYZU,3); // XYZU 軸為非對稱 S 曲線。
i8094H_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094H_SET_K(cardNo, AXIS_XYZU, 500); // XYZU 軸 K=500 PPS/S^2。
i8094H_SET_L(cardNo, AXIS_XYZU, 300); // XYZU 軸 L=300 PPS/S^2。
i8094H_SET_SV(cardNo, AXIS_XYZU, 200); //設定 XYZU 軸初始速度=200 PPS。
i8094H_SET_AO(cardNo, AXIS_XYZU, 9); // XYZU 軸減速(保留脈波數)= 9 PPS。
i8094H_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 i8094H_SET_SV(BYTE cardNo, BYTE 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： **i8094H_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 i8094H_SET_V(BYTE cardNo, BYTE 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：
`i8094H_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 i8094H_SET_A(BYTE cardNo, BYTE 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，i8094H_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：
`i8094H_SET_MAX_V(1, AXIS_X, 20000);`
`//最小加速度單位值: 20,000 ÷ 64 = 312.5 × n ≐ 313...625...938...。`
`//最大加速度值: 20,000 × 125 = 2,500,000。`
`i8094H_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 i8094H_SET_D(BYTE cardNo, BYTE 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，i8094H_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：
i8094H_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: $20,000 \times 125 = 2,500,000$ 。
i8094H_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 i8094H_SET_K(BYTE cardNo, BYTE 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)
參考 2.5 所設定，i8094H_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: 2,000,000,000

回應： 無

範例：
i8094H_SET_MAX_V(1, AXIS_X, 20000);
//最小加速度變化率單位值: $20,000 \times 0.0119211 = 238.422 \times n \approx 238...476...$ 。
i8094H_SET_K(1, AXIS_X, 10000);
//設定第 1 卡 X 軸，加速度變化率為 10,000 PPS/ Sec²。

6.1.7 設定軸減速度變化率

● ****void i8094H_SET_L(BYTE cardNo, BYTE 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/ Sec²)
參考 2.5 所設定，i8094H_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: 2,000,000,000

回應： 無

範例：
i8094H_SET_MAX_V(1, AXIS_X, 20000);
//最小減速度變化率單位值: $20,000 \times 0.0119211 = 238.422 \times n \approx 238...476...$ 。
i8094H_SET_L(1, AXIS_X, 10000);
//設定第 1 卡 X 軸，減速度變化率為 10,000 PPS/ Sec²。

6.1.8 設定軸減速(保留脈波數)

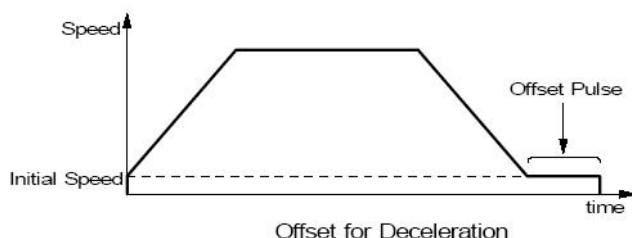
● ****void i8094H_SET_AO(BYTE cardNo, BYTE axis, long data)**

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：
i8094H_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

- ****void i8094H_FIXED_MOVE(BYTE cardNo, BYTE 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 (-2,000,000,000 ~ +2,000,000,000)

回應： 有錯誤發生(欲讀錯誤碼請搭配使用 i8094H_GET_ERROR_CODE)

範例：
BYTE cardNo=1; //設定第 1 號卡。
i8094H_SET_MAX_V(cardNo, AXIS_XYZU, 20000); //設定最高速 20K PPS。
i8094H_NORMAL_SPEED(cardNo, AXIS_XYZU, 0); //設定 XYZU 軸對稱 T 曲線。
i8094H_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094H_SET_A(cardNo, AXIS_XYZU, 1000); //設定 XYZU 軸加速度 1000 PPS/S。
i8094H_SET_SV(cardNo, AXIS_XYZU, 2000); //設定 XYZU 初始速度 2000 PPS。
i8094H_SET_AO(cardNo, AXIS_XYZU, 9); //XYZU 軸減速(保留脈波數)= 9 PPS。
i8094H_FIXED_MOVE(cardNo, AXIS_XYZU, 10000); //XYZU 移動 10000 Pulse。

- ****void i8094H_SET_PULSE(BYTE cardNo, BYTE axis, DWORD data)**

功能： 在單軸固定步數輸出時，可於途中改變輸出步數，但無法改變方向。

參數：
cardNo: 指定卡號
axis: 指定軸號碼 (參考表 2-1)
data: 指定步數 (0 ~ +2,000,000,000)

回應： 無

範例：
BYTE cardNo=1; //設定第 1 號卡。
i8094H_SET_MAX_V(cardNo, AXIS_XYZU, 20000); //設定最高速 20K PPS。
i8094H_NORMAL_SPEED(cardNo, AXIS_XYZU, 0); //設定 XYZU 軸對稱 T 曲線。
i8094H_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094H_SET_A(cardNo, AXIS_XYZU, 1000); //設定 XYZU 軸加速度 1000 PPS/S。
i8094H_SET_SV(cardNo, AXIS_XYZU, 2000); //設定 XYZU 初始速度 2000 PPS。
i8094H_SET_AO(cardNo, AXIS_XYZU, 9); //XYZU 軸減速(保留脈波數)= 9 PPS。
i8094H_FIXED_MOVE(cardNo, AXIS_XYZU, 10000); //XYZU 移動 10000 Pulse。
i8094H_SET_PULSE(cardNo, AXIS_XYZU, 9000); //改變移動步數 9000 Pulse。

6.1.10 連續脈波輸出

- ****void i8094H_CONTINUE_MOVE(BYTE cardNo, BYTE 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： 有錯誤發生(欲讀錯誤碼請搭配使用 i8094H_GET_ERROR_CODE)

範例：
BYTE cardNo=1; //設定第 1 號卡。
i8094H_SET_MAX_V(cardNo, AXIS_XYZU, 20000); //設定最高速 20K PPS。
i8094H_NORMAL_SPEED(cardNo, AXIS_XYZU, 0); //設定 XYZU 軸對稱 T 曲線。
i8094H_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094H_SET_A(cardNo, AXIS_XYZU,1000); //設定 XYZU 軸加速度 1000 PPS/S。
i8094H_SET_SV(cardNo, AXIS_XYZU, 2000); //設定 XYZU 初始速度 2000 PPS。
i8094H_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 i8094H_AXIS_ASSIGN(BYTE cardNo, BYTE axis1, BYTE axis2, BYTE 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：
i8094H_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 i8094H_VECTOR_SPEED(BYTE cardNo, BYTE 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。
        i8094H_SET_MAX_V(cardNo, AXIS_XYZU, 20000); //設 XYZU 最高速 20K PPS

//=====
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 0);
//二軸(直線&弧&圓)固定向量速度 VSV=VV，設 VV 即可。
i8094H_SET_VV(cardNo, 1000); //設定第 1 卡，向量定速度為 1000 PPS。
i8094H_LINE_2D(cardNo, 12000, 10000); //執行向量 2D 補間。

//=====
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 1);
//二軸直線對稱 T 曲線 (VSV、VV、VA、VAO)。
i8094H_SET_VSV(cardNo, 500); //設定向量初始速度為 500 PPS。
i8094H_SET_VV(cardNo, 2000); //設定向量速度為 2000 PPS。
i8094H_SET_VA(cardNo, 1000); //設定向量加速速度為 1000 PPS。
i8094H_LINE_2D(cardNo, 20000, 10000); //執行向量 2D 補間。

//=====
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 2);
//二軸直線對稱 S 曲線(VSV、VV、VA、VK、AO)。
i8094H_SET_VSV(cardNo, 200); //設定向量初始速度為 500 PPS。
i8094H_SET_VV(cardNo, 2000); //設定向量速度為 2000 PPS。
i8094H_SET_VK(cardNo, 500); //設定 VK=500 PPS/S^2。
i8094H_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094H_LINE_2D(cardNo, 10000, 10000); //執行向量 2D 補間。

//=====
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 3);
//二軸直線非對稱 T 曲線 (VSV、VV、VA、VD、VAO)。
i8094H_SET_VSV(cardNo, 100); //設定向量初始速度為 500 PPS。
i8094H_SET_VV(cardNo, 2000); //設定向量速度為 2000 PPS。
i8094H_SET_VA(cardNo, 1000); //設定向量加速度為 1000 PPS/s。
i8094H_SET_VD(cardNo, 500); //設定向量減速度為 500 PPS/s。
i8094H_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094H_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;
i8094H_SET_MAX_V(cardNo, AXIS_XYZU, 8000);
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 4);
//二軸直線非對稱 S 曲線 (VSV、VV、VK、VL、VAO)。
i8094H_SET_VSV(cardNo, sv); //設定向量初始速度為 sv PPS。
i8094H_SET_VV(cardNo, v); //設定向量速度為 v PPS。
i8094H_SET_VK(cardNo, 500); //設定 VK=500 PPS/S^2。
i8094H_SET_VL(cardNo, 300); //設定 VL=300 PPS/S^2。
i8094H_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094H_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;
i8094H_SET_MAX_V(cardNo, AXIS_XYZU, 8000);
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 5);
//二軸(弧&圓)對稱 T 曲線 (VSV、VV、VA、VAO)。
i8094H_SET_VSV(cardNo, sv); //設定向量初始速度為 sv PPS。
i8094H_SET_VV(cardNo, v); //設定向量速度為 v PPS。
i8094H_SET_VA(cardNo, a); //設定向量加速度為 a PPS/s。
i8094H_SET_VAO(cardNo, 0); //設定軸向量減速(保留脈波數)0 Pulse。
i8094H_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;
i8094H_SET_MAX_V(cardNo, AXIS_XYZU, 8000);
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 6);
//二軸(弧&圓)對稱 T 曲線 (VSV、VV、VA、VAO)。
i8094H_SET_VSV(cardNo, sv); //設定向量初始速度為 sv PPS。
i8094H_SET_VV(cardNo, v); //設定向量速度為 v PPS。
i8094H_SET_VA(cardNo, a); //設定向量加速度為 a PPS/s。
i8094H_SET_VD(cardNo, d); //設定向量減速度為 d PPS/s。
i8094H_SET_VAO(cardNo, 0); //設定軸向量減速(保留脈波數)0 Pulse。
i8094H_CIRCLE_CW(cardNo, c1, c2); //執行二軸順時針圓形補間。

```

```

//=====

```

```

i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 7);
//三軸直線固定向量速度 (VSV=VV)。
i8094H_SET_VSV(cardNo, 1000); //設定向量初始速度為 1000 PPS。
i8094H_SET_VV(cardNo, 1000); //設定向量速度為 1000 PPS。
i8094H_LINE_3D(cardNo, 10000, 10000,10000); //執行向量 3D 補間。

//=====
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 8);
//三軸直線對稱 T 曲線 (VSV、VV、VA、VAO)。
i8094H_SET_VSV(cardNo, 100); //設定向量初始速度為 1000 PPS。
i8094H_SET_VV(cardNo, 3000); //設定向量速度為 3000 PPS。
i8094H_SET_VA(cardNo, 500); //設定向量加速度為 500 PPS/s。
i8094H_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094H_LINE_3D(cardNo, 10000, 1000,20000); //執行向量 3D 補間。

//=====
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 9);
//三軸直線對稱 S 曲線 (VSV、VV、VK、VAO)
i8094H_SET_VSV(cardNo, 100); //設定向量初始速度為 1000 PPS。
i8094H_SET_VV(cardNo, 3000); //設定向量速度為 3000 PPS。
i8094H_SET_VK(cardNo, 500); //設定 VK=500 PPS/S^2。
i8094H_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094H_LINE_3D(cardNo, 10000, 1000,1000); //執行向量 3D 補間。

//=====
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 10);
//三軸直線非對稱 T 曲線 (VSV、VV、VA、VD、VAO)。
i8094H_SET_VSV(cardNo, 100); //設定向量初始速度為 1000 PPS。
i8094H_SET_VV(cardNo, 2000); //設定向量速度為 3000 PPS。
i8094H_SET_VA(cardNo, 1000); //設定向量加速度為 1000 PPS/s。
i8094H_SET_VD(cardNo, 500); //設定向量減速度為 500 PPS/s。
i8094H_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。
i8094H_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;
i8094H_SET_MAX_V(cardNo, AXIS_XYZU, 8000);
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 11);

```

```
//三軸直線非對稱 S 曲線 (VSV、VV、VK、VL、VAO)。  
i8094H_SET_VSV(cardNo, sv); //設定向量初始速度為 sv PPS。  
i8094H_SET_VV(cardNo, v); //設定向量速度為 v PPS。  
i8094H_SET_VK(cardNo, 500); //設定 VK=500 PPS/S^2。  
i8094H_SET_VL(cardNo, 300); //設定 VL=300 PPS/S^2。  
i8094H_SET_VAO(cardNo, 20); //設定軸向量減速(保留脈波數)20 Pulse。  
i8094H_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 i8094H_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： **i8094H_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 i8094H_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： **i8094H_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 i8094H_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，**i8094H_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： **i8094H_SET_MAX_V(1, AXIS_X, 20000);**
//最小向量加速度單位值: 20,000 ÷ 64 = 312.5 × n ≐ 313...625...938...。
//最大向量加速度值: 20,000 × 125 = 2,500,000。
i8094H_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 i8094H_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，i8094H_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： i8094H_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: $20,000 \times 125 = 2,500,000$ 。
i8094H_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 i8094H_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)
參考 2.5 所設定，i8094H_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: 2,000,000,000

回應： 無

範例： i8094H_SET_MAX_V(1, AXIS_X, 20000);
//最小向量加速度變化率單位值: $20,000 \times 0.0119211 = 238.422 \times n \doteq 238...476...$ 。
i8094H_SET_VK(1, 10000);
//設定第 1 卡 X 軸，向量加速度變化率為 10,000 PPS/ Sec²。

6.2.8 設定軸向量減速度變化率

● ****void i8094H_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/ Sec²)
參考 2.5 所設定，i8094H_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: **2,000,000,000**

回應： 無

範例： `i8094H_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...。`
`i8094H_SET_VL(1, 10000);`
`//設定第 1 卡 X 軸，向量減速度變化率為 10,000 PPS/ Sec2。`

6.2.9 設定軸向量減速(保留脈波數)

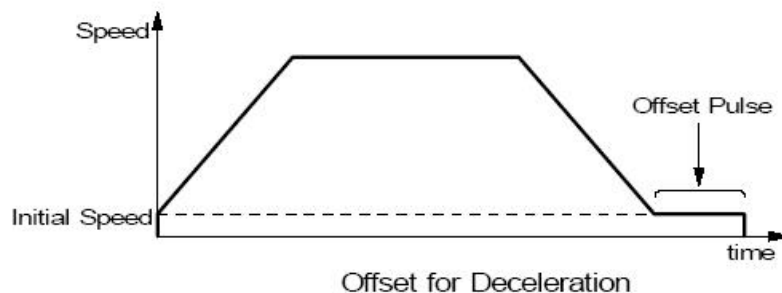
● ****void i8094H_SET_VAO(BYTE cardNo, long data)**

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： `i8094H_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

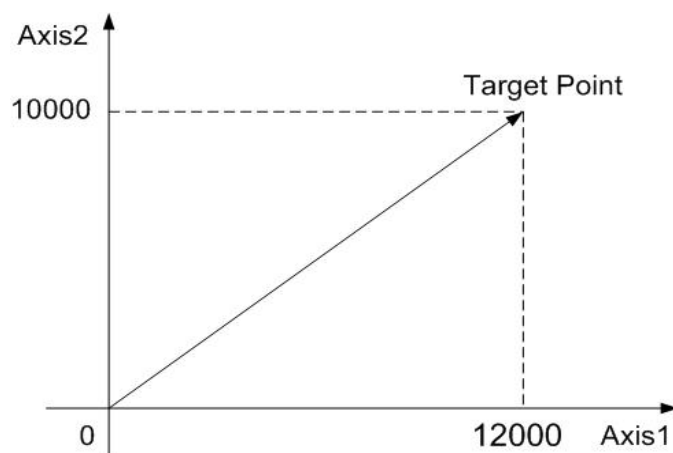
● ****void i8094H_LINE_2D(BYTE cardNo, long fp1, long fp2)**

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 (-2,000,000,000 ~ +2,000,000,000)
fp2: 指定第二軸 Pulse 數 (-2,000,000,000 ~ +2,000,000,000)

回應： 有錯誤發生(欲讀錯誤碼請搭配使用 i8094H_GET_ERROR_CODE)

範例：
`i8094H_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

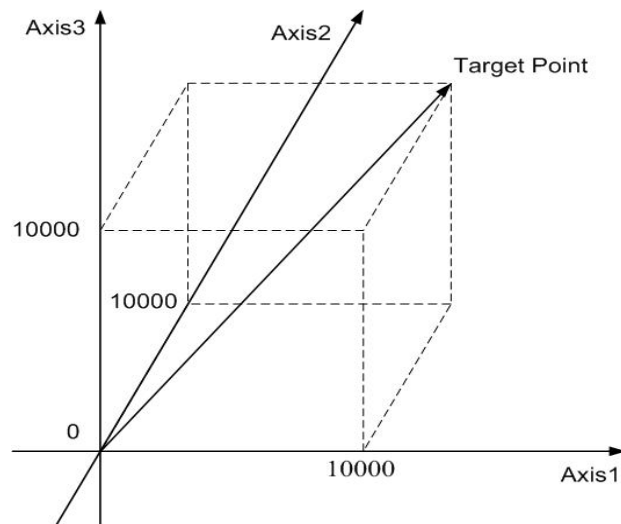
● ****void i8094H_LINE_3D(BYTE cardNo, long fp1, long fp2, long fp3)**

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 (-2,000,000,000 ~ +2,000,000,000)
fp2: 指定第二軸 Pulse 數 (-2,000,000,000 ~ +2,000,000,000)
fp3: 指定第三軸 Pulse 數 (-2,000,000,000 ~ +2,000,000,000)

回應： 有錯誤發生(欲讀錯誤碼請搭配使用 i8094H_GET_ERROR_CODE)

範例：
`i8094H_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

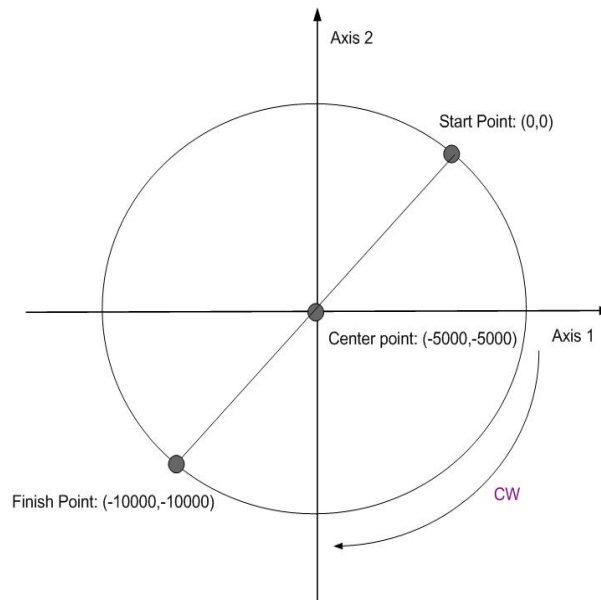
● ****void i8094H_ARC_CW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2)**

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 (-2,000,000,000 ~ +2,000,000,000)
cp2: 指定第二軸圓弧中心相對位置 (-2,000,000,000 ~ +2,000,000,000)
fp1: 指定第一軸圓弧終點相對位置 (-2,000,000,000 ~ +2,000,000,000)
fp2: 指定第二軸圓弧終點相對位置 (-2,000,000,000 ~ +2,000,000,000)

回應： 有錯誤發生(欲讀錯誤碼請搭配使用 i8094H_GET_ERROR_CODE)

範例：
i8094H_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

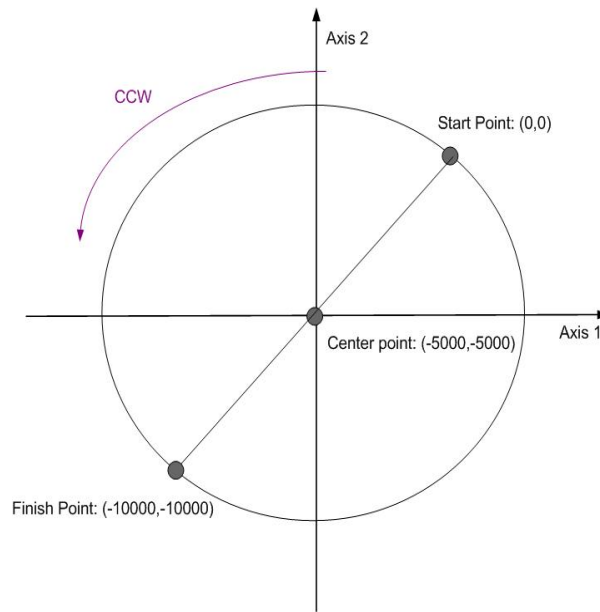
- ****void i8094H_ARC_CCW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2)**

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 (-2,000,000,000 ~ +2,000,000,000)
cp2: 指定第二軸圓弧中心相對位置 (-2,000,000,000 ~ +2,000,000,000)
fp1: 指定第一軸圓弧終點相對位置 (-2,000,000,000 ~ +2,000,000,000)
fp2: 指定第二軸圓弧終點相對位置 (-2,000,000,000 ~ +2,000,000,000)

回應： 有錯誤發生(欲讀錯誤碼請搭配使用 i8094H_GET_ERROR_CODE)

範例：
i8094H_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

- ****void i8094H_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 (-2,000,000,000 ~ +2,000,000,000)
cp2: 指定第二軸圓弧中心相對位置 (-2,000,000,000 ~ +2,000,000,000)

回應: 有錯誤發生(欲讀錯誤碼請搭配使用 i8094H_GET_ERROR_CODE)

範例: **i8094H_CIRCLE_CW(1, 0, 10000);**
//設定第 1 卡，執行二軸順時針圓形補間。

- ****void i8094H_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 (-2,000,000,000 ~ +2,000,000,000)
cp2: 指定第二軸圓弧中心相對位置 (-2,000,000,000 ~ +2,000,000,000)

回應: 有錯誤發生(欲讀錯誤碼請搭配使用 i8094H_GET_ERROR_CODE)

範例: **i8094H_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

6.3.1 設定同步運動條件

- ****void i8094H_SYNC_ACTION(**
BYTE cardNo, BYTE axis1, BYTE axis2, DWORD nSYNC,
BYTE nDRV, BYTE nLATCH, BYTE nPRESET, BYTE nOUT,
BYTE nINT, BYTE isrNoX, BYTE isrNoY, BYTE isrNoZ, BYTE isrNoU)

功能： 同步運動條件的設定。
 (完全硬體處理，不會耗用 WinCon、I8000 系統資源)

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

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

axis2 設 0，表示同步運動軸為主軸自己

nSYNC: 同步運動條件因子，可複選，如下表說明

號碼	代號	說明
0x00000000		除能同步運動條件因子
0x00000001	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 必須和 i8094H_SET_COMPARE 並用(6.3.2)
0x00000002	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 必須和 i8094H_SET_COMPARE 並用(6.3.2)
0x00000004	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 必須和 i8094H_SET_COMPARE 並用(6.3.2)
0x00000008	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 必須和 i8094H_SET_COMPARE 並用(6.3.2)
0x00000010	D-STA	同步運動發生於驅動開始時
0x00000020	D-END	同步運動發生於驅動結束時
0x00000040	IN3 ↑	同步運動發生於 nIN3 信號正邊緣觸發從低到高準位
0x00000080	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 ↑ (0x00000001 + 0x00000040 = 0x00000041)

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 移動中 之軸設定無效 步數設定需搭配 OPSET 設定新的位置
2	FDRV-	負方向固定脈波驅動，對 移動中 之軸設定無效 步數設定需搭配 OPSET 設定新的位置
3	CDRV+	正方向連續脈波驅動，對 移動中 之軸設定無效
4	CDRV-	負方向連續脈波驅動，對 移動中 之軸設定無效
5	SSTOP	減速停止
6	ISTOP	直接停止

nLATCH: 同步位置門鎖項目，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 **i8094H_GET_LATCH** 並用(6.3.3)。

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] 連續運動指令 CONTINUE_MOVE 之軸無法設定新的位置
4	VLSET	設定新的速度(V)，[V ← PRESET]

必須和 **i8094H_SET_PRESET** 並用(6.3.4)。

nOUT: 設定外部觸發訊號，如下表說明

號碼	代號	說明
0		取消輸出外部觸發訊號
1	OUT	設定輸出外部觸發訊號

必須和 **i8094H_SET_OUT** 並用(6.3.5)。

nINT: 設定中斷功能，如下表說明

號碼	代號	說明
0		取消中斷功能
1	INT	設定同時產生中斷訊號

- 將針對 **axis2** 所設定之同步運動軸，產生中斷，請個別填入如下 **isrNoX**、**isrNoY**、**isrNoZ**、**isrNoU** 相對的中斷服務程式號碼。
- 必須和 **i8094H_ENABLE_INT** 並用(6.3.6)。

isrNoX: ISR1 ~ ISR20：指定同步運動軸 **X** 的中斷服務程式號碼，參考(7.2.1)

不使用請填 0

isrNoY: ISR1 ~ ISR20 : 指定同步運動軸 **Y** 的中斷服務程式號碼，參考(7.2.1)
 不使用請填 0

isrNoZ: ISR1 ~ ISR20 : 指定同步運動軸 **Z** 的中斷服務程式號碼，參考(7.2.1)
 不使用請填 0

isrNoU: ISR1 ~ ISR20 : 指定同步運動軸 **U** 的中斷服務程式號碼，參考(7.2.1)
 不使用請填 0

回應: 無

範例: //範例 1. 當 U 軸 IN3 收到正邊緣觸發信號，便改變速度及 LATCH encoder 值。
i8094H_SYNC_ACTION(cardNo, AXIS_U, 0, 0X00000040, 0, 2, 4, 0, 0, 0);
i8094H_SET_MAX_V(cardNo, AXIS_U, 5000); //設定 U 軸最高速 5K PPS。
i8094H_NORMAL_SPEED(cardNo, AXIS_U, 0); //設定 U 軸為對稱 T 曲線。
i8094H_SET_V(cardNo, AXIS_U, 2000); //設定 U 軸速度=2,000 PPS。
i8094H_SET_A(cardNo, AXIS_U, 100000); //設定 U 軸加速度=100K PPS/S。
i8094H_SET_SV(cardNo, AXIS_U, 100); //設定 U 軸初始速度=100 PPS。
i8094H_FIXED_MOVE(cardNo, AXIS_U, 10000); //設定 U 軸移動 10,000 Pulse。
i8094H_SET_PRESET(cardNo, AXIS_U, 100); //設定 U 軸新的速度值=100 PPS。

```
while (i8094H_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 = i8094H_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。
i8094H_SYNC_ACTION(cardNo, AXIS_U, 2, 0X00000001, 1, 0, 3, 0, 0, 0);
i8094H_SET_COMPARE(cardNo, AXIS_U, 0, 1, 5000);
//設定 COMP+的值=5,000，來源參考 U 軸 EP。
i8094H_SET_MAX_V(cardNo, AXIS_YU, 9000); //設定 YU 軸最高速 9K PPS。
i8094H_NORMAL_SPEED(cardNo, AXIS_YU, 0); //設定 YU 軸為對稱 T 曲線。
i8094H_SET_V(cardNo, AXIS_YU, 3000); //設定 YU 軸速度=3,000 PPS。
i8094H_SET_A(cardNo, AXIS_YU, 200000); //設定 YU 軸加速度=200K PPS/S。
i8094H_SET_SV(cardNo, AXIS_YU, 200); //設定 YU 軸初始速度=200 PPS。
i8094H_FIXED_MOVE(cardNo, AXIS_U, 10000); //設定 U 軸移動 10,000 Pulse。
i8094H_SET_PRESET(cardNo, AXIS_Y, 2000); //設定 Y 軸移動 2,000 PPS。

6.3.2 設定COMPARE值

- ****void i8094H_SET_COMPARE(BYTE cardNo, BYTE axis, BYTE nSELECT, BYTE 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: (-2,000,000,000 ~ +2,000,000,000)

回應： 無

範例：
`i8094H_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 i8094H_GET_LATCH(BYTE cardNo, BYTE axis)**

功能： 讀取同步位置閘鎖值。

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： 位置閘鎖值 -2,000,000,000 ~ +2,000,000,000

範例：
`long data = i8094H_GET_LATCH(1, AXIS_Y);`
`//讀取第 1 卡 Y 軸，同步位置閘鎖值。`

6.3.4 設定PRESET資料

● ****void i8094H_SET_PRESET(BYTE cardNo, BYTE axis, long data)**

功能： 選擇同步資料設定方式(各同步運動軸無法單獨設定)。

參數：
cardNo: 指定卡號
axis: 同步運動軸號碼，必需同 6.3.1 的 **axis2** (i8094H_SYNC_ACTION)
data: LP: (-2,000,000,000 ~ +2,000,000,000)
EP: (-2,000,000,000 ~ +2,000,000,000)
P: (-2,000,000,000 ~ +2,000,000,000)
V: 最大值請參考 2.5，如同步運動軸為兩軸以上，請將其 i8094H_SET_MAX_V 皆設為相同值

回應： 無

範例： 請參考 6.3.1 之範例

6.3.5 設定OUT資料

● ****void i8094H_SET_OUT(BYTE cardNo, BYTE axis, BYTE outEdge, BYTE PulseWidth)**

功能： 輸出脈波設定。

參數：
cardNo: 指定卡號
axis: 指定軸號碼(參考表 2-1)，目前只支援 X、Y 兩軸
outEdge: trigger OUT 觸發邏輯: 0=低準位觸發, 1=高準位觸發
PulseWidth: trigger OUT 輸出脈波寬度
0 = 10 uSec
1 = 20 uSec
2 = 100 uSec
3 = 200 uSec
4 = 1,000 uSec
5 = 2,000 uSec
6 = 10,000 uSec
7 = 20,000 uSec

回應： 無

範例： 請參考 6.3.1 之範例

6.3.6 控制器對軸卡(i8094H)中斷功能

- ***void i8094H_ENABLE_INT(BYTE cardNo)**

功能： 致能中斷功能。

參數： **cardNo:** 指定卡號

回應： 無

範例： **i8094H_ENABLE_INT(1);**
//設定第 1 卡，致能其中斷功能。

- ***void i8094H_DISABLE_INT(BYTE cardNo)**

功能： 除能中斷功能。

參數： **cardNo:** 指定卡號

回應： 無

範例： **i8094H_DISABLE_INT(1);**
//解除第 1 卡其中斷功能。

● ****void i8094H_INTFACTOR_ENABLE(BYTE cardNo, BYTE axis, BYTE nINT, BYTE isrNo)**

功能： 設定觸發中斷條件因子。

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 i8094H_SET_COMPARE 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 i8094H_SET_COMPARE 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 i8094H_SET_COMPARE 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 i8094H_SET_COMPARE 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

isrNo: ISR1 ~ ISR20：指定中斷服務程式號碼，參考(7.2.1)

回應： 無

範例：
i8094H_SET_COMPARE(0, AXIS_X, 0, 0, 5000);
i8094H_SET_COMPARE(0, AXIS_Y, 0, 0, 10000);
i8094H_INTFACTOR_ENABLE(1, AXIS_X, 4, ISR1);
i8094H_INTFACTOR_ENABLE(1, AXIS_Y, 4, ISR2);

i8094H_SET_LP(0, AXIS_XY, 0);
i8094H_SET_MAX_V(0, AXIS_XY, 1000000);
i8094H_NORMAL_SPEED(0, AXIS_XY, 0); //設定 XY 為對稱 T 曲線。
i8094H_SET_V(0, AXIS_XY, 5000); //設定 XY 軸速度=2000 PPS。
i8094H_SET_A(0, AXIS_XY, 10000); //設定 XY 軸加速度 1000 PPS/S。
i8094H_SET_SV(0, AXIS_XY, 200); //設定 XY 初始速度 2000 PPS。
i8094H_SET_AO(0, AXIS_XY, 0); //XY 軸減速(保留脈波數)= 9 PPS。
i8094H_FIXED_MOVE(0, AXIS_XY, 20000);

loop:
if (i8094H_STOP_WAIT(0, AXIS_XY) == NO) goto loop;

備註： 勿和軟體極限功能(2.10)共用。

● ****void i8094H_INTFACTOR_DISABLE(BYTE cardNo, BYTE axis, BYTE nINT)**

功能: 解除觸發中斷條件因子。

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	中斷發生於第一個脈波產生時，可視為驅動開始時
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
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
3	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
4	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
5	C-END	中斷發生於等速段結束，意即減速段開始時
6	C-STA	中斷發生於等速段開始時，意即加速段開始時
7	D-END	中斷發生於減速段結束時，可視為驅動結束時

回應: 無

範例: `i8094H_INTFACTOR_DISABLE(1, AXIS_XYZU, 1);`
`i8094H_INTFACTOR_DISABLE(1, AXIS_XYZU, 2);`
`i8094H_INTFACTOR_DISABLE(1, AXIS_XYZU, 3);`
`i8094H_INTFACTOR_DISABLE(1, AXIS_XYZU, 4);`
`// 設卡號等於 1 時，除能 4 軸的位置計數器與比較暫存器的中斷條件。`

6.3.7 軸卡(i8094H)對控制器中斷功能

- **BYTE i8094H_ENABLE_RINT(BYTE cardNo)**

功能： 致能中斷功能。

參數： **cardNo:** 指定卡號

回應： 0： 失敗
1： 成功

範例： **i8094H_ENABLE_RINT(1);**
//設定第 1 卡，致能其中斷功能。

- **void i8094H_DISABLE_RINT(BYTE cardNo)**

功能： 除能中斷功能。

參數： **cardNo:** 指定卡號

回應： 無

範例： **i8094H_DISABLE_RINT(1);**
//設定第 1 卡，除能其中斷功能。

● **BYTE** i8094H_RINT_WAIT(**BYTE** cardNo, **BYTE** waitNo, **DWORD** TimeOut)

功能： 等待中斷狀態碼回應。

參數： **cardNo**: 指定卡號

waitNo: 設定中斷條件，可複選，如下表說明

號碼	說明
0x01	Line Scan 完成，(請參考 6.4.8)
0x02	Macro Program 完成，(請參考 7.2.2)
0x04	使用者定義 RINT 完成，(請參考 7.2.8)
0x08	連續補間被中斷
0x10	
0x20	預留系統使用
0x40	Axes Error，(請配合 3.6，i8094H_GET_ERROR_CODE)
0x80	Module Error，(請配合 2.3 重置軸卡)

例：選 Axes Error 和 Module Error (0x40 + 0x80 = 0xC0)

TimeOut: 單位 ms，設“0”表示無 Time Out 功能

回應： 0： Time Out
非零值： 收到中斷狀態碼

範例： **BYTE** irec = i8094H_ENABLE_RINT(CardNo);

//以下的 code 請寫在 Thread Function。

irec = i8094H_RINT_WAIT(CardNo, 0xc8, 0);

switch (irec)

{

case 0x08: AfxMessageBox(
 TEXT("check Interrupt event is Finished (waitNo : 0x08)"), MB_OK, -1);
 break;

case 0x40: AfxMessageBox(
 TEXT("check Interrupt event is Finished (waitNo : 0x40)"), MB_OK, -1);
 break;

case 0x80: AfxMessageBox(
 TEXT("check Interrupt event is Finished (waitNo : 0x80)"), MB_OK, -1);
 break;

default:
 break;

}

AfxMessageBox(_T("ScanTime_Thread STOP !!"));

6.4 客製特殊應用

6.4.1 二軸矩形連續補間

- `*void i8094H_RECTANGLE(`
`BYTE cardNo, BYTE axis1, BYTE axis2,`
`BYTE nAcc, BYTE Sp, BYTE nDir, long Lp, long Wp, long Rp,`
`DWORD RSV, DWORD RV, DWORD RA, DWORD RD)`

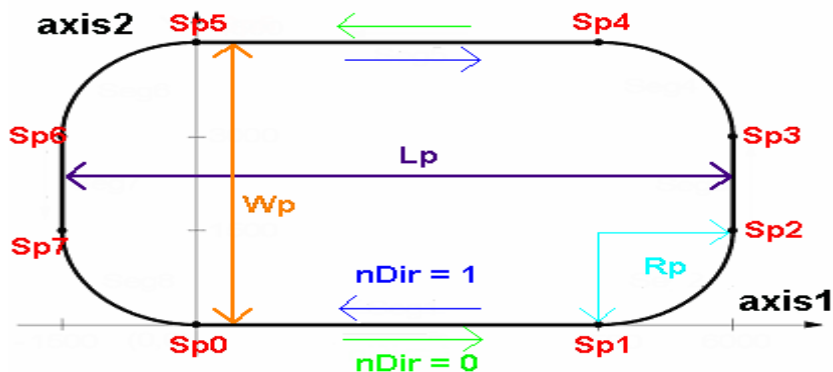
功能： 執行二軸矩形補間。
(不會耗用 WinCon、I8000 系統資源)

參數：

<code>cardNo:</code>	指定卡號
<code>axis1:</code>	指定第一軸號碼: X、Y、Z、U (1、2、4、8)
<code>axis2:</code>	指定第二軸號碼: X、Y、Z、U (1、2、4、8)
<code>nAcc:</code>	0 → 定速度補間 1 → 對稱 T 曲線加減速補間
<code>Sp:</code>	設定起點 0 ~ 7 (Sp0 ~ Sp7 如下圖所示)
<code>nDir:</code>	設定方向 0、1 (CCW or CW)
<code>Lp:</code>	設定長度 Pulse 數 (1 ~ 2,000,000,00)
<code>Wp:</code>	設定寬度 Pulse 數 (1 ~ 2,000,000,000)
<code>Rp:</code>	設定圓弧半徑 Pulse 數 (1 ~ 2,000,000,000)
<code>RSV:</code>	設定矩形補間向量起始速度(PPS)
<code>RV:</code>	設定矩形補間向量速度(PPS)
<code>RA:</code>	設定矩形補間向量加速度(PPS/Sec)
<code>RD:</code>	設定矩形末段補間向量減速度(PPS/Sec)

回應： 有錯誤發生(欲讀錯誤碼請搭配使用 `i8094H_GET_ERROR_CODE`)

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。`
`i8094H_SET_MAX_V(1, AXIS_XYZU, 16000); //最高速度為 16000 PPS。`
`i8094H_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 i8094H_LINE_2D_INITIAL(BYTE cardNo, BYTE axis1, BYTE 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：

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)
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 i8094H_LINE_2D_CONTINUE(BYTE cardNo, BYTE nType, long fp1, long fp2)** 並用。

- ***void i8094H_LINE_2D_CONTINUE**(**BYTE cardNo**, **BYTE nType**, **long fp1**, **long fp2**)

功能: 執行二軸直線連續補間。
(不會耗用 WinCon、I8000 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 (-2,000,000,000 ~ +2,000,000,000)
fp2:	指定第二軸 Pulse 數 (-2,000,000,000 ~ +2,000,000,000)

回應: 有錯誤發生(欲讀錯誤碼請搭配使用 i8094H_GET_ERROR_CODE)

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;
i8094H_SET_MAX_V(cardNo, AXIS_XYZU,160000L);
i8094H_LINE_2D_INITIAL(cardNo, AXIS_X, AXIS_Y, sv, v, a);
for (loop1 = 0; loop1 < 10000; loop1++)
{
    i8094H_LINE_2D_CONTINUE (cardNo, 0, 100, 100);
    i8094H_LINE_2D_CONTINUE (cardNo, 0, -100, -100);
}
i8094H_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 i8094H_LINE_3D_INITIAL(BYTE cardNo, BYTE axis1, BYTE axis2, BYTE 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: **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)
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 i8094H_LINE_3D_CONTINUE(BYTE cardNo, BYTE nType, long fp1, long fp2, long fp3)** 並用。

- ***void i8094H_LINE_3D_CONTINUE(BYTE cardNo, BYTE nType, long fp1, long fp2, long fp3)**

功能: 執行三軸直線連續補間。
(不會耗用 WinCon、I8000 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 (-2,000,000,000 ~ +2,000,000,000)
fp2: 指定第二軸 Pulse 數 (-2,000,000,000 ~ +2,000,000,000)
fp3: 指定第三軸 Pulse 數 (-2,000,000,000 ~ +2,000,000,000)

回應: 有錯誤發生(欲讀錯誤碼請搭配使用 **i8094H_GET_ERROR_CODE**)

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;
i8094H_SET_MAX_V(cardNo, AXIS_XYZU,160000L);

```

i8094H_LINE_3D_INITIAL(cardNo, AXIS_X, AXIS_Y, AXIS_Z, sv, v, a);
for (loop1 = 0; loop1 < 10000; loop1++)
{
    i8094H_LINE_3D_CONTINUE(cardNo, 0, 100, 100, 100);
    i8094H_LINE_3D_CONTINUE(cardNo, 0, -100, -100, -100);
}
i8094H_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 i8094H_MIX_2D_INITIAL(BYTE cardNo, BYTE axis1, BYTE axis2, BYTE 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
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)
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 i8094H_MIX_2D_CONTINUE(BYTE cardNo, BYTE nAcc, BYTE nType, long cp1, long cp2, long fp1, long fp2)**並用。

- ***void i8094H_MIX_2D_CONTINUE(BYTE cardNo, BYTE nAcc, BYTE nType, long cp1, long cp2, long fp1, long fp2)**

功能： 執行二軸直線和圓弧連續補間。
(不會耗用 WinCon、I8000 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 → i8094H_LINE_2D(BYTE cardNo, long fp1, long fp2) 2 → i8094H_ARC_CW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2) 3 → i8094H_ARC_CCW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2) 4 → i8094H_CIRCLE_CW(BYTE cardNo, long cp1, long cp2) 5 → i8094H_CIRCLE_CCW(BYTE cardNo, long cp1, long cp2)
cp1:	指定第一軸圓、弧中心相對位置 (-2,000,000,000 ~ +2,000,000,000)
cp2:	指定第二軸圓、弧中心相對位置 (-2,000,000,000 ~ +2,000,000,000)
fp1:	指定第一軸 Pulse 數 (-2,000,000,000 ~ +2,000,000,000)
fp2:	指定第二軸 Pulse 數 (-2,000,000,000 ~ +2,000,000,000)

回應： 有錯誤發生(欲讀錯誤碼請搭配使用 i8094H_GET_ERROR_CODE)

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;
i8094H_SET_MAX_V(cardNo, AXIS_XYZU,160000L);
i8094H_MIX_2D_INITIAL(cardNo, AXIS_X, AXIS_Y, 1, sv, v, a);
for (loop1 = 0; loop1 < 10000; loop1++)
{
    i8094H_MIX_2D_CONTINUE (cardNo, 0, 1, 0, 0, 100, 100);
    i8094H_MIX_2D_CONTINUE (cardNo, 0, 2, 100, 0, 100, 100);
}
i8094H_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)

- `void i8094H_CONTINUE_INTP(`
`BYTE cardNo, BYTE axis1, BYTE axis2, BYTE axis3,`
`BYTE nAcc, DWORD VSV, DWORD VV, DWORD VA, DWORD VD)`

功能: 執行多點連續補間(對稱 T 曲線), 請先將運動陣列資料寫入。
(不會耗用 WinCon、I8000 系統資源)

參數:

<code>cardNo:</code>	指定卡號
<code>axis1:</code>	指定第一軸號碼: X、Y、Z、U (1、2、4、8)
<code>axis2:</code>	指定第二軸號碼: X、Y、Z、U (1、2、4、8)
<code>axis3:</code>	指定第三軸號碼: X、Y、Z、U (1、2、4、8)
<code>nAcc:</code>	0 → 定速度補間 (vv) 1 → 對稱 T 曲線加減速補間 (vsv、vv、va、vd)
<code>VSV:</code>	設定補間向量起始速度(PPS)
<code>VV:</code>	設定補間向量速度(PPS)
<code>VA:</code>	設定補間向量加速度(PPS/Sec)
<code>VD:</code>	設定末段補間向量減速度(PPS/Sec)

回應: 有錯誤發生(欲讀錯誤碼請搭配使用 `i8094H_GET_ERROR_CODE`)

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。`
`i8094H_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};`
`i8094H_CONTINUE_INTP_ARRAY(`
`cardNo, nType, cp1, cp2, fp1, fp2, fp3);`
`//=====`
`i8094H_CONTINUE_INTP(cardNo, AXIS_X, AXIS_Y, 0, 1, sv, v, a, d);`
`//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。`

- **void i8094H_CONTINUE_INTP_ARRAY(**
BYTE cardNo, BYTE nType[], long cp1[], long cp2[],
long fp1[], long fp2[], long fp3[])

功能: 多點連續補間的運動陣列資料。

參數: **cardNo:** 指定卡號
nType[]: 連續補間點最大: 有 0 ~ 99 共 100 點
 1 → i8094H_LINE_2D(BYTE cardNo, long fp1, long fp2)
 2 → i8094H_ARC_CW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2)
 3 → i8094H_ARC_CCW(BYTE cardNo, long cp1, long cp2, long fp1, long fp2)
 4 → i8094H_CIRCLE_CW(BYTE cardNo, long cp1, long cp2)
 5 → i8094H_CIRCLE_CCW(BYTE cardNo, long cp1, long cp2)
 6 → i8094H_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 (-2,000,000,000 ~ +2,000,000,000)
cp2[]: 指定第二軸圓、弧中心相對位置 (-2,000,000,000 ~ +2,000,000,000)
fp1[]: 指定第一軸 Pulse 數 (-2,000,000,000 ~ +2,000,000,000)
 指定第一軸圓弧終點相對位置
fp2[]: 指定第二軸 Pulse 數 (-2,000,000,000 ~ +2,000,000,000)
 指定第二軸圓弧終點相對位置
fp3[]: 指定第三軸 Pulse 數 (-2,000,000,000 ~ +2,000,000,000)
 (二軸和三軸無法混合使用，所有未使用數值請填 0)

回應: 無

範例: 同上 i8094H_CONTINUE_INTP 範例。

6.4.6 三軸螺旋運動

- `*void i8094H_HELIX_3D(`
`BYTE cardNo, BYTE axis1, BYTE axis2, BYTE axis3, BYTE nDir,`
`DWORD VV, long cp1, long cp2, long cycle, long pitch)`

功能： 執行螺旋運動(定速)。

(不會耗用 WinCon、I8000 系統資源)

參數：

<code>cardNo:</code>	指定卡號
<code>axis1:</code>	指定圓形運動第一軸號碼: X、Y、Z、U (1、2、4、8)
<code>axis2:</code>	指定圓形運動第二軸號碼: X、Y、Z、U (1、2、4、8)
<code>axis3:</code>	指定同動單軸號碼: X、Y、Z、U (1、2、4、8)
<code>nDir:</code>	0 → 圓形運動 CW 1 → 圓形運動 CCW
<code>VV:</code>	設定螺旋向量速度(PPS)
<code>cp1:</code>	指定第一軸圓中心相對位置 (-2,000,000,000 ~ +2,000,000,000)
<code>cp2:</code>	指定第二軸圓中心相對位置 (-2,000,000,000 ~ +2,000,000,000)
<code>cycle:</code>	圓形運動循環次數 (-2,000,000,000 ~ +2,000,000,000)
<code>pitch:</code>	單軸節距 (-2,000,000,000 ~ +2,000,000,000)

回應： 有錯誤發生(欲讀錯誤碼請搭配使用 `i8094H_GET_ERROR_CODE`)

範例： `BYTE cardNo=1; //設定第 1 號卡。`

```
//=====
i8094H_SET_MAX_V(cardNo, AXIS_XYZU,16000L); //設最高速度為 16K PPS。
long v=50000; //設定向量速度為 PPS。
i8094H_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。
```

```
//=====
i8094H_SET_MAX_V(cardNo, AXIS_XYZU,16000L); //設最高速度為 16K PPS。
long v=100000; //設定向量速度為 PPS。
i8094H_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 i8094H_RATIO_INITIAL(BYTE cardNo, BYTE axis1, BYTE axis2, DWORD SV, DWORD V, DWORD A, BYTE CMX, BYTE CDV)**

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)
CMX: 設定兩軸的比例分子 (1 ~ +127)
CDV: 設定兩軸的比例分母 (1 ~ +127)

回應: 無

範例: 和 **void i8094H_RATIO_2D(BYTE cardNo, BYTE nType, long data, BYTE nDir)** 並用。

- ***void i8094H_RATIO_2D(BYTE cardNo, BYTE nType, long data, BYTE nDir)**

功能: 執行比例連續運動。
(不會耗用 WinCon、I8000 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 (-2,000,000,000 ~ +2,000,000,000)
nDir: 比例運動第二軸方向:
0 → 正轉 CW
1 → 反轉 CCW

回應: 有錯誤發生(欲讀錯誤碼請搭配使用 i8094H_GET_ERROR_CODE)

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;

```

i8094H_SET_MAX_V(cardNo, AXIS_XYZU,160000L);
i8094H_RATIO_INITIAL(cardNo, AXIS_U, AXIS_X, sv, v, a, 9, 25);
for (loop2 = 0; loop2 < 5; loop2++)
{
    for (loop1 = 0; loop1 < 5; loop1++)
    {
        i8094H_RATIO_2D(cardNo, 0, 3600, 0);
        i8094H_RATIO_2D(cardNo, 0, 3600, 1);
    }
    i8094H_RATIO_2D(cardNo, 0, 7200, 0);
    i8094H_RATIO_2D(cardNo, 0, 3600, 1);
}
i8094H_RATIO_2D(cardNo, 1, 7200, 0);
//設定第 1 卡，執行 U、X 兩軸比例(0.36)運動。

```

6.4.8 同步輸出Line Scan運動

- **void i8094H_LINE_SCAN(BYTE cardNo, BYTE axis, BYTE Type, BYTE outEdge, BYTE PulseWidth, long Pitch)**

功能: 等距 Line Scan trigger out : Max speed < 100KHz (Pulse Width 10 uS)。
不等距 Line Scan trigger out : Max speed < 18KHz。

參數:

cardNo: 指定卡號

axis: 指定運動軸號碼：等距 X 或 Y (1、2)
不等距 X (1)

Type: 0 = 等距運動 (參考 C+，LP)
1 = 不等距運動 (參考 C+，LP)
2 = 等距運動 (參考 C+，EP)
3 = 不等距運動 (參考 C+，EP)

outEdge: trigger OUT 觸發邏輯: 0=低準位觸發, 1=高準位觸發

PulseWidth: trigger OUT 輸出脈波寬度

0 = 10 uSec
1 = 20 uSec
2 = 100 uSec
3 = 200 uSec
4 = 1,000 uSec
5 = 2,000 uSec
6 = 10,000 uSec
7 = 20,000 uSec

Pitch: 指定間距步數 (-2,000,000,000 ~ +2,000,000,000)

回應: 無

範例: **long** lp1, lp2, lp3, lp4;
BYTE irec;
i8094H_SET_MAX_V(CardNo, AXIS_XY, 4000000); //設定 XY 軸最高速 10K PPS。
i8094H_NORMAL_SPEED(CardNo, AXIS_XY, 0); //設定 XY 軸為對稱 T 曲線。
i8094H_SET_V(CardNo, AXIS_XY, 2000000); //設定 XY 軸速度=2,000 PPS。
i8094H_SET_A(CardNo, AXIS_XY, 20000000); //設定 XY 軸加速度=200K PPS/S。
i8094H_SET_SV(CardNo, AXIS_XY, 2000000); //設定 XY 軸初始速度=200 PPS。
i8094H_LINE_SCAN(CardNo, AXIS_X, 0, 0, 0, -39);
i8094H_LINE_SCAN(CardNo, AXIS_Y, 0, 0, 0, -79);
i8094H_LINE_SCAN_START(CardNo, AXIS_XY, 0, -4000000);

irec = i8094H_RINT_WAIT(CardNo, 0x01, 10000);
lp1 = i8094H_GET_LP(CardNo, AXIS_X);
lp2 = i8094H_GET_LP(CardNo, AXIS_Y);

- **void** i8094H_LINE_SCAN_START(**BYTE** cardNo, **BYTE** axis, **BYTE** Type, **long** Position)

功能: 啟動 Line Scan trigger out 運動。

執行不等距運動期間(等距運動即無限制):

a. 所有 ISR 將停止執行

b. 請勿使用如下命令

i8094H_READ_bVAR
i8094H_READ_VAR
i8094H_READ_FLONG
i8094H_GET_LP
i8094H_GET_EP
i8094H_GET_CV
i8094H_GET_CA
i8094H_GET_CVAR
i8094H_GET_DI
i8094H_GET_ERROR
i8094H_GET_ERROR_CODE
i8094H_FRNET_IN
i8094H_HOME_WAIT
i8094H_GET_LATCH
i8094H_STOP_WAIT
i8094H_CLEAR_STOP

參數: **cardNo:** 指定卡號
axis: 指定運動軸號碼 : 等距 X 或 Y 或 XY (1、2、3)
不等距 X (1)

Type: 0 =等距運動 (參考 C+ , LP)
 1 =不等距運動 (參考 C+ , LP , 需配合設定
i8094H_LINE_SCAN_OFFSET)
 2 =等距運動 (參考 C+ , EP)
 3 =不等距運動 (參考 C+ , EP , 需配合設定
i8094H_LINE_SCAN_OFFSET)
Position: 指定總步數，移動距離 (-2,000,000,000 ~ +2,000,000,000)

回應: 無

範例: 同上

- **void i8094H_LINE_SCAN_OFFSET(BYTE cardNo, DWORD Total, char Offset[])**

功能: 設定不等距間距補正步數。

參數: **cardNo:** 指定卡號
Total: Pitch 最大補正點數 (0 ~ 40960)
Offset: 指定 Pitch 補正步數 (-128 ~ +127)

回應: 無

範例: **long lp1;**
BYTE irec;
i8094H_SET_MAX_V(CardNo, AXIS_X, 4000000); //設定 X 軸最高速 10K PPS。
i8094H_NORMAL_SPEED(CardNo, AXIS_X, 0); //設定 X 軸為對稱 T 曲線。
i8094H_SET_V(CardNo, AXIS_X, 2000000); //設定 X 軸速度=2,000 PPS。
i8094H_SET_A(CardNo, AXIS_X, 20000000); //設定 X 軸加速度=200K PPS/S。
i8094H_SET_SV(CardNo, AXIS_X, 2000000); //設定 X 軸初始速度=200 PPS。
char offset[40960];
for (lp1 = 0; lp1 < 40960; lp1++)
{offset[lp1] = 0;}
i8094H_LINE_SCAN(CardNo, AXIS_X, 1, 0, 1, 105);
i8094H_LINE_SCAN_OFFSET(CardNo, 40000, offset);
i8094H_LINE_SCAN_START(CardNo, AXIS_X, 1, 4200000);

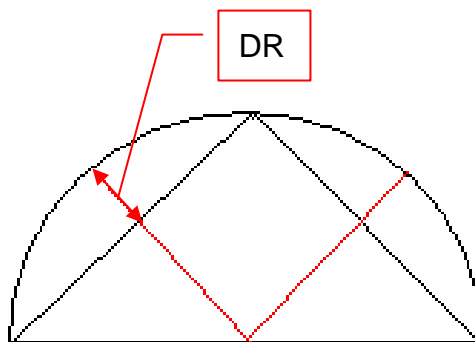
irec = i8094H_RINT_WAIT(CardNo, 0x01, 10000);
lp1 = i8094H_GET_LP(CardNo, AXIS_X);

6.4.9 三軸圓弧補間

- `long i8094H_ARC_3D(BYTE cardNo, BYTE axis1, BYTE axis2, BYTE axis3, long *P1, long *P2, DWORD DR, DWORD VV)`

功能：三點定一 3D 圓弧，圓弧的起點(0, 0, 0)，圓弧的第二點(P1[0], P1[1], P1[2])，圓弧的結束點(P2[0], P2[1], P2[2])，以三軸直線補間逼近(定速度)。

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)
P1: 指定圓弧中點相對位置: (P1[0]_axis1, P1[1]_axis2, P1[2]_axis3)
P2: 指定圓弧終點相對位置: (P2[0]_axis1, P2[1]_axis2, P2[2]_axis3)
DR: 設定直線逼近圓弧的誤差值 (+1 ~ +2,000,000,000 Pulse)



VV: 設定向量速度(PPS)

回應：
< 0 (負值) 向量速度設定太大，回傳執行最大的向量速度值(PPS)
> 0 (正值) 設定正確，回傳執行 3D 圓弧耗費的時間(ms)
如有錯誤發生(欲讀錯誤碼請搭配使用 `i8094H_GET_ERROR_CODE`)

範例: `long P1[3], P2[3];`

```
P1[0] = -10000;      P1[1] = 10000;      P1[2] = 5000;  
P2[0] = -10000;      P2[1] = -10000;     P2[2] = 10000;
```

```
long rec = i8094H_ARC_3D(  
    1, AXIS_X, AXIS_Y, AXIS_Z, P1, P2, 5, 1000000);  
//DR=5 Pulse、VV=1,000,000 PPS, rec 回傳值: +45 (ms)
```

```
long rec = i8094H_ARC_3D(  
    1, AXIS_X, AXIS_Y, AXIS_Z, P1, P2, P3, 1, 1000000);  
//DR=1 Pulse、VV=1,000,000, rec 回傳值: -724397  
//表示軸卡以 VV=724,397 PPS 執行
```

6.4.10 三軸混合連續補間

- **void i8094H_MIX_3D_INITIAL**(**BYTE cardNo**, **BYTE axis1**, **BYTE axis2**, **BYTE axis3**, **DWORD VSV**, **DWORD VV**, **DWORD VA**)

功能： 三軸直線和圓弧連續補間初始設定，對稱 T 曲線加減速。

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)
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 **long i8094H_MIX_3D_CONTINUE**(
BYTE cardNo, **BYTE nAcc**, **BYTE nType**, **long *P1**, **long *P2**, **DWORD DR**)並用。

- **long i8094H_MIX_3D_CONTINUE**(**BYTE cardNo**, **BYTE nAcc**, **BYTE nType**, **long *P1**, **long *P2**, **DWORD DR**)

功能： 執行三軸直線和圓弧連續補間。

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: 0 → **i8094H_LINE_3D(cardNo, *P1)**
1 → **i8094H_ARC_3D(cardNo, *P1, *P2, DR)**
P1: 指定三軸直線相對位置: (P1[0]_axis1, P1[1]_axis2, P1[2]_axis3)
指定圓弧中點相對位置: (P1[0]_axis1, P1[1]_axis2, P1[2]_axis3)
P2: 指定圓弧終點相對位置: (P2[0]_axis1, P2[1]_axis2, P2[2]_axis3)
DR: 設定直線逼近圓弧的誤差值 (+1 ~ +2,000,000,000 Pulse)

回應：
< 0 (負值) 向量速度設定太大，回傳圓弧最大的向量速度值(PPS)
> 0 (正值) 設定正確，回傳執行 3D 圓弧耗費的時間(ms)
如有錯誤發生(欲讀錯誤碼請搭配使用 **i8094H_GET_ERROR_CODE**)

```

範例:  i8094H_MIX_3D_INITIAL(
CardNo, AXIS_X, AXIS_Y, AXIS_Z, 100000 , 100000, 100000);
long P1[3], P2[3];
long rec1, rec2, rec3, rec4;
long lp1, lp2, lp3, lp4;
DWORD t0, t1, t;
P1[0] = 10000;    P1[1] = 10000;    P1[2] = 10000;
rec1 = i8094H_MIX_3D_CONTINUE(CardNo, 0, 0, P1, P2, 1);

// max RTC speed = 5KHz
P1[0] = -10000;    P1[1] = 10000;    P1[2] = 5000;
P2[0] = -10000;    P2[1] = -10000;    P2[2] = 10000;
rec2 = i8094H_MIX_3D_CONTINUE(CardNo, 0, 1, P1, P2, 1);

// max RTC speed = 5KHz
P1[0] = -10000;    P1[1] = 10000;    P1[2] = 5000;
P2[0] = -10000;    P2[1] = -10000;    P2[2] = 10000;
rec3 = i8094H_MIX_3D_CONTINUE(CardNo, 0, 1, P1, P2, 1);

P1[0] = -10000;    P1[1] = -10000;    P1[2] = -10000;
rec4 = i8094H_MIX_3D_CONTINUE(CardNo, 1, 0, P1, P2, 1);

loop:
//Sleep(100);
if (i8094H_STOP_WAIT(CardNo, AXIS_XYZU) == NO) goto loop;

lp1 = i8094H_GET_LP(CardNo, AXIS_X);
lp2 = i8094H_GET_LP(CardNo, AXIS_Y);
lp3 = i8094H_GET_LP(CardNo, AXIS_Z);
lp4 = i8094H_GET_LP(CardNo, AXIS_U);

```


6.5 其他功能

6.5.1 設定軸暫停

- ****void i8094H_DRV_HOLD(BYTE cardNo, BYTE 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 i8094H_DRV_START(BYTE cardNo, BYTE 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。
i8094H_DRV_HOLD(cardNo, AXIS_XYU); //設定 XYU 三軸暫停移動。

i8094H_SET_MAX_V(cardNo, AXIS_U, 10000); //設定 U 軸最高速 10K PPS。
i8094H_NORMAL_SPEED(cardNo, AXIS_U, 0); //設定 U 軸對稱 T 曲線。
i8094H_SET_V(cardNo, AXIS_U, 2000); //設定 U 軸速度=2,000 PPS。
i8094H_SET_A(cardNo, AXIS_U,1000); //設定 U 軸加速度 1,000 PPS/S。
i8094H_SET_SV(cardNo, AXIS_U, 2000); //設定 U 初始速度 2,000 PPS。
i8094H_SET_AO(cardNo, AXIS_U, 9); // U 軸減速(保留脈波數)= 9 PPS。
i8094H_SET_MAX_V(cardNo, AXIS_XY, 20000); //設定 XY 最高速 20K PPS。
i8094H_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。
i8094H_VECTOR_SPEED(cardNo, 0); //二軸直線固定速度 VSV=VV，設 VV 即可。
i8094H_SET_VV(cardNo, 5000); //設定第 1 卡，向量定速度為 5,000 PPS。

i8094H_FIXED_MOVE(cardNo, AXIS_U, 5000); //U 移動 5,000 Pulse。
i8094H_LINE_2D(cardNo, 12000, 10000); //執行直線補間移動。

i8094H_DRV_START(cardNo, AXIS_XYU); //開始 XYU 三軸同時移動。
```

6.5.3 等待完成軸運動

- **BYTE i8094H_STOP_WAIT(BYTE cardNo, BYTE 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。**
i8094H_SET_MAX_V(cardNo, AXIS_XYZU, 20000); //設定最高速 20K PPS。
i8094H_NORMAL_SPEED(cardNo, AXIS_XYZU, 0); //設定 XYZU 軸對稱 T 曲線。
i8094H_SET_V(cardNo, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094H_SET_A(cardNo, AXIS_XYZU,1000); //設定 XYZU 軸加速度 1000 PPS/S。
i8094H_SET_SV(cardNo, AXIS_XYZU, 2000); //設定 XYZU 初始速度 2000 PPS。
i8094H_SET_AO(cardNo, AXIS_XYZU, 9); // XYZU 軸減速(保留脈波數)= 9 PPS。
i8094H_FIXED_MOVE(cardNo, AXIS_XYZU, 10000); //XYZU 移動 10000 Pulse。

```
if (i8094H_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(補間)軸停止

- ***void i8094H_STOP_SLOWLY(BYTE cardNo, BYTE 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: `i8094H_STOP_SLOWLY(1, AXIS_XY);`
`//設定第 1 卡 XY 軸, 減速停止。`

- ***void i8094H_STOP_SUDDENLY(BYTE cardNo, BYTE 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: `i8094H_STOP_SUDDENLY(1, AXIS_ZU);`
`//設定第 1 卡 ZU 軸, 立即(緊急)停止。`

- ***void i8094H_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: `i8094H_VSTOP_SLOWLY(1);`
`//設定第 1 卡補間軸, 減速停止。`

- ***void i8094H_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: `i8094H_VSTOP_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 i8094H_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: `i8094H_VSTOP_SUDDENLY(1);`
`i8094H_CLEAR_STOP(1);`
`//清除第 1 卡錯誤狀態。`

6.5.6 補間動作結束

- ***void i8094H_INTP_END(BYTE cardNo, BYTE 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: `i8094H_INTP_END(1, 0);`
`//設定第 1 卡二軸補間，動作結束。`

7 i8094H 專用特殊功能

- **i8094H** 開機後系統的內定值如下:

```
i8094H_SET_PULSE_MODE(cardNo, AXIS_XYZU, 0);  
i8094H_SET_MAX_V(cardNo, AXIS_XYZU, 200000L);  
i8094H_SET_HLMT(cardNo, AXIS_XYZU, 0, 0);  
i8094H_LIMITSTOP_MODE(cardNo, AXIS_XYZU, 0);  
i8094H_SET_NHOME(cardNo, AXIS_XYZU, 0);  
i8094H_SET_HOME_EDGE(cardNo, AXIS_XYZU, 0);  
i8094H_CLEAR_SLMT(cardNo, AXIS_XYZU);  
i8094H_SET_ENCODER(cardNo, AXIS_XYZU, 0, 0, 0);  
i8094H_SERVO_OFF(cardNo, AXIS_XYZU);  
i8094H_SET_ALARM(cardNo, AXIS_XYZU, 0, 0);  
i8094H_SET_INPOS(cardNo, AXIS_XYZU, 0, 0);  
i8094H_SET_FILTER(cardNo, AXIS_XYZU, 0, 0);  
i8094H_VRING_DISABLE(cardNo, AXIS_XYZU);  
i8094H_AVTRI_DISABLE(cardNo, AXIS_XYZU);  
i8094H_EXD_DISABLE(cardNo, AXIS_XYZU);
```

- 可以使用在下載參數表(參考 7.1)中的函式如下:

```
i8094H_SET_PULSE_MODE  
i8094H_SET_MAX_V  
i8094H_SET_HLMT  
i8094H_LIMITSTOP_MODE  
i8094H_SET_NHOME  
i8094H_SET_HOME_EDGE  
i8094H_SET_SLMT (i8094H_CLEAR_SLMT)  
i8094H_SET_ENCODER  
i8094H_SERVO_ON (i8094H_SERVO_OFF)  
i8094H_SET_ALARM  
i8094H_SET_INPOS  
i8094H_SET_FILTER  
i8094H_VRING_ENABLE (i8094H_VRING_DISABLE)  
i8094H_AVTRI_ENABLE (i8094H_AVTRI_DISABLE)
```

- 可以將函式的返回值(參考 7.2.3)設為變數值如下:

i8094H_GET_LP

i8094H_GET_EP

i8094H_GET_DI

i8094H_GET_DI_ALL

i8094H_GET_ERROR

i8094H_GET_ERROR_CODE

i8094H_FRNET_DI

i8094H_HOME_WAIT

i8094H_GET_LATCH

i8094H_STOP_WAIT

7.1 初始化參數表

可使用下面兩種方法來下載參數表：

- 需配合 i8094H_Tools 工具程式下載參數表到 i8094H，詳細操作內容請參考 i8094H_Getting_Started_1.1tc.pdf 手冊。

- **void i8094H_LOAD_INITIAL(BYTE cardNo)**

功能： 亦可使用函式將設定寫入 i8094H 的參數表(參考表 2-1a)，再利用應用程式使用 i8094H_LOAD_INITIAL，令 i8094H 套用參數表的初始化設定值。

參數： **cardNo:** 指定卡號

回應： 無

範例：
i8094H_SET_PULSE_MODE(1, INITIAL_XYZU, 0);
//將第 1 卡 X Y Z U 軸，脈波輸出模式皆設為 0，寫入參數表。
i8094H_SET_MAX_V(1, INITIAL_XY, 200000L);
//將第 1 卡 X Y 軸，最高速為 200K PPS，寫入參數表。
i8094H_SET_HLMT(1, INITIAL_XYZU, 0, 0);
//將第 1 卡 X Y Z U 軸，其"前後極限"觸發邏輯，全部為低準位觸發寫入參數表。
i8094H_LIMITSTOP_MODE(1, INITIAL_X, 0);
//將第 1 卡 X 軸，碰觸前後極限後立即停止，寫入參數表。
i8094H_SET_NHOME(1, INITIAL_XY, 0);
//將第 1 卡 X Y 軸，其"近原點"開關，觸發邏輯全部為低準位觸發，寫入參數表。

i8094H_LOAD_INITIAL(1);
//令 i8094H 套用參數表的初始化設定值。

7.2 巨集程式的創建

7.2.1 建立巨集程式碼

- **void i8094H_MP_CREATE(BYTE cardNo, BYTE mpNo)**

- **void i8094H_MP_CREATE(BYTE cardNo, BYTE isrNo)**

功能： 在 i8094H 卡上自建巨集程式，然後利用 i8094H_MP_CALL 執行。

1. 所有建在 i8094H 卡上的巨集程式碼，皆能斷電保持資料。
2. 不可使用在巨集程式內。

參數： **cardNo:** 指定卡號
mpNo: 指定巨集程式的編號(MP1 ~ MP157)
isrNo: 指定中斷服務程式的編號(ISR1 ~ ISR20)

回應： 無

範例： i8094H_MP_CREATE(1, MP21); //開始將巨集程式寫入 i8094H 中。

```
//=====
//下面這些程式並不會被執行，而是將程式碼寫入 i8094H 中待命。
i8094H_SET_MAX_V(1, AXIS_XYZU, 20000); //設定軸最高速 20K PPS。
i8094H_NORMAL_SPEED(1, AXIS_XYZU, 0); //設定 XYZU 為對稱 T 曲線。
i8094H_SET_V(1, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
i8094H_SET_A(1, AXIS_XYZU, 1000); //設定 XYZU 軸加速度 1000 PPS/S。
i8094H_SET_SV(1, AXIS_XYZU, 2000); //設定 XYZU 初始速度 2000 PPS。
i8094H_SET_AO(1, AXIS_XYZU, 9); //XYZU 軸減速(保留脈波數)= 9 PPS。
i8094H_FIXED_MOVE(1, AXIS_XYZU, 10000); //XYZU 移動 10000 Pulse。
i8094H_MP_CLOSE(1); //結束第 1 卡，巨集程式寫入 i8094H 中。
//=====
```

- ***void i8094H_MP_CLOSE(BYTE cardNo)**

功能： 結束巨集程式寫入 i8094H，需搭配 i8094H_MP_CREATE 使用。

參數： **cardNo:** 指定卡號

回應： 無

範例： 同上

7.2.2 執行巨集程式

- ***void i8094H_MP_CALL(BYTE cardNo, BYTE mpNo)**

功能: i8094H 將執行指定編號的巨集程式，MP 最多可以 CALL 7 層，ISR 只有 1 層。

參數: **cardNo:** 指定卡號
mpNo: 指定巨集程式的編號(MP1 ~ MP157)

回應: 無

範例: i8094H_MP_CREATE(1, MP21); //開始將 21 號巨集程式寫入 i8094H 中。

```
//=====
```

```
//下面這些程式並不會被執行，而是將程式碼寫入 i8094H 中待命。
```

```
i8094H_SET_MAX_V(1, AXIS_XYZU, 20000); //設定軸最高速 20K PPS。
```

```
i8094H_NORMAL_SPEED(1, AXIS_XYZU, 0); //設定 XYZU 為對稱 T 曲線。
```

```
i8094H_SET_V(1, AXIS_XYZU, 2000); //設定 XYZU 軸速度=2000 PPS。
```

```
i8094H_SET_A(1, AXIS_XYZU, 1000); //設定 XYZU 軸加速度 1000 PPS/S。
```

```
i8094H_SET_SV(1, AXIS_XYZU, 2000); //設定 XYZU 初始速度 2000 PPS。
```

```
i8094H_SET_AO(1, AXIS_XYZU, 9); //XYZU 軸減速(保留脈波數)= 9 PPS。
```

```
i8094H_FIXED_MOVE(1, AXIS_XYZU, 10000); //XYZU 移動 10000 Pulse。
```

```
i8094H_MP_CLOSE(1); //結束第 1 卡，0 號巨集程式寫入 i8094H 中。
```

```
//=====
```

```
i8094H_MP_CALL(1, MP21); //呼叫第 1 卡 i8094H，執行 21 號巨集程式。
```

7.2.3 自定義變數(Variable)

- ****void i8094H_MP_SET_VAR(BYTE cardNo, long varNo, long varNo1)**
- ****void i8094H_MP_SET_VAR(BYTE cardNo, long varNo, long IVar1)**

功能: 設定變數值 VARn = VARn。
設定變數值 VARn = 立即資料。

參數: **cardNo:** 指定卡號
varNo: 巨集程式可使用的變數: VAR0 ~ VAR511
varNo1: 使用變數值: VAR0 ~ VAR511
IVar1: 立即資料 (-2,000,000,000 ~ +2,000,000,000)

回應: 無

範例: 請參考 i8094H_MP_VAR_CALCULATE 的範例。

- ****void i8094H_MP_SET_RVAR(BYTE cardNo, long varNo)**

功能: 設定變數值 VARn = 函式的返回值自動轉為(long)。

參數: **cardNo:** 指定卡號
varNo: 巨集程式可使用的變數: VAR0 ~ VAR511

回應: 無

範例: 請參考 i8094H_MP_VAR_CALCULATE 的範例。

- ****void i8094H_MP_VAR_CALCULATE(**
BYTE cardNo, long varNo, BYTE Operator, long varNo1, long varNo2)
- ****void i8094H_MP_VAR_CALCULATE(**
BYTE cardNo, long varNo, BYTE Operator, long IVar1, long varNo2)

功能: 變數計算, $varNo (+-*/) varNo1 = varNo2$ 。
 變數計算, $varNo (+-*/)$ 立即資料 = $varNo2$ 。

參數: **cardNo:** 指定卡號
varNo: 被運算的變數: VAR0 ~ VAR511
Operator: ‘+’ 加法
 ‘-’ 減法
 ‘*’ 乘法
 ‘/’ 除法(無條件捨去, 取整數位)
varNo1: 立即計算的變數值: VAR0 ~ VAR511
varNo2: 計算結果存入的變數: VAR0 ~ VAR511
IVar1: 立即資料 (-2,000,000,000 ~ +2,000,000,000)

回應: 無

範例: **i8094H_MP_CREATE(1, MP100); //開始將 100 號巨集程式寫入 i8094H 中。**
//=====
//下面這些程式並不會被執行, 而是將程式碼寫入 i8094H 中待命。
i8094H_MP_SET_VAR(1, VAR1, 100); //VAR1 = 100。
i8094H_MP_SET_VAR(1, VAR2, 200); //VAR2 = 200。
i8094H_MP_SET_VAR(1, VAR10, 10); //VAR10 = 10。
//-----
//VAR6 = i8094H_FRNET_RA(1, 8)
i8094H_FRNET_RA(1, 8);
i8094H_MP_SET_RVAR(1, VAR6); //VAR6 = 第 1 卡 RA8 目前輸入。
//-----
//VAR5 = i8094H_GET_LP(1, AXIS_X)
i8094H_GET_LP(1, AXIS_X); //讀取第 1 卡 X 軸目前指令位置。
i8094H_MP_SET_RVAR(1, VAR5); //VAR5 = 第 1 卡 X 軸目前指令位置值。
//-----
i8094H_MP_VAR_CALCULATE(1, VAR1, '+', VAR2, VAR3);
//VAR1 + VAR2 = VAR3。
i8094H_MP_VAR_CALCULATE(1, VAR3, '-', VAR1, VAR3);
//VAR3 - VAR1 = VAR3。
i8094H_MP_VAR_CALCULATE(1, VAR3, '*', VAR10, VAR2);
//VAR3 x VAR10 = VAR3。
i8094H_MP_VAR_CALCULATE(1, VAR3, '/', VAR2, VAR1);
//VAR3 / VAR2 = VAR1。
//最終結果 VAR1 = 10、VAR2 = 200、VAR3 = 2,000、VAR10 = 10。
i8094H_MP_CLOSE(1); //結束第 1 卡, 100 號巨集程式寫入 i8094H 中。
//=====

7.2.4 程序循環(FOR~NEXT)

- ***void i8094H_MP_FOR(BYTE cardNo, long varNo)**
- ***void i8094H_MP_FOR(BYTE cardNo, long IVar)**

功能： 程序執行循環命令，最多可以處理 7 層。

參數： **cardNo:** 指定卡號
varNo: 巨集程式可使用的變數: VAR0 ~ VAR511
IVar: 立即資料 (-2,000,000,000 ~ +2,000,000,000)

回應： 無

範例： **i8094H_MP_CREATE(1, MP100); //開始將 100 號巨集程式寫入 i8094H 中。**
//=====
//下面這些程式並不會被執行，而是將程式碼寫入 i8094H 中待命。
i8094H_SET_MAX_V(1, AXIS_X, 20000); //設定軸最高速 20K PPS。
i8094H_NORMAL_SPEED(1, AXIS_X, 0); //設定 X 軸為對稱 T 曲線。
i8094H_SET_V(1, AXIS_X, 2000); //設定 X 軸速度 2,000 PPS。
i8094H_SET_A(1, AXIS_X, 1000); //設定 X 軸加速度 1,000 PPS/S。
i8094H_SET_SV(1, AXIS_X, 2000); //設定 X 軸初始速度 2,000 PPS。
i8094H_SET_AO(1, AXIS_X, 0); //X 軸減速(保留脈波數)= 0 PPS。
//-----
i8094H_MP_SET_VAR(1, VAR1, 100); //VAR1 = 100。
i8094H_MP_FOR(1, VAR1);
//令 X 軸前後移動 1,000 Pulse，循環 100 次。
//亦可直接傳入立即運算值 i8094H_MP_FOR(1, 100)。
i8094H_FIXED_MOVE(1, AXIS_X, 1000); //X 軸移動 1,000 Pulse。
i8094H_FIXED_MOVE(1, AXIS_X, -1000); //X 軸移動-1,000 Pulse。
i8094H_MP_NEXT(1);
i8094H_MP_CLOSE(1); //結束第 1 卡，100 號巨集程式寫入 i8094H 中。
//=====
i8094H_MP_CALL(1, MP100); //呼叫第 1 卡 i8094H，執行 100 號巨集程式。

- ***void i8094H_MP_NEXT(BYTE cardNo)**

功能： 需搭配 i8094H_MP_FOR 使用。

參數： **cardNo:** 指定卡號

回應： 無

範例： 同上

7.2.5 程序條件(IF~ELSE)

- ****void i8094H_MP_IF(BYTE cardNo, long varNo, WORD Operator, long varNo1)**
- ****void i8094H_MP_IF(BYTE cardNo, long varNo, WORD Operator, long IVar1)**

功能: 程序條件敘述式，MP 最多可以處理 7 層，ISR 只有 1 層。

參數: **cardNo:** 指定卡號
varNo: 運算的變數: VAR0 ~ VAR511
Operator: ‘<’ 小於
 ‘>’ 大於
 ‘<=’ 小於等於
 ‘>=’ 大於等於
 ‘==’ 等於
 ‘!’ 不等於
(Operator: 以”單引號”表示的單一或複合運算子)
varNo1: 條件變數: VAR0 ~ VAR511
IVar1: 立即資料 (-2,000,000,000 ~ +2,000,000,000)

回應: 無

範例: `i8094H_MP_CREATE(1, MP100); //開始將 100 號巨集程式寫入 i8094H 中。`
`//=====`
`//下面這些程式並不會被執行，而是將程式碼寫入 i8094H 中待命。`
`i8094H_SET_MAX_V(1, AXIS_X, 20000); //設定軸最高速 20K PPS。`
`i8094H_NORMAL_SPEED(1, AXIS_X, 0); //設定 X 軸為對稱 T 曲線。`
`i8094H_SET_V(1, AXIS_X, 2000); //設定 X 軸速度 2,000 PPS。`
`i8094H_SET_A(1, AXIS_X, 1000); //設定 X 軸加速度 1,000 PPS/S。`
`i8094H_SET_SV(1, AXIS_X, 2000); //設定 X 初始速度 2,000 PPS。`
`i8094H_SET_AO(1, AXIS_X, 0); //X 軸減速(保留脈波數)= 0 PPS。`
`//-----`
`i8094H_MP_SET_VAR(1, VAR1, 100); //VAR1 = 100。`
`i8094H_MP_SET_VAR(1, VAR2, 200); //VAR2 = 200。`
`i8094H_MP_IF(1, VAR1, '<', VAR2);`
`i8094H_FIXED_MOVE(1, AXIS_X, 1000); //X 軸移動 1,000 Pulse。`
`i8094H_MP_ELSE(1);`
`i8094H_FIXED_MOVE(1, AXIS_X, -1000); //X 軸移動-1,000 Pulse。`
`i8094H_MP_IF_END(1);`
`i8094H_MP_CLOSE(1); //結束第 1 卡，100 號巨集程式寫入 i8094H 中。`
`//=====`
`i8094H_MP_CALL(1, MP100); //呼叫第 1 卡 i8094H，執行 100 號巨集程式。`

● ****void i8094H_MP_ELSE(BYTE cardNo)**

功能： 需搭配 i8094H_MP_IF 使用。

參數： **cardNo:** 指定卡號

回應： 無

範例： 同上

● ****void i8094H_MP_IF_END(BYTE cardNo)**

功能： 結束 i8094H_MP_IF 程序。

參數： **cardNo:** 指定卡號

回應： 無

範例： 同上

7.2.6 延時器(TIMER)

- ***void i8094H_MP_TIMER(BYTE cardNo, long varNo)**
- ***void i8094H_MP_TIMER(BYTE cardNo, long ms)**

功能： 程序延時命令。

參數： **cardNo:** 指定卡號
varNo: 巨集程式可使用的變數: VAR0 ~ VAR511
ms: 立即資料 (0 ~ +2,000,000,000) ms

回應： 無

範例： **i8094H_MP_TIMER(1, 200);** //程序延時 200 ms。

7.2.7 等待軸運動停止(MP專用)

- ***void i8094H_MP_STOP_WAIT(BYTE cardNo, BYTE axis)**

功能： 等待軸完全停止，MP 的程序才能再繼續執行。

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

回應： 無

範例： 無

7.2.8 使用者定義RINT

- ****void i8094H_MP_SET_RINT(BYTE cardNo)**

功能: 使用者定義 RINT 的設定，當 MP 執行到此 Function，軸卡便會對 Controller 發出 **0x04** 的中斷。

參數: **cardNo:** 指定卡號

回應: 無

範例: 無