

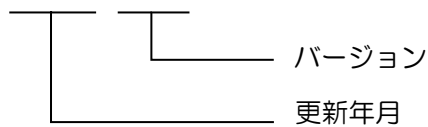
# HIMC

## API ユーザーマニュアル

## 改訂履歴

バージョンは、表紙の下部にも記載されています。

MH05UJ01-2303\_V0.5



日付	バージョン	適用機種	改定内容
2022 年 6 月 30 日	0.5	iA Studio 2.0	(1) 章・セクションの追加： 8 章 - AIO 関数 (2) API の追加： ■ 2 章 HIMC_SetEconMode HIMC_IsEconMode ■ 3 章 HIMC_MoveTrq HIMC_MovePVT HIMC_IsAcc ■ 6 章 HIMC_ArcAngle2D HIMC_SetGrpAngMotionProfile HIMC_GetGrpCoordTrans HIMC_SetGrpCoordTrans HIMC_GetGrpPoseCmd HIMC_GetGrpPoseFb HIMC_CircRel ■ 7 章 HIMC_SetGPIInvert HIMC_SetGPOInvert HIMC_BindEMO HIMC_GetAllGPIInvertSt HIMC_GetAllGPOInvertSt ■ 10 章 HIMC_SetPT_PosArray HIMC_SetPT_StateArray HIMC_SetPT_StartIndex HIMC_SetPT_EndIndex ■ 12 章 HIMC_SetCompAlgType ■ 14 章 HIMC_LoadHMPLTask ■ 15 章 HIMC_SetHmplMsgEvtCallback (3) 変数を追加： セクション 3.1.1、セクション 6.1.1 (4) エラーメッセージの追加： セクション 17.1.1、セクション 17.1.2、 セクション 17.1.3
Sep. 15 <sup>th</sup> , 2021	0.4	iA Studio 1.4	(1) 章・セクションの追加： セクション 7.1.1 - GPIO 変数 (2) API の追加： ■ 2 章 HIMC_GetFirmwareVer ■ 3 章 HIMC_GetVelFb HIMC_GetVelErr HIMC_GetCurrFb

日付	バージョン	適用機種	改定内容
			HIMC_SetVelScale HIMC_GetVelScale HIMC_SetRollover HIMC_GetRolloverTurns HIMC_IsDriveErr HIMC_IsPosErr ■ 6 章 HIMC_JogGroup  HIMC_JogGroupAxis HIMC_SetGrpVelScale HIMC_GetGrpVelScale ■ 11 章 HIMC_SetupComp3D (3) 変数を追加： セクション 3.1.1、セクション 6.1.1、セクション 15.1.2 (4) エラーメッセージの追加： セクション 16.1.1、セクション 16.1.2、セクション 16.1.3、セクション 19.1
2020 年 6 月 30 日	0.3	iA Studio 1.3	(1) C#、LabVIEW、Python の対応する関数名を追加 (2) 付録の追加： ■ セクション 19.1：API エラーコード (3) API の修正： ■ セクション 6.2 HIMC_GroupReset → HIMC_ResetGroup ■ 17 章 PosTriggerPar: Remove parameter “polarity” (4) API の追加： ■ セクション 3.2 HIMC_Halt HIMC_Resume ■ セクション 3.3 HIMC_SetAccTime HIMC_SetDecTime HIMC_GetKillDec HIMC_SetKillDec HIMC_IgnorePE ■ 4 章 HIMC_GetGearRatio HIMC_IsInGear HIMC_IsGearMaster HIMC_IsGearSlave ■ 5 章 HIMC_GetGantryPairID HIMC_IsGantryPair ■ セクション 6.2 HIMC_HaltGroup HIMC_ResumeGroup HIMC_ArcCW2D  HIMC_ArcCCW2D ■ セクション 6.3 HIMC_GetGrpKin HIMC_GetGrpMaxVel

日付	バージョン	適用機種	改定内容
			HIMC_SetGrpVel HIMC_GetGrpMaxAcc HIMC_SetGrpAcc HIMC_SetGrpAccTime HIMC_GetGrpMaxDec HIMC_SetGrpDec HIMC_SetGrpDecTime HIMC_GetGrpSMTIME HIMC_SetGrpSMTIME HIMC_GetGrpCoordSys HIMC_SetGrpCoordSys HIMC_GetGrpBufferMode HIMC_SetGrpBufferMode HIMC_GetGrpTransMode HIMC_SetGrpTransMode HIMC_SetGrpTransPrm HIMC_GetGrpCmdNum ■ セクション 6.4 HIMC_IsGrpErrorStop ■ 11 章 HIMC_GetCompPos ■ セクション 15.3 HIMC_RunSlvPdIFunc HIMC_StopSlvPdIFunc HIMC_IsSlvPdIFuncRunning
2019 年 4 月 2 日	0.2	ia Studio 1.1.3772.0	(1) 章の配置を修正 (2) 「シンクロモーション機能」「フィルター機能」の章を追加 (3) API の名前を変更： ■ セクション 3.2 HIMC_GetSwPositiveLimitPos → HIMC_GetSWRL HIMC_SetSwPositiveLimitPos → HIMC_SetSWRL HIMC_GetSwNegativeLimitPos → HIMC_GetSWLL HIMC_SetSwNegativeLimitPos → HIMC_SetSWLL ■ 13 章 HIMC_StartHMPLTask → HIMC_StartTask HIMC_StopHMPLTask → HIMC_StopTask (4) API 変数の修正： ■ セクション 6.1.4 HIMC_LinAbs HIMC_LinRel HIMC_CircAbs ■ 9 章 HIMC_SetPosTriggerConfig (5) API の追加： ■ 2 章 HIMC_ConnectToSimulator HIMC_ConnectToEthernet HIMC_IsSystemOper HIMC_IsSystemPreOp HIMC_IsSystemError HIMC_StopAll HIMC_RescanMoE ■ セクション 3.2 HIMC_GetPosOffset HIMC_GetPosOut

日付	バージョン	適用機種	改定内容
			HIMC_GetVelOut HIMC_GetAccOut HIMC_IgnoreHWL HIMC_IgnoreSWL HIMC_GetAxisNum ■ セクション 6.2 HIMC_RemoveAxisFromGrp HIMC_SetGrpKin HIMC_Reset ■ 7 章 HIMC_ToggleHimcGpo HIMC_ToggleSlaveGpo ■ 8 章 HIMC_SetTableValue HIMC_GetTableValue ■ 11 章 HIMC_SetupComp2D ■ 13 章 HIMC_StartTaskFunc HIMC_StopAllTask HIMC_IsTaskStop ■ 15 章 HIMC_GetSlvVar HIMC_SetSlvVar HIMC_GetSlvSt HIMC_SetSlvSt ■ 16 章 HIMC_GetAxisLastErr HIMC_ClearAxisLastErr HIMC_GetGrpLastErr HIMC_ClearGrpLastErr (6) 列挙の追加 ■ 18 章 ShaperMode
2018 年 4 月 17 日	0.1	iA Studio 1.0.2461.0	初版

# 目次

1.	はじめに .....	1-1
1.1	API の仕組み .....	1-2
1.2	API 環境のセットアップ .....	1-2
1.2.1	C / C++ .....	1-2
1.2.2	C# .....	1-2
1.2.3	LabView .....	1-2
1.2.4	Python .....	1-2
1.3	法的免責事項 .....	1-3
2.	HIMC システム機能 .....	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_IsSystemOper .....	2-7
2.7	HIMC_IsSystemPreOp .....	2-8
2.8	HIMC_IsSystemError .....	2-9
2.9	HIMC_DisableAll .....	2-10
2.10	HIMC_StopAll .....	2-11
2.11	HIMC_EStop .....	2-12
2.12	HIMC_GetSlaveNum .....	2-13
2.13	HIMC_GetFirmwareVer .....	2-14
2.14	HIMC_RescanMoE .....	2-15
2.15	HIMC_SetEconMode .....	2-16
2.16	HIMC_IsEconMode .....	2-17
3.	軸機能 .....	3-1
3.1	概要 .....	3-3
3.1.1	軸変数 .....	3-6
3.2	軸モーション制御 .....	3-9
3.2.1	HIMC_Enable .....	3-9
3.2.2	HIMC_Disable .....	3-10
3.2.3	HIMC_Reset .....	3-11
3.2.4	HIMC_MoveAbs .....	3-12
3.2.5	HIMC_MoveRel .....	3-13
3.2.6	HIMC_MoveVel .....	3-14
3.2.7	HIMC_MoveTrq .....	3-15
3.2.8	HIMC_MovePVT .....	3-16
3.2.9	HIMC_Stop .....	3-18
3.2.10	HIMC_Halt .....	3-19
3.2.11	HIMC_Resume .....	3-20
3.3	軸設定 .....	3-21
3.3.1	HIMC_GetMaxVel .....	3-21
3.3.2	HIMC_SetVel .....	3-22
3.3.3	HIMC_GetMaxAcc .....	3-23
3.3.4	HIMC_SetAcc .....	3-24
3.3.5	HIMC_SetAccTime .....	3-25
3.3.6	HIMC_GetMaxDec .....	3-26
3.3.7	HIMC_SetDec .....	3-27

3.3.8	HIMC_SetDecTime.....	3-28
3.3.9	HIMC_GetKillDec.....	3-29
3.3.10	HIMC_SetKillDec.....	3-30
3.3.11	HIMC_GetSWRL.....	3-31
3.3.12	HIMC_SetSWRL.....	3-32
3.3.13	HIMC_GetSWLL.....	3-33
3.3.14	HIMC_SetSWLL.....	3-34
3.3.15	HIMC_GetSMTIME.....	3-35
3.3.16	HIMC_SetSMTIME.....	3-36
3.3.17	HIMC_GetMoveTime.....	3-37
3.3.18	HIMC_GetSettlingTime.....	3-38
3.3.19	HIMC_SetPos.....	3-39
3.3.20	HIMC_GetPosFb.....	3-40
3.3.21	HIMC_GetPosOffset.....	3-41
3.3.22	HIMC_GetPosErr.....	3-42
3.3.23	HIMC_GetVelFb.....	3-43
3.3.24	HIMC_GetVelErr.....	3-44
3.3.25	HIMC_GetCurrFb.....	3-45
3.3.26	HIMC_GetRefPos.....	3-46
3.3.27	HIMC_GetRefVel.....	3-47
3.3.28	HIMC_GetRefAcc.....	3-48
3.3.29	HIMC_GetPosOut.....	3-49
3.3.30	HIMC_GetVelOut.....	3-50
3.3.31	HIMC_GetAccOut.....	3-51
3.3.32	HIMC_IgnoreHWL.....	3-52
3.3.33	HIMC_IgnoreSWL.....	3-53
3.3.34	HIMC_IgnorePE.....	3-54
3.3.35	HIMC_GetAxisNum.....	3-55
3.3.36	HIMC_SetVelScale.....	3-56
3.3.37	HIMC_GetVelScale.....	3-57
3.3.38	HIMC_SetRollover.....	3-58
3.3.39	HIMC_GetRolloverTurns.....	3-59
3.3.40	HIMC_GetMultiAxesFeedbackPos.....	3-60
3.4	軸の状態.....	3-61
3.4.1	HIMC_IsEnabled.....	3-61
3.4.2	HIMC_IsMoving.....	3-62
3.4.3	HIMC_IsInPos.....	3-63
3.4.4	HIMC_IsErrorStop.....	3-64
3.4.5	HIMC_IsGantry.....	3-65
3.4.6	HIMC_IsGrouped.....	3-66
3.4.7	HIMC_IsSync.....	3-67
3.4.8	HIMC_IsHWLL.....	3-68
3.4.9	HIMC_IsHWRL.....	3-69
3.4.10	HIMC_IsSWLL.....	3-70
3.4.11	HIMC_IsSWRL.....	3-71
3.4.12	HIMC_IsDriveErr.....	3-72
3.4.13	HIMC_IsPosErr.....	3-73
3.4.14	HIMC_IsCompActive.....	3-74
3.4.15	HIMC_IsAcc.....	3-75
4.	シンクロモーション機能.....	4-1
4.1	概要.....	4-2
4.1.1	シンクロモーション変数.....	4-2

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.	ガントリ機能 .....	5-1
5.1	概要 .....	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.	グループ機能 .....	6-1
6.1	概要 .....	6-3
6.1.1	グループ変数.....	6-6
6.1.2	座標系 .....	6-9
6.1.3	運動学 .....	6-13
6.1.4	バッファモード .....	6-13
6.1.5	Transition モード .....	6-15
6.2	グループモーションコントロール.....	6-17
6.2.1	HIMC_EnableGroup.....	6-17
6.2.2	HIMC_DisableGroup .....	6-18
6.2.3	HIMC_ResetGroup.....	6-19
6.2.4	HIMC_StopGroup .....	6-20
6.2.5	HIMC_HaltGroup .....	6-21
6.2.6	HIMC_ResumeGroup.....	6-22
6.2.7	HIMC_JogGroup.....	6-23
6.2.8	HIMC_JogGroupAxis .....	6-24
6.2.9	HIMC_LineAbs2D.....	6-25
6.2.10	HIMC_LineAbs3D .....	6-26
6.2.11	HIMC_LineRel2D.....	6-27
6.2.12	HIMC_LineRel3D .....	6-28
6.2.13	HIMC_Arc2D .....	6-29
6.2.14	HIMC_ArcCW2D.....	6-31
6.2.15	HIMC_ArcCCW2D.....	6-32
6.2.16	HIMC_Circle2D .....	6-33
6.2.17	HIMC_ArcAngle2D .....	6-35
6.3	グループ設定 .....	6-37
6.3.1	HIMC_AddAxesToGrp.....	6-37
6.3.2	HIMC_RemoveAxisFromGrp .....	6-38
6.3.3	HIMC_SetupGroup .....	6-39
6.3.4	HIMC_UngrpAllAxes .....	6-40
6.3.5	HIMC_GetGroupID.....	6-41
6.3.6	HIMC_SetGrpMotionProfile.....	6-42
6.3.7	HIMC_SetGrpAngMotionProfile.....	6-44
6.3.8	HIMC_GetGrpKin.....	6-46
6.3.9	HIMC_SetGrpKin .....	6-47
6.3.10	HIMC_GetGrpMaxVel .....	6-48
6.3.11	HIMC_SetGrpVel .....	6-49
6.3.12	HIMC_GetGrpMaxAcc .....	6-50



6.3.13	HIMC_SetGrpAcc .....	6-51
6.3.14	HIMC_SetGrpAccTime.....	6-52
6.3.15	HIMC_GetGrpMaxDec.....	6-53
6.3.16	HIMC_SetGrpDec.....	6-54
6.3.17	HIMC_SetGrpDecTime .....	6-55
6.3.18	HIMC_GetGrpSMTime.....	6-56
6.3.19	HIMC_SetGrpSMTime .....	6-57
6.3.20	HIMC_GetGrpCoordSys .....	6-58
6.3.21	HIMC_SetGrpCoordSys .....	6-59
6.3.22	HIMC_GetGrpBufferMode .....	6-60
6.3.23	HIMC_SetGrpBufferMode .....	6-61
6.3.24	HIMC_GetGrpTransMode .....	6-62
6.3.25	HIMC_SetGrpTransMode .....	6-63
6.3.26	HIMC_SetGrpTransPrm.....	6-64
6.3.27	HIMC_GetGrpCmdNum.....	6-65
6.3.28	HIMC_SetGrpVelScale .....	6-66
6.3.29	HIMC_GetGrpVelScale .....	6-67
6.3.30	HIMC_GetGrpCoordTrans.....	6-68
6.3.31	HIMC_SetGrpCoordTrans .....	6-69
6.3.32	HIMC_GetGrpPoseCmd .....	6-70
6.3.33	HIMC_GetGrpPoseFb.....	6-71
6.4	グループステータス .....	6-72
6.4.1	HIMC_IsGrpEnabled .....	6-72
6.4.2	HIMC_IsGrpMoving.....	6-73
6.4.3	HIMC_IsGrpInPos .....	6-74
6.4.4	HIMC_IsGrpErrorStop.....	6-75
6.5	高度なグループモーションコントロール .....	6-76
6.5.1	HIMC_LinAbs .....	6-76
6.5.2	HIMC_LinRel .....	6-78
6.5.3	HIMC_CircAbs .....	6-80
6.5.4	HIMC_CircRel .....	6-82
7.	GPIO 機能.....	7-1
7.1	概要 .....	7-2
7.1.1	GPIO 変数 .....	7-2
7.2	コントローラー IO 設定 .....	7-3
7.2.1	HIMC_SetHimcGpo.....	7-3
7.2.2	HIMC_ToggleHimcGpo .....	7-4
7.2.3	HIMC_SetHimcMultiGpo.....	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	スレーブ IO 設定.....	7-9
7.3.1	HIMC_SetSlaveGpo .....	7-9
7.3.2	HIMC_ToggleSlaveGpo .....	7-10
7.3.3	HIMC_SetSlaveMultiGpo.....	7-11
7.4	コントローラー IO ステータス .....	7-12
7.4.1	HIMC_GetHimcGpi.....	7-12
7.4.2	HIMC_GetHimcGpo .....	7-13
7.4.3	HIMC_GetHimcMultiGpi .....	7-14
7.4.4	HIMC_GetHimcMultiGpo .....	7-15
7.4.5	HIMC_GetHimcGpiNum .....	7-16

7.4.6	HIMC_GetHimcGpoNum .....	7-17
7.4.7	HIMC_GetAllGPIInvertSt .....	7-18
7.4.8	HIMC_GetAllGPOInvertSt .....	7-19
7.5	スレーブ IO ステータス .....	7-20
7.5.1	HIMC_GetSlaveGpi .....	7-20
7.5.2	HIMC_GetSlaveGpo .....	7-21
7.5.3	HIMC_GetSlaveMultiGpi .....	7-22
7.5.4	HIMC_GetSlaveMultiGpo .....	7-23
7.5.5	HIMC_GetSlaveGpiNum .....	7-24
7.5.6	HIMC_GetSlaveGpoNum .....	7-25
8.	AIO 機能 .....	8-1
8.1	概要 .....	8-2
8.2	スレーブ AIO 設定 .....	8-3
8.2.1	HIMC_SetSlvAOMonitor .....	8-3
8.2.2	HIMC_SetSlvAIParam .....	8-5
8.2.3	HIMC_SetSlvAOParam .....	8-6
8.2.4	HIMC_SetSlvAO .....	8-7
8.3	スレーブ AIO ステータス .....	8-8
8.3.1	HIMC_GetSlvAIType .....	8-8
8.3.2	HIMC_GetSlvAIScale .....	8-9
8.3.3	HIMC_GetSlvAIOffset .....	8-10
8.3.4	HIMC_GetSlvAOScale .....	8-11
8.3.5	HIMC_GetSlvAOOffset .....	8-12
8.3.6	HIMC_IsSlvAOBound .....	8-13
8.3.7	HIMC_GetSlvAI .....	8-14
8.3.8	HIMC_GetSlvAO .....	8-15
9.	ユーザーテーブル機能 .....	9-1
9.1	概要 .....	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.	ポジショントリガー機能 .....	10-1
10.1	概要 .....	10-2
10.1.1	PT 変数 .....	10-2
10.1.2	PT 機能の利用の流れ .....	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.	タッチプローブ機能 .....	11-1
11.1	概要 .....	11-2
11.2	HIMC_EnableTouchProbe .....	11-3
11.3	HIMC_DisableTouchProbe .....	11-4
11.4	HIMC_IsTouchProbeEnabled .....	11-5
11.5	HIMC_IsTouchProbeTriggered .....	11-6

11.6	HIMC_GetTouchProbePos .....	11-7
12.	ダイナミックエラー補償機能.....	12-1
12.1	概要 .....	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.....	12-9
12.7	HIMC_GetCompPos.....	12-10
12.8	HIMC_SetCompAlgType.....	12-11
13.	フィルター機能 .....	13-1
13.1	概要 .....	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 タスク機能 .....	14-1
14.1	概要 .....	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.	コールバック機能.....	15-1
15.1	HIMC_SetHmplEvtCallback.....	15-2
15.2	HIMC_SetErrorCallback .....	15-3
15.3	HIMC_SetHmplMsgEvtCallback .....	15-4
16.	変数と関数演算機能.....	16-1
16.1	概要 .....	16-2
16.1.1	ドライバ変数一覧.....	16-3
16.1.2	コントローラ変数一覧 .....	16-4
16.2	ドライバ可変運転 .....	16-7
16.2.1	HIMC_GetSlvVar .....	16-7
16.2.2	HIMC_SetSlvVar.....	16-8
16.2.3	HIMC_GetSlvSt.....	16-9
16.2.4	HIMC_SetSlvSt.....	16-10
16.3	ドライバ機能の運転.....	16-11
16.3.1	HIMC_RunSlvPdIFunc.....	16-11
16.3.2	HIMC_StopSlvPdIFunc .....	16-12
16.3.3	HIMC_IsSlvPdIFuncRunning.....	16-13
16.4	コントローラ変数操作 .....	16-14
16.4.1	HIMC_GetVariableByID .....	16-14
16.4.2	HIMC_SetVariableByID.....	16-15
16.4.3	HIMC_GetVariableListByID .....	16-16
16.4.4	HIMC_SetVariableListByID .....	16-17
16.4.5	HIMC_GetGlobalVariables .....	16-18

	16.4.6	HIMC_SetGlobalVariables.....	16-19
17.		HIMC エラー機能.....	17-1
	17.1	概要 .....	17-2
	17.1.1	システムエラーメッセージ .....	17-3
	17.1.2	軸エラーメッセージ .....	17-6
	17.1.3	グループのエラーメッセージ .....	17-8
	17.2	HIMC_GetLastError .....	17-10
	17.3	HIMC_GetAxisLastErr .....	17-11
	17.4	HIMC_ClearAxisLastErr.....	17-12
	17.5	HIMC_GetGrpLastErr .....	17-13
	17.6	HIMC_ClearGrpLastErr .....	17-14
	17.7	HIMC_GetErrorInformation.....	17-15
18.		データ構造 .....	18-1
	18.1	ComInfo .....	18-2
	18.2	CoordPosition .....	18-3
	18.3	MotionProfile.....	18-4
	18.4	CenterPosition .....	18-5
	18.5	NormalVector .....	18-6
	18.6	PosTriggerPar.....	18-7
19.		列挙 .....	19-1
	19.1	ComType .....	19-2
	19.2	CoordSystem.....	19-3
	19.3	MotionBufferMode .....	19-4
	19.4	MotionTransitionMode .....	19-5
	19.5	ShaperMode .....	19-6
20.		付録 .....	20-7
	20.1	API エラーコード .....	20-8

# 1. はじめに

1.1	API の仕組み .....	1-2
1.2	API 環境のセットアップ .....	1-2
1.2.1	C / C++ .....	1-2
1.2.2	C# .....	1-2
1.2.3	LabView .....	1-2
1.2.4	Python .....	1-2
1.3	法的免責事項 .....	1-3

## 1.1 API の仕組み

このリファレンスガイドで紹介する関数は、主に C 言語で使用されます。他の API 環境の対応名については、各セクションの説明を参照してください。

注：

iA Studio 1.3 (同梱)以降で採用されているモーション変数の単位：  
直線運動 (mm)、回転運動 (deg)、時間 (ms)

iA Studio 1.2 (付属) 以前で採用されているモーション変数の単位：  
直線運動 (m)、回転運動 (rad)、時間 (s)

## 1.2 API 環境のセットアップ

### 1.2.1 C / C++

1. <インストールディレクトリ> \documents\Software Guides\API\_examples\cpp\vs\_project に移動します。
2. copy\_required\_file.bat を実行します。
3. プロジェクトファイル api\_example.sln を開きます。

### 1.2.2 C#

1. <インストールディレクトリ> \documents\Software Guides\API\_examples\c\_sharp に移動します。
2. copy\_required\_file.bat を実行します。
3. プロジェクトファイル api\_example.sln を開きます。

### 1.2.3 LabView

1. <インストールディレクトリ> \documents\Software Guides\API\_examples\labview に移動します。
2. copy\_required\_file.bat を実行します。
3. プロジェクトファイル example.lvproj を開きます。

### 1.2.4 Python

1. <インストールディレクトリ> \documents\Software Guides\API\_examples\python に移動します。
2. copy\_required\_file.bat を実行します。
3. example python example.py を直接実行します。

## 1.3 法的免責事項

ユーザーは、特定の用途のために、このガイドで提供されているサンプルコードを採用または変更できます。ただし、正確性、有効性、および安全性は、異なるアプリケーションシナリオでは保証できません。ユーザーは、ソフトウェア実装の安全性と有効性について全責任を負う必要があります。

(このページはブランクになっています)



## 2. HIMC システム機能

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_IsSystemOper.....	2-7
2.7	HIMC_IsSystemPreOp .....	2-8
2.8	HIMC_IsSystemError.....	2-9
2.9	HIMC_DisableAll .....	2-10
2.10	HIMC_StopAll .....	2-11
2.11	HIMC_EStop.....	2-12
2.12	HIMC_GetSlaveNum .....	2-13
2.13	HIMC_GetFirmwareVer .....	2-14
2.14	HIMC_RescanMoE.....	2-15
2.15	HIMC_SetEconMode .....	2-16
2.16	HIMC_IsEconMode .....	2-17

## 2.1 HIMC\_ConnectCtrl

### 目的

コントローラーに接続します。

### 構文

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

### パラメーター

com_info [in]	接続の情報を格納する構造体
p_ctrl_id [out]	正常に接続されたコントローラーID を受け取るバッファーへのポインター。 ユーザーは、この ID を他の API 関数でを使用して、このコントローラーを 操作できます。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

HIMC に接続するには、HIMC\_ConnectToSimulator または HIMC\_ConnectToEthernet を使用することをお勧めします。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_ConnectCtrl
LabVIEW	--
Python	--

## 2.2 HIMC\_ConnectToSimulator

### 目的

シミュレーターに接続します。

### 構文

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

### パラメーター

**p\_ctrl\_id [out]**      正常に接続されたコントローラーID を受け取るバッファーへのポインター。ユーザーは、この ID を他の API 関数で使用して、このコントローラーを操作できます。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 2.3 HIMC\_ConnectToEthernet

### 目的

イーサネットに接続します。

### 構文

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

### パラメーター

ip\_address [in] IP アドレスの文字列

port [in] ポートの文字列

p\_ctrl\_id [out] 正常に接続されたコントローラーID を受け取るバッファーへのポインター。ユーザーは、この ID を他の API 関数でを使用して、このコントローラーを操作できます。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 2.4 HIMC\_DisconnectCtrl

目的

コントローラーとの接続を切断します。

構文

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

パラメーター

ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
                          HIMC\_ConnectCtrl を呼び出して取得する必要があります。

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_DisconnectCtrl
LabVIEW	HIMC Disconnect Ctrl.vi
Python	Disconnect when class is deleted

## 2.5 HIMC\_RebootController

### 目的

コントローラーを再起動します。

### 構文

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

### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_RebootController
LabVIEW	HIMC Reboot Controller.vi
Python	RebootController

## 2.6 HIMC\_IsSystemOper

### 目的

HIMC システムが「操作」状態にあるかどうかを問い合わせます。その場合、モーション関連の機能を使用できます。

### 構文

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

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- isOper [out]      HIMC システムの「操作」状態を受け取るバッファーへのポインター。  
HIMC システムが「SystemOper」状態の場合、値は 1 になります。  
それ以外の場合は、0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 2.7 HIMC\_IsSystemPreOp

### 目的

HIMC システムが「操作前」の状態にあるかどうかを問い合わせます。その場合、HIMC とスレーブ間の通信は確立されますが、モーション関連の機能は使用できません。

### 構文

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

### パラメーター

- ctrl\_id [in]**      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- isPreOp [out]**      HIMC システムの「操作前」状態を受け取るためのバッファへのポインター。  
HIMC システムが「SystemPreOp」状態の場合、値は 1 になります。  
それ以外の場合は、0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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



## 2.8 HIMC\_IsSystemError

### 目的

HIMC システムが「エラー」状態にあるかどうかを問い合わせます。そうである場合、HIMC とスレーブ間の通信でエラーが発生します。

### 構文

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

### パラメーター

- ctrl\_id [in]**      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- isError [out]**      HIMC システムの「エラー」状態を受け取るバッファーへのポインター。  
HIMC システムが「SystemError」状態の場合、値は 1 になります。  
それ以外の場合は、0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 2.9 HIMC\_DisableAll

### 目的

すべての軸とすべての軸グループを無効にします。

### 構文

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

### パラメーター

ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

軸と軸グループのモーションキューがクリアされます。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_DisableAll
LabVIEW	HIMC Disable All.vi
Python	DisableAll

## 2.10 HIMC\_StopAll

### 目的

キル減速で全軸、全軸グループを停止させ、「有効」状態にします。

### 構文

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

### パラメーター

ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

軸と軸グループのモーションキューがクリアされます。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_StopAll
LabVIEW	HIMC Stop All.vi
Python	StopAll

## 2.11 HIMC\_EStop

### 目的

コントローラー内のすべての実行プログラム(すべての HMPL タスクを含む)を停止し、すべての軸とすべての軸グループを無効にします。

### 構文

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

### パラメーター

ctrl\_id [in]            HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_EStop
LabVIEW	HIMC E Stop.vi
Python	EStop

## 2.12 HIMC\_GetSlaveNum

### 目的

コントローラーに接続されているスレーブの数を取得します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
p_slave_num [out]	コントローラーに接続されているスレーブの数を受け取るバッファーへのポインター。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 2.13 HIMC\_GetFirmwareVer

### 目的

コントローラーのファームウェアバージョンを取得します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
p_ver_buf [out]	返されたファームウェアバージョンの文字列を受け取るバッファへのポインター。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 2.14 HIMC\_RescanMoE

### 目的

コントローラーとスレーブ間の接続を再スキャンします。

### 構文

```
int HIMC_RescanMoE(
    int ctrl_id
);
```

### パラメーター

ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_RescanMoE
LabVIEW	HIMC Rescan MoE.vi
Python	RescanMoE

## 2.15 HIMC\_SetEconMode

### 目的

HIMC API パフォーマンスモードを設定します。

### 構文

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

### パラメーター

**ctrl\_id [in]**      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**mode [in]**      HIMC API パフォーマンスモード。  
0：ハイパフォーマンスモード(初期値)  
1：エコモード

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

- (1) ハイパフォーマンスモードは、HIMC API の平均応答速度は高速ですが、CPU 使用率はエコモードよりも高くなります。
- (2) エコモードは CPU 使用率を削減できますが、HIMC API の平均応答時間は増加します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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



## 2.16 HIMC\_IsEconMode

### 目的

HIMC API パフォーマンスモードが Econ モードかどうかを照会します。

### 構文

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

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID。  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- p\_is\_econ [in]      HIMC API パフォーマンスモードを受け取るバッファへのポインター。  
エコモードの場合、値は 1 になります。それ以外の場合は、0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

(このページはブランクになっています)

## 3. 軸機能

3.1	概要 .....	3-3
3.1.1	軸変数 .....	3-6
3.2	軸モーション制御 .....	3-9
3.2.1	HIMC_Enable .....	3-9
3.2.2	HIMC_Disable .....	3-10
3.2.3	HIMC_Reset .....	3-11
3.2.4	HIMC_MoveAbs .....	3-12
3.2.5	HIMC_MoveRel .....	3-13
3.2.6	HIMC_MoveVel .....	3-14
3.2.7	HIMC_MoveTrq .....	3-15
3.2.8	HIMC_MovePVT .....	3-16
3.2.9	HIMC_Stop .....	3-18
3.2.10	HIMC_Halt .....	3-19
3.2.11	HIMC_Resume .....	3-20
3.3	軸設定 .....	3-21
3.3.1	HIMC_GetMaxVel .....	3-21
3.3.2	HIMC_SetVel .....	3-22
3.3.3	HIMC_GetMaxAcc .....	3-23
3.3.4	HIMC_SetAcc .....	3-24
3.3.5	HIMC_SetAccTime .....	3-25
3.3.6	HIMC_GetMaxDec .....	3-26
3.3.7	HIMC_SetDec .....	3-27
3.3.8	HIMC_SetDecTime .....	3-28
3.3.9	HIMC_GetKillDec .....	3-29
3.3.10	HIMC_SetKillDec .....	3-30
3.3.11	HIMC_GetSWRL .....	3-31
3.3.12	HIMC_SetSWRL .....	3-32
3.3.13	HIMC_GetSWLL .....	3-33
3.3.14	HIMC_SetSWLL .....	3-34
3.3.15	HIMC_GetSMTTime .....	3-35
3.3.16	HIMC_SetSMTTime .....	3-36
3.3.17	HIMC_GetMoveTime .....	3-37
3.3.18	HIMC_GetSettlingTime .....	3-38
3.3.19	HIMC_SetPos .....	3-39
3.3.20	HIMC_GetPosFb .....	3-40
3.3.21	HIMC_GetPosOffset .....	3-41
3.3.22	HIMC_GetPosErr .....	3-42
3.3.23	HIMC_GetVelFb .....	3-43
3.3.24	HIMC_GetVelErr .....	3-44
3.3.25	HIMC_GetCurrFb .....	3-45
3.3.26	HIMC_GetRefPos .....	3-46
3.3.27	HIMC_GetRefVel .....	3-47
3.3.28	HIMC_GetRefAcc .....	3-48

3.3.29	HIMC_GetPosOut.....	3-49
3.3.30	HIMC_GetVelOut.....	3-50
3.3.31	HIMC_GetAccOut.....	3-51
3.3.32	HIMC_IgnoreHWL .....	3-52
3.3.33	HIMC_IgnoreSWL .....	3-53
3.3.34	HIMC_IgnorePE.....	3-54
3.3.35	HIMC_GetAxisNum .....	3-55
3.3.36	HIMC_SetVelScale.....	3-56
3.3.37	HIMC_GetVelScale .....	3-57
3.3.38	HIMC_SetRollover.....	3-58
3.3.39	HIMC_GetRolloverTurns .....	3-59
3.3.40	HIMC_GetMultiAxesFeedbackPos.....	3-60
3.4	軸の状態.....	3-61
3.4.1	HIMC_IsEnabled.....	3-61
3.4.2	HIMC_IsMoving .....	3-62
3.4.3	HIMC_IsInPos.....	3-63
3.4.4	HIMC_IsErrorStop .....	3-64
3.4.5	HIMC_IsGantry .....	3-65
3.4.6	HIMC_IsGrouped.....	3-66
3.4.7	HIMC_IsSync .....	3-67
3.4.8	HIMC_IsHWLL .....	3-68
3.4.9	HIMC_IsHWRL .....	3-69
3.4.10	HIMC_IsSWLL .....	3-70
3.4.11	HIMC_IsSWRL.....	3-71
3.4.12	HIMC_IsDriveErr.....	3-72
3.4.13	HIMC_IsPosErr .....	3-73
3.4.14	HIMC_IsCompActive.....	3-74
3.4.15	HIMC_IsAcc .....	3-75



図 3.1.3 が示すように、滑らかな時間を追加すると基準速度が遅くなりますが、システムの安定性を高めるために高加速度によって生成されるジャークを効果的に下げることができます。ジャークと最大加速度、滑らかな時間の関係は次のようになります。

$$\text{ジャーク} = \text{最大加速度} / \text{スムーズ時間 (Ts)}$$

総加速時間は、次の式で求められます。

$$\text{合計加速時間 (T)} = \text{T カurve加速時間 (Ta)} + \text{スムーズ時間 (Ts)}$$

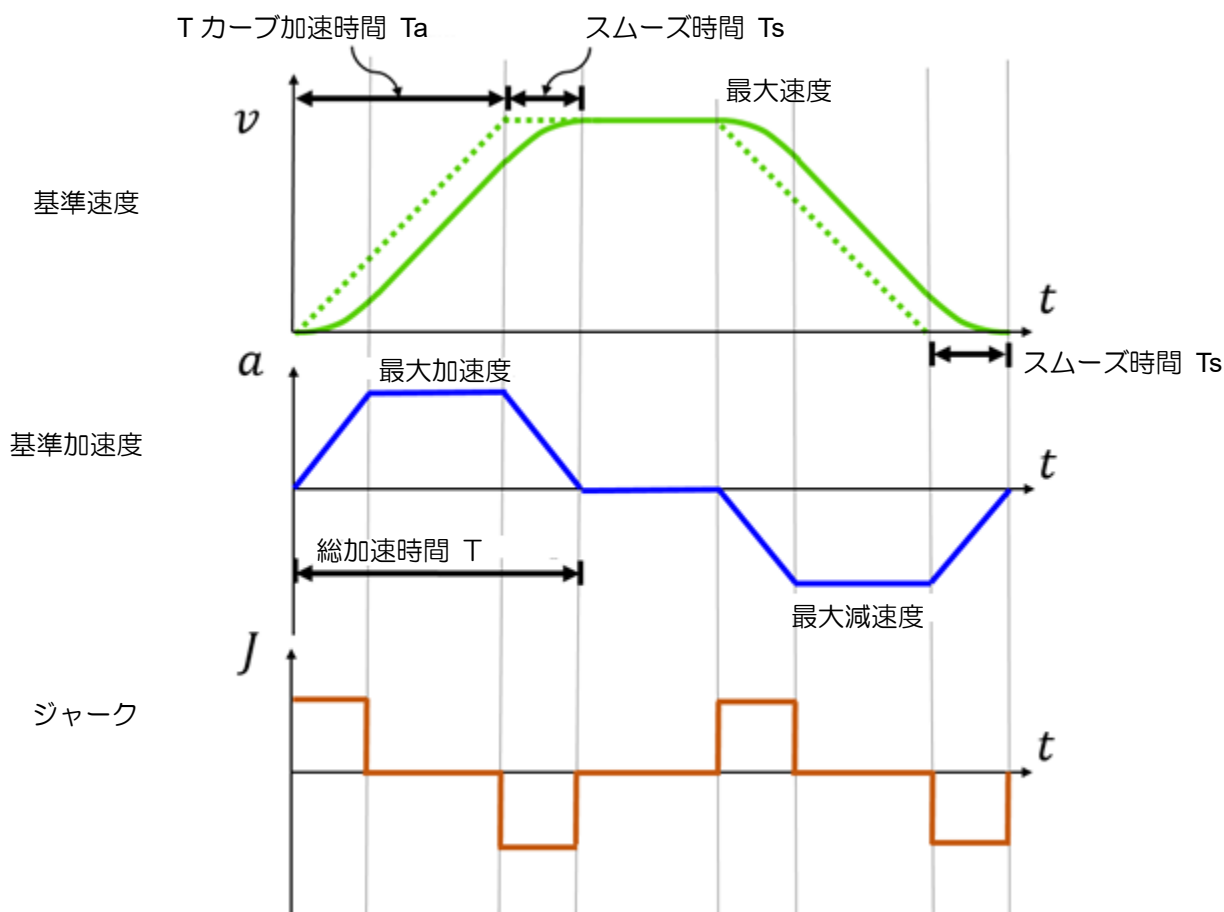


図 3.1.3

さらに、HIMC 軸モーションコマンドは、動的ターゲット位置と速度計画の変更をサポートします(オンザフライ修正)。軸移動中、ユーザーは軸の目標位置、最大速度、最大加減速度を変更できます。HIMC のプロファイルジェネレーターは、新しいコマンドとモーションパラメーターに基づいて、新しいターゲット位置に移動します。

軸の動作状態は、図 3.1.4 に示すように「移動中」と「インポジション」に分けられます。セクション I で軸位置計画コマンドを送信し続け、セクション II に入る前に終了します。コントローラーは、設定されたターゲット半径とデバウンス時間に基づいて、軸が所定の位置にあるかどうかを判断します。

デバウンス時間後に軸フィードバック位置が参照位置のターゲット半径内に留まる場合、軸位置はインポジションと見なされます。このとき、コントローラー内部で「軸インポジション」のステータスが確立されます。ただし、デバウンス時間中に軸フィードバック位置が基準位置のターゲット半径を超えると、整定時間の計算がリセットされます。次に軸フィードバック位置がターゲット半径に入るまで、デバウンス時間のインポジション状態はチェックされません。

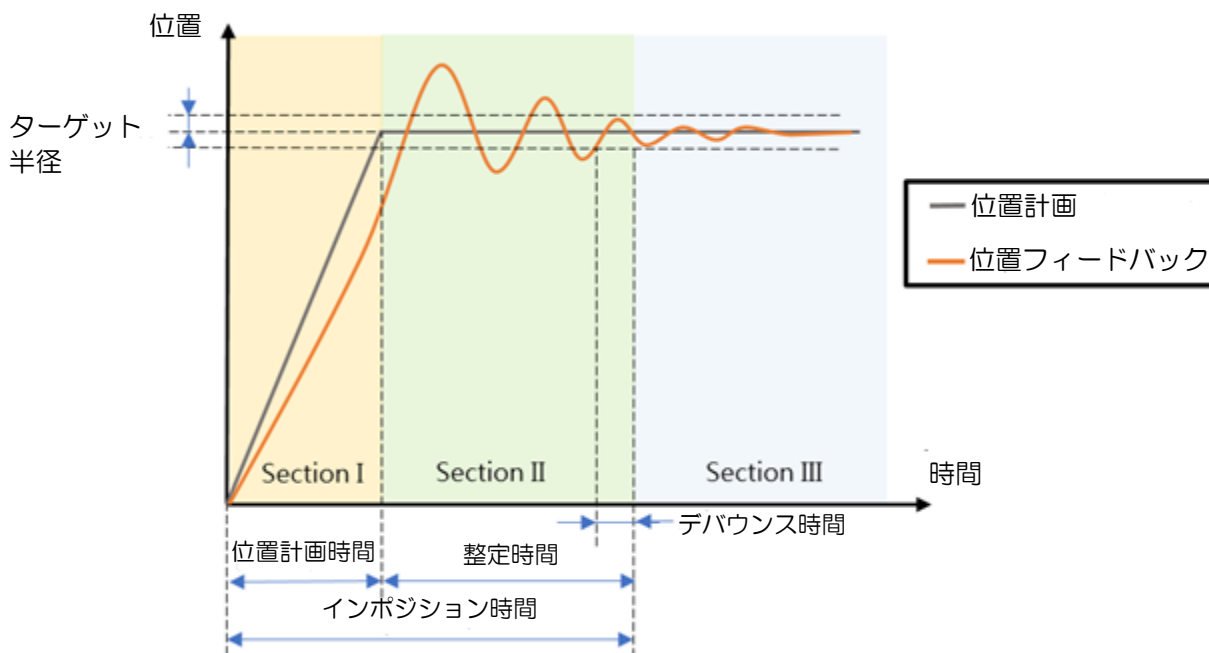


図 3.1.4

図 3.1.4 を参照すると、軸の動作ステータスは次のように記述されます。

1. セクション I：軸が動いていて、位置が合っていません。
2. セクション II：軸は動いていませんが、位置が合っていません。
3. セクション III：軸は動いておらず、所定の位置にあります。

軸が「同期」状態にある場合、軸グループまたはマスター軸は軸モーションのモーションプロファイルを生成します。軸自体はモーションプロファイルを生成しません。代わりに、軸グループまたはマスター軸によって計画された参照位置にのみ従います。

## 3.1.1 軸変数

軸変数は、モーションコマンド変数、プロファイルジェネレーター変数、ステータス変数の3つのカテゴリに分類されます。ユーザーは、iA Studio のスコープマネージャーを介して目的の変数を選択できます（「iA Studio ユーザーガイド」のセクション 4.8 を参照）。詳細な説明を表 3.1.1.1 から表 3.1.1.5 に示します。

表 3.1.1.1 軸のモーションコマンド変数

名称	変数	単位	説明
Sync Position Setpoint	sync_pos_set	mm または deg	同期位置セットポイント。これは、軸が同期動作（軸グループ、カムまたはギア操作など）にある場合に追従する目標位置です。
Position Command	pg_pos_cmd	mm または deg	プロファイルジェネレーターの位置コマンド。これは、軸が個別の動作（2 点間）にある場合に追従する目標位置です。
Reference Position	ref_pos	mm または deg	参考位置。これは、定義済みのモーションプロファイルに従ってプロファイルジェネレーターから生成された位置セットポイントです。
Reference Velocity	ref_vel	mm/s または deg/s	基準速度。これは、定義済みのモーションプロファイルに従ってプロファイルジェネレーターから生成された速度設定値です。
Reference Acceleration	ref_acc	mm/s <sup>2</sup> または deg/s <sup>2</sup>	参照加速度。これは、定義済みのモーションプロファイルに従ってプロファイルジェネレーターから生成された加速度設定値です。
Position Compensation	pos_comp_set	mm または deg	位置補正值。動的誤差補正機能の誤差マップの出力です。機能が無効になっている場合はゼロになります。
Compensated Position	pos_comp	mm または deg	補正位置指令値。基準位置と位置補正值の和です。
Position Offset	pos_offset	mm または deg	位置オフセット。デフォルト値はゼロです。ユーザーがモーターを動かさずに新しい軸位置を設定すると、ゼロ以外になります。
Position Output	pos_out	mm または deg	位置出力。位置オフセットなしの軸位置指令です。
Velocity Output	vel_out	mm/s または deg/s	速度出力。速度オフセットなしの軸速度指令です。
Acceleration Output	acc_out	mm/s <sup>2</sup> または deg/s <sup>2</sup>	加速出力。加速度オフセットなしの軸加速度指令です。
Offsetted Position Output	offset_pos_out	mm または deg	オフセット付き位置出力。これは、位置オフセットを含む最終的に計算された軸位置コマンドです。値は単位「カウント」に変換され、対応するスレーブに送信されます。
Raw Position Feedback	pos_fb_raw	mm または deg	生の位置フィードバック。これは、スレーブから読み取ったエンコーダフィードバックです。
Offsetted Position Feedback	offset_pos_fb	mm または deg	オフセット位置フィードバック。位置オフセット付きの位置フィードバックです。
Position Feedback	pos_fb	mm または deg	位置フィードバック。これは軸座標系です。
Velocity Feedback	vel_fb	mm/s または deg/s	速度フィードバック。これは軸座標系です。
Position Error	pos_err	mm または deg	位置エラー。これは、位置出力と生の位置フィードバックの違いです。



名称	変数	単位	説明
Velocity Error	vel_err	mm/s または deg/s	速度エラー。これは、速度出力と生の速度フィードバックの差です。
Move Time	movetime	ms	移動時間
Settling Time	settlingtime	ms	整定時間

表 3.1.1.2 軸のプロファイルジェネレーター変数

名称	変数	単位	説明
Max. Profile Velocity	max_vel	mm/s または deg/s	最大プロファイル速度。必ずしも達していない。
Max. Profile Acceleration	max_acc	mm/s <sup>2</sup> または deg/s <sup>2</sup>	最大プロファイル加速。必ずしも達していない。
Profile Deceleration	max_dec	mm/s <sup>2</sup> または deg/s <sup>2</sup>	最大プロファイル減速。必ずしも達していない。
Smooth Time	sm_factor	ms	スムーズな時間。入力範囲は 0 ~ 500 です。値を大きくすると、モーション中の機械的振動を減らすことができますが、合計モーション時間は影響を受けます。

表 3.1.1.3 軸のステータス変数

名称	変数	単位	説明
Fault Status	fault_status	N/A	軸のエラーステータス; ビット定義については、表 3.1.1.4 を参照してください。
Motion Status	motion_status	N/A	軸のモーションステータス; ビット定義については、表 3.1.1.5 を参照してください。

表 3.1.1.4 軸エラーステータスのビット定義

Bit	名称	説明	デフォルトの応答
0	Error Stop	「エラー停止」状態の軸	N/A
1	Drive fault	スレーブドライバーの障害	コントローラーは軸を無効にします
2	Position error	位置誤差が保護限界を超えています	コントローラーは軸を無効にします
3	Hardware right limit	軸ハードウェア右リミットがトリガーされました	コントローラーはモーションを停止します
4	Hardware left limit	軸ハードウェアの左リミットがトリガーされました	コントローラーはモーションを停止します
5	Software right limit	軸ソフト右リミット発動	コントローラーはモーションを停止します
6	Software left limit	軸ソフトウェアの左リミットがトリガーされました	コントローラーはモーションを停止します

表 3.1.1.5 軸モーションステータスのビット定義

Bit	名称	説明	備考
0	Enabled	軸が有効になります	N/A
1	Moving	軸が動いています	セクション 3.1 を参照してください
2	In Position	軸はインポジションです	セクション 3.1 を参照してください
3	Synchronous	軸は「同期」状態にあります	軸が軸グループ内にあるか、同期モーションのスレーブ軸です。
4	Group	軸は軸グループにあります	セクション 6.1 を参照してください
5	Gantry	軸はガントリー軸です	セクション 5.1 を参照してください
6	Input Shape	入力シェイプフィルタを有効にします	セクション 13.1 を参照してください
7	VSF	VSF フィルターを有効にします	セクション 13.1 を参照してください
8	Gear	軸は電子ギアスレーブ軸です	セクション 4.1 を参照してください
9	Cam	軸は電子カムスレーブ軸です	まだサポートされていません
10	Accelerating	軸は加速しています	セクション 3.1 を参照してください
11	Homed	軸が原点復帰を完了します	N/A

## 3.2 軸モーション制御

### 3.2.1 HIMC\_Enable

#### 目的

軸を有効にします

#### 構文

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

#### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

axis\_id [in]      Axis index

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_Enable
LabVIEW	HIMC Enable.vi
Python	Enable

3.2.2 HIMC\_Disable

目的  
軸を無効にします

構文

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

パラメーター

ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
                          HIMC\_ConnectCtrl を呼び出して取得する必要があります。

axis\_id [in]           Axis index

戻りの値  
関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

備考  
軸のモーションキューがクリアされます。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_Disable
LabVIEW	HIMC Disable.vi
Python	Disable

### 3.2.3 HIMC\_Reset

#### 目的

「エラー停止」状態の軸をリセットします

#### 構文

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

#### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

axis\_id [in]      Axis index

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_Reset
LabVIEW	HIMC Reset.vi
Python	Reset

## 3.2.4 HIMC\_MoveAbs

### 目的

軸を絶対目標位置に移動します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
pos [in]	絶対目標位置の値 パラメーター単位：mm または deg

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_MoveAbs
LabVIEW	HIMC Move Abs.vi
Python	MoveAbs

### 3.2.5 HIMC\_MoveRel

#### 目的

相対距離で軸を移動します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
rel_dist [in]	相対距離に対する値 パラメーター単位：mm または deg

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_MoveRel
LabVIEW	HIMC Move Rel.vi
Python	MoveRel

## 3.2.6 HIMC\_MoveVel

### 目的

特定の速度で終わりのない動きを開始すること

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
vel [in]	特定の速度の値 モーションの方向を示すために、正または負のいずれかになります。 パラメーター単位：mm/s または deg/s

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_MoveVel
LabVIEW	HIMC Move Vel.vi
Python	MoveVel



### 3.2.7 HIMC\_MoveTrq

#### 目的

特定のトルクで終わりのない運動を開始すること

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
torque_cmd [in]	トルク指令 パラメーター単位：N-m

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 備考

- (1) この機能は、「Profile Torque」モードにのみ適用されます。
- (2) トルク指令がモーターの連続トルクより大きい場合、モーターは連続トルクの値で動きます。

#### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_MoveTrq
LabVIEW	HIMC Move Trq.vi
Python	MoveTrq

## 3.2.8 HIMC\_MovePVT

### 目的

指定された位置(P)、速度(V)、時間(T)に基づいて、指定された位置に軸を移動します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
time [in]	ユーザーが指定した時間配列へのポインター パラメーター単位：ms
pos [in]	ユーザーが指定した位置配列へのポインター パラメーター単位：mm または deg
vel [in]	ユーザーが指定した速度配列へのポインター パラメーター単位：mm/s または deg/s
num_pt [in]	PVT モーションポイントの数。その最大値は 50 です 時間の長さ、位置、および速度の配列は、このパラメーターと同じでなければなりません

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_MovePVT
LabVIEW	HIMC Move PVT.vi
Python	MovePVT

## 3.2.9 HIMC\_Stop

### 目的

軸の動きを止めること

### 構文

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

### パラメーター

**ctrl\_id [in]**      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**axis\_id [in]**      Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

軸のモーションキューがクリアされます

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_Stop
LabVIEW	HIMC Stop.vi
Python	Stop

### 3.2.10 HIMC\_Halt

#### 目的

軸の動きを停止するには、その速度は 0 に設定されます。

#### 構文

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

#### パラメーター

**ctrl\_id [in]**      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**axis\_id [in]**      Axis index

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 備考

軸が所定の位置にない場合、軸は動き続けます。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_Halt
LabVIEW	HIMC Halt.vi
Python	Halt

3.2.11 HIMC\_Resume

目的

「停止」状態から軸の動きを再開すること。

構文

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

パラメーター

ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
                         HIMC\_ConnectCtrl を呼び出して取得する必要があります。

axis\_id [in]           Axis index

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_Resume
LabVIEW	HIMC Resume.vi
Python	Resume

## 3.3 軸設定

### 3.3.1 HIMC\_GetMaxVel

#### 目的

軸の最大プロファイル速度を取得します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_vel [out]	軸の最大プロファイル速度を受け取るバッファーへのポインター パラメーター単位：mm/s または deg/s

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetMaxVel
LabVIEW	HIMC Get Max Vel.vi
Python	GetMaxVel

## 3.3.2 HIMC\_SetVel

### 目的

軸の最大プロファイル速度を設定します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
vel [in]	軸の新しい最大プロファイル速度 パラメーター単位：mm/s または deg/s 入力範囲：ゼロ以外の正の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetVel
LabVIEW	HIMC Set Vel.vi
Python	SetVel



### 3.3.3 HIMC\_GetMaxAcc

#### 目的

軸の最大プロファイル加速度を取得します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl から取得する必要があります。
axis_id [in]	Axis index
p_acc [out]	軸の最大プロファイル加速度を受け取るバッファーへのポインター。 パラメーター単位：mm/s <sup>2</sup> または deg/s <sup>2</sup>

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetMaxAcc
LabVIEW	HIMC Get Max Acc.vi
Python	GetMaxAcc

## 3.3.4 HIMC\_SetAcc

### 目的

軸の最大プロファイル加速度を設定します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
acc [in]	軸の新しい最大プロファイル加速度 パラメーター単位：mm/s <sup>2</sup> または deg/s <sup>2</sup> 入力範囲：ゼロ以外の正の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetAcc
LabVIEW	HIMC Set Acc.vi
Python	SetAcc

### 3.3.5 HIMC\_SetAccTime

#### 目的

軸の加速時間を設定します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
acc_time [in]	軸の加速時間 パラメーター単位：ms 入力範囲：ゼロ以外の正の値

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 3.3.6 HIMC\_GetMaxDec

### 目的

軸の最大プロファイル減速度を取得します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_dec [out]	軸の最大プロファイル減速度を受け取るバッファーへのポインター パラメーター単位：mm/s <sup>2</sup> または deg/s <sup>2</sup>

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetMaxDec
LabVIEW	HIMC Get Max Dec.vi
Python	GetMaxDec

### 3.3.7 HIMC\_SetDec

#### 目的

軸の最大プロファイル減速度を設定します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
dec [in]	軸の新しい最大プロファイル減速度 パラメーター単位：mm/s <sup>2</sup> または deg/s <sup>2</sup> 入力範囲：ゼロ以外の正の値

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_SetDec
LabVIEW	HIMC Set Dec.vi
Python	SetDec

## 3.3.8 HIMC\_SetDecTime

### 目的

軸の減速時間を設定します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
dec_time [in]	軸の減速時間 パラメーター単位：ms 入力範囲：ゼロ以外の正の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

### 3.3.9 HIMC\_GetKillDec

#### 目的

軸のキル減速度を取得します

#### 構文

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

#### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID
- HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]      Axis index
- p\_kill\_dec [out]   軸のキル減速を受け取るバッファへのポインター  
                  パラメーター単位：mm/s<sup>2</sup> または deg/s<sup>2</sup>

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 3.3.10 HIMC\_SetKillDec

### 目的

軸のキル減速度を設定します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
kill_dec [in]	軸の新しいキル減速 パラメーター単位：mm/s <sup>2</sup> または deg/s <sup>2</sup> 入力範囲：ゼロ以外の正の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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



### 3.3.11 HIMC\_GetSWRL

#### 目的

軸のソフトリミット位置を取得します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_right_limit_pos [out]	軸のソフトウェア右リミット位置を受け取るバッファーへのポインター パラメーター単位：mm または deg

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetSWRL
LabVIEW	HIMC Get SWRL.vi
Python	GetSWRL

## 3.3.12 HIMC\_SetSWRL

### 目的

軸のソフトウェア右リミット位置を設定します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
right_limit_pos [in]	軸の新しいソフトウェア右リミット位置 パラメーター単位：mm または deg

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetSWRL
LabVIEW	HIMC Set SWRL.vi
Python	SetSWRL

### 3.3.13 HIMC\_GetSWLL

#### 目的

軸のソフト左リミット位置を取得します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_left_limit_pos [out]	軸のソフトウェア左リミット位置を受け取るバッファへのポインター パラメーター単位：mm または deg

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetSWLL
LabVIEW	HIMC Get SWLL.vi
Python	GetSWLL

3.3.14 HIMC\_SetSWLL

目的

軸のソフトウェア左リミット位置を設定します

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index
- left\_limit\_pos [in]

軸の新しいソフトウェア左リミット位置  
パラメーター単位：mm または deg

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_SetSWLL
LabVIEW	HIMC Set SWLL.vi
Python	SetSWLL

### 3.3.15 HIMC\_GetSMTTime

#### 目的

軸のプロファイルの滑らかな時間を取得します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_smooth_time [out]	軸のプロファイルの滑らかな時間を受け取るバッファーへのポインター パラメーター単位：ms

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetSMTTime
LabVIEW	HIMC Get SM Time.vi
Python	GetSMTTime

## 3.3.16 HIMC\_SetSMTIME

### 目的

軸のプロファイルスムーズ時間を設定します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
smooth_time [in]	軸の新しいプロファイルスムーズ時間 パラメーター単位：ms 入力範囲：0 ～ 500

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

軸が移動中の場合、この機能は適用されません

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetSMTIME
LabVIEW	HIMC Set SM Time.vi
Python	SetSMTIME

### 3.3.17 HIMC\_GetMoveTime

#### 目的

軸の移動時間を取得します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_move_time [out]	軸の移動時間を受け取るバッファーへのポインター パラメーター単位：ms

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

3.3.18 HIMC\_GetSettlingTime

目的

軸の整定時間を取得します

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index
- p\_settling\_time [out]

軸の整定時間を受け取るバッファーへのポインター  
パラメーター単位：ms

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

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



### 3.3.19 HIMC\_SetPos

#### 目的

軸の位置を設定し、原点オフセットを変更します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
pos [in]	軸の現在位置の値 パラメーター単位：mm または deg

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 備考

軸が「同期」状態、軸グループに追加された状態、またはエラー状態にある場合、この機能は適用されません。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_SetPos
LabVIEW	HIMC Set Pos.vi
Python	SetPos

## 3.3.20 HIMC\_GetPosFb

### 目的

軸のフィードバック位置を取得します

### 構文

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

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]      Axis index
- p\_pos [out]      軸のフィードバック位置を受け取るバッファへのポインター  
パラメーター単位：mm または deg

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

### 3.3.21 HIMC\_GetPosOffset

#### 目的

軸の位置オフセットを取得します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_pos [out]	軸の位置オフセットを受け取るバッファーへのポインター パラメーター単位：mm または deg

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 3.3.22 HIMC\_GetPosErr

### 目的

軸の位置誤差を取得します

### 構文

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

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]      Axis index
- p\_pos\_err [out]   軸の位置誤差を受け取るバッファへのポインター  
パラメーター単位：mm または deg

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

### 3.3.23 HIMC\_GetVelFb

#### 目的

軸の速度フィードバックを取得します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_vel [out]	軸の速度フィードバックを受け取るバッファへのポインター パラメーター単位：mm/s または deg/s

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

3.3.24 HIMC\_GetVelErr

目的

軸の速度誤差を取得します

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index
- p\_vel\_err [out]

軸の速度エラーを受け取るバッファへのポインター  
パラメーター単位：mm/s または deg/s

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

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

### 3.3.25 HIMC\_GetCurrFb

#### 目的

軸の現在のフィードバック(二乗値)を取得します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_curr [out]	軸の現在のフィードバック(二乗値)を受け取るバッファへのポインター パラメーター単位：A <sup>2</sup>

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 備考

現在のフィードバックを取得するには、まず「CurABS」を PDO 通信オブジェクトとして構成する必要があります(「iA Studio ユーザーガイド」のセクション 4.13 を参照)。PDO オブジェクトが構成されていない場合、「\*p\_curr」の値は 0 になります。

#### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetCurrFb
LabVIEW	HIMC Get Curr Fb.vi
Python	GetCurrFb

3.3.26 HIMC\_GetRefPos

目的

軸の基準位置を取得します

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index
- p\_pos [out]

軸の基準位置を受け取るバッファーへのポインター  
パラメーター単位：mm または deg

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

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



### 3.3.27 HIMC\_GetRefVel

#### 目的

軸の基準速度を取得します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_vel [out]	軸の基準速度を受け取るバッファへのポインター パラメーター単位：mm/s または deg/s

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 3.3.28 HIMC\_GetRefAcc

### 目的

軸の基準加速度を取得します

### 構文

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

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]      Axis index
- p\_acc [out]      軸の基準加速度を受け取るバッファへのポインター  
パラメーター単位：mm/s<sup>2</sup> または deg/s<sup>2</sup>

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

### 3.3.29 HIMC\_GetPosOut

#### 目的

コントローラーからスレーブドライバーに送信された軸の位置コマンド出力を取得します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_pos [out]	軸の位置コマンド出力を受け取るバッファーへのポインター パラメーター単位：mm または deg

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 3.3.30 HIMC\_GetVelOut

### 目的

コントローラーからスレーブドライバーに送信された軸の速度コマンド出力を取得します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_vel [out]	軸の速度コマンド出力を受け取るバッファへのポインター パラメーター単位：mm/s または deg/s

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

### 3.3.31 HIMC\_GetAccOut

#### 目的

コントローラーからスレーブドライバーに送信された軸の加速度コマンド出力を取得します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_acc [out]	軸の加速度コマンド出力を受け取るバッファーへのポインター パラメーター単位：mm/s <sup>2</sup> または deg/s <sup>2</sup>

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

3.3.32 HIMC\_IgnoreHWL

目的

ハードウェア制限保護の警告メッセージを無視します。

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index
- cmd [in]

メッセージを無視するには、「1」に設定します。  
メッセージを有効にするには、「0」に設定します（初期値）。

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_IgnoreHWL
LabVIEW	HIMC Ignore HWL.vi
Python	IgnoreHWL

### 3.3.33 HIMC\_IgnoreSWL

#### 目的

ソフトウェア制限保護の警告メッセージを無視します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
cmd [in]	メッセージを無視するには、「1」に設定します。 メッセージを有効にするには、「0」に設定します（初期値）。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_IgnoreSWL
LabVIEW	HIMC Ignore SWL.vi
Python	IgnoreSWL

## 3.3.34 HIMC\_IgnorePE

### 目的

位置偏差制限の警告メッセージを無視します。

### 構文

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

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]      Axis index
- cmd [in]          メッセージを無視するには、「1」に設定します。  
メッセージを復元するには、「0」に設定します（初期値）。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_IgnorePE
LabVIEW	HIMC Ignore PE.vi
Python	IgnorePE



### 3.3.35 HIMC\_GetAxisNum

#### 目的

コントローラーに接続されている軸数を取得します。

#### 構文

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

#### パラメーター

- ctrl\_id [in]            HIWIN モーションコントローラーのコントローラーID  
                         HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- num [out]              コントローラーに接続されている軸の数を受け取るバッファへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

3.3.36 HIMC\_SetVelScale

目的

軸運動の速度スケールを設定します

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index
- vel\_scale [in]

軸運動の新しい速度スケール  
入力範囲：0 ～ 100

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

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

### 3.3.37 HIMC\_GetVelScale

#### 目的

軸運動の速度スケールを取得します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_vel_scale [out]	軸の動きの速度スケールを受け取るバッファへのポインター その範囲は 0 ~ 100 です。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 3.3.38 HIMC\_SetRollover

### 目的

軸の位置ロールオーバー値を設定します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
rollover_val [in]	軸の位置ロールオーバー値 パラメーター単位：mm または deg 入力範囲：ゼロ以外の正の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

- (1) パラメーター「rollover\_val」が 0 に設定されている場合、関数はクローズされます。
- (2) この機能は、軸が無効になっている場合にのみ適用されます。
- (3) 軸が軸グループに追加されている場合、この機能は適用されません。

### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetRollover
LabVIEW	HIMC Set Rollover.vi
Python	SetRollover

### 3.3.39 HIMC\_GetRolloverTurns

#### 目的

軸がロールオーバーモードのときに回転数を取得します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_turns [out]	軸がロールオーバーモードのときに回転数を受け取るバッファへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 3.3.40 HIMC\_GetMultiAxesFeedbackPos

### 目的

軸リストのフィードバック位置を取得します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
p_axes_id_array [in]	軸 ID のリストを格納するバッファへのポインター
num_of_axes [in]	軸数
p_pos_array [out]	軸のリストのフィードバック位置を受け取るバッファへのポインター パラメーター単位：mm または deg

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 3.4 軸の状態

### 3.4.1 HIMC\_IsEnabled

#### 目的

軸の「有効」ステータスを問い合わせます

#### 構文

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

#### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]      Axis index
- p\_enabled [out]    軸の有効ステータスを受け取るバッファへのポインター  
軸が「有効」状態の場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_IsEnabled
LabVIEW	HIMC Is Enabled.vi
Python	IsEnabled

## 3.4.2 HIMC\_IsMoving

### 目的

軸の「移動」ステータスを照会します。軸が移動している場合、PG(プロファイルジェネレーター)は新しい位置を出力し続けます。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_moving [out]	軸の移動状態を受け取るバッファーへのポインター 軸が「Moving」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_IsMoving
LabVIEW	HIMC Is Moving.vi
Python	IsMoving



### 3.4.3 HIMC\_IsInPos

#### 目的

軸の「インポジション」ステータスを照会します。軸が所定の位置にある場合、位置エラーは、特定の期間(デバウンス時間)の間、エラーウィンドウ(ターゲット半径)内に保持されます。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_in_position [out]	軸のインポジションステータスを受け取るバッファーへのポインター 軸が「InPos」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

3.4.4 HIMC\_IsErrorStop

目的

軸が「エラー停止」状態かどうかを問い合わせます

構文

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

パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_errorstop [out]	軸のエラー停止ステータスを受け取るバッファーへのポインター 軸が「ErrorStop」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

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

### 3.4.5 HIMC\_IsGantry

#### 目的

軸が「ガントリー」状態にあるかどうかを照会します

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_gantry [out]	軸のガントリーステータスを受け取るバッファへのポインター 軸が「ガントリー」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_IsGantry
LabVIEW	HIMC Is Gantry.vi
Python	IsGantry

## 3.4.6 HIMC\_IsGrouped

### 目的

軸が軸グループにグループ化されているかどうかを照会します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_grouped [out]	軸のグループ化されたステータスを受け取るバッファへのポインター 軸が「グループ化」状態にある場合、値は 1 になります。それ以外の場合、 値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_IsGrouped
LabVIEW	HIMC Is Grouped.vi
Python	IsGrouped

### 3.4.7 HIMC\_IsSync

#### 目的

軸が「同期」状態にあるかどうかを問い合わせます

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_sync [out]	軸の同期ステータスを受け取るバッファへのポインター 軸が「同期」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_IsSync
LabVIEW	HIMC Is Sync.vi
Python	IsSync

3.4.8 HIMC\_IsHWLL

目的

軸がハードウェアの左リミットに達しているかどうかを照会します。

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index
- p\_is\_hwll [out]

軸の HWLL ステータスを受け取るバッファーへのポインター  
軸が「HWLL」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_IsHWLL
LabVIEW	HIMC Is HWLL.vi
Python	IsHWLL

### 3.4.9 HIMC\_IsHWRL

#### 目的

軸がハードウェアの右リミットに達しているかどうかを照会します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_hwrl [out]	軸の HWRL ステータスを受け取るバッファーへのポインター 軸が「HWRL」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_IsHWRL
LabVIEW	HIMC Is HWRL.vi
Python	IsHWRL

3.4.10 HIMC\_IsSWLL

目的

軸がソフトウェアの左リミットに達しているかどうかを問い合わせます。

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index
- p\_is\_swll [out]

軸の SWLL ステータスを受け取るバッファーへのポインター  
軸が「SWLL」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_IsSWLL
LabVIEW	HIMC Is SWLL.vi
Python	IsSWLL



### 3.4.11 HIMC\_IsSWRL

#### 目的

軸がソフトウェアの右リミットに達しているかどうかを問い合わせます。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_swrl [out]	軸の SWRL ステータスを受け取るバッファへのポインター 軸が「SWRL」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_IsSWRL
LabVIEW	HIMC Is SWRL.vi
Python	IsSWRL

3.4.12 HIMC\_IsDriveErr

目的

軸がドライバーアラームをトリガーするかどうかを問い合わせます

構文

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

パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_driveerr [out]	軸の DriveErr ステータスを受け取るバッファーへのポインター 軸が「DriveErr」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

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

### 3.4.13 HIMC\_IsPosErr

#### 目的

軸の位置誤差が保護限界を超えているかどうかを問い合わせます

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_poserr [out]	軸の PosErr ステータスを受け取るバッファーへのポインター 軸が「PosErr」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 備考

エラー保護限界は、コントローラー内の軸の位置エラーの許容範囲を示します。

#### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

3.4.14 HIMC\_IsCompActive

目的

補正機能が有効かどうかを問い合わせます

構文

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

パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_compactive [out]	軸の補正アクティブステータスを受け取るバッファーへのポインター 軸が「CompActive」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

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

### 3.4.15 HIMC\_IsAcc

#### 目的

軸が加速しているかどうかを問い合わせます

#### 構文

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

#### パラメーター

- ctrl\_id [in]** HIWIN モーションコントローラーのコントローラー ID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]** Axis index
- p\_is\_acc [out]** 軸の加速ステータスを受け取るバッファーへのポインター  
軸が「Acc」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_IsAcc
LabVIEW	HIMC Is Acc.vi
Python	IsAcc

(このページはブランクになっています)

## 4. シンクロモーション機能

4.1	概要 .....	4-2
4.1.1	シンクロモーション変数 .....	4-2
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 概要

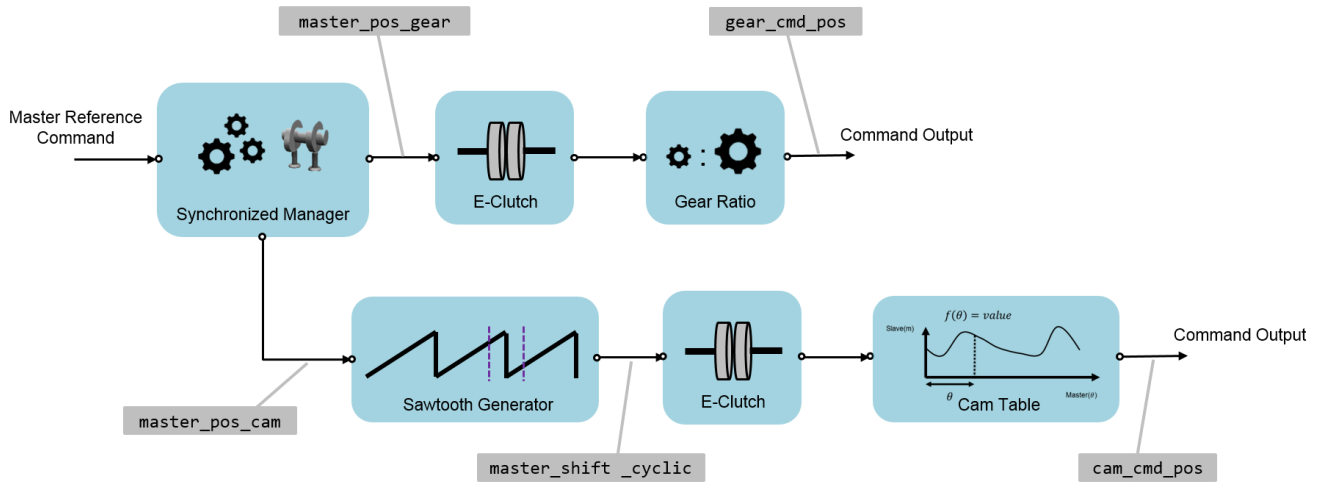


図 4.1.1

ユーザーは、ある軸と別の軸の間の同期モーションを定義できます。先行軸であるマスター軸が最初に位置コマンドを生成し、次にスレーブ軸がモーション構成に基づいてマスター軸を参照します。主従関係が一定であれば、動きは電子ギアです。一方、スレーブ軸がパターンに従う必要がある場合、モーションは電子カムです。図 4.1.2 では、軸 0 がマスター軸として機能し、軸 1、2、3、および 4 が先行します。軸 1、2、および 3 は電子ギアを採用し、軸 4 は電子カムを採用しています。

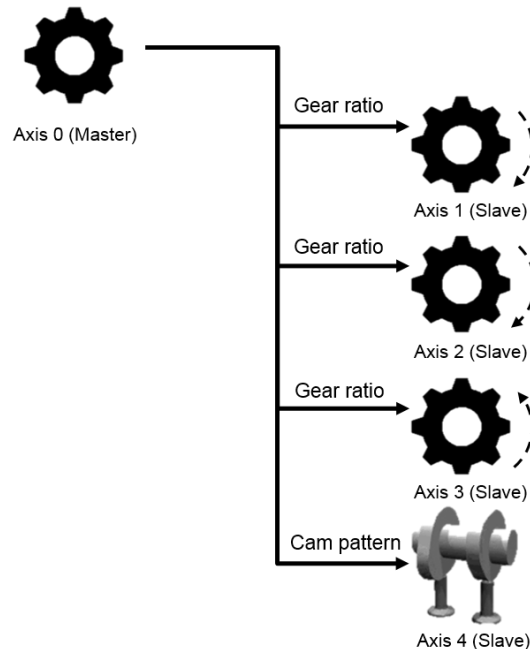


図 4.1.2

### 4.1.1 シンクロモーション変数

一般的な同期モーション変数を表 4.1.1.1 に示します。ユーザーは、iA Studio のスコープマネージャーを介して目的の変数を選択できます(「iA Studio ユーザーガイド」のセクション 4.8 を参照)。



表 4.1.1.1

名称	変数	単位	説明
Raw Master Position	master_pos_gear	mm または deg	マスター軸の位置指令
Gear Command Position	gear_cmd_pos	mm または deg	スレーブ軸の位置指令出力
Gear Ratio	gear_ratio	mm または deg	ギア比

4.2 HIMC\_EnableGear

目的

2つの軸をマスター、スレーブ関係で結合します

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_master\_id [in]

マスターAxis index
- axis\_slave\_id [in]

スレーブ Axis index

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

備考

この機能は、両方の軸が有効な場合にのみ適用されます。

要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_EnableGear
LabVIEW	HIMC Enable Gear.vi
Python	EnableGear

## 4.3 HIMC\_DisableGear

### 目的

マスター、スレーブ関係から 2 つの軸を切り離して、2 つの独立した軸にします

### 構文

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

### パラメーター

ctrl\_id [in]                      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

axis\_slave\_id [in]              スレーブ Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_DisableGear
LabVIEW	HIMC Disable Gear.vi
Python	DisableGear

## 4.4 HIMC\_GearIn

### 目的

スレーブ軸の状態を解放から使用に変更します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_master_id [in]	マスターAxis index
axis_slave_id [in]	スレーブ Axis index
gear_ratio [in]	ギア比の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

この機能は、両方の軸が有効な場合にのみ適用されます。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GearIn
LabVIEW	HIMC Gear In.vi
Python	GearIn

## 4.5 HIMC\_GearOut

### 目的

スレーブ軸の状態をエンゲージからディスエンゲージに変更します

### 構文

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

### パラメーター

ctrl\_id [in]                      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

axis\_slave\_id [in]              スレーブ Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GearOut
LabVIEW	HIMC Gear Out.vi
Python	GearOut

## 4.6 HIMC\_GetGearRatio

### 目的

スレーブ軸のギア比を取得します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_slave_id [in]	スレーブ Axis index
p_ratio [out]	スレーブ軸のギア比を受け取るバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 4.7 HIMC\_IsInGear

### 目的

スレーブ軸が「エンゲージ」状態にあるかどうかを問い合わせます

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_in_gear [out]	スレーブ軸の有効ステータスを受け取るバッファへのポインター スレーブ軸が「InGear」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 4.8 HIMC\_IsGearMaster

### 目的

軸がマスター軸かどうかを問い合わせます

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_gear_master [out]	軸がマスター軸かどうかのステータスを受け取るバッファーへのポインター 軸が「GearMaster」状態にある場合、値は 1 になります。それ以外の場合、 値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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



## 4.9 HIMC\_IsGearSlave

### 目的

軸がスレーブ軸かどうかを問い合わせます

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_gear_slave [out]	スレーブ軸かどうかのステータスを受け取るバッファへのポインター 軸が「GearSlave」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

(このページはブランクになっています)

# 5. ガントリー機能

5.1	概要 .....	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 概要

ガントリー構成は、図 5.1.1 に示すように、右側(RHS)軸と左側(LHS)軸のペアを仮想直線軸とヨー軸のペアに変換します。ガントリー構成を確立した後、ユーザーは、RHS 軸に直線軸方向コマンドを与えて、RHS 軸と LHS 軸の両方を同じ方向に駆動し、LHS 軸にヨー軸方向の回転運動コマンドを与えることができます。

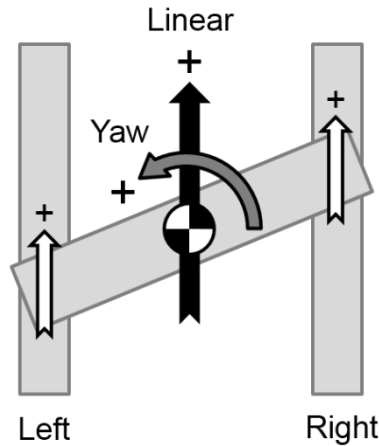


図 5.1.1

ガントリー構成では、直線軸とヨー軸の位置フィードバックは次のように定義されます。

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

$Pos_{linear}$  : 直線軸の位置フィードバック       $Pos_{yaw}$  : ヨー軸の位置フィードバック

$Pos_{RHS}$  : 右軸の位置フィードバック       $Pos_{LHS}$  : 左軸の位置フィードバック

図 5.1.2 は、リニア軸、ヨー軸、RHS 軸、LHS 軸の位置フィードバックの回路図です。

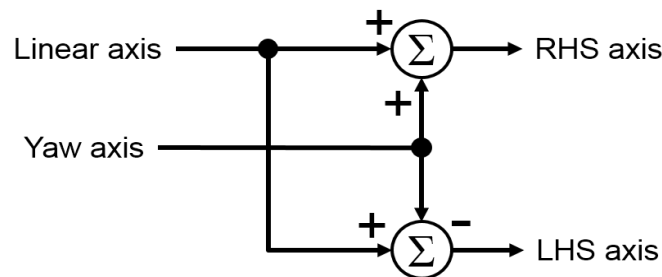


図 5.1.2

## 5.2 HIMC\_EnableGantryPair

### 目的

ガントリーペアをセットアップします

### 構文

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

### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

lhs\_axis\_id [in]    左側 Axis index

rhs\_axis\_id [in]    右側 Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

この機能は、両方の軸が無効になっている場合にのみ適用されます。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 5.3 HIMC\_DisableGantryPair

### 目的

ガントリーペアを分割します

### 構文

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

### パラメーター

**ctrl\_id [in]**           HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**axis\_id [in]**           ガントリーペアのいずれかの Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

この機能は、両方の軸が無効になっている場合にのみ適用されます。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 5.4 HIMC\_GetGantryPairID

### 目的

任意のガントリー軸のガントリーペア ID を取得します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	ガントリーペアのいずれかの Axis index
p_id [out]	ガントリーペア ID を受け取るバッファーへのポインター 入力軸がガントリー軸でない場合は-1 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 5.5 HIMC\_IsGantryPair

### 目的

2つの軸がガントリーペアであるかどうかを照会します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id_1 [in]	Axis index 1
axis_id_2 [in]	Axis index 2
p_is_gantry_pair [out]	2つの軸がガントリーペアであるかどうかのステータスを受け取るバッファへのポインター。2つの軸が「GantryPair」状態にある場合、値は1になります。それ以外の場合、値は0になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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



## 6. グループ機能

6.1	概要 .....	6-3
6.1.1	グループ変数.....	6-6
6.1.2	座標系 .....	6-9
6.1.3	運動学 .....	6-13
6.1.4	バッファモード .....	6-13
6.1.5	Transition モード .....	6-15
6.2	グループモーションコントロール.....	6-17
6.2.1	HIMC_EnableGroup .....	6-17
6.2.2	HIMC_DisableGroup .....	6-18
6.2.3	HIMC_ResetGroup .....	6-19
6.2.4	HIMC_StopGroup .....	6-20
6.2.5	HIMC_HaltGroup .....	6-21
6.2.6	HIMC_ResumeGroup.....	6-22
6.2.7	HIMC_JogGroup.....	6-23
6.2.8	HIMC_JogGroupAxis .....	6-24
6.2.9	HIMC_LineAbs2D .....	6-25
6.2.10	HIMC_LineAbs3D .....	6-26
6.2.11	HIMC_LineRel2D .....	6-27
6.2.12	HIMC_LineRel3D .....	6-28
6.2.13	HIMC_Arc2D .....	6-29
6.2.14	HIMC_ArcCW2D.....	6-31
6.2.15	HIMC_ArcCCW2D.....	6-32
6.2.16	HIMC_Circle2D .....	6-33
6.2.17	HIMC_ArcAngle2D .....	6-35
6.3	グループ設定 .....	6-37
6.3.1	HIMC_AddAxesToGrp.....	6-37
6.3.2	HIMC_RemoveAxisFromGrp .....	6-38
6.3.3	HIMC_SetupGroup .....	6-39
6.3.4	HIMC_UngrpAllAxes .....	6-40
6.3.5	HIMC_GetGroupID .....	6-41
6.3.6	HIMC_SetGrpMotionProfile.....	6-42
6.3.7	HIMC_SetGrpAngMotionProfile .....	6-44
6.3.8	HIMC_GetGrpKin.....	6-46
6.3.9	HIMC_SetGrpKin .....	6-47
6.3.10	HIMC_GetGrpMaxVel .....	6-48
6.3.11	HIMC_SetGrpVel .....	6-49
6.3.12	HIMC_GetGrpMaxAcc .....	6-50
6.3.13	HIMC_SetGrpAcc .....	6-51
6.3.14	HIMC_SetGrpAccTime.....	6-52
6.3.15	HIMC_GetGrpMaxDec.....	6-53
6.3.16	HIMC_SetGrpDec.....	6-54
6.3.17	HIMC_SetGrpDecTime .....	6-55
6.3.18	HIMC_GetGrpSMTime.....	6-56

6.3.19	HIMC_SetGrpSMTTime .....	6-57
6.3.20	HIMC_GetGrpCoordSys .....	6-58
6.3.21	HIMC_SetGrpCoordSys .....	6-59
6.3.22	HIMC_GetGrpBufferMode .....	6-60
6.3.23	HIMC_SetGrpBufferMode .....	6-61
6.3.24	HIMC_GetGrpTransMode .....	6-62
6.3.25	HIMC_SetGrpTransMode .....	6-63
6.3.26	HIMC_SetGrpTransPrm.....	6-64
6.3.27	HIMC_GetGrpCmdNum.....	6-65
6.3.28	HIMC_SetGrpVelScale .....	6-66
6.3.29	HIMC_GetGrpVelScale .....	6-67
6.3.30	HIMC_GetGrpCoordTrans.....	6-68
6.3.31	HIMC_SetGrpCoordTrans .....	6-69
6.3.32	HIMC_GetGrpPoseCmd .....	6-70
6.3.33	HIMC_GetGrpPoseFb.....	6-71
6.4	グループステータス .....	6-72
6.4.1	HIMC_IsGrpEnabled .....	6-72
6.4.2	HIMC_IsGrpMoving.....	6-73
6.4.3	HIMC_IsGrpInPos .....	6-74
6.4.4	HIMC_IsGrpErrorStop.....	6-75
6.5	高度なグループモーションコントロール .....	6-76
6.5.1	HIMC_LinAbs .....	6-76
6.5.2	HIMC_LinRel.....	6-78
6.5.3	HIMC_CircAbs .....	6-80
6.5.4	HIMC_CircRel .....	6-82

## 6.1 概要

HIMC は、LineAbs2D / 3D、LineRel2D / 3D、Arc2D、Circle2D などの多軸直線および円弧同時補間機能の軸グループモーションコマンドを提供します。軸モーションコマンドと比較して、軸グループモーションコマンドは、各軸の同期を確実にします。各軸のモーションの開始時間と停止時間は同じであり、コントローラーは、ユーザーが指定した基準速度に基づいて各軸のモーション速度を調整します。

図 6.1.1 に HIMC 軸グループモーションコマンドのパラメータフロー図を示します。各軸の位置フィードバックは、順運動学の計算を通過するため、機械座標系での軸グループの位置フィードバック(直交位置フィードバック)が取得されます。ユーザーによって与えられたターゲットコマンドに基づいて、コントローラーは、軸グループのモーションプロファイル(図 6.1.2 が示すように)による空間内の補間コマンド(デカルト位置コマンド)を計画し、逆運動学を使用して各軸の対応する位置コマンドを計算します。

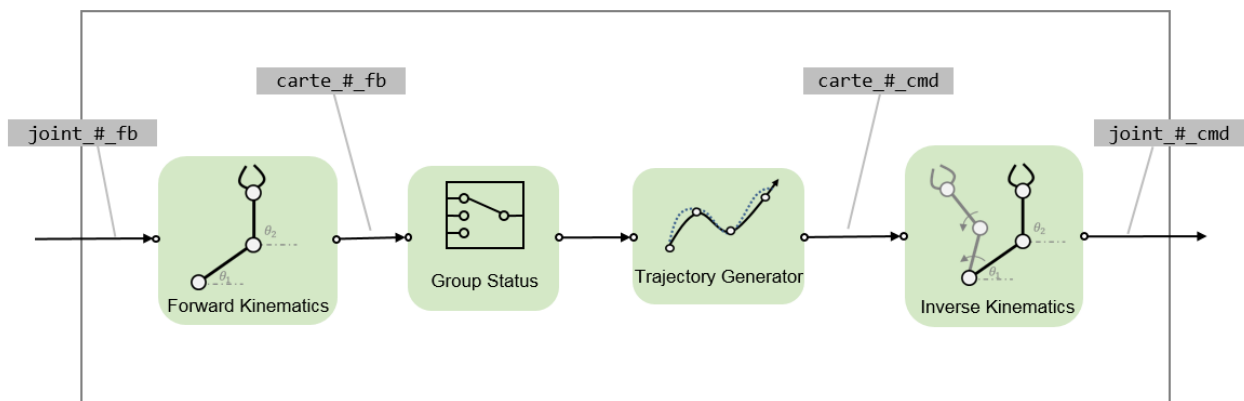


図 6.1.1

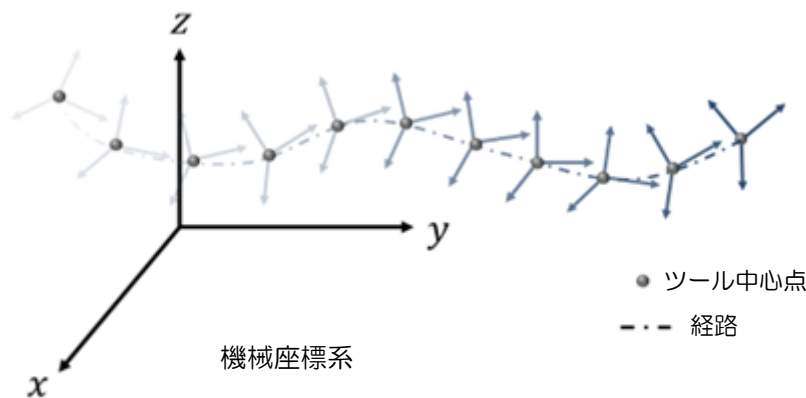


図 6.1.2

軸グループモーションコマンドでは、HIMC は空間内の各セグメントの移動距離を計算します。軸移動コマンドとは異なり、速度計画は空間内の軸グループの移動方向に沿って計画され、移動方向は移動コマンドの方向に基づいて変化します。

軸グループモーションコマンドは、軸モーションコマンドに似ています。また、図 6.1.3 に示すように、Sカーブ速度計画も採用しています。空間内の軸群の動きは、並進と回転の 2 つの部分で構成されます。並進指令は XYZ の位置指令で構成され、回転指令は ABC の回転指令で構成されます。軸グループを使用すると、ユーザーは、プロファイルジェネレーターの最大速度、最大加速度、最大減速度、滑らかな時間を含む、平行移動と回転の速度計画パラメーターを設定できます。

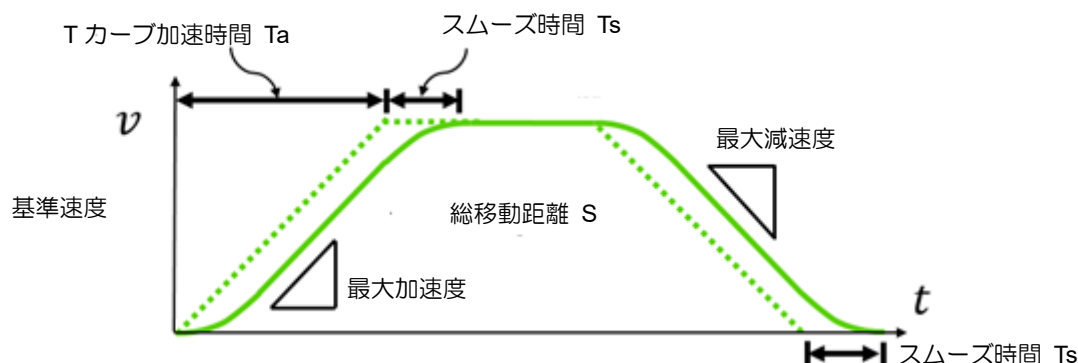


図 6.1.3

図 6.1.4 に示すように、各軸グループのモーションコマンドは 1 つのセグメントとして表示されます。動作中、各セグメントの並進コマンドと回転コマンド、およびユーザーが設定した速度計画パラメーターに基づいて、HIMC は並進コマンドと回転コマンドの移動時間を計算します。長い移動時間の速度計画パラメーターは、軸グループの送り速度として表示されます。移動時間の短い方は、送り速度指令で配分された動作指令に従って移動します。

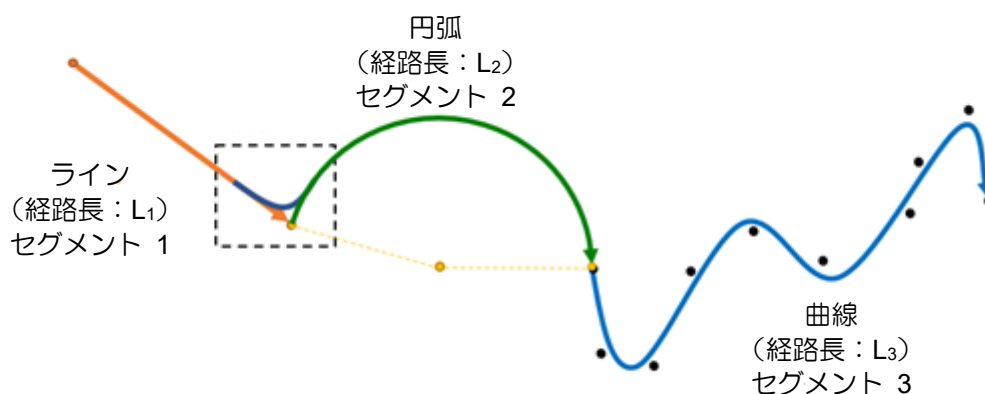


図 6.1.4

HIMC には、軸グループコマンド用の組み込みバッファがあります。各モーションコマンドのセグメントは、このバッファに配置されます。同時に最大 512 個のモーションコマンドを受け入れることができます。この容量制限を超えるモーションコマンドはコントローラーによって破棄され、エラーメッセージが表示されます。セグメントのモーションコマンド間では、ユーザーが設定したバッファモードとトランジションモードに基づいて、速度とパスが計画されます。選択したモードによって、計画された速度と経路が変更される場合があります。図 6.1.4 を例にとると、各セグメントの速度ハンドオー

バーを設定するためにバッファモードが使用される場合、この軸モーションの全長は「 $S=L_1$  (直線)+ $L_2$  (円弧)+ $L_3$  (曲線)」になります。詳細については、セクション 6.1.4 および 6.1.5 を参照してください。

軸グループのモーションステータスは、軸のモーションステータスに似ています。また、「移動」と「定位置」に分けることもできます。動作中は、図 3.1.4 に示すように、以下を含む 3 つのフェーズがあります：

1. 軸グループが動いていて、位置が合っていない
2. 軸グループは動いていませんが、位置が合っていない
3. 軸グループは移動しておらず、所定の位置にあります

ターゲット半径とデバウンス時間を使用して軸がインポジションかどうかをチェックする軸モーションコマンドとは異なり、軸グループモーションコマンドは、グループ内のすべての軸がインポジションかどうかをチェックします。つまり、軸グループがインポジションの場合、グループ内のすべての軸が「インポジション」状態になります。

## 6.1.1 グループ変数

軸グループ変数は、モーションコマンド変数、プロファイルジェネレーター変数、ステータス変数の3つのカテゴリに分類されます。ユーザーは、iA Studio のスコープマネージャーを介して目的の変数を選択できます(「iA Studio ユーザーガイド」のセクション 4.8 を参照)。詳細な説明を表 6.1.1.1 から表 6.1.1.5 に示します。

表 6.1.1.1 軸グループのモーションコマンド変数

名称	変数	単位	説明
Cartesian Position Command	carte_pose_cmd	mm または deg	機械座標系 (MCS) の軸グループの空間位置コマンド。[X Y Z A B C] の値を含む配列です。
Cartesian Velocity Command	carte_vel_cmd	mm/s または deg/s	機械座標系 (MCS) の軸グループの空間速度コマンド。[X Y Z A B C] の値を含む配列です。
Cartesian Position Feedback	carte_pose_fb	mm または deg	機械座標系 (MCS) の軸グループの空間位置フィードバック。[X Y Z A B C] の値を含む配列です。
Axis Position Command	joint_pos_cmd	mm または deg	軸座標系(ACS)での軸グループの軸位置指令の配列です。
Axis Velocity Command	joint_vel_cmd	mm/s または deg/s	軸座標系(ACS)での軸グループの軸速度コマンドの配列です。
Axis Acceleration Command	joint_acc_cmd	mm/s <sup>2</sup> または deg/s <sup>2</sup>	軸座標系(ACS)の軸グループの軸加速度コマンドの配列です。
Axis Position Feedback	joint_pos_fb	mm または deg	軸座標系(ACS)での軸グループの軸位置フィードバックの配列です。
Cartesian Position Error	carte_pose_err	mm または deg	機械座標系(MCS)の軸グループの空間位置エラー。[X Y Z A B C] の値を含む配列です。
Reference Group Position	grp_pg_pos	mm または deg	軸グループの基準位置。これは、軸グループコマンドのモーションプロファイルに従って、プロファイルジェネレーターから生成された位置セットポイントです。
Reference Group Velocity	grp_pg_vel	mm/s または deg/s	軸グループの基準速度。これは、軸グループコマンドのモーションプロファイルに従って、プロファイルジェネレーターから生成された速度設定値です。
Reference Group Acceleration	grp_pg_acc	mm/s <sup>2</sup> または deg/s <sup>2</sup>	軸グループの基準加速度。これは、軸グループコマンドのモーションプロファイルに従って、プロファイルジェネレーターから生成された加速度設定値です。

表 6.1.1.2 軸グループのプロファイルジェネレーター変数

名称	変数	単位	説明
Group Max. Linear Profile Velocity	grp_lin_vel	mm/s	軸グループの最大線形プロファイル速度。 必ずしも達していない。
Group Max. Linear Profile Acceleration	grp_lin_acc	mm/s <sup>2</sup>	軸グループの最大線形プロファイル加速度。 必ずしも達していない。
Group Max. Linear Profile Deceleration	grp_lin_dec	mm/s <sup>2</sup>	軸グループの最大線形プロファイル減速度。 必ずしも達していない。
Group Linear Smooth Time	grp_lin_sf	ms	軸グループの線形プロファイルの平滑化時間。入力範囲は 0 ～ 500 です。値を大きくすると、モーション中の機械的振動を減らすことができますが、合計モーション時間は影響を受けます。
Group Max. Angular Profile Velocity	grp_ang_vel	deg/s	軸グループの最大角プロファイル速度。 必ずしも達していない。
Group Max. Angular Profile Acceleration	grp_ang_acc	deg/s <sup>2</sup>	軸グループの最大角度プロファイル加速度。 必ずしも達していない。
Group Max. Angular Profile Deceleration	grp_ang_dec	deg/s <sup>2</sup>	軸グループの最大角度プロファイル減速。 必ずしも達していない。
Group Angular Smooth Time	grp_ang_sf	ms	軸グループの角度プロファイルの平滑化時間。入力範囲は 0 ～ 500 です。値を大きくすると、モーション中の機械的振動を減らすことができますが、合計モーション時間は影響を受けます。

表 6.1.1.3 軸グループのステータス変数

名称	変数	単位	説明
Group Fault Status	grp_fault_status	N/A	軸グループのエラーステータス：ビット定義については、表 6.1.1.4 を参照してください。
Group Motion Status	grp_motion_status	N/A	軸グループのモーションステータス：ビット定義については、表 6.1.1.5 を参照してください。

表 6.1.1.4 軸グループエラーステータスのビット定義

Bit	名称	説明	デフォルトの応答
0	Error Stop	Axis group at “error stop” state	N/A
1	Axis Fault	Slave drive fault	コントローラーは軸を無効にします。 軸グループが同期していません。
2	Software Limit	Axis software limit triggered	コントローラーがモーションを停止します。 軸グループが同期していません。

表 6.1.1.5 軸群モーションステータスのビット定義

Bit	名称	説明	備考
0	Enabled	軸グループが有効になります	N/A
1	Moving	軸群が動いています	N/A
2	In Position	軸グループはインポジションです	軸グループ内のすべての軸がインポジションです。
3	Input Shape	軸グループの入力形状フィルターを有効にします	セクション 13.1 を参照してください



## 6.1.2 座標系

表 6.1.2.1 は、軸座標系(ACS)、機械座標系(MCS)、製品座標系(PCS)、ワークピース座標系(WCS)、グローバル座標系、および座標オフセットを含む、HIMC の座標系の定義と説明を示しています。

表 6.1.2.1

HMPL 定義	説明
CS_ACS	軸座標系。 これは、個々のモーターの動きに関連しています。
CS_MCS	機械座標系。 (「ワールド座標系」または「ベース座標系」と呼ばれることもあります) これは機械の原点が固定された座標系であり、キネマティクス変換を介して ACS にリンクされています (セクション 6.1.3 を参照)。空間内の位置と方向を示すために、合計で 6 つの次元があります (3 つの並進、3 つの回転)。
CS_PCS	製品座標系 (CNC プログラムでは「プログラム座標系」)。 製品やワークに取り付けて、座標変換パラメーターを設定できます。
CS_WCS# (#=1~15)	ワーク座標系。 ワーク原点を設定するために使用され、最大 15 の独立したワーク座標系を提供します。デフォルトはオフセットなしです。Product Coordinate System に依存するため、座標変換パラメーターを設定できます。
CS_GLOBAL	グローバル座標系。 グローバルなゼロ点を設定するために使用され、各軸グループのグローバルな空間関係を確立できます。 まだサポートされていません。
CS_OFFSET	座標オフセット。 仮のゼロ点設定に使用します。デフォルトはオフセットなしです。つまり、オフセットの座標原点はマシンの座標原点です。Product Coordinate System に依存するため、座標変換パラメーターを設定できます。

図 6.1.2.1 に回転軸 2 軸のスカラロボットの ACS、MCS、PCS の関係例を示します。ACS と MCS は、順運動学と逆運動学によって変換されます（セクション 6.1.3 を参照）。一方、MCS と PCS には座標変換関係があります。座標系上の位置は、座標の並進と回転によって取得されます

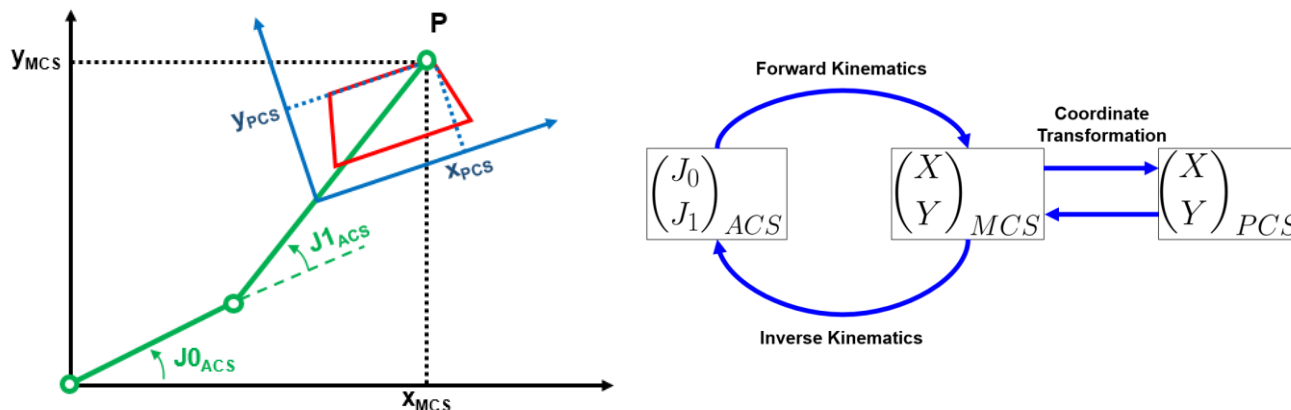


図 6.1.2.1

MCS を PCS に変換する場合、HIMC は必要に応じてマシンのワーク座標(WCS1~15)と座標オフセット (OFFSET)を設定できます。座標系の設定には、並進 3 自由度(X、Y、Z)と回転 3 自由度(A、B、C)を使用して空間での姿勢を示します。

HIMC は、固定角度で「Roll-Pitch-Yaw」回転規則を採用しています。図 6.1.2.2 が示すように、X 軸に沿って回転する自由度はロール、角度 A です。Y 軸に沿って回転する自由度はピッチ、角度 B です。Z 軸に沿って回転する自由度は、Yaw、角度 C です。この回転規則は、図 6.1.2.3 に示すように、スペース内のオブジェクトの向きを示すためにテイトブライアン角度で ZYX のシーケンスを使用するのと同じです。

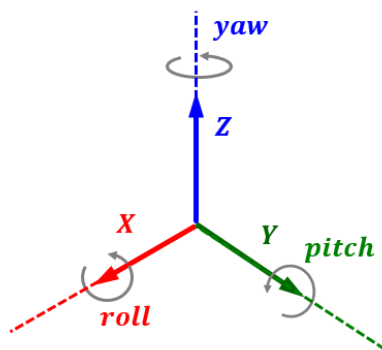


図 6.1.2.2

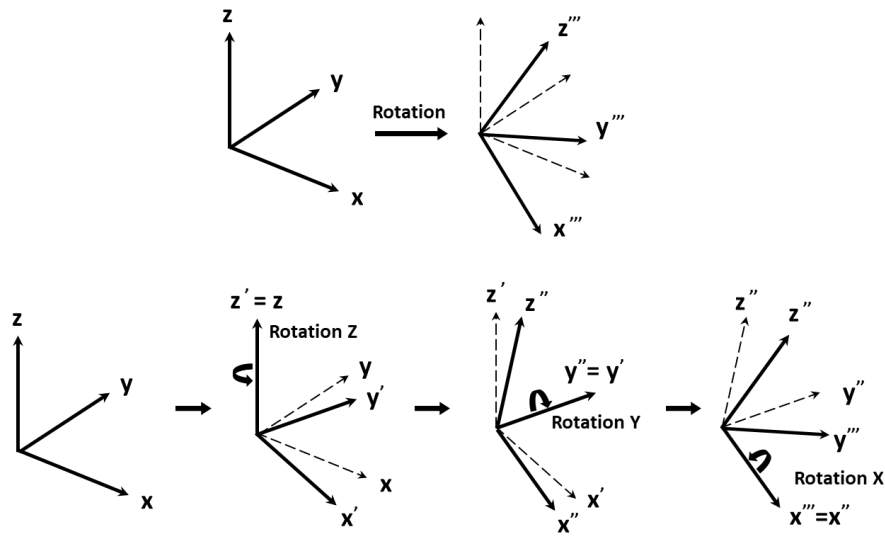


図 6.1.2.3

座標オフセットがない場合、各 WCS と MCS の関係は図 6.1.2.4 のようになります。

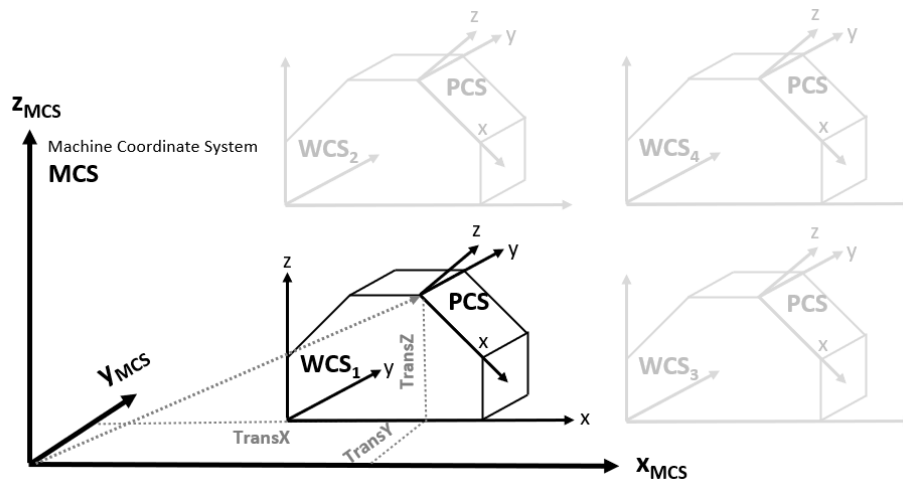
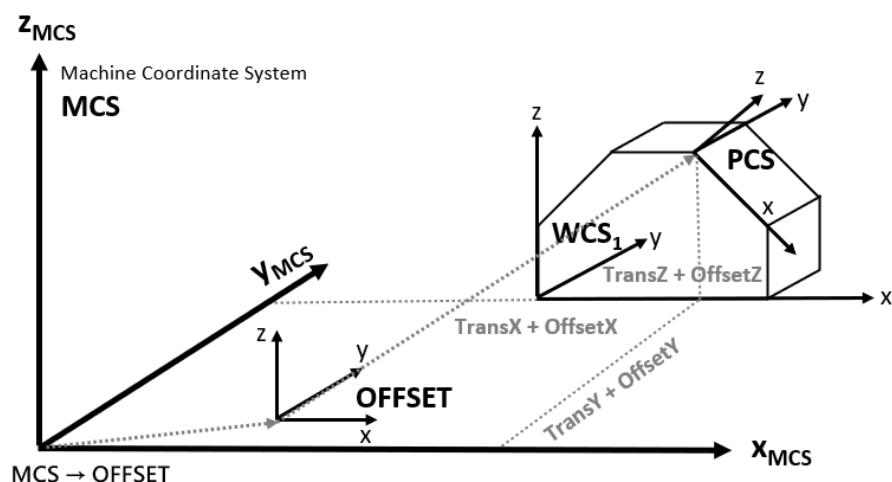


図 6.1.2.4

座標オフセットを加えると、WCS と MCS の関係は図 6.1.2.5 のようになります。座標オフセットの変換が追加されます。



### 图 6.1.2.5

上記の機能に基づいて、ユーザーは HIMC で座標変換のパラメーターを定義し、座標系間の変換関係を確立できます。図 6.1.2.6 に座標系の関係を示します。わかりやすくするために、図では XY 平面の座標のみを示しています。実際のアプリケーションでは、ユーザーは座標変換に 6 つの自由度 (X、Y、Z、A、B、C) を設定できます。

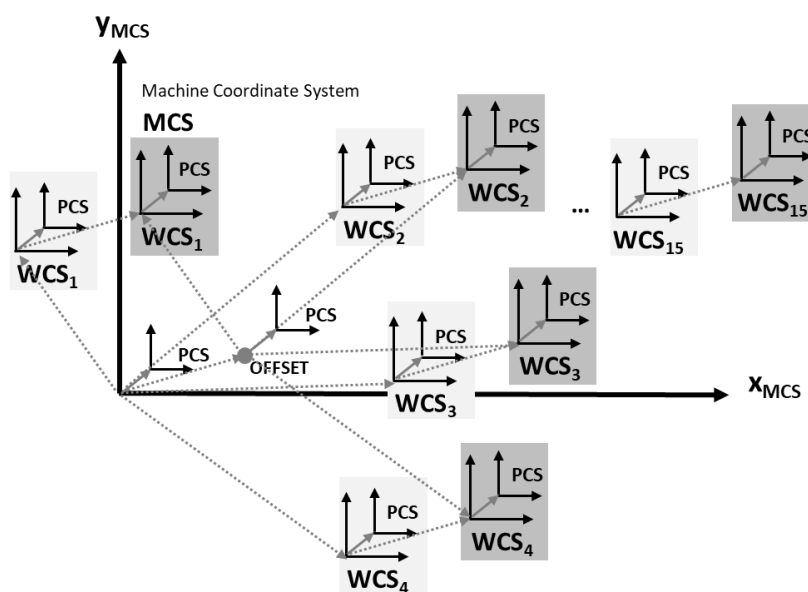


图 6.1.2.6

### 6.1.3 運動学

キネマティクスは、主に ACS(軸座標系)と MCS(機械座標系)の間の変換を扱います。フォワードキネマティクスは、ACS の各軸の位置フィードバックから MCS の座標位置までの計算です。逆に、逆運動学は、MCS の座標位置から ACS の各軸の位置までの計算です。表 6.1.3.1 に、HIMC が提供するキネマティクスコンフィギュレーションの定義を示します。

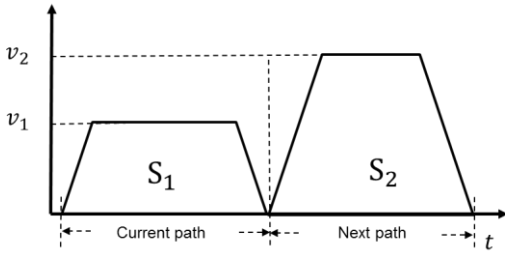
表 6.1.3.1

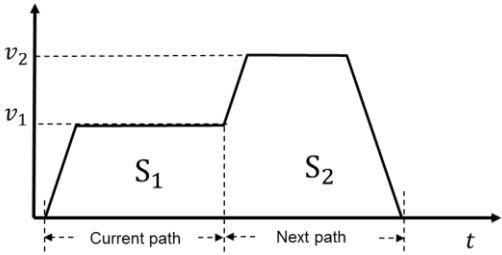
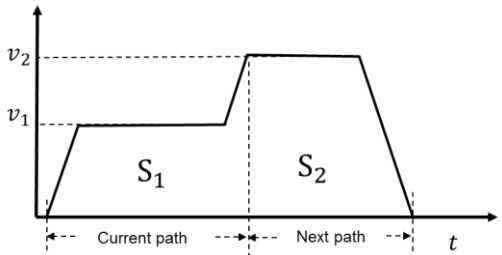
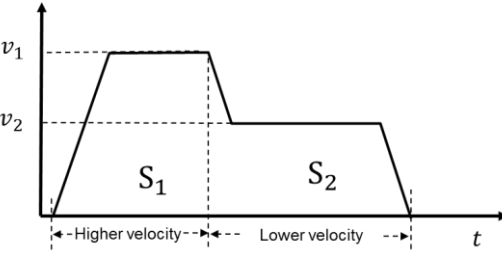
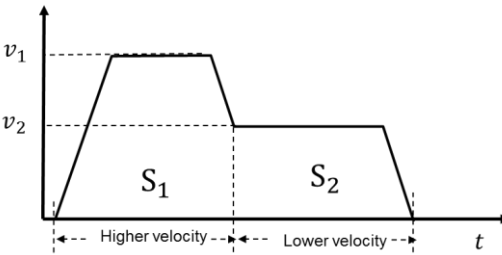
ID	名称	説明
1	Cartesian	連携モーショングループの各軸を直交座標の X、Y、Z、A、B、C 軸にそれぞれマッピングします。関節空間で許容される最大軸数は 6 です。(軸グループのデフォルト)
2	SCARA	(サポートされていません)
3	WAFER	(サポートされていません)
4	6-Axis Articulated Robot	(サポートされていません)

### 6.1.4 バッファモード

バッファモードは、隣接するパスの終点での速度プロファイルを決めます。ユーザーはこの設定を使用して、隣接する 2 つのパスのプロファイル速度を計画できます。表 6.1.4.1 に、HIMC が提供するバッファモードの定義を示します。

表 6.1.4.1

HMPL 定義	説明
BM_ABORT	進行中のモーションを中止し、すぐに次のモーションを開始します
BM_BUFF	<p>現在のパスが完了したら、次のパスを開始します。</p> 

BM_PREV	<p>速度は、現在のパスの速度とブレンドされます。(ブレンド)</p> 
BM_NEXT	<p>速度は、次のパスの速度とブレンドされます。(ブレンド)</p> 
BM_HIGH	<p>速度は、2つのパス間でより高い速度とブレンドされます。(ブレンド)</p> 
BM_LOW	<p>速度は、2つのパスの間の低い速度とブレンドされます。(ブレンド)</p> 

## 6.1.5 Transition モード

トランジションモードは、隣接するパス間のトランジションカーブのタイプを決定します。この設定により、HIMC は、設定されたパラメーターに基づいて 2 つの線形モーションコマンド間のコーナースムージングの計算を行います (表 6.1.5.1 を参照)。より良い計画パスを実現するために、これは元のモーションプロファイルに影響します。

表 6.1.5.1

HMPL 定義	説明	対応パラメーター	単位
TM_NONE	なし: トランジション曲線を挿入しません (デフォルトモード)	N/A	N/A
TM_START_VEL (Not supported)	開始速度	TPStartVelocity	mm/s または deg/s
TM_CONST_VEL	等速度	TPVelocity	mm/s または deg/s
TM_CORNER_DIST	コーナー距離	TPCornerDistance	mm または deg
TM_MAX_CORNER_DEV (Not supported)	最大コーナー偏差	TPCornerDeviation	mm または deg

高度なグループモーション制御の関数「HIMC\_LinAbs」および「HIMC\_LinRel」をトランジションモード TM\_CONST\_VEL で使用して円運動の速度を割り当てる場合、円の開始位置からコーナーまでの距離は、セクション 6.3.25 で設定されたトランジションモードの距離パラメーターによって図のように決定されます。トランジションモード TM\_CORNER\_DIST を使用して、円の開始位置からコーナーまでの距離を割り当てると、円運動の速度は、セクション 6.3.25 で設定されたトランジションモードの速度パラメーターによって決定されます。

トランジションモードで計算されたスムージング円半径がセグメントの長さを超える場合、セグメント間のトランジションモード関数は無視されます。

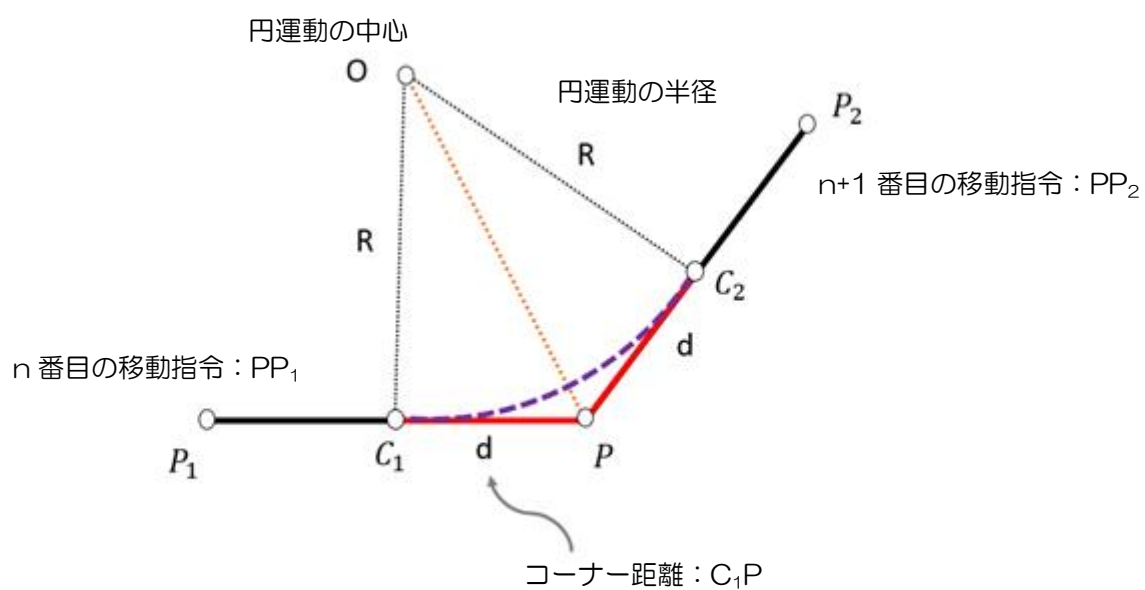


図 6.1.5.1



## 6.2 グループモーションコントロール

### 6.2.1 HIMC\_EnableGroup

#### 目的

軸グループを有効にします

#### 構文

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

#### パラメーター

**ctrl\_id [in]**      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**group\_id [in]**      Axis group index

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 備考

この関数を実行する前に、グループ内のすべての軸を有効にする必要があります。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_EnableGroup
LabVIEW	HIMC Enable Group.vi
Python	EnableGroup

6.2.2 HIMC\_DisableGroup

目的

軸グループを無効にします

構文

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

パラメーター

ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
                         HIMC\_ConnectCtrl を呼び出して取得する必要があります。

group\_id [in]           Axis group index

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_DisableGroup
LabVIEW	HIMC Disable Group.vi
Python	DisableGroup

## 6.2.3 HIMC\_ResetGroup

### 目的

軸グループの状態を「グループエラー停止」から「グループスタンバイ」に変更します

### 構文

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

### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

group\_id [in]      Axis group index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

この機能は、すべてのエラーがクリアされた後にのみ使用できます。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_ResetGroup
LabVIEW	HIMC Reset Group.vi
Python	ResetGroup

## 6.2.4 HIMC\_StopGroup

### 目的

軸グループの動作を停止します

### 構文

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

### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

group\_id [in]      Axis group index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

軸グループのモーションキューがクリアされます。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_StopGroup
LabVIEW	HIMC Stop Group.vi
Python	StopGroup

## 6.2.5 HIMC\_HaltGroup

### 目的

軸グループの動作を停止するのにその速度は 0 に設定します

### 構文

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

### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

group\_id [in]      Axis group index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

軸グループが所定の位置にない場合、移動を続けます。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_HaltGroup
LabVIEW	HIMC Halt Group.vi
Python	HaltGroup

6.2.6 HIMC\_ResumeGroup

目的

「停止」状態から軸グループの動作を再開します

構文

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

パラメーター

ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
                         HIMC\_ConnectCtrl を呼び出して取得する必要があります。

group\_id [in]           Axis group index

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_ResumeGroup
LabVIEW	HIMC Resume Group.vi
Python	ResumeGroup

## 6.2.7 HIMC\_JogGroup

### 目的

軸群を機械座標系の指定方向に特定の速度で無限運動を開始させる

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
carte_dir [in]	機械座標系での運動方向 0 ～ 5 の数字は機械座標系の 6-DOF {X, Y, Z, A, B, C}を順番に表します。
jog_vel [in]	特定の速度の値 その正/負の値は、運動方向の同方向/逆方向の動きを表します。 パラメーター単位：mm/s または deg/s

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_JogGroup
LabVIEW	HIMC Jog Group.vi
Python	JogGroup

## 6.2.8 HIMC\_JogGroupAxis

### 目的

軸グループ内の特定の軸に、軸座標系で特定の速度で無限運動を開始させること

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
grp_axis [in]	軸グループで Axis index. 番号 0 ～ 8 は、各軸が軸グループに追加される順番を表します。
jog_vel [in]	特定の速度の値 その正/負の値は、運動方向の同方向/逆方向の動きを表します。 パラメーター単位：mm/s または deg/s

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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



## 6.2.9 HIMC\_LineAbs2D

### 目的

機械座標系の絶対目標位置に向けて、軸グループ上で補間された 2 次元の直線移動を指令します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
end_x [in]	X 座標での絶対ターゲット位置の値
end_y [in]	Y 座標での絶対ターゲット位置の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_LineAbs2D
LabVIEW	HIMC Line Abs2D.vi
Python	LineAbs2D

## 6.2.10 HIMC\_LineAbs3D

### 目的

機械座標系の絶対目標位置に向かって、軸グループ上で補間された 3 次元の直線移動を命令します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
end_x [in]	X 座標での絶対ターゲット位置の値
end_y [in]	Y 座標での絶対ターゲット位置の値
end_z [in]	Z 座標での絶対ターゲット位置の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_LineAbs3D
LabVIEW	HIMC Line Abs3D.vi
Python	LineAbs3D

## 6.2.11 HIMC\_LineRel2D

### 目的

機械座標系の相対位置に向かって、軸グループ上で補間された 2 次元の直線移動を指令します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
distance_x [in]	X 座標での相対距離の値
distance_y [in]	Y 座標での相対距離の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_LineRel2D
LabVIEW	HIMC Line Rel2D.vi
Python	LineRel2D

## 6.2.12 HIMC\_LineRel3D

### 目的

機械座標系の相対位置に向かって、軸グループ上で補間された 3 次元の直線移動を指令します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
distance_x [in]	X 座標での相対距離の値
distance_y [in]	Y 座標での相対距離の値
distance_z [in]	Z 座標での相対距離の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_LineRel3D
LabVIEW	HIMC Line Rel3D.vi
Python	LineRel3D

## 6.2.13 HIMC\_Arc2D

### 目的

機械座標系の絶対目標位置に向かって、軸グループ上で補間された 2 次元円運動を指令します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
border_x [in]	X 座標での境界の絶対位置の値
border_y [in]	Y 座標での境界の絶対位置の値
end_x [in]	X 座標での絶対終了位置の値
end_y [in]	Y 座標での絶対終了位置の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

## 他の API 環境の対応名称

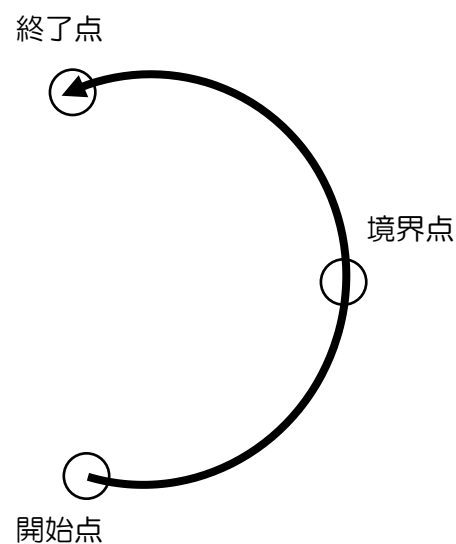
C#	HIMC_Arc2D
LabVIEW	HIMC Arc2D.vi
Python	Arc2D

### 長所

ユーザーは境界点(動きの中で最も遠い点)を指定し、マシンがそこに到達できることを確認できます。

### 短所

1つのコマンドで角度  $< 2\pi$  に制限されます。



## 6.2.14 HIMC\_ArcCW2D

### 目的

機械座標系の時計回りの絶対目標位置に向かって、軸グループ上で補間された 2 次元円運動を指令します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
center_x [in]	X 座標の絶対中心位置の値
center_y [in]	Y 座標の絶対中心位置の値
end_x [in]	X 座標での絶対終了位置の値
end_y [in]	Y 座標での絶対終了位置の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_ArcCW2D
LabVIEW	HIMC ArcCW2D.vi
Python	ArcCW2D

## 6.2.15 HIMC\_ArcCCW2D

### 目的

機械座標系の反時計回りの絶対目標位置に向かって、軸グループ上で補間された 2 次元円運動を指令します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
center_x [in]	X 座標の絶対中心位置の値
center_y [in]	Y 座標の絶対中心位置の値
end_x [in]	X 座標での絶対終了位置の値
end_y [in]	Y 座標での絶対終了位置の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_ArcCCW2D
LabVIEW	HIMC ArcCCW2D.vi
Python	ArcCCW2D



## 6.2.16 HIMC\_Circle2D

### 目的

機械座標系の絶対目標位置に向かって、軸グループ上で補間された 2 次元円運動を指令します

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
center_x [in]	X 座標の絶対中心位置の値
center_y [in]	Y 座標の絶対中心位置の値
end_x [in]	X 座標での絶対終了位置の値
end_y [in]	Y 座標での絶対終了位置の値
turns [in]	始点を基準とした円形パスのターン数 円形パスの方向と合計角度を決定します

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

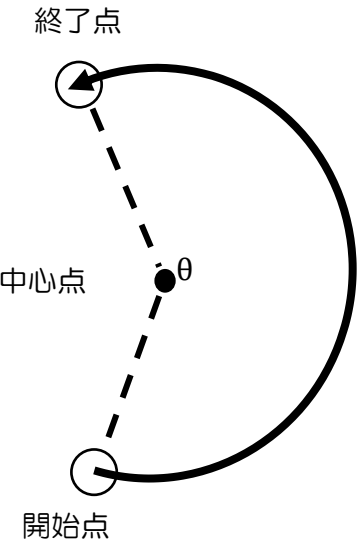
- パラメーター turns は、円軌道の回転方向を表します。  
turns >= 0 の場合、反時計回りに移動します。turns < 0 の場合、時計回りに移動します。
- || turns || 時 <= 1、円形軌道の合計移動角度は < 360°です。  
円軌道の移動角度の合計が >= 360° (つまり、1 回転または複数回転) の場合、|| turns || >= 2 でなければなりません。
- この関数を使用する場合、turns = 0 と turns = 1 の動作は同じです。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_Circle2D
LabVIEW	HIMC Circle2D.vi
Python	Circle2D



長所  
角度に制限はありません。

短所  
ユーザーは境界点(動きの最も遠い点)を指定できません。そのため、境界点に到達しない場合があります。

turns の値は、円運動の方向を決定します。※  $\text{angle} = \theta + \text{turns} \times 360$  ※  
turns ≥ 0 は C.C.W を示します。方向、turns < 0 は C.W.方向を示します。turns = 0 の動きは、turns = 1 の動きと同じであることに注意してください。次の表では、例として  $\theta = 210^\circ$  を使用しています。

Turns	計算	角度 ( 度 )
-2	$210 - 2 \times 360^\circ$	$-510^\circ$
-1	$210 - 1 \times 360^\circ$	$-150^\circ$
0	$210 + 0 \times 360^\circ$	$210^\circ$
1	$210 + 0 \times 360^\circ$	$210^\circ$
2	$210 + 1 \times 360^\circ$	$570^\circ$

## 6.2.17 HIMC\_ArcAngle2D

### 目的

与えられた角度に基づいて、機械座標系の絶対目標位置に向かって、軸グループ上で補間された 2 次元円運動を指令します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
center_x [in]	X 座標の絶対中心位置の値
center_y [in]	Y 座標の絶対中心位置の値
angle [in]	絶対中心位置に対する始点と終点の角度。 円運動の方向と回転角度を決定します。 パラメーター単位：deg

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

パラメーター「angle」は、円軌道の回転方向を表します。

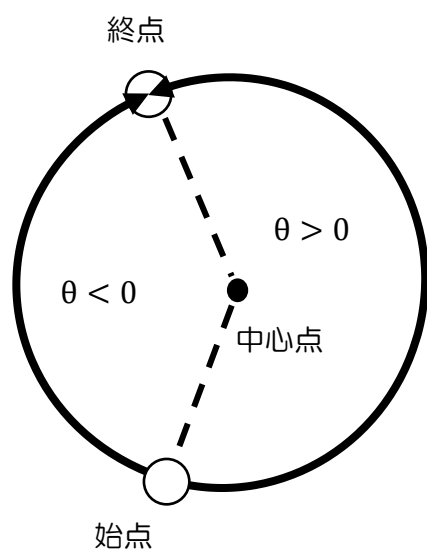
「angle」> 0 の場合、反時計回りに移動します。「angle」< 0 の場合、時計回りに移動します。

## 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

## 他の API 環境の対応名称

C#	HIMC_ArcAngle2D
LabVIEW	HIMC Arc Angle2D.vi
Python	ArcAngle2D



$\theta$  の値は、円運動の方向を決定します。

$\theta > 0$  : 反時計回りに移動

$\theta < 0$  : 時計回りに移動

## 6.3 グループ設定

### 6.3.1 HIMC\_AddAxesToGrp

#### 目的

特定の順序で軸グループに軸を追加します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
num_of_axes [in]	軸グループに追加する軸の数。(最大: 9)
p_axis_list [in]	軸 ID のシーケンスリストを格納するバッファへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

6.3.2 HIMC\_RemoveAxisFromGrp

目的

軸グループから最後の軸を削除します。

構文

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

パラメーター

ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
                          HIMC\_ConnectCtrl を呼び出して取得する必要があります。

group\_id [in]           Axis group index

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

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

### 6.3.3 HIMC\_SetupGroup

#### 目的

特定の順序で軸グループを設定します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
num_of_axes [in]	軸数。(最大: 9)
p_axis_list [in]	軸 ID のシーケンスリストを格納するバッファへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_SetupGroup
LabVIEW	HIMC Setup Group.vi
Python	SetupGroup

6.3.4 HIMC\_UngrpAllAxes

目的

軸グループをグループ解除して無効にします。

構文

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

パラメーター

ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
                         HIMC\_ConnectCtrl を呼び出して取得する必要があります。

group\_id [in]           Axis group index

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

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



### 6.3.5 HIMC\_GetGroupID

#### 目的

軸が属する軸グループ ID を取得します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_group_id [out]	軸が属する軸グループ ID を受け取るバッファへのポインター 値が-1 の場合、軸がどの軸グループにも属していないことを示します。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 6.3.6 HIMC\_SetGrpMotionProfile

### 目的

軸グループの TCP 線形モーションパラメーターを設定します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
max_velocity [in]	軸グループの最大線形プロファイル速度。 パラメーター単位：mm/s 入力範囲：0 ～ 5000
max_acceleration [in]	軸グループの線形プロファイルの最大加速度。 パラメーター単位：mm/s <sup>2</sup> 入力範囲：>0 ～ 50000 (加速度を 0 にすることはできません)
max_deceleration [in]	軸グループの線形プロファイルの最大減速度。 パラメーター単位：mm/s <sup>2</sup> 入力範囲：>0 ～ 50000 (減速度を 0 にすることはできません)
smooth_time [in]	軸グループの線形プロファイルの平滑化時間。 パラメーター単位：ms 入力範囲：0 ～ 500

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

線形グループモーションプロファイルの既定値は、速度、加速、減速、およびスムーズ時間に対してそれぞれ [100、500、500、50] です。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 6.3.7 HIMC\_SetGrpAngMotionProfile

### 目的

軸グループの TCP 角運動パラメーターを設定します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
max_velocity [in]	軸グループの最大角プロファイル速度。 パラメーター単位：deg/s 入力範囲：0 ～ 7200
max_acceleration [in]	軸グループの最大角プロファイル加速度。 パラメーター単位：deg/s <sup>2</sup> 入力範囲：>0 ～ 72000 (加速度を 0 にすることはできません)
max_deceleration [in]	軸グループの最大角度プロファイル減速。 パラメーター単位：deg/s <sup>2</sup> 入力範囲：>0 ～ 72000 (減速度を 0 にすることはできません)
smooth_time [in]	軸グループの角度プロファイルの平滑化時間。 パラメーター単位：ms 入力範囲：0 ～ 500

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

角度グループモーションプロファイルのデフォルト値は、速度、加速、減速、およびスムーズ時間に対してそれぞれ [360, 1800, 1800, 50] です。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

## 6.3.8 HIMC\_GetGrpKin

### 目的

軸グループのキネマティクスタイプを取得します。

### 構文

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

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- group\_id [in]      Axis group index
- p\_grp\_kin [out]      軸グループのキネマティクスタイプを受け取るバッファへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

### 6.3.9 HIMC\_SetGrpKin

#### 目的

軸グループのキネマティクスタイプを設定します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
kin_type [in]	軸グループの新しいキネマティクスタイプ。詳細については、セクション 6.1.3 を参照してください。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 6.3.10 HIMC\_GetGrpMaxVel

### 目的

軸グループの最大プロファイル速度を取得します。

### 構文

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

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- group\_id [in]      Axis group index
- p\_grp\_vel [out]    軸グループの最大プロファイル速度を受け取るバッファーへのポインター  
パラメーター単位：mm/s または deg/s

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetGrpMaxVel
LabVIEW	HIMC Get Grp Max Vel.vi
Python	GetGrpMaxVel



### 6.3.11 HIMC\_SetGrpVel

#### 目的

軸グループの最大プロファイル速度を設定します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
vel [in]	軸グループの新しい最大プロファイル速度。 パラメーター単位：mm/s または deg/s 入力範囲：0 ～ 5000

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 6.3.12 HIMC\_GetGrpMaxAcc

### 目的

軸グループの最大プロファイル加速度を取得します。

### 構文

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

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- group\_id [in]      Axis group index
- p\_grp\_acc [out]    軸グループの最大プロファイル加速度を受け取るバッファーへのポインター  
パラメーター単位：mm/s<sup>2</sup> または deg/s<sup>2</sup>

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetGrpMaxAcc
LabVIEW	HIMC Get Grp Max Acc.vi
Python	GetGrpMaxAcc

### 6.3.13 HIMC\_SetGrpAcc

#### 目的

軸グループの最大プロファイル加速度を設定します。

#### 構文

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

#### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- group\_id [in]     Axis group index
- acc [in]          軸グループの新しい最大プロファイル加速度。  
パラメーター単位：mm/s<sup>2</sup> または deg/s<sup>2</sup>  
入力範囲：>0 ~ 50000 (加速度を 0 にすることはできません)

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 6.3.14 HIMC\_SetGrpAccTime

### 目的

軸グループの加速時間を設定します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
acc_time [in]	軸グループの加速時間。 パラメーター単位：ms 入力範囲：ゼロ以外の正の値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

### 6.3.15 HIMC\_GetGrpMaxDec

#### 目的

軸グループの最大プロファイル減速度を取得します。

#### 構文

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

#### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- group\_id [in]      Axis group index
- p\_grp\_dec [out]   軸グループの最大プロファイル減速度を受け取るバッファーへのポインター  
パラメーター単位：mm/s<sup>2</sup> または deg/s<sup>2</sup>

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetGrpMaxDec
LabVIEW	HIMC Get Grp Max Dec.vi
Python	GetGrpMaxDec

## 6.3.16 HIMC\_SetGrpDec

### 目的

軸グループの最大プロファイル減速度を設定します。

### 構文

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

### パラメーター

- ctrl\_id [in]**      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- group\_id [in]**      Axis group index
- dec [in]**      軸グループの新しい最大プロファイル減速度。  
パラメーター単位：mm/s<sup>2</sup> または deg/s<sup>2</sup>  
入力範囲：>0 ~ 50000 (減速度を 0 にすることはできません)

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

### 6.3.17 HIMC\_SetGrpDecTime

#### 目的

軸グループの減速時間を設定します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
dec_time [in]	軸群の減速時間です。 パラメーター単位：ms 入力範囲：ゼロ以外の正の値

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 6.3.18 HIMC\_GetGrpSMTime

### 目的

軸グループのプロファイルスムーズタイムを取得します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
p_grp_smooth_time [out]	軸グループのプロファイルの滑らかな時間を受け取るバッファーへのポインター パラメーター単位：ms

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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



### 6.3.19 HIMC\_SetGrpSMTime

#### 目的

軸グループのプロファイルスムーズ時間を設定します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
smooth_time [in]	軸グループの新しいプロファイルスムーズ時間。 パラメーター単位：ms 入力範囲：0 ～ 500

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 6.3.20 HIMC\_GetGrpCoordSys

### 目的

軸グループの座標系を取得します。

### 構文

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

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
p_grp_coord_sys [out]	軸グループの座標系を受け取るバッファーへのポインター。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

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

### 6.3.21 HIMC\_SetGrpCoordSys

#### 目的

軸グループの座標系を設定します。

#### 構文

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

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
coord_sys [in]	軸グループの新しい座標系。詳細については、セクション 6.1.2 を参照してください。 例：1. kCoord_MCS 2. kCoord_WCS1   kCoord_PCS 3. kCoord_OFFSET   kCoord_WCS2   kCoord_PCS

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

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

## 6.3.22 HIMC\_GetGrpBufferMode

### 目的

軸グループのバッファーマードを取得します。

### 構文

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

### パラメーター

ctrl\_id [in]                      HIWIN モーションコントローラーのコントローラーID  
                                  HIMC\_ConnectCtrl を呼び出して取得する必要があります。

group\_id [in]                    Axis group index

p\_grp\_buffer\_mode [out]      軸グループのバッファーマードを受け取るバッファへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetGrpBufferMode
LabVIEW	HIMC Get Buffer Mode.vi
Python	GetGrpBufferMode

### 6.3.23 HIMC\_SetGrpBufferMode

#### 目的

軸グループのバッファーマードを設定します。

#### 構文

```
int HIMC_SetGrpBufferMode(
    int ctrl_id,
    int group_id,
    int buffer_mode
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
buffer_mode [in]	軸グループの新しいバッファーマード。詳細については、セクション 6.1.4 を参照してください。 入力範囲：0 ～ 5

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_SetGrpBufferMode
LabVIEW	HIMC Set Grp Buffer Mode.vi
Python	SetGrpBufferMode

## 6.3.24 HIMC\_GetGrpTransMode

### 目的

軸グループの transition モードを取得します。

### 構文

```
int HIMC_GetGrpTransMode(
    int ctrl_id,
    int group_id,
    int *p_grp_trans_mode
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
p_grp_trans_mode [out]	軸グループの transition モードを受け取るバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetGrpTransMode
LabVIEW	HIMC Get Trans Mode.vi
Python	GetGrpTransMode

### 6.3.25 HIMC\_SetGrpTransMode

#### 目的

軸グループの transition モードを設定します。

#### 構文

```
int HIMC_SetGrpTransMode(
    int ctrl_id,
    int group_id,
    int trans_mode
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
trans_mode [in]	軸グループの新しい transition モード。詳細については、セクション 6.1.5 を参照してください。 入力範囲：0 ～ 4

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_SetGrpTransMode
LabVIEW	HIMC Set Grp Trans Mode.vi
Python	SetGrpTransMode

## 6.3.26 HIMC\_SetGrpTransPrm

### 目的

軸グループの transition モードのパラメーターを設定します。

### 構文

```
int HIMC_SetGrpTransPrm(
    int    ctrl_id,
    int    group_id,
    double trans_vel,
    double trans_dis
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
trans_vel [in]	軸グループの新しい transition モードの速度パラメーター。 詳細については、セクション 6.1.5 を参照してください。
trans_dis [in]	軸グループの新しい transition モードの距離パラメーター 詳細については、セクション 6.1.5 を参照してください。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetGrpTransPrm
LabVIEW	HIMC Set Grp Trans Prm.vi
Python	SetGrpTransPrm



### 6.3.27 HIMC\_GetGrpCmdNum

#### 目的

コマンドバッファ内の軸グループのコマンド数を取得します。

#### 構文

```
int HIMC_GetGrpCmdNum(
    int ctrl_id,
    int group_id,
    int *p_grp_cmd_num
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
p_grp_cmd_num [out]	コマンドバッファ内の軸グループのコマンド数を受け取るバッファへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetGrpCmdNum
LabVIEW	HIMC Get Grp Cmd Num.vi
Python	GetGrpCmdNum

## 6.3.28 HIMC\_SetGrpVelScale

### 目的

軸群モーションの速度スケールを設定します。

### 構文

```
int HIMC_SetGrpVelScale(
    int    ctrl_id,
    int    group_id,
    double vel_scale
);
```

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- group\_id [in]      Axis group index
- vel\_scale [in]      軸グループモーションの新しい速度スケール。  
入力範囲：0 ～ 100

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetGrpVelScale
LabVIEW	HIMC Set Grp Vel Scale.vi
Python	SetGrpVelScale

### 6.3.29 HIMC\_GetGrpVelScale

#### 目的

軸群モーションの速度スケールを取得します。

#### 構文

```
int HIMC_GetGrpVelScale(
    int    ctrl_id,
    int    group_id,
    double *p_grp_vel_scale
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis index
p_grp_vel_scale [out]	軸グループモーションの速度スケールを受け取るバッファへのポインター。その範囲は 0 ～ 100 です。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetGrpVelScale
LabVIEW	HIMC Get Grp Vel Scale.vi
Python	GetGrpVelScale

## 6.3.30 HIMC\_GetGrpCoordTrans

### 目的

軸グループの座標系の変換パラメーターを取得します。

### 構文

```
int HIMC_GetGrpCoordTrans(
    int    ctrl_id,
    int    group_id,
    int    coord_sys,
    double *trans_param
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis index
coord_sys [in]	座標系 詳細については、セクション 19.2 CoordSystem を参照してください。
trans_param [out]	6-DOF {X, Y, Z, A, B, C}の変換パラメーターを含む 6 要素配列へのポインター。 パラメーター単位：mm (X, Y, Z)、deg (A, B, C)

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetGrpCoordTrans
LabVIEW	HIMC Get Grp Coord Trans.vi
Python	GetGrpCoordTrans

### 6.3.31 HIMC\_SetGrpCoordTrans

#### 目的

軸グループの座標系の変換パラメーターを設定します。

#### 構文

```
int HIMC_SetGrpCoordTrans(
    int    ctrl_id,
    int    group_id,
    int    coord_sys,
    double *trans_param
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis index
coord_sys [in]	座標系 詳細については、セクション 19.2 CoordSystem を参照してください。
trans_param [in]	6-DOF {X, Y, Z, A, B, C}の変換パラメーターを含む 6 要素配列へのポインター パラメーター単位：mm (X, Y, Z)、deg (A, B, C)

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_SetGrpCoordTrans
LabVIEW	HIMC Set Grp Coord Trans.vi
Python	SetGrpCoordTrans

## 6.3.32 HIMC\_GetGrpPoseCmd

### 目的

群座標系の姿勢指令を取得します。

### 構文

```
int HIMC_GetGrpPoseCmd(
    int    ctrl_id,
    int    group_id,
    int    coord_sys,
    double *pose_cmd
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis index
coord_sys [in]	座標系 詳細については、セクション 19.2 CoordSystem を参照してください。
pose_cmd [out]	6-DOF {X, Y, Z, A, B, C}のポーズコマンドを含む 6 要素配列へのポインター パラメーター単位：mm (X, Y, Z)、deg (A, B, C)

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetGrpPoseCmd
LabVIEW	HIMC Get Grp Pose Cmd.vi
Python	GetGrpPoseCmd

### 6.3.33 HIMC\_GetGrpPoseFb

#### 目的

軸グループの座標系のポーズフィードバックを取得します。

#### 構文

```
int HIMC_GetGrpPoseFb(
    int    ctrl_id,
    int    group_id,
    int    coord_sys,
    double *pose_fb
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID. HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis index
coord_sys [in]	座標系 詳細については、セクション 19.2 CoordSystem を参照してください。
pose_fb [out]	6-DOF {X, Y, Z, A, B, C}のポーズフィードバックを含む 6 要素配列へのポインター パラメーター単位：mm (X, Y, Z)、deg (A, B, C)

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetGrpPoseFb
LabVIEW	HIMC Get Grp Pose Fb.vi
Python	GetGrpPoseFb

## 6.4 グループステータス

### 6.4.1 HIMC\_IsGrpEnabled

#### 目的

軸グループの「有効」ステータスを問い合わせます

#### 構文

```
int HIMC_IsGrpEnabled(
    int ctrl_id,
    int group_id,
    int *p_is_grp_enabled
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID. HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
p_is_grp_enabled [out]	軸グループの有効ステータスを受け取るバッファーへのポインター 軸グループが「GrpEnabled」状態の場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_IsGrpEnabled
LabVIEW	HIMC Is Grp Enabled.vi
Python	IsGrpEnabled



## 6.4.2 HIMC\_IsGrpMoving

### 目的

軸グループの「移動」ステータスを問い合わせます。軸グループが移動している場合、PG(プロファイルジェネレーター)は新しい位置を出力し続けます。

### 構文

```
int HIMC_IsGrpMoving(
    int ctrl_id,
    int group_id,
    int *p_is_grp_moving
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
p_is_grp_moving [out]	軸群の移動状態を受け取るバッファーへのポインター 軸グループが「GrpMoving」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_IsGrpMoving
LabVIEW	HIMC Is Grp Moving.vi
Python	IsGrpMoving

## 6.4.3 HIMC\_IsGrpInPos

### 目的

軸グループの「インポジション」ステータスを照会します。軸グループがインポジションの場合、グループ内のすべての軸がインポジションです。

### 構文

```
int HIMC_IsGrpInPos(
    int ctrl_id,
    int group_id,
    int *p_is_grp_inpos
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
p_is_grp_inpos [out]	軸グループのインポジションステータスを受け取るバッファーへのポインター 軸グループが「GrpInPos」状態にある場合、値は 1 になります。それ以外の 場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_IsGrpInPos
LabVIEW	HIMC Is Grp In Pos.vi
Python	IsGrpInPos

## 6.4.4 HIMC\_IsGrpErrorStop

### 目的

軸グループが「エラー停止」状態かどうかを問い合わせます。

### 構文

```
int HIMC_IsGrpErrorStop(
    int ctrl_id,
    int group_id,
    int *p_is_grp_errorstop
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
p_is_grp_errorstop [out]	軸グループのエラー停止ステータスを受け取るバッファへのポインター 軸グループが「GrpErrorStop」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_IsGrpErrorStop
LabVIEW	HIMC Is Grp Error Stop.vi
Python	IsGrpErrorStop

## 6.5 高度なグループモーションコントロール

### 6.5.1 HIMC\_LinAbs

#### 目的

特定の座標系の絶対位置に向かって、軸グループ上で補間された直線移動を命令します。

#### 構文

```
int HIMC_LinAbs(
    int ctrl_id,
    int group_id,
    CoordPosition *target_pos,
    MotionProfile *motion_profile,
    CoordSystem coord_sys,
    MotionBufferMode buff_mode,
    MotionTransitionMode trans_mode,
    double trans_par
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
target_pos [in]	ツール中心点(エンドエフェクタ)の 6-DOF ターゲット位置を格納するバッファへのポインター パラメーター単位：mm (X, Y, Z)、deg (A, B, C) 詳細については、セクション 18.2 CoordPosition を参照してください。
motion_profile [in]	ツールの中心点(エンドエフェクタ)のモーションプロファイル設定を格納するバッファへのポインター 詳細については、セクション 18.3 MotionProfile を参照してください。
coord_sys [in]	該当する座標系を指定します 詳細については、セクション 19.2 CoordSystem を参照してください。
buff_mode [in]	パスバッファモードを指定します 詳細については、セクション 19.3 MotionBufferMode を参照してください。
trans_mode [in]	パス transition モードを指定します 詳細については、セクション 19.4 MotionTransitionMode を参照してください。
trans_par [in]	特定の transition モードのパラメーターを指定します。 不要な場合はゼロを記入してください。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_LinAbs
LabVIEW	HIMC Lin Abs.vi
Python	LinAbs

## 6.5.2 HIMC\_LinRel

### 目的

特定の座標系の相対位置に向かって、軸グループ上で補間された直線移動を命令します。

### 構文

```
int HIMC_LinRel(
    int ctrl_id,
    int group_id,
    CoordPosition *relative_dist,
    MotionProfile *motion_profile,
    CoordSystem coord_sys,
    MotionBufferMode buff_mode,
    MotionTransitionMode trans_mode,
    double trans_par
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
relative_dist [in]	ツールの中心点（エンドエフェクタ）の 6-DOF の相対距離を格納するバッファへのポインター パラメーターの単位：mm（X、Y、Z）、deg（A、B、C） 詳細については、セクション 18.2 CoordPosition を参照してください
motion_profile [in]	ツールの中心点（エンドエフェクタ）のモーションプロファイル設定を格納するバッファへのポインター 詳細については、セクション 18.3 MotionProfile を参照してください。
coord_sys [in]	該当する座標系を指定します 詳細については、セクション 19.2 CoordSystem を参照してください。
buff_mode [in]	パスバッファモードを指定します 詳細については、セクション 19.3 MotionBufferMode を参照してください。
trans_mode [in]	パス Transition モードを指定します 詳細については、セクション 19.4 MotionTransitionMode を参照してください。

**trans\_par [in]**            特定の transition モードのパラメーターを指定します。  
                                  不要な場合はゼロを記入してください

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_LinRel
LabVIEW	HIMC Lin Rel.vi
Python	LinRel

## 6.5.3 HIMC\_CircAbs

### 目的

特定の座標系の絶対位置に向かって、軸グループ上で補間された円運動を指令します。

### 構文

```
int HIMC_CircAbs(
    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,
    double trans_par
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
center_pos [in]	中心点の位置を格納するバッファへのポインター パラメーター単位：mm 詳細については、セクション 18.4 CenterPosition を参照してください。
normal_vector [in]	円の法線ベクトルを格納するバッファへのポインター 詳細については、セクション 18.5 NormalVector を参照してください。
turns [in]	始点を基準とした円形パスのターン数 円形パスの方向と合計角度を決定します
target_pos [in]	ツール中心点（エンドエフェクタ）の 6-DOF ターゲット位置を格納するバッファへのポインター パラメーターの単位：mm（X、Y、Z）、deg（A、B、C） 詳細については、セクション 18.2 CoordPosition を参照してください。
motion_profile [in]	ツールの中心点（エンドエフェクタ）のモーションプロファイル設定を格納するバッファへのポインター 詳細については、セクション 18.3 MotionProfile を参照してください。
coord_sys [in]	該当する座標系を指定します 詳細については、セクション 19.2 CoordSystem を参照してください。
buff_mode [in]	パスバッファモードを指定します 詳細については、セクション 19.3 MotionBufferMode を参照してください。
trans_mode [in]	パス transition モードを指定します 詳細については、セクション 19.4 MotionTransitionMode を参照してください。



trans\_par [in]            特定の transition モードのパラメーターを指定します。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_CircAbs
LabVIEW	HIMC Circ Abs.vi
Python	CircAbs

## 6.5.4 HIMC\_CircRel

### 目的

特定の座標系の相対位置に向かって、軸グループ上で補間された円運動を指令します。

### 構文

```
int HIMC_CircRel(
    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,
    double trans_par
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
center_pos [in]	中心点の位置を格納するバッファへのポインター パラメーター単位：mm 詳細については、セクション 18.4 CenterPosition を参照してください。
normal_vector [in]	円の法線ベクトルを格納するバッファへのポインター 詳細については、セクション 18.5 NormalVector を参照してください。
turns [in]	始点を基準とした円形パスのターン数 円形パスの方向と合計角度を決定します
relative_dist [in]	ツールの中心点（エンドエフェクタ）の 6-DOF の相対距離を格納するバッファへのポインター パラメーターの単位：mm（X、Y、Z）、deg（A、B、C） 詳細については、セクション 18.2 CoordPosition を参照してください。
motion_profile [in]	ツールの中心点（エンドエフェクタ）のモーションプロファイル設定を格納するバッファへのポインター 詳細については、セクション 18.3 MotionProfile を参照してください。
coord_sys [in]	該当する座標系を指定します 詳細については、セクション 19.2 CoordSystem を参照してください。
buff_mode [in]	パスバッファモードを指定します 詳細については、セクション 19.3 MotionBufferMode を参照してください。
trans_mode [in]	パス transition モードを指定します 詳細については、セクション 19.4 MotionTransitionMode を参照してください。

`trans_par [in]`          特定の transition モードのパラメータを指定します。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_CircRel
LabVIEW	HIMC Circ Rel.vi
Python	CircRel

(このページはブランクになっています)

## 7. GPIO 機能

7.1	概要 .....	7-2
7.1.1	GPIO 変数 .....	7-2
7.2	コントローラー IO 設定 .....	7-3
7.2.1	HIMC_SetHimcGpo .....	7-3
7.2.2	HIMC_ToggleHimcGpo .....	7-4
7.2.3	HIMC_SetHimcMultiGpo .....	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	スレーブ IO 設定 .....	7-9
7.3.1	HIMC_SetSlaveGpo .....	7-9
7.3.2	HIMC_ToggleSlaveGpo .....	7-10
7.3.3	HIMC_SetSlaveMultiGpo .....	7-11
7.4	コントローラー IO ステータス .....	7-12
7.4.1	HIMC_GetHimcGpi .....	7-12
7.4.2	HIMC_GetHimcGpo .....	7-13
7.4.3	HIMC_GetHimcMultiGpi .....	7-14
7.4.4	HIMC_GetHimcMultiGpo .....	7-15
7.4.5	HIMC_GetHimcGpiNum .....	7-16
7.4.6	HIMC_GetHimcGpoNum .....	7-17
7.4.7	HIMC_GetAllGPInvertSt .....	7-18
7.4.8	HIMC_GetAllGPOInvertSt .....	7-19
7.5	スレーブ IO ステータス .....	7-20
7.5.1	HIMC_GetSlaveGpi .....	7-20
7.5.2	HIMC_GetSlaveGpo .....	7-21
7.5.3	HIMC_GetSlaveMultiGpi .....	7-22
7.5.4	HIMC_GetSlaveMultiGpo .....	7-23
7.5.5	HIMC_GetSlaveGpiNum .....	7-24
7.5.6	HIMC_GetSlaveGpoNum .....	7-25

## 7.1 概要

HIMC は、8 セットの汎用入出力(GPIO)ピンを提供します。ハードウェアの遅延時間は 1ms 以内で、電源は 24V です。スレーブデバイスは、MoE 通信を介してコントローラーに接続し、その IO ステータスを更新できます。IO の数は、スレーブデバイスによって異なります。この章で提供される「SetGPO」や「SetSlvGPO」などの機能を使用して、ユーザーは HIMC とスレーブの出力ピンの信号をそれぞれ設定できます。また、ユーザーは入力/出力ピンの信号ステータスを照会できます。iA Studio の機能モジュール「デジタル IO」(「iA Studio ユーザーガイド」のセクション 4.4 を参照)を使用すると、HIMC およびスレーブの入出力状態を監視および設定できます。

HIMC のデジタル入力「I8」は非常停止信号(「HIMC インストールガイド」のセクション 3.3 を参照)であり、立ち上がりエッジ信号を受信するとトリガーされます。この時点で、すべての軸が無効になり、すべての HMPL タスクが停止します。

注：立ち上がりエッジ信号がトリガーされた後、ユーザーは軸を再度有効にするか、HMPL タスクを再実行できます。

### 7.1.1 GPIO 変数

GPIO 変数は、コントローラーの汎用入出力変数(システム変数)とスレーブの汎用入出力変数(スレーブ変数)の 2 つのカテゴリに分けられます。ユーザーは、iA Studio のスコープマネージャーを介して目的の変数を選択できます(「iA Studio ユーザーガイド」のセクション 4.8 を参照)。詳細な説明を表 7.1.1.1 および表 7.1.1.2 に示します。

表 7.1.1.1 コントローラーの汎用入出力変数

名称	変数	単位	説明
HIMC GPO	himc_gpo_bits	N/A	コントローラーの汎用出力ピンの状態
HIMC GPI	himc_gpi_bits	N/A	コントローラーの汎用入力ピンの状態

表 7.1.1.2 スレーブの汎用入出力変数

名称	変数	単位	説明
Slave GPO	gpoStBits_1	N/A	スレーブの汎用出力ピンの状態
Slave GPI	gpiStBits_1	N/A	スレーブの汎用入力ピンの状態

## 7.2 コントローラー IO 設定

### 7.2.1 HIMC\_SetHimcGpo

#### 目的

コントローラーの汎用出力の状態を設定します

#### 構文

```
int HIMC_SetHimcGpo(
    int ctrl_id,
    int gpo_idx,
    char state
);
```

#### パラメーター

**ctrl\_id [in]**      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**gpo\_idx [in]**      汎用出力インデックス

**state [in]**      設定する状態

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_SetHimcGpo
LabVIEW	HIMC Set Himc Gpo.vi
Python	SetHimcGpo

## 7.2.2 HIMC\_ToggleHimcGpo

### 目的

コントローラーの汎用出力の状態を切り替えます

### 構文

```
int HIMC_ToggleHimcGpo(
    int ctrl_id,
    int gpo_idx
);
```

### パラメーター

**ctrl\_id [in]**      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**gpo\_idx [in]**      汎用出力インデックス

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_ToggleHimcGpo
LabVIEW	HIMC Toggle Himc Gpo.vi
Python	ToggleHimcGpo



## 7.2.3 HIMC\_SetHimcMultiGpo

### 目的

コントローラーの複数の汎用出力の状態を設定します

### 構文

```
int HIMC_SetHimcMultiGpo(
    int ctrl_id,
    int start_idx,
    int num,
    char *p_state_array
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
start_idx [in]	設定する汎用出力の開始インデックス
num [in]	設定する汎用出力の数
p_state_array [in]	インデックス範囲 [start_idx, ..., start_idx + num – 1]の汎用出力の状態を設定するためのバッファへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetHimcMultiGpo
LabVIEW	HIMC Set Himc Multi Gpo.vi
Python	SetHimcMultiGpo

## 7.2.4 HIMC\_SetGPIInvert

### 目的

コントローラーの汎用入力の反転状態を設定します

### 構文

```
int HIMC_SetGPIInvert(
    int ctrl_id,
    int gpi_idx,
    char invert
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
gpi_idx [in]	汎用出力インデックス
invert [in]	設定する反転状態

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetGPIInvert
LabVIEW	HIMC Set GPI Invert.vi
Python	SetGPIInvert

## 7.2.5 HIMC\_SetGPOInvert

### 目的

コントローラーの汎用出力の反転状態を設定します

### 構文

```
int HIMC_SetGPOInvert(
    int ctrl_id,
    int gpo_idx,
    char invert
);
```

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- gpo\_idx [in]      汎用出力インデックス
- invert [in]      設定する反転状態

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetGPOInvert
LabVIEW	HIMC Set GPO Invert.vi
Python	SetGPOInvert

7.2.6 HIMC\_BindEMO

目的

汎用入力ピンを E-Stop にバインドするように設定します

構文

```
int HIMC_BindEMO(  
    int ctrl_id,  
    int gpi_idx  
);
```

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- gpi\_idx [in]

汎用出力インデックス、デフォルト値は 8 です。  
0 に設定すると、すべての汎用入力ピンが E-Stop にバインドされません。

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_BindEMO
LabVIEW	HIMC Bind EMO.vi
Python	BindEMO

## 7.3 スレーブ IO 設定

### 7.3.1 HIMC\_SetSlaveGpo

#### 目的

スレーブの汎用出力の状態を設定します

#### 構文

```
int HIMC_SetSlaveGpo(
    int ctrl_id,
    int slave_id,
    int gpo_idx,
    char state
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
gpo_idx [in]	汎用出力インデックス
state [in]	設定する状態

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_SetSlaveGpo
LabVIEW	HIMC Set Slave Gpo.vi
Python	SetSlaveGpo

## 7.3.2 HIMC\_ToggleSlaveGpo

### 目的

スレーブの汎用出力の状態を切り替えます

### 構文

```
int HIMC_ToggleSlaveGpo(
    int ctrl_id,
    int slave_id,
    int gpo_idx
);
```

### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

slave\_id [in]      スレーブ ID

gpo\_idx [in]      汎用出力インデックス

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_ToggleSlaveGpo
LabVIEW	HIMC Toggle Slave Gpo.vi
Python	ToggleSlaveGpo

### 7.3.3 HIMC\_SetSlaveMultiGpo

#### 目的

スレーブの複数の汎用出力の状態を設定します

#### 構文

```
int HIMC_SetSlaveMultiGpo(
    int ctrl_id,
    int slave_id,
    int start_idx,
    int num,
    char *p_state_array
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
start_idx [in]	設定する汎用出力の開始インデックス
num [in]	設定する汎用出力の数
p_state_array [in]	インデックス範囲 [start_idx, ..., start_idx + num – 1]の汎用出力の状態を設定するためのバッファへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_SetSlaveMultiGpo
LabVIEW	HIMC Set Slave Multi Gpo.vi
Python	SetSlaveMultiGpo

## 7.4 コントローラー IO ステータス

### 7.4.1 HIMC\_GetHimcGpi

#### 目的

コントローラーの汎用入力の状態を取得します

#### 構文

```
int HIMC_GetHimcGpi(
    int    ctrl_id,
    int    gpi_idx,
    char *p_state
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
gpi_idx [in]	汎用出力インデックス
p_state [out]	特定の入力の状態を受け取るバッファへのポインター 入力が「オン」状態の場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetHimcGpi
LabVIEW	HIMC Get Himc Gpi.vi
Python	GetHimcGpi



## 7.4.2 HIMC\_GetHimcGpo

### 目的

コントローラーの汎用出力の状態を取得します

### 構文

```
int HIMC_GetHimcGpo(
    int    ctrl_id,
    int    gpo_idx,
    char *p_state
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
gpo_idx [in]	汎用出力インデックス
p_state [out]	特定の出力の状態を受け取るバッファへのポインター 出力が「オン」状態の場合、値は 1 になります。それ以外の場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetHimcGpo
LabVIEW	HIMC Get Himc Gpo.vi
Python	GetHimcGpo

7.4.3 HIMC\_GetHimcMultiGpi

目的

コントローラーの複数の汎用入力の状態を取得します

構文

```
int HIMC_GetHimcMultiGpi(  
    int ctrl_id,  
    int start_idx,  
    int num,  
    char *p_state_array  
);
```

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- start\_idx [in]

取得する汎用入力の開始インデックス
- num [in]

取得する汎用入力の数
- p\_state\_array [out]

インデックス範囲 [start\_idx, ..., start\_idx + num – 1]の汎用入力の状態を受け取るバッファへのポインター

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_GetHimcMultiGpi
LabVIEW	HIMC Get Himc Multi Gpi.vi
Python	GetHimcMultiGpi

## 7.4.4 HIMC\_GetHimcMultiGpo

### 目的

コントローラーの複数の汎用出力の状態を取得します

### 構文

```
int HIMC_GetHimcMultiGpo(
    int ctrl_id,
    int start_idx,
    int num,
    char *p_state_array
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
start_idx [in]	取得する汎用出力の開始インデックス
num [in]	取得する汎用出力の数
p_state_array [out]	インデックス範囲 [start_idx, ..., start_idx + num – 1]の汎用出力の状態を受け取るバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetHimcMultiGpo
LabVIEW	HIMC Get Himc Multi Gpo.vi
Python	GetHimcMultiGpo

### 7.4.5 HIMC\_GetHimcGpiNum

#### 目的

コントローラーの汎用入力の数を取得します

#### 構文

```
int HIMC_GetHimcGpiNum(  
    int ctrl_id,  
    int *p_gpi_num  
);
```

#### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
                  HIMC\_ConnectCtrl を呼び出して取得する必要があります。

p\_gpi\_num [out]    コントローラーの汎用入力の数を受け取るバッファーへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetHimcGpiNum
LabVIEW	HIMC Get Himc Gpi Num.vi
Python	GetHimcGpiNum

## 7.4.6 HIMC\_GetHimcGpoNum

### 目的

コントローラーの汎用出力の数を取得します

### 構文

```
int HIMC_GetHimcGpoNum(
    int ctrl_id,
    int *p_gpo_num
);
```

### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

p\_gpo\_num [out]    コントローラーの汎用出力の数を受け取るバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetHimcGpoNum
LabVIEW	HIMC Get Himc Gpo Num.vi
Python	GetHimcGpoNum

## 7.4.7 HIMC\_GetAllGPIInvertSt

### 目的

コントローラーのすべての汎用入力の反転状態を取得します

### 構文

```
int HIMC_GetAllGPIInvertSt(
    int    ctrl_id,
    char  *p_state
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
p_state [out]	すべての汎用入力の反転状態を受け取るバッファーへのポインター 入力が「反転」状態の場合、値は 1 になります。それ以外の場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetAllGPIInvertSt
LabVIEW	HIMC Get Get All GPI Invert St.vi
Python	GetAllGPIInvertSt

## 7.4.8 HIMC\_GetAllGPOInvertSt

### 目的

コントローラーのすべての汎用出力の反転状態を取得します

### 構文

```
int HIMC_GetAllGPOInvertSt(
    int    ctrl_id,
    char *p_state
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
p_state [out]	すべての汎用出力の反転状態を受け取るバッファへのポインター 出力が「反転」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetAllGPOInvertSt
LabVIEW	HIMC Get Get All GPO Invert St.vi
Python	GetAllGPOInvertSt

## 7.5 スレーブ IO ステータス

### 7.5.1 HIMC\_GetSlaveGpi

#### 目的

スレーブの汎用入力の状態を取得します

#### 構文

```
int HIMC_GetSlaveGpi(
    int ctrl_id,
    int slave_id,
    int gpi_idx,
    char *p_state
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
gpi_idx [in]	汎用入力インデックス。
p_state [out]	特定の入力の状態を受け取るバッファへのポインター 入力が「オン」状態の場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetSlaveGpi
LabVIEW	HIMC Get Slave Gpi.vi
Python	GetSlaveGpi



## 7.5.2 HIMC\_GetSlaveGpo

### 目的

スレーブの汎用出力の状態を取得します

### 構文

```
int HIMC_GetSlaveGpo(
    int ctrl_id,
    int slave_id,
    int gpo_idx,
    char *p_state
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
gpo_idx [in]	汎用出力インデックス
p_state [out]	特定の出力の状態を受け取るバッファへのポインター 出力が「オン」状態の場合、値は 1 になります。それ以外の場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetSlaveGpo
LabVIEW	HIMC Get Slave Gpo.vi
Python	GetSlaveGpo

## 7.5.3 HIMC\_GetSlaveMultiGpi

### 目的

スレーブの複数の汎用入力の状態を取得します

### 構文

```
int HIMC_GetSlaveMultiGpi(
    int ctrl_id,
    int slave_id,
    int start_idx,
    int num,
    char *p_state_array
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
start_idx [in]	取得する汎用入力の開始インデックス
num [in]	取得する汎用入力の数
p_state_array [out]	インデックス範囲 [start_idx, ..., start_idx + num - 1]の汎用入力の状態を受け取るバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetSlaveMultiGpi
LabVIEW	HIMC Get Slave Multi Gpi.vi
Python	GetSlaveMultiGpi

## 7.5.4 HIMC\_GetSlaveMultiGpo

### 目的

スレーブの複数の汎用出力の状態を取得します

### 構文

```
int HIMC_GetSlaveMultiGpo(
    int ctrl_id,
    int slave_id,
    int start_idx,
    int num,
    char *p_state_array
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
start_idx [in]	取得する汎用出力の開始インデックス
num [in]	取得する汎用出力の数
p_state_array [out]	インデックス範囲 [start_idx, ..., start_idx + num – 1]の汎用出力の状態を受け取るバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetSlaveMultiGpo
LabVIEW	HIMC Get Slave Multi Gpo.vi
Python	GetSlaveMultiGpo

7.5.5 HIMC\_GetSlaveGpiNum

目的

スレーブの汎用入力の数を取得します

構文

```
int HIMC_GetSlaveGpiNum(  
    int ctrl_id,  
    int slave_id,  
    int *p_gpi_num  
);
```

パラメーター

ctrl\_id [in]            HIWIN モーションコントローラーのコントローラーID  
                         HIMC\_ConnectCtrl を呼び出して取得する必要があります。

slave\_id [in]            スレーブ ID

p\_gpi\_num [out]    スレーブの汎用入力の数を受け取るバッファーへのポインター

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_GetSlaveGpiNum
LabVIEW	HIMC Get Slave Gpi Num.vi
Python	GetSlaveGpiNum

## 7.5.6 HIMC\_GetSlaveGpoNum

### 目的

スレーブの汎用出力の数を取得します

### 構文

```
int HIMC_GetSlaveGpoNum(
    int ctrl_id,
    int slave_id,
    int *p_gpo_num
);
```

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- slave\_id [in]      スレーブ ID
- p\_gpo\_num [out]      スレーブの汎用出力の数を受け取るバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetSlaveGpoNum
LabVIEW	HIMC Get Slave Gpo Num.vi
Python	GetSlaveGpoNum

(このページはブランクになっています)

## 8. AIO 機能

8.1	概要 .....	8-2
8.2	スレーブ AIO 設定 .....	8-3
8.2.1	HIMC_SetSlvAOMonitor .....	8-3
8.2.2	HIMC_SetSlvAIParam .....	8-5
8.2.3	HIMC_SetSlvAOParam .....	8-6
8.2.4	HIMC_SetSlvAO .....	8-7
8.3	スレーブ AIO ステータス .....	8-8
8.3.1	HIMC_GetSlvAIType .....	8-8
8.3.2	HIMC_GetSlvAIScale .....	8-9
8.3.3	HIMC_GetSlvAIOffset .....	8-10
8.3.4	HIMC_GetSlvAOScale .....	8-11
8.3.5	HIMC_GetSlvAOOffset .....	8-12
8.3.6	HIMC_IsSlvAOBound .....	8-13
8.3.7	HIMC_GetSlvAI .....	8-14
8.3.8	HIMC_GetSlvAO .....	8-15

## 8.1 概要

AIO 機能により、アナログ入力(AI)またはアナログ出力(AO)機能を備えたスレーブは、関連するパラメータを読み取り、設定することができます。現在、AIO 関数は ETA3 にのみ適用されます。ドライバーのアナログ信号ピンの信号操作はサポートできません。ETA3 の AIO モジュールは、4 組のアナログ入力と 8 組のアナログ出力 (4 組の電圧型アナログ出力、4 組の電流型アナログ出力)を提供します。設定に基づいて、アナログ入力は電圧または電流になります。測定・出力できる電圧範囲は±10V、電流範囲は 4~20mA です。詳細な仕様については、「ETA3 インストールガイド」を参照してください。



# 8.2 スレーブ AIO 設定

## 8.2.1 HIMC\_SetSlvAOMonitor

### 目的

コントローラー変数をアナログ出力にバインドするように設定します。

### 構文

```
int HIMC_SetSlvAOMonitor(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ao_idx,
    int var_id
);
```

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- slave\_id [in]      スレーブ ID
- slot\_idx [in]      アナログ出力モジュールの取り付けスロットのインデックス
- ao\_idx [in]      アナログ出力モジュールのアナログ出力チャンネル
- var\_id      コントローラー変数と軸 ID  
次の表は、コントローラー変数と軸 ID の定義を示しています。  
コントローラー変数の詳細については、セクション 16.1.2 を参照してください。  
例：HIMC\_REF\_VEL | HIMC\_AXIS\_2

コントローラー変数	定義	コントローラー変数	定義
HIMC_REF_POS	軸の基準位置	HIMC_POS_FB	軸の位置フィードバック
HIMC_REF_VEL	軸の基準速度	HIMC_VEL_FB	軸の速度フィードバック
HIMC_REF_ACC	軸の基準加速度	HIMC_ACC_FB	軸の加速度フィードバック
		HIMC_CUR_FB	軸の現在のフィードバック

軸 ID	定義
HIMC_AXIS_0	軸 0
HIMC_AXIS_1	軸 1
...	...
HIMC_AXIS_15	軸 15

## 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

## 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

## 他の API 環境の対応名称

C#	HIMC_SetSlvAOMonitor
LabVIEW	HIMC Set Slv AO Monitor.vi
Python	SetSlvAOMonitor

## 8.2.2 HIMC\_SetSlvAIParam

### 目的

スレーブのアナログ入力パラメーターを設定します

### 構文

```
int HIMC_SetSlvAIParam(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ai_idx,
    int ai_type,
    double ai_scale,
    double ai_offset
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
slot_idx [in]	アナログ入力モジュールの取り付けスロットのインデックス
ai_idx [in]	アナログ入力モジュールのアナログ入力チャンネル
ai_type [in]	アナログ入力タイプ 0：電圧タイプ（初期値） 1：電流タイプ
ai_scale [in]	アナログ入力と測定物理量のスケール
ai_offset [in]	アナログ入力と測定物理量のオフセット

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetSlvAIParam
LabVIEW	HIMC Set Slv AI Param.vi
Python	SetSlvAIParam

## 8.2.3 HIMC\_SetSlvAOParam

### 目的

スレーブのアナログ出力パラメーターを設定します

### 構文

```
int HIMC_SetSlvAOParam(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ao_idx,
    int ao_en_bind,
    double ao_scale,
    double ao_offset
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
slot_idx [in]	アナログ出力モジュールの取り付けスロットのインデックス
ao_idx [in]	アナログ出力モジュールのアナログ出力チャンネル
ao_en_bind [in]	0：コントローラー変数にバインドされたアナログ出力の機能は OFF (初期値) 1：コントローラー変数にバインドされたアナログ出力の機能が ON です
ao_scale [in]	アナログ出力とコントローラー変数のスケール
ao_offset [in]	アナログ出力とコントローラー変数のオフセット

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetSlvAOParam
LabVIEW	HIMC Set Slv AO Param.vi
Python	SetSlvAOParam

## 8.2.4 HIMC\_SetSlvAO

### 目的

スレーブのアナログ出力値を設定します

### 構文

```
int HIMC_SetSlvAO(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ao_idx,
    double ao_value
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
slot_idx [in]	アナログ出力モジュールの取り付けスロットのインデックス
ao_idx [in]	アナログ出力モジュールのアナログ出力チャンネル
ao_value [in]	アナログ出力値 パラメーター単位：V または mA

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetSlvAO
LabVIEW	HIMC Set Slv AO.vi
Python	SetSlvAO

## 8.3 スレーブ AIO ステータス

### 8.3.1 HIMC\_GetSlvAIType

#### 目的

スレーブのアナログ入力タイプを取得します

#### 構文

```
int HIMC_GetSlvAIType(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ai_idx,
    int *p_ai_type
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
slot_idx [in]	アナログ入力モジュールの取り付けスロットのインデックス
ai_idx [in]	アナログ入力モジュールのアナログ入力チャンネル
p_ai_type [out]	スレーブのアナログ入力タイプを受け取るバッファーへのポインター 0：電圧タイプ、1：電流タイプ

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetSlvAIType
LabVIEW	HIMC Get Slv AI Type.vi
Python	GetSlvAIType

### 8.3.2 HIMC\_GetSlvAIScale

#### 目的

スレーブのアナログ入力のスケールと測定された物理量を取得します

#### 構文

```
int HIMC_GetSlvAIScale(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ai_idx,
    double *p_ai_scale
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
slot_idx [in]	アナログ入力モジュールの取り付けスロットのインデックス
ai_idx [in]	アナログ入力モジュールのアナログ入力チャンネル
p_ai_scale [out]	スレーブのアナログ入力のスケールと観測された物理量を受け取るバッファへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetSlvAIScale
LabVIEW	HIMC Get Slv AI Scale.vi
Python	GetSlvAIScale

## 8.3.3 HIMC\_GetSlvAIOffset

### 目的

スレーブのアナログ入力オフセットと測定物理量を取得します

### 構文

```
int HIMC_GetSlvAIOffset(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ai_idx,
    double *p_ai_offset
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
slot_idx [in]	アナログ入力モジュールの取り付けスロットのインデックス
ai_idx [in]	アナログ入力モジュールのアナログ入力チャンネル
p_ai_offset [out]	スレーブのアナログ入力オフセットと測定物理量を受け取るバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetSlvAIOffset
LabVIEW	HIMC Get Slv AI Offset.vi
Python	GetSlvAIOffset



### 8.3.4 HIMC\_GetSlvAOScale

#### 目的

アナログ出力とコントローラー変数のスレーブのスケールを取得します

#### 構文

```
int HIMC_GetSlvAOScale(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ao_idx,
    double *p_ao_scale
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
slot_idx [in]	アナログ出力モジュールの取り付けスロットのインデックス
ao_idx [in]	アナログ出力モジュールのアナログ出力チャンネル
p_ao_scale [out]	スレーブのアナログ出力のスケールとコントローラー変数を受け取るバッファーへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetSlvAOScale
LabVIEW	HIMC Get Slv AO Scale.vi
Python	GetSlvAOScale

## 8.3.5 HIMC\_GetSlvAOOffset

### 目的

アナログ出力とコントローラー変数のスレーブのオフセットを取得します

### 構文

```
int HIMC_GetSlvAOOffset(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ao_idx,
    double *p_ao_offset
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
slot_idx [in]	アナログ出力モジュールの取り付けスロットのインデックス
ao_idx [in]	アナログ出力モジュールのアナログ出力チャンネル
p_ao_offset [out]	アナログ出力とコントローラー変数のスレーブのオフセットを受け取るバッファ ーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetSlvAOOffset
LabVIEW	HIMC Get Slv AO Offset.vi
Python	GetSlvAOOffset

### 8.3.6 HIMC\_IsSlvAOBound

#### 目的

スレーブのアナログ出力がコントローラー変数にバインドされているかどうかを照会します

#### 構文

```
int HIMC_IsSlvAOBound(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ao_idx,
    int *p_is_bound
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
slot_idx [in]	アナログ出力モジュールの取り付けスロットのインデックス
ao_idx [in]	アナログ出力モジュールのアナログ出力チャンネル
p_is_bound [out]	スレーブのアナログ出力のバインドされたステータスを受け取るバッファーへのポインター スレーブが「SlvAOBound」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_IsSlvAOBound
LabVIEW	HIMC Is Slv AO Bound.vi
Python	IsSlvAOBound

## 8.3.7 HIMC\_GetSlvAI

### 目的

スレーブのアナログ入力の物理量測定値を取得します

### 構文

```
int HIMC_GetSlvAI(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ai_idx,
    double *p_ai_value
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
slot_idx [in]	アナログ入力モジュールの取り付けスロットのインデックス
ai_idx [in]	アナログ入力モジュールのアナログ入力チャンネル
p_ai_value [out]	スレーブのアナログ入力の測定された物理量値を受け取るバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetSlvAI
LabVIEW	HIMC Get Slv AI.vi
Python	GetSlvAI

### 8.3.8 HIMC\_GetSlvAO

#### 目的

スレーブのアナログ出力値を取得します

#### 構文

```
int HIMC_GetSlvAO(
    int ctrl_id,
    int slave_id,
    int slot_idx,
    int ao_idx,
    double *p_ao_value
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ ID
slot_idx [in]	アナログ出力モジュールの取り付けスロットのインデックス
ao_idx [in]	アナログ出力モジュールのアナログ出力チャンネル
p_ao_value [out]	スレーブのアナログ出力値を受け取るバッファへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetSlvAO
LabVIEW	HIMC Get Slv AO.vi
Python	GetSlvAO

(このページはブランクになっています)

## 9. ユーザーテーブル機能

9.1	概要 .....	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 概要

HIMC は、最大 512,000 個の double 型変数データ (500K バイト) を格納できる空きメモリ領域をユーザーに提供します。この章で提供される関数を使用すると、ユーザーはメモリ空間にアクセスできます。書き込まれた値は、コントローラーのランダムアクセスメモリ (RAM) に保存されます。関数「HIMC\_SaveUserTable」を使用すると、メモリ空間のユーザーテーブルのデータが HIMC のハードディスク空間に保存されます。HIMC の電源を入れ直した後、関数「HIMC\_LoadUserTable」を使用して、保存されたデータがユーザーテーブルのメモリ空間に再度コピーされます。

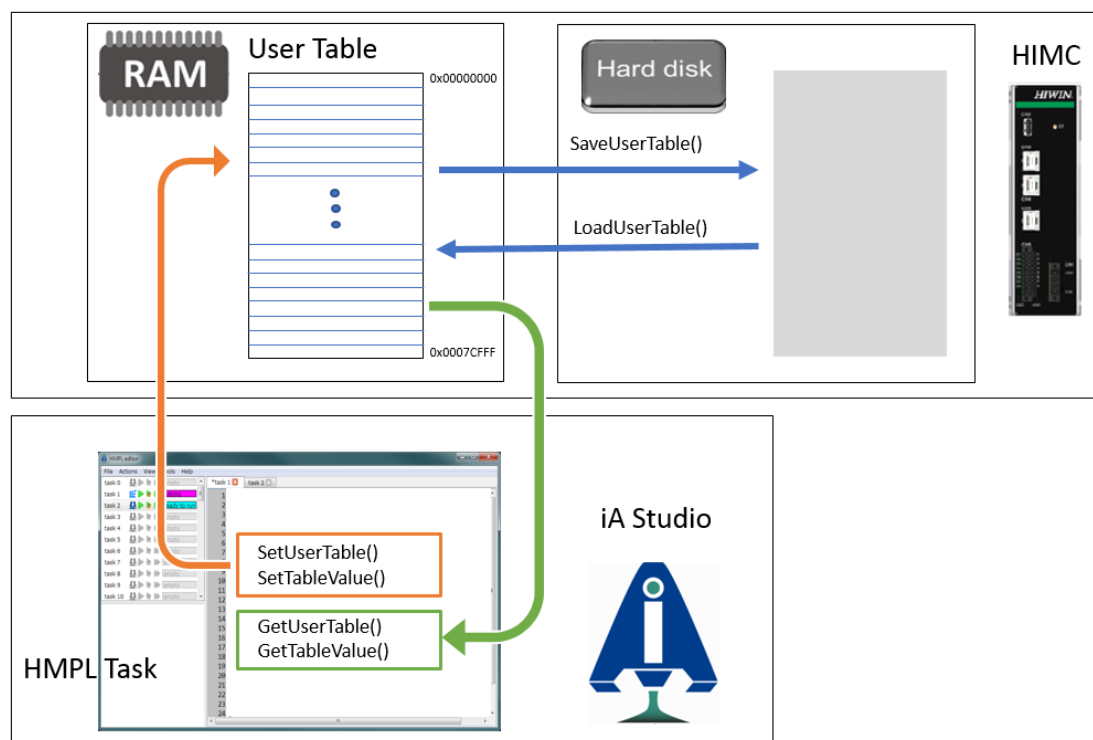


図 9.1.1

注：ユーザーは、iA Studio のテーブルビューアーを介してユーザーテーブルの変数値にアクセスできます(「iA Studio ユーザーガイド」のセクション 4.11 を参照)。これには、HIMC のメモリとハードディスクへの読み込みと保存が含まれます。

注意：

動的エラー補正関数で使用するエラーマップは、ユーザーテーブルのメモリ空間に保存されます。動的エラー補正を有効にする場合、ユーザーは他のユーザーテーブル値へのアクセスが構築されたエラー補正值に影響を与えないようにする必要があります。



## 9.2 HIMC\_SetUserTable

### 目的

ユーザーテーブルデーターをコントローラーに設定します

### 構文

```
int HIMC_SetUserTable(
    int    ctrl_id,
    double *p_user_table_data,
    int    start_idx,
    int    number_of_doubles_to_write
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
p_user_table_data [in]	データーをユーザーテーブルに書き込むためのバッファへのポインター
start_idx [in]	ユーザーテーブルの開始インデックス
number_of_doubles_to_write [in]	書き込まれるユーザーテーブルの数

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetUserTable
LabVIEW	HIMC Set User Table.vi
Python	SetUserTable

## 9.3 HIMC\_GetUserTable

### 目的

コントローラーからユーザーテーブルデータを取得します

### 構文

```
int HIMC_GetUserTable(
    int    ctrl_id,
    double *p_user_table_data,
    int    start_idx,
    int    number_of_doubles_to_read
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
p_user_table_data [out]	ユーザーテーブルからデータを受け取るバッファーへのポインター
start_idx [in]	ユーザーテーブルの開始インデックス
number_of_doubles_to_read [in]	読み取るユーザーテーブルの数

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetUserTable
LabVIEW	HIMC Get User Table.vi
Python	GetUserTable

## 9.4 HIMC\_SetTableValue

### 目的

ユーザーテーブルの特定のインデックスにデータを書き込みます

### 構文

```
int HIMC_SetTableValue(
    int    ctrl_id,
    int    index,
    double value
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
index [in]	ユーザーテーブルのインデックス
value [in]	入力データ

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetTableValue
LabVIEW	HIMC Set Table Value.vi
Python	SetTableValue

## 9.5 HIMC\_GetTableValue

### 目的

ユーザーテーブルの特定のインデックスからデータを取得します

### 構文

```
int HIMC_GetTableValue(
    int    ctrl_id,
    int    index,
    double *value
);
```

### パラメーター

**ctrl\_id [in]**           HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**index [in]**            ユーザーテーブルのインデックス

**value [out]**           データを受け取るバッファへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetTableValue
LabVIEW	HIMC Get Table Value.vi
Python	GetTableValue

## 9.6 HIMC\_SaveUserTable

### 目的

ユーザーテーブルデータを RAM のメモリに保存します

### 構文

```
int HIMC_SaveUserTable(
    int ctrl_id,
    int start_idx,
    int num_data
);
```

### パラメーター

**ctrl\_id [in]**      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**start\_idx [in]**      ユーザーテーブルの開始インデックス

**num\_data [in]**      格納する要素の数

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SaveUserTable
LabVIEW	HIMC Save User Table.vi
Python	SaveUserTable

9.7 HIMC\_LoadUserTable

目的

ユーザーテーブルデータをメモリから RAM にロードします

構文

```
int HIMC_LoadUserTable(  
    int ctrl_id,  
    int start_idx,  
    int num_data  
);
```

パラメーター

- ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
                      HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- start\_idx [in]        ユーザーテーブルの開始インデックス
- num\_data [in]        格納する要素の数

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_LoadUserTable
LabVIEW	HIMC Load User Table.vi
Python	LoadUserTable

## 10. ポジショントリガー機能

10.1	概要 .....	10-2
10.1.1	PT 変数 .....	10-2
10.1.2	PT 機能の利用の流れ .....	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 概要

HMPL コマンドを使用すると、ユーザーは一部の HIWIN ドライバーで PT(ポジショントリガー)関連の機能を操作できます。PT 関連の機能を操作する前に、互換性のあるドライブについて HIWIN または地域の代理店にお問い合わせください。

注：HIWIN ドライバーが PT 関連機能を使用するための要件は、(1)デジタルエンコーダー、(2)完全な原点復帰手順です。

### 10.1.1 PT 変数

PT 関連の関数は、表 10.1.1.1 に示す変数に基づいて動作します。

表 10.1.1.1

名称	タイプ	単位	説明
status	int	true / false	PT 機能のステータス。PT がまだ機能しているかどうかを示します。
start position	double	mm または deg	PT 機能の開始位置。PT 出力信号列はここから始まります。
end position	double	mm または deg	PT 機能の終了位置。この時点以降、PT 出力信号は送信されません。
interval	double	mm または deg	連続する PT 出力間の位置間隔
pulse width	int	ns	各 PT 出力信号の幅 1. D1-N シリーズドライバーの場合、範囲 25 ns から 100,000 ns で、最小増分は 25 ns です。たとえば、25、50、... 100,000 ns です。 2. E1 シリーズドライバーの場合、範囲は 20 ns ~ 80,000 ns で、最小増分は 20 ns です。たとえば、20、40、... 80,000 ns です。
position array	double	mm または deg	ランダムインターバル PT 機能のトリガー位置
status array	int	high / low	ランダムインターバル PT 機能のステータス出力
start index	int	-	ランダム間隔 PT 機能の開始インデックス
end index	int	-	ランダム間隔 PT 機能の終了インデックス



図 10.1.1.1 では、極性は「アクティブハイ」に設定されています。

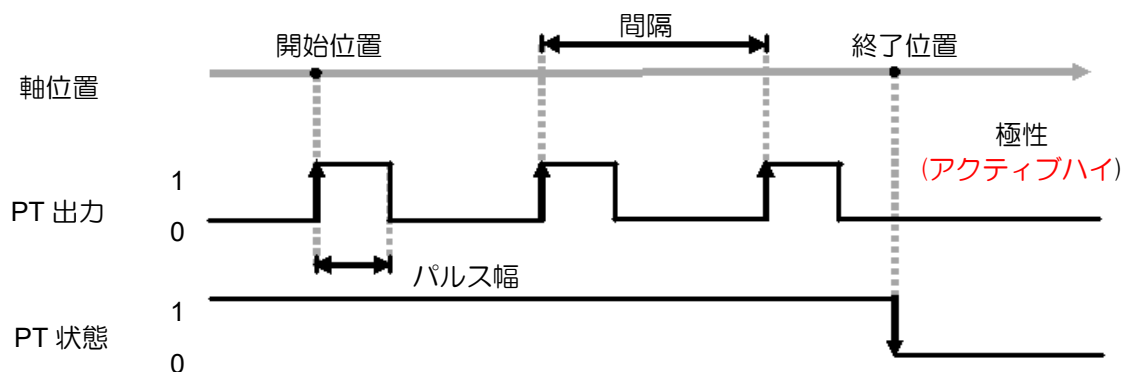


図 10.1.1.1

図 10.1.1.2 では、極性は「アクティブロー」に設定されています。

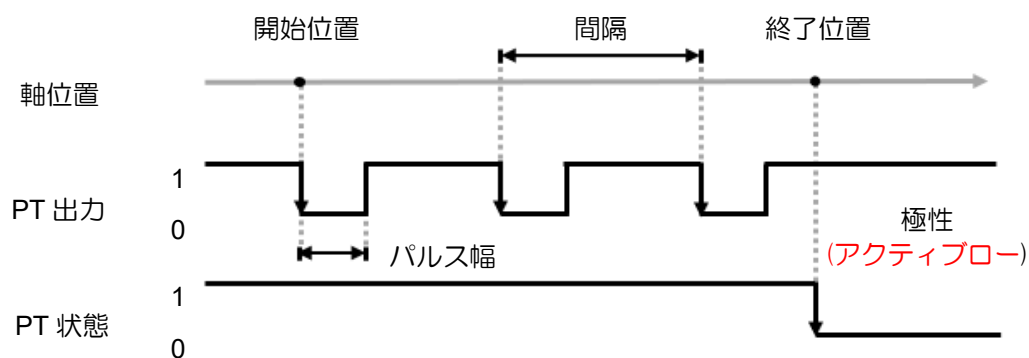


図 10.1.1.2

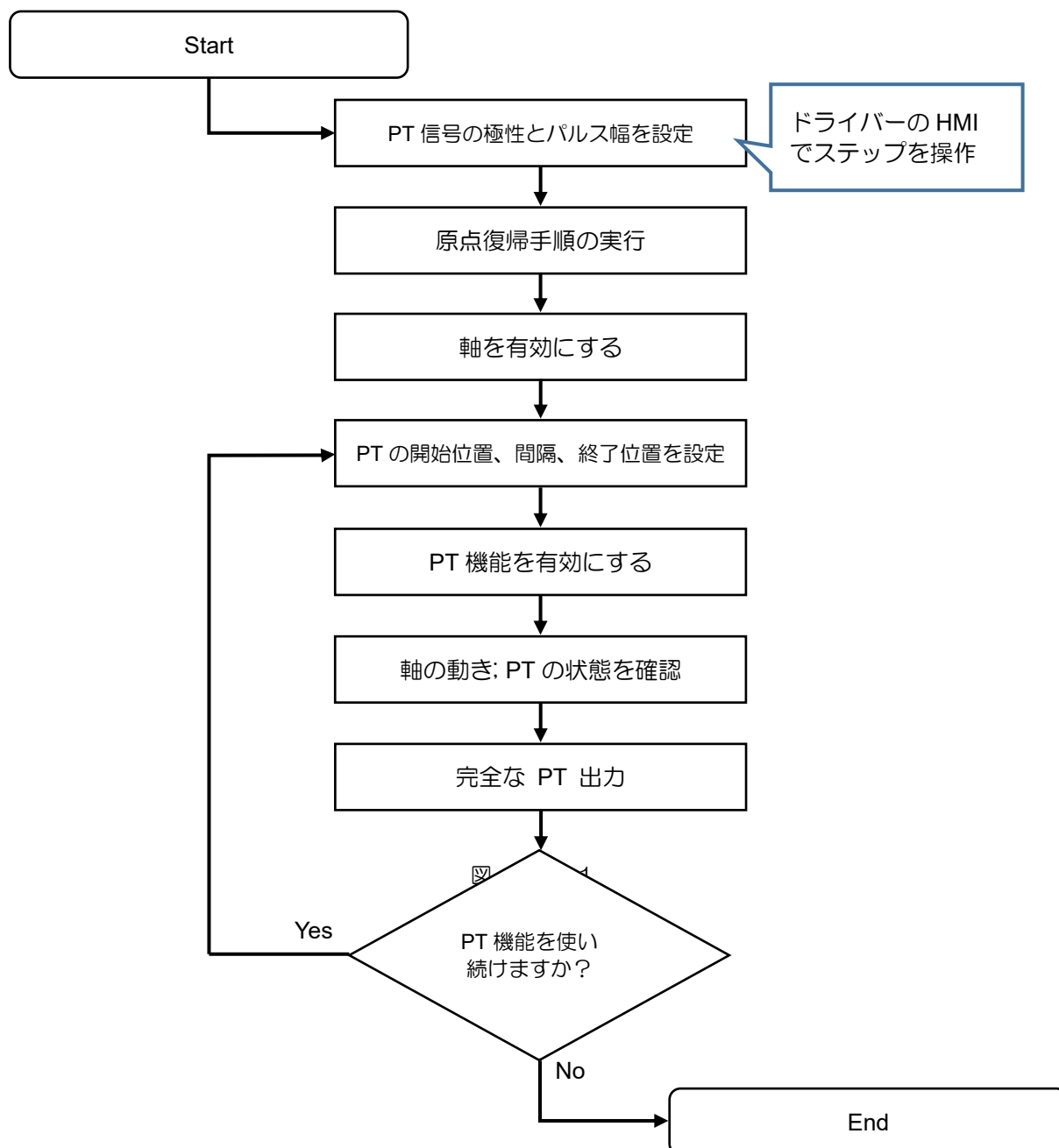
制限：

PT 機能の間隔と軸の速度は、式「速度 < 間隔 × 位置サンプリングレート」に適合する必要があります。間隔が 100 μm に設定され、位置のサンプリングレートが 16 K の場合、速度は 1600 mm/s 未満である必要があります。

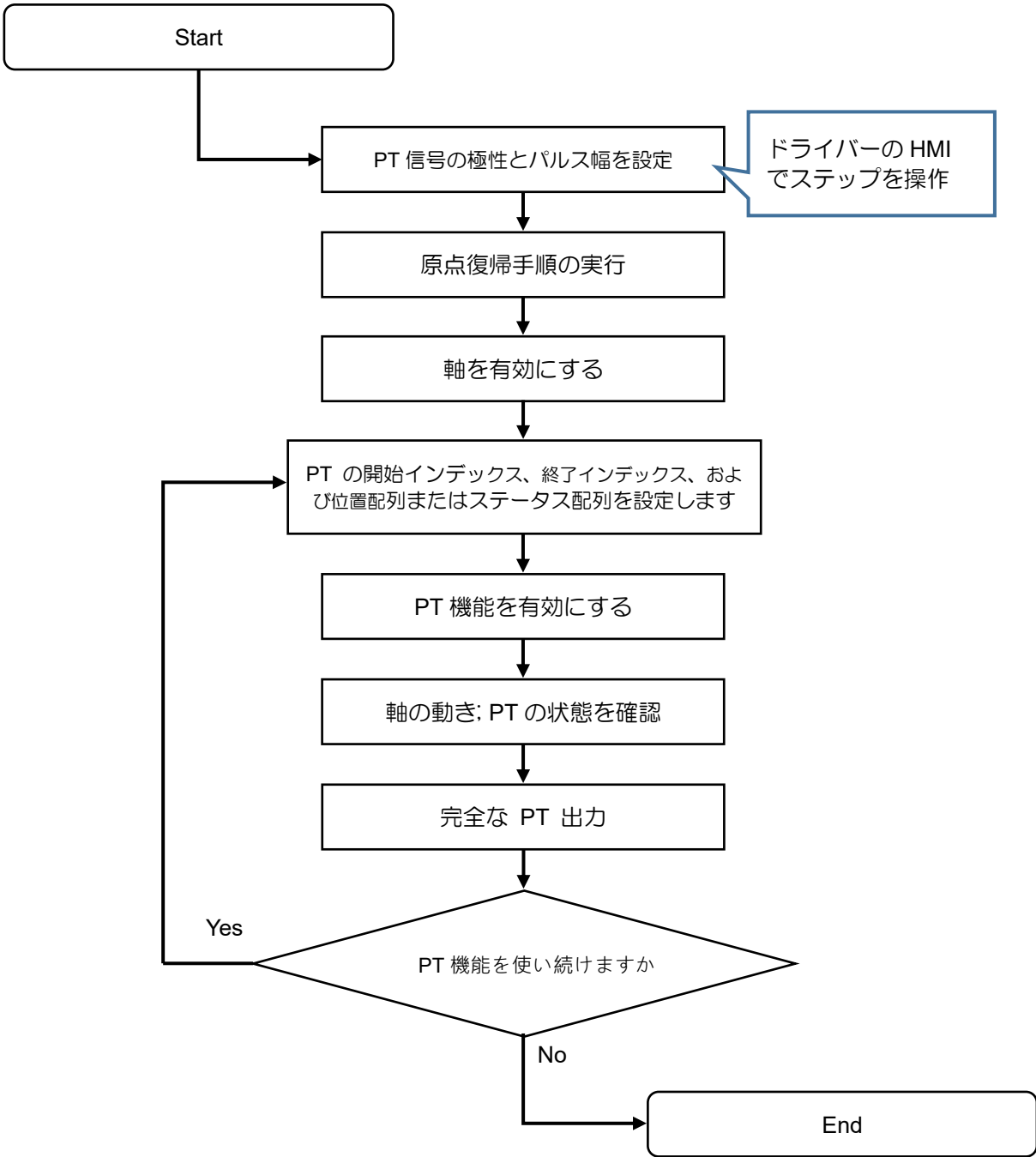
注：PT 機能の出力極性(アクティブ ハイ/ロー)を調整するには、ドライバーの HMI に移動して設定します。設定保存後、ドライバーの電源を入れ直して出力極性を有効にします。

## 10.1.2 PT 機能の利用の流れ

### ◆ 定間隔 PT 機能



◆ ランダムインターバル PT 機能



10.2 HIMC\_EnablePT

目的

軸の位置トリガー機能を有効にします

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_EnablePT
LabVIEW	HIMC Enable PT.vi
Python	EnablePT

## 10.3 HIMC\_DisablePT

### 目的

軸の位置トリガー機能を無効にします

### 構文

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

### パラメーター

**ctrl\_id [in]**            HIWIN モーションコントローラーのコントローラーID  
                         HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**axis\_id [in]**            Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_DisablePT
LabVIEW	HIMC Disable PT.vi
Python	DisablePT

## 10.4 HIMC\_IsPTEnabled

### 目的

位置トリガー機能が有効かどうかを問い合わせます

### 構文

```
int HIMC_IsPTEnabled(
    int ctrl_id,
    int axis_id,
    int *p_is_pt_enabled
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_pt_enabled [out]	位置トリガーの有効状態を受け取るバッファーへのポインター 軸が「PTEnabled」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_IsPTEnabled
LabVIEW	HIMC Is PT Enabled.vi
Python	IsPTEnabled

## 10.5 HIMC\_SetPosTriggerConfig

### 目的

位置トリガー構成を軸に設定します

### 構文

```
int HIMC_SetPosTriggerConfig(  
    int ctrl_id,  
    int axis_id,  
    PosTriggerPar *pos_trigger_par  
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
pos_trigger_par [in]	軸に設定された位置トリガー構成を含むバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetPosTriggerConfig
LabVIEW	HIMC Set Pos Trigger Config.vi
Python	SetPosTriggerConfig

10.6 HIMC\_SetPT\_PosArray

目的

ランダム位置トリガー機能のトリガー位置を設定します

構文

```
int HIMC_SetPT_PosArray(  
    int    ctrl_id,  
    int    axis_id,  
    int    index,  
    double trigger_pos  
);
```

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index
- index [in]

トリガー位置のインデックス
- trigger\_pos [in]

複数のトリガー位置を含む配列  
パラメーター単位：mm または deg

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_SetPT_PosArray
LabVIEW	HIMC Set PT Pos Array.vi
Python	SetPT_PosArray



## 10.7 HIMC\_SetPT\_StateArray

### 目的

ランダム位置トリガー機能のステータス出力を設定します

### 構文

```
int HIMC_SetPT_StateArray(  
    int ctrl_id,  
    int axis_id,  
    int index,  
    int state  
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
index [in]	ステータス出力のインデックス
state [in]	複数のステータス出力を含む配列

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetPT_StateArray
LabVIEW	HIMC Set PT State Array.vi
Python	SetPT_StateArray

## 10.8 HIMC\_SetPT\_StartIndex

### 目的

ランダムポジショントリガー機能の開始インデックスを設定します

### 構文

```
int HIMC_SetPT_StartIndex(
    int ctrl_id,
    int axis_id,
    int index
);
```

### パラメーター

- ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]      Axis index
- index [in]      インデックスを開始します

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetPT_StartIndex
LabVIEW	HIMC Set PT Start Index.vi
Python	SetPT_StartIndex

## 10.9 HIMC\_SetPT\_EndIndex

### 目的

ランダムポジショントリガー機能の終了インデックスを設定します

### 構文

```
int HIMC_SetPT_EndIndex(  
    int ctrl_id,  
    int axis_id,  
    int index  
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
index [in]	インデックスを終了します

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetPT_EndIndex
LabVIEW	HIMC Set PT End Index.vi
Python	SetPT_EndIndex

(このページはブランクになっています)

# 11. タッチプローブ機能

11.1	概要 .....	11-2
11.2	HIMC_EnableTouchProbe .....	11-3
11.3	HIMC_DisableTouchProbe .....	11-4
11.4	HIMC_IsTouchProbeEnabled .....	11-5
11.5	HIMC_IsTouchProbeTriggered .....	11-6
11.6	HIMC_GetTouchProbePos .....	11-7

## 11.1 概要

タッチプローブ機能は、原点復帰手順で使用されるラッチ機能です (AC モーター、ダイレクトドライブモーター、リニアモーターに適用可能)。エンコーダー入力信号のエッジトリガーでエンコーダーの位置フィードバック値を取り込みます。図 11.1.1 に示すように、ドライバーがエンコーダーインデックス信号を通過すると、タッチプローブ機能がトリガーされ、インデックス信号の位置が記録されます。タッチプローブ機能を使用すると、ユーザーはタッチプローブ機能がトリガーされているかどうかをコントローラーに問い合わせることができ、コントローラーはラッチによって記録されたインデックス信号の位置を取得できます。

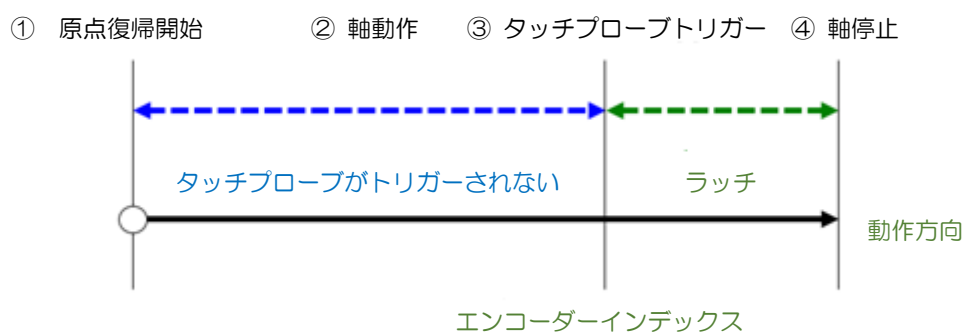


図 11.1.1

## 11.2 HIMC\_EnableTouchProbe

### 目的

軸のタッチプローブ機能を有効にします

### 構文

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

### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

axis\_id [in]      Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_EnableTouchProbe
LabVIEW	HIMC Enable Touch Probe.vi
Python	EnableTouchProbe

11.3 HIMC\_DisableTouchProbe

目的

軸のタッチプローブ機能を無効にします

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_DisableTouchProbe
LabVIEW	HIMC Disable Touch Probe.vi
Python	DisableTouchProbe



## 11.4 HIMC\_IsTouchProbeEnabled

### 目的

タッチプローブ機能が有効かどうかを問い合わせます

### 構文

```
int HIMC_IsTouchProbeEnabled(  
    int ctrl_id,  
    int axis_id,  
    int *p_is_probe_enabled  
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_probe_enabled [out]	タッチプローブの有効状態を受け取るバッファへのポインター 軸が「TouchProbeEnabled」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_IsTouchProbeEnabled
LabVIEW	HIMC Is Touch Probe Enabled.vi
Python	IsTouchProbeEnabled

## 11.5 HIMC\_IsTouchProbeTriggered

### 目的

タッチプローブがトリガーされているかどうかを問い合わせます

### 構文

```
int HIMC_IsTouchProbeTriggered(
    int ctrl_id,
    int axis_id,
    int *p_is_probe_triggered
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_is_probe_triggered [out]	タッチプローブのトリガー状態を受け取るバッファへのポインター 軸が「TouchProbeTriggered」状態にある場合、値は 1 になります。 それ以外の場合は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_IsTouchProbeTriggered
LabVIEW	HIMC Is Touch Probe Triggered.vi
Python	IsTouchProbeTriggered

## 11.6 HIMC\_GetTouchProbePos

### 目的

軸のタッチプローブ位置を取得します

### 構文

```
int HIMC_GetTouchProbePos(  
    int    ctrl_id,  
    int    axis_id,  
    double *p_get_probe_pos  
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
p_get_probe_pos [out]	軸のタッチプローブ位置を受け取るバッファーへのポインター パラメーター単位：mm または deg

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

軸がガントリーモードの場合、この機能は適用されません。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetTouchProbePos
LabVIEW	HIMC Get Touch Probe Pos.vi
Python	GetTouchProbePos

(このページはブランクになっています)

## 12. ダイナミックエラー補償機能

12.1	概要 .....	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 .....	12-9
12.7	HIMC_GetCompPos.....	12-10
12.8	HIMC_SetCompAlgType.....	12-11

## 12.1 概要

HIMC は、動的な 1D / 2D / 3D エラー補正機能を提供します。関連するエラー測定と計算結果に基づいて、ユーザーはエラーマップを作成し、HIMC で設定を行うことができます。設定パラメーターには、補正軸、参照軸、マップ点の間隔、マップ点のベース、マップ点の数、および各マップ点の補正值が含まれます。補正值の設定に関しては、HIMC ユーザーテーブルのメモリ空間が使用され、エラーマップの最初の ID 位置がユーザーテーブルに提供されます。

注 1：ユーザーテーブルの使用方法と説明については、9 章を参照してください。

注 2：動的エラー補正を有効にする前に、軸は原点復帰を完了して補正軸と基準軸の座標位置を固定する必要があります。

ユーザーは、同じ軸を補正軸と基準軸の両方として選択することも、複数の異なる軸を補正軸の基準軸として選択することもできます。たとえば、補正軸は Z 軸で、基準軸は X 軸と Y 軸です。補正軸の補正值は、基準軸の移動位置に応じて変化します。軸の移動中、確立されたエラーマップは、線形補間を使用してマップポイント間の補正コマンド値を計算するため、補正值が連続的に維持されます。軸の位置がエラーマップによって確立された範囲を超えると、図 12.1.1 に示すように、HIMC は最も近いマップポイントの補正值を補正コマンドとして使用します。

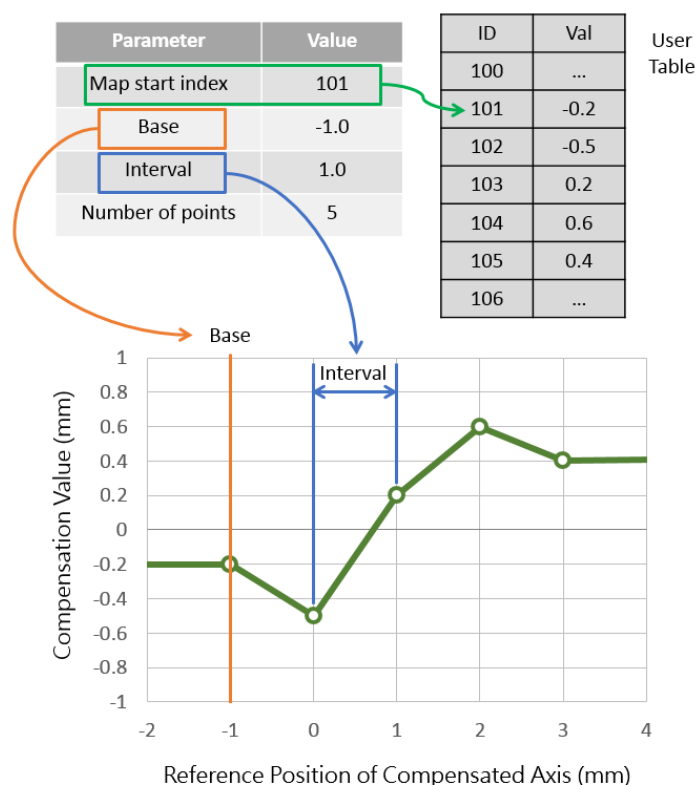


図 12.1.1

動的誤差補正を有効にした後、コントローラーの軸制御コマンドで、出力基準位置は、既知の測定誤差を排除するために補正される変位を追加します。図 3.1.1 に示すように、関係は次のように記述できます：

$$\text{基準位置} + \text{位置補正} = \text{位置出力}$$

動的エラー補正を有効にすると、ユーザーは iA Studio のスコープマネージャーを介して変数を観察できます。

- 補正值：軸→モーション変数→位置補正
- 基準位置（補正なし）：軸 → モーション変数 → 基準位置
- 基準位置（補正あり）：軸→モーション変数→位置出力

制限：

HIMC では、補正コマンドはプロファイルジェネレーターを経由せず、コントローラーによってプリセットされる最大補正值は 1 mm です。補正值が 1 mm より大きい場合、システムはエラーメッセージを表示してユーザーに通知します。

動的エラー補正を有効にする場合、補正する基準座標を固定する必要があります。したがって、ユーザーは軸の原点オフセットを変更できません。

## 12.2 HIMC\_EnableComp

### 目的

軸の動的エラー補正を有効にします

### 構文

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

### パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

軸が有効な場合、この機能は適用されません。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_EnableComp
LabVIEW	HIMC Enable Comp.vi
Python	EnableComp



## 12.3 HIMC\_DisableComp

### 目的

軸の動的エラー補正を無効にします

### 構文

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

### パラメーター

**ctrl\_id [in]**           HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**axis\_id [in]**           Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

- (1) 電流フィードバックとして、軸の基準位置がリセットされます。
- (2) 軸が有効な場合、この機能は適用されません。
- (3) 動的誤差補正の設定がクリアされます。  
動的エラー補正を再度有効にするには、設定をリセットする必要があります。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_DisableComp
LabVIEW	HIMC Disable Comp.vi
Python	DisableComp

## 12.4 HIMC\_SetupComp

### 目的

軸の 1 次元の動的誤差補正を設定します

### 構文

```
int HIMC_SetupComp(
    int    ctrl_id,
    int    axis_id,
    int    start_idx,
    double base_val,
    double interval,
    int    num_pt,
    int    ref_axis_id
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
start_idx [in]	ユーザーテーブル内のマップポイントの開始インデックス
base_val [in]	ベース値（マップ入力の最小値） パラメーター単位：mm または deg
interval [in]	隣接するマップポイント間の一定の間隔 パラメーター単位：mm または deg
num_pt [in]	マップポイントの数
ref_axis_id [in]	基準軸のインデックス

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_SetupComp
LabVIEW	HIMC Setup Comp.vi
Python	SetupComp

## 12.5 HIMC\_SetupComp2D

### 目的

軸の 2 次元動的エラー補正を設定します

### 構文

```
int HIMC_SetupComp2D(
    int    ctrl_id,
    int    axis_id,
    int    start_idx,
    double *base_val,
    double *interval,
    int    *num_pt,
    int    *ref_axis_id
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
start_idx [in]	ユーザーテーブル内のマップポイントの開始インデックス
base_val [in]	各次元のベース値(マップ入力の最小値)を含む 2 要素配列へのポインター パラメーター単位：mm または deg
interval [in]	隣接するマップポイント間の各次元の一定間隔を含む 2 要素配列へのポインター パラメーター単位：mm または deg
num_pt [in]	各次元のマップポイント数を含む 2 要素配列へのポインター
ref_axis_id [in]	各次元の参照軸のインデックスを含む 2 要素配列へのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetupComp2D
LabVIEW	HIMC Setup Comp2D.vi
Python	SetupComp2D

## 12.6 HIMC\_SetupComp3D

### 目的

軸の 3 次元動的誤差補正を設定します

### 構文

```
int HIMC_SetupComp3D(
    int    ctrl_id,
    int    axis_id,
    int    start_idx,
    double *base_val,
    double *interval,
    int    *num_pt,
    int    *ref_axis_id
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
start_idx [in]	ユーザーテーブル内のマップポイントの開始インデックス
base_val [in]	各次元のベース値(マップ入力の最小値)を含む 3 要素配列へのポインター パラメーター単位: mm または deg
interval [in]	隣接するマップポイント間の各次元の一定間隔を含む 3 要素配列へのポインター パラメーター単位: mm または deg
num_pt [in]	各次元のマップポイント数を含む 3 要素配列へのポインター
ref_axis_id [in]	各次元の参照軸のインデックスを含む 3 要素配列へのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.4
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetupComp3D
LabVIEW	HIMC Setup Comp3D.vi
Python	SetupComp3D

## 12.7 HIMC\_GetCompPos

### 目的

コントローラーからドライバーに送信された軸の誤差補正值を取得します

### 構文

```
int HIMC_GetCompPos(  
    int    ctrl_id,  
    int    group_id,  
    double *p_comp_pos  
);
```

### パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- group\_id [in]

Axis group index
- p\_comp\_pos [out]

軸の誤差補正值を受け取るバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetCompPos
LabVIEW	HIMC Get Comp Pos.vi
Python	GetCompPos

## 12.8 HIMC\_SetCompAlgType

### 目的

軸の動的誤差補正の補間方法を設定します

### 構文

```
int HIMC_SetCompAlgType(  
    int ctrl_id,  
    int axis_id,  
    int alg_type  
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
alg_type [in]	動的エラー補正の補間方法 0：一次線形補間（初期値） 1：3 次スプライン補間

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

3 次元動的誤差補正は、3 次スプライン補間をサポートしていません。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetCompAlgType
LabVIEW	HIMC Set Comp Alg Type.vi
Python	SetCompAlgType

(このページはブランクになっています)



## 13. フィルター機能

13.1	概要 .....	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 概要

フィルター関数は、プロファイルジェネレーターの位置コマンドを修正するために使用されます。現在、HMPL は、平滑時間、振動抑制フィルター(VSF)、入力整形フィルター(InShape)の 3 種類のフィルターを提供しています。

Smooth time はモーターをスムーズに加速させて滑らかな動きを実現し、VSF と InShape は移動中のモーターの振動(特に機構の負荷が片持ちの場合)を抑制します。「周波数」と「減衰比」をチューニングすることで、制振効果が得られます。

VSF と InShape は同時に使用できませんが、どちらかをスムーズに使用できます。

その上、協調運動になると、Axis InShape 関数は役に立ちません。ユーザーは、振動を抑えるために Group InShape 機能を採用する必要があります。

注：フィルターを使用すると、移動時間が長くなり、デバウンス時間が短くなります。

## 13.2 HIMC\_EnableAxisVsf

### 目的

軸の VSF フィルターを有効にします

### 構文

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

### パラメーター

ctrl\_id [in]      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

axis\_id [in]      Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

モーターが動いているときは、この機能は適用されません。モーターが予期せぬ振動を発生させます。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_EnableAxisVsf
LabVIEW	HIMC Enable Axis Vsf.vi
Python	EnableAxisVsf

### 13.3 HIMC\_DisableAxisVsf

目的

軸の VSF フィルターを無効にします

構文

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

パラメーター

- ctrl\_id [in]

HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]

Axis index

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_DisableAxisVsf
LabVIEW	HIMC Disable Axis Vsf.vi
Python	DisableAxisVsf

## 13.4 HIMC\_SetAxisVsf

### 目的

軸の VSF フィルターのパラメーターを設定します

### 構文

```
int HIMC_SetAxisVsf(
    int ctrl_id,
    int axis_id,
    double frequency,
    double damping_ratio
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
frequency [in]	システム周波数 パラメーター単位：Hz 入力範囲：0.1 ～ 200
damping_ratio [in]	減衰比 入力範囲：0.7 ～ 1.5 (1.0 を推奨)

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

モーターが動いているときは、この機能は適用されません。モーターが予期せぬ振動を発生させます。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

## 他の API 環境の対応名称

C#	HIMC_SetAxisVsf
LabVIEW	HIMC Set Axis Vsf.vi
Python	SetAxisVsf

## 13.5 HIMC\_EnableAxisInShape

### 目的

軸の InShape フィルターを有効にします

### 構文

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

### パラメーター

ctrl\_id [in]                      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

axis\_id [in]                      Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

モーターが動いているときは、この機能は適用されません。モーターが予期せぬ振動を発生させます。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_EnableAxisInshape
LabVIEW	HIMC Enable Axis Inshape.vi
Python	EnableAxisInshape

### 13.6 HIMC\_DisableAxisInShape

目的

軸の InShape フィルターを無効にします

構文

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

パラメーター

- ctrl\_id [in]
- HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]
- Axis index

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_DisableAxisInshape
LabVIEW	HIMC Disable Axis Inshape.vi
Python	DisableAxisInshape



## 13.7 HIMC\_SetAxisInShape

### 目的

軸の InShape フィルターのパラメーターを設定します

### 構文

```
int HIMC_SetAxisInShape(  
    int ctrl_id,  
    int axis_id,  
    double frequency,  
    double damping_ratio,  
    ShaperMode shaper_type  
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
frequency [in]	システム周波数 パラメーター単位：Hz 入力範囲：3.0 ～ 300
damping_ratio [in]	減衰比 入力範囲：0.0 ～ 0.3
shaper_type [in]	Shaper_Normal と Shaper_Robust の 2 つのシェーパータイプがあります。 Shaper_Robust は Shaper_Normal よりも堅牢ですが、Shaper_Normal は振動を抑えるのに十分な強度があります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

- (1) この機能は、モーターが動いているときは適用できません。モーターが予期せぬ振動を発生させます。
- (2) 周波数と減衰比の初期値は、それぞれ 5.5Hz と 0.03 です。

## 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

## 他の API 環境の対応名称

C#	HIMC_SetAxisInshape
LabVIEW	HIMC Set Axis Inshape.vi
Python	SetAxisInshape

## 13.8 HIMC\_EnableGrpInShape

### 目的

軸グループの InShape フィルターを有効にします

### 構文

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

### パラメーター

ctrl\_id [in]                      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

group\_id [in]                    Axis group index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

モーターが動いているときは、この機能は適用されません。モーターが予期せぬ振動を発生させます。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_EnableGrpInShape
LabVIEW	HIMC Enable Grp In Shape.vi
Python	EnableGrpInShape

### 13.9 HIMC\_DisableGrpInShape

目的

軸グループの InShape フィルターを無効にします

構文

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

パラメーター

ctrl\_id [in]                      HIWIN モーションコントローラーのコントローラーID  
                                 HIMC\_ConnectCtrl を呼び出して取得する必要があります。

group\_id [in]                    Axis group index

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_DisableGrpInShape
LabVIEW	HIMC Disable Grp In Shape.vi
Python	DisableGrpInShape

## 13.10 HIMC\_SetGrpInShape

### 目的

軸グループの InShape フィルターのパラメーターを設定します

### 構文

```
int HIMC_SetGrpInShape(  
    int ctrl_id,  
    int group_id,  
    double frequency,  
    double damping_ratio,  
    ShaperMode shaper_type  
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
group_id [in]	Axis group index
frequency [in]	システム周波数 パラメーター単位：Hz 入力範囲：3.0 ～ 300
damping_ratio [in]	減衰比 入力範囲：0.0 ～ 0.3
shaper_type [in]	Shaper_Normal と Shaper_Robust の 2 つのシェーパータイプがあります。 Shaper_Robust は Shaper_Normal よりも堅牢ですが、Shaper_Normal は振動を抑えるのに十分な強度があります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

- (1) この機能は、モーターが動いているときは適用できません。モーターが予期せぬ振動を発生させます。
- (2) 周波数と減衰比の初期値は、それぞれ 5.5Hz と 0.03 です。

## 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

## 他の API 環境の対応名称

C#	HIMC_SetGrpInShape
LabVIEW	HIMC Set Grp In Shape.vi
Python	SetGrpInShape

# 14. HMPL タスク機能

14.1	概要 .....	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 概要

HIMC には、ユーザーがアプリケーションに基づいてモーションプロファイルコマンドを練習するための 64 のビルトイン HMPL タスクがあります。どの HMPL タスクでも、ユーザーは HMPL タスク機能を使用して他の HMPL タスクを開始または停止できます。HMPL タスクが実行されている場合、ユーザーは再実行を要求できません。代わりに、ユーザーはタスクの実行が完了し、タスクが「停止」ステータスになるまで待つ必要があります。ただし、ユーザーは HMPL タスクが現在実行されているかどうかを照会し、それに応じてアプリケーションの複数の HMPL タスクの順序を制御できます。



## 14.2 HIMC\_StartTask

### 目的

HMPL タスクの実行を開始します

### 構文

```
int HIMC_StartTask(
    int ctrl_id,
    int task_id
);
```

### パラメーター

ctrl\_id [in]            HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

task\_id [in]            HMPL タスク ID

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_StartTask
LabVIEW	HIMC Start Task.vi
Python	StartTask

14.3 HIMC\_StartTaskFunc

目的

HMPL タスクで関数の実行を開始します

構文

```
int HIMC_StartTaskFunc(  
    int ctrl_id,  
    int task_id,  
    char *func_name  
);
```

パラメーター

ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
                      HIMC\_ConnectCtrl を呼び出して取得する必要があります。

task\_id [in]           HMPL タスク ID

func\_name [in]       HMPL タスクに関数名を格納するためのバッファーへのポインター

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_StartTaskFunc
LabVIEW	HIMC Start Task Func.vi
Python	StartTaskFunc

## 14.4 HIMC\_StopTask

### 目的

HMPL タスクの実行を停止します

### 構文

```
int HIMC_StopTask(
    int ctrl_id,
    int task_id
);
```

### パラメーター

ctrl\_id [in]            HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

task\_id [in]            HMPL タスク ID

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_StopTask
LabVIEW	HIMC Stop Task.vi
Python	StopTask

# 14.5 HIMC\_StopAllTask

目的

すべての HMPL タスク(呼び出し元を含む)の実行を停止します

構文

```
int HIMC_StopAllTask(  
    int ctrl_id  
);
```

パラメーター

ctrl\_id [in]            HIWIN モーションコントローラーのコントローラーID  
                         HIMC\_ConnectCtrl を呼び出して取得する必要があります。

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_StopAllTask
LabVIEW	HIMC Stop All Task.vi
Python	StopAllTask

## 14.6 HIMC\_IsTaskStop

### 目的

HMPL タスクの実行が停止されているかどうかを照会します

### 構文

```
int HIMC_IsTaskStop(
    int ctrl_id,
    int task_id,
    int *isStop
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
task_id [in]	HMPL タスク ID
isStop [out]	HMPL タスクのステータスを受け取るバッファへのポインター HMPL タスクが「TaskStop」状態にある場合、値は 1 になります。それ以外の場合、値は 0 になります。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_IsTaskStop
LabVIEW	HIMC Is Task Stop.vi
Python	IsTaskStop

## 14.7 HIMC\_LoadHMPLTask

### 目的

HMPL ファイルをコントローラーにロードします

### 構文

```
int HIMC_LoadHMPLTask(
    int ctrl_id,
    int task_id,
    const char *file_name
);
```

### パラメーター

- ctrl\_id [in]**           HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- task\_id [in]**           HMPL タスク ID
- file\_name [in]**       読み込まれる HMPL ファイルパスを格納するバッファーへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

この機能は、「iA\_Studio」フォルダ内の実行ファイル「HMPL\_compiler.exe」で使用する必要があります。実行ファイルをコピーし、ユーザー開発プログラムの「bin¥Debug」および「bin¥Release」フォルダに配置します。

### 要件

サポートされる最小バージョン	iA Studio 2.0
Executable	HMPL_compiler.exe
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_LoadHMPLTask
LabVIEW	HIMC Load HMPL Task.vi
Python	LoadHMPLTask

# 15. コールバック機能

15.1	HIMC_SetHmplEvtCallback.....	15-2
15.2	HIMC_SetErrorCallback .....	15-3
15.3	HIMC_SetHmplMsgEvtCallback .....	15-4

15.1 HIMC\_SetHmplEvtCallback

目的

HMPL タスクから送信されたイベントを取得するためのコールバック関数を登録します

構文

```
int HIMC_SetHmplEvtCallback(  
    int ctrl_id,  
    HMPLEventCBFuncPtr hmpl_event_cb_func_ptr  
);
```

パラメーター

- ctrl\_id [in]
- HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- hmpl\_event\_cb\_func\_ptr [in]
- コールバック関数へのポインター  
関数のプロトタイプは void func(int arg)です。

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_SetHmplEvtCallback
LabVIEW	--
Python	SetHmplEvtCallback



## 15.2 HIMC\_SetErrorCallback

### 目的

コントローラーから送信されたエラーID を取得するためのコールバック関数を登録します

### 構文

```
int HIMC_SetErrorCallback(  
    int ctrl_id,  
    HimcErrorCBFuncPtr himc_error_cb_func_ptr  
);
```

### パラメーター

ctrl\_id [in]                      HIWIN モーションコントローラーのコントローラーID  
                                 HIMC\_ConnectCtrl を呼び出して取得する必要があります。

himc\_error\_cb\_func\_ptr [in]      コールバック関数へのポインター  
                                 関数のプロトタイプは「void func(int arg)」です。

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 備考

エラーID については、「iA Studio ユーザーガイド」の 5 章を参照してください。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetErrorCallback
LabVIEW	--
Python	SetErrorCallback

15.3 HIMC\_SetHmplMsgEvtCallback

目的

HMPL タスクから送信された文字列メッセージを取得するためのコールバック関数を登録します。

構文

```
int HIMC_SetHmplMsgEvtCallback(  
    int ctrl_id,  
    HmplMsgEventCBFuncPtr hmpl_msg_cb_func_ptr  
);
```

パラメーター

- ctrl\_id [in]
- HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- hmpl\_msg\_cb\_func\_ptr [in]
- コールバック関数へのポインター  
関数のプロトタイプは「void func(const char \*text)」です。

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

備考

文字列メッセージの最大長は 128 文字です。

要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_SetHmplMsgEvtCallback
LabVIEW	--
Python	SetHmplMsgEvtCallback

## 16. 変数と関数演算機能

16.1	概要 .....	16-2
16.1.1	ドライバー変数一覧 .....	16-3
16.1.2	コントローラー変数一覧 .....	16-4
16.2	ドライバー可変運転 .....	16-7
16.2.1	HIMC_GetSlvVar .....	16-7
16.2.2	HIMC_SetSlvVar .....	16-8
16.2.3	HIMC_GetSlvSt .....	16-9
16.2.4	HIMC_SetSlvSt .....	16-10
16.3	ドライバー機能の運転 .....	16-11
16.3.1	HIMC_RunSlvPdIFunc .....	16-11
16.3.2	HIMC_StopSlvPdIFunc .....	16-12
16.3.3	HIMC_IsSlvPdIFuncRunning .....	16-13
16.4	コントローラー変数操作 .....	16-14
16.4.1	HIMC_GetVariableByID .....	16-14
16.4.2	HIMC_SetVariableByID .....	16-15
16.4.3	HIMC_GetVariableListByID .....	16-16
16.4.4	HIMC_SetVariableListByID .....	16-17
16.4.5	HIMC_GetGlobalVariables .....	16-18
16.4.6	HIMC_SetGlobalVariables .....	16-19

## 16.1 概要

HIMC は、ドライバーとコントローラーの変数操作機能をユーザーに提供します。ドライバーの変数操作では、ドライバー変数の文字列を指定してその値にアクセスし、MoE 通信を介してデータ交換を行う必要があります。コントローラーの変数操作に関しては、コントローラーの変数 ID リストに基づいて、特定の変数のアドレッシング ID を指定してアクセスする必要があります。ドライバーとコントローラーのパラメーターの定義は、それぞれセクション 16.1.1 と 16.1.2 に記載されています。

### 注意：

特定のパーボーズが必要ない場合は、ユーザーが関連する HMI と機能を使用して関連するシステム変数にアクセスすることをお勧めします。変数操作機能を使用する場合、ユーザーは変数へのアクセスと値の入力の安全性を確保する必要があります。

## 16.1.1 ドライバー変数一覧

E1 シリーズドライバーの Pt パラメーターの説明については、「E1 シリーズドライバーユーザーマニュアル」の 15 章を参照してください。表 16.1.1.1 に示す Pt パラメーターには、関連する機能のビット定義があるため、パラメーター名は PtXXX.all にする必要があります。その他のパラメーターについては、PtXXX を使用します。

表 16.1.1.1

基本機能設定用 パラメーター (Pt0XX)	チューニング用 パラメーター (Pt1XX)	位置関連 パラメーター (Pt2XX)	トルク関連 パラメーター ( Pt4XX )	I/O 設定用 パラメーター (Pt5XX)	回生抵抗設定 パラメーター (Pt6XX)
Pt000.all	Pt10B.all	Pt200.all	Pt408.all	Pt50A.all	Pt70A.all
Pt001.all	Pt139.all	Pt207.all	Pt416.all	Pt50B.all	Pt70B.all
Pt002.all	Pt140.all	Pt22A.all	Pt423.all	Pt50C.all	Pt710.all
Pt006.all	Pt170.all			Pt50D.all	
Pt007.all				Pt50E.all	
Pt008.all				Pt50F.all	
Pt009.all				Pt511.all	
Pt010.all				Pt512.all	
Pt00A.all				Pt513.all	
Pt00B.all				Pt514.all	
Pt00D.all				Pt515.all	
Pt00E.all				Pt516.all	
Pt00F.all				Pt517.all	
				Pt519.all	
				Pt51A.all	

変数操作関数を使用して変数にアクセスする場合、ユーザーは、変数の変更によって生成される安全性の問題を含め、設定された変数の正確性を確認する必要があります。関連する操作要件を除き、可変書き込みエラーによるドライバーの機能への影響を避けるために、Thunder HMI を使用することをお勧めします。

注：現在、この章で提供されている関数は、D シリーズドライバーのパラメーターをサポートしていません。関連する HMI および機能からパラメーターにアクセスしてください。

## 16.1.2 コントローラー変数一覧

HIMC は、コントローラー変数のアドレス指定 ID として 32 ビットを取ります。そのタイプは 0x□□□□□□□□で、「0x」は値が 16 進数であることを示します。変数演算機能により、HIMC が提供するシステム変数、軸変数、軸グループ変数にアクセスできます。ID のアドレッシングの規則は、次のように説明されます：

1. アドレッシング ID の 1 番目と 2 番目の値は、「コントローラー変数のカテゴリ」を示します。0x00□□□□□□はシステム変数に属します。0x83□□□□□□は軸変数に属します。0x82□□□□□□は軸グループ変数に属します。
2. アドレッシング ID の 3 番目と 4 番目の値は、「軸 ID または軸グループ ID」を示します。たとえば、軸変数 0x8302□□□□は Axis index.02 を格納する変数であり、軸グループ変数 0x8201□□□□は Axis group index.01 を格納する変数です。
3. アドレッシング ID の 5 ～ 8 番目の値は、「コントローラーのシステム、軸または軸グループ変数のアドレッシング位置」を示します。パラメーターのリストと説明については、表 16.1.2.1 から表 16.1.2.3 を参照してください。

表 16.1.2.1

システム変数		
アドレッシング ID	変数名	説明
0x0000012c	HCV_ID_fclk	システム実行クロック（250us あたり 1 カウントアップ）
0x0000012e	HCV_ID_timeInMs	システム実行時間 (ms)
0x000007d0	HCV_ID_user_table	ユーザーが自由に使える double[512000]配列変数
0x00002328	HCV_ID_ltest0	ユーザーが自由に使える int 変数
0x00002329	HCV_ID_ltest1	ユーザーが自由に使える int 変数
0x0000232a	HCV_ID_ltest2	ユーザーが自由に使える int 変数
0x0000232b	HCV_ID_ltest3	ユーザーが自由に使える int 変数
0x0000232c	HCV_ID_ltest4	ユーザーが自由に使える int 変数
0x0000232d	HCV_ID_ltest5	ユーザーが自由に使える int 変数
0x0000232e	HCV_ID_ltest6	ユーザーが自由に使える int 変数
0x0000232f	HCV_ID_ltest7	ユーザーが自由に使える int 変数
0x00002330	HCV_ID_ltest8	ユーザーが自由に使える int 変数
0x00002331	HCV_ID_ltest9	ユーザーが自由に使える int 変数
0x0000235a	HCV_ID_dtest0	ユーザーが自由に使える double 変数
0x0000235b	HCV_ID_dtest1	ユーザーが自由に使える double 変数
0x0000235c	HCV_ID_dtest2	ユーザーが自由に使える double 変数
0x0000235d	HCV_ID_dtest3	ユーザーが自由に使える double 変数
0x0000235e	HCV_ID_dtest4	ユーザーが自由に使える double 変数
0x0000235f	HCV_ID_dtest5	ユーザーが自由に使える double 変数
0x00002360	HCV_ID_dtest6	ユーザーが自由に使える double 変数

システム変数		
アドレッシング ID	変数名	説明
0x00002361	HCV_ID_dtest7	ユーザーが自由に使える double 変数
0x00002362	HCV_ID_dtest8	ユーザーが自由に使える double 変数
0x00002363	HCV_ID_dtest9	ユーザーが自由に使える double 変数
0x0000238c	HCV_ID_mtest	ユーザーが自由に使える double[10] 配列変数

表 16.1.2.2

軸変数		
アドレッシング ID	変数名	説明
0x83□□0015	HCV_ID_motion_type	モーションタイプ
0x83□□0033	HCV_ID_pos_tr	インポジション収束半径
0x83□□0034	HCV_ID_pos_tr_t	インポジション整定時間
0x83□□0065	HCV_ID_sw_RL	ソフトウェア右側リミット
0x83□□0066	HCV_ID_sw_LL	ソフトウェア左側リミット
0x83□□0067	HCV_ID_vel_lim	最大速度制限
0x83□□0068	HCV_ID_acc_lim	最大加速度制限
0x83□□0069	HCV_ID_dec_lim	最大減速度限界
0x83□□0079	HCV_ID_max_pos_err	位置誤差限界
0x83□□007a	HCV_ID_max_comp_lim	位置補正限界
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	スムーズ時間
0x83□□00db	HCV_ID_vel_scale	速度スケール (0~100)
0x83□□00dd	HCV_ID_p2p_del	P2P モーション待ち時間
0x83□□00de	HCV_ID_p2p_pos1	P2P ポジション 1
0x83□□00df	HCV_ID_p2p_pos2	P2P ポジション 2
0x83□□00e0	HCV_ID_p2p_repeat	P2P モーションを繰り返す
0x83□□00e1	HCV_ID_rlt_dist	相対移動距離
0x83□□00e2	HCV_ID_en_motionManager	Motion Manager の動作軸選択
0x83□□00e3	HCV_ID_acc_time	加速時間
0x83□□00e4	HCV_ID_dec_time	減速時間
0x83□□00e9	HCV_ID_map_io_type	誤差補正タイプ
0x83□□0117	HCV_ID_rollover_turns	ロールオーバーターン
0x83□□0119	HCV_ID_rollover_val	ロールオーバー値
0x83□□012d	HCV_ID_servo_type	サーボ制御モード (CSP : 0、PP : 5)
0x83□□0193	HCV_ID_gant_pair	ガントリー構成のガントリーペア ID
0x83□□01f7	HCV_ID_en_delay	軸有効化のタイムアウト
0x83□□01ff	HCV_ID_fb_ratio_pos	ドライバの位置分解能: 長さの単位 (分母)

軸変数		
アドレッシング ID	変数名	説明
0x83□□0200	HCV_ID_fb_ratio_cnt	ドライバーの位置分解能: 単位: カウント (分子)
0x83□□0263	HCV_ID_last_err	軸エラーコード
0x83□□03c2	HCV_ID_gear_ratio	ギア比

注：記号□□は 16 進形式の軸 ID になります。たとえば、01 は Axis index.01 を表します。0f は Axis index.15 を表します。

表 16.1.2.3

軸グループ変数		
アドレッシング ID	変数名	説明
0x82□□0002	HCV_ID_grp_num_axis	軸グループの軸数
0x82□□00c9	HCV_ID_grp_lin_vel_lim	軸グループの速度制限
0x82□□00ca	HCV_ID_grp_lin_acc_lim	軸グループの加速度制限
0x82□□00cb	HCV_ID_grp_lin_dec_lim	軸グループの減速度制限
0x82□□00d0	HCV_ID_grp_lin_vel	軸グループの目標速度
0x82□□00d1	HCV_ID_grp_lin_acc	軸グループの目標加速度
0x82□□00d2	HCV_ID_grp_lin_dec	軸グループの目標減速度
0x82□□00d3	HCV_ID_grp_lin_sf	軸グループのスムーズ時間
0x82□□00f0	HCV_ID_grp_lin_acc_time	軸グループの目標加速時間
0x82□□00f1	HCV_ID_grp_lin_dec_time	軸グループの目標減速時間
0x82□□00e4	HCV_ID_grp_coord_sys	軸グループの座標系
0x82□□00e5	HCV_ID_grp_buffer_mode	軸グループのバッファモード
0x82□□00e6	HCV_ID_grp_trans_mode	軸グループの Transition モード
0x82□□00e7	HCV_ID_grp_trans_vel	軸グループの Transition 速度
0x82□□00e8	HCV_ID_grp_trans_dis	軸グループの Tramsition 距離
0x82□□00eb	HCV_ID_grp_vel_scale	軸グループの速度スケール (0~100)
0x82□□00da	HCV_ID_grp_shaper_fr	軸グループの InShape フィルターの頻度
0x82□□00db	HCV_ID_grp_shaper_xi	軸グループの InShape フィルターの減衰比
0x82□□038f	HCV_ID_grp_last_err	軸群エラーコード

注：記号□□は 16 進形式の軸グループ ID になります。たとえば、01 は Axis group index.01 を表します。0f は、Axis group index.15 を表します。



## 16.2 ドライバー可変運転

### 16.2.1 HIMC\_GetSlvVar

#### 目的

スレーブの変数値を取得します

#### 構文

```
int HIMC_GetSlvVar(  
    int ctrl_id,  
    int slave_id,  
    const char *var_name,  
    double *p_var  
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ Axis
var_name [in]	変数名を格納するバッファへのポインター
p_var [out]	変数の値を受け取るバッファへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetSlvVar
LabVIEW	HIMC Get Slv Var.vi
Python	GetSlvVar

## 16.2.2 HIMC\_SetSlvVar

### 目的

スレーブの変数値を設定します

### 構文

```
int HIMC_SetSlvVar(
    int ctrl_id,
    int slave_id,
    const char *var_name,
    double value
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ index
var_name [in]	変数名を格納するバッファへのポインター
value [in]	変数の新しい値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetSlvVar
LabVIEW	HIMC Set Slv Var.vi
Python	SetSlvVar

### 16.2.3 HIMC\_GetSlvSt

#### 目的

スレーブの状態を取得します

#### 構文

```
int HIMC_GetSlvSt(
    int ctrl_id,
    int slave_id,
    const char *st_name,
    int *on_off
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ index
st_name [in]	状態名を格納するバッファへのポインター
on_off [out]	スレーブの状態を受け取るバッファへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetSlvSt
LabVIEW	HIMC Get Slv St.vi
Python	GetSlvSt

## 16.2.4 HIMC\_SetSlvSt

### 目的

スレーブの状態を設定します

### 構文

```
int HIMC_SetSlvSt(
    int ctrl_id,
    int slave_id,
    const char *st_name,
    int on_off
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ index
st_name [in]	状態名を格納するバッファーへのポインター
on_off [in]	スレーブの新しい状態

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetSlvSt
LabVIEW	HIMC Set Slv St.vi
Python	SetSlvSt

## 16.3 ドライバー機能の運転

### 16.3.1 HIMC\_RunSlvPdlFunc

#### 目的

スレーブの PDL 機能を実行します

#### 構文

```
int HIMC_RunSlvPdlFunc(
    int ctrl_id,
    int slave_id,
    const char *input_pdl_func_name
);
```

#### パラメーター

ctrl\_id [in]                      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

slave\_id [in]                      スレーブ index

input\_pdl\_func\_name [in]      実行する PDL 関数名を格納するバッファーへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 備考

この関数を呼び出すとき、ユーザーはスレーブが PDL 関数を本当に実行したことを確認する必要があります。そうしないと、この関数は失敗します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_RunSlvPdlFunc
LabVIEW	HIMC Run Slv Pdl Func.vi
Python	RunSlvPdlFunc

16.3.2 HIMC\_StopSlvPdlFunc

目的

スレーブの PDL 機能の実行を停止します

構文

```
int HIMC_StopSlvPdlFunc(  
    int ctrl_id,  
    int slave_id,  
    const char *input_pdl_func_name  
);
```

パラメーター

- ctrl\_id [in]
- HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- slave\_id [in]
- スレーブ index
- input\_pdl\_func\_name [in]
- 実行を停止する PDL 関数名を格納するバッファへのポインター

戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

備考

この関数を呼び出すとき、ユーザーはスレーブが PDL 関数を本当に実行したことを確認する必要があります。そうしないと、この関数は失敗します。

要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_StopSlvPdlFunc
LabVIEW	HIMC Stop Slv Pdl Func.vi
Python	StopSlvPdlFunc

### 16.3.3 HIMC\_IsSlvPdlFuncRunning

#### 目的

スレーブの PDL 機能がまだ実行中かどうかを問い合わせます

#### 構文

```
int HIMC_IsSlvPdlFuncRunning(
    int ctrl_id,
    int slave_id,
    const char *input_pdl_func_name,
    int *p_ret_is_running
);
```

#### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
slave_id [in]	スレーブ index
input_pdl_func_name [in]	実行状況を問い合わせる PDL 関数名を格納するバッファへのポインター
p_ret_is_running [out]	PDL 関数の実行ステータスを受け取るバッファへのポインター PDL 関数が「SlvPdlFuncRunning」状態にある場合、値は 1 になります。 それ以外の場合、値は 0 になります。

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 1.3
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_IsSlvPdlFuncRunning
LabVIEW	HIMC Is Slv Pdl Func Running.vi
Python	IsSlvPdlFuncRunning

## 16.4 コントローラー変数操作

### 16.4.1 HIMC\_GetVariableByID

#### 目的

コントローラーの変数値を ID で取得します

#### 構文

```
int HIMC_GetVariableByID(  
    int    ctrl_id,  
    int    var_id,  
    double *p_val  
);
```

#### パラメーター

- ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
                  HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- var\_id [in]           HIMC コントローラー変数 ID  
                  定義については、セクション 16.1.2 を参照してください。
- p\_val [out]           変数の値を受け取るバッファーへのポインター

#### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

#### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

#### 他の API 環境の対応名称

C#	HIMC_GetVariableByID
LabVIEW	HIMC Get Variable By ID.vi
Python	GetVariableByID



## 16.4.2 HIMC\_SetVariableByID

### 目的

コントローラーの変数値を ID で設定します

### 構文

```
int HIMC_SetVariableByID(
    int    ctrl_id,
    int    var_id,
    double val
);
```

### パラメーター

- ctrl\_id [in]**      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- var\_id [in]**      HIMC コントローラー変数 ID  
定義については、セクション 16.1.2 を参照してください。
- val [in]**          変数の新しい値

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetVariableByID
LabVIEW	HIMC Set Variable By ID.vi
Python	SetVariableByID

## 16.4.3 HIMC\_GetVariableListByID

### 目的

コントローラーのリスト変数の値を ID で取得します

### 構文

```
int HIMC_GetVariableListByID(
    int    ctrl_id,
    int    *p_var_id,
    int    num,
    double *p_val
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
p_var_id [in]	リスト変数の HIMC コントローラー変数 ID を格納するバッファへのポインター 定義については、セクション 16.1.2 を参照してください。
num [in]	変数の数
p_val [out]	リスト変数の値を受け取るバッファへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetVariableListByID
LabVIEW	HIMC Get Variable List By ID.vi
Python	GetVariableListByID

## 16.4.4 HIMC\_SetVariableListByID

### 目的

値を設定して、コントローラーの変数を ID でリストします

### 構文

```
int HIMC_SetVariableListByID(
    int    ctrl_id,
    int    *p_var_id,
    int    num,
    double *p_val
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
p_var_id [in]	リスト変数の HIMC コントローラー変数 ID を格納するバッファへのポインター 定義については、セクション 16.1.2 を参照してください。
num [in]	変数の数
p_val [in]	リスト変数の新しい値を格納するバッファへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_SetVariableListByID
LabVIEW	HIMC Set Variable List By ID.vi
Python	SetVariableListByID

## 16.4.5 HIMC\_GetGlobalVariables

### 目的

コントローラーのリストグローバル変数の値を取得します

### 構文

```
int HIMC_GetGlobalVariables(
    int    ctrl_id,
    char    **pp_var_name_array,
    int    length,
    double *p_output_array
);
```

### パラメーター

- ctrl\_id [in]                      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- pp\_var\_name\_array [in]          リストグローバル変数の名前を格納するバッファへのポインター
- length [in]                      グローバル変数の数
- p\_output\_array [out]            リストグローバル変数の値を受け取るバッファへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Executable	HMPL_compiler.exe
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetGlobalVariables
LabVIEW	HIMC Get Global Variables.vi
Python	GetGlobalVariables

## 16.4.6 HIMC\_SetGlobalVariables

### 目的

値を設定して、コントローラーのグローバル変数をリストします

### 構文

```
int HIMC_SetGlobalVariables(
    int    ctrl_id,
    char    **pp_var_name_array,
    int    length,
    double *p_input_array
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
pp_var_name_array [in]	リストグローバル変数の名前を格納するバッファへのポインター
length [in]	グローバル変数の数
p_input_array [in]	リストグローバル変数の新しい値を格納するバッファへのポインター

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Executable	HMPL_compiler.exe
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetGlobalVariables
LabVIEW	HIMC Get Global Variables.vi
Python	GetGlobalVariables

(このページはブランクになっています)

## 17. HIMC エラー機能

17.1	概要 .....	17-2
17.1.1	システムエラーメッセージ .....	17-3
17.1.2	軸エラーメッセージ .....	17-6
17.1.3	グループのエラーメッセージ .....	17-8
17.2	HIMC_GetLastError.....	17-10
17.3	HIMC_GetAxisLastErr .....	17-11
17.4	HIMC_ClearAxisLastErr.....	17-12
17.5	HIMC_GetGrpLastErr .....	17-13
17.6	HIMC_ClearGrpLastErr .....	17-14
17.7	HIMC_GetErrorInformation.....	17-15

## 17.1 概要

HIMC は、関連するエラーメッセージを表す 32 ビットエラーコードを提供します。この章で提供される機能を使用して、ユーザーはシステム、軸、および軸グループのエラーコードを取得またはクリアできます。(各カテゴリのエラーコード、名前、および説明は、セクション 17.1.1 ~ 17.1.3 に記載されています。) エラーコードのタイプは 0x□□□□□□□□で、「0x」は値が 16 進数であることを示します。その規則は、次のように説明されるコントローラー変数のアドレス指定 ID の規則と同じです。

1. エラーコードの 1 番目と 2 番目の値は、「コントローラー変数のカテゴリ」を示します。0x00□□□□□□はシステム変数に属します。0x83□□□□□□は軸変数に属します。0x82□□□□□□は軸グループ変数に属します。
2. エラーコードの 3 番目と 4 番目の値は、「軸 ID または軸グループ ID」を示します。たとえば、軸変数 0x8302□□□□は Axis index.02 を格納する変数であり、軸グループ変数 0x8201□□□□は Axis group index.01 を格納する変数です。
3. エラーコードの 5 ~ 8 番目の値は「変数 ID」を示します。詳細については、セクション 17.1.1 ~ 17.1.3 を参照してください。

注：関数の戻り値は 10 進数であるため、正しいエラーコードを取得するには、ユーザーが自分でそれを 16 進数に変換する必要があります。



## 17.1.1 システムエラーメッセージ

表 17.1.1.1

システムエラーコード		
エラーコード	エラー名	説明
0x00000001	eERR_HCV_ID_NOT_FOUND	変数 ID が見つかりませんでした
0x00000002	eERR_DATA_EXCEEDED	要求されたデータは範囲外です
0x00000003	eERR_HCV_IS_READ_ONLY	読み取り専用パラメーター
0x00000004	eERR_HCV_VALUE_OUT_OF_RANGE	入力値が範囲外です
0x00000064	eERR_EMERGENCY_STOP	非常停止が作動しました。すべての軸を無効にし、すべてのタスクを停止します。
0x000000ff	eERR_MOE_NOT_READY	MoE は準備ができていません
0x00000100	eERR_MAIL_BOX_BUSY	コントローラーとスレーブ間のメールボックスがビジーです
0x00000101	eERR_VAR_NOT_IN_SLV_DB	スレーブ変数が見つかりませんでした
0x00000102	eERR_VAR_NOT_REGYET	スレーブ変数を読み取ることができません
0x00000103	eERR_READ_VAR_NO_RECV	スレーブからの応答はありませんでした
0x00000104	eERR_PREV_SLV_CMD_NOT_FIN	スレーブへの前のコマンドが終了していません
0x00000105	eERR_SLV_ID_INVALID	スレーブ ID が無効です
0x00000106	eERR_PDO_NUM_EXCEED	PDO の数が範囲外です
0x00000107	eERR_NOT_VALID_TASKID	タスク ID が無効です
0x00000108	eERR_TASK_IS_RUNNING	タスクはすでに実行中です
0x00000109	eERR_FUNC_NOT_IN_TASK	関数がタスクに見つかりませんでした
0x0000010a	eERR_TASK_EMPTY	タスクは空です
0x0000010b	eERR_TASK_NOT_RUNNING	タスクは実行されていません
0x0000012c	eERR_NIC_INIT_TOUT	mega-ulink のネットワークポートの準備ができていません
0x0000012d	eERR_HARDWARE_MISMATCH	ハードウェアが認識されていません
0x0000012e	eERR_SLAVE_NUM_MISMATCH	スレーブの数は構成とは異なります
0x0000012f	eERR_INVALID_PDO	PDO が無効です
0x00000130	eERR_INVALID_MCK_CNFG	モーションカーネルの構成が無効です
0x00000136	eERR_MOE_SEND_FAIL	mega-ulink パケットの送信に失敗しました
0x00000137	eERR_MOE_RECV_FAIL	mega-ulink パケットの受信に失敗しました
0x00000138	eERR_HIMC_LOAD_CONFIG_FAIL	SSD からの構成のロードに失敗しました。もう一度保存してください
0x00000139	eERR_HIMC_SAVE_CONFIG_FAIL	HIMC への構成の保存に失敗しました。もう一度保存してください
0x0000013a	eERR_HIMC_SAVE_CONFIG_COPY_FAIL	HIMC への構成の保存に失敗しました。ファイルを SAVE フォルダに保存できません。
0x0000013b	eERR_HIMC_SAVE_UPDATE_PRM_TIMEOUT	HIMC への構成の保存に失敗しました。Prm 値の更新タイムアウト。
0x000001f4	eERR_ISR_NOT_STABLE	割り込みの周期は安定していません
0x000001f5	eERR_MCK_OVERLOAD	モーションカーネルが過負荷になっています
0x000001f6	eERR_ISR_OVERLOAD	CPU が過負荷になっています
0x000001f7	eERR_MOE_ISR_NOT_STABLE	MoE では、割り込みの周期が安定しません
0x000003e8	eERR_PP_MODE_NOT_SUPPORTED	この機能は PP モードではサポートされていません

システムエラーコード		
エラーコード	エラー名	説明
0x00001388	eERR_HMPL_INVALID_ARG	引数は HMPL では無効です
0x00001389	eERR_HMPL_INVALID_PTR	ポインタは HMPL では無効です
0x0000138a	eERR_HMPL_STACK_OVERFLOW	HMPL のスタックオーバーフロー
0x0000138b	eERR_HMPL_ILLEGAL_MEM_OP	HMPL ではメモリの操作は違法です
0x0000138c	eERR_HMPL_MOTION_NOT_READY	モーション関数は同期状態で呼び出す必要があります
0x0000138d	eERR_HMPL_STR_TOO_LONG	文字列の長さが範囲外です
0x0000138e	eERR_HMPL_INVALID_STR_FORMAT	文字列形式が無効です
0x0000138f	eERR_HMPL_ARG_OUT_OF_RANGE	引数が範囲外です
0x00001392	eERR_HMPL_ASCII_AGENT_RUNNING	ASCII エージェントはすでに実行中です
0x0000139c	eERR_HMPL_CANNOT_RUN_IN_DEBUG	関数はデバッグモードでは実行できません
0x000013a6	eERR_HMPL_TOO_MANY_BRK_POINT	タスクにブレークポイントが多すぎます
0x000013ec	eERR_HMPL_MUTEX_LOCK_TWICE	同じタスクで同じミューテックスを 2 回ロックすることはできません
0x00001450	eERR_HMPL_INVALID_SYS_TIME_MEMORY	バッファが小さすぎます。最小サイズは 30 バイトである必要があります
0x00001451	eERR_HMPL_NOT_SUPPORTED	この HMPL 関数は、このプラットフォームではサポートされていません
0x00001452	eERR_HMPL_CLIENT_NOT_CONNECTED	クライアントが切断されているため送信できません
0x0000176f	eERR_HMPL_INTERNAL_ERROR	HMPL 内部エラー
0x00001770	eERR_HMPL_EXEC_FAILED	HMPL 関数の実行に失敗しました
0x00001771	eERR_HMPL_ASM_LOAD_FAILED	HMPL のコンパイルに失敗しました。アセンブリファイルが空であるか、生成されていません
0x00001772	eERR_HMPL_STARTTASK_TIMEOUT	HMPL StartTask 関数のタイムアウト
0x00001773	eERR_HMPL_STOPTASK_TIMEOUT	HMPL StopTask 関数のタイムアウト
0x000017d4	eERR_ASCII_CONNECT_TIMEOUT	ASCII クライアント接続タイムアウト
0x000017d5	eERR_ASCII_CONNECT_FAILED	ASCII クライアント接続に失敗しました。IP とポートを確認してください
0x000017d6	eERR_ASCII_MULTI_CONNECTING	同時に接続する複数の ASCII クライアント
0x000017d7	eERR_ASCII_MULTI_DISCONNECTING	複数の ASCII クライアントが同時に切断します
0x000017d8	eERR_ASCII_DISCONNECT_TIMEOUT	ASCII クライアント切断タイムアウト
0x000017de	eERR_ASCII_RECV_TIMEOUT	ASCII クライアント受信タイムアウト。後でもう一度やり直してください
0x000017df	eERR_ASCII_RECV_FAIL	ASCII クライアントの受信に失敗しました。接続がまだ生きているかどうかを確認してください
0x000017e0	eERR_ASCII_MULTI_RECVING	複数の ASCII クライアントが同時に受信しています
0x000017e8	eERR_ASCII_SEND_TIMEOUT	ASCII クライアント送信タイムアウト。後でもう一度やり直してください
0x000017e9	eERR_ASCII_SEND_FAIL	ASCII クライアントの送信に失敗しました。接続がまだ生きているかどうかを確認してください
0x000017ea	eERR_ASCII_MULTI_SENDING	同時に送信する複数の ASCII クライアント
0x00001838	eERR_MODBUS_CONNECT_TIMEOUT	Modbus クライアント接続タイムアウト

システムエラーコード		
エラーコード	エラー名	説明
0x00001839	eERR_MODBUS_CONNECT_FAILED	Modbus クライアント接続に失敗しました。ip を確認してください
0x0000183a	eERR_MODBUS_MULTI_CONNECTING	同時に接続する複数の Modbus クライアント
0x0000183b	eERR_MODBUS_MULTI_DISCONNECTING	複数の Modbus クライアントが同時に切断されます
0x0000183c	eERR_MODBUS_DISCONNECT_TIMEOUT	Modbus クライアントの切断タイムアウト
0x0000183d	eERR_MODBUS_DATALENGTH_ERR	Modbus クライアントの読み書きデータ数が制限を超えています
0x0000183e	eERR_MODBUS_SOCKET_BUSY	Modbus クライアントは、同時に 2 つ以上のコマンドを処理します
0x0000183f	eERR_MODBUS_JOB_TIMEOUT	Modbus クライアントジョブの実行タイムアウト。後でもう一度やり直してください
0x00001840	eERR_MODBUS_JOB_FAIL	Modbus クライアントジョブの実行に失敗しました。接続がまだ生きているかどうかを確認してください

## 17.1.2 軸エラーメッセージ

次のエラーコードは、軸のエラーまたは無効な操作が原因で表示されます。記号□□は 16 進形式の軸 ID になります。たとえば、01 は Axis index.01 を表します。0f は Axis index.15 を表します。

表 17.1.2.1

軸エラーコード		
エラーコード	エラー名	説明
0x83□□000a	eERR_AXIS_CMD_UNKOWN	コマンド名は不明です
0x83□□001e	eERR_AXIS_CMD_QUEUE_FULL	軸コマンドキューがいっぱいです
0x83□□0064	eERR_AXIS_CMD_INVALID_STATE	軸は、現在のモーション状態ではコマンドを実行できません。
0x83□□006e	eERR_AXIS_CMD_INVALID_ENABLED	このコマンドは、有効になっている間は使用できません
0x83□□0078	eERR_AXIS_CMD_INVALID_DISABLED	このコマンドは、無効になっている間は使用できません
0x83□□0082	eERR_AXIS_CMD_INVALID_MOVING	軸は移動中にコマンドを実行できません
0x83□□008c	eERR_AXIS_CMD_INVALID_STOPPING	軸が停止している場合、コマンドは無効です
0x83□□0096	eERR_AXIS_CMD_INVALID_ERROR_STATE	軸が ErrorStop 状態の場合、コマンドは無効です
0x83□□00a0	eERR_AXIS_CMD_INVALID_IN_SYNC	軸が同期動作状態の場合、コマンドは無効です
0x83□□00aa	eERR_AXIS_CMD_INVALID_GEAR_MASTER	軸がギアマスタ軸の場合、コマンドは無効です
0x83□□00b4	eERR_AXIS_CMD_INVALID_PP_MODE	軸が PP モードの場合、コマンドは無効です
0x83□□00c8	eERR_AXIS_CMD_INVALID_INPUTSHAPING_ENABLED	位置指令整形機能が有効な場合、軸は指令を実行できません
0x83□□00d2	eERR_AXIS_CMD_INVALID_COMP_ENABLE D	動的補正が有効になっている場合、軸はコマンドを実行できません
0x83□□00dc	eERR_AXIS_CMD_INVALID_GANTRY_MODE	軸は、ガントリーモードでコマンドを実行できません
0x83□□00e6	eERR_AXIS_CMD_INVALID_GROUPED	軸が軸グループにある場合、このコマンドは使用できません
0x83□□012c	eERR_AXIS_CMD_INVALID_PARAMETER	軸コマンドのパラメーターが不正です
0x83□□0136	eERR_AXIS_CMD_INVALID_POS	軸目標位置が許容範囲外です
0x83□□0140	eERR_AXIS_CMD_INVALID_VEL	軸速度設定が許容範囲外です
0x83□□014a	eERR_AXIS_CMD_INVALID_ACC	軸加速度の設定が許容範囲外です
0x83□□0154	eERR_AXIS_CMD_INVALID_DEC	軸減速度の設定が許容範囲外です
0x83□□015e	eERR_AXIS_CMD_INVALID_JERK	軸ジャークの設定が許容範囲外です
0x83□□0168	eERR_AXIS_CMD_INVALID_SM_TIME	軸スムーズ時間設定が許容範囲外です
0x83□□0172	eERR_AXIS_CMD_INVALID_KILL_DEC	軸キル減速度の設定が許容範囲外です
0x83□□017c	eERR_AXIS_CMD_INVALID_VEL_SCALE	軸速度スケールの設定が許容範囲外です
0x83□□0190	eERR_AXIS_COMP_NOT_CNFG	軸の動的補正設定が正しく構成されていません
0x83□□01c2	eERR_AXIS_CMD_INVALID_MASTER_SLAVE_CONNECTION	マスター、スレーブ関係の設定が無効です
0x83□□01cc	eERR_AXIS_CMD_INVALID_SLAVE_ID	スレーブ ID の設定が無効です
0x83□□01d6	eERR_AXIS_CMD_INVALID_GEAR_RATIO	スレーブ軸のギア比設定が許容範囲外です

軸エラーコード		
エラーコード	エラー名	説明
0x83□□01f4	eERR_AXIS_CMD_INVALID_ROLLOVER_POS	軸のロールオーバー位置が無効です。正の値にする必要があります
0x83□□03f2	eERR_AXIS_DRIVE_FAULT	ドライブが障害を報告しました。ドライバー内の対応するエラーメッセージを確認してください
0x83□□03fc	eERR_AXIS_DRIVE_ABNORMAL_DISABLE	ドライバーが異常に無効になっています
0x83□□0406	eERR_AXIS_DRIVE_ENABLE_TOUT	ドライバーを有効にするのに時間がかかりすぎました
0x83□□0410	eERR_AXIS_DRIVE_CLEAR_ERROR_TOUT	ドライバーエラーのクリアに時間がかかりすぎました
0x83□□041a	eERR_AXIS_DRIVE_DISABLE_TOUT	ドライバーを無効にするのに時間がかかりすぎました
0x83□□0456	eERR_AXIS_VEL_LIMIT	基準速度が速度制限を超えました
0x83□□0460	eERR_AXIS_ACC_LIMIT	基準加速度が加速度制限を超えています
0x83□□046a	eERR_AXIS_CURR_LIMIT	現在のコマンドが現在の制限を超えました
0x83□□0474	eERR_AXIS_DAMPINGRATIO_LIMIT	軸の減衰比設定が許容範囲外です
0x83□□047e	eERR_AXIS_FREQUENCY_LIMIT	軸の周波数設定が許容範囲外です
0x83□□07da	eERR_AXIS_SWRL	軸基準位置が右ソフトリミットに達しました
0x83□□07e4	eERR_AXIS_SWLL	軸基準位置が左ソフトリミットに達しました
0x83□□07ee	eERR_AXIS_HWRL	軸右ハードウェアリミット信号がトリガーされました
0x83□□07f8	eERR_AXIS_HWLL	軸左ハードウェアリミット信号がトリガーされました
0x83□□0802	eERR_AXIS_COMP_LIMIT	軸補正位置が最大補正限界を超えました
0x83□□083e	eERR_AXIS_PERR	軸位置偏差が保護限界を超えました。モーターの動きに機械的な干渉がないかどうかを最初に確認してください
0x83□□0848	eERR_AXIS_VERR	軸速度エラーが保護限界を超えました。モーターの動きに機械的な干渉がないかどうかを最初に確認してください。
0x83□□08a2	eERR_AXIS_PVT_MOTION_VEL_LIMIT	軸 PVT モーションの速度が保護限界を超えました。指定されたパラメーターが有効かどうかを最初に確認してください。
0x83□□08ac	eERR_AXIS_PVT_MOTION_ACC_LIMIT	軸 PVT モーションの加速度が保護限界を超えました。指定されたパラメーターが有効かどうかを最初に確認してください
0x83□□08b6	eERR_AXIS_PVT_MOTION_INVALID_TIME	軸 PVT モーションのタイムシーケンスが無効です。指定されたパラメーターが有効かどうかを最初に確認してください。
0x83□□0bb8	eERR_AXIS_CTRL_ERR	軸内部制御エラー
0x83□□0fa0	eERR_AXIS_CMD_GEAR_DISABLED	ギアが無効になっている間は、ギアコマンドを使用できません



## 17.1.3 グループのエラーメッセージ

以下のエラーコードは、軸グループでのエラーまたは無効な操作により表示されます。記号□□は 16 進形式の軸グループ ID になります。たとえば、01 は Axis group index.01 を表します。0f は、Axis group index.15 を表します。

表 17.1.3.1

軸グループのエラーコード		
エラーコード	エラー名	説明
0x82□□000a	eERR_CRD_CMD_UNKNOWN	軸グループコマンドが不明です
0x82□□0028	eERR_CRD_CMD_AXIS_DUPLICATED	軸は既にグループにあるため、追加できませんでした。
0x82□□0032	eERR_CRD_CMD_GRP_SIZE_EMPTY	軸グループが空です
0x82□□003c	eERR_CRD_CMD_GRP_SIZE_FULL	軸グループがいっぱいで、これ以上軸を保持できません
0x82□□0046	eERR_CRD_CMD_INVALID_MOVING	軸群移動中はコマンド無効です
0x82□□0050	eERR_CRD_CMD_INVALID_DISABLED	軸グループが無効の場合、コマンドは無効です
0x82□□005a	eERR_CRD_CMD_INVALID_INPUTSHAPING_PARAMETER_INCOMPLETE	軸グループインシェイプ機能のパラメーターが不完全です。
0x82□□001e	eERR_CRD_CMD_INVALID_KIN_SETTING	キネマティクスタイプの設定が無効です
0x82□□001f	eERR_CRD_CMD_INVALID_SPECIFIC_KIN	軸グループが特定のキネマティクスタイプにある場合、このコマンドは無効です
0x82□□006e	eERR_CRD_CMD_INVALID_STATE	軸グループは、現在のモーション状態ではコマンドを実行できません
0x82□□0078	eERR_CRD_CMD_QUEUE_FULL	最後のコマンドが完了するまでお待ちください
0x82□□00d2	eERR_CRD_CMD_INVALID_POS	軸グループの目標位置または姿勢が許容範囲外です
0x82□□00dc	eERR_CRD_CMD_INVALID_LIN_VEL	軸群の線速度設定が許容範囲外です
0x82□□00e6	eERR_CRD_CMD_INVALID_LIN_ACC	軸グループの直線加速度設定が許容範囲外です
0x82□□00f0	eERR_CRD_CMD_INVALID_LIN_DEC	軸グループの直線減速度の設定が許容範囲外です
0x82□□00fa	eERR_CRD_CMD_INVALID_LIN_JERK	軸グループの直線加減速設定が許容範囲外です
0x82□□0104	eERR_CRD_CMD_INVALID_LIN_SM_TIME	軸グループの直線平滑時間設定が許容範囲外です
0x82□□010e	eERR_CRD_CMD_INVALID_DAMPINGRATIO	軸群の減衰比設定が許容範囲外です
0x82□□0118	eERR_CRD_CMD_INVALID_FREQUENCY	軸グループの周波数設定が許容範囲外です
0x82□□0140	eERR_CRD_CMD_INVALID_ANG_VEL	軸グループの角速度設定が許容範囲外です
0x82□□014a	eERR_CRD_CMD_INVALID_ANG_ACC	軸グループの角加速度設定が許容範囲外です
0x82□□0154	eERR_CRD_CMD_INVALID_ANG_DEC	軸群の減速度設定が許容範囲外です
0x82□□015e	eERR_CRD_CMD_INVALID_ANG_JERK	軸群のジャークの設定が許容範囲外です
0x82□□0168	eERR_CRD_CMD_INVALID_ANG_SM_TIME	軸グループの角度平滑時間設定が許容範囲外です
0x82□□0190	eERR_CRD_CMD_INVALID_VEL_SCALE	軸グループの速度スケールが許容範囲外です
0x82□□019a	eERR_CRD_CMD_INVALID_TRANS_VEL	軸グループの移行速度が無効です
0x82□□01a4	eERR_CRD_CMD_INVALID_TRANS_DIS	軸グループの transition 距離が無効です
0x82□□01b8	eERR_CRD_CMD_TRANS_MODE_UNKNOWN	パス transition モード名が不明です
0x82□□01c2	eERR_CRD_CMD_COORD_SYS_UNKNOWN	座標系が不明です

軸グループのエラーコード		
エラーコード	エラー名	説明
0x82□□01cc	eERR_CRD_CMD_BLEND_MODE_UNKNOWN	パスブレンドモード名が不明です
0x82□□01fe	eERR_CRD_CMD_LIN_INVALID_PARAM	パラメーターは、直線経路計画では無効です
0x82□□0262	eERR_CRD_CMD_CIRC_INVALID_PARAM	パラメーターは円形経路計画では無効です
0x82□□026c	eERR_CRD_CMD_CIRC_INVALID_CENTER	円形パスの中心位置が始点/終点に近すぎます
0x82□□0276	eERR_CRD_CMD_CIRC_ANGLE_SMALL	円形パスの中心角が小さすぎます
0x82□□0280	eERR_CRD_CMD_CIRC_INVALID_RADIUS	円形パスの半径が無効です
0x82□□028a	eERR_CRD_CMD_CIRC_INVALID_COORD	円形パスの座標系が無効です
0x82□□02c6	eERR_CRD_CMD_BEZIER_INVALID_PARAM	パラメーターは、ベジエ曲線パス計画では無効です
0x82□□02d0	eERR_CRD_CMD_BSPLINE_INVALID_PARAM	パラメーターは BSpline 曲線パス計画では無効です
0x82□□02da	eERR_CRD_CMD_CURVE_INVALID_START_POS	開始位置は曲線パス計画では無効です
0x82□□03f2	eERR_CRD_AXIS_ABNORMALLY_DISABLED	軸グループ内の 1 つまたは複数の軸が異常に無効になっています
0x82□□03fc	eERR_CRD_AXIS_SWL	軸グループの軸の 1 つがソフトウェアリミットに達しています

## 17.2 HIMC\_GetLastError

### 目的

コントローラーの最新のエラーコードを取得します

### 構文

```
int HIMC_GetLastError(
    int ctrl_id,
    int *p_error_code
);
```

### パラメーター

**ctrl\_id [in]**                      HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**p\_error\_code [out]**              コントローラーの最新のエラーコードを受け取るバッファへのポインター  
定義については、セクション 17.1.1 を参照してください

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetLastError
LabVIEW	HIMC Get Last Error.vi
Python	GetLastError



## 17.3 HIMC\_GetAxisLastErr

### 目的

軸の最新のエラーコードを取得します

### 構文

```
int HIMC_GetAxisLastErr(
    int ctrl_id,
    int axis_id,
    int *err_code
);
```

### パラメーター

ctrl_id [in]	HIWIN モーションコントローラーのコントローラーID HIMC_ConnectCtrl を呼び出して取得する必要があります。
axis_id [in]	Axis index
err_code [out]	軸の最新のエラーコードを受け取るバッファーへのポインター 定義については、セクション 17.1.2 を参照してください

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_GetAxisLastErr
LabVIEW	HIMC Get Axis Last Err.vi
Python	GetAxisLastErr

## 17.4 HIMC\_ClearAxisLastErr

### 目的

軸の最新のエラーコードをクリアします

### 構文

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

### パラメーター

- ctrl\_id [in]           HIWIN モーションコントローラーのコントローラーID  
                      HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- axis\_id [in]           Axis index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_ClearAxisLastErr
LabVIEW	HIMC Clear Axis Last Err.vi
Python	ClearAxisLastErr

## 17.5 HIMC\_GetGrpLastErr

### 目的

軸グループの最新のエラーコードを取得します

### 構文

```
int HIMC_GetGrpLastErr(
    int ctrl_id,
    int group_id,
    int *err_code
);
```

### パラメーター

- ctrl\_id [in]** HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。
- group\_id [in]** Axis group index
- err\_code [out]** 軸グループの最新のエラーコードを受け取るバッファへのポインター  
定義については、セクション 17.1.3 を参照してください

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_ClearAxisLastErr
LabVIEW	HIMC Clear Axis Last Err.vi
Python	ClearAxisLastErr

## 17.6 HIMC\_ClearGrpLastErr

### 目的

軸グループの最新のエラーコードをクリアします

### 構文

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

### パラメーター

**ctrl\_id [in]**           HIWIN モーションコントローラーのコントローラーID  
HIMC\_ConnectCtrl を呼び出して取得する必要があります。

**group\_id [in]**       Axis group index

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

### 要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	HIMC_ClearGrpLastErr
LabVIEW	HIMC Clear Grp Last Err.vi
Python	ClearGrpLastErr

## 17.7 HIMC\_GetErrorInformation

### 目的

指定したエラーID の情報を取得します

### 構文

```
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  
);
```

### パラメーター

error_id [in]	エラーID を指定します
p_name [out]	指定されたエラーID のエラー名を受け取るバッファへのポインター
name_buff_len [in]	エラー名を受け取るバッファの最大文字数を指定します
p_name_actual_len [out]	エラー名の実際の文字数を受け取るバッファへのポインター。(終端のヌル文字を除外します。)
p_description [out]	指定されたエラーID のエラーの説明を受け取るバッファへのポインター
description_buff_len [in]	エラーの説明を受け取るバッファの最大文字数を指定します
p_description_actual_len [out]	エラーの説明の実際の文字数を受け取るバッファへのポインター。(終端のヌル文字を除外します。)

### 戻りの値

関数が成功した場合は int 値 0 を返し、関数が失敗した場合はゼロ以外の値を返します。

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	HIMC_GetErrorInformation
LabVIEW	HIMC Get Error Information.vi
Python	GetErrorInformation

# 18. データ構造

18.1	ComInfo .....	18-2
18.2	CoordPosition .....	18-3
18.3	MotionProfile.....	18-4
18.4	CenterPosition .....	18-5
18.5	NormalVector.....	18-6
18.6	PosTriggerPar.....	18-7

18.1 ComInfo

目的

接続タイプとその情報を定義します

構文

```
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;
```

パラメーター

ComType	接続タイプ
TCP_IP	ネットワーク接続パラメーター、ip および port を含む構造体
RS232	RS232 接続パラメーター、com ポート名、およびボーレートを含む構造体
Simulator	Simulator 接続パラメーターと autoExecExe を含む構造体

要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	ComInfo
LabVIEW	--
Python	ComInfo



## 18.2 CoordPosition

### 目的

座標モーションの位置または距離を定義します

### 構文

```
typedef struct {
    double x_pos;
    double y_pos;
    double z_pos;
    double a_pos;
    double b_pos;
    double c_pos;
} CoordPosition, *PCoordPosition;
```

### パラメーター

x_pos	終点の直線位置 X。単位：mm
y_pos	終点の直線位置 Y。単位：mm
z_pos	終点の直線位置 Z。単位：mm
a_pos	終点の方向角 A。単位：deg
b_pos	終点の方向角度 B。単位：deg
c_pos	終点の方向角度 C。単位：deg

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	CoordPosition
LabVIEW	--
Python	CoordPosition

## 18.3 MotionProfile

### 目的

モーションプロファイル設定を定義します

### 構文

```
typedef struct {
    double max_vel;
    double max_acc;
    double max_dec;
    double smooth_time;
} MotionProfile, *PMotionProfile;
```

### パラメーター

max_vel	直線運動の最大接線速度 単位: mm/s または deg/s 範囲: 0 ~ 5000
max_acc	直線運動の接線方向の最大加速度 単位: mm/s <sup>2</sup> または deg/s <sup>2</sup> 範囲: >0 ~ 50000 (加速度を 0 にすることはできません)
max_dec	直線運動の接線方向の最大減速度 単位: mm/s <sup>2</sup> または deg/s <sup>2</sup> 範囲: >0 ~ 50000 (減速度を 0 にすることはできません)
smooth_time	直線運動の滑らかな時間 単位: ms 範囲: 0 ~ 500

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	MotionProfile
LabVIEW	--
Python	MotionProfile

## 18.4 CenterPosition

### 目的

中心位置の設定を定義します

### 構文

```
typedef struct {  
    double x_pos;  
    double y_pos;  
    double z_pos;  
} CenterPosition, *PCenterPosition;
```

### パラメーター

x\_pos      X 軸の位置。単位：mm  
y\_pos      Y 軸の位置。単位：mm  
z\_pos      Z 軸の位置。単位：mm

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	CenterPosition
LabVIEW	--
Python	CenterPosition

## 18.5 NormalVector

### 目的

法線ベクトルの設定を定義します

### 構文

```
typedef struct {
    double x_vector;
    double y_vector;
    double z_vector;
} NormalVector, *PNormalVector;
```

### パラメーター

x\_vector      X 方向ベクトル  
y\_vector      Y 方向ベクトル  
z\_vector      Z 方向ベクトル

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	NormalVector
LabVIEW	--
Python	NormalVector

## 18.6 PosTriggerPar

### 目的

位置トリガー設定を定義します

### 構文

```
typedef struct {  
    double start_pos;  
    double end_pos;  
    double interval;  
    int pulse_width;  
} PosTriggerPar, *PPosTriggerPar;
```

### パラメーター

start_pos	PT 機能の開始位置 単位：mm または deg
end_pos	PT 機能の終了位置 単位：mm または deg
interval	連続する PT 出力間の位置間隔 単位：mm または deg
pulse_width	各 PT 出力信号の幅 単位：ns

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	PosTriggerPar
LabVIEW	--
Python	PosTriggerPar

(このページはブランクになっています)

# 19. 列拳

19.1	ComType .....	19-2
19.2	CoordSystem.....	19-3
19.3	MotionBufferMode .....	19-4
19.4	MotionTransitionMode .....	19-5
19.5	ShaperMode .....	19-6

## 19.1 ComType

### 目的

接続タイプの列挙

### 構文

```
typedef enum {
    COM_TYPE_TCPIP,
    COM_TYPE_RS232,
    COM_TYPE_SIMULATOR
} ComType;
```

### パラメーター

COM\_TYPE\_TCPIP                      接続タイプは TCPIP です

COM\_TYPE\_RS232                      接続タイプは RS232 です

COM\_TYPE\_SIMULATOR              接続タイプはシミュレーターです

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	ComType
LabVIEW	--
Python	ComType



## 19.2 CoordSystem

### 定義

### 座標系

### 構文

```
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;
```

### 説明

セクション 6.1.2 を参照してください

### 要件

サポートされる最小バージョン	iA Studio 2.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	CoordSystem
LabVIEW	--
Python	CoordSystem

## 19.3 MotionBufferMode

### 定義

隣接する座標モーションセグメント間のバッファモード

### 構文

```
typedef enum {
    kBM_Buffered = 0,
    kBM_Aborting = 1,
    kBM_BlendingLow = 2,
    kBM_BlendingPrevious = 3,
    kBM_BlendingNext = 4,
    kBM_BlendingHigh = 5
} MotionBufferMode;
```

### 説明

セクション 6.1.4 を参照してください

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	MotionBufferMode
LabVIEW	--
Python	MotionBufferMode

## 19.4 MotionTransitionMode

### 定義

隣接する座標モーションセグメント間の Transition モード

### 構文

```
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  
} MotionTransitionMode;
```

### 説明

セクション 6.1.5 を参照してください

### 要件

サポートされる最小バージョン	iA Studio 0.23
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

### 他の API 環境の対応名称

C#	MotionTransitionMode
LabVIEW	--
Python	MotionTransitionMode

## 19.5 ShaperMode

定義

入力整形フィルター (InShape) のフィルターモード

構文

```
typedef enum {
    Shaper_Normal = 0,
    Shaper_Robust
} ShaperMode;
```

パラメーター

Shaper\_Normal            通常の入力 shaping フィルター  
Shaper\_Robust           堅牢な入力 shaping フィルター

要件

サポートされる最小バージョン	iA Studio 1.1
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

他の API 環境の対応名称

C#	ShaperMode
LabVIEW	--
Python	ShaperMode

# 20. 付録



20.1    API エラーコード .....20-8

## 20.1 API エラーコード

API でコントローラーにアクセスすると、次のエラーコードが表示されます。

表 20.1.1

API エラーコード		
エラーコード	エラー名	説明
0x01000000	eERR_API_COMM_ERR	コントローラーとの通信中にエラーが発生しました
0x0100000a	eERR_API_CONNECT_FAIL	コントローラーに接続できません
0x01000014	eERR_API_TOUT	タイムアウト期間が経過したため、この操作は返されました
0x0100001e	eERR_API_ACCESS_REJECT	リクエストは拒否されました
0x01000028	eERR_API_FIFO_MISMATCH	致命的な API エラーです
0x01000032	eERR_API_FIFO_FULL	ネットワークがビジーです
0x0100003c	eERR_API_HIMC_NOT_READY	HIMC の準備ができていません
0x01000046	eERR_API_PROTOCOL_MISMATCH	致命的な API エラーです
0x01000050	eERR_API_INPUT_ARG_ERR	引数が無効です
0x0100005a	eERR_API_NOT_SUPPORT	API は、このバージョンではサポートされていません
0x01000064	eERR_API_BUSY	API がビジーです
0x0100006e	eERR_API_FILE_TRANS_FAIL	ファイル送信に失敗しました
0x01000078	eERR_API_ID_NOT_FOUND	接続 ID が見つかりませんでした。まだ接続されていない可能性があります
0x01000082	eERR_API_SLV_DB_NOT_READY	スレーブは準備ができていません
0x0100008c	eERR_API_SLV_ID_INVALID	スレーブ ID が無効です
0x01000096	eERR_API_INVALID_VAR_ID	変数 ID が無効です
0x010000a0	eERR_API_VAR_VAL_OUT_OF_RANGE	値が範囲外です
0x010000a5	eERR_API_VAR_IS_READ_ONLY	変数は読み取り専用です
0x010000aa	eERR_API_FS_ACCESS_DENIED	ファイルシステムにアクセスできません。権限を確認してください
0x010000b4	eERR_API_TASK_ID_INVALID	タスク ID が無効です
0x010000be	eERR_API_TASK_EMPTY	タスクは空です
0x010000c3	eERR_API_TASK_FUNC_NOT_FOUND	関数が見つかりません
0x010000c8	eERR_API_TASK_NOT_RUNNING	タスクは実行されていません
0x010000d2	eERR_API_TASK_IS_RUNNING	タスクはすでに実行中です
0x010000d7	eERR_API_TOO_MANY_BRK_POINT	タスクにブレークポイントが多すぎます
0x010000dc	eERR_API_INVALID_ERROR_ID	エラー ID が無効です
0x010000e6	eERR_API_INSUFFICIENT_BUFFER	バッファが不足しています
0x010000f0	eERR_API_STR_TOO_LONG	文字列の長さが範囲外です
0x000000fa	eERR_API_HIMC_VERSION_MISMATCH	API は、このコントローラーのバージョンと互換性がありません
0x010003e8	eERR_API_MOTION_ERROR	モーション制御エラー。エラーログを確認してください
0x0100270f	eERR_API_FATAL	致命的な API エラーです

HIMC API リファレンスガイド  
バージョン：V0.5      2023 年 3 月改定

- 
1. HIWIN は HIWIN Mikrosystem Corp., HIWIN Technologies Corp., ハイウィン株式会社の登録商標です。ご自身の権利を保護するため、模倣品を購入することは避けてください。
  2. 実際の製品は、製品改良等に対応するため、このカタログの仕様や写真と異なる場合があります。
  3. HIWIN は「貿易法」および関連規制の下で制限された技術や製品を販売・輸出しません。制限された HIWIN 製品を輸出する際には、関連する法律に従って、所管当局によって承認を受けます。また、核・生物・化学兵器やミサイルの製造または開発に使用することは禁じます。
-