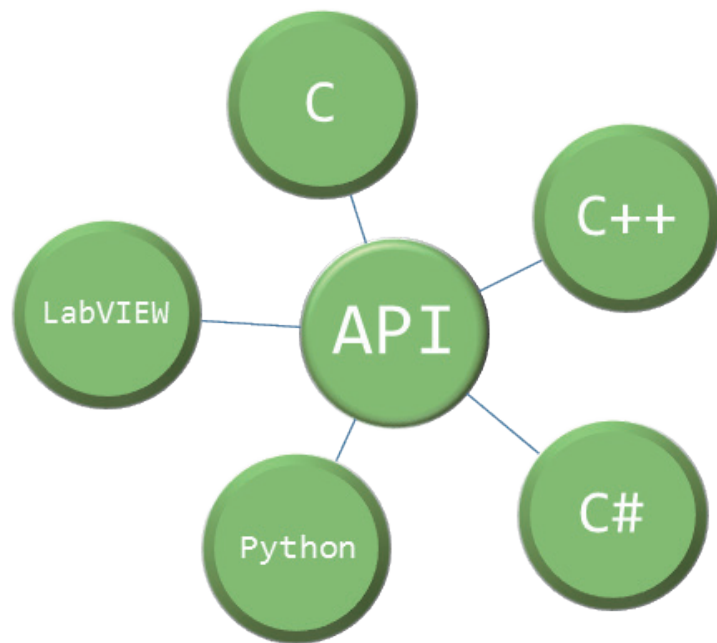




HIMC 系列

API 參考指南

修訂紀錄

手冊版次資訊亦標記於手冊封面右下角。

MH05UC01-2510_V1.2

手冊版次

發行年份與月份

發行日期	版次	適用軟體	更新內容
2025/10/31	1.2	iA Studio 3.3.0	(1) 新增章節： ■ 第 19 章—從站函式 (2) 新增函式： ■ 第 2 章 HIMC_IsConnected HIMC_GetModelNum HIMC_GetSerialNum ■ 第 3 章 HIMC_MovePVT HIMC_GetVel HIMC_GetAcc HIMC_GetDec HIMC_SetPosWindow HIMC_GetPosWindow HIMC_SetPosWindowTime HIMC_GetPosWindowTime ■ 第 6 章 HIMC_GetGrpVel HIMC_GetGrpAcc HIMC_GetGrpDec ■ 第 7 章 HIMC_GetSlvAllGPIByStartIdx HIMC_GetSlvAllGPOByStartIdx ■ 第 11 章 HIMC_EnableTouchProbe1 HIMC_EnableTouchProbe2 HIMC_DisableTouchProbe1 HIMC_DisableTouchProbe2 HIMC_IsTouchProbe1Enabled

發行日期	版次	適用軟體	更新內容
			HIMC_IsTouchProbe2Enabled HIMC_IsTouchProbe1Triggered HIMC_IsTouchProbe2Triggered HIMC_GetTouchProbe1Pos HIMC_GetTouchProbe2Pos HIMC_SetTouchProbe1Func HIMC_SetTouchProbe2Func ■ 第 18 章 HIMC_IsHomeSwitch (3) 函式修訂： ■ 第 2 章 HIMC_GetFirmwareVer HIMC_GetFirmwareDate ■ 第 3 章 HIMC_MoveVel HIMC_SetVel HIMC_SetAcc HIMC_SetDec HIMC_SetOpMode ■ 第 6 章 HIMC_SetGrpMotionProfile HIMC_SetGrpVel HIMC_SetGrpAcc HIMC_SetGrpDec ■ 第 8 章 HIMC_SetSlvAORaw HIMC_GetSlvAIRaw HIMC_GetSlvAORaw ■ 第 16 章 HIMC_ReadSDO HIMC_ReadPDO ■ 第 17 章 HIMC_GetDriveErr (4) 功能說明更新 / 新增： ■ 第 1 章 更新 API 如何運作的說明、更新 API 環境設定、更新版本說明、新增數據型態的說明。 ■ 第 10、11、18 章 虛擬機、虛擬軸不支援位置觸發相關函式、Touch Probe 相關函式、MoveHome 函式。

發行日期	版次	適用軟體	更新內容
			■ 第 16 章 修正變數說明 (HCV_ID_fb_ratio_pos 、 HCV_ID_fb_ratio_cnt 、 HCV_ID_fb_curr_ratio_curr 、 HCV_ID_fb_curr_ratio_cnt) 。 (5) 更新錯誤訊息： 17.1.1 節、17.1.2 節、17.1.3 節、22.1 節
2025/02/28	1.1	iA Studio 3.1.0	(1) 新增函式： ■ 第 2 章 HIMC_IsSystemPreOp HIMC_GetFirmwareDate HIMC_GetFirmwareHash HIMC_GetModelNumber HIMC_GetApiVer ■ 第 3 章 HIMC_GetPosFbComp HIMC_GetBufferMode HIMC_GetCmdNum ■ 第 6 章 HIMC_SetGrpLookAheadPrm HIMC_SetGrpQueueSize ■ 第 10 章 HIMC_SetPT_PosArray HIMC_SetPT_StateArray HIMC_SetPT_StartIndex HIMC_SetPT_EndIndex ■ 第 18 章 HIMC_SetHomedStatus (2) 函式與數據結構修訂： ■ 第 2 章 HIMC_IsSystemOper ■ 第 3 章 HIMC_GetCurrFb - 回傳電流值 HIMC_SetOpMode ■ 第 8 章 HIMC_SetSlvAORaw HIMC_GetSlvAORaw HIMC_GetSlvAIRaw ■ 第 16 章 HIMC_ReadSDO

發行日期	版次	適用軟體	更新內容
			HIMC_WriteSDO HIMC_ReadPDO HIMC_WritePDO HIMC_FroceWritePDO - 參數型態 ■ 第 20 章 MotionBufferMode MotionTransitionMode - 新增參數 (3) 功能說明更新 / 新增： ■ 第 3 章 軸變數新增 CoE 變數。 ■ 第 6 章 更新速度緩衝模式說明、更新路徑過渡模式說明、新增速度前瞻功能。 ■ 第 10 章 新增 Random PT 變數說明與流程。 ■ 第 18 章 更新歸原點方法說明。 ■ 第 19 章 新增數據結構 Data_t。 (4) 新增錯誤訊息： 17.1.1 節、17.1.2 節、17.1.3 節、21.1 節
2023/03/15	1.0	iA Studio 3.0.0	(1) HIMC 支援 CoE 通訊：新增 Read/Write SDO 與 PDO 函式，移除 Get/Set Slave Var 與 Run PDL 相關函式。 (2) 新增章節： ■ 第 18 章—歸原點函式 (3) 新增函式： ■ 第 2 章 HIMC_IsSystemInit HIMC_GetECATSt HIMC_GetSlvECATSt HIMC_ScanNetwork ■ 第 3 章 HIMC_SetOpMode HIMC_SetBufferMode ■ 第 8 章 HIMC_SetSlvAOHex HIMC_GetSlvAOHex HIMC_GetSlvAIHex ■ 第 11 章

發行日期	版次	適用軟體	更新內容
			HIMC_SetTouchProbeFunc ■ 第 16 章 HIMC_GetDriveErr (4) 移除第 10 章中 Random PT 的相關功能。

相關文件

透過相關文件，使用者可快速了解此手冊的定位，以及各手冊、產品之間的關聯性。詳細內容請至本公司官網→下載中心→手冊總覽閱覽 (https://www.hiwinmikro.tw/Downloads/ManualOverview_TC.htm) 。

目錄

1.	序言	1-1
1.1	API 如何運作	1-2
1.2	API 環境設定	1-3
1.2.1	C / C++	1-3
1.2.2	C#	1-3
1.2.3	LabVIEW	1-4
1.2.4	Python	1-4
1.3	版本說明	1-5
1.4	法律免責聲明	1-5
1.5	數據型態	1-6
2.	HIMC 系統函式	2-1
2.1	HIMC_ConnectCtrl	2-2
2.2	HIMC_ConnectToSimulator	2-3
2.3	HIMC_ConnectToEthernet	2-4
2.4	HIMC_DisconnectCtrl	2-5
2.5	HIMC_RebootController	2-6
2.6	HIMC_IsConnected	2-7
2.7	HIMC_IsSystemInit	2-8
2.8	HIMC_IsSystemOper	2-9
2.9	HIMC_IsSystemPreOp	2-10
2.10	HIMC_IsSystemError	2-11
2.11	HIMC_GetECATSt	2-12
2.12	HIMC_GetSlvECATSt	2-13
2.13	HIMC_DisableAll	2-14
2.14	HIMC_StopAll	2-15
2.15	HIMC_EStop	2-16
2.16	HIMC_GetSlaveNum	2-17
2.17	HIMC_GetFirmwareVer	2-18
2.18	HIMC_GetFirmwareDate	2-19
2.19	HIMC_GetFirmwareHash	2-20
2.20	HIMC_GetModelNum	2-21
2.21	HIMC_GetSerialNum	2-22
2.22	HIMC_GetApiVer	2-23
2.23	HIMC_ScanNetwork	2-24
2.24	HIMC_SetEconMode	2-25
2.25	HIMC_IsEconMode	2-26

3.	軸函式	3-1
3.1	概述	3-4
3.1.1	軸變數	3-7
3.2	軸運動控制	3-12
3.2.1	HIMC_Enable	3-12
3.2.2	HIMC_Disable	3-13
3.2.3	HIMC_Reset	3-14
3.2.4	HIMC_MoveAbs	3-15
3.2.5	HIMC_MoveRel	3-16
3.2.6	HIMC_MoveVel	3-17
3.2.7	HIMC_MoveTrq	3-18
3.2.8	HIMC_MovePVT	3-19
3.2.9	HIMC_Stop	3-21
3.2.10	HIMC_Halt	3-22
3.2.11	HIMC_Resume	3-23
3.3	軸設定	3-24
3.3.1	HIMC_GetVel	3-24
3.3.2	HIMC_SetVel	3-25
3.3.3	HIMC_GetAcc	3-26
3.3.4	HIMC_SetAcc	3-27
3.3.5	HIMC_SetAccTime	3-28
3.3.6	HIMC_GetDec	3-29
3.3.7	HIMC_SetDec	3-30
3.3.8	HIMC_SetDecTime	3-31
3.3.9	HIMC_GetKillDec	3-32
3.3.10	HIMC_SetKillDec	3-33
3.3.11	HIMC_GetSWRL	3-34
3.3.12	HIMC_SetSWRL	3-35
3.3.13	HIMC_GetSWLL	3-36
3.3.14	HIMC_SetSWLL	3-37
3.3.15	HIMC_GetSMTIME	3-38
3.3.16	HIMC_SetSMTIME	3-39
3.3.17	HIMC_GetMoveTime	3-40
3.3.18	HIMC_GetSettlingTime	3-41
3.3.19	HIMC_SetPos	3-42
3.3.20	HIMC_GetPosFb	3-43
3.3.21	HIMC_GetPosFbComp	3-44
3.3.22	HIMC_GetPosOffset	3-45
3.3.23	HIMC_GetPosErr	3-46
3.3.24	HIMC_GetVelFb	3-47
3.3.25	HIMC_GetVelErr	3-48
3.3.26	HIMC_GetCurrFb	3-49

3.3.27	HIMC_GetRefPos.....	3-50
3.3.28	HIMC_GetRefVel	3-51
3.3.29	HIMC_GetRefAcc.....	3-52
3.3.30	HIMC_GetPosOut	3-53
3.3.31	HIMC_GetVelOut	3-54
3.3.32	HIMC_GetAccOut	3-55
3.3.33	HIMC_IgnoreHWL	3-56
3.3.34	HIMC_IgnoreSWL	3-57
3.3.35	HIMC_IgnorePE	3-58
3.3.36	HIMC_GetAxisNum	3-59
3.3.37	HIMC_SetVelScale	3-60
3.3.38	HIMC_GetVelScale	3-61
3.3.39	HIMC_SetRollover	3-62
3.3.40	HIMC_GetRolloverTurns	3-64
3.3.41	HIMC_SetOpMode.....	3-65
3.3.42	HIMC_SetBufferMode.....	3-66
3.3.43	HIMC_GetBufferMode	3-67
3.3.44	HIMC_GetCmdNum.....	3-68
3.3.45	HIMC_GetMultiAxesFeedbackPos	3-69
3.3.46	HIMC_SetPosWindow.....	3-70
3.3.47	HIMC_GetPosWindow	3-71
3.3.48	HIMC_SetPosWindowTime.....	3-72
3.3.49	HIMC_GetPosWindowTime.....	3-73
3.4	軸狀態	3-74
3.4.1	HIMC_IsEnabled	3-74
3.4.2	HIMC_IsMoving	3-75
3.4.3	HIMC_IsInPos	3-76
3.4.4	HIMC_IsErrorStop	3-77
3.4.5	HIMC_IsGantry	3-78
3.4.6	HIMC_IsGrouped	3-79
3.4.7	HIMC_IsSync	3-80
3.4.8	HIMC_IsHWLL.....	3-81
3.4.9	HIMC_IsHWRL	3-82
3.4.10	HIMC_IsSWLL	3-83
3.4.11	HIMC_IsSWRL.....	3-84
3.4.12	HIMC_IsDriveErr	3-85
3.4.13	HIMC_IsPosErr.....	3-86
3.4.14	HIMC_IsCompActive.....	3-87
3.4.15	HIMC_IsAcc	3-88
4.	同步運動函式	4-1
4.1	概述	4-2

4.1.1	同步運動變數	4-3
4.2	HIMC_EnableGear	4-4
4.3	HIMC_DisableGear	4-5
4.4	HIMC_GearIn	4-6
4.5	HIMC_GearOut	4-7
4.6	HIMC_GetGearRatio	4-8
4.7	HIMC_IsInGear	4-9
4.8	HIMC_IsGearMaster	4-10
4.9	HIMC_IsGearSlave	4-11
5.	龍門函式	5-1
5.1	概述	5-2
5.2	HIMC_EnableGantryPair	5-3
5.3	HIMC_DisableGantryPair	5-4
5.4	HIMC_GetGantryPairID	5-5
5.5	HIMC_IsGantryPair	5-6
6.	軸群組函式	6-1
6.1	概述	6-3
6.1.1	軸群組變數	6-6
6.1.2	座標系統	6-9
6.1.3	運動學	6-13
6.1.4	速度緩衝模式	6-13
6.1.5	路徑過渡模式	6-16
6.2	軸群組運動控制	6-18
6.2.1	HIMC_EnableGroup	6-18
6.2.2	HIMC_DisableGroup	6-19
6.2.3	HIMC_ResetGroup	6-20
6.2.4	HIMC_StopGroup	6-21
6.2.5	HIMC_HaltGroup	6-22
6.2.6	HIMC_ResumeGroup	6-23
6.2.7	HIMC_JogGroup	6-24
6.2.8	HIMC_JogGroupAxis	6-25
6.2.9	HIMC_LineAbs2D	6-26
6.2.10	HIMC_LineAbs3D	6-27
6.2.11	HIMC_LineRel2D	6-28
6.2.12	HIMC_LineRel3D	6-29
6.2.13	HIMC_Arc2D	6-30
6.2.14	HIMC_ArcCW2D	6-32
6.2.15	HIMC_ArcCCW2D	6-33
6.2.16	HIMC_ArcAngle2D	6-34
6.2.17	HIMC_Circle2D	6-36

6.3	軸群組設定	6-38
6.3.1	HIMC_AddAxesToGrp	6-38
6.3.2	HIMC_RemoveAxisFromGrp	6-39
6.3.3	HIMC_SetupGroup	6-40
6.3.4	HIMC_UngrpAllAxes	6-41
6.3.5	HIMC_GetGroupID	6-42
6.3.6	HIMC_SetGrpMotionProfile	6-43
6.3.7	HIMC_SetGrpAngMotionProfile	6-45
6.3.8	HIMC_GetGrpKin	6-47
6.3.9	HIMC_SetGrpKin	6-48
6.3.10	HIMC_GetGrpVel	6-49
6.3.11	HIMC_SetGrpVel	6-50
6.3.12	HIMC_GetGrpAcc	6-51
6.3.13	HIMC_SetGrpAcc	6-52
6.3.14	HIMC_SetGrpAccTime	6-53
6.3.15	HIMC_GetGrpDec	6-54
6.3.16	HIMC_SetGrpDec	6-55
6.3.17	HIMC_SetGrpDecTime	6-56
6.3.18	HIMC_GetGrpSMTime	6-57
6.3.19	HIMC_SetGrpSMTime	6-58
6.3.20	HIMC_GetGrpCoordSys	6-59
6.3.21	HIMC_SetGrpCoordSys	6-60
6.3.22	HIMC_GetGrpBufferMode	6-61
6.3.23	HIMC_SetGrpBufferMode	6-62
6.3.24	HIMC_GetGrpTransMode	6-63
6.3.25	HIMC_SetGrpTransMode	6-64
6.3.26	HIMC_SetGrpTransPrm	6-65
6.3.27	HIMC_GetGrpCmdNum	6-67
6.3.28	HIMC_SetGrpVelScale	6-68
6.3.29	HIMC_GetGrpVelScale	6-69
6.3.30	HIMC_GetGrpCoordTrans	6-70
6.3.31	HIMC_SetGrpCoordTrans	6-71
6.3.32	HIMC_GetGrpPoseCmd	6-72
6.3.33	HIMC_GetGrpPoseFb	6-73
6.3.34	HIMC_SetGrpLookAheadPrm	6-74
6.3.35	HIMC_SetGrpQueueSize	6-75
6.4	軸群組狀態	6-76
6.4.1	HIMC_IsGrpEnabled	6-76
6.4.2	HIMC_IsGrpMoving	6-77
6.4.3	HIMC_IsGrpInPos	6-78
6.4.4	HIMC_IsGrpErrorStop	6-79
6.5	進階軸群組運動控制	6-80

6.5.1	HIMC_LineAbs.....	6-80
6.5.2	HIMC_LineRel	6-82
6.5.3	HIMC_CircleAbs.....	6-84
6.5.4	HIMC_CircleRel	6-86
7.	GPIO 函式.....	7-1
7.1	概述	7-2
7.1.1	GPIO 變數	7-2
7.2	控制器 IO 設定	7-3
7.2.1	HIMC_SetGPO	7-3
7.2.2	HIMC_ToggleGPO	7-4
7.2.3	HIMC_SetAllGPO.....	7-5
7.2.4	HIMC_SetGPInvert	7-6
7.2.5	HIMC_SetGPOInvert	7-7
7.2.6	HIMC_BindEMO.....	7-8
7.3	從站 IO 設定.....	7-9
7.3.1	HIMC_SetSlvGPO	7-9
7.3.2	HIMC_ToggleSlvGPO	7-11
7.3.3	HIMC_SetSlvAllGPO.....	7-12
7.4	控制器 IO 狀態	7-13
7.4.1	HIMC_GetGPI.....	7-13
7.4.2	HIMC_GetGPO	7-14
7.4.3	HIMC_GetAllGPI	7-15
7.4.4	HIMC_GetAllGPO	7-16
7.5	從站 IO 狀態.....	7-17
7.5.1	HIMC_GetSlvGPI.....	7-17
7.5.2	HIMC_GetSlvGPO	7-18
7.5.3	HIMC_GetSlvAllGPI	7-19
7.5.4	HIMC_GetSlvAllGPO	7-20
7.5.5	HIMC_GetSlvAllGPIByStartIdx	7-21
7.5.6	HIMC_GetSlvAllGPOByStartIdx	7-23
8.	AIO 函式	8-1
8.1	概述	8-2
8.2	從站 AIO 設定.....	8-3
8.2.1	HIMC_SetSlvAType	8-3
8.2.2	HIMC_SetSlvAOType.....	8-4
8.2.3	HIMC_SetSlvAORaw	8-5
8.2.4	HIMC_SetSlvAO	8-6
8.3	從站 AIO 狀態.....	8-7
8.3.1	HIMC_GetSlvAType.....	8-7
8.3.2	HIMC_GetSlvAOType	8-8

8.3.3	HIMC_GetSlvAIRaw.....	8-9
8.3.4	HIMC_GetSlvAI.....	8-10
8.3.5	HIMC_GetSlvAORaw.....	8-11
8.3.6	HIMC_GetSlvAO	8-12
8.4	從站 AO 綁定 HIMC 內部記憶體變數	8-13
8.4.1	HIMC_SetSlvAOMonitor	8-13
8.4.2	HIMC_SetSlvAOParam.....	8-15
8.4.3	HIMC_GetSlvAOScale.....	8-17
8.4.4	HIMC_GetSlvAOOffset.....	8-18
8.4.5	HIMC_IsSlvAOBound.....	8-19
9.	User Table 函式.....	9-1
9.1	概述	9-2
9.2	HIMC_SetUserTable	9-3
9.3	HIMC_GetUserTable.....	9-4
9.4	HIMC_SetTableValue	9-5
9.5	HIMC_GetTableValue	9-6
9.6	HIMC_SaveUserTable	9-7
9.7	HIMC_LoadUserTable.....	9-8
10.	位置觸發函式	10-1
10.1	概述	10-2
10.1.1	PT 變數.....	10-2
10.1.2	PT 功能使用流程	10-4
10.2	HIMC_EnablePT.....	10-6
10.3	HIMC_DisablePT	10-7
10.4	HIMC_IsPTEnabled.....	10-8
10.5	HIMC_SetPosTriggerConfig.....	10-9
10.6	HIMC_SetPT_PosArray.....	10-10
10.7	HIMC_SetPT_StateArray	10-11
10.8	HIMC_SetPT_StartIndex.....	10-12
10.9	HIMC_SetPT_EndIndex.....	10-13
11.	Touch Probe 函式.....	11-1
11.1	概述	11-2
11.2	HIMC_EnableTouchProbe1	11-3
11.3	HIMC_EnableTouchProbe2	11-4
11.4	HIMC_DisableTouchProbe1.....	11-5
11.5	HIMC_DisableTouchProbe2.....	11-6
11.6	HIMC_IsTouchProbe1Enabled	11-7
11.7	HIMC_IsTouchProbe2Enabled	11-8
11.8	HIMC_IsTouchProbe1Triggered.....	11-9

11.9	HIMC_IsTouchProbe2Triggered.....	11-10
11.10	HIMC_GetTouchProbe1Pos	11-11
11.11	HIMC_GetTouchProbe2Pos	11-12
11.12	HIMC_SetTouchProbe1Func.....	11-13
11.13	HIMC_SetTouchProbe2Func.....	11-15
12.	動態誤差補償函式.....	12-1
12.1	概述	12-2
12.2	HIMC_EnableComp	12-4
12.3	HIMC_DisableComp	12-5
12.4	HIMC_SetupComp	12-6
12.5	HIMC_SetupComp2D	12-8
12.6	HIMC_SetupComp3D	12-10
12.7	HIMC_GetCompPos	12-12
12.8	HIMC_SetCompAlgType	12-13
13.	濾波器函式.....	13-1
13.1	概述	13-2
13.2	HIMC_EnableAxisVsf.....	13-3
13.3	HIMC_DisableAxisVsf.....	13-4
13.4	HIMC_SetAxisVsf	13-5
13.5	HIMC_EnableAxisInShape.....	13-7
13.6	HIMC_DisableAxisInShape	13-8
13.7	HIMC_SetAxisInShape.....	13-9
13.8	HIMC_EnableGrpInShape	13-11
13.9	HIMC_DisableGrpInShape.....	13-12
13.10	HIMC_SetGrpInShape.....	13-13
14.	HMPL Task 函式	14-1
14.1	概述	14-2
14.2	HIMC_StartTask.....	14-3
14.3	HIMC_StartTaskFunc.....	14-4
14.4	HIMC_StopTask.....	14-5
14.5	HIMC_StopAllTask	14-6
14.6	HIMC_IsTaskStop	14-7
14.7	HIMC_LoadHMPLTask.....	14-8
15.	回調函式	15-1
15.1	HIMC_SetHmplEvtCallback	15-2
15.2	HIMC_SetErrorCallback.....	15-3
15.3	HIMC_SetHmplMsgEvtCallback	15-4

16.	變數與函式操作函式	16-1
16.1	概述	16-2
16.1.1	控制器變數列表	16-3
16.2	驅動器變數操作.....	16-7
16.2.1	HIMC_ReadSDO	16-7
16.2.2	HIMC_WriteSDO	16-9
16.2.3	HIMC_ReadPDO	16-11
16.2.4	HIMC_WritePDO	16-12
16.2.5	HIMC_ForceWritePDO	16-13
16.2.6	HIMC_ReleasePDO	16-14
16.3	控制器變數操作.....	16-15
16.3.1	HIMC_GetVariableByID	16-15
16.3.2	HIMC_SetVariableByID	16-16
16.3.3	HIMC_GetVariableListByID	16-17
16.3.4	HIMC_SetVariableListByID	16-18
16.3.5	HIMC_GetGlobalVariables	16-19
16.3.6	HIMC_SetGlobalVariables	16-20
17.	HIMC 錯誤函式	17-1
17.1	概述	17-2
17.1.1	系統錯誤訊息	17-3
17.1.2	軸錯誤訊息.....	17-6
17.1.3	軸群組錯誤訊息	17-10
17.2	HIMC_GetLastError	17-12
17.3	HIMC_GetAxisLastErr	17-13
17.4	HIMC_ClearAxisLastErr	17-14
17.5	HIMC_GetGrpLastErr.....	17-15
17.6	HIMC_ClearGrpLastErr	17-16
17.7	HIMC_GetDriveErr.....	17-17
17.8	HIMC_GetErrorInformation.....	17-18
18.	歸原點函式.....	18-1
18.1	概述	18-2
18.2	HIMC_MoveHome	18-7
18.3	HIMC_SetHomeMethod	18-8
18.4	HIMC_SetHomeSwitchVel	18-9
18.5	HIMC_SetHomeZeroVel	18-10
18.6	HIMC_SetHomeAcc	18-11
18.7	HIMC_SetHomeOffset.....	18-12
18.8	HIMC_SetHomeTimeout	18-13
18.9	HIMC_SetHomedStatus.....	18-14
18.10	HIMC_IsHomed.....	18-15

18.11	HIMC_IsHoming.....	18-16
18.12	HIMC_IsHomeSwitch	18-17
19.	從站函式	19-1
19.1	HIMC_GetSlvChNum	19-2
19.2	HIMC_GetSlvSlotNum	19-3
20.	數據結構	20-1
20.1	ComInfo	20-2
20.2	CoordPosition.....	20-4
20.3	MotionProfile	20-5
20.4	CenterPosition.....	20-7
20.5	NormalVector	20-8
20.6	TransPrm.....	20-9
20.7	PosTriggerPar	20-10
20.8	Data_t.....	20-11
21.	列舉	21-1
21.1	ComType.....	21-2
21.2	CoordSystem	21-3
21.3	MotionBufferMode	21-5
21.4	MotionTransitionMode	21-6
21.5	ShaperMode.....	21-7
22.	附錄	22-1
22.1	API 錯誤代碼.....	22-2

(此頁有意留白。)

1. 序言

1.	序言	1-1
1.1	API 如何運作	1-2
1.2	API 環境設定	1-3
1.2.1	C / C++	1-3
1.2.2	C#	1-3
1.2.3	LabVIEW	1-4
1.2.4	Python	1-4
1.3	版本說明	1-5
1.4	法律免責聲明	1-5
1.5	數據型態	1-6

1.1 API 如何運作

HIMC 系列 API 是以 C/C++ 語言開發的動態函式庫 (DLL)，提供一組應用程式介面 (API)，協助使用者與 HIMC^{*1} 功能進行互動。使用者可透過匯入 DLL 並呼叫其提供的 API 函式，撰寫應用所需的運動控制邏輯與程式，以實現運動控制、狀態讀取與參數設定等操作。此參考指南裡所介紹的函式以 C/C++ 為主，欲使用其他 API 環境，請參考該語言的 DLL 呼叫機制，對應函式名稱請參閱各小節的說明。

HIMC 系列 API 提供 32-bit 與 64-bit 兩種版本，請依應用程式編譯架構選擇對應的版本。若執行環境與 DLL 架構不符，將導致載入失敗或執行錯誤。

- 32-bit：<安裝目錄>\HIMC_API.dll
- 64-bit：<安裝目錄>\x64_api\HIMC_API.dll

註：

*1：包含以 HIMC 系列命名的機種。

1.2 API 環境設定

執行 API 前，使用環境需安裝 Visual Studio 2012(VC++ 11.0)的轉發套件(Redistributable Package)，安裝檔路徑如下：

- 32-bit：<安裝目錄>\vcredist_x86.exe
- 64-bit：<安裝目錄>\x64_api\vcredist_x64.exe

1.2.1 C / C++

1. 先至<安裝目錄>\examples\API\cpp\vs_project。
2. 執行 copy_required_file.bat。
3. 開啟專案檔 api_example.sln。

環境關聯檔案

Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll、msvcpr110.dll、msvcr110.dll
HMPL 函式檔	HMPL_compiler.exe

1.2.2 C#

1. 先至<安裝目錄>\examples\API\c_sharp。
2. 執行 copy_required_file.bat。
3. 開啟專案檔 api_example.sln。

環境關聯檔案

C# Interface Definition	HIMC_API.cs
DLL	HIMC_API.dll、msvcpr110.dll、msvcr110.dll
EXE	HMPL_compiler.exe

1.2.3 LabVIEW

1. 先至<安裝目錄>\examples\API\labview。
2. 執行 copy_required_file.bat。
3. 開啟專案檔 example.lvproj。

環境關聯檔案

LabVIEW 虛擬儀器檔	<安裝目錄>\examples\API\labview\lib\Vis\#.vi
DLL	HIMC_API.dll、msvcp110.dll、msvcr110.dll
EXE	HMPL_compiler.exe

1.2.4 Python

1. 先至<安裝目錄>\examples\API\python。
2. 執行 copy_required_file.bat。
3. 可直接執行範例 python example.py。

環境關聯檔案

Python Interface Definition	himc_api.py
DLL	HIMC_API.dll、msvcp110.dll、msvcr110.dll
EXE	HMPL_compiler.exe

1.3 版本說明

搭配 CoE 通訊的 HIMC 控制器（產品型號 MC-XX-XX-01-XX），需使用 iA Studio 3.0 以上的軟體版本；
搭配 MoE 通訊的 HIMC 控制器（產品型號 MC-XX-01-00-XX），則需使用 iA Studio 2.X 以下的軟體版本。
請依照下表使用對應的軟體版本。

表 1.3.1 軟體版本對應控制器產品

軟體版本	控制器		通訊格式	產品型號
iA Studio 3.2 以上		HIMC3	EtherCAT	MC-XX-03-01-XX
iA Studio 3.1 iA Studio 3.0		HIMC	EtherCAT	MC-XX-01-01-XX
iA Studio 2.X 以下			mega-ulink	MC-XX-01-00-XX

註：

iA Studio 1.3 以上所採用的運動變數單位：線性運動（mm）、旋轉運動（deg）、時間（ms）；

iA Studio 1.2 以下所採用的運動變數單位：線性運動（m）、旋轉運動（rad）、時間（s）。

1.4 法律免責聲明

使用者可因特定用途，採用或修改本指南所提供的任一示例代碼。但是，在不同的應用場景下，無法保證其正確性、有效性及安全性。使用者須為軟體執行的安全性及有效性負全責。

1.5 數據型態

數據型態用於宣告變數或取得函式之回傳值。變數的型態決定其佔據儲存空間的容量及有效值。做數據型態的轉換時，數值必須落在目標資料類型的範圍內，否則會發生溢位或截斷，導致數值被更改。

表 1.5.1

型態	描述	位元組 (Byte)	有效值
char int8_t	8 位元整數	1	-128 ~ 127
unsigned char uint8_t	8 位元正整數	1	0 ~ 255
short int16_t	16 位元整數	2	-32768 ~ 32767
unsigned short uint16_t	16 位元正整數	2	0 ~ 65535
int int32_t	32 位元整數	4	-2147483648 ~ 2147483647
unsigned int uint32_t	32 位元正整數	4	0 ~ 4294967295
long long int64_t	64 位元整數	8	-9223372036854775808 ~ 9223372036854775807
unsigned long long uint64_t	64 位元正整數	8	0 ~ 18446744073709551615
float	32 位元浮點數型 (精確到小數點後 6 位)	4	1.17549e-38 ~ 3.40282e+38
double	64 位元浮點數型 (精確到小數點後 15 位)	8	2.225074e-308 ~ 1.797693e+308
int* char* double* ...	指標型態，含特定型態變數之儲存位置地址。	8	N/A
void	具 void 回傳型態的函式不回傳任何值。	N/A	N/A
void*	泛指指標型態，含任意型態變數之儲存位置地址。	8	N/A

範例

```
void main()
{
    // 數值截斷
    int64_t i64 = 70000;
    int16_t i16 = (int16_t)i64;
    printf("%d", i16);    // 4464

    // 數值溢位
    int64_t i64 = -1;
    uint16_t u16 = (uint16_t)i64;
    printf("%d", u16);    // 65535
}
```

(此頁有意留白。)

2. HIMC 系統函式

2.	HIMC 系統函式.....	2-1
2.1	HIMC_ConnectCtrl	2-2
2.2	HIMC_ConnectToSimulator	2-3
2.3	HIMC_ConnectToEthernet.....	2-4
2.4	HIMC_DisconnectCtrl	2-5
2.5	HIMC_RebootController.....	2-6
2.6	HIMC_IsConnected.....	2-7
2.7	HIMC_IsSystemInit.....	2-8
2.8	HIMC_IsSystemOper	2-9
2.9	HIMC_IsSystemPreOp	2-10
2.10	HIMC_IsSystemError	2-11
2.11	HIMC_GetECATSt	2-12
2.12	HIMC_GetSlvECATSt	2-13
2.13	HIMC_DisableAll.....	2-14
2.14	HIMC_StopAll.....	2-15
2.15	HIMC_EStop.....	2-16
2.16	HIMC_GetSlaveNum	2-17
2.17	HIMC_GetFirmwareVer	2-18
2.18	HIMC_GetFirmwareDate	2-19
2.19	HIMC_GetFirmwareHash.....	2-20
2.20	HIMC_GetModelNum.....	2-21
2.21	HIMC_GetSerialNum.....	2-22
2.22	HIMC_GetApiVer	2-23
2.23	HIMC_ScanNetwork	2-24
2.24	HIMC_SetEconMode.....	2-25
2.25	HIMC_IsEconMode	2-26

2.1 HIMC_ConnectCtrl

用途

建立控制器的連線。

語法

```
int HIMC_ConnectCtrl(  
    const ComInfo com_info,  
    int *p_ctrl_id  
);
```

參數

com_info [in] 儲存連線資訊的結構。

p_ctrl_id [out] 指標型態的記憶體，用來儲存連線成功的控制器 ID。
可透過其他 API 函式傳入此 ID 來操作該連線中的控制器。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

建議使用 HIMC_ConnectToSimulator 或 HIMC_ConnectToEthernet 連線至 HIMC。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ConnectCtrl
LabVIEW	-
Python	-

2.2 HIMC_ConnectToSimulator

用途

建立虛擬機的連線。

語法

```
int HIMC_ConnectToSimulator(  
    int *p_ctrl_id  
);
```

參數

p_ctrl_id [out] 指標型態的記憶體，用來儲存連線成功的控制器 ID。
可透過其他 API 函式傳入此 ID 來操作該連線中的控制器。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ConnectToSimulator
LabVIEW	HIMC Connect To Simulator.vi
Python	宣告 HimcAPI("Simulator")

2.3 HIMC_ConnectToEthernet

用途

建立 Ethernet 的連線。

語法

```
int HIMC_ConnectToEthernet(
    const char *ip_address,
    const char *port,
    int *p_ctrl_id
);
```

參數

ip_address [in] IP 位址的字串。

port [in] 埠的字串。

p_ctrl_id [out] 指標型態的記憶體，用來儲存連線成功的控制器 ID。
可透過其他 API 函式傳入此 ID 來操作該連線中的控制器。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ConnectToEthernet
LabVIEW	HIMC Connect To Ethernet.vi
Python	宣告 HimcAPI("Ethernet")

2.4 HIMC_DisconnectCtrl

用途

切斷控制器的連線。

語法

```
int HIMC_DisconnectCtrl(  
    int ctrl_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisconnectCtrl
LabVIEW	HIMC Disconnect Ctrl.vi
Python	當 class 被刪除即斷線

2.5 HIMC_RebootController

用途

重新啟動控制器。

語法

```
int HIMC_RebootController(
    int ctrl_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_RebootController
LabVIEW	HIMC Reboot Controller.vi
Python	RebootController

2.6 HIMC_IsConnected

用途

詢問控制器是否處於連線 (connected) 狀態。

語法

```
bool HIMC_IsConnected(  
    int ctrl_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

回傳值

若控制器處於連線狀態，將回傳 **bool** 型態的值 **true**。若控制器處於其他狀態，則回傳 **bool** 型態的值 **false**。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsConnected
LabVIEW	-
Python	IsConnected

2.7 HIMC_IsSystemInit

用途

詢問 HIMC 系統是否處於初始 (initializing) 狀態。

語法

```
int HIMC_IsSystemInit(
    int ctrl_id,
    int *p_is_init
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_is_init [out] 指標型態的記憶體，用來儲存 HIMC 系統的初始狀態。
 若 HIMC 系統處於初始狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsSystemInit
LabVIEW	HIMC Is System Init.vi
Python	IsSystemInit

2.8 HIMC_IsSystemOper

用途

詢問 HIMC 系統是否處於運行 (operation) 狀態。若是，則可使用運動與 PDO 相關的函式。

語法

```
int HIMC_IsSystemOper(  
    int ctrl_id,  
    int *p_is_op  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_is_op [out] 指標型態的記憶體，用來儲存 HIMC 系統狀態。
 若 HIMC 系統處於運行狀態且 EtherCAT 通訊處於運行狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.0.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsSystemOper
LabVIEW	HIMC Is System Oper.vi
Python	IsSystemOper

2.9 HIMC_IsSystemPreOp

用途

詢問 HIMC 系統是否處於預運行 (pre-operation) 狀態。若是，HIMC 與從站間的連線已建立完成，但無法使用運動與 PDO 相關的函式。

語法

```
int HIMC_IsSystemPreOp(
    int ctrl_id,
    int *p_is_preop
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。

須透過呼叫 HIMC_ConnectCtrl 函式來取得。

p_is_preop [out] 指標型態的記憶體，用來儲存 HIMC 系統狀態。

若 HIMC 系統處於運行狀態且 EtherCAT 通訊處於預運行狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsSystemPreOp
LabVIEW	HIMC Is System Pre Op.vi
Python	IsSystemPreOp

2.10 HIMC_IsSystemError

用途

詢問 HIMC 系統是否處於錯誤 (error) 狀態。

語法

```
int HIMC_IsSystemError(  
    int ctrl_id,  
    int *p_is_error  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_is_error [out] 指標型態的記憶體，用來儲存 HIMC 系統的錯誤狀態。
 若 HIMC 系統處於錯誤狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsSystemError
LabVIEW	HIMC Is System Error.vi
Python	IsSystemError

2.11 HIMC_GetECATSt

用途

取得控制器的通訊狀態 (EtherCAT State Machine) 。

語法

```
int HIMC_GetECATSt(
    int ctrl_id,
    int *p_state
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID 。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得 。
- p_state [out] 指標型態的記憶體，用來儲存控制器通訊狀態 。
- 1 : Init、2 : Pre-Operation、4 : Safe-Operation、8 : Operation

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0** 。若失敗，則回傳**非零值** 。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetECATSt
LabVIEW	HIMC Get ECAT St.vi
Python	GetECATSt

2.12 HIMC_GetSlvECATSt

用途

取得從站的通訊狀態（EtherCAT State Machine）。

語法

```
int HIMC_GetSlvECATSt(
    int ctrl_id,
    int slv_id,
    int *p_state
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_id [in] 從站編號。
- p_state [out] 指標型態的記憶體，用來儲存從站通訊狀態。
 1：Init、2：Pre-Operation、4：Safe-Operation、8：Operation

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvECATSt
LabVIEW	HIMC Get Slv ECAT St.vi
Python	GetSlvECATSt

2.13 HIMC_DisableAll

用途

解激磁全部的軸和軸群組。

語法

```
int HIMC_DisableAll(
    int ctrl_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

此函式會清除軸和軸群組原有的路徑規畫。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisableAll
LabVIEW	HIMC Disable All.vi
Python	DisableAll

2.14 HIMC_StopAll

用途

以緊急減速度停止全部的軸和軸群組，並讓軸和軸群組維持在激磁狀態。

語法

```
int HIMC_StopAll(  
    int ctrl_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

此函式會清除軸和軸群組原有的路徑規劃。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_StopAll
LabVIEW	HIMC Stop All.vi
Python	StopAll

2.15 HIMC_EStop

用途

緊急停止控制器內部的所有執程序 (包含全部的 HMPL task) ，並解激磁全部的軸和軸群組。

語法

```
int HIMC_EStop(
    int ctrl_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID 。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0** 。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_EStop
LabVIEW	HIMC E Stop.vi
Python	EStop

2.16 HIMC_GetSlaveNum

用途

取得連接至控制器的從站數量。

語法

```
int HIMC_GetSlaveNum(
    int ctrl_id,
    int *p_slv_num
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

p_slv_num [out] 指標型態的記憶體，用來儲存連接至控制器的從站數量。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlaveNum
LabVIEW	HIMC Get Slave Num.vi
Python	GetSlaveNum

2.17 HIMC_GetFirmwareVer

用途

取得控制器的韌體版本。

語法

```
int HIMC_GetFirmwareVer(  
    int ctrl_id,  
    char *p_ver_buf  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_ver_buf [out] 指標型態的記憶體，用來儲存回傳的韌體版本字串。
 若為 N/A，表示讀取失敗或無法取得資料。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetFirmwareVer
LabVIEW	HIMC Get Firmware Ver.vi
Python	GetFirmwareVer

2.18 HIMC_GetFirmwareDate

用途

取得控制器的韌體檔案日期。

語法

```
int HIMC_GetFirmwareDate(  
    int ctrl_id,  
    char *p_date_buf  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_date_buf [out] 指標型態的記憶體，用來儲存回傳的韌體檔案日期字串。
 若為 N/A，表示讀取失敗或無法取得資料。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetFirmwareDate
LabVIEW	-
Python	GetFirmwareDate

2.19 HIMC_GetFirmwareHash

用途

取得控制器的韌體 Hash ID。

語法

```
int HIMC_GetFirmwareHash(  
    int ctrl_id,  
    uint64_t *p_hash  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

p_hash [out] 指標型態的記憶體，用來儲存回傳的韌體 Hash ID。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetFirmwareHash
LabVIEW	-
Python	GetFirmwareHash

2.20 HIMC_GetModelNum

用途

取得控制器的產品型號。

語法

```
int HIMC_GetModelNum(  
    int ctrl_id,  
    char *p_mod_buf  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_mod_buf [out] 指標型態的記憶體，用來儲存回傳的產品型號字串。
 若為 N/A，表示讀取失敗或無法取得資料。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetModelNum
LabVIEW	-
Python	GetModelNum

2.21 HIMC_GetSerialNum

用途

取得控制器的產品序號。

語法

```
int HIMC_GetSerialNum(
    int ctrl_id,
    char *p_sn_buf
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
p_sn_buf [out]	指標型態的記憶體，用來儲存回傳的產品序號字串。 若為 N/A，表示讀取失敗或無法取得資料。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSerialNum
LabVIEW	-
Python	GetSerialNum

2.22 HIMC_GetApiVer

用途

取得 API 的軟體版本。

語法

```
int HIMC_GetApiVer(  
    char *p_ver_buf  
);
```

參數

p_ver_buf [out] 指標型態的記憶體，用來儲存回傳的 API 軟體版本字串。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetApiVer
LabVIEW	-
Python	GetApiVer

2.23 HIMC_ScanNetwork

用途

重新掃描控制器與從站的連線。

語法

```
int HIMC_ScanNetwork(  
    int ctrl_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ScanNetwork
LabVIEW	HIMC Scan Network.vi
Python	ScanNetwork

2.24 HIMC_SetEconMode

用途

設置 HIMC API 效能模式。

語法

```
int HIMC_SetEconMode(
    int ctrl_id,
    bool mode
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

mode [in] HIMC API 效能模式。
0：高效能模式（預設值）
1：節能模式

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

- (1) 高效能模式有較快的 HIMC API 平均響應速度，但 CPU 使用率較節能模式高。
- (2) 節能模式可降低 CPU 使用率，但會增加 HIMC API 平均響應時間。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetEconMode
LabVIEW	HIMC Set Econ Mode.vi
Python	SetEconMode

2.25 HIMC_IsEconMode

用途

詢問 HIMC API 效能模式是否為節能模式。

語法

```
int HIMC_IsEconMode(  
    int ctrl_id,  
    bool *p_is_econ  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_is_econ [out] 指標型態的記憶體，用來儲存 HIMC API 效能模式。
 若為節能模式，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsEconMode
LabVIEW	HIMC Is Econ Mode.vi
Python	IsEconMode

3. 軸函式

3.	軸函式	3-1
3.1	概述	3-4
3.1.1	軸變數	3-7
3.2	軸運動控制	3-12
3.2.1	HIMC_Enable	3-12
3.2.2	HIMC_Disable	3-13
3.2.3	HIMC_Reset	3-14
3.2.4	HIMC_MoveAbs	3-15
3.2.5	HIMC_MoveRel	3-16
3.2.6	HIMC_MoveVel	3-17
3.2.7	HIMC_MoveTrq	3-18
3.2.8	HIMC_MovePVT	3-19
3.2.9	HIMC_Stop	3-21
3.2.10	HIMC_Halt	3-22
3.2.11	HIMC_Resume	3-23
3.3	軸設定	3-24
3.3.1	HIMC_GetVel	3-24
3.3.2	HIMC_SetVel	3-25
3.3.3	HIMC_GetAcc	3-26
3.3.4	HIMC_SetAcc	3-27
3.3.5	HIMC_SetAccTime	3-28
3.3.6	HIMC_GetDec	3-29
3.3.7	HIMC_SetDec	3-30
3.3.8	HIMC_SetDecTime	3-31
3.3.9	HIMC_GetKillDec	3-32
3.3.10	HIMC_SetKillDec	3-33
3.3.11	HIMC_GetSWRL	3-34
3.3.12	HIMC_SetSWRL	3-35
3.3.13	HIMC_GetSWLL	3-36
3.3.14	HIMC_SetSWLL	3-37
3.3.15	HIMC_GetSMTTime	3-38
3.3.16	HIMC_SetSMTTime	3-39
3.3.17	HIMC_GetMoveTime	3-40
3.3.18	HIMC_GetSettlingTime	3-41
3.3.19	HIMC_SetPos	3-42
3.3.20	HIMC_GetPosFb	3-43
3.3.21	HIMC_GetPosFbComp	3-44
3.3.22	HIMC_GetPosOffset	3-45

3.3.23	HIMC_GetPosErr.....	3-46
3.3.24	HIMC_GetVelFb.....	3-47
3.3.25	HIMC_GetVelErr.....	3-48
3.3.26	HIMC_GetCurrFb.....	3-49
3.3.27	HIMC_GetRefPos.....	3-50
3.3.28	HIMC_GetRefVel.....	3-51
3.3.29	HIMC_GetRefAcc.....	3-52
3.3.30	HIMC_GetPosOut.....	3-53
3.3.31	HIMC_GetVelOut.....	3-54
3.3.32	HIMC_GetAccOut.....	3-55
3.3.33	HIMC_IgnoreHWL.....	3-56
3.3.34	HIMC_IgnoreSWL.....	3-57
3.3.35	HIMC_IgnorePE.....	3-58
3.3.36	HIMC_GetAxisNum.....	3-59
3.3.37	HIMC_SetVelScale.....	3-60
3.3.38	HIMC_GetVelScale.....	3-61
3.3.39	HIMC_SetRollover.....	3-62
3.3.40	HIMC_GetRolloverTurns.....	3-64
3.3.41	HIMC_SetOpMode.....	3-65
3.3.42	HIMC_SetBufferMode.....	3-66
3.3.43	HIMC_GetBufferMode.....	3-67
3.3.44	HIMC_GetCmdNum.....	3-68
3.3.45	HIMC_GetMultiAxesFeedbackPos.....	3-69
3.3.46	HIMC_SetPosWindow.....	3-70
3.3.47	HIMC_GetPosWindow.....	3-71
3.3.48	HIMC_SetPosWindowTime.....	3-72
3.3.49	HIMC_GetPosWindowTime.....	3-73
3.4	軸狀態.....	3-74
3.4.1	HIMC_IsEnabled.....	3-74
3.4.2	HIMC_IsMoving.....	3-75
3.4.3	HIMC_IsInPos.....	3-76
3.4.4	HIMC_IsErrorStop.....	3-77
3.4.5	HIMC_IsGantry.....	3-78
3.4.6	HIMC_IsGrouped.....	3-79
3.4.7	HIMC_IsSync.....	3-80
3.4.8	HIMC_IsHWLL.....	3-81
3.4.9	HIMC_IsHWRL.....	3-82
3.4.10	HIMC_IsSWLL.....	3-83
3.4.11	HIMC_IsSWRL.....	3-84

3.4.12	HIMC_IsDriveErr	3-85
3.4.13	HIMC_IsPosErr.....	3-86
3.4.14	HIMC_IsCompActive.....	3-87
3.4.15	HIMC_IsAcc	3-88

3.1 概述

HIMC 提供單軸運動命令，包含點到點（P2P）、吋動（JOG）與同步運動命令，使用者可依應用需求使用相關的運動函式功能。圖 3.1.1 為 HIMC 軸運動控制流程圖，運動命令會經過內建的軌跡規劃器（PG·Profile Generator）產生軸的參考位置命令（Reference Position）；而輸出的參考位置命令疊加上軸的誤差補償值，即產生給驅動器的位置輸出命令（Position Output）。

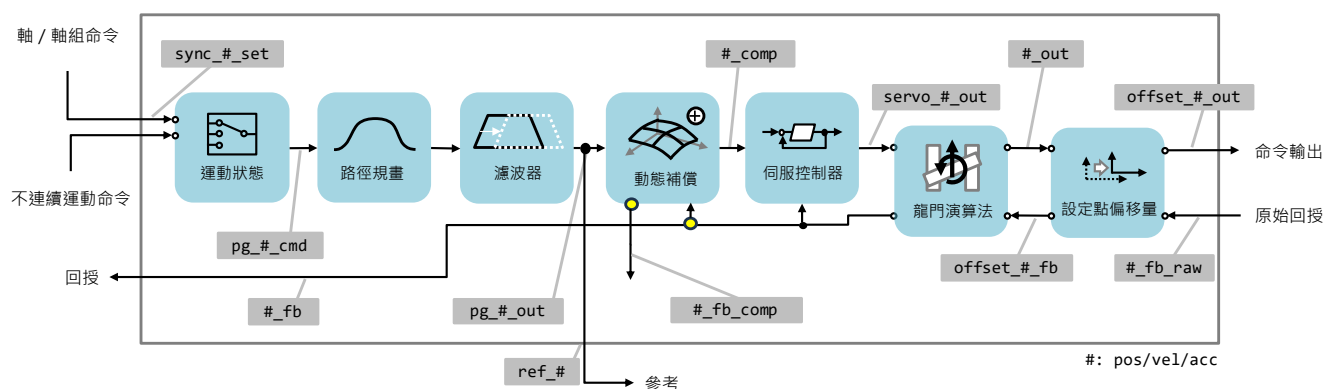


圖 3.1.1

HIMC 的軌跡規劃器內建 S-Curve 速度規劃，如圖 3.1.2 所示。使用者可設定軌跡規劃器的最大速度、最大加速度、最大減速度與平滑時間。

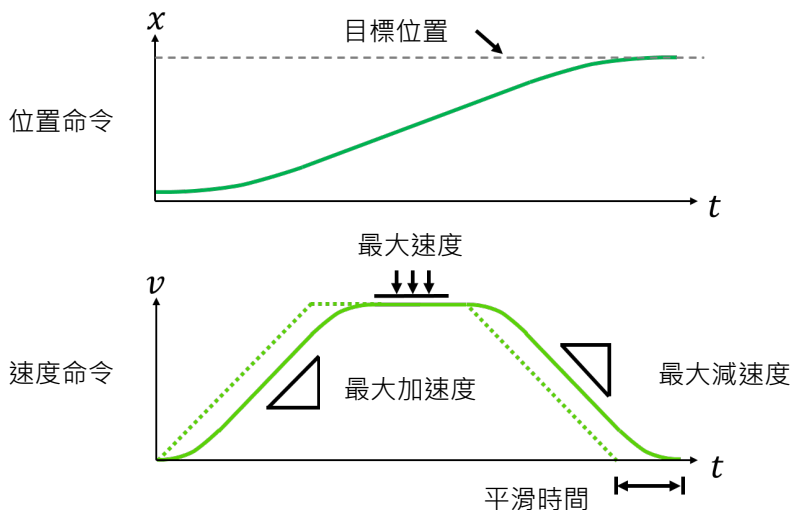


圖 3.1.2

如圖 3.1.3 所示，加入平滑時間會造成軸速度命令延遲，但可以有效降低高加減速所產生的急跳度 (Jerk)，以增加系統的穩定性。急跳度、最大加速度與平滑時間的關係如下：

$$\text{急跳度} = \text{最大加速度} / \text{平滑時間 } T_s$$

總加速度時間則可由下列關係式得到：

$$\text{總加速度時間 } T = \text{T-Curve 加速度時間 } T_a + \text{平滑時間 } T_s$$

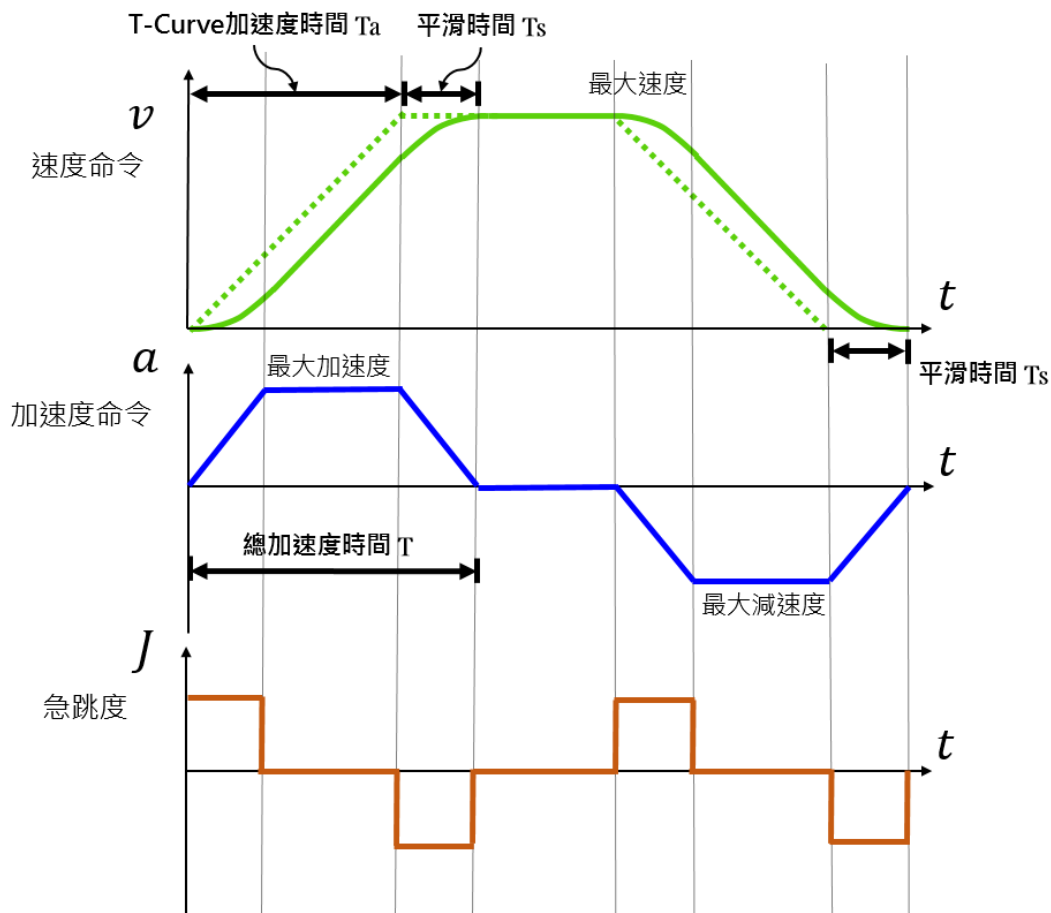


圖 3.1.3

此外，HIMC 軸運動命令支援動態地變更目標位置與速度規劃 (On the Fly Modification)。使用者可以在軸運動的過程中，改變軸的目標位置命令、最大速度與最大加 / 減速度；HIMC 的軌跡規劃器會依據新的命令與軸運動參數，移動到新的目標位置。

軸的運動狀態分成移動中 (Moving) 與是否到位 (In-Position)，如圖 3.1.4 所示。在區域 I 中持續送出軸目標位置，並在進入區域 II 前結束；控制器會依據所設定的位置目標框 (Position Window) 與位置目標框反彈跳時間 (Position Window Time)，判斷軸是否已經到位。

軸位置回授若在軸位置命令的目標框內，經過反彈跳時間後，會被視為軸位置已經到位，此時控制器內部的軸到位狀態成立；若在反彈跳時間內的任一時刻，軸位置回授超出位置命令的目標框，則整定時間的計算將被重設，等待下一次軸位置回授進入目標框內後，才會重新計算是否滿足反彈跳時間的到位條件。

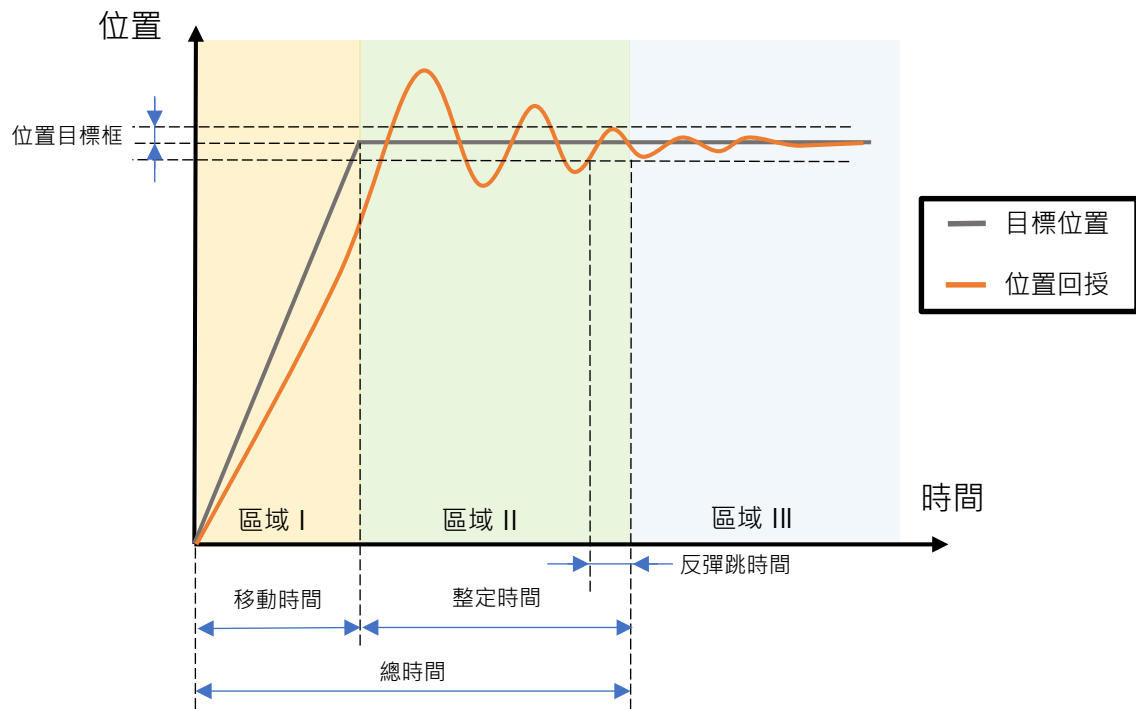


圖 3.1.4

參考圖 3.1.4，軸運動狀態說明如下：

1. 區域 I：軸移動中 (Moving)，尚未到位 (Not In-Position)。
2. 區域 II：軸停止移動 (Not Moving)，但尚未到位 (Not In-Position)。
3. 區域 III：軸停止移動 (Not Moving)，已到位 (In-Position)。

而若軸在同步運動狀態時，軸運動會交由軸群組或主軸來產生規劃的運動軌跡，軸本身不規劃運動軌跡，僅追隨由軸群組或主軸規劃的位置命令。

3.1.1 軸變數

軸變數分成運動命令變數、運動規劃變數、狀態變數與 CoE 物件變數，使用者可利用 iA Studio 的 Scope Manager(請參閱《iA Studio 軟體使用手冊》4.8 節)選擇欲觀測的變數。詳細說明如表 3.1.1.1 至表 3.1.1.6。

表 3.1.1.1 軸運動命令變數

名稱	變數	單位	描述
Sync Position Setpoint	sync_pos_set	毫米或角度	同步位置設定點。當軸處於同步運動模式(如軸群組、凸輪或齒輪傳動操作)時，此點為其目標位置。
Position Command	pg_pos_cmd	毫米或角度	軌跡規劃位置命令。當軸處於不連續運動模式(點對點)時，此點為其目標位置。
Reference Position	ref_pos	毫米或角度	參考位置。根據預先定義好的運動軌跡，經軌跡規劃器產生的位置設定點。
Reference Velocity	ref_vel	毫米/秒或 角度/秒	參考速度。根據預先定義好的運動軌跡，經軌跡規劃器產生的速度設定點。
Reference Acceleration	ref_acc	毫米/秒 ² 或 角度/秒 ²	參考加速度。根據預先定義好的運動軌跡，經軌跡規劃器產生的加速度設定點。
Position Compensation	comp_pos_set	毫米或角度	位置誤差補償值。此為動態誤差補償功能的補償輸出。若不使用此功能，其值為 0。
Compensated Position	pos_comp	毫米或角度	補償後的位置命令值。此為參考位置與位置誤差補償值的加總。
Position Offset	pos_offset	毫米或角度	位置偏移量。預設值為 0。若使用者在馬達未移動時設一個新的軸位置，其值則不是 0。
Position Output	pos_out	毫米或角度	位置輸出。此為無位置偏移量的軸位置命令。
Velocity Output	vel_out	毫米/秒或 角度/秒	速度輸出。此為無速度偏移量的軸速度命令。
Acceleration Output	acc_out	毫米/秒 ² 或 角度/秒 ²	加速度輸出。此為無加速度偏移量的軸加速度命令。
Offsetted Position Output	offset_pos_out	毫米或角度	具位置偏移量的位置輸出。此為最終計算而成的軸位置命令。該值將轉換為單位 count 並傳送到相應的從站。
Raw Position	pos_fb_raw	毫米或角度	原始位置回授。由從站讀取該站收到的編碼器回授。

名稱	變數	單位	描述
Feedback			
Offsetted Position Feedback	offset_pos_fb	毫米或角度	被偏移的位置回授。此為具位置偏移量的位置回授。
Position Feedback	pos_fb	毫米或角度	位置回授。位於軸座標系統中。
Velocity Feedback	vel_fb	毫米/秒或 角度/秒	速度回授。位於軸座標系統中。
Position Error	pos_err	毫米或角度	位置誤差。此為位置輸出與原始位置回授間的差值。
Velocity Error	vel_err	毫米/秒或 角度/秒	速度誤差。此為速度輸出與原始速度回授間的差值。
Move Time	movetime	毫秒	移動時間。
Settling Time	settlingtime	毫秒	整定時間。

表 3.1.1.2 軸運動規劃變數

名稱	變數	單位	描述
Max. Profile Velocity	max_vel	毫米/秒或 角度/秒	最大速度。不一定會達到。
Max. Profile Acceleration	max_acc	毫米/秒 ² 或 角度/秒 ²	最大加速度。不一定會達到。
Profile Deceleration	max_dec	毫米/秒 ² 或 角度/秒 ²	最大減速度。不一定會達到。
Smooth Time	sm_factor	毫秒	平滑時間。輸入範圍為 0 ~ 500。增加該值可減少運動期間的機械振動，但會影響總運動時間。

表 3.1.1.3 軸狀態變數

名稱	變數	單位	描述
Fault Status	fault_status	無	軸錯誤狀態，位元定義請參閱表 3.1.1.4。
Motion Status	motion_status	無	軸運動狀態，位元定義請參閱表 3.1.1.5。

表 3.1.1.4 軸錯誤狀態位元定義

位元	名稱	描述	預設反應
0	Error Stop	軸處於 error stop 狀態。	無。
1	Drive fault	從站驅動器的錯誤。	控制器將軸解激磁。
2	Position error	位置誤差超過保護範圍。	控制器將軸解激磁。
3	Hardware right limit	觸發軸的硬體右極限。	控制器停止軸的運動。
4	Hardware left limit	觸發軸的硬體左極限。	控制器停止軸的運動。
5	Software right limit	觸發軸的軟體右極限。	控制器停止軸的運動。
6	Software left limit	觸發軸的軟體左極限。	控制器停止軸的運動。

表 3.1.1.5 軸運動狀態位元定義

位元	名稱	描述	備註
0	Enabled	軸激磁狀態。	無。
1	Moving	軸移動中。	請參閱 3.1 節。
2	In Position	軸到位。	請參閱 3.1 節。
3	Synchronous	軸在同步狀態。	軸在軸群組中或為電子齒輪從軸。
4	Group	軸在軸群組中。	請參閱 6.1 節。
5	Gantry	軸為龍門軸。	請參閱 5.1 節。
6	Input Shape	開啟 Input Shape 濾波器。	請參閱 13.1 節。
7	VSF	開啟 VSF 濾波器。	請參閱 13.1 節。
8	Gear	軸為電子齒輪從軸。	請參閱 4.1 節。
9	Cam	軸為電子凸輪從軸。	目前不支援。
10	Accelerating	軸加速中。	請參閱 3.1 節。
11	Homed	軸完成歸原點。	無。
12	Homing	軸歸原點中。	無。
13	Home Switch	近原點開關觸發。	無。

表 3.1.1.6 軸 CoE 物件變數

名稱	變數	單位	描述
Power Drive State	power_drive_st	無	Power Drive State (PDS)。
Controlword	coe_controlword	無	軸控制物件。
Statusword	coe_statusword	無	軸狀態物件。
Modes of Operation	servo_type	無	軸控制模式寫入值。
Modes of Operational Display	coe_op_modes_display	無	軸控制模式顯示值。
Target Position	coe_pos_cmd	count	目標位置命令。
Target Velocity	coe_vel_cmd	count/s	目標速度命令。
Target Torque	coe_trq_cmd	無	目標轉矩命令。
Position Actual Value	coe_pos_fb	count	位置回授值。
Velocity Actual Value	coe_vel_fb	count/s	速度回授值。
Torque Actual Value	coe_trq_fb	無	轉矩回授值。
Touch Probe Function	tp_function	無	Touch Probe 功能設定值。
Touch Probe Status	tp_status	無	Touch Probe 功能狀態。
Touch Probe 1 Positive Value	tp1_positive_edge	count	Touch Probe 1 在正緣的位置值。
Touch Probe 1 Negative Value	tp1_negative_edge	count	Touch Probe 1 在負緣的位置值。
Touch Probe 2 Positive Value	tp2_positive_edge	count	Touch Probe 2 在正緣的位置值。
Touch Probe 2 Negative Value	tp2_negative_edge	count	Touch Probe 2 在負緣的位置值。
Negative Limit Switch	di_bit_HWLL	無	負極限開關訊號。
Positive Limit Switch	di_bit_HWRL	無	正極限開關訊號。

名稱	變數	單位	描述
Home Switch	di_bit_HOME	無	近原點開關訊號。

3.2 軸運動控制

3.2.1 HIMC_Enable

用途

激磁軸。

語法

```
int HIMC_Enable(
    int ctrl_id,
    int axis_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將物件 0x6040 (Control word) 、 0x6041 (Status word) 配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_Enable
LabVIEW	HIMC Enable.vi
Python	Enable

3.2.2 HIMC_Disable

用途

解激磁軸。

語法

```
int HIMC_Disable(  
    int ctrl_id,  
    int axis_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

- (1) 此函式會清除軸原有的路徑規劃。
- (2) 使用此函式需將物件 0x6040 (Control word)、0x6041 (Status word) 配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_Disable
LabVIEW	HIMC Disable.vi
Python	Disable

3.2.3 HIMC_Reset

用途

當軸處於 ErrorStop 狀態時，重新設定軸。

語法

```
int HIMC_Reset(
    int ctrl_id,
    int axis_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

- (1) 當軸處於 error stop 狀態時，執行此函式。
- (2) 使用此函式需將物件 0x6040 (Control word)、0x6041 (Status word) 配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_Reset
LabVIEW	HIMC_Reset.vi
Python	Reset

3.2.4 HIMC_MoveAbs

用途

將軸移動至絕對目標位置。

語法

```
int HIMC_MoveAbs(
    int    ctrl_id,
    int    axis_id,
    double pos
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- pos [in] 絕對目標位置的值。
 參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將對應的命令配置為 PDO，例如 CSP 模式為 0x607A（Target position）。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_MoveAbs
LabVIEW	HIMC Move Abs.vi
Python	MoveAbs

3.2.5 HIMC_MoveRel

用途

將軸移動相對距離。

語法

```
int HIMC_MoveRel(
    int    ctrl_id,
    int    axis_id,
    double rel_dist
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

rel_dist [in] 相對距離的值。
參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將對應的命令配置為 PDO，例如 CSP 模式為 0x607A（Target position）。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_MoveRel
LabVIEW	HIMC Move Rel.vi
Python	MoveRel

3.2.6 HIMC_MoveVel

用途

設定運動速度，並以此速度持續移動。

語法

```
int HIMC_MoveVel(  
    int    ctrl_id,  
    int    axis_id,  
    double vel  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- vel [in] 移動速度的值。
 數值的正負值表示運動方向。
 參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

使用此函式需將對應的命令配置為 PDO，例如 CSP 模式為 0x607A (Target position)。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_MoveVel
LabVIEW	HIMC Move Vel.vi
Python	MoveVel

3.2.7 HIMC_MoveTrq

用途

以特定轉矩持續移動。

語法

```
int HIMC_MoveTrq(
    int    ctrl_id,
    int    axis_id,
    double trq_cmd
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

trq_cmd [in] 轉矩命令。
 參數單位：N-m (牛頓-米)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

- (1) 此函式僅適用於 Profile Torque 模式。
- (2) 若轉矩命令大於馬達的連續轉矩，馬達會以連續轉矩的值移動。
- (3) 使用此函式需將物件 0x6071 (Target torque) 配置為 PDO，且需設定馬達的轉矩常數 (force constant)。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_MoveTrq
LabVIEW	HIMC Move Trq.vi
Python	MoveTrq

3.2.8 HIMC_MovePVT

用途

依給定的位置 (P)、速度 (V) 與時間 (T)，將軸移動至指定位置。

語法

```
int HIMC_MovePVT(  
    int    ctrl_id,  
    int    axis_id,  
    double *time,  
    double *pos,  
    double *vel,  
    int    num_pt  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- time [in] 指向使用者給定的時間陣列的指標。
 參數單位：ms (毫秒)
- pos [in] 指向使用者給定的位置陣列的指標。
 參數單位：mm (毫米) 或 deg (角度)
- vel [in] 指向使用者給定的速度陣列的指標。
 參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)
- num_pt [in] PVT 運動點的數量，其最大值為 32。
 時間、位置與速度陣列的長度皆須與此參數一致。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將對應的命令配置為 PDO，例如 CSP 模式為 0x607A (Target position)。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_MovePVT
LabVIEW	-
Python	MovePVT

3.2.9 HIMC_Stop

用途

停止軸的運動。

語法

```
int HIMC_Stop(  
    int ctrl_id,  
    int axis_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

此函式會清除軸原有的路徑規劃。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_Stop
LabVIEW	HIMC Stop.vi
Python	Stop

3.2.10 HIMC_Halt

用途

暫停軸的運動，軸運動速度被設定為 0。

語法

```
int HIMC_Halt(
    int ctrl_id,
    int axis_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

使用此函式需將物件 0x6040 (Control word)、0x6041 (Status word) 配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_Halt
LabVIEW	HIMC Halt.vi
Python	Halt

3.2.11 HIMC_Resume

用途

由軸暫停的狀態中恢復軸的運動。

語法

```
int HIMC_Resume(  
    int ctrl_id,  
    int axis_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將物件 0x6040 (Control word) 、0x6041 (Status word) 配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_Resume
LabVIEW	HIMC Resume.vi
Python	Resume

3.3 軸設定

3.3.1 HIMC_GetVel

用途

取得軸的運動速度目標值。

語法

```
int HIMC_GetVel(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_vel  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_vel [out] 指標型態的記憶體，用來儲存軸的運動速度目標值。
 參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetVel
LabVIEW	-
Python	GetVel

3.3.2 HIMC_SetVel

用途

設置軸的運動速度目標值。

語法

```
int HIMC_SetVel(  
    int    ctrl_id,  
    int    axis_id,  
    double vel  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- vel [in] 軸的新運動速度目標值。
 參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)
 輸入範圍：非零正值

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetVel
LabVIEW	HIMC Set Vel.vi
Python	SetVel

3.3.3 HIMC_GetAcc

用途

取得軸的運動加速度目標值。

語法

```
int HIMC_GetAcc(
    int    ctrl_id,
    int    axis_id,
    double *p_acc
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_acc [out] 指標型態的記憶體，用來儲存軸的運動加速度目標值。
參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetAcc
LabVIEW	-
Python	GetAcc

3.3.4 HIMC_SetAcc

用途

設置軸的運動加速度目標值。

語法

```
int HIMC_SetAcc(
    int    ctrl_id,
    int    axis_id,
    double acc
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- acc [in] 軸的新運動加速度目標值。
參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)
輸入範圍：非零正值

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetAcc
LabVIEW	HIMC Set Acc.vi
Python	SetAcc

3.3.5 HIMC_SetAccTime

用途

設置軸的加速度時間。

語法

```
int HIMC_SetAccTime(  
    int    ctrl_id,  
    int    axis_id,  
    double acc_time  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

acc_time [in] 軸的加速度時間。
 參數單位：ms (毫秒)
 輸入範圍：非零正值

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetAccTime
LabVIEW	HIMC Set Acc Time.vi
Python	SetAccTime

3.3.6 HIMC_GetDec

用途

取得軸的運動減速度目標值。

語法

```
int HIMC_GetDec(
    int    ctrl_id,
    int    axis_id,
    double *p_dec
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_dec [out] 指標型態的記憶體，用來儲存軸的運動減速度目標值。
 參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetDec
LabVIEW	-
Python	GetDec

3.3.7 HIMC_SetDec

用途

設置軸的運動減速度目標值。

語法

```
int HIMC_SetDec(
    int    ctrl_id,
    int    axis_id,
    double dec
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- dec [in] 軸的新運動減速度目標值。
參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)
輸入範圍：非零正值

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetDec
LabVIEW	HIMC Set Dec.vi
Python	SetDec

3.3.8 HIMC_SetDecTime

用途

設置軸的減速度時間。

語法

```
int HIMC_SetDecTime(
    int    ctrl_id,
    int    axis_id,
    double dec_time
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- dec_time [in] 軸的減速度時間。
參數單位：ms (毫秒)
輸入範圍：非零正值

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetDecTime
LabVIEW	HIMC Set Dec Time.vi
Python	SetDecTime

3.3.9 HIMC_GetKillDec

用途

取得軸的緊急減速度。

語法

```
int HIMC_GetKillDec(
    int    ctrl_id,
    int    axis_id,
    double *p_kill_dec
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_kill_dec [out] 指標型態的記憶體，用來儲存軸的緊急減速度。
參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetKillDec
LabVIEW	HIMC Get Kill Dec.vi
Python	GetKillDec

3.3.10 HIMC_SetKillDec

用途

設置軸的緊急減速度。

語法

```
int HIMC_SetKillDec(
    int    ctrl_id,
    int    axis_id,
    double kill_dec
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- kill_dec [in] 軸的新緊急減速度。
 參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)
 輸入範圍：非零正值

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetKillDec
LabVIEW	HIMC Set Kill Dec.vi
Python	SetKillDec

3.3.11 HIMC_GetSWRL

用途

取得軸的軟體右極限位置。

語法

```
int HIMC_GetSWRL(
    int    ctrl_id,
    int    axis_id,
    double *p_right_limit_pos
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
p_right_limit_pos [out]	指標型態的記憶體，用來儲存軸的軟體右極限位置。 參數單位：mm (毫米) 或 deg (角度)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSWRL
LabVIEW	HIMC Get SWRL.vi
Python	GetSWRL

3.3.12 HIMC_SetSWRL

用途

設置軸的軟體右極限位置。

語法

```
int HIMC_SetSWRL(  
    int    ctrl_id,  
    int    axis_id,  
    double right_limit_pos  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in]

軸編號。
- right_limit_pos [in]

軸的新軟體右極限位置。
參數單位：mm (毫米) 或 deg (角度)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetSWRL
LabVIEW	HIMC Set SWRL.vi
Python	SetSWRL

3.3.13 HIMC_GetSWLL

用途

取得軸的軟體左極限位置。

語法

```
int HIMC_GetSWLL(
    int    ctrl_id,
    int    axis_id,
    double *p_left_limit_pos
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
p_left_limit_pos [out]	指標型態的記憶體，用來儲存軸的軟體左極限位置。 參數單位：mm (毫米) 或 deg (角度)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSWLL
LabVIEW	HIMC Get SWLL.vi
Python	GetSWLL

3.3.14 HIMC_SetSWLL

用途

設置軸的軟體左極限位置。

語法

```
int HIMC_SetSWLL(  
    int    ctrl_id,  
    int    axis_id,  
    double left_limit_pos  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in]

軸編號。
- left_limit_pos [in]

軸的新軟體左極限位置。
參數單位：mm (毫米) 或 deg (角度)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetSWLL
LabVIEW	HIMC Set SWLL.vi
Python	SetSWLL

3.3.15 HIMC_GetSMTime

用途

取得軸的平滑時間。

語法

```
int HIMC_GetSMTime(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_smooth_time  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_smooth_time [out] 指標型態的記憶體，用來儲存軸的平滑時間。
 參數單位：ms (毫秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSMTime
LabVIEW	HIMC Get SM Time.vi
Python	GetSMTime

3.3.16 HIMC_SetSMTime

用途

設置軸的平滑時間。

語法

```
int HIMC_SetSMTime(
    int    ctrl_id,
    int    axis_id,
    double smooth_time
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
smooth_time [in]	軸的新平滑時間。 參數單位：ms (毫秒) 輸入範圍：0 ~ 500

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

當軸正在移動時，此函式不適用。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetSMTime
LabVIEW	HIMC Set SM Time.vi
Python	SetSMTime

3.3.17 HIMC_GetMoveTime

用途

取得軸的移動時間。

語法

```
int HIMC_GetMoveTime(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_move_time  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_move_time [out] 指標型態的記憶體，用來儲存軸的移動時間。
 參數單位：ms (毫秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetMoveTime
LabVIEW	HIMC Get Move Time.vi
Python	GetMoveTime

3.3.18 HIMC_GetSettlingTime

用途

取得軸的整定時間。

語法

```
int HIMC_GetSettlingTime(
    int    ctrl_id,
    int    axis_id,
    double *p_settling_time
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_settling_time [out] 指標型態的記憶體，用來儲存軸的整定時間。
參數單位：ms (毫秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSettlingTime
LabVIEW	HIMC Get Settling Time.vi
Python	GetSettlingTime

3.3.19 HIMC_SetPos

用途

設定軸位置，改變原點偏移量。

語法

```
int HIMC_SetPos(
    int    ctrl_id,
    int    axis_id,
    double pos
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- pos [in] 軸目前位置的值。
 參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

當軸處於同步狀態、加入軸群組或處於錯誤狀態時，此功能不適用。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetPos
LabVIEW	HIMC Set Pos.vi
Python	SetPos

3.3.20 HIMC_GetPosFb

用途

取得軸的位置回授。

語法

```
int HIMC_GetPosFb(
    int    ctrl_id,
    int    axis_id,
    double *p_pos
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_pos [out] 指標型態的記憶體，用來儲存軸的位置回授。
 參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

使用此函式需將物件 0x6064（Position actual value）配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetPosFb
LabVIEW	HIMC Get Pos Fb.vi
Python	GetPosFb

3.3.21 HIMC_GetPosFbComp

用途

取得加入位置補償值的軸位置回授。

語法

```
int HIMC_GetPosFbComp(
    int    ctrl_id,
    int    axis_id,
    double *p_pos
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_pos [out] 指標型態的記憶體，用來儲存加入位置補償值的軸位置回授。
 參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將物件 0x6064（Position actual value）配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetPosFbComp
LabVIEW	-
Python	GetPosFbComp

3.3.22 HIMC_GetPosOffset

用途

取得軸的位置偏移量。

語法

```
int HIMC_GetPosOffset(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_pos  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_pos [out] 指標型態的記憶體，用來儲存軸的位置偏移量。
 參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetPosOffset
LabVIEW	HIMC Get Pos Offset.vi
Python	GetPosOffset

3.3.23 HIMC_GetPosErr

用途

取得軸的位置誤差。

語法

```
int HIMC_GetPosErr(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_pos_err  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

p_pos_err [out] 指標型態的記憶體，用來儲存軸的位置誤差。
 參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetPosErr
LabVIEW	HIMC Get Pos Err.vi
Python	GetPosErr

3.3.24 HIMC_GetVelFb

用途

取得軸的速度回授。

語法

```
int HIMC_GetVelFb(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_vel  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_vel [out] 指標型態的記憶體，用來儲存軸的速度回授。
 參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetVelFb
LabVIEW	HIMC Get Vel Fb.vi
Python	GetVelFb

3.3.25 HIMC_GetVelErr

用途

取得軸的速度誤差。

語法

```
int HIMC_GetVelErr(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_vel_err  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_vel_err [out] 指標型態的記憶體，用來儲存軸的速度誤差。
 參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetVelErr
LabVIEW	HIMC Get Vel Err.vi
Python	GetVelErr

3.3.26 HIMC_GetCurrFb

用途

取得軸的電流回授。

語法

```
int HIMC_GetCurrFb(
    int    ctrl_id,
    int    axis_id,
    double *p_curr
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_curr [out] 指標型態的記憶體，用來儲存軸的電流回授。
 參數單位：安培 (A)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將物件 0x6077 (Torque actual value) 配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetCurrFb
LabVIEW	-
Python	GetCurrFb

3.3.27 HIMC_GetRefPos

用途

取得軸的參考位置。

語法

```
int HIMC_GetRefPos(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_pos  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

p_pos [out] 指標型態的記憶體，用來儲存軸的參考位置。
 參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetRefPos
LabVIEW	HIMC Get Ref Pos.vi
Python	GetRefPos

3.3.28 HIMC_GetRefVel

用途

取得軸的參考速度。

語法

```
int HIMC_GetRefVel(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_vel  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_vel [out] 指標型態的記憶體，用來儲存軸的參考速度。
 參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetRefVel
LabVIEW	HIMC Get Ref Vel.vi
Python	GetRefVel

3.3.29 HIMC_GetRefAcc

用途

取得軸的參考加速度。

語法

```
int HIMC_GetRefAcc(
    int    ctrl_id,
    int    axis_id,
    double *p_acc
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_acc [out] 指標型態的記憶體，用來儲存軸的參考加速度。
參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetRefAcc
LabVIEW	HIMC Get Ref Acc.vi
Python	GetRefAcc

3.3.30 HIMC_GetPosOut

用途

取得由控制器送至驅動器的位置命令輸出。

語法

```
int HIMC_GetPosOut(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_pos  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_pos [out] 指標型態的記憶體，用來儲存軸的位置命令輸出。
 參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetPosOut
LabVIEW	HIMC Get Pos Out.vi
Python	GetPosOut

3.3.31 HIMC_GetVelOut

用途

取得由控制器送至驅動器的速度命令輸出。

語法

```
int HIMC_GetVelOut(
    int    ctrl_id,
    int    axis_id,
    double *p_vel
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_vel [out] 指標型態的記憶體，用來儲存軸的速度命令輸出。
參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetVelOut
LabVIEW	HIMC Get Vel Out.vi
Python	GetVelOut

3.3.32 HIMC_GetAccOut

用途

取得由控制器送至驅動器的加速度命令輸出。

語法

```
int HIMC_GetAccOut(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_acc  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_acc [out] 指標型態的記憶體，用來儲存軸的加速度命令輸出。
 參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetAccOut
LabVIEW	HIMC Get Acc Out.vi
Python	GetAccOut

3.3.33 HIMC_IgnoreHWL

用途

忽略硬體極限的警告通知。

語法

```
int HIMC_IgnoreHWL(  
    int ctrl_id,  
    int axis_id,  
    int cmd  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- cmd [in] 設為 1：忽略通知；設為 0：恢復通知（預設值）。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IgnoreHWL
LabVIEW	HIMC Ignore HWL.vi
Python	IgnoreHWL

3.3.34 HIMC_IgnoreSWL

用途

忽略軟體極限的警告通知。

語法

```
int HIMC_IgnoreSWL(  
    int ctrl_id,  
    int axis_id,  
    int cmd  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- cmd [in] 設為 1：忽略通知；設為 0：恢復通知（預設值）。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IgnoreSWL
LabVIEW	HIMC Ignore SWL.vi
Python	IgnoreSWL

3.3.35 HIMC_IgnorePE

用途

忽略位置誤差極限的警告通知。

語法

```
int HIMC_IgnorePE(
    int ctrl_id,
    int axis_id,
    int cmd
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- cmd [in] 設為 1：忽略通知；設為 0：恢復通知（預設值）。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IgnorePE
LabVIEW	HIMC Ignore PE.vi
Python	IgnorePE

3.3.36 HIMC_GetAxisNum

用途

取得連接至控制器的軸數量。

語法

```
int HIMC_GetAxisNum(  
    int ctrl_id,  
    int *num  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- num [out] 指標型態的記憶體，用來儲存連接至控制器的軸數量。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetAxisNum
LabVIEW	HIMC Get Axis Num.vi
Python	GetAxisNum

3.3.37 HIMC_SetVelScale

用途

設置軸運動的速度百分比。

語法

```
int HIMC_SetVelScale(  
    int    ctrl_id,  
    int    axis_id,  
    double vel_scale  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

vel_scale [in] 軸運動的新速度百分比。
 輸入範圍：0 ~ 100

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetVelScale
LabVIEW	HIMC Set Vel Scale.vi
Python	SetVelScale

3.3.38 HIMC_GetVelScale

用途

取得軸運動的速度百分比。

語法

```
int HIMC_GetVelScale(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_vel_scale  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in]

軸編號。
- p_vel_scale [out]

指標型態的記憶體，用來儲存軸運動的速度百分比。
數值範圍為 0 ~ 100。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetVelScale
LabVIEW	HIMC Get Vel Scale.vi
Python	GetVelScale

3.3.39 HIMC_SetRollover

用途

設置軸在單圈模式下的位置上限值。

語法

```
int HIMC_SetRollover(  
    int    ctrl_id,  
    int    axis_id,  
    double rollover_val  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
rollover_val [in]	軸在單圈模式下的位置上限值。 參數單位：mm (毫米) 或 deg (角度) 輸入範圍：零或大於零的正值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

- (1) 若參數 rollover_val 設定為 0，即關閉此功能。
- (2) 當軸處於解激磁狀態時，此函式才適用。
- (3) 當軸加入軸群組時，此功能不適用。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetRollover
LabVIEW	HIMC Set Rollover.vi
Python	SetRollover

3.3.40 HIMC_GetRolloverTurns

用途

取得軸在單圈模式下的翻轉圈數。

語法

```
int HIMC_GetRolloverTurns(  
    int ctrl_id,  
    int axis_id,  
    int *p_turns  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_turns [out] 指標型態的記憶體，用來儲存軸在單圈模式下的翻轉圈數。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetRolloverTurns
LabVIEW	HIMC Get Rollover Turns.vi
Python	GetRolloverTurns

3.3.41 HIMC_SetOpMode

用途

設置軸的操作模式。

語法

```
int HIMC_SetOpMode(  
    int ctrl_id,  
    int axis_id,  
    int op_mode  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- op_mode [in] 軸的新操作模式。
 輸入範圍：8 (週期同步位置)、9 (週期同步速度)、10 (週期同步轉矩)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetOpMode
LabVIEW	HIMC Set Op Mode.vi
Python	SetOpMode

3.3.42 HIMC_SetBufferMode

用途

設置軸的速度緩衝模式。

語法

```
int HIMC_SetBufferMode(  
    int ctrl_id,  
    int axis_id,  
    int buf_mode  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- buf_mode [in] 軸的新速度緩衝模式。
 輸入範圍：0 (立即停止模式)、1 (緩衝模式)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetBufferMode
LabVIEW	HIMC Set Buffer Mode.vi
Python	SetBufferMode

3.3.43 HIMC_GetBufferMode

用途

取得軸的速度緩衝模式。

語法

```
int HIMC_GetBufferMode(  
    int ctrl_id,  
    int axis_id,  
    int* buf_mode  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- buf_mode [out] 指標型態的記憶體，用來儲存軸的速度緩衝模式。
 0：立即停止模式、1：緩衝模式

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetBufferMode
LabVIEW	HIMC Get Buffer Mode.vi
Python	GetBufferMode

3.3.44 HIMC_GetCmdNum

用途

取得軸暫存的命令數量。

語法

```
int HIMC_GetCmdNum(
    int ctrl_id,
    int axis_id,
    int* cmd_num
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- cmd_num [out] 指標型態的記憶體，用來儲存軸暫存的命令數量。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetCmdNum
LabVIEW	HIMC Get Cmd Num.vi
Python	GetCmdNum

3.3.45 HIMC_GetMultiAxesFeedbackPos

用途

取得多軸的位置回授。

語法

```
int HIMC_GetMultiAxesFeedbackPos(  
    int ctrl_id,  
    int *p_axes_id_array,  
    int num_of_axes,  
    int *p_pos_array  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_axes_id_array [in] 指標型態的記憶體，用來儲存多軸的 ID。
- num_of_axes [in] 軸的數量。
- p_pos_array [out] 指標型態的記憶體，用來儲存多軸的位置回授。
 參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetMultiAxesFeedbackPos
LabVIEW	HIMC Get Multi Axes Feedback Pos.vi
Python	GetMultiAxesFeedbackPos

3.3.46 HIMC_SetPosWindow

用途

設置軸的位置目標框。

語法

```
int HIMC_SetPosWindow(
    int ctrl_id,
    int axis_id,
    double pos_window
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
pos_window [in]	軸的新位置目標框。 參數單位：mm (毫米) 或 deg (角度) 輸入範圍：非零正值

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetPosWindow
LabVIEW	-
Python	SetPosWindow

3.3.47 HIMC_GetPosWindow

用途

取得軸的位置目標框。

語法

```
int HIMC_GetPosWindow(  
    int ctrl_id,  
    int axis_id,  
    double *pos_window  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in]

軸編號。
- pos_window [out]

指標型態的記憶體，用來儲存軸的位置目標框。
參數單位：mm (毫米) 或 deg (角度)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetPosWindow
LabVIEW	-
Python	GetPosWindow

3.3.48 HIMC_SetPosWindowTime

用途

設置軸的位置目標框反彈跳時間。

語法

```
int HIMC_SetPosWindowTime(  
    int ctrl_id,  
    int axis_id,  
    double pos_window_time  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
pos_window_time [in]	軸的新位置目標框反彈跳時間。 參數單位：ms (毫秒) 輸入範圍：非零正值

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetPosWindowTime
LabVIEW	-
Python	SetPosWindowTime

3.3.49 HIMC_GetPosWindowTime

用途

取得軸的位置目標框反彈跳時間。

語法

```
int HIMC_GetPosWindowTime(  
    int ctrl_id,  
    int axis_id,  
    double *pos_window_time  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in]

軸編號。
- pos_window_time [out]

指標型態的記憶體，用來儲存軸的位置目標框反彈跳時間。
參數單位：ms (毫秒)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetPosWindowTime
LabVIEW	-
Python	GetPosWindowTime

3.4 軸狀態

3.4.1 HIMC_IsEnabled

用途

詢問軸的激磁狀態。

語法

```
int HIMC_IsEnabled(
    int ctrl_id,
    int axis_id,
    int *p_enabled
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_enabled [out] 指標型態的記憶體，用來儲存軸的激磁狀態。
 若軸為激磁狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將物件 0x6041 (Status word) 配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsEnabled
LabVIEW	HIMC Is Enabled.vi
Python	IsEnabled

3.4.2 HIMC_IsMoving

用途

詢問軸的移動狀態。若軸正在移動，軌跡規劃器 (PG) 會持續輸出新的位置。

語法

```
int HIMC_IsMoving(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_moving  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_moving [out] 指標型態的記憶體，用來儲存軸的移動狀態。
 若軸正在移動，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsMoving
LabVIEW	HIMC Is Moving.vi
Python	IsMoving

3.4.3 HIMC_IsInPos

用途

詢問軸的到位狀態。若軸已到位，位置誤差會小於所設定的目標框，並維持一段時間（反彈跳時間）。

語法

```
int HIMC_IsInPos(  
    int ctrl_id,  
    int axis_id,  
    int *p_in_position  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_in_position [out] 指標型態的記憶體，用來儲存軸的到位狀態。
 若軸已到位，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsInPos
LabVIEW	HIMC Is In Pos.vi
Python	IsInPos

3.4.4 HIMC_IsErrorStop

用途

詢問軸是否處於 error stop 狀態。

語法

```
int HIMC_IsErrorStop(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_errorstop  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in]

軸編號。
- p_is_errorstop [out]

指標型態的記憶體，用來儲存軸的 error stop 狀態。
若軸處於 error stop 狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsErrorStop
LabVIEW	HIMC Is Error Stop.vi
Python	IsErrorStop

3.4.5 HIMC_IsGantry

用途

詢問軸是否處於龍門狀態。

語法

```
int HIMC_IsGantry(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_gantry  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
p_is_gantry [out]	指標型態的記憶體，用來儲存軸的龍門狀態。 若軸處於龍門狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsGantry
LabVIEW	HIMC Is Gantry.vi
Python	IsGantry

3.4.6 HIMC_IsGrouped

用途

詢問軸是否被歸類至一個軸群組。

語法

```
int HIMC_IsGrouped(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_grouped  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_grouped [out] 指標型態的記憶體，用來儲存軸的 grouped 狀態。
 若軸被歸類至一個軸群組，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsGrouped
LabVIEW	HIMC Is Grouped.vi
Python	IsGrouped

3.4.7 HIMC_IsSync

用途

詢問軸是否處於同步狀態。

語法

```
int HIMC_IsSync(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_sync  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_sync [out] 指標型態的記憶體，用來儲存軸的同步狀態。
 若軸處於同步狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsSync
LabVIEW	HIMC Is Sync.vi
Python	IsSync

3.4.8 HIMC_IsHWLL

用途

詢問軸是否觸發硬體左極限（HWLL）。

語法

```
int HIMC_IsHWLL(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_hwll  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_hwll [out] 指標型態的記憶體，用來儲存軸的 HWLL 狀態。
 若軸處於 HWLL 狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

使用此函式需將物件 0x60FD（Digital inputs）配置為 PDO，指定 bit 0 為左極限輸入。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsHWLL
LabVIEW	HIMC Is HWLL.vi
Python	IsHWLL

3.4.9 HIMC_IsHWRL

用途

詢問軸是否觸發硬體右極限 (HWRL) 。

語法

```
int HIMC_IsHWRL(
    int ctrl_id,
    int axis_id,
    int *p_is_hwrl
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_hwrl [out] 指標型態的記憶體，用來儲存軸的 HWRL 狀態。
若軸處於 HWRL 狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將物件 0x60FD (Digital inputs) 配置為 PDO，指定 bit 1 為右極限輸入。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsHWRL
LabVIEW	HIMC Is HWRL.vi
Python	IsHWRL

3.4.10 HIMC_IsSWLL

用途

詢問軸是否觸發軟體左極限（SWLL）。

語法

```
int HIMC_IsSWLL(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_swll  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_swll [out] 指標型態的記憶體，用來儲存軸的 SWLL 狀態。
 若軸處於 SWLL 狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsSWLL
LabVIEW	HIMC Is SWLL.vi
Python	IsSWLL

3.4.11 HIMC_IsSWRL

用途

詢問軸是否觸發軟體右極限（SWRL）。

語法

```
int HIMC_IsSWRL(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_swrl  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_swrl [out] 指標型態的記憶體，用來儲存軸的 SWRL 狀態。
 若軸處於 SWRL 狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsSWRL
LabVIEW	HIMC Is SWRL.vi
Python	IsSWRL

3.4.12 HIMC_IsDriveErr

用途

詢問軸是否觸發驅動器警報。

語法

```
int HIMC_IsDriveErr(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_driveerr  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in]

軸編號。
- p_is_driveerr [out]

指標型態的記憶體，用來儲存軸的 DriveErr 狀態。
若軸觸發驅動器警報，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

使用此函式需將物件 0x6041 (Status word) 配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsDriveErr
LabVIEW	HIMC Is Drive Err.vi
Python	IsDriveErr

3.4.13 HIMC_IsPosErr

用途

詢問軸的位置誤差是否超過保護範圍。

語法

```
int HIMC_IsPosErr(
    int ctrl_id,
    int axis_id,
    int *p_is_poserr
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
p_is_poserr [out]	指標型態的記憶體，用來儲存軸的 PosErr 狀態。 若軸的位置誤差超過保護範圍，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

誤差保護範圍係指控制器內的軸位置誤差容許值。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsPosErr
LabVIEW	HIMC Is Pos Err.vi
Python	IsPosErr

3.4.14 HIMC_IsCompActive

用途

詢問補償功能是否啟用。

語法

```
int HIMC_IsCompActive(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_compactive  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_compactive [out] 指標型態的記憶體，用來儲存軸的補償功能狀態。
 若軸已啟用補償功能，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsCompActive
LabVIEW	HIMC Is Comp Active.vi
Python	IsCompActive

3.4.15 HIMC_IsAcc

用途

詢問軸是否正在加速。

語法

```
int HIMC_IsAcc(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_acc,  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_acc [out] 指標型態的記憶體，用來儲存軸的加速狀態。
 若軸正在加速，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsAcc
LabVIEW	HIMC Is Acc.vi
Python	IsAcc

4. 同步運動函式

4.	同步運動函式	4-1
4.1	概述	4-2
4.1.1	同步運動變數	4-3
4.2	HIMC_EnableGear	4-4
4.3	HIMC_DisableGear	4-5
4.4	HIMC_GearIn	4-6
4.5	HIMC_GearOut	4-7
4.6	HIMC_GetGearRatio	4-8
4.7	HIMC_IsInGear	4-9
4.8	HIMC_IsGearMaster	4-10
4.9	HIMC_IsGearSlave	4-11

4.1 概述

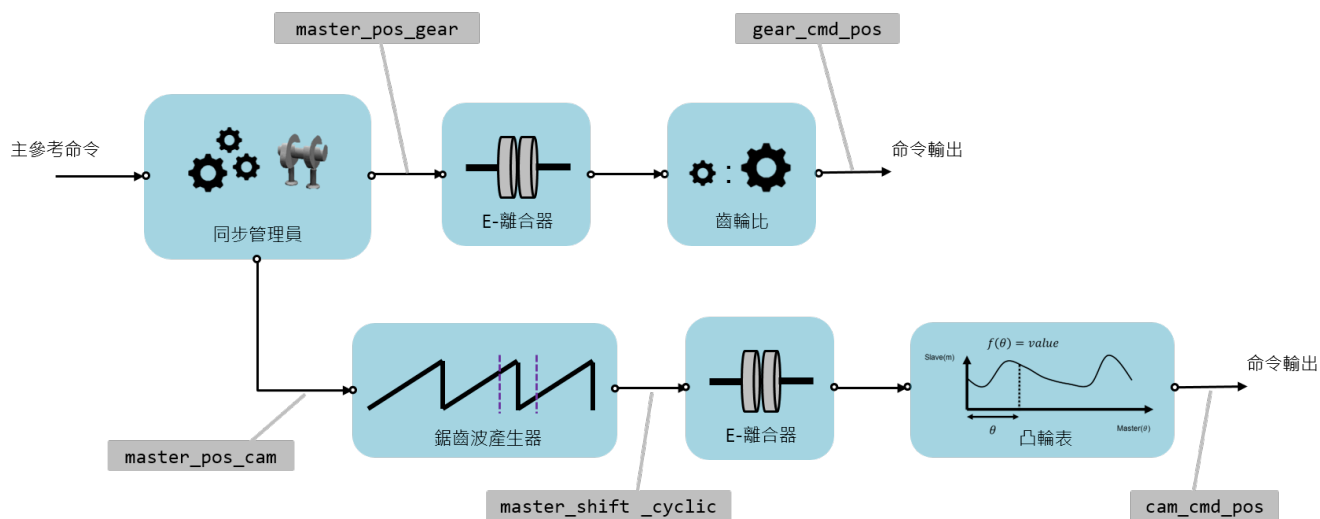


圖 4.1.1

使用者可定義兩軸間的同步運動。作為引導軸的主軸生成位置命令後，從軸會依運動配置參考主軸。若主從關係固定不變，則為電子齒輪傳動；若從軸須遵循某個模式，則為電子凸輪傳動。圖 4.1.2 中，軸 0 作為主軸，引導軸 1、軸 2、軸 3 與軸 4。軸 1、軸 2 與軸 3 採電子齒輪傳動，軸 4 則採電子凸輪傳動。

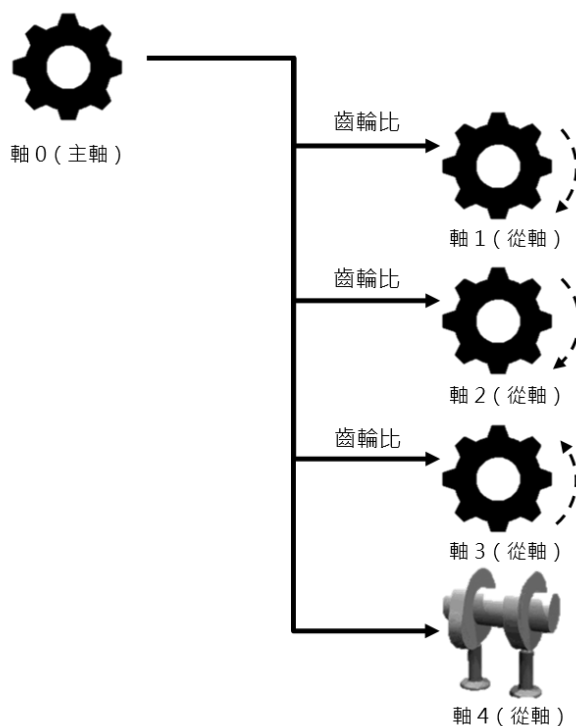


圖 4.1.2

4.1.1 同步運動變數

表 4.1.1.1 為常見的同步運動變數，使用者可利用 iA Studio 的 Scope Manager (請參閱《iA Studio 軟體使用手冊》4.8 節) 選擇欲觀測的變數。

表 4.1.1.1

名稱	變數	單位	描述
Raw Master Position	master_pos_gear	毫米 或 角度	主軸的位置命令。
Gear Command Position	gear_cmd_pos	毫米 或 角度	從軸輸出位置命令。
Gear Ratio	gear_ratio	毫米 或 角度	齒輪比。

4.2 HIMC_EnableGear

用途

結合兩軸，使其成為主從關係。

語法

```
int HIMC_EnableGear(  
    int ctrl_id,  
    int axis_master_id,  
    int axis_slv_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_master_id [in] 主軸編號。

axis_slv_id [in] 從軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

當兩軸都已激磁，此函式才適用。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_EnableGear
LabVIEW	HIMC Enable Gear.vi
Python	EnableGear

4.3 HIMC_DisableGear

用途

解除兩軸的主從關係，使其恢復兩獨立軸。

語法

```
int HIMC_DisableGear(  
    int ctrl_id,  
    int axis_slv_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_slv_id [in] 從軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisableGear
LabVIEW	HIMC Disable Gear.vi
Python	DisableGear

4.4 HIMC_GearIn

用途

更改從軸的狀態：空檔→咬合。

語法

```
int HIMC_GearIn(  
    int ctrl_id,  
    int axis_master_id,  
    int axis_slv_id,  
    double gear_ratio  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_master_id [in] 主軸編號。
- axis_slv_id [in] 從軸編號。
- gear_ratio [in] 齒輪比的值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

當兩軸都已激磁，此函式才適用。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GearIn
LabVIEW	HIMC Gear In.vi
Python	GearIn

4.5 HIMC_GearOut

用途

更改從軸的狀態：咬合→空檔。

語法

```
int HIMC_GearOut(  
    int ctrl_id,  
    int axis_slv_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_slv_id [in] 從軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GearOut
LabVIEW	HIMC Gear Out.vi
Python	GearOut

4.6 HIMC_GetGearRatio

用途

取得從軸的電子齒輪比。

語法

```
int HIMC_GetGearRatio(  
    int ctrl_id,  
    int axis_slv_id,  
    double *p_ratio  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_slv_id [in] 從軸編號。
- p_ratio [out] 指標型態的記憶體，用來儲存從軸的電子齒輪比。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGearRatio
LabVIEW	HIMC Get Gear Ratio.vi
Python	GetGearRatio

4.7 HIMC_IsInGear

用途

詢問從軸是否處於咬合狀態。

語法

```
int HIMC_IsInGear(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_in_gear  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_in_gear [out] 指標型態的記憶體，用來儲存從軸的咬合狀態。
 若從軸處於咬合狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsInGear
LabVIEW	HIMC Is In Gear.vi
Python	IsInGear

4.8 HIMC_IsGearMaster

用途

詢問軸是否為主軸。

語法

```
int HIMC_IsGearMaster(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_gear_master  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
p_is_gear_master [out]	指標型態的記憶體，用來儲存軸是否為主軸的狀態。 若軸為主軸，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsGearMaster
LabVIEW	HIMC Is Gear Master.vi
Python	IsGearMaster

4.9 HIMC_IsGearSlave

用途

詢問軸是否為從軸。

語法

```
int HIMC_IsGearSlave(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_gear_slv  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in]

軸編號。
- p_is_gear_slv [out]

指標型態的記憶體，用來儲存軸是否為從軸的狀態。
若軸為從軸，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsGearSlave
LabVIEW	HIMC Is Gear Slave.vi
Python	IsGearSlave

(此頁有意留白。)

5. 龍門函式

5.	龍門函式.....	5-1
5.1	概述	5-2
5.2	HIMC_EnableGantryPair	5-3
5.3	HIMC_DisableGantryPair.....	5-4
5.4	HIMC_GetGantryPairID	5-5
5.5	HIMC_IsGantryPair.....	5-6

5.1 概述

龍門配置將一對右側軸（RHS）和左側軸（LHS）轉換為一對假想的線性軸（Linear）和旋轉軸（Yaw），如圖 5.1.1。使用者在建立龍門配置之後，可對右側軸下達線性軸方向的命令，以相同方向驅動右側軸和左側軸；對左側軸則下達旋轉軸方向的旋轉運動命令。

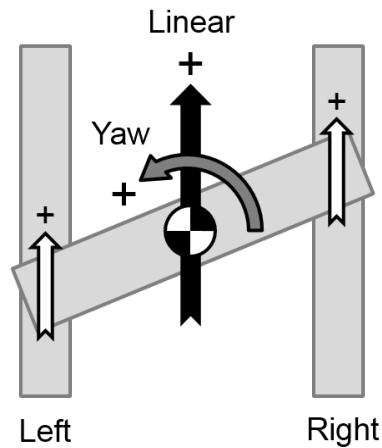


圖 5.1.1

在龍門配置中，線性軸與旋轉軸的位置回授定義如下：

$$Pos_{linear} = \frac{Pos_{RHS} + Pos_{LHS}}{2}; \quad Pos_{yaw} = \frac{Pos_{RHS} - Pos_{LHS}}{2}$$

Pos_{linear} : 線性軸的位置回授 Pos_{yaw} : 旋轉軸的位置回授

Pos_{RHS} : 右側軸的位置回授 Pos_{LHS} : 左側軸的位置回授

圖 5.1.2 為線性軸、旋轉軸、右側軸與左側軸配置的位置回授示意圖。

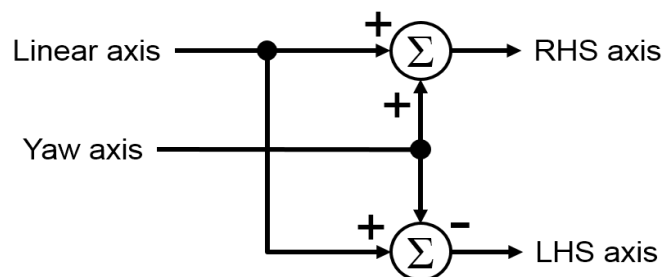


圖 5.1.2

5.2 HIMC_EnableGantryPair

用途

建立一對龍門。

語法

```
int HIMC_EnableGantryPair(  
    int ctrl_id,  
    int lhs_axis_id,  
    int rhs_axis_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- lhs_axis_id [in] 左側軸之編號。
- rhs_axis_id [in] 右側軸之編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

當兩軸都處於解激磁狀態時，此函式才適用。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_EnableGantryPair
LabVIEW	HIMC Enable Gantry Pair.vi
Python	EnableGantryPair

5.3 HIMC_DisableGantryPair

用途

分開一對龍門。

語法

```
int HIMC_DisableGantryPair(  
    int ctrl_id,  
    int axis_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 龍門中任一軸之編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

當兩軸都處於解激磁狀態時，此函式才適用。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisableGantryPair
LabVIEW	HIMC Disable Gantry Pair.vi
Python	DisableGantryPair

5.4 HIMC_GetGantryPairID

用途

取得任意龍門軸的龍門 ID。

語法

```
int HIMC_GetGantryPairID(  
    int ctrl_id,  
    int axis_id,  
    int *p_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 龍門中任一軸之編號。
- p_id [out] 指標型態的記憶體，用來儲存龍門 ID。
 若輸入軸本身不是龍門軸，則值為-1。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGantryPairID
LabVIEW	HIMC Get Gantry Pair ID.vi
Python	GetGantryPairID

5.5 HIMC_IsGantryPair

用途

詢問任意兩軸是否為一對龍門。

語法

```
int HIMC_IsGantryPair(  
    int ctrl_id,  
    int axis_id_1,  
    int axis_id_2,  
    int *p_is_gantry_pair  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id_1 [in] 軸編號 1。
- axis_id_2 [in] 軸編號 2。
- p_is_gantry_pair [out] 指標型態的記憶體，用來儲存兩軸是否為一對龍門的狀態。
 若兩軸為一對龍門，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsGantryPair
LabVIEW	HIMC Is Gantry Pair.vi
Python	IsGantryPair

6. 軸群組函式

6.	軸群組函式	6-1
6.1	概述	6-3
6.1.1	軸群組變數	6-6
6.1.2	座標系統	6-9
6.1.3	運動學	6-13
6.1.4	速度緩衝模式	6-13
6.1.5	路徑過渡模式	6-16
6.2	軸群組運動控制	6-18
6.2.1	HIMC_EnableGroup	6-18
6.2.2	HIMC_DisableGroup	6-19
6.2.3	HIMC_ResetGroup	6-20
6.2.4	HIMC_StopGroup	6-21
6.2.5	HIMC_HaltGroup	6-22
6.2.6	HIMC_ResumeGroup	6-23
6.2.7	HIMC_JogGroup	6-24
6.2.8	HIMC_JogGroupAxis	6-25
6.2.9	HIMC_LineAbs2D	6-26
6.2.10	HIMC_LineAbs3D	6-27
6.2.11	HIMC_LineRel2D	6-28
6.2.12	HIMC_LineRel3D	6-29
6.2.13	HIMC_Arc2D	6-30
6.2.14	HIMC_ArcCW2D	6-32
6.2.15	HIMC_ArcCCW2D	6-33
6.2.16	HIMC_ArcAngle2D	6-34
6.2.17	HIMC_Circle2D	6-36
6.3	軸群組設定	6-38
6.3.1	HIMC_AddAxesToGrp	6-38
6.3.2	HIMC_RemoveAxisFromGrp	6-39
6.3.3	HIMC_SetupGroup	6-40
6.3.4	HIMC_UngrpAllAxes	6-41
6.3.5	HIMC_GetGroupID	6-42
6.3.6	HIMC_SetGrpMotionProfile	6-43
6.3.7	HIMC_SetGrpAngMotionProfile	6-45
6.3.8	HIMC_GetGrpKin	6-47
6.3.9	HIMC_SetGrpKin	6-48
6.3.10	HIMC_GetGrpVel	6-49
6.3.11	HIMC_SetGrpVel	6-50
6.3.12	HIMC_GetGrpAcc	6-51

6.3.13	HIMC_SetGrpAcc	6-52
6.3.14	HIMC_SetGrpAccTime	6-53
6.3.15	HIMC_GetGrpDec	6-54
6.3.16	HIMC_SetGrpDec	6-55
6.3.17	HIMC_SetGrpDecTime	6-56
6.3.18	HIMC_GetGrpSMTime	6-57
6.3.19	HIMC_SetGrpSMTime	6-58
6.3.20	HIMC_GetGrpCoordSys	6-59
6.3.21	HIMC_SetGrpCoordSys	6-60
6.3.22	HIMC_GetGrpBufferMode	6-61
6.3.23	HIMC_SetGrpBufferMode	6-62
6.3.24	HIMC_GetGrpTransMode	6-63
6.3.25	HIMC_SetGrpTransMode	6-64
6.3.26	HIMC_SetGrpTransPrm	6-65
6.3.27	HIMC_GetGrpCmdNum	6-67
6.3.28	HIMC_SetGrpVelScale	6-68
6.3.29	HIMC_GetGrpVelScale	6-69
6.3.30	HIMC_GetGrpCoordTrans	6-70
6.3.31	HIMC_SetGrpCoordTrans	6-71
6.3.32	HIMC_GetGrpPoseCmd	6-72
6.3.33	HIMC_GetGrpPoseFb	6-73
6.3.34	HIMC_SetGrpLookAheadPrm	6-74
6.3.35	HIMC_SetGrpQueueSize	6-75
6.4	軸群組狀態	6-76
6.4.1	HIMC_IsGrpEnabled	6-76
6.4.2	HIMC_IsGrpMoving	6-77
6.4.3	HIMC_IsGrpInPos	6-78
6.4.4	HIMC_IsGrpErrorStop	6-79
6.5	進階軸群組運動控制	6-80
6.5.1	HIMC_LineAbs	6-80
6.5.2	HIMC_LineRel	6-82
6.5.3	HIMC_CircleAbs	6-84
6.5.4	HIMC_CircleRel	6-86

6.1 概述

HIMC 提供多軸直線與圓弧同動插補功能的軸群組運動命令，包含 LineAbs2D / 3D、LineRel2D / 3D、Arc2D、Circle2D 等。與軸運動命令相較，軸群組運動命令保證群組內各軸的同動性，各軸運動的起始與停止時間一致，控制器會依使用者給定的速度規劃命令對各軸運動速度進行調配。HIMC 控制器的基礎功能，支援最多 4 軸的軸群組運動命令（產品型號 MC-XX-XX-XX-00），若有 5 軸同動(或以上)的軸群組運動功能需求，請向本公司或當地經銷商諮詢相關資訊。

圖 6.1.1 為 HIMC 軸群組運動命令的參數流程圖。由各軸的位置回授（Axis Position Feedback），經過正向運動學計算，得到軸群組在機器座標系統中的位置回授（Cartesian Position Feedback）；而根據使用者所給的目標命令，控制器依軸群組的運動軌跡（Motion Profile）規劃空間中的插補命令（Cartesian Position Command），如圖 6.1.2 所示，並由反向運動學計算各軸馬達的對應位置命令（Axis Position Command）。

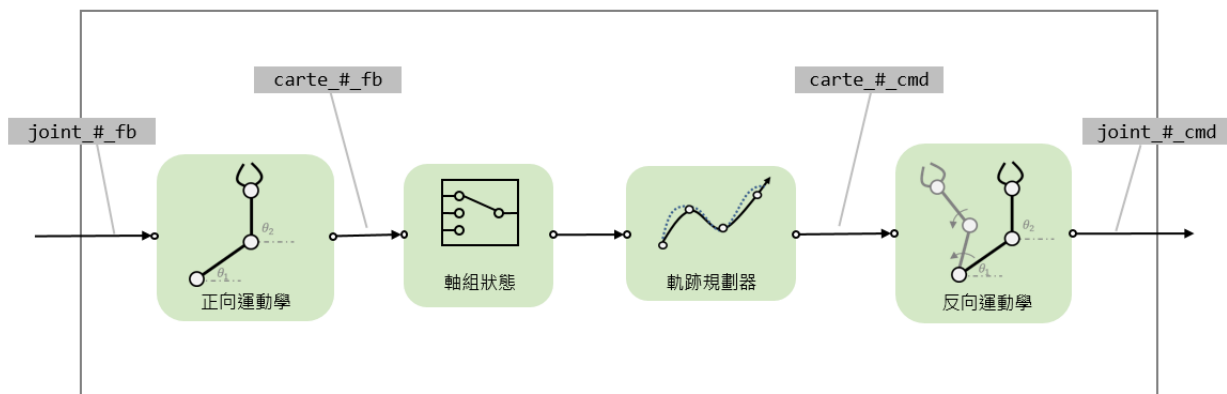


圖 6.1.1

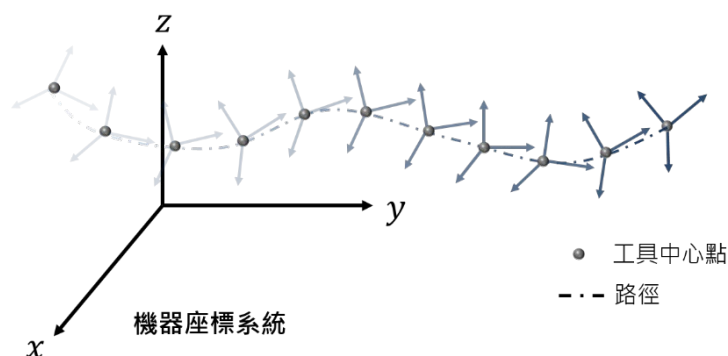


圖 6.1.2

HIMC 會在軸群組運動命令中計算各線段在空間中移動的距離。與軸運動命令不同，其速度規劃是沿著軸群組在空間中移動的方向進行規劃，其移動方向會隨著運動命令的方向而改變。

軸群組運動命令與軸運動命令（請參閱第 3 章）相似，採用 S-Curve 速度規劃，如圖 6.1.3 所示。軸群組在空間中的運動包含平移（Translation）與旋轉（Rotation）兩部分。平移命令由 XYZ 的位置命令所組成；旋轉命令則由 ABC 的旋轉命令所組成。透過軸群組，使用者可設置平移與旋轉的速度規劃參數，包含軌跡規劃器的最大速度、最大加速度、最大減速度與平滑時間。

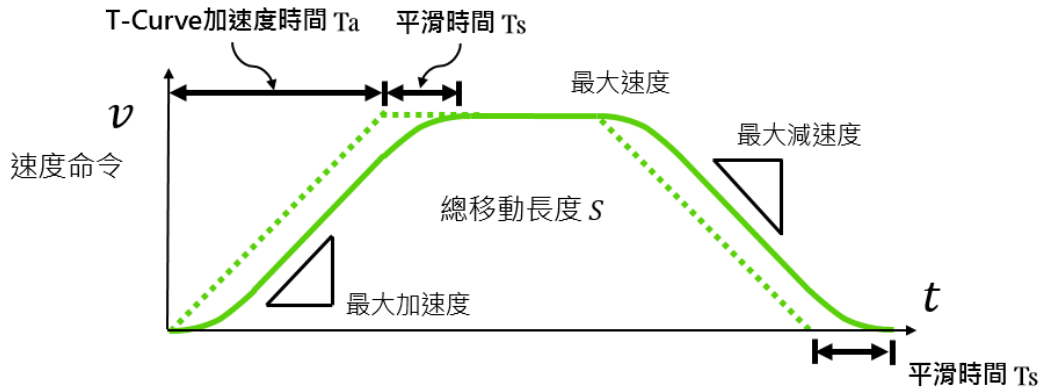


圖 6.1.3

每一個軸群組的運動命令會被視為個別線段（Segment），如圖 6.1.4 所示。在運動過程中，HIMC 會依各個別線段的平移命令與旋轉命令，以及使用者設置的速度規劃參數，計算平移命令與旋轉命令的移動時間，並取移動時間較長者的速度規劃參數作為軸群組的進給速度（Feed Rate）；移動時間較短者，則依此進給速度命令分配後的運動命令移動。

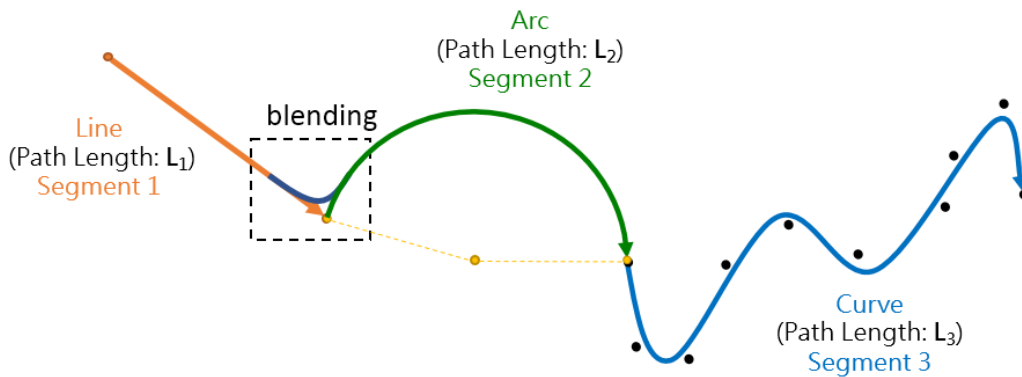


圖 6.1.4

HIMC 內建軸群組命令緩衝空間，每個運動命令的線段都會被放置在此空間，最多能同時接受 512 個運動命令。大於此容量限制的運動命令則會被控制器捨去，並提示錯誤訊息。各個線段的運動命令之間會根據使用者所設定的速度緩衝模式與路徑過渡模式進行速度與路徑的規劃，規劃後的速度曲線與空間路徑有可能會依選擇的模式不同而改變。以圖 6.1.4 為例，若使用速度緩衝模式設定各線段的速度交接，此軸群組的運動總長度為 $S = L_1(\text{Line}) + L_2(\text{Arc}) + L_3(\text{Curve})$ ，詳細說明請參閱 6.1.4 與 6.1.5 節。

軸群組的運動狀態與軸的運動狀態相似，分成移動中 (Moving) 與是否到位 (In-Position) 。在運動過程中，有如圖 3.1.4 的三個階段，包含：

1. 軸群組移動中 (Moving) ，尚未到位 (Not In-Position) 。
2. 軸群組停止移動 (Not Moving) ，但尚未到位 (Not In-Position) 。
3. 軸群組停止移動 (Not Moving) ，已到位 (In-Position) 。

與軸運動命令利用位置目標框 (Position Window) 與反彈跳時間 (Position Window Time) 計算軸運動是否到位不同，軸群組命令藉由判斷軸群組底下所有的軸是否已經到位，來決定軸群組的運動是否已經到位；意即當軸群組已經到位時，軸群組底下的各軸也皆在到位的運動狀態。

6.1.1 軸群組變數

軸群組變數分成運動命令變數、運動規劃變數與狀態變數，使用者可利用 iA Studio 的 Scope Manager(請參閱《iA Studio 軟體使用手冊》4.8 節) 選擇欲觀測的變數。詳細說明如表 6.1.1.1 至表 6.1.1.5。

表 6.1.1.1 軸群組運動命令變數

名稱	變數	單位	描述
Cartesian Position Command	carte_pose_cmd	毫米或角度	軸群組在機器座標系統 (MCS) 中的空間位置命令。為一陣列變數，存放 [X Y Z A B C] 的數值。
Cartesian Velocity Command	carte_vel_cmd	毫米/秒或 角度/秒	軸群組在機器座標系統 (MCS) 中的空間速度命令。為一陣列變數，存放 [X Y Z A B C] 的數值。
Cartesian Position Feedback	carte_pose_fb	毫米或角度	軸群組在機器座標系統 (MCS) 中的空間位置回授。為一陣列變數。存放 [X Y Z A B C] 的數值。
Axis Position Command	joint_pos_cmd	毫米或角度	軸群組在軸座標系統 (ACS) 中的軸位置命令。為一陣列變數。
Axis Velocity Command	joint_vel_cmd	毫米/秒 或 角度/秒	軸群組在軸座標系統 (ACS) 中的軸速度命令。為一陣列變數。
Axis Acceleration Command	joint_acc_cmd	毫米/秒 ² 或 角度/秒 ²	軸群組在軸座標系統 (ACS) 中的軸加速度命令。為一陣列變數。
Axis Position Feedback	joint_pos_fb	毫米或角度	軸群組在軸座標系統 (ACS) 中的軸位置回授。為一陣列變數。
Cartesian Position Error	carte_pose_err	毫米或角度	軸群組在機器座標系統 (MCS) 中的空間位置誤差。為一陣列變數，存放 [X Y Z A B C] 的數值。
Reference Group Position	grp_pg_pos	毫米或角度	軸群組參考位置。根據軸群組命令的運動軌跡，經軌跡規劃器產生的位置設定點。
Reference Group Velocity	grp_pg_vel	毫米/秒或 角度/秒	軸群組參考速度。根據軸群組命令的運動軌跡，經軌跡規劃器產生的速度設定點。
Reference	grp_pg_acc	毫米/秒 ² 或	軸群組參考加速度。根據軸群組命令的運動軌跡，經軌

名稱	變數	單位	描述
Group Acceleration		角度/秒 ²	跡規劃器產生的加速度設定點。

表 6.1.1.2 軸群組運動規劃變數

名稱	變數	單位	描述
Group Max. Linear Profile Velocity	grp_lin_vel	毫米/秒	軸群組線性運動最大速度。不一定會達到。
Group Max. Linear Profile Acceleration	grp_lin_acc	毫米/秒 ²	軸群組線性運動最大加速度。不一定會達到。
Group Max. Linear Profile Deceleration	grp_lin_dec	毫米/秒 ²	軸群組線性運動最大減速度。不一定會達到。
Group Linear Smooth Time	grp_lin_sf	毫秒	軸群組線性運動平滑時間。輸入範圍為 0 ~ 500。增加該值可減少運動期間的機械振動，但會影響總運動時間。
Group Max. Angular Profile Velocity	grp_ang_vel	角度/秒	軸群組旋轉運動最大速度。不一定會達到。
Group Max. Angular Profile Acceleration	grp_ang_acc	角度/秒 ²	軸群組旋轉運動最大加速度。不一定會達到。
Group Max. Angular Profile Deceleration	grp_ang_dec	角度/秒 ²	軸群組旋轉運動最大減速度。不一定會達到。
Group Angular Smooth Time	grp_ang_sf	毫秒	軸群組旋轉運動平滑時間。輸入範圍為 0 ~ 500。增加該值可減少運動期間的機械振動，但會影響總運動時間。

表 6.1.1.3 軸群組狀態變數

名稱	變數	單位	描述
Group Fault Status	grp_fault_status	無	軸群組錯誤狀態，位元定義請參閱表 6.1.1.4。
Group Motion Status	grp_motion_status	無	軸群組運動狀態，位元定義請參閱表 6.1.1.5。

表 6.1.1.4 軸群組錯誤狀態位元定義

位元	名稱	描述	預設反應
0	Error Stop	軸群組處於 error stop 狀態。	無。
1	Axis Fault	從站驅動器的錯誤。	控制器將軸解激磁，軸群組脫離同步。
2	Software Limit	觸發軸的軟體極限。	控制器停止軸的運動，軸群組脫離同步。

表 6.1.1.5 軸群組運動狀態位元定義

位元	名稱	描述	備註
0	Enabled	軸群組激磁狀態。	無。
1	Moving	軸群組移動中。	無。
2	In Position	軸群組到位。	軸群組內所有軸到位。
3	Input Shape	開啟軸群組 Input Shape 濾波器。	請參閱 13.1 節。

6.1.2 座標系統

表 6.1.2.1 為 HIMC 的座標系統定義與說明，包含軸座標系統 (Axis Coordinate System · ACS)、機器座標系統 (Machine Coordinate System · MCS)、產品座標系統 (Product Coordinate System · PCS)、工件座標系統 (Workpiece Coordinate System · WCS)、全局座標系統 (Global Coordinate System) 與座標偏移量 (Coordinate Offset)。

表 6.1.2.1

HMPL 定義	描述
CS_ACS	軸座標系統，與個別馬達運動有關。
CS_MCS	機器座標系統 (又稱「大地座標系統」或「基座標系統」)。 在機器上具有固定原點的座標系統，藉運動學轉換 (參閱 6.1.3 節) 與 ACS 連接。 共 6 個維度來表示空間中的位置與方位角 (3 平移、3 旋轉)。
CS_PCS	產品座標系統 (在 CNC 程式中稱「程式座標系統」)，依附在產品或工件上。 可設定座標轉換參數。
CS_WCS# (#=1~15)	工件座標系統，用來設定工件零點。最多提供 15 個獨立的工件座標系統，預設無偏移量。 相依於產品座標系統，可設定座標轉換參數。
CS_GLOBAL	全局座標系統，用來設定全局零點。可建立各個軸群組的全局空間關係。 目前不支援。
CS_OFFSET	座標偏移量，用來設定暫時零點。預設無偏移量，即偏移量座標原點為機器座標原點。 相依於產品座標系統，可設定座標轉換參數。

圖 6.1.2.1 以具有兩旋轉軸的 SCARA 機器人為例，說明 ACS、MCS 與 PCS 之間的關係。ACS 與 MCS 之間透過正逆向運動學轉換（參閱 6.1.3 節），而 MCS 與 PCS 之間則存在相對座標的轉換關係，透過座標的平移與旋轉得到在座標系上的位置。

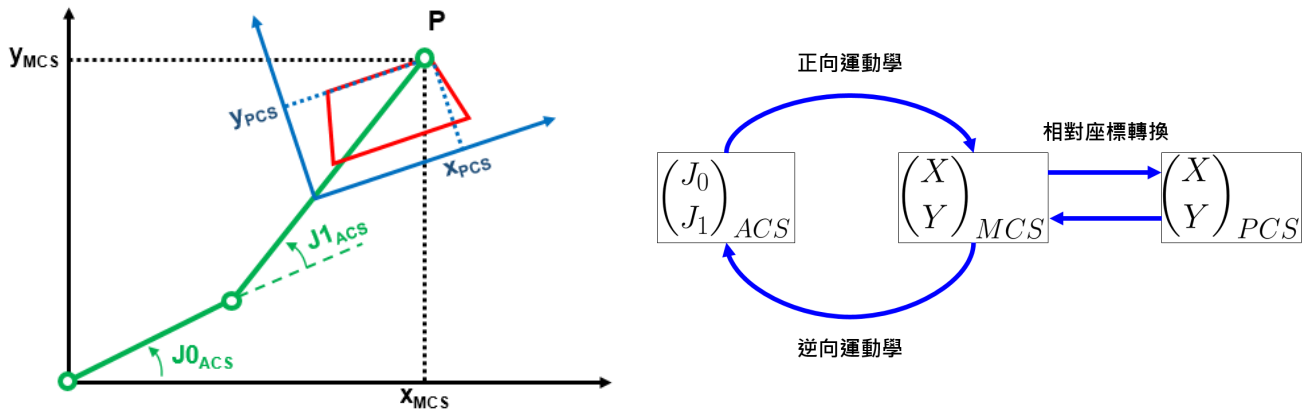


圖 6.1.2.1

將 MCS 轉換至 PCS 時，HIMC 可依需求設定機台的工件座標（WCS1~15）與座標偏移量（OFFSET）。其中，座標系統的設定使用 3 個平移自由度（X、Y、Z）與 3 個旋轉自由度（A、B、C）來表示空間中的姿態（Pose）。

HIMC 採用固定座標系下的 Roll-Pitch-Yaw 旋轉慣例（Rotation Convention）。如圖 6.1.2.2 所示，沿 X 軸旋轉的自由度為 Roll，為角度 A；沿 Y 軸旋轉的自由度為 Pitch，為角度 B；沿 Z 軸旋轉的自由度為 Yaw，為角度 C。此旋轉慣例等同於使用 Tait-Bryan angles 的 ZYX 順序來表示物體在空間中的方位角（Orientation），如圖 6.1.2.3 所示。

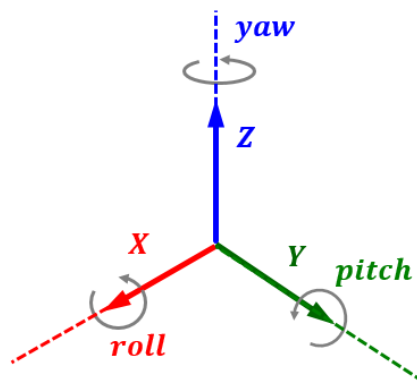


圖 6.1.2.2

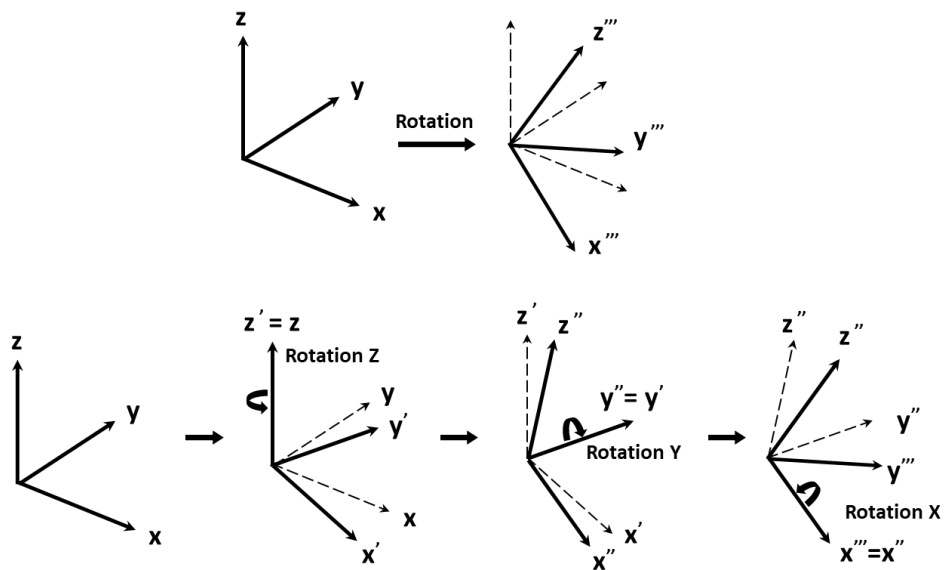


圖 6.1.2.3

假設無座標偏移量，各個 WCS 與 MCS 的對應關係如圖 6.1.2.4 所示。

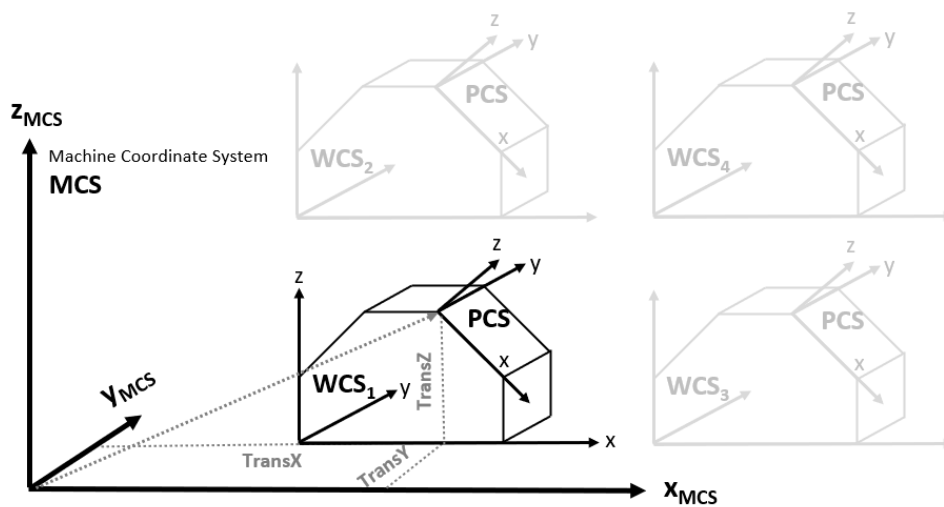


圖 6.1.2.4

若加入座標偏移量，WCS 與 MCS 的對應關係則如圖 6.1.2.5 所示，兩者之間會加入座標偏移量的轉換。

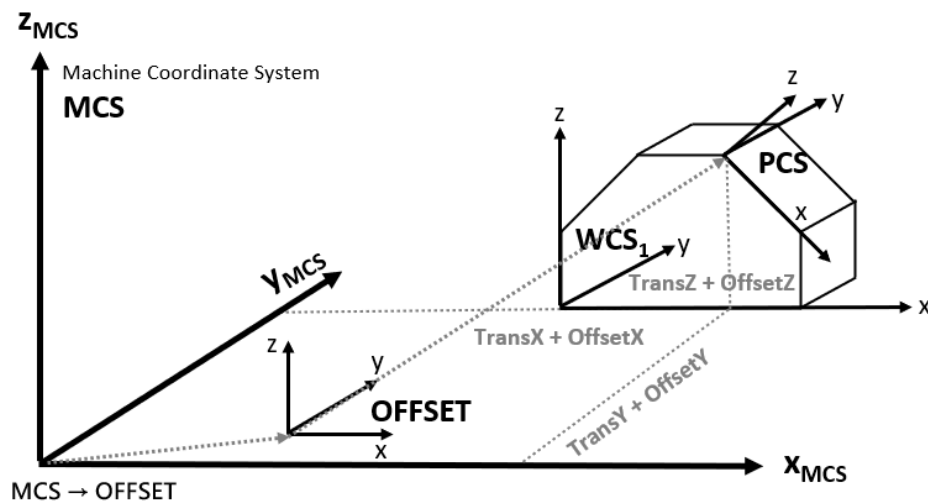


圖 6.1.2.5

依上述座標系統的功能，使用者可在 HIMC 定義座標轉換的參數，依應用需求建立各座標系統之間的轉換關係。圖 6.1.2.6 為各個座標系統之間的關係示意圖，為了讓使用者容易理解，圖中僅採用 XY 平面的座標作為示意，實際應用可設定 6 個自由度的座標轉換參數（X、Y、Z、A、B、C）。

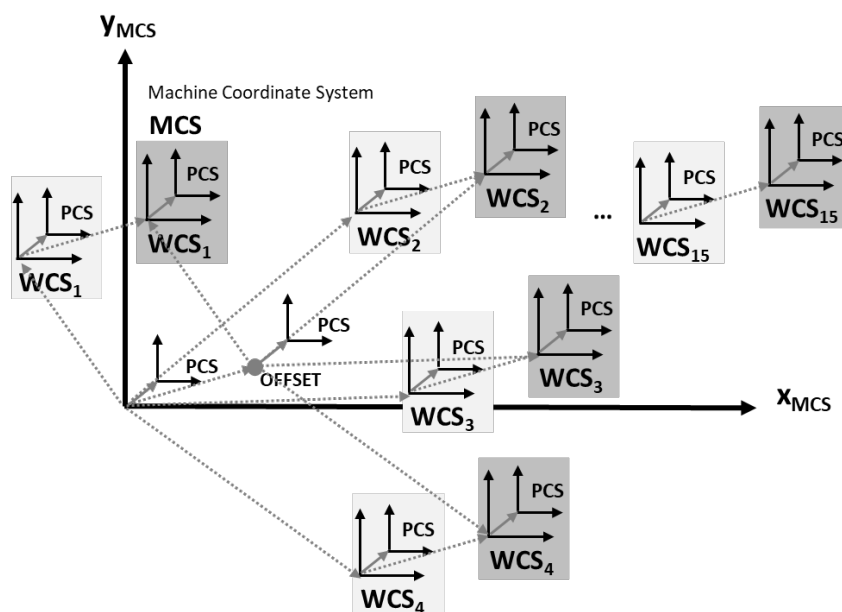


圖 6.1.2.6

6.1.3 運動學

運動學主要處理 ACS (軸座標系統) 與 MCS (機器座標系統) 之間的轉換關係。由 ACS 中各軸的位置回授計算在 MCS 的空間座標位置，為正向運動學 (Forward Kinematics)；反之，由 MCS 的空間座標位置計算在 ACS 中各軸的位置，則為逆向運動學 (Inverse Kinematics)。表 6.1.3.1 為 HIMC 提供的運動學組態定義。

表 6.1.3.1

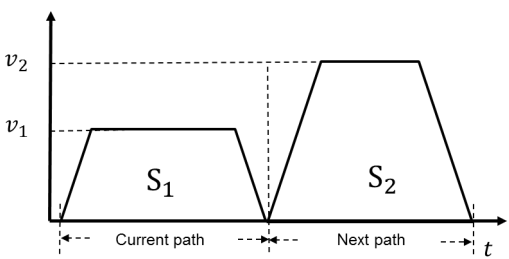
ID	名稱	描述
1	Cartesian	將協調運動中的每個軸分別映射到笛卡爾坐標系的 X、Y、Z、A、B、C 軸。關節空間中，最多可允許 6 個軸。(軸群組的預設值)
2	SCARA	(未開放)
3	WAFER	(未開放)
4	6-Axis Articulated Robot	(未開放)

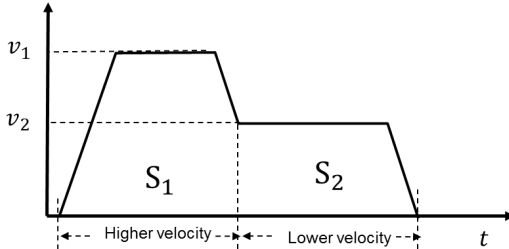
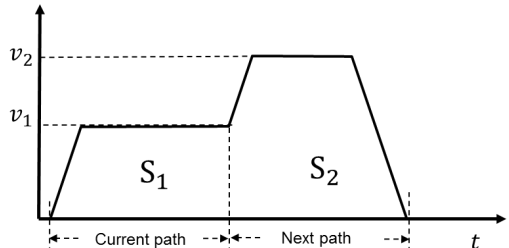
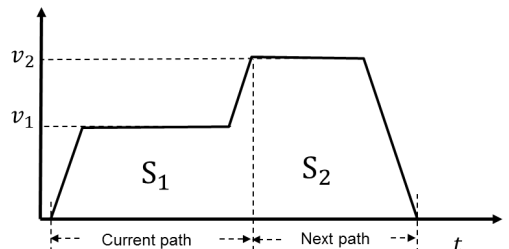
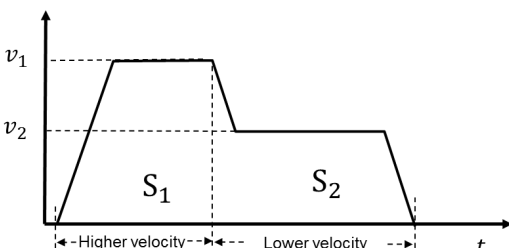
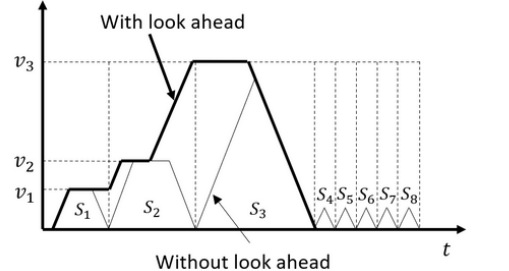
6.1.4 速度緩衝模式

速度緩衝模式決定了相鄰路徑端點處的速度，使用者可透過此設置規劃相鄰兩段路徑的軌跡速度。表 6.1.4.1 為 HIMC 提供的速度緩衝模式定義。

註：若當前路徑到位前，軸群組沒有接收到下一個路徑命令，速度緩衝模式功能將會被忽略。

表 6.1.4.1

API 定義	描述
kBM_Aborting	中止當前的運動，並立即執行下一個運動。
kBM_Buffered	完成當前的路徑 (current path) 後，再開始下一個路徑 (next path)。 

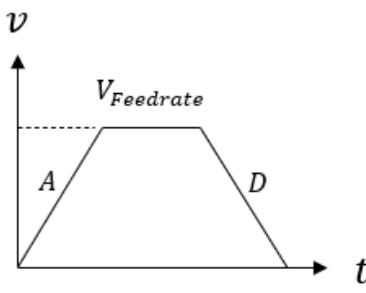
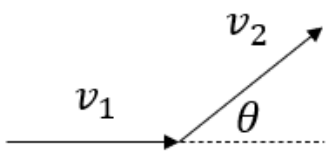
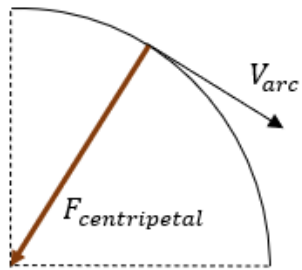
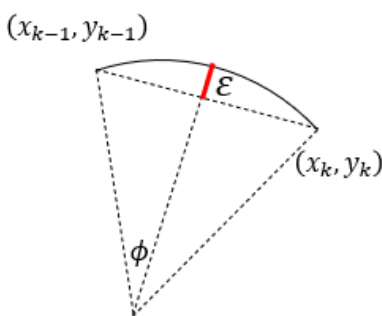
kBM_BlendingLow	其交接速度為兩路徑中較低的速度。(Blending) 
kBM_BlendingPrevious	其交接速度為當前路徑的速度。(Blending) 
kBM_BlendingNext	其交接速度為下一路徑的速度。(Blending) 
kBM_BlendingHigh	其交接速度為兩路徑中較高的速度。(Blending) 
kBM_LookAhead	由速度前瞻功能計算的運動距離與給定的限制條件，由控制器決定移動的速度。 

速度前瞻運動 (Look Ahead Motion)

速度前瞻功能是控制器於軸群組命令緩衝空間中的運動命令，藉由往前預看的方式，計算運動命令的移動距離，並提前對運動路徑上的轉折處，依機構的加速度限制條件進行加減速度的運動規劃。表 6.1.4.2，說明速度前瞻運動功能在運動過程中，所計算的速度參數，包含：

- (a) 進給速度($V_{Feedrate}$)：軸群組命令的速度條件。
- (b) 角落速度(V_{corner})：依使用者設定的轉角加速度(A_{corner})限制，與運動方向的夾角計算而得。轉角加速度限制的預設值為 5000 mm/s^2 。
- (c) 圓弧速度(V_{arc})：依使用者設定的圓弧加速度(A_{arc})限制條件計算而得。圓弧加速度的預設值為 100 mm/s^2 。
- (d) 最大弦誤差速度(V_{chord})：依使用者設定的弦誤差(ϵ)條件計算而得。弦誤差預設不啟用，預設值為 0。

表 6.1.4.2

	
(a) 進給速度 $V_{Feedrate}$	(b) 角落速度 $V_{corner} = \frac{A_{corner} * \Delta T}{1 - \cos\theta}$
	
(c) 圓弧速度 $V_{arc} = \sqrt{A_{arc} * r}$	(d) 最大弦誤差速度 $V_{chord} = \frac{2 * r * \phi}{\Delta T}$

使用速度前瞻功能時，命令緩衝區的指令愈多，控制器運動核心的計算時間會增加，可能出現 MCK Overload 的錯誤訊息。此時可透過增加通訊周期或減少緩衝區大小的方式來排除此錯誤。

6.1.5 路徑過渡模式

路徑過渡模式決定了相鄰路徑間的過渡曲線類型，如表 6.1.5.1 所示。透過此設置，控制器會依據使用者所設定的模式與參數，在前後兩運動命令之間進行轉角平滑化的計算，決定路徑平滑的起點與終點，並搭配速度緩衝模式及運動軌跡產生結果，如表 6.1.5.2 所示。請注意此規劃方式會影響原先的運動軌跡。

表 6.1.5.1

API 定義	描述
KTM_None	無：不插入過渡曲線（預設模式）
kTM_StartVelocity（未開放）	使用速度參數作為過渡路徑的起始速度，目前不支援此功能
kTM_Velocity	使用速度參數作為過渡路徑的等速速度
kTM_CornerDistance	使用距離參數設定過渡路徑平滑的起點到角落的距離
kTM_MaxCornerDeviation	使用誤差參數決定過渡路徑與原路徑最大誤差值
kTM_MaxCornerCurvature	使用曲率參數決定過渡路徑之最大曲率數值

註：

- (1) 若路徑過渡模式計算出的修剪範圍超過任一線段的長度，該線段間的過渡模式功能將會被忽略。
- (2) 過渡參數設定方式參考 6.3.26 節。
- (3) kTM_Velocity 與 kTM_CornerDistance 行為相同。

表 6.1.5.2 路徑過渡模式與速度緩衝模式

路徑過渡 模式	Trajectory of TCP	Speed of TCP	
		kBM_Buffered	Others
kTM_None			
kTM_StartVelocity			
kTM_CornerRadius			
kTM_MaxCornerDeviation			
kTM_MaxCornerCurvature			

6.2 軸群組運動控制

6.2.1 HIMC_EnableGroup

用途

致能軸群組。

語法

```
int HIMC_EnableGroup(
    int ctrl_id,
    int group_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

group_id [in] 軸群組編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

在執行此函式前，須激磁軸群組內的所有軸。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_EnableGroup
LabVIEW	HIMC Enable Group.vi
Python	EnableGroup

6.2.2 HIMC_DisableGroup

用途

停用軸群組。

語法

```
int HIMC_DisableGroup(  
    int ctrl_id,  
    int group_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisableGroup
LabVIEW	HIMC Disable Group.vi
Python	DisableGroup

6.2.3 HIMC_ResetGroup

用途

更改軸群組的狀態：Group Error Stop → Group Standby。

語法

```
int HIMC_ResetGroup(  
    int ctrl_id,  
    int group_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

group_id [in] 軸群組編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ResetGroup
LabVIEW	HIMC Reset Group.vi
Python	ResetGroup

6.2.4 HIMC_StopGroup

用途

停止軸群組的運動。

語法

```
int HIMC_StopGroup(  
    int ctrl_id,  
    int group_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

此函式會清除軸群組原有的路徑規劃。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_StopGroup
LabVIEW	HIMC Stop Group.vi
Python	StopGroup

6.2.5 HIMC_HaltGroup

用途

暫停軸群組的運動，軸運動速度被設定為 0。

語法

```
int HIMC_HaltGroup(  
    int ctrl_id,  
    int group_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

group_id [in] 軸群組編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

若軸群組尚未到位，軸群組會繼續移動。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_HaltGroup
LabVIEW	HIMC Halt Group.vi
Python	HaltGroup

6.2.6 HIMC_ResumeGroup

用途

由軸群組暫停的狀態中恢復軸群組的運動。

語法

```
int HIMC_ResumeGroup(  
    int ctrl_id,  
    int group_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

group_id [in] 軸群組編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ResumeGroup
LabVIEW	HIMC Resume Group.vi
Python	ResumeGroup

6.2.7 HIMC_JogGroup

用途

命令一個軸群組，在機器座標系統中的指定方向以特定速度持續移動。

語法

```
int HIMC_JogGroup(
    int    ctrl_id,
    int    group_id,
    int    carte_dir,
    double jog_vel
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- carte_dir [in] 機器座標系統中的運動方向。
數字 0 ~ 5 依序代表機器座標系統中的 6-DOF { X, Y, Z, A, B, C }。
- jog_vel [in] 移動速度的值。
數值的正負值表示往運動方向的同方向或反方向移動。
參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_JogGroup
LabVIEW	HIMC Jog Group.vi
Python	JogGroup

6.2.8 HIMC_JogGroupAxis

用途

命令一個軸群組中的特定軸，在軸座標系統中以特定速度持續移動。

語法

```
int HIMC_JogGroupAxis(
    int    ctrl_id,
    int    group_id,
    int    grp_axis,
    double jog_vel
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- grp_axis [in] 軸群組中的軸編號。
數字 0 ~ 8 依序代表各軸加入軸群組的順序。
- jog_vel [in] 移動速度的值。
數值的正負值表示往運動方向的同方向或反方向移動。
參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_JogGroupAxis
LabVIEW	HIMC Jog Group Axis.vi
Python	JogGroupAxis

6.2.9 HIMC_LineAbs2D

用途

命令一個軸群組，二維內插線性移動至機器座標系統中的絕對位置。

語法

```
int HIMC_LineAbs2D(
    int    ctrl_id,
    int    group_id,
    double end_x,
    double end_y
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- end_x [in] 絕對目標位置在 X 軸的值。
- end_y [in] 絕對目標位置在 Y 軸的值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_LineAbs2D
LabVIEW	HIMC Line Abs2D.vi
Python	LineAbs2D

6.2.10 HIMC_LineAbs3D

用途

命令一個軸群組，三維內插線性移動至機器座標系統中的絕對位置。

語法

```
int HIMC_LineAbs3D(  
    int    ctrl_id,  
    int    group_id,  
    double end_x,  
    double end_y,  
    double end_z  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
end_x [in]	絕對目標位置在 X 軸的值。
end_y [in]	絕對目標位置在 Y 軸的值。
end_z [in]	絕對目標位置在 Z 軸的值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_LineAbs3D
LabVIEW	HIMC Line Abs3D.vi
Python	LineAbs3D

6.2.11 HIMC_LineRel2D

用途

命令一個軸群組，二維內插線性移動至機器座標系統中的相對位置。

語法

```
int HIMC_LineRel2D(
    int    ctrl_id,
    int    group_id,
    double distance_x,
    double distance_y
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- distance_x [in] 相對距離在 X 軸的值。
- distance_y [in] 相對距離在 Y 軸的值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_LineRel2D
LabVIEW	HIMC Line Rel2D.vi
Python	LineRel2D

6.2.12 HIMC_LineRel3D

用途

命令一個軸群組，三維內插線性移動至機器座標系統中的相對位置。

語法

```
int HIMC_LineRel3D(
    int    ctrl_id,
    int    group_id,
    double distance_x,
    double distance_y,
    double distance_z
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
distance_x [in]	相對距離在 X 軸的值。
distance_y [in]	相對距離在 Y 軸的值。
distance_z [in]	相對距離在 Z 軸的值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_LineRel3D
LabVIEW	HIMC Line Rel3D.vi
Python	LineRel3D

6.2.13 HIMC_Arc2D

用途

命令一個軸群組，二維內插圓弧移動至機器座標系統中的絕對位置。

語法

```
int HIMC_Arc2D(
    int    ctrl_id,
    int    group_id,
    double border_x,
    double border_y,
    double end_x,
    double end_y
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- border_x [in] 絕對中繼位置在 X 軸的值。
- border_y [in] 絕對中繼位置在 Y 軸的值。
- end_x [in] 絕對終點位置在 X 軸的值。
- end_y [in] 絕對終點位置在 Y 軸的值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

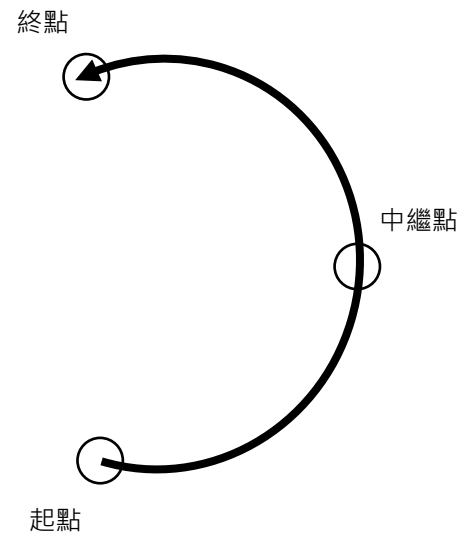
C#	HIMC_Arc2D
LabVIEW	HIMC Arc2D.vi
Python	Arc2D

優點

使用者可指定中繼點（運動中的最遠點），並確保機器可取得此點。

缺點

在單一命令中，其角度被限制為 $<2\pi$ 。



6.2.14 HIMC_ArcCW2D

用途

命令一個軸群組，二維內插圓弧順時針移動至機器座標系統中的絕對位置。

語法

```
int HIMC_ArcCW2D(
    int    ctrl_id,
    int    group_id,
    double center_x,
    double center_y,
    double end_x,
    double end_y
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- center_x [in] 絕對中心位置在 X 軸的值。
- center_y [in] 絕對中心位置在 Y 軸的值。
- end_x [in] 絕對終點位置在 X 軸的值。
- end_y [in] 絕對終點位置在 Y 軸的值。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ArcCW2D
LabVIEW	HIMC Arc CW2D.vi
Python	ArcCW2D

6.2.15 HIMC_ArcCCW2D

用途

命令一個軸群組，二維內插圓弧逆時針移動至機器座標系統中的絕對位置。

語法

```
int HIMC_ArcCCW2D(  
    int    ctrl_id,  
    int    group_id,  
    double center_x,  
    double center_y,  
    double end_x,  
    double end_y  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- center_x [in] 絕對中心位置在 X 軸的值。
- center_y [in] 絕對中心位置在 Y 軸的值。
- end_x [in] 絕對終點位置在 X 軸的值。
- end_y [in] 絕對終點位置在 Y 軸的值。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ArcCCW2D
LabVIEW	HIMC Arc CCW2D.vi
Python	ArcCCW2D

6.2.16 HIMC_ArcAngle2D

用途

命令一個軸群組依給定角度，二維內插圓弧移動至機器座標系統中的絕對位置。

語法

```
int HIMC_ArcAngle2D(  
    int    ctrl_id,  
    int    group_id,  
    double center_x,  
    double center_y,  
    double angle  
);
```

參數

- | | |
|---------------|---|
| ctrl_id [in] | HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。 |
| group_id [in] | 軸群組編號。 |
| center_x [in] | 絕對中心位置在 X 軸的值。 |
| center_y [in] | 絕對中心位置在 Y 軸的值。 |
| angle [in] | 起點與終點相對於絕對中心位置的夾角，決定了圓弧移動的方向及旋轉角度。
參數單位：deg (角度) |

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

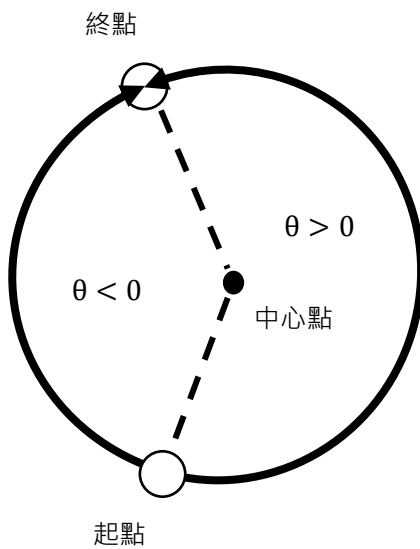
參數 angle 代表圓軌跡旋轉的方向，若 angle > 0：逆時針移動；若 angle < 0：順時針移動。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ArcAngle2D
LabVIEW	HIMC Arc Angle2D.vi
Python	ArcAngle2D



θ 的值決定了圓弧移動的方向。

$\theta > 0$: 逆時針移動

$\theta < 0$: 順時針移動

6.2.17 HIMC_Circle2D

用途

命令一個軸群組，二維內插圓周運動至機器座標系統中的絕對位置。

語法

```
int HIMC_Circle2D(  
    int    ctrl_id,  
    int    group_id,  
    double center_x,  
    double center_y,  
    double end_x,  
    double end_y,  
    int    turns  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
center_x [in]	絕對中心位置在 X 軸的值。
center_y [in]	絕對中心位置在 Y 軸的值。
end_x [in]	絕對終點位置在 X 軸的值。
end_y [in]	絕對終點位置在 Y 軸的值。
turns [in]	相對於起點的圓周運動圈數，決定了圓周運動的方向及總角度。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

- (1) 參數 turns 代表圓軌跡旋轉的方向，若 turns >= 0：逆時針移動；若 turns < 0：順時針移動。
- (2) 當 ||turns|| <= 1 時，圓軌跡的總移動角度<360°；若圓軌跡總移動角度>=360°（即一圈，或一圈以上），||turns|| 須 >= 2。
- (3) 使用此函式時，turns = 0 與 turns = 1 的行為定義相同。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

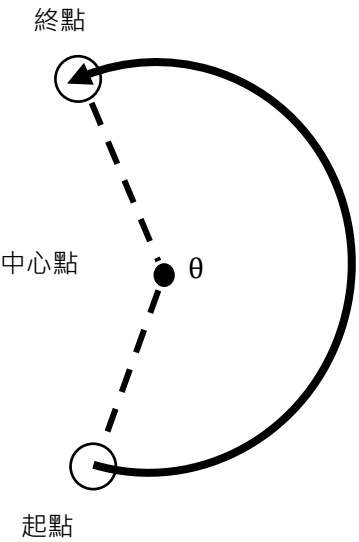
C#	HIMC_Circle2D
LabVIEW	HIMC Circle2D.vi
Python	Circle2D

優點

無角度限制。

缺點

使用者無法指定中繼點（運動中的最遠點）。
因此，機器不一定可以取得此點。



圈數的值決定了圓周運動的方向。

※ 角度 = $\theta + \text{圈數} \times 360$

圈數 ≥ 0 為正轉，圈數 < 0 為反轉。（注意：圈數 = 0 與 圈數 = 1 的運動軌跡相同。）

下表以 $\theta = 210^\circ$ 為例。

圈數	計算	角度
-2	$210 - 2 \times 360^\circ$	-510°
-1	$210 - 1 \times 360^\circ$	-150°
0	$210 + 0 \times 360^\circ$	210°
1	$210 + 0 \times 360^\circ$	210°
2	$210 + 1 \times 360^\circ$	570°

6.3 軸群組設定

6.3.1 HIMC_AddAxesToGrp

用途

將軸加入一個具有特定序列的軸群組中。

語法

```
int HIMC_AddAxesToGrp(
    int ctrl_id,
    int group_id,
    int num_of_axes,
    int *p_axis_list,
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- num_of_axes [in] 要被加入軸群組的軸數量。(最大為 9)
- p_axis_list [in] 指標型態的記憶體，用來儲存軸 ID 的序列。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_AddAxesToGrp
LabVIEW	HIMC Add Axes To Grp.vi
Python	AddAxesToGrp

6.3.2 HIMC_RemoveAxisFromGrp

用途

從一個具有特定序列的軸群組中移除最後一軸。

語法

```
int HIMC_RemoveAxisFromGrp(  
    int ctrl_id,  
    int group_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

group_id [in] 軸群組編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_RemoveAxisFromGrp
LabVIEW	HIMC Remove Axis From Grp.vi
Python	RemoveAxisFromGrp

6.3.3 HIMC_SetupGroup

用途

設定並致能一個具有特定序列的軸群組。

語法

```
int HIMC_SetupGroup(
    int ctrl_id,
    int group_id,
    int num_of_axes,
    int *p_axis_list
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

group_id [in] 軸群組編號。

num_of_axes [in] 軸數量。(最大為 9)

p_axis_list [in] 指標型態的記憶體，用來儲存軸 ID 的序列。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetupGroup
LabVIEW	HIMC Setup Group.vi
Python	SetupGroup

6.3.4 HIMC_UngrpAllAxes

用途

拆散軸群組。

語法

```
int HIMC_UngrpAllAxes(  
    int ctrl_id,  
    int group_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

group_id [in] 軸群組編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_UngrpAllAxes
LabVIEW	HIMC Ungrp All Axes.vi
Python	UngrpAllAxes

6.3.5 HIMC_GetGroupID

目的

取得軸所屬的軸群組 ID。

語法

```
int HIMC_GetGroupID(
    int ctrl_id,
    int axis_id,
    int *p_group_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

p_group_id [out] 指標型態的記憶體，用來儲存軸所屬的軸群組 ID。
若其值為-1，代表此軸不屬於任何軸群組。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGroupID
LabVIEW	HIMC Get Group ID.vi
Python	GetGroupID

6.3.6 HIMC_SetGrpMotionProfile

用途

設置軸群組的 TCP 線性運動參數。

語法

```
int HIMC_SetGrpMotionProfile(  
    int    ctrl_id,  
    int    group_id,  
    double max_velocity,  
    double max_acceleration,  
    double max_deceleration,  
    double smooth_time  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
max_velocity [in]	軸群組線性運動的最大速度。 參數單位：mm/s (毫米/秒) 輸入範圍：0 ~ 5000
max_acceleration [in]	軸群組線性運動的最大加速度。 參數單位：mm/s ² (毫米/秒 ²) 輸入範圍：>0 ~ 50000 (加速度不能為 0)
max_deceleration [in]	軸群組線性運動的最大減速度。 參數單位：mm/s ² (毫米/秒 ²) 輸入範圍：>0 ~ 50000 (加速度不能為 0)
smooth_time [in]	軸群組線性運動的平滑時間。 參數單位：ms (毫秒) 輸入範圍：0 ~ 500

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

軸群組線性運動的參數預設值分別為 [100 (速度) , 5000 (加速度) , 5000 (減速度) , 50 (平滑時間)]。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpMotionProfile
LabVIEW	HIMC Set Grp Motion Profile.vi
Python	SetGrpMotionProfile

6.3.7 HIMC_SetGrpAngMotionProfile

用途

設置軸群組的 TCP 旋轉運動參數。

語法

```
int HIMC_SetGrpAngMotionProfile(  
    int    ctrl_id,  
    int    group_id,  
    double max_velocity,  
    double max_acceleration,  
    double max_deceleration,  
    double smooth_time  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
max_velocity [in]	軸群組旋轉運動的最大速度。 參數單位：deg/s (角度/秒) 輸入範圍：0 ~ 7200
max_acceleration [in]	軸群組旋轉運動的最大加速度。 參數單位：deg/s ² (角度/秒 ²) 輸入範圍：>0 ~ 72000 (加速度不能為 0)
max_deceleration [in]	軸群組旋轉運動的最大減速度。 參數單位：deg/s ² (角度/秒 ²) 輸入範圍：>0 ~ 72000 (加速度不能為 0)
smooth_time [in]	軸群組旋轉運動的平滑時間。 參數單位：ms (毫秒) 輸入範圍：0 ~ 500

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

軸群組旋轉運動的參數預設值分別為 [360 (速度) , 1800 (加速度) , 1800 (減速度) , 50 (平滑時間)]。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpAngMotionProfile
LabVIEW	HIMC Set Grp Ang Motion Profile.vi
Python	SetGrpAngMotionProfile

6.3.8 HIMC_GetGrpKin

用途

取得軸群組的運動學模式。

語法

```
int HIMC_GetGrpKin(  
    int ctrl_id,  
    int group_id,  
    int *p_grp_kin  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- p_grp_kin [out] 指標型態的記憶體，用來儲存軸群組的運動學模式。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpKin
LabVIEW	HIMC Get Grp Kin.vi
Python	GetGrpKin

6.3.9 HIMC_SetGrpKin

用途

設置軸群組的運動學模式。

語法

```
int HIMC_SetGrpKin(
    int ctrl_id,
    int group_id,
    int kin_type
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

group_id [in] 軸群組編號。

kin_type [in] 軸群組的新運動學模式，請參閱 6.1.3 節。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpKin
LabVIEW	HIMC Set Grp Kin.vi
Python	SetGrpKin

6.3.10 HIMC_GetGrpVel

用途

取得軸群組的運動速度目標值。

語法

```
int HIMC_GetGrpVel(
    int    ctrl_id,
    int    group_id,
    double *p_grp_vel
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- p_grp_vel [out] 指標型態的記憶體，用來儲存軸群組的運動速度目標值。
參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpVel
LabVIEW	-
Python	GetGrpVel

6.3.11 HIMC_SetGrpVel

用途

設置軸群組的運動速度目標值。

語法

```
int HIMC_SetGrpVel(
    int    ctrl_id,
    int    group_id,
    double vel
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- vel [in] 軸群組的新運動速度目標值。
 參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)
 輸入範圍：0 ~ 5000

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpVel
LabVIEW	HIMC Set Grp Vel.vi
Python	SetGrpVel

6.3.12 HIMC_GetGrpAcc

用途

取得軸群組的運動加速度目標值。

語法

```
int HIMC_GetGrpAcc(
    int    ctrl_id,
    int    group_id,
    double *p_grp_acc
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- p_grp_acc [out] 指標型態的記憶體，用來儲存軸群組的運動加速度目標值。
參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpAcc
LabVIEW	-
Python	GetGrpAcc

6.3.13 HIMC_SetGrpAcc

用途

設置軸群組的運動加速度目標值。

語法

```
int HIMC_SetGrpAcc(
    int    ctrl_id,
    int    group_id,
    double acc
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- acc [in] 軸群組的新運動加速度目標值。
參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)
輸入範圍：>0 ~ 50000 (加速度不能為 0)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpAcc
LabVIEW	HIMC Set Grp Acc.vi
Python	SetGrpAcc

6.3.14 HIMC_SetGrpAccTime

用途

設置軸群組的加速度時間。

語法

```
int HIMC_SetGrpAccTime(  
    int    ctrl_id,  
    int    group_id,  
    double acc_time  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- acc_time [in] 軸群組的加速度時間。
 參數單位：ms (毫秒)
 輸入範圍：非零正值

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpAccTime
LabVIEW	HIMC Set Grp Acc Time.vi
Python	SetGrpAccTime

6.3.15 HIMC_GetGrpDec

用途

取得軸群組的運動減速度目標值。

語法

```
int HIMC_GetGrpDec(
    int    ctrl_id,
    int    group_id,
    double *p_grp_dec
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- p_grp_dec [out] 指標型態的記憶體，用來儲存軸群組的運動減速度目標值。
 參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpDec
LabVIEW	-
Python	GetGrpDec

6.3.16 HIMC_SetGrpDec

用途

設置軸群組的運動減速度目標值。

語法

```
int HIMC_SetGrpDec(  
    int    ctrl_id,  
    int    group_id,  
    double dec  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- dec [in] 軸群組的新運動減速度目標值。
 參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)
 輸入範圍：>0 ~ 50000 (減速度不能為 0)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpDec
LabVIEW	HIMC Set Grp Dec.vi
Python	SetGrpDec

6.3.17 HIMC_SetGrpDecTime

用途

設置軸群組的減速度時間。

語法

```
int HIMC_SetGrpDecTime(
    int    ctrl_id,
    int    group_id,
    double dec_time
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- dec_time [in] 軸群組的減速度時間。
 參數單位：ms (毫秒)
 輸入範圍：非零正值

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpDecTime
LabVIEW	HIMC Set Grp Dec Time.vi
Python	SetGrpDecTime

6.3.18 HIMC_GetGrpSMTime

用途

取得軸群組的平滑時間。

語法

```
int HIMC_GetGrpSMTime(  
    int    ctrl_id,  
    int    group_id,  
    double *p_grp_smooth_time  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in]

軸群組編號。
- p_grp_smooth_time [out]

指標型態的記憶體，用來儲存軸群組的平滑時間。
參數單位：ms (毫秒)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpSMTime
LabVIEW	HIMC Get Grp SM Time.vi
Python	GetGrpSMTime

6.3.19 HIMC_SetGrpSMTime

用途

設置軸群組的平滑時間。

語法

```
int HIMC_SetGrpSMTime(
    int    ctrl_id,
    int    group_id,
    double smooth_time
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- smooth_time [in] 軸群組的新平滑時間。
 參數單位：ms (毫秒)
 輸入範圍：0 ~ 500

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpSMTime
LabVIEW	HIMC Set Grp SM Time.vi
Python	SetGrpSMTime

6.3.20 HIMC_GetGrpCoordSys

用途

取得軸群組的座標系統。

語法

```
int HIMC_GetGrpCoordSys(  
    int ctrl_id,  
    int group_id,  
    int *p_grp_coord_sys  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- p_grp_coord_sys [out] 指標型態的記憶體，用來儲存軸群組的座標系統。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpCoordSys
LabVIEW	HIMC Get Grp Coord Sys.vi
Python	GetGrpCoordSys

6.3.21 HIMC_SetGrpCoordSys

用途

設置軸群組的座標系統。

語法

```
int HIMC_SetGrpCoordSys(
    int ctrl_id,
    int group_id,
    int coord_sys
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- coord_sys [in] 軸群組的新座標系統，請參閱 6.1.2 節。
範例：1. kCoord_MCS
2. kCoord_WCS1 | kCoord_PCS
3. kCoord_OFFSET | kCoord_WCS2 | kCoord_PCS

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpCoordSys
LabVIEW	HIMC Set Grp Coord Sys.vi
Python	SetGrpCoordSys

6.3.22 HIMC_GetGrpBufferMode

用途

取得軸群組的速度緩衝模式。

語法

```
int HIMC_GetGrpBufferMode(  
    int ctrl_id,  
    int group_id,  
    int *p_grp_buffer_mode  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- p_grp_buffer_mode [out] 指標型態的記憶體，用來儲存軸群組的速度緩衝模式。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpBufferMode
LabVIEW	HIMC Get Grp Buffer Mode.vi
Python	GetGrpBufferMode

6.3.23 HIMC_SetGrpBufferMode

用途

設置軸群組的速度緩衝模式。

語法

```
int HIMC_SetGrpBufferMode(
    int ctrl_id,
    int group_id,
    int buffer_mode
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- buffer_mode [in] 軸群組的新速度緩衝模式，請參閱 6.1.4 節。
 輸入範圍：0 ~ 5

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpBufferMode
LabVIEW	HIMC Set Grp Buffer Mode.vi
Python	SetGrpBufferMode

6.3.24 HIMC_GetGrpTransMode

用途

取得軸群組的路徑過渡模式。

語法

```
int HIMC_GetGrpTransMode(  
    int ctrl_id,  
    int group_id,  
    int *p_grp_trans_mode  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- p_grp_trans_mode [out] 指標型態的記憶體，用來儲存軸群組的路徑過渡模式。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpTransMode
LabVIEW	HIMC Get Grp Trans Mode.vi
Python	GetGrpTransMode

6.3.25 HIMC_SetGrpTransMode

用途

設置軸群組的路徑過渡模式。

語法

```
int HIMC_SetGrpTransMode(  
    int ctrl_id,  
    int group_id,  
    int trans_mode  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- trans_mode [in] 軸群組的新路徑過渡模式，請參閱 6.1.5 節。
 輸入範圍：0 ~ 4

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpTransMode
LabVIEW	HIMC Set Grp Trans Mode.vi
Python	SetGrpTransMode

6.3.26 HIMC_SetGrpTransPrm

用途

設置軸群組的路徑過渡模式參數。

語法

```
int HIMC_SetGrpTransPrm(
    int    ctrl_id,
    int    group_id,
    double trans_vel,
    double trans_dis,
    double trans_dev,
    double trans_curv
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- trans_vel [in] 軸群組新的路徑過渡模式速度參數，請參閱 6.1.5 節。
- trans_dis [in] 軸群組新的路徑過渡模式距離參數，請參閱 6.1.5 節。
- trans_dev [in] 軸群組新的路徑過渡模式最大誤差參數，請參閱 6.1.5 節。
- trans_curv [in] 軸群組新的路徑過渡模式最大曲率參數，請參閱 6.1.5 節。

表 6.3.26.1 路徑過渡參數設定範圍

路徑過渡參數	單位	Max.	Min.
trans_vel	mm/s	< 5000	> 0
trans_dis	mm	< [len _{min}] = MIN([len _{s1}],[len _{s2}])	> 0
trans_dev	mm	< [trans_dis] × $\frac{1}{2} \cos \frac{\theta}{2}$ 註：θ為角落角度	> 0
trans_curv	mm ⁻¹	< 10 ⁷	> $\frac{6 \times \sin ([\theta])}{[len_{min}] \sqrt{2 - 2 \cos ([\theta])}^3}$

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpTransPrm
LabVIEW	HIMC Set Grp Trans Prm.vi
Python	SetGrpTransPrm

6.3.27 HIMC_GetGrpCmdNum

用途

取得軸群組的在命令緩衝區的命令數量。

語法

```
int HIMC_GetGrpCmdNum(  
    int ctrl_id,  
    int group_id,  
    int *p_grp_cmd_num  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in]

軸群組編號。
- p_grp_cmd_num [out]

指標型態的記憶體，用來儲存軸群組在命令緩衝區的命令數量。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpCmdNum
LabVIEW	HIMC Get Grp Cmd Num.vi
Python	GetGrpCmdNum

6.3.28 HIMC_SetGrpVelScale

用途

設置軸群組運動的速度百分比。

語法

```
int HIMC_SetGrpVelScale(
    int    ctrl_id,
    int    group_id,
    double vel_scale
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- vel_scale [in] 軸群組運動的新速度百分比。
輸入範圍：0 ~ 100

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpVelScale
LabVIEW	HIMC Set Grp Vel Scale.vi
Python	SetGrpVelScale

6.3.29 HIMC_GetGrpVelScale

用途

取得軸群組運動的速度百分比。

語法

```
int HIMC_GetGrpVelScale(
    int    ctrl_id,
    int    group_id,
    double *p_grp_vel_scale
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- p_grp_vel_scale [out] 指標型態的記憶體，用來儲存軸群組運動的速度百分比。
數值範圍為 0 ~ 100。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpVelScale
LabVIEW	HIMC Get Grp Vel Scale.vi
Python	GetGrpVelScale

6.3.30 HIMC_GetGrpCoordTrans

用途

取得軸群組座標系統的轉換參數。

語法

```
int HIMC_GetGrpCoordTrans(
    int    ctrl_id,
    int    group_id,
    int    coord_sys,
    double *trans_param
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- coord_sys [in] 座標系統。
請參閱 21.2 節 CoordSystem。
- trans_param [out] 指向六元素陣列的指標，內含 6-DOF { X, Y, Z, A, B, C } 中的轉換參數。
參數單位：X, Y, Z 為 mm (毫米) ； A, B, C 為 deg (角度) 。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpCoordTrans
LabVIEW	HIMC Get Grp Coord Trans.vi
Python	GetGrpCoordTrans

6.3.31 HIMC_SetGrpCoordTrans

用途

設置軸群組座標系統的轉換參數。

語法

```
int HIMC_SetGrpCoordTrans(
    int    ctrl_id,
    int    group_id,
    int    coord_sys,
    double *trans_param
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- coord_sys [in] 座標系統。
請參閱 21.2 節 CoordSystem。
- trans_param [in] 指向六元素陣列的指標，內含 6-DOF { X, Y, Z, A, B, C } 中的轉換參數。
參數單位：X, Y, Z 為 mm (毫米) ； A, B, C 為 deg (角度) 。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpCoordTrans
LabVIEW	HIMC Set Grp Coord Trans.vi
Python	SetGrpCoordTrans

6.3.32 HIMC_GetGrpPoseCmd

用途

取得軸群組座標系統的姿態命令。

語法

```
int HIMC_GetGrpPoseCmd(
    int    ctrl_id,
    int    group_id,
    int    coord_sys,
    double *pose_cmd
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- coord_sys [in] 座標系統。
請參閱 21.2 節 CoordSystem。
- pose_cmd [out] 指向六元素陣列的指標，內含 6-DOF { X, Y, Z, A, B, C } 中的姿態命令。
參數單位：X, Y, Z 為 mm (毫米) ； A, B, C 為 deg (角度) 。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpPoseCmd
LabVIEW	HIMC Get Grp Pose Cmd.vi
Python	GetGrpPoseCmd

6.3.33 HIMC_GetGrpPoseFb

用途

取得軸群組座標系統的姿態回授。

語法

```
int HIMC_GetGrpPoseFb(
    int    ctrl_id,
    int    group_id,
    int    coord_sys,
    double *pose_fb
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- coord_sys [in] 座標系統。
請參閱 21.2 節 CoordSystem。
- pose_fb [out] 指向六元素陣列的指標，內含 6-DOF { X, Y, Z, A, B, C } 中的姿態回授。
參數單位：X, Y, Z 為 mm (毫米) ； A, B, C 為 deg (角度) 。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpPoseFb
LabVIEW	HIMC Get Grp Pose Fb.vi
Python	GetGrpPoseFb

6.3.34 HIMC_SetGrpLookAheadPrm

用途

設置軸群組速度前瞻功能的運動參數。

語法

```
int HIMC_SetGrpLookAheadPrm(  
    int    ctrl_id,  
    int    group_id,  
    int    prm_id,  
    double value  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
prm_id [in]	速度前瞻功能參數。 0：角落最大加速度、1：圓弧最大加速度、2：最大弦誤差
value [in]	速度前瞻功能參數的設定值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpLookAheadPrm
LabVIEW	HIMC Set Grp Look Ahead Prm.vi
Python	SetGrpLookAheadPrm

6.3.35 HIMC_SetGrpQueueSize

用途

設置軸群組命令緩衝區的大小。

語法

```
int HIMC_SetGrpQueueSize(  
    int ctrl_id,  
    int group_id,  
    int queue_size  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in]

軸群組編號。
- queue_size [in]

軸群組命令緩衝區的大小，若輸入值為 n，緩衝區的大小為 2ⁿ。
輸入範圍：0 ~ 10

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpQueueSize
LabVIEW	HIMC Set Grp Queue Size.vi
Python	SetGrpQueueSize

6.4 軸群組狀態

6.4.1 HIMC_IsGrpEnabled

用途

詢問軸群組的啟用狀態。

語法

```
int HIMC_IsGrpEnabled(  
    int ctrl_id,  
    int group_id,  
    int *p_is_grp_enabled  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
p_is_grp_enabled [out]	指標型態的記憶體，用來儲存軸群組的啟用狀態。 若已啟用軸群組，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsGrpEnabled
LabVIEW	HIMC Is Grp Enabled.vi
Python	IsGrpEnabled

6.4.2 HIMC_IsGrpMoving

用途

詢問軸群組的移動狀態。若軸群組正在移動，軌跡規劃器（PG）會持續輸出新的位置。

語法

```
int HIMC_IsGrpMoving(  
    int ctrl_id,  
    int group_id,  
    int *p_is_grp_moving  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in]

軸群組編號。
- p_is_grp_moving [out]

指標型態的記憶體，用來儲存軸群組的移動狀態。
若軸群組正在移動，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsGrpMoving
LabVIEW	HIMC Is Grp Moving.vi
Python	IsGrpMoving

6.4.3 HIMC_IsGrpInPos

用途

詢問軸群組的到位狀態。若軸群組已到位，軸群組底下的各軸皆到位。

語法

```
int HIMC_IsGrpInPos(
    int ctrl_id,
    int group_id,
    int *p_is_grp_inpos
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- p_is_grp_inpos [out] 指標型態的記憶體，用來儲存軸群組的到位狀態。
 若軸群組已到位，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsGrpInPos
LabVIEW	HIMC Is Grp In Pos.vi
Python	IsGrpInPos

6.4.4 HIMC_IsGrpErrorStop

用途

詢問軸群組是否處於 error stop 狀態。

語法

```
int HIMC_IsGrpErrorStop(
    int ctrl_id,
    int group_id,
    int *p_is_grp_errorstop
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- p_is_grp_errorstop [out] 指標型態的記憶體，用來儲存軸群組的 error stop 狀態。
 若軸群組處於 error stop 狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsGrpErrorStop
LabVIEW	HIMC Is Grp Error Stop.vi
Python	IsGrpErrorStop

6.5 進階軸群組運動控制

6.5.1 HIMC_LineAbs

用途

命令一個軸群組，內插線性移動至特定座標系統中的絕對位置。

語法

```
int HIMC_LineAbs(
    int ctrl_id,
    int group_id,
    CoordPosition *target_pos,
    MotionProfile *motion_profile,
    CoordSystem coord_sys,
    MotionBufferMode buff_mode,
    MotionTransitionMode trans_mode,
    TransPrm *trans_prm
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
target_pos [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的 6 自由度目標位置。 參數單位：X, Y, Z 為 mm（毫米）；A, B, C 為 deg（角度）。 請參閱 20.2 節 CoordPosition。
motion_profile [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的運動設定。 請參閱 20.3 節 MotionProfile。
coord_sys [in]	指定合適的座標系統。 請參閱 21.2 節 CoordSystem。
buff_mode [in]	指定速度緩衝模式。 請參閱 21.3 節 MotionBufferMode。

- trans_mode [in]** 指定路徑過渡模式。
請參閱 21.4 節 MotionTransitionMode。
- trans_prm [in]** 指定路徑過渡模式參數的指標。
請參閱 20.6 節 TransPrm。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_LineAbs
LabVIEW	HIMC Line Abs.vi
Python	LineAbs

6.5.2 HIMC_LineRel

用途

命令一個軸群組，內插線性移動至特定座標系統中的相對位置。

語法

```
int HIMC_LineRel(
    int ctrl_id,
    int group_id,
    CoordPosition *relative_dist,
    MotionProfile *motion_profile,
    CoordSystem coord_sys,
    MotionBufferMode buff_mode,
    MotionTransitionMode trans_mode,
    TransPrm *trans_prm
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
relative_dist [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的 6 自由度相對距離。 參數單位：X, Y, Z 為 mm（毫米）；A, B, C 為 deg（角度）。 請參閱 20.2 節 CoordPosition。
motion_profile [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的運動設定。 請參閱 20.3 節 MotionProfile。
coord_sys [in]	指定合適的座標系統。 請參閱 21.2 節 CoordSystem。
buff_mode [in]	指定速度緩衝模式。 請參閱 21.3 節 MotionBufferMode。
trans_mode [in]	指定路徑過渡模式。 請參閱 21.4 節 MotionTransitionMode。

trans_prm [in] 指定路徑過渡模式參數的指標。
請參閱 20.6 節 TransPrm。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_LineRel
LabVIEW	HIMC Line Rel.vi
Python	LineRel

6.5.3 HIMC_CircleAbs

用途

命令一個軸群組，內插圓周運動至特定座標系統中的絕對位置。

語法

```
int HIMC_CircleAbs(  
    int ctrl_id,  
    int group_id,  
    CenterPosition *center_pos,  
    NormalVector *normal_vector,  
    int turns,  
    CoordPosition *target_pos,  
    MotionProfile *motion_profile,  
    CoordSystem coord_sys,  
    MotionBufferMode buff_mode,  
    MotionTransitionMode trans_mode,  
    TransPrm *trans_prm  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
center_pos [in]	指標型態的記憶體，用來儲存中心點。 參數單位：mm (毫米) 請參閱 20.4 節 CenterPosition。
normal_vector [in]	指標型態的記憶體，用來儲存圓的法向量。 請參閱 20.5 節 NormalVector。
turns [in]	相對於起點的圓周運動圈數，決定了圓周運動的方向及總角度。
target_pos [in]	指標型態的記憶體，用來儲存工具中心點 (末端點) 的 6 自由度目標位置。 參數單位：X, Y, Z 為 mm (毫米) ； A, B, C 為 deg (角度)。 請參閱 20.2 節 CoordPosition。

motion_profile [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的運動設定。 請參閱 20.3 節 MotionProfile。
coord_sys [in]	指定合適的座標系統。 請參閱 21.2 節 CoordSystem。
buff_mode [in]	指定速度緩衝模式。 請參閱 21.3 節 MotionBufferMode。
trans_mode [in]	指定路徑過渡模式。 請參閱 21.4 節 MotionTransitionMode。
trans_prm [in]	指定路徑過渡模式參數的指標。 請參閱 20.6 節 TransPrm。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_CircleAbs
LabVIEW	HIMC Circle Abs.vi
Python	CircleAbs

6.5.4 HIMC_CircleRel

用途

命令一個軸群組，內插圓周運動至特定座標系統中的相對位置。

語法

```
int HIMC_CircleRel(  
    int ctrl_id,  
    int group_id,  
    CenterPosition *center_pos,  
    NormalVector *normal_vector,  
    int turns,  
    CoordPosition *relative_dist,  
    MotionProfile *motion_profile,  
    CoordSystem coord_sys,  
    MotionBufferMode buff_mode,  
    MotionTransitionMode trans_mode,  
    TransPrm *trans_prm  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
center_pos [in]	指標型態的記憶體，用來儲存中心點。 參數單位：mm (毫米) 請參閱 20.4 節 CenterPosition。
normal_vector [in]	指標型態的記憶體，用來儲存圓的法向量。 請參閱 20.5 節 NormalVector。
turns [in]	相對於起點的圓周運動圈數，決定了圓周運動的方向及總角度。
relative_dist [in]	指標型態的記憶體，用來儲存工具中心點 (末端點) 的 6 自由度相對距離。 參數單位：X, Y, Z 為 mm (毫米) ； A, B, C 為 deg (角度)。 請參閱 20.2 節 CoordPosition。

motion_profile [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的運動設定。 請參閱 20.3 節 MotionProfile。
coord_sys [in]	指定合適的座標系統。 請參閱 21.2 節 CoordSystem。
buff_mode [in]	指定速度緩衝模式。 請參閱 21.3 節 MotionBufferMode。
trans_mode [in]	指定路徑過渡模式。 請參閱 21.4 節 MotionTransitionMode。
trans_prm [in]	指定路徑過渡模式參數的指標。 請參閱 20.6 節 TransPrm。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_CircleRel
LabVIEW	HIMC Circle Rel.vi
Python	CircleRel

(此頁有意留白。)

7. GPIO 函式

7.	GPIO 函式.....	7-1
7.1	概述	7-2
7.1.1	GPIO 變數	7-2
7.2	控制器 IO 設定	7-3
7.2.1	HIMC_SetGPO	7-3
7.2.2	HIMC_ToggleGPO	7-4
7.2.3	HIMC_SetAllGPO	7-5
7.2.4	HIMC_SetGPInvert	7-6
7.2.5	HIMC_SetGPOInvert	7-7
7.2.6	HIMC_BindEMO	7-8
7.3	從站 IO 設定	7-9
7.3.1	HIMC_SetSlvGPO	7-9
7.3.2	HIMC_ToggleSlvGPO	7-11
7.3.3	HIMC_SetSlvAllGPO	7-12
7.4	控制器 IO 狀態	7-13
7.4.1	HIMC_GetGPI	7-13
7.4.2	HIMC_GetGPO	7-14
7.4.3	HIMC_GetAllGPI	7-15
7.4.4	HIMC_GetAllGPO	7-16
7.5	從站 IO 狀態	7-17
7.5.1	HIMC_GetSlvGPI	7-17
7.5.2	HIMC_GetSlvGPO	7-18
7.5.3	HIMC_GetSlvAllGPI	7-19
7.5.4	HIMC_GetSlvAllGPO	7-20
7.5.5	HIMC_GetSlvAllGPIByStartIdx	7-21
7.5.6	HIMC_GetSlvAllGPOByStartIdx	7-23

7.1 概述

HIMC 提供各 8 組通用輸入 / 輸出 (GPIO) 腳位，硬體延遲時間 1ms 以內，24V。從站裝置可透過 CoE 通訊連結控制器並更新從站上的 IO 狀態，IO 的數量依從站裝置而定。使用者可利用本章提供的函式，如 SetGPO 與 SetSlvGPO 來分別設定 HIMC 與從站輸出腳位的訊號，也可以詢問輸入 / 輸出腳位的訊號狀態。另可利用 iA Studio 功能模組 Digital IO (請參閱《iA Studio 軟體操作手冊》4.4 節) 來觀察與設定 HIMC 與從站的輸入 / 輸出狀態。

HIMC 的數位輸入 I8 為 E-Stop 的訊號源 (請參閱《HIMC 安裝指南》3.3 節)，接收到正緣訊號時會被觸發，此時所有軸都會被解激磁，且所有 HMPL task 都會被停止執行。

註：觸發正緣訊號後，使用者可重新激磁軸或重新執行 HMPL task。

7.1.1 GPIO 變數

使用者可利用 iA Studio 的 Scope Manager (請參閱《iA Studio 軟體使用手冊》4.8 節) 選擇欲觀測的控制器通用輸入 / 輸出系統變數。詳細說明如表 7.1.1.1。

表 7.1.1.1 控制器通用輸入 / 輸出變數

名稱	變數	單位	描述
HIMC GPO	himc_gpo_bits	無	控制器通用輸出腳位狀態。
HIMC GPI	himc_gpi_bits	無	控制器通用輸入腳位狀態。

7.2 控制器 IO 設定

7.2.1 HIMC_SetGPO

用途

設置控制器通用輸出的狀態。

語法

```
int HIMC_SetGPO(  
    int ctrl_id,  
    int gpo_idx,  
    char state  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- gpo_idx [in] 通用輸出編號。
- state [in] 欲設置的狀態。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGPO
LabVIEW	HIMC Set GPO.vi
Python	SetGPO

7.2.2 HIMC_ToggleGPO

用途

切換控制器通用輸出的狀態。

語法

```
int HIMC_ToggleGPO(  
    int ctrl_id,  
    int gpo_idx  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

gpo_idx [in] 通用輸出編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ToggleGPO
LabVIEW	HIMC Toggle GPO.vi
Python	ToggleGPO

7.2.3 HIMC_SetAllGPO

用途

設置控制器多個通用輸出的狀態。

語法

```
int HIMC_SetAllGPO(  
    int ctrl_id,  
    int all_gpo_state  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- all_gpo_state [in] 所有通用輸出的狀態值。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetAllGPO
LabVIEW	HIMC Set All GPO.vi
Python	SetAllGPO

7.2.4 HIMC_SetGPInvert

用途

設置控制器通用輸入的反相狀態。

語法

```
int HIMC_SetGPInvert(
    int ctrl_id,
    int gpi_idx,
    char invert
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

gpi_idx [in] 通用輸入編號。

invert [in] 欲設置的反相狀態。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGPInvert
LabVIEW	HIMC Set GPI Invert.vi
Python	SetGPInvert

7.2.5 HIMC_SetGPOInvert

用途

設置控制器通用輸出的反相狀態。

語法

```
int HIMC_SetGPOInvert(  
    int ctrl_id,  
    int gpo_idx,  
    char invert  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- gpo_idx [in] 通用輸出編號。
- invert [in] 欲設置的反相狀態。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGPOInvert
LabVIEW	HIMC Set GPO Invert.vi
Python	SetGPOInvert

7.2.6 HIMC_BindEMO

用途

設置欲綁定 E-Stop 的通用輸入腳位。

語法

```
int HIMC_BindEMO(
    int ctrl_id,
    int gpi_idx
);
```

參數

- ctrl_id [in]** HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- gpi_idx [in]** 通用輸入編號，預設值為 8。
 若設為 0，則所有通用輸入腳位皆不綁定 E-Stop。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_BindEMO
LabVIEW	HIMC Bind EMO.vi
Python	Bind EMO

7.3 從站 IO 設定

7.3.1 HIMC_SetSlvGPO

用途

設置從站通用輸出的狀態。

語法

```
int HIMC_SetSlvGPO(  
    int ctrl_id,  
    int slv_slot_id,  
    int gpo_idx,  
    int on_off  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- gpo_idx [in] 通用輸出編號。
- on_off [in] 欲設置的狀態。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

使用此函式需將 Digital output 物件配置為 PDO，例如驅動器需設定 0x60FE(Digital outputs)為 PDO。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetSlvGPO
LabVIEW	HIMC Set Slv GPO.vi
Python	SetSlvGPO

7.3.2 HIMC_ToggleSlvGPO

用途

切換從站通用輸出的狀態。

語法

```
int HIMC_ToggleSlvGPO(
    int ctrl_id,
    int slv_slot_id,
    int gpo_idx
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- gpo_idx [in] 通用輸出編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將 Digital output 物件配置為 PDO，例如驅動器需設定 0x60FE(Digital outputs)為 PDO。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ToggleSlvGPO
LabVIEW	HIMC Toggle Slv GPO.vi
Python	ToggleSlvGPO

7.3.3 HIMC_SetSlvAllGPO

用途

設置從站多個通用輸出的狀態。

語法

```
int HIMC_SetSlvAllGPO(
    int ctrl_id,
    int slv_slot_id,
    int all_gpo_state
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- all_gpo_state [in] 所有通用輸出的狀態值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將 Digital output 物件配置為 PDO，例如驅動器需設定 0x60FE(Digital outputs)為 PDO。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetSlvAllGPO
LabVIEW	HIMC Set Slv All GPO.vi
Python	SetSlvAllGPO

7.4 控制器 IO 狀態

7.4.1 HIMC_GetGPI

用途

取得控制器通用輸入的狀態。

語法

```
int HIMC_GetGPI(
    int    ctrl_id,
    int    gpi_idx,
    char *p_state
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- gpi_idx [in] 通用輸入編號。
- p_state [out] 指標型態的記憶體，用來儲存特定輸入的狀態。
 若輸入為導通狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGPI
LabVIEW	HIMC Get GPI.vi
Python	GetGPI

7.4.2 HIMC_GetGPO

用途

取得控制器通用輸出的狀態。

語法

```
int HIMC_GetGPO(
    int    ctrl_id,
    int    gpo_idx,
    char *p_state
);
```

參數

- ctrl_id [in]** HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- gpo_idx [in]** 通用輸出編號。
- p_state [out]** 指標型態的記憶體，用來儲存特定輸出的狀態。
若輸出為導通狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGPO
LabVIEW	HIMC Get GPO.vi
Python	GetGPO

7.4.3 HIMC_GetAllGPI

用途

取得控制器多個通用輸入的狀態。

語法

```
int HIMC_GetAllGPI(  
    int ctrl_id,  
    int *p_state  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_state [out]

指標型態的記憶體，用來儲存通用輸入的狀態。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetAllGPI
LabVIEW	HIMC Get All GPI.vi
Python	GetAllGPI

7.4.4 HIMC_GetAllGPO

用途

取得控制器多個通用輸出的狀態。

語法

```
int HIMC_GetAllGPO(
    int ctrl_id,
    int *p_state
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

p_state [out] 指標型態的記憶體，用來儲存通用輸出的狀態。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetAllGPO
LabVIEW	HIMC Get All GPO.vi
Python	GetAllGPO

7.5 從站 IO 狀態

7.5.1 HIMC_GetSlvGPI

用途

取得從站通用輸入的狀態。

語法

```
int HIMC_GetSlvGPI(  
    int ctrl_id,  
    int slv_slot_id,  
    int gpi_idx,  
    int *p_state  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- gpi_idx [in] 通用輸入編號。
- p_state [out] 指標型態的記憶體，用來儲存特定輸入的狀態。
 若輸入為導通狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

使用此函式需將 Digital input 物件配置為 PDO，例如驅動器需設定 0x60FD (Digital inputs) 為 PDO。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvGPI
LabVIEW	HIMC Get Slv GPI.vi
Python	GetSlvGPI

7.5.2 HIMC_GetSlvGPO

用途

取得從站通用輸出的狀態。

語法

```
int HIMC_GetSlvGPO(
    int ctrl_id,
    int slv_slot_id,
    int gpo_idx,
    int *p_state
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- gpo_idx [in] 通用輸出編號。
- p_state [out] 指標型態的記憶體，用來儲存特定輸出的狀態。
若輸出為導通狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

使用此函式需將 Digital output 物件配置為 PDO，例如驅動器需設定 0x60FE(Digital outputs)為 PDO。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvGPO
LabVIEW	HIMC Get Slv GPO.vi
Python	GetSlvGPO

7.5.3 HIMC_GetSlvAllGPI

用途

詢問從站所有通用輸入的狀態。

語法

```
int HIMC_GetSlvAllGPI(
    int ctrl_id,
    int slv_slot_id,
    int *p_state
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- p_state [out] 指標型態的記憶體，用來儲存從站所有通用輸入的狀態值。
若第 1 與第 4 個 GPI 腳位為 TRUE，則回傳值為 9。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將 Digital input 物件配置為 PDO，例如驅動器需設定 0x60FD (Digital inputs) 為 PDO。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAllGPI
LabVIEW	HIMC Get Slv All GPI.vi
Python	GetSlvAllGPI

7.5.4 HIMC_GetSlvAllGPO

用途

詢問從站所有通用輸入的狀態。

語法

```
int HIMC_GetSlvAllGPO(
    int ctrl_id,
    int slv_slot_id,
    int *p_state
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- p_state [out] 指標型態的記憶體，用來儲存從站所有通用輸出的狀態值。
 若第 2 與第 3 個 GPO 腳位為 TRUE，則回傳值為 6。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將 Digital output 物件配置為 PDO，例如驅動器需設定 0x60FE(Digital outputs)為 PDO。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAllGPO
LabVIEW	HIMC Get Slv All GPO.vi
Python	GetSlvAllGPO

7.5.5 HIMC_GetSlvAllGPIByStartIdx

用途

依照起始編號取得從站所有通用輸入的狀態。

語法

```
int HIMC_GetSlvAllGPIByStartIdx(  
    int ctrl_id,  
    int slv_slot_id,  
    int start_idx  
    int *p_state  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- start_idx [in] 通用輸入編號起始值。
- p_state [out] 指標型態的記憶體，用來儲存從站由起始編號後的所有通用輸入的狀態值。
 若起始編號為 8，第 8 與第 11 個 GPI 腳位為 TRUE，則回傳值為 9。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

使用此函式需將 Digital input 物件配置為 PDO，例如驅動器需設定 0x60FD (Digital inputs) 為 PDO。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAllGPIByStartIdx
LabVIEW	-
Python	GetSlvAllGPIByStartIdx

7.5.6 HIMC_GetSlvAllGPOByStartIdx

用途

依照起始編號取得從站所有通用輸出的狀態。

語法

```
int HIMC_GetSlvAllGPOByStartIdx(  
    int ctrl_id,  
    int slv_slot_id,  
    int start_idx  
    int *p_state  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- start_idx [in] 通用輸出編號起始值。
- p_state [out] 指標型態的記憶體，用來儲存從站由起始編號後的所有通用輸出的狀態值。
 若起始編號為 8，第 9 與第 10 個 GPO 腳位為 TRUE，則回傳值為 6。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

使用此函式需將 Digital output 物件配置為 PDO，例如驅動器需設定 0x60FE(Digital outputs)為 PDO。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAllGPOByStartIdx
LabVIEW	-
Python	GetSlvAllGPOByStartIdx

8. AIO 函式

8.	AIO 函式	8-1
8.1	概述	8-2
8.2	從站 AIO 設定	8-3
8.2.1	HIMC_SetSlvAlType	8-3
8.2.2	HIMC_SetSlvAOType	8-4
8.2.3	HIMC_SetSlvAORaw	8-5
8.2.4	HIMC_SetSlvAO	8-6
8.3	從站 AIO 狀態	8-7
8.3.1	HIMC_GetSlvAlType	8-7
8.3.2	HIMC_GetSlvAOType	8-8
8.3.3	HIMC_GetSlvAlRaw	8-9
8.3.4	HIMC_GetSlvAl	8-10
8.3.5	HIMC_GetSlvAORaw	8-11
8.3.6	HIMC_GetSlvAO	8-12
8.4	從站 AO 綁定 HIMC 內部記憶體變數	8-13
8.4.1	HIMC_SetSlvAOMonitor	8-13
8.4.2	HIMC_SetSlvAOParam	8-15
8.4.3	HIMC_GetSlvAOScale	8-17
8.4.4	HIMC_GetSlvAOOffset	8-18
8.4.5	HIMC_IsSlvAOBound	8-19

8.1 概述

利用 AIO 函式，具類比輸入 (AI) 或類比輸出 (AO) 功能的從站可讀取和設定相關參數。其中 HMPL 提供使用者指定數位類比轉換形式設定，說明如表 8.1。

表 8.1.1 類比資料轉換形式定義

轉換形式	轉換說明	描述	附註																								
kDAC_NONE	無轉換	此轉換形式只能使用 Raw 函式操作。	-																								
kDAC_N10_P10	-10~10	將原始類比值轉換為-10~10。	常見於類比 電壓模組																								
kDAC_P0_P10	0~10	將原始類比值轉換為 0~10。																									
kDAC_N5_P5	-5~5	將原始類比值轉換為-5~5。																									
kDAC_P0_P5	0~5	將原始類比值轉換為 0~5。																									
kDAC_P0_P20	0~20	將原始類比值轉換為 0~20。																									
kDAC_P4_P20	4~20	將原始類比值轉換為 4~20。	常見於類比 電流模組																								
kDAC_N20_P20	-20~20	將原始類比值轉換為-20~20。																									
kDAC_SIGNED	高位元為正負號定義	使用時為有符號轉換。 以 16 bit ±10V 類比裝置為例，只設定 kDAC_N10_P10 時轉換表如下： <table><tr><th>原始數值</th><th>轉換數值</th></tr><tr><td>65535</td><td>10</td></tr><tr><td>49152</td><td>5</td></tr><tr><td>32768</td><td>0</td></tr><tr><td>16384</td><td>-5</td></tr><tr><td>0</td><td>-10</td></tr></table> 同時設定 kDAC_SIGNED kDAC_N10_P10 時轉換表如下： <table><tr><th>原始數值</th><th>轉換數值</th></tr><tr><td>32767</td><td>10</td></tr><tr><td>16384</td><td>5</td></tr><tr><td>0</td><td>0</td></tr><tr><td>-16384</td><td>-5</td></tr><tr><td>-32768</td><td>-10</td></tr></table>	原始數值	轉換數值	65535	10	49152	5	32768	0	16384	-5	0	-10	原始數值	轉換數值	32767	10	16384	5	0	0	-16384	-5	-32768	-10	轉換方式 請參考 從站裝置的 使用手冊
原始數值	轉換數值																										
65535	10																										
49152	5																										
32768	0																										
16384	-5																										
0	-10																										
原始數值	轉換數值																										
32767	10																										
16384	5																										
0	0																										
-16384	-5																										
-32768	-10																										

8.2 從站 AIO 設定

8.2.1 HIMC_SetSlvAIOType

用途

設置從站的類比輸入值轉換形式。

語法

```
int HIMC_SetSlvAIOType(  
    int ctrl_id,  
    int slv_slot_id,  
    int ai_idx,  
    int range_type  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ai_idx [in] 類比輸入通道位置。
- range_type [in] 類比資料轉換形式，請參考表 8.1。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetSlvAIOType
LabVIEW	HIMC Set Slv AO Type.vi
Python	SetSlvAIOType

8.2.2 HIMC_SetSlvAOType

用途

設置從站的類比輸出值轉換形式。

語法

```
int HIMC_SetSlvAOType(
    int ctrl_id,
    int slv_slot_id,
    int ao_idx,
    int range_type
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ao_idx [in] 類比輸出通道位置。
- range_type [in] 類比資料轉換形式，請參考表 8.1。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetSlvAOType
LabVIEW	HIMC Set Slv AO Type.vi
Python	SetSlvAOType

8.2.3 HIMC_SetSlvAORaw

用途

設置從站的類比輸出原始值。

語法

```
int HIMC_SetSlvAORaw(  
    int ctrl_id,  
    int slv_slot_id,  
    int ao_idx,  
    int ao_raw_val  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ao_idx [in] 類比輸出通道位置。
- ao_raw_val [in] 類比輸出原始值。
 若需調整數值顯示範圍，請參考 1.5 節自行轉換型態。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將 Analog output 物件配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetSlvAORaw
LabVIEW	-
Python	SetSlvAORaw

8.2.4 HIMC_SetSlvAO

用途

設置從站的類比輸出值。

語法

```
int HIMC_SetSlvAO(
    int ctrl_id,
    int slv_slot_id,
    int ao_idx,
    double ao_val
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ao_idx [in] 類比輸出通道位置。
- ao_val [in] 類比輸出值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將 Analog output 物件配置為 PDO 且需設定轉換形式。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetSlvAO
LabVIEW	HIMC Set Slv AO.vi
Python	SetSlvAO

8.3 從站 AIO 狀態

8.3.1 HIMC_GetSlvAIType

用途

取得從站的類比輸入類型。

語法

```
int HIMC_GetSlvAIType(
    int ctrl_id,
    int slv_slot_id,
    int ai_idx,
    int *p_ai_type
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ai_idx [in] 類比輸入通道位置。
- p_ai_type [out] 指標型態的記憶體，用來儲存從站的類比輸入類型，請參考表 8.1。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAIType
LabVIEW	HIMC Get Slv AI Type.vi
Python	GetSlvAIType

8.3.2 HIMC_GetSlvAOType

用途

取得從站的類比輸入類型。

語法

```
int HIMC_GetSlvAOType(
    int ctrl_id,
    int slv_slot_id,
    int ao_idx,
    int *p_ao_type
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ao_idx [in] 類比輸出通道位置。
- p_ao_type [out] 指標型態的記憶體，用來儲存從站的類比輸出類型，請參考表 8.1。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAOType
LabVIEW	HIMC Get Slv AO Type.vi
Python	GetSlvAOType

8.3.3 HIMC_GetSlvAIRaw

用途

取得從站類比輸入原始值。

語法

```
int HIMC_GetSlvAIRaw(
    int ctrl_id,
    int slv_slot_id,
    int ai_idx,
    int *p_ai_raw_val
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ai_idx [in] 類比輸入通道位置。
- p_ai_raw_val [out] 指標型態的記憶體，用來儲存類比輸入原始值。
 若需調整數值顯示範圍，請參考 1.5 節自行轉換型態。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將 Analog input 物件配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAIRaw
LabVIEW	-
Python	GetSlvAIRaw

8.3.4 HIMC_GetSlvAI

用途

取得從站類比輸入值。

語法

```
int HIMC_GetSlvAI(
    int ctrl_id,
    int slv_slot_id,
    int ai_idx,
    double *p_ai_val
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ai_idx [in] 類比輸入通道位置。
- p_ai_val [out] 指標型態的記憶體，用來儲存類比輸入值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將 Analog input 物件配置為 PDO 且需設定轉換形式。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAI
LabVIEW	HIMC Get Slv AI.vi
Python	GetSlvAI

8.3.5 HIMC_GetSlvAORaw

用途

取得從站類比輸出原始值。

語法

```
int HIMC_GetSlvAORaw(
    int ctrl_id,
    int slv_slot_id,
    int ao_idx,
    int *p_ao_raw_val
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ao_idx [in] 類比輸出通道位置。
- p_ao_raw_val [out] 指標型態的記憶體，用來儲存類比輸出原始值。
 若需調整數值顯示範圍，請參考 1.5 節自行轉換型態。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將 Analog output 物件配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAORaw
LabVIEW	-
Python	GetSlvAORaw

8.3.6 HIMC_GetSlvAO

用途

取得從站類比輸出值。

語法

```
int HIMC_GetSlvAO(  
    int ctrl_id,  
    int slv_slot_id,  
    int ao_idx,  
    double *p_ao_val  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ao_idx [in] 類比輸出通道位置。
- p_ao_val [out] 指標型態的記憶體，用來儲存類比輸出值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將 Analog output 物件配置為 PDO 且需設定轉換形式。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAO
LabVIEW	HIMC Get Slv AO.vi
Python	GetSlvAO

8.4 從站 AO 綁定 HIMC 內部記憶體變數

8.4.1 HIMC_SetSlvAOMonitor

用途

設置欲與類比輸出綁定的控制器變數。

語法

```
int HIMC_SetSlvAOMonitor(  
    int ctrl_id,  
    int slv_slot_id,  
    int ao_idx,  
    int var_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ao_idx [in] 類比輸出通道位置。
- var_id [in] 控制器變數與軸 ID。
 控制器變數與軸 ID 的定義如下表，更多的控制器變數請參閱 16.1.1 節。
 範例：HMPL_AXIS_0 | HMPL_REF_VEL

控制器變數	定義說明	控制器變數	定義說明
HMPL_REF_POS	軸參考位置	HMPL_POS_FB	軸位置回授
HMPL_REF_VEL	軸參考速度	HMPL_VEL_FB	軸速度回授
HMPL_REF_ACC	軸參考加速度	HMPL_ACC_FB	軸加速度回授
		HMPL_CUR_FB	軸電流回授

軸 ID	定義說明
HMPL_AXIS_0	軸 0
HMPL_AXIS_1	軸 1
...	...
HMPL_AXIS_15	軸 15

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將 Analog output 物件配置為 PDO 且需設定轉換形式。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetSlvAOMonitor
LabVIEW	HIMC Set Slv AO Monitor.vi
Python	SetSlvAOMonitor

8.4.2 HIMC_SetSlvAOParam

用途

設置從站的類比轉換控制器參數。

語法

```
int HIMC_SetSlvAOParam(  
    int ctrl_id,  
    int slv_slot_id,  
    int ao_idx,  
    int ao_en_bind,  
    double ao_scale,  
    double ao_offset  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ao_idx [in] 類比輸出通道位置。
- ao_en_bind [in] 0：關閉類比輸出綁定控制器變數的功能（預設值）
 1：開啟類比輸出綁定控制器變數的功能
- ao_scale [in] 類比輸出控制器變數的比例增益。
- ao_offset [in] 類比輸出控制器變數的偏移量。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetSlvAOParam
LabVIEW	HIMC Set Slv AO Param.vi
Python	SetSlvAOParam

8.4.3 HIMC_GetSlvAOScale

用途

取得從站類比輸出與控制器變數的比例增益。

語法

```
int HIMC_SetSlvAOScale(
    int ctrl_id,
    int slv_slot_id,
    int ao_idx,
    double *p_ao_scale
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ao_idx [in] 類比輸出通道位置。
- p_ao_scale [in] 指標型態的記憶體，用來儲存類比輸出控制器變數的比例增益。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAOScale
LabVIEW	HIMC Get Slv AO Scale.vi
Python	GetSlvAOScale

8.4.4 HIMC_GetSlvAOffset

用途

取得從站類比輸出與控制器變數的偏移量。

語法

```
int HIMC_SetSlvAOffset(  
    int ctrl_id,  
    int slv_slot_id,  
    int ao_idx,  
    double *p_ao_offset  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ao_idx [in] 類比輸出通道位置。
- p_ao_offset [in] 指標型態的記憶體，用來儲存類比輸出與控制器變數的偏移量。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvAOffset
LabVIEW	HIMC Get Slv AO Offset.vi
Python	GetSlvAOffset

8.4.5 HIMC_IsSlvAOBound

用途

詢問從站的類比輸出是否綁定控制器變數。

語法

```
int HIMC_IsSlvAOBound(  
    int ctrl_id,  
    int slv_slot_id,  
    int ao_idx,  
    int *p_is_bound  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ao_idx [in] 類比輸出通道位置。
- p_ao_bound [in] 指標型態的記憶體，用來儲存類比輸出是否綁定控制器變數的狀態。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsSlvAOBound
LabVIEW	HIMC Is Slv AO Bound.vi
Python	IsSlvAOBound

(此頁有意留白。)

9. User Table 函式

9.	User Table 函式.....	9-1
9.1	概述	9-2
9.2	HIMC_SetUserTable	9-3
9.3	HIMC_GetUserTable.....	9-4
9.4	HIMC_SetTableValue	9-5
9.5	HIMC_GetTableValue	9-6
9.6	HIMC_SaveUserTable	9-7
9.7	HIMC_LoadUserTable.....	9-8

9.1 概述

HIMC 提供使用者可自由使用的記憶體空間，最多可儲存 512,000 個 double 型態的變數資料(500K Bytes)。使用者可利用本章提供的函式存取記憶體空間，寫入的值會存放在控制器的隨機存取記憶體 (RAM) 中。透過 HIMC_SaveUserTable 函式，將 User Table 在記憶體空間的資料儲存到 HIMC 的硬碟空間中；在 HIMC 斷電重開之後，透過 HIMC_LoadUserTable 函式，將保存的資料重新複製到 User Table 的記憶體空間。

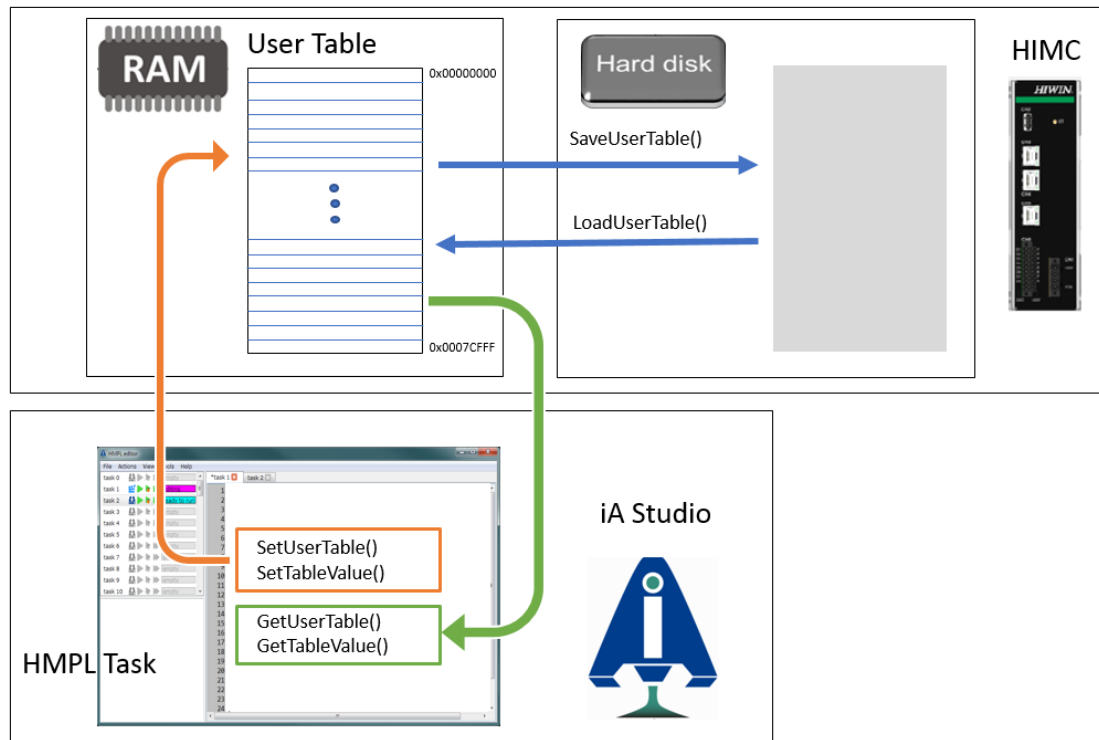


圖 9.1.1

註：使用者可利用 iA Studio 的 Table Viewer (請參閱《iA Studio 軟體使用手冊》4.11 節) 來存取 User Table 的變數值，包含載入與存入 HIMC 的記憶體與硬碟之中。

注意：

動態誤差補償函式所使用的誤差補償表 (Error map) 為儲存在 User Table 的記憶體空間。啟用動態誤差補償時，使用者需自行確保其他 User Table 數值的存取不會影響到所建立的誤差補償值。

9.2 HIMC_SetUserTable

用途

設置 User Table 的數據至控制器中。

語法

```
int HIMC_SetUserTable(
    int    ctrl_id,
    double *p_user_table_data,
    int    start_idx,
    int    number_of_doubles_to_write
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_user_table_data [in] 指標型態的記憶體，將數據寫入 User Table。
- start_idx [in] User Table 的起始編號。
- number_of_doubles_to_write [in] 被寫入之 User Table 的數量。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetUserTable
LabVIEW	HIMC Set User Table.vi
Python	SetUserTable

9.3 HIMC_GetUserTable

用途

從控制器中取得 User Table 的數據。

語法

```
int HIMC_GetUserTable(
    int    ctrl_id,
    double *p_user_table_data,
    int    start_idx,
    int    number_of_doubles_to_read
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
p_user_table_data [out]	指標型態的記憶體，用來儲存來自 User Table 的數據。
start_idx [in]	User Table 的起始編號。
number_of_doubles_to_read [in]	被讀取之 User Table 的數量。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetUserTable
LabVIEW	HIMC Get User Table.vi
Python	GetUserTable

9.4 HIMC_SetTableValue

用途

將數據寫入 User Table 的特定編號中。

語法

```
int HIMC_SetTableValue(  
    int    ctrl_id,  
    int    index,  
    double value  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- index [in] User Table 的編號。
- value [in] 輸入數據。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetTableValue
LabVIEW	HIMC Set Table Value.vi
Python	SetTableValue

9.5 HIMC_GetTableValue

用途

從 User Table 的特定編號中取得數據。

語法

```
int HIMC_GetTableValue(
    int    ctrl_id,
    int    index,
    double *value
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

index [in] User Table 的編號。

value [out] 指標型態的記憶體，用來儲存數據。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetTableValue
LabVIEW	HIMC Get Table Value.vi
Python	GetTableValue

9.6 HIMC_SaveUserTable

用途

將在 RAM 中的 User Table 數據存入永久記憶體中。

語法

```
int HIMC_SaveUserTable(  
    int ctrl_id,  
    int start_idx,  
    int num_data  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- start_idx [in] User Table 的起始編號。
- num_data [in] 被儲存元素的數量。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SaveUserTable
LabVIEW	HIMC Save User Table.vi
Python	SaveUserTable

9.7 HIMC_LoadUserTable

用途

將永久記憶體中的 User Table 數據載入至 RAM。

語法

```
int HIMC_LoadUserTable(
    int ctrl_id,
    int start_idx,
    int num_data
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- start_idx [in] User Table 的起始編號。
- num_data [in] 被載入元素的數量。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_LoadUserTable
LabVIEW	HIMC Load User Table.vi
Python	LoadUserTable

10. 位置觸發函式

10.	位置觸發函式	10-1
10.1	概述	10-2
10.1.1	PT 變數	10-2
10.1.2	PT 功能使用流程	10-4
10.2	HIMC_EnablePT.....	10-6
10.3	HIMC_DisablePT	10-7
10.4	HIMC_IsPTEnabled.....	10-8
10.5	HIMC_SetPosTriggerConfig.....	10-9
10.6	HIMC_SetPT_PosArray.....	10-10
10.7	HIMC_SetPT_StateArray	10-11
10.8	HIMC_SetPT_StartIndex.....	10-12
10.9	HIMC_SetPT_EndIndex.....	10-13

10.1 概述

HIMC 位置觸發函式僅適用於搭配大銀驅動器，使用者可利用 HMPL 命令操作 PT (位置觸發) 相關功能。操作 PT 相關功能前，請向本公司或當地經銷商諮詢相容的驅動器。

註 1：大銀驅動器使用 PT 相關功能的條件為 (1) 須為數位式編碼器 (2) 須先執行完驅動器歸原點流程。

註 2：虛擬機、虛擬軸不支援 PT 相關函式。

10.1.1 PT 變數

表 10.1.1.1 為操作 PT 相關功能的變數介紹。

表 10.1.1.1

名稱	型態	單位	描述
狀態	int	true / false	PT 功能的狀態，顯示 PT 是否仍在運行。
起點	double	毫米 或 角度	PT 功能的起點。PT 輸出訊號序列從此點開始。
終點	double	毫米 或 角度	PT 功能的終點。此點後不再發送 PT 輸出訊號。
間距	double	毫米 或 角度	連續 PT 輸出的位置間距。
脈波寬度	int	奈秒	每個 PT 輸出訊號的寬度。 範圍： 1. E1 系列驅動器為 20 奈秒至 80,000 奈秒，20 奈秒為最小增加單位。例如，20、40、... 80,000。
位置陣列	double	毫米 或 角度	非等間距 PT 功能的觸發位置。
狀態陣列	int	high / low	非等間距 PT 功能的狀態輸出。
起始索引值	int	-	非等間距 PT 功能的起始索引值。
結束索引值	int	-	非等間距 PT 功能的結束索引值。

圖 10.1.1.1 中，極性被設定為 active high。

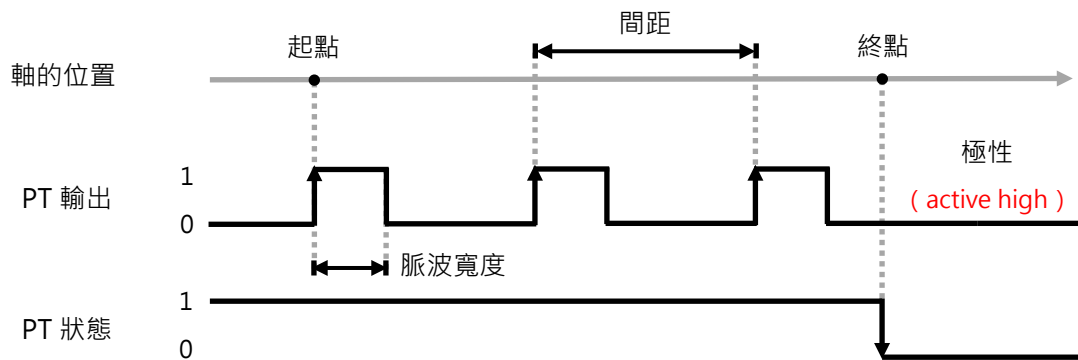


圖 10.1.1.1

圖 10.1.1.2 中，極性被設定為 active low。

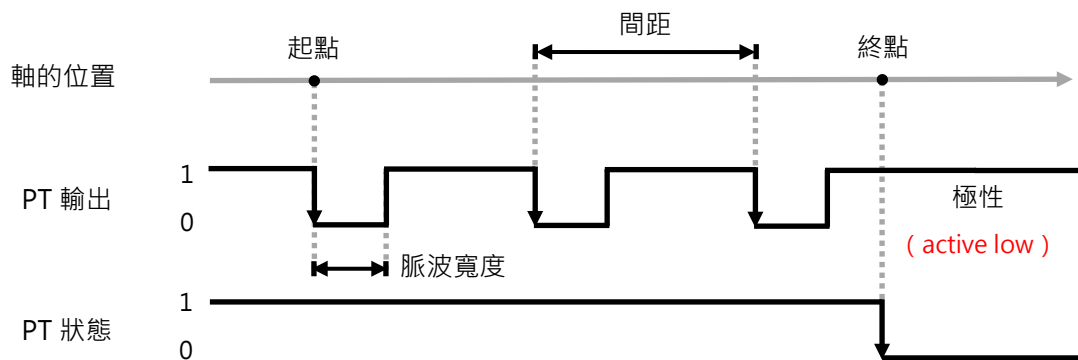


圖 10.1.1.2

限制條件：

PT 功能的間距與軸的移動速度須滿足『速度 < 間距 x 位置取樣頻率』。

若間距設定為 100 μm，位置取樣頻率為 16 K，則移動速度須小於 1600 mm/s。

註：欲調整 PT 功能的輸出極性 (active high / low)，請至驅動器介面設定，儲存設定後斷電重開驅動器，使輸出極性生效。

10.1.2 PT 功能使用流程

◆ 等間距 PT 功能

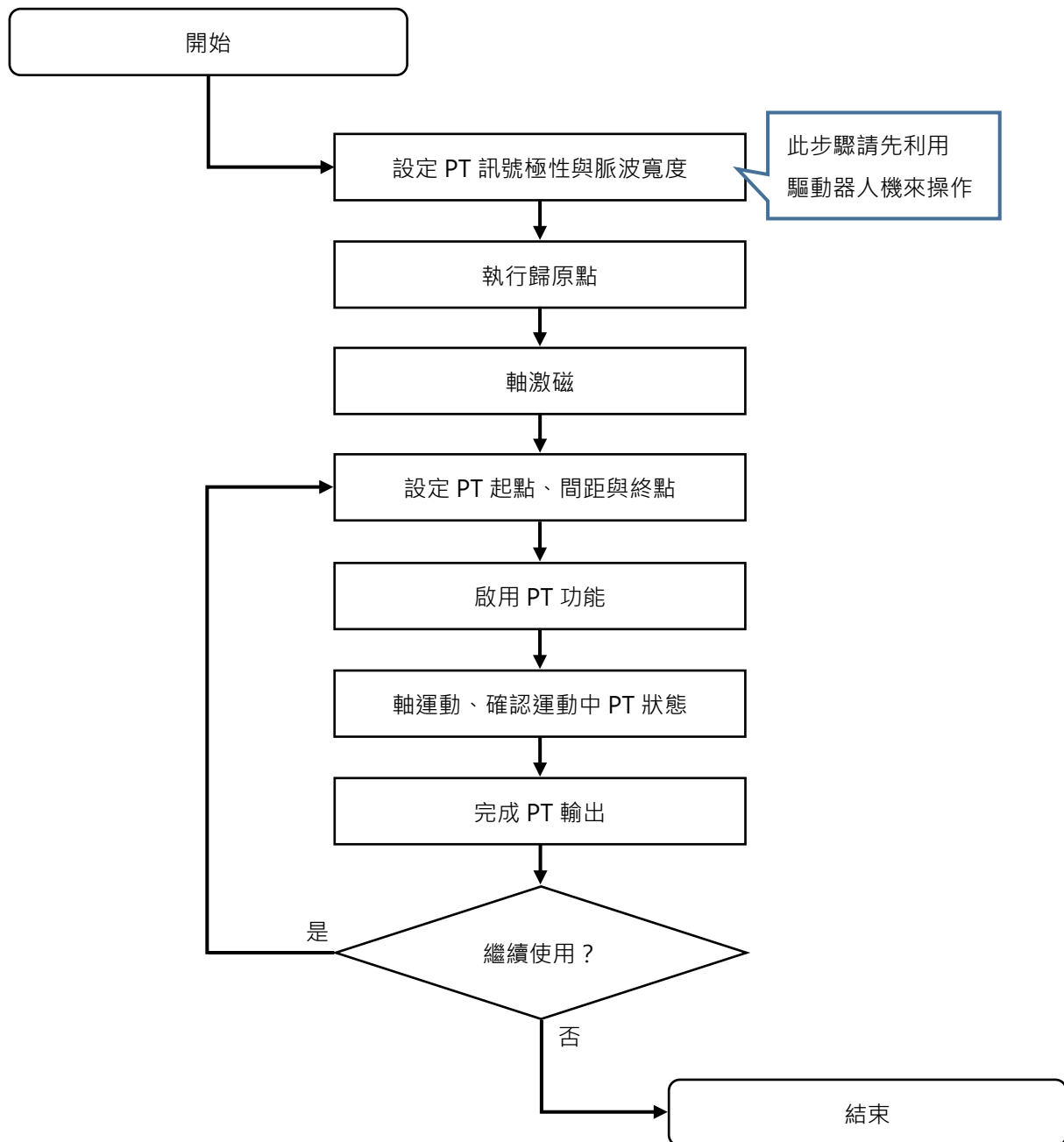


圖 10.1.2.1

◆ 非等間距 PT 功能

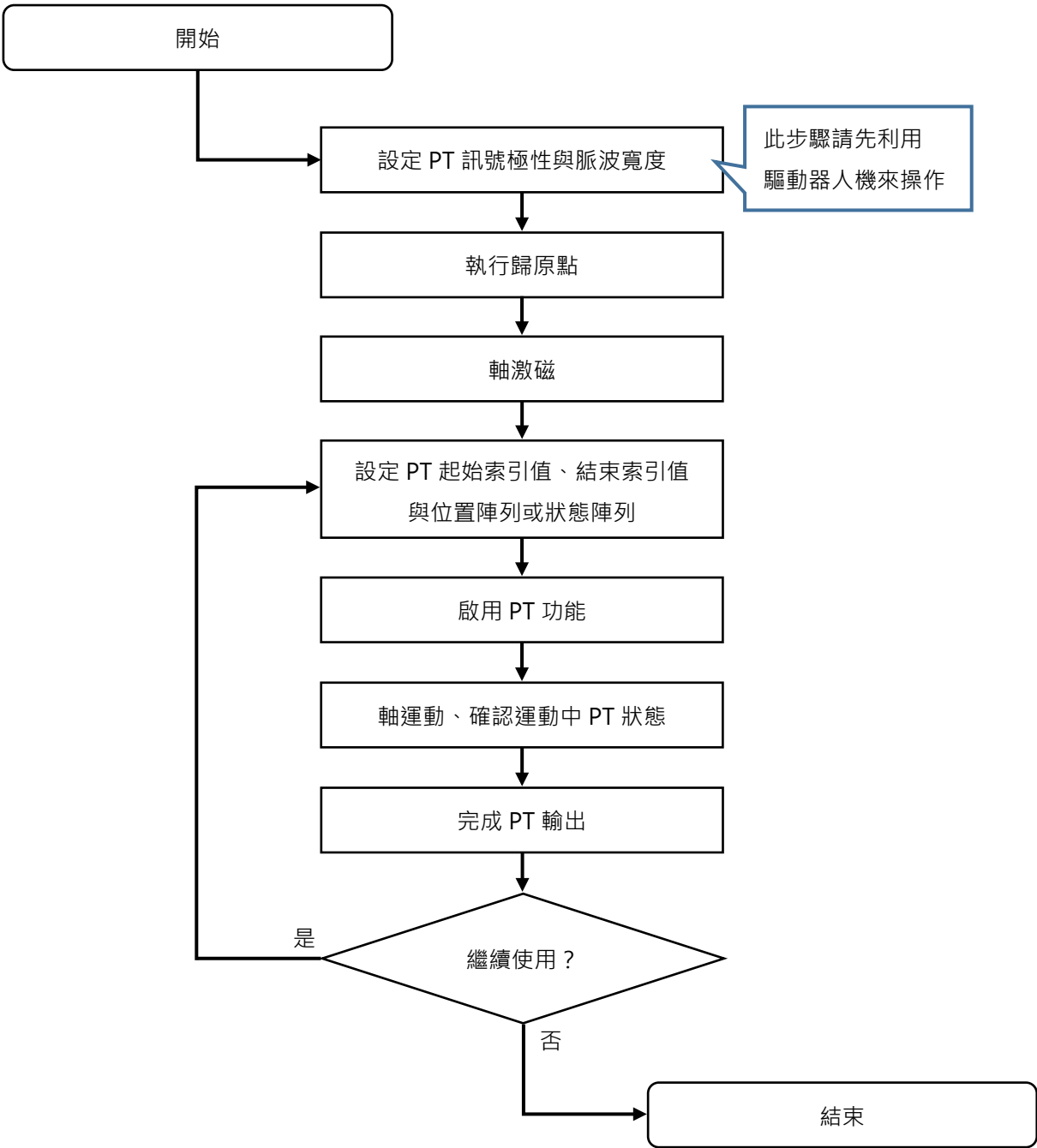


圖 10.1.2.2

10.2 HIMC_EnablePT

用途

致能軸的位置觸發功能。

語法

```
int HIMC_EnablePT(  
    int ctrl_id,  
    int axis_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_EnablePT
LabVIEW	HIMC Enable PT.vi
Python	EnablePT

10.3 HIMC_DisablePT

用途

停用軸的位置觸發功能。

語法

```
int HIMC_DisablePT(  
    int ctrl_id,  
    int axis_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisablePT
LabVIEW	HIMC Disable PT.vi
Python	DisablePT

10.4 HIMC_IsPTEnabled

用途

詢問是否已啟用位置觸發功能。

語法

```
int HIMC_IsPTEnabled(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_pt_enabled  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_pt_enabled [out] 指標型態的記憶體，用來儲存位置觸發的啟用狀態。
 若軸處於 PT 啟用狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsPTEnabled
LabVIEW	HIMC Is PT Enabled.vi
Python	IsPTEnabled

10.5 HIMC_SetPosTriggerConfig

用途

設定軸的位置觸發配置。

語法

```
int HIMC_SetPosTriggerConfig(  
    int ctrl_id,  
    int axis_id,  
    PosTriggerPar *pos_trigger_par  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- pos_trigger_par [in] 指標型態的記憶體，用來儲存軸的位置觸發配置。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetPosTriggerConfig
LabVIEW	HIMC Set Pos Trigger Config.vi
Python	SetPosTriggerConfig

10.6 HIMC_SetPT_PosArray

用途

設置非等間距位置觸發功能的觸發位置。

語法

```
int HIMC_SetPT_PosArray(  
    int    ctrl_id,  
    int    axis_id,  
    int    index,  
    double trigger_pos  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

index [in] 觸發位置的索引值。

trigger_pos [in] 一陣列，內含多個觸發位置。
 參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetPT_PosArray
LabVIEW	HIMC Set PT Pos Array.vi
Python	SetPT_PosArray

10.7 HIMC_SetPT_StateArray

用途

設置非等間距位置觸發功能的狀態輸出。

語法

```
int HIMC_SetPT_StateArray(  
    int ctrl_id,  
    int axis_id,  
    int index,  
    int state  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- index [in] 狀態輸出的索引值。
- state [in] 一陣列，內含多個狀態輸出。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetPT_StateArray
LabVIEW	HIMC Set PT State Array.vi
Python	SetPT_StateArray

10.8 HIMC_SetPT_StartIndex

用途

設置非等間距位置觸發功能的起始索引值。

語法

```
int HIMC_SetPT_StartIndex(  
    int ctrl_id,  
    int axis_id,  
    int index  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

index [in] 起始索引值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetPT_StartIndex
LabVIEW	HIMC Set PT Start Index.vi
Python	SetPT_StartIndex

10.9 HIMC_SetPT_EndIndex

用途

設置非等間距位置觸發功能的結束索引值。

語法

```
int HIMC_SetPT_EndIndex(  
    int ctrl_id,  
    int axis_id,  
    int index  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- index [in] 結束索引值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetPT_EndIndex
LabVIEW	HIMC Set PT End Index.vi
Python	SetPT_EndIndex

(此頁有意留白。)

11. Touch Probe 函式

11.	Touch Probe 函式.....	11-1
11.1	概述	11-2
11.2	HIMC_EnableTouchProbe1	11-3
11.3	HIMC_EnableTouchProbe2	11-4
11.4	HIMC_DisableTouchProbe1.....	11-5
11.5	HIMC_DisableTouchProbe2.....	11-6
11.6	HIMC_IsTouchProbe1Enabled	11-7
11.7	HIMC_IsTouchProbe2Enabled	11-8
11.8	HIMC_IsTouchProbe1Triggered.....	11-9
11.9	HIMC_IsTouchProbe2Triggered.....	11-10
11.10	HIMC_GetTouchProbe1Pos	11-11
11.11	HIMC_GetTouchProbe2Pos	11-12
11.12	HIMC_SetTouchProbe1Func.....	11-13
11.13	HIMC_SetTouchProbe2Func.....	11-15

11.1 概述

Touch Probe 函式為位置鎖存 (Latch) 功能，透過編碼器輸入訊號的邊緣觸發來補捉編碼器的位置回授數值，用於歸原點程序中（適用於 AC、DM、LM 馬達）。如圖 11.1.1 所示，驅動器經過其編碼器 Index 訊號時，會觸發 Touch Probe 功能並記錄此 Index 訊號的位置。使用者可透過 Touch Probe 函式詢問控制器的 Touch Probe 功能是否已被觸發，控制器亦可獲得被 Touch Probe 鎖存記錄的 Index 訊號位置。

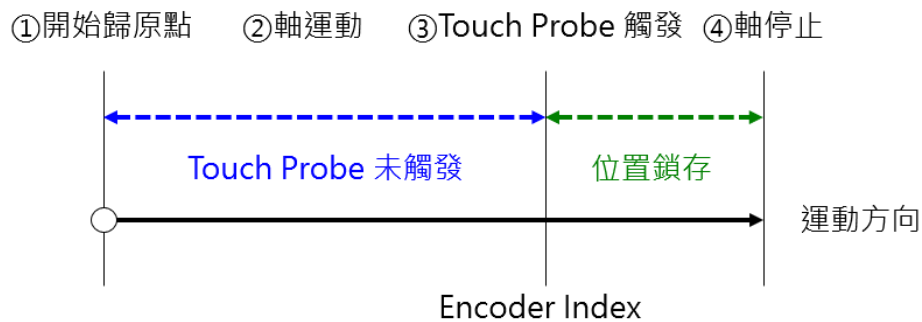


圖 11.1.1

註 1：Touch Probe 功能受驅動器支援的通道與鎖存方向限制。

註 2：虛擬機、虛擬軸不支援 Touch Probe 相關函式。

11.2 HIMC_EnableTouchProbe1

用途

啟用軸的 Touch Probe 1 功能。

語法

```
int HIMC_EnableTouchProbe1(  
    int ctrl_id,  
    int axis_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_EnableTouchProbe1
LabVIEW	-
Python	EnableTouchProbe1

11.3 HIMC_EnableTouchProbe2

用途

啟用軸的 Touch Probe 2 功能。

語法

```
int HIMC_EnableTouchProbe2(
    int ctrl_id,
    int axis_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_EnableTouchProbe2
LabVIEW	-
Python	EnableTouchProbe2

11.4 HIMC_DisableTouchProbe1

用途

停用軸的 Touch Probe 1 功能。

語法

```
int HIMC_DisableTouchProbe1(  
    int ctrl_id,  
    int axis_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisableTouchProbe1
LabVIEW	-
Python	DisableTouchProbe1

11.5 HIMC_DisableTouchProbe2

用途

停用軸的 Touch Probe 2 功能。

語法

```
int HIMC_DisableTouchProbe2(  
    int ctrl_id,  
    int axis_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisableTouchProbe2
LabVIEW	-
Python	DisableTouchProbe2

11.6 HIMC_IsTouchProbe1Enabled

用途

取得軸的 Touch Probe 1 啟用狀態。

語法

```
int HIMC_IsTouchProbe1Enabled(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_probe_enabled  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in]

軸編號。
- p_is_probe_enabled [out]

指標型態的記憶體，用來儲存 Touch Probe 1 的啟用狀態。
若軸處於 Touch Probe 1 啟用狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsTouchProbe1Enabled
LabVIEW	-
Python	IsTouchProbe1Enabled

11.7 HIMC_IsTouchProbe2Enabled

用途

取得軸的 Touch Probe 2 啟用狀態。

語法

```
int HIMC_IsTouchProbe2Enabled(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_probe_enabled  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_probe_enabled [out] 指標型態的記憶體，用來儲存 Touch Probe 2 的啟用狀態。
 若軸處於 Touch Probe 2 啟用狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsTouchProbe2Enabled
LabVIEW	-
Python	IsTouchProbe2Enabled

11.8 HIMC_IsTouchProbe1Triggered

用途

取得軸的 Touch Probe 1 觸發狀態。

語法

```
int HIMC_IsTouchProbe1Triggered(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_probe_triggered  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_probe_triggered [out] 指標型態的記憶體，用來儲存 Touch Probe 1 的觸發狀態。
 若軸處於 Touch Probe 1 觸發狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsTouchProbe1Triggered
LabVIEW	-
Python	IsTouchProbe1Triggered

11.9 HIMC_IsTouchProbe2Triggered

用途

取得軸的 Touch Probe 2 觸發狀態。

語法

```
int HIMC_IsTouchProbe2Triggered(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_probe_triggered  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

p_is_probe_triggered [out] 指標型態的記憶體，用來儲存 Touch Probe 2 的觸發狀態。
 若軸處於 Touch Probe 2 觸發狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsTouchProbe2Triggered
LabVIEW	-
Python	IsTouchProbe2Triggered

11.10 HIMC_GetTouchProbe1Pos

用途

取得軸的 Touch Probe 1 鎖存位置。

語法

```
int HIMC_GetTouchProbe1Pos(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_get_probe_pos  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in]

軸編號。
- p_get_probe_pos [out]

指標型態的記憶體，用來儲存軸的 Touch Probe 1 鎖存位置。
參數單位：mm (毫米) 或 deg (角度)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetTouchProbe1Pos
LabVIEW	-
Python	GetTouchProbe1Pos

11.11 HIMC_GetTouchProbe2Pos

用途

取得軸的 Touch Probe 2 鎖存位置。

語法

```
int HIMC_GetTouchProbe2Pos(
    int    ctrl_id,
    int    axis_id,
    double *p_get_probe_pos
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
p_get_probe_pos [out]	指標型態的記憶體，用來儲存軸的 Touch Probe 2 鎖存位置。 參數單位：mm (毫米) 或 deg (角度)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetTouchProbe2Pos
LabVIEW	-
Python	GetTouchProbe2Pos

11.12 HIMC_SetTouchProbe1Func

用途

設定軸的 Touch Probe 1 功能。

語法

```
int HIMC_SetTouchProbe1Func(  
    int ctrl_id,  
    int axis_id,  
    int tp_input,  
    int cont_trigger,  
    int detect_edge  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- tp_input [in] Touch probe 1 訊號來源。
 輸入範圍：0 (外部訊號)、1 (Z-phase)
- cont_trigger [in] Touch probe 1 觸發模式。
 輸入範圍：0 (單次觸發)、1 (連續觸發)
- detect_edge [in] Touch probe 1 邊緣偵測模式。
 輸入範圍：0 (正緣觸發)、1 (負緣觸發)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetTouchProbe1Func
LabVIEW	-
Python	SetTouchProbe1Func

11.13 HIMC_SetTouchProbe2Func

用途

設定軸的 Touch Probe 2 功能。

語法

```
int HIMC_SetTouchProbe2Func(  
    int ctrl_id,  
    int axis_id,  
    int tp_input,  
    int cont_trigger,  
    int detect_edge  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- tp_input [in] Touch probe 2 訊號來源。
 輸入範圍：0 (外部訊號)、1 (Z-phase)
- cont_trigger [in] Touch probe 2 觸發模式。
 輸入範圍：0 (單次觸發)、1 (連續觸發)
- detect_edge [in] Touch probe 2 邊緣偵測模式。
 輸入範圍：0 (正緣觸發)、1 (負緣觸發)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetTouchProbe2Func
LabVIEW	-
Python	SetTouchProbe2Func

12. 動態誤差補償函式

12.	動態誤差補償函式.....	12-1
12.1	概述	12-2
12.2	HIMC_EnableComp	12-4
12.3	HIMC_DisableComp	12-5
12.4	HIMC_SetupComp	12-6
12.5	HIMC_SetupComp2D	12-8
12.6	HIMC_SetupComp3D	12-10
12.7	HIMC_GetCompPos	12-12
12.8	HIMC_SetCompAlgType	12-13

12.1 概述

HIMC 提供動態 1D / 2D / 3D 誤差補償功能，依據相關誤差量測與計算結果，使用者可建立誤差補償表並在 HIMC 進行設定。設定的參數包含被補償軸、參考軸、補償點的間距 (Interval)、補償點的起始位置 (Base)、補償點的數量 (Number) 與各補償點的補償值 (Value)；其中補償值的設定須使用 HIMC User Table 的記憶體空間，並提供誤差補償表的第一個 ID 位置於 User Table 中。

註 1：有關 User Table 的使用與詳細說明，請參閱第 9 章。

註 2：啟用動態誤差補償前，軸須先完成歸原點以固定被補償軸與參考軸的座標位置。

至於被補償軸與參考軸，可選擇同一軸，既是被補償軸，也是參考軸；亦可選擇多個不同的軸作為被補償軸的參考軸，如被補償軸為 Z 軸，參考軸為 X 和 Y 軸；而被補償軸的補償值會依據參考軸的運動位置而變化。建立的誤差補償表會在軸運動過程中，以線性插補的方式計算補償點之間的補償命令值，使運動過程中的補償值能夠保持連續，而不會產生命令跳躍的情形。而當軸位置超過補償表所建立的範圍時，HIMC 會使用最近補償點的補償值作為其補償命令，如圖 12.1.1 所示。

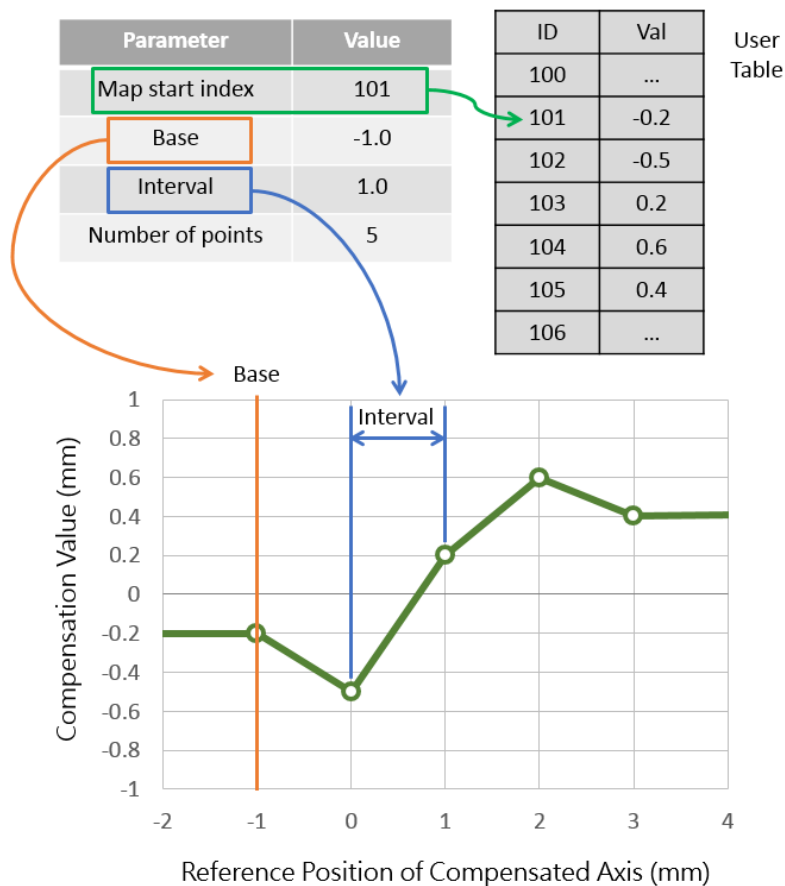


圖 12.1.1

啟用動態誤差補償後，控制器的軸控制命令中，輸出的位置命令會疊加上欲補償的位移量以消除量測已知的誤差量，如圖 3.1.1 所示，其關係為：

參考位置命令 (Reference Position) + 補償值 (Position Compensation) = 位置命令輸出 (Position Output)

啟用動態誤差補償後，可利用 iA Studio 的 Scope Manger 來觀察變數。

- 補償值：Axis → Motion Variable → Position Compensation
- 位置命令 (不含補償)：Axis → Motion Variable → Reference Position
- 位置命令 (含補償)：Axis → Motion Variable → Position Output

限制：

補償命令在 HIMC 中沒有經過路徑的軌跡規劃器，控制器預設最大的誤差補償值為 1 mm。若補償值大於 1 mm，系統會顯示錯誤訊息以提示使用者。

啟用動態誤差補償時，須固定補償的參考座標，故無法變更軸的原點偏移量。

12.2 HIMC_EnableComp

用途

啟用軸的動態誤差補償。

語法

```
int HIMC_EnableComp(
    int ctrl_id,
    int axis_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

當軸處於激磁狀態，此函式不適用。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_EnableComp
LabVIEW	HIMC Enable Comp.vi
Python	EnableComp

12.3 HIMC_DisableComp

用途

停用軸的動態誤差補償。

語法

```
int HIMC_DisableComp(  
    int ctrl_id,  
    int axis_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

- (1) 軸的參考位置將被重新設定為目前位置。
- (2) 當軸處於激磁狀態，此函式不適用。
- (3) 此函式會清除原有的動態誤差補償設定。欲再啟用動態誤差補償，須重新設定。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisableComp
LabVIEW	HIMC Disable Comp.vi
Python	DisableComp

12.4 HIMC_SetupComp

用途

設定軸的一維動態誤差補償。

語法

```
int HIMC_SetupComp(  
    int    ctrl_id,  
    int    axis_id,  
    int    start_idx,  
    double base_val,  
    double interval,  
    int    num_pt,  
    int    ref_axis_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- start_idx [in] User Table 中，補償點的起始編號。
- base_val [in] 起始位置（補償輸入的最小值）。
 參數單位：mm（毫米）或 deg（角度）
- interval [in] 相鄰補償點的間距。
 參數單位：mm（毫米）或 deg（角度）
- num_pt [in] 補償點的數量。
- ref_axis_id [in] 參考軸的編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetupComp
LabVIEW	HIMC Setup Comp.vi
Python	SetupComp

12.5 HIMC_SetupComp2D

用途

設定軸的二維動態誤差補償。

語法

```
int HIMC_SetupComp2D(
    int    ctrl_id,
    int    axis_id,
    int    start_idx,
    double *base_val,
    double *interval,
    int    *num_pt,
    int    *ref_axis_id
);
```

參數

- | | |
|------------------|---|
| ctrl_id [in] | HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。 |
| axis_id [in] | 軸編號。 |
| start_idx [in] | User Table 中，補償點的起始編號。 |
| base_val [in] | 指向兩元素陣列的指標，內含各維的起始位置（補償輸入的最小值）。
參數單位：mm（毫米）或 deg（角度） |
| interval [in] | 指向兩元素陣列的指標，內含各維相鄰補償點的間距。
參數單位：mm（毫米）或 deg（角度） |
| num_pt [in] | 指向兩元素陣列的指標，內含各維補償點的數量。 |
| ref_axis_id [in] | 指向兩元素陣列的指標，內含各維參考軸的編號。 |

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetupComp2D
LabVIEW	HIMC Setup Comp2D.vi
Python	SetupComp2D

12.6 HIMC_SetupComp3D

用途

設定軸的三維動態誤差補償。

語法

```
int HIMC_SetupComp3D(
    int    ctrl_id,
    int    axis_id,
    int    start_idx,
    double *base_val,
    double *interval,
    int    *num_pt,
    int    *ref_axis_id
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
start_idx [in]	User Table 中，補償點的起始編號。
base_val [in]	指向三元素陣列的指標，內含各維的起始位置（補償輸入的最小值）。 參數單位：mm（毫米）或 deg（角度）
interval [in]	指向三元素陣列的指標，內含各維相鄰補償點的間距。 參數單位：mm（毫米）或 deg（角度）
num_pt [in]	指向三元素陣列的指標，內含各維補償點的數量。
ref_axis_id [in]	指向三元素陣列的指標，內含各維參考軸的編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetupComp3D
LabVIEW	HIMC Setup Comp3D.vi
Python	SetupComp3D

12.7 HIMC_GetCompPos

用途

取得由控制器送至驅動器的軸誤差補償值。

語法

```
int HIMC_GetCompPos(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_comp_pos  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_comp_pos [out] 指標型態的記憶體，用來儲存軸的誤差補償值。
 參數單位：mm (毫米) 或 deg (角度)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetCompPos
LabVIEW	HIMC Get Comp Pos.vi
Python	GetCompPos

12.8 HIMC_SetCompAlgType

用途

設置軸的動態誤差補償插補方式。

語法

```
int HIMC_SetCompAlgType(  
    int ctrl_id,  
    int axis_id,  
    int alg_type  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

alg_type [in] 動態誤差補償插補方式。
 0：一階線性插補（預設值）
 1：三階樣條插補

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

三維動態誤差補償不支援三階樣條插補。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetCompAlgType
LabVIEW	HIMC Set Comp Alg Type.vi
Python	SetCompAlgType

(此頁有意留白。)

13. 濾波器函式

13.	濾波器函式	13-1
13.1	概述	13-2
13.2	HIMC_EnableAxisVsf	13-3
13.3	HIMC_DisableAxisVsf	13-4
13.4	HIMC_SetAxisVsf	13-5
13.5	HIMC_EnableAxisInShape	13-7
13.6	HIMC_DisableAxisInShape	13-8
13.7	HIMC_SetAxisInShape	13-9
13.8	HIMC_EnableGrpInShape	13-11
13.9	HIMC_DisableGrpInShape	13-12
13.10	HIMC_SetGrpInShape	13-13

13.1 概述

利用濾波器函式，修正軌跡規劃器的位置命令。目前，HMPL 提供三種濾波器：平滑時間、VSF 與 InShape。

平滑時間讓馬達平穩地加速，以實現平穩運動；而 VSF 與 InShape 在運動期間抑制馬達的振動（尤其當機構的負載為懸臂時）。透過調整「頻率」和「阻尼比」，即可達到抑制震動的效果。

不能同時使用 VSF 及 InShape，但兩者皆可與平滑時間搭配。

此外，Axis InShape 函式無法對協調運動產生作用，使用者須採用 Group InShape 函式來抑制振動。

註：使用濾波器會增加移動時間，減少整定時間。

13.2 HIMC_EnableAxisVsf

用途

啟用軸的 VSF 濾波器。

語法

```
int HIMC_EnableAxisVsf(  
    int ctrl_id,  
    int axis_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_EnableAxisVsf
LabVIEW	HIMC Enable Axis Vsf.vi
Python	EnableAxisVsf

13.3 HIMC_DisableAxisVsf

用途

停用軸的 VSF 濾波器。

語法

```
int HIMC_DisableAxisVsf(  
    int ctrl_id,  
    int axis_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisableAxisVsf
LabVIEW	HIMC Disable Axis Vsf.vi
Python	DisableAxisVsf

13.4 HIMC_SetAxisVsf

用途

設置軸的 VSF 濾波器參數。

語法

```
int HIMC_SetAxisVsf(  
    int ctrl_id,  
    int axis_id,  
    double frequency,  
    double damping_ratio  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- frequency [in] 系統頻率。
 參數單位：Hz (赫茲)
 輸入範圍：0.1 ~ 200
- damping_ratio [in] 阻尼比。
 輸入範圍：0.7 ~ 1.5 (建議輸入 1.0)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetAxisVsf
LabVIEW	HIMC Set Axis Vsf.vi
Python	SetAxisVsf

13.5 HIMC_EnableAxisInShape

用途

啟用軸的 InShape 濾波器。

語法

```
int HIMC_EnableAxisInShape(  
    int ctrl_id,  
    int axis_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_EnableAxisInshape
LabVIEW	HIMC Enable Axis In Shape.vi
Python	EnableAxisInShape

13.6 HIMC_DisableAxisInShape

用途

停用軸的 InShape 濾波器。

語法

```
int HIMC_DisableAxisInShape(  
    int ctrl_id,  
    int axis_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisableAxisInshape
LabVIEW	HIMC Disable Axis In Shape.vi
Python	DisableAxisInShape

13.7 HIMC_SetAxisInShape

用途

設置軸的 InShape 濾波器參數。

語法

```
int HIMC_SetAxisInShape(  
    int ctrl_id,  
    int axis_id,  
    double frequency,  
    double damping_ratio,  
    ShaperMode shaper_type  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
frequency [in]	系統頻率。 參數單位：Hz (赫茲) 輸入範圍：3.0 ~ 300
damping_ratio [in]	阻尼比。 輸入範圍：0.0 ~ 0.3
shaper_type [in]	有兩種 shaper 類型：Shaper_Normal 與 Shaper_Robust。 Shaper_Robust 的效果比 Shaper_Normal 更強，但 Shaper_Normal 的強度足以抑制振動。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

- (1) 當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。
- (2) 系統頻率與阻尼比的預設值分別為 5.5Hz 與 0.03。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetAxisInShape
LabVIEW	HIMC Set Axis In Shape.vi
Python	SetAxisInShape

13.8 HIMC_EnableGrpInShape

用途

啟用軸群組的 InShape 濾波器。

語法

```
int HIMC_EnableGrpInShape(
    int ctrl_id,
    int group_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

group_id [in] 軸群組編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_EnableGrpInShape
LabVIEW	HIMC Enable Grp In Shape.vi
Python	EnableGrpInShape

13.9 HIMC_DisableGrpInShape

用途

停用軸群組的 InShape 濾波器。

語法

```
int HIMC_DisableGrpInShape(
    int ctrl_id,
    int group_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

group_id [in] 軸群組編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_DisableGrpInShape
LabVIEW	HIMC Disable Grp In Shape.vi
Python	DisableGrpInShape

13.10 HIMC_SetGrpInShape

用途

設置軸群組的 InShape 濾波器參數。

語法

```
int HIMC_SetGrpInShape(  
    int ctrl_id,  
    int group_id,  
    double frequency,  
    double damping_ratio,  
    ShaperMode shaper_type  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
frequency [in]	系統頻率。 參數單位：Hz (赫茲) 輸入範圍：3.0 ~ 300
damping_ratio [in]	阻尼比。 輸入範圍：0.0 ~ 0.3
shaper_type [in]	有兩種 shaper 類型：Shaper_Normal 與 Shaper_Robust。 Shaper_Robust 的效果比 Shaper_Normal 更強，但 Shaper_Normal 的強度足以抑制振動。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

- (1) 當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。
- (2) 系統頻率與阻尼比的預設值分別為 5.5Hz 與 0.03。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGrpInShape
LabVIEW	HIMC Set Grp In Shape.vi
Python	SetGrpInShape

14. HMPL Task 函式

14.	HMPL Task 函式	14-1
14.1	概述	14-2
14.2	HIMC_StartTask.....	14-3
14.3	HIMC_StartTaskFunc.....	14-4
14.4	HIMC_StopTask.....	14-5
14.5	HIMC_StopAllTask	14-6
14.6	HIMC_IsTaskStop	14-7
14.7	HIMC_LoadHMPLTask.....	14-8

14.1 概述

HIMC 內建 64 個 HMPL task 讓使用者能實作應用所需的運動規劃命令。在任一 HMPL task 中，使用者可透過 HMPL task 函式來啟動或停止其他的 HMPL task。當 HMPL task 正在執行中，則無法重複要求已在執行中的 HMPL task 重新執行，須等到 task 完成執行、進入停止狀態為止。但使用者可以詢問 HMPL task 目前是否處於正在執行的狀態，並依此對多個 HMPL task 進行應用所需的順序控制。

14.2 HIMC_StartTask

用途

開始執行 HMPL task。

語法

```
int HIMC_StartTask(  
    int ctrl_id,  
    int task_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

task_id [in] HMPL task ID。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_StartTask
LabVIEW	HIMC Start Task.vi
Python	StartTask

14.3 HIMC_StartTaskFunc

用途

開始執行 HMPL task 中的一個函式。

語法

```
int HIMC_StartTaskFunc(  
    int ctrl_id,  
    int task_id,  
    char *func_name  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- task_id [in] HMPL task ID。
- func_name [in] 指標型態的記憶體，用來儲存 HMPL task 中的函式名稱。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_StartTaskFunc
LabVIEW	HIMC Start Task Func.vi
Python	StartTaskFunc

14.4 HIMC_StopTask

用途

停止執行 HMPL task。

語法

```
int HIMC_StopTask(  
    int ctrl_id,  
    int task_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- task_id [in] HMPL task ID。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_StopTask
LabVIEW	HIMC Stop Task.vi
Python	StopTask

14.5 HIMC_StopAllTask

用途

停止執行所有 HMPL task (包含呼叫者) 。

語法

```
int HIMC_StopAllTask(  
    int ctrl_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID 。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得 。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0** 。若失敗，則回傳**非零值** 。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_StopAllTask
LabVIEW	HIMC Stop All Task.vi
Python	StopAllTask

14.6 HIMC_IsTaskStop

用途

詢問 HMPL task 是否已停止執行。

語法

```
int HIMC_IsTaskStop(  
    int ctrl_id,  
    int task_id,  
    int *isStop  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- task_id [in] HMPL task ID。
- isStop [out] 指標型態的記憶體，用來儲存 HMPL task 的狀態。
 若 HMPL task 已停止，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsTaskStop
LabVIEW	HIMC Is Task Stop.vi
Python	IsTaskStop

14.7 HIMC_LoadHMPLTask

用途

將 HMPL 檔案載入控制器。

語法

```
int HIMC_LoadHMPLTask(
    int ctrl_id,
    int task_id,
    const char *file_name
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- task_id [in] HMPL task ID。
- file_name [in] 指標型態的記憶體，用來儲存欲載入的 HMPL 檔案路徑。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

此函式須與 iA_Studio 資料夾內的 HMPL_compiler.exe 執行檔搭配使用。請複製此執行檔，並放置於使用者自行開發程式內的 bin\Debug 與 bin\Release 資料夾。

需求版本與檔案

最低支援版本	iA Studio 2.0
Executable	HMPL_compiler.exe
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_LoadHMPLTask
LabVIEW	HIMC Load HMPL Task.vi
Python	LoadHMPLTask

15. 回調函式

15.	回調函式	15-1
15.1	HIMC_SetHmplEvtCallback	15-2
15.2	HIMC_SetErrorCallback.....	15-3
15.3	HIMC_SetHmplMsgEvtCallback	15-4

15.2 HIMC_SetErrorCallback

用途

顯示回調函式，以取得控制器所傳送的錯誤 ID。

語法

```
int HIMC_SetErrorCallback(  
    int ctrl_id,  
    HimcErrorCBFuncPtr himc_error_cb_func_ptr  
);
```

參數

- ctrl_id [in]

HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- himpl_error_cb_func_ptr [in]

回調函式的指標，其原型為 void func(int arg)。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

錯誤 ID 請參閱《iA Studio 軟體使用手冊》第 5 章。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetErrorCallback
LabVIEW	-
Python	SetErrorCallback

15.3 HIMC_SetHimplMsgEvtCallback

用途

顯示回調函式，以取得 HMPL task 所傳送的字串訊息。

語法

```
int HMC_SetHmplMsgEvtCallback(  
    int ctrl_id,  
    HmplMsgEventCBFuncPtr hmpl_msg_cb_func_ptr  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
--------------	--

hmpl_msg_cb_func_ptr [in]	回調函式的指標，其原型為 <code>void func(const char *text)</code> 。
---------------------------	---

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

字串訊息的長度上限為 128 個字元。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetHmplMsgEvtCallback
LabVIEW	-
Python	SetHmplMsgEvtCallback

16. 變數與函式操作函式

16.	變數與函式操作函式	16-1
16.1	概述	16-2
16.1.1	控制器變數列表	16-3
16.2	驅動器變數操作	16-7
16.2.1	HIMC_ReadSDO	16-7
16.2.2	HIMC_WriteSDO	16-9
16.2.3	HIMC_ReadPDO	16-11
16.2.4	HIMC_WritePDO	16-12
16.2.5	HIMC_ForceWritePDO	16-13
16.2.6	HIMC_ReleasePDO	16-14
16.3	控制器變數操作	16-15
16.3.1	HIMC_GetVariableByID	16-15
16.3.2	HIMC_SetVariableByID	16-16
16.3.3	HIMC_GetVariableListByID	16-17
16.3.4	HIMC_SetVariableListByID	16-18
16.3.5	HIMC_GetGlobalVariables	16-19
16.3.6	HIMC_SetGlobalVariables	16-20

16.1 概述

HIMC 提供使用者驅動器與控制器的變數操作函式。對於驅動器的變數操作，使用者須給定驅動器物件字典的索引來存取其數值，並透過 CoE 通訊進行資料交換；而控制器的變數操作，使用者須根據控制器的變數 ID 列表，給定特定變數的定址 ID 來做存取。16.1.1 節為控制器的變數定義說明。

注意：

若無特定目的需求，建議使用者利用相關人機介面與函式來存取相關的系統變數。使用變數操作函式時，使用者須自行確保存取變數與輸入數值的安全性。

16.1.1 控制器變數列表

HIMC 使用 32 個位元做為控制器變數的定址 ID，其型式為 0x□□□□□□□□，其中 0x 表示數值為十六進位制。透過變數操作函式，使用者可存取 HIMC 提供的系統變數、軸變數與軸群組變數。定址 ID 的規則說明如下：

1. 定址 ID 的第 1~2 個數值表示『控制器變數的類別』，系統變數為 0x00□□□□□□、軸變數為 0x83□□□□□□、軸群組變數為 0x82□□□□□□。
2. 定址 ID 的第 3~4 個數值表示『軸 ID 或軸群組 ID』。例如：軸變數 0x8302□□□□為存取軸編號 02 的變數；軸群組變數 0x8201□□□□為存取軸群組編號 01 的變數。
3. 定址 ID 的第 5~8 個數值表示『控制器系統、軸或軸群組變數的定址位置』，變數列表與說明請參考表 16.1.1.1 至表 16.1.1.3。

表 16.1.1.1

系統變數		
定址 ID	變數名稱	說明
0x0000012c	HCV_ID_fclk	系統執行 Clock (每 250 us 增加 1 count)
0x0000012e	HCV_ID_timeInMs	系統執行時間 (ms)
0x000007d0	HCV_ID_user_table	使用者可自由使用的 double[512000]陣列變數
0x00002328	HCV_ID_ltest0	使用者可自由使用的 int 變數
0x00002329	HCV_ID_ltest1	使用者可自由使用的 int 變數
0x0000232a	HCV_ID_ltest2	使用者可自由使用的 int 變數
0x0000232b	HCV_ID_ltest3	使用者可自由使用的 int 變數
0x0000232c	HCV_ID_ltest4	使用者可自由使用的 int 變數
0x0000232d	HCV_ID_ltest5	使用者可自由使用的 int 變數
0x0000232e	HCV_ID_ltest6	使用者可自由使用的 int 變數
0x0000232f	HCV_ID_ltest7	使用者可自由使用的 int 變數
0x00002330	HCV_ID_ltest8	使用者可自由使用的 int 變數
0x00002331	HCV_ID_ltest9	使用者可自由使用的 int 變數
0x0000235a	HCV_ID_dtest0	使用者可自由使用的 double 變數
0x0000235b	HCV_ID_dtest1	使用者可自由使用的 double 變數
0x0000235c	HCV_ID_dtest2	使用者可自由使用的 double 變數
0x0000235d	HCV_ID_dtest3	使用者可自由使用的 double 變數
0x0000235e	HCV_ID_dtest4	使用者可自由使用的 double 變數
0x0000235f	HCV_ID_dtest5	使用者可自由使用的 double 變數
0x00002360	HCV_ID_dtest6	使用者可自由使用的 double 變數
0x00002361	HCV_ID_dtest7	使用者可自由使用的 double 變數

系統變數		
定址 ID	變數名稱	說明
0x00002362	HCV_ID_dtest8	使用者可自由使用的 double 變數
0x00002363	HCV_ID_dtest9	使用者可自由使用的 double 變數
0x0000238c	HCV_ID_mtest	使用者可自由使用的 double[10]陣列變數

表 16.1.1.2

軸變數		
定址 ID	變數名稱	說明
0x83□□0015	HCV_ID_motion_type	運動型態
0x83□□0033	HCV_ID_pos_tr	位置目標框
0x83□□0034	HCV_ID_pos_tr_t	位置目標框反彈跳時間
0x83□□0065	HCV_ID_sw_RL	軟體右極限
0x83□□0066	HCV_ID_sw_LL	軟體左極限
0x83□□0067	HCV_ID_vel_lim	最大速度限制
0x83□□0068	HCV_ID_acc_lim	最大加速度限制
0x83□□0069	HCV_ID_dec_lim	最大減速度限制
0x83□□0079	HCV_ID_max_pos_err	位置誤差限制
0x83□□007a	HCV_ID_max_comp_lim	位置補償限制
0x83□□00a0	HCV_ID_home_status	歸原點狀態
0x83□□00a1	HCV_ID_home_method	歸原點方法
0x83□□00a2	HCV_ID_home_fast_vel	快速歸原點速度
0x83□□00a3	HCV_ID_home_slow_vel	慢速歸原點速度
0x83□□00a4	HCV_ID_home_timeout	歸原點逾時時間
0x83□□00a5	HCV_ID_home_acc	歸原點加速度
0x83□□00a6	HCV_ID_home_offset	歸原點位置偏移量
0x83□□00d3	HCV_ID_max_vel	目標速度
0x83□□00d4	HCV_ID_max_acc	目標加速度
0x83□□00d5	HCV_ID_max_dec	目標減速度
0x83□□00d7	HCV_ID_sm_factor	平滑時間
0x83□□00db	HCV_ID_vel_scale	速度百分比 (0~100)
0x83□□00dd	HCV_ID_p2p_del	P2P 運動等待時間
0x83□□00de	HCV_ID_p2p_pos1	P2P 位置 1
0x83□□00df	HCV_ID_p2p_pos2	P2P 位置 2
0x83□□00e0	HCV_ID_p2p_repeat	重覆 P2P 運動
0x83□□00e1	HCV_ID_rlt_dist	相對移動距離
0x83□□00e2	HCV_ID_en_motionManager	Motion Manager 運動軸選擇
0x83□□00e3	HCV_ID_acc_time	加速度時間

軸變數		
定址 ID	變數名稱	說明
0x83□□00e4	HCV_ID_dec_time	減速度時間
0x83□□00e9	HCV_ID_map_io_type	誤差補償類型
0x83□□0117	HCV_ID_rollover_turns	單圈模式 rollover 次數
0x83□□0119	HCV_ID_rollover_val	單圈模式 rollover 設定值
0x83□□0193	HCV_ID_gant_pair	龍門配置的龍門 ID
0x83□□01f7	HCV_ID_en_delay	軸激磁 Time Out 時間
0x83□□01ff	HCV_ID_fb_ratio_pos	驅動器位置解析度，長度單位（分子）
0x83□□0200	HCV_ID_fb_ratio_cnt	驅動器位置解析度，單位 count（分母）
0x83□□0209	HCV_ID_fb_curr_ratio_curr	驅動器電流解析度，電流單位（分子）
0x83□□020a	HCV_ID_fb_curr_ratio_cnt	驅動器電流解析度，單位 count（分母）
0x83□□0213	HCV_ID_rotor_inertia	馬達轉子慣量比
0x83□□0214	HCV_ID_force_constant	馬達轉矩常數
0x83□□0263	HCV_ID_last_err	軸錯誤代碼
0x83□□03c2	HCV_ID_gear_ratio	電子齒輪比

註：符號□□會是該軸的 ID，ID 會以十六進位制表示。例如：01 為軸編號 01；0f 為軸編號 15。

表 16.1.1.3

軸群組變數		
定址 ID	變數名稱	說明
0x82□□0002	HCV_ID_grp_num_axis	軸群組軸數
0x82□□00c9	HCV_ID_grp_lin_vel_lim	軸群組線性運動速度限制
0x82□□00ca	HCV_ID_grp_lin_acc_lim	軸群組線性運動加速度限制
0x82□□00cb	HCV_ID_grp_lin_dec_lim	軸群組線性運動減速度限制
0x82□□00d4	HCV_ID_grp_ang_vel_lim	軸群組旋轉運動速度限制
0x82□□00d5	HCV_ID_grp_ang_acc_lim	軸群組旋轉運動加速度限制
0x82□□00d6	HCV_ID_grp_ang_dec_lim	軸群組旋轉運動減速度限制
0x82□□00d0	HCV_ID_grp_lin_vel	軸群組目標速度
0x82□□00d1	HCV_ID_grp_lin_acc	軸群組目標加速度
0x82□□00d2	HCV_ID_grp_lin_dec	軸群組目標減速度
0x82□□00d3	HCV_ID_grp_lin_sf	軸群組平滑時間
0x82□□00f0	HCV_ID_grp_lin_acc_time	軸群組目標加速度時間
0x82□□00f1	HCV_ID_grp_lin_dec_time	軸群組目標減速度時間
0x82□□00e7	HCV_ID_grp_ang_vel	軸群組旋轉運動目標速度
0x82□□00e8	HCV_ID_grp_ang_acc	軸群組旋轉運動目標加速度
0x82□□00e9	HCV_ID_grp_ang_dec	軸群組旋轉運動目標減速度
0x82□□00ea	HCV_ID_grp_ang_sf	軸群組旋轉運動平滑時間

軸群組變數		
定址 ID	變數名稱	說明
0x82□□00eb	HCV_ID_grp_ang_acc_time	軸群組旋轉運動目標加速度時間
0x82□□00ec	HCV_ID_grp_ang_dec_time	軸群組旋轉運動目標減速度時間
0x82□□00e4	HCV_ID_grp_coord_sys	軸群組座標系統
0x82□□00e5	HCV_ID_grp_buffer_mode	軸群組速度緩衝模式
0x82□□00e6	HCV_ID_grp_trans_mode	軸群組路徑過渡模式
0x82□□00e7	HCV_ID_grp_trans_vel	軸群組路徑過渡速度
0x82□□00e8	HCV_ID_grp_trans_dis	軸群組路徑過渡距離
0x82□□00f6	HCV_ID_grp_trans_dev	軸群組路徑過渡偏移量
0x82□□00f7	HCV_ID_grp_trans_curvature	軸群組路徑過渡曲率
0x82□□00eb	HCV_ID_grp_vel_scale	軸群組速度百分比 (0~100)
0x82□□00da	HCV_ID_grp_shaper_fr	軸群組 InShape 濾波器頻率
0x82□□00db	HCV_ID_grp_shaper_xi	軸群組 InShape 濾波器阻尼比
0x82□□038f	HCV_ID_grp_last_err	軸群組錯誤代碼

註：符號□□會是該軸群組的 ID，ID 會以十六進位制表示。例如：01 為軸群組編號 01；0f 為軸群組編號 15。

16.2 驅動器變數操作

16.2.1 HIMC_ReadSDO

用途

透過 SDO 讀取從站的物件數值。

語法

```
int HIMC_ReadSDO(  
    int ctrl_id,  
    int slv_id,  
    int obj_index,  
    int obj_subindex,  
    int size_bytes,  
    int64_t *value  
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
slv_id [in]	從站編號。
obj_index [in]	從站物件的索引。
obj_subindex [in]	從站物件的子索引。
size_bytes [in]	從站物件的 Byte 長度。
value [out]	指標型態的記憶體，用來儲存讀取的物件數值。 若需調整數值顯示範圍，請參考 1.5 節自行轉換型態。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ReadSDO
LabVIEW	HIMC Read SDO.vi
Python	ReadSDO

16.2.2 HIMC_WriteSDO

用途

透過 SDO 將數值寫入從站的物件。

語法

```
int HIMC_WriteSDO(  
    int ctrl_id,  
    int slv_id,  
    int obj_index,  
    int obj_subindex,  
    int size_bytes,  
    int64_t value  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_id [in] 從站編號。
- obj_index [in] 從站物件的索引。
- obj_subindex [in] 從站物件的子索引。
- size_bytes [in] 從站物件的 Byte 長度。
- value [in] 欲寫入的物件數值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_WriteSDO
LabVIEW	HIMC Write SDO.vi
Python	WriteSDO

16.2.3 HIMC_ReadPDO

用途

透過 PDO 讀取從站已配置的 PDO 物件數值。

語法

```
int HIMC_ReadPDO(  
    int ctrl_id,  
    int slv_id,  
    int obj_index,  
    int obj_subindex,  
    int64_t *value  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_id [in] 從站編號。
- obj_index [in] 從站物件的索引。
- obj_subindex [in] 從站物件的子索引。
- value [out] 指標型態的記憶體，用來儲存讀取的物件數值。
 若需調整數值顯示範圍，請參考 1.5 節自行轉換型態。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ReadPDO
LabVIEW	HIMC Read PDO.vis
Python	ReadPDO

16.2.4 HIMC_WritePDO

用途

透過 PDO 將數值寫入從站的已配置的 PDO 物件。

語法

```
int HIMC_WritePDO(
    int ctrl_id,
    int slv_id,
    int obj_index,
    int obj_subindex,
    int64_t value
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_id [in] 從站編號。
- obj_index [in] 從站物件的索引。
- obj_subindex [in] 從站物件的子索引。
- value [in] 欲寫入的物件數值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_WritePDO
LabVIEW	HIMC Write PDO.vi
Python	WritePDO

16.2.5 HIMC_ForceWritePDO

用途

透過 PDO 將數值強制寫入從站的已配置的 PDO 物件。

語法

```
int HIMC_ForceWritePDO(
    int ctrl_id,
    int slv_id,
    int obj_index,
    int obj_subindex,
    int64_t value
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_id [in] 從站編號。
- obj_index [in] 從站物件的索引。
- obj_subindex [in] 從站物件的子索引。
- value [in] 欲寫入的物件數值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ForceWritePDO
LabVIEW	HIMC Force Write PDO.vi
Python	ForceWritePDO

16.2.6 HIMC_ReleasePDO

用途

釋放被強制寫入的 PDO 物件，與 HIMC_ForceWritePDO 函式搭配使用。

語法

```
int HIMC_ReleasePDO(
    int ctrl_id,
    int slv_id,
    int obj_index,
    int obj_subindex
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。

slv_id [in] 從站編號。

obj_index [in] 從站物件的索引。

obj_subindex [in] 從站物件的子索引。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ReleasePDO
LabVIEW	HIMC Release PDO.vi
Python	ReleasePDO

16.3 控制器變數操作

16.3.1 HIMC_GetVariableByID

用途

藉 ID 取得控制器的變數值。

語法

```
int HIMC_GetVariableByID(
    int    ctrl_id,
    int    var_id,
    double *p_val
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- var_id [in] HIMC 控制器變數 ID，HCV ID 定義請參閱 16.1.1 節。
- p_val [out] 指標型態的記憶體，用來儲存變數值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetVariableByID
LabVIEW	HIMC Get Variable By ID.vi
Python	GetVariableByID

16.3.2 HIMC_SetVariableByID

用途

藉 ID 設置控制器的變數值。

語法

```
int HIMC_SetVariableByID(
    int    ctrl_id,
    int    var_id,
    double val
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- var_id [in] HIMC 控制器變數 ID，HCV ID 定義請參閱 16.1.1 節。
- val [in] 新的變數值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetVariableByID
LabVIEW	HIMC Set Variable By ID.vi
Python	SetVariableByID

16.3.3 HIMC_GetVariableListByID

用途

藉 ID 取得控制器的多個變數值。

語法

```
int HIMC_GetVariableListByID(
    int    ctrl_id,
    int    *p_var_id,
    int    num,
    double *p_val
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_var_id [in] 指標型態的記憶體，用來儲存多個 HIMC 控制器變數的 ID。
HCV ID 定義請參閱 16.1.1 節。
- num [in] 變數的數量。
- p_val [out] 指標型態的記憶體，用來儲存多個變數值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetVariableListByID
LabVIEW	HIMC Get Variable List By ID.vi
Python	GetVariableListByID

16.3.4 HIMC_SetVariableListByID

用途

藉 ID 設置控制器的多個變數值。

語法

```
int HIMC_SetVariableListByID(
    int    ctrl_id,
    int    *p_var_id,
    int    num,
    double *p_val
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_var_id [in] 指標型態的記憶體，用來儲存多個 HIMC 控制器變數的 ID。
 HCV ID 定義請參閱 16.1.1 節。
- num [in] 變數的數量。
- p_val [in] 指標型態的記憶體，用來儲存多個新的變數值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetVariableListByID
LabVIEW	HIMC Set Variable List By ID.vi
Python	SetVariableListByID

16.3.5 HIMC_GetGlobalVariables

用途

取得控制器的多個全域變數值。

語法

```
int HIMC_GetGlobalVariables(  
    int    ctrl_id,  
    char   **pp_var_name_array,  
    int    length,  
    double *p_output_array  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- pp_var_name_array [in] 指標型態的記憶體，用來儲存多個全域變數的名稱。
- length [in] 全域變數的數量。
- p_output_array [out] 指標型態的記憶體，用來儲存多個全域變數值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
執行檔	HMPL_compiler.exe
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGlobalVariables
LabVIEW	HIMC Get Global Variables.vi
Python	GetGlobalVariables

16.3.6 HIMC_SetGlobalVariables

用途

設置控制器的多個全域變數值。

語法

```
int HIMC_SetGlobalVariables(
    int    ctrl_id,
    char    **pp_var_name_array,
    int    length,
    double *p_input_array
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- pp_var_name_array [in] 指標型態的記憶體，用來儲存多個全域變數的名稱。
- length [in] 全域變數的數量。
- p_input_array [in] 指標型態的記憶體，用來儲存多個新的全域變數值。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
執行檔	HMPL_compiler.exe
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetGlobalVariables
LabVIEW	HIMC Set Global Variables.vi
Python	SetGlobalVariables

17. HIMC 錯誤函式

17.	HIMC 錯誤函式	17-1
17.1	概述	17-2
17.1.1	系統錯誤訊息	17-3
17.1.2	軸錯誤訊息	17-6
17.1.3	軸群組錯誤訊息	17-10
17.2	HIMC_GetLastError	17-12
17.3	HIMC_GetAxisLastErr	17-13
17.4	HIMC_ClearAxisLastErr	17-14
17.5	HIMC_GetGrpLastErr	17-15
17.6	HIMC_ClearGrpLastErr	17-16
17.7	HIMC_GetDriveErr	17-17
17.8	HIMC_GetErrorInformation	17-18

17.1 概述

HIMC 使用 32 位元的錯誤代碼來表示相關的錯誤訊息。透過本章提供的函式，使用者可取得或清除系統、軸與軸群組的錯誤代碼（各類別之錯誤代碼、名稱與說明分別條列於 17.1.1 節至 17.1.3 節）。錯誤代碼的型式為 0x□□□□□□□□，其中 0x 表示數值為十六進位制。其規則與控制器變數定址 ID 相同，說明如下：

1. 錯誤代碼的第 1~2 個數值表示『控制器變數的類別』，系統變數為 0x00□□□□□□、軸變數為 0x83□□□□□□、軸群組變數為 0x82□□□□□□。
2. 錯誤代碼的第 3~4 個數值表示『軸 ID 或軸群組 ID』。例如：軸變數 0x8302□□□□為存取軸編號 02 的變數；軸群組變數 0x8201□□□□為存取軸群組編號 01 的變數。
3. 錯誤代碼的第 5~8 個數值表示『變數 ID』，詳細說明請參閱 17.1.1 節至 17.1.3 節。

註：由於函式回傳值為十進位制，使用者須自行轉換成十六進位制，才能對應到正確的錯誤代碼。

17.1.1 系統錯誤訊息

表 17.1.1.1

系統錯誤代碼		
錯誤代碼	錯誤名稱	說明
0x00000001	eERR_HCV_ID_NOT_FOUND	無法找到此變數 ID。
0x00000002	eERR_DATA_EXCEEDED	要求的資料超出範圍。
0x00000003	eERR_HCV_IS_READ_ONLY	此參數唯讀。
0x00000004	eERR_HCV_VALUE_OUT_OF_RANGE	輸入值超出範圍限制。
0x00000050	eERR_EWM_CALLBACK_BUSY	EWM 回呼函式處於忙碌狀態。
0x00000064	eERR_EMERGENCY_STOP	控制器已緊急停止。
0x00000107	eERR_NOT_VALID_TASKID	此 task ID 無效。
0x00000108	eERR_TASK_IS_RUNNING	此 task 正在執行中。
0x00000109	eERR_FUNC_NOT_IN_TASK	此 task 未包含此函式。
0x0000010a	eERR_TASK_EMPTY	此 task 沒有內容。
0x0000010b	eERR_TASK_NOT_RUNNING	此 task 未在執行。
0x0000012c	eERR_NIC_INIT_TOUT	總線通訊網路未就緒。
0x0000012d	eERR_HARDWARE_MISMATCH	無法辨識的硬體。
0x0000012e	eERR_SLAVE_NUM_MISMATCH	從站數量與組態不符。
0x00000130	eERR_INVALID_MCK_CNFG	運動核心 (motion kernel) 組態無效。
0x00000138	eERR_HIMC_LOAD_CONFIG_FAIL	無法從 SSD 下載組態，請再儲存一次。
0x00000139	eERR_HIMC_SAVE_CONFIG_FAIL	無法將組態存至 HIMC，請再儲存一次。
0x0000013a	eERR_HIMC_SAVE_CONFIG_COPY_FAIL	無法將組態存至 HIMC。無法將檔案存至 SAVE 資料夾。
0x0000013c	eERR_ETHERCAT_LICENSE_MISMATCH	EtherCAT 授權不匹配。
0x000001f4	eERR_ISR_NOT_STABLE	中斷週期不穩定。
0x000041f4	eWRN_ISR_NOT_STABLE	中斷週期不穩定預警。
0x000001f5	eERR_MCK_OVERLOAD	運動核心 (motion kernel) 過載。
0x0000C1f5	eDBG_MCK_OVERLOAD	運動核心 (motion kernel) 過載。(預警)
0x000001f6	eERR_ISR_OVERLOAD	CPU 過載。
0x0000C1f6	eDBG_ISR_OVERLOAD	CPU 過載。(預警)
0x00001388	eERR_HMPL_INVALID_ARG	HMPL 內的引數無效。
0x00001389	eERR_HMPL_INVALID_PTR	HMPL 內的指標 (pointer) 無效。
0x0000138a	eERR_HMPL_STACK_OVERFLOW	HMPL 內堆疊溢位。
0x0000138b	eERR_HMPL_ILLEGAL_MEM_OP	此記憶體操作在 HMPL 內是非法的。
0x0000138c	eERR_HMPL_MOTION_NOT_READY	控制器狀態未就緒，無法執行此 HMPL。
0x0000138d	eERR_HMPL_STR_TOO_LONG	字串長度超過限制。
0x0000138e	eERR_HMPL_INVALID_STR_FORMAT	字串格式錯誤。

系統錯誤代碼		
錯誤代碼	錯誤名稱	說明
0x0000138f	eERR_HMPL_ARG_OUT_OF_RANGE	輸入的引數超過範圍。
0x00001392	eERR_HMPL_ASCII_AGENT_RUNNING	ASCII agent 執行中，無法同時執行多個 ASCII agent。
0x0000139c	eERR_HMPL_CANNOT_RUN_IN_DEBUG	此 HMPL 函式無法在偵錯模式中執行。
0x000013a6	eERR_HMPL_TOO_MANY_BRK_POINT	此 task 有太多中斷點。
0x000013ec	eERR_HMPL_MUTEX_LOCK_TWICE	重複鎖定已鎖定的互斥鎖。
0x00001450	eERR_HMPL_INVALID_SYS_TIME_MEMORY	記憶體太小，至少要 30 Bytes。
0x00001451	eERR_HMPL_NOT_SUPPORTED	此平台不支援此 HMPL 函式。
0x00001452	eERR_HMPL_CLIENT_NOT_CONNECTED	無法連線至客戶端。
0x000014b4	eERR_HMPL_NOT_IN_OP_MODE	此 HMPL 函式無法在非同步狀態中執行。
0x0000176f	eERR_HMPL_INTERNAL_ERROR	HMPL 內部錯誤。
0x00001770	eERR_HMPL_EXEC_FAILED	HMPL 執行失敗。
0x00001771	eERR_HMPL_ASM_LOAD_FAILED	HMPL 編譯失敗。組合語言檔案不存在或未產生。
0x00001772	eERR_HMPL_STARTTASK_TIMEOUT	HMPL StartTask 函式逾時。
0x00001773	eERR_HMPL_STOPTASK_TIMEOUT	HMPL StopTask 函式逾時。
0x000017d4	eERR_ASCII_CONNECT_TIMEOUT	ASCII 用戶端連線逾時。
0x000017d5	eERR_ASCII_CONNECT_FAILED	ASCII 用戶端連線失敗，請檢查 ip 及連接埠。
0x000017d6	eERR_ASCII_MULTI_CONNECTING	多個 ASCII 用戶端同時連線。
0x000017d7	eERR_ASCII_MULTI_DISCONNECTING	多個 ASCII 用戶端同時斷線。
0x000017d8	eERR_ASCII_DISCONNECT_TIMEOUT	ASCII 用戶端斷線逾時。
0x000017de	eERR_ASCII_RECV_TIMEOUT	ASCII 用戶端接收資料逾時，請稍候再試。
0x000017df	eERR_ASCII_RECV_FAIL	ASCII 用戶端接收資料失敗，請確認連線是否正常。
0x000017e0	eERR_ASCII_MULTI_RECVING	多個 ASCII 用戶端同時接受資料。
0x000017e8	eERR_ASCII_SEND_TIMEOUT	ASCII 用戶端傳送資料逾時，請稍候再試。
0x000017e9	eERR_ASCII_SEND_FAIL	ASCII 用戶端傳送資料失敗，請確認連線是否正常。
0x000017ea	eERR_ASCII_MULTI_SENDING	多個 ASCII 用戶端同時傳送資料。
0x00001838	eERR_MODBUS_CONNECT_TIMEOUT	Modbus 用戶端連線逾時。
0x00001839	eERR_MODBUS_CONNECT_FAILED	Modbus 用戶端連線失敗，請檢查 ip。
0x0000183a	eERR_MODBUS_MULTI_CONNECTING	多個 Modbus 用戶端同時連線。
0x0000183b	eERR_MODBUS_MULTI_DISCONNECTING	多個 Modbus 用戶端同時斷線。
0x0000183c	eERR_MODBUS_DISCONNECT_TIMEOUT	Modbus 用戶端斷線逾時。
0x0000183d	eERR_MODBUS_DATALENGTH_ERR	Modbus 用戶端的讀寫資料數量超過上限。
0x0000183e	eERR_MODBUS_SOCKET_BUSY	Modbus 用戶端同時處理兩個以上的命令。

系統錯誤代碼		
錯誤代碼	錯誤名稱	說明
0x0000183f	eERR_MODBUS_JOB_TIMEOUT	Modbus 用戶端執行任務逾時，請稍候再試。
0x00001840	eERR_MODBUS_JOB_FAIL	Modbus 用戶端執行任務失敗，請確認連線是否正常。
0x0000b037	eMSG_HIMC_SET_DEFAULT	還原為出廠預設。
0x0000b038	eMSG_HIMC_REBOOT	重新啟動控制器。
0x0000b039	eMSG_HIMC_BOOT	啟動控制器。
0x0000b03a	eMSG_HIMC_INFO	控制器資訊。
0x0000b03b	eMSG_HIMC_STORE_CONFIG	儲存控制器組態。
0x0000b03e	eMSG_API_MAIN_ID_CHANGE_GET	取得存取優先權。
0x0000b03f	eMSG_API_MAIN_ID_CHANGE_RELEASE	釋放存取優先權。
0x0000b2c8	eMSG_START_HMI_SCOPE	啟動示波器監控。
0x0000b2c9	eMSG_STOP_HMI_SCOPE	停止示波器監控。
0x00003fff	eERR_SYS_LOG	控制器錯誤。
0x00007fff	eWRN_SYS_LOG	控制器警告。
0x0000bfff	eMSG_SYS_LOG	控制器訊息。
0x0000ffff	eDBG_SYS_LOG	控制器偵錯。

17.1.2 軸錯誤訊息

以下錯誤代碼會出現在軸發生錯誤或操作無效時。錯誤發生時，符號□□會是該軸的 ID，ID 會以十六進位制表示。例如：01 為軸編號 01；0f 為軸編號 15。

表 17.1.2.1

軸錯誤代碼		
錯誤代碼	錯誤名稱	說明
0x83□□000a	eERR_AXIS_CMD_UNKOWN	未知的命令名稱。
0x83□□001e	eERR_AXIS_CMD_QUEUE_FULL	軸命令佇列 (queue) 已滿。
0x83□□0050	eERR_AXIS_CMD_INVALID_IN_SIM	虛擬機無法使用此命令。
0x83□□0060	eERR_AXIS_CMD_INVALID_IN_VIRTUAL	虛擬軸無法使用此命令。
0x83□□0064	eERR_AXIS_CMD_INVALID_STATE	軸在當前的狀態下無法執行此命令。
0x83□□006e	eERR_AXIS_CMD_INVALID_ENABLED	軸激磁時無法使用此命令。
0x83□□0078	eERR_AXIS_CMD_INVALID_DISABLED	軸解激磁時無法使用此命令。
0x83□□0082	eERR_AXIS_CMD_INVALID_MOVING	軸移動時無法執行此命令。
0x83□□008c	eERR_AXIS_CMD_INVALID_STOPPING	軸停止時無法執行此命令。
0x83□□0096	eERR_AXIS_CMD_INVALID_ERROR_STATE	軸發生錯誤時無法執行此命令。
0x83□□00a0	eERR_AXIS_CMD_INVALID_IN_SYNC	軸處於同步運動狀態時，此命令無效。
0x83□□00aa	eERR_AXIS_CMD_INVALID_GEAR_MASTER	軸為電子齒輪主軸時，此命令無效。
0x83□□00b4	eERR_AXIS_CMD_INVALID_PP_MODE	當軸在 PP 模式下，此命令無效。
0x83□□00be	eERR_AXIS_CMD_INVALID_MAP_SWITCHING	當軸在切換補償表時，此命令無效。
0x83□□00c8	eERR_AXIS_CMD_INVALID_INPUTSHAPING_ENABLED	軸位置命令塑型功能啟用時，此命令無效。
0x83□□00d2	eERR_AXIS_CMD_INVALID_COMP_ENABLED	動態補償功能啟用時，此命令無效。
0x83□□00dc	eERR_AXIS_CMD_INVALID_GANTRY_MODE	軸處於龍門模式時，此命令無效。
0x83□□00e6	eERR_AXIS_CMD_INVALID_GROUPED	若軸已在群組內，此命令無效。
0x83□□00f0	eERR_AXIS_CMD_INVALID_CONTROL_MODE	在當前的控制模式下，此命令無效。
0x83□□00fa	eERR_AXIS_CMD_INVALID_OP_MODE	運行模式無效。
0x83□□0104	eERR_AXIS_CMD_INVALID_BUFFER_MODE	軸緩衝模式無效。
0x83□□0105	eERR_AXIS_CMD_INVALID_SETBUFFERMODE	軸有未完成的命令時，無法使用此命令。
0x83□□010e	eERR_AXIS_CMD_INVALID_TP_ENABLED	Touch Probe 啟用時，無法使用此命令。
0x83□□012c	eERR_AXIS_CMD_INVALID_PARAMETER	命令參數無效。
0x83□□0136	eERR_AXIS_CMD_INVALID_POS	軸目標位置超出允許範圍。
0x83□□0140	eERR_AXIS_CMD_INVALID_VEL	軸速度設定超出允許範圍。

軸錯誤代碼		
錯誤代碼	錯誤名稱	說明
0x83□□014a	eERR_AXIS_CMD_INVALID_ACC	軸加速度設定超出允許範圍。
0x83□□0154	eERR_AXIS_CMD_INVALID_DEC	軸減速度設定超出允許範圍。
0x83□□015e	eERR_AXIS_CMD_INVALID_JERK	軸急跳度 (jerk) 設定超出允許範圍。
0x83□□0168	eERR_AXIS_CMD_INVALID_SM_TIME	軸平滑時間設定超出允許範圍。
0x83□□0172	eERR_AXIS_CMD_INVALID_KILL_DEC	軸緊急減速度設定超出允許範圍。
0x83□□017c	eERR_AXIS_CMD_INVALID_VEL_SCALE	軸速度百分比設定超出允許範圍。
0x83□□0190	eERR_AXIS_COMP_NOT_CNFG	軸動態補償設定未妥善配置。
0x83□□01c2	eERR_AXIS_CMD_INVALID_MASTER_SLAVE_CONNECTION	主從軸關係設定無效。
0x83□□01cc	eERR_AXIS_CMD_INVALID_SLAVE_ID	從軸 ID 設定無效。
0x83□□01d6	eERR_AXIS_CMD_INVALID_GEAR_RATIO	從軸齒輪比設定超出允許範圍。
0x83□□01f4	eERR_AXIS_CMD_INVALID_ROLLOVER_POS	無效的軸單圈模式位置設置，應調整為正值。
0x83□□01fe	eERR_AXIS_CMD_INVALID_ROLLOVER_PP_MODE	單圈模式在軌跡位置模式下不支援。切換至軌跡位置模式前，請將單圈模式設定值重設為 0。
0x83□□0208	eERR_AXIS_CMD_INVALID_FORCECONST_T RQMODE	轉矩模式下無效的轉矩常數。請先設定正確的轉矩常數。
0x83□□0258	eERR_AXIS_CMD_INVALID_NO_SPECIFIC_P DO	PDO 需具有特定 CoE 物件才能執行此命令。
0x83□□03f2	eERR_AXIS_DRIVE_FAULT	驅動器發生錯誤。
0x83□□03fc	eERR_AXIS_DRIVE_ABNORMAL_DISABLE	驅動器不正常解激磁。
0x83□□0406	eERR_AXIS_DRIVE_ENABLE_TOUT	激磁驅動器的時間過長。
0x83□□0407	eERR_AXIS_DRIVE_ENABLE_TOUT_RMT	激磁驅動器的時間過長。 請檢查驅動器 Access 設定。
0x83□□0410	eERR_AXIS_DRIVE_CLEAR_ERROR_TOUT	清除驅動器錯誤的時間過長。
0x83□□041a	eERR_AXIS_DRIVE_DISABLE_TOUT	解激磁驅動器的時間過長。
0x83□□0424	eERR_AXIS_DRIVE_HOME_TOUT	軸歸原點的時間過長。
0x83□□042e	eERR_AXIS_DRIVE_HOME_FAILED	軸歸原點錯誤。請檢查驅動器錯誤代碼。
0x83□□0456	eERR_AXIS_VEL_LIMIT	軸速度超過速度極限。
0x83□□4456	eWRN_AXIS_VEL_LIMIT	軸速度受速度極限制警告。
0x83□□0460	eERR_AXIS_ACC_LIMIT	軸加速度超過加速度極限。
0x83□□046a	eERR_AXIS_CURR_LIMIT	軸電流超過電流極限。
0x83□□0474	eERR_AXIS_DAMPINGRATIO_LIMIT	軸的阻尼比設定超出允許範圍。
0x83□□047e	eERR_AXIS_FREQUENCY_LIMIT	軸的頻率設定超出允許範圍。
0x83□□07da	eERR_AXIS_SWRL	軸位置命令碰觸到右側軟體極限。
0x83□□07e4	eERR_AXIS_SWLL	軸位置命令碰觸到左側軟體極限。

軸錯誤代碼		
錯誤代碼	錯誤名稱	說明
0x83□□07ee	eERR_AXIS_HWRL	軸的右側硬體極限訊號被觸發。
0x83□□07f8	eERR_AXIS_HWLL	軸的左側硬體極限訊號被觸發。
0x83□□0802	eERR_AXIS_COMP_LIMIT	軸誤差補償位置超出最大誤差極限。
0x83□□083e	eERR_AXIS_PERR	軸位置誤差超出位置誤差限制。請檢查是否有機構干涉。
0x83□□0848	eERR_AXIS_VERR	軸速度誤差超出速度誤差限制。請檢查是否有機構干涉。
0x83□□08a2	eERR_AXIS_PVT_MOTION_VEL_LIMIT	軸 PVT 運動的速度超過保護範圍。請檢查給定的參數是否有效。
0x83□□08ac	eERR_AXIS_PVT_MOTION_ACC_LIMIT	軸 PVT 運動的加速度超過保護範圍。請檢查給定的參數是否有效。
0x83□□08b6	eERR_AXIS_PVT_MOTION_INVALID_TIME	軸 PVT 運動的時序無效。請檢查給定的參數是否有效。
0x83□□0bb8	eERR_AXIS_CTRL_ERR	軸內部錯誤。
0x83□□0fa0	eERR_AXIS_CMD_GEAR_DISABLED	當電子齒輪停用時，電子齒輪指令不被允許執行。
0x83□□0fa1	eERR_AXIS_CMD_INVALID_AXIS_IN_CAM	當軸為電子凸輪時，電子齒輪命令無效。
0x83□□1388	eERR_CAM_CMD_INVALID_ENGAGE_WINDOW	電子凸輪咬合視窗超出允許範圍。
0x83□□1389	eERR_CAM_CMD_INVALID_ENGAGE_POSITION	電子凸輪咬合位置超出電子凸輪表範圍。
0x83□□138a	eERR_CAM_CMD_INVALID_MASTER_SCALE_FACTOR	電子凸輪主軸比例因子超出允許範圍。
0x83□□138b	eERR_CAM_CMD_INVALID_CAM_SCALE_FACTOR	電子凸輪比例因子超出允許範圍。
0x83□□138c	eERR_CAM_CMD_INVALID_CAMTABLE_ID	電子凸輪表 ID 超出允許範圍。
0x83□□138d	eERR_CAM_CMD_INVALID_DISENGAGE_WINDOW	電子凸輪脫離視窗超出允許範圍。
0x83□□138e	eERR_CAM_CMD_INVALID_DISENGAGE_POSITION	電子凸輪脫離位置超出允許範圍。
0x83□□138f	eERR_CAM_CMD_INVALID_OPERATION_IN_ENGAGED_STATE	咬合狀態下，電子凸輪命令無效。
0x83□□1390	eERR_CAM_CMD_INVALID_START_MODE	電子凸輪起始模式不符合有效列舉數值。
0x83□□1391	eERR_CAM_CMD_INVALID_MOVE_MODE	電子凸輪移動模式不符合有效列舉數值。
0x83□□1392	eERR_CAM_ENGAGED_FAILED	由於窗口過小，電子凸輪主軸可能通過咬合視窗。
0x83□□1393	eERR_CAM_CMD_NOTINUSE	目前不支援軌跡模式的終止。

軸錯誤代碼		
錯誤代碼	錯誤名稱	說明
0x83□□1394	eERR_CAM_CMD_CAM_DISABLED	電子凸輪停用時，電子凸輪命令不被允許。
0x83□□1395	eERR_CAM_CMD_INVALID_AXIS_NOT_INCAM	當軸不為電子凸輪時，電子凸輪命令無效。
0x83□□1396	eERR_CAM_CMD_INVALID_AXIS_NOT_IN_DISENGAGED	當軸未脫離時，電子凸輪命令無效。
0x83□□1397	eERR_CAM_CMD_INVALID_AXIS_IN_GEAR	當軸為電子齒輪時，電子凸輪命令無效。

17.1.3 軸群組錯誤訊息

以下錯誤代碼會出現在軸群組發生錯誤或操作無效時。錯誤發生時，符號□□會是該軸群組的 ID，ID 會以十六進位制表示。例如：01 為軸群組編號 01；0f 為軸群組編號 15。

表 17.1.3.1

軸群組錯誤代碼		
錯誤代碼	錯誤名稱	說明
0x82□□000a	eERR_CRD_CMD_UNKNOWN	未知的軸群組命令。
0x82□□0014	eERR_CRD_CMD_REACH_MAX_NUM_AXIS	軸群組達到最大軸數。
0x82□□001e	eERR_CRD_CMD_INVALID_KIN_SETTING	無效的運動類型。
0x82□□001f	eERR_CRD_CMD_INVALID_SPECIFIC_KIN	軸群組處於特殊運動學模式，此命令無效。
0x82□□0028	eERR_CRD_CMD_AXIS_DUPLICATED	無法新增軸，因該軸已在群組內。
0x82□□0032	eERR_CRD_CMD_GRP_SIZE_EMPTY	此軸群組已空。
0x82□□003c	eERR_CRD_CMD_GRP_SIZE_FULL	此軸群組已滿，無法新增軸。
0x82□□0046	eERR_CRD_CMD_INVALID_MOVING	軸群組移動時，此命令無效。
0x82□□0050	eERR_CRD_CMD_INVALID_DISABLED	軸群組未激磁，此命令無效。
0x82□□005a	eERR_CRD_CMD_INVALID_INPUTSHAPING_PARAMETER_INCOMPLETE	軸群組塑型功能的參數不完全。
0x82□□006e	eERR_CRD_CMD_INVALID_STATE	軸群組在目前的運動狀態無法執行此命令。
0x82□□0078	eERR_CRD_CMD_QUEUE_FULL	請待前一命令執行完畢。
0x82□□0082	eERR_CRD_CMD_GRP_AXIS_INVALID	軸群組無效。
0x82□□008c	eERR_CRD_CMD_QUEUE_IS_NOT_EMPTY	軸群組命令佇列 (queue) 不是空的。
0x82□□0096	eERR_CRD_CMD_INVALID_QUEUE_SIZE	軸群組命令佇列 (queue) 數量無效。
0x82□□00d2	eERR_CRD_CMD_INVALID_POS	軸群組的目標位置或方向超出允許範圍。
0x82□□00dc	eERR_CRD_CMD_INVALID_LIN_VEL	軸群組的線性速度設定超出允許範圍。
0x82□□00e6	eERR_CRD_CMD_INVALID_LIN_ACC	軸群組的線性加速度設定超出允許範圍。
0x82□□00f0	eERR_CRD_CMD_INVALID_LIN_DEC	軸群組的線性減速度設定超出允許範圍。
0x82□□00fa	eERR_CRD_CMD_INVALID_LIN_JERK	軸群組的線性急跳度 (jerk) 設定超出允許範圍。
0x82□□0104	eERR_CRD_CMD_INVALID_LIN_SM_TIME	軸群組的線性平滑時間設定超出允許範圍。
0x82□□010e	eERR_CRD_CMD_INVALID_DAMPINGRATIO	軸群組的阻尼比設定超出允許範圍。
0x82□□0118	eERR_CRD_CMD_INVALID_FREQUENCY	軸群組的頻率設定超出允許範圍。
0x82□□0140	eERR_CRD_CMD_INVALID_ANG_VEL	軸群組的旋轉速度設定超出允許範圍。
0x82□□014a	eERR_CRD_CMD_INVALID_ANG_ACC	軸群組的旋轉加速度設定超出允許範圍。
0x82□□0154	eERR_CRD_CMD_INVALID_ANG_DEC	軸群組的旋轉減速度設定超出允許範圍。
0x82□□015e	eERR_CRD_CMD_INVALID_ANG_JERK	軸群組的旋轉急跳度 (jerk) 設定超出允許範圍。

軸群組錯誤代碼		
錯誤代碼	錯誤名稱	說明
0x82□□0168	eERR_CRD_CMD_INVALID_ANG_SM_TIME	軸群組的旋轉平滑時間設定超出允許範圍。
0x82□□0190	eERR_CRD_CMD_INVALID_VEL_SCALE	軸群組的速度百分比超出可允許範圍。
0x82□□019a	eERR_CRD_CMD_INVALID_TRANS_VEL	軸群組的過渡速度無效。
0x82□□01a4	eERR_CRD_CMD_INVALID_TRANS_DIS	軸群組的過渡距離無效。
0x82□□01a5	eERR_CRD_CMD_INVALID_TRANS_DEV	軸群組的過渡誤差無效。
0x82□□01a6	eERR_CRD_CMD_INVALID_TRANS_CURVE	軸群組的過渡曲率無效。
0x82□□01b8	eERR_CRD_CMD_TRANS_MODE_UNKNOW N	未知的 Transition 模式。
0x82□□01c2	eERR_CRD_CMD_COORD_SYS_UNKNOWN	未知的座標系統。
0x82□□01cc	eERR_CRD_CMD_BLEND_MODE_UNKNOW N	未知的 Blending 模式。
0x82□□01fe	eERR_CRD_CMD_LIN_INVALID_PARAM	線性路徑規劃的參數無效。
0x82□□0262	eERR_CRD_CMD_CIRC_INVALID_PARAM	圓弧路徑規劃的參數無效。
0x82□□026c	eERR_CRD_CMD_CIRC_INVALID_CENTER	圓弧路徑的中心點太接近起點或終點。
0x82□□0276	eERR_CRD_CMD_CIRC_ANGLE_SMALL	圓弧路徑的中心角度太小。
0x82□□0280	eERR_CRD_CMD_CIRC_INVALID_RADIUS	圓弧路徑的半徑無效。
0x82□□028a	eERR_CRD_CMD_CIRC_INVALID_COORD	圓弧路徑的座標系統無效。
0x82□□02c6	eERR_CRD_CMD_BEZIER_INVALID_PARAM	貝茲路徑規劃的參數無效。
0x82□□02d0	eERR_CRD_CMD_BSPLINE_INVALID_PARAM	BSpline 曲線路徑規劃的參數無效。
0x82□□02da	eERR_CRD_CMD_CURVE_INVALID_STARTP OS	曲線路徑規劃的起始位置無效。
0x82□□02e4	eERR_CRD_CMD_COORD_INVALID_PARAM	座標轉換的參數無效。
0x82□□02ee	eERR_CRD_CMD_NURBS_INVALID_PARAM	NURBS 曲線路徑規劃的參數無效。
0x82□□02f8	eERR_CRD_CMD_LOOKAHEAD_INVALID_PA RAM	軸群組的速度前瞻參數無效。
0x82□□03f2	eERR_CRD_AXIS_ABNORMALLY_DISABLED	軸群組的一軸或多軸不正常解激磁。
0x82□□03fc	eERR_CRD_AXIS_SWL	軸群組的一軸超出軟體極限。

17.2 HIMC_GetLastError

用途

取得控制器的最新錯誤代碼。

語法

```
int HIMC_GetLastError(  
    int ctrl_id,  
    int *p_error_code  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_error_code [out] 指標型態的記憶體，用來儲存控制器的最新錯誤代碼。
 代碼定義請參閱 17.1.1 節。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetLastError
LabVIEW	HIMC Get Last Error.vi
Python	GetLastError

17.3 HIMC_GetAxisLastError

用途

取得軸的最新錯誤代碼。

語法

```
int HIMC_GetAxisLastError(  
    int ctrl_id,  
    int axis_id,  
    int *err_code  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- err_code [out] 指標型態的記憶體，用來儲存軸的最新錯誤代碼。
 代碼定義請參閱 17.1.2 節。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetAxisLastError
LabVIEW	HIMC Get Axis Last Err.vi
Python	GetAxisLastError

17.4 HIMC_ClearAxisLastError

用途

清除軸的最新錯誤代碼。

語法

```
int HIMC_ClearAxisLastError(  
    int ctrl_id,  
    int axis_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ClearAxisLastError
LabVIEW	HIMC Clear Axis Last Err.vi
Python	ClearAxisLastError

17.5 HIMC_GetGrpLastErr

用途

取得軸群組的最新錯誤代碼。

語法

```
int HIMC_GetGrpLastErr(  
    int ctrl_id,  
    int group_id,  
    int *err_code  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- err_code [out] 指標型態的記憶體，用來儲存軸群組的最新錯誤代碼。
 代碼定義請參閱 17.1.3 節。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetGrpLastErr
LabVIEW	HIMC Get Grp Last Err.vi
Python	GetGrpLastErr

17.6 HIMC_ClearGrpLastError

用途

清除軸群組的最新錯誤代碼。

語法

```
int HIMC_ClearGrpLastError(  
    int ctrl_id,  
    int group_id  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_ClearGrpLastError
LabVIEW	HIMC Clear Grp Last Err.vi
Python	ClearGrpLastError

17.7 HIMC_GetDriveErr

用途

取得驅動器的錯誤代碼。

語法

```
int HIMC_GetDriveErr(  
    int ctrl_id,  
    int axis_id,  
    int *err_code  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸群組編號。
- err_code [in] 指標型態的記憶體，用來儲存驅動器的錯誤代碼。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetDriveErr
LabVIEW	HIMC Get Drive Err.vi
Python	GetDriveErr

17.8 HIMC_GetErrorInformation

用途

取得特定錯誤 ID 的資訊。

語法

```
int HIMC_GetErrorInfomation(  
    int error_id,  
    char *p_name,  
    int name_buff_len,  
    int *p_name_actual_len,  
    char *p_description,  
    int description_buff_len,  
    int *p_description_actual_len,  
);
```

參數

error_id [in]	指定一個錯誤 ID。
p_name [out]	指標型態的記憶體，用來儲存特定錯誤 ID 的錯誤名稱。
name_buff_len [in]	指定儲存錯誤名稱的最大字元數。
p_name_actual_len [out]	指標型態的記憶體， 用來儲存錯誤名稱的實際字元數（不包括空字元。）
p_description [out]	指標型態的記憶體，用來儲存特定錯誤 ID 的錯誤描述。
description_buff_len [in]	指定儲存錯誤描述的最大字元數。
p_description_actual_len [out]	指標型態的記憶體， 用來儲存錯誤描述的實際字元數（不包括空字元。）

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetErrorInformation
LabVIEW	HIMC Get Error Information.vi
Python	GetErrorInformation

(此頁有意留白。)

18. 歸原點函式

18.	歸原點函式	18-1
18.1	概述	18-2
18.2	HIMC_MoveHome	18-7
18.3	HIMC_SetHomeMethod	18-8
18.4	HIMC_SetHomeSwitchVel	18-9
18.5	HIMC_SetHomeZeroVel	18-10
18.6	HIMC_SetHomeAcc	18-11
18.7	HIMC_SetHomeOffset	18-12
18.8	HIMC_SetHomeTimeout	18-13
18.9	HIMC_SetHomedStatus	18-14
18.10	HIMC_IsHomed	18-15
18.11	HIMC_IsHoming	18-16
18.12	HIMC_IsHomeSwitch	18-17

18.1 概述

HIMC 支援 CiA 402 歸原點模式，使用者可依據機台配置設定各軸的歸原點方法，表 18.1.1 列出 CiA 402 歸原點模式所定義的歸原點方法，詳細圖示與歸原點流程則如表 18.1.2 所示。

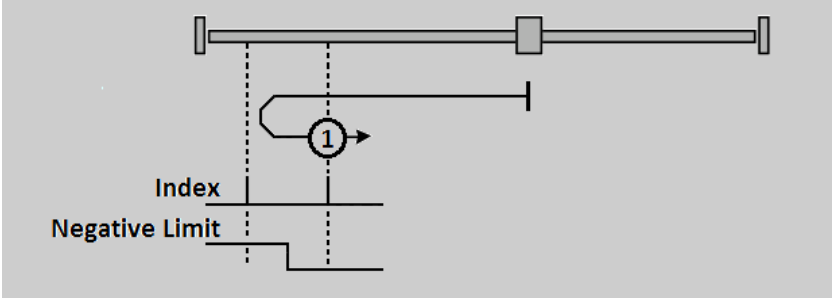
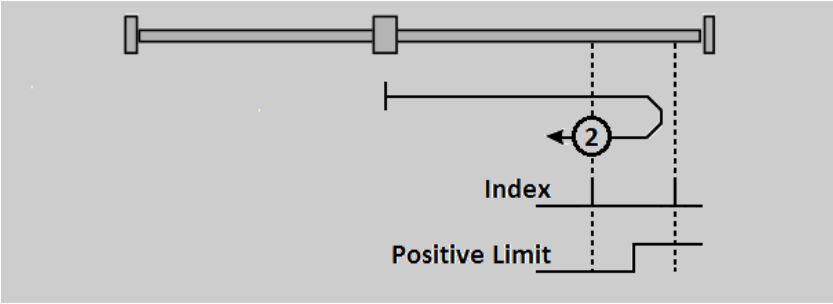
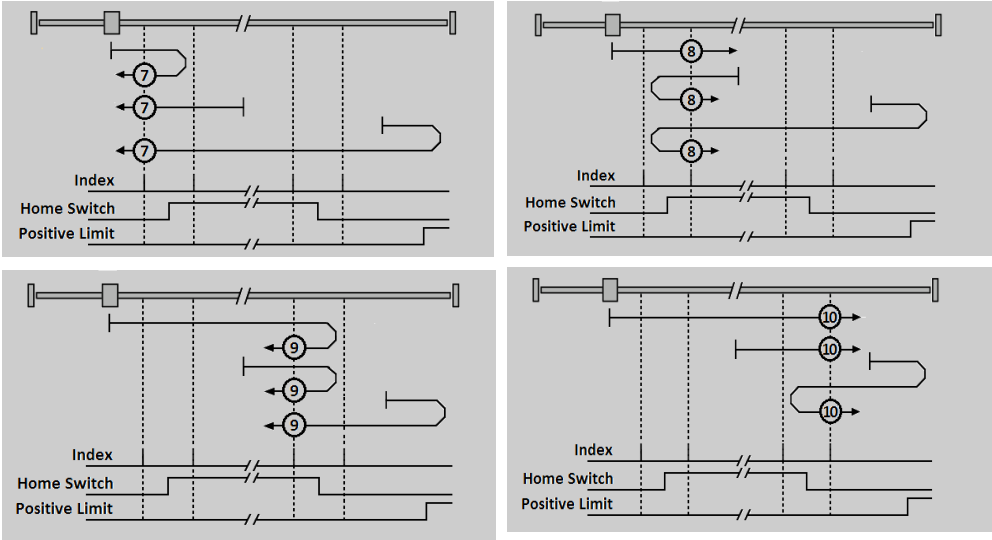
註 1：歸原點方法受驅動器支援的方法限制。

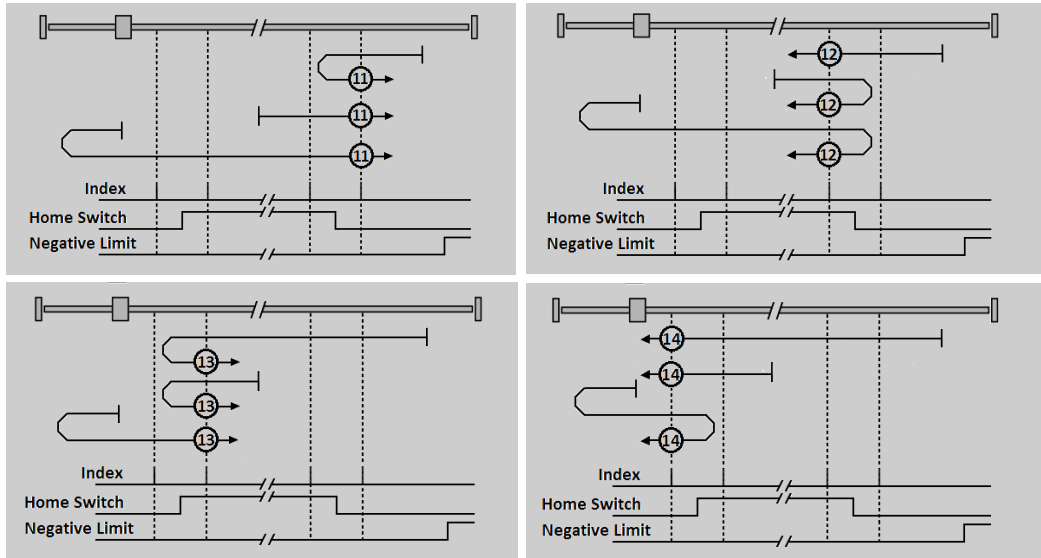
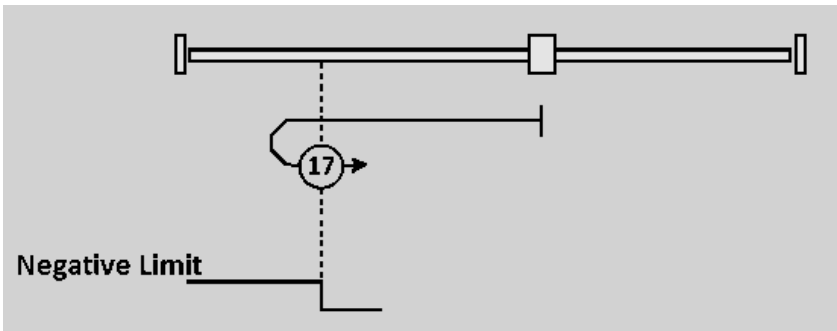
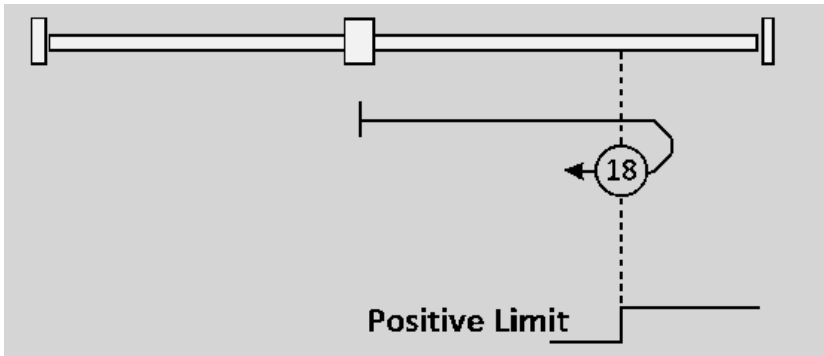
註 2：虛擬機、虛擬軸不支援 HIMC_MoveHome 函式。

表 18.1.1

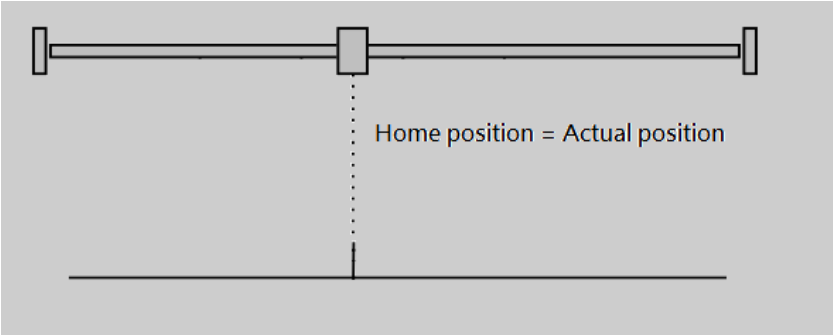
歸原點方法	描述
1	以負極限開關 (negative limit switch) 和 index 脈衝歸原點
2	以正極限開關 (positive limit switch) 和 index 脈衝歸原點
7~10	以原點開關 (home switch) 和 index 脈衝歸原點—正初始方向
11~14	以原點開關 (home switch) 和 index 脈衝歸原點—負初始方向
17	以負極限開關 (negative limit switch) 歸原點
18	以正極限開關 (positive limit switch) 歸原點
23~26	以原點開關 (home switch) 歸原點—正初始方向
27~30	以原點開關 (home switch) 歸原點—負初始方向
33~34	以 index 脈衝歸原點
37	以當前位置歸原點

表 18.1.2

歸原點方法	歸原點流程
1	 若負極限開關失效，運動的初始方向向左。原點位置位於負極限開關失效的右側第一個 index 脈衝處。若無指定負極限，歸原點將失敗。
2	 若正極限開關失效，運動的初始方向向右。原點位置位於正極限開關失效的左側第一個 index 脈衝處。若無指定正極限，歸原點將失敗。
7~10 歸原點方法	 運動的初始方向取決於所尋找的原點開關緣。若一開始原點開關有效，方法7與8的初始方向為負，其他的初始方向為正。若無指定原點開關和正極限，歸原點將失敗。

歸原點方法	歸原點流程
11~14	 運動的初始方向取決於所尋找的原點開關緣。若一開始原點開關有效，方法11與12的初始方向為正，其他的初始方向為負。若無指定原點開關和負極限，歸原點將失敗。
17	 若負極限開關失效，運動的初始方向向左。原點位置位於負極限開關失效的右側。若無指定負極限，歸原點將失敗。
18	 若正極限開關失效，運動的初始方向向右。原點位置位於正極限開關失效的左側。若無指定正極限，歸原點將失敗。

歸原點方法	歸原點流程	
23~26		
	運動的初始方向取決於所尋找的原點開關緣。若一開始原點開關有效，方法 23 與 24 的初始方向為負，其他的初始方向為正。若無指定原點開關和正極限，歸原點將失敗。	
27~30		
	運動的初始方向取決於所尋找的原點開關緣。若一開始原點開關有效，方法27與28的初始方向為正，其他的初始方向為負。若無指定原點開關和負極限，歸原點將失敗。	
33~34		

歸原點方法	歸原點流程
	歸原點方向分別為負 (33) 或正 (34) 。原點位置位於所選方向中的 index 脈衝處。
37	
	馬達的當前位置被定義為原點位置。在此方法中，驅動器不需要處於Operation enabled狀態。 6062h (position demand value) = 6064h (position actual value) = 607Ch (home offset) 6063h (position actual internal value) = 60FCh (position demand internal value) = 0

18.2 HIMC_MoveHome

用途

執行軸的歸原點程序。

語法

```
int HIMC_MoveHome(  
    int ctrl_id,  
    int axis_id  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

備註

使用此函式需將物件 0x6060 (Mode of operation) 、0x6061 (Mode of operation display) 配置為 PDO。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_MoveHome
LabVIEW	HIMC Move Home.vi
Python	MoveHome

18.3 HIMC_SetHomeMethod

用途

設置歸原點程序的歸原點方法。

語法

```
int HIMC_SetHomeMethod(  
    int ctrl_id,  
    int axis_id,  
    int method  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- method [in] 歸原點方法，請參閱表 18.1.1 與表 18.1.2。預設值為 33。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetHomeMethod
LabVIEW	HIMC Set Home Method.vi
Python	SetHomeMethod

18.4 HIMC_SetHomeSwitchVel

用途

設置歸原點程序的快速歸原點速度。

語法

```
int HIMC_SetHomeSwitchVel(  
    int    ctrl_id,  
    int    axis_id,  
    double fast_vel  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- fast_vel [in] 快速歸原點速度，預設值為 20。
 參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetHomeSwitchVel
LabVIEW	HIMC Set Home Switch Vel.vi
Python	SetHomeSwitchVel

18.5 HIMC_SetHomeZeroVel

用途

設置歸原點程序的慢速歸原點速度。

語法

```
int HIMC_SetHomeZeroVel(  
    int    ctrl_id,  
    int    axis_id,  
    double slow_vel  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- slow_vel [in] 慢速歸原點速度，預設值為 5。
 參數單位：mm/s (毫米/秒) 或 deg/s (角度/秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetHomeZeroVel
LabVIEW	HIMC Set Home Zero Vel.vi
Python	SetHomeZeroVel

18.6 HIMC_SetHomeAcc

用途

設置歸原點程序的歸原點加速度。

語法

```
int HIMC_SetHomeAcc(
    int    ctrl_id,
    int    axis_id,
    double acc
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- acc [in] 歸原點加速度，預設值為 2000。
 參數單位：mm/s² (毫米/秒²) 或 deg/s² (角度/秒²)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetHomeAcc
LabVIEW	HIMC Set Home Acc.vi
Python	SetHomeAcc

18.7 HIMC_SetHomeOffset

用途

設置歸原點程序的原點偏移量。

語法

```
int HIMC_SetHomeOffset(  
    int    ctrl_id,  
    int    axis_id,  
    double offset  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- offset [in] 原點偏移量，預設值為 0。
 參數單位：mm（毫米）或 deg（角度）

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetHomeOffset
LabVIEW	HIMC Set Home Offset.vi
Python	SetHomeOffset

18.8 HIMC_SetHomeTimeout

用途

設置歸原點程序的逾時時間。

語法

```
int HIMC_SetHomeTimeout(  
    int ctrl_id,  
    int axis_id,  
    int timeout  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- timeout [in] 逾時時間，預設值為 120,000。
 參數單位：ms (毫秒)

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetHomeTimeout
LabVIEW	HIMC Set Home Timeout.vi
Python	SetHomeTimeout

18.9 HIMC_SetHomedStatus

用途

設置軸的歸原點狀態。

語法

```
int HIMC_SetHomedStatus(  
    int ctrl_id,  
    int axis_id,  
    int is_homed  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- is_homed [in] 設為 1：歸原點已完成；設為 0：歸原點未完成（預設值）。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_SetHomedStatus
LabVIEW	-
Python	-

18.10 HIMC_IsHomed

用途

詢問軸是否已完成歸原點程序。

語法

```
int HIMC_IsHomed(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_homed  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_homed [out] 指標型態的記憶體，用來儲存軸是否已完成歸原點程序。
 若軸已完成歸原點程序，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsHomed
LabVIEW	HIMC Is Homed.vi
Python	IsHomed

18.11 HIMC_IsHoming

用途

詢問軸是否正在執行歸原點程序。

語法

```
int HIMC_IsHoming(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_homing  
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。

axis_id [in] 軸編號。

p_is_homing [out] 指標型態的記憶體，用來儲存軸是否正在執行歸原點程序。
 若軸正在執行歸原點程序，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsHoming
LabVIEW	HIMC Is Homing.vi
Python	IsHoming

18.12 HIMC_IsHomeSwitch

用途

取得軸的近原點開關狀態。

語法

```
int HIMC_IsHomeSwitch(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_home_switch  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_home_switch [out] 指標型態的記憶體，用來儲存軸的近原點開關狀態。
 若軸的近原點開關為觸發狀態，其值為 1，否則為 0。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_IsHomeSwitch
LabVIEW	-
Python	IsHomeSwitch

(此頁有意留白。)

19. 從站函式

19.	從站函式	19-1
19.1	HIMC_GetSlvChNum	19-2
19.2	HIMC_GetSlvSlotNum	19-3

19.1 HIMC_GetSlvChNum

用途

依照通道類型取得從站對應的通道數量。

語法

```
int HIMC_GetSlvChNum(
    int ctrl_id,
    int slv_slot_id,
    int ch_type,
    int *p_ch_num
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_slot_id [in] 從站 ID 與其插槽 ID，若從站無插槽則可忽略插槽 ID。
- ch_type [in] 通道類型。
 1：通用輸入、2：通用輸出、3：類比輸入、4：類比輸出
- p_ch_num [out] 指標型態的記憶體，用來儲存從站的通道數量。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvChNum
LabVIEW	-
Python	GetSlvChNum

19.2 HIMC_GetSlvSlotNum

用途

取得從站的插槽數量。

語法

```
int HIMC_GetSlvSlotNum(  
    int ctrl_id,  
    int slv_id,  
    int *p_slot_num  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slv_id [in] 從站 ID。
- p_slot_num [out] 指標型態的記憶體，用來儲存從站的插槽數量。

回傳值

若函式執行成功，將回傳 **int** 型態的值 **0**。若失敗，則回傳**非零值**。

需求版本與檔案

最低支援版本	iA Studio 3.3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	HIMC_GetSlvSlotNum
LabVIEW	-
Python	GetSlvSlotNum

(此頁有意留白。)

20. 數據結構

20.	數據結構	20-1
20.1	ComInfo	20-2
20.2	CoordPosition.....	20-4
20.3	MotionProfile.....	20-5
20.4	CenterPosition.....	20-7
20.5	NormalVector	20-8
20.6	TransPrm.....	20-9
20.7	PosTriggerPar	20-10
20.8	Data_t.....	20-11

20.1 ComInfo

用途

定義連線類型和資訊。

語法

```
typedef struct {
    ComType type;
    struct {
        char ip[20];
        char port[12];
    } TCP_IP;

    struct {
        char com_port_name[80];
        int baud_rate;
    } RS232;

    struct {
        char autoExecExe;
    } Simulator;
} ComInfo;
```

成員

- ComType 連線類型。
- TCP_IP 含網絡連線參數、IP 與端口的結構。
- RS232 含 RS232 連線參數、序列埠名稱與波特率的結構。
- Simulator 含 Simulator 連線參數與 autoExecExe 的結構。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	ComInfo
LabVIEW	-
Python	ComInfo

20.2 CoordPosition

用途

定義座標運動的位置或距離。

語法

```
typedef struct {  
    double x_pos;  
    double y_pos;  
    double z_pos;  
    double a_pos;  
    double b_pos;  
    double c_pos;  
} CoordPosition, *PCoordPosition;
```

成員

x_pos 終點的線性位置 X。單位：mm (毫米)

y_pos 終點的線性位置 Y。單位：mm (毫米)

z_pos 終點的線性位置 Z。單位：mm (毫米)

a_pos 終點的定向角 A。單位：deg (角度)

b_pos 終點的定向角 B。單位：deg (角度)

c_pos 終點的定向角 C。單位：deg (角度)

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	CoordPosition
LabVIEW	-
Python	CoordPosition

20.3 MotionProfile

用途

定義運動軌跡設定。

語法

```
typedef struct {
    double max_vel;
    double max_acc;
    double max_dec;
    double smooth_time;
} MotionProfile, *PMotionProfile;
```

成員

max_vel	線性運動的最大切向速度。 單位：mm/s (毫米/秒) 或 deg/s (角度/秒) 範圍：0 ~ 5000
max_acc	線性運動的最大切向加速度。 單位：mm/s ² (毫米/秒 ²) 或 deg/s ² (角度/秒 ²) 範圍：>0 ~ 50000 (加速度不能為 0)
max_dec	線性運動的最大切向減速度。 單位：mm/s ² (毫米/秒 ²) 或 deg/s ² (角度/秒 ²) 範圍：>0 ~ 50000 (減速度不能為 0)
smooth_time	線性運動的平滑時間。 單位：ms (毫秒) 範圍：0 ~ 500

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	MotionProfile
LabVIEW	-
Python	MotionProfile

20.4 CenterPosition

用途

定義中心點設定。

語法

```
typedef struct {
    double x_pos;
    double y_pos;
    double z_pos;
} CenterPosition, *PCenterPosition;
```

成員

- x_pos X 軸位置。單位：mm (毫米)
- y_pos Y 軸位置。單位：mm (毫米)
- z_pos Z 軸位置。單位：mm (毫米)

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	CenterPosition
LabVIEW	-
Python	CenterPosition

20.5 NormalVector

用途

定義法向量設定。

語法

```
typedef struct {  
    double x_vector;  
    double y_vector;  
    double z_vector;  
} NormalVector, *PNormalVector;
```

成員

x_vector X 方向向量。

y_vector Y 方向向量。

z_vector Z 方向向量。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	NormalVector
LabVIEW	-
Python	NormalVector

20.6 TransPrm

用途

定義位置觸發設定。

語法

```
typedef struct {
    double trans_vel,
    double trans_dis,
    double trans_dev;
    double trans_curv;
} TransPrm, *PTransPrm;
```

成員

- trans_vel

軸群組新的路徑過渡模式速度參數。
單位：mm/s (毫米/秒)
- trans_dis

軸群組新的路徑過渡模式距離參數。
單位：mm (毫米)
- trans_dev

軸群組新的路徑過渡模式最大誤差參數。
單位：mm (毫米)
- trans_curv

軸群組新的路徑過渡模式最大曲率參數。
單位：mm⁻¹ (毫米⁻¹)

需求版本與檔案

最低支援版本	iA Studio 3.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	TransPrm
LabVIEW	-
Python	TransPrm

20.7 PosTriggerPar

用途

定義位置觸發設定。

語法

```
typedef struct {  
    double start_pos;  
    double end_pos;  
    double interval;  
    int pulse_width;  
} PosTriggerPar, *PPosTriggerPar;
```

成員

start_pos	PT 功能的起點。 單位：mm (毫米) 或 deg (角度)
end_pos	PT 功能的終點。 單位：mm (毫米) 或 deg (角度)
interval	連續 PT 輸出的位置間距。 單位：mm (毫米) 或 deg (角度)
pulse_width	每個 PT 輸出信號的寬度。 單位：ns (奈秒)

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	PosTriggerPar
LabVIEW	-
Python	PosTriggerPar

20.8 Data_t

用途

定義 CoE 物件之數值型態。

語法

```
typedef union {  
    unsigned char bytes[65];  
    unsigned char uint8_;  
    unsigned short uint16_;  
    unsigned int uint32_;  
    unsigned uint64_t uint64_;  
    signed char int8_;  
    short int16_;  
    int int32_;  
    int64_t int64_;  
    float float_;  
    double double_;  
} Data_t;
```

成員

bytes	型態為字串的值。(註：不支援 C#)
uint8_	型態為無符號字元的值。
uint16_	型態為無符號短整數的值。
uint32_	型態為無符號整數的值。
uint64_	型態為無符號長整數的值。
int8_	型態為字元的值。
int16_	型態為短整數的值。
int32_	型態為整數的值。
int64_	型態為長整數的值。
float_	型態為浮點數的值。
double_	型態為雙精度浮點數的值。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	Data_t
LabVIEW	-
Python	Data_t

21. 列舉

21.	列舉	21-1
21.1	ComType.....	21-2
21.2	CoordSystem	21-3
21.3	MotionBufferMode	21-5
21.4	MotionTransitionMode	21-6
21.5	ShaperMode.....	21-7

21.1 ComType

定義

連線類型列舉。

語法

```
typedef enum {  
    COM_TYPE_TCPIP,  
    COM_TYPE_RS232,  
    COM_TYPE_SIMULATOR  
} ComType;
```

成員

COM_TYPE_TCPIP 連線類型為 TCPIP。

COM_TYPE_RS232 連線類型為 RS232。

COM_TYPE_SIMULATOR 連線類型為虛擬機。

需求版本與檔案

最低支援版本	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	ComType
LabVIEW	-
Python	ComType

21.2 CoordSystem

定義

座標系統。

語法

```
typedef enum {  
    kCoord_ACS = 0,  
    kCoord_MCS = 1,  
    kCoord_PCS = 2,  
    kCoord_GLOBAL = 3,  
    kCoord_WCS1 = 1 << 8,  
    kCoord_WCS2 = 2 << 8,  
    kCoord_WCS3 = 3 << 8,  
    kCoord_WCS4 = 4 << 8,  
    kCoord_WCS5 = 5 << 8,  
    kCoord_WCS6 = 6 << 8,  
    kCoord_WCS7 = 7 << 8,  
    kCoord_WCS8 = 8 << 8,  
    kCoord_WCS9 = 9 << 8,  
    kCoord_WCS10 = 10 << 8,  
    kCoord_WCS11 = 11 << 8,  
    kCoord_WCS12 = 12 << 8,  
    kCoord_WCS13 = 13 << 8,  
    kCoord_WCS14 = 14 << 8,  
    kCoord_WCS15 = 15 << 8,  
    kCoord_OFFSET = 1 << 15  
} CoordSystem;
```

說明

請參閱 6.1.2 節。

需求版本與檔案

最低支援版本	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	CoordSystem
LabVIEW	-
Python	CoordSystem

21.3 MotionBufferMode

定義

相鄰協調運動段之間的緩衝模式。

語法

```
typedef enum {
    kBM_Aborting = 0,
    kBM_Buffered = 1,
    kBM_BlendingLow = 2,
    kBM_BlendingPrevious = 3,
    kBM_BlendingNext = 4,
    kBM_BlendingHigh = 5,
    kBM_LookAhead = 6
} MotionBufferMode;
```

說明

請參閱 6.1.4 節。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	MotionBufferMode
LabVIEW	-
Python	MotionBufferMode

21.4 MotionTransitionMode

定義

相鄰協調運動段之間的過渡模式。

語法

```
typedef enum {
    kTM_NONE = 0,
    kTM_StartVelocity = 1,
    kTM_ConstantVelocity = 2,
    kTM_CornerDistance = 3,
    kTM_MaxCornerDeviation = 4,
    kTM_PLCOpenReserved_05 = 5,
    kTM_PLCOpenReserved_06 = 6,
    kTM_PLCOpenReserved_07 = 7,
    kTM_PLCOpenReserved_08 = 8,
    kTM_PLCOpenReserved_09 = 9,
    kTM_PLCOpenReserved_10 = 10,
    kTM_MaxCornerCurvature = 11
} MotionTransitionMode;
```

說明

請參閱 6.1.5 節。

需求版本與檔案

最低支援版本	iA Studio 3.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	MotionTransitionMode
LabVIEW	-
Python	MotionTransitionMode

21.5 ShaperMode

定義

InShape 的濾波器模式。

語法

```
typedef enum {  
    Shaper_Normal = 0,  
    Shaper_Robust  
} ShaperMode;
```

成員

- Shaper_Normal 普通的 InShape。
- Shaper_Robust 強化的 InShape。

需求版本與檔案

最低支援版本	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

其他 API 環境的對應名稱

C#	ShaperMode
LabVIEW	-
Python	ShaperMode

(此頁有意留白。)

22. 附錄

22.	附錄.....	22-1
22.1	API 錯誤代碼.....	22-2

22.1 API 錯誤代碼

以下錯誤代碼會出現在以 API 存取控制器發生錯誤時。

表 22.1.1

API 錯誤代碼		
錯誤代碼	錯誤名稱	說明
0x01000000	eERR_API_COMM_ERR	與控制器通訊時發生錯誤。
0x0100000a	eERR_API_CONNECT_FAIL	無法連線至控制器。
0x01000014	eERR_API_TOUT	操作因逾時被退回。
0x0100001e	eERR_API_ACCESS_REJECT	要求被拒。
0x01000028	eERR_API_FIFO_MISMATCH	API 錯誤。
0x01000032	eERR_API_FIFO_FULL	網路忙碌。
0x0100003c	eERR_API_HIMC_NOT_READY	HIMC 尚未備妥。
0x01000046	eERR_API_PROTOCOL_MISMATCH	API 錯誤。
0x01000050	eERR_API_INPUT_ARG_ERR	引數無效。
0x0100005a	eERR_API_NOT_SUPPORT	此版本並未支援此 API。
0x01000064	eERR_API_BUSY	API 忙碌。
0x0100006e	eERR_API_FILE_TRANS_FAIL	檔案傳輸失敗。
0x01000073	eERR_API_ZIP_CORRUPT	壓縮歸檔文件已損壞。
0x01000078	eERR_API_ID_NOT_FOUND	找不到此連線 ID，請確認是否連線。
0x01000082	eERR_API_SLV_DB_NOT_READY	從站尚未備妥。
0x0100008c	eERR_API_SLV_ID_INVALID	從站 ID 無效。
0x01000096	eERR_API_INVALID_VAR_ID	變數 ID 無效。
0x010000a0	eERR_API_VAR_VAL_OUT_OF_RANGE	數值超出範圍。
0x010000a5	eERR_API_VAR_IS_READ_ONLY	變數只能讀取。
0x010000aa	eERR_API_FS_ACCESS_DENIED	無法存取檔案系統，請檢查您的存取權限。
0x010000b4	eERR_API_TASK_ID_INVALID	此 task ID 無效。
0x010000be	eERR_API_TASK_EMPTY	此 task 沒有內容。
0x010000c3	eERR_API_TASK_FUNC_NOT_FOUND	在 task 內無法找到此函式。
0x010000c8	eERR_API_TASK_NOT_RUNNING	此 task 未在執行。
0x010000d2	eERR_API_TASK_IS_RUNNING	此 task 正在執行中。
0x010000d7	eERR_API_TOO_MANY_BRK_POINT	此 task 有太多中斷點。
0x010000dc	eERR_API_INVALID_ERROR_ID	無效的錯誤代碼。
0x010000e6	eERR_API_INSUFFICIENT_BUFFER	緩存空間不足。
0x010000f0	eERR_API_STR_TOO_LONG	字串長度超過限制。
0x010000fa	eERR_API_HIMC_VERSION_MISMATCH	API 與控制器版本不相容。
0x010003e8	eERR_API_MOTION_ERROR	運動控制錯誤，請檢查錯誤 log。

API 錯誤代碼		
錯誤代碼	錯誤名稱	說明
0x010087d0	eMSG_API_PACKET_SYS_CALL	API 系統呼叫訊息。
0x010047da	eWRN_API_PACKET_BUSY	API 封包忙碌。
0x010047e4	eWRN_API_PACKET_INVALID_SYS_CALL_ID	API 系統呼叫 ID 無效。
0x01008bb8	eMSG_API_CONNECT	API 用戶端已連線。
0x01008bb9	eMSG_API_DISCONNECT	API 用戶端已斷線。
0x01008bba	eMSG_API_VERSION	API 用戶端版本。
0x0100270f	eERR_API_FATAL	API 運行中錯誤。
0x01003fff	eERR_API_LOG	API 錯誤。
0x01007fff	eWRN_API_LOG	API 警告。
0x0100bfff	eMSG_API_LOG	API 訊息。
0x0100ffff	eDBG_API_LOG	API 偵錯。