



ADLINK
TECHNOLOGY INC.

AMP-204C / AMP-208C

DSP-based 4/8 軸高階脈衝式運動控制卡

使用手冊



手冊版本 : 1.00

更新日期 : August 15, 2013

料號 : 50-15089-1000



Recycled Paper

Advance Technologies; Automate the World.

更動記錄

更動	日期	更動內容
1.00	2013-08-15	初版



請注意，本文件為 AMP-204C / AMP-208C 使用手冊的初版。本公司雖已盡全力確使文件內容在出版當時均屬精準，後續版本仍可能對本文件內規格及操作有或大或小的修改，甚或本文件所無的全新章節。
如需其他資料或有任何問題，請瀏覽本公司網站
(<http://www.adlinktech.com>) 或洽服務資訊所列的銷售據點。

前言

Copyright 2013 ADLINK Technology, Inc.

本文件內容受版權法保護。版權所有。非經事前書面許可，嚴禁對本文件之任何部分以任何機械、電子、或任何其他方式進行任何形式之複製。

免責聲明

本文件內容如因可靠性、設計、及功能改善而需進行任何變更，恕不另行通知。本文件內容不代表本公司的義務。

不論在任何情況下，本公司對任何因使用或無法使用本產品或本文所導致之直接、間接、特殊、偶然、或必然之損害不負任何責任，即使已知有此種損害產生的可能性亦然。

環境責任

凌華科技遵守歐盟之 Restriction of Hazardous Substances (RoHS) 規範及 Waste Electrical and Electronic Equipment (WEEE) 規範，致力於承擔其全球環境保護之責。環境保護是凌華科技的第一優先原則。本公司已採取確保其產品、製程、零件、及原料對環境衝擊降至盡可能最低程度的措施。本公司已鼓勵其客戶按各國家或公司之產品處理及 / 或回收規定，進行其產品的處理。

商標

本文件內所提及之產品名稱僅供辨識之用，且可能為各該公司之商標及 / 或註冊商標。

慣例

請注意本下列文件所用到的慣例，以確保使用者正確執行相關的工作及指令。



註釋

其他有助於使用者執行其作業的資訊、協助、及技巧。



注意

執行作業時，有關防止**輕微**人身傷害、零件損壞、資料遺失、及 / 或程式損毀的資訊。



警告

執行作業時，有關防止**嚴重**人身傷害、零件損壞、資料遺失、及 / 或程式損毀的資訊。

目錄

更動記錄 ii

前言	iii
圖目錄	ix
表目錄	xiii
1 簡介	1
1.1 產品規格	4
1.2 軟體支援	8
軟體支援函式庫	8
MotionCreatorPro 2	8
1.3 端子板	8
2 開始安裝	9
2.1 確認包裝內項目	9
2.2 AMP-204C / AMP-208C 外觀輪廓圖	10
2.3 硬體安裝	12
硬體組態	12
硬體安裝程序	12
問題排除	13
2.4 軟體安裝流程	14
2.5 主要連接器訊號定義	16
AMP-204C : P1 連接器	16
AMP-208C : P1-A/B 連接器	18
AMP-204C/208C : P2 連接器	21
2.6 DIP Switch	23
SW2 : 卡片編號開關 (Card ID Switch)	23
2.7 IDE 44p – DSUB 37p 排線	24
2.8 專用端子板 - DIN-825-GP4	25

連接器定義	26
P1 連接器：可連接至 PCI-8254 / PCI-8258 / AMP-204C / AMP-208C	28
S1, S2 : EDO/ALM_RST 選擇開關	37
3 訊號連接介面	39
3.1 脈衝控制命令訊號 (Pulse Command):	40
3.2 編碼器輸入訊號 (Encoder Input, EA & EB & EZ):	43
3.3 緊急停止訊號 (Emergency Stop Input)	45
3.4 機械極限開關訊號 (PEL/MEL signals Input)	46
3.5 原點開關訊號 (ORG signals Input)	48
3.6 到位 / 零速度檢出訊號 (INP signals Input)	49
3.7 伺服警報訊號 (ALM signals Input)	50
3.8 伺服激磁訊號 (SVON signals Output)	51
3.9 比較觸發訊號 (Comapre & Trigger Output):	52
3.10 泛用數位輸出入訊號 (Digital signals Output/Input)	54
4 運動控制功能原理介紹	59
4.1 運動控制模式暨介面簡介	60
4.1.1 運動控制介面 (Motion Control Interface)	60
4.1.2 控制週期 (Control Cycle)	66
4.2 運動控制操作 (Motion Control Operations)	68
4.2.1 座標系統 (Coordinated System)	68
4.2.2 單位轉換因子 (Unit Factor)	69
4.2.3 加減速曲線 (Acc/Deceleration Profile)	72
4.3 歸零運動 (Home Move)	78
4.3.1 ORG 訊號定位 - Home Mode 0	81
4.3.2 EL 訊號定位 - Home Mode 1	88
4.3.3 單一 EZ 訊號定位	91
4.4 速度運動 (Velocity Move)	94
4.5 JOG 運動 (Jog Move)	97
4.6 點對點運動 (Point-to-Point Move)	101
4.6.1 點對點運動 (Point-to-Point Move)	101

4.6.2	同時啟動 (Synchronous Start)	102
4.6.3	動態運動參數切換 (On The Fly Change)	103
4.6.4	連續 PTP 運動 (Continuous PTP Move)	103
4.7	多軸補間運動 (Interpolation)	106
4.7.1	直線補間 (Linear Interpolation)	106
4.7.2	圓弧補間 (Arc Interpolation)	108
4.7.3	連續補間運動 (Continuous Interpolation)	116
4.8	運動狀態監控 (Motion Status Monitoring)	122
4.8.1	運動狀態 (Motion Status)	123
4.9	應用功能 (Application Functions)	132
4.9.1	電子齒輪 (Electronic Gearing)	132
4.9.2	高速位置比較觸發 (High Speed Position Compare Trigger)	134
4.9.3	PWM 控制 (雷射控制) (VAO Table Control)	140
4.9.4	運動控制暨 I/O 取樣功能 (Sampling function)	148
4.9.5	同動控制 (Simultaneous Move)	153
4.9.6	點表運動	156
4.10	安全保護 (Safety Protection)	161
4.10.1	硬體保護 (Hardware Protection)	161
4.10.2	軟體保護 (Software Protection)	164
4.11	本機中斷 (Host Interrupt)	168
重要安全資訊	177	
服務資訊	179	

preliminary

圖目錄

圖 1-1 :	AMP-204C/208C 系統方塊圖	2
圖 1-2 :	系統安裝流程圖	3
圖 2-1 :	AMP-204C 產品外觀輪廓圖	10
圖 2-2 :	AMP-208C 產品外觀輪廓圖	11
圖 2-3 :	DIN-825-GP4 外觀圖	25
圖 2-4 :	DIN-825-GP4 機構圖	26
圖 3-1 :	脈衝控制命令訊號 Line Driver 型連接範例圖	41
圖 3-2 :	脈衝控制命令訊號 Open-Collector 型連接範例圖	42
圖 3-3 :	編碼器輸入訊號 Line Driver 型連接範例圖	44
圖 3-4 :	緊急停止訊號連接範例圖	45
圖 3-5 :	機械極限開關訊號連接範例圖	47
圖 3-6 :	原點開關訊號連接範例圖	48
圖 3-7 :	到位檢出訊號連接範例圖	49
圖 3-8 :	伺服警報訊號連接範例圖	50
圖 3-9 :	伺服激磁訊號連接範例圖	51
圖 3-10 :	比較觸發訊號 Line Driver 型連接範例圖	52
圖 3-11 :	比較觸發訊號 Open-Collector 型連接範例圖	53
圖 3-12 :	泛用數位輸出入訊號連接範例圖	55
圖 3-13 :	泛用數位輸出入訊號連接範例圖	58
圖 4-1 :	脈衝式訊號格式	61
圖 4-2 :	控制週期	67
圖 4-3 :	控制器座標系統方塊圖	68
圖 4-4 :	梯型速度曲線之速度 / 加速度 / 加加速度對時間 關係圖	72
圖 4-5 :	自動規劃最大速度示意圖	73
圖 4-6 :	S 形速度曲線之速度 / 加速度 / 加加速度 對時間 關係圖	74
圖 4-7 :	自動規劃最大速度示意圖	76
圖 4-8 :	Home mode 0 (Case: ORG)	82
圖 4-9 :	Home mode 0 (Case: ORG)	84
圖 4-10 :	Home mode 0 (Case: ORG)	85
圖 4-11 :	Home mode 0 反向 (Case: ORG+EZ)	86
圖 4-12 :	Home mode 0 減速停止 (Case: ORG)	87
圖 4-13 :	Home mode 1 (Case: EL)	88
圖 4-14 :	Home mode 1 (Case: EL+EZ)	90
圖 4-15 :	Home mode 2 (Case: EZ)	92
圖 4-16 :	Home mode 2 反向 (Case: EZ)	93
圖 4-17 :	JOG 運動的 V-T 圖與 JOG-ON 訊號的關係圖	97

圖 4-18 :	Jog 時進模式示意圖	98
圖 4-19 :	T-curve V-T 速度曲線圖	101
圖 4-20 :	動態改變位置速度示意圖	103
圖 4-21 :	三段連續位置 V-T 圖	104
圖 4-22 :	三段連續位置 V-T 圖 (速度自動連接 (1))	104
圖 4-23 :	三段連續位置 V-T 圖 (速度自動連接 (2))	104
圖 4-24 :	三段連續位置 V-T 圖 (速度自動連接 (3))	105
圖 4-25 :	三段連續位置 V-T 圖 (速度自動連接 (4))	105
圖 4-26 :	二維直線補間運動	107
圖 4-27 :	三維圓弧補間運動 (方法一)	109
圖 4-28 :	空間中法向量定義	110
圖 4-29 :	空間中決定圓弧方向	110
圖 4-30 :	三維圓弧補間運動 (方法二)	111
圖 4-31 :	三維圓弧補間範例圖	112
圖 4-32 :	三維螺旋補間運動 (方法一)	113
圖 4-33 :	三維螺旋補間運動 (方法二)	114
圖 4-34 :	連續補間運動 (Buffer) 示意圖	116
圖 4-35 :	速度混合 (方法一)	117
圖 4-36 :	速度混合 (方法二)	118
圖 4-37 :	速度混合 (方法三)	118
圖 4-38 :	速度混合 (方法四)	119
圖 4-39 :	速度混合 (方法五)	119
圖 4-40 :	速度混合 (方法六)	120
圖 4-41 :	速度混合 (方法七)	120
圖 4-42 :	連續補間運動範例圖	121
圖 4-43 :	運動狀態監視流程圖	122
圖 4-44 :	不同運動訊號與運動關係圖	125
圖 4-45 :	Motion Done (MDN) 訊號與運動關係圖	126
圖 4-46 :	Motion Done (MDN), In-homing (HMV) 訊號與運動關係圖	127
圖 4-47 :	WAIT 訊號與運動關係圖	128
圖 4-48 :	JOG, Motion Done (MDN) 訊號與運動關係圖	129
圖 4-49 :	異常停止訊號 (ASTP) 與運動關係圖	129
圖 4-50 :	擬和運動 (BLD) 訊號與運動關係圖	130
圖 4-51 :	前 / 後位移量事件訊號與運動關係圖	131
圖 4-52 :	電子齒輪自動接和速率調整	133
圖 4-53 :	比較觸發方塊圖	135
圖 4-54 :	線性比較觸發範例圖	137
圖 4-55 :	表格比較觸發範例圖	138
圖 4-56 :	表格比較觸發方塊圖	139

圖 4-57 :	訊號取樣架構圖	148
圖 4-58 :	中斷流程圖	168

preliminary

preliminary

表目錄

表 1-1 :	脈衝型伺服驅動器專用纜線對照表.....	8
表 4-1 :	編碼器輸入格式	63
表 4-2 :	編碼器輸入格式	63
表 4-3 :	卡片參數表 (Board parameter table).....	146
表 4-4 :	運動核心 (Motion kernel) 訊號表.....	149

preliminary

This page intentionally left blank.

preliminary

1 簡介

AMP-204C / AMP-208C 為凌華科技 DSP-based 高階運動控制卡，可支援 4/8 軸脈衝 (Pulse type) 訊號命令，提供開迴路 (Open-loop) 控制選擇，可支援各式伺服驅動器上位置 (Position) 命令。

AMP-204C / AMP-208C 藉由高速 PCI 匯排流 (Bus) 完成與作業系統 (operating system) 之運動控制命令、回授資料及參數調整等資料交換。搭配凌華科技自製 Softmotion 核心，可提供即時 T/S 速度曲線規劃 (T/S speed profile planning)、點對點運動 (Point-to-Point move)、多軸補間運動 (Multi-dimension interpolation move)、主 / 從軸控制 (Master/Slave motion) 等豐富運動控制函式。

下圖一為 AMP-204C/208C 系統方塊圖，AMP-204C/208C 採用 TI (Texas Instrument) DSP (Digital Signal Processor) 作為主運算單元，同時整合高速大容量 FPGA (Field programmable Gate Array)，高速編碼器輸出單元，2/4 組高速位置比對及脈衝輸出 (Position compare and trigger output)，運動及泛用 I/O 輸出入 (Motion & general purposed I/O) 及邏輯控制。為了避免錯誤的配線燒毀 AMP-204C/208C，故將大部分光隔離電路設計至專用端子板 DIN-825-GP4。而藉由各式凌華科技所提供之抗撓曲線材，可輕易與市售知名伺服驅動器 (servo drive) 連接。

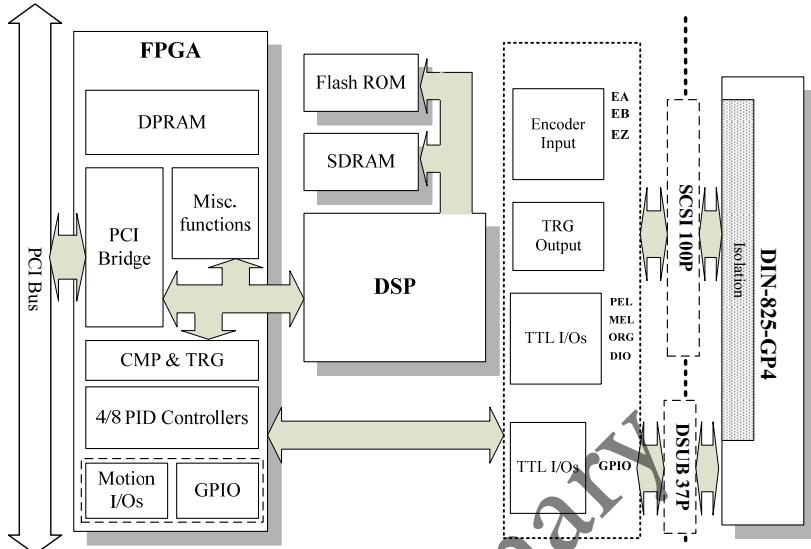


圖 1-1 : AMP-204C/208C 系統方塊圖

圖形化運動控制介面 – “MotionCreatorPro 2”為 Windows-based 運動控制軟體開發工具。藉由即時性的圖形化介面可完成運動控制狀態以及 I/O 觀測。可同步擷取運動量曲線及資料以進行分析。內含 “Setup Wizard” 指引使用者逐步完成硬體安裝、硬體配線及單軸操作。此工具可有效節省使用者的開發時程及成本。

Windows Programming Libraries 支援 Windows 編譯環境，如 Visual Studio C++ 6.0, Microsoft .NET framework 的 VB.NET, C++, 以及 Borland 的 C++Builder 等。使用者亦可在安裝資料夾中找到範例程式。下圖二之流程圖將指導您如何使用本手冊，並快速有效的找到對應章節。

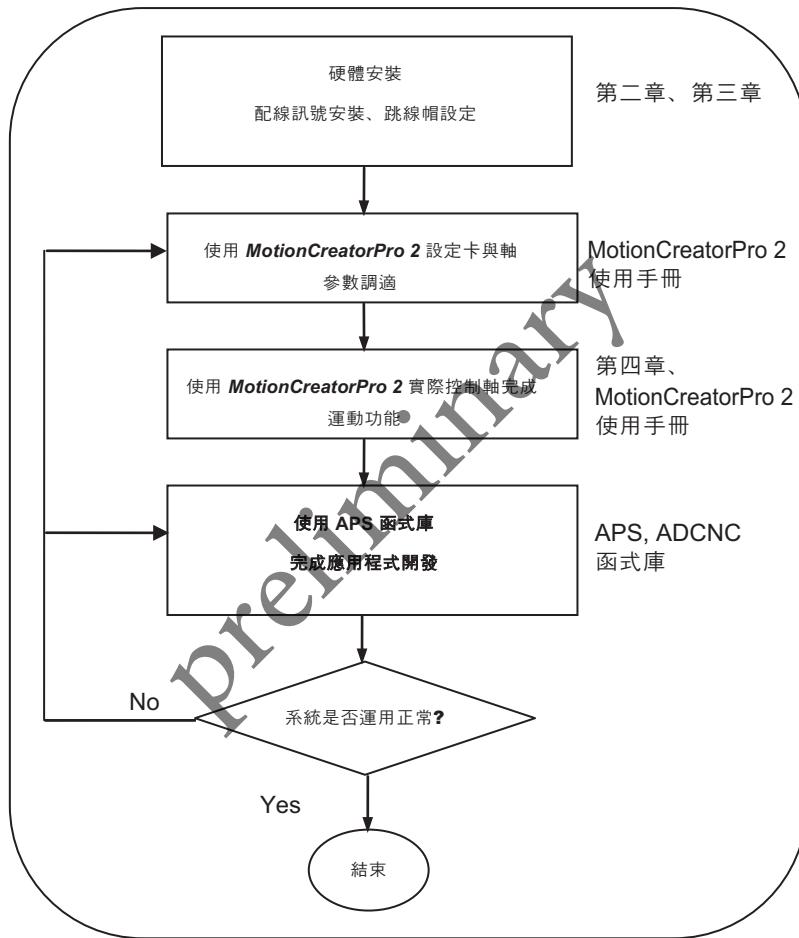


圖 1-2：系統安裝流程圖

1.1 產品規格

	項目	規格敘述
系統	匯排流資訊	PCI Rev. 2.2, 33MHz
	PCI 匯排流寬度	32-bit
	PCI 匯排流電壓	3.3V, 5V
	PCI 匯排流 IRQ 設定	Assigned by PCI controller
處理器 (DSP)	型號	TI 375MHz floating DSP
	記憶體配備 (程式及資料)	DDR2 SDRAM: 64Mx16bit Flash ROM: 16M-bit
板對板介面	連接器	1x SCSI-II 100P for AMP-204C
		1x Dual SCSI VHDCI 100P for AMP-208C
運動控制	最大支援軸數	4/8 軸 for AMP-204C / AMP-208C
	軌跡更新率	500us, 1ms, 2ms (可程式)
	位置 / 速度命令範圍	32 bit
	加速度 / 減速度範圍	32 bit
	編碼器輸入頻率	20 MHz @ 4x AB
	編碼器輸入模式	CW/CCW, 1x/2x/4x AB Phase
	編碼器輸入介面	±12 volts, TTL compatible
	運動控制相關 I/O	正 (Plus)/ 負 (Minus) 極限 訊號 (End limit)
		Zero-position for each axis
I/O 介面	驅動器相關 I/O	伺服使能 (Servo ON)
		到位訊號 (In-position) / 零速度檢出 (Zero-Speed)
		警報 (Alarm)
		20/24-CH 輸入 & 20/24-CH TTL 輸出 (光隔離設計於 DIN-825-GP4)
泛用數位輸出入	泛用 I/O	

	項目	規格敘述
運動控制功能	速度波形規劃 (Speed Profile Planning)	梯型曲線 (Trapezoidal Curve) 及 S 型曲線 (S-Curve)
	軌跡規劃 (Trajectory Planing)	吋動 (Jogging) 點對點運動
	線性補間：2-6 軸	在線位置 / 速度變更
		3 軸圓弧補間
		3 軸螺旋補間
歸零運動 (Home Return)	3 軸螺線補間	
	使用者定義 (可參考 zero-position, limit switch, EZ 等訊號)	

項目	規格敘述
點表 (Point table)	各軸獨立支援 50 點緩衝記憶區 (BUFs)
	支援點對點 / 直線 / 圓弧及螺旋補間
	支援 Dwell 功能
	支援暫停 / 恢復功能 (Pause/Resume)
	支援 DO 功能
運動狀態監測 (Motion Monitoring)	運動控制相關 I/O 監測
	運動狀態監測
同步運動 (Synchronous move)	4/8 軸對應 AMP-204C / AMP-208C
工業應用	主從軸控制
	高達 4/8 軸 (包含龍門控制 (Gantry Control))
	資料取樣
中斷	運動速度波形 / 運動狀態 / 運動控制相關 I/O
	系統錯誤診斷
	看門狗 (Watchdog timer)
運動狀態事件 / 錯誤報警 / 運動到位 / 緊急停止等	依手冊規劃
位置比對暨 觸發輸出	脈衝輸出介面
	差分輸出
	觸發通道
	2/4 對應 AMP-204C / AMP-208C
	脈衝邏輯
	可程式 (active-high or active-low)
	觸發輸出頻率
最小脈寬	線性比對觸發 : 1MHz
	FIFO 比對觸發 : 255K~1MHz
	100 ns (可程式)
位置比較模式	FIFO 與線性比對
FIFO 容量	255 點 (每通道獨立)

	項目	規格敘述
PWM 控制	最大通道數	2/4 CH 對應 AMP-204C / AMP-208C
	控制模式	● 固定頻率，變動占空比 ● 變動頻率，固定占空比 ● 變動頻率，變動占空比
	解析度	16 bit

環境條件

	項目
工作環境溫度	0~55°C
儲存環境溫度	-20~75°C
工作環境濕度	10~90%RH, avoid condensation
儲存環境濕度	10~90%RH, avoid condensation
雜訊阻抗	Noise voltage 1500V.P.P, Noise frequency 25~60Hz using noise simulator
環境條件	Minimal corrosive gas, dust
散熱條件	Selfcooling
產品功耗	+3.3V @ 0.8 A typical +5V @ 0.8 A typical +/-12V @ 0.5 A typical

1.2 軟體支援

1.2.1 軟體支援函式庫

AMP-204C / AMP-208C 支援 Windows XP/7 32/64 bit 操作系統，並提供完整的函式庫與動態連接檔 (DLL)，讓使用者可以輕易的完成其應用程式。

1.2.2 MotionCreatorPro 2

MotionCreatorPro 2 是專為凌華科技運動控制產品所開發的使用者介面，其可運作於一般 Windows 環境中。藉由 MotionCreatorPro 2，使用者可以輕易的完成如卡參數、軸參數的設定。而經由其中設置精靈 (Setup Wizard)，更能讓使用者在短短幾分鐘內完成硬體安裝、訊號配置及單軸操控 (Single-axis manipulation)，能有效縮短應用開發時程。MotionCreatorPro 2 提供一切單軸或補間運動操作頁面，能讓使用者同時檢視其整體機構設計及電控設計是否正確。

1.3 端子板

凌華科技提供 AMP-204C/208C 專用端子板 - 「DIN-825-GP4」，其可藉由特殊纜線輕易的與市售伺服驅動器連接，如三菱 (Mitsubishi) J3A 系列、安川 (Yaskawa) Sigma V 系列，亦可藉由單端開放纜線自行與其他廠牌伺服或步進驅動器連接。支援廠牌及專用纜線如下：

脈衝命令：

纜線	支援廠牌
HSL-4XMO-DM	Mitsubishi J2S 系列
4XMO-DM-J3	Mitsubishi J3A 系列
HSL-4XMO-DP	Panasonic A4, A5 系列
HSL-4XMO-DY	Yaskawa Sigma V 系列
4XMO-DA	Delta A2 系列
4XMO-OPEN	泛用型

表 1-1：脈衝型伺服驅動器專用纜線對照表

2 開始安裝

本章節將教導如何成功安裝 AMP-204C / AMP-208C，並完成硬體及軟體安裝，以及各式 I/O 接線。

- 確認包裝內項目
- 硬體安裝
- 軟體安裝
- 各式 I/O 接線

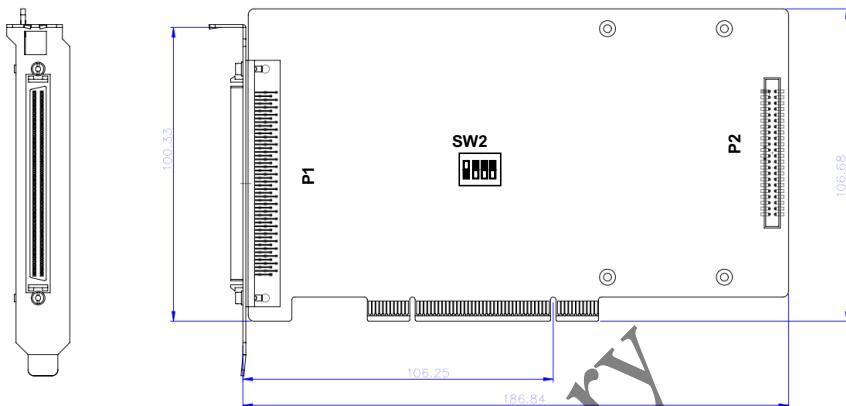
2.1 確認包裝內項目

除本手冊之外，產品包裝紙盒內應包含：

- AMP-204C 或 AMP-208C 一張
- IDE 44p – DSUB 37p 排線 (flat cable) 一條
- 產品保證卡

如果有任何以上項目遺失或是毀損，請速與您的經銷商聯絡。並保留完整產品包裝及內容物，以利新品更換或維修。

2.2 AMP-204C / AMP-208C 外觀輪廓圖



所有尺寸單位均為公釐。

註釋

圖 2-1 : AMP-204C 產品外觀輪廓圖

P1 : 為主要運動控制命令 (Motion Command)、位置回授 (Position feedback) 及相關伺服訊號回授 (Servo I/O feedback) 使用。(使用 SCSI 100-PINS 連接器 (connector))

P2 : 為 16 通道數位 TTL 輸入與 16 通道數位 TTL 輸出使用。(使用 DSUB 37-PINS 連接器)

SW2 : 卡片 ID(Card ID) 設定 (0-15)

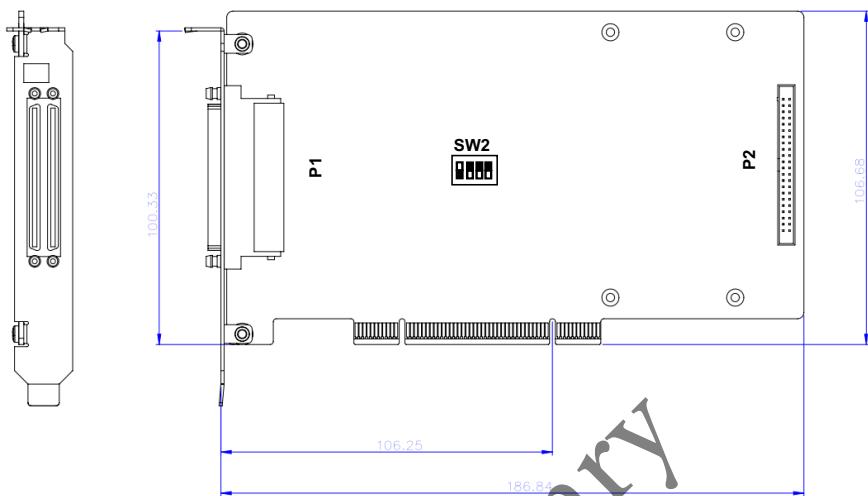


圖 2-2 : AMP-208C 產品外觀輪廓圖

P1 : 為主要運動控制命令、位置回授及相關伺服訊號回授使用。(使用 SCSI-VHDCI 200-PINS 連接器)

P2 : 為 16 通道數位 TTL 輸入與 16 通道數位 TTL 輸出使用。(使用 DSUB 37-PINS 連接器)

SW2 : 卡片 ID 設定 (0-15)

2.3 硬體安裝

2.3.1 硬體組態

AMP-204C/208C 採用 PCI Rev. 2.2 匯排流規格。系統 BIOS 可自動配置記憶體與 IRQ 通道。

專用端子板 DIN-825-GP4，提供光隔離電路 (Isolation circuitry) 與指示燈號，搭配專用纜線，可輕易連接各式知名伺服驅動器及步進驅動器。

2.3.2 硬體安裝程序

1. 閱讀此手冊並將訊號輸出入調整至適當模式
2. 關閉電腦電源及所有端子板上相關電源，並將 AMP-204C/208C 安裝至電腦內 32-bit PCI 插槽。(通常為白色插槽) (請確保做好靜電保護處理)
3. 利用 SCSI 100p 纜線，連接 AMP-204C/208C 及 DIN-825-GP4
4. 完成 DIN-825-GP4 板上運動控制相關極限開關，伺服專用訊號及泛用數位訊號配線
5. 完成與伺服或步進驅動器連接
6. 開啟系統電源，包含電腦電源、端子板相關電源、24Vdc 電源
7. 利用 MotionCreatorPro 2 確認所有 I/O 訊號及伺服操作的正確性



注意

為了防止人身觸電事故、保證電氣設備正常運行，請務必將電源端子之 Shielding 端與大地 (Earth) 連接。



注意

為安全起見，建議使用者初次使用此卡片時，務必將馬達驅動器與負載脫離，在未完成控制系統的安裝及調試前，不要將馬達驅動器與任何機械裝置連接。待調整卡片以及驅動器參數得以控制馬達，方可進行系統的連接，否則可能造成嚴重的後果。

2.3.3 問題排除

如果安裝後電腦無法正常開啟或運動控制操作時有異常情況發生。請依循下列建議進行問題排除。若已經依以下步驟進行但問題仍未排除，請洽詢您所購買的供應商，尋求技術服務。

故障現象	可能原因
安裝驅動程式之後在Windows 裝置管理員仍看不到這張卡	請關機重新確認卡已正確插入 PCI bus 中與從 Windows 控制台 "新增移除程式" 中確認驅動程式是否正確安裝完畢
電腦安裝驅動程式之後， MotionCreatorPro2 無法開啟	請確認系統中是否安裝 .NET framework v3.5 以上版本
接上馬達之後發生 MotionCreatorPro2 上無訊號是亮著，但是馬達也完全不會動	請確認端子板上有輸入外部 24Vdc 電源
使用 MotionCreatorPro2 時驅動器的控制燈號都正常但是驅動器發生警報	請確認軸參數設定警報邏輯 (ALM) 是否錯誤與 EMG 電路配置是否正確
輸出命令的數值與編碼器迴授的數值不相同	請確認迴授訊號 (CW/CCW,1xAB,2xAB,4xAB) 設定是否與驅動器設置相同
位置控制時，馬達時只能單向移動不能雙向往復移動	請確認輸出訊號的形式 (CW/CCW, OUT/DIR) 是否與馬達驅動器設定不同

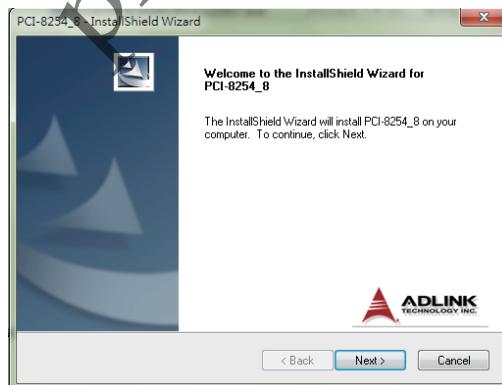
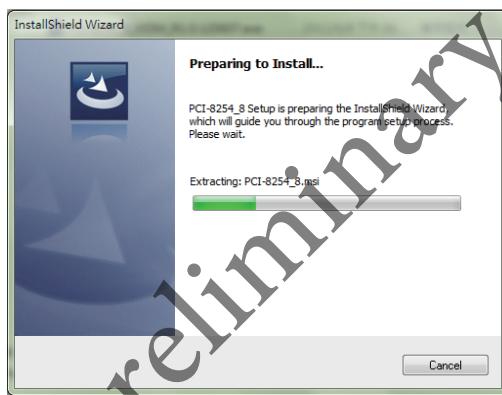
2.4 軟體安裝流程

安裝 Windows 驅動程式流程：

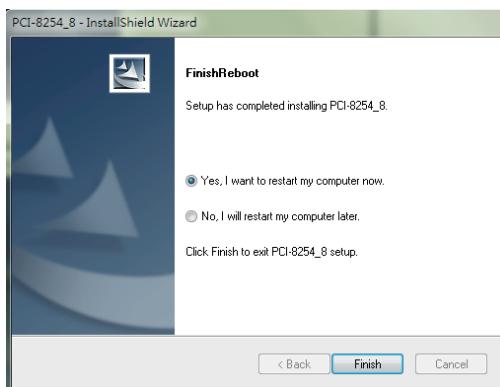
Step 1. 步驟 1：執行 AMP-204C / AMP-208C WDM 檔案，並自動執行安裝流程。



Step 2. 跟隨提示點擊 "next"，完成安裝。



Step 3. 完成安裝，重新開啟電腦。



Step 4. 確認 Windows 裝置管理員是否可以正確辨識 AMP-204C / AMP-208C 。



建議：請定期至凌華科技官方網頁下載最新的安裝軟體，以保持最佳的使用環境。

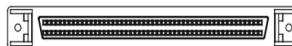
(<http://www.adlinktech.com/Motion-Control/index.php>)



2.5 主要連接器訊號定義

2.5.1 AMP-204C : P1 連接器

- P1



No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	DGND	--	Digital ground	51	IEMG	I	Emergency stop input
2	DGND	--	Digital ground	52	Rsv.	--	Reserved
3	Rsv.	--	Reserved	53	Rsv.	--	Reserved
4	Rsv.	--	Reserved	54	Rsv.	--	Reserved
5	Rsv.	--	Reserved	55	Rsv.	--	Reserved
6	Rsv.	--	Reserved	56	Rsv.	--	Reserved
7	Rsv.	--	Reserved	57	Rsv.	--	Reserved
8	Rsv.	--	Reserved	58	Rsv.	--	Reserved
9	Rsv.	--	Reserved	59	Rsv.	--	Reserved
10	Rsv.	--	Reserved	60	Rsv.	--	Reserved
11	EA5V	--	5V Power	61	DGND	--	Digital ground
12	EA5V	--	5V Power	62	DGND	--	Digital ground
13	OUT1+	○	Pulse output (+), (1)	63	OUT3+	○	Pulse output (+), (3)
14	OUT1-	○	Pulse output (-), (1)	64	OUT3-	○	Pulse output (-), (3)
15	DIR1+	○	Direction output (+), (1)	65	DIR3+	○	Direction output (+), (3)
16	DIR1-	○	Direction output (-), (1)	66	DIR3-	○	Direction output (-), (3)
17	OUT2+	○	Pulse output (+), (2)	67	OUT4+	○	Pulse output (+), (4)
18	OUT2-	○	Pulse output (-), (2)	68	OUT4-	○	Pulse output (-), (4)
19	DIR2+	○	Direction output (+), (2)	69	DIR4+	○	Direction output (+), (4)
20	DIR2-	○	Direction output (-), (2)	70	DIR4-	○	Direction output (-), (4)
21	TRG1+	○	Trigger output (+), (1)	71	TRG2+	○	Trigger output (+), (2)
22	TRG1-	○	Trigger output (-), (1)	72	TRG2-	○	Trigger output (-), (2)
23	EA1+		Encoder A-phase (+),(1)	73	EA3+		Encoder A-phase (+),(3)
24	EA1-		Encoder A-phase (-),(1)	74	EA3-		Encoder A-phase (-),(3)
25	EB1+		Encoder B-phase (+),(1)	75	EB3+		Encoder B-phase (+),(3)
26	EB1-		Encoder B-phase (-),(1)	76	EB3-		Encoder B-phase (-),(3)

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
27	EZ1+		Encoder Z-phase +(1)	77	EZ3+		Encoder Z-phase +(3)
28	EZ1-		Encoder Z-phase -(1)	78	EZ3-		Encoder Z-phase -(3)
29	EA2+		Encoder A-phase +(2)	79	EA4+		Encoder A-phase +(4)
30	EA2-		Encoder A-phase -(2)	80	EA4-		Encoder A-phase -(4)
31	EB2+		Encoder B-phase +(2)	81	EB4+		Encoder B-phase +(4)
32	EB2-		Encoder B-phase -(2)	82	EB4-		Encoder B-phase -(4)
33	EZ2+		Encoder Z-phase +(2)	83	EZ4+		Encoder Z-phase +(4)
34	EZ2-		Encoder Z-phase -(2)	84	EZ4-		Encoder Z-phase -(4)
35	ALM1		Servo alarm,(1)	85	ALM3		Servo alarm,(3)
36	ORG1		Home limit, (1)	86	ORG3		Home limit, (3)
37	SVON1	○	Servo-ON, (1)	87	SVON3	○	Servo-ON, (3)
38	PEL1		Positive limit, (1)	88	PEL3		Positive limit, (3)
39	INP1		In-Position (1)	89	INP3		In-Position (3)
40	MEL1		Negative limit, (1)	90	MEL3		Negative limit, (3)
41	ALM2		Servo alarm,(2)	91	ALM4		Servo alarm,(4)
42	ORG2		Home limit, (2)	92	ORG4		Home limit, (4)
43	SVON2	○	Servo-ON, (2)	93	SVON4	○	Servo-ON, (4)
44	PEL2		Positive limit, (2)	94	PEL4		Positive limit, (4)
45	INP2		In-Position (2)	95	INP4		In-Position (4)
46	MEL2		Negative limit, (2)	96	MEL4		Negative limit, (4)
47	EDO1	○	Digital Output, (1)	97	EDO3	○	Digital Output, (3)
48	EDI1		Digital Input, (1)	98	EDI3		Digital Input, (3)
49	EDO2	○	Digital Output, (2)	99	EDO4	○	Digital Output, (4)
50	EDI2		Digital Input, (2)	100	EDI4		Digital Input, (4)

2.5.2 AMP-208C : P1-A/B 連接器

- P1-A



No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	DGND	--	Digital ground	51	IEMG		Emergency stop input
2	DGND	--	Digital ground	52	Rsv.	--	Reserved
3	Rsv.	--	Reserved	53	Rsv.	--	Reserved
4	Rsv.	--	Reserved	54	Rsv.	--	Reserved
5	Rsv.	--	Reserved	55	Rsv.	--	Reserved
6	Rsv.	--	Reserved	56	Rsv.	--	Reserved
7	Rsv.	--	Reserved	57	Rsv.	--	Reserved
8	Rsv.	--	Reserved	58	Rsv.	--	Reserved
9	Rsv.	--	Reserved	59	Rsv.	--	Reserved
10	Rsv.	--	Reserved	60	Rsv.	--	Reserved
11	EA5V	--	5V power	61	DGND	--	Digital ground
12	EA5V	--	5V power	62	DGND	--	Digital ground
13	OUT1+	○	Pulse output (+), (1)	63	OUT3+	○	Pulse output (+), (3)
14	OUT1-	○	Pulse output (-), (1)	64	OUT3-	○	Pulse output (-), (3)
15	DIR1+	○	Direction output (+), (1)	65	DIR3+	○	Direction output (+), (3)
16	DIR1-	○	Direction output (-), (1)	66	DIR3-	○	Direction output (-), (3)
17	OUT2+	○	Pulse output (+), (2)	67	OUT4+	○	Pulse output (+), (4)
18	OUT2-	○	Pulse output (-), (2)	68	OUT4-	○	Pulse output (-), (4)
19	DIR2+	○	Direction output (+), (2)	69	DIR4+	○	Direction output (+), (4)
20	DIR2-	○	Direction output (-), (2)	70	DIR4-	○	Direction output (-), (4)
21	TRG1+	○	Trigger output (+), (1)	71	TRG2+	○	Trigger output (+), (2)
22	TRG1-	○	Trigger output (-), (1)	72	TRG2-	○	Trigger output (-), (2)
23	EA1+		Encoder A-phase (+),(1)	73	EA3+		Encoder A-phase +(3)
24	EA1-		Encoder A-phase (-),(1)	74	EA3-		Encoder A-phase -(3)
25	EB1+		Encoder B-phase (+),(1)	75	EB3+		Encoder B-phase +(3)
26	EB1-		Encoder B-phase (-),(1)	76	EB3-		Encoder B-phase -(3)
27	EZ1+		Encoder Z-phase (+),(1)	77	EZ3+		Encoder Z-phase +(3)
28	EZ1-		Encoder Z-phase (-),(1)	78	EZ3-		Encoder Z-phase -(3)

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
29	EA2+		Encoder A-phase (+),(2)	79	EA4+		Encoder A-phase (+),(4)
30	EA2-		Encoder A-phase (-),(2)	80	EA4-		Encoder A-phase (-),(4)
31	EB2+		Encoder B-phase (+),(2)	81	EB4+		Encoder B-phase (+),(4)
32	EB2-		Encoder B-phase (-),(2)	82	EB4-		Encoder B-phase (-),(4)
33	EZ2+		Encoder Z-phase (+),(2)	83	EZ4+		Encoder Z-phase (+),(4)
34	EZ2-		Encoder Z-phase (-),(2)	84	EZ4-		Encoder Z-phase (-),(4)
35	ALM1		Servo alarm,(1)	85	ALM3		Servo alarm,(3)
36	ORG1		Home limit, (1)	86	ORG3		Home limit, (3)
37	SVON1	○	Servo-ON, (1)	87	SVON3	○	Servo-ON, (3)
38	PEL1		Positive limit, (1)	88	PEL3		Positive limit, (3)
39	INP1		In-Position (1)	89	INP3		In-Position (3)
40	MEL1		Negative limit, (1)	90	MEL3		Negative limit, (3)
41	ALM2		Servo alarm,(2)	91	ALM4		Servo alarm,(4)
42	ORG2		Home limit, (2)	92	ORG4		Home limit, (4)
43	SVON2	○	Servo-ON, (2)	93	SVON4	○	Servo-ON, (4)
44	PEL2		Positive limit, (2)	94	PEL4		Positive limit, (4)
45	INP2		In-Position (2)	95	INP4		In-Position (4)
46	MEL2		Negative limit, (2)	96	MEL4		Negative limit, (4)
47	EDO1	○	Digital Output, (1)	97	EDO3	○	Digital Output, (3)
48	EDI1		Digital Input, (1)	98	EDI3		Digital Input, (3)
49	EDO2	○	Digital Output, (2)	99	EDO4	○	Digital Output, (4)
50	EDI2		Digital Input, (2)	100	EDI4		Digital Input, (4)

• **P1-B**

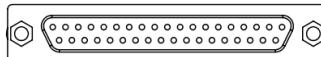
No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	Rsv.	--	Reserved	51	Rsv.	--	Reserved
2	Rsv.	--	Reserved	52	Rsv.	--	Reserved
3	Rsv.	--	Reserved	53	Rsv.	--	Reserved
4	Rsv.	--	Reserved	54	Rsv.	--	Reserved
5	Rsv.	--	Reserved	55	Rsv.	--	Reserved
6	Rsv.	--	Reserved	56	Rsv.	--	Reserved

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
7	Rsv.	--	Reserved	57	Rsv.	--	Reserved
8	Rsv.	--	Reserved	58	Rsv.	--	Reserved
9	Rsv.	--	Reserved	59	Rsv.	--	Reserved
10	Rsv.	--	Reserved	60	Rsv.	--	Reserved
11	EA5V	--	5V power	61	DGND	--	Digital ground
12	EA5V	--	5V power	62	DGND	--	Digital ground
13	OUT5+	O	Pulse output (+), (5)	63	OUT7+	O	Pulse output (+), (7)
14	OUT5-	O	Pulse output (-), (5)	64	OUT7-	O	Pulse output (-), (7)
15	DIR5+	O	Direction output (+), (5)	65	DIR7+	O	Direction output (+), (7)
16	DIR5-	O	Direction output (-), (5)	66	DIR7-	O	Direction output (-), (7)
17	OUT6+	O	Pulse output (+), (6)	67	OUT8+	O	Pulse output (+), (8)
18	OUT6-	O	Pulse output (-), (6)	68	OUT8-	O	Pulse output (-), (8)
19	DIR6+	O	Direction output (+), (6)	69	DIR8+	O	Direction output (+), (8)
20	DIR6-	O	Direction output (-), (6)	70	DIR8-	O	Direction output (-), (8)
21	TRG3+	O	Trigger output (+), (3)	71	TRG4+	O	Trigger output (+), (4)
22	TRG3-	O	Trigger output (-), (3)	72	TRG4-	O	Trigger output (-), (4)
23	EA5+	I	Encoder A-phase (+),(5)	73	EA7+	I	Encoder A-phase (+),(7)
24	EA5-	I	Encoder A-phase (-),(5)	74	EA7-	I	Encoder A-phase (-),(7)
25	EB5+	I	Encoder B-phase (+),(5)	75	EB7+	I	Encoder B-phase (+),(7)
26	EB5-	I	Encoder B-phase (-),(5)	76	EB7-	I	Encoder B-phase (-),(7)
27	EZ5+	I	Encoder Z-phase (+),(5)	77	EZ7+	I	Encoder Z-phase (+),(7)
28	EZ5-	I	Encoder Z-phase (-),(5)	78	EZ7-	I	Encoder Z-phase (-),(7)
29	EA6+	I	Encoder A-phase (+),(6)	79	EA8+	I	Encoder A-phase (+),(8)
30	EA6-	I	Encoder A-phase (-),(6)	80	EA8-	I	Encoder A-phase (-),(8)
31	EB6+	I	Encoder B-phase (+),(6)	81	EB8+	I	Encoder B-phase (+),(8)
32	EB6-	I	Encoder B-phase (-),(6)	82	EB8-	I	Encoder B-phase (-),(8)
33	EZ6+	I	Encoder Z-phase (+),(6)	83	EZ8+	I	Encoder Z-phase (+),(8)
34	EZ6-	I	Encoder Z-phase (-),(6)	84	EZ8-	I	Encoder Z-phase (-),(8)
35	ALM5	I	Servo alarm,(5)	85	ALM7	I	Servo alarm,(7)
36	ORG5	I	Home limit, (5)	86	ORG7	I	Home limit, (7)
37	SVON5	O	Servo-ON, (5)	87	SVON7	O	Servo-ON, (7)

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
38	PEL5	I	Positive limit, (5)	88	PEL7	I	Positive limit, (7)
39	INP5	I	In-Position (5)	89	INP7	I	In-Position (7)
40	MEL5	I	Negative limit, (5)	90	MEL7	I	Negative limit, (7)
41	ALM6	I	Servo alarm,(6)	91	ALM8	I	Servo alarm,(8)
42	ORG6	I	Home limit, (6)	92	ORG8	I	Home limit, (8)
43	SVON6	O	Servo-ON, (6)	93	SVON8	O	Servo-ON, (8)
44	PEL6	I	Positive limit, (6)	94	PEL8	I	Positive limit, (8)
45	INP6	I	In-Position (6)	95	INP8	I	In-Position (8)
46	MEL6	I	Negative limit, (6)	96	MEL8	I	Negative limit, (8)
47	EDO5	O	Digital Output, (5)	97	EDO7	O	Digital Output, (7)
48	EDI5	I	Digital Input, (5)	98	EDI7	I	Digital Input, (7)
49	EDO6	O	Digital Output, (6)	99	EDO8	O	Digital Output, (8)
50	EDI6	I	Digital Input, (6)	100	EDI8	I	Digital Input, (8)

2.5.3 AMP-204C/208C : P2 連接器

- P2



No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	Rsv.	--	Reserved	20	VDD	I	+5V power supply input
2	TDI1	I	TTL input, (1)	21	TDO1	O	TTL output, (1)
3	TDI2	I	TTL input, (2)	22	TDO2	O	TTL output, (2)
4	TDI3	I	TTL input, (3)	23	TDO3	O	TTL output, (3)
5	TDI4	I	TTL input, (4)	24	TDO4	O	TTL output, (4)
6	TDI5	I	TTL input, (5)	25	TDO5	O	TTL output, (5)
7	TDI6	I	TTL input, (6)	26	TDO6	O	TTL output, (6)
8	TDI7	I	TTL input, (7)	27	TDO7	O	TTL output, (7)
9	TDI8	I	TTL input, (8)	28	TDO8	O	TTL output, (8)
10	TDI9	I	TTL input, (9)	29	TDO9	O	TTL output, (9)
11	TDI10	I	TTL input, (10)	30	TDO10	O	TTL output, (10)
12	TDI11	I	TTL input, (11)	31	TDO11	O	TTL output, (11)
13	TDI12	I	TTL input, (12)	32	TDO12	O	TTL output, (12)

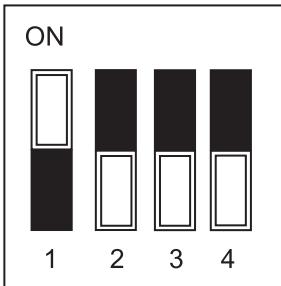
No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
14	TDI13		TTL input, (13)	33	TDO13	○	TTL output, (13)
15	TDI14		TTL input, (14)	34	TDO14	○	TTL output, (14)
16	TDI15		TTL input, (15)	35	TDO15	○	TTL output, (15)
17	TDI16		TTL input, (16)	36	TDO16	○	TTL output, (16)
18	DGND	-	Digital ground	37	DGND	-	Digital ground
19	VDD		+5V power supply input	--	--	--	--

preliminary

2.6 DIP Switch

2.6.1 SW2 : 卡片編號開關 (Card ID Switch)

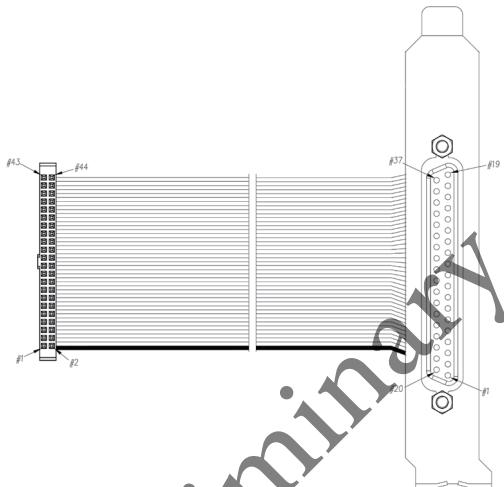
此開關被使用於調整各卡片編號，方便使用者於應用程式中能輕易辨識該卡的卡片編號。例如當使用者將卡編號調整至 "0-0-0-1" (OFF-OFF-OFF-ON)，則意謂該卡編號為 "1"，調整編號表如下：



Card ID	Switch Setting (ON=1)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111 (預設)

2.7 IDE 44p – DSUB 37p 排線

本產品包裝內附贈一條 IDE 44 pin 至 DSUB 37 pin 的 IDE 排線一條。其功能為 AMP-204C / AMP-208C P2 延伸 16 通道數位輸入及 16 通道數位輸出使用。



2.8 專用端子板 - DIN-825-GP4

「DIN-825-GP4」為 PCI-8254 / PCI-8258 以及 AMP-204C / AMP-208C 專用端子板，其可藉由特殊纜線輕易的與市售伺服驅動器連接，如三菱 (Mitsubishi) J3A 系列、安川 (Yaskawa) Sigma V 系列，亦可藉由單端開放纜線自行與其他廠牌伺服或步進驅動器連接。



DIN-825-GP4 支援 PCI-8254 / PCI-8258 以及 AMP-204C / AMP-208C，請不要與其他凌華科技運動控制產品連接使用，以免燒毀該端子板。

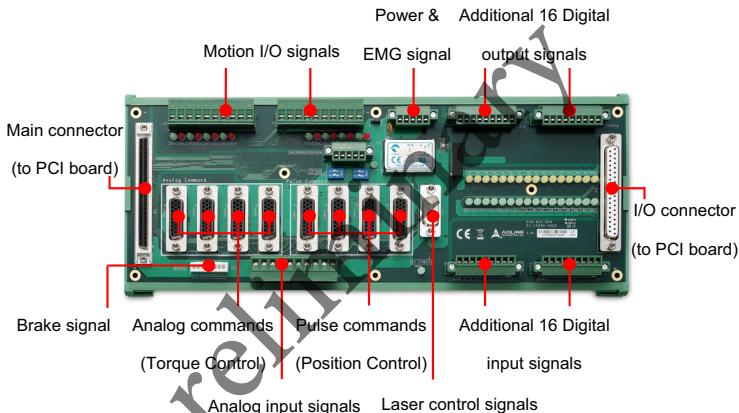


圖 2-3 : DIN-825-GP4 外觀圖

2.8.1 連接器定義

1. P1：為一組 SCSI 100-PINS 連接器，主要運動控制訊號使用。
2. CMA1–4：為四組 26-PINS 連接器，主要為連接伺服驅動器進行 S/T 模式控制，輸出類比控制命令使用。
3. CMP1–4：為四組 26-PINS 連接器，主要為連接伺服驅動器進行 P 模式控制或步進驅動器，輸出脈衝控制命令使用。可藉由專用纜線連接至 Mitsubishi J3A 系列，Yaskawa Sigma II, III & V 系列，Panasonic MINAS A4&A5
4. J1–J3：為 3 組 10-PINS 螺絲鎖附型 (Screwed 系列，Delta A2 系列，或藉由單端開放之纜線，連接至其他廠牌伺服驅動器及步進驅動器 type) 連接器，可連接所有類比輸入訊號、比較觸發訊號、各軸機械正負極限開關及原點‘訊號。
5. J4：為一組 8-PINS 連接器，主要為連接煞車訊號 (Brake Signal) 時使用。
6. J5：為一組 5-PINS 連接器，主要為端子板主電源及緊急停止訊號使用。

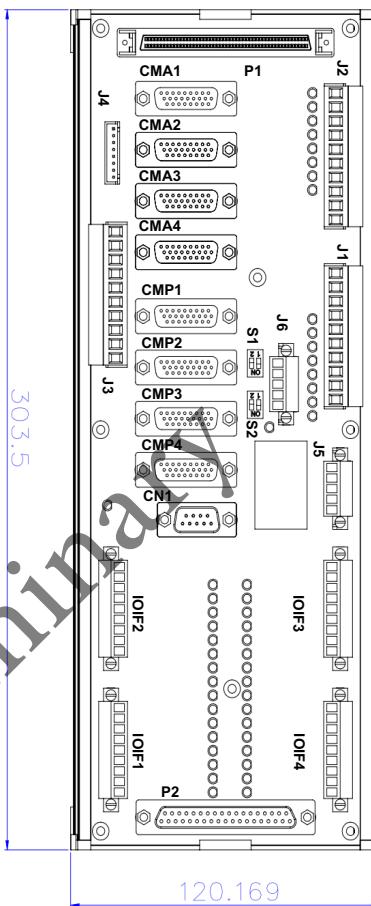
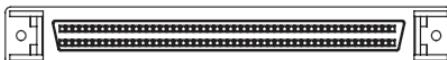


圖 2-4 : DIN-825-GP4 機構圖

7. J6：為一組 5-PINS 連接器，主要為 4 組數位帶隔離輸出 (Isolated digital output) 通道
8. P2：為一組 DSUB 37-PINS 連接器，主要為連接控制器上 16 通道數位輸出訊號及 16 通道數位輸入訊號 (TTL)。
9. IOIF1-IOIF4：為四組 9-PINS 連接器，主要為泛用 16 通道數位輸出訊號及 16 通道數位輸入訊號使用。
10. CN1：為一組 9-PINS 連接器，主要為 Laser Control 使用。

preliminary

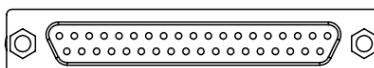
2.8.2 P1 連接器：可連接至 PCI-8254 / PCI-8258 / AMP-204C / AMP-208C



• P1 :

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	DGND	--	Digital ground	51	IEMG		Emergency stop input
2	DGND	--	Digital ground	52	Rsv.	--	Reserved
3	Rsv.	--	Reserved	53	Rsv.	--	Reserved
4	Rsv.	--	Reserved	54	Rsv.	--	Reserved
5	Rsv.	--	Reserved	55	Rsv.	--	Reserved
6	Rsv.	--	Reserved	56	Rsv.	--	Reserved
7	Rsv.	--	Reserved	57	Rsv.	--	Reserved
8	Rsv.	--	Reserved	58	Rsv.	--	Reserved
9	Rsv.	--	Reserved	59	Rsv.	--	Reserved
10	Rsv.	--	Reserved	60	Rsv.	--	Reserved
11	EA5V	--	5V power	61	DGND	--	Digital ground
12	EA5V	--	5V power	62	DGND	--	Digital ground
13	OUT1+	○	Pulse output (+), (1)	63	OUT3+	○	Pulse output (+), (3)
14	OUT1-	○	Pulse output (-), (1)	64	OUT3-	○	Pulse output (-), (3)
15	DIR1+	○	Direction output (+), (1)	65	DIR3+	○	Direction output (+), (3)
16	DIR1-	○	Direction output (-), (1)	66	DIR3-	○	Direction output (-), (3)
17	OUT2+	○	Pulse output (+), (2)	67	OUT4+	○	Pulse output (+), (4)
18	OUT2-	○	Pulse output (-), (2)	68	OUT4-	○	Pulse output (-), (4)
19	DIR2+	○	Direction output (+), (2)	69	DIR4+	○	Direction output (+), (4)
20	DIR2-	○	Direction output (-), (2)	70	DIR4-	○	Direction output (-), (4)
21	TRG1+	○	Trigger output (+), (1)	71	TRG2+	○	Trigger output (+), (2)
22	TRG1-	○	Trigger output (-), (1)	72	TRG2-	○	Trigger output (-), (2)
23	EA1+		Encoder A-phase (+),(1)	73	EA3+		Encoder A-phase (+),(3)
24	EA1-		Encoder A-phase (-),(1)	74	EA3-		Encoder A-phase (-),(3)
25	EB1+		Encoder B-phase (+),(1)	75	EB3+		Encoder B-phase (+),(3)
26	EB1-		Encoder B-phase (-),(1)	76	EB3-		Encoder B-phase (-),(3)

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
27	EZ1+		Encoder Z-phase (+),(1)	77	EZ3+		Encoder Z-phase (+),(3)
28	EZ1-		Encoder Z-phase (-),(1)	78	EZ3-		Encoder Z-phase (-),(3)
29	EA2+		Encoder A-phase (+),(2)	79	EA4+		Encoder A-phase (+),(4)
30	EA2-		Encoder A-phase (-),(2)	80	EA4-		Encoder A-phase (-),(4)
31	EB2+		Encoder B-phase (+),(2)	81	EB4+		Encoder B-phase (+),(4)
32	EB2-		Encoder B-phase (-),(2)	82	EB4-		Encoder B-phase (-),(4)
33	EZ2+		Encoder Z-phase (+),(2)	83	EZ4+		Encoder Z-phase (+),(4)
34	EZ2-		Encoder Z-phase (-),(2)	84	EZ4-		Encoder Z-phase (-),(4)
35	ALM1		Servo alarm,(1)	85	ALM3		Servo alarm,(3)
36	ORG1		Home limit, (1)	86	ORG3		Home limit, (3)
37	SVON1	○	Servo-ON, (1)	87	SVON3	○	Servo-ON, (3)
38	PEL1		Positive limit, (1)	88	PEL3		Positive limit, (3)
39	INP1		In-Position (1)	89	INP3		In-Position (3)
40	MEL1		Negative limit, (1)	90	MEL3		Negative limit, (3)
41	ALM2		Servo alarm,(2)	91	ALM4		Servo alarm,(4)
42	ORG2		Home limit, (2)	92	ORG4		Home limit, (4)
43	SVON2	○	Servo-ON, (2)	93	SVON4	○	Servo-ON, (4)
44	PEL2		Positive limit, (2)	94	PEL4		Positive limit, (4)
45	INP2		In-Position (2)	95	INP4		In-Position (4)
46	MEL2		Negative limit, (2)	96	MEL4		Negative limit, (4)
47	EDO1	○	Digital Output, (1)	97	EDO3	○	Digital Output, (3)
48	EDI1		Digital Input, (1)	98	EDI3		Digital Input, (3)
49	EDO2	○	Digital Output, (2)	99	EDO4	○	Digital Output, (4)
50	EDI2		Digital Input, (2)	100	EDI4		Digital Input, (4)



• P2 :

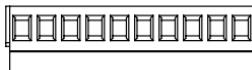
No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	Rsv.	--	Reserved	20	VDD	○	+5V power supply Output
2	TDI1		TTL input, (1)	21	TDO1	○	TTL output, (1)

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
3	TDI2		TTL input, (2)	22	TDO2	○	TTL output, (2)
4	TDI3		TTL input, (3)	23	TDO3	○	TTL output, (3)
5	TDI4		TTL input, (4)	24	TDO4	○	TTL output, (4)
6	TDI5		TTL input, (5)	25	TDO5	○	TTL output, (5)
7	TDI6		TTL input, (6)	26	TDO6	○	TTL output, (6)
8	TDI7		TTL input, (7)	27	TDO7	○	TTL output, (7)
9	TDI8		TTL input, (8)	28	TDO8	○	TTL output, (8)
10	TDI9		TTL input, (9)	29	TDO9	○	TTL output, (9)
11	TDI10		TTL input, (10)	30	TDO10	○	TTL output, (10)
12	TDI11		TTL input, (11)	31	TDO11	○	TTL output, (11)
13	TDI12		TTL input, (12)	32	TDO12	○	TTL output, (12)
14	TDI13		TTL input, (13)	33	TDO13	○	TTL output, (13)
15	TDI14		TTL input, (14)	34	TDO14	○	TTL output, (14)
16	TDI15		TTL input, (15)	35	TDO15	○	TTL output, (15)
17	TDI16		TTL input, (16)	36	TDO16	○	TTL output, (16)
18	EGND	-	External power ground	37	EGND	-	External power ground
19	VDD		+5V power supply input	--	--	--	--



• J1 :

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	DICOM	--	Digital input common	6	EDI4		Isolated digital input, (4)
2	EDI3		Isolated digital input, (3)	7	PEL4		Positive limit, (4)
3	PEL3		Positive limit, (3)	8	ORG4		Origin Signal, (4)
4	ORG3		Origin Signal, (3)	9	MEL4		Negative limit, (4)
5	MEL3		Negative limit, (3)	10	DOCOM	--	Digital output common



- **J2 :**

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	DICOM	--	Digital input common	6	EDI2		Isolated digital input, (2)
2	EDI1		Isolated digital input, (1)	7	PEL2		Positive limit, (2)
3	PEL1		Positive limit, (1)	8	ORG2		Origin Signal, (2)
4	ORG1		Origin Signal, (1)	9	MEL2		Negative limit, (2)
5	MEL1		Negative limit, (1)	10	DOCOM	--	Digital output common



註釋

1. DICOM 建議連接至外部電源供應 (一般為 24VDC)
2. DOCOM 建議連接至外部電源的地 (GND)



- **J3 :**

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	DGND	--	Isolated digital ground	6	AGND	--	Analog ground
2	TRG2-	○	Trigger output (-), (2)	7	AI4		Analog input, (4)
3	TRG2+	○	Trigger output (+), (2)	8	AI3		Analog input, (3)
4	TRG1-	○	Trigger output (-), (1)	9	AI2		Analog input, (2)
5	TRG1+	○	Trigger output (+), (1)	10	AI1		Analog input, (1)



- **J4 : Brake Connector**

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	BRAKE 1+	--	Brake signal (+), (1)	6	BRAKE 3+		Brake signal (+), (3)
2	BRAKE 1-		Brake signal (-), (1)	7	BRAKE 3-		Brake signal (-), (3)
3	BRAKE 2+		Brake signal (+), (2)	8	BRAKE 4+		Brake signal (+), (4)
4	BRAKE 2-		Brake signal (-), (2)	9	BRAKE 4-		Brake signal (-), (4)



• J5

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	I24V	--	Ext. power supply, +24V	4	DOCOM	--	Digital output common
2	IGND	--	Ext. power ground	5	EEMG	I	Ext. Emergency signal
3	DICOM	--	Digital input common	6	--	--	--



註釋

1. DICOM 建議連接至外部電源供應 (一般為 24VDC)

2. DOCOM 建議連接至外部電源的地 (GND)

• J6



No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	EDO1	○	Digital output, (1)	4	EDO4	○	Digital output, (4)
2	EDO2	○	Digital output, (2)	5	DOCOM	○	Digital output common
3	EDO3	○	Digital output, (3)	6	--	○	--

• IOIF1 :



No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	DI1	I	Additional isolated digital input, (1)	6	DI6	I	Additional isolated digital input, (6)
2	DI2	I	Additional isolated digital input, (2)	7	DI7	I	Additional isolated digital input, (7)
3	DI3	I	Additional isolated digital input, (3)	8	DI8	I	Additional isolated digital input, (8)
4	DI4	I	Additional isolated digital input, (4)	9	DICOM	--	Digital input common
5	DI5	I	Additional isolated digital input, (5)	--	--	--	--

• **IOIF2 :**

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	DI9	I	Additional isolated digital input, (9)	6	DI14	I	Additional isolated digital input, (14)
2	DI10	I	Additional isolated digital input, (10)	7	DI15	I	Additional isolated digital input, (15)
3	DI11	I	Additional isolated digital input, (11)	8	DI16	I	Additional isolated digital input, (16)
4	DI12	I	Additional isolated digital input, (12)	9	DICOM	--	Digital input common
5	DI13	I	Additional isolated digital input, (13)	--	--	--	--

• **IOIF3 :**

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	※DO1	O	Additional isolated digital output, (1)	6	DO6	O	Additional isolated digital output, (6)
2	※DO2	O	Additional isolated digital output, (2)	7	DO7	O	Additional isolated digital output, (7)
3	※DO3	O	Additional isolated digital output, (3)	8	DO8	O	Additional isolated digital output, (8)
4	※DO4	O	Additional isolated digital output, (4)	9	DOCOM	--	Digital output common
5	※DO5	O	Additional isolated digital output, (5)	--	--	--	--



註釋

※ 表示該數位輸出電流可達 250mA

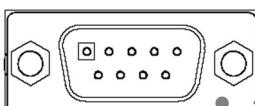
- **IOIF4 :**

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	DO9	O	Additional isolated digital output, (9)	6	DO14	O	Additional isolated digital output, (14)
2	DO10	O	Additional isolated digital output, (10)	7	DO15	O	Additional isolated digital output, (15)
3	DO11	O	Additional isolated digital output, (11)	8	DO16	O	Additional isolated digital output, (16)
4	DO12	O	Additional isolated digital output, (12)	9	DOCOM	--	Digital output common
5	DO13	O	Additional isolated digital output, (13)	--	--	--	--



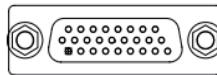
註釋

1. DICOM 建議連接至外部電源供應 (一般為 24VDC)
2. DOCOM 建議連接至外部電源的地 (GND)



- **CN1:**

No.	Name	I/O	Function of Axis	No.	Name	I/O	Function of Axis
1	EDO4+	O	Digital output (+), (4)	6	EDO4-	O	Digital output (-), (4)
2	TRG1+	O	Trigger output (+), (1)	7	TRG1-	O	Trigger output (-), (1)
3	TRG2+	O	Trigger output (+), (2)	8	TRG2-	O	Trigger output (-), (2)
4	Rsv.	--	Reserved	9	DGND	--	Digital ground
5	Rsv.	--	Reserved				



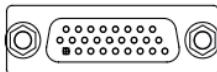
• CMA1-CMA4 (僅適用於 PCI-8254/8258) :

No.	Name	I/O	Function	No.	Name	I/O	Function	No.	Name	I/O	Function
1	SYON	O	Servo On signal	10	ALM_RST / DO	O	Reset driver signal / Digital output signal	19	EMG	I	Emergency signal
2	ZSP	I	Zero speed signal	11	ALM	I	Servo alarm signal	20	GIND	--	Ext. power ground
3	Rev.		Reserved	12	12V	--	Ext. power supply, +24V	21	GIND	--	Ext. power ground
4	Rev.	--	Reserved	13	IGND	--	Ext. power ground	22	GIND	--	Ext. power ground
5	AOUT-	O	Analog command output(-)	14	BRAKE-	O	Brake signal(-)	23	Rev.	--	Reserved
6	AOUT+	O	Analog command output(+)	15	AGND	--	Analog ground	24	Rev.	--	Reserved
7	EA-	I	Encoder A-phase(-)	16	EB	I	Encoder B-phase(-)	25	E2-	I	Encoder Z-phase(-)
8	EA+	I	Encoder A-phase(+)	17	EB+	I	Encoder B-phase(+)	26	E2+	I	Encoder Z-phase(+)
9	BRAKE+	O	Brake signal(+)	18	AGND	--	Analog ground				



ALM_RST / DO : 此訊號可以藉由 S1 或 S2 開關被選擇為泛用數位輸出訊號 (EDO) 或警報清除功能 (ALM_RST) 使用

註釋



- **CMP1~CMP4 :**

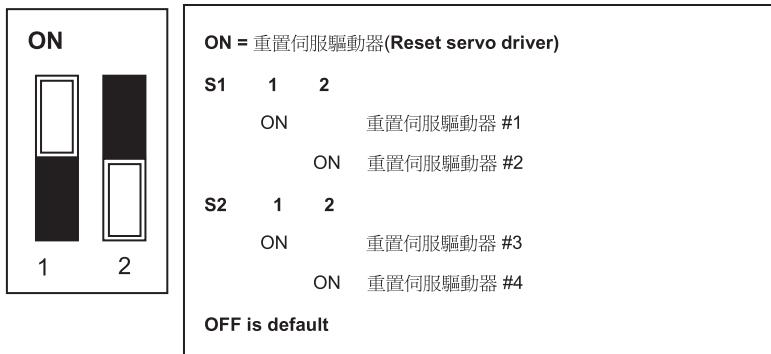
No.	Name	I/O	Function	No.	Name	I/O	Function	No.	Name	I/O	Function
1	SVON	O	Servo On signal	10	ALM_RST / DO	O	Reset driver signal / Digital output signal	19	EMG	I	Emergency signal
2	INP	I	In-position signal	11	ALM	I	Servo alarm signal	20	IGND	--	Ext. power ground
3	ERC	O	Brk ctr. ctr. signal	12	12AV	--	Ext. power supply, +24V	21	IGND	--	Ext. power ground
4	RDY	I	Servo ready signal	13	IGND	--	Ext. power ground	22	IGND	--	Ext. power ground
5	OUT	O	Pulse signal(+)	14	BRKE-	O	Brake signal(-)	23	DIR-	O	Dir. Signal(-)
6	OUT+	O	Pulse signal(-)	15	IGND	--	Ext. power ground	24	DIR+	O	Dir. Signal(+)
7	EA-	I	Encoder A-phase(-)	16	EB+	I	Encoder B-phase(+)	25	EZ-	I	Encoder Z-phase(-)
8	EA+	I	Encoder A-phase(+)	17	EB+	I	Encoder B-phase(-)	26	EZ+	I	Encoder Z-phase(+)
9	BRKE+	O	Brake signal(+)	18	IGND	--	Ext. power ground				



ALM_RST / DO : 此訊號可以藉由 S1 或 S2 開關被選擇為泛用數位輸出訊號 (EDO) 或警報清除功能 (ALM_RST) 使用

註釋

2.8.3 S1, S2 : EDO/ALM_RST 選擇開關



DIN-825-GP4 搭載 4 組伺服驅動器 (servo drive) 重置 (reset) 訊號。經由設定開關 S1 及 S2，可設定 CMA1~CMA4 PIN 10 及 CMP1~CMP4 PIN 10 為重置伺服驅動器使用或是 J6 連結器 DO.1~DO.4 使用。

preliminary

3 訊號連接介面

AMP-204C / AMP-208C 必須搭配其專用端子板 DIN-825-GP4，與伺服或步進馬達驅動器連接。所有機械相關 I/O、伺服相關 I/O 之光隔離電路被設置於 DIN-825-GP4，以防止任何不當訊號連接對主要控制器 AMP-204C / AMP-208C 造成毀損。並且可有效減低使用者進行客服維修時，置換控制器相關產品之難度，以及減少其所需時間。以下章節將對各式不同之機械相關 I/O (Mechanical I/O)、伺服相關 I/O (Servo I/O) 訊號連接時所需注意之事項詳加敘述。本章包含有：

- 第 3.1 節：** 脈衝控制命令訊號
- 第 3.2 節：** 編碼器輸入訊號
- 第 3.3 節：** 緊急停止訊號
- 第 3.4 節：** 機械極限開關訊號
- 第 3.5 節：** 原點開關訊號
- 第 3.6 節：** 到位 / 零速度訊號
- 第 3.7 節：** 伺服警報訊號
- 第 3.8 節：** 伺服激磁訊號
- 第 3.9 節：** 比較觸發訊號
- 第 3.10 節：** 泛用數位輸出入訊號

3.1 脈衝控制命令訊號 (Pulse Command):

AMP-204C / AMP-208C 亦提供 4/8 脈衝控制命令通道，每組脈衝控制命令支援高達 6.5MHz 輸出頻率。

一般而言，伺服驅動器可被設定為 P (位置) 模式，意謂將使用 AMP-204C / AMP-208C 的脈衝命令控制，做開迴路控制(Open-loop Control)。

除了伺服驅動器外，步進馬達驅動器 (Stepper drive) 亦使用脈衝命令介面為主要控制輸入命令。以下為差分脈衝訊號至 DIN-825-GP4 上之脈衝命令輸出之對應腳位：

CMP _x Pin No (x=1~4)	Signal Name	Description (n=1~8)	Axis #
6	OUT+	Pulse signal, (+) (n)	1~8
5	OUT-	Pulse signal, (-) (n)	1~8
24	DIR+	Dir. Signal, (+) (n)	1~8
23	DIR-	Dir. Signal, (-) (n)	1~8



註釋

AMP-208C 需搭配 2 組 DIN-825-GP4 以完成八軸運動控制功能

#1 負責軸 1~ 軸 4 ; #2 負責軸 5~ 軸 8

一般無論是伺服馬達驅動器或步進馬達驅動器可能會使用下列兩種輸入介面，

1. Line Driver 輸入介面，可提供較佳的抗雜訊 (Noise-resistant) 能力及較長的配線長度。
- 訊號連接範例圖：

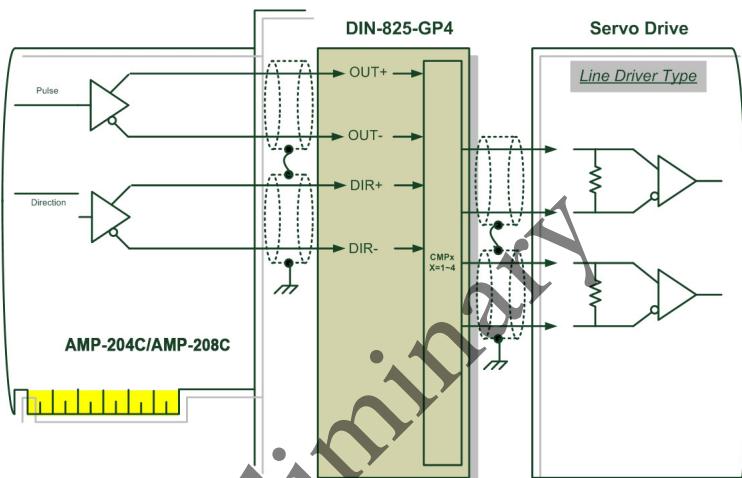


圖 3-1：脈衝控制命令訊號 Line Driver 型連接範例圖

2. Open-Collector 輸入介面，可藉由調整提升電阻 (Pull-up resistance)，提高訊號通過的電流量，但僅適用於較短的配線設置。

- 訊號連接範例圖：

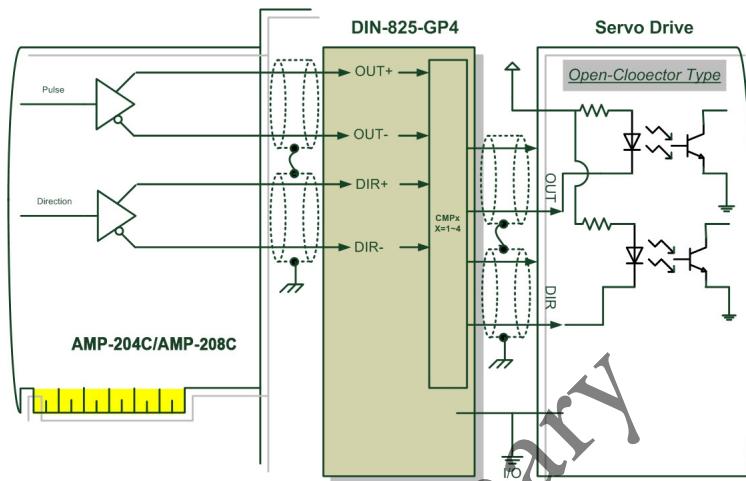


圖 3-2：脈衝控制命令訊號 Open-Collector 型連接範例圖



為了防止不當配線造成控制器上 Line Driver 元件毀損，建議連接控制器之 OUT-, DIR- 腳位至馬達驅動器之 OUT, DIR 腳位

注意



控制器使用 Line Driver 元件 -26LS31，其最大 Sink Current 規為 20mA，請不要超出此規格使用，以免毀損該元件

注意

3.2 編碼器輸入訊號 (Encoder Input, EA & EB & EZ):

AMP-204C / AMP-208C 分別提供 4/8 編碼器輸入通道，其可允許單端輸入頻率高達 5MHz，其中各通道包含有 EA、EB 及 EZ 訊號。而每組 EA、EB、EZ 訊號包含一對差分訊號，如 EA 訊號包含 EA+ 及 EA-。編碼器使用原理請參考第 4.1.1.4 節。以下為 DIN-825-GP4 上之編碼器輸入對應腳位：

CMAx / CMPx Pin No (x=1~4)	Signal Name	Description (n=1~8)	Axis #
8	EA+	Encoder A-phase (+),(n)	1~8
7	EA-	Encoder A-phase (-),(n)	1~8
17	EB+	Encoder B-phase (+),(n)	1~8
16	EB-	Encoder B-phase (-),(n)	1~8
26	EZ+	Encoder Z-phase (+),(n)	1~8
25	EZ-	Encoder Z-phase (-),(n)	1~8



AMP-208C 需搭配 2 組 DIN-825-GP4 以完成八軸運動控制

功能

#1 負責軸 1~ 軸 4 ; #2 負責軸 5~ 軸 8



注意

控制器使用 Line Receiver 元件 -26LS32，其最大 Sink Current 規為 20mA@5V，請不要超出此規格使用，以免毀損該元件

- 訊號連接範例圖：

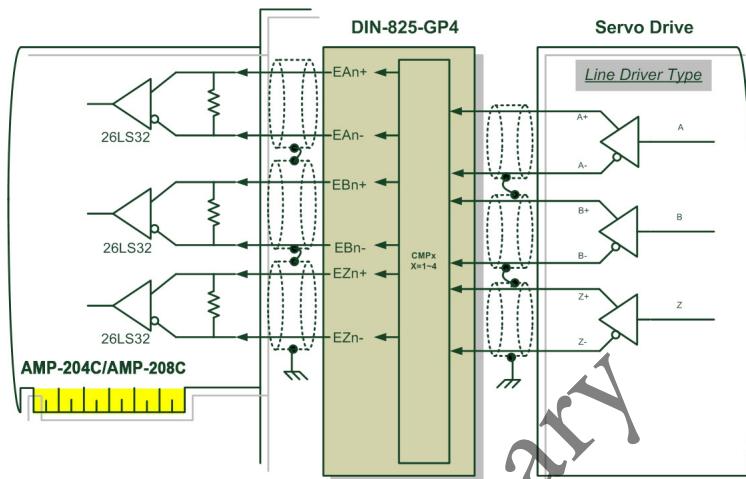


圖 3-3：編碼器輸入訊號 Line Driver 型連接範例圖

3.3 緊急停止訊號 (Emergency Stop Input)

AMP-204C / AMP-208C 提供一組硬體輸入緊急停止訊號 (EMG)。當外部緊急停止訊號被觸發時，會立即停止所有運動控制命令。此外，DIN-825-GP4 亦設計將外部緊急停止訊號同步送達伺服 / 步進馬達驅動器，可即時停止各馬達之運動。以下為 DIN-825-GP4 上之緊急停止訊號輸入對應腳位：

J5 Pin No	Signal Name	Description	Axis #
5	EEMG	External emergency stop input (外部輸入)	-

CMPx / CMAX Pin No (x=1~4)	Signal Name (n=1~8)	Description	Axis #
19	EMG(n)	Emergency stop (輸出至驅動器)	1~8



AMP-208C 需搭配 2 組 DIN-825-GP4 以完成八軸運動控制功能
#1 負責軸 1~ 軸 4 ; #2 負責軸 5~ 軸 8

- 訊號連接範例圖：

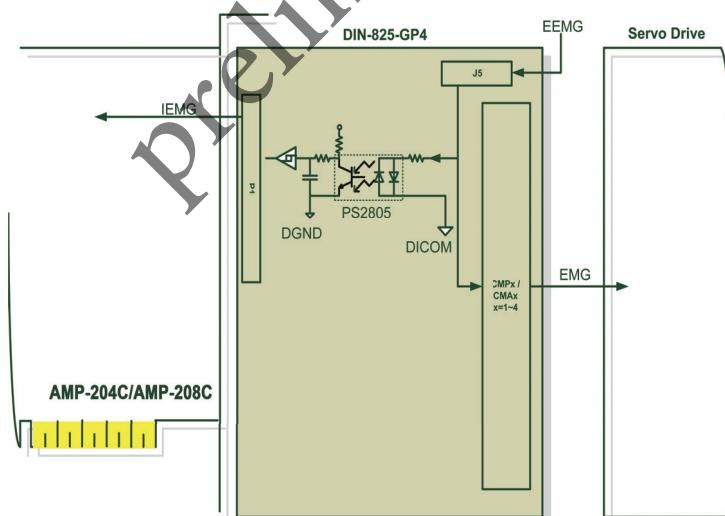


圖 3-4：緊急停止訊號連接範例圖

3.4 機械極限開關訊號 (PEL/MEL signals Input)

AMP-204C / AMP-208C 提供 4/8 組機械極限開關 (End-limited switch) 輸入通道。正極限開關 (Plus Limited Switch, PEL) 被使用於當運動往正方向行進時之機械保護開關，當此開關被觸發時，AMP-204C / AMP-208C 會立即停止正方向運動。負極限開關 (Minus Limited Switch, MEL) 被使用於當運動往負方向行進時之機械保護開關，當此開關被觸發時，AMP-204C / AMP-208C 會立即停止負方向運動。以下為 DIN-825-GP4 上之機械極限開關訊號輸入對應腳位：

J1/J2 Pin No	Signal Name	Description	Axis #
3	PEL(3) / PEL(1)	正極限開關輸入 (3) / (1)	3 / 1
7	PEL(4) / PEL(2)	正極限開關輸入 (4) / (2)	4 / 2
5	MEL(3) / MEL(1)	負極限開關輸入 (3) / (1)	3 / 1
9	MEL(4) / MEL(2)	負極限開關輸入 (4) / (2)	4 / 2



註釋

AMP-208C 需搭配 2 組 DIN-825-GP4 以完成八軸運動控制

功能

#1 負責軸 1~ 軸 4 ; #2 負責軸 5~ 軸 8

• 訊號連接範例圖：

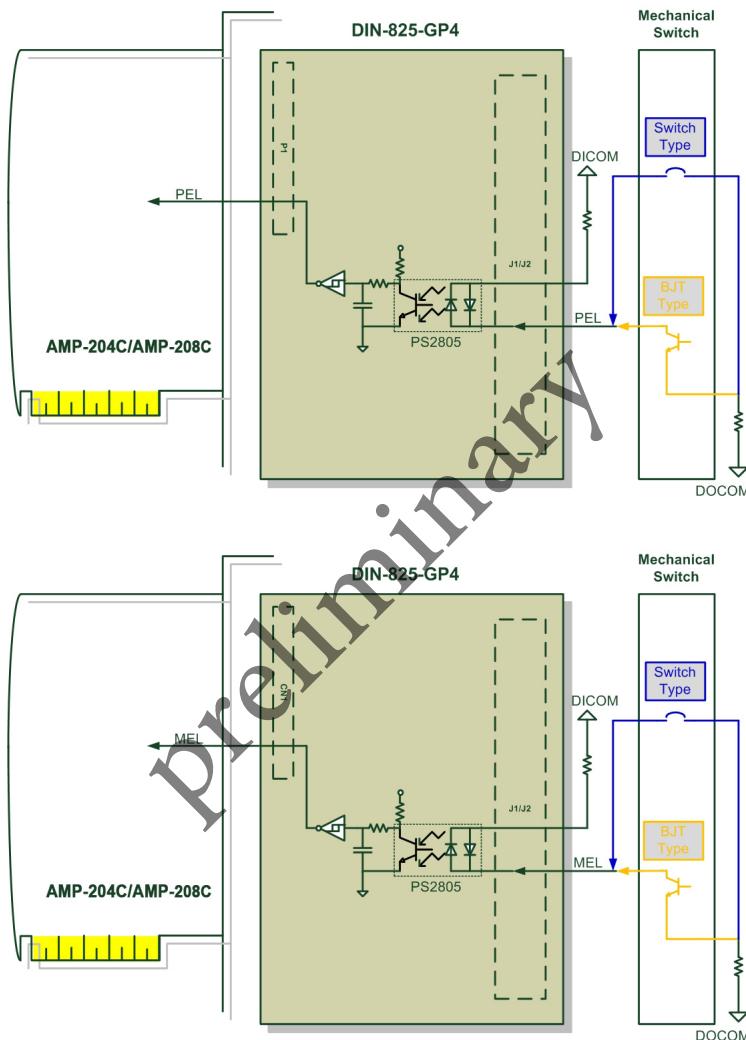


圖 3-5：機械極限開關訊號連接範例圖

3.5 原點開關訊號 (ORG signals Input)

AMP-204C / AMP-208C 提供 4/8 組原點開關輸入通道。搭配 4.4 節所述之歸零運動，可使得機構回到最初定義之原點 (Orign position)(亦稱為零位移點，Zero position)。以下為 DIN-825-GP4 上之原點開關訊號輸入對應腳位：

J1/J2 Pin No	Signal Name	Description	Axis #
4	ORG(3) / ORG(1)	原點開關輸入 (3) / (1)	3 / 1
8	ORG(4) / ORG(2)	原點開關輸入 (4) / (2)	4 / 2



AMP-208C 需搭配 2 組 DIN-825-GP4 以完成八軸運動控制
功能
#1 負責軸 1~ 軸 4 ; #2 負責軸 5~ 軸 8

- 訊號連接範例圖：

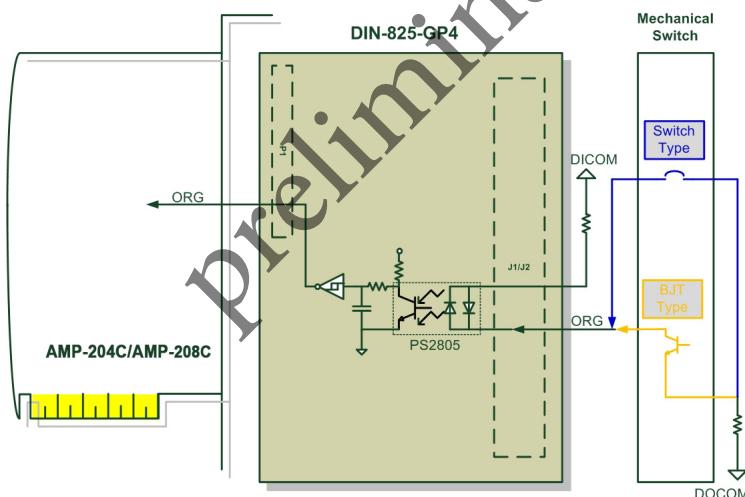


圖 3-6：原點開關訊號連接範例圖

3.6 到位 / 零速度檢出訊號 (INP signals Input)

AMP-204C / AMP-208C 提供 4/8 組到位 (In-position, INP) 輸入通道。搭配 4.3 節所述，可作為各式運動之到位事件觸發 (event trigger) 來源。一般而言，伺服驅動器被設置為位置模式 (P mode) 時，當運動到位時，驅動器會發出一脈衝訊號 (通常被稱為 INP)，通知控制器已到位。(in-position)。以下為 DIN-825-GP4 上之到位 / 零速度檢出訊號輸入對應腳位：

CMPx Pin No (x=1~4)	Signal Name (n=1~4)	Description	Axis #
2	INP(n)	到位訊號 (僅供脈衝輸出模式時使用)	1~4



AMP-208C 需搭配 2 組 DIN-825-GP4 以完成八軸運動控制功能
#1 負責軸 1~ 軸 4 ; #2 負責軸 5~ 軸 8

- 訊號連接範例圖：

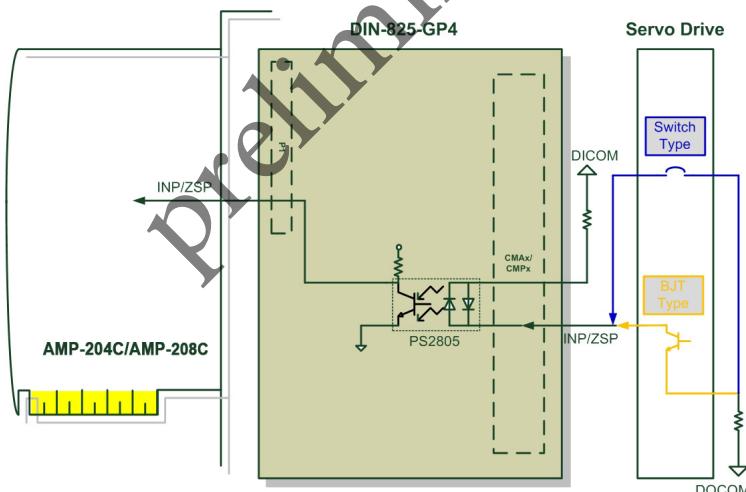


圖 3-7：到位檢出訊號連接範例圖

3.7 伺服警報訊號 (ALM signals Input)

AMP-204C / AMP-208C 提供 4/8 組伺服警報 (servo alarm) 輸入通道。搭配 4.11 節所述，可作為運動中斷事件觸發來源。一般而言，當伺服驅動器在行進間發生異常時，會輸出一脈衝訊號 (ALM)，通知控制器有異常發生。以下為 DIN-825-GP4 上之伺服警報訊號輸入對應腳位：

CMAX / CMPx Pin No (x=1~4)	Signal Name (n=1~4)	Description	Axis #
11	ALM(n)	伺服警報訊號	1~4



AMP-208C 需搭配 2 組 DIN-825-GP4 以完成八軸運動控制功能
#1 負責軸 1~ 軸 4 ; #2 負責軸 5~ 軸 8

- 訊號連接範例圖：

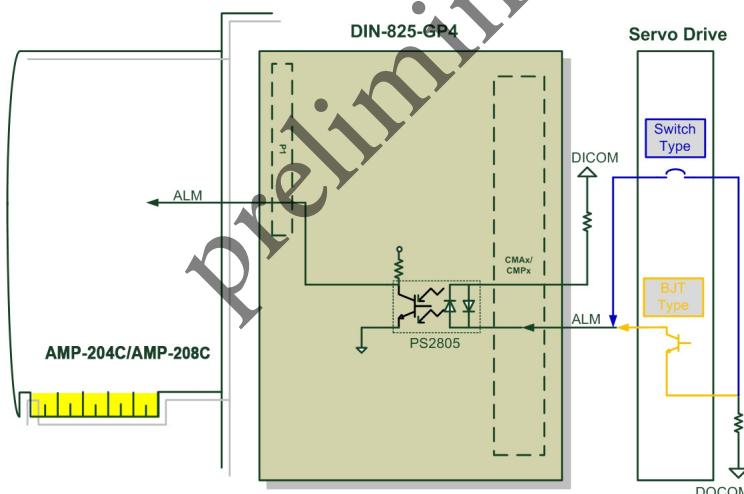


圖 3-8：伺服警報訊號連接範例圖

3.8 伺服激磁訊號 (SVON signals Output)

AMP-204C / AMP-208C 提供 4/8 組伺服激磁 (servo on) 輸出通道，並利用此伺服激磁訊號使得伺服驅動器激磁 (utilize the servo-on signal to enable servo drive)，並允許脈衝或類比命令輸入。當行進間 (motion) 發生異常時，AMP-204C / AMP-208C 會自動關閉此訊號，並停止所有運動控制命令。以下為 DIN-825-GP4 上之伺服激磁訊號輸出對應腳位：

CMAx / CMPx Pin No (x=1~4)	Signal Name (n=1~4)	Description	Axis #
1	SVON(n)	伺服激磁訊號	1~4



AMP-208C 需搭配 2 組 DIN-825-GP4 以完成八軸運動控制功能

#1 負責軸 1~ 軸 4 ; #2 負責軸 5~ 軸 8

- 訊號連接範例圖：

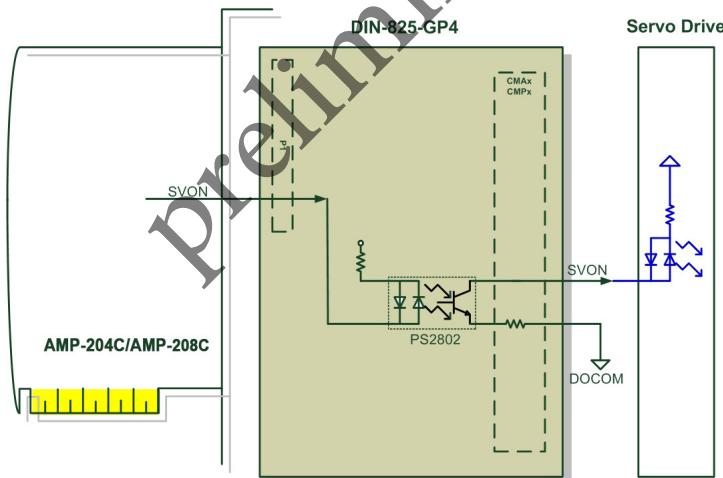


圖 3-9：伺服激磁訊號連接範例圖

3.9 比較觸發訊號 (Comapre & Trigger Output):

AMP-204C / AMP-208C 提供 2/4 比較觸發脈衝輸出通道，每組比較觸發通道可輸出高達 1MHz 脈衝命令。使用方式及細節請參考第 4.10.2 節。以下為差分脈衝訊號至 DIN-825-GP4 上之脈衝命令輸出之對應腳位：

J3 Pin No	Signal Name	Description
2	TRG2-/TRG4-	Trigger output (-), (2)/(4)
3	TRG2+ / TGR4+	Trigger output (+), (2)/(4)
4	TRG1-/TRG3-	Trigger output (-), (1)/(3)
5	TRG1+/TRG3+	Trigger output (+), (1)/(3)



註釋

AMP-204C / AMP-208C 之比較觸發脈衝輸出通道使用 Line Driver 輸出介面，可提供較佳的抗雜訊能力及較長的配線長度，其中 Trigger output (3) & (4) 需要由 #2 DIN-825-GP4 完成配線。

- 訊號連接範例圖：

1. Line Driver 介面：

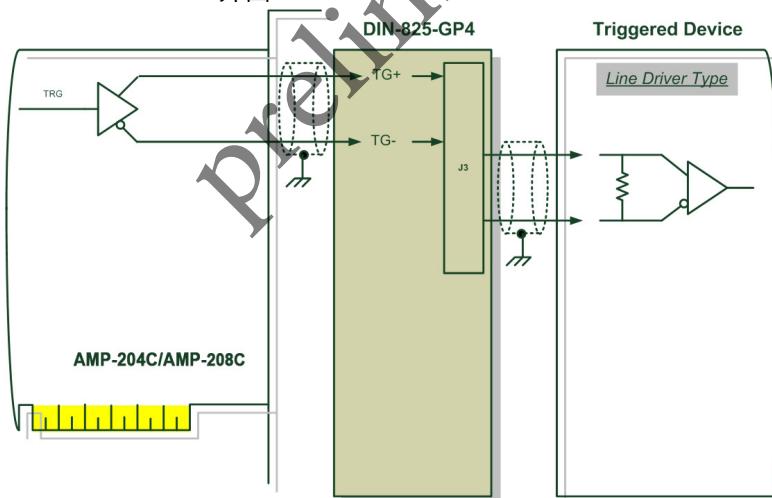


圖 3-10：比較觸發訊號 Line Driver 型連接範例圖

2. Open-Collector 介面：

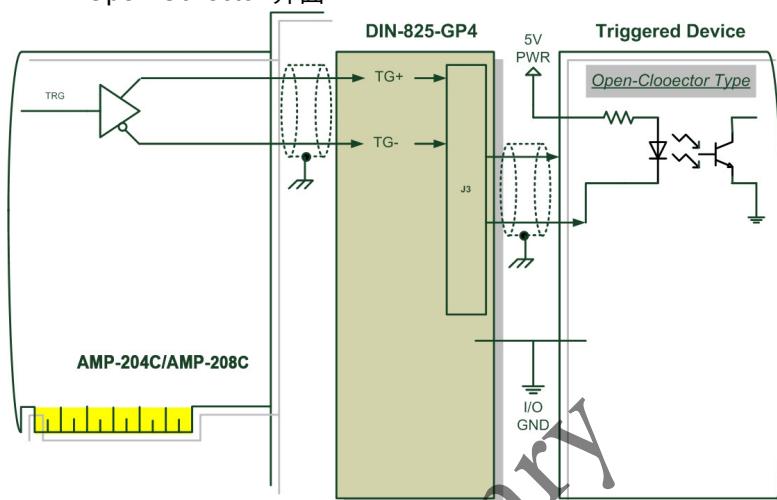


圖 3-11：比較觸發訊號 Open-Collector 型連接範例圖

3.10 泛用數位輸出入訊號 (Digital signals Output/Input)

AMP-204C / AMP-208C 提供共 20/24 組數位輸出入通道。以下為 DIN-825-GP4 上之泛用數位輸出入訊號對應腳位：

J1/J2 Pin No.	Signal Name	Description
2	EDI(3) / EDI (1)	泛用數位輸入訊號 (3), (1)
6	EDI(4) / EDI (2)	泛用數位輸入訊號 (4), (2)

J6 Pin No.	Signal Name	Description
1	EDO(1)	泛用數位輸出訊號 (1)
2	EDO(2)	泛用數位輸出訊號 (2)
3	EDO(3)	泛用數位輸出訊號 (3)
4	EDO(4)	泛用數位輸出訊號 (4)



註釋

AMP-208C 需搭配 2 組 DIN-825-GP4 以完成八軸運動控制
功能
#1 負責軸 1~ 軸 4 ; #2 負責軸 5~ 軸 8



註釋

1. DICOM 建議連接至外部電源供應 (一般為 24VDC)
2. DOCOM 建議連接至外部電源的地 (GND)

• 訊號連接範例圖：

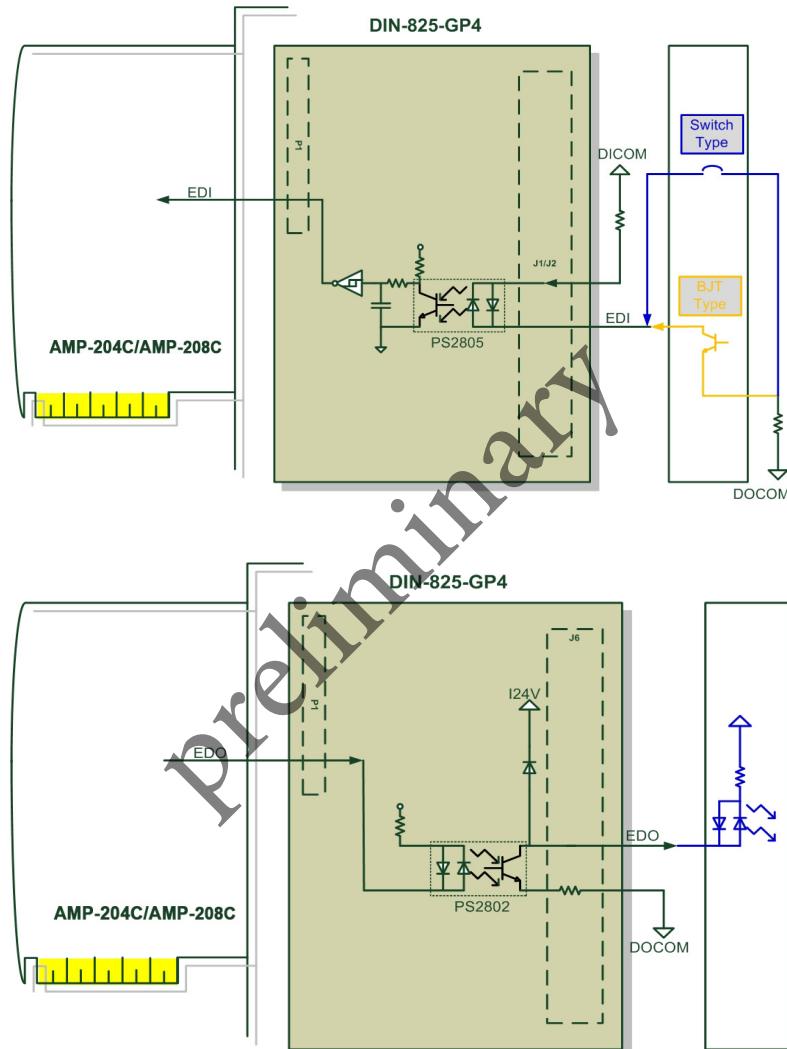


圖 3-12：泛用數位輸出入訊號連接範例圖

IOIF1 Pin No.	Signal Name	Description
1~8	DI(1)~(8)	泛用 IOIF2 數位輸入訊號 (1)~(8)

IOIF2 Pin No.	Signal Name	Description
1~8	DI(9)~(16)	泛用數位輸入訊號 (9)~(16)

IOIF3 Pin No.	Signal Name	Description	Axis #
※1~5	DO(1)~(5)	泛用數位輸出訊號 (1)~(5)	-



註釋

※ 表示該數位輸出電流可達 250mA

IOIF3 Pin No.	Signal Name	Description	Axis #
6~8	DO(6)~(8)	泛用數位輸出訊號 (6)~(8)	-

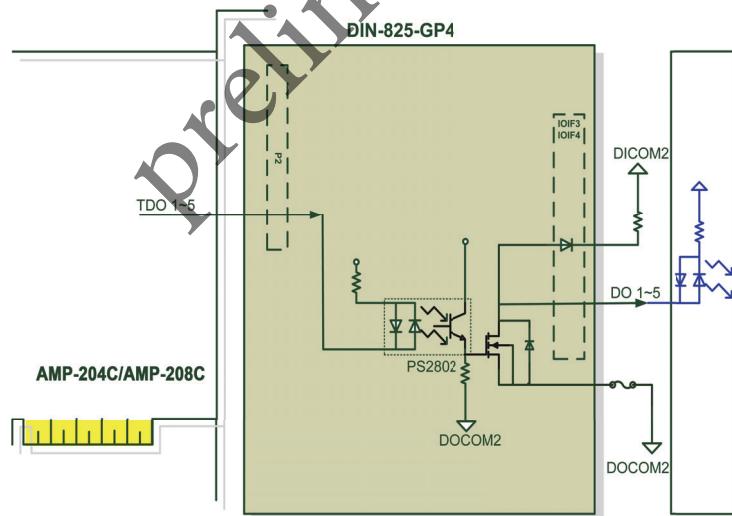
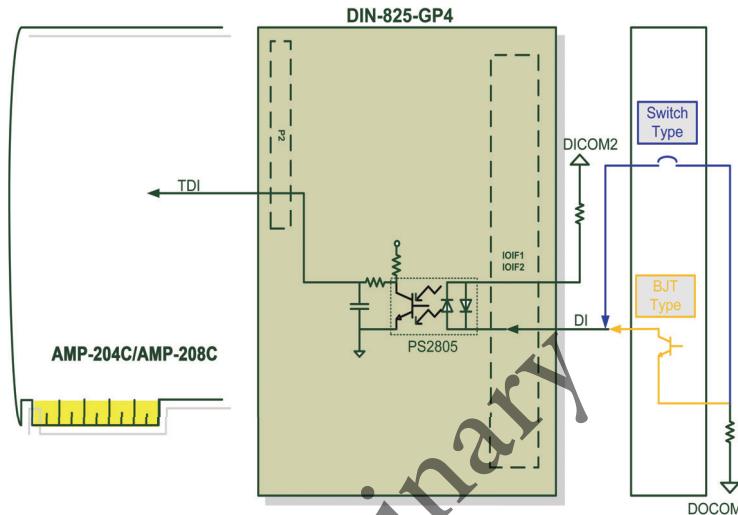
IOIF4 Pin No.	Signal Name	Description	Axis #
1~8	DO(9)~(16)	泛用數位輸出訊號 (9)~(16)	-



註釋

1. DICOM 建議連接至外部電源供應 (一般為 24VDC)
2. DOCOM 建議連接至外部電源的地 (GND)

• 訊號連接範例圖：



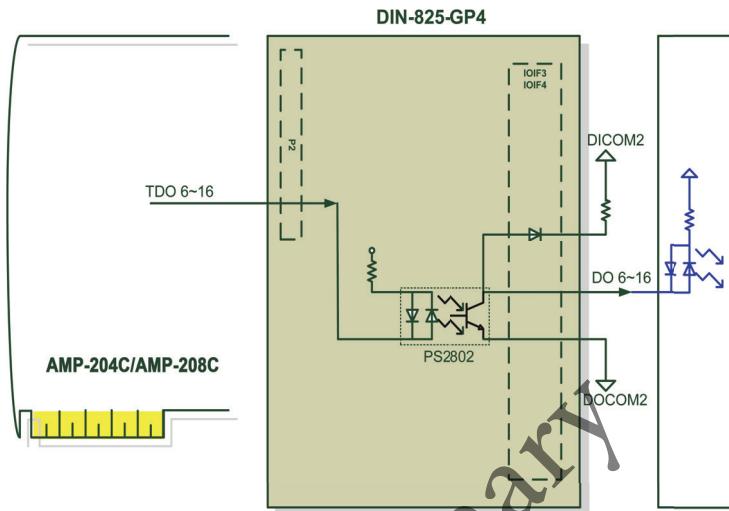


圖 3-13：泛用數位輸出入訊號連接範例圖

4 運動控制功能原理介紹

本章節將介紹此AMP-204C / AMP-208C運動控制原理及注意事項。

以下為各節簡述：

- 第 4.1 節：** 運動控制模式暨介面簡介
- 第 4.2 節：** 運動控制操作 (Motion Control Operations)
- 第 4.3 節：** 歸零運動 (Home Return Move)
- 第 4.4 節：** 速度運動 (Velocity Move)
- 第 4.5 節：** JOG 運動 (Jog Move)
- 第 4.6 節：** 點對點運動 (Point-to-Point Move)
- 第 4.7 節：** 多軸補間運動 (Interpolation)
- 第 4.8 節：** 運動狀態監控 (Motion Status Monitoring)
- 第 4.9 節：** 應用功能 (Application Functions)
- 第 4.10 節：** 安全保護 (Safety Protection)
- 第 4.11 節：** 本機中斷 (Host Interrupt)

4.1 運動控制模式暨介面簡介

本節將介紹本控制器「AMP-204C」及「AMP-208C」在進行運動控制前的基礎設定以及核心運作的基礎觀念。

4.1.1 運動控制介面 (Motion Control Interface)

4.1.1.1 控制模式及輸出介面

其設定方式可利用 **MotionCreatorPro2** 應用程式進行設定，此設定值會儲存在控制器上的非揮發性記憶體 (Non-volatile memory，一般指 ROM) 上，爾後控制器開機時會自動載入此設定。另外，可利用下列 API 讀出目前設定值，用來做設定值的確認。

`APS_get_eep_curr_drv_ctrl_mode()`

4.1.1.2 脈衝式控制 (Pulse Type)

此種控制模式可用於控制步進電機 (Stepping motor) 或者設定為位置控制模式 (P control mode) 輸入訊號為脈衝格式的伺服電機 (Servo motor)。控制器的輸出介面為 OUT / DIR[1..8] 腳位。(請參考第三章)

此控制架構即所謂的開迴路 (Open-loop) 或半閉迴路 (Semi closed-loop)，亦即上位控制器以數位脈衝的訊號格式輸出位置命令至下位的步進電機或是伺服電機，而閉迴路控制在伺服驅動器中完成。此模式下，脈衝的數量表示實際機構所行走之距離 (需換算機械位移與脈衝之間關係)，同時，脈衝輸出頻率亦即表示機構所行之速度 (一般以 PPS, pulse per second 表示)。

此模式下須注意被驅動電機所能接受的脈衝訊號格式。必須正確的設定脈衝格式訊號，馬達才會正常作動，若設定錯誤，可能會出現馬達不運轉，運轉方向錯誤或馬達不正常抖動。一般在軟體初始化過程後進行任何運動控制前必須被正確地設定。本控制器提供兩種脈衝訊號輸出格式：

- OUT/DIR 訊號格式：此模式下 OUT 訊號表示輸出的脈衝頻率及數量，DIR 表示機構位移為正 / 反向
- CW/CCW 訊號格式：此模式下 CW/CCW 訊號同時表示機構位移方向及脈衝輸出頻率及數量

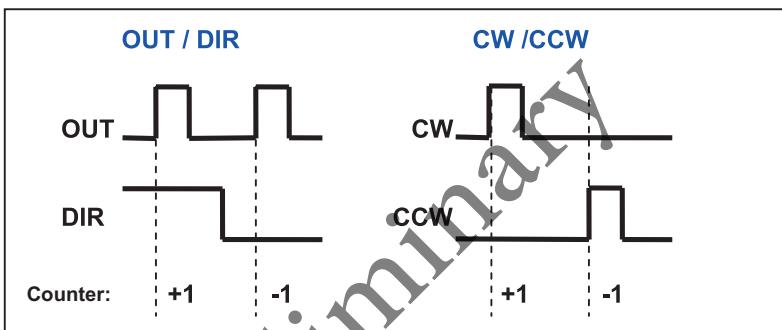


圖 4-1：脈衝式訊號格式

其設定方式於軸參數中設定：

Param. No.	Define symbol	Description
81h (129)	PRA_PULSE_OUT_MODE	脈波輸出格式設定

4.1.1.3 編碼器 (Encoder)

本控制器的位置編碼器 (Encoder input) 支援 9 種數位訊號輸入格式，如下表所示。



注意

在進行運動控制前必須正確設定位置編碼器，尤其若控制模式採用類比輸出的全閉迴路控制時 (Closed-loop control)，若設定不正確可能造成伺服馬達暴衝現象。



註釋

您可以使用 MotionCreatePro 2 軟體來設定並測試，其方式先用手動方式旋轉馬達或 (或移動載台) 來確認馬達或光學尺 (linear scale) 到控制器這端的編碼器訊號是否正確。

No	Decode Mode	Positive direction		Negative direction	
		EA	EB	EA	EB
0	OUT/DIR (1)		High		Low
1	CW/CCW (1)		Low	Low	
2	1X AB				
3	2x AB				
4	4x AB				
5	OUT/DIR (2)		High		Low
6	OUT/DIR (3)		Low		High
7	OUT/DIR (4)		Low		High

No	Decode Mode	Positive direction		Negative direction	
		EA	EB	EA	EB
8	CW/CCW (2)		High	High	

表 4-1：編碼器輸入格式

- 軸參數設定：

Param. No.	Define symbol	Description
80h (128)	PRA_ENCODER_MODE	編碼器 (Encoder) 輸入訊號格式
85h (133)	PRA_ENCODER_DIR	編碼器 (Encoder) 計數方向設定

表 4-2：編碼器輸入格式

- 軸參數設定 API：

`APS_set_axis_param(); // 寫入軸參數`

`APS_get_axis_param(); // 讀出軸參數`

4.1.1.4 運動控制 I/O (Motion control I/O)

本控制器定義常見的運動控制 I/O 訊號，整理如下表：

Param.	Defined Symbol	Type	Description
0	ALM	Input	伺服電機警報輸入訊號 (Servo alarm)
1	PEL	Input	正端機械極限輸入訊號 (Plus end limit)
2	MEL	Input	負端機械極限輸入訊號 (Minus end limit)
3	ORG	Input	原點輸入訊號 (Home input)
4	EMG	Input	緊急停止輸入訊號 (Emergency stop input)
5	EZ	Input	編碼器 Z 向輸入訊號 (Servo index input)
6	INP	Input	伺服到位輸入訊號 (In-Position input)
7	SVON	Output	伺服激磁輸出訊號 (Servo ON output status)

其中，ALM, EZ 和 INP 是由伺服驅動器 (Servo drive) 所發出的訊號，SVON 訊號 (Servo on) 是伺服驅動器的接收訊號用來啟動伺服電機。

另外 PEL, MEL, ORG 和 EMG 訊號是屬於機械 IO。其與安全性相關的如 EMG, PEL 和 MEL，或者是用來配合運動控制使用，如定位運動會使用 OGR, PEL, MEL, EZ 等。

使用下列 API 函式可取得 I/O 狀態，變數中每個位元代表該軸運動控制 I/O 的狀態。

I32 APS_set_servo_on(I32 Axis_ID, I32 Servo_on);

I32 APS_motion_status(I32 Axis_ID);

- **訊號反向**

這些訊號邏輯 (Logic) 可經由軟體的方式設定是否反向，相關軸參數如下：

- **卡片參數 (Board parameter) :**

Param. No.	Define symbol	Description
00h (0)	PRA_EL_LOGIC	PEL/MEL input logic
01h (1)	PRA_ORG_LOGIC	ORG input logic
04h (4)	PRA_ALM_LOGIC	Set ALM logic
05h (5)	PRA_ZSP_LOGIC / PRA_INP_LOGIC	Set INP logic
06h (6)	PRA_EZ_LOGIC	Set EZ logic

- **卡片參數 (Board parameter) :**

Param. No.	Define symbol	Description	Value	Default
00h (0)	PRS_EMG_LOGIC	EMG input logic	0 : Not inverse 1 : Inverse	0

- **濾波 (Filter)**

為避免外部雜訊干擾，導致運動控制不正常作動，針對運動控制 I/O 本控制器設計濾波器可過濾高頻雜訊 (High-frequency noise)，濾波器預設是啟動的。

4.1.2 控制週期 (Control Cycle)

一般而言，運動控制器內有三種不同控制週期來執行各種不同的工作，這三種週期分別為

1. 伺服控制週期
2. 運動控制週期
3. 系統工作週期

4.1.2.1 伺服控制週期 (Servo Control Cycle)

伺服控制週期亦即執行一次閉迴路控制所需的時間，本控制器的伺服控制週期高達 20KHz，即一個週期的時間為 50 microsecond，，在這個控制週期中會完成如 PID 演算 (PID compensation)、濾波演算 (Filter compensation) 等伺服控制相關工作。

4.1.2.2 運動控制週期 (Motion control cycle)

運動控制週期預設值為 1KHz，即一個週期的時間為 1 millisecond，在這個控制週期中會完成如主機間通訊 (Host communication)，運動控制，軌跡計算 (Trajectory calculation)，資料採集 (Data sampling)，周邊硬體元件控制等相關工作。

4.1.2.3 系統工作週期 (Host Control Cycle)

系統工作週期預設值為 0.5KHz，即一個週期的時間為 2 ms (millisecond)，在這個控制週期中主要執行如主機間通訊，看門狗程序 (Watch dog)，內核更新 (Kernel update) 程序，參數管理等和一些較非即時性的工作。

伺服控制週期為獨立運作，而運動控制和系統工作周期的工作是在同一處理器中完成，控制器會自動完成排程的工作，運動控制週期的優先權高於系統工作，如下圖所示。

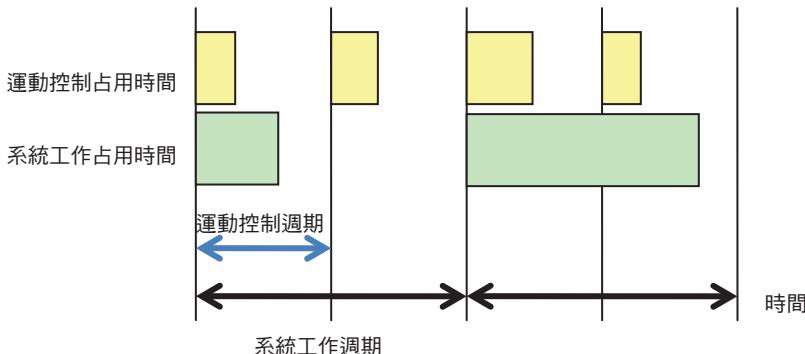


圖 4-2：控制週期

運動程式會在運動控制週期中被執行，並且使用者可讓該程序直接控制每個運動控制週期中有哪些工作要執行，讓使用者可以更精準的完成即時性的工作，但也必須注意處理器的使用 (DSP loading)。

控制器在執行程序時，可能會因為外部的訊號，使用者的操作，演算法 (Algorithm) 的流程等讓處理器工作量難以具體的預測，所以一般情況下盡量讓運動控週期對處理器的使用率低於 70%，其餘 30% 保留給系統工作和預留處理某瞬間的工作量高峰。

若工作量超過控制週期 (工作超載，Overloading)，可能會產生不預期的結果，因此本控制器提供一些函式，工具讓您方便監控處理器的使用量，便於調整您的控制程序，若使用量超過處理器的負荷，控制器會紀錄並發出警告 (中斷，請參考中斷章節)，讓你可以在程序上做適當的回應處理。

相關 API 使用方式如下：

```

get_motion_control_timing() // 取得當下運動控制週期的使用量
get_max_motion_control_timing() // 取得運動控制週期的最大使用量
get_host_control_timing() // 取得當下系統工作週期的使用量
get_max_host_control_timing() // 取得系統工作週期的最大使用量
reset_max_motion_control_timing()
reset_max_host_control_timing()
get_over_cycle_event()
get_over_cycle_count()
reset_over_cycle_count()

```

4.2 運動控制操作 (Motion Control Operations)

本章將介紹控制器所提供之運動控制模式，及其基本原理，目的是讓使用者可更精準掌握控制器的運動控制能力，協助完成特定應用。

4.2.1 座標系統 (Coordinated System)

本控制器採用直角座標系統 (卡式座標系，Cartesian coordinate system)，在此座標系統中可以進行一到多軸的運動，而座標系統中的各軸則一對一的對應至馬達上。座標系統內的軸和實際控制的馬達存在一層轉換的關係，而這層關係讓使用者可以自由設定自己的座標系統。下圖為座標系統關係，其中單位轉換因子在下小節介紹。

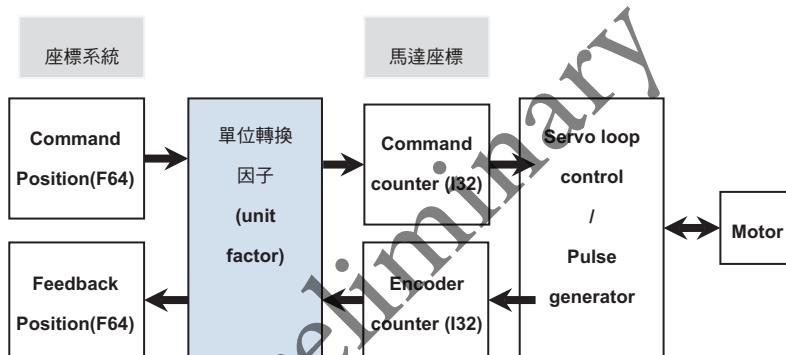


圖 4-3：控制器座標系統方塊圖

可透過下列 API 讀出或設定座標命令位置及實際座標位置

`I32 APS_get_command_f(I32 Axis_ID, F64 *Command);` // 命令位置讀取
`I32 APS_set_command_f(I32 Axis_ID, F64 Command);` // 命令位置設定
`I32 APS_get_position_f(I32 Axis_ID, F64 *Position);` // 實際位置讀取
`I32 APS_set_position_f(I32 Axis_ID, F64 Position);` // 實際位置設定

I32 作標格式相容 API，同上列 API 功能

`I32 APS_get_command(I32 Axis_ID, I32 *Command);`

```
I32 APS_set_command(I32 Axis_ID, I32 Command);
I32 APS_get_position( I32 Axis_ID, I32 *Position );
I32 APS_set_position(I32 Axis_ID, I32 Position);
```

下列 API 可以讀取馬達座標

```
I32 APS_get_encoder( I32 Axis_ID, I32 *Encoder );
I32 APS_get_command_counter( I32 Axis_ID, I32 *Counter );
```



閉迴路控制過程中，任意設定 Command counter 以及 Encoder counter 是不允許的，因此不提供相關設定 API。

註釋

4.2.2 單位轉換因子 (Unit Factor)

透過設定單位因子，馬達的位置單位（或移動機構）可以與座標系統的物理距離單位能夠有實際的對應。其計算公式如下

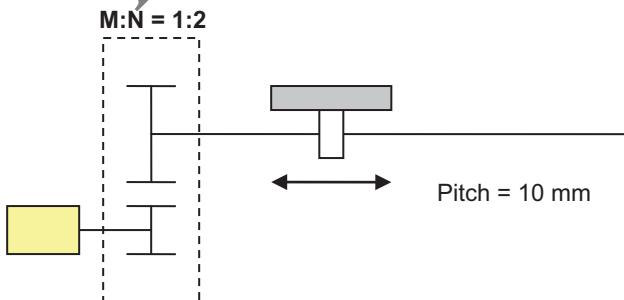
$$\text{Unit Factor} = \frac{\text{Encoder Resolution}}{\text{Pitch (User define unit)}} \times \frac{N \text{ (減速齒數)}}{M \text{ (驅動齒數)}}$$

以下以三個範例來說明如何計算單位轉換因子 (Unit factor)：

例 1：滾珠螺桿載台

假設馬達旋轉一周產生的編碼器計數（解析度）為 10000，螺桿的導程 (Pitch) 為 10 mm，若使用者欲使用的距離單位為 Micrometer，則：

減速機齒輪比

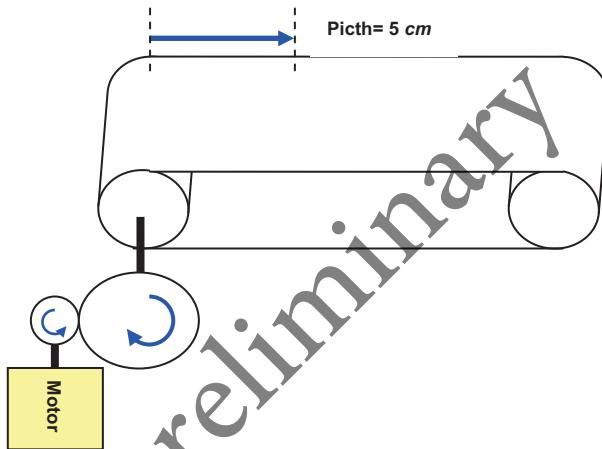


Unit factor 的計算方式如下：

$$\text{Unit factor} = \frac{10000}{10\text{mm} \times 1000 \mu\text{m}} \times \frac{2}{1} = 2$$

例 2：輸送帶系統

馬達旋轉一圈的脈波數為 8192，皮帶輪旋轉一周輸送帶位移 5cm，齒輪比 (Gear ratio) 為 1:2，若欲使用 Millimeter 為距離單位，則距離因子 (Unit factor) 為：



Unit factor 的計算方式如下：

$$\text{Unit factor} = \frac{8192}{5\text{cm} \times 100 \text{ mm}} \times \frac{2}{1} = 32.768$$

例 3：線性馬達 (Linear motor) 系統使用光學尺 (Linear scale)

加設光學精度 (Resolution) 為 1 Micrometer，若欲使用 Millimeter 為距離單位，則距離因子 (Unit factor) 為：

$$\text{Unit factor} = \frac{1}{1\mu\text{m} \times 1000 \text{ mm}} = 0.001$$

Unit factor 可在軸參數中設定：

Param. No.	Define symbol	Description	Value	Default
86h (134)		Unit factor	F64 value	1

一般而言，在進行運動控制應用設計前，單位的定義應先被確認固定，再進行其他與位置相關的參數設定。



註釋

若操作中 Unit factor 有設定有變動，其他和距離單位相關的參數 (如位置單位，速度單位和加速度單位等) 單位意義也將會有影響，使用者應自行變更調整相關設定。

相關軸參數如下列表：

Param. No.	Define symbol
07h (7)	PRA_SD_DEC
0Ah (10)	PRA_SPEL_POS0
0Bh (11)	PRA_SMEL_POS1
13h (19)	PRA_HOME_ACC
15h (21)	PRA_HOME_VM
17h (26)	PRA_HOME_SHIFT
19h (25)	PRA_HOME_V0
1Bh (27)	PRA_HOME_POS
21h (33)	PRA_ACC
22h (34)	PRA_DEC
23h (35)	PRA_VS
24h (36)	PRA_VM
25h (37)	PRA_VE
2Ah (42)	PRA_PRE_EVENT_DIST
2Bh (43)	PRA_POST_EVENT_DIST
43h (67)	PRA_JG_ACC
44h (68)	PRA_JG_DEC
45h (69)	PRA_JG_VM
46h (70)	PRA_JG_OFFSET



註釋

部分 API 亦有位置相關的輸入參數端的編碼器訊號是否正確。

4.2.3 加減速曲線 (Acc/Deceleration Profile)

一個基本的運動命令通常包含：1. 距離，2. 速度，3. 加速度等資料，透過這些運動命令參數，控制器會自動規劃計算加減速度曲線 (Acceleration & deceleration) 使運動能在使用者要求下平順地完成。本控制器提供下列加速曲線：

1. 梯形速度曲線 (Trapezoidal speed profile, T-curve)
2. S 形速度曲線 (S-Curve)

4.2.3.1 梯形速度曲線 (Trapezoidal speed profile, T-curve)

梯形速度曲線（也稱 "T- 曲線"，T-curve）意指期加速和減速區遵循一階線性 (First-order linear) 速度曲線（等加速度）。如下圖速度 - 時間圖 (V-T) 所示

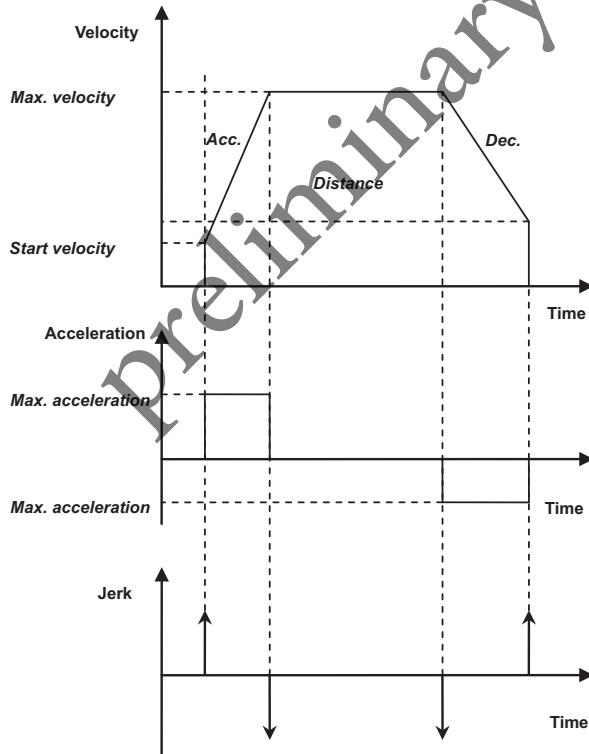


圖 4-4：梯型速度曲線之速度 / 加速度 / 加加速度 對時間關係圖

在 V-T 圖中，梯形曲線內的面積代表移動距離。當使用者所設定的移動距離不足時，控制器會自動調整（調降）最大速度，但保持加速度設定。如下圖所示：

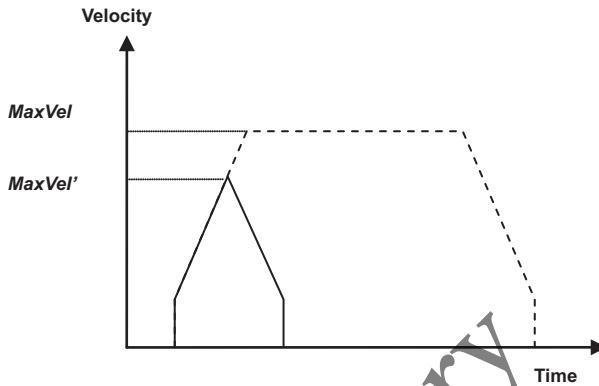


圖 4-5：自動規劃最大速度示意圖

使用者設定的最大速為 ($MaxVel$)，虛線表示在足夠距離下的速度曲線 (Speed profile)，但因移動距離不足，控制器自動調整最大速為 ($MaxVel'$)。但加減速率則保持不變，以保持最佳 (最短) 的移動時間。

4.2.3.2 S 形速度曲線 (S-Curve)

S 形曲線是指在加加速區 (Jerk) 的速度曲線遵循二階曲線 (Second-order profile)。有助於減少馬達在啟動和停止時的振動，如下圖 (t1, t3, t5, t7) 所示。

為了加快加速和減速的時間，我們需要插入到這些地區的線性部分 (t2, t6) 以維持最大加速度率，而加速度率 - 時間圖 (A-T 圖) 呈現梯形。

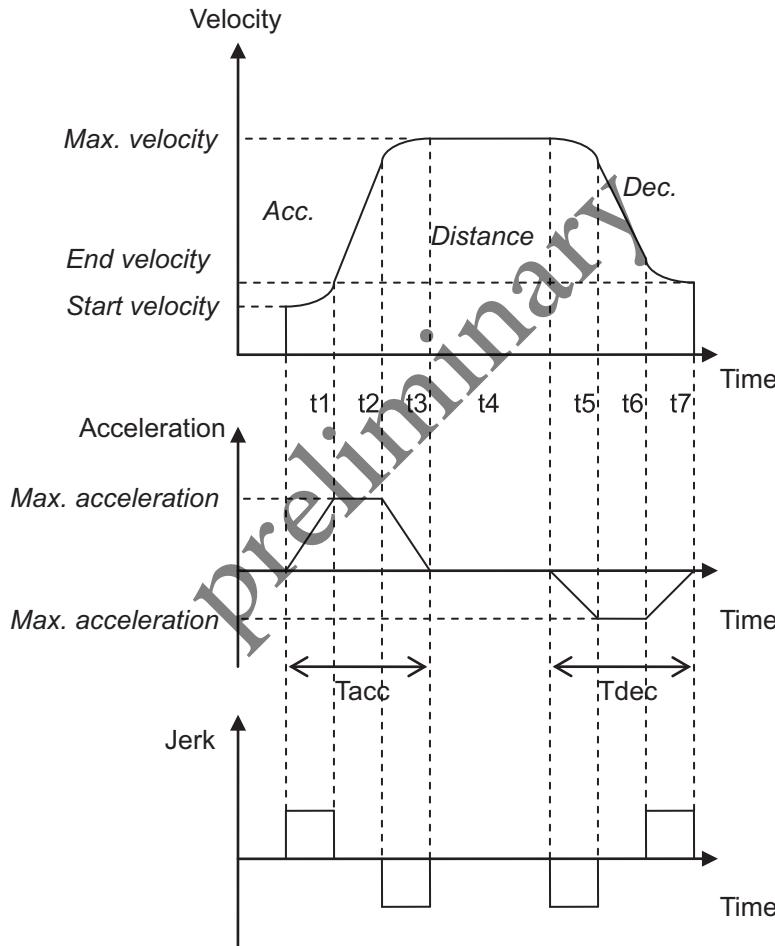


圖 4-6 : S 形速度曲線之速度 / 加速度 / 加加速度 對時間關係圖

本控制器使用 S-factor (S) 來控制加加速 (Jerk) 的比率, 其公式如下

$$S = \frac{t1}{t1+t2}; \text{ and } t1 = t2$$

S 的值介於 0 到 1, 因此當

$S = 0$ 速度曲線為 T 型曲線 (T – curve)

$S > 0$ 且 $S \leq 1$: s 型曲線 (S – curve)

其中, 當 $S = 1$ 時我們又稱純 S 型曲線 (Pure S – curve), 其 A-T 圖呈三角形。

由上述公式可知, 當 S 的值調的越大表示加速曲線越平滑, Jerk 值會越小, 可助降低機台的震動, 但相對的, 運動過程所花費的時間也就越大。反之 S 越小 Jerk 會越大, 但運動時間最短。

如同 T 型曲線, 當移動距離不足時, 控制器會自動修正最大速度以維持運動的順暢。加速度 (ACC, DEC) 和 S 因子則保持一至以維持加速度, 而加加速度 (Jerk) 則會改變。如下圖

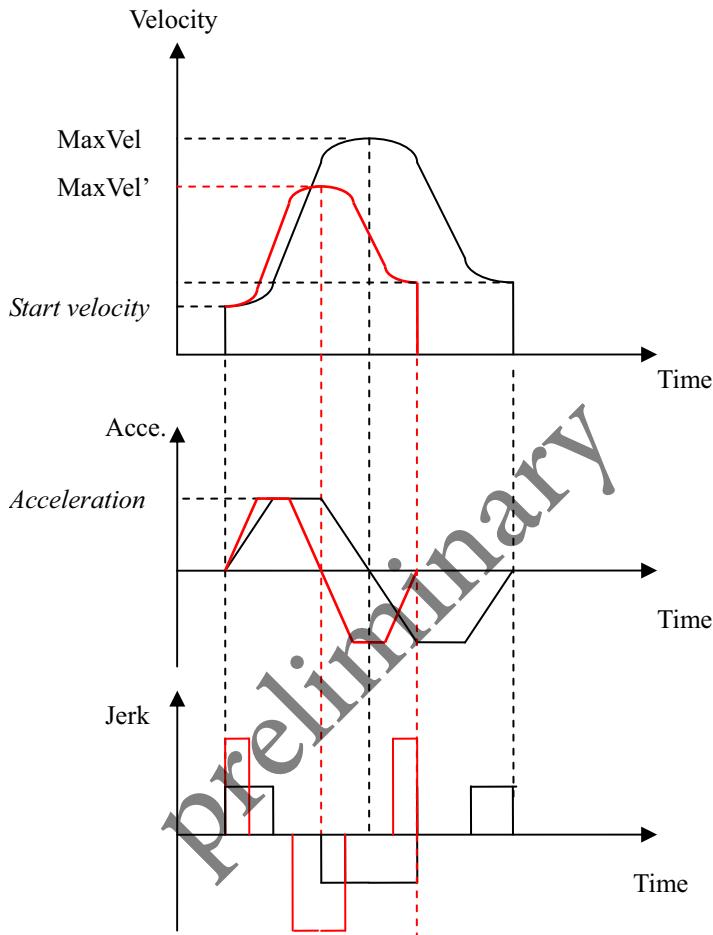


圖 4-7：自動規劃最大速度示意圖

上述加速曲線及其規則適用於單軸點對點運動 (PTP)，速度型運動 (Velocity move)，定位運動 (Homing) 和多軸的補間運動 (Interpolation)。

- **相關軸參數**

Param. No.	Define symbol	Description
12h (18)	PRA_HOME_CURVE	Home move S-factor
20h (32)	PRA_SF	Move S-factor
42h (66)	PRA_JG_SF	Jog S-factor



有些 API 中也可以直接設定 S-factor，請參閱 Function library 手冊

4.3 歸零運動 (Home Move)

本章將介紹控制器所提供之運動控制模式，及其基本原理，目的是讓使用者更精準掌握控制器的運動控制能力，協助完成特定應用。

一般運動控制系統在電源開啟後 (Power on) 進行任何運動控制前，會先進行定位運動來確立整個系統的座標 (Coordinate) 的原點 (Zero position)。

一般常見的步進馬達，伺服驅動器或搭配光學尺的線性馬達機構採用的是增量型編碼器 (Incremental type encoder)，此類裝置在進行定位時通常會配合使用一些機械訊號來配合完成定位。這些機械訊號包括 ORG, EZ, PEL 和 MEL 等。有些伺服驅動器提供絕對型編碼器 (Absolute type encoder)，例如三菱電機的 J3-B type (SSCNET 3)，此類的驅動器在最初系統初剛建立完成後也必須進行至少一次的定位運動，當絕對座標系確立後，爾後就無須再進行定位運動。

當定位指令下達時，控制器會透過一些外部訊號來進行原點位置 (Original position / Zero position) 的搜尋。當定位運動完成後，控制器會停在原點上，命令位置 (Command position) 和回授位置 (Feedback position) 會歸零。上述自動完成定位運動的所有動作將由控制器自動完成，使用者只需等待定位動作完成。

要完成定位運動的操作的方式如下述步驟：

- 設定 Home mode 及相關參數
- 啟動 Home move
- 等待 Home move 是否正確完成 (可使用輪詢 (Polling) 或使用中 (Interrupt) 的方式)
- Home move 完成或者針對不同錯誤情形進行處理

相關的 APS API 如下：

I32 APS_home_move(I32 Axis_ID);

相關的軸參數 (Axis parameter):

Param. No.	Define symbol	Description
10h (16)	PRA_HOME_MODE	Home mode 設定
11h (17)	PRA_HOME_DIR	Homing 方向設定
12h (18)	PRA_HOME_CURVE	Home move S-factor

Param. No.	Define symbol	Description
13h (19)	PRA_HOME_ACC	Homing 加減速率設定
15h (21)	PRA_HOME_VM	Homing 最大速率
17h (23)	PRA_HOME_SHIFT	Home 位置和定位訊號的偏移距離
18h (24)	PRA_HOME_EZA	EZ alignment enable
19h (25)	PRA_HOME_VO	Homing 離開 ORG 訊號的速率
1Bh (27)	PRA_HOME_POS	Homing 完成後位置命令設定

• 例：

```
#include "APS168.h"
#include "APS_define.h"
#include "ErrorDef.h"

void home_move_example()
{
    //This example shows how home move operates

    I32 axis_id = 0;
    I32 return_code;
    I32 msts;

    // 1. Select home mode and config home parameters
    APS_set_axis_param( axis_id, PRA_HOME_MODE, 0 ); //Set home mode
    APS_set_axis_param( axis_id, PRA_HOME_DIR, 1 ); //Set home direction
    APS_set_axis_param( axis_id, PRA_HOME_CURVE, 0 ); // Set acceleration pattern (T-curve)
    APS_set_axis_param( axis_id, PRA_HOME_ACC, 1000000 ); // Set homing acceleration rate
    APS_set_axis_param( axis_id, PRA_HOME_VM, 100000 ); // Set homing maximum velocity.
    APS_set_axis_param( axis_id, PRA_HOME_VO, 50000 ); // Set homing
    APS_set_axis_param( axis_id, PRA_HOME_EZA, 0 ); // Set homing
    APS_set_axis_param( axis_id, PRA_HOME_SHIFT, 0 ); // Set homing
    APS_set_axis_param( axis_id, PRA_HOME_POS, 0 ); // Set homing
```

```
// 2. Start home move
return_code = APS_home_move( axis_id ); //Start homing
if( return_code != ERR_NoError )
{ /* Error handling */ }

// 3. Wait for home move done,
do{
    Sleep( 100 );

    msts = APS_motion_status( axis_id );// Get motion status
    msts = ( msts >> MTS_NSTP ) & 1; // Get motion done bit
}while( msts == 1 );

// 4. Check home move success or not
msts = APS_motion_status( axis_id ); // Get motion status
msts = ( msts >> MTS_ASTP ) & 1; // Get abnormal stop bit
if( msts == 1 )
{ // Error handling ...
    I32 stop_code;
    APS_get_stop_code( axis_id, &stop_code );
}

}else
{ // Homing success.
}

}
```

本控制器依不同硬體平台，提供數種自動歸零程序 (Auto-home searching)，其中可以任意參考三種不同的機械訊號，如 ORG, EL 及單一 EZ 訊號。以根據不同參考訊號定義三種歸零模式，使用者亦可彈性結合任意三種訊號來規劃適合的歸零程序。此外每一種模式還提供數種參數來完成各種定位的需求。這三種定位模式是依據這些系統配置而設計，已涵蓋一般常見硬體配置。這三種模式分別為：

1. ORG 訊號定位 (Home mode = 0)(Home return by ORG signal)
2. EL 訊號定位 (Home mode = 1)
3. 單一 EZ 訊號定位 (Home mode = 2)

下面小節針對這三種定位方式流程以及相關參數設定來做說明。

4.3.1 ORG 訊號定位 - Home Mode 0

根據不同的起始位置，分成三種狀況來討論：

狀況 A：起始位置介於 MEL 和 ORG 訊號間或處於 MEL 訊號上。

狀況 B：起始位置在 ORG 訊號上

狀況 C：起始位置介於 PEL 和 ORG 訊號間或處於 PEL 訊號上。

下圖表為速度和位置的示意圖，來表示這三種狀況的定位動作，下圖中三個灰色區域由左至右分別代表 MEL 訊號為 "ON" 的區域，ORG ON 的區域及 PEL ON 的區域。一開始以最大速度 VM 往前尋找 ORG，當接觸到 ORG 訊號，減速到完全停止。接著以 VM 速度反方向離開 ORG 訊號。最後重新以低速 VO 尋找 ORG 完成回 Home 程序。

- **相關軸參數 (Axis parameter) 設置**

軸參數	軸參數值	軸參數值敘述
PRA_HOME_MODE	0	採用 Home mode 0 (ORG 訊號定位)
PRA_HOME_DIR	0	採正方向前進定位
PRA_HOME_EZA	0	是否進一步對齊 EZ 訊號, 0: 否, 1: 是
PRA_HOME_S	0	S 曲線因子
PRA_HOME_ACC	ACC	加減速度 (距離單位 / 秒 ²)
PRA_HOME_VS	VS	啟動速度 (距離單位 / 秒)
PRA_HOME_VM	VM	搜尋原點速度 (距離單位 / 秒)
PRA_HOME_VO	VO	定位速度 (距離單位 / 秒)
PRA_HOME_SHIFT	0	最後定位位置與對齊訊號的偏移量 (距離單位 / 脈衝)

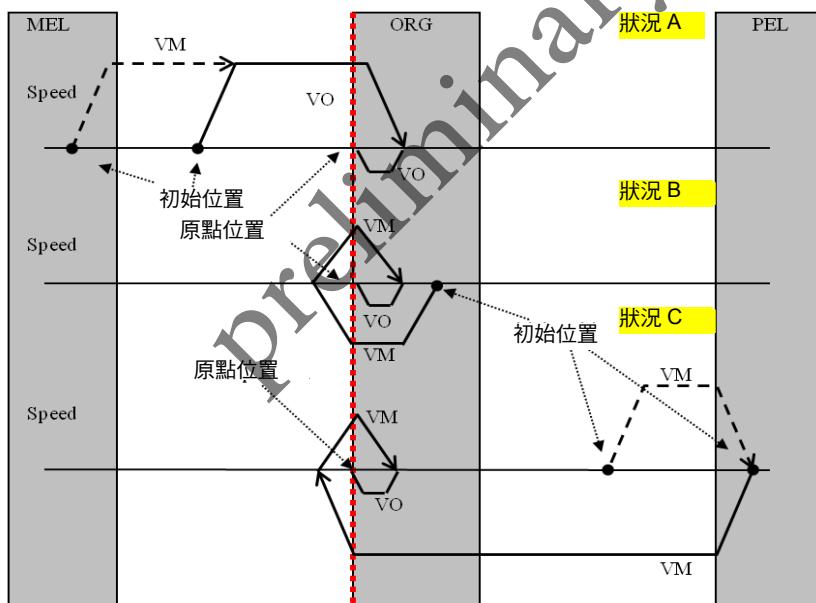


圖 4-8 : Home mode 0 (Case: ORG)

一般機械裝置上的 ORG 訊號有兩個方向邊緣 (訊號擋片 (Fender) 的兩端)，上圖顯示當軸參數中的定位方向參數設定為正方向時 (PRA_HOME_DIR)，控制軸會由正方向 (位置命令遞增的方向) 進行搜尋。最後停在 ORG 訊號的左側邊緣上 (靠近 MEL 機械訊號裝置端)。

反之，軸參數中的定位方向參數設定為負方向時 (PRA_HOME_DIR)，控制軸會由負方向 (位置命令遞減的方向) 進行搜尋。最後停在 ORG 訊號的右側邊緣上 (靠近 PEL 機械訊號裝置端)。下圖顯示當軸參數 "PRA_DIR" 設定為負方向時，定位運動的示意圖：

- **相關軸參數設置**

軸參數	軸參數值	軸參數值敘述
PRA_HOME_MODE	0	用 Home mode 0 (ORG 訊號定位)
PRA_HOME_DIR	1	採負方向前進定位
PRA_HOME_EZA	0	是否進一步對齊 EZ 訊號，0：否，1：是
PRA_HOME_S	0	S 曲線因子
PRA_HOME_ACC	ACC	加減速度 (距離單位 / 秒 ²)
PRA_HOME_VS	VS	啟動速度 (距離單位 / 秒)
PRA_HOME_VM	VM	搜尋原點速度 (距離單位 / 秒)
PRA_HOME_VO	VO	定位速度 (距離單位 / 秒)
PRA_HOME_SHIFT	0	定位位置偏移量 (距離單位 / 脈)

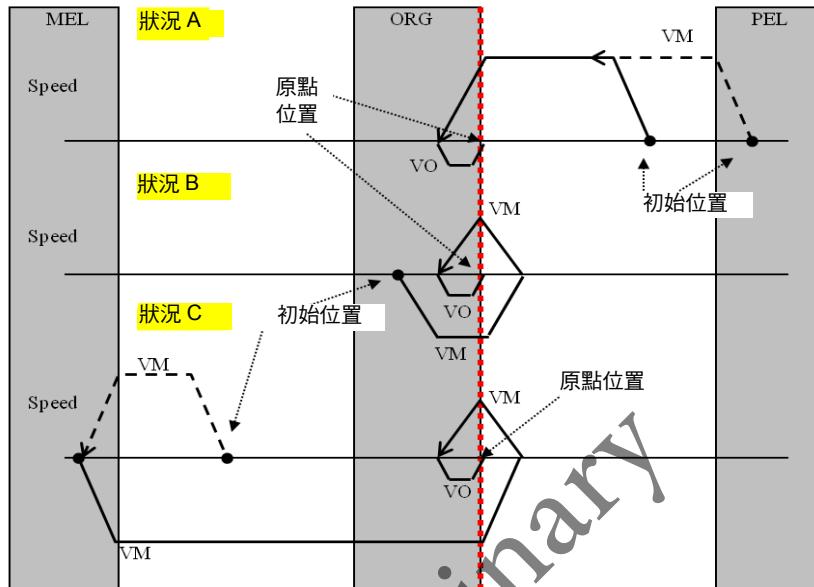


圖 4-9 : Home mode 0 (Case: ORG)

當軸參數 PRA_HOME_EZA 設定為 1 時，表示進行 EZ 對齊，當控制軸找到 ORG 邊緣後，再進一步往定位方向前進，直到第一個 EZ 訊號被偵測到並將控制軸定位至 EZ 邊緣上定位運動始完成。

其動作如下圖：

- **相關軸參數設置**

軸參數	軸參數值	軸參數值敘述
PRA_HOME_MODE	0	Home mode 0
PRA_HOME_DIR	0	採正方向前進定位
PRA_HOME_EZA	1	是否進一步對齊 EZ 訊號, 0 : 否, 1: 是
PRA_HOME_S	0	S 曲線因子
PRA_HOME_ACC	ACC	加減速度 (距離單位 / 秒 ²)
PRA_HOME_VS	VS	啟動速度 (距離單位 / 秒)
PRA_HOME_VM	VM	搜尋原點速度 (距離單位 / 秒)

軸參數	軸參數值	軸參數值敘述
PRA_HOME_VO	VO	定位速度 (距離單位 / 秒)
PRA_HOME_SHIFT	0	定位位置偏移量 (距離單位)

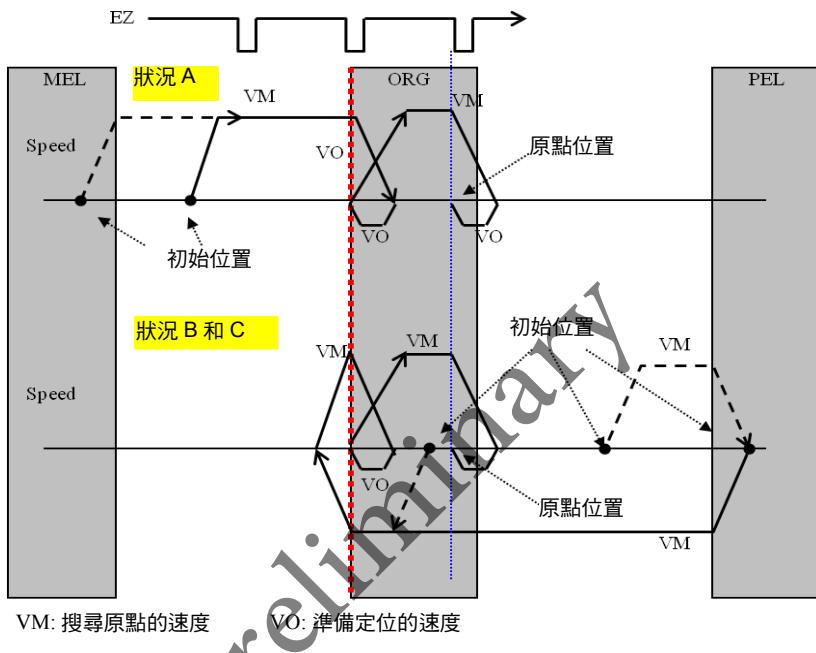


圖 4-10 : Home mode 0 (Case: ORG)

下圖為負方向的例子：

- **相關軸參數設置**

軸參數	軸參數值	軸參數值敘述
PRA_HOME_MODE	0	用 Home mode 0 (ORG 訊號定位)
PRA_HOME_DIR	1	採負方向前進定位
PRA_HOME_EZA	1	是否進一步對齊 EZ 訊號, 0 : 否, 1: 是
PRA_HOME_S	0	S 曲線因子
PRA_HOME_ACC	ACC	加減速度 (距離單位 / 秒 ²)
PRA_HOME_VS	VS	啟動速度 (距離單位 / 秒)
PRA_HOME_VM	VM	搜尋原點速度 (距離單位 / 秒)

軸參數	軸參數值	軸參數值敘述
PRA_HOME_VO	VO	定位速度 (距離單位 / 秒)
PRA_HOME_SHIFT	0	定位位置偏移量 (距離單位)

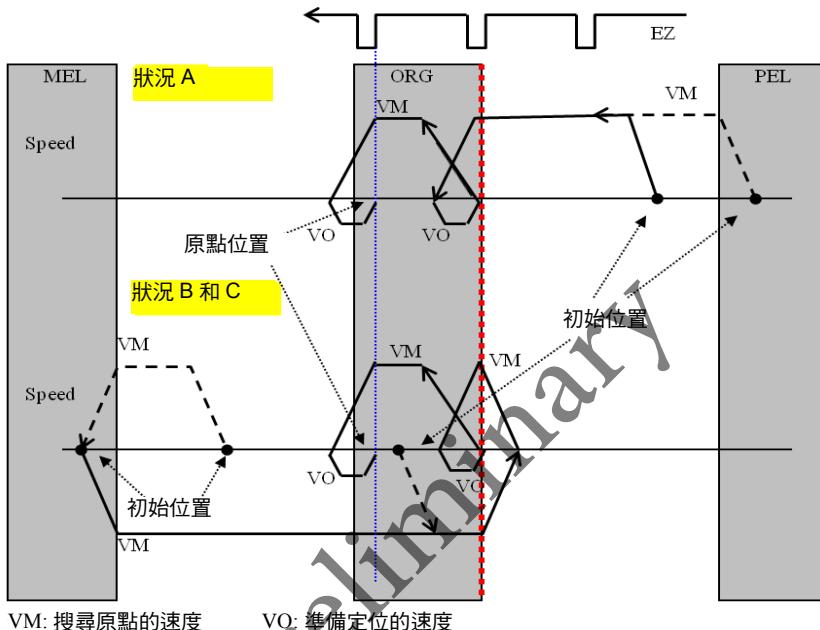


圖 4-11 : Home mode 0 反向 (Case: ORG+EZ)

另外可以設定定位位置偏移量，進行最後位置的微調工作。下面是一個設定的例子和其示意圖：

• 相關軸參數設置

軸參數	軸參數值	軸參數值敘述
PRA_HOME_MODE	0	採用 Home mode 0 (ORG 訊號定位)
PRA_HOME_DIR	0	採正方向前進定位
PRA_HOME_EZA	0	是否進一步對齊 EZ 訊號, 0 : 否, 1: 是
PRA_HOME_S	0	S 曲線因子
PRA_HOME_ACC	ACC	加減速度 (距離單位 / 秒 ²)
PRA_HOME_VS	VS	啟動速度 (距離單位 / 秒)
PRA_HOME_VM	VM	搜尋原點速度 (距離單位 / 秒)
PRA_HOME_VO	VO	定位速度 (距離單位 / 秒)
PRA_HOME_SHIFT	偏移量	定位位置偏移量 (距離單位)

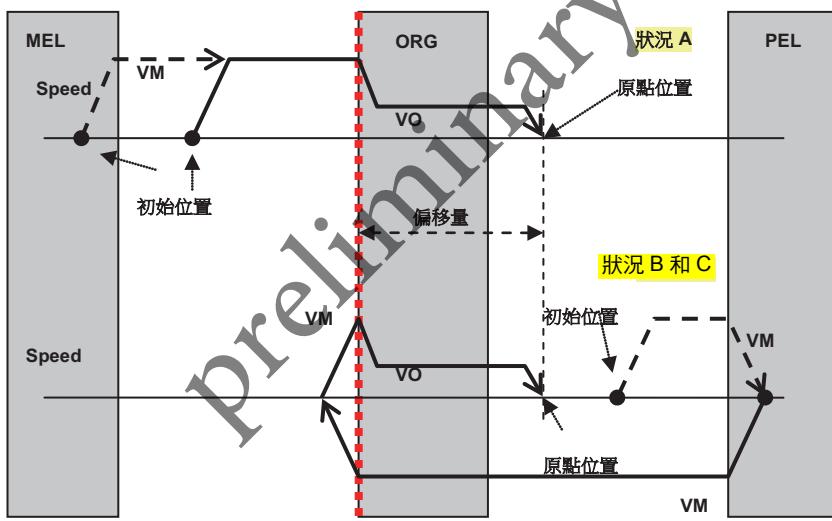


圖 4-12 : Home mode 0 減速停止 (Case: ORG)

4.3.2 EL 訊號定位 - Home Mode 1

這是基於使用 PEL 或 MEL 機械訊號為主的定位運動，當命令下達後，控制軸會搜尋 PEL 或 MEL 訊號的位置，最終停在該訊號邊緣上，另外也可設定是否再進一步對齊 EZ 訊號以及偏移量。

下圖 1，設定 Home mode 1 (EZ 訊號) 定位模式，正方向定位，關閉 EZ 對齊。完成定位運動後控制軸將停在 PEL 訊號邊緣上。

- **相關軸參數設置**

軸參數	軸參數值	軸參數值敘述
PRA_HOME_MODE	1	採用 Home mode 1 (EL 訊號定位)
PRA_HOME_DIR	0	採正方向前進定位
PRA_HOME_EZA	0	是否進一步對齊 EZ 訊號，0：否，1：是
PRA_HOME_S	0	S 曲線因子
PRA_HOME_ACC	ACC	加減速度 (距離單位 / 秒 ²)
PRA_HOME_VS	VS	啟動速度 (距離單位 / 秒)
PRA_HOME_VM	VM	搜尋原點速度 (距離單位 / 秒)
PRA_HOME_VO	VO	定位速度 (距離單位 / 秒)
PRA_HOME_SHIFT	0	定位位置偏移量 (距離單位)

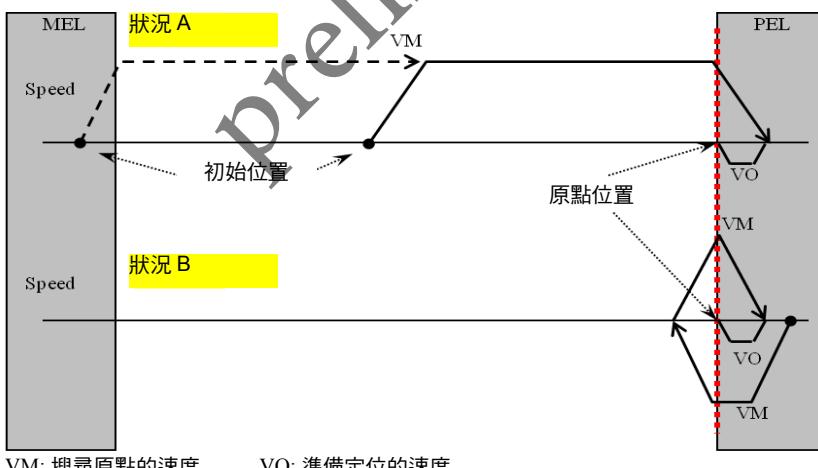
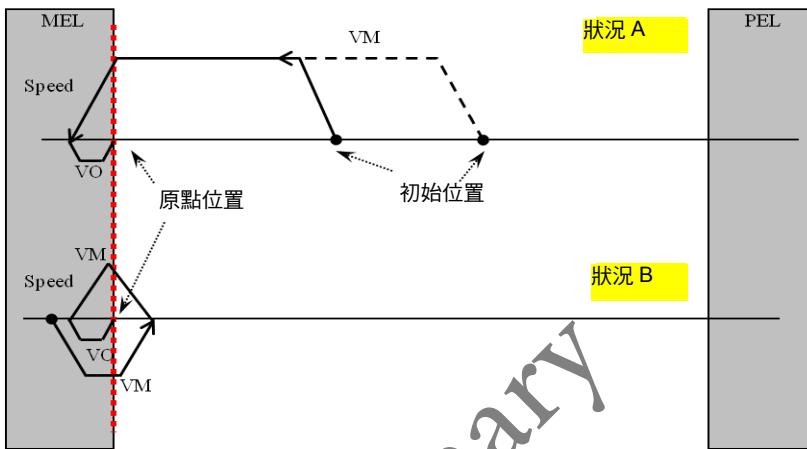


圖 4-13 : Home mode 1 (Case: EL)

EL 定位模式：正方向定位運動，控制軸將停在 PEL 極限邊緣上

若設定 EL 訊號定位模式，負方向定位，關閉 EZ 對齊。完成定位運動後控制軸將停在 MEL 訊號邊緣上如下圖所示。



VM: 搜尋原點的速度

VO: 準備定位的速度

下圖，設定 Home mode 1 (EZ 訊號) 定位模式，正方向定位，設定 EZ 對齊。完成定位運動後控制軸將停在 EZ 訊號邊緣上。

- **相關軸參數設置**

軸參數	軸參數值	軸參數值敘述
PRA_HOME_MODE	1	採用 Home mode 1 (EZ 訊號) 定位
PRA_HOME_DIR	0	採正方向前進定位
PRA_HOME_EZA	1	是否進一步對齊 EZ 訊號，0：否，1：是
PRA_HOME_S	0	S 曲線因子
PRA_HOME_ACC	ACC	加減速度 (距離單位 / 秒 ²)
PRA_HOME_VS	VS	啟動速度 (距離單位 / 秒)
PRA_HOME_VM	VM	搜尋原點速度 (距離單位 / 秒)
PRA_HOME_VO	VO	定位速度 (距離單位 / 秒)
PRA_HOME_SHIFT	0	定位位置偏移量 (距離單位)

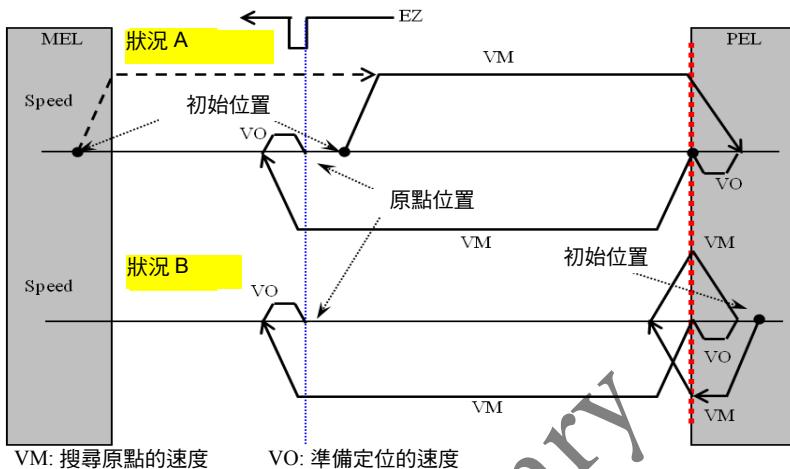
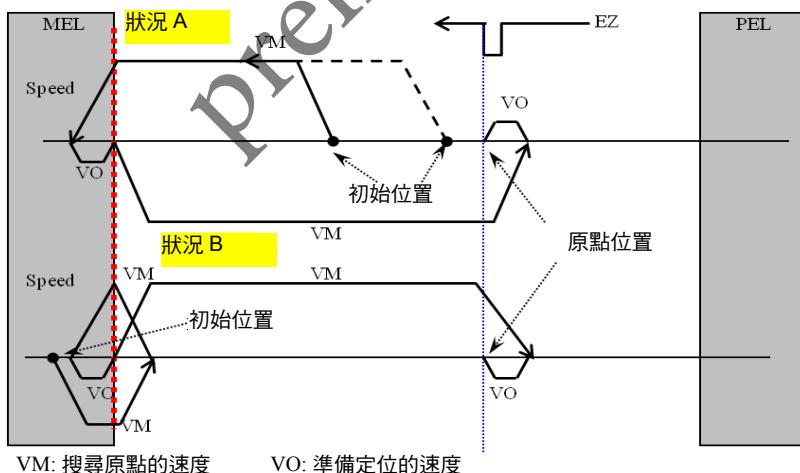


圖 4-14 : Home mode 1 (Case: EL+EZ)

若設定 EL 訊號定位模式，負方向定位，設定 EZ 對齊。則完成定位運動後控制軸也將同樣停在 EZ 訊號邊緣上如下圖所示：



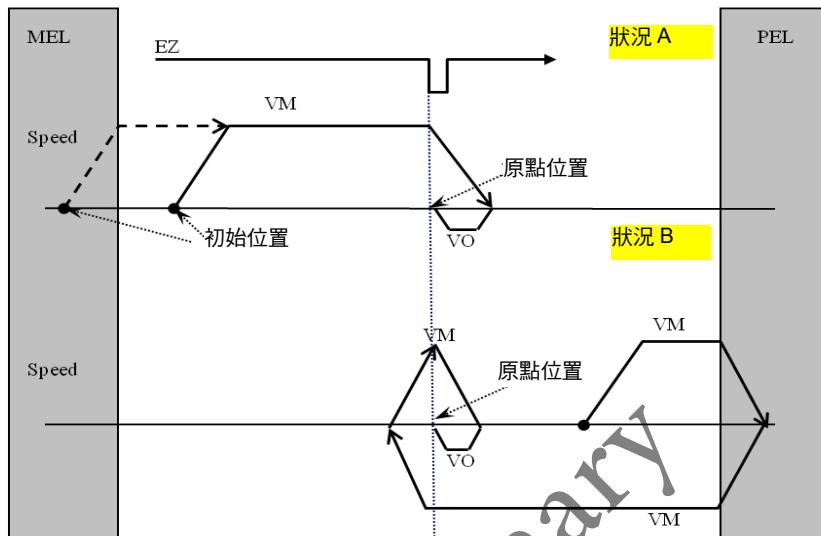
4.3.3 單一 EZ 訊號定位

一般線性馬達機構只會設置一位置標示訊號 (Position mark / EZ mark)，因此，本模式即用在此種機構配置上。

下圖，設定 "Home mode 2 (單一 EZ 訊號)" 定位模式，正方向定位。完成定位運動後控制軸將停在 EZ 訊號邊緣上。

- **相關軸參數設置**

軸參數	軸參數值	軸參數值敘述
PRA_HOME_MODE	2	採用 Home mode 2 (單一 EZ 訊號) 定位
PRA_HOME_DIR	0	採正方向前進定位
PRA_HOME_EZA	0	是否進一步對齊 EZ 訊號，0：否，1：是
PRA_HOME_S	0	S 曲線因子
PRA_HOME_ACC	ACC	加減速度 (距離單位 / 秒 ²)
PRA_HOME_VS	VS	啟動速度 (距離單位 / 秒)
PRA_HOME_VM	VM	搜尋原點速度 (距離單位 / 秒)
PRA_HOME_VO	VO	定位速度 (距離單位 / 秒)
PRA_HOME_SHIFT	0	定位位置偏移量 (距離單位)



VM: 搜尋原點的速度

VO: 準備定位的速度

圖 4-15 : Home mode 2 (Case: EZ)

下圖，設定 "Home mode 2 (單一 EZ 訊號)" 定位模式，負方向定位。完成定位運動後控制軸將停在 EZ 訊號邊緣上。

- **相關軸參數設置**

軸參數	軸參數值	軸參數值敘述
PRA_HOME_MODE	2	採用 Home mode 2 (單一 EZ 訊號) 定位
PRA_HOME_DIR	1	採負方向前進定位
PRA_HOME_EZA	0	是否進一步對齊 EZ 訊號，0 : 否，1: 是
PRA_HOME_S	0	S 曲線因子
PRA_HOME_ACC	ACC	加減速度 (距離單位 / 秒 ²)
PRA_HOME_VS	VS	啟動速度 (距離單位 / 秒)
PRA_HOME_VM	VM	搜尋原點速度 (距離單位 / 秒)
PRA_HOME_VO	VO	定位速度 (距離單位 / 秒)
PRA_HOME_SHIFT	0	定位位置偏移量 (距離單位)

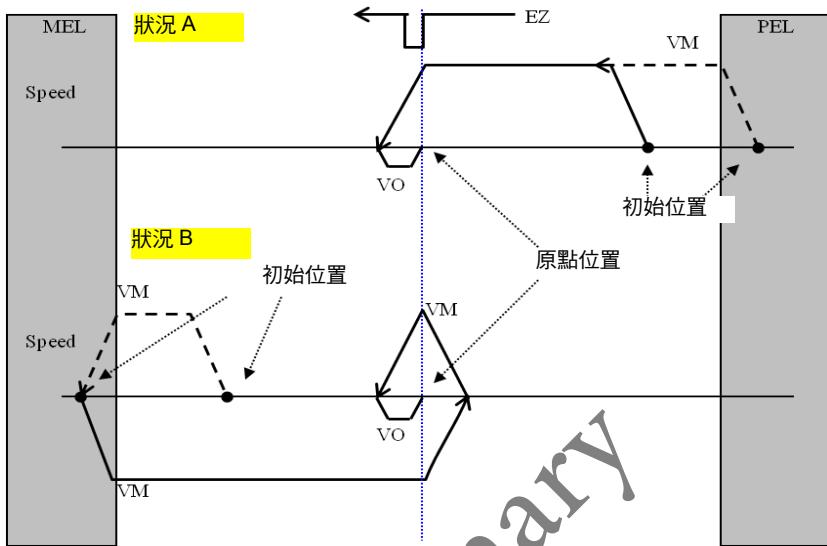


圖 4-16 : Home mode 2 反向 (Case: EZ)



此模式下參數：PRA_HOME_EZA，無作用

4.4 速度運動 (Velocity Move)

此運動模式係指，當指令下達後該運動軸會依照指定的速度曲線運行，運動將會持續下去直到下達停止運動的命令。另外在 Velocity move 支援下述功能：

- 動態改變最大速：運動中可任意改變不同的最大速度。
- 動態下達點對點運動 (PTP) 指令：運動中將速度運動切換成點對點運動移動至特定位置。
- 同步觸發 (亦稱同動)：此運動可設定為觸發啟動，指令下達時，該軸會進入等待觸發訊號，當接收到觸發訊號後才開始運動。另外當多軸處於等待觸發狀態，可以同時發送觸發訊號來達成同時啟動的目的，但須注意是各軸運動是獨立的，結束時間會依照所設定的參數決定。

相關 APS API 如下：

I32 APS_vel(...); // 下達速度運動 (使用 F64 資料格式)
I32 APS_vel_all(...); // 下達速度運動並指定所有速度參數
I32 APS_stop_move(...); // 減速停止
I32 APS_emg_stop(...); // 立即停止
I32 APS_stop_move_multi(...); // 多軸同時下達停止
I32 APS_emg_stop_multi(...); // 多軸同時下達立即停止
I32 APS_velocity_move(...); // 下達速度運動 (使用 I32 資料格式)
I32 APS_move_trigger(...); // 下達同步啟動指令

- **相關的軸參數**

Param. No.	Define symbol	Description
20h (32)	PRA_CURVE	S 曲線因子
21h (33)	PRA_ACC	加速度 (距離單位 / 秒 ²)
23h (34)	PRA_VS	啟動速度 (距離單位 / 秒)
24h (35)	PRA_VM	最大速度 (距離單位 / 秒)

- **例 1:**

設定參數，啟動 Velocity move，其範例流程如下，

1. 2 秒後改變最大速度
2. 2 秒後改變最大速度
3. 2 秒後減速停止

```
#include "APS168.h"
#include "APS_define.h"
#include "ErrorCodesDef.h"

void velocity_move_example()
{
    I32 axis_id = 0;
    F64 speed_1 = 500.0;
    F64 speed_2 = 1000.0;
    F64 speed_3 = 600.0;

    APS_set_axis_param_f( axis_id, PRA_STP_DEC, 10000.0 );
    APS_set_axis_param_f( axis_id, PRA_CURVE, 0.5 ); //Set acceleration rate
    APS_set_axis_param_f( axis_id, PRA_ACC, 10000.0 ); //Set acceleration rate
    APS_set_axis_param_f( axis_id, PRA_DEC, 10000.0 ); //Set deceleration rate

    APS_vel( axis_id, 0, speed_1, 0 ); // Start a velocity move
    Sleep( 2000 ); // Wait 2 second
    APS_vel( axis_id, 0, speed_2, 0 ); // Change speed on the fly
    Sleep( 2000 ); // Wait 2 second
    APS_vel( axis_id, 0, speed_3, 0 ); // Change speed on the fly
    Sleep( 2000 ); // Wait 2 second

    APS_stop_move( axis_id ); // Stop
}
```



運動控制輸入訊號 EMG, ALM, PEL, MEL 等會導致運動終止，
請參考**安全保護**



速度運動模式下，目標位置 (Target position) 會隨時更新，和
命令位置 (Command position) 相同。

preliminary

4.5 JOG 運動 (Jog Move)

Jog 操作，常見於機台上的控制面板，主要目的在手動控制運動軸的移動，亦可與機械開關透過數位輸入訊號 (Digital input) 作結合運用，利用 DI 訊號當作 Jog 運動啟動訊號，因此使用者只要設定好相關參數即可讓機台上的控制面板開關來操作 Jog 運動，無須撰寫控制程式。

Jog 運動模式分為下列兩種：

1. **連續模式 (Continuous mode)**：此種模式其行為類似速度運動模式 (Velocity move)，給定最大速度及加速曲線。當 JOG-ON 訊號 (*) 發生上緣觸發 (Rising edge event)，指定的控制軸開始進行速度運動。當 JOG-ON 訊號發生下緣觸發 (Falling edge event)，則該軸會減速停止。見下圖：

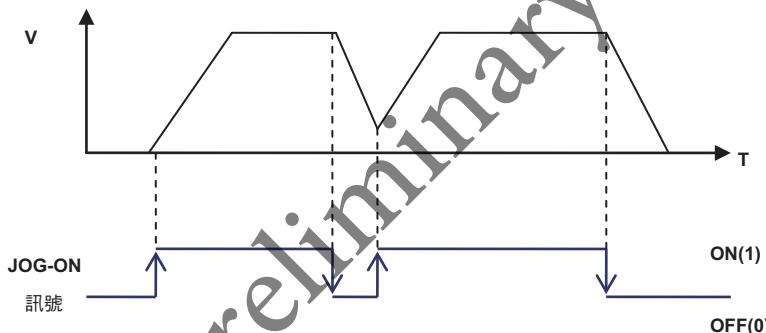


圖 4-17：JOG 運動的 V-T 圖與 JOG-ON 訊號的關係圖



JOG-ON 訊號為一軟體數位訊號，ON 為 1，OFF 為 0

1. **吋進模式 (Step mode)**：此模式除速度參數外，需給定位移量，因此可以容易預測停止位置。當 JOG-ON 控制訊號上緣觸發驅動後，被控制軸會移動給定的位移量 (Offset) 然後停止，暫停一段時間 (或稱延遲時間)，若控制訊號在延遲時間內持續為 ON，則控制軸以設定的速度曲線運動直到控制訊號解除，和連續運動不同之處在於停止時其總位移量 (Total displacement) 會自動計算為設定位移量 (Offset) 的整倍數。因此便於在調機時可以達到較精密的位移量控制。

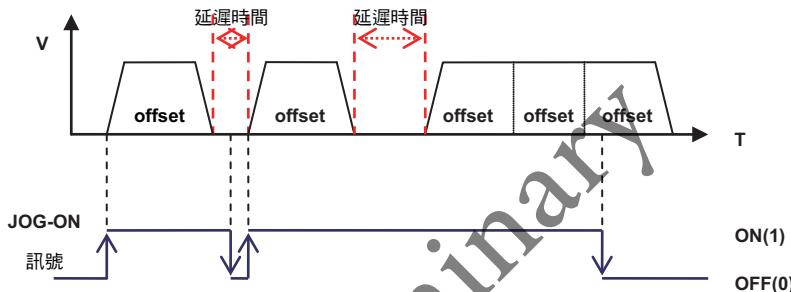


圖 4-18：Jog 吋進模式示意圖

- JOG-ON 與數位輸入 (Digital input) 訊號連動：

其 JOG-ON 控制訊號除了可以使用 API 函式呼叫的方式下達外，也可以將數位輸入訊號設定為控制訊號。設定軸參數 (48h, 49h, 4Ah, 4Bh) 方式分為兩種，以 DI 點數量來區分：

1. 使用兩個 DI 通道，分別將兩個通道設定為正方向運動 JOG-ON 訊號與負方向運動 JOG-ON
2. 使用一個 DI 通道，指定為 JOG-ON 訊號，其方向性由軸參數決定。

相關 APS API 如下：

I32 APS_jog_on (...); // 下達速度運動

- **相關的軸參數**

參數碼	參數定義	參數值意義
40h ()	PRA_JG_MODE	設定 JOG 模式 [0 : Continuous, 1 : Step]
41h ()	PRA_JG_DIR	設定 JOG 方向 [0 : Negative, 1 : Positive direction]
42h ()	PRA_JG_SF	設定 JOG S factor [0 ~ 1]
43h ()	PRA_JG_ACC	設定 JOG 加速率 (Acceleration) [Value > 0]
44h ()	PRA_JG_DEC	設定 JOG 減速率 (Deceleration) [Value > 0]
45h ()	PRA_JG_VM	設定 JOG 最大速 (Max.velocity) [Value > 0]
46h ()	PRA_JG_OFFSET	設定 JOG 偏移位置 . Step 模式使用 [Value >= 0]
47h ()	PRA_JG_DELAY	設定 JOG 延遲時間 , Step 模式使用 [0 ~ 10,000,000] us
48h ()	PRA_JG_MAP_DI_EN	設定 JOG 信號與 DI 信號是否關聯，即其 DI 訊號是否反向
49h ()	PRA_JG_P_JOG_DI	指定某一通道 DI 訊號至正向 JOG 信號
4Ah ()	PRA_JG_N_JOG_DI	指定某一通道 DI 訊號至反向 JOG 信號
4Bh ()	PRA_JG_JOG_DI	指定某一通道 DI 訊號至 JOG 信號，方向設定依照 0x41

- **例：**

```
#include "APS168.h"
#include "APS_define.h"
#include "ErrorDef.h"

void jog_move_example()
{
    //This example shows how jog move work
}
```



1. 運動控制輸入訊號EMG, ALM, PEL, MEL等會導致運動終止。請參考安全保護章節。
2. 連續模式下，目標位置 (Target position) 會隨時更新，和命令位置 (Command position) 相同。
3. 當控制軸為 Jog 運動時，其他運動指令將被禁止以避免誤動作。
4. 若控制軸在執行其他運動時 (如定位運動等)，Jog 命令將會被忽略。

preliminary

4.6 點對點運動 (Point-to-Point Move)

4.6.1 點對點運動 (Point-to-Point Move)

點對點運動 (Point-to-Point move, PTP move) 係指，在預定的速度規劃下控制一軸由 A 位置移動到 B 位置。依照給定的位置參數，可區分為相對位置運動 (Relative move) 和絕對位置運動 (Absolute move)。

速度曲線部分本控制器提供 T-Curve 和 S-Curve 速度曲線及可透過 S-factor 來進行調整，其中包含起始速度 (Start velocity)，最大速度 (Maximum velocity)，結束速度 (End velocity)，加 / 減速率 (Acceleration / Deceleration) 參數可以個別調整，如下圖所示。其細節說明請參考 4.2.3 節 - 加減速曲線。

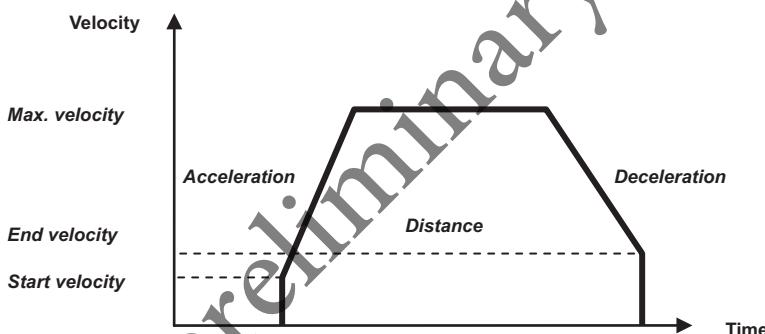


圖 4-19 : T-curve V-T 速度曲線圖

此外，為了偵測點對點運動是否已經結束，我們可以使用 `APS_motion_status()` 將每一軸的運動狀態 (Motion status) 資訊取回，關於運動狀態的描述可以參考第 4.9 節。運動過程中可使用 `APS_stop_move()` 或 `APS_emg_stop()` 來中止運動。

相關的 APS API 如下：

I32 APS_ptp(); // PTP move

I32 APS_ptp_v(); // 帶最大速度參數 PTP move

I32 APS_ptp_all(); // 帶所有速度參數 PTP move

I32 APS_relative_move(); // Relative PTP move (使用 I32 資料格式)

I32 APS_absolute_move(); // Absolute PTP move (使用 I32 資料格式)

I32 APS_stop_move(); // 減速停止

I32 APS_emg_stop(); // 立即停止

- **相關的軸參數**

Param. No.	Define symbol	Description
07h (7)	PRA_SD_DEC	stop_move() 減速率
20h (32)	PRA_SF	S 因子 (S-factor)
21h (33)	PRA_ACC	加速度 (Acceleration rate)
22h (34)	PRA_DEC	減速度 (Deceleration rate)
23h (35)	PRA_VS	啟動速度 (Start velocity)
24h (36)	PRA_VM	最大速度 (Maximum velocity)
25h (37)	PRA_VE	結束速度 (End velocity)

4.6.2 同時啟動 (Synchronous Start)

同步啟動 (亦稱同動)：此運動可設定為觸發啟動，指令下達時，該軸會進入等待觸發訊號，當接收到觸發訊號後才開始運動。另外當多軸處於等待觸發狀態，可以同時發送觸發訊號來達成同時啟動的目的，但須注意是各軸為獨立運動，結束時間會依照所設定的位移量及加速曲線來決定。

4.6.3 動態運動參數切換 (On The Fly Change)

在 PTP 運動過程中可動態改變位置及速度參數，動態改變方式如下：

1. 只動態改變新的位置，速度參數沿用上一次的命令。
2. 只動態改變最大速度，目標位置不變。
3. 動態改變新的位置及速度曲線，意即下達全新的PTP命令。

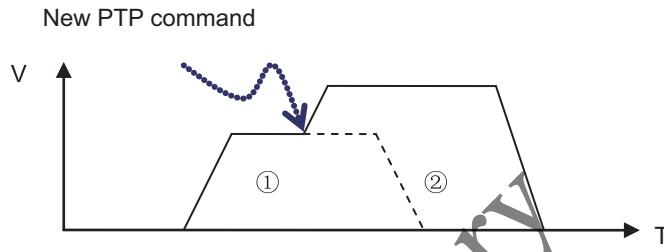


圖 4-20：動態改變位置速度示意圖

4.6.4 連續 PTP 運動 (Continuous PTP Move)

每一軸配有一運動緩衝佇列 (Motion buffer)，其容量為 10 組。當第一個 PTP 命令下達時，該軸立刻啟動運行，運動過程中可連續下達 PTP 命令，後續命令會被存入緩衝之中等待執行。當第一個運動到達指定位置後，控制器自動接續執行緩衝器中的 PTP 命令，依此類推直到沒有新的命令為止。使用者可以指定速度的接續方式，依照速度連接的方式區分為：

1. 緩衝連續 (Buffered)：以佇列為單位速度接續
2. 低速連續 (Blend low)：接續時以最大速較低者相接
3. 高速連續 (Blend high)：接續時以最大速較高者相接
4. 前連續 (Blend previous)：接續時以前者最大速相接
5. 後連續 (Blend next)：接續時以後者的最大速相接

以 3 個連續 PTP 運動為例，依照速度連續的方式設定不同，則 V-T 圖和程式的範例如下：

1. 緩衝連續 (Buffered)

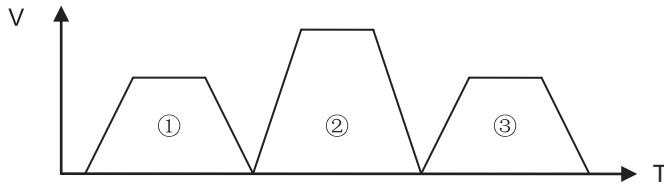


圖 4-21：三段連續位置 V-T 圖

2. 低速連續 (Blend low)：接續時以最大速較低者相接

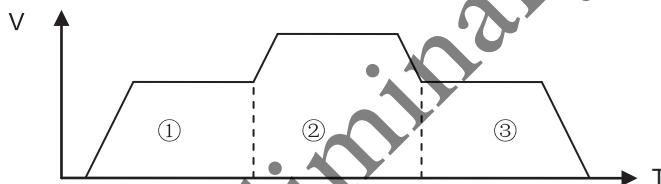


圖 4-22：三段連續位置 V-T 圖 (速度自動連接 (1))

3. 高速連續 (Blend high)：接續時以最大速較高者相接

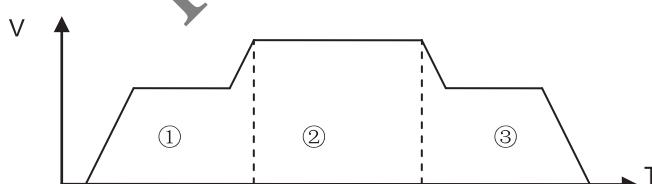


圖 4-23：三段連續位置 V-T 圖 (速度自動連接 (2))

4. 前連續 (Blend previous) : 接續時以前者最大速相接

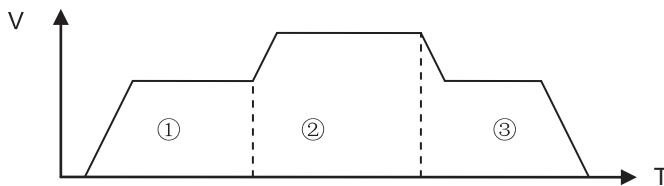


圖 4-24 : 三段連續位置 V-T 圖 (速度自動連接 (3))

5. 後連續 (Blend next): 接續時以後者的最大速相接

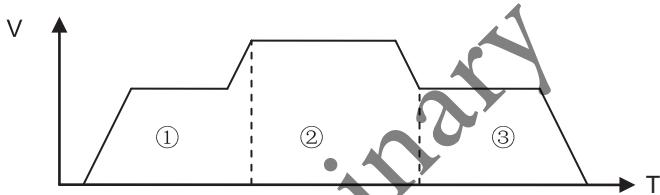


圖 4-25 : 三段連續位置 V-T 圖 (速度自動連接 (4))

4.7 多軸補間運動 (Interpolation)

補間運動係指多軸數的軌跡運動，使用者設定軌跡的特徵如圓心，終點資訊等以及速度資訊，控制器會計算路徑與時間的關係，而執行補間運動的各軸會同時啟動，運動完成後同時停止。

本控制器支援多種補間運動包括：任意 2 到 6 軸直線補間，任意 2 軸的圓弧補間，任意 3 軸的圓弧補間，任意 3 軸螺旋線補間運動。

補間運動通常是需要兩個軸以上，軸號的指定方式通常會採用陣列 (Array) 的方式傳入，在此我們定義輸入軸號陣列 (Axes array) 中，的第一個軸號即為參考軸 (Reference axis)。

參考軸 (Reference axis) 意指，補間運動的速度曲線的設定參考該軸的軸參數 (Axis parameters) 設定，而速度曲線的設定皆為向量的方式 (合成值)。舉例來說，當我們要執行一個 2 軸的直線補間運動，分別為 Axis 1 和 Axis 2，因此我們宣告一個陣列來指定軸號如下例。下例中，axes_array 的第一個軸號為 Axis 1，所以補間運動所要設定的起始速度、最大速、末速和加速度等速度參數皆使用 Axis1 的軸參數，反之，若 Axis 2 陣列中的第一個元素，axis 2 即為參考軸。

```
I32 axes_array[2] = { 1, 2 };
```

本章下面幾個小節分別說明直線、圓弧及螺旋補間的原理及操作方式，最後則會談到連續補間運動 (Continuous interpolation)。

4.7.1 直線補間 (Linear Interpolation)

本控制器支援最高任意六軸的直線補間運動。當直線補間運動下達，相關各軸會同時啟動並依照指定的位置 (絕對或相對)、速度、加速度等加速曲線來進行移動，最後所有相關運動軸也會同時停止，速度曲線設定方式皆採用合成向量。

假設進行 N (N=2~6) 軸的直線補間運動，各軸的位移量分別為 $\Delta X_0, \Delta X_1 \dots \Delta X_{N-1}$ ，則合成位移量 ΔP 為：

$$\Delta P = \sqrt{(\Delta X_0)^2 + (\Delta X_1)^2 + \dots + (\Delta X_{N-1})^2}$$

若設定的合成速度為 V ，則各軸的速度 V_n 為：

$$V_n = \frac{\Delta X_n}{\Delta P} \times V$$

以二維直線補間為例，S 為起點座標，E 為終點座標如下圖：

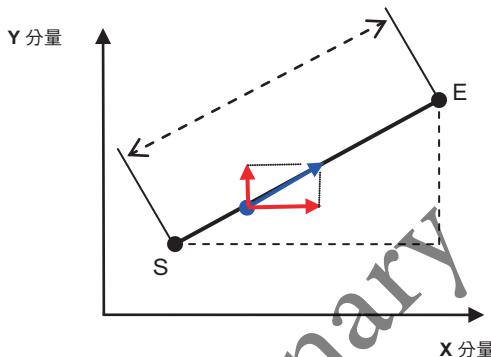


圖 4-26：二維直線補間運動

ΔX 和 ΔY 分別為 X 軸和 Y 軸的位移量，而補間的距離設定方式是依照各軸的分量下去設定(例如相對距離 ΔX 和 ΔY ，或給定終點 E 的絕對座標)，若要計算合成位移量 ΔP ，則使用下列公式：

$$\Delta P = \sqrt{(\Delta X)^2 + (\Delta Y)^2}$$

速度，加速度的設定方式則採用和成向量，若要計算各軸分量，以合成速度 V 為例，各軸的速度分量 V_x ， V_y ：

$$V_x = \frac{\Delta X}{\Delta P} \times V \quad V_y = \frac{\Delta Y}{\Delta P} \times V$$

相關的 APS API 如下：

I32 APS_line(); // 多軸直線補間
I32 APS_line_v(); // 帶有最大速設定的多軸直線補間
I32 APS_line_all(); // 帶有所有速度設定的多軸直線補間
I32 APS_stop_move(); // 減速停止
I32 APS_emg_stop(); // 立即停止
*I32 APS_absolute_linear_move(); // 直線補間給定絕對位置
(使用 I32 資料格式)*
*I32 APS_relative_linear_move(); // 直線補間給定相對位置
(使用 I32 資料格式)*

- **相關的軸參數**

Param. No.	Define symbol	Description
07h (7)	PRA_SD_DEC	stop_move() 減速率
20h (32)	PRA_SF	S 因子 (S-factor)
21h (33)	PRA_ACC	加速度 (Acceleration rate)
22h (34)	PRA_DEC	減速度 (Deceleration rate)
23h (35)	PRA_VS	啟動速度 (Start velocity)
24h (36)	PRA_VM	最大速度 (Maximum velocity)
25h (37)	PRA_VE	結束速度 (End velocity)

4.7.2 圓弧補間 (Arc Interpolation)

本控制器支援二維，三維圓弧補間，並提供多樣化的輸入方式，來應付各種應用的需求。下面分別介紹 2D 及 3D 的圓弧運動及其各種命令建構方式。

4.7.2.1 3D 圓弧補間 (3D Arc Interpolation)

圓弧的描述方式分為

方法一：給定圓心、角度及法向量

方法二：給定圓心、終點

相關指令整理如下：

函式名稱	函式說明
APS_arc3_ca APS_arc3_ca_v APS_arc3_ca_all	執行三維圓弧補間，輸入圓心 (Center)，角度 (Angle) 和法向量 (Normal vector)
APS_arc3_ce APS_arc3_ce_v APS_arc3_ce_all	執行三維圓弧補間，輸入圓心 (Center) 和終點 (End point) 限制：無法執行半圓或全圓

➤ 方法一：給定圓心，角度及法向量，如下圖：

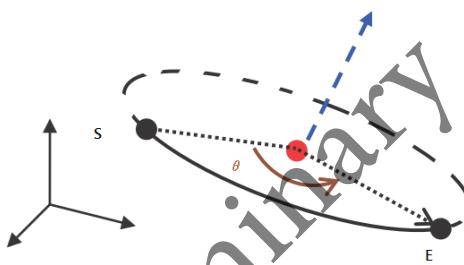


圖 4-27：三維圓弧補間運動（方法一）

此種輸入方式，讓您很容易的建構出一個圓弧路徑，並且沒有半圓或全圓的限制。您必須注意法向量 (Normal vector) 是否正確，若您輸入的法向量有偏差，本控制器預設會自動幫您修正法向量的輸入值。

- **法向量自動校正 (Auto Normal Vector Correct):**

若輸入的法向量和圓心到起點的向量非正交向量 (Not orthogonal vector)，控制器自動校正輸入的法向量，其校正方式採 Gram-Schmidt 正交化法。

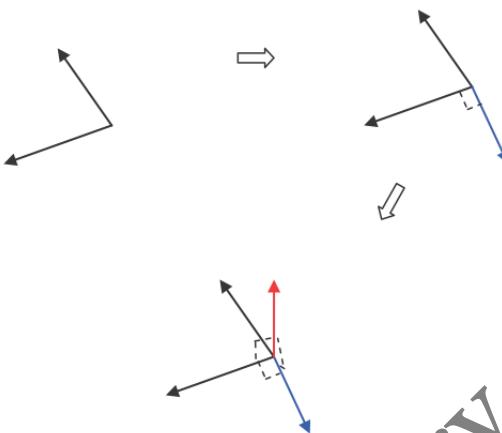


圖 4-28：空間中法向量定義

- 如何決定圓弧方向以及多圈數的路徑

使用右手定則 (Right-hand grip rule)，如下圖，大拇指方向為法向量方向，其他四指為正旋轉的方向。若角度 Angle 輸入負值則選轉方向相反。

若要執行多圈數 (大於 360 度的圓) 可直接設定角度例如 旋轉 2 圈 = 720 度。

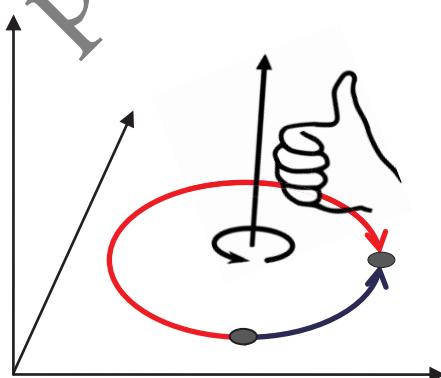


圖 4-29：空間中決定圓弧方向

另外，因限於計算機的計算精度，終點位置座標可能會有些許誤差，若要精確計算至終點位置，則可採用方法二輸入終點位置的輸入方式（下一小節介紹）

➤ 方法二：給定 圓心、終點

此方法您只需輸入圓心和終點的位置，利用此方法來建構圓弧路徑的好處是，不需要求得法向量 (Normal vector)，且可以很精確的掌握最後終點位置來滿足循跡 (Contour) 或需精確定位的應用。採用惟有下列兩點限制：

1. 無法執行角度為 180 度的半圓 (Half circle)
2. 無法執行角度為 360 度的全圓 (Full circle)

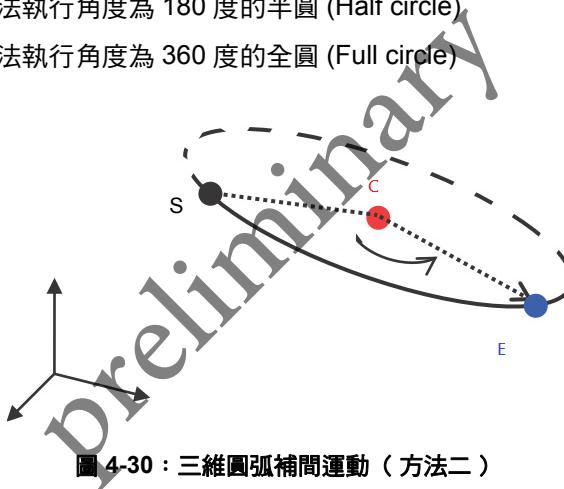


圖 4-30：三維圓弧補間運動（方法二）

• 如何決定圓弧方向

利用右手定則 (Right-hand grip rule)，大拇指為平面的法向量方向，此時四指的方向為正旋轉方向，如下圖所示。

利用輸入參數 "I16 Dir" 來設定，若 $Dir >= 0$ 則為正向旋轉，若 $DIR <= -1$ 為反方向旋轉

• 多圈數（圓弧角度大於 360 度）的路徑

同樣利用輸入參數 "I16 Dir" 來設定旋轉角度 angle，其公式如下

$$\text{Angle} = \theta + \text{Dir} * 2\pi$$

舉例來說若，若 $\theta = 30$ 度，則

Dir	計算方式	Angle (度 Degree)
0	$30 + 0 \times 360$	30
1	$30 + 1 \times 360$	390
2	$30 + 2 \times 360$	750
-1	$30 + (-1) \times 360$	-330
-2	$30 + (-2) \times 360$	-690

• 例：

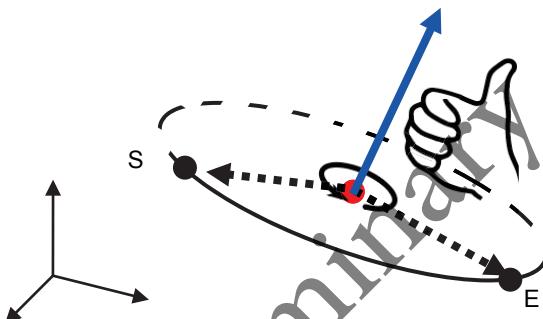


圖 4-31：三維圓弧補間範例圖

4.7.2.2 2D 圓弧補間 (2D Arc Interpolation)

如同 3D 圓弧，2D 圓弧提供下列的描述方式：

方法一：給定圓心，角度

方法二：給定圓心，終點

其設定方式皆與 3D 圓弧相似，細節請參考 3D 圓弧補間

相關的 APS API 如下

```

I32 APS_arc2_ca();
I32 APS_arc2_ca_v();
I32 APS_arc2_ca_all();
I32 APS_arc2_ce();
I32 APS_arc2_ce_v();
I32 APS_arc2_ce_all();

```

4.7.2.3 3D 螺旋插補 (Helical Interpolation)

本控制器支援三維螺旋補間或稱圓錐補間 (Spiral-Helix interpolation)，並提供多樣化的輸入方式，來應付各種應用的需求。建構方式如下：

方法一：給定圓心及角度 (Center-Angle)

方法二：給定圓心及終點 (Center-End)

下面分別介紹這些方法。

➤ 方法一：給定圓心及角度 (Center-Angle)

描述圓錐曲線的參數整理如下表及下圖

參數	說明
圓心 (Center point)	圓心位置 (相對或絕對)
角度 (Angle)	起點和投影在起點所在的圓平面上的終點角度 (如下圖所示)，其正負號代表方向。
法向量 (Normal vector)	起點圓平面的法線向量
高度 (Height)	圓錐高度 (相對高度)
終點半徑 (Final radius)	終點所在的圓半徑

• 例：

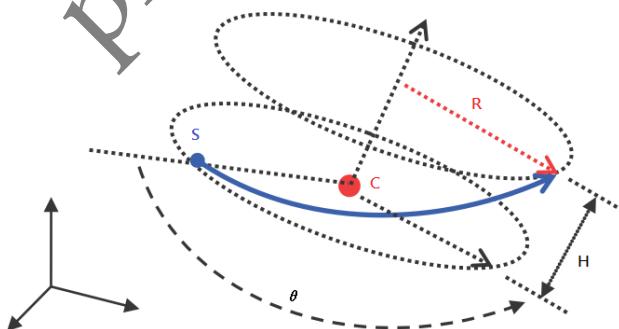


圖 4-32：三維螺旋補間運動（方法一）

➤ 方法二：給定圓心及終點 (Center-End)

描述圓錐曲線的參數整理如下表及下圖

參數	說明
圓心 (Center point)	圓心位置 (相對或絕對)
法向量 (Normal vector)	起點圓平面的法線向量
終點 (End point)	圓錐終點位置 (相對或絕對)
方向 (Direction)	旋轉方向及圈數

方向參數的設定原理和 3D 圓弧相同，細節說明請參考前一章節。

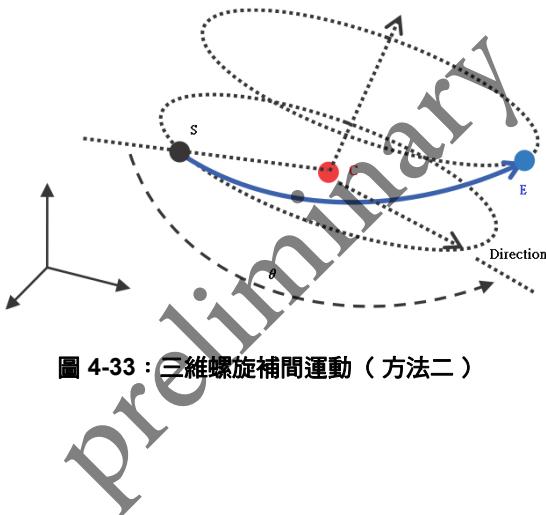


圖 4-33：三維螺旋補間運動（方法二）

上述幾種螺旋補間曲線輸入方式皆需要給定法向量，若法向量輸入有偏差，控制器會自動作修正，其修正方式請參考章節 4.8.2 圓弧補間章節

相關的 APS API 如下：

輸入方式	API	說明
方法一 Center-Angle	I32 APS_sprial_ca();	啟動三維螺旋補間
	I32 APS_sprial_ca_v();	啟動三維螺旋補間 + 最大速參數
	I32 APS_sprial_ca_all();	啟動三維螺旋補間 + 所有速度參數設定
方法二 Center-End	I32 APS_sprial_ce();	啟動三維螺旋補間
	I32 APS_sprial_ce_v();	啟動三維螺旋補間 + 最大速參數
	I32 APS_sprial_ce_all();	啟動三維螺旋補間 + 所有速度參數設定

相關的軸參數：

Param. No.	Define symbol	Description
07h (7)	PRA_SD_DEC	stop_move()，減速率
20h (32)	PRA_SF	S 因子 (S-factor)
21h (33)	PRA_ACC	加速度 (Acceleration rate)
22h (34)	PRA_DEC	減速度 (Deceleration rate)
23h (35)	PRA_VS	啟動速度 (Start velocity)
24h (36)	PRA_VM	最大速度 (Maximum velocity)
25h (37)	PRA_VE	結束速度 (End velocity)

4.7.3 連續補間運動 (Continuous Interpolation)

連續補間係指控制器連續執行多個補間路徑，補間路徑包括前面小節所介紹的，直線，圓弧和螺旋補間。其操作方式是連下執行多個補間指令，下達的補間指令會被存入控制器的緩衝區中等待被執行，如下圖。

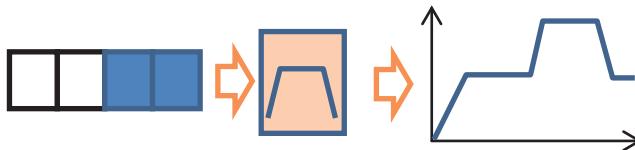


圖 4-34：連續補間運動 (Buffer) 示意圖

連續補間運動只有一個規則就是補間的維度和軸號必須相同，舉例來說 3D 的直線必須和 3D 圓弧相連接，不能和 2D 的圓弧進行連接，其使用的軸號也必須相同。另外針對兩個相鄰的路徑其速度接續的方式，本控制器提供下列七種設定方式：

1. 中止 - 混和 (Aborting and blending)
2. 中止 - 強迫 (Aborting forced)
3. 中止 - 減速 (Aborting stop)
4. 緩衝 (Buffered)
5. 混和 - 減速 (Blending when deceleration start)
6. 混和 - 殘餘距離 (Blending when residue-distance met)
7. 混和 - 殘距比 (Blending when residue-distance % met)

其設定的方式是在輸入參數 "Flag" 中指定，參數說明細節請參考 APS API 使用手冊。

基本上，前面三種方式 (1)、(2)、(3) 在新的補間命令下達時會立刻中止目前正在運行補間運動且執行所下達的補間運動，倘若運動緩衝 (Motion buffer) 中尚有命令未被執行，此時也會被清除，這三種方式在行為上的差別在於其中止的方式，一般被運用在於需要立刻改變目前的補間路徑上。

其他 (4)、(5)、(6)、(7) 的方式，會依緩衝順序執行。第四種方式，特色在於完全依照補間的路徑及速度規劃依序執行，路徑不會產生誤差。

而其中 (5)、(6)、(7) 方法採用的是速度混和的做法，而其好處是軌跡會進行平滑的處理，因此不會有振動問題，且運動速度不會下降，這些方法的不同處在於控制其開始混和的時機點，這關係到混和過程的實際計算路徑和使用者規劃的路徑誤差，使用者可以依照需求選擇調整。下面分別說明這七種速度連續方式：

1. 中止 - 混和 (Aborting and blending)

當 “中止連續” 的補間指令下達後，補間指令會立即生效，以不減速的方式轉移至新的命令。為避免震動 (避免速度不連續) 控制器會自動平滑轉移的軌跡，使每一軸的速度分量維持連續。以下圖為例，第一個線性補間指令由 S1 位置移動至 E1 位置，在途中執行一 “中止連續”的線性補間移動至 E2 位置。左圖為軌跡圖，控制器自動計算平滑軌跡 (紅色線段) 轉換至新的補間指令，右圖為其軌跡合成速度 - 時間 (V-T)。

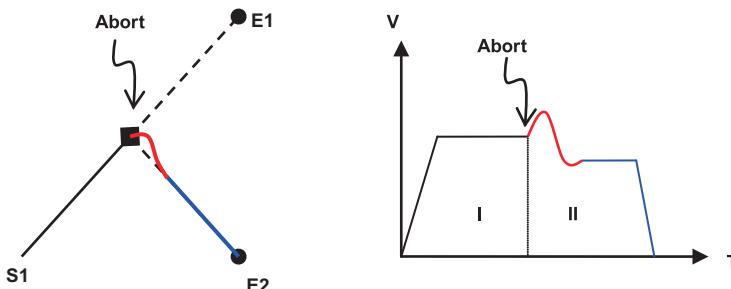


圖 4-35：速度混合 (方法一)

2. 中止 - 強迫 (Aborting forced)

此種指令的特徵是指令下達後，軌跡立即轉換到新指令上，控制器將不會進行平滑處理，也因此運動軌跡會和指令完全吻合。這種模式下，各軸的速度分量可能會有不連續的情況，使用者必須注意轉換時的速度與角度以避免震動產生。

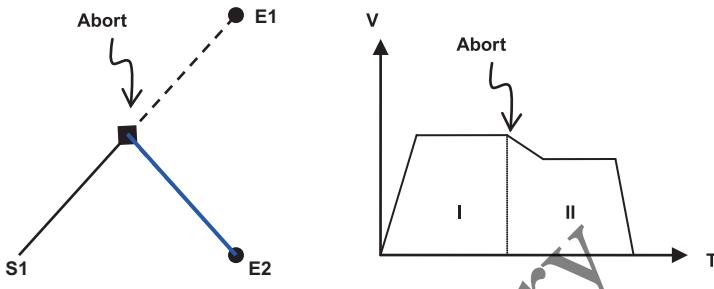


圖 4-36：速度混合 (方法二)

3. 中止 - 減速 (Aborting stop)

指令下達後，原補間指令會先減速停止 (減速率可以設定)，停止後再接續新的補間指令。

以下圖為例，執行一直線補間由 S1 至 E1，途中在 E1' 處下達 “中止 - 減速” 的補間指令，控制器接收指令後以減速設定減速，停止於 S2 然後在移動至 E2 位置。須注意，若採用相對距離的位置給定方式，其相對的起點會以 E1' 為基準而非 S2。

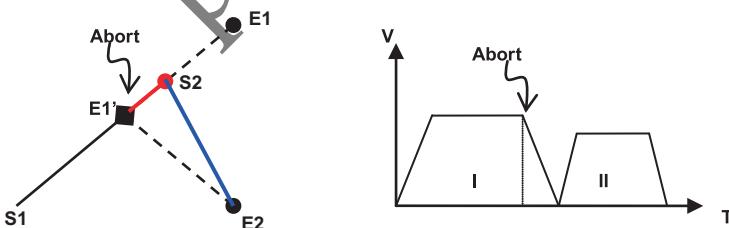


圖 4-37：速度混合 (方法三)

4. 緩衝 (Buffered)

當新指令下達後會先存入運動緩衝 (Motion buffer)，待原補間完成後，接續執行緩衝內的命令。以下圖為例，執行一直線補間由 S1 至 E1，途中下達 “緩衝 (Buffered)” 的補間指令，控制器接收指令後先存入緩衝內，待補間完成後再由 E1 移動至 E2 位置。速度曲線的部分完全依照使用者的設定，若要讓兩補間連接時不減速或稍微減速，可以事先規劃前一段的結束速度與下一段的起始速度。

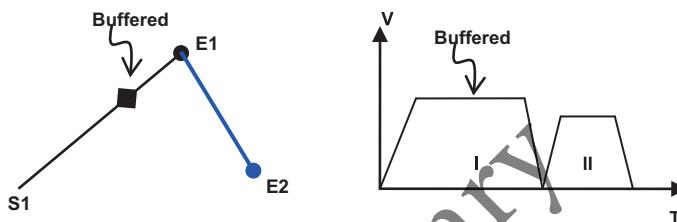


圖 4-38：速度混合 (方法四)

5. 混和 - 減速 (Blending when deceleration start)

當控制器收到新指令後會先將其存入運動緩衝 (Motion buffer)，待原補間指令開始減速時，新的補間指令開始啟動進行擬合，如下圖。可以透過調整減速率來決定開始擬合的時間，減速率越高擬合區域越少，和原補間路徑的誤差也會越小。

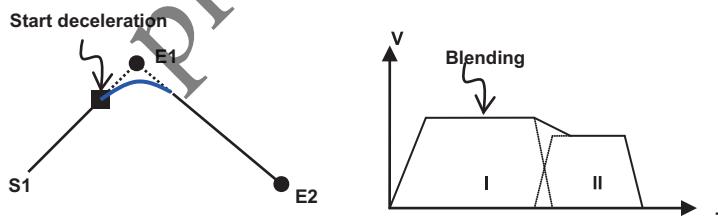


圖 4-39：速度混合 (方法五)

6. 混和-殘餘距離 (Blending when residue-distance met)

當控制器收到新指令後會先將其存入運動緩衝 (Motion buffer)，使用者可以指定一位移量 (如下圖的 Distance，此處稱為殘餘距離)，待原補間路徑距離目標位置小於指定的殘餘距離時 (圖中 E1)，新的補間指令開始啟動進行擬合，如下圖。

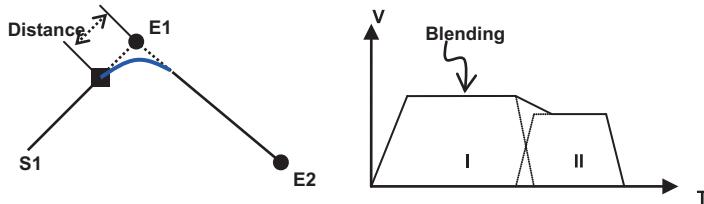


圖 4-40：速度混合 (方法六)

7. 混和-殘距比 (Blending when residue-distance % met)

和方法 6 類似，但給定的方式是殘距比 (殘餘距離佔整體補間距離的百分比)，如下圖中的 P % 。

舉例來說，以下圖為例，若殘距比設定 10%，直線補間的 S1 到 E1 的距離為 1000，則當運動軸移動到 900 時 (相對起點位置)，下一個補間指令開始進行擬合。

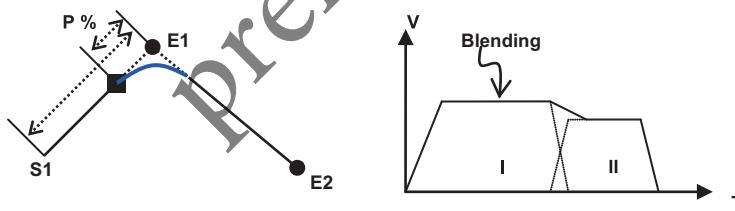


圖 4-41：速度混合 (方法七)

- 例：

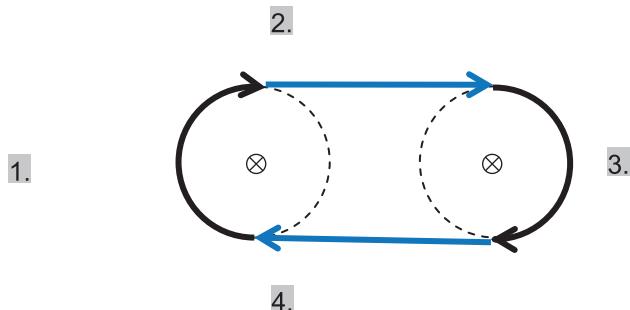


圖 4-42：連續補間運動範例圖

4.8 運動狀態監控 (Motion Status Monitoring)

在進行運動流程控制的過程中，我們常需要對控制軸的運動狀態進行監控，並在適當的時機轉換到下一個流程的控制。舉例來說，當系統初始化時，上位控制程式（使用者的控制程式）的第一個步驟是先對各控制軸進行定位（Homing）的動作，當指令下達後控制器便開始執行定位運動，此時控制程式必須等待定位運動結束，在這裡一般的做法是採用輪詢（Polling）方式偵測定位是否結束，亦即每隔一段時間讀取控制器的運動狀態訊號，當運動完成後才能對控制軸進行下一階段的控制。

另外，運動過程中可能出現例外狀況，上位控制程式也必須能夠檢測出狀況，並針對各種狀況進行例外處理。例如，在定位運動過程中按下緊急停止按鈕（Emergency stop），或是運動過程中觸發極限訊號（End limit signal）等狀況。下圖為定位運動的基本流程圖。

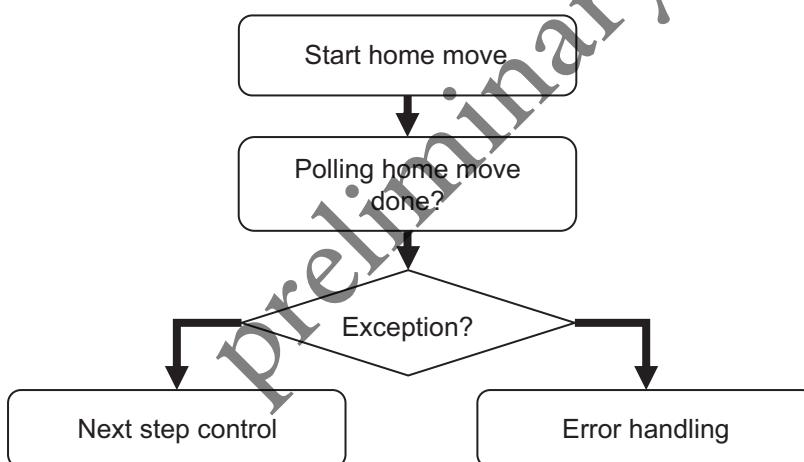


圖 4-43：運動狀態監視流程圖

下面 4.9.1 小節將描述本控制器所提供的運動控制狀態及其行為，在 4.9.2 小節描述錯誤停止代碼功能。

4.8.1 運動狀態 (Motion Status)

讀取各軸的運動狀態使用下列 API 函式：

I32 APS_motion_status();

以軸為單位的運動狀態整合整合在回傳值 I32 變數之內（32 bit 整數），每個位元所代表的運動狀態和其意義如下表：

Bit No.	7	6	5	4	3	2	1	0
Status		HMV	MDN	DIR	DEC	ACC	VM	CSTP
Bit No.	15	14	13	12	11	10	9	8
Status	JOG				PTB	WAIT		
Bit No.	23	22	21	20	19	18	17	16
Status					POSTD	PRED	BLD	ASTP
Bit No.	31	30	29	28	27	26	25	24
Status				GER				

下表簡述運動狀態意義：

Bit No.	Define	Description
0	CSTP	單一運動指令結束
1	VM	在最大速
2	ACC:	加速中
3	DEC:	減速中
4	DIR:	運動方向 . 1: 正方向, 0: 負方向
5	MDN	運動指令結束
6	HMV	執行定位運動中
7~9	--	(保留)
10	WAIT	等待運動觸發中
11	PTB	執行 PTB 運動中
12~14	--	(保留)

Bit No.	Define	Description
15	JOG	Jog 運動中
16	ASTP	不正常停止，執行下個運動後才清除此信號
17	BLD	該軸正在進行擬和運動
18	PRED	前位移事件，執行下個運動後才清除此信號
19	POSTD	後位移事件，執行下個運動後才清除此信號
20~27	--	(保留)
28	GER	齒輪跟隨中 (In geared)，該軸為從軸 (Slave axis)
29~31	--	(保留)

下面依位元順序說明各運動狀態：

Bit0~ bit4:

CSTP : Command stop，當該訊號為 ON 時，表示控制器沒有輸出移動命令。

VM : 當該訊號為 ON 時，表示運動已達最大速度設定。

ACC : 當該訊號為 ON 時，表示運動正在加速中。

DEC : 當該訊號為 ON 時，表示運動正在減速中。

DIR : 當該訊號為 ON 時，表示往正方向運動。若運動停止時，DIR 會保持前停止前一刻的狀態。

其運動與訊號的關係圖如下：

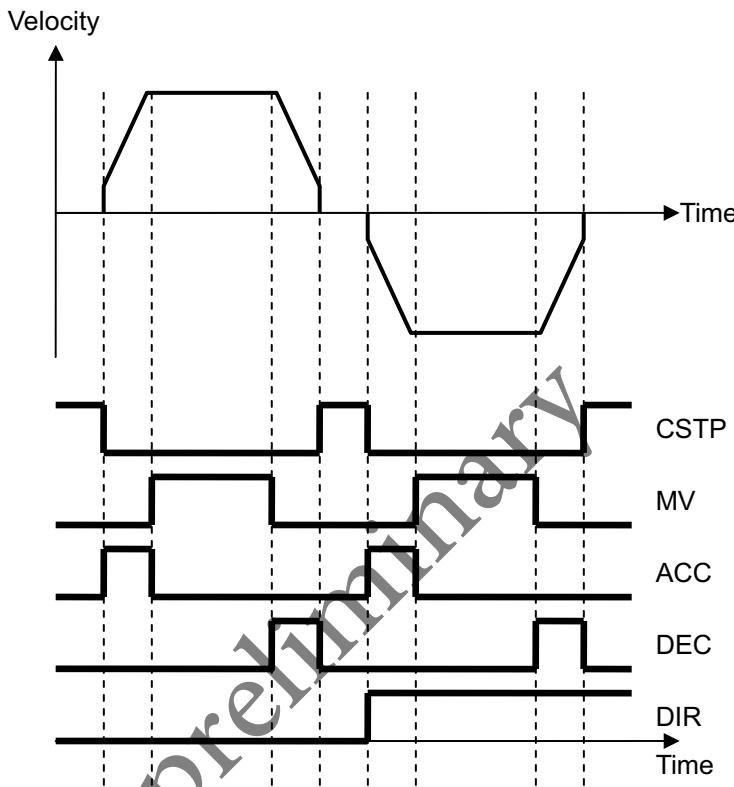


圖 4-44：不同運動訊號與運動關係圖

Bit 5：運動完成 (Motion Done) – MDN

單一運動指令或連續運動指令完成。單一運動指令系指單軸的點對點運動，多軸的點對點運動。連續運動泛指，如定位運動 (Home)，連續運動等一系列運動所組成之連續運動指令。利用此訊號，使用者便很容易地使用輪詢 (Polling) 或是使用中斷事件產生便可運動流程規劃。

注意，當運動異常停止此訊號也會產生，要進一步檢查運動是否有異常停止可以檢查 ASTP 訊號。

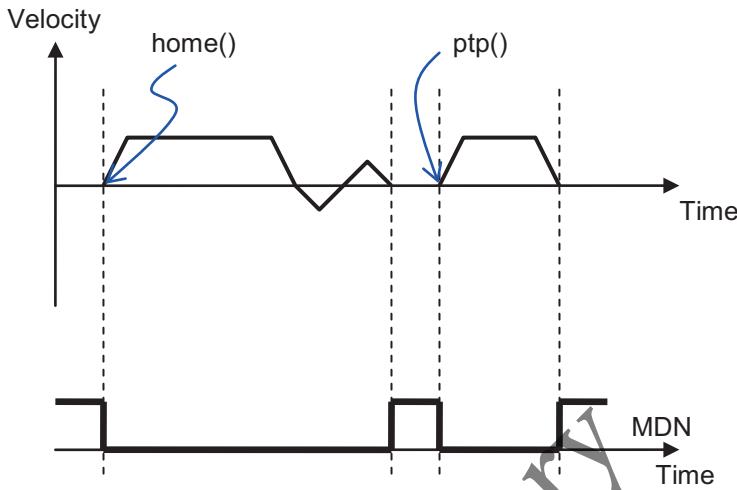


圖 4-45 : Motion Done (MDN) 訊號與運動關係圖

Bit 6 : 定位運動訊號 (In Homing) – HMV

當控制器接收到定位運動指令 `home()` 並開始進行定位運動時 HMV 訊號會被致能 ($= 1$)。當定位運動完成或定位運動被終止時，此訊號會此訊號會關閉 ($= 0$)

詳細的定位運動描述請參考 4.4 節

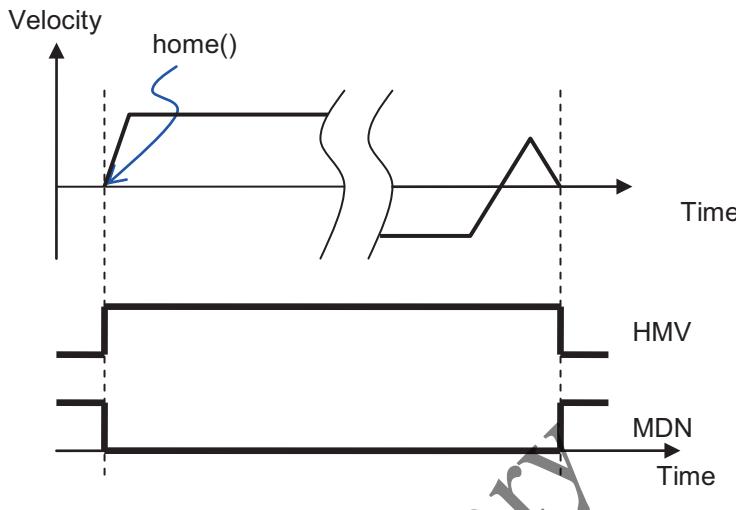


圖 4-46 : Motion Done (MDN), In-homing (HMV) 訊號與運動關係圖

Bit10 : 等待運動觸發 (Wait Move Trigger) – WAIT

當此信號為 ON 時表示該軸正處於等待運動觸發的狀態。當使用者發出觸發訊號：使用 : move_trigger() 函式來觸發正在等待的軸。

下列運動控制函式的輸入參數 Flag 設定為 MF_WAIT (0x00100)時，亦即將該指令設定為觸發啟動。該目標軸不會立即啟動，且 WAIT 訊號變為 ON 。

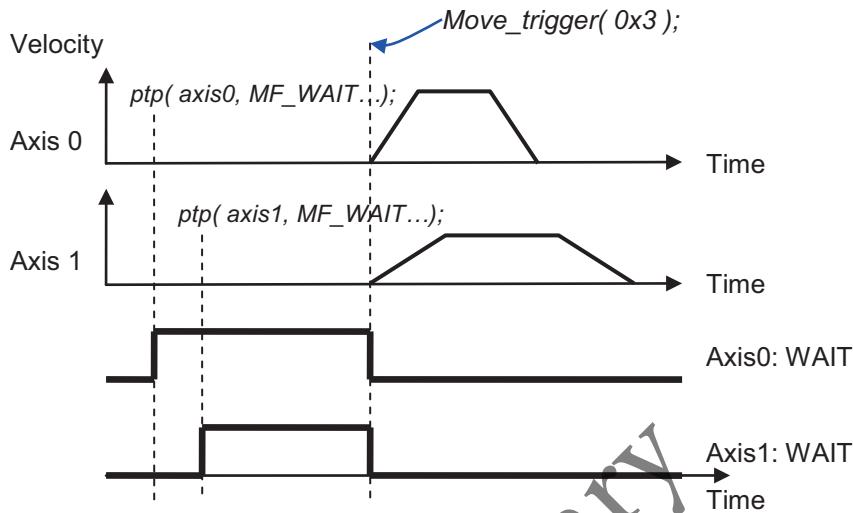


圖 4-47 : WAIT 訊號與運動關係圖

Bit11 : Point Buffer 運動訊號 – PTB

當 Point buffer 運動啟動時，此訊號為 ON，當運動結束時為 OFF。

Bit15 : Jog 運動訊號 – JOG

當該軸正在進行 Jog 運動時，JOG 訊號為 ON，當 Jog 運動結束 JOG 訊號為 OFF，

詳細的 Jog 運動功能請參考 4.6 節，

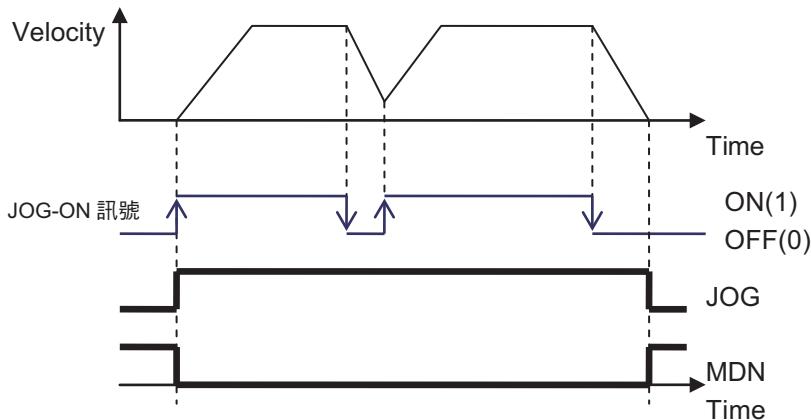


圖 4-48 : JOG, Motion Done (MDN) 訊號與運動關係圖

Bit 16 : 異常停止 (Abnormal stop) – ASTP

當運動因為一些原因而被迫終止，此訊號會產生。異常停止的原因包括如下表所示，使用者可用 `get_stop_code()` 函式取回錯誤停止的代碼 (Stop code)。可利用此代碼進行後續的錯誤處理程序。下小節將說明停止原因。

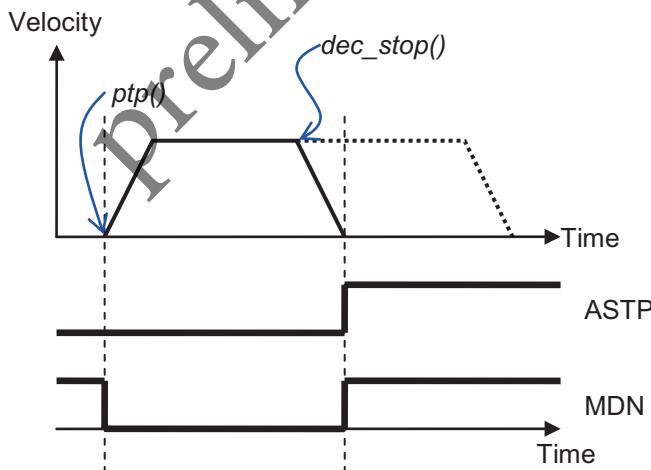


圖 4-49 : 異常停止訊號 (ASTP) 與運動關係圖

Bit17：擬和運動中 – BLD

連續補間運動中有數種速度接續的方式，其中混和接續(Blending)方式在兩路徑交接處有一過渡(Transition)區域(如下圖)，BLD 訊號代表該軸進入此區域。

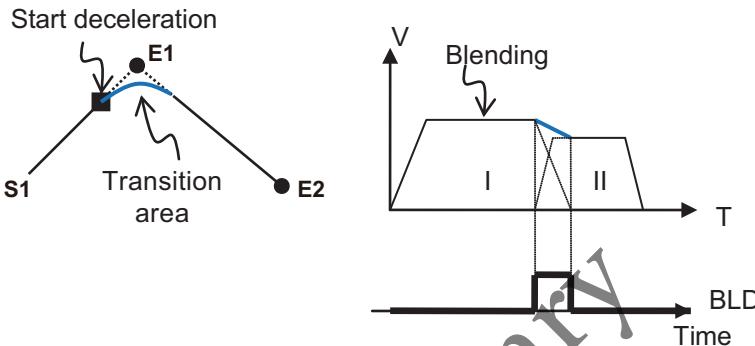


圖 4-50：擬和運動 (BLD) 訊號與運動關係圖

Bit18, Bit19：前 / 後位移量事件 : (Pre/Post-distance Event)

每筆位置運動命令可指定前位移量(Pre-distance)或後位移量(Post-distance)，當移動位移量滿足設定的條件時控制器發出訊號。

前位移量指當點對點運動(Point to point move)開始時控制器會記錄運動的位移量，當位移量大於前位移量時，前位移量事件(Pre-distance event)便產生。相對的，當該點對點運動所剩餘的位移量小於後位移量時，後位移量事件(Post-distance event)便產生。其運動與訊號的關係圖如下：

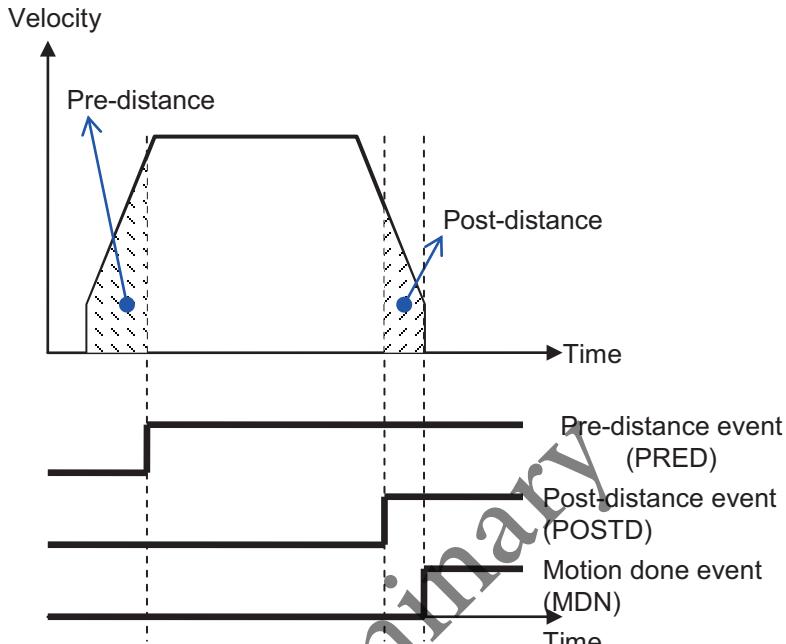


圖 4-51：前 / 後位移量事件訊號與運動關係圖

4.9 應用功能 (Application Functions)

4.9.1 電子齒輪 (Electronic Gearing)

電子齒輪功能係指使用者可以設定某一軸 (從軸 Slave axis) 和另一軸 (主軸 Master axis) 的運動關係，而這個關係類似機械上的齒輪結構，兩個齒輪間的關係一般使用齒輪比 (Gear ratio) 來表示，舉例來說一組齒輪 X 和 Y，其齒輪比為 1:2，當 X(主軸) 旋轉 1 圈時，Y(從軸) 旋轉 2 圈。同理，透過設定電子齒輪比，當主軸進行控制運動時從軸根據使用者設定的齒輪比運動。

本控制器提供 2 種模式分別為標準模式及龍門模式。這兩種模式最主要的差別在於龍門模式是特別針對雙驅動式龍門機構 (Dual drives gantry mechanism) 所設計，這種結構主要使用兩個馬達來驅動一個剛性連接的機構，因此在安全性上以及控制的行為上有特殊的設計。以下兩個小節分別就這兩種模式來說明。

4.9.1.1 標準模式 (Standard Mode)

標準模式下的電子齒輪的設定方式先選定一個從軸 (Slave axis)，所有的參數和命令皆以此從軸為目標做設定，在啟動電子齒輪模式前必須先設定下列軸參數：

Param. No.	Define symbol	Description
60h(96)	PRA_GEAR_MASTER	設定主軸 (Gear master)
61h(97)	PRA_GEAR_ENGAGE_RATE	設定齒輪接合速率 (Engage rate)
62h(98)	PRA_GEAR_RATIO	設定齒輪比 (Gear ratio)

設定完後使用 APS_start_gear(從軸 ID) 來啟動電子齒輪功能。當功能被啟動後，主軸移動多少距離，從軸會依照齒輪比的設定移動。考慮當主軸並非靜止的情況，使用者可以設定適當的齒輪接合速率 (Engage rate) 讓從軸以設定的速率由零開始變化到使用者設定的齒輪比，以避免從軸產生很高的瞬間加速造成震動。另外，此模式下齒輪比也可以動態改變，而改變的過程也依照接合速率轉換。

需增加對 Engage rate 的說明，例如 $Engage\ rate = Gear\ ratio / Engage\ time$

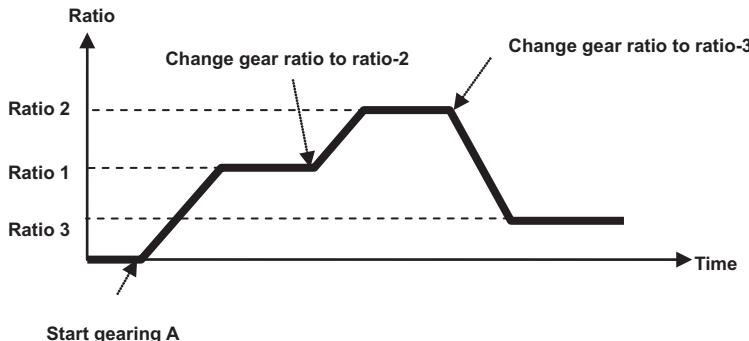


圖 4-52：電子齒輪自動接和速率調整

有幾個狀況會解除標準模式的齒輪關係：

1. 使用者手動解除齒輪關係使用 `APS_start_gear()`
2. 當從軸的 `EMG` / `ALM` / `PEL` / `MEL` / `ALM` 信號為 `ON` 時，主軸若在運行中不會被影響。
3. 從軸收到 `stop()`, `emg_stop()`, `servo_off()` 指令

4.9.1.2 龍門模式 (Gantry Mode)

龍門模式是依照雙驅的龍門機構 (Dual drives gantry mechanism)，所以有以下特點：

1. 除了使用者手動解除齒輪關係外，齒輪關係將不會被解除。
2. 若從軸的 `EMG` / `ALM` / `PEL` / `MEL` / `ALM` 信號為 `ON` 時，主從軸的運動將會停止。
3. 當從軸收到 `stop()`, `emg_stop()`, `servo_off()` 指令時，主從軸的運動將會停止。
4. 不允許變更齒輪比，固定為 1:1
5. 接合速率的設定將被忽略。

另外，此模式下有兩層誤置誤差的保護機制，控制器在每個運動週期檢查兩軸的位置誤差是否超過誤差設定，若超過第一層 (Level 1) 位置誤差設定值時進行減速停止。若超過第二層 (Level 2) 位置誤差設定值時會對兩軸進行 Servo-Off 動作。

其操作方式是對從軸的軸參數設定 1: 選定跟隨的主軸 ,2: 兩階段的位置誤差保護。設定完後使用 APS_start_gear(從軸 ID) 來啟動龍門模式。當模式啟動後只需操作主軸即可，從軸的動作會和主軸完全一致。

Param. No.	Define symbol	Description
60h(96)	PRA_GEAR_MASTER	設定主軸 (Gear master)
63h(99)	PRA_GANTRY_PROTECT_1	Gantry mode 第一階保護
64h(100)	PRA_GANTRY_PROTECT_2	Gantry mode 第二階保護

4.9.2 高速位置比較觸發 (High Speed Position Compare Trigger)

本控制器提供比較觸發 (Compare trigger) 功能供使用者應用，其架構如下圖所示。觸發訊號會經由 TRG0~3 發出，觸發訊號輸出有兩種形式可以設定，分別為脈衝輸出 (Pulse output) 和準位切換輸出 (Level toggle output)，其中脈衝訊號的長度和邏輯也可透過內部 PWMn 模組來調整。PWM 產生訊號來源有兩種，一種是透過手動觸發 (Manual trigger) 方式產生觸發訊號，即使用者呼叫 API 即產生觸發訊號。另一種為比較觸發 (Compare trigger) 方式，而比較觸發又分為線性比較觸發 (Linear compare) 和表格觸發 (Table compare) 兩種。對於 PWM 來說這些訊號來源都可以任意搭配選擇。下面小節分別說明手動觸發和比較觸發。

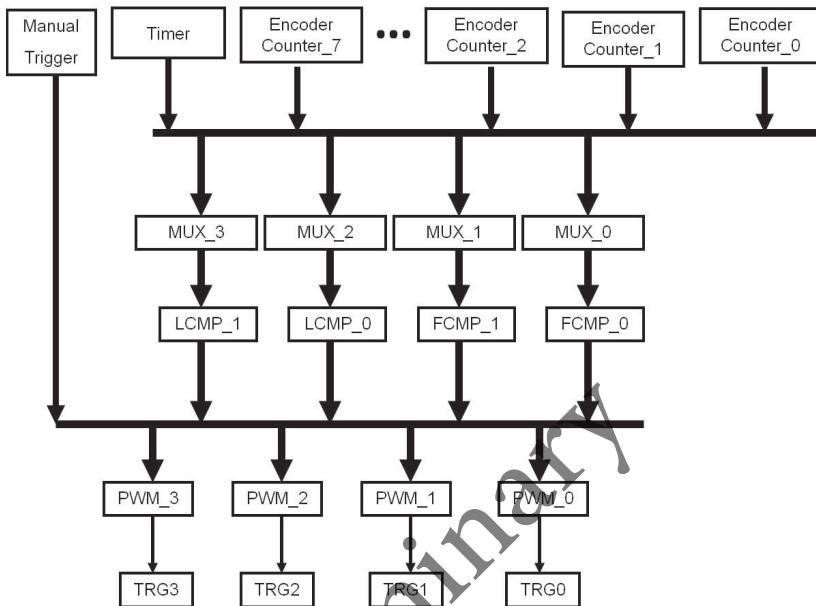


圖 4-53：比較觸發方塊圖

TRG / PWM / Timer 相關參數設定		
NO	Define	Description
0x06	TGR_TRG_EN	TRG0~3 輸出開關
0x10~0x13	TGR_TRGx_SRC	設定 TRG0~3 觸發來源，可以選擇多重來源
0x14~0x17	TGR_TRGx_PWD	設定 TRG0~3 脈衝寬度
0x18~1B	TGR_TRGx_LOGIC	設定 TRG0~3 邏輯準位
0x1C~1F	TGR_TRGx_TGL	設定 TRG0~3 觸發輸出格式
0x20	TIMR_ITV	設定 Timer 間隔
0x21	TIMR_DIR	設定 Timer 計數方向
0x22	TIMR_RING_EN	設定 Timer counter 溢位行為
0x23	TIMR_EN	啟動 Timer

比較觸發詳細的相關參數列表請參考 APS Library 使用手冊。設定參數 APIs 如下

APS_set_trigger_param();

APS_get_trigger_param();

比較器的來源除了編碼計數器 (Encoder counter) 外也可選擇使用內部的 Timer，其相關的 API 如下：

APS_get_timer_counter(); // 讀取 timer counter

APS_set_timer_counter(); // 設定 timer counter

4.9.2.1 手動觸發 (Manual Trigger)

使用 APS_set_trigger_manual() API 即可對輸出脈波訊號，使用前先設定 TRG 的觸發來源有手動觸發。以下是一個操作範例：

NO	Define	Description
0x10~0x13	TGR_TRGx_SRC	設定 TRG0~3 觸發來源，可以選擇多重來源

4.9.2.2 比較觸發 (Compare Trigger)

比較觸發係指當比較器 (Comparator, CMP) 來源值與要比較值相同時，則會產生觸發訊號。比較器來源可分為兩種，第一種是各軸的位置計數器 (Encoder / Counter_0~7)，另一個則為時間計數器 (Timer)。比較方式也可分為兩種形式，第一種為線性比較觸發 (Linear compare)，另一種則稱為表格比較觸發 (Table compare trigger)，下兩小節分別說明運作原理及操作方式。

4.9.2.2.1 線性比較觸發 (Linear Compare Trigger)

使用者在使用線性比較器時首先需要決定比較對象：位置計數器或是時間計數器。接著要設定開始點 (Start point)、總點數 (Repeat times) 以及點間隔 (Interval)。我們以下面位置 - 時間圖來說明如何設定，其中 P1 為比較開始點，總點數設為 4，點間隔為 L，因此可得 4 個彼此間隔為 L 的比較點 P1~P4。當馬達位置經過比較點時，TRG 就會依序發出脈衝訊號。比較方向性可由點間隔的正負決定。線性比較觸發的比較速率高達 1MHz，可比較的點數高達 32 bit 整數值。

使用下列 API 設定線性比教的起始點，總點數和間隔

APS_set_trigger_linear();

線性比較觸發相關參數		
NO	Define	Description
0x00	TGR_LCMP0_SRC	線性比較器 (LCMP0) 比較來源
0x01	TGR_LCMP1_SRC	線性比較器 (LCMP1) 比較來源

• 例：

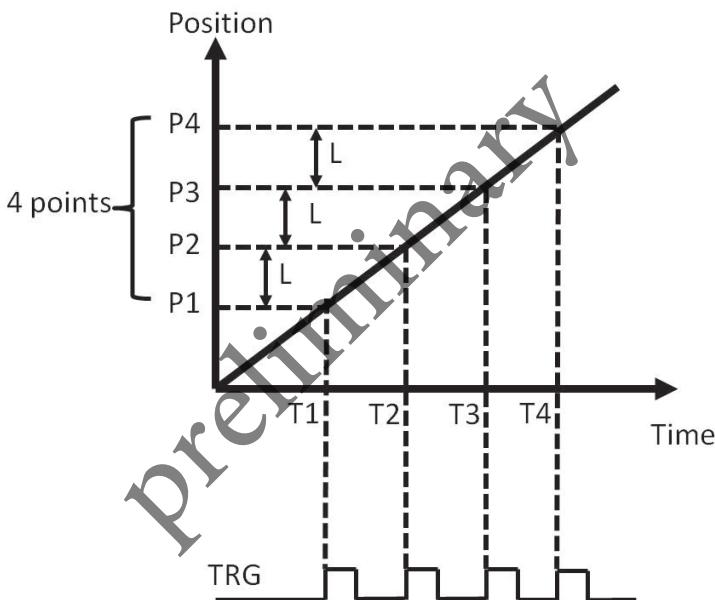


圖 4-54：線性比較觸發範例圖

4.9.2.2.2 表格比較觸發 (Table Compare Trigger)

和線性比較觸發不同的是表格比較觸發可讓使用者自由決定比較點，換句話說，也就是每個比較點的間隔可以是變動的。由下圖所示，使用者任意填入 P1~P4 的點，當馬達位置到達時，則發出觸發訊號。

表格比較觸發相關參數		
0x02	TGR_TCMP0_SRC	設定 Table comparator CH0 比較對象
0x03	TGR_TCMP1_SRC	設定 Table comparator CH1 比較對象
0x04	TGR_TCMP0_DIR	設定 Table comparator CH0 比較方向
0x05	TGR_TCMP1_DIR	設定 Table comparator CH1 比較方向

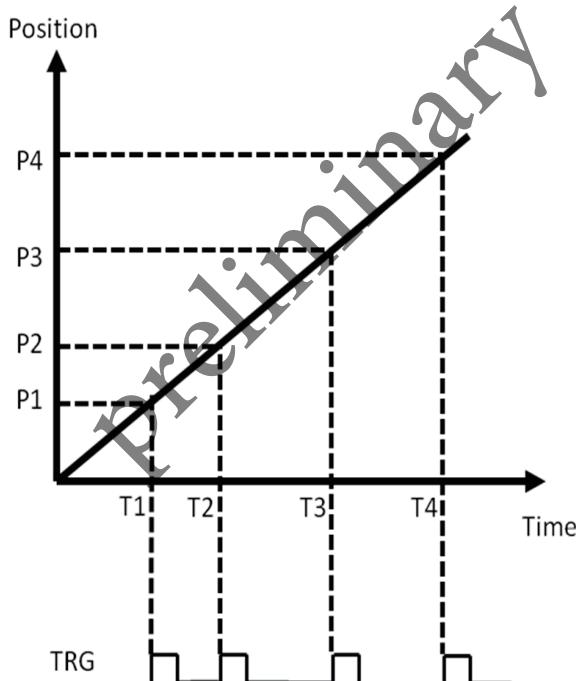


圖 4-55：表格比較觸發範例圖

控制器和硬體內部有兩層的 FIFO 緩衝設計以加速比較速率，硬體的 FIFO 大小為 255 筆，其比較速率高達 1 MHz，控制器上處理器會設定 999 筆 FIFO 緩衝，控制器在每運動控制週期執行填點工作。另外在 APS 函數庫中，使用者只須傳入任意大小的位置陣列 (視系統記憶體大小限制)APS 函式庫將動態載入所有比較位置到控制器中，因此若點數較多時使用者也無須額外撰寫程式來將比較點動態填入控制器中。

將比較表格陣列傳入的 API:

`APS_set_trigger_table();`

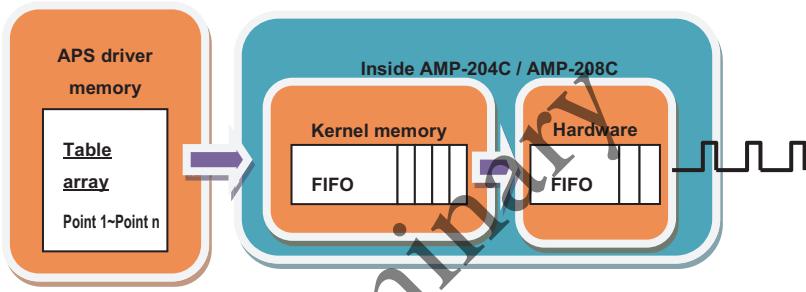


圖 4-56：表格比較觸發方塊圖

4.9.3 PWM 控制 (雷射控制) (VAO Table Control)

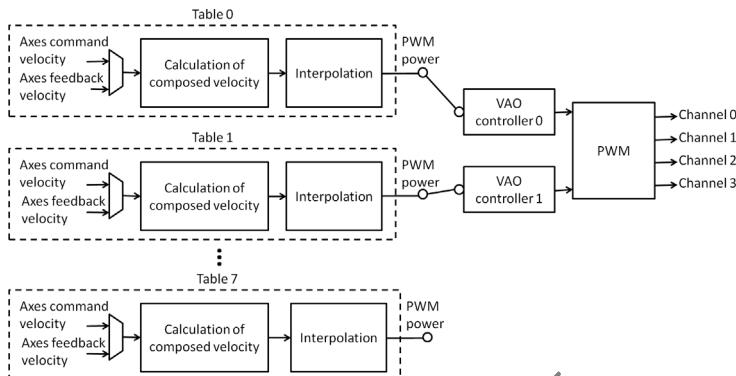
4.9.3.1 整體架構介紹

近年來，雷射切割應用已被廣泛使用於各種加工，例如金屬、非金屬或複合材料等。此種應用與運動控制之間存在著高度相關性，因此，ADLINK DSP-based 運動控制卡為了滿足此種需求而發展了 VAO 模組。VAO 模組功能在於透過速度資訊控制雷射功率以達到高品質的切割效果。一般而言，常見的雷射功率控制方式是透過改變 PWM(Pulse-width modulation) 來達成，因此 ADLINK DSP-based 運動控制卡也擁有不同形式 PWM 控制方式以提供使用者應用在不同場合。

下圖說明了 VAO 模組的運作流程。VAO 控制器 (VAO controller) 負責項目為在不同速度下根據速度功率表計算結果控制 PWM 輸出。AMP-204C / AMP-208C 內有兩組 VAO 控制器且可依實際狀況指定不同輸出通道 (channel)，因此可達成 (1) 同時多通道輸出相同 PWM 或是 (2) 不同通道輸出不同 PWM 等不同需求。每個 VAO 控制器可任意切換不同速度功率表因此可滿足多層切割的要求。AMP-204C / AMP-208C 內有八組速度功率表 (Table 0~7)，其會根據不同軸之間的合成速度透過內插方式計算出相對應的 PWM 設定值以控制雷射輸出功率。不同軸的速度來源可以是運動命令速度 (Command velocity) 或是回授速度 (Feedback velocity)，其中回授速度可使用內建濾波器 (filter) 將雜訊消除。目前回授速度的更新速度為 1KHz。VAO 模組可以搭配點表運動 (Point table) 使用。

使用者在使用 VAO 模組之前，必須設定好 PWM 控制模式與速度與對應功率表，而函式 (function) 的使用必須搭配 VAO 參數表 (VAO parameter table) 以設定參數。相關說明和設定流程會在後續部分依序介紹。

VAO 模組架構圖



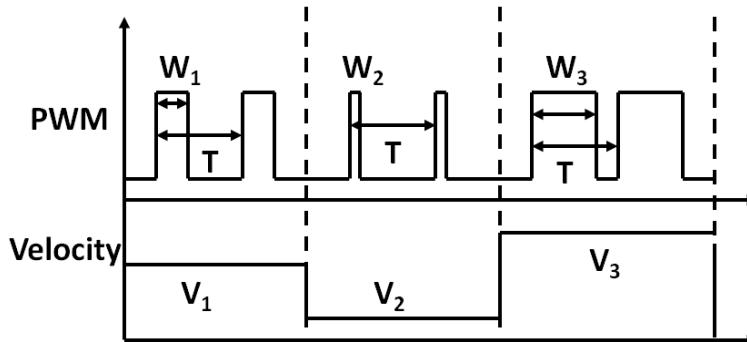
4.9.3.2 控制模式

目前AMP-204C / AMP-208C VAO模組支援的控制模式有以下三種:

a. Mode1: PWM mode

此種控制模式為在固定 PWM 頻率 (frequency) 下根據不同速度調整 PWM 工作週期 (duty cycle)，如下圖所示，固定輸出 PWM 頻率為 $1/T$ ，在 V_1 、 V_2 及 V_3 三種不同速度下，PWM 輸出會根據速度功率表計算結果調整 PWM 工作週期 W_1/T 、 W_2/T 及 W_3/T ，其中 PWM 脍寬 (width) W_1 、 W_2 與 W_3 的寬度皆不同。有關速度功率表部分在接下來章節會做介紹。使用者想使用此種控制模式時其流程為

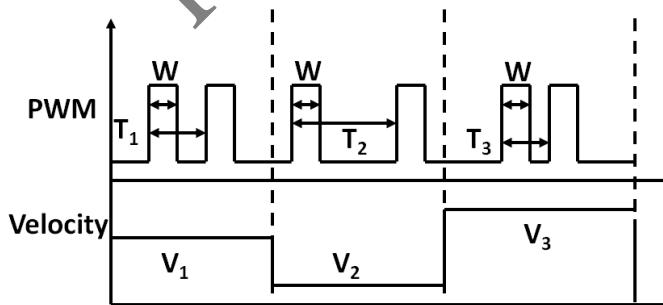
1. 設定控制模式。使用 `APS_set_vao_param()` 設定 `VAO_TABLE_OUTPUT_TYPE` 參數的值為 0x1。
2. 設定 PWM 固定輸出頻率。使用 `APS_set_vao_param()` 設定 `VAO_TABLE_PWM_Config` 參數，在此種模式下參數的單位為 Hz。目前支援的頻率輸入範圍為 3Hz ~ 50MHz。



b. Mode 2: PWM frequency mode with fixed width

此種控制模式為在固定 PWM 脍寬 (width) 下根據速度調整 PWM 頻率。如下圖所示，在 V_1 、 V_2 及 V_3 三種不同速度下，PWM 脍寬固定為 W ，速度功率表會隨著速度改變而產生不同 PWM 頻率 $1/T_1$ 、 $1/T_2$ 與 $1/T_3$ 。使用此種模式的流程為

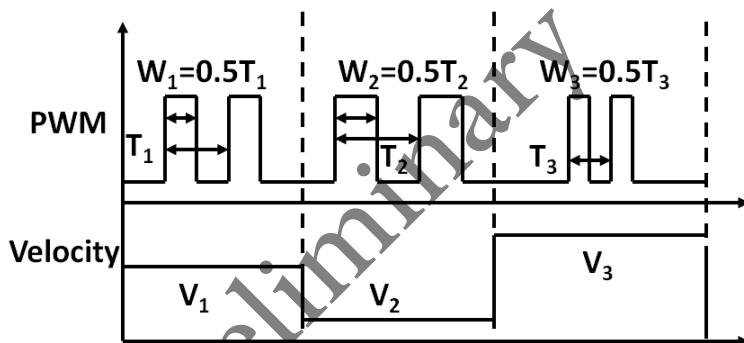
1. 設定控制模式。使用 `APS_set_vao_param()` 設定 `VAO_TABLE_OUTPUT_TYPE` 參數為 0x2。
2. 設定 PWM 固定輸出脈寬。使用 `APS_set_vao_param()` 設定 `VAO_TABLE_PWM_Config` 參數，在此種模式下參數的單位為 ns。而目前支援的輸入範圍是 20ns ~ 335544320 ns。



c. Mode 3: PWM frequency mode with fixed duty cycle

此種控制模式為在固定 PWM 工作週期 (duty cycle) 下根據速度調整 PWM 頻率。如下圖所示，在不同速度下其得到的工作週期 W_1/T_1 、 W_2/T_2 及 W_3/T_3 都會相同的，但是對應的頻率和脈寬則是根據速度功率表調變的。

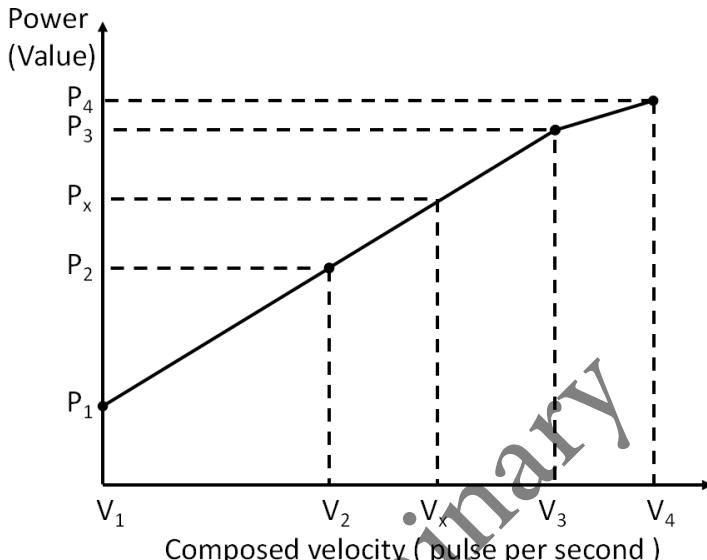
1. 設定控制模式。使用 `APS_set_vao_param()` 設定 `VAO_TABLE_OUTPUT_TYPE` 參數為 0x3。
2. 設定 PWM 固定工作週期。使用 `APS_set_vao_param()` 設定 `VAO_TABLE_PWM_Config` 參數，在此種模式下參數的單位為 %。而目前支援可輸入範圍為 0.05% ~ 100%。



4.9.3.3 速度與對應功率表

前面提到了速度與對應功率表的目的在於根據合成速度計算出相對應 PWM 功率 (power) 給 VAO 控制器以調整實際 PWM 輸出訊號。VAO 模組提供了八組速度與對應功率表給使用者以達成分層切割的目的，每個表可填入 32 組速度與對應的功率。以下將說明如何透過速度資訊計算相對應的功率。以下圖為例，橫軸為各軸合成速度，而縱軸則為相對應的 PWM 功率 (Power)。須注意的是，功率的單位會隨著控制模式不同，舉例來說，當使用者設定控制模式為 Mode1: PWM mode 時，此時對應功率為 PWM 工作週期 (duty cycle)，若是控制模式為 Mode 2: PWM frequency mode with fixed width，則對應功率為 PWM 頻率。假設已經得到各軸合成速度 V_X ，則對應功率 P_X 可由內插方法得知為

$$P_X = (P_3 - P_2) * (V_X - V_2) / (V_3 - V_2) + P_2$$



下列表格列出了在不同控制模式下速度與對應功率表所能設定功率的範圍和解析度。

Mode	Power output range	Resolution
1: PWM mode	Duty cycle: 0~2000 (0.05%~100%)	0.05%
2: PWM frequency mode with fixed width	Frequency: 3Hz ~ 50MHz	1 Hz
3: PWM frequency mode with fixed duty cycle	Frequency: 3Hz ~ 50MHz	1 Hz

4.9.3.4 輸出設定

目前 VAO 模組支援 4 通道 PWM 輸出，使用者可以任意決定 VAO 控制器的輸出通道，也可以讓多通道同時輸出相同控制訊號。目前則是透過函式 `APS_start_vao()` 開啟相對應 VAO 模組。

4.9.3.5 VAO 參數表 (VAO parameter table)

VAO 參數表讓使用者可以決定控制模式和速度功率表的相關設定，以下表格將說明各個 VAO 參數的定義以提供使用者使用。

NO	Define	Description	Value	Default:
0x00 + (2 * N) Note: N is TableNo, range is 0 ~ 7.	VAO_TABLE_OUTPUT_TYPE	Table output type	1: PWM mode 2: PWM frequency mode with fixed width 3: PWM frequency mode with fixed duty cycle	1
0x01 + (2 * N) Note: N is TableNo, range is 0 ~ 7.	VAO_TABLE_INPUT_TYPE	Table input type	0: Feedback speed 1: Command speed	0
0x10 + N Note: N is TableNo, range is 0 ~ 7.	VAO_TABLE_PWM_Config	Configure PWM according to output type	a. Mode 1 - set a fixed frequency (3~50M Hz) b. Mode 2 - set a fixed Pulse Width (20~335544300 ns) c. Mode 3 - set a fixed duty cycle: N * 0.05 %. (N: 1 ~ 2000)	100
0x20 + N Note: N is TableNo, range is 0 ~ 7.	VAO_TABLE_SRC	Specify axisID for VAO table.	Bit0: Axis 0 On Bit1: Axis 1 On Bit2: Axis 2 On Bit3: Axis 3 On	0x01
0x30~	Reserved	Reserved	Reserved	Reserved

4.9.3.6 數位輸出 (digital output) 與 PWM 的對應功能

VAO 模組支援一種特殊的對應功能可讓使用者利用數位輸出控制 PWM 開或關。此種功能可配合點表 (point table) 使用，利用點表內數位輸出控制選項可讓使用者在指定位置開啟或關閉 PWM 訊號輸出。在使用此種功能時，使用者必須利用卡片參數 (Board parameter) 指定並建立 PWM 與數位輸出的對應關係如下圖所示。其中 DO0~7 為數位輸出。舉例來說，假設指定 PWM 0 對應到 DO2，同時設定邏輯 (logic) 為 1，則當 DO2 由 low 變 high 時則 PWM 0 開始輸出，而當 DO2 由 high 變 low 則 PWM 0 會停止輸出。反之若邏輯為 0 則當 DO2 由 high 變 low 時則 PWM 0 開始輸出。需注意當開啟 VAO

模組 PWM 輸出時，若對應關係已經建立時，PWM 輸出是否啟動或停止將由當時相對應的數位輸出狀態及邏輯決定。以下為簡單使用範例

1. 使用 APS_set_board_param() 依照卡片參數表設定 PWM 輸出通道與相對應數位輸出以及判斷邏輯。
2. 在點表 (Point table) 指定點的選項 (Option) 中打開 DO_Enable, 決定 DO_Channels 及 DO_ON 或 DO_OFF。

對應關係圖

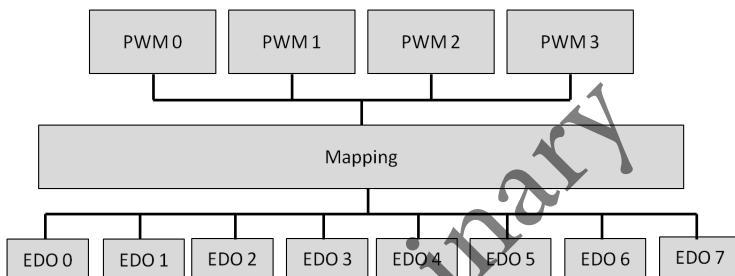


表 4-3：卡片參數表 (Board parameter table)

NO.	SYMBOL	Description	Default
110h	PRB_PWM0_MAP_DO	(1) -1: Disable mapping ; > 0: Enable mapping (2) Bit0~7: Specify a Do channel. (3) Bit8: Select logic; Set to 1: Turning on Do maps enabling PWM0. Turning off Do maps disabling PWM0. Set to 0: Turning on Do maps disabling PWM0. Turning off Do maps enabling PWM0.	-1
111h	PRB_PWM1_MAP_DO	Please see description of PRB_PWM0_MAP_DO	-1
112h	PRB_PWM2_MAP_DO	Please see description of PRB_PWM0_MAP_DO	-1
113h	PRB_PWM3_MAP_DO	Please see description of PRB_PWM0_MAP_DO	-1

4.9.3.7 操作流程範例

以下部分將說明不同控制模式的操作流程供使用者參考。

Mode	Description
1: PWM mode	<p>a. VAO parameter table - <code>APS_set_vao_param()</code> 0x00: set to 1 – PWM mode 0x01: set to 1 – command speed 0x10: set to 1000 – set fixed frequency to 1000 Hz 0x20: set to 3 – Axis0 and Axis1 are selected</p> <p>b. “Velocity to Power” mapping lookup table - <code>APS_set_vao_table()</code> Duty cycle range: 0 ~ 2000 units (Be equal to 0 ~ 100 %) Points range: 1 ~ 32 points</p> <p>c. Switch VAO table - <code>APS_switch_vao_table()</code> d. Enable VAO output channel - <code>APS_start_vao()</code></p>
2: PWM frequency mode with fixed width	<p>a. VAO parameter table - <code>APS_set_vao_param()</code> 0x00: set to 2 – PWM frequency mode with fixed width 0x01: set to 1 – command speed 0x10: set to 1000 – set fixed pulse width to 1000 ns 0x20: set to 3 – Axis0 and Axis1 are selected</p> <p>b. “Velocity to Power” mapping lookup table - <code>APS_set_vao_table()</code> Frequency range: 1 ~ 25Mhz for PCI-8253/6 Points range: 1 ~ 32 points</p> <p>c. Switch VAO table - <code>APS_switch_vao_table()</code> d. Enable VAO output channel - <code>APS_start_vao()</code></p>
3: PWM frequency mode with fixed duty cycle	<p>a. VAO parameter table - <code>APS_set_vao_param()</code> 0x00: set to 3 – PWM frequency mode with fixed duty cycle 0x01: set to 1 – command speed 0x10: set to 200 – set fixed duty cycle to 10%. (200 * 0.05 %) 0x20: set to 3 – Axis0 and Axis1 are selected</p> <p>b. “Velocity to Power” mapping lookup table - <code>APS_set_vao_table()</code> Frequency range: 1 ~ 25Mhz for PCI-8253/6 Points range: 1 ~ 32 points</p> <p>c. Switch VAO table - <code>APS_switch_vao_table()</code> d. Enable VAO output channel - <code>APS_start_vao()</code></p>

4.9.4 運動控制暨 I/O 取樣功能 (Sampling function)

4.9.4.1 取樣來源 Sampling source

本控制卡支援取樣不同訊號以供使用者分析。訊號來源主要分為兩種，第一種屬於運動核心訊號，第二種則為閉迴路控制訊號。以下圖為例說明，底層的運動核心和控制器的更新速度 (Update rate) 分別是 1ms 與 250us(控制器的更新速度是可調的)，而我們取樣機制 (Sampling process) 的取樣速度 (Sampling rate) 則是 1ms，最後透過 APS library 將取樣訊號送到 MotionCreatorPro2 或是其他應用程式中呈現出來。建議若想得知整體系統控制效能，觀察閉迴路控制訊號會是較好的選擇。須注意，若使用者選取脈衝控制模式 (Pulse mode)，取樣閉迴路控制訊號是無效的。以下表格將說明各訊號的意義，實際操作頁面及步驟詳述於 MotionCreatorPro 2 使用手冊中第四小節。

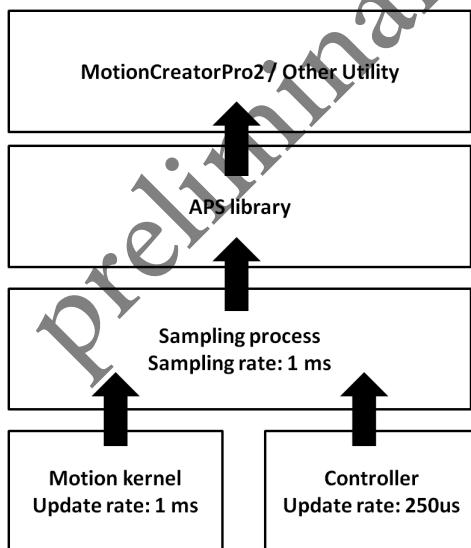


圖 4-57：訊號取樣架構圖

表 4-4：運動核心 (Motion kernel) 訊號表

訊號名稱	範圍	資料形態	描述
SAMP_SRC_COM_POS	Axis 0~7	Integer	位置命令 (command position); 單位 : pulse
SAMP_SRC_FBK_POS	Axis 0~7	Integer	回饋位置 (feedback position); 單位 : pulse
SAMP_SRC_CMD_VEL	Axis 0~7	Integer	速度命令 (command velocity); 單位 : pulse/sec
SAMP_SRC_FBK_VEL	Axis 0~7	Integer	回饋速度 (feedback velocity); 單位 : pulse/sec
SAMP_SRC_MIO	Axis 0~7	Integer	運動 I/O (motion IO), 定義請參考註 1
SAMP_SRC_MSTS	Axis 0~7	Integer	運動狀態 (motion status), 定義請參考註 2
SAMP_SRC_MSTS_ACC	Axis 0~7	Integer	運動狀態 : 加速段 (motion status acc); 0: 無加速 1: 目前加速
SAMP_SRC_MSTS_MV	Axis 0~7	Integer	運動狀態 : 等速段 (motion status at max velocity); 0: 無等速 1: 目前等速
SAMP_SRC_MSTS_DEC	Axis 0~7	Integer	運動狀態 : 減速段 (motion status DEC); 0: 無減速 1: 目前減速
SAMP_SRC_MSTS_CSTP	Axis 0~7	Integer	運動狀態 : 運動命令停止 (motion status CSTP); 0: 運動進行中 1: 運動命令停止
SAMP_SRC_MSTS_MDN	Axis 0~7	Integer	運動狀態 : 運動完成 (motion status MDN); 0: 運動進行中 1: 運動完成
SAMP_SRC_MIO_INP	Axis 0~7	Integer	運動狀態 : 運動到位 (motion status INP); 0: 運動未到位 1: 運動到位
SAMP_SRC_MIO_ORG	Axis 0~7	Integer	運動狀態 : ORG 訊號 (motion status OGR); 0: 無 ORG 訊號 1: 觸碰到 ORG 訊號
SAMP_SRC_CONTROL_VOL	Axis 0~7	Integer	輸出電壓 (Control command voltage); 單位 : mV
SAMP_GTY_DEVIATION	Axis 0~7	Integer	龍門運動時指定軸與對應主軸 (或從軸) 之間回饋位置偏移量 (Gantry deviation); 單位 : pulse
SAMP_SRC_ENCODER_RAW	Axis 0~7	Integer	驅動器回饋位置原始訊號 (Encoder raw data); 單位 : pulse
SAMP_SRC_ERR_POS	Axis 0~7	Integer	位置誤差 (Error position); 單位 : pulse
SAMP_SRC_COM_POS_F64	Axis 0~7	Double	同 SAMP_SRC_COM_POS, 但以浮點數表示
SAMP_SRC_FBK_POS_F64	Axis 0~7	Double	同 SAMP_SRC_FBK_POS, 但以浮點數表示
SAMP_SRC_CMD_VEL_F64	Axis 0~7	Double	同 SAMP_SRC_CMD_VEL, 但以浮點數表示

訊號名稱	範圍	資料形態	描述
SAMP_SRC_FBK_VEL_F64	Axis 0~7	Double	同 SAMP_SRC_FBK_VEL，但以浮點數表示
SAMP_SRC_CONTROL_VOL_F64	Axis 0~7	Double	同 SAMP_SRC_CONTROL_VOL，但以浮點數表示
SAMP_SRC_ERR_POS_F64	Axis 0~7	Double	同 SAMP_SRC_ERR_POS，但以浮點數表示
SAMP_PWM_FREQUENCY_F64	Channel 0~3	Double	PWM 訊號頻率 (PWM frequency); 單位 : Hz
SAMP_PWM_DUTY_CYCLE_F64	Channel 0~3	Double	PWM 訊號佔空比 (PWM duty cycle); 單位 : 百分比 (%)
SAMP_PWM_WIDTH_F64	Channel 0~3	Double	PWM 訊號寬度 (PWM width); 單位 : ns
SAMP_VAO_COMP_VEL_F64	No. 0~1	Double	控制雷射功率的合成速度 (Composed velocity for Laser power control); 單位 : pulse/sec
SAMP_PTBUFF_COMP_VEL_F64	Table 0~1	Double	點表運動合成速度 (Composed velocity of point table); 單位 : pulse/sec
SAMP_PTBUFF_COMP_ACC_F64	Table 0~1	Double	點表運動合成加速度 (Composed acceleration of point table); 單位 : pulse/sec ²

註 1: 運動 I/O 定義表 (Motion IO definition table)

7	6	5	4	3	2	1	0
SVON	INP	EZ	EMG	ORG	MEL	PEL	ALM
15	14	13	12	11	10	9	8
			SMEL	SPEL			

Bit number detail description:

Bit	Define	Description
0	ALM	Servo alarm input status
1	PEL	Positive end limit
2	MEL	Minus end limit
3	ORG	Original input (Home input)
4	EMG	Emergency stop input
5	EZ	Servo index input
6	INP	In-Position input
7	SVON	Servo ON output status
...		
11	SPEL	1: Soft-positive-end limit condition match.
12	SMEL	1: Soft-minus-end limit condition match

註 2: 運動狀態定義表 (Motion status definition table)

7	6	5	4	3	2	1	0
	HMV	MDN	DIR	DEC	ACC	VM	CSTP
15	14	13	12	11	10	9	8
JOG				PTB	WAIT		
23	22	21	20	19	18	17	16
				POSTD	PRED	BLD	ASTP
31	30	29	28	27	26	25	24
			GER				

Bit number detail description:

Bit	Define	Description
0	CSTP	Command stopped (But it could be in motion)
1	VM	In maximum velocity
2	ACC:	In acceleration
3	DEC:	In deceleration
4	DIR:	Move direction. 1:Positive direction, 0:Negative direction
5	MDN	Motion done. 0: In motion, 1: Motion done (It could be abnormal stop)
6	HMV	In homing
...		
10	WAIT	Axis is in waiting state. (Wait move trigger)
11	PTB	Axis is in point buffer moving. (When this bit on, MDN and ASTP will be cleared)
...		
15	JOG	In jogging
16	ASTP	0: Stop normally, 1: abnormal stop. When axis in motion, this bit will be clear.
17	BLD	Axis (Axes) in blending moving
18	PRED	Pre-distance event, 1: event arrived. The event will be clear when axis start moving
19	POSTD	Post-distance event. 1: event arrived. The event will be clear when axis start moving
...		
28	GER	1: In geared (This axis as slave axis and it follow a master specified in axis parameter.)
29	--	

4.9.5 同動控制 (Simultaneous Move)

4.9.5.1 同時啟動 (Simultaneous Start)

同步啟動 (亦稱同動)：此運動可設定為觸發啟動，指令下達時，該軸會進入等待觸發訊號的狀態，當接收到觸發訊號後才開始運動。另外當多軸處於等待觸發的狀態時，可以同時發送觸發訊號來達成同時啟動的目的，但須注意是各軸為獨立運動，結束時間會依照所設定的位移量及加速曲線來決定。

請依據下列二個步驟來達成同時啟動的功能：

- a 將軸運動設定為觸發啟動，並監測軸狀態是否處於等待觸發的狀態
 - b 發出觸發訊號來執行同步啟動
-
- a 將軸運動設定為觸發啟動的模式，監測軸狀態是否處於等待觸發的狀態
- 透過控制器提供的運動函式的 Option 參數，可以設定為觸發啟動的模式，指令下達後，該軸會進入等待觸發訊號的狀態。

以 APS_ptp 為例子，函式原型為

`I32 APS_ptp(I32 Axis_ID, I32 Option, ...);`

其中 Option 的定義為如下表，當 Bit 8 設定為 1 時，指定的軸會被設定為觸發啟動的模式。

7	6	5	4	3	2	1	Bit : 0
							Absolute(0) / Relative(1)
15	14	13	12	11	10	9	8
Buffer mode							Wait trigger

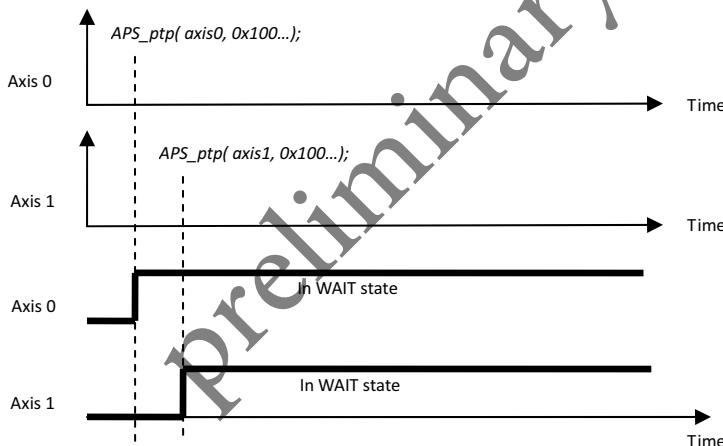
當軸運動被設定為觸發啟動後，該軸會進入等待觸發訊號的狀態，也就是下表中的 Bit 10 的 WAIT 訊號，會被顯示為 ON. 可以透過控制器提供的函示庫來顯示其訊號狀態，此運動狀態監控函式為

```
I32 APS_motion_status();
```

Motion status definition table							
7	6	5	4	3	2	1	Bit : 0
--	HMV	MDN	DIR	DEC	ACC	VM	CSTP
15	14	13	12	11	10	9	8
JOG	--	--	--	PTB	WAIT	--	--

下圖為運動被設定為等待觸發的示意圖

Velocity



相關的 APS API 如下

```
I32 APS_ptp();I32 APS_ptp_v();I32 APS_ptp_all();I32
APS_line();I32 APS_line_v();  

I32APS_line_all();I32 APS_vel();I32 APS_vel_all();I32
APS_arc2_ca();I32 APS_arc2_ca_v();  

I32 APS_arc2_ca_all();I32 APS_arc2_ce();I32
APS_arc2_ce_v();I32 APS_arc2_ce_all();
```

```

I32 APS_arc3_ca();I32 APS_arc3_ca_v();I32 APS_arc3_ca_all();
);I32 APS_arc3_ce();
I32 APS_arc3_ce_v();I32 APS_arc3_ce_all();I32
APS_arc3_ca();I32 APS_arc3_ca_v();
I32 APS_arc3_ca_all();I32 APS_sprial_ca();I32
APS_sprial_ca_v();I32 APS_sprial_ca_all();
I32 APS_sprial_ce();I32 APS_sprial_ce_v();I32
APS_sprial_ce_all();

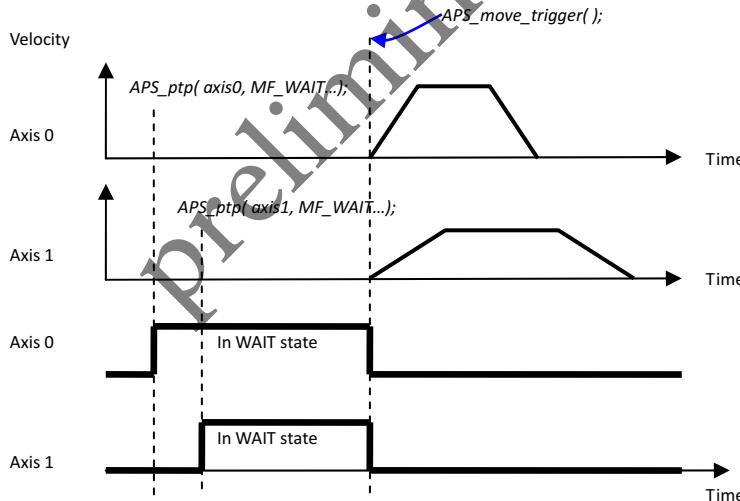
```

b. 發出觸發訊號來執行同步啟動

透過函式發出觸發訊號，可以達成多軸的同時啟動。函式的原形為：

```
I32 APS_move_trigger( );
```

下圖為發出觸發訊號後的多軸同時啟動示意圖：



相關的 APS API 如下：

```
I32 APS_move_trigger(); // 發出觸發訊號
```

```
I32 APS_stop_move_multi(); // 同步減速停止
```

```
I32 APS_emg_stop_multi(); // 同步 Emg 停止
```

- 例：

```
#include "APS168.h"
#include "APS_define.h"
#include "ErrorCodeDef.h"

void simultaneous_move_example()
{
    //This example shows how to execute a simultaneous move
    I32 option = 0x100; //bit 8 = 1
    I32 return_code = 0;
    I32 dimension = 2;
    I32 axis_array[2] = { 0, 1 };

    return_code = APS_ptp( 0, option, 10000, 0 ); //axis 0 設定為觸發啟動模式
    return_code = APS_ptp( 1, option, 20000, 0 ); //axis 1 設定為觸發啟動模式
    return_code = APS_move_trigger( dimension, axis_array ); //發出觸發訊號讓axis 0 & 1同時啟動
}
```

4.9.6 點表運動

控制器提供兩組的點表功能，每一組點表提供 50 個 buffer 點的空間，透過監控 buffer 點的空間使用狀態，並不斷的重覆載入這 50 個 buffer 點空間，可以實現需要大量點數的點表功能，在實際應用上不會受限於點數限制。

透過控制器提供的點表運動功能，使用者可以透過相關的函式來達成多線段的連續運動。點表運動的運動指令包含了直線運動，圓弧運動，螺旋補間運動以及暫停 (dwell)，命令指令則包含數位輸出以及 VAO table 切換，使用者可以依據其應用領域來程式化相關需求。

4.9.6.1 點表參數設定

在點表內主要的參數，可以分為下列三個項目，分別為：

- a 設定運動參數
- b 設定命令指令
- c 設定運動指令到點表中

a. 設定運動參數

在執行運動指令之前，使用者需要先設定運動參數，運動參數包含絕對、相對運動、最大速度、結束速度、加減速度、S 因子以及相鄰路徑其速度接續的方式等，根據應用來給予適當的速度與路徑的規劃。需注意的是，這些運動參數的特性只需要設定一次就會被程式記憶起來，其他運動命令會自動的套用先前的設定，除非使用者要變動參數的設定，否則不需要將參數依據所有運動命令做重複設定。

運動參數設定	對應 APS 函式
絕對 / 相對運動	APS_pt_set_absolute / APS_pt_set_relative
最大速度	APS_pt_set_vm
結束速度	APS_pt_set_ve
加速度	APS_pt_set_acc
減速度	APS_pt_set_dec
加速度和減速度	APS_pt_set_acc_dec
S 因子	APS_pt_set_s
相鄰路徑其速度接續的方式 (請參考 4.11.3 小節)	APS_pt_set_trans_buffered (緩衝) APS_pt_set_trans_inp (緩衝並且到位) APS_pt_set_trans_blend_dec (混和 - 減速) APS_pt_set_trans_blend_dist (混和 - 殘餘距離) APS_pt_set_trans_blend_pcnt (混和 - 殘距比)

b. 設定命令指令

命令指令為同步指令，會伴隨著點表中的運動指令一同被執行，也就是說在運動過程中，可以在不同的運動線段上同步的控制數位輸出。

命令指令 (伴隨著運動同步被執行)	對應 APS 函式
數位輸出 (DO)	APS_pt_ext_set_do_ch
Vao table 切換	APS_pt_ext_set_table_no

c. 設定運動指令到點表中

點表提供了下列的運動指令，包含直線、圓弧以及螺旋補間，透過對應的 APS 函式，將運動指令填入到點表中。

運動指令	對應 APS 函式
直線補間	APS_pt_line
圓弧補間	APS_pt_arc2_ca / APS_pt_arc2_ce APS_pt_arc3_ca / APS_pt_arc3_ce
螺旋補間	APS_pt_sprial_ca / APS_pt_sprial_ce
暫停	APS_pt_dwell

透過上述的三個步驟，設定相關的運動參數，設定需同步的命令指令，最後將運動指令設入到點表中存放，照著這個步驟，依序的把所有圖形線段存入到點表內存放。

4.9.6.2 實現點表運動

控制器也提供兩組的點表功能，每一組點表提供 50 個 buffer 點的空間，透過監控 buffer 點的空間使用狀態，並不斷的重覆載入這 50 個 buffer 點空間，可以實現大量點數的運動圖形的點表功能，在實際應用上不會受限於點數限制。

實現一個點表運動，分為下列三個項目：

- a 致能 / 禁能點表功能
- b 監控 buffer 空間，並且填點
- c 啟動 / 停止點表運動

a. 致能 / 禁能點表功能

使用點表功能需要先將點表致能，同時指定點表的 ID(0~1)，指定運動維度，以及指定軸號碼。當點表功能結束後，則必須把它做禁能設定。

I32 APS_pt_enable(I32 Board_ID, I32 PtId, I32 Dimension, I32 *AxisArr);

點表功能	對應 APS 函式
致能點表功能	APS_pt_enable
禁能點表功能	APS_pt_disable

b. 監控 buufer 空間，並且填點

每一組點表提供 50 個 buffer 點空間來供使用，應用上透過監控這些 buffer，並同時將運動指令填入點表中（參考 4.11.1 節），依序的將所有圖形點動態載入。

點表功能	對應 APS 函式
監控 buffer 狀態	APS_get_pt_status

c. 啟動 / 停止點表運動

在致能點表，並依序的填入運動指令到 buffer 中之後，使用者可以下達啟動點表的功能。此時，運動核心程式將會根據 buffer 中的運動指令依序的執行，直到使用者中斷點表或是 buffer 點數已經執行完畢。

點表功能	對應 APS 函式
啟動點表運動	APS_pt_start
結束點表運動	APS_pt_stop

• 例：

```
#include "APS168.h"
#include "APS_define.h"
#include "ErrorCodeDef.h"

void pt_move_example()
{
    //This example shows how pt move operation
    I32 ret;
    I32 Board_ID = 0;
    I32 PtBld = 0; //Point table 0
    I32 Dimension = 2; //2D Dimension
    I32 AxisArr[2] = { 0, 1 }; //Set Axis 0 & Axis 1 to point table 0
    PTLINE Prof;
    PTSTS Status;

    //Enable point table id 0 for 2D dimension with aixs 0 and axis 1.
    APS_pt_enable( Board_ID , PtBld, Dimension, & AxisArr ); //Enable point table id 0

    //Get status of point table id 0 to monitor buffer
    APS_get_pt_status( Board_ID , PtBld, &Status );
    if ( !( Status.Bitsts & 0x02 ) ) //Point buffer is not full
    {
        //Push move into point buffer
        Prof.Dim = 2;
        Prof.Pos[0] = 10000;
        Prof.Pos[1] = 10000;
        ret = APS_pt_line( Board_ID, PtBld, &Prof, &Status );
    }
    //Start point table move
    APS_pt_start( Board_ID, PtBld, 0 );
}
```

4.10 安全保護 (Safety Protection)

在設備的操作的過程可能會遭遇到錯誤或者必須緊急停止的狀況，一般的做法是將運轉中的機構設備做停止的動作。本控制器提供一些安全機制來偵測預先設定好的錯誤行為，當錯誤行為發生時控制器會做適當的處理，以保護人員安全及防止設備的損壞。這些安全機制中，有些需要外接硬體訊號，有些則利用軟體的方式檢查周期的檢查，下面小節將分別介紹這些安全機制。

4.10.1 硬體保護 (Hardware Protection)

控制器提供了外部硬體訊號的偵測保護機制，這些訊號包含緊急停止 (EMG)，伺服警報 (ALM) 以及機構正負極限 (PEL, MEL)。以下各小章節將詳細描述其原理。

4.10.1.1 緊急停止 (EMG)

EMG 硬體輸入腳位，參照下表：

P1A Pin No	Signal Name
51	IEMG

EMG 訊號為一個硬體的輸入訊號，當 EMG 訊號為 ON 時將導致控制器操作在下列幾種行為：

1. 若軸在運動狀態下，當 EMG 訊號為 ON，控制器會立即的停止相對應的軸運動。此時該軸的錯誤停止的代碼(Stop code) 被設定為 "1" (STOP_EMG)，且運動狀態為異常停止狀態 (ASTP)。
2. 若軸是在非運動狀態下，且 EMG 訊號是 ON，此時控制器不會運行使用者的運動指令，同時讓軸的 stop code = STOP_EMG(1)，且運動狀態為異常停止狀態 (ASTP)。

相關 APIs:

`APS_motion_status();` // 讀取運動狀態 (ASTP)

`APS_get_stop_code();` // 讀取錯誤停止的代碼

4.10.1.2 伺服警報 (ALM)

ALM 硬體輸入腳位以及相對應的軸號，參照下表：

P1A Pin No	Signal Name	Axis #	P1B Pin No	Signal Name	Axis #
35	ALM1	0	35	ALM5	4
41	ALM2	1	41	ALM6	5
85	ALM3	2	85	ALM7	6
91	ALM4	3	91	ALM8	7

ALM 訊號為一個硬體的輸入訊號，來自伺服驅動器的 ALM 訊號，透過 ALM 腳位輸入到控制器內。當 ALM 訊號為 ON 時，將導致控制器操作在下列幾種行為：

1. 若軸是在運動狀態下，當 ALM 訊號被致動，控制器會立即的停止相對應的軸運動，此時該軸的錯誤停止的代碼 (Stop code) 被設定為 "2" (STOP_ALM)，且運動狀態為異常停止狀態 (ASTP = ON)。
2. 若軸是在非運動狀態下，且 ALM 訊號是致動的，此時控制器不會運行使用者的運動指令，同時讓該軸的錯誤停止的代碼 (Stop code) 被設定為 "2" (STOP_ALM)，且運動狀態為異常停止狀態 (ASTP = ON)。

4.10.1.3 正負極限訊號 (PEL / MEL)

EL 硬體輸入腳位以及相對應的軸號，參照下表：

P1A Pin No	Signal Name	Axis #	P1B Pin No	Signal Name	Axis #
38	PEL1	0	40	MEL1	0
44	PEL2	1	46	MEL2	1
88	PEL3	2	90	MEL3	2
94	PEL4	3	96	MEL4	3
38	PEL5	4	40	MEL5	4
44	PEL6	5	46	MEL6	5
88	PEL7	6	90	MEL7	6
94	PEL8	7	96	MEL8	7

EL 訊號為一個硬體的輸入訊號，分為 PEL 和 MEL 兩種訊號。PEL 為正方向的極限訊號，MEL 為負方向的極限訊號。致動 EL 訊號將導致控制器操作在下列幾種行為：

1. 若軸是在正方向運動狀態下，當 PEL 訊號被致動，控制器會停止相對應的軸運動，此時該軸的錯誤停止的代碼(Stop code)被設定為 "4" (STOP_PEL)，且運動狀態為異常停止狀態 (ASTP = ON)。
2. 若軸是在負方向運動狀態下，當 MEL 訊號被致動，控制器會停止相對應的軸運動，此時該軸的錯誤停止的代碼(Stop code)被設定為 "5" (STOP_MEL)，且運動狀態為異常停止狀態 (ASTP = ON)。
3. 若軸是在非運動狀態下，且 PEL 訊號是致動的，此時控制器不會運行使用者的正方向運動指令，同時讓該軸的錯誤停止的代碼(Stop code)被設定為 "4" (STOP_PEL)，且運動狀態為異常停止狀態 (ASTP = ON)。
4. 若軸是在非運動狀態下，且 MEL 訊號是致動的，此時控制器不會運行使用者的負方向運動指令，同時讓該軸的錯誤停止的代碼(Stop code)被設定為 "5" (STOP_MEL)，且運動狀態為異常停止狀態 (ASTP = ON)。
5. 有兩種停止模式可以選擇，分別為減速停止和立即停止，軸參數標號為 PRA_EL_MODE(0x02)。

4.10.2 軟體保護 (Software Protection)

控制器提供了軟體保護機制，包含軟體極限以及位置誤差保護。

4.10.2.1 軟體極限 (Soft-limit Signal)

軟體極限的功能和硬體極限類似，差別在於軟體極限是以檢查各軸所在的位置來產生極限訊號，同樣區分為正極限 (SPEL) 和負極限 (SMEL) 兩種訊號。要使用軟體極限的步驟如下：

1. 設定軟體極限的位置，軸參數如下表中的 PRA_SPEL_POS 和 PRA_SMEL_POS。
2. 設定極限訊號產生時的停止模式，可選擇減速停止和立即停止，參照軸參數標號為 PRA_EL_MODE(0x02)，PRA_SD_DEC (0x07)
3. 啟動軟體極限功能，軸參數如下表中的 PRA_SPEL_EN (0x08) 和 PRA_SMEL_EN(0x09)。

通常啟用軟體極限功能之前，會先進行定位 (Homing) 的動作已確立坐標系的極限位置。

NO	Define	Description
02h	PRA_EL_MODE	EL訊號停止模式 若選擇減速模式，減速率參考參數：PRA_SD_DEC.
07h	PRA_SD_DEC	停止減速率設定
08h	PRA_SPEL_EN	Soft PEL 開關
09h	PRA_SMEL_EN	Soft MEL 開關
0Ah	PRA_SPEL_POS	Soft PEL 位置
0Bh	PRA_SMEL_POS	Soft MEL 位置

當軟體極限功能啟動後，可以透過控制器提供的函示庫來顯示其訊號狀態。此 IO 監控函式為

APS_motion_io_status();

當軟體極限訊號為 ON 時將導致控制器操作在下列幾種行為：

1. 若軸是在正方向運動狀態下，當 SPEL 訊號被致動，控制器會停止相對應的軸運動，此時軸的停止碼 (stop code) = STOP_SPEL (6)，且運動狀態為異常停止狀態 (ASTP)。
2. 若軸是在負方向運動狀態下，當 SMEL 訊號被致動，控制器會停止相對應的軸運動，此時軸的停止碼 (stop code) = STOP_SMEL (7)，且運動狀態為異常停止狀態 (ASTP)。
3. 若軸是在非運動狀態下，且 SPEL 訊號是致動的，此時控制器不會運行使用者的正方向運動指令，同時讓軸的停止碼 (stop code) = STOP_SPEL (6)，且運動狀態為異常停止狀態 (ASTP)。
4. 若軸是在非運動狀態下，且 SMEL 訊號是致動的，此時控制器不會運行使用者的負方向運動指令，同時讓軸的停止碼 (stop code) = STOP_SMEL (7)，且運動狀態為異常停止狀態 (ASTP)。

4.10.2.2 位置誤差保護 (Position Error Protection)

位置誤差保護是一種軟體保護的機制，主要是監控命令計數器與回授計數器的差值，此差值定義為位置誤差。當位置誤差過大時，控制器會送出 Servo off 訊號，其使用時機可以在伺服調機前設定使用。

此功能在軸參數中設定，設定如下表所示，當位置誤差 (PRA_ERR_POS_LEVEL) 設定為 "0" 時，表示不啟動位置誤差保護 (Disable)。當位置誤差設定為非零值，則啟動位置誤差保護。

NO	Define	Description
124h	PRA_ERR_POS_LEVEL	位置誤差保護設定

位置保護將導致控制器操作在下列幾種行為：

當位置誤差大於使用者的設定值時，控制器會直接 Servo off，此時軸的停止碼 (Stop code) = STOP_ERROR_LEVEL，且運動狀態為異常停止狀態 (ASTP)。

4.10.2.3 看門狗 (Watch Dog)

看門狗的保護機制，是控制器內部有實作一個計時器，當計時器逾時，會啟動事先設定好的回應動作，回應動作包括伺服電機的解除激磁 (Servo off)，關閉數位輸出以及關閉 PWM 輸出等。當使用者啟動看門狗機制後，使用者的程式必須處在可回應的狀態，在計時器尚未逾時前，不間斷的重置看門狗，讓計時器重新開始計時。只要使用者程式一直保持在可回應狀態，那麼就不會觸發相對應的事件。換句話說，看門狗的功能是監控上位的控制程式是否進入停滯 (當機) 的狀態，當停滯狀態發生時，控制器可以觸發保護機制來關閉訊號輸出。

在 Windows 下使用中斷的方式如下列三大步驟：

1. 設定計時器逾時的觸發事件
2. 啟動看門狗保護機制
3. 在週期時間內重置計時器

其相關的 APS APIs 如下列：

```
APS_wdt_set_action_event();  
APS_wdt_get_action_event();  
APS_wdt_start();  
APS_wdt_get_timeout_period();  
APS_wdt_reset_counter();
```

詳細的使用方式描述如下：

1. 設定計時器逾時的觸發事件：

使用 **APS_wdt_set_action_event()** 函式來設定觸發事件。

使用 **APS_wdt_get_action_event()** 函式來獲得觸發事件。

2. 啟動看門狗保護機制：

使用者使用 **APS_wdt_start()** 設定一個逾時的時間，同時啟動看門狗，此時內部計時器就會開始運作。若是逾時的時間設定為 0，則代表看門狗不啟動。

使用 **APS_wdt_get_timeout_period()** 可以讀取設定的逾時時間。

3. 不間斷的重置計時器

在啟動看門狗後，需要在設定的逾時時間內，去重置看門狗，讓計時器可以歸零，並重新計時。若計時器逾時了，會依據步驟 1 的設定，去觸發相關的事件。

使用 **APS_wdt_reset_counter()** 去重置看門狗。

- 例：

```
void watchdog_example()
{
    // This example shows how interrupt functions work.

    I32 board_id = 0;

    I32 timer_no = 0; // Only timer 0 to be used

    I32 time_out = 10; // Time out is 10*100 ms = 1 sec

    I32 EventByBit = 0x01; // Action event is defined by bit.

        // Bit0: Motor servo off
        // Bit1: Digital output off
        // Bit2: PWM off

    I32 ret = 0; // return code

    //Step 1: 設定計時器觸發後的事件
    ret = APS_wdt_set_action_event( board_id, timer_no, EventByBit );

    //Step 2: 啟動看門狗保護機制
    ret = APS_wdt_start( board_id, timer_no, time_out );
    //使用者的timer，while loop，或者是可回應點

    timer(500ms)//每500ms去重置一次計時器

    {

        //Step 3: 不間斷的重置計時器

        ret = APS_wdt_reset_counter( board_id, timer_no );
        ...Do Something
    }
}
```

4.11 本機中斷 (Host Interrupt)

中斷係指當某一指定的事件產生，裝置（本控制器）發出硬體中斷訊號給作業系統，而作業系統再通知驅動程式執行相對應的中斷服務函式 (Interrupt service routine) 的過程，下圖為中斷流程示意。

中斷與輪詢的機制，常被使用來偵測某事件的產生。若使用輪詢的機制，會重覆地占用 CPU 的時間，來偵測事件的產生，導致 CPU 的使用率過高。而透過中斷的機制，在事件產生後，裝置會通知 CPU 事件已經產生，這過程中不會消耗掉多餘的 CPU 時間，因此可以降低 CPU 的使用率，並且在等待中斷的同時程式可以處理其他工作達到多工且有效率利用 CPU 資源。

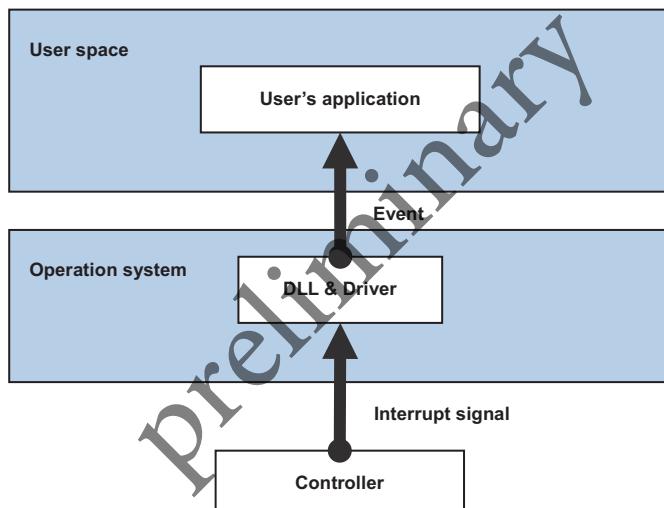


圖 4-58：中斷流程圖

本控制器所提供的中斷事件類別區分為：

1. 軸中斷 (Axis interrupt)
2. 系統中斷 (System interrupt)
3. 數位輸入中斷 (Digital input interrupt)

與控制軸相關的事件均歸類為軸中斷，數位輸入中斷包含上緣觸發 (Rising edge interrupt) 和下緣觸發 (Falling edge interrupt)。其餘事件歸類為系統中斷。

下表描述本控器所有的中斷事件類別，其中 Item 0~7 分別為各控制軸相關的中斷 (註 :AMP-204C 為 Item 0~3, 4~7 部分為保留)，Item 8 為系統相關中斷，Item 9 和 10 為數位輸入中斷。

- **Interrupt Item 總覽 :**

Item	Item 類別描述
0~7	軸 0~7 中斷 (AMP-204C 之 Item 4 ~ 7 保留)
8	系統中斷
9	數位輸入上緣觸發中斷
10	數位輸入下緣觸發中斷

每個 Item 分別有最多 32 種中斷事件 (32 bit)，詳細定義請參考下列各表格：

- **Item = 0~7 : 軸中斷事件總覽**

Bit No.	7	6	5	4	3	2	1	0
Factor	--	IEMG	IINP	IEZ	IORG	IMEL	IPEL	IALM
Bit No.	15	14	13	12	11	10	9	8
Factor	ISPEL	--	IASTP	IMDN	IDECK	IACC	IVM	ICSTP
Bit No.	23	22	21	20	19	18	17	16
Factor	--	--	--	--	IPOSTD	IPRED	--	ISMEL
Bit No.	31	30	29	28	27	26	25	24
Factor	--	--	--	--	--	--	--	--

• 軸中斷事件描述：

bit.	符號	中斷事件說明
0	IALM	ALM 訊號發生
1	IPEL	PEL 訊號發生
2	IMEL	MEL 訊號發生
3	IORG	ORG 訊號發生
4	IEZ	馬達 Z 相訊號 (EZ) 發生
5	IINP	驅動器到位 INP 訊號發生
6	IEMG	緊急停止訊號 EMG 發生 (同系統 IEMG)
7	--	保留 (Reserved) , 設定為 0
8	ICSTP	CSTP 訊號發生
9	IVM	最大速
10	IACC	開始加速
11	IDEC	開始減速
12	IMDN	運動完成
13	IASTP	不正常停止
14	--	保留 (Reserved) , 設定為 0
15	ISPEL	Soft PEL 發生
16	ISMEL	Soft MEL 發生
17	--	保留 (Reserved) , 設定為 0
18	IPRED	Pre-distance 事件發生
19	IPOSTD	Post-distance 事件發生
20~	--	保留 (Reserved) , 設定為 0

- Item = 8 : 系統中斷事件總覽

Bit No.	7	6	5	4	3	2	1	0
Factor	--	IHOV	IMOV	IFCF1	IFCF0	ILCF1	ILCF0	IEMG
Bit No.	15	14	13	12	11	10	9	8
Factor	--	--	--	--	--	--	--	--
Bit No.	23	22	21	20	19	18	17	16
Factor	--	--	--	--	--	--	--	--
Bit No.	31	30	29	28	27	26	25	24
Factor	--	--	--	--	--	--	--	--

- 系統中斷事件描述

bit.	符號	中斷事件說明
0	IEMG	緊急停止訊號 EMG 發生
1	ILCF0	線性比較器 NO.0 (Linear comparator 0) 比較結束
2	ILCF1	線性比較器 NO.1 (Linear comparator 1) 比較結束
3	IFCF0	FIFO 比較器 NO.0 (FIFO comparator 0) 比較結束
4	IFCF1	FIFO 比較器 NO.1 (FIFO comparator 1) 比較結束
5	IMOV	運動控制迴路超載
6	IHOV	系統工作迴路超載
7	--	保留 (Reserved) , 設定為 0

- Item = 9 : 數位輸入上緣觸發中斷

Bit No.	7	6	5	4	3	2	1	0
Factor	IDIR7	IDIR6	IDIR5	IDIR4	IDIR3	IDIR2	IDIR1	IDIR0
Bit No.	15	14	13	12	11	10	9	8
Factor	IDIR15 (TTL7)	IDIR14 (TTL6)	IDIR13 (TTL5)	IDIR12 (TTL4)	IDIR11 (TTL3)	IDIR10 (TTL2)	IDIR9 (TTL1)	IDIR8 (TTL0)
Bit No.	23	22	21	20	19	18	17	16
Factor	IDIR23 (TTL15)	IDIR22 (TTL14)	IDIR21 (TTL13)	IDIR20 (TTL12)	IDIR19 (TTL11)	IDIR18 (TTL10)	IDIR17 (TTL9)	IDIR16 (TTL8)
Bit No.	31	30	29	28	27	26	25	24
Factor	--	--	--	--	--	--	--	--

- Item = 10 : 數位輸入下緣觸發中斷 :

Bit No.	7	6	5	4	3	2	1	0
Factor	IDIF7	IDIF6	IDIF5	IDIF4	IDIF3	IDIF2	IDIF1	IDIF0
Bit No.	15	14	13	12	11	10	9	8
Factor	IDIF15 (TTL7)	IDIF14 (TTL6)	IDIF13 (TTL5)	IDIF12 (TTL4)	IDIF11 (TTL3)	IDIF10 (TTL2)	IDIF9 (TTL1)	IDIF8 (TTL0)
Bit No.	23	22	21	20	19	18	17	16
Factor	IDIF23 (TTL15)	IDIF22 (TTL14)	IDIF21 (TTL13)	IDIF20 (TTL12)	IDIF19 (TTL11)	IDIF18 (TTL10)	IDIF17 (TTL9)	IDIF16 (TTL8)
Bit No.	31	30	29	28	27	26	25	24
Factor	--	--	--	--	--	--	--	--



偵測數位輸入訊號 (DI) 狀態改變由控制器在每個運動周期內偵測，因此外部輸入訊號變化周期必須大於運動週期才能產生中斷

在 Windows 下使用中斷的方式如下列五大步驟：

1. 設定中斷事件
2. 啟動中斷總開關
3. 等待中斷觸發
4. 重置中斷為觸發狀態
5. 關閉中斷事件和中斷總開關

其相關的 APS APIs 如下列：

```

I32 APS_int_enable( I32 Board_ID, I32 Enable );
I32 APS_set_int_factor( I32 Board_ID, I32 Item_No, I32 Factor_No, Enable );
I32 APS_get_int_factor( I32 Board_ID, I32 Item_No, I32 Factor_No, *Enable );
HANDLE APS_int_no_to_handle( I32 Int_No );
I32 APS_wait_single_int( I32 Int_No, I32 Time_Out );
I32 APS_wait_multiple_int( I32 Int_Count, I32 *Int_No_Array, I32 Wait_All, I32 Time_Out );
I32 APS_reset_int( I32 Int_No );
I32 APS_set_int( I32 Int_No );

```

詳細的使用方式描述如下：

1. 設定中斷事件：

使用 **APS_set_int_factor()** 設定要等待的中斷事件，若成功設定此函式回傳此中斷事件的 Event number，使用者必須使用一變數將 Event number 紀錄待後面步驟 Wait 函式中使用。

依照應用需求 **APS_set_int_factor()** 函式也可以關閉已被開啟的中斷事件。

2. 啟動中斷總開關：

以控制器為單位的中斷裝置總開關，開啟後作業系統才可接收到硬體裝置的中斷信號。使用 **APS_int_enable()** 開啟。

3. 等待中斷觸發

若等待單一中斷事件使用 **APS_wait_single_int()**，或者使用 **APS_wait_multiple_int()** 同時等待數個中斷事件。

當程序進入該函式後會進入睡眠狀態，亦即該程序（或執行緒）不再耗費 CPU 資源直到中斷事件發生或者 Timeout，當 Wait 函式返回後，使用者可透過檢查其回傳值來確認所等待的事件是否發生，進而處理後續的應用流程。

4. 重置中斷為觸發狀態

當事件發生，程序離開 Wait 函式後，該中斷事件會處於被觸發的狀態 (Signaled state)，所以若要重新等待該事件，必須先將該中斷狀態手動重置為未觸發狀態 (Non-signaled state)。若未重置的情況下再次呼叫 Wait 函式，wait 函式會直接返回。重置所使用的函式：**APS_reset_int()**

5. 關閉中斷事件和中斷總開關

最後，若不再使用中斷功能，使用 **APS_set_int_factor()** 函式關閉個別中斷事件，且關閉中斷總開關 **APS_int_enable()** 來釋放所有中斷相關資源。

- 例：

```

void interrupt_example()
{
    // This example shows how interrupt functions work.

    I32 board_id = 0;
    I32 int_no;    // Interrupt number
    I32 return_code; // function return code
    I32 item = 0;  // Axis #0 interrupt
    I32 factor = ( 1 << 12 ); // bit 12 IMDN interrupt

    //Step 1: 設定要等待的中斷事件, factor = IMDN
    int_no = APS_set_int_factor( board_id, item, factor, 1 );
    //Step 2: 設定中斷總開關
    APS_int_enable( board_id, 1 ); // Enable the interrupt main switch
    //Step 3: 等待中斷觸發
    return_code = APS_wait_single_int( int_no, -1 ); //Wait interrupt
    if( return_code == ERR_NoError )
    {
        //Interrupt occurred
        //Step 4: 重置中斷為觸發狀態
        APS_reset_int( int_no );
    }
    // Step 5: 關閉中斷事件和中斷總開關
    APS_set_int_factor( board_id, item, factor, 0 );
    APS_int_enable( board_id, 0 );
}

```

另外，若使用者想要使用 win32 的 Event handle，可在步驟 1 後使用 APS_int_no_to_handle() 將 Event number 轉換成 win32 下的 Event handle 格式

- 例：

```
#include <windows.h> // Using event handle
#include "APS168.h"
#include "ErrorCodeDef.h"

void interrupt_with_win32_example()
{
    // This example shows how interrupt functions work.

    I32 board_id = 0;
    I32 int_no;      // Interrupt number
    DWORD return_code; // function return code
    I32 item = 0;    // Axis #0 interrupt
    I32 factor = ( 1 << 12 ); // bit 12 IMDN interrupt
    HANDLE handle;

    //Step 1: 設定要等待的中斷事件, factor = IMDN
    int_no = APS_set_int_factor( board_id, item, factor, 1 );
    handle = APS_int_no_to_handle( int_no );
    //Step 2: 設定中斷總開關
    APS_int_enable( board_id, 1 ); // Enable the interrupt main switch
    //Step 3: 等待中斷觸發
    return_code = WaitForSingleObject( handle, INFINITE );
    if( return_code == ERR_NoError )
    { //Interrupt occurred
        //Step 4: 重置中斷為為觸發狀態
        ResetEvent( handle );
    }
    // Step 5: 關閉中斷事件和中斷總開關
    APS_set_int_factor( board_id, item, factor, 0 );
    APS_int_enable( board_id, 0 );
}
```

重要安全資訊

為保護使用者安全，處理 / 操作本設備前，請詳閱並遵守本手冊及相關設備上的各項**指令、警告、警戒、及注意事項**。

- ▶ 請詳閱下列安全指令。
- ▶ 請妥善保留本手冊以備將來參考。
- ▶ 請詳閱本手冊規格節內所述有關產品作業環境的說明。
- ▶ 安裝 / 裝載或拆裝 / 拆除設備時：
 - ▷ 關閉電源並拔除任何電源線 / 纜線。
- ▶ 為防電擊及 / 或損壞設備：
 - ▷ 切勿將產品置於水或液體源附近；
 - ▷ 設備務必避開高熱或高濕之處；
 - ▷ 設備務必通風良好（切勿遮蓋通風口）；
 - ▷ 勿必使用指定的電壓及電源設定；
 - ▷ 設備務必在方便可用的電氣插座附近安裝及使用；
 - ▷ 勿必將電源線固定妥當（切勿壓 / 裹任何物件在電源線上）；
 - ▷ 勿必在穩固的表面及 / 或指定的安裝架上安裝 / 附掛及操作設備；
 - ▷ 設備長期不使用時，請關閉電源並拔除設備的電源線。

- ▶ 切勿自行維修本設備。本設備限由合格人員進行維修。
本設備可使用鋰電池提供不中斷、備援或緊急電力。



不當的電池可能有爆炸之虞。廢舊電池請按法規處理。

- ▶ 如有下列情況，本產品限由授權的技術人員處理：
 - ▷ 電源線或插頭受損；
 - ▷ 設備進水；
 - ▷ 設備暴露於高濕 / 高水氣的環境；
 - ▷ 設備無法運作，或不按使用手冊所描述方式運作；
 - ▷ 遭摔落及 / 或損壞；
 - ▷ 有明顯的破損。

服務資訊

如需任何服務或協助請聯繫本公司。

ADLINK Technology, Inc.

Address: 9F, No.166 Jian Yi Road, Zhonghe District
New Taipei City 235, Taiwan
新北市中和區建一路 166 號 9 樓
Tel: +886-2-8226-5877
Fax: +886-2-8226-5717
Email: service@adlinktech.com

Ampro ADLINK Technology, Inc.

Address: 5215 Hellyer Avenue, #110, San Jose, CA 95138, USA
Tel: +1-408-360-0200
Toll Free: +1-800-966-5200 (USA only)
Fax: +1-408-360-0222
Email: info@adlinktech.com

ADLINK Technology (China) Co., Ltd.

Address: 上海市浦东新区张江高科技园区芳春路 300 号 (201203)
300 Fang Chun Rd., Zhangjiang Hi-Tech Park,
Pudong New Area, Shanghai, 201203 China
Tel: +86-21-5132-8988
Fax: +86-21-5132-3588
Email: market@adlinktech.com

ADLINK Technology Beijing

Address: 北京市海淀区上地东路 1 号盈创动力大厦 E 座 801 室(100085)
Rm. 801, Power Creative E, No. 1,
Shang Di East Rd., Beijing, 100085 China
Tel: +86-10-5885-8666
Fax: +86-10-5885-8626
Email: market@adlinktech.com

ADLINK Technology Shenzhen

Address: 深圳市南山区科技园南区高新南七道 数字技术园
A1 栋 2 楼 C 区 (518057)
2F, C Block, Bldg. A1, Cyber-Tech Zone, Gao Xin Ave. Sec. 7,
High-Tech Industrial Park S., Shenzhen, 518054 China
Tel: +86-755-2643-4858
Fax: +86-755-2664-6353
Email: market@adlinktech.com

LiPPERT ADLINK Technology GmbH

Address: Hans-Thoma-Strasse 11, D-68163, Mannheim, Germany
Tel: +49-621-43214-0
Fax: +49-621 43214-30
Email: emea@adlinktech.com

ADLINK Technology, Inc. (French Liaison Office)

Address: 15 rue Emile Baudot, 91300 Massy CEDEX, France
Tel: +33 (0) 1 60 12 35 66
Fax: +33 (0) 1 60 12 35 66
Email: france@adlinktech.com

ADLINK Technology Japan Corporation

Address: 〒101-0045 東京都千代田区神田鍛冶町 3-7-4
神田 374 ビル 4F
KANDA374 Bldg. 4F, 3-7-4 Kanda Kajicho,
Chiyoda-ku, Tokyo 101-0045, Japan
Tel: +81-3-4455-3722
Fax: +81-3-5209-6013
Email: japan@adlinktech.com

ADLINK Technology, Inc. (Korean Liaison Office)

Address: 서울시 서초구 서초동 1675-12 모인터빌딩 8 층
8F Mointer B/D, 1675-12, Seocho-Dong, Seocho-Gu,
Seoul 137-070, Korea
Tel: +82-2-2057-0565
Fax: +82-2-2057-0563
Email: korea@adlinktech.com

ADLINK Technology Singapore Pte. Ltd.

Address: 84 Genting Lane #07-02A, Cityneon Design Centre,
Singapore 349584
Tel: +65-6844-2261
Fax: +65-6844-2263
Email: singapore@adlinktech.com

ADLINK Technology Singapore Pte. Ltd. (Indian Liaison Office)

Address: 1st Floor, #50-56 (Between 16th/17th Cross) Margosa Plaza,
Margosa Main Road, Malleswaram, Bangalore-560055, India
Tel: +91-80-65605817, +91-80-42246107
Fax: +91-80-23464606
Email: india@adlinktech.com

ADLINK Technology, Inc. (Israeli Liaison Office)

Address: 6 Hasadna St., Kfar Saba 44424, Israel
Tel: +972-9-7446541
Fax: +972-9-7446542
Email: israel@adlinktech.com