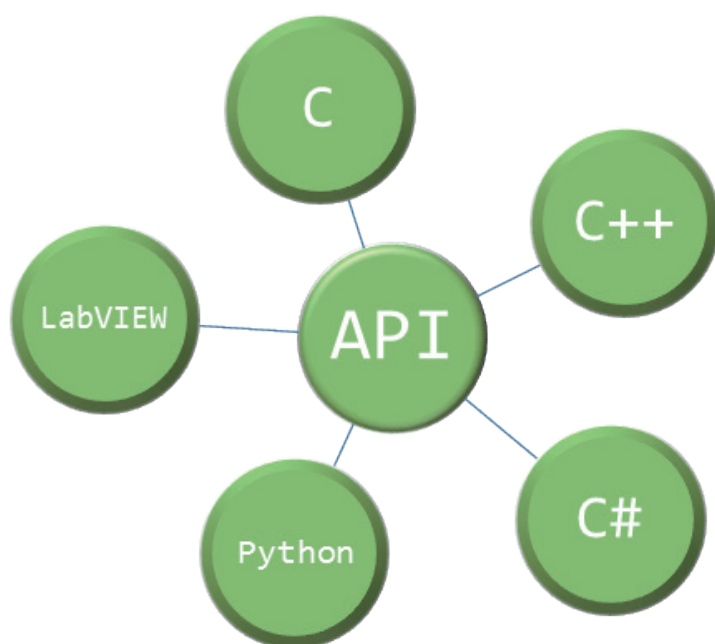


HIWIN®



HIMC

API 參考指南

修訂紀錄

發行日期	版次	適用軟體	更新內容
2019/04/02	0.1	iA Studio 1.1.3772.0	第一版發行。

目錄

1.	序言.....	1-1
1.1	HIMC API 設定.....	1-2
1.1.1	C / C++	1-2
1.1.2	C#	1-2
1.2	法律免責聲明	1-2
2.	HIMC 系統函式.....	2-1
2.1	HIMC_ConnectCtrl	2-2
2.2	HIMC_ConnectToSimulator	2-3
2.3	HIMC_ConnectToEthernet.....	2-4
2.4	HIMC_DisconnectCtrl	2-5
2.5	HIMC_RebootController.....	2-6
2.6	HIMC_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_RescanMoE.....	2-14
3.	軸函式	3-1
3.1	軸運動控制	3-3
3.1.1	HIMC_Enable	3-3
3.1.2	HIMC_Disable.....	3-4
3.1.3	HIMC_Reset.....	3-5
3.1.4	HIMC_MoveAbs	3-6
3.1.5	HIMC_MoveRel.....	3-7
3.1.6	HIMC_MoveVel.....	3-8
3.1.7	HIMC_Stop	3-9
3.2	軸設定	3-10
3.2.1	HIMC_GetMaxVel	3-10
3.2.2	HIMC_SetVel.....	3-11

目錄

3.2.3	HIMC_GetMaxAcc.....	3-12
3.2.4	HIMC_SetAcc.....	3-13
3.2.5	HIMC_GetMaxDec.....	3-14
3.2.6	HIMC_SetDec.....	3-15
3.2.7	HIMC_GetSWRL.....	3-16
3.2.8	HIMC_SetSWRL.....	3-17
3.2.9	HIMC_GetSWLL.....	3-18
3.2.10	HIMC_SetSWLL.....	3-19
3.2.11	HIMC_GetSMTime.....	3-20
3.2.12	HIMC_SetSMTime.....	3-21
3.2.13	HIMC_GetMoveTime.....	3-22
3.2.14	HIMC_GetSettlingTime.....	3-23
3.2.15	HIMC_SetPos.....	3-24
3.2.16	HIMC_GetPosFb.....	3-25
3.2.17	HIMC_GetPosOffset.....	3-26
3.2.18	HIMC_GetPosErr.....	3-27
3.2.19	HIMC_GetRefPos.....	3-28
3.2.20	HIMC_GetRefVel.....	3-29
3.2.21	HIMC_GetRefAcc.....	3-30
3.2.22	HIMC_GetPosOut.....	3-31
3.2.23	HIMC_GetVelOut.....	3-32
3.2.24	HIMC_GetAccOut.....	3-33
3.2.25	HIMC_IgnoreHWL.....	3-34
3.2.26	HIMC_IgnoreSWL.....	3-35
3.2.27	HIMC_GetAxisNum.....	3-36
3.2.28	HIMC_GetMultiAxesFeedbackPos.....	3-37
3.3	軸狀態.....	3-38
3.3.1	HIMC_IsEnabled.....	3-38
3.3.2	HIMC_IsMoving.....	3-39
3.3.3	HIMC_IsInPos.....	3-40
3.3.4	HIMC_IsErrorStop.....	3-41
3.3.5	HIMC_IsGantry.....	3-42
3.3.6	HIMC_IsGrouped.....	3-43

目錄

3.3.7	HIMC_IsSync	3-44
3.3.8	HIMC_IsHWLL.....	3-45
3.3.9	HIMC_IsHWRL	3-46
3.3.10	HIMC_IsSWLL	3-47
3.3.11	HIMC_IsSWRL.....	3-48
3.3.12	HIMC_IsCompActive.....	3-49
4.	同步運動函式	4-1
4.1	HIMC_EnableGear	4-2
4.2	HIMC_DisableGear.....	4-3
4.3	HIMC_GearIn.....	4-4
4.4	HIMC_GearOut.....	4-5
5.	龍門函式	5-1
5.1	HIMC_EnableGantryPair	5-2
5.2	HIMC_DisableGantryPair.....	5-3
6.	軸群組函式	6-1
6.1	軸群組運動控制.....	6-2
6.1.1	HIMC_EnableGroup	6-2
6.1.2	HIMC_DisableGroup.....	6-3
6.1.3	基本運動命令	6-4
6.1.3.1	HIMC_LineAbs2D	6-4
6.1.3.2	HIMC_LineAbs3D	6-5
6.1.3.3	HIMC_LineRel2D	6-6
6.1.3.4	HIMC_LineRel3D	6-7
6.1.3.5	HIMC_Arc2D	6-8
6.1.3.6	HIMC_Circle2D	6-10
6.1.4	進階運動命令	6-12
6.1.4.1	HIMC_LinAbs	6-12
6.1.4.2	HIMC_LinRel.....	6-14
6.1.4.3	HIMC_CircAbs	6-16
6.1.5	HIMC_StopGroup	6-18

目錄

6.2	軸群組設定	6-19
6.2.1	HIMC_AddAxesToGrp	6-19
6.2.2	HIMC_RemoveAxisFromGrp	6-20
6.2.3	HIMC_SetupGroup	6-21
6.2.4	HIMC_UngrpAllAxes	6-22
6.2.5	HIMC_GetGroupID	6-23
6.2.6	HIMC_SetGrpMotionProfile	6-24
6.2.7	HIMC_SetGrpKin	6-26
6.2.8	HIMC_GroupReset	6-27
6.3	軸群組狀態	6-28
6.3.1	HIMC_IsGrpEnabled	6-28
6.3.2	HIMC_IsGrpMoving	6-29
6.3.3	HIMC_IsGrpInPos	6-30
7.	GPIO 函式	7-1
7.1	HIMC_GetHimcGpi	7-2
7.2	HIMC_GetHimcGpo	7-3
7.3	HIMC_SetHimcGpo	7-4
7.4	HIMC_GetHimcMultiGpi	7-5
7.5	HIMC_GetHimcMultiGpo	7-6
7.6	HIMC_SetHimcMultiGpo	7-7
7.7	HIMC_GetHimcGpiNum	7-8
7.8	HIMC_GetHimcGpoNum	7-9
7.9	HIMC_ToggleHimcGpo	7-10
7.10	HIMC_GetSlaveGpi	7-11
7.11	HIMC_GetSlaveGpo	7-12
7.12	HIMC_SetSlaveGpo	7-13
7.13	HIMC_GetSlaveMultiGpi	7-14
7.14	HIMC_GetSlaveMultiGpo	7-15
7.15	HIMC_SetSlaveMultiGpo	7-16
7.16	HIMC_GetSlaveGpiNum	7-17
7.17	HIMC_GetSlaveGpoNum	7-18
7.18	HIMC_ToggleSlaveGpo	7-19

目錄

8.	User Table 函式.....	8-1
8.1	HIMC_SetUserTable	8-2
8.2	HIMC_GetUserTable.....	8-3
8.3	HIMC_SetTableValue	8-4
8.4	HIMC_GetTableValue	8-5
8.5	HIMC_SaveUserTable	8-6
8.6	HIMC_LoadUserTable.....	8-7
9.	位置觸發函式.....	9-1
9.1	HIMC_EnablePT.....	9-2
9.2	HIMC_DisablePT	9-3
9.3	HIMC_IsPTEnabled.....	9-4
9.4	HIMC_SetPosTriggerConfig.....	9-5
10.	Touch Probe 函式.....	10-1
10.1	HIMC_EnableTouchProbe.....	10-2
10.2	HIMC_DisableTouchProbe	10-3
10.3	HIMC_IsTouchProbeEnabled.....	10-4
10.4	HIMC_IsTouchProbeTriggered	10-5
10.5	HIMC_GetTouchProbePos.....	10-6
11.	動態誤差補償函式.....	11-1
11.1	HIMC_EnableComp	11-2
11.2	HIMC_DisableComp.....	11-3
11.3	HIMC_SetupComp	11-4
11.4	HIMC_SetupComp2D	11-6
12.	濾波器函式.....	12-1
12.1	HIMC_EnableAxisVsf.....	12-2
12.2	HIMC_DisableAxisVsf.....	12-3
12.3	HIMC_SetAxisVsf	12-4
12.4	HIMC_EnableAxisInShape.....	12-6
12.5	HIMC_DisableAxisInShape	12-7

目錄

12.6	HIMC_SetAxisInShape	12-8
12.7	HIMC_EnableGrpInShape	12-10
12.8	HIMC_DisableGrpInShape	12-11
12.9	HIMC_SetGrpInShape	12-12
13.	HMPL Task 函式	13-1
13.1	HIMC_StartTask	13-2
13.2	HIMC_StartTaskFunc	13-3
13.3	HIMC_StopTask	13-4
13.4	HIMC_StopAllTask	13-5
13.5	HIMC_IsTaskStop	13-6
14.	回調函式	14-1
14.1	HIMC_SetHmplEvtCallback	14-2
14.2	HIMC_SetErrorCallback	14-3
15.	參數操作函式	15-1
15.1	HIMC_GetVariableByID	15-2
15.2	HIMC_SetVariableByID	15-3
15.3	HIMC_GetVariableListByID	15-4
15.4	HIMC_SetVariableListByID	15-5
15.5	HIMC_GetGlobalVariables	15-6
15.6	HIMC_SetGlobalVariables	15-7
15.7	HIMC_GetSlvVar	15-8
15.8	HIMC_SetSlvVar	15-9
15.9	HIMC_GetSlvSt	15-10
15.10	HIMC_SetSlvSt	15-11
16.	HIMC 錯誤函式	16-1
16.1	HIMC_GetLastError	16-2
16.2	HIMC_GetAxisLastErr	16-3
16.3	HIMC_ClearAxisLastErr	16-4
16.4	HIMC_GetGrpLastErr	16-5

目錄

16.5	HIMC_ClearGrpLastErr	16-6
16.6	HIMC_GetErrorInformation.....	16-7
17.	數據結構	17-1
17.1	ComInfo	17-2
17.2	CoordPosition.....	17-3
17.3	MotionProfile.....	17-4
17.4	CenterPosition.....	17-5
17.5	NormalVector	17-6
17.6	PosTriggerPar	17-7
18.	列舉.....	18-1
18.1	ComType.....	18-2
18.2	CoordSystem	18-3
18.3	MotionBufferMode	18-4
18.4	MotionTransitionMode	18-5
18.5	ShaperMode.....	18-6

(此頁有意留白。)

1. 序言



1.	序言.....	1-1
1.1	HIMC API 設定.....	1-2
1.1.1	C / C++	1-2
1.1.2	C#	1-2
1.2	法律免責聲明	1-2

1.1 HIMC API 設定

1.1.1 C / C++

將檔案 HIMC_API.h、HIMC_API.lib 與 HIMC_API.dll 放在執行檔所在的資料夾中。

1.1.2 C#

將檔案 HIMC_API.dll 放在執行檔所在的資料夾中。

1.2 法律免責聲明

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

2. HIMC 系統函式

2.	HIMC 系統函式	2-1
2.1	HIMC_ConnectCtrl	2-2
2.2	HIMC_ConnectToSimulator	2-3
2.3	HIMC_ConnectToEthernet.....	2-4
2.4	HIMC_DisconnectCtrl	2-5
2.5	HIMC_RebootController.....	2-6
2.6	HIMC_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_RescanMoE.....	2-14

2.1 HIMC_ConnectCtrl

用途

建立控制器的連線。

語法

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

參數

com_info [in] 儲存連線資訊的結構。

p_ctrl_id [out] 指標型態的記憶體，用來儲存控制器 ID。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.2 HIMC_ConnectToSimulator

用途

建立模擬器的連線。

語法

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

參數

p_ctrl_id [out] 指標型態的記憶體，用來儲存控制器 ID。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.3 HIMC_ConnectToEthernet

用途

建立 Ethernet 的連線。

語法

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

參數

ip_address [in] IP 位址的字串。

port [in] 埠的字串。

p_ctrl_id [out] 指標型態的記憶體，用來儲存控制器 ID。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.4 HIMC_DisconnectCtrl

用途

切斷控制器的連線。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.5 HIMC_RebootController

用途

重新啟動控制器。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.6 HIMC_IsSystemOper

用途

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

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.7 HIMC_IsSystemPreOp

用途

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

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.8 HIMC_IsSystemError

用途

詢問 HIMC 系統是否處於錯誤 (error) 狀態。若是，HIMC 與從站間的連線發生錯誤。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.9 HIMC_DisableAll

用途

解激磁全部的軸和軸群組。

語法

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

參數

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

回傳值

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

備註

此函式會清除軸和軸群組原有的路徑規畫。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.10 HIMC_StopAll

用途

以急剎加速度停止全部的軸和軸群組。

語法

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

參數

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

回傳值

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

備註

此函式會清除軸和軸群組原有的路徑規畫。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.11 HIMC_EStop

用途

緊急停止控制器。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.12 HIMC_GetSlaveNum

用途

取得連接至控制器的從站數量。

語法

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

參數

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

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

2.13 HIMC_RescanMoE

用途

當 HIMC 與從站失去連線時，重新掃描 MoE。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3. 軸函式

3.	軸函式	3-1
3.1	軸運動控制	3-3
3.1.1	HIMC_Enable	3-3
3.1.2	HIMC_Disable	3-4
3.1.3	HIMC_Reset	3-5
3.1.4	HIMC_MoveAbs	3-6
3.1.5	HIMC_MoveRel	3-7
3.1.6	HIMC_MoveVel	3-8
3.1.7	HIMC_Stop	3-9
3.2	軸設定	3-10
3.2.1	HIMC_GetMaxVel	3-10
3.2.2	HIMC_SetVel	3-11
3.2.3	HIMC_GetMaxAcc	3-12
3.2.4	HIMC_SetAcc	3-13
3.2.5	HIMC_GetMaxDec	3-14
3.2.6	HIMC_SetDec	3-15
3.2.7	HIMC_GetSWRL	3-16
3.2.8	HIMC_SetSWRL	3-17
3.2.9	HIMC_GetSWLL	3-18
3.2.10	HIMC_SetSWLL	3-19
3.2.11	HIMC_GetSMTIME	3-20
3.2.12	HIMC_SetSMTIME	3-21
3.2.13	HIMC_GetMoveTime	3-22
3.2.14	HIMC_GetSettlingTime	3-23
3.2.15	HIMC_SetPos	3-24
3.2.16	HIMC_GetPosFb	3-25
3.2.17	HIMC_GetPosOffset	3-26
3.2.18	HIMC_GetPosErr	3-27
3.2.19	HIMC_GetRefPos	3-28
3.2.20	HIMC_GetRefVel	3-29
3.2.21	HIMC_GetRefAcc	3-30
3.2.22	HIMC_GetPosOut	3-31
3.2.23	HIMC_GetVelOut	3-32
3.2.24	HIMC_GetAccOut	3-33
3.2.25	HIMC_IgnoreHWL	3-34
3.2.26	HIMC_IgnoreSWL	3-35
3.2.27	HIMC_GetAxisNum	3-36
3.2.28	HIMC_GetMultiAxesFeedbackPos	3-37

3.3	軸狀態	3-38
3.3.1	HIMC_IsEnabled	3-38
3.3.2	HIMC_IsMoving	3-39
3.3.3	HIMC_IsInPos	3-40
3.3.4	HIMC_IsErrorStop	3-41
3.3.5	HIMC_IsGantry	3-42
3.3.6	HIMC_IsGrouped	3-43
3.3.7	HIMC_IsSync	3-44
3.3.8	HIMC_IsHWLL	3-45
3.3.9	HIMC_IsHWRL	3-46
3.3.10	HIMC_IsSWLL	3-47
3.3.11	HIMC_IsSWRL	3-48
3.3.12	HIMC_IsCompActive	3-49

3.1 軸運動控制

3.1.1 HIMC_Enable

用途

激磁軸。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.1.2 HIMC_Disable

用途

解激磁軸。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

備註

此函式會清除軸原有的路徑規畫。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.1.3 HIMC_Reset

用途

當軸處於 ErrorStop 狀態時，重新設定軸。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.1.4 HIMC_MoveAbs

用途

將軸移動至絕對目標位置。

語法

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

參數

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

axis_id [in] 軸編號。

pos [in] 絕對目標位置的值。
參數單位：meter (公尺) 或 radian (弧度)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.1.5 HIMC_MoveRel

用途

將軸移動相對距離。

語法

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

參數

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

axis_id [in] 軸編號。

rel_dist [in] 相對距離的值。
參數單位：meter (公尺) 或 radian (弧度)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.1.6 HIMC_MoveVel

用途

以特定速度持續移動。

語法

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

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- vel [in] 移動速度的值。
可為正也可為負，以表示運動的方向。
參數單位：m/s (公尺/秒) 或 rad/s (弧度/秒)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.1.7 HIMC_Stop

用途

停止軸的運動。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

備註

此函式會清除軸原有的路徑規畫。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2 軸設定

3.2.1 HIMC_GetMaxVel

用途

取得軸的最大速度。

語法

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

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_vel [out] 指標型態的記憶體，用來儲存軸的最大速度。
參數單位：m/s (公尺/秒) 或 rad/s (弧度/秒)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.2 HIMC_SetVel

用途

設置軸的最大速度。

語法

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

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- vel [in] 軸的新最大速度。
 參數單位：m/s (公尺/秒) 或 rad/s (弧度/秒)
 輸入範圍：非零正值

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.3 HIMC_GetMaxAcc

用途

取得軸的最大加速度。

語法

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

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_acc [out] 指標型態的記憶體，用來儲存軸的最大加速度。
 參數單位： m/s^2 (公尺/秒²) 或 rad/s^2 (弧度/秒²)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.4 HIMC_SetAcc

用途

設置軸的最大加速度。

語法

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

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- acc [in] 軸的新最大加速度。
參數單位： m/s^2 (公尺/秒²) 或 rad/s^2 (弧度/秒²)
輸入範圍：非零正值

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.5 HIMC_GetMaxDec

用途

取得軸的最大減速度。

語法

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

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_dec [out] 指標型態的記憶體，用來儲存軸的最大減速度。
 參數單位： m/s^2 (公尺/秒²) 或 rad/s^2 (弧度/秒²)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.6 HIMC_SetDec

用途

設置軸的最大減速度。

語法

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

參數

- ctrl_id [in]** HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in]** 軸編號。
- dec [in]** 軸的新最大減速度。
參數單位：m/s² (公尺/秒²) 或 rad/s² (弧度/秒²)
輸入範圍：非零正值

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.7 HIMC_GetSWRL

用途

取得軸的軟體右極限位置。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.8 HIMC_SetSWRL

用途

設置軸的軟體右極限位置。

語法

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

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- right_limit_pos [in] 軸的新軟體右極限位置。
參數單位：meter (公尺) 或 radian (弧度)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.9 HIMC_GetSWLL

用途

取得軸的軟體左極限位置。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.10 HIMC_SetSWLL

用途

設置軸的軟體左極限位置。

語法

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

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- left_limit_pos [in] 軸的新軟體左極限位置。
參數單位：meter (公尺) 或 radian (弧度)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.11 HIMC_GetSMTime

用途

取得軸的平滑時間。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.12 HIMC_SetSMTIME

用途

設置軸的平滑時間。

語法

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

參數

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

axis_id [in] 軸編號。

smooth_time [in] 軸的新平滑時間。
參數單位：second (秒)
輸入範圍：0.0~0.5

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.13 HIMC_GetMoveTime

用途

取得軸的路徑規劃時間。

語法

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

參數

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

axis_id [in] 軸編號。

p_move_time [out] 指標型態的記憶體，用來儲存軸的路徑規劃時間。
參數單位：second (秒)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.14 HIMC_GetSettlingTime

用途

取得軸的整定時間。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.15 HIMC_SetPos

用途

標出軸目前位置的值。

語法

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

參數

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

axis_id [in] 軸編號。

pos [in] 軸目前位置的值。
參數單位：meter (公尺) 或 radian (弧度)

回傳值

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

備註

當軸處於同步模式、加入軸群組或處於錯誤狀態時，此功能不適用。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.16 HIMC_GetPosFb

用途

取得軸的回授位置。

語法

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

參數

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

axis_id [in] 軸編號。

p_pos [out] 指標型態的記憶體，用來儲存軸的回授位置。
參數單位：meter (公尺) 或 radian (弧度)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.17 HIMC_GetPosOffset

用途

取得軸的位置偏移量。

語法

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

參數

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

axis_id [in] 軸編號。

p_pos [out] 指標型態的記憶體，用來儲存軸的位置偏移量。
參數單位：meter (公尺) 或 radian (弧度)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.18 HIMC_GetPosErr

用途

取得軸的跟隨誤差。

語法

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

參數

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

axis_id [in] 軸編號。

p_pos_err [out] 指標型態的記憶體，用來儲存軸的跟隨誤差。
 參數單位：meter (公尺) 或 radian (弧度)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.19 HIMC_GetRefPos

用途

取得軸的參考位置。

語法

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

參數

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

axis_id [in] 軸編號。

p_pos [out] 指標型態的記憶體，用來儲存軸的參考位置。
參數單位：meter (公尺) 或 radian (弧度)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.20 HIMC_GetRefVel

用途

取得軸的參考速度。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.21 HIMC_GetRefAcc

用途

取得軸的參考加速度。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.22 HIMC_GetPosOut

用途

取得由控制器送至驅動器的位置命令輸出。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.23 HIMC_GetVelOut

用途

取得由控制器送至驅動器的速度命令輸出。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.24 HIMC_GetAccOut

用途

取得由控制器送至驅動器的加速度命令輸出。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.25 HIMC_IgnoreHWL

用途

忽略觸碰到硬體極限的警告通知。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.26 HIMC_IgnoreSWL

用途

忽略觸碰到軟體極限的警告通知。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.27 HIMC_GetAxisNum

用途

取得連接至控制器的軸數量。

語法

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

參數

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

num [out] 指標型態的記憶體，用來儲存連接至控制器的軸數量。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.2.28 HIMC_GetMultiAxesFeedbackPos

用途

取得多軸的回授位置。

語法

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

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
p_axes_id_array [in]	指標型態的記憶體，用來儲存多軸的 ID。
num_of_axes [in]	軸的數量。
p_pos_array [out]	指標型態的記憶體，用來儲存多軸的回授位置。 參數單位：meter (公尺) 或 radian (弧度)

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3 軸狀態

3.3.1 HIMC_IsEnabled

用途

詢問軸的激磁狀態。

語法

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

參數

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

axis_id [in] 軸編號。

p_enabled [out] 指標型態的記憶體，用來儲存軸的激磁狀態。
若已激磁軸，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3.2 HIMC_IsMoving

用途

詢問軸的移動狀態。若軸正在移動，軌跡規劃器 (PG) 會持續輸出新的位置。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3.3 HIMC_IsInPos

用途

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

語法

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

參數

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

axis_id [in] 軸編號。

p_in_position [out] 指標型態的記憶體，用來儲存軸的到位狀態。
若軸已到位，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3.4 HIMC_IsErrorStop

用途

詢問軸是否處於 error stop 狀態。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3.5 HIMC_IsGantry

用途

詢問軸是否處於龍門狀態。

語法

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

參數

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

axis_id [in] 軸編號。

p_is_gantry [out] 指標型態的記憶體，用來儲存軸的龍門狀態。
若軸處於龍門狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3.6 HIMC_IsGrouped

用途

詢問軸是否被歸類至一個軸群組。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3.7 HIMC_IsSync

用途

詢問軸是否處於同步狀態。

語法

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

參數

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

axis_id [in] 軸編號。

p_is_sync [out] 指標型態的記憶體，用來儲存軸的同步狀態。
若軸處於同步狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3.8 HIMC_IsHWLL

用途

詢問軸是否觸發硬體左極限 (HWLL) 。

語法

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

參數

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

axis_id [in] 軸編號。

p_is_hwll [out] 指標型態的記憶體，用來儲存軸的 HWLL 狀態。
若軸處於 HWLL 狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2184.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3.9 HIMC_IsHWRL

用途

詢問軸是否觸發硬體右極限 (HWRL) 。

語法

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

參數

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

axis_id [in] 軸編號。

p_is_hwrl [out] 指標型態的記憶體，用來儲存軸的 HWRL 狀態。
若軸處於 HWRL 狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2184.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3.10 HIMC_IsSWLL

用途

詢問軸是否觸發軟體左極限 (SWLL) 。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2184.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3.11 HIMC_IsSWRL

用途

詢問軸是否觸發軟體右極限（SWRL）。

語法

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

參數

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

axis_id [in] 軸編號。

p_is_swrl [out] 指標型態的記憶體，用來儲存軸的 SWRL 狀態。
若軸處於 SWRL 狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2184.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

3.3.12 HIMC_IsCompActive

用途

詢問補償功能是否有效。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2184.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

(此頁有意留白。)

4. 同步運動函式



4.	同步運動函式	4-1
4.1	HIMC_EnableGear	4-2
4.2	HIMC_DisableGear	4-3
4.3	HIMC_GearIn	4-4
4.4	HIMC_GearOut	4-5

4.1 HIMC_EnableGear

用途

結合兩軸，使其成為主從關係。

語法

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

參數

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

axis_master_id [in] 主軸編號。

axis_slave_id [in] 從軸編號。

回傳值

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

備註

當兩軸都已激磁，此函式才適用。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

4.2 HIMC_DisableGear

用途

解除兩軸的主從關係，使其恢復兩獨立軸。

語法

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

參數

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

axis_slave_id [in] 從軸編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

4.3 HIMC_GearIn

用途

更改從軸的狀態：空檔→咬合。

語法

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

參數

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

回傳值

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

備註

當兩軸都已激磁，此函式才適用。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

4.4 HIMC_GearOut

用途

更改從軸的狀態：咬合→空檔。

語法

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

參數

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

axis_slave_id [in] 從軸編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

(此頁有意留白。)

5. 龍門函式



5.	龍門函式	5-1
5.1	HIMC_EnableGantryPair	5-2
5.2	HIMC_DisableGantryPair	5-3

5.1 HIMC_EnableGantryPair

用途

建立一對龍門。

語法

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

參數

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

lhs_axis_id [in] 左側軸之編號。

rhs_axis_id [in] 右側軸之編號。

回傳值

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

備註

當兩個軸都處於激磁狀態時，此函式才適用。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

5.2 HIMC_DisableGantryPair

用途

分開一對龍門。

語法

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

參數

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

axis_id [in] 龍門中任一軸之編號。

回傳值

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

備註

當兩個軸都處於解激磁狀態時，此函式才適用。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

(此頁有意留白。)

6. 軸群組函式

6.	軸群組函式	6-1
6.1	軸群組運動控制	6-2
6.1.1	HIMC_EnableGroup	6-2
6.1.2	HIMC_DisableGroup	6-3
6.1.3	基本運動命令	6-4
6.1.3.1	HIMC_LineAbs2D	6-4
6.1.3.2	HIMC_LineAbs3D	6-5
6.1.3.3	HIMC_LineRel2D	6-6
6.1.3.4	HIMC_LineRel3D	6-7
6.1.3.5	HIMC_Arc2D	6-8
6.1.3.6	HIMC_Circle2D	6-10
6.1.4	進階運動命令	6-12
6.1.4.1	HIMC_LinAbs	6-12
6.1.4.2	HIMC_LinRel	6-14
6.1.4.3	HIMC_CircAbs	6-16
6.1.5	HIMC_StopGroup	6-18
6.2	軸群組設定	6-19
6.2.1	HIMC_AddAxesToGrp	6-19
6.2.2	HIMC_RemoveAxisFromGrp	6-20
6.2.3	HIMC_SetupGroup	6-21
6.2.4	HIMC_UngrpAllAxes	6-22
6.2.5	HIMC_GetGroupID	6-23
6.2.6	HIMC_SetGrpMotionProfile	6-24
6.2.7	HIMC_SetGrpKin	6-26
6.2.8	HIMC_GroupReset	6-27
6.3	軸群組狀態	6-28
6.3.1	HIMC_IsGrpEnabled	6-28
6.3.2	HIMC_IsGrpMoving	6-29
6.3.3	HIMC_IsGrpInPos	6-30

6.1 軸群組運動控制

6.1.1 HIMC_EnableGroup

用途

致能軸群組。

語法

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

參數

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

group_id [in] 軸群組編號。

回傳值

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

備註

在執行此函式前，須激磁軸群組內的所有軸。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.1.2 HIMC_DisableGroup

用途

解致能軸群組。

語法

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

參數

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

group_id [in] 軸群組編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.1.3 基本運動命令

6.1.3.1 HIMC_LineAbs2D

用途

命令一個軸群組，二維內插線性移動至機器座標系統中的絕對位置。

語法

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

參數

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

group_id [in] 軸群組編號。

end_x [in] 絕對目標位置在 X 軸的值。

end_y [in] 絕對目標位置在 Y 軸的值。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.1.3.2 HIMC_LineAbs3D

用途

命令一個軸群組，三維內插線性移動至機器座標系統中的絕對位置。

語法

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

參數

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

group_id [in] 軸群組編號。

end_x [in] 絕對目標位置在 X 軸的值。

end_y [in] 絕對目標位置在 Y 軸的值。

end_z [in] 絕對目標位置在 Z 軸的值。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.1.3.3 HIMC_LineRel2D

用途

命令一個軸群組，二維內插線性移動至機器座標系統中的相對位置。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.1.3.4 HIMC_LineRel3D

用途

命令一個軸群組，三維內插線性移動至機器座標系統中的相對位置。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.1.3.5 HIMC_Arc2D

用途

命令一個軸群組，二維內插圓弧移動至機器座標系統中的絕對位置。

語法

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

參數

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

group_id [in] 軸群組編號。

border_x [in] 絕對中繼位置在 X 軸的值。

border_y [in] 絕對中繼位置在 Y 軸的值。

end_x [in] 絕對終點位置在 X 軸的值。

end_y [in] 絕對終點位置在 Y 軸的值。

回傳值

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

需求版本與檔案

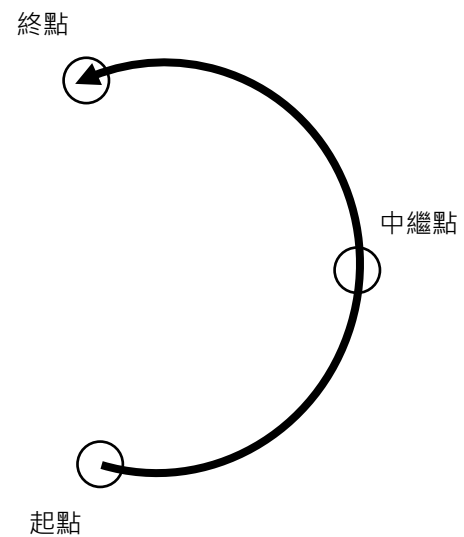
最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

優點

使用者可指定中繼點（運動中的最遠點），
並確保機器可取得此點。

缺點

在單一命令中，其角度被限制為 $< 2\pi$ 。



6.1.3.6 HIMC_Circle2D

用途

命令一個軸群組，二維內插圓周運動至機器座標系統中的絕對位置。

語法

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

參數

- table border="0">
- | | |
| --- | --- |
| ctrl_id [in] | HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。 |
- | | |
| --- | --- |
| group_id [in] | 軸群組編號。 |
- | | |
| --- | --- |
| center_x [in] | 絕對中心位置在 X 軸的值。 |
- | | |
| --- | --- |
| center_y [in] | 絕對中心位置在 Y 軸的值。 |
- | | |
| --- | --- |
| end_x [in] | 絕對終點位置在 X 軸的值。 |
- | | |
| --- | --- |
| end_y [in] | 絕對終點位置在 Y 軸的值。 |
- | | |
| --- | --- |
| turns [in] | 相對於起點的圓周運動圈數，決定了圓周運動的方向及總角度。 |

回傳值

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

需求版本與檔案

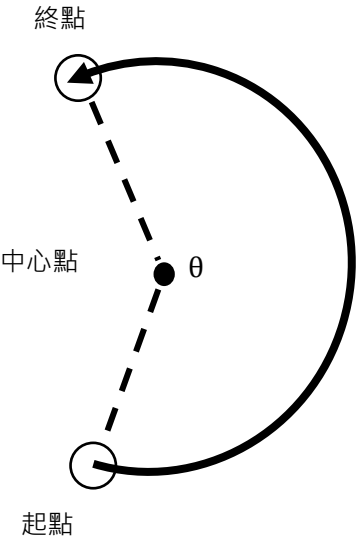
最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

優點

無角度限制。

缺點

使用者無法指定中繼點（運動中的最遠點）。
因此，機器不一定可以取得此點。



圈數的值決定了圓周運動的方向。

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

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

下表以 $\theta = 210^\circ$ 為例。

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

6.1.4 進階運動命令

6.1.4.1 HIMC_LinAbs

用途

命令一個軸群組，內插線性移動至特定座標系統中的絕對位置。

語法

```
int HIMC_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
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
target_pos [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的 6 自由度目標位置。 請參考節 17.2 CoordPosition。
motion_profile [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的運動設定。 請參考節 17.3 MotionProfile。
coord_sys [in]	指定合適的座標系統。 請參考節 18.2 CoordSystem。

- buff_mode [in]** 指定路徑緩衝模式。
請參考節 18.3 MotionBufferMode。
- trans_mode [in]** 指定路徑過渡模式。
請參考節 18.4 MotionTransitionMode。
- trans_par [in]** 指定特定過渡模式的參數。若不使用，請填入 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.1.4.2 HIMC_LinRel

用途

命令一個軸群組，內插線性移動至特定座標系統中的相對位置。

語法

```
int HIMC_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
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
relative_dist [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的 6 自由度相對距離。 請參考節 17.2 CoordPosition。
motion_profile [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的運動設定。 請參考節 17.3 MotionProfile。
coord_sys [in]	指定合適的座標系統。 請參考節 18.2 CoordSystem。
buff_mode [in]	指定路徑緩衝模式。 請參考節 18.3 MotionBufferMode。

- trans_mode [in]** 指定路徑過渡模式。
請參考節 18.4 MotionTransitionMode。
- trans_par [in]** 指定特定過渡模式的參數。若不使用，請填入 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.1.4.3 HIMC_CircAbs

用途

命令一個軸群組，內插圓周運動至特定座標系統中的絕對位置。

語法

```
int HIMC_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
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
center_pos [in]	指標型態的記憶體，用來儲存中心點。 請參考節 17.4 CenterPosition。
normal_vector [in]	指標型態的記憶體，用來儲存圓的法向量。 請參考節 17.5 NormalVector。
turns [in]	相對於起點的圓周運動圈數，決定了圓周運動的方向及總角度。

target_pos [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的 6 自由度目標位置。 請參考節 17.2 CoordPosition。
motion_profile [in]	指標型態的記憶體，用來儲存工具中心點（末端點）的運動設定。 請參考節 17.3 MotionProfile。
coord_sys [in]	指定合適的座標系統。 請參考節 18.2 CoordSystem。
buff_mode [in]	指定路徑緩衝模式。 請參考節 18.3 MotionBufferMode。
trans_mode [in]	指定路徑過渡模式。 請參考節 18.4 MotionTransitionMode。
trans_par [in]	指定特定過渡模式的參數。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.1.5 HIMC_StopGroup

用途

停止軸群組的運動。

語法

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

參數

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

group_id [in] 軸群組編號。

回傳值

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

備註

此函式會清除軸群組原有的路徑規畫。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.2 軸群組設定

6.2.1 HIMC_AddAxesToGrp

用途

將軸加入一個具有特定序列的軸群組中。

語法

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

參數

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

group_id [in] 軸群組編號。

num_of_axes [in] 要被加入軸群組的軸數量。(最大為 9)

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.2.2 HIMC_RemoveAxisFromGrp

用途

從一個具有特定序列的軸群組中移除最後一軸。

語法

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

參數

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

group_id [in] 軸群組編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.2.3 HIMC_SetupGroup

用途

設定並致能一個具有特定序列的軸群組。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.2.4 HIMC_UngrpAllAxes

用途

拆散軸群組。

語法

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

參數

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

group_id [in] 軸群組編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.2.5 HIMC_GetGroupID

目的

取得軸所屬的軸群組 ID。

語法

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

參數

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

axis_id [in] 軸編號。

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2188.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.2.6 HIMC_SetGrpMotionProfile

用途

在路徑上為一個群組運動設定 TCP 的最大切向運動。

語法

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

參數

ctrl_id [in]	<p>HIWIN 運動控制器的控制器 ID。</p> <p>須透過呼叫 HIMC_ConnectCtrl 函式來取得。</p>
group_id [in]	<p>軸群組編號。</p>
max_velocity [in]	<p>軸的新最大速度。</p> <p>參數單位：m/s (公尺/秒) 或 rad/s (弧度/秒)</p> <p>輸入範圍：非零正值</p>
max_acceleration [in]	<p>軸的新最大加速度。</p> <p>參數單位：m/s² (公尺/秒²) 或 rad/s² (弧度/秒²)</p> <p>輸入範圍：非零正值</p>
max_deceleration [in]	<p>軸的新最大減速度。</p> <p>參數單位：m/s² (公尺/秒²) 或 rad/s² (弧度/秒²)</p> <p>輸入範圍：非零正值</p>

smooth_time [in] 軸的新平滑時間。
參數單位：second (秒)
輸入範圍：0.0~0.5

回傳值

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

備註

軸群組運動的參數預設值分別為 [0.1 (速度) , 0.5 (加速度) , 0.5 (減速度) , 0.05 (平滑時間)]。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.2.7 HIMC_SetGrpKin

用途

設置軸群組的運動學模式。

語法

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

參數

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

group_id [in] 軸群組編號。

kin_type [in] 運動學模式。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.2.8 HIMC_GroupReset

用途

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

語法

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

參數

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

group_id [in] 軸群組編號。

回傳值

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

備註

清除所有的錯誤之後，才可使用此函式。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.3 軸群組狀態

6.3.1 HIMC_IsGrpEnabled

用途

詢問軸群組的致能狀態。

語法

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

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- p_is_grp_enabled [out] 指標型態的記憶體，用來儲存軸群組的致能狀態。
若已致能軸群組，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2184.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.3.2 HIMC_IsGrpMoving

用途

詢問軸群組的移動狀態。若軸群組正在移動，軌跡規劃器（PG）會持續輸出新的位置。

語法

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

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

6.3.3 HIMC_IsGrpInPos

用途

詢問軸群組的到位狀態。若軸群組已到位，跟隨誤差會小於所設定的目標框，並維持一段時間（反彈跳時間）。

語法

```
int HIMC_IsGrpInPos(
    int ctrl_id,
    int group_id,
    int *p_is_grp_inpos
);
```

參數

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

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7. GPIO 函式

7.	GPIO 函式.....	7-1
7.1	HIMC_GetHimcGpi	7-2
7.2	HIMC_GetHimcGpo.....	7-3
7.3	HIMC_SetHimcGpo	7-4
7.4	HIMC_GetHimcMultiGpi.....	7-5
7.5	HIMC_GetHimcMultiGpo	7-6
7.6	HIMC_SetHimcMultiGpo.....	7-7
7.7	HIMC_GetHimcGpiNum	7-8
7.8	HIMC_GetHimcGpoNum	7-9
7.9	HIMC_ToggleHimcGpo.....	7-10
7.10	HIMC_GetSlaveGpi	7-11
7.11	HIMC_GetSlaveGpo.....	7-12
7.12	HIMC_SetSlaveGpo	7-13
7.13	HIMC_GetSlaveMultiGpi.....	7-14
7.14	HIMC_GetSlaveMultiGpo	7-15
7.15	HIMC_SetSlaveMultiGpo.....	7-16
7.16	HIMC_GetSlaveGpiNum	7-17
7.17	HIMC_GetSlaveGpoNum	7-18
7.18	HIMC_ToggleSlaveGpo.....	7-19

7.1 HIMC_GetHimcGpi

用途

詢問控制器通用輸入的狀態。

語法

```
int HIMC_GetHimcGpi(
    int    ctrl_id,
    int    gpi_idx,
    char *p_state
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- gpi_idx [in] 通用輸入編號。
- p_state [out] 指標型態的記憶體，用來儲存特定輸入的狀態。
若輸入為導通狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2041.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.2 HIMC_GetHimcGpo

用途

詢問控制器通用輸出的狀態。

語法

```
int HIMC_GetHimcGpo(
    int    ctrl_id,
    int    gpo_idx,
    char *p_state
);
```

參數

- ctrl_id [in]** HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- gpo_idx [in]** 通用輸出編號。
- p_state [out]** 指標型態的記憶體，用來儲存特定輸出的狀態。
若輸出為導通狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2041.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.3 HIMC_SetHimcGpo

用途

設置控制器通用輸出的狀態。

語法

```
int HIMC_SetHimcGpo(  
    int ctrl_id,  
    int gpo_idx,  
    char state  
);
```

參數

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

gpo_idx [in] 通用輸出編號。

state [in] 要設置為特定輸出的狀態。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2041.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.4 HIMC_GetHimcMultiGpi

用途

詢問控制器多個通用輸入的狀態。

語法

```
int HIMC_GetHimcMultiGpi(
    int ctrl_id,
    int start_idx,
    int num,
    char *p_state_array
);
```

參數

- ctrl_id [in]** HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- start_idx [in]** 要取得的通用輸入的起始編號。
- num [in]** 要取得的通用輸入數量。
- p_state_array [out]** 指標型態的記憶體，用來儲存通用輸入的狀態，
其矩陣排序為[start_idx, ..., start_idx + num - 1]。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2111.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.5 HIMC_GetHimcMultiGpo

用途

詢問控制器多個通用輸出的狀態。

語法

```
int HIMC_GetHimcMultiGpo(
    int ctrl_id,
    int start_idx,
    int num,
    char *p_state_array
);
```

參數

- ctrl_id [in]** HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- start_idx [in]** 要取得的通用輸出的起始編號。
- num [in]** 要取得的通用輸出數量。
- p_state_array [out]** 指標型態的記憶體，用來儲存通用輸出的狀態，
其矩陣排序為[start_idx, ..., start_idx + num - 1]。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2111.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.6 HIMC_SetHimcMultiGpo

用途

設置控制器多個通用輸出的狀態。

語法

```
int HIMC_SetHimcMultiGpo(
    int ctrl_id,
    int start_idx,
    int num,
    char *p_state_array
);
```

參數

- ctrl_id [in]** HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- start_idx [in]** 要設置的通用輸出的起始編號。
- num [in]** 要設置的通用輸出數量。
- p_state_array [in]** 指標型態的記憶體，用來設置通用輸出的狀態，
其矩陣排序為[start_idx, ..., start_idx + num - 1]。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2111.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.7 HIMC_GetHimcGpiNum

用途

取得控制器通用輸入的數量。

語法

```
int HIMC_GetHimcGpiNum(
    int ctrl_id,
    int *p_gpi_num
);
```

參數

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

p_gpi_num [out] 指標型態的記憶體，用來儲存控制器通用輸入的數量。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2041.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.8 HIMC_GetHimcGpoNum

用途

取得控制器通用輸出的數量。

語法

```
int HIMC_GetHimcGpoNum(
    int ctrl_id,
    int *p_gpo_num
);
```

參數

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

p_gpo_num [out] 指標型態的記憶體，用來儲存控制器通用輸出的數量。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2041.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.9 HIMC_ToggleHimcGpo

用途

切換控制器通用輸出的狀態。

語法

```
int HIMC_ToggleHimcGpo(
    int ctrl_id,
    int gpo_idx
);
```

參數

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

gpo_idx [in] 通用輸出編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.10 HIMC_GetSlaveGpi

用途

詢問從站通用輸入的狀態。

語法

```
int HIMC_GetSlaveGpi(
    int ctrl_id,
    int slave_id,
    int gpi_idx,
    char *p_state
);
```

參數

- ctrl_id [in]** HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slave_id [in]** 從站 ID。
- gpi_idx [in]** 通用輸入編號。
- p_state [out]** 指標型態的記憶體，用來儲存特定輸入的狀態。
若輸入為導通狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2041.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.11 HIMC_GetSlaveGpo

用途

詢問從站通用輸出的狀態。

語法

```
int HIMC_GetSlaveGpo(
    int ctrl_id,
    int slave_id,
    int gpo_idx,
    char *p_state
);
```

參數

- ctrl_id [in]** HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slave_id [in]** 從站 ID。
- gpo_idx [in]** 通用輸出編號。
- p_state [out]** 指標型態的記憶體，用來儲存特定輸出的狀態。
若輸出為導通狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2041.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.12 HIMC_SetSlaveGpo

用途

設置從站通用輸出的狀態。

語法

```
int HIMC_SetSlaveGpo(
    int ctrl_id,
    int slave_id,
    int gpo_idx,
    char state
);
```

參數

- ctrl_id [in]** HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slave_id [in]** 從站 ID。
- gpo_idx [in]** 通用輸出編號。
- state [in]** 要設置為特定輸出的狀態。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2041.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.13 HIMC_GetSlaveMultiGpi

用途

詢問從站多個通用輸入的狀態。

語法

```
int HIMC_GetSlaveMultiGpi(
    int ctrl_id,
    int slave_id,
    int start_idx,
    int num,
    char *p_state_array
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slave_id [in] 從站 ID。
- start_idx [in] 要取得的通用輸入的起始編號。
- num [in] 要取得的通用輸入數量。
- p_state_array [out] 指標型態的記憶體，用來儲存通用輸入的狀態，
其矩陣排序為[start_idx, ..., start_idx + num – 1]。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2111.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.14 HIMC_GetSlaveMultiGpo

用途

詢問從站多個通用輸出的狀態。

語法

```
int HIMC_GetSlaveMultiGpo(
    int ctrl_id,
    int slave_id,
    int start_idx,
    int num,
    char *p_state_array
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slave_id [in] 從站 ID。
- start_idx [in] 要取得的通用輸出的起始編號。
- num [in] 要取得的通用輸出數量。
- p_state_array [out] 指標型態的記憶體，用來儲存通用輸出的狀態，
其矩陣排序為[start_idx, ..., start_idx + num - 1]。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2111.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.15 HIMC_SetSlaveMultiGpo

用途

設置從站多個通用輸出的狀態。

語法

```
int HIMC_SetSlaveMultiGpo(
    int ctrl_id,
    int slave_id,
    int start_idx,
    int num,
    char *p_state_array
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- slave_id [in] 從站 ID。
- start_idx [in] 要設置的通用輸出的起始編號。
- num [in] 要設置的通用輸出數量。
- p_state_array [out] 指標型態的記憶體，用來儲存通用輸出的狀態，
其矩陣排序為[start_idx, ..., start_idx + num – 1]。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2111.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.16 HIMC_GetSlaveGpiNum

用途

取得從站通用輸入的數量。

語法

```
int HIMC_GetSlaveGpiNum(
    int ctrl_id,
    int slave_id,
    int *p_gpi_num
);
```

參數

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

slave_id [in] 從站 ID。

p_gpi_num [out] 指標型態的記憶體，用來儲存從站通用輸入的數量。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2041.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.17 HIMC_GetSlaveGpoNum

用途

取得從站通用輸出的數量。

語法

```
int HIMC_GetSlaveGpoNum(
    int ctrl_id,
    int slave_id,
    int *p_gpo_num
);
```

參數

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

slave_id [in] 從站 ID。

p_gpo_num [out] 指標型態的記憶體，用來儲存從站通用輸出的數量。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2041.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

7.18 HIMC_ToggleSlaveGpo

用途

切換從站通用輸出的狀態。

語法

```
int HIMC_ToggleSlaveGpo(
    int ctrl_id,
    int slave_id,
    int gpo_idx
);
```

參數

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

slave_id [in] 從站 ID。

gpo_idx [in] 通用輸出編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

(此頁有意留白。)

8. User Table 函数

8.	User Table 函数.....	8-1
8.1	HIMC_SetUserTable	8-2
8.2	HIMC_GetUserTable.....	8-3
8.3	HIMC_SetTableValue	8-4
8.4	HIMC_GetTableValue	8-5
8.5	HIMC_SaveUserTable	8-6
8.6	HIMC_LoadUserTable.....	8-7

8.1 HIMC_SetUserTable

用途

設置 User Table 的數據至控制器中。

語法

```
int HIMC_SetUserTable(
    int    ctrl_id,
    double *p_user_table_data,
    int    start_idx,
    int    number_of_doubles_to_write
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
p_user_table_data [in]	指標型態的記憶體，將數據寫入 User Table。
start_idx [in]	User Table 的起始編號。
number_of_doubles_to_write [in]	被寫入之 User Table 的數量。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

8.2 HIMC_GetUserTable

用途

從控制器中取得 User Table 的數據。

語法

```
int HIMC_GetUserTable(
    int    ctrl_id,
    double *p_user_table_data,
    int    start_idx,
    int    number_of_doubles_to_read
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
p_user_table_data [out]	指標型態的記憶體，用來儲存來自 User Table 的數據。
start_idx [in]	User Table 的起始編號。
number_of_doubles_to_read [in]	被讀取之 User Table 的數量。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

8.3 HIMC_SetTableValue

用途

將數據寫入 User Table 的特定編號中。

語法

```
int HIMC_SetTableValue(
    int    ctrl_id,
    int    index,
    double value
);
```

參數

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

index [in] User Table 的編號。

value [in] 輸入數據。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

8.4 HIMC_GetTableValue

用途

從 User Table 的特定編號中取得數據。

語法

```
int HIMC_GetTableValue(
    int    ctrl_id,
    int    index,
    double *value
);
```

參數

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

index [in] User Table 的編號。

value [out] 指標型態的記憶體，用來儲存數據。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

8.5 HIMC_SaveUserTable

用途

將在 RAM 中的 User Table 數據存入永久記憶體中。

語法

```
int HIMC_SaveUserTable(  
    int ctrl_id,  
    int start_idx,  
    int num_data  
);
```

參數

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

start_idx [in] User Table 的起始編號。

num_data [in] 被儲存元素的數量。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2184.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

8.6 HIMC_LoadUserTable

用途

將永久記憶體中的 User Table 數據載入至 RAM。

語法

```
int HIMC_LoadUserTable(
    int ctrl_id,
    int start_idx,
    int num_data
);
```

參數

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

start_idx [in] User Table 的起始編號。

num_data [in] 被載入元素的數量。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2184.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

(此頁有意留白。)

9. 位置觸發函式



9.	位置觸發函式	9-1
9.1	HIMC_EnablePT	9-2
9.2	HIMC_DisablePT	9-3
9.3	HIMC_IsPTEnabled	9-4
9.4	HIMC_SetPosTriggerConfig	9-5

9.1 HIMC_EnablePT

用途

致能軸的位置觸發功能。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

9.2 HIMC_DisablePT

用途

解致能軸的位置觸發功能。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

9.3 HIMC_IsPTEnabled

用途

詢問是否已致能位置觸發功能。

語法

```
int HIMC_IsPTEnabled(
    int ctrl_id,
    int axis_id,
    int *p_is_pt_enabled
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_pt_enabled [out] 指標型態的記憶體，用來儲存位置觸發的致能狀態。
若軸處於 PT enabled 狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

9.4 HIMC_SetPosTriggerConfig

用途

設定軸的位置觸發配置。

語法

```
int HIMC_SetPosTriggerConfig(  
    int ctrl_id,  
    int axis_id,  
    PosTriggerPar *pos_trigger_par  
);
```

參數

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

axis_id [in] 軸編號。

pos_trigger_par [in] 指標型態的記憶體，用來儲存軸的位置觸發配置。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

(此頁有意留白)

10. Touch Probe 函式



10.	Touch Probe 函式.....	10-1
10.1	HIMC_EnableTouchProbe.....	10-2
10.2	HIMC_DisableTouchProbe	10-3
10.3	HIMC_IsTouchProbeEnabled.....	10-4
10.4	HIMC_IsTouchProbeTriggered	10-5
10.5	HIMC_GetTouchProbePos.....	10-6

10.1 HIMC_EnableTouchProbe

用途

致能軸的 Touch Probe 功能。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

10.2 HIMC_DisableTouchProbe

用途

解致能軸的 Touch Probe 功能。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

10.3 HIMC_IsTouchProbeEnabled

用途

詢問是否已致能 Touch Probe 功能。

語法

```
int HIMC_IsTouchProbeEnabled(
    int ctrl_id,
    int axis_id,
    int *p_is_probe_enabled
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- p_is_probe_enabled [out] 指標型態的記憶體，用來儲存 Touch Probe 的致能狀態。
若軸處於 touch probe enabled 狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

10.4 HIMC_IsTouchProbeTriggered

用途

詢問是否已觸發 Touch Probe 功能。

語法

```
int HIMC_IsTouchProbeTriggered(
    int ctrl_id,
    int axis_id,
    int *p_is_probe_triggered
);
```

參數

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

axis_id [in] 軸編號。

p_is_probe_triggered [out] 指標型態的記憶體，用來儲存 Touch Probe 的觸發狀態。
若軸處於 touch probe triggered 狀態，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

10.5 HIMC_GetTouchProbePos

用途

取得軸的 touch probe 位置。

語法

```
int HIMC_GetTouchProbePos(
    int    ctrl_id,
    int    axis_id,
    double *p_get_probe_pos
);
```

參數

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

axis_id [in] 軸編號。

p_get_probe_pos [out] 指標型態的記憶體，用來儲存軸的 touch probe 位置。
參數單位：meter (公尺) 或 radian (弧度)

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

備註

當軸處於 gantry 模式，此函式不適用。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

11. 動態誤差補償函式

11.	動態誤差補償函式.....	11-1
11.1	HIMC_EnableComp	11-2
11.2	HIMC_DisableComp.....	11-3
11.3	HIMC_SetupComp	11-4
11.4	HIMC_SetupComp2D	11-6

11.1 HIMC_EnableComp

用途

啟動軸的動態誤差補償。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

備註

當軸處於激磁狀態，此函式不適用。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

11.2 HIMC_DisableComp

用途

取消軸的動態誤差補償。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

備註

- (1) 軸的參考位置將被重新設定為目前位置。
- (2) 當軸處於激磁狀態，此函式不適用。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

11.3 HIMC_SetupComp

用途

設定軸的一維動態誤差補償。

語法

```
int HIMC_SetupComp(
    int    ctrl_id,
    int    axis_id,
    int    start_idx,
    double base_val,
    double interval,
    int    num_pt,
    int    ref_axis_id
);
```

參數

- table border="0">
- | | |
| --- | --- |
| ctrl_id [in] | HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。 |
- | | |
| --- | --- |
| axis_id [in] | 軸編號。 |
- | | |
| --- | --- |
| start_idx [in] | User Table 中，補償點的起始編號。 |
- | | |
| --- | --- |
| base_val [in] | 基值（補償輸入的最小值）。 參數單位：meter（公尺）或 radian（弧度） |
- | | |
| --- | --- |
| interval [in] | 相鄰補償點的間距。 參數單位：meter（公尺）或 radian（弧度） |
- | | |
| --- | --- |
| num_pt [in] | 補償點的數量。 |
- | | |
| --- | --- |
| ref_axis_id [in] | 參考軸的編號。 |

回傳值

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

備註

當被補償的軸處於解激磁狀態，此函式不適用。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

11.4 HIMC_SetupComp2D

用途

設定軸的二維動態誤差補償。

語法

```
int HIMC_SetupComp2D(
    int    ctrl_id,
    int    axis_id,
    int    start_idx,
    double *base_val,
    double *interval,
    int    *num_pt,
    int    *ref_axis_id
);
```

參數

- table border="0">
- | | |
| --- | --- |
| ctrl_id [in] | HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。 |
- | | |
| --- | --- |
| axis_id [in] | 軸編號。 |
- | | |
| --- | --- |
| start_idx [in] | User Table 中，補償點的起始編號。 |
- | | |
| --- | --- |
| base_val [in] | 指向兩元素陣列的指標，內含各維的基值（補償輸入的最小值）。 參數單位：meter（公尺）或 radian（弧度） |
- | | |
| --- | --- |
| interval [in] | 指向兩元素陣列的指標，內含各維相鄰補償點的間距。 參數單位：meter（公尺）或 radian（弧度） |
- | | |
| --- | --- |
| num_pt [in] | 指向兩元素陣列的指標，內含各維補償點的數量。 |
- | | |
| --- | --- |
| ref_axis_id [in] | 指向兩元素陣列的指標，內含各維參考軸的編號。 |

回傳值

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

備註

當被補償的軸處於解激磁狀態，此函式不適用。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

(此頁有意留白。)

12. 濾波器函式

12.	濾波器函式.....	12-1
12.1	HIMC_EnableAxisVsf.....	12-2
12.2	HIMC_DisableAxisVsf.....	12-3
12.3	HIMC_SetAxisVsf	12-4
12.4	HIMC_EnableAxisInShape.....	12-6
12.5	HIMC_DisableAxisInShape	12-7
12.6	HIMC_SetAxisInShape.....	12-8
12.7	HIMC_EnableGrpInShape	12-10
12.8	HIMC_DisableGrpInShape.....	12-11
12.9	HIMC_SetGrpInShape.....	12-12

12.1 HIMC_EnableAxisVsf

用途

啟動軸的 VSF 濾波器。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

備註

當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

12.2 HIMC_DisableAxisVsf

用途

取消軸的 VSF 濾波器。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

12.3 HIMC_SetAxisVsf

用途

設置軸的 VSF 濾波器參數。

語法

```
int HIMC_SetAxisVsf(
    int ctrl_id,
    int axis_id,
    double frequency,
    double damping_ratio
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
frequency [in]	系統頻率。 參數單位：Hz (赫茲) 輸入範圍：0.1~200
damping_ratio [in]	阻尼比。 輸入範圍：0.7~1.5 (建議輸入 1.0)

回傳值

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

備註

當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

12.4 HIMC_EnableAxisInShape

用途

啟動軸的 InShape 濾波器。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

備註

當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

12.5 HIMC_DisableAxisInShape

用途

取消軸的 InShape 濾波器。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

12.6 HIMC_SetAxisInShape

用途

設置軸的 InShape 濾波器參數。

語法

```
int HIMC_SetAxisInShape(
    int ctrl_id,
    int axis_id,
    double frequency,
    double damping_ratio,
    ShaperMode shaper_type
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
axis_id [in]	軸編號。
frequency [in]	系統頻率。 參數單位：Hz (赫茲) 輸入範圍：3.0~300
damping_ratio [in]	阻尼比。 輸入範圍：0.0~0.3
shaper_type [in]	有兩種 shaper 類型： Shaper_Normal 與 Shaper_Robust 。 Shaper_Robust 的效果比 Shaper_Normal 更強，但 Shaper_Normal 的強度足以抑制振動。

回傳值

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

備註

- (1) 當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。
- (2) 系統頻率與阻尼比的預設值分別為 5.5Hz 與 0.03。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

12.7 HIMC_EnableGrpInShape

用途

啟動軸群組的 InShape 濾波器。

語法

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

參數

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

group_id [in] 軸群組編號。

回傳值

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

備註

當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

12.8 HIMC_DisableGrpInShape

用途

取消軸群組的 InShape 濾波器。

語法

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

參數

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

group_id [in] 軸群組編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

12.9 HIMC_SetGrpInShape

用途

設置軸群組的 InShape 濾波器參數。

語法

```
int HIMC_SetGrpInShape(
    int ctrl_id,
    int group_id,
    double frequency,
    double damping_ratio,
    ShaperMode shaper_type
);
```

參數

ctrl_id [in]	HIWIN 運動控制器的控制器 ID。 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
group_id [in]	軸群組編號。
frequency [in]	系統頻率。 參數單位：Hz (赫茲) 輸入範圍：3.0~300
damping_ratio [in]	阻尼比。 輸入範圍：0.0~0.3
shaper_type [in]	有兩種 shaper 類型： Shaper_Normal 與 Shaper_Robust 。 Shaper_Robust 的效果比 Shaper_Normal 更強，但 Shaper_Normal 的強度足以抑制振動。

回傳值

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

備註

- (1) 當馬達正在移動時，此函式不適用。否則，馬達將產生非預期的振動。
- (2) 系統頻率與阻尼比的預設值分別為 5.5Hz 與 0.03。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

(此頁有意留白。)

13. HMPL Task 函式

13.	HMPL Task 函式	13-1
13.1	HIMC_StartTask.....	13-2
13.2	HIMC_StartTaskFunc.....	13-3
13.3	HIMC_StopTask.....	13-4
13.4	HIMC_StopAllTask	13-5
13.5	HIMC_IsTaskStop	13-6

13.1 HIMC_StartTask

用途

開始執行 HMPL task。

語法

```
int HIMC_StartTask(
    int ctrl_id,
    int task_id
);
```

參數

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

task_id [in] HMPL task ID。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

13.2 HIMC_StartTaskFunc

用途

開始執行 HMPL task 中的一個函式。

語法

```
int HIMC_StartTaskFunc(
    int ctrl_id,
    int task_id,
    char *func_name
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- task_id [in] HMPL task ID。
- func_name [in] 指標型態的記憶體，用來儲存 HMPL task 中的函式名稱。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

13.3 HIMC_StopTask

用途

停止執行 HMPL task。

語法

```
int HIMC_StopTask(
    int ctrl_id,
    int task_id
);
```

參數

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

task_id [in] HMPL task ID。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

13.4 HIMC_StopAllTask

用途

停止執行所有 HMPL task (包含呼叫者) 。

語法

```
int HIMC_StopAllTask(
    int ctrl_id
);
```

參數

ctrl_id [in] HIWIN 運動控制器的控制器 ID 。

 須透過呼叫 HIMC_ConnectCtrl 函式來取得 。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

13.5 HIMC_IsTaskStop

用途

詢問是否已停止執行 HMPL task。

語法

```
int HIMC_IsTaskStop(
    int ctrl_id,
    int task_id,
    int *isStop
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- task_id [in] HMPL task ID。
- isStop [out] 指標型態的記憶體，用來儲存 HMPL task 的狀態。
 若 HMPL task 已停止，其值為 1，否則為 0。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

14. 回調函式



14.	回調函式	14-1
14.1	HIMC_SetHmplEvtCallback	14-2
14.2	HIMC_SetErrorCallback.....	14-3

14.1 HIMC_SetHmplEvtCallback

用途

顯示回調函式，以取得 HMPL task 所傳送的事件。

語法

```
int HIMC_SetHmplEvtCallback(
    int ctrl_id,
    HMPLEventCBFuncPtr hmpl_event_cb_func_ptr
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- hmpl_event_cb_func_ptr [in] 回調函式的指標，其原型為 void func(int arg)。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

14.2 HIMC_SetErrorCallback

用途

顯示回調函式，以取得控制器所傳送的錯誤 ID。

語法

```
int HIMC_SetErrorCallback(
    int ctrl_id,
    HimcErrorCBFuncPtr himc_error_cb_func_ptr
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- hmc_error_cb_func_ptr [in] 回調函式的指標，其原型為 void func(int arg)。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2080.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

(此頁有意留白。)

15. 參數操作函式

15.	參數操作函式	15-1
15.1	HIMC_GetVariableByID	15-2
15.2	HIMC_SetVariableByID	15-3
15.3	HIMC_GetVariableListByID	15-4
15.4	HIMC_SetVariableListByID	15-5
15.5	HIMC_GetGlobalVariables	15-6
15.6	HIMC_SetGlobalVariables	15-7
15.7	HIMC_GetSlvVar	15-8
15.8	HIMC_SetSlvVar	15-9
15.9	HIMC_GetSlvSt	15-10
15.10	HIMC_SetSlvSt	15-11

15.1 HIMC_GetVariableByID

用途

藉 ID 取得控制器的變數值。

語法

```
int HIMC_GetVariableByID(  
    int    ctrl_id,  
    int    var_id,  
    double *p_val  
);
```

參數

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

var_id [in] 變數 ID。
 註：從 hcv_id.h 檔案中取得此 ID。

p_val [out] 指標型態的記憶體，用來儲存變數值。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

15.2 HIMC_SetVariableByID

用途

藉 ID 設置控制器的變數值。

語法

```
int HIMC_SetVariableByID(  
    int    ctrl_id,  
    int    var_id,  
    double val  
);
```

參數

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

var_id [in] 變數 ID。
 註：從 hcv_id.h 檔案中取得此 ID。

val [in] 新的變數值。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

15.3 HIMC_GetVariableListByID

用途

藉 ID 取得控制器的多個變數值。

語法

```
int HIMC_GetVariableListByID(
    int    ctrl_id,
    int    *p_var_id,
    int    num,
    double *p_val
);
```

參數

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

p_var_id [in] 指標型態的記憶體，用來儲存多個變數的 ID。
註：從 hcv_id.h 檔案中取得 ID。

num [in] 變數的數量。

p_val [out] 指標型態的記憶體，用來儲存多個變數值。

回傳值

若函式執行成功，將回傳 int 型態的值 0。若失敗，則回傳非零值。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

15.4 HIMC_SetVariableListByID

用途

藉 ID 設置控制器的多個變數值。

語法

```
int HIMC_SetVariableListByID(  
    int    ctrl_id,  
    int    *p_var_id,  
    int    num,  
    double *p_val  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- p_var_id [in] 指標型態的記憶體，用來儲存多個變數的 ID。
 註：從 hcv_id.h 檔案中取得 ID。
- num [in] 變數的數量。
- p_val [in] 指標型態的記憶體，用來儲存多個新的變數值。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

15.5 HIMC_GetGlobalVariables

用途

取得控制器的多個全域變數值。

語法

```
int HIMC_GetGlobalVariables(
    int    ctrl_id,
    char    **pp_var_name_array,
    int    length,
    double *p_output_array
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- pp_var_name_array [in] 指標型態的記憶體，用來儲存多個全域變數的名稱。
- length [in] 全域變數的數量。
- p_output_array [out] 指標型態的記憶體，用來儲存多個全域變數值。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

15.6 HIMC_SetGlobalVariables

用途

設置控制器的多個全域變數值。

語法

```
int HIMC_SetGlobalVariables(  
    int    ctrl_id,  
    char    **pp_var_name_array,  
    int    length,  
    double *p_input_array  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- pp_var_name_array [in] 指標型態的記憶體，用來儲存多個全域變數的名稱。
- length [in] 全域變數的數量。
- p_input_array [in] 指標型態的記憶體，用來儲存多個新的全域變數值。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

15.7 HIMC_GetSlvVar

用途

取得從站的變數值。

語法

```
int HIMC_GetSlvVar(
    int ctrl_id,
    int slave_id,
    const char *var_name,
    double *p_var
);
```

參數

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

slave_id [in] 從站編號。

var_name [in] 指標型態的記憶體，用來儲存變數名稱。

p_var [out] 指標型態的記憶體，用來儲存變數值。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

15.8 HIMC_SetSlvVar

用途

設置從站的變數值。

語法

```
int HIMC_SetSlvVar(  
    int ctrl_id,  
    int slave_id,  
    const char *var_name,  
    double value  
);
```

參數

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

slave_id [in] 從站編號。

var_name [in] 指標型態的記憶體，用來儲存變數名稱。

value [in] 新的變數值。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

15.9 HIMC_GetSlvSt

用途

取得從站的狀態。

語法

```
int HIMC_GetSlvSt(
    int ctrl_id,
    int slave_id,
    const char *st_name,
    int *on_off
);
```

參數

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

slave_id [in] 從站編號。

st_name [in] 指標型態的記憶體，用來儲存狀態名稱。

on_off [out] 指標型態的記憶體，用來儲存從站的狀態。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

15.10 HIMC_SetSlvSt

用途

設置從站的狀態。

語法

```
int HIMC_SetSlvSt(  
    int ctrl_id,  
    int slave_id,  
    const char *st_name,  
    int on_off  
);
```

參數

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

slave_id [in] 從站編號。

st_name [in] 指標型態的記憶體，用來儲存狀態名稱。

on_off [in] 新的從站狀態。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

(此頁有意留白。)

16. HIMC 錯誤函式

16.	HIMC 錯誤函式	16-1
16.1	HIMC_GetLastError	16-2
16.2	HIMC_GetAxisLastErr	16-3
16.3	HIMC_ClearAxisLastErr	16-4
16.4	HIMC_GetGrpLastErr	16-5
16.5	HIMC_ClearGrpLastErr	16-6
16.6	HIMC_GetErrorInformation	16-7

16.1 HIMC_GetLastError

用途

取得控制器的最新錯誤代碼。

語法

```
int HIMC_GetLastError(
    int ctrl_id,
    int *p_error_code
);
```

參數

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

p_error_code [out] 指標型態的記憶體，用來儲存控制器的最新錯誤代碼。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

16.2 HIMC_GetAxisLastErr

用途

取得軸的最新錯誤代碼。

語法

```
int HIMC_GetAxisLastErr(  
    int ctrl_id,  
    int axis_id,  
    int *err_code  
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- axis_id [in] 軸編號。
- err_code [out] 指標型態的記憶體，用來儲存軸的最新錯誤代碼。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

16.3 HIMC_ClearAxisLastErr

用途

清除軸的最新錯誤代碼。

語法

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

參數

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

axis_id [in] 軸編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

16.4 HIMC_GetGrpLastErr

用途

取得軸群組的最新錯誤代碼。

語法

```
int HIMC_GetGrpLastErr(
    int ctrl_id,
    int group_id,
    int *err_code
);
```

參數

- ctrl_id [in] HIWIN 運動控制器的控制器 ID。
 須透過呼叫 HIMC_ConnectCtrl 函式來取得。
- group_id [in] 軸群組編號。
- err_code [out] 指標型態的記憶體，用來儲存軸群組的最新錯誤代碼。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

16.5 HIMC_ClearGrpLastErr

用途

清除軸群組的最新錯誤代碼。

語法

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

參數

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

group_id [in] 軸群組編號。

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

16.6 HIMC_GetErrorInformation

用途

取得特定錯誤 ID 的資訊。

語法

```
int HIMC_GetErrorInfomation(  
    int error_id,  
    char *p_name,  
    int name_buff_len,  
    int *p_name_actual_len,  
    char *p_description,  
    int description_buff_len,  
    int *p_description_actual_len,  
);
```

參數

error_id [in]	指定一個錯誤 ID。
p_name [out]	指標型態的記憶體，用來儲存特定錯誤 ID 的錯誤名稱。
name_buff_len [in]	指定儲存錯誤名稱的最大字元數。
p_name_actual_len [out]	指標型態的記憶體， 用來儲存錯誤名稱的實際字元數（不包括空字元。）
p_description [out]	指標型態的記憶體，用來儲存特定錯誤 ID 的錯誤描述。
description_buff_len [in]	指定儲存錯誤描述的最大字元數。
p_description_actual_len [out]	指標型態的記憶體， 用來儲存錯誤描述的實際字元數（不包括空字元。）

回傳值

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

需求版本與檔案

最低支援版本	iA Studio 0.23.2101.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

17. 數據結構

17.	數據結構	17-1
17.1	ComInfo	17-2
17.2	CoordPosition.....	17-3
17.3	MotionProfile.....	17-4
17.4	CenterPosition.....	17-5
17.5	NormalVector	17-6
17.6	PosTriggerPar	17-7

17.1 ComInfo

用途

定義連線類型和資訊。

語法

```
typedef struct {
    ComType type;
    struct {
        char ip[20];
        char port[12];
    } TCP_IP;

    struct {
        char com_port_name[80];
        int baud_rate;
    } RS232;

    struct {
        char autoExecExe;
    } Simulator;
} ComInfo;
```

成員

- ComType 連線類型。
- TCP_IP 含網絡連線參數、IP 與端口的結構。
- RS232 含 RS232 連線參數、序列埠名稱與波特率的結構。
- Simulator 含 Simulator 連線參數與 autoExecExe 的結構。

需求版本與檔案

最低支援版本	iA Studio 0.23.1892.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

17.2 CoordPosition

用途

定義座標運動的位置或距離。

語法

```
typedef struct {
    double x_pos;
    double y_pos;
    double z_pos;
    double a_pos;
    double b_pos;
    double c_pos;
} CoordPosition, *PCoordPosition;
```

成員

x_pos 終點的線性位置 X。單位：meter (公尺)

y_pos 終點的線性位置 Y。單位：meter (公尺)

z_pos 終點的線性位置 Z。單位：meter (公尺)

a_pos 終點的定向角 A。單位：radian (弧度)

b_pos 終點的定向角 B。單位：radian (弧度)

c_pos 終點的定向角 C。單位：radian (弧度)

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

17.3 MotionProfile

用途

定義運動軌跡設定。

語法

```
typedef struct {
    double max_vel;
    double max_acc;
    double max_dec;
    double smooth_time;
} MotionProfile, *PMotionProfile;
```

成員

max_vel	<p>線性運動的最大切向速度。</p> <p>單位：m/s (公尺/秒) 或 rad/s (弧度/秒)</p> <p>範圍：非零正值</p>
max_acc	<p>線性運動的最大切向加速度。</p> <p>單位：m/s² (公尺/秒²) 或 rad/s² (弧度/秒²)</p> <p>範圍：非零正值</p>
max_dec	<p>線性運動的最大切向減速度。</p> <p>單位：m/s² (公尺/秒²) 或 rad/s² (弧度/秒²)</p> <p>範圍：非零正值</p>
smooth_time	<p>線性運動的平滑時間。</p> <p>單位：second (秒)</p> <p>範圍：0.0~0.5</p>

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

17.4 CenterPosition

用途

定義中心點設定。

語法

```
typedef struct {
    double x_pos;
    double y_pos;
    double z_pos;
} CenterPosition, *PCenterPosition;
```

成員

x_pos X 軸位置。單位：meter (公尺)

y_pos Y 軸位置。單位：meter (公尺)

z_pos Z 軸位置。單位：meter (公尺)

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

17.5 NormalVector

用途

定義法向量設定。

語法

```
typedef struct {
    double x_vector;
    double y_vector;
    double z_vector;
} NormalVector, *PNormalVector;
```

成員

x_vector X 方向向量。

y_vector Y 方向向量。

z_vector Z 方向向量。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

17.6 PosTriggerPar

用途

定義位置觸發設定。

語法

```
typedef struct {
    double start_pos;
    double end_pos;
    double interval;
    int pulse_width;
    int polarity;
} PosTriggerPar, *PPosTriggerPar;
```

成員

- start_pos** PT 功能的起點。
單位：meter (公尺) 或 radian (弧度)
- end_pos** PT 功能的終點。
單位：meter (公尺) 或 radian (弧度)
- interval** 連續 PT 輸出的位置間距。
單位：meter (公尺) 或 radian (弧度)
- pulse_width** 每個 PT 輸出信號的寬度。
單位：nanometer (奈秒)
- polarity** 電子信號極性輸出。0 表示常態為低態，1 表示常態為高態。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

(此頁有意留白。)

18. 列舉



18.	列舉	18-1
18.1	ComType.....	18-2
18.2	CoordSystem	18-3
18.3	MotionBufferMode	18-4
18.4	MotionTransitionMode	18-5
18.5	ShaperMode.....	18-6

18.1 ComType

定義

連線類型列舉。

語法

```
typedef enum {
    COM_TYPE_TCPIP,
    COM_TYPE_RS232,
    COM_TYPE_SIMULATOR
} ComType;
```

成員

COM_TYPE_TCPIP	連線類型為 TCPIP。
COM_TYPE_RS232	連線類型為 RS232。
COM_TYPE_SIMULATOR	連線類型為模擬器。

需求版本與檔案

最低支援版本	iA Studio 0.23.1892.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

18.2 CoordSystem

定義

座標系統。

語法

```
typedef enum {
    kCoord_ACS = 0,
    kCoord_MCS = 1,
    kCoord_PCS = 2
} CoordSystem;
```

成員

- kCoord_ACS 軸座標系統，與個別馬達運動有關。
- kCoord_MCS 機器座標系統，藉運動轉換與 ACS 連接。
- kCoord_PCS 產品座標系統，依附在產品或工件上。

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

18.3 MotionBufferMode

定義

相鄰協調運動段之間的緩衝模式。

語法

```
typedef enum {
    kBM_Buffered = 0,
    kBM_Aborting = 1,
    kBM_BlendingLow = 2,
    kBM_BlendingPrevious = 3,
    kBM_BlendingNext = 4,
    kBM_BlendingHigh = 5
} MotionBufferMode;
```

成員

kBM_Buffered	完成當前的運動後，再開始下一個路徑。(預設模式)
kBM_BlendingPrevious	其交接速度為上一個路徑的速度。
其他成員	(不使用)

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

18.4 MotionTransitionMode

定義

相鄰協調運動段之間的過渡模式。

語法

```
typedef enum {
    kTM_NONE = 0,
    kTM_StartVelocity = 1,
    kTM_ConstantVelocity = 2,
    kTM_CornerDistance = 3,
    kTM_MaxCornerDeviation = 4,
    kTM_PLCOpenReserved_05 = 5,
    kTM_PLCOpenReserved_06 = 6,
    kTM_PLCOpenReserved_07 = 7,
    kTM_PLCOpenReserved_08 = 8,
    kTM_PLCOpenReserved_09 = 9
} MotionTransitionMode;
```

成員

kTM_None 不插入過渡曲線。(預設模式)

其他成員 (不使用)

需求版本與檔案

最低支援版本	iA Studio 0.23.2170.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll

18.5 ShaperMode

定義

輸入修正法 (InShape) 的濾波器模式。

語法

```
typedef enum {
    Shaper_Normal = 0,
    Shaper_Robust
} ShaperMode;
```

成員

Shaper_Normal 普通的輸入修正法。

Shaper_Robust 強化的輸入修正法。

需求版本與檔案

最低支援版本	iA Studio 1.1.3761.0
Header	HIMC_API.h
Library	HIMC_API.lib
DLL	HIMC_API.dll