EtherCAT Master Software Manual

English Ver. 1.0.0, OCT. 2018

WARRANTY

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

WARNING

ICP DAS assumes no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

COPYRIGHT

TRADEMARK

Names are used for identification only and may be registered trademarks of their respective companies.

CONTACT US

If you have any questions, please feel free to contact us via email at:

service@icpdas.com; **service.icpdas@gmail.com**

SUPPORT

ECAT-M801-8AX
ECAT-M801-16AX
ECAT-M801-32AX
ECAT-M801-8AX/S
ECAT-M801-16AX/S
ECAT-M801-32AX/S

Contens

1. Introduction 10
2. Software Installation 11
2.1. Obtaining the Driver Installer Package 11
2.2. Driver Installing Procedure 12
2.3. Uninstalling the Driver 19
2.4. Driver Installing Manually 21
2.4.1. Step1 21
2.4.2. Step2 24
2.4.3. Step3 25
3. EcatUtility 29
3.1. Device Operation Toolbar 30
3.1.1. Device Initialization Steps 30
3.1.2. Network Information Edit Steps 31
3.1.3. Start/Stop the EtherCAT Operation Task Steps 33
3.2. Message Panel 34
3.3. Device Network Status 34
3.4. Slave Operation Page 35
3.4.1. Basic Slave Operation Steps 36
3.4.2. Slave PDO and DI/DO LED Operation Steps 36
3.5. Motion Control Initialization Toolbar 37
3.5.1. Motion Control Parameter File Editing Steps 38
3.5.2. Motion Control Initialization Steps 41
3.6. Motion Control Page 42
3.6.1. Single-Axis Motion Control Page 42
3.6.2. Group Motion Control Page 46
3.6.3. Show Position Page 50
3.6.4. 3D Show Position Page 52
3.7. Device I/O Operation Page 54
3.7.1. Device DO control operation step 55
3.7.2. Device DI control operation step 55
3.8. PID Control Page 56
3.8.1. PID Control Page 56
4. Function Overview 59
4.1. Device Operation Flow 59
4.2. Slave Operation Flow 61

4.3. Motion Control Flow 62
4.3.1. Motion Control Initialization 62
4.3.2. Axis Motion Control 64
4.3.3. Axis Homing 66
4.3.4. Axis Error Process 68
4.3.5. Group Moving 70
4.4. Use motion Library in Windows 72
4.4.1. For Visual Studio 72
5. Device Operation Functions 74
5.1. ECAT_GetDeviceCnt 74
5.2. ECAT_OpenDevice 76
5.3. ECAT_CloseDevice 78
5.4. ECAT_GetDeviceSerialNo 80
5.5. ECAT_GetDeviceDI 82
5.6. ECAT_GetDeviceDIBit 84
5.7. ECAT_GetDeviceDO 86
5.8. ECAT_GetDeviceDOBit 88
5.9. ECAT_SetDeviceDO 90
5.10. ECAT_SetDeviceDOBit 92
5.11. ECAT_SetDeviceEncProperty 94
5.12. ECAT_GetDeviceEncProperty 96
5.13. ECAT_GetDeviceEncCount 98
5.14. ECAT_ResetDeviceEncCount 100
5.15. ECAT_SetDeviceCmpTrigProperty 102
5.16. ECAT_GetDeviceCmpTrigProperty 104
5.17. ECAT_SetDeviceCmpTrigData 106
5.18. ECAT_SetDeviceContCmpTrigData 108
5.19. ECAT_SetDeviceEmg 110
5.20. ECAT_GetDeviceEmg 112
5.21. ECAT_GetDeviceEmgStatus 114
5.22. ECAT_SetDeviceEmgSoftSig 116
5.23. ECAT_SetDeviceMPG 118
5.24. ECAT_GetDeviceMPG 121
5.25. ECAT_GetDeviceState 123
5.26. ECAT_StartDeviceOpTask 126
5.27. ECAT_StopDeviceOpTask 129
5.28. ECAT_SetTimer 131
5.29. ECAT_ SetTimerStop 133

5.30. ECAT_WaitforTimer ... 135
5.31. ECAT_GetProcessTime ... 137
6. Slave Operation Functions ... 139
6.1. ECAT_GetSlaveInfo ... 139
6.2. ECAT_GetSlaveSdoObject ... 142
6.3. ECAT_SetSlaveSdoObject ... 144
6.4. ECAT_SetSlaveRxPdoData ... 146
6.5. ECAT_GetSlaveRxPdoData ... 148
6.6. ECAT_GetSlaveTxPdoData ... 150
6.7. ECAT_GetSlaveDI ... 152
6.8. ECAT_GetSlaveDIBit ... 154
6.9. ECAT_GetSlaveDO ... 156
6.10. ECAT_GetSlaveDOBit ... 158
6.11. ECAT_SetSlaveDO ... 160
6.12. ECAT_SetSlaveDOBit ... 162
6.13. ECAT_SetSlaveAoProperty ... 164
6.14. ECAT_GetSlaveAoProperty ... 166
6.15. ECAT_SetSlaveAoRawData ... 168
6.16. ECAT_GetSlaveAoRawData ... 170
6.17. ECAT_SetSlaveAoVoltData ... 172
6.18. ECAT_GetSlaveAoVoltData ... 174
6.19. ECAT_SetSlaveEncProperty ... 176
6.20. ECAT_GetSlaveEncProperty ... 179
6.21. ECAT_GetSlaveEncCount ... 182
6.22. ECAT_ResetSlaveEncCount ... 184
6.23. ECAT_SetSlaveEncLatch ... 186
6.24. ECAT_GetSlaveEncLatch ... 188
6.25. ECAT_GetSlaveEncLatchCnt ... 190
7. Motion Control Functions ... 192
7.1. Motion Control Initialization ... 192
7.1.1. ECAT_McInit ... 192
7.2. Axis Parameter Settings ... 195
7.2.1. ECAT_McSetAxisServoOn ... 195
7.2.2. ECAT_McSetAxisPPU ... 197
7.2.3. ECAT_McGetAxisPPU ... 199
7.2.4. ECAT_McSetProfileData ... 201
7.2.5. ECAT_McGetProfileData ... 203
7.2.6. ECAT_McSetProfileCSV ... 205

7.2.7. ECAT_McGetProfileCSV 208
7.2.8. ECAT_McSetAxisAccTime 211
7.2.9. ECAT_McGetAxisAccTime 213
7.2.10. ECAT_McSetAxisAccDecType 215
7.2.11. ECAT_McGetAxisAccDecType 217
7.2.12. ECAT_McSetAxisEncoderPPR 219
7.2.13. ECAT_McGetAxisEncoderPPR 221
7.2.14. ECAT_McSetAxisMotorPPR 223
7.2.15. ECAT_McGetAxisMotorPPR 225
7.2.16. ECAT_McSetEcamTable 227
7.2.17. ECAT_McGetEcamTable 229
7.2.18. ECAT_McConfigEcamTable 231
7.2.19. ECAT_McSetAxisTouchProbeProperty 233
7.2.20. ECAT_McGetAxisTouchProbeProperty 235
7.2.21. ECAT_McGetAxisTouchProbeValue 237
7.2.22. ECAT_McSetAxisVelocityFeedForwardGain 239
7.2.23. ECAT_McGetAxisVelocityFeedForwardGain 241
7.2.24. ECAT_McSetAxisPosSoftwareLimit 243
7.2.25. ECAT_McGetAxisPosSoftwareLimit 245
7.2.26. ECAT_McSetAxisPosSoftwareLimitStatus 247
7.2.27. ECAT_McGetAxisPosSoftwareLimitStatus 250
7.2.28. ECAT_OpenMotionConfig 252
7.3. Axis Status 255
7.3.1. ECAT_McGetAxisActualPos 255
7.3.2. ECAT_McGetAxisCommandPos 258
7.3.3. ECAT_McGetAxisActualVel 261
7.3.4. ECAT_McGetAxisActualPosVel 264
7.3.5. ECAT_McGetAxisState 267
7.3.6. ECAT_McGetAxisLastError 270
7.3.7. ECAT_McGetAxisDriveError 272
7.3.8. ECAT_McGetAxisDI 275
7.3.9. ECAT_McGetAxisHomeState 278
7.4. Axis Homing 280
7.4.1. ECAT_McSetAxisHomeMethod 280
7.4.2. ECAT_McGetAxisHomeMethod 282
7.4.3. ECAT_McSetAxisHomeSpeed 284
7.4.4. ECAT_McGetAxisHomeSpeed 286
7.4.5. ECAT_McSetAxisHomeAcc 288

7.4.6. ECAT_McGetAxisHomeAcc 290
7.4.7. ECAT_McSetAxisHomeOffset 292
7.4.8. ECAT_McGetAxisHomeOffset 294
7.4.9. ECAT_McAxisHome 296
7.5. Axis Moving 299
7.5.1. ECAT_McAxisErrorReset 299
7.5.2. ECAT_McAxisMoveAbs 301
7.5.3. ECAT_McAxisMoveRel 304
7.5.4. ECAT_McAxisChangePos 307
7.5.5. ECAT_McAxisChangeVel 310
7.5.6. ECAT_McAxisMoveVel 313
7.5.7. ECAT_McAxisMoveVelByPos 315
7.5.8. ECAT_McAxisGearIn 317
7.5.9. ECAT_McAxisGearOut 320
7.5.10. ECAT_McAxisMoveProfile 322
7.5.11. ECAT_McAxisMoveProfileCSV 324
7.5.12. ECAT_McAxisCamIn 326
7.5.13. ECAT_McAxisCamPhaseShift 330
7.5.14. ECAT_McAxisCamOut 335
7.5.15. ECAT_McAxisTangentInGroup 337
7.5.16. ECAT_McAxisGantryIn 347
7.5.17. ECAT_McAxisGantryMaxPosDiff 350
7.5.18. ECAT_McAxisGantryMaxPosDiffStatus 353
7.5.19. ECAT_McAxisStop 356
7.5.20. ECAT_McAxisQuickStop 359
7.6. Group Parameter Setting 362
7.6.1. ECAT_McAddAxisToGroup 362
7.6.2. ECAT_McRemoveAxisFromGroup 364
7.6.3. ECAT_McUngroupAllAxes 366
7.6.4. ECAT_McSetGroupCmdMode 368
7.6.5. ECAT_McSetGroupCmdModeEx 370
7.6.6. ECAT_McGetGroupCmdMode 373
7.6.7. ECAT_McSetGroupAccTime 375
7.6.8. ECAT_McGetGroupAccTime 377
7.6.9. ECAT_McSetGroupAccDecType 379
7.6.10. ECAT_McGetGroupAccDecType 381
7.7. Group Status 383
7.7.1. ECAT_McGetGroupState 383

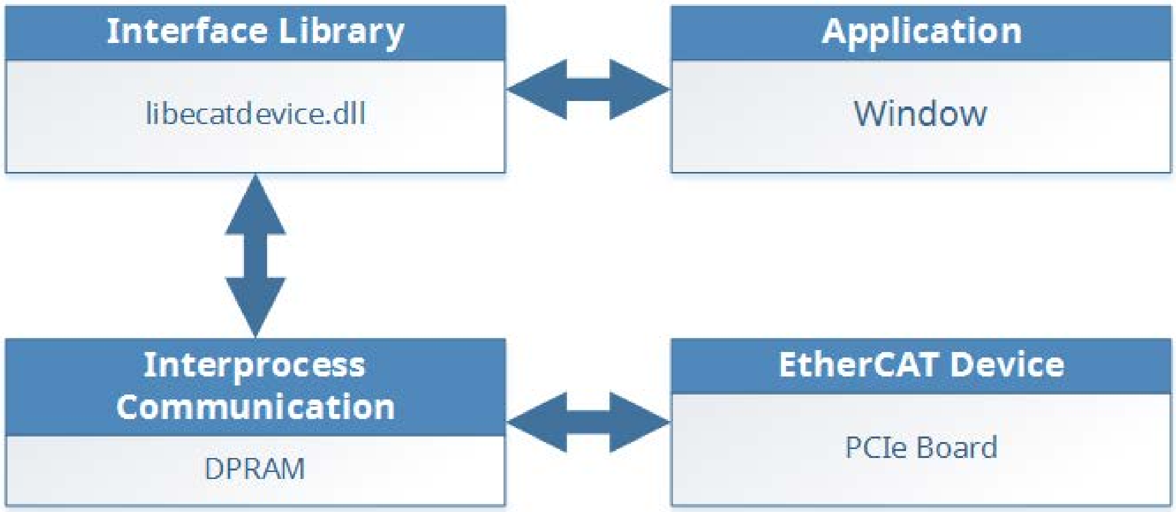
7.7.2. ECAT_McGetGroupCmdBuffer 386
7.7.3. ECAT_McSetGroupVelLimitStatus 388
7.7.4. ECAT_McGetGroupVelLimitStatus 390
7.7.5. ECAT_McSetGroupVelLimitValue 392
7.7.6. ECAT_McGetGroupVelLimitValue 394
7.8. Group Moving 396
7.8.1. ECAT_McGroupMoveLineAbs 396
7.8.2. ECAT_McGroupMoveLineRel 400
7.8.3. ECAT_McGroupMoveLineAbs_PT 404
7.8.4. ECAT_McGroupMoveLineRel_PT 407
7.8.5. ECAT_McGroupMoveLineAbs_PVT 410
7.8.6. ECAT_McGroupMoveLineRel_PVT 414
7.8.7. ECAT_McGroupMoveCircularAbs_CP_Angle 418
7.8.8. ECAT_McGroupMoveCircularRel_CP_Angle 422
7.8.9. ECAT_McGroupMoveCircularAbs_CP_EP 425
7.8.10. ECAT_McGroupMoveCircularRel_CP_EP 429
7.8.11. ECAT_McGroupMoveCircularAbs_BP_EP 432
7.8.12. ECAT_McGroupMoveCircularRel_BP_EP 436
7.8.13. ECAT_McGroupMove3DCircularAbs_CP_Angle 439
7.8.14. ECAT_McGroupMove3DCircularRel_CP_Angle 443
7.8.15. ECAT_McGroupMove3DCircularAbs_CP_EP 446
7.8.16. ECAT_McGroupMove3DCircularRel_CP_EP 450
7.8.17. ECAT_McGroupMove3DCircularAbs_BP_EP 453
7.8.18. ECAT_McGroupMove3DCircularRel_BP_EP 457
7.8.19. ECAT_McGroupMoveHelicalAbs 460
7.8.20. ECAT_McGroupMoveHelicalRel 465
7.8.21. ECAT_McGroupMove3DHelicalAbs_CP_Angle 468
7.8.22. ECAT_McGroupMove3DHelicalRel_CP_Angle 473
7.8.23. ECAT_McGroupMoveConicalHelixAbs 476
7.8.24. ECAT_McGroupMoveConicalHelixRel 480
7.8.25. ECAT_McGroupMove3DConicalHelixAbs_CP_Angle 483
7.8.26. ECAT_McGroupMove3DConicalHelixRel_CP_Angle 487
7.8.27. ECAT_McGroupMoveProfile 490
7.8.28. ECAT_McGroupMoveProfileCSV 493
7.8.29. ECAT_McGroupMoveDwell 496
7.8.30. ECAT_McGroupMoveDO 499
7.8.31. ECAT_McGroupMoveAO 503
7.8.32. ECAT_McGroupMoveBlendingSync 507

7.8.33. ECAT_McGroupStop 511
7.8.34. ECAT_McGroupQuickStop 514
7.8.35. ECAT_McSetGroupHold 517
7.9. PID Controller 520
7.9.1. ECAT_PidGetSetPointValue 521
7.9.2. ECAT_PidSetSetPointValue 523
7.9.3. ECAT_PidGetProcessVariable 525
7.9.4. ECAT_PidGetSampleTime 528
7.9.5. ECAT_PidSetSampleTime 530
7.9.6. ECAT_PidGetStatus 532
7.9.7. ECAT_PidSetStatus 534
7.9.8. ECAT_PidGetSimulateMode 536
7.9.9. ECAT_PidSetSimulateMode 538
7.9.10. ECAT_PidGetParameter 540
7.9.11. ECAT_PidSetParameter 542
7.9.12. ECAT_PidGetProcessVariableModule 544
7.9.13. ECAT_PidSetProcessVariableModule 546
7.9.14. ECAT_PidGetControlOutputModule 549
7.9.15. ECAT_PidSetControlOutputModule 551
7.9.16. ECAT_PidGetControlOutputValue 554
7.9.17. ECAT_PidGetSimulateFeedback 557
7.9.18. ECAT_PidGet_Sp_Err_Op_Pv 560
7.10. Stewart Platform 563
7.10.1. ECAT_McSetStewartPlatform_M1 565
7.10.2. ECAT_McSetStewartPlatform_M1 568
7.10.3. ECAT_McSetStewartPlatform_M2 571
7.10.4. ECAT_McGetStewartPlatform_M2 574
7.10.5. ECAT_McStwartPlatformMoveAbs_PT 577
7.11. Motion Data Recorder 581
7.11.1. ECAT_McSetMotionRecord 581
7.11.2. ECAT_McGetMotionRecordState 583
7.11.3. ECAT_McClearMotionRecord 585
7.11.4. ECAT_McSetMotionRecordParam 587
7.11.5. ECAT_McGetMotionRecordParam 589
7.11.6. ECAT_McGetMotionRecordValue 591
8. Appendix 595
8.1. Error Codes 595
8.2. SDO Abort messages 600

8.3. Revision History 602

1. Introduction

For developing applications on EtherCAT Master series cards, ICP DAS provides users with a shared library libecatdevice (.dll) to support the use in Windows operating systems. It provides powerful, easy-to-use functions for developing applications and speed-up the developing process . The library architecture is shown in the following figure. The user programs are developed on PC. PC is cummunicated with ECAT-M801 via APIs which use DRPRM (dual-port RAM) as the bridge.

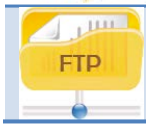


2. Software Installation

This chapter shows where to get and how to install the driver package and utility.

2.1. Obtaining the Driver Installer Package

The driver installer package can be found on the supplied CD-ROM, or can be downloaded from ether ICP DAS FTP or web site. The location and addresses are in the table below:

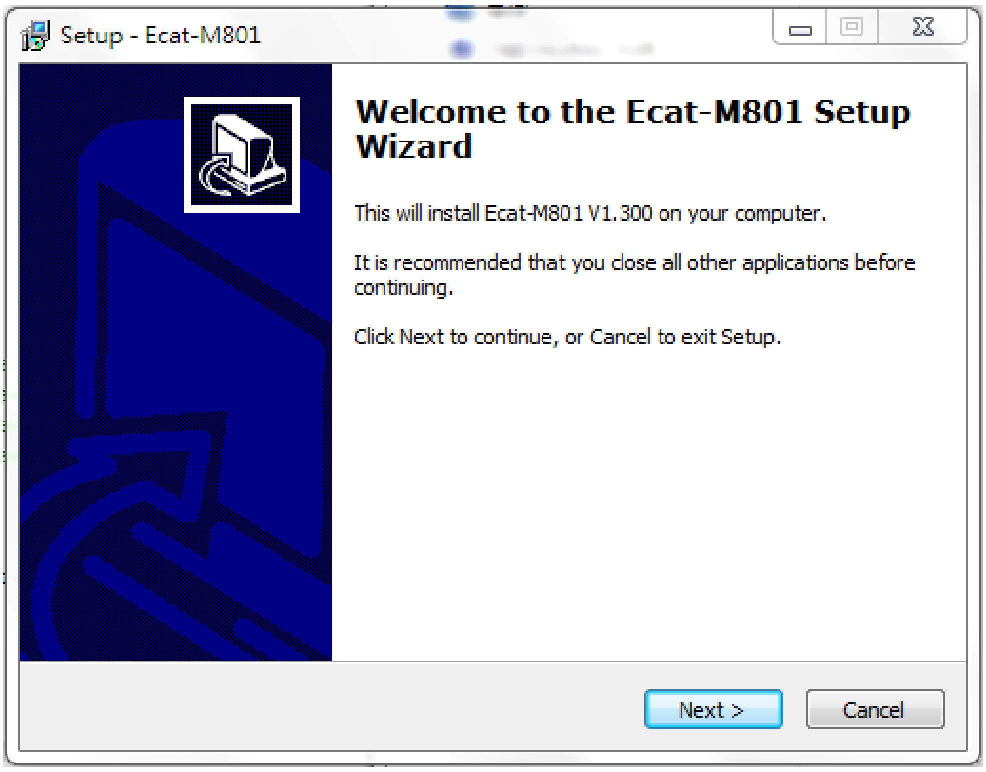
	http://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/master/ecat-m801/manual/
	ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/master/ecat-m801/manual/

2.2. Driver Installing Procedure

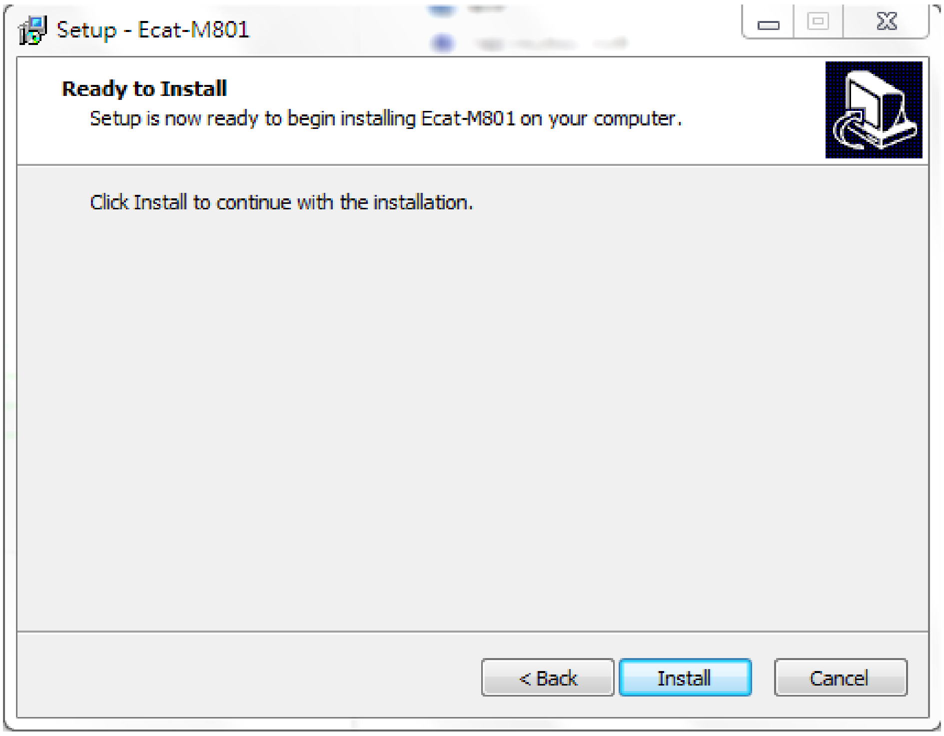
To install drivers, follow the procedure described below:

1. Double-Click "ECAT-M801_vx.xx.xx_setup.exe" to install driver.

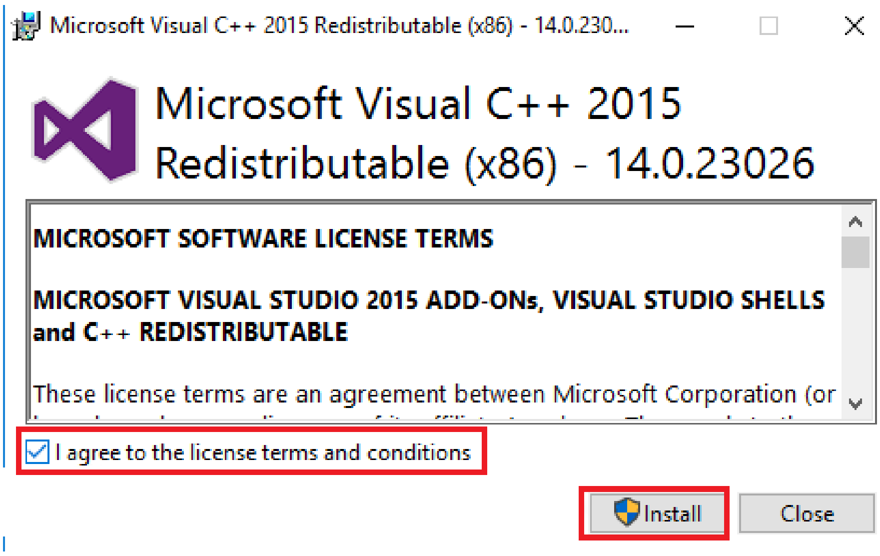
2. Click ther “Next >” button.



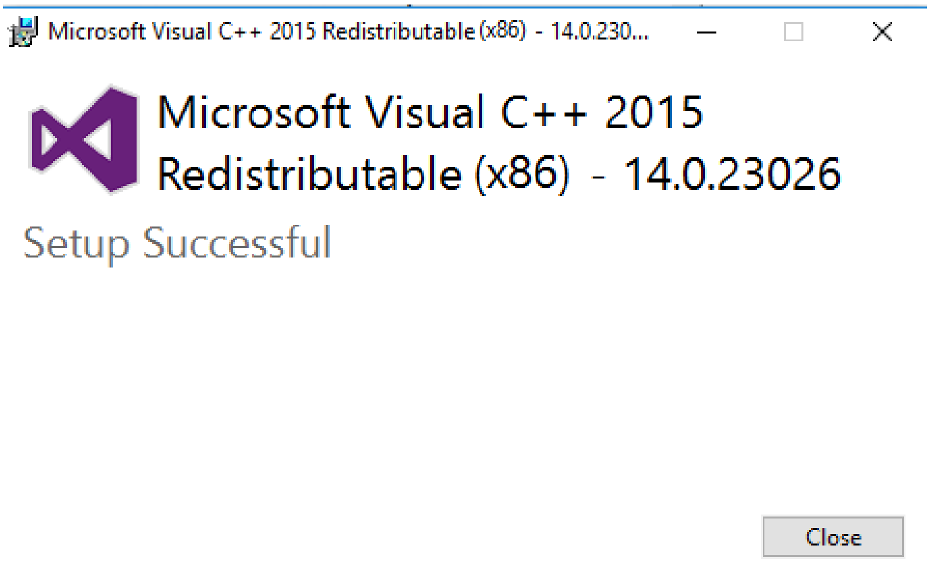
3. Select the installation folder, the default path is C:\icpdas\Ecat-M801, Click the "Install" button to continue.



4. Check "I agree to the license terms and conditions", then click the "Install" button to continue.

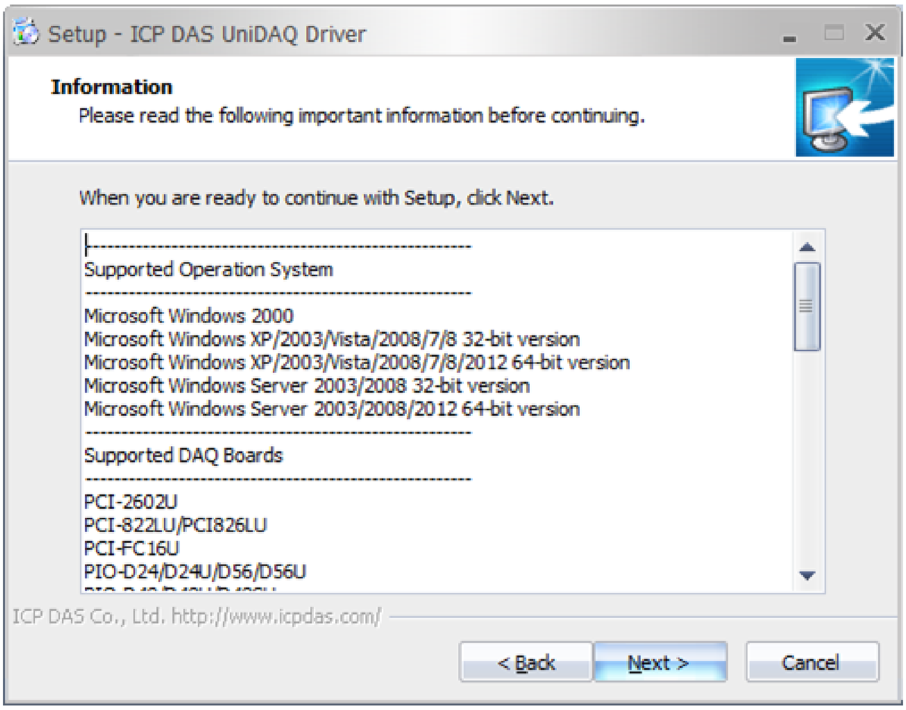


5. Click the “Close” button to continue.

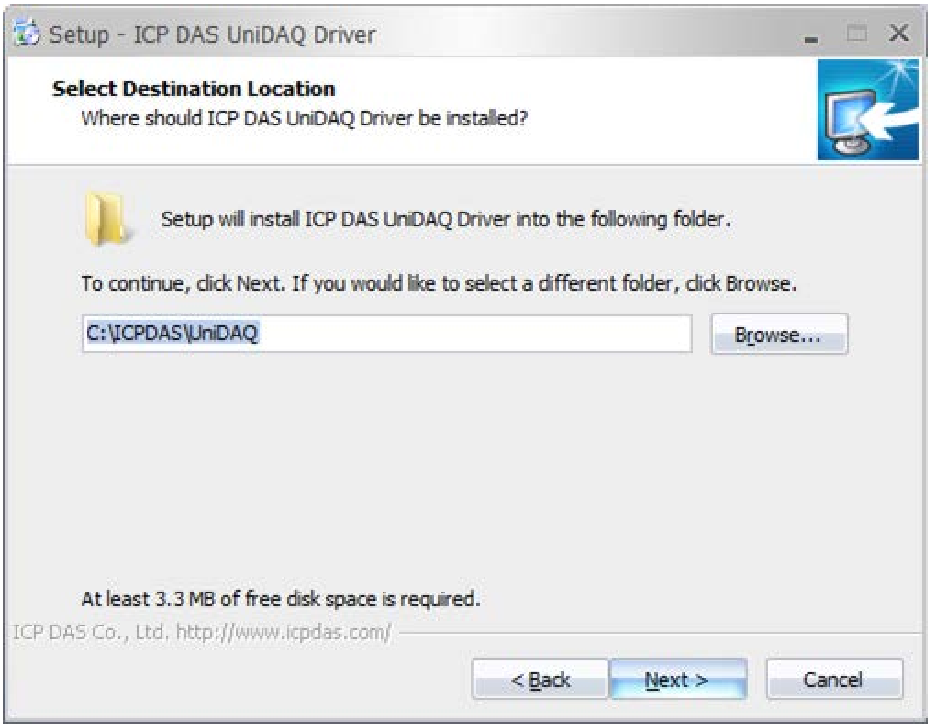


6. Click "Next >"

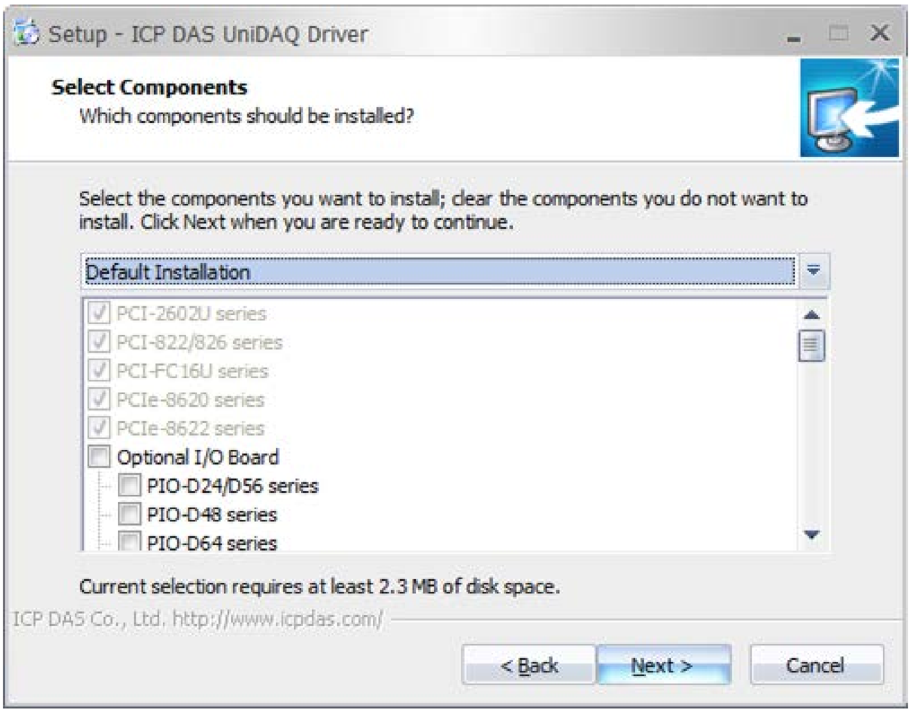
7. Click "Next >"



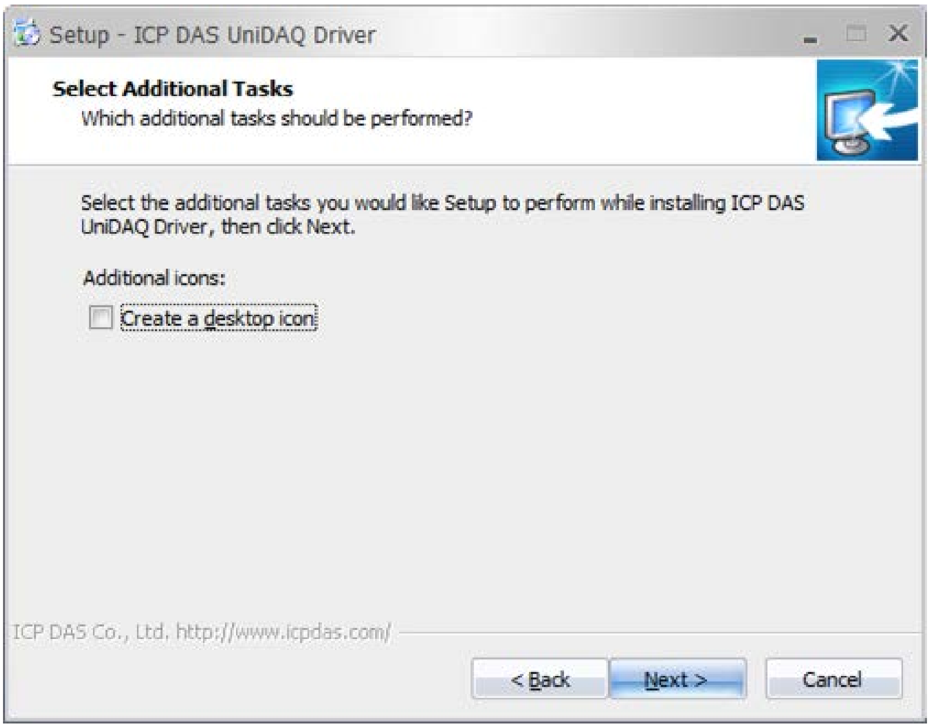
8. Click "Next >"

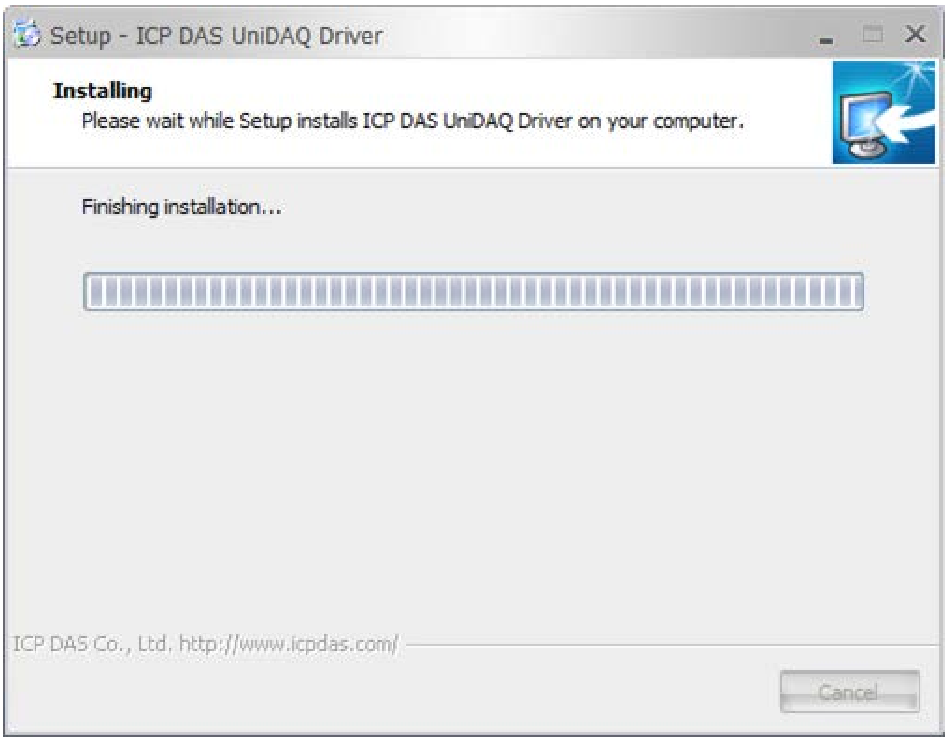


9. Click "Next >"

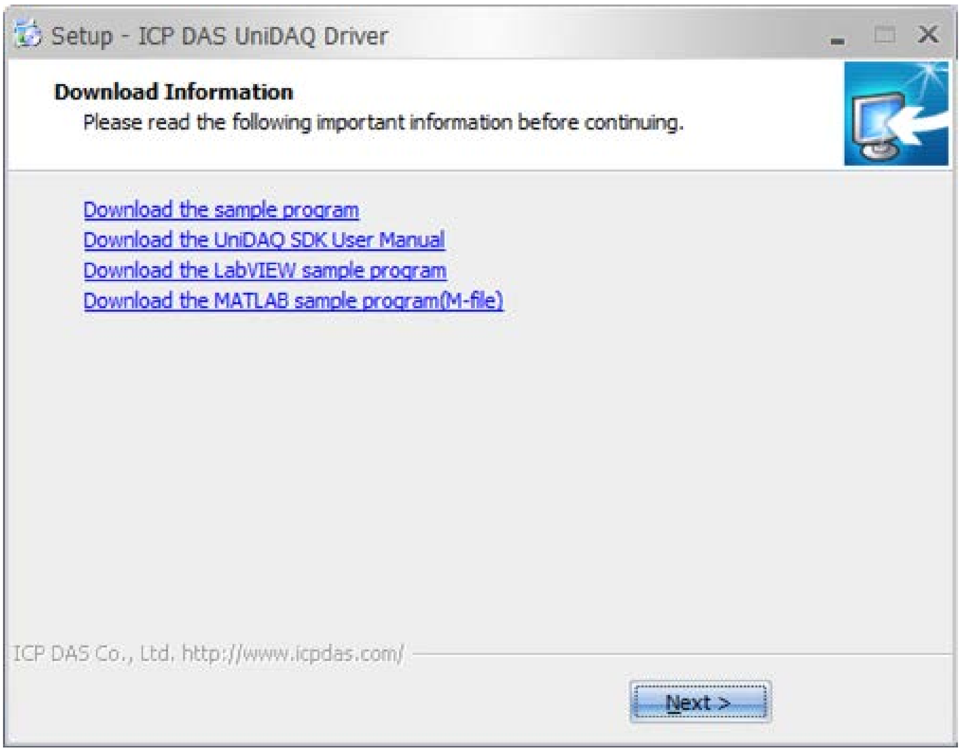


10. Click "Next >"



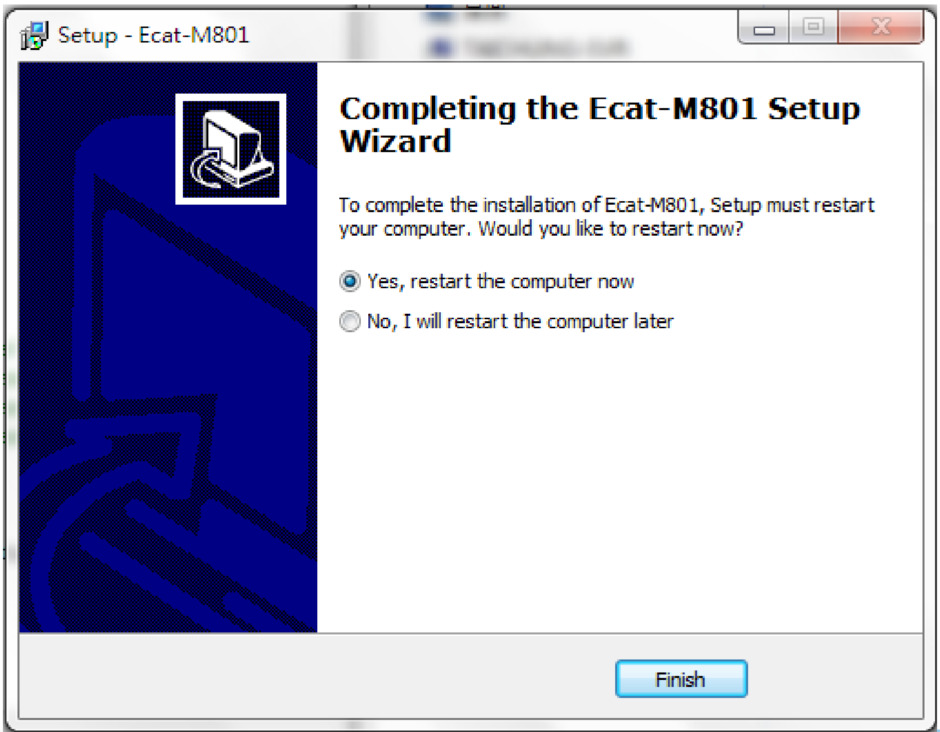


11. Click "Next >"



12. Click "Finish"

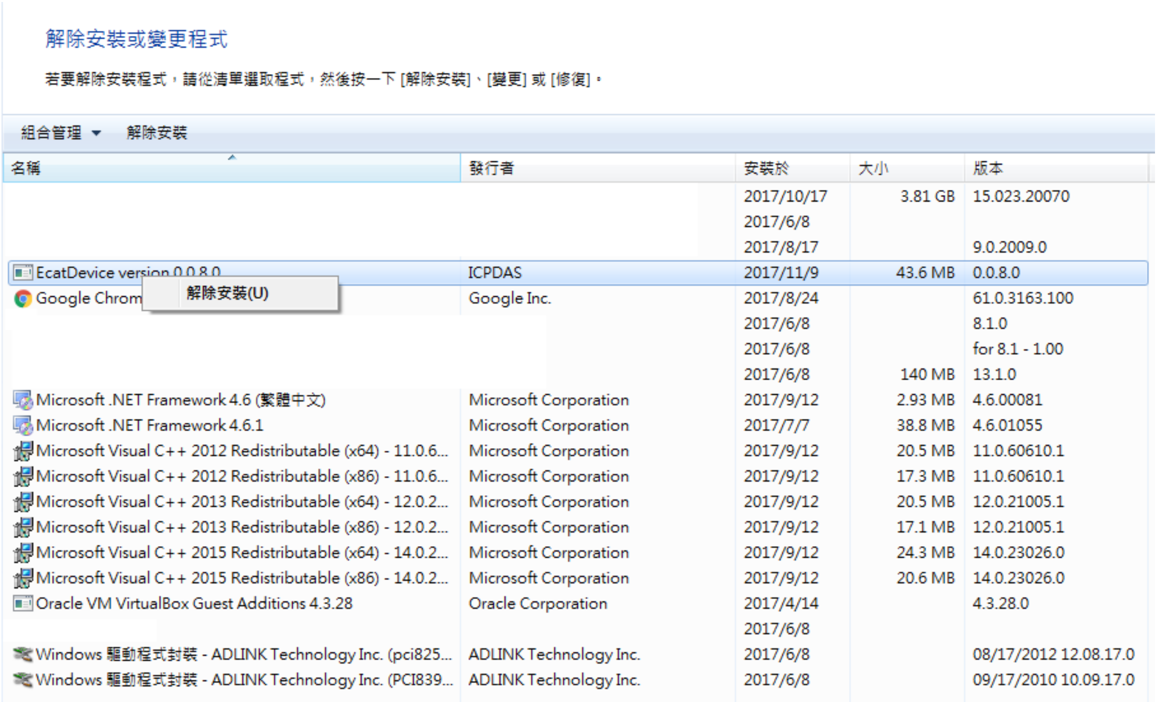
13. Click the “Finish” button and restart the computer.



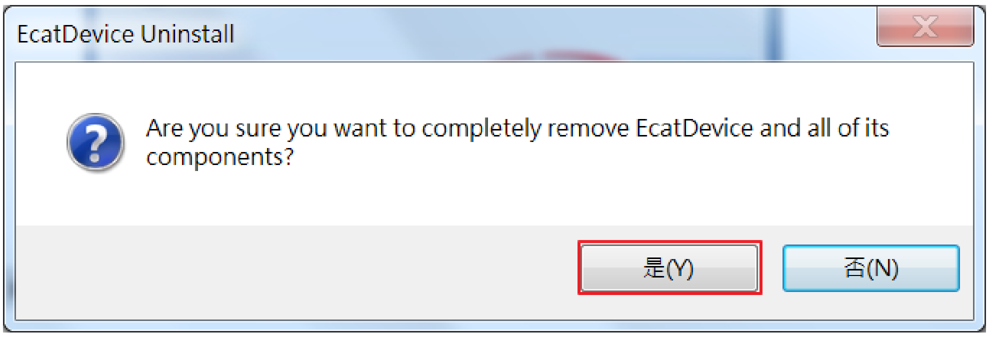
2.3. Uninstalling the Driver

ICPDAS driver includes an uninstall utility to help users remove the software from your computer. To uninstall the software, complete the following procedures:

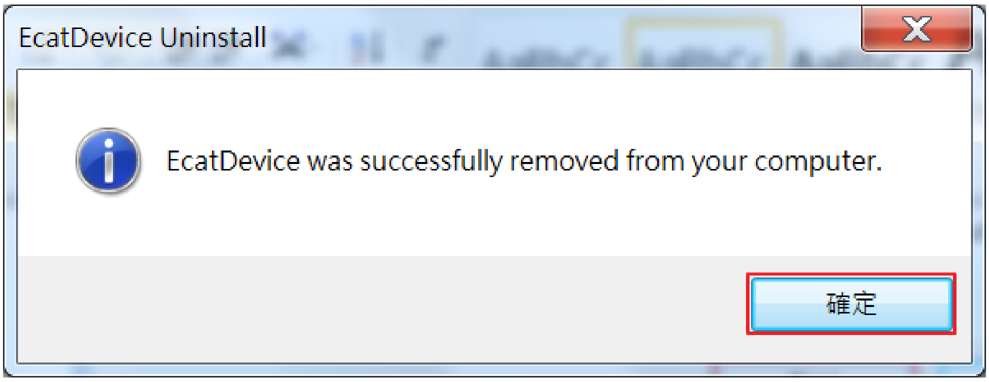
1. Select **Control Panel >> Add/Remove Programs** from the Windows **Start** menu.
2. Click the **Install/Uninstall tab** and highlight the item EcatDevice Windows Driver and then click the remove button.



3. When the message box loads, click the Yes(Y) button to uninstall the software.



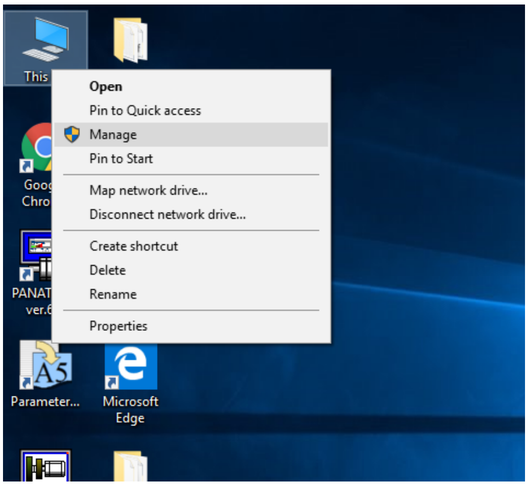
4. After the uninstallation process is complete, a dialog box will be displayed to you that the driver was successfully removed. Click the "OK" button to finish the uninstallation process.



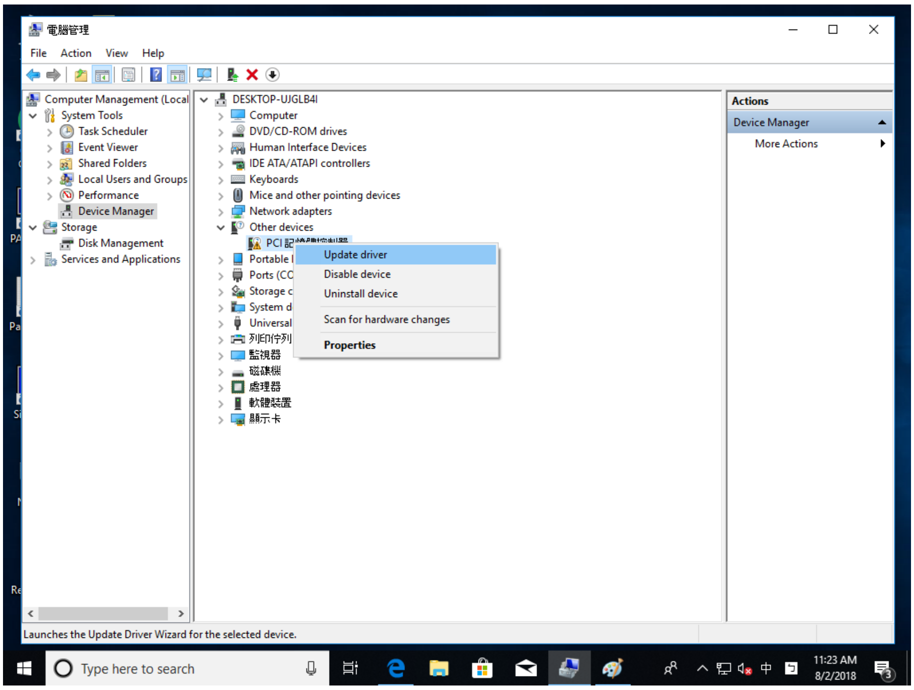
2.4. Driver Installing Manually

2.4.1. Step1

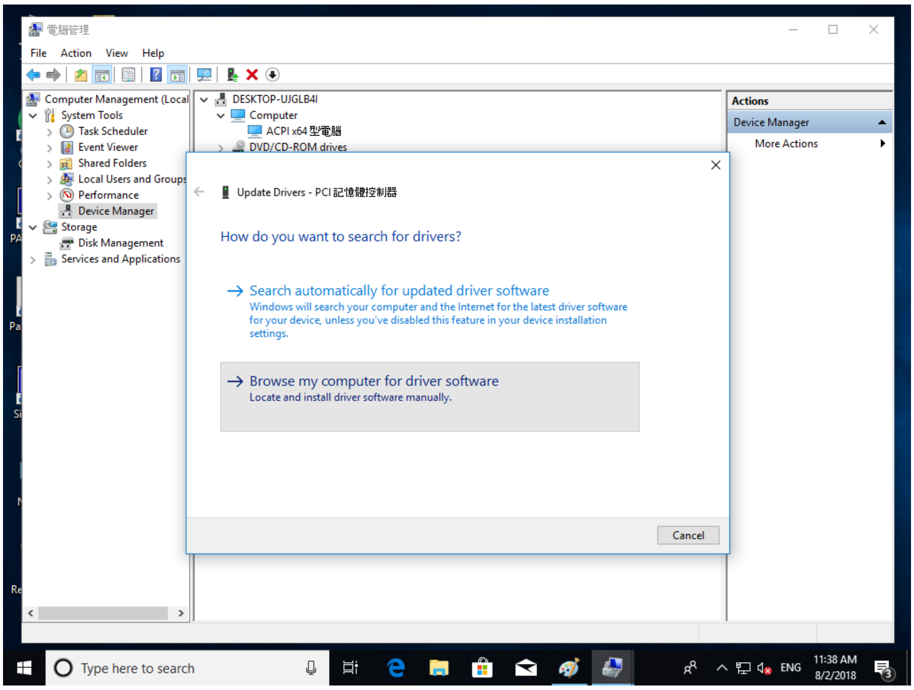
1. Right-click **This PC** (or Computer) and click Manage.



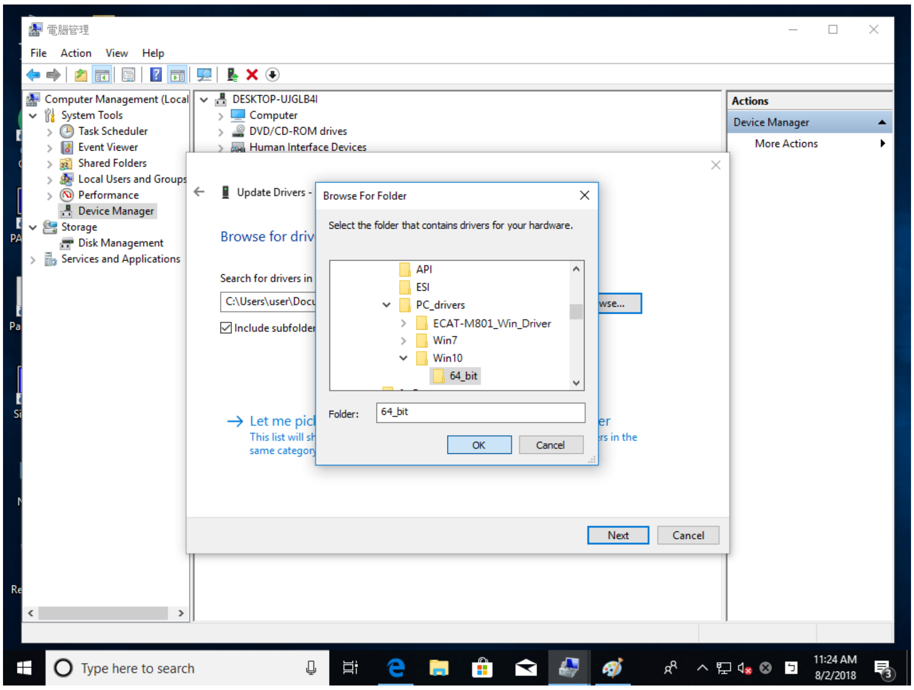
2. In the Computer Management window, on the left-hand side, click Device Manager.
3. Right-click the device, and select Update Driver.



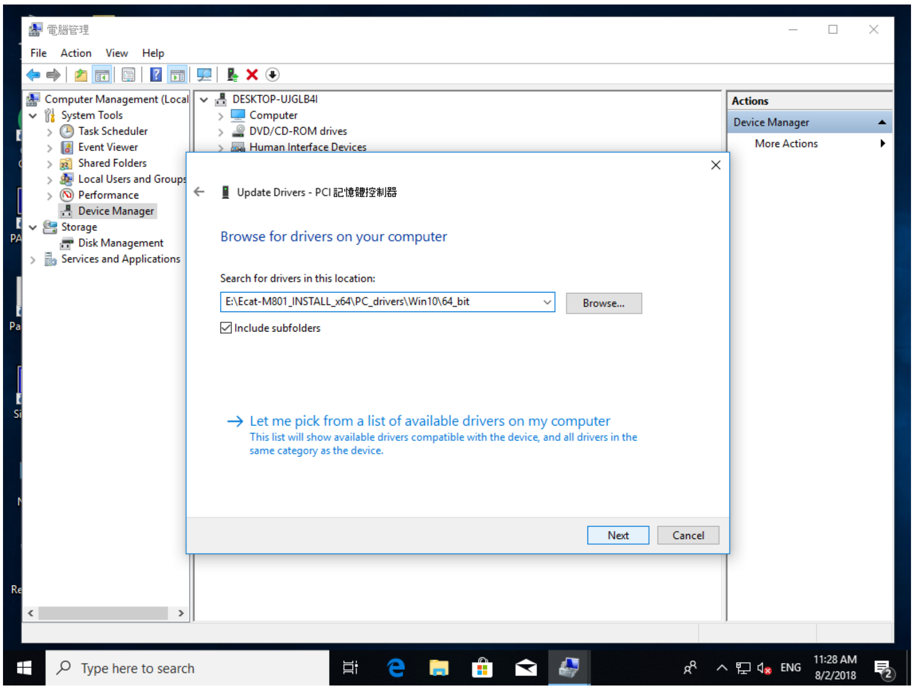
4. In the Update Driver Software window, select Browse my computer for driver software.

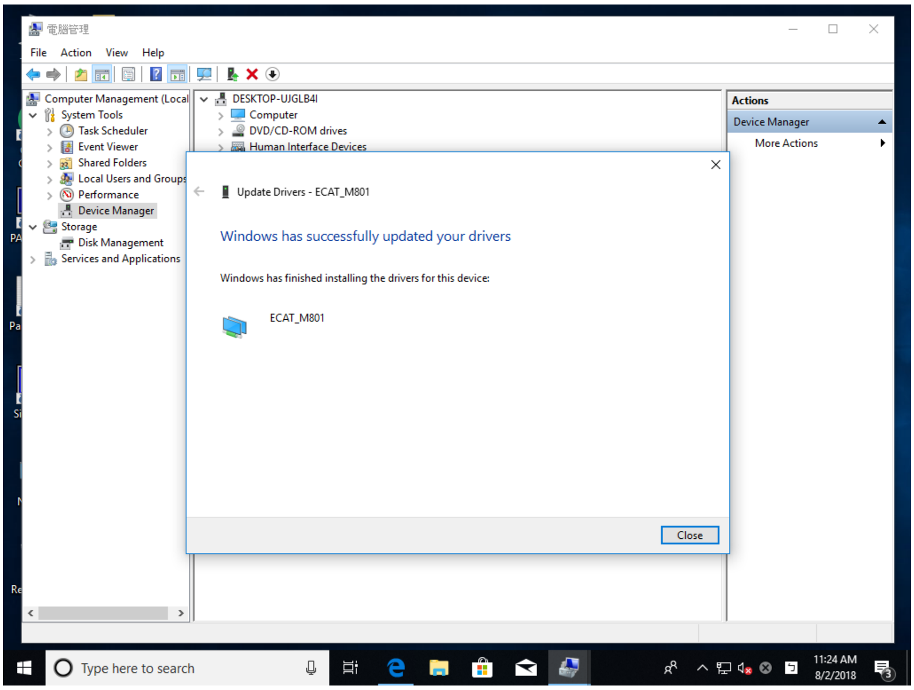


5. Click Browse and navigate to the folder that named 「Ecat-M801_INSTALL_x64\PC_drivers\Win10\64bit」. Click OK when this folder is selected.



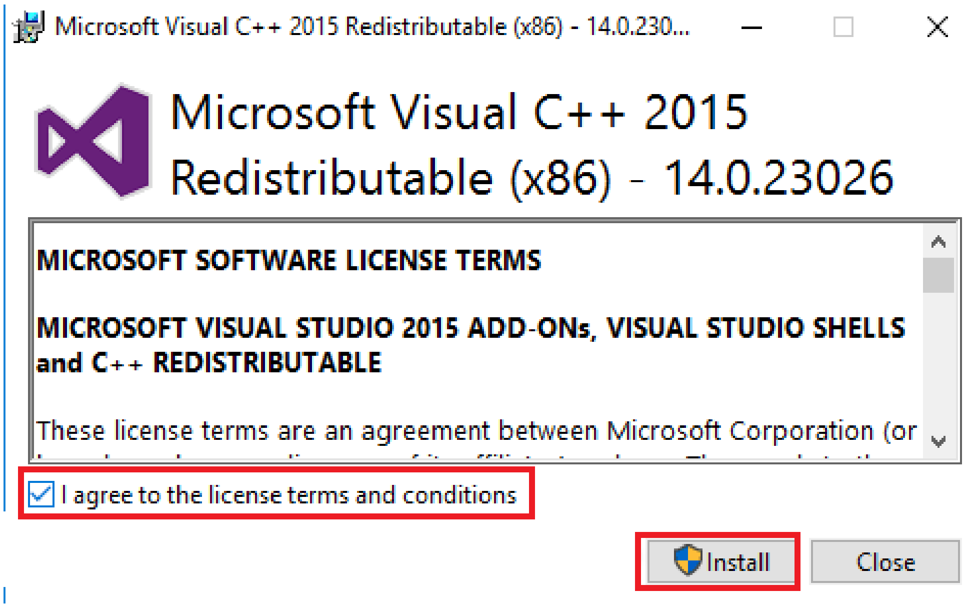
6. Back in the Update Driver Software window, click Next. Windows will search for the driver and install it automatically. When done, click Close.



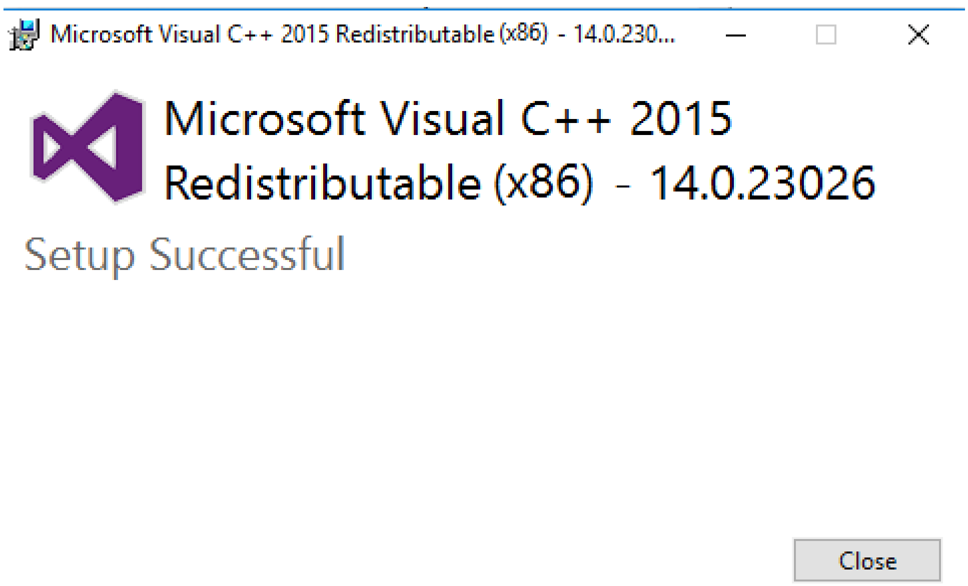


2.4.2. Step2

1. Install vc_redist.x86_2015.exe, the file is in folder which named 「Ecat-M801_INSTALL_x64\PC_drivers」, check I agree to the license terms and conditions, and Click the **Install** to continue.



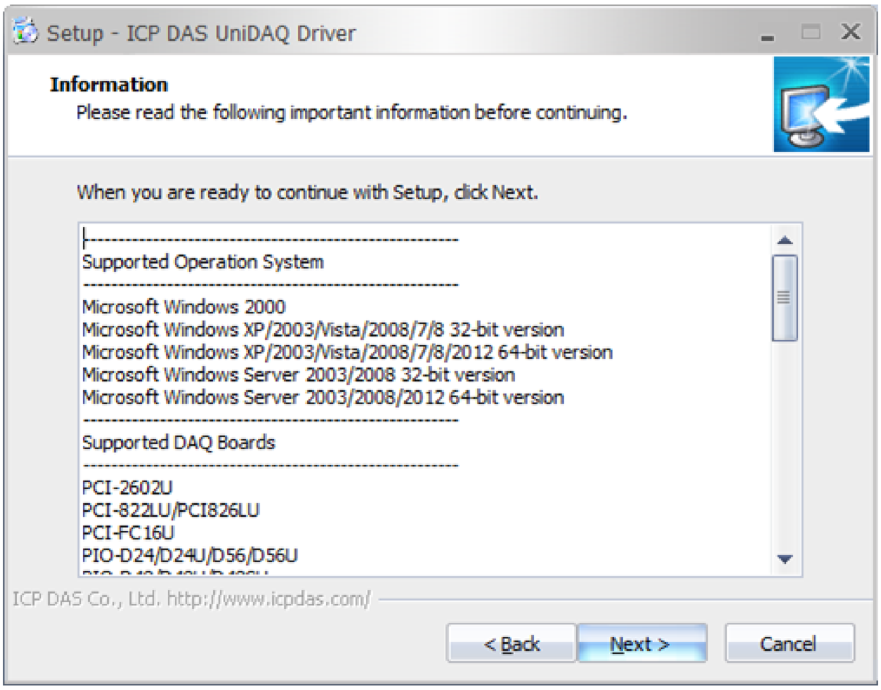
2. Click the **Close** to continue.



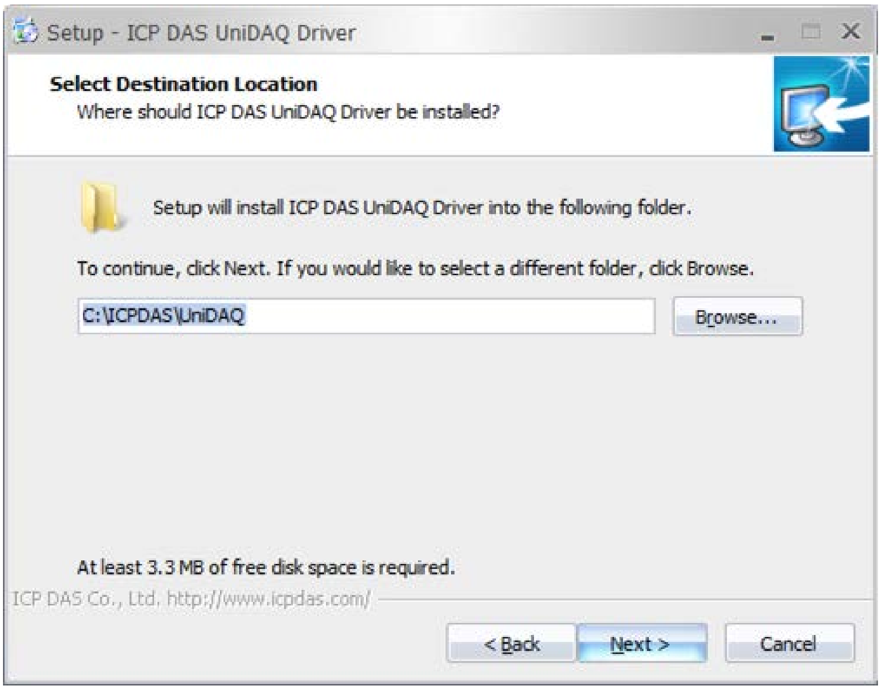
2.4.3. Step3

1. Install UniDAQ_Win_Setup_1.3.2.0_0807.exe, the file is in folder which name 「Ecat-M801_INSTALL_x64\PC_drivers」, click "Next >"

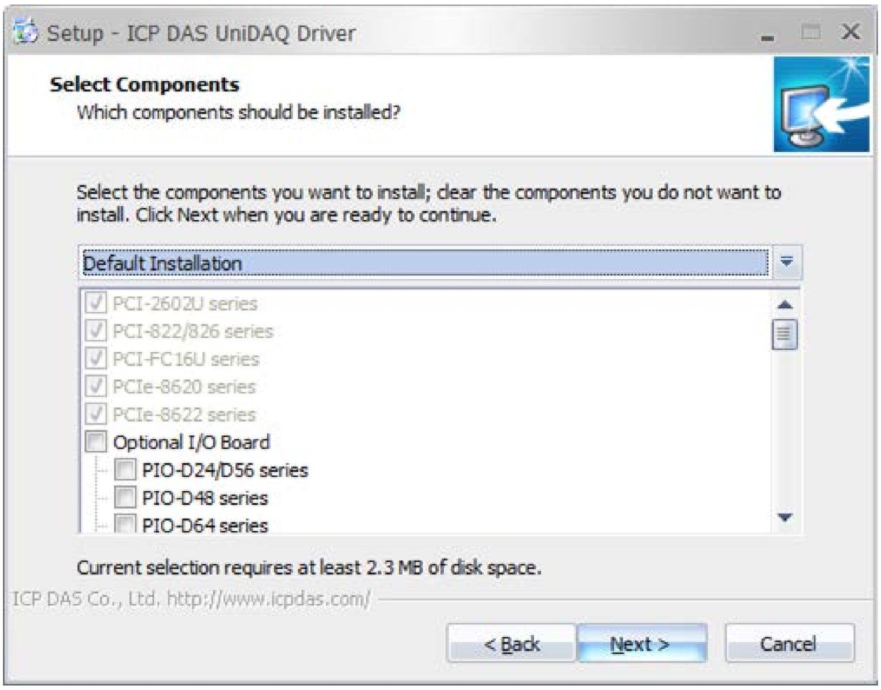
2. Click "Next >"



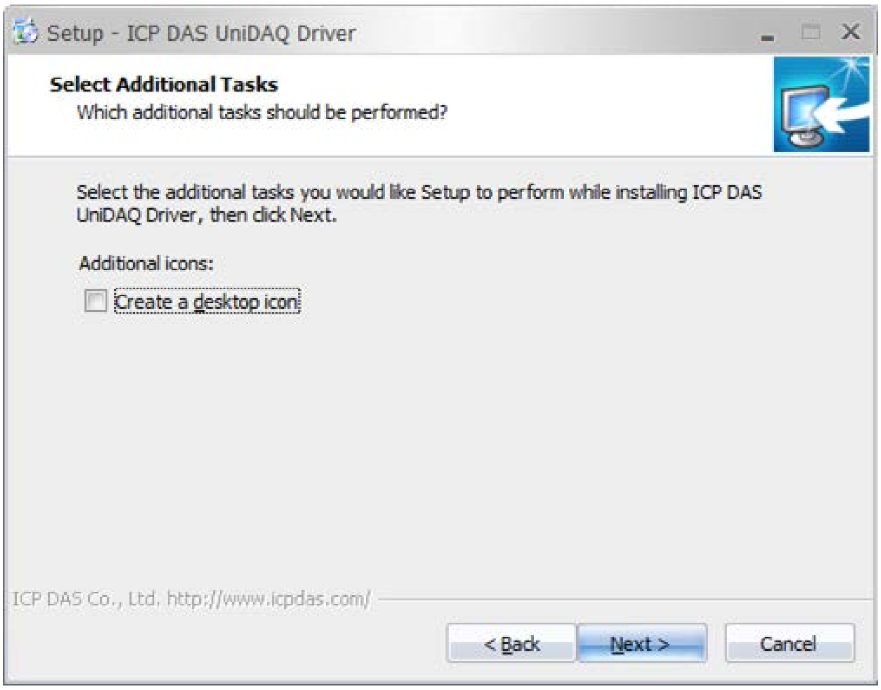
3. Click Next>

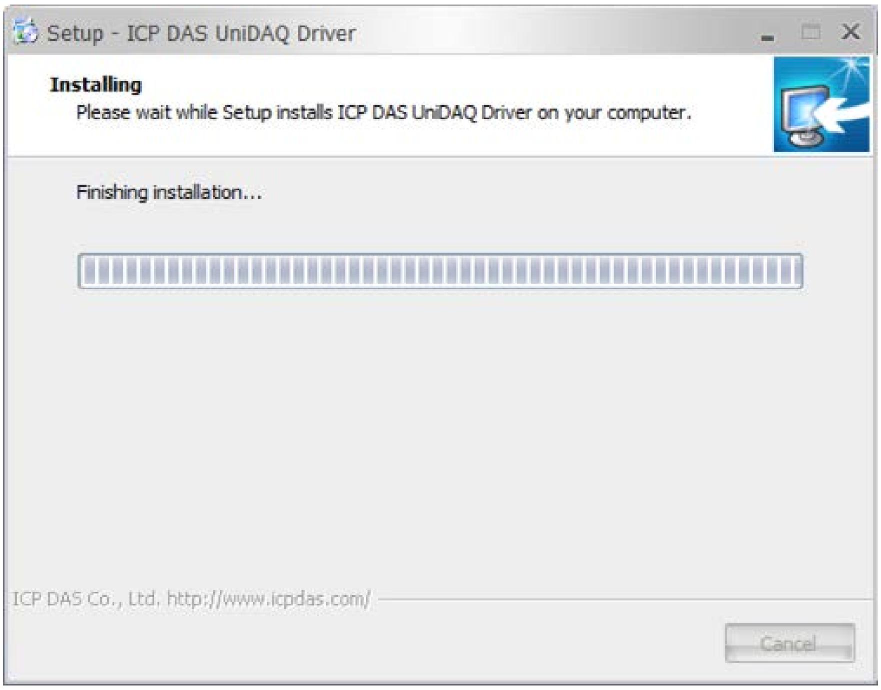


4. Click "Next >"



5. Click "Next >"

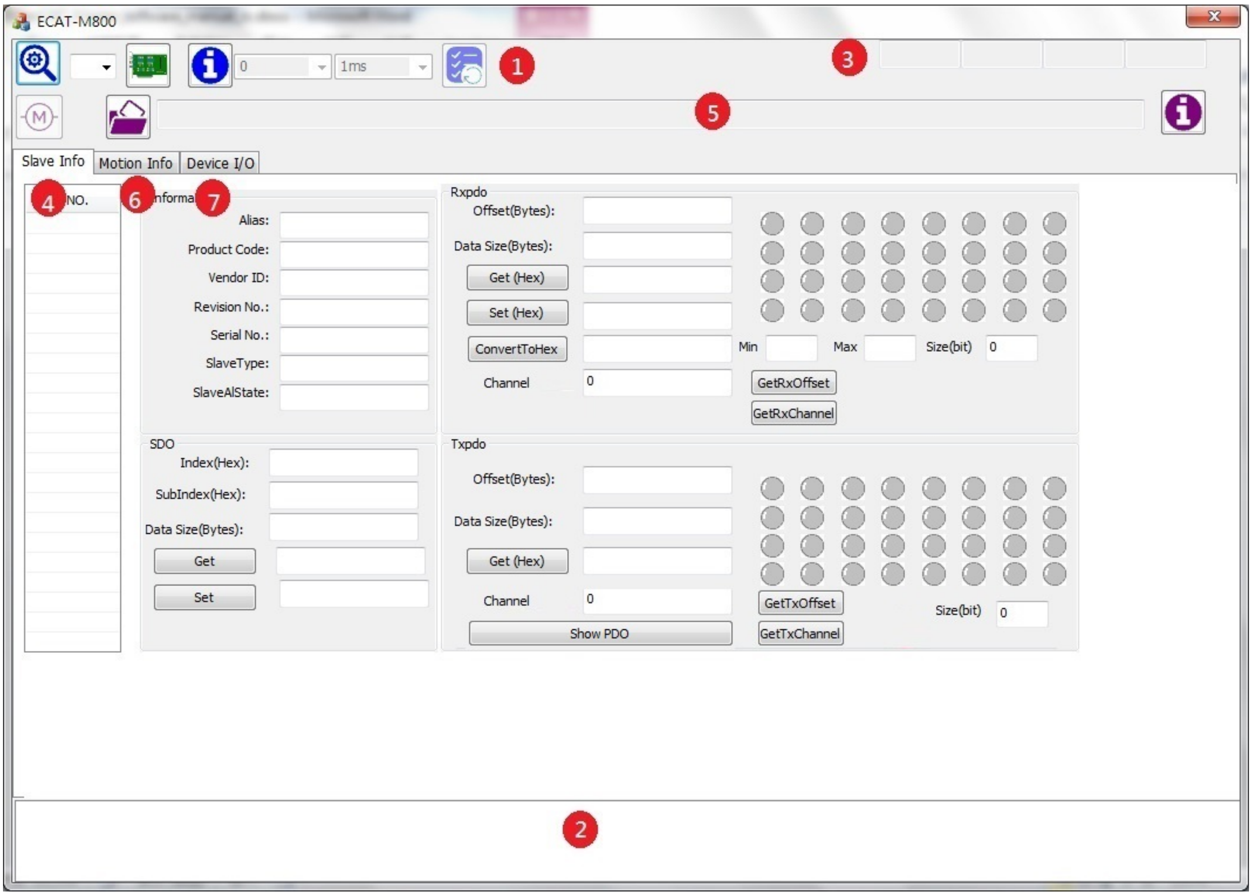

Setup - ICP DAS UniDAQ Driver
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing ICP DAS UniDAQ Driver, then click Next.
Additional icons:
Create a desktop icon
ICP DAS Co., Ltd. http://www.icpdas.com/
< Back
Next >
Cancel


Setup - ICP DAS UniDAQ Driver
Installing
Please wait while Setup installs ICP DAS UniDAQ Driver on your computer.
Finishing installation...
ICP DAS Co., Ltd. http://www.icpdas.com/
Cancel

6. Click "Finish" to finish the installation.

3. EcatUtility

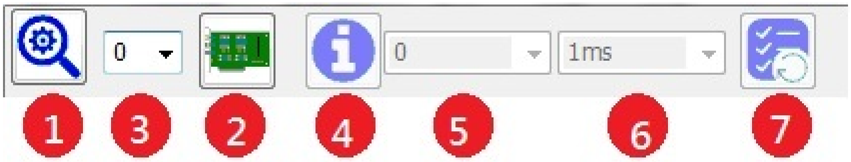
EcatUtility is a software tool for users to use ECAT-M801 card on EtherCAT applications. It allow users to edit the device network information, to test slave modules, and to do motion control function tests. This software tool contains several parts shown in the following figure and table.



Item	Description
(1)	Device operation toolbar
(2)	Message panel
(3)	Device network status
(4)	Slave Operation page
(5)	Toolbar for the initialization of Motion Control
(6)	Motion control page
(7)	Device I/O operation page

3.1. Device Operation Toolbar

The device operation toolbar is show below,and the description of each control item is shown in its following table.



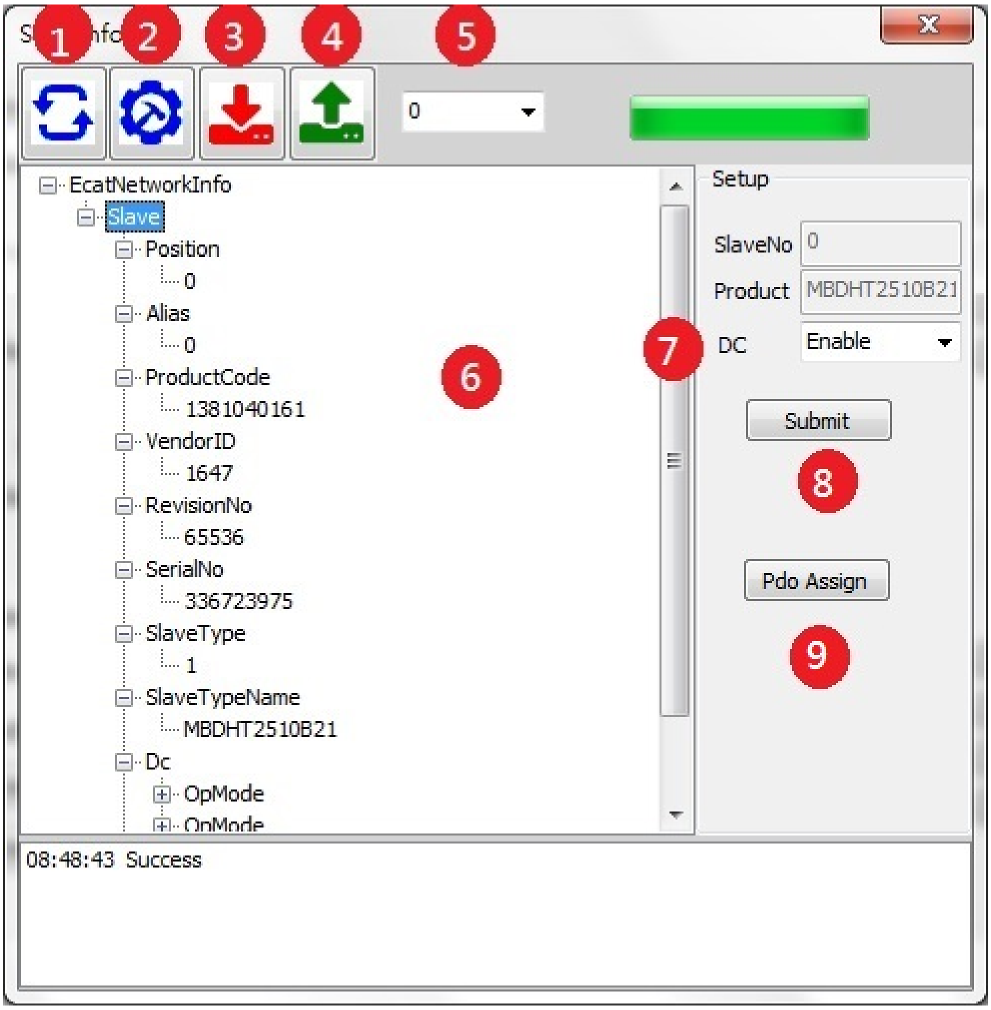
Item	Description
(1)	Get all the number of available devices and shown them in (3)
(2)	Open or Close device which is chosen in (3)
(3) 0	Device number list (shown the chosen card number here)
(4)	Network information edit page (for configuring slaves)
(5) 0	Network information number list. The chosen file number will be used to store network information for downloading into ECAT-M801 card.
(6) 1000000	Cycle time list. Unit is in micro-seconds. The chosen cycle time is used for cyclic communication when EtherCAT system goes into OP state.
(7)	Start or Stop the device EtherCAT operation task

3.1.1. Device Initialization Steps

1. Click to get the number of devices (i.e. control cards) and shown them in (3).
2. Select the device number (i.e. card number) from the device number list, and click to open the specified device communication operation.
3. If is clicked again, it will turn off device communication.

3.1.2. Network Information Edit Steps

Click on the device operation toolbar to enter the network information edit page. The descriptions of control items are shown in the following figure and table.

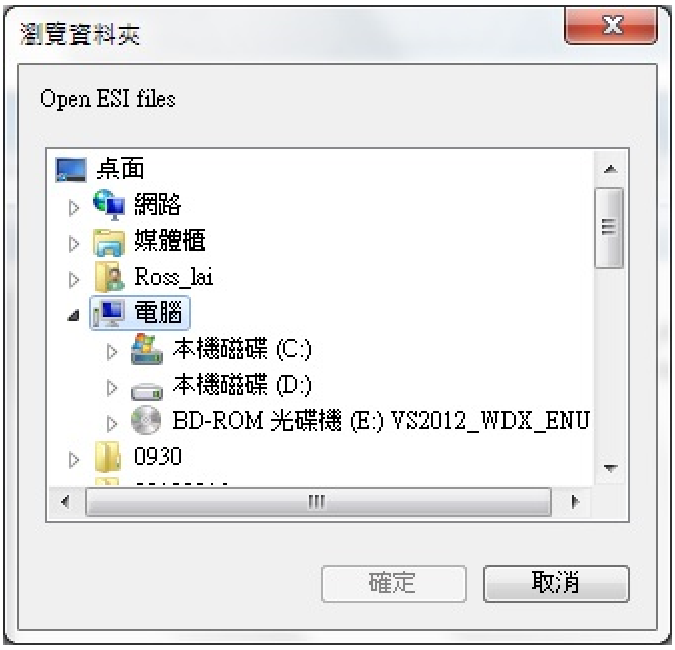


Item	Description
(1)	Scan EtherCAT network for finding slaves.
(2)	Build network information from ESI files
(3)	Send the network information into ECAT-M801 card. The network information file number is chosed in (5).
(4)	Retrieve network information from ECAT-M801 card. The network information file number is chosed in (5).

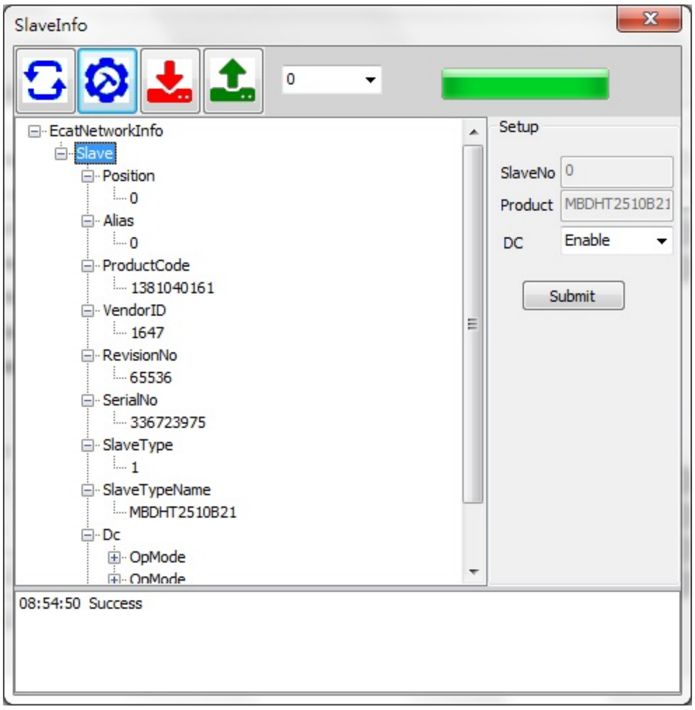
(5) 0	The file number (file name) of network information for sending and retrieving.
(6)	Network information panel
(7) DC Enable	DC setting of the selected slave. If the slave is capable of DC communication and meets the system cycle time setting, it can be set to **Enable**.
(8) Submit	Apply the configuration change of this slave. It must be apply before switching to edit other slaves; otherwize, system does not accept the change.
(9) Pdo Assign	Do PDO assignment. If users are familiar with EtherCAT technology and know how to that, it is for them to chage the default PDO assignment.

Descriptions:

1. Click to scan all the connected slaves.
2. After click , users must choose an ESI File Directory for utility program to search related ESI files.



Then the utility program will retrieve detailed slave informations and show them on the info panel.



3. Click Slave to select a slave. User can select its DC setting on the right-hand side. Click Submit to apply the setting before switching to edit other slaves.
4. After setting all slaves, select a preferred network information number from the network information list and click to send it into the device (ECAT-M801 control card).
5. If needed, a previous configuration file in the device can be retrieved by clicking .
6. Step2 assign default Pdos, Pdo Assign provide user to assign Pdos mapping.

3.1.3. Start/Stop the EtherCAT Operation Task Steps

1. After the user completes the steps of editing network information, he can select a network information number from the device operation toolbar.
2. Choose a suitable communication cycle time in the cycle list.
3. Click to start EtherCAT operation task. If there is no error message appeared, wait for device network status to change to OP. Then, users can start the related EtherCAT operation.
4. If is clicked again, it will stop the EtherCAT operation task.

3.2. Message Panel

As shown below, when any software operation error occurs, the message panel will show the error message, occurred time and error code. To clear all information in this pannel, please move the mouse curser on the Message panel, click the right mouse button, and then choose "Clear" in the right-click menu.

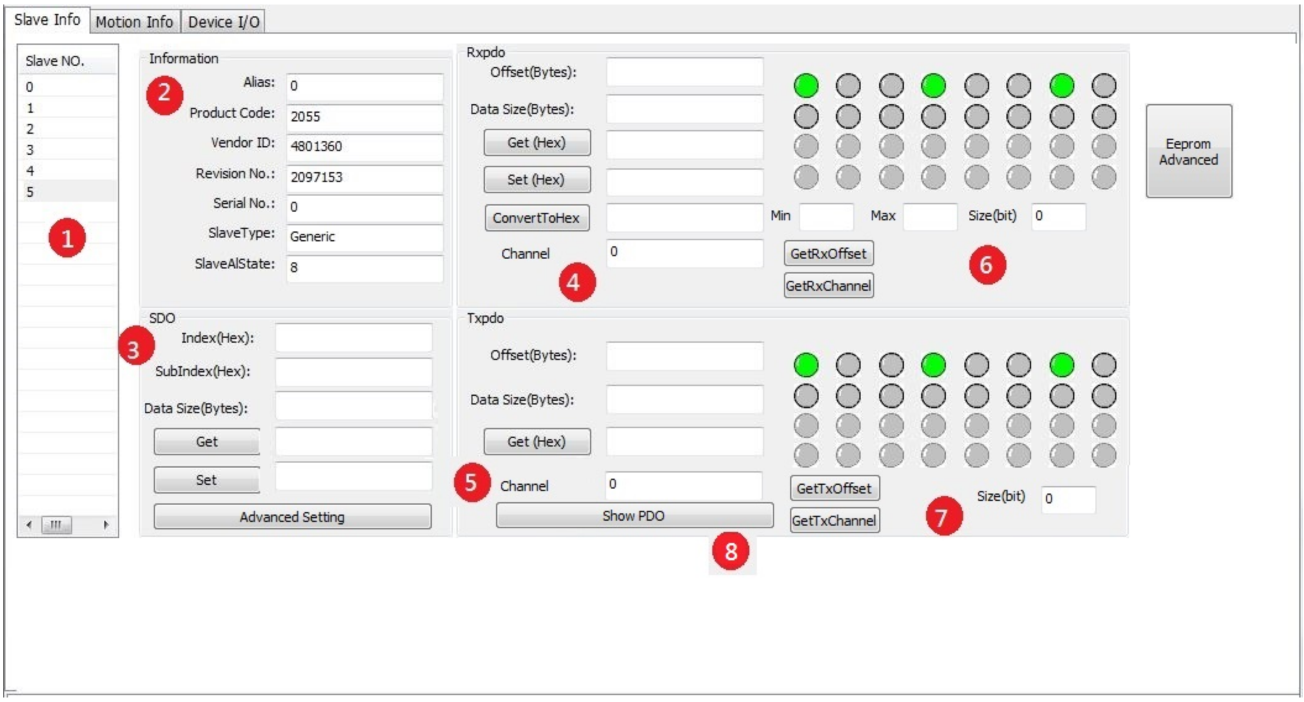
3.3. Device Network Status

As shown below, when the device communication operation is enabled, the device network statuses are updated continuously. The description of each status is in the following table.

Item	Description
(1)	The total number of responding slaves
(2)	EtherCAT AL states of all slaves (EtherCAT states: INIT, PREOP, SAFEOP, OP)
(3)	EtherCAT working counter value It provides an indication for communication status.
(4)	Network link status of EtherCAT It indicates a good wire connection or not. Down: Link Down Up: Link Up

3.4. Slave Operation Page

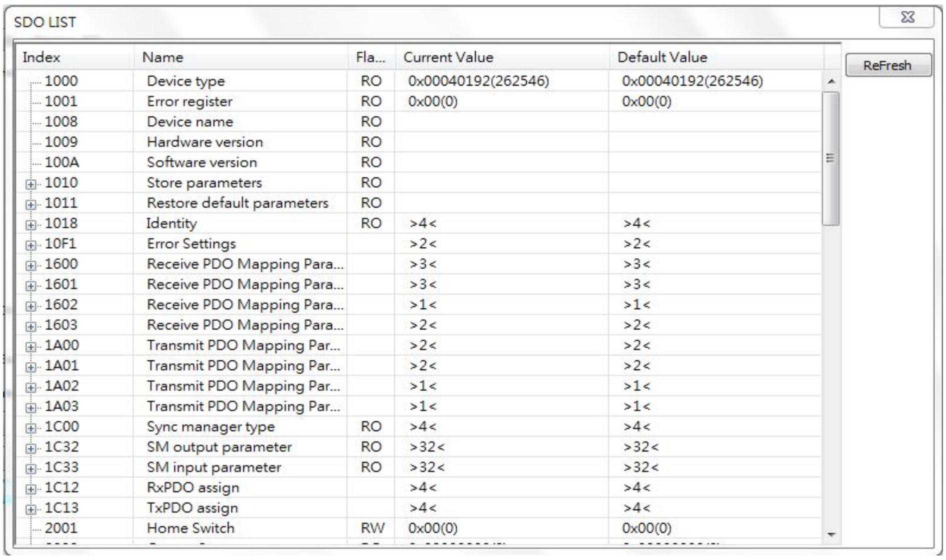
The following page is for Slave operation. The descriptions are in the table.



Item	Description
(1)	Slave list (all the scanned slaves are listed here)
(2)	Slave information of the selected slave
(3)	SDO read/write for the selected slave
(4)	Read/Write utput objects (RxPDO) of the selected slave
(5)	Read input objects (TxPDO) of the selected slave
(6)	Control digital outputs and display the DO status of the selected slave
(7)	Display the digital input status of the selected slave
(8)	Show PDO It can used to show all defined objects.

3.4.1. Basic Slave Operation Steps

1. After executing the device initialization steps, the user can select a slave from the list of slaves. The related slave informations will be displayed in the slave information group box.
2. User can read/write SDO objects by entering the Index, SubIndex and Data Size in SDO read/write group box, and clicking "Get/Set" to read/write Object value.
3. The Advance Setting button provides users with easy access to read and write SDO. Press the ReFresh button to update the data.


SDO LIST

ReFresh

Index	Name	Fla...	Current Value	Default Value
1000	Device type	RO	0x00040192(262546)	0x00040192(262546)
1001	Error register	RO	0x00(0)	0x00(0)
1008	Device name	RO		
1009	Hardware version	RO		
100A	Software version	RO		
1010	Store parameters	RO		
1011	Restore default parameters	RO		
1018	Identity	RO	>4<	>4<
10F1	Error Settings		>2<	>2<
1600	Receive PDO Mapping Para...		>3<	>3<
1601	Receive PDO Mapping Para...		>3<	>3<
1602	Receive PDO Mapping Para...		>1<	>1<
1603	Receive PDO Mapping Para...		>2<	>2<
1A00	Transmit PDO Mapping Par...		>2<	>2<
1A01	Transmit PDO Mapping Par...		>2<	>2<
1A02	Transmit PDO Mapping Par...		>1<	>1<
1A03	Transmit PDO Mapping Par...		>1<	>1<
1C00	Sync manager type	RO	>4<	>4<
1C32	SM output parameter	RO	>32<	>32<
1C33	SM input parameter	RO	>32<	>32<
1C12	RxPDO assign		>4<	>4<
1C13	TxPDO assign		>4<	>4<
2001	Home Switch	RW	0x00(0)	0x00(0)

3.4.2. Slave PDO and DI/DO LED Operation Steps

1. After the user completes the start EtherCAT operation task steps, he can access PDO by entering the Offset and Data Size in the PDO read/write group box and then clicking "Get/Set" to read/write the slave PDO data. The data to be access are composed of bytes; and all the bytes are separated by commas. For example, writing 2-byte data, 0x02FF, the user has to enter a string **FF,02** to the write text box. It means that the first data to be written is 0xFF and the second byte is 0x02. If data is a double word, 0x12345678, please take the little endian expression as 78,56,34,12.
2. DO/DI LED operations include some further processing on RxPDO and TxPDO data and shows the status on LED display. A DO slave module has RxPDO objects mapping to digital outputs. A DI slave module has TxPDO objects mapping to digital inputs.

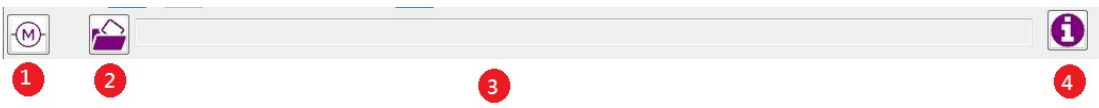
Users can change digital outputs by writing data to RxPDO objects and get their values by reading them. In the same way, user can get the values of digital inputs by reading TxPDO objects.

3. If users would like to know how many channels the module has, they can use GetRxChannel or GetTxChannel to obtain the number and size of the channels.
4. If the module has multiple channels, users can enter channel and size, then press GetRxOffset or GetTxOffset to get the offset of the channel.
5. ConvertToHex button function: Enter the decimal value you want to convert to a hexadecimal number. The MIN and MAX define the range of that decimal value; and Size (bit) defines the range of the hexadecimal value. Note: here this hexadecimal value is a signed value.

   EX: We want to send a 10 voltages output command to an analog output module. If the AO output range is 0V to 10V; and the resolution of the AO channel is 8-bit. Here, a value to be converted will be: 10; the range will be MIN: 0 and MAX: 10; and the output Size (bit) is 8. The result of the converted value is going to be **7F. Use 7F as input to an analog output channel will produce a 10 Voltages output.**
6. Show PDO button function: Show RxPDO and TxPDO objects.

3.5. Motion Control Initialization Toolbar

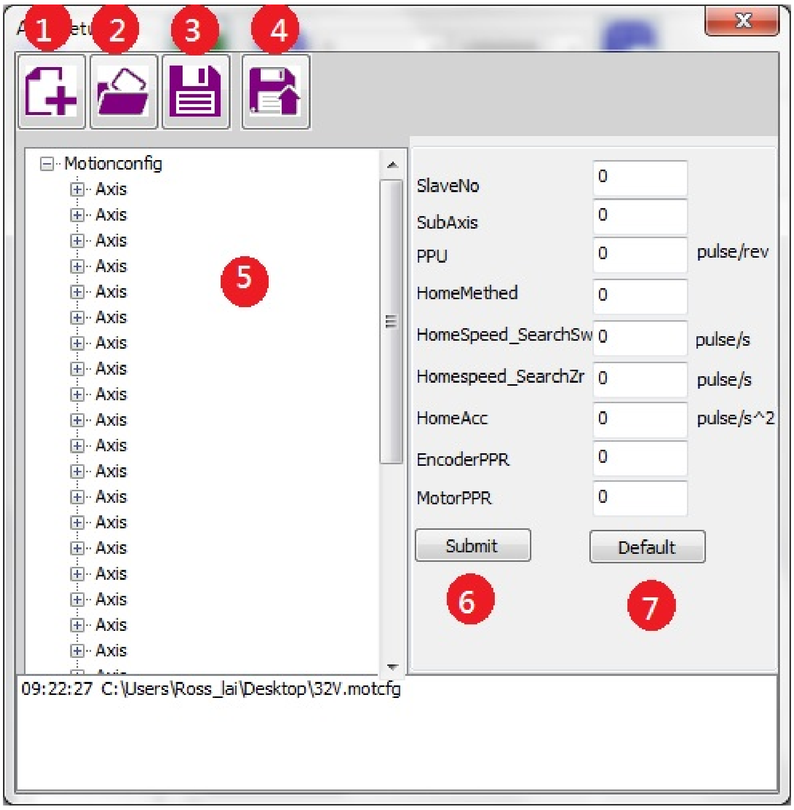
For motion control applicaitons, this basic configuration is necessary. Parameters for defining axes must be initialized before starting motion control. The motion control initialization toolbar is shown below, and the descriptions are shown in the following table.



Item	Description
(1)	Start to initialize axes for motion control according to a file selected by (2).
(2)	Open the file dialog for selecting a parameter file
(3)	Path information of the parameter file is shown here.
(4)	Open the edit page of motion control parameter file

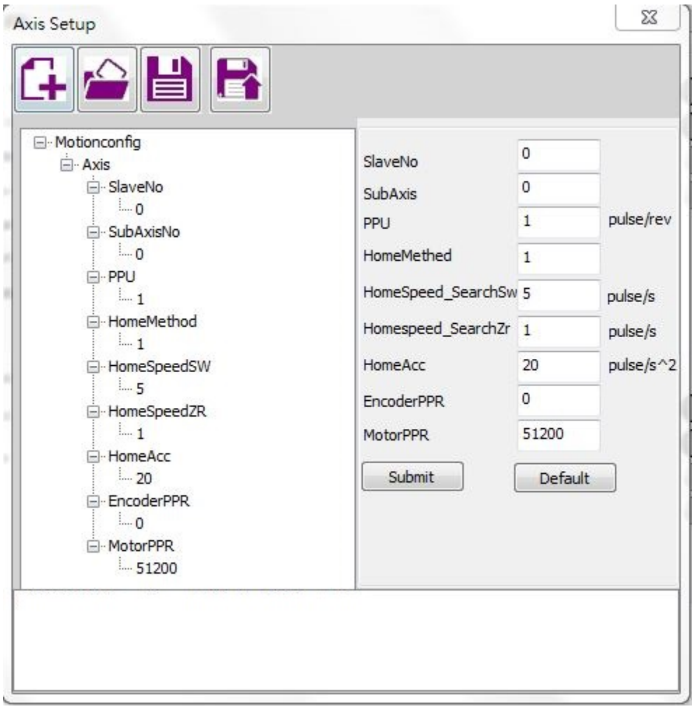
3.5.1. Motion Control Parameter File Editing Steps

After clicking on the Motion Control Initialization Toolbar, the Control Parameter Edit page is opened as follows. The description of each control item is shown in the following table.

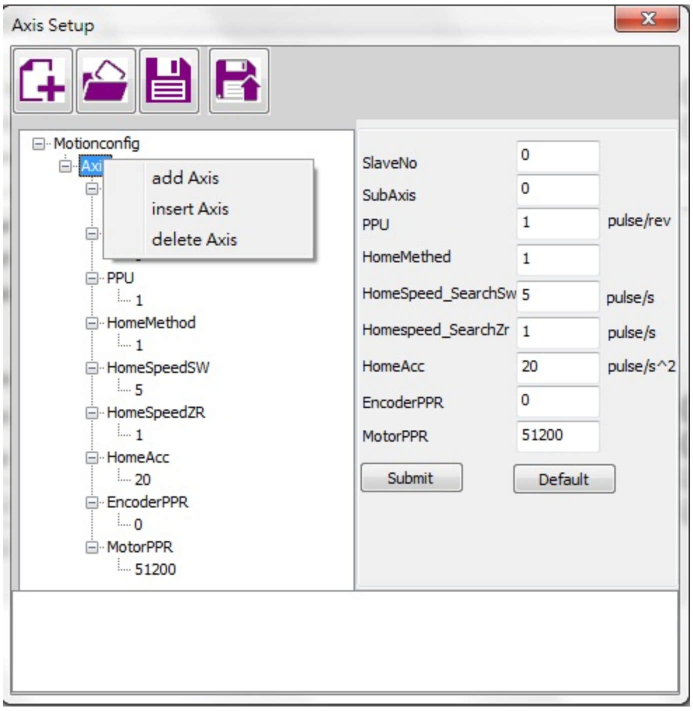


Item	Description
(1)	Create a new parameter file
(2)	Open an existed parameter file
(3)	Save the current parameter file
(4)	Save as another parameter file
(5)	Parameter information panel
(6)	Submit (apply) selected axis settings. **If you need these changes, do click it before switching to edit any other axis.**
(7)	Get the default values for the selected axis

1. Click to create a new parameter file. An axis is created automatically.

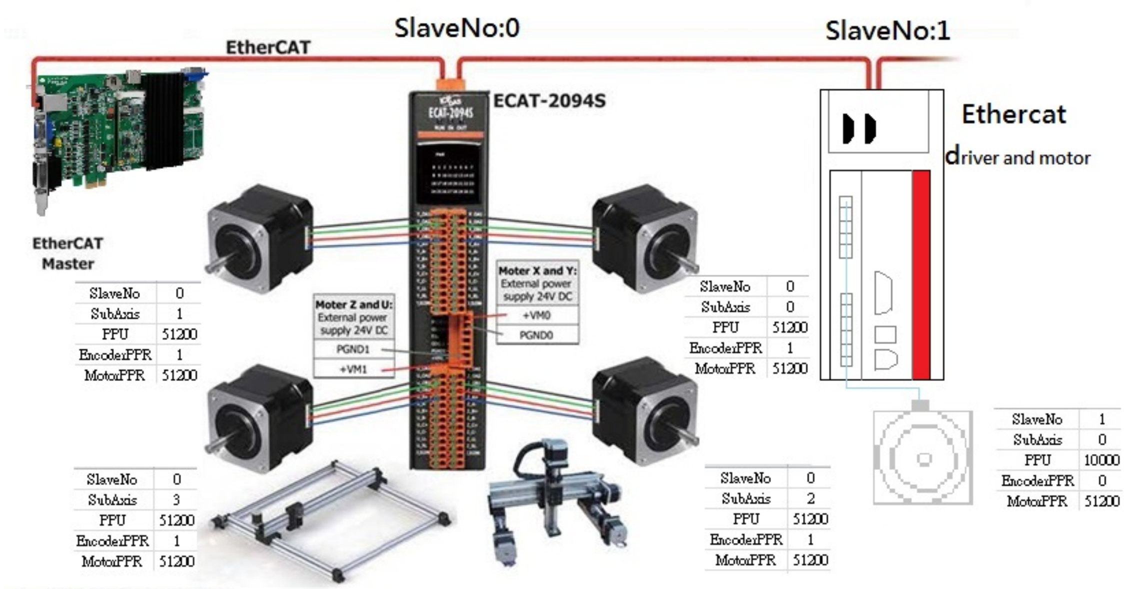


2. Set the **SlaveNo** first. This is the axis number to be operated for this slave. **Note: When this SlaveNo is set to be 65535, it becomes a virtual axis.** SubAxis is for configuring mutiple axes on one slave, such as some multi-axis stepper motor drivers. Right now, the ECAT-2094S is supported. Set 0 to Subaxis for the first axis of 2094s; set 1 to Subaxis for the second axis; and so on. Enter suitable values for PPU (pulses per unit), HomeMethod, HomeSpeed_SearchSw (speed for searching switch), HomeSpeed_SearchZr (speed for searching index), HomeAcc (acceleration), EncoderPPR (pulse per revolution of encoder whcih is defined for appending an encoder to a stepper motor), MotorPPR (pulse per revoltion of motor) parameters.
3. Choose an axis node by clicking the right-hand mouse button; a small menu will pop-up. Choose "**add Axis**" to add an axis after the last node. Click "**insert Axis**" to insert an axis right after the current node. Click "**delete Axis**" to delete the selected axis.



4. After editing an axis, click **Submit** button to confirm the changes. If not, the changes will not take effect. Any time you click , the changed contents are save to file.
5. Click to save the contents into a new parameter file.

Example: Following ECAT-2094S has 4 axes. Another servo drive is a standard CiA402 drive.



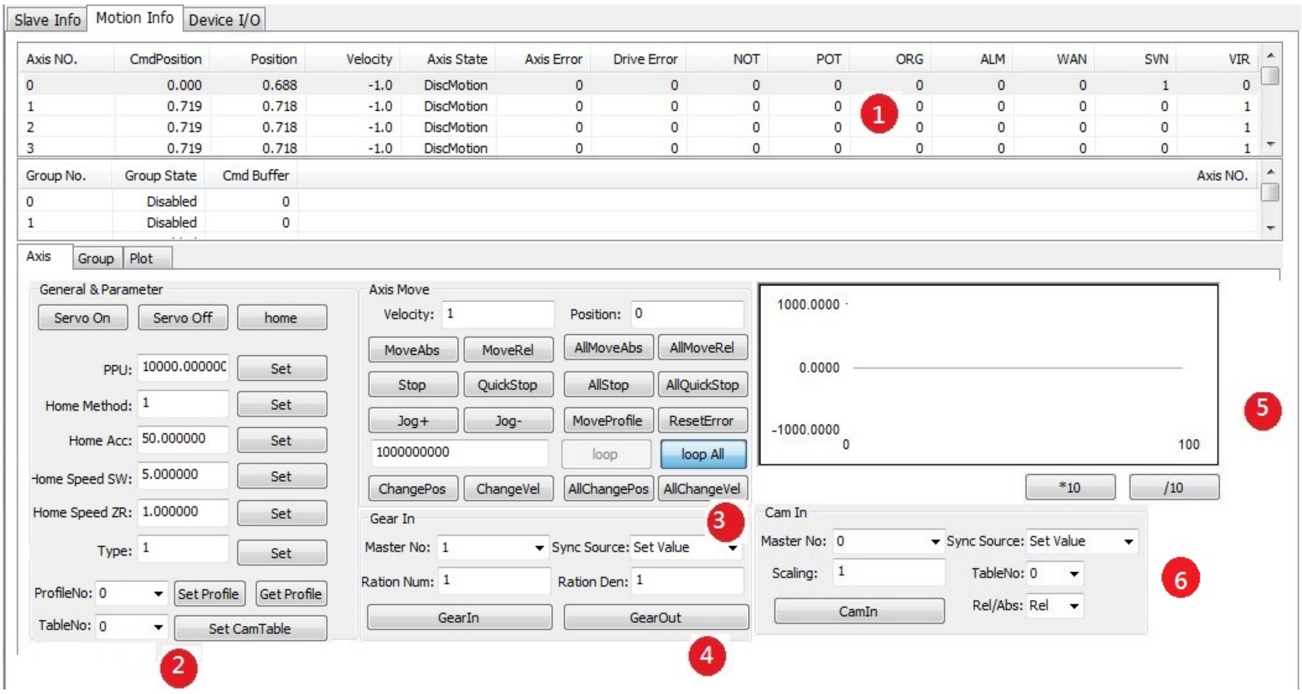
3.5.2. Motion Control Initialization Steps

1. After the user has completed the motion control parameter file editing step, click on the motion control initialization toolbar to open the edited parameter file.
2. Click will use this parameter file to initializing the every single-axis definition. To configure groups for motion control, further steps need to be implemented.

3.6. Motion Control Page

After the initialization of the motion control, the user can start to do motion control operations. The motion control page includes two parts: (1) single-axis motion control page. (2) group motion control page.

3.6.1. Single-Axis Motion Control Page



Item	Description
(1)	Single-axis motion Information
(2)	Single-axis parameter settings
(3)	Single-axis motion control function tests
(4)	Gear function settings and testings
(5)	Display of single-axis Position and Velocity
(6)	E-Cam function settings and testings

Single-axis motion informations

Item	Description
Axis No.	Axis number
Position	Axis position
Velocity	Axis velocity
Axis State	Axis state
Axis Error	Axis last error
Drive Error	Axis drive error
NOT	Negative limit switch
POT	Positive limit switch
ORG	Home switch
ALM	Alarm
WAN	Warning
SVN	Servo ON/OFF state
VIR	Virtual Axis (when slave number is 65535)

Single-axis parameter settings

1. Choose an axis by clicking a slave number in the single-axis motion information panel.
2. Click the "Servo ON/OFF" button to enable or disable the drive. But if it is a virtual axis, the motion control can be started without this operation.
3. Click the "Home" button to start homing of this axis.
4. Click the "Set" button to apply the change of parameters.
5. Parameters can be modified. After press **Set**, it will take effect. However, these changes can not save back to the configuraiton file.

Single-axis motion control functions

1. An axis is selected in the single-axis motion information panel, and it is enabled.
2. Set values of "Velocity" and "Position" parameters.
3. Click the "MoveAbs" or "MoveRel" button to do a single-axis motion control test.

"MoveAbs" can move the selected axis in absolute position mode; while **"MoveRel"** is moving by a relative distance.

4. Clicking "Stop" or "QuickStop" button can stop this single-axis motion control test.
5. To control of all axes, set the velocity "Velocity" and "Position" parameters. Then click the **"All MoveAbs"** or **"All MoveRel"** button to use the same parameter settings to perform single-axis motion control for all axes. Click the "All Stop" or "All QuickStop" button to stop all axes.
6. When the "Jog+" or "Jog-" button is clicked, the "Velocity" parameter is used to start a movement with a specified velocity. Click "Stop" or "QuickStop" to stop this motion.
7. The edit box beside the **loop** and **loop** buttons is used for enter a loop number. Set this value first. When loop or loop All is clicked, the axis or axes will move back and forth between the current position and the set position.

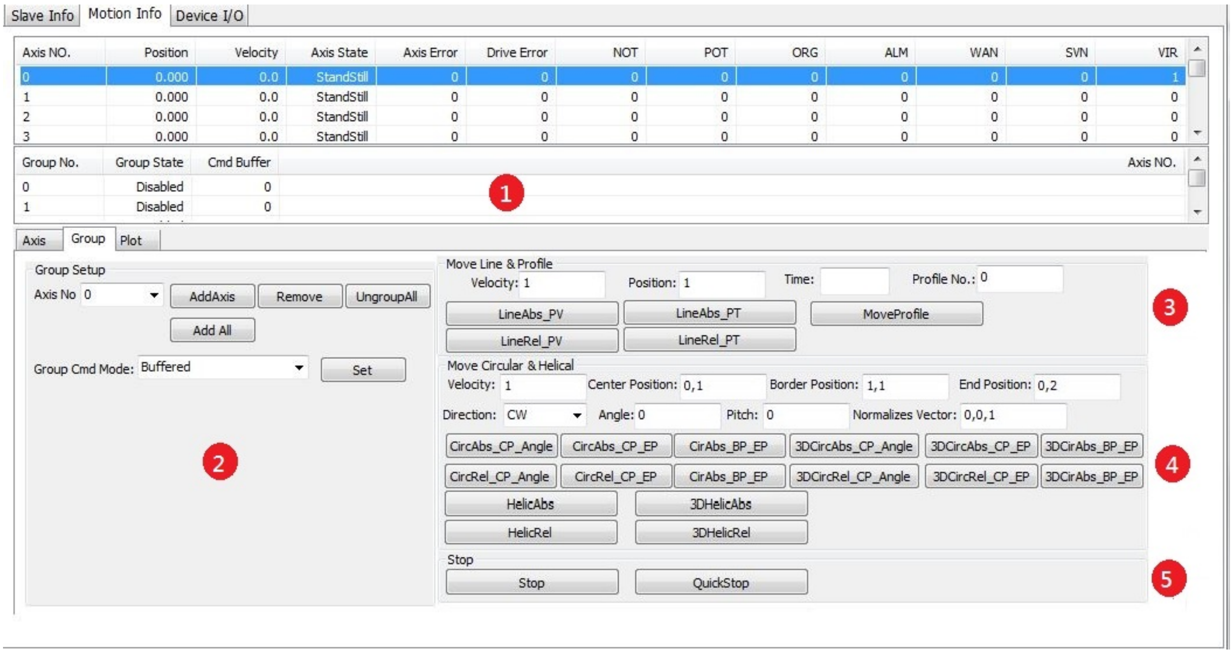
Gear function settings and testing

1. An axis is selected in the single-axis motion information panel, and it is enabled.
2. Set its "Master No" parameter. This master axis will be the reference axis.
3. Next, set the slave reference source. The reference source can be either the master's command set value or the master's actual position value.
4. The electronic gear ratio is composed by a numerator and a denuminator. Set the numerator in the edit box with label "Ratio Num", and set the denominator in the edit box with label "Ratio Den". Source value multiplied by the gear ratio will be the reference command of the slave axis.
5. Next, click the "GearIn" button to start the gear motion. The state of the slave axis will change into SyncMotion. After that, the slave axis will follow any motion of the master axis with the gear ratio defined before.
6. Click the "Gearout" button will stop the synchronized motion. The state of the slave axis will change from synchronized motion to be the continuous motion. If you want to stop the gear motion, click the "Stop" or "QuickStop" button to stop this following motion control.

eCam function settings and testing

1. An axis is selected in the single-axis motion information panel, and it is enabled.
2. Set its "Master No" parameter. This master axis will be the reference axis.
3. Next, set the slave reference source. The reference source can be either the master's command set value or the master's actual position value.
4. Next, Set the Scaling,TableNo and Rel/Abs to define how slave following master axis.
5. Next, click the "CamIn" button to start the eCam motion. The state of the slave axis will change into SyncMotion. After that, the slave axis will follow any motion of the master axis by the definition in CamTable mentioned before.
6. If you want to stop the eCam motion, click the "Stop" or "QuickStop" button to stop this following motion control.

3.6.2. Group Motion Control Page



Item	Description
(1)	Group motion information panel
(2)	Group motion parameter settings
(3)	Group linear motion and Profile motion tests
(4)	Group circular motion and helical motion tests
(5)	Group stop function tests

Group motion information panel

1. Click "Group" to switch to the group motion information panel. The definition of each item is explained as follows.

Item	Description
Group No.	Group number
GroupState	Group state
Cmd Buffer	The number of commands in command buffer (Each group command buffer has a limited size. This item shows the remaining commands in this group buffer.)
Axis No.	All the axis numbers of this group is listed here

Group motion parameter settings

1. Select a specified group number in the group motion information panel.
2. If there is not any axis number in the group, the group state is Disabled. User cann select the desired axis number from the "Axis No" and click the "Add" button to add this specified axis number to the group. This process can be performed as many axes as user wants to.
3. Click the "Remove" button can remove a specified axis number from the group.
4. Click the "Ungroup All" button can remove all axes from the group.
5. Click the "Set" button to apply settings.

Group linear motion control

1. Select a specified group number in the group motion information panel.
2. Set "Velocity" and "Position" parameters. Use commas to separate each position inputs. For example, when starting two-axis linear interpolation moving in absulute position method, users can input 50,100 in the position edit box to move the first axis to 50 and the second axis to 100.
3. Next, click the "Line Abs" or "Line Rel" button to start the multi-axis linear interpolation moving in absolute or relative mode.
4. While moving, click "Stop" or "QuickStop" to stop the group motion.

Group circular motion control

1. Select a specified group number in the group motion information panel.
2. Set "Velocity", "Center Position", "Angle" parameters. Use the comma to separate the data of center position.
3. Click the "Circular Abs" or "Circular Rel" button to start circular interpolation moving according to your desired absolute or relative mode.
4. While moving, click "Stop" or "QuickStop" to stop the group motion.

Item	Description
CircAbs_CP_Angle	Start group 2D circular interpolation motion by setting a center position and an angle in the absolute mode.
CircRel_CP_Angle	Start group 2D circular interpolation motion by setting a center position and an angle in the relative mode.
CircAbs_CP_EP	Start group 2D circular interpolation motion by setting a center position and an end position in the absolute mode.
CircRel _CP_EP	Start group 2D circular interpolation motion by setting a center position and an end position in the relative mode.
CircAbs_BP_EP	Start group 2D circular interpolation motion by setting a border position and en end position in the absolute mode.
CircRel _BP_EP	Start group 2D circular interpolation motion by setting a border position and en end position in the relative mode.
3D CircAbs_CP_Angle	Start group 3D circular interpolation motion by setting a center position and an angle in the absolute mode.
3D CircRel _CP_Angle	Start group 3D circular interpolation motion by setting a center position and an angle in the relative mode.
3D CircAbs_CP_EP	Start group 3D circular interpolation motion by setting

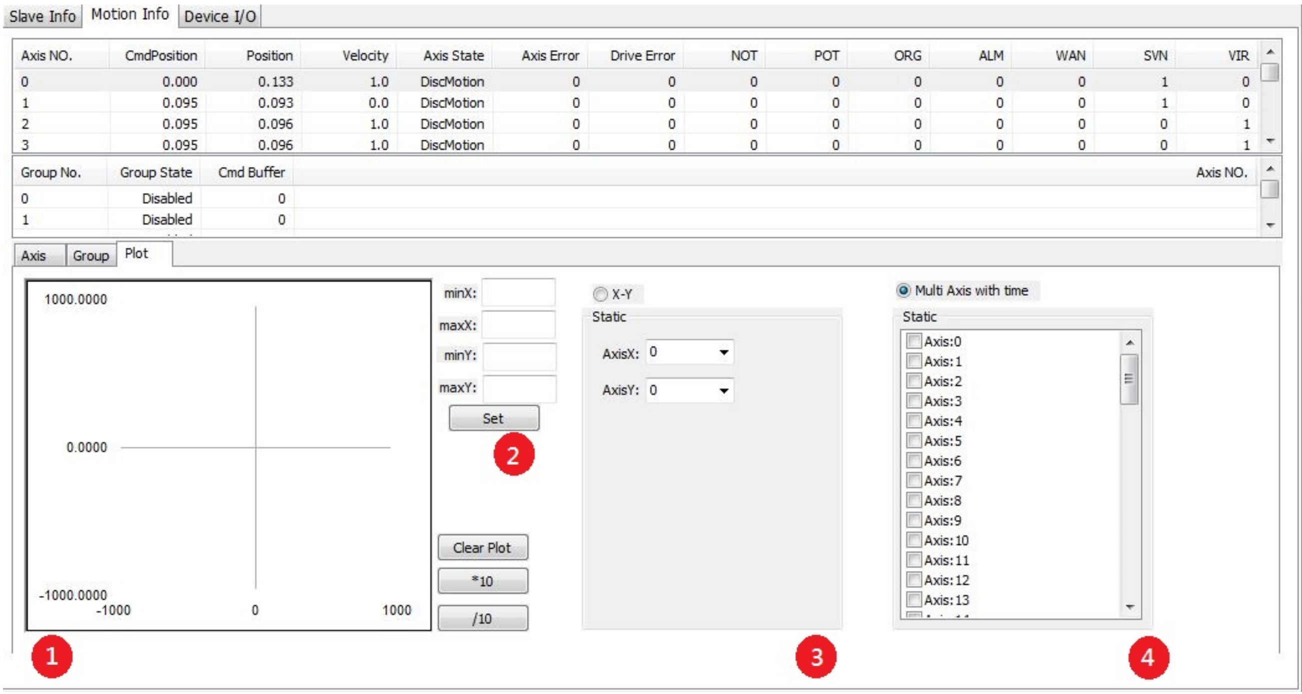
	a center position and an end position in the absolute mode.
3D CircRel _CP_EP	Start group 3D circular interpolation motion by setting a center position and an end position in the relative mode.
3D CircAbs_BP_EP	Start group 3D circular interpolation motion by setting a border position and an end position in the absolute mode.
3D CircRel _BP_EP	Start group 3D circular interpolation motion by setting a border position and an end position in the relative mode.

Group helical motion control

1. Select a specified group number in the group motion information panel.
2. Set "Velocity", "Center Position", "Angle", "Pitch" parameters. Use a comma to separate the two inputs of the center position.
3. Click the "Helical Abs" or "Helical Rel" button to start a helical interpolation motion according to your desired absolute or relative mode..
4. While moving, the group motion can be stopped by clicking "Stop" or "QuickStop".

Item	Description
Helical Abs	Start the helical interpolation motion of a group in the absolute mode.
Helical Rel	Start the helical interpolation motion of a group in the relative mode.
3D Helical Abs	Start the 3D helical interpolation motion of a group in the absolute mode.
3D Helical Rel	Start the 3D helical interpolation motion of a group in the relative mode.

3.6.3. Show Position Page



Item	Description
(1)	Motion position display area This plot can be determined by (3) or (4).
(2)	Motion position range and scaling settings
(3)	Select "X-Y" plot for (1) and choose axis for X and Y
(4)	Select multiple axes Position vs Time plot and choose axes for this plot.

Motion position display area

Item	Description
minX	minimum value of x-axis
maxX	maximum value of x-axis

minY	minimum value of y-axis
maxY	maximum value of y-axis

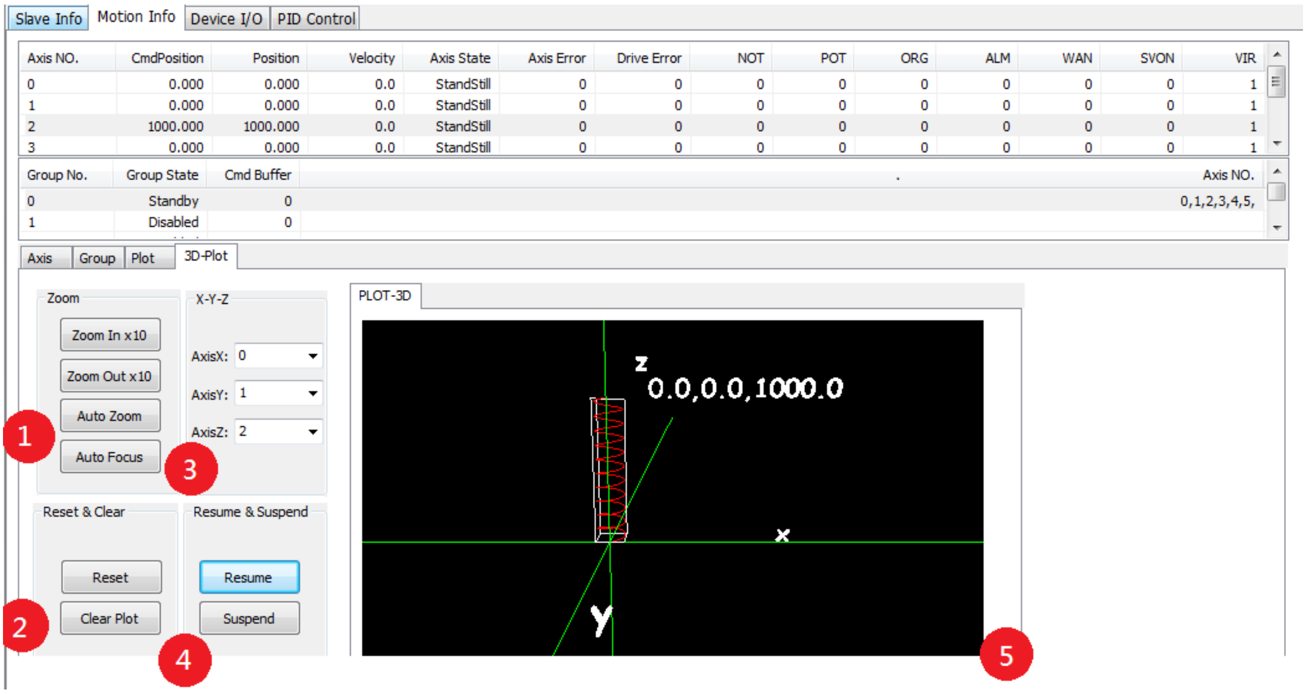
X-Y display

1. Set the X axis and Y axis to observe their 2D position variation through time.

multiple axes display

1. Check axes that you are interested in; and then you can observe position vs time of these axes

3.6.4. 3D Show Position Page



Note:Only show data within an hour.

Item	Description
(1)	Zoom In/Out
(2)	Reset/Clear
(3)	Axis Setting
(4)	Resume/Suspend the plotting
(5)	Motion position display area

Zoom In/Out

Item	Description
Zoom In x10	10X Zoom In
Zoom Out x10	10X Zoom Out

Auto Zoom	Auto Zoom In/Out
Auto Focus	Auto Focus

Reset/Clear

Item	Description
Reset	Reset Motion position display area
Clear Plot	Clear Motion position display area

X-Y-Z Axis Setting

Item	Description
AxisX: 0	Setting of X-Axis
AxisY: 1	Setting of Y-Axis
AxisZ: 2	Setting of Z-Axis

Motion position display area

Horizontal and Vertical movement: press middle mouse button and drag.

Rotation around X-Axis and Y-Axis: press left mouse button and drag.

Zoom IN/Out: press right mouse button and drag.

3.7. Device I/O Operation Page

Switch to the device I/O operation page as shown below. These are local I/O, not EtherCAT I/O, provided by the ECAT-M801 control card. The description of each control item is shown in the following table.

Item	Description
(1)	Device DO control
(2)	Device DI control

3.7.1. Device DO control operation step

1. Click the "Set" button to write the data for all DO channels.
2. Click the "Get" button to get the DO settings. If the "Bit" option is selected, enter the bit number in the edit box and click the "Get" button to get the bit value of the specified bit number.

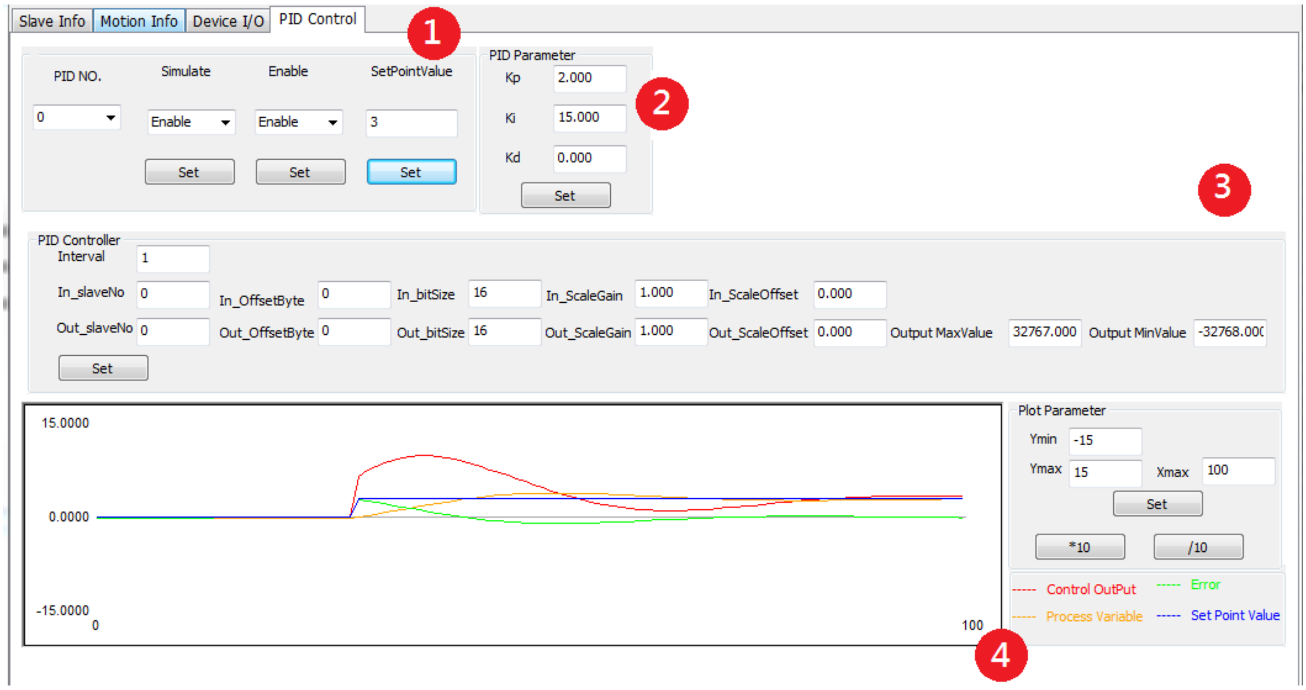
3.7.2. Device DI control operation step

1. Click the "Get" button to get the DI data. If the "Bit" option is selected, enter the bit number in the edit box and click the "Get" button to get the bit value of the specified bit number.

3.8. PID Control Page

After the user start and successfully enter EtherCAT operation task, the user can test PID Controller.

3.8.1. PID Control Page



item	Description
(1)	Status of PID Controller
(2)	Parameters of PID Controller
(3)	Input/Output module settings for PID Controller
(4)	Plots for PID Controller

Status of PID Controller

item	Description
PID No.	PID Controller Number
Simulate	Enable simulation or not
Enable	Activate PID Controller or not
SetPointValue	Setting the Setpoint value (i.e. system command)

Parameters of PID Controller

1. Choose PID Controller Number.
2. Set PID Controller Input module and Output Module. Refer to (3).
3. Set PID Parameters.
4. Set Simulate value as "**Enable**" to activate simulation. Set Simulate value as "**disable**" will activate the measurement and control function of the Input module and Output Module, respectively.
5. Set Enable as "**Enable**" to activate PID Controller. "**Disable**" will stop PID control.

Input/Output module settings for PID Controller

item	Description
Interval	Control Interval of PID Controller，Unit: Ethercat CycleTime
In_slaveNo	Measuring channel is located in this slave module
In_OffsetByte	TxPdo Offset of the measuring channel
In_bitSize	DataSize of this measuring channel, Unit:bit
In_ScaleGain	Scale gain for conversing digital value into physical value
In_ScaleOffset	Scale offset for conversing digital value into physical value
Out_slaveNo	Control output channel is located in this slave module
Out_OffsetByte	RxPdo Offset of this control output channel
Out_bitSize	DataSize of this control output channel，Unit:bit

Out_ScaleGain	Scale gain for conversing physical value into digital value
Out_ScaleOffset	Scale offset for conversing physical value into digital value
Output MaxValue	Maximum Limitation of Control Output
Output MinValue	Minimum Limitation of Control Output

4. Function Overview

4.1. Device Operation Flow

As shown in Figure 4.1, the user can call the *GetDeviceCnt* function to find out how many devices (cards) can be used. Each device should has a unique Card ID. The Card ID is set by four-bit dip-switch on the ECAT-M801 control card. Then, according to the Card ID, call *OpenDevice* function to open that device. After this device is opened, the EtherCAT cyclic communication does not start yet. Some basic device operation functions should be used to configure the communication before the cyclic communication can be started.

At first, the user can use *GetDeviceState* to get the current states of the EtherCAT network. These states includes the number of currently connected slaves, the AL status, network link status, etc. Next, the *GetSlaveInfo* function can be called for each slave to get the slave information. If some SDO objects need to be read/written, the *GetSlaveSdoObject* and *SetSlaveSdoObject* functions can be used for these purposes. These functions will do acyclic communication through EtherCAT Bus.

Before starting the operation task of EtherCAT, please use the utility program to create and edit at least one EtherCAT network information file and write the the system information into the device. Then, in your program, call *StartDeviceOpTask* function to start the EtherCAT operation task. This function will command slaves to enter into the OP state. The user can use *GetDeviceState* to get the current states. If there is no error and the AL state reaches OP, the PDO cyclic data communication is on. Motion control operations can be configured and started. To stop the EtherCAT communication, *StopDeviceOpTask* function must be called. To close the device operation (close a card), use *CloseDevice* function to do it.

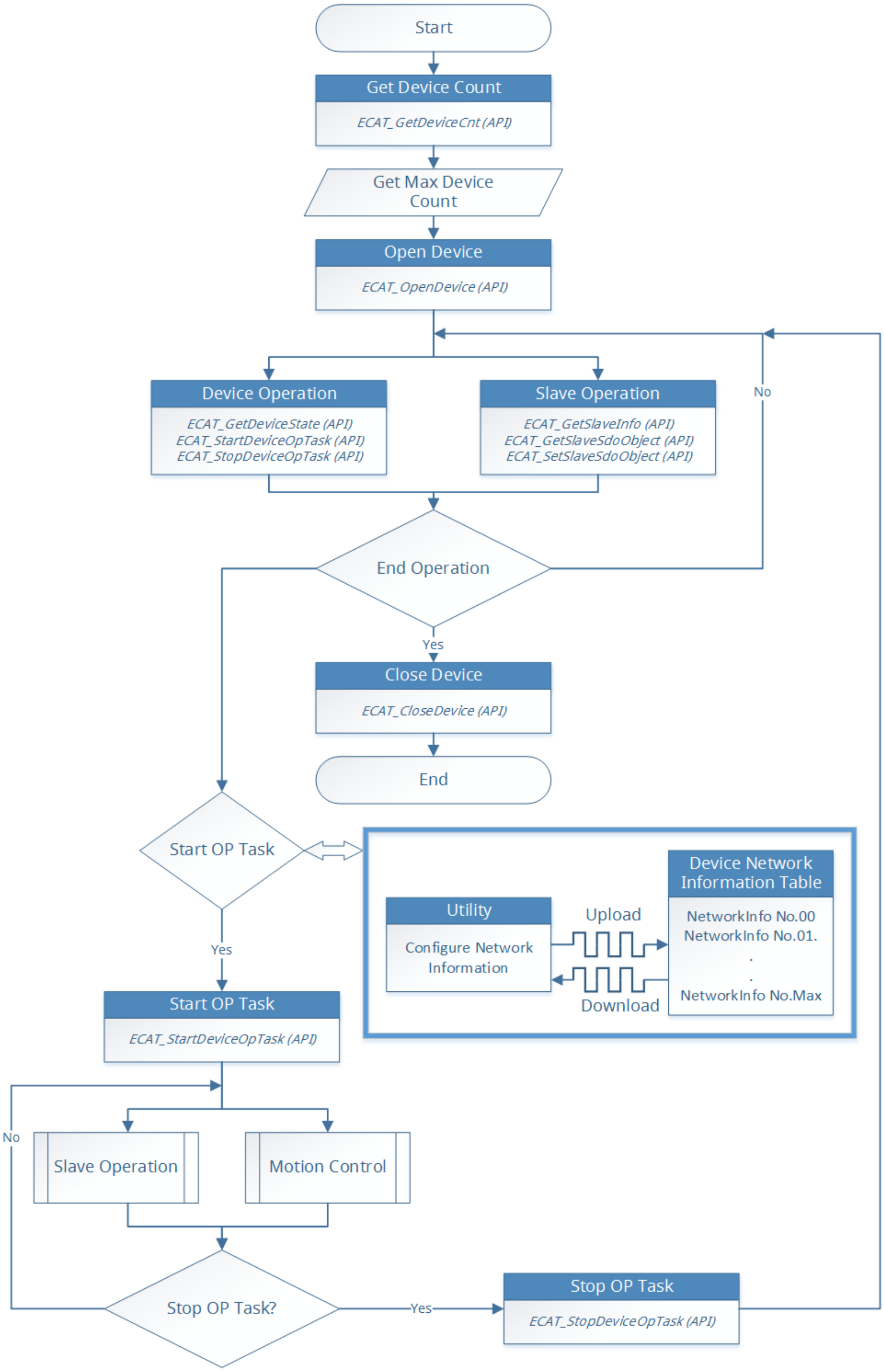

Start
Get Device Count
ECAT_GetDeviceCnt (API)
Get Max Device Count
Open Device
ECAT_OpenDevice (API)
Device Operation
ECAT_GetDeviceState (API)
ECAT_StartDeviceOpTask (API)
ECAT_StopDeviceOpTask (API)
Slave Operation
ECAT_GetSlaveInfo (API)
ECAT_GetSlaveSdoObject (API)
ECAT_SetSlaveSdoObject (API)
No
End Operation
Yes
Close Device
ECAT_CloseDevice (API)
End
Start OP Task
Device Network Information Table
Utility
Configure Network Information
Upload
Download
NetworkInfo No.00
NetworkInfo No.01.
NetworkInfo No.Max
Yes
Start OP Task
ECAT_StartDeviceOpTask (API)
No
Slave Operation
Motion Control
Stop OP Task?
Yes
Stop OP Task
ECAT_StopDeviceOpTask (API)

Figure 4.1

4.2. Slave Operation Flow

As shown in Figure 4.2, Slave operation can be divided into two parts. First, do the basic operation of the device. The *GetSlaveInfo*, *GetSlaveSdoObject*, *SetSlaveSdoObject* functions are provided. Next, make EtherCAT communication enter into OP state; then read/write functions of RxPDO, TxPDO can be called to get/set object values.

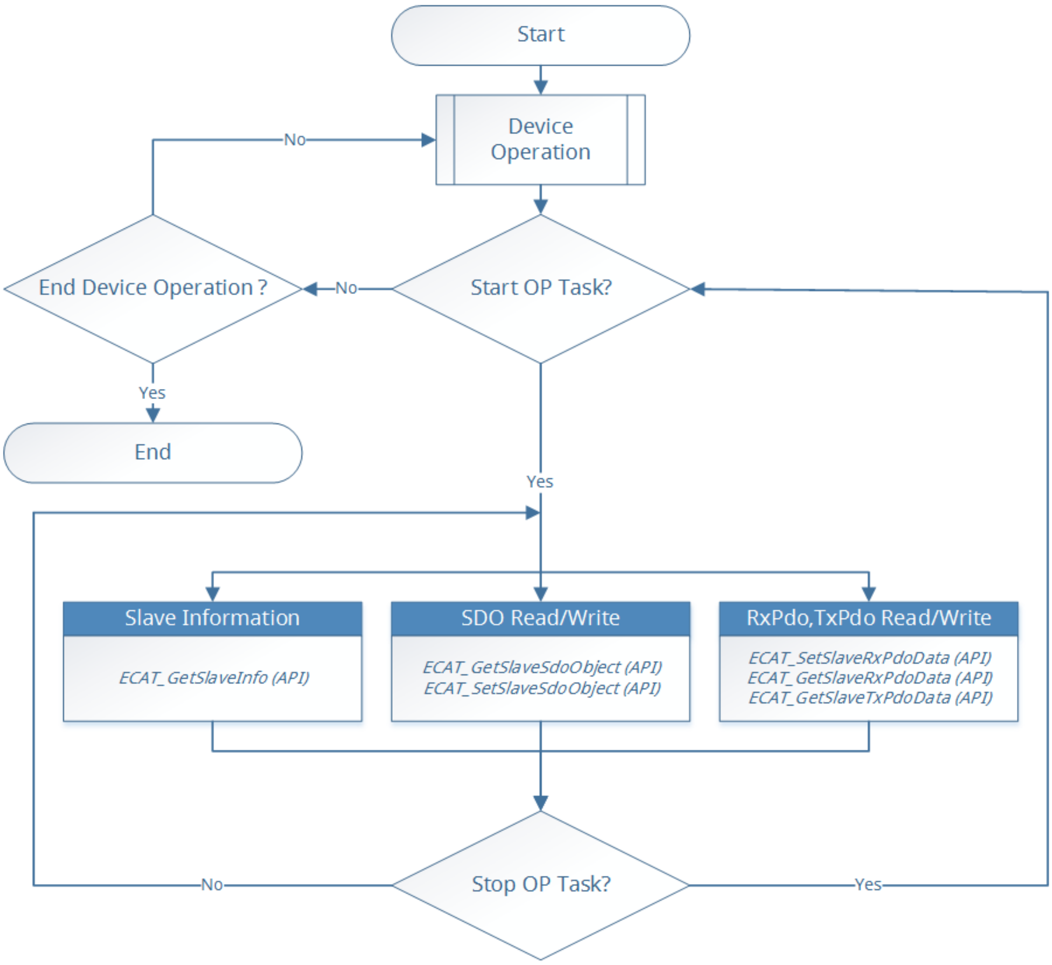


Figure 4.2

4.3. Motion Control Flow

4.3.1. Motion Control Initialization

As show in Figure 4.3, before starting the motion control operation, the initialization operation needs to be performed first. The initialization will assign different axis numbers to specified slaves. The device performs motion control according to those axis numbers.

Call *McInit* function to initialize the motion control. If the initialization is successful, the user can start various motion operations, such as axis homing, axis operation (single axis motion functions), axis error processing and group operation (multi-axis motion functions).

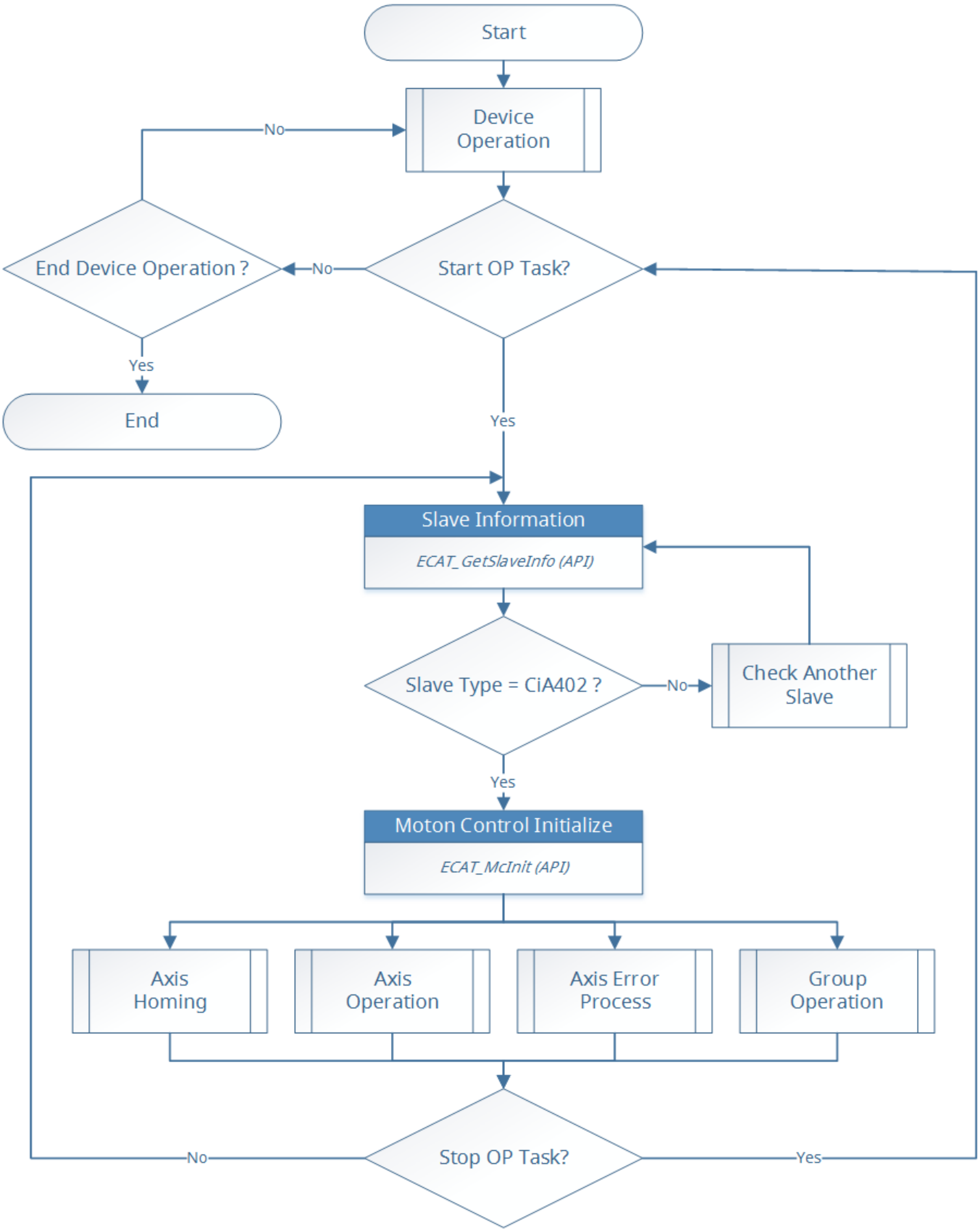

Start
Device Operation
No
End Device Operation ?
No
Start OP Task?
Yes
End
Yes
Slave Information
ECAT_GetSlaveInfo (API)
Slave Type = CiA402 ?
No
Check Another Slave
Yes
Moton Control Initialize
ECAT_McInit (API)
Axis Homing
Axis Operation
Axis Error Process
Group Operation
No
Stop OP Task?
Yes

Figure 4.3

4.3.2. Axis Motion Control

As shown in Figure 4.4, users need to configure various parameters before performing single-axis operation. After setting these parameters, the user can call *McGetAxisState* to get the state of an axis. If the state is Standstill, it means that axis is currently stopped and ready to receive a new motion command. After successfully calling a motion function, the axis state will change from the current Standstill state to a suitable state, such as Discrete Motion, Continuous Moiton, or Synchronized Moiton. If the state is in either one of these three states, it indicates that axis is moving.

When an axis is moving, the user can call stop functions to stop its motion. Only when the axis state changes to Standstill a new motion command can be issued again. If any error occurs while moving, the state of that axis will change to ErrorStop. In ErrorStop state, users need to deal with this error.

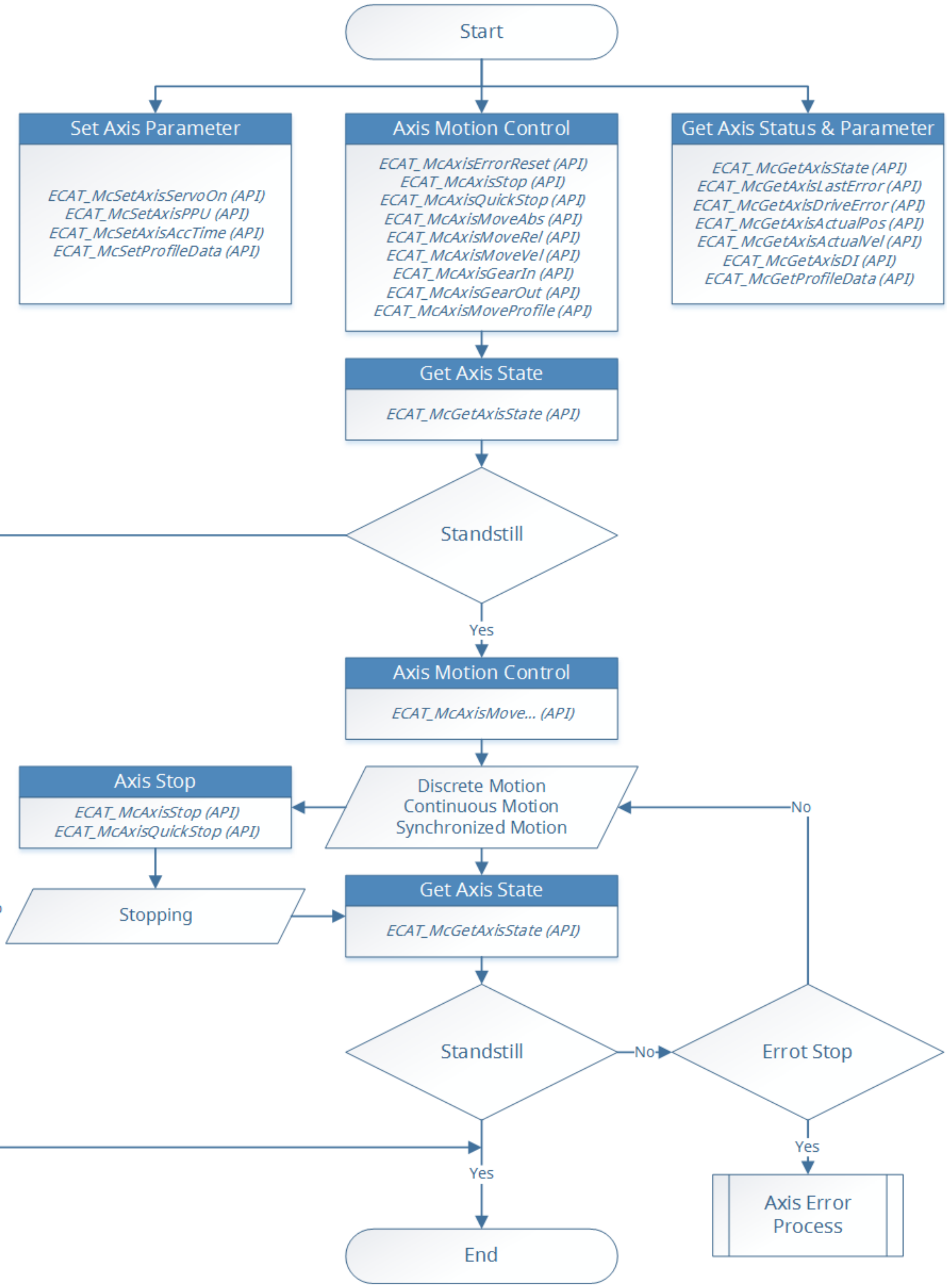

Start
Set Axis Parameter
ECAT_McSetAxisServoOn (API)
ECAT_McSetAxisPPU (API)
ECAT_McSetAxisAccTime (API)
ECAT_McSetProfileData (API)
Axis Motion Control
ECAT_McAxisErrorReset (API)
ECAT_McAxisStop (API)
ECAT_McAxisQuickStop (API)
ECAT_McAxisMoveAbs (API)
ECAT_McAxisMoveRel (API)
ECAT_McAxisMoveVel (API)
ECAT_McAxisGearIn (API)
ECAT_McAxisGearOut (API)
ECAT_McAxisMoveProfile (API)
Get Axis Status & Parameter
ECAT_McGetAxisState (API)
ECAT_McGetAxisLastError (API)
ECAT_McGetAxisDriveError (API)
ECAT_McGetAxisActualPos (API)
ECAT_McGetAxisActualVel (API)
ECAT_McGetAxisDI (API)
ECAT_McGetProfileData (API)
Get Axis State
ECAT_McGetAxisState (API)
Standstill
Yes
No
Axis Motion Control
ECAT_McAxisMove... (API)
Axis Stop
ECAT_McAxisStop (API)
ECAT_McAxisQuickStop (API)
Discrete Motion
Continuous Motion
Synchronized Motion
No
Stopping
Get Axis State
ECAT_McGetAxisState (API)
Standstill
No
Errot Stop
Yes
Yes
Axis Error
Process
End

Figure 4.4

4.3.3. Axis Homing

As shown in Figure 4.5, before starting homing of an axis, parameters such as the home method, home speed, home acceleration, home offset and so on must be set. In single-axis motion control, *McGetAxisState* function can be called to get axis state. If the state is Standstill, that axis is currently stopped and ready to receive a new motion command. After successfully calling homing function, the axis status changes from the Standstill to the Homing. It indicates the axis is homing now.

The user can call the stop function to stop the axis homing. When the axis state changes from Homing to the Standstill, a new motion command can be issued. If any error occurs while homing, the state of the axis will be changed from Homing to ErrorStop. In this state, users need to deal with this error.

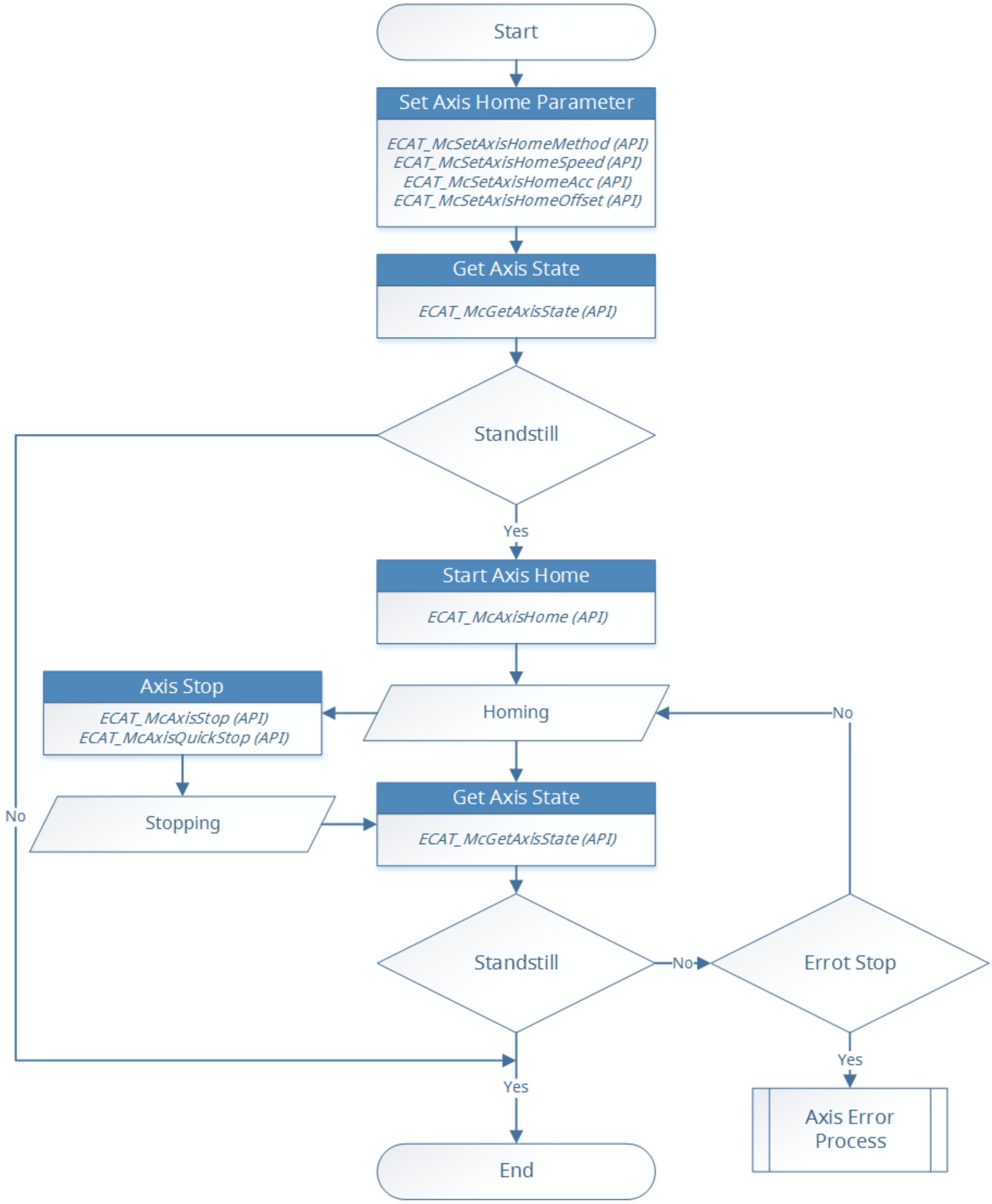
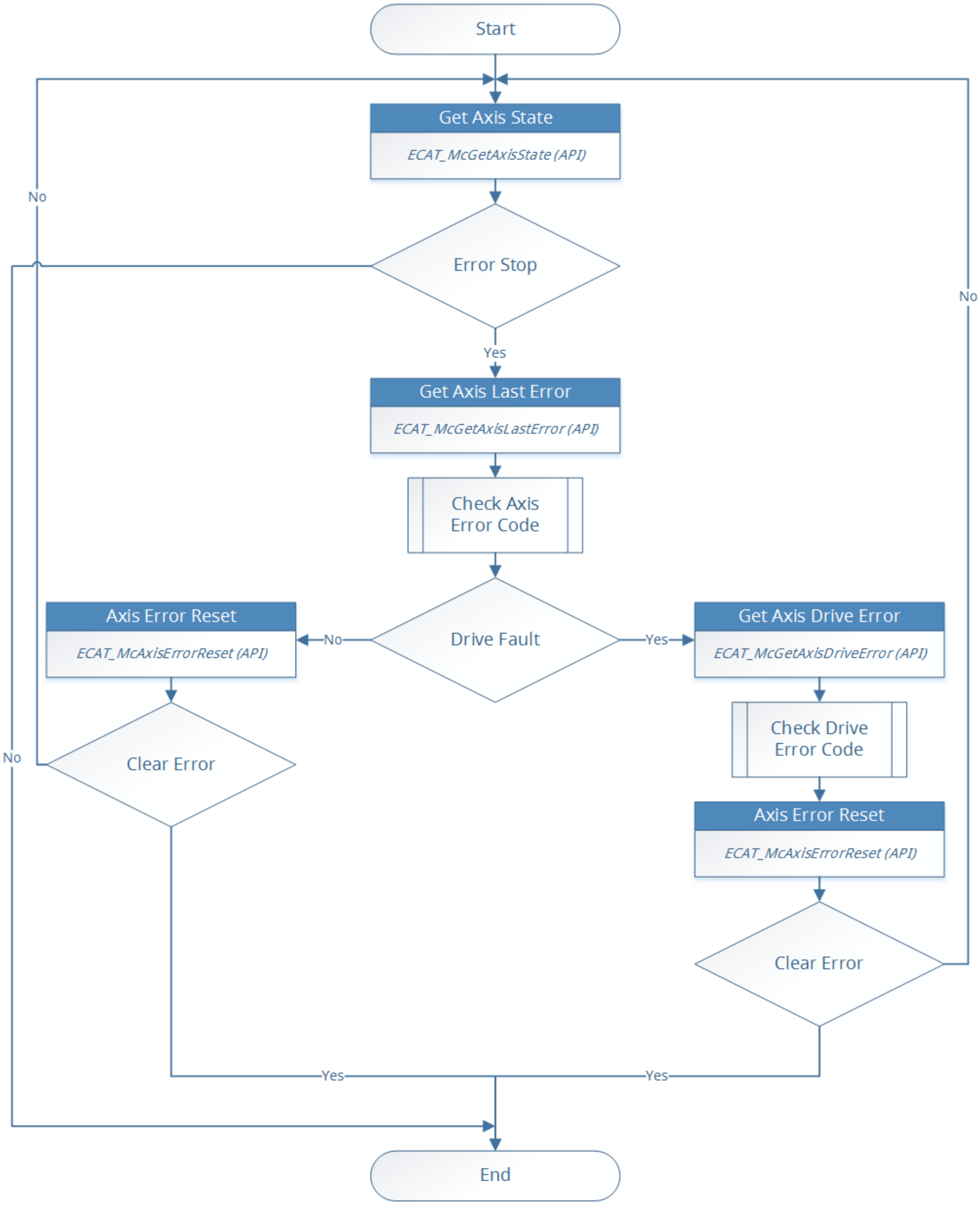

Start
Set Axis Home Parameter
ECAT_McSetAxisHomeMethod (API)
ECAT_McSetAxisHomeSpeed (API)
ECAT_McSetAxisHomeAcc (API)
ECAT_McSetAxisHomeOffset (API)
Get Axis State
ECAT_McGetAxisState (API)
Standstill
Yes
No
Start Axis Home
ECAT_McAxisHome (API)
Axis Stop
ECAT_McAxisStop (API)
ECAT_McAxisQuickStop (API)
Homing
No
Stopping
Get Axis State
ECAT_McGetAxisState (API)
Standstill
No
Errot Stop
Yes
Yes
Axis Error
Process
End

Figure 4.5

4.3.4. Axis Error Process

As shown in Figrue 4.6, when the axis state is ErrorStop, *McGetAxisLastError* function can be used to get the error code. From the error code, the cause of error can be determined. The error handling includes two parts: (1) If the error is not a servo drive error, the user can call *McAxisErrorReset* to clear the error. The axis state will be changed from ErrorStop to Standstill. (2) If the error comes from a servo drive, *McGetAxisDriveError* function can be called to get the drive's error code, and then call *McAxisErrorReset* to clear its error. Some servo drive errors can be cleared by the reset command; but some can not. If the reset command does not change the axis state back to Standstill, please restart (turn the power off than on) the servo drive to clear the its error.


Start
Get Axis State
ECAT_McGetAxisState (API)
No
Error Stop
No
Yes
Get Axis Last Error
ECAT_McGetAxisLastError (API)
Check Axis
Error Code
Axis Error Reset
ECAT_McAxisErrorReset (API)
No
Drive Fault
Yes
Get Axis Drive Error
ECAT_McGetAxisDriveError (API)
Check Drive
Error Code
No
Clear Error
Axis Error Reset
ECAT_McAxisErrorReset (API)
Clear Error
Yes
Yes
End

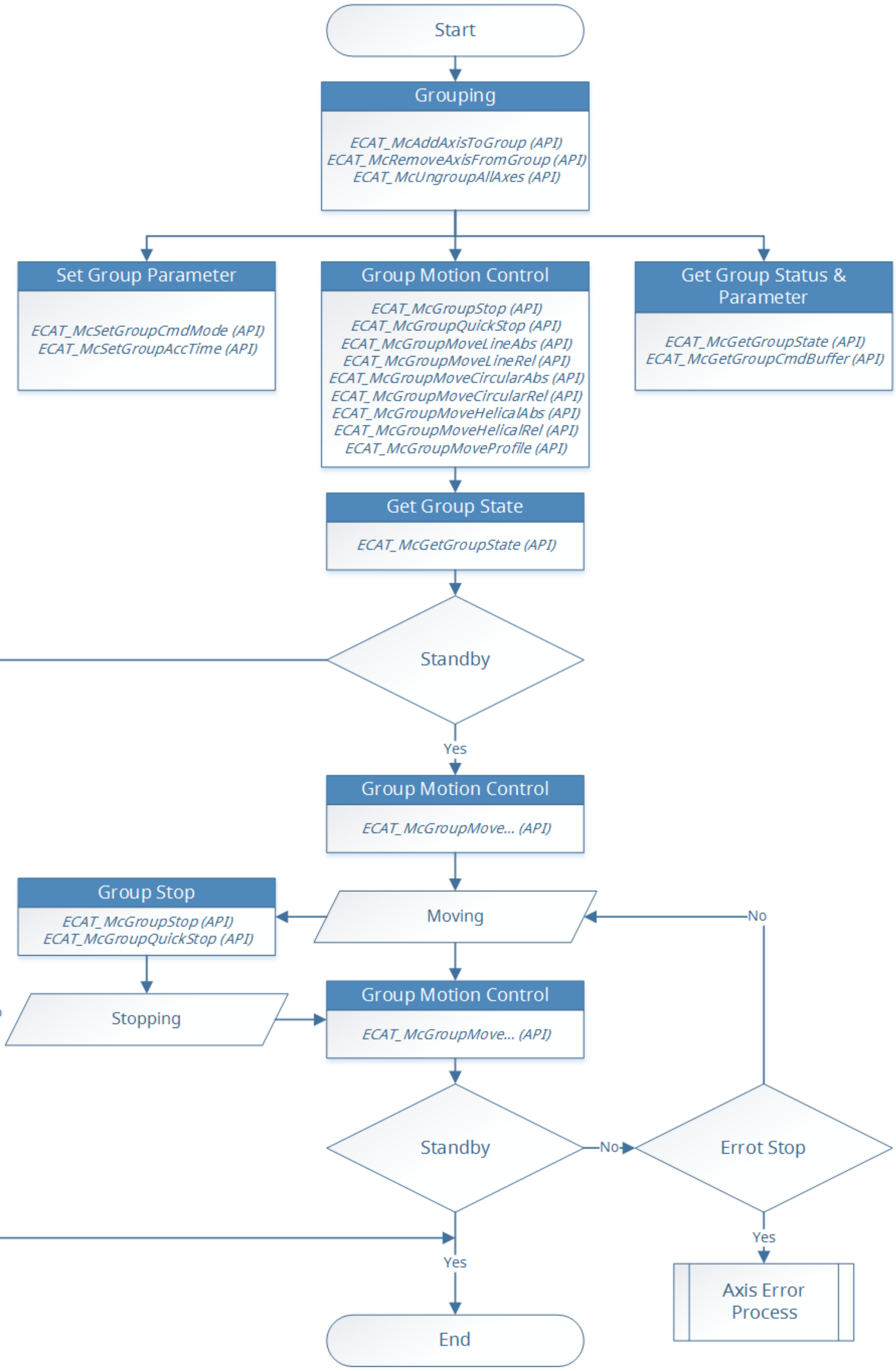
Figrue 4.6

4.3.5. Group Moving

The user can use the group motion control to do the multu-axis interpolation motion. Before using the group motion, the user needs to create a group and add axes to it. *McAddAxisToGroup* funciton can add an axis to the specified group; *McRemoveAxis* function can remove an axis from the specified group; *McUngroupAllAxes* function can remove all axes from the specified group. After a group is created and has enough axes to do some multi-axis motion, users can use group motion commands to do applications, as shown in Figrue 4.7.

McGetGroupState function can get the state of a group. If the state is Standby, the group motion is currently stopped. Users can issue a new motion command. Immediately after a motion function is successfully called, the group state changes from Standby to Moving.

Users can call stop functions to stop the group motion. When the stop command is completed, group state will change from Moving to Standby. In Standby state, the group is ready for executing another motion command. If any error occurs while moving, the state of that group will change from Moving to ErrorStop. In this state, users have to deal with this error.

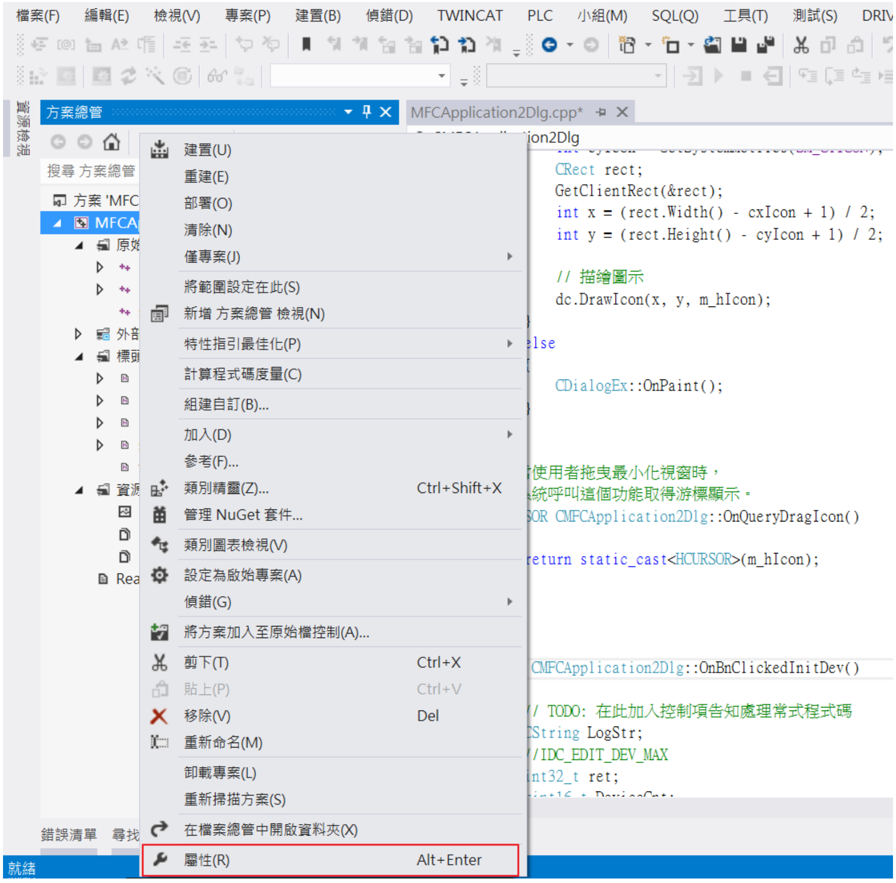

Start
Grouping
ECAT_McAddAxisToGroup (API)
ECAT_McRemoveAxisFromGroup (API)
ECAT_McUngroupAllAxes (API)
Set Group Parameter
ECAT_McSetGroupCmdMode (API)
ECAT_McSetGroupAccTime (API)
Group Motion Control
ECAT_McGroupStop (API)
ECAT_McGroupQuickStop (API)
ECAT_McGroupMoveLineAbs (API)
ECAT_McGroupMoveLineRel (API)
ECAT_McGroupMoveCircularAbs (API)
ECAT_McGroupMoveCircularRel (API)
ECAT_McGroupMoveHelicalAbs (API)
ECAT_McGroupMoveHelicalRel (API)
ECAT_McGroupMoveProfile (API)
Get Group Status & Parameter
ECAT_McGetGroupState (API)
ECAT_McGetGroupCmdBuffer (API)
Get Group State
ECAT_McGetGroupState (API)
Standby
Yes
No
Group Motion Control
ECAT_McGroupMove... (API)
Group Stop
ECAT_McGroupStop (API)
ECAT_McGroupQuickStop (API)
Moving
No
Stopping
Group Motion Control
ECAT_McGroupMove... (API)
Standby
No
Errot Stop
Yes
Yes
Axis Error Process
End

Figrue 4.7

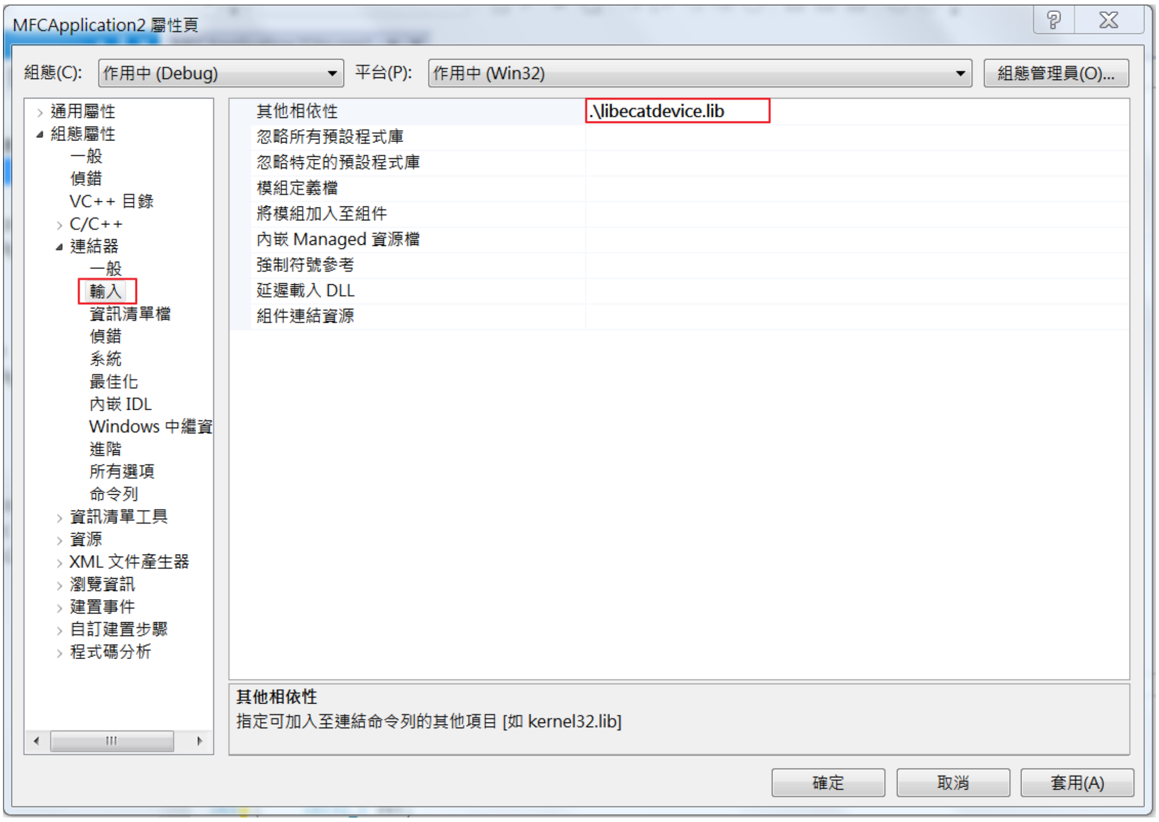
4.4. Use motion Library in Windows

4.4.1. For Visual Studio

1. Create a new project, Select **File->New->Project**.
2. Right-click the project node in Solution Explorer and choose **Properties** to open the property page dialog box.



3. Select **Configuration Properties->Linker->Input->Additional Dependencies**; enter **libecatdevice.lib** file in additional dependencies.


MFCApplication2 屬性頁
組態(C): 作用中 (Debug)
平台(P): 作用中 (Win32)
組態管理員(O)...
通用屬性
組態屬性
一般
偵錯
VC++ 目錄
C/C++
連結器
一般
輸入
資訊清單檔
偵錯
系統
最佳化
內嵌 IDL
Windows 中繼資
進階
所有選項
命令列
資訊清單工具
資源
XML 文件產生器
瀏覽資訊
建置事件
自訂建置步驟
程式碼分析
其他相依性
.\libecatdevice.lib
忽略所有預設程式庫
忽略特定的預設程式庫
模組定義檔
將模組加入至組件
內嵌 Managed 資源檔
強制符號參考
延遲載入 DLL
組件連結資源
其他相依性
指定可加入至連結命令列的其他項目 [如 kernel32.lib]
確定
取消
套用(A)

5. Device Operation Functions

5.1. ECAT_GetDeviceCnt

Description:

Get the number of available devices.

Syntax:

int32_t ECAT_GetDeviceCnt (uint16_t *DeviceCnt, uint8_t CardID[])

Parameters:

Name	Type	IN or OUT	Description
DeviceCnt	uint16_t	OUT	number of available devices
CardID	uint8_t *	OUT	Card ID of each device

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceCnt;
uint8_t CardID[CARD_DEVICE_NO_MAX];
ret = ECAT_GetDeviceCnt(&DeviceCnt, CardID);
if(ret < 0)
{
	printf("Failed to get device count:%d\n", ret);
}
else
{
	printf("Device Count%u \n", DeviceCnt);
}
```


5.2. ECAT_OpenDevice

Description:

Open a device with the specified Card ID.

Syntax:

int32_t ECAT_OpenDevice(uint16_t DeviceNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
ret = ECAT_OpenDevice(DeviceNo);
if(ret < 0)
{
	printf("Failed to open device:%d\n", ret);
}
else
{
	printf("Open device successfully! \n");
}
```

5.3. ECAT_CloseDevice

Description:

Close a device (card) with the specified Card ID.

Syntax:

int32_t ECAT_CloseDevice(uint16_t DeviceNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_CloseDevice(DeviceNo);
if(ret < 0)
{
	printf("Failed to close device:%d\n", ret);
}
else
{
	printf("Close device successfully! \n");
}
```

5.4. ECAT_GetDeviceSerialNo

Description:

Get the hardware serial number.

Syntax:

int32_t ECAT_GetDeviceSerialNo(uint16_t DeviceNo, uint8_t *SerialNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SerialNo	uint8_t *	IN	Hardware serial number (array size is 8 Bytes)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint8_t SerialNo[8];
ret = ECAT_GetDeviceSerialNo(DeviceNo, SerialNo);
if(ret < 0)
{
	printf("Failed to get device serial No.:%d\n", ret);
}
else
{
	printf("serial number = %x %x %x %x %x %x %x %x\n",
		SerialNo[0],SerialNo[1],SerialNo[2],SerialNo[3],
		SerialNo[4],SerialNo[5],SerialNo[6],SerialNo[7]);
}
```

5.5. ECAT_GetDeviceDI

Description:

Get the on-board digital input data of the specified device. These digital inputs has nothing to do with EtherCAT bus.

Syntax:

int32_t ECAT_GetDeviceDI(uint16_t DeviceNo, uint32_t *Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Value	uint32*	OUT	Digital input data (only lower 13 bits are available)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_GetDeviceDI(DeviceNo, &Value);
if(ret < 0)
{
        printf("Failed to get device DI:%d\n", ret);
}
else
{
        printf("DI:%u! \n", Value);
}
```

5.6. ECAT_GetDeviceDIBit

Description:

Get a bit state of a device's on-board digital input.

Syntax:

int32_t ECAT_GetDeviceDIBit(uint16_t DeviceNo, uint16_t BitNo, uint32_t *Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
BitNo	uint16	IN	Bit number (0 ~ 12)
Value	uint32*	OUT	Bit data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value;
uint16_t BitNo = 2;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_GetDeviceDIBit(DeviceNo, BitNo, &Value);
if(ret < 0)
{
    printf("Failed to get device DI:%d\n", ret);
}
else
{
    printf("DI_Bit[%u]:%u! \n", BitNo, Value);
}
```


5.7. ECAT_GetDeviceDO

Description:

Get the on-board digital output data of a specified device.

Syntax:

int32_t ECAT_GetDeviceDO(uint16_t DeviceNo, uint32_t *Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Value	uint32*	OUT	Digital output data (only lower 13 bits are available)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_GetDeviceDO(DeviceNo, &Value);
if(ret < 0)
{
        printf("Failed to get device DO:%d\n", ret);
}
else
{
        printf("DO:%u! \n", Value);
}
```

5.8. ECAT_GetDeviceDOBit

Description:

Get a bit state of a device's on-board digital output.

Syntax:

int32_t ECAT_GetDeviceDOBit(uint16_t DeviceNo, uint16_t BitNo, uint32_t *Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
BitNo	uint16	IN	Bit number (0 ~ 12)
Value	uint32*	OUT	Bit data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value;
uint16_t BitNo = 2;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_GetDeviceDOBit(DeviceNo, BitNo, &Value);
if(ret < 0)
{
	printf("Failed to get device DO:%d\n", ret);
}
else
{
	printf("DO_Bit[%u]:%u! \n", BitNo,Value);
}
```

5.9. ECAT_SetDeviceDO

Description:

Set the on-board digital output data of a device.

Syntax:

int32_t ECAT_SetDeviceDO(uint16_t DeviceNo, uint32_t Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Value	uint32	OUT	Digital output data (only lower 13 bits are available)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value = 0x000F;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_SetDeviceDO(DeviceNo, Value);
if(ret < 0)
{
	printf("Failed to set device DO:%d\n", ret);
}
else
{
	printf("DO:%u! \n", Value);
}
```

5.10. ECAT_SetDeviceDOBit

Description:

Set a bit data of a device's on-board digital output.

Syntax:

int32_t ECAT_SetDeviceDOBit(uint16_t DeviceNo, uint16_t BitNo, uint32_t Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
BitNo	uint16	IN	Bit number (0 ~ 12)
Value	uint32*	OUT	Bit data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value = 1;
uint16_t BitNo = 2;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_SetDeviceDOBit(DeviceNo, BitNo, Value);
if(ret < 0)
{
    printf("Failed to set device DO:%d\n", ret);
}
else
{
    printf("DO_Bit[%u]:%u! \n", BitNo, Value);
}
```


5.11. ECAT_SetDeviceEncProperty

Description:

Set the on-board encoder mode of a device.

Syntax:

int32_t ECAT_SetDeviceEncProperty(uint16_t DeviceNo, uint16_t EncNo, uint8_t Mode, uint8_t InvertCnt, uint8_t LPF)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
EncNo	uint16_t	IN	Encoder interface channel number (0 ~ 1)
Mode	uint8_t	IN	Encoder Mode 1: CW/CCW 2: Pulse/Dir 3: A/B Phase
InvertCnt	uint8_t	IN	Invert the counting direction
LPF	uint8_t	IN	Low pass filter (As shown in Table 5.1)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 5.1: Low Pass Filter Definition

Macro Definition	Value	Description
DEV_ENC_LPF_4_MHZ	0	4MHz
DEV_ENC_LPF_3P6_MHZ	1	3.6MHz
DEV_ENC_LPF_1P8_MHZ	2	1.8MHz
DEV_ENC_LPF_950_KHZ	4	950KHz
DEV_ENC_LPF_480_KHZ	8	480KHz
DEV_ENC_LPF_240_KHZ	16	240KHz
DEV_ENC_LPF_120_KHZ	32	120KHz
DEV_ENC_LPF_60_KHZ	64	60KHz
DEV_ENC_LPF_30_KHZ	128	30KHz

Example:

[C/C++]

```
int32_t ret;
uint16_t EncNo = 0;
uint8_t Mode = 3; //A/B Phase
uint8_t InvertCnt = 1; //Enable Reverse
uint8_t LPF = 0;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_SetDeviceEncProperty(DeviceNo, EncNo, Mode, InvertCnt, LPF);
if(ret != 0)
{
    printf("Failed to set encoder mode:%d\n", ret);
}
else
{
    printf("Set encoder mode successfully! \n");
}
```


5.12. ECAT_GetDeviceEncProperty

Description:

Get the on-board encoder mode of a device.

Syntax:

int32_t ECAT_GetDeviceEncProperty(uint16_t DeviceNo, uint16_t EncNo, uint8_t *Mode, uint8_t * InvertCnt, uint8_t *LPF)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
EncNo	uint16_t	IN	Encoder interface channel number (0 ~ 1)
Mode	uint8_t	OUT	Encoder Mode 1: CW/CCW 2: Pulse/Dir 3: A/B Phase
InvertCnt	uint8_t	OUT	Invert the counting direction
LPF	uint8_t	OUT	Low pass filter (As shown in Table 5.1)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t EncNo = 0;
uint8_t Mode;
uint8_t InvertCnt;
uint8_t LPF;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_GetDeviceEncProperty(DeviceNo, EncNo, &Mode, & InvertCnt, &LPF);
if(ret != 0)
{
	printf("Failed to get encoder mode:%d\n", ret);
}
else
{
	printf("Encoder mode:%u\n", Mode);
	printf("Encoder reverse:%u\n", ReverseCnt);
	printf("Encoder Low Pass Filter:%u\n", LPF);
}
```


5.13. ECAT_GetDeviceEncCount

Description:

Get a on board encoder counter value of a device.

Syntax:

int32_t ECAT_GetDeviceEncCount(uint16_t DeviceNo, uint16_t EncNo, int32_t *Cnt)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
EncNo	uint16_t	IN	Encoder interface channel number (0 ~ 1)
Cnt	int32_t *	OUT	Encoder counter value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t EncNo = 0;
int32_t Cnt;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_GetDeviceEncCount(DeviceNo, EncNo, &Cnt);
if(ret != 0)
{
    printf("Failed to get encoder count:%d\n", ret);
}
else
{
    printf("Encoder count:%d\n", Cnt);
}
```

5.14. ECAT_ResetDeviceEncCount

Description:

Clear a on-board encoder counter value of a device.

Syntax:

int32_t ECAT_ResetDeviceEncCount(uint16_t DeviceNo, uint16_t EncNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
EncNo	uint16_t	IN	Encoder interface channel number (0 ~ 1)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t EncNo = 0;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_ResetDeviceEncCount(DeviceNo, EncNo);
if(ret != 0)
{
	printf("Failed to clear encoder count:%d\n", ret);
}
else
{
	printf("Clear encoder count successfully!\n");
}
```


5.15. ECAT_SetDeviceCmpTrigProperty

Description:

Set the on-board device compare-trigger related properties.

Syntax:

int32_t ECAT_SetDeviceCmpTrigProperty(uint16_t DeviceNo, uint16_t EncNo, uint32_t PulseWidth)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
EncNo	uint16_t	IN	Encoder interface channel number (0 ~ 1)
PulseWidth	uint32_t	IN	Output Pulse width setting, the unit is 0.016us, and the maximum value is 0x7fffffff x 0.016us

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t EncNo = 0;
uint32_t PulseWidth = 100000;
ret = ECAT_SetDeviceCmpTrigProperty(DeviceNo, EncNo, PulseWidth);
if(ret != 0)
        printf("Failed to set compare triger property:%d\n", ret);
```

5.16. ECAT_GetDeviceCmpTrigProperty

Description:

Get the on-board device compare-trigger related properties.

Syntax:

int32_t ECAT_GetDeviceCmpTrigProperty(uint16_t DeviceNo, uint16_t EncNo, uint32_t *PulseWidth)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
EncNo	uint16_t	IN	Encoder interface channel number (0 ~ 1)
PulseWidth	uint32_t *	OUT	Output Pulse width setting value, the unit is 0.016us, and the maximum value is 0x7fffffff x 0.016us

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t EncNo = 0;
uint32_t PulseWidth;
ret = ECAT_GetDeviceCmpTrigProperty(DeviceNo, EncNo, &PulseWidth);
if(ret != 0)
{
	printf("Failed to get compare triger property:%d\n", ret);
}
else
{
	printf("Compare triger pulse width:%u\n", PulseWidth);
}
```


5.17. ECAT_SetDeviceCmpTrigData

Description:

According to the setting value, start a single compare-trigger function for a on-board encoder interface channel.

Syntax:

int32_t ECAT_SetDeviceCmpTrigData(uint16_t DeviceNo, uint16_t EncNo, int32_t CmpData)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
EncNo	uint16_t	IN	Encoder interface chaneel number (0 ~ 1)
CmpData	int32_t	IN	Single compare-trigger data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EncNo = 0;
int32_t CmpData = 1000;
ret = ECAT_SetDeviceCmpTrigData(DeviceNo, EncNo, CmpData);
if(ret != 0)
        printf("Failed to set compare triger data:%d\n",ret);
```

5.18. ECAT_SetDeviceContCmpTrigData

Description:

Start a continuous or a multiple compare-trigger function.

Syntax:

int32_t ECAT_SetDeviceContCmpTrigData(uint16_t DeviceNo, uint16_t EncNo, int32_t Start, uint32_t Interval, uint32_t Times)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
EncNo	uint16_t	IN	Encoder interface channel number (0 ~ 1)
Start	int32_t	IN	Start position for this compare-trigger operation
Interval	uint32_t	IN	Trigger interval (i.e. position increment)
Times	uint32_t	IN	Set 0 for continuous compare-trigger; a number greater than 0 is the number for multiple compare -trigger actions

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EncNo = 0;
int32_t Start = 1000;
uint32_t Interval = 200;
uint32_t Times = 10;
ret = ECAT_SetDeviceContCmpTrigData(DeviceNo, EncNo, Start, Interval, Times);
if(ret != 0)
    printf("Failed to set continus compare triger data:%d\n", ret);
```


5.19. ECAT_SetDeviceEmg

Description:

Set the device emergency stop signal related configurations.

Syntax:

int32_t ECAT_SetDeviceEmg(uint16_t DeviceNo, uint8_t Source, uint8_t Enable, uint8_t Logic, uint16_t SlaveNo, uint8_t ServoOff)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
Source	uint8_t	IN	Emergency stop signal source (As show in Table 5.2)
Enable	uint8_t	IN	Enable/ Disable emergency stop
Logic	uint8_t	IN	Emergency stop signal logic level 0: Low 1: High
SlaveNo	uint16_t	IN	Slave number
BitNo	uint16_t	IN	Bit number
ServoOff	uint8_t	IN	Servo Off when emergency stop triggered 0: N 1: Y

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 5.2: Emergency stop signal source

Macro Definition	Value	Description
DEV_EMG_SOURCE_OB_DI	0	On-Board DI
DEV_EMG_SOURCE_SLAVE_DI	1	Slave DI

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint8_t Source, Logic, Enable, ServoOff;
uint16_t SlaveNo, BitNo;
/* On board DI settings*/
Source = DEV_EMG_SOURCE_OB_DI;
Logic = 0; // Low active
Enable = 1;
ServoOff = 1;
ret = ECAT_SetDeviceEmg(DeviceNo, Source, Enable, Logic, 0,   0, ServoOff)
if(ret != 0)
    printf("Failed to enable emergency:%d\n",ret);
/* Slave DI settings*/
Source = DEV_EMG_SOURCE_SLAVE_DI;
Logic = 0; // Low active
Enable = 1;
SlaveNo = 0;
BitNo = 1;
ret = ECAT_SetDeviceEmg(DeviceNo, Source, Enable, Logic, SlaveNo, BitNo, ServoOff)
if(ret != 0)
    printf("Failed to enable emergency:%d\n",ret);
```


5.20. ECAT_GetDeviceEmg

Description:

Get the configurations of the device emergency stop signal.

Syntax:

int32_t ECAT_GetDeviceEmg(uint16_t DeviceNo, uint8_t *Source, uint8_t *Enable, uint8_t *Logic, uint16_t *SlaveNo, uint16_t *BitNo, uint8_t *ServoOff)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Source	uint8_t *	OUT	Emergency stop signal source. 0: On board DI 1: Slave DI (Please refer to Table 5.2)
Enable	uint8_t *	OUT	Enable / Disable emergency stop
Logic	uint8_t *	OUT	Emergency stop signal logic level 0: Low 1: High
SlaveNo	uint16_t *	OUT	Slave number
BitNo	uint16_t *	OUT	Bit number
ServoOff	uint8_t *	OUT	Servo Off when emergency stop triggered 0: N 1: Y

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint8_t Source, Logic, Enable, ServoOff;
uint16_t SlaveNo, BitNo;

ret = ECAT_GetDeviceEmg(DeviceNo, &Source, &Enable, &Logic, &SlaveNo, &BitNo, &ServoOff)
if(ret != 0)
    printf("Failed to get emergency settings:%d\n", ret);
else{
    printf("Emergency source:%d\n", Source);
    printf("Emergency enable:%d\n", Enable);
    printf("Emergency logic:%d\n", Logic);
    printf("Emergency SlaveNo:%d\n", SlaveNo);
    printf("Emergency BitNo:%d\n", BitNo);
    printf("Emergency ServoOff:%d\n", ServoOff);
}
```


5.21. ECAT_GetDeviceEmgStatus

Description:

Get emergency stop signal status.

Syntax:

int32_t ECAT_GetDeviceEmgStatus(uint16_t DeviceNo, uint8_t *Status)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
Status	uint8_t *	OUT	Emergency stop signal status

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint8_t Status;

ret = ECAT_GetDeviceEmgStatus(DeviceNo, &Status)
if(ret != 0)
	printf("Failed to get emergency status:%d\n", ret);
else
	printf("Emergency Status:%d\n", Status);
```


5.22. ECAT_SetDeviceEmgSoftSig

Description:

Use this function to produce an emergency stop.

Syntax:

int32_t ECAT_SetDeviceEmgSoftSig (uint16_t DeviceNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint8_t Source, Logic, Enable, ServoOff;
uint16_t SlaveNo, BitNo;
/* On board DI settings*/
Source = DEV_EMG_SOURCE_OB_DI;
Logic = 0; // Low active
Enable = 1;
ServoOff = 0;
ret = ECAT_SetDeviceEmg(DeviceNo, Source, Enable, Logic, 0, 0, ServoOff)
if(ret != 0)
	printf("Failed to enable emergency:%d\n", ret);

ret = ECAT_SetDeviceEmgSoftSig (DeviceNo)
if(ret != 0)
	printf("Failed to set emergency software signal:%d\n", ret);
```

5.23. ECAT_SetDeviceMPG

Description:

Configure device local I/O into a manual pulse generator. The MPG pin definitions are shown in Table 5.3 and Table 5.4. Up to 7 axes can be defined for control, and they are labeled as X, Y, Z, 4, 5, 6, and 7. Three multipliers are defined here: x1, x10, and x100. Encoder interface are defined in Table 5.4.

Syntax:

int32_t ECAT_SetDeviceMPG(uint16_t DeviceNo, uint8_t Enable, uint16_t *AxisNo, uint16_t AxisCount)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Enable	uint8_t	IN	Enable/Disable MPG function 0: Disable 1: Enable
AxisNo	uint16_t *	IN	A pointer points to an axis number array. Axes under this MPG control are listed here. Axis numbers are assigned when users use utility program to configure servo drives as axes for this EtherCAT control system.
AxisCount	uint16_t	IN	Size of this axis number array

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 5.3: CON1 connector for MPG signal pin definitions

Pin Number	Pin Assignment	MPG Signal	Pin Number	Pin Assignment	MPG Signal
1	DI0	X	8	DI7	x1
2	DI1	Y	19	DI8	x10
3	DI2	Z	20	DI9	x100
4	DI3	4	9	EXT. GNC	0V
5	DI4	5	18	EXT. PWR	+24V
6	DI5	6			
7	DI6	7			

Table 5.4: CON2 connector MPG pin definitions

Pin Number	Pin Assignment	MPG Signal
1	1A-	$\overline{A}$
6	1A+	A
2	1B-	$\overline{B}$
7	1B+	B

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint8_t Enable = 1;
uint16_t AxisNo[4];
uint16_t AxisCount = 4;
AxisNo[0] = 0;
```



```
AxisNo[1] = 1;
AxisNo[2] = 2;
AxisNo[3] = 3;

ret = ECAT_SetDeviceMPG(DeviceNo, Enable, AxisNo, AxisCount);
if (ret != 0)
{
	printf("Failed to set device MPG:%d\n", ret);
}
```

5.24. ECAT_GetDeviceMPG

Description:

Get the manual pulse generator (MPG) configuration of this device (card). The MPG pin definitions are shown in Table 5.3 and Table 5.4. Up to 7 axes can be defined for control, and they are labeled as X, Y, Z, 4, 5, 6, and 7. Three multipliers are defined here: x1, x10, and x100. Encoder interface are defined in Table 5.4.

Syntax:

int32_t ECAT_GetDeviceMPG(uint16_t DeviceNo, uint8_t *Enable, uint16_t *AxisNo, uint16_t *AxisCount)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Enable	uint8_t *	OUT	Enable/Disable MPG function 0: Disable 1: Enable
AxisNo	uint16_t *	OUT	A pointer points to an axis number array. Axes under this MPG control are listed here. Axis numbers are assigned when users use utility program to configure servo drives as axes for this EtherCAT control system.
AxisCount	uint16_t *	OUT	Size of this axis number array

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint8_t Enable;
uint16_t AxisNo[4];
uint16_t i, AxisCount;

ret = ECAT_GetDeviceMPG(DeviceNo, &Enable, AxisNo, &AxisCount);
if (ret != 0){
	printf("Failed to get device MPG:%d\n",ret);
}
else{
	printf("MPG enable:%d\n",Enable);
	for (i = 0; i < AxisCount; i++)
		printf("MPG axis number[%d]:%d\n", i, AxisNo[i]);
}
```


5.25. ECAT_GetDeviceState

Description:

Get the EtherCAT network status of a device. This function is always called for checking if the system is running normally.

Syntax:

int32_t ECAT_GetDeviceState(uint16_t DeviceNo, uint32_t *LinkUp, uint32_t *SlavesResp, uint32_t *AlState, uint32_t *Wc)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
LinkUp	uint32_t*	OUT	Network link status of Ethernet (EtherCAT) 0: Link Down 1: Link Up
SlavesResp	uint32_t*	OUT	Sum of responding slaves on this EtherCAT network system
AlState	uint32_t*	OUT	AL state of EtherCAT master. AL states are defined shown in Table 5.5.
Wc	uint32_t*	OUT	EtherCAT working counter value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 5.5: EtherCAT AL states

Macro Definition	Value	Description
ECAT_AS_INIT	0x00	Init
ECAT_AS_PREOP	0x02	Pre-Operational
ECAT_AS_SAFEOP	0x04	Safe-Operational
ECAT_AS_OP	0x08	Operational

Example:

[C/C++]

```
int32_t ret;
char buffer[1024];
char StrAlState[255];
uint32_t DeviceNo = 0;
uint32_t LinkUp, SlavesResp, AlState, Wc;
ret = ECAT_OpenDevice(DeviceNo);
ret = ECAT_GetDeviceState(DeviceNo, &LinkUp, &SlavesResp, &AlState, &Wc);
if(ret < 0)
	printf("Failed to get device state:%d\n", ret);
else
{
	if(AlState == ECAT_AS_INIT)
		sprintf(StrAlState,"INIT");
	else if(AlState == ECAT_AS_PREOP)
		sprintf(StrAlState,"PREOP");
	else if(AlState == ECAT_AS_SAFEOP)
		sprintf(StrAlState,"SAFEOP");
	else if(AlState == ECAT_AS_OP)
		sprintf(StrAlState,"OP");
	else
		sprintf(StrAlState,"Invalid");
	sprintf(buffer,"Slave(s):%u | AL State:%s | Link is :%s | Wc:%-u "
				, SlavesResp, StrAlState, LinkUp? "up" : "down", Wc);
	printf("%s\n", buffer);
}
```


5.26. ECAT_StartDeviceOpTask

Description:

Start the device EtherCAT operation task. At least one network informaiton must be pre-loaded into this card. This configuration file is used for checking the real system is the same as configured one. This function takes some time to finish. Most of the motion funcitons can be only called when the system goes into OP state. After this function is called, users must further use function *ECAT_GetDeviceState* to check if this operation finishes successfully.

Syntax:

int32_t ECAT_StartDeviceOpTask(uint16_t DeviceNo, uint16_t NetworkInfoNo, uint8_t EnumCycleTime, uint64_t EnableDC, uint32_t WcErrCnt)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
NetworkInfoNo	uint16_t	IN	Network information number (Configured by the EtherCAT utility)
EnumCycleTime	uint8_t	IN	Cycle time number (As show in Table 5.6)
EnableDC	uint8_t	IN	DC mode is enabled of each slave 0: Disable 1: Enable
WcErrCnt	uint32_t	IN	Working counter error count

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 5.6: Cycle time number

Macro Definition	Value	Description
DEV_OP_CYCLE_TIME_1MS	0	1ms
DEV_OP_CYCLE_TIME_2MS	1	2ms
DEV_OP_CYCLE_TIME_3MS	2	3ms
DEV_OP_CYCLE_TIME_4MS	3	4ms
DEV_OP_CYCLE_TIME_5MS	4	5ms
DEV_OP_CYCLE_TIME_6MS	5	6ms
DEV_OP_CYCLE_TIME_7MS	6	7ms
DEV_OP_CYCLE_TIME_8MS	7	8ms
DEV_OP_CYCLE_TIME_9MS	8	9ms
DEV_OP_CYCLE_TIME_10MS	9	10ms
DEV_OP_CYCLE_TIME_11MS	10	11ms
DEV_OP_CYCLE_TIME_12MS	11	12ms
DEV_OP_CYCLE_TIME_13MS	12	13ms
DEV_OP_CYCLE_TIME_14MS	13	14ms
DEV_OP_CYCLE_TIME_15MS	14	15ms
DEV_OP_CYCLE_TIME_16MS	15	16ms

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo    = 0;
uint16_t NetworkInfoNo = 0;
uint8_t EnumCycleTime = DEV_OP_CYCLE_TIME_1MS;
uint8_t EnableDC = 1;
int32_t flag = 1;
ret = ECAT_OpenDevice(DeviceNo);

ret = ECAT_StartDeviceOpTask(DeviceNo, NetworkInfoNo, EnumCycleTime, EnableDC);
if(ret < 0)
```



```
{
    printf("Failed to start device op task:%d\n",ret);
}
else
{
    printf("Start device op task successfully! \n");
}
```

5.27. ECAT_StopDeviceOpTask

Description:

Stop the device EtherCAT operation task.

Syntax:

int32_t ECAT_StopDeviceOpTask(uint16_t DeviceNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo     = 0;
uint16_t NetworkInfoNo = 0;
...
ret = ECAT_StopDeviceOpTask(DeviceNo);
if(ret < 0)
{
        printf("Failed to stop device op task:%d\n",ret);
}
else
{
        printf("stop device op task successfully! \n");
}
```


5.28. ECAT_SetTimer

Description:

Set Timer Interval. This timer is only available when system is in OP state. In OP state, there is a cyclic task running inside ECAT_M801. This card takes advantage of this cyclic task to provide the host PC a timer function. *ECAT_SetTimer* function configures its time interval. A companion function *ECAT_WaitforTimer* is the one to wait for this interval. A thread calls *ECAT_WaitforTimer* will be suspended until time up.

Syntax:

int32_t ECAT_SetTimer(uint16_t DeviceNo, uint32_t Interval)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Interval	uint32_t	IN	Time Interval, unit: Cycle Time

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo    = 0;
uint32_t Interval = 100;

ret = ECAT_SetTimer(DeviceNo, Interval);
if(ret < 0)
{
      printf("Failed to Set Timer:%d\n", ret);
}
else
{
      printf("Set Timer successfully! \n");
}
while(1)
{
      ECAT_WaitforTimer(DeviceNo);
      //do something …
}
```


5.29. ECAT_ SetTimerStop

Description:

Disable Timer. This timer is only available when system is in OP state. In OP state, there is a cyclic task running inside ECAT_M801. This card takes advantage of this cyclic task to provide the host PC a timer function. *ECAT_SetTimer* function configures its time interval. A companion function *ECAT_WaitforTimer* is the one to wait for this interval. A thread calls *ECAT_WaitforTimer* will be suspended until time up.

Syntax:

int32_t ECAT_ SetTimerStop(DeviceNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo    = 0;
ret = ECAT_SetTimerStop(DeviceNo);
if(ret < 0)
{
        printf("Failed to Set Timer Stop:%d\n",ret);
}
else
{
        printf("Set Timer Stop successfully! \n");
}
```


5.30. ECAT_WaitforTimer

Description:

Wait until time up. This timer is only available when system is in OP state. In OP state, there is a cyclic task running inside ECAT_M801. This card takes advantage of this cyclic task to provide the host PC a timer function. *ECAT_SetTimer* function configures its time interval. A companion function *ECAT_WaitforTimer* is the one to wait for this interval. A thread calls *ECAT_WaitforTimer* will be suspended until time up.

Syntax:

int32_t ECAT_WaitforTimer(uint16_t DeviceNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo    = 0;
uint32_t Interval = 100;


ret = ECAT_SetTimer(DeviceNo, Interval);
if(ret < 0)
{
	printf("Failed to Set Timer:%d\n", ret);
}
else
{
	printf("Set Timer successfully! \n");
}
while(1)
{
	ret = ECAT_WaitforTimer(DeviceNo);
	if(ret == 0)
	{
		//do something…
	}
}
```


5.31. ECAT_GetProcessTime

Description:

Get the processing time of an EtherCAT communication cycle. This is an average time for successive 1000 cycles; the unit is in micro-second.

Warn: the processing time may change according to the quantity of slaves and the called APIs. It is better to keep this value under 50% of EtherCAT cycle time.

Syntax:

int32_t ECAT_GetProcessTime(uint16_t DeviceNo, double *Time)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Time	double*	OUT	Processing time of an EtherCAT cycle Unit: ms

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
double Time;
ret = ECAT_GetProcessTime(DeviceNo, &Time);
if(ret < 0)
{
    printf("Failed to get Process Time:%d\n", ret);
}
else
{
    printf("Process Time:%f \n", Time);
}
```

6. Slave Operation Functions

6.1. ECAT_GetSlaveInfo

Description:

Get slave information of a slave.

Syntax:

int32_t ECAT_GetSlaveInfo(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t *Alias, uint32_t *ProductCode, uint32_t *VendorID, uint32_t *RevisionNo, uint32_t *SerialNo, uint8_t *AlState, uint32_t *SlaveType)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
Alias	uint16_t*	OUT	Alias
ProductCode	uint32_t*	OUT	Product Code
VendorID	uint32_t*	OUT	Vendor ID
RevisionNo	uint32_t*	OUT	Revision number
SerialNo	uint32_t*	OUT	Serial number
AlState	uint8_t*	OUT	EtherCAT AL State of this slave
SlaveType	uint32_t*	OUT	Slave Type (As show in Table 6.1)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 6.1: Slave Type

Macro Definition	Value	Description
SLAVE_TYPE_GENERIC	0	Generic
SLAVE_TYPE_CiA402	1	CiA 402 drive
SLAVE_TYPE_STEPPER_MOTOR	2	Single Axis Stepper Motor (especially, ECAT-2091S)
SLAVE_TYPE_4_AXIS_STEPPER_MOTOR	3	4-Axis Stepper Motor (especially, ECAT-2094S)

Example:

[C/C++]

```
int32_t ret;
int16_t i;
uint16_t SlaveCnt;
uint16_t DeviceNo = 0;
uint16_t Alias;
uint32_t ProductCode, VendorID, RevisionNo, SerialNo, SlaveType;
...
ret = ECAT_OpenDevice(DeviceNo);
if(ret < 0)
{
	printf("Failed to open device:%d\n",ret);
}
else
{
	for(i=0;i<SlaveCnt;i++)
	{
		ret = ECAT_GetSlaveInfo(DeviceNo, i, &Alias, &ProductCode,
				&VendorID, &RevisionNo, &SerialNo, &SlaveType);
		if(ret < 0)
		{
			printf("Failed to get slave infomation:%d\n",ret);
		}
		else
```



```
            {
                printf("Slave(%u)-+\n"
                    "           |-ProductCode:0x%X\n"
                    "           |-VendorID:0x%X\n"
                    "           |-RevisionNo:0x%X\n"
                    "           |-SerialNo:0x%X\n"
                    "           |-SlaveType:%d\n"
                    "\n"
                    , i ,ProductCode, VendorID, RevisionNo, SerialNo, SlaveType);
            }
        }
    }
```

6.2. ECAT_GetSlaveSdoObject

Description:

Get SDO data from a slave. Read a data object by means of service data object communication.

Syntax:

int32_t ECAT_GetSlaveSdoObject(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t Index, uint8_t SubIndex, uint16_t DataSize, uint32_t *ObjectVal, uint32_t *AbortCode)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
Index	uint16_t	IN	Object index
SubIndex	uint8_t	IN	Object sub-index
DataSize	uint16_t	IN	Size of data
ObjectVal	uint32_t*	OUT	Data buffer (read-out data)
AbortCode	uint32_t*	OUT	Abort code of the SDO (Please refer to Appendix "SDO Abort messages")

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret ;
uint32_t DeviceNo = 0;
uint16_t i;
uint16_t SlaveNo = 0;
uint16_t OffsetByte, DataSize;
uint16_t Index = 0x607C; //Home Offset Index
uint8_t SubIndex = 0x00; //Home Offset SubIndex
uint16_t DataSize = 4; //4 byte
uint32_t ObjectVal = 0;

...
ret = ECAT_GetSlaveSdoObject(DeviceNo, SlaveNo, Index, SubIndex,
          DataSize, &ObjectVal, &AbortCode);
if(ret < 0)
{
       printf("Failed to get sdo object:%d\n", ret);
}
else
{
       printf("Get sdo object successfully!\n");
}
```

6.3. ECAT_SetSlaveSdoObject

Description:

Set SDO data to a slave. Write a data object by means of service data object communication.

Syntax:

int32_t ECAT_SetSlaveSdoObject(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t Index, uint8_t SubIndex, uint16_t DataSize, uint32_t ObjectVal, uint32_t *AbortCode)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
Index	uint16_t	IN	Object index
SubIndex	uint8_t	IN	Object sub-index
DataSize	uint16_t	IN	Size of data
ObjectVal	uint32_t	IN	Data buffer (data for writing)
AbortCode	uint32_t*	OUT	Abort code of the SDO (Please refer to Appendix "SDO Abort messages")

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret ;
uint32_t DeviceNo = 0;
uint16_t i;
uint16_t SlaveNo = 0;
uint16_t OffsetByte, DataSize;
uint16_t Index = 0x607C; //Home Offset Index
uint8_t SubIndex = 0x00; //Home Offset SubIndex
uint16_t DataSize = 4; //4 byte
uint32_t ObjectVal = 100;

...
ret = ECAT_SetSlaveSdoObject(DeviceNo, SlaveNo, Index, SubIndex,
          DataSize, ObjectVal, &AbortCode);
if(ret < 0)
{
       printf("Failed to set sdo object:%d\n",ret);
}
else
{
       printf("set sdo object successfully!\n");
}
```

6.4. ECAT_SetSlaveRxPdoData

Description:

Set RxPDO data to a slave. Trasnfer process data to the RxPDO of a slave by means of cyclic communication.

Syntax:

int32_t ECAT_SetSlaveRxPdoData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, uint8_t *Data)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
OffsetByte	uint16_t	IN	Byte offset
DataSize	uint16_t	IN	Size of data (RW_PDO_DATA_SIZE_MAX macro is the maximum size of the data. It is 512.)
Data	uint8_t*	IN	Data buffer

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret ;
uint32_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte, DataSize;
uint8_t Data[RW_PDO_DATA_SIZE_MAX];

//Example-ECAT-2055-32
OffsetByte = 0;
DataSize = 2; //2 bytes
Data[0] = 0xFF;
Data[1] = 0xAA;

...
ret = ECAT_SetSlaveRxPdoData(DeviceNo, SlaveNo, OffsetByte, DataSize, Data);
if(ret < 0)
{
	printf("Failed to set RxPdo data:%d\n", ret);
}
else
{
	printf("Set RxPdo data successfully!\n");
}
```


6.5. ECAT_GetSlaveRxPdoData

Description:

Get RxPDO data of a slave. Read process data from the RxPDO of a slave by means of cyclic communication.

Syntax:

int32_t ECAT_GetSlaveRxPdoData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, uint8_t *Data)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
OffsetByte	uint16_t	IN	Byte offset
DataSize	uint16_t	IN	Size of data (RW_PDO_DATA_SIZE_MAX macro is the maximum size of the data. It is 512.)
Data	uint8_t*	IN	Data buffer

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret ;
uint16_t i;
uint32_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte,DataSize;
uint8_t Data[RW_PDO_DATA_SIZE_MAX];

//Example-ECAT-2055-32
OffsetByte = 0;
DataSize = 2; //2 bytes


...
ret = ECAT_GetSlaveRxPdoData(DeviceNo, SlaveNo, OffsetByte, DataSize, Data);
if(ret < 0)
{
	printf("Failed to get RxPdo data:%d\n", ret);
}
else
{
	for(i=0;i<DataSize;i++)
	{
		printf("Data[%u]:0x%X\n", i, Data[i]);
	}
}
```

6.6. ECAT_GetSlaveTxPdoData

Description:

Get TxPDO data of a slave. Read process data from the TxPDO of a slave by means of cyclic communication.

Syntax:

int32_t ECAT_GetSlaveTxPdoData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, uint8_t *Data)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
OffsetByte	uint16_t	IN	Byte offset
DataSize	uint16_t	IN	Size of data (RW_PDO_DATA_SIZE_MAX macro is the maximum size of the data. It is 512.)
Data	uint8_t*	IN	Data buffer

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret ;
uint16_t i;
uint32_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte,DataSize;
uint8_t Data[RW_PDO_DATA_SIZE_MAX];

//Example-ECAT-2055-32
OffsetByte = 0;
DataSize = 2; //2 bytes


...
ret = ECAT_GetSlaveTxPdoData(DeviceNo, SlaveNo, OffsetByte, DataSize, Data);
if(ret < 0)
{
    printf("Failed to get TxPdo data:%d\n", ret);
}
else
{
    for(i=0;i<DataSize;i++)
    {
        printf("Data[%u]:0x%X\n", i, Data[i]);
    }
}
```


6.7. ECAT_GetSlaveDI

Description:

Get the digital input data of a slave. If a slave is a simple digital input slave, users can use this API to get DI values. Function *ECAT_GetSlaveTxPdoData* can also do it; but users have to enter more parameters for the same purpose.

Syntax:

int32_t ECAT_GetSlaveDI(uint16_t DeviceNo, uint16_t SlaveNo, uint32_t *Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
Value	uint32_t*	OUT	Digital input data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint32_t Value;

ret = ECAT_GetSlaveDI(DeviceNo, SlaveNo, &Value);
if(ret < 0)
	printf("Failed to get slave DI:%d\n", ret);
else
	printf("DI:%u! \n",Value);
```


6.8. ECAT_GetSlaveDIBit

Description:

Get a bit state of a slave's digital input.

Syntax:

int32_t ECAT_GetSlaveDIBit(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t BitNo, uint32_t *Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
BitNo	uint16_t	IN	Bit number
Value	uint32_t*	OUT	Bit data (0 or 1)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t BitNo = 2;
uint32_t Value;

ret = ECAT_GetSlaveDIBit(DeviceNo, SlaveNo, BitNo, &Value);
if(ret < 0)
	printf("Failed to get slave DI:%d\n", ret);
else
	printf("DI_Bit[%u]:%u! \n", BitNo, Value);
```


6.9. ECAT_GetSlaveDO

Description:

Get the digital output data of a slave. If a slave is a simple digital output slave, users can use this API to get DO states. Function *ECAT_GetSlaveRxPdoData* can also do it; but users have to enter more parameters for the same purpose.

Syntax:

int32_t ECAT_GetSlaveDO(uint16_t DeviceNo, uint16_t SlaveNo, uint32_t *Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
SlaveNo	uint16_t	IN	Slave number
Value	uint32_t*	OUT	Digital output data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint32_t Value;

ret = ECAT_GetSlaveDO(DeviceNo, SlaveNo, &Value);
if(ret < 0)
	printf("Failed to get slave DO:%d\n", ret);
else
	printf("DO:%u! \n",Value);
```


6.10. ECAT_GetSlaveDOBit

Description:

Get a bit state of a slave's digital output.

Syntax:

int32_t ECAT_GetSlaveDOBit(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t BitNo, uint32_t *Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
BitNo	uint16_t	IN	Bit number
Value	uint32_t*	OUT	Bit data (0 or 1)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t BitNo = 2;
uint32_t Value;

ret = ECAT_GetSlaveDOBit(DeviceNo, SlaveNo, BitNo, &Value);
if(ret < 0)
	printf("Failed to get slave DO bit:%d\n", ret);
else
	printf("DO_Bit[%u]:%u! \n", BitNo, Value);
```


6.11. ECAT_SetSlaveDO

Description:

Set the digital output data of a slave. If a slave is a simple digital input slave, users can use this API to set DO values. Function *ECAT_SetSlaveRxPdoData* can also do it; but users have to enter more parameters for the same purpose.

Syntax:

int32_t ECAT_SetSlaveDO(uint16_t DeviceNo, uint16_t SlaveNo, uint32_t Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
SlaveNo	uint16_t	IN	Slave number
Value	uint32_t	IN	Digital output data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint32_t Value = 255;

ret = ECAT_SetSlaveDO(DeviceNo, SlaveNo, Value);
if(ret < 0)
	printf("Failed to set slave DO:%d\n", ret);
else
	printf("Set slave DO successfully! \n");
```

6.12. ECAT_SetSlaveDOBit

Description:

Set a bit data of a slave's digital output.

Syntax:

int32_t ECAT_SetSlaveDOBit(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t BitNo, uint32_t Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
BitNo	uint16_t	IN	Bit number
Value	uint32_t	IN	Bit data (0 or 1)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t BitNo = 2;
uint32_t Value = 1;

ret = ECAT_GetSlaveDOBit(DeviceNo, SlaveNo, BitNo, Value);
if(ret < 0)
	printf("Failed to set slave DI bit:%d\n", ret);
else
	printf("Set slave DO bit successfully! \n");
```


6.13. ECAT_SetSlaveAoProperty

Description:

Set the AO channel property value.

Syntax:

int32_t ECAT_SetSlaveAoProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, uint8_t Range)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
ChannelNo	uint16_t	IN	Channel number
Range	uint8_t	IN	AO range code (As show in Table 6.2)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 6.2: AO range code

Macro Definition	Value	Description
SLAVE_AO_UNI_5V	0	0 ~ 5V
SLAVE_AO_BI_5V	1	±5V
SLAVE_AO_UNI_10V	2	0 ~ 10V
SLAVE_AO_BI_10V	3	±10V

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
uint8_t Range = SLAVE_AO_UNI_10V;

ret = ECAT_SetSlaveAoProperty(DeviceNo, SlaveNo, ChannelNo, Range);
if(ret != 0)
    printf("Failed to set slave AO settings:%d\n", ret);
else
    printf("Set slave AO settins successfully! \n");
```

6.14. ECAT_GetSlaveAoProperty

Description:

Get the AO channel property value.

Syntax:

int32_t ECAT_GetSlaveAoProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t *ChannelNo, uint8_t *Range)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
ChannelNo	uint16_t	IN	Channel number
Range	uint8_t *	OUT	AO range code (As show in Table 6.2)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
uint8_t Range;

ret = ECAT_GetSlaveAoProperty(DeviceNo, SlaveNo, ChannelNo, &Range);
if(ret != 0)
	printf("Failed to get slave AO settings:%d\n", ret);
else
	printf("AO range:%d\n", Range);
```

6.15. ECAT_SetSlaveAoRawData

Description:

Set the binary value of an analog output channel.

Syntax:

int32_t ECAT_SetSlaveAoRawData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, int16_t Data)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
ChannelNo	uint16_t	IN	Channel number
Data	int16_t	IN	AO binary value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
int16_t Data = 0xFF;

ret = ECAT_SetSlaveAoRawData(DeviceNo, SlaveNo, ChannelNo, Data);
if(ret != 0)
        printf("Failed to set slave AO raw data:%d\n", ret);
else
        printf("Set slave AO raw data successfully! \n");
```


6.16. ECAT_GetSlaveAoRawData

Description:

Get the binary value of an analog output channel.

Syntax:

int32_t ECAT_GetSlaveAoRawData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, int16_t *Data)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
ChannelNo	uint16_t	IN	Channel number
Data	int16_t *	OUT	AO binary value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
int16_t Data;

ret = ECAT_GetSlaveAoRawData(DeviceNo, SlaveNo, ChannelNo, &Data);
if(ret != 0)
    printf("Failed to get slave AO raw data:%d\n", ret);
else
    printf("AO raw data:%d\n", Data);
```


6.17. ECAT_SetSlaveAoVoltData

Description:

Set the voltage output value of a specified analog output channel.

Syntax:

int32_t ECAT_SetSlaveAoVoltData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, double Data)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
ChannelNo	uint16_t	IN	Channel number
Data	double	IN	AO voltage value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
double Data = 5.5;

ret = ECAT_SetSlaveAoVoltData(DeviceNo, SlaveNo, ChannelNo, Data);
if(ret != 0)
    printf("Failed to set slave AO volt data:%d\n", ret);
else
    printf("Set slave AO volt data successfully! \n");
```


6.18. ECAT_GetSlaveAoVoltData

Description:

Get the voltage output value (a floating-point value) of a specified analog output channel.

Syntax:

int32_t ECAT_GetSlaveAoVoltData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, double *Data)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
ChannelNo	uint16_t	IN	Channel number
Data	Double *	OUT	AO voltage value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
double Data;

ret = ECAT_GetSlaveAoVoltData(DeviceNo, SlaveNo, ChannelNo, &Data);
if(ret != 0)
    printf("Failed to get slave AO volt data:%d\n", ret);
else
    printf("AO volt data:%d\n", Data);
```


6.19. ECAT_SetSlaveEncProperty

Description:

Set the encoder property value. This function is designed for encoder module ECAT-2093 and ECAT-2092.

Syntax:

int32_t ECAT_SetSlaveEncProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t Mode, uint8_t InvertCnt, uint8_t LPF)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder inteface channel number
Mode	uint8_t	IN	Encoder Mode 1: CW/CCW 2: Pulse/Dir 3: A/B Phase
InvertCnt	uint8_t	IN	Invert count (change counting direction)
LPF	uint8_t	IN	Low pass filter

Low-pass filter settings:

Low pass filter number	Maximum Input Frequency	
	Pulse/Direction counting mode Clockwise/Counterclockwise mode	Quadrant counting mode
0	4MHz (filter disabled)	6MHz (filter disabled)
1	4MHz	1MHz
2	2MHz	500KHz
3	1MHz	250KHz
4	640KHz	160KHz
5	320KHz	80KHz
6	160KHz	40Hz
7	80KHz	20KHz
8	40KHz	10KHz

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Mode = 3; //A/B Phase
uint8_t InvertCnt = 1; //Enable Reverse
uint8_t LPF = 0;
ret = ECAT_SetSlaveEncProperty(DeviceNo, SlaveNo, EncNo, Mode, InvertCnt, LPF);
if(ret != 0)
    printf("Failed to set encoder property:%d\n",ret);
else
    printf("Set encoder property successfully! \n");
```


6.20. ECAT_GetSlaveEncProperty

Description:

Get the encoder property value. This function is designed for encoder module ECAT-2093 and ECAT-2092.

Syntax:

int32_t ECAT_GetSlaveEncProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t *Mode, uint8_t *InvertCnt, uint8_t *LPF)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder interface number
Mode	uint8_t *	OUT	Encoder Mode 1: CW/CCW 2: Pulse/Dir 3: A/B Phase
InvertCnt	uint8_t *	OUT	Invert count (change counting direction)
LPF	uint8_t *	OUT	Low pass filter

Low-pass filter settings:

Low pass filter number	Maximum Input Frequency	
	Pulse/Direction counting mode Clockwise/Counterclockwise mode	Quadrant counting mode
0	4MHz (filter disabled)	6MHz (filter disabled)
1	4MHz	1MHz
2	2MHz	500KHz
3	1MHz	250KHz
4	640KHz	160KHz
5	320KHz	80KHz
6	160KHz	40Hz
7	80KHz	20KHz
8	40KHz	10KHz

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Mode;
uint8_t ReverseCnt ;
uint8_t LPF;
ret = ECAT_GetSlaveEncProperty(DeviceNo, SlaveNo, EncNo, &Mode, &ReverseCnt, &LPF);
if(ret != 0)
{
	printf("Failed to get encoder property:%d\n", ret);
}
else
{
	printf("Encoder mode:%u\n", Mode);
	printf("Encoder reverse:%u\n", ReverseCnt);
	printf("Encoder Low Pass Filter:%u\n", LPF);
}
```

6.21. ECAT_GetSlaveEncCount

Description:

Get the encoder counter value. This function is designed for encoder module ECAT-2093 and ECAT-2092.

Syntax:

int32_t ECAT_GetSlaveEncCount(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, int32_t *Cnt)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder interface number
Cnt	int32_t *	OUT	Encoder counter value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t Cnt;
ret = ECAT_ GetSlaveEncCount(DeviceNo, SlaveNo , EncNo, &Cnt);
if(ret != 0)
{
	printf("Failed to get encoder count:%d\n",ret);
}
else
{
	printf("Encoder count:%d\n",Cnt);
}
```


6.22. ECAT_ResetSlaveEncCount

Description:

Clear the encoder counter value. This function is designed for encoder module ECAT-2093 and ECAT-2092.

Syntax:

```
int32_t ECAT_ResetSlaveEncCount(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo)
```

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder interface number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
ret = ECAT_ ResetSlaveEncCount(DeviceNo, SlaveNo , EncNo);
if(ret != 0)
{
    printf("Failed to reset encoder count:%d\n", ret);
}
else
{
    printf("Reset encoder count successfully!\n");
}
```

6.23. ECAT_SetSlaveEncLatch

Description:

Set the position latch function property value. This function is designed for encoder module ECAT-2093 and ECAT-2092.

Syntax:

int32_t ECAT_SetSlaveEncLatch(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t Enable)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder interface number
Enable	uint8_t	IN	Enable/Disable latch

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Enable = 1;

ret = ECAT_SetSlaveEncLatch(DeviceNo, SlaveNo, EncNo, Enable);
if(ret != 0)
{
    printf("Failed to set encoder latch:%d\n", ret);
}
else
{
    printf("Set encoder latch successfully!\n");
}
```


6.24. ECAT_GetSlaveEncLatch

Description:

Get the position latch function property value. This function is designed for encoder module ECAT-2093 and ECAT-2092.

Syntax:

int32_t ECAT_GetSlaveEncLatch(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t *Enable)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder interface number
Enable	uint8_t *	OUT	Enable/Disable latch

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Enable;

ret = ECAT_SetSlaveEncLatch(DeviceNo, SlaveNo, EncNo, &Enable);
if(ret != 0)
{
	printf("Failed to get encoder latch:%d\n", ret);
}
else
{
	printf("Encoder latch enable:%u\n", Enable);
}
```


6.25. ECAT_GetSlaveEncLatchCnt

Description:

Get the latch count. This function is designed for encoder module ECAT-2093 and ECAT-2092.

Syntax:

int32_t ECAT_GetSlaveEncLatchCnt(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, int32_t *Cnt)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder interface number
Cnt	int32_t *	OUT	Latch count

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t Cnt;

ret = ECAT_ GetSlaveEncLatchCnt(DeviceNo, SlaveNo, EncNo, &Cnt);
if(ret != 0)
{
      printf("Failed to get encoder latch count:%d\n",ret);
}
else
{
      printf("Encoder latch count:%u\n", Cnt);
}
```

7. Motion Control Functions

7.1. Motion Control Initialization

7.1.1. ECAT_McInit

Description:

Initialize parameters for motion control.

Syntax:

int32_t ECAT_McInit(uint16_t DeviceNo, uint16_t SlaveNo[], uint16_t SubAxisNo[], uint16_t AxisCount)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t[]	IN	An array of Slave number. Each index of this array is an slave number.
SubAxisNo	uint16_t[]	IN	Sub-axis number. In general, a slave only has an axis. But some slave has several axes. Several sub-axis numbers are provided for this kind of slave. With the combination of save number and sub-axis number, the system can have all axes be defined and used individually.
AxisCount	uint16_t	IN	Set the number of axes (MC_AXIS_NO_MAX macro is the maximum number of axes)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisCount=0;
uint16_t McSlaveNo[MC_AXIS_NO_MAX];
uint16_t McSubAxisNo[MC_AXIS_NO_MAX];
McSlaveNo[0] = 0;
McSlaveNo[1] = 1;
McSlaveNo[2] = 2;
McSlaveNo[3] = 2;
McSlaveNo[4] = 2;
McSlaveNo[5] = 2;
McSubAxisNo [0] = 0;
McSubAxisNo [1] = 0;
McSubAxisNo [2] = 0;
McSubAxisNo [3] = 1;
McSubAxisNo [4] = 2;
McSubAxisNo [5] = 3;

...
AxisCount = 6;
ret = ECAT_McInit(DeviceNo, McSlaveNo, McSubAxisNo , AxisCount);
if(ret < 0)
{
    printf("Failed to initialize motion control:%d\n", ret);
}
else
{
    printf("Initialize motion control successfully\n");
}
```

7.2. Axis Parameter Settings

7.2.1. ECAT_McSetAxisServoOn

Description:

Set an axis (a drive) to be servo ON or OFF state.

Syntax:

int32_t ECAT_McSetAxisServoOn(uint16_t DeviceNo, uint16_t AxisNo, uint16_t State)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
State	uint16_t	IN	Servo Driver state 0: OFF 1: ON

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t State = 1;
...
ret = ECAT_McSetAxisServoOn(DeviceNo, AxisNo, State);
if(ret < 0)
{
	printf("Failed to set axis ServoOn:%d\n", ret);
}
else
{
	printf("Set axis ServoOn successfully!\n");
}
```

7.2.2. ECAT_McSetAxisPPU

Description:

Set pulse per unit to a specific axis.

Syntax:

int32_t ECAT_McSetAxisPPU(uint16_t DeviceNo, uint16_t AxisNo, double PPU)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
PPU	double	IN	Pulse Per Unit

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double PPU = 100000;
...
ret = ECAT_McSetAxisPPU(DeviceNo, AxisNo, PPU);
if(ret < 0)
{
	printf("Failed to set axis PPU:%d\n", ret);
}
else
{
	printf("Set axis PPU successfully!\n");
}
```

7.2.3. ECAT_McGetAxisPPU

Description:

Get pulse per unit of a axis.

Syntax:

int32_t ECAT_McGetAxisPPU(uint16_t DeviceNo, uint16_t AxisNo, double *PPU)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
PPU	Double*	OUT	Pulse Per Unit

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double PPU;
...
ret = ECAT_McGetAxisPPU(DeviceNo, AxisNo, &PPU);
if(ret < 0)
{
	printf("Failed to get axis PPU:%d\n", ret);
}
else
{
	printf("Axis[%u] PPU:%f\n", AxisNo, PPU);
}
```


7.2.4. ECAT_McSetProfileData

Description:

Set a position array data into a buffer number for profile motion.

Syntax:

int32_t ECAT_McSetProfileData(uint16_t DeviceNo, uint16_t ProfileNo, double *Data, uint16_t DataSize)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
ProfileNo	uint16_t	IN	Profile number, available number range 0~15
Data	double*	IN	Data buffer, can store up to 3000 double-type data
DataSize	uint16_t	IN	Size of data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProfileNo = 0;
double Data[MC_PROFILE_DATA_MAX];
uint16_t DataSize = 10;
Data[0] =   0.00005;
Data[1] =   0.00015;
Data[2] =   0.00030;
Data[3] =   0.00050;
Data[4] =   0.00075;
Data[5] =   0.00105;
Data[6] =   0.00140;
Data[7] =   0.00180;
Data[8] =   0.00225;
Data[9] =   0.00275;

ret = ECAT_McSetProfileData(DeviceNo, ProfileNo, Data, DataSize);
if(ret < 0)
{
    printf("Failed to set profile data:%d\n", ret);
}
else
{
    printf("Set set profile data successfully!\n");
}
```


7.2.5. ECAT_McGetProfileData

Description:

Get a position array from a profile buffer number

Syntax:

int32_t ECAT_McGetProfileData(uint16_t DeviceNo, uint16_t ProfileNo, double *Data, uint16_t DataSize)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
ProfileNo	uint16_t	IN	Profile number, available number range 0~15
Data	double*	OUT	Data buffer, can store up to 3000 double-type data
DataSize	uint16_t	IN	Size of data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t i;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProfileNo = 0;
double Data[MC_PROFILE_DATA_MAX];
uint16_t DataSize = 10;
...
ret = ECAT_McGetProfileData(DeviceNo, ProfileNo, Data, DataSize);
if(ret < 0)
{
	printf("Failed to get profile data:%d\n",ret);
}
else
{
	printf("Set get profile data successfully!\n");
	for(i=0; i<DataSize; i++)
		printf("Data[%u]:%f\n", i, Data[i]);
}
```

7.2.6. ECAT_McSetProfileCSV

Description:

Write position data to a CSV file. This file contain data for profile motion. The data format is shown in Figure 7.1.

Syntax:

int32_t ECAT_McSetProfileCSV(uint16_t DeviceNo, uint16_t ProfileNo, uint32_t Offset,char *Data, uint32_t DataSize, uint8_t LastFlag)

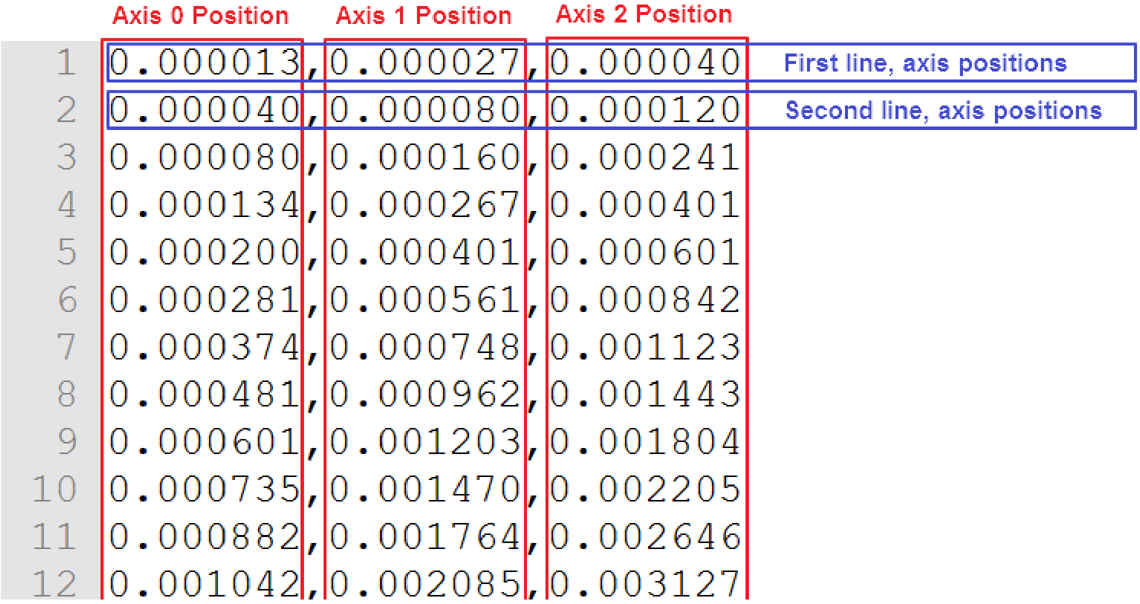
Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
ProfileNo	uint16_t	IN	Profile number, available number range 0~15
Offset	uint32_t	IN	File offset
Data	char *	IN	Data buffer
DataSize	uint32_t	IN	Size of the data
LastFlag	uint8_t	IN	Write end flag 0: more data will be written 1: this is the last write action

Return:

0: Success.

Others: Refer to Appendix "Error Codes".



1	0.000013,	0.000027,	0.000040
2	0.000040,	0.000080,	0.000120
3	0.000080,	0.000160,	0.000241
4	0.000134,	0.000267,	0.000401
5	0.000200,	0.000401,	0.000601
6	0.000281,	0.000561,	0.000842
7	0.000374,	0.000748,	0.001123
8	0.000481,	0.000962,	0.001443
9	0.000601,	0.001203,	0.001804
10	0.000735,	0.001470,	0.002205
11	0.000882,	0.001764,	0.002646
12	0.001042,	0.002085,	0.003127

Figure 7.1

Example:

[C/C++]

```
FILE *pFile;
size_t file_Size;
char *buffer;
size_t result;
int32_t ret;
uint16_t ProfileNo = 0;
uint8_t LastFlag = 1;
char *file_name = "D:\xxx.csv"

pFile = fopen(file_name, "rb" );
if (pFile==NULL) {
	printf("Failed to open file:%s", file_name);
	return;
}

// obtain file size:
fseek (pFile, 0, SEEK_END);
file_Size = ftell(pFile);
```

```
fseek (pFile, 0, SEEK_SET);

// allocate memory to contain the whole file:
buffer = (char*)malloc(sizeof(char)*file_Size);
if (buffer == NULL) {
	printf("Failed to allocate memory");
	fclose(pFile);
	return;
}

// copy the file into the buffer:
result = fread(buffer, 1, file_Size, pFile);
if (result != file_Size) {
	printf("Failed to read from file");
	goto out_close;
}

/* the whole file is now loaded in the memory buffer. */
ret = ECAT_McSetProfileCSV(DeviceNo, ProfileNo, 0, buffer, file_Size, LastFlag);
if(ret != 0)
	printf("Failed to set profile csv data:%d", ret);

out_close:
	fclose(pFile);
	free(buffer);
```

7.2.7. ECAT_McGetProfileCSV

Description:

Get position data from a CSV file. This file is used for profile moiton. The format is shown in Figure 7.1.

Syntax:

int32_t ECAT_McGetProfileCSV(uint16_t DeviceNo, uint16_t ProfileNo, uint32_t *Offset, char *Data, uint32_t *DataSize, uint8_t *LastFlag)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
ProfileNo	uint16_t	IN	Profile number, available number range 0~15
Offset	uint32_t *	OUT	File offset
Data	char *	OUT	Data buffer
DataSize	uint32_t *	OUT	Size of the data
LastFlag	uint8_t *	OUT	Read end flag 0: more data can be read 1: reach the end of file

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
FILE * pFile;
char Data[2048];
int32_t ret;
char *file_name = "D:\xxx.csv"
uint16_t ProfileNo = 0;
uint8_t LastFlag;
uint32_t DataSize;
uint32_t Offset = 0;

pFile = fopen(file_name, "wb" );
if (pFile==NULL) {
	printf("Failed to create file:%s", file_name);
	return;
}

while(1)
{
	DataSize = 2048;
	LastFlag = 0;
	if((ret = ECAT_McGetProfileCSV(DeviceNo, ProfileNo, &Offset, Data,
	&DataSize, &LastFlag)) != 0) {
		printf("Failed to get profile csv data:%d", ret);
		fclose(pFile);
		return;
	}

	if (fwrite(Data , 1, DataSize, pFile) != DataSize) {
		printf("Failed to Write File");
		fclose(pFile);
		return;
	}

	if(LastFlag) {
		fclose(pFile);
		break;
```



```
    }
};
```

7.2.8. ECAT_McSetAxisAccTime

Description:

Set acceleration time of an axis.

Syntax:

int32_t ECAT_McSetAxisAccTime(uint16_t DeviceNo, uint16_t AxisNo, uint16_t Time_ms)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Time_ms	uint16_t	IN	Acceleration time (Unit: millisecond)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Time_ms = 500;
...
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
}
else
{
    printf("Set axis acceleration time successfully!\n");
}
```

7.2.9. ECAT_McGetAxisAccTime

Description:

Get acceleration time of an axis.

Syntax:

int32_t ECAT_McGetAxisAccTime(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *Time_ms)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
AxisNo	uint16_t	IN	Axis number
Time_ms	uint16_t*	OUT	Acceleration time (Unit: millisecond)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Time_ms;
ret = ECAT_McGetAxisAccTime(DeviceNo, AxisNo, &Time_ms);
if(ret < 0)
{
    printf("Failed to get axis acceleration time:%d\n", ret);
}
else
{
    printf("Axis[%u] Acceleration Time(ms):%f\n", AxisNo, Time_ms);
}
```

7.2.10. ECAT_McSetAxisAccDecType

Description:

Set acceleration type of an axis.

Syntax:

int32_t ECAT_McSetAxisAccDecType(uint16_t DeviceNo, uint16_t AxisNo, uint16_t Type)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Type	uint16_t	IN	Acceleration Type 1:T-Curve 2:S-Curve

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Type = 1; //T-Curve
ret = ECAT_McSetAxisAccDecType (DeviceNo, AxisNo, Type);
if(ret < 0)
{
    printf("Failed to set axis AccDecType:%d\n", ret);
}
else
{
    printf("Set axis AccDecType successfully!\n");
}
```


7.2.11. ECAT_McGetAxisAccDecType

Description:

Get acceleration type of an axis.

Syntax:

int32_t ECAT_McGetAxisAccDecType(uint16_t DeviceNo, uint16_t AxisNo, uint16_t * Type)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Type	uint16_t*	OUT	Acceleration Type 1:T-Curve 2:S-Curve

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Type;
ret = ECAT_McGetAxisAccDecType (DeviceNo, AxisNo, &Type);
if(ret < 0)
{
	printf("Failed to get axis AccDecType:%d\n", ret);
}
else
{
	printf("Axis[%u] AccDecType:%f\n", AxisNo, Type);
}
```


7.2.12. ECAT_McSetAxisEncoderPPR

Description:

Set encoder pulse per revolution value of an axis.

Syntax:

int32_t ECAT_McSetAxisEncoderPPR(uint16_t DeviceNo, uint16_t AxisNo, uint32_t PPR)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
PPR	Uint32_t	IN	Pulse per revolution

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t PPR =4000;
...
ret = ECAT_McSetAxisEncoderPPR (DeviceNo, AxisNo, PPR);
if(ret < 0)
{
	printf("Failed to set axis encoder PPR:%d\n", ret);
}
else
{
	printf("Set axis encoder PPR successfully!\n");
}
```


7.2.13. ECAT_McGetAxisEncoderPPR

Description:

Get encoder pulse per revolution of an axis.

Syntax:

int32_t ECAT_McGetAxisEncoderPPR(uint16_t DeviceNo, uint16_t AxisNo, uint32_t *PPR)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
PPR	Uint32_t*	OUT	Pulse per revolution

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t PPR ;
ret = ECAT_McGetAxisEncoderPPR (DeviceNo, AxisNo, &PPR);
if(ret < 0)
{
    printf("Failed to get axis encoder PPR:%d\n", ret);
}
else
{
    printf("Axis[%u] encoder PPR :%f\n", AxisNo, PPR);
}
```


7.2.14. ECAT_McSetAxisMotorPPR

Description:

Set motor pulse per revolution of an axis.

Syntax:

int32_t ECAT_McSetAxisMotorPPR(uint16_t DeviceNo, uint16_t AxisNo, uint32_t PPR)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
PPR	Uint32_t	IN	Pulse per revolution

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t PPR =4000;
ret = ECAT_McSetAxisMotorPPR (DeviceNo, AxisNo, PPR);
if(ret < 0)
{
    printf("Failed to set axis motor PPR:%d\n", ret);
}
else
{
    printf("Set axis motor PPR successfully!\n");
}
```


7.2.15. ECAT_McGetAxisMotorPPR

Description:

Get motor pulse per revolution of an axis.

Syntax:

int32_t ECAT_McGetAxisMotorPPR(uint16_t DeviceNo, uint16_t AxisNo, uint32_t *PPR)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
PPR	Uint32_t*	OUT	Pulse per revolution

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t PPR ;
...
ret = ECAT_McGetAxisMotorPPR (DeviceNo, AxisNo, &PPR);
if(ret < 0)
{
    printf("Failed to get axis motor PPR:%d\n", ret);
}
else
{
    printf("Axis[%u] motor PPR :%f\n", AxisNo, PPR);
}
```


7.2.16. ECAT_McSetEcamTable

Description:

Set the slave position data for an E-CAM table. There are two tables available for this card.

Syntax:

int32_t ECAT_McSetEcamTable(uint16_t DeviceNo, uint16_t TableNo, double *Data, uint16_t DataSize)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
TableNo	uint16_t	IN	E-CAM table number (0 or 1)
Data	double*	IN	Slave position data (Unit: user unit)
DataSize	uint16_t	IN	Size of data (Up to 360)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t TableNo = 0;
double Data[360];
uint16_t DataSize = 360;
/**
***Write E-CAM Table data to Data[360]

**/
ret = ECAT_McSetEcamTable(DeviceNo, TableNo, Data, DataSize);

if(ret < 0)
{
    printf("Failed to set E-CAM table data:%d\n", ret);
}
else
{
    printf("Set E-CAM table data successfully!\n");
}
```

7.2.17. ECAT_McGetEcamTable

Description:

Get the slave position data from an E-CAM table. There are two tables available for this card.

Syntax:

int32_t ECAT_McGetEcamTable(uint16_t DeviceNo, uint16_t TableNo, double *Data, uint16_t DataSize)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
TableNo	uint16_t	IN	E-CAM table number (0 or 1)
Data	double*	OUT	Slave position data (Unit: user unit)
DataSize	uint16_t	IN	Size of data (Up to 360)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t TableNo = 0;
double Data[360];
uint16_t DataSize = 360;
ret = ECAT_McGetEcamTable(DeviceNo, TableNo, Data, DataSize);

if(ret < 0)
{
	printf("Failed to get E-CAM table data:%d\n",ret);
}
else
{
	printf("Get E-CAM table data successfully!\n");
	for(i=0;i<DataSize;i++)
		printf("Data[%u]:%f\n", i, Data[i]);
}
```


7.2.18. ECAT_McConfigEcamTable

Description:

Set data property of an E-CAM table.

Syntax:

int32_t ECAT_McConfigEcamTable(uint16_t DeviceNo, uint16_t TableNo, uint8_t SlaveAbs)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
TableNo	uint16_t	IN	E-CAM table number (0 or 1)
SlaveAbs	uint8_t	IN	Slave positon data type 0: Relative position 1: Absolute position

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t TableNo = 0;
uint8_t SlaveAbs = 0;
ret = ECAT_McConfigEcamTable(DeviceNo, TableNo, SlaveAbs);

if(ret < 0)
{
	printf("Failed to configure E-CAM table parameter:%d\n", ret);
}
else
{
	printf("Configure E-CAM table parameter successfully!\n");
}
```


7.2.19. ECAT_McSetAxisTouchProbeProperty

Description:

Configure Touch Probe function of an axis. Servo drives can have up to two Touch Probe inputs. But some have one, and some have none.

Syntax:

int32_t ECAT_McSetAxisTouchProbeProperty(uint16_t DeviceNo, uint16_t AxisNo, uint16_t ProbeNo, uint8_t Enable, uint8_t Logic)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
ProbeNo	uint16_t	IN	Touch Probe number 1: Touch Probe 1 input 2: Touch Probe 2 input
Enable	uint8_t	IN	Enable/Disable Touch Probe function
Logic	uint8_t	IN	Touch Probe logic level 0: Falling edge 1: Rising edge

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProbeNo = 1;
uint8_t Enable = 1;
uint8_t Logic = 1; //rising edge
ret = ECAT_McSetAxisTouchProbeProperty((DeviceNo, AxisNo, ProbeNo, Enable, Logic);
if(ret < 0)
{
	printf("Failed to set Touch Probe property:%d\n", ret);
}
else
{
	printf("Set Touch Probe property successfully!\n");
}
```

7.2.20. ECAT_McGetAxisTouchProbeProperty

Description:

Get the property settings of Touch Probe function of an axis. Servo drives can have up to two Touch Probe inputs. But some have one, and some have none.

Syntax:

int32_t ECAT_McGetAxisTouchProbeProperty(uint16_t DeviceNo, uint16_t AxisNo, uint16_t ProbeNo, uint8_t *Enable, uint8_t *Logic)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
ProbeNo	uint16_t	IN	Touch Probe number 1: Touch Probe 1 input 2: Touch Probe 2 input
Enable	uint8_t *	OUT	Enable/Disable Touch Probe function
Logic	uint8_t *	OUT	Touch Probe logic level 0: Falling edge 1: Rising edge

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProbeNo = 1;
uint8_t Enable;
uint8_t Logic;
ret = ECAT_McGetAxisTouchProbeProperty(DeviceNo, AxisNo, ProbeNo, &Enable, &Logic);
if(ret < 0)
{
	printf("Failed to get Touch Probe property:%d\n", ret);
}
else
{
	printf("Touch Probe[%u]->Enable:%u\n", ProbeNo, Enable);
	printf("Touch Probe[%u]->Logic:%u\n", ProbeNo, Logic);
}
```


7.2.21. ECAT_McGetAxisTouchProbeValue

Description:

Get the Touch Probe value of an axis.

Syntax:

int32_t ECAT_McGetAxisTouchProbeValue(uint16_t DeviceNo, uint16_t AxisNo, uint16_t ProbeNo, double *Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
AxisNo	uint16_t	IN	Axis number
ProbeNo	uint16_t	IN	Touch Probe number 1: Touch Probe 1 input 2: Touch Probe 2 input
Value	double *	OUT	Touch Probe Value (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProbeNo = 1;
double Value;
ret = ECAT_McGetAxisTouchProbeValue(DeviceNo, AxisNo, ProbeNo, &Value);
if(ret < 0)
{
	printf("Failed to get Touch Probe value:%d\n", ret);
}
else
{
	printf("Touch Probe[%u]->Value:%f\n", ProbeNo, Value);
}
```


7.2.22. ECAT_McSetAxisVelocityFeedForwardGain

Description:

Set Velocity Feed Forward Gain of an axis. Note: Only for CiA402 servo drives. In general, the feed forward velocity can help in the performance of position tracking control. This function defines the contribution of the feed forward velocity for position control.

Syntax:

int32_t ECAT_McSetAxisVelocityFeedForwardGain(uint16_t DeviceNo, uint16_t AxisNo, double Gain)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Gain	double	IN	Velocity Feed Forward Gain range: 0 ~ 1

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Gain = 0.95;
ret = ECAT_McSetAxisVelocityFeedForwardGain(DeviceNo, AxisNo, Gain);
if(ret < 0)
{
	printf("Failed to set axis Velocity Feed Forward Gain%d\n", ret);
}
else
{
	printf("Set axis Velocity Feed Forward Gain successfully!\n");
}
```

7.2.23. ECAT_McGetAxisVelocityFeedForwardGain

Description:

Get Velocity Feed Forward Gain of an axis. Note: Only for CiA402 servo drives. In general, the feed forward velocity can help in the performance of position tracking control. This function defines the contribution of the feed forward velocity for position control.

Syntax:

int32_t ECAT_McGetAxisVelocityFeedForwardGain(uint16_t DeviceNo, uint16_t AxisNo, double * Gain)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Gain	Double*	OUT	Velocity Feed Forward Gain

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Gain;
ret = ECAT_McGetAxistVelocityFeedForwardGain(DeviceNo, AxisNo, &Gain);
if(ret < 0)
{
    printf("Failed to get axis Velocity Feed Forward Gain:%d\n", ret);
}
else
{
    printf("Axis[%u] Velocity Feed Forward Gain:%f\n", AxisNo, Gain);
}
```

7.2.24. ECAT_McSetAxisPosSoftwareLimit

Description:

Set position software limit to a specific axis. Notice:Only for CiA402 and Virtual axis.

Syntax:

int32_t ECAT_McSetAxisPosSoftwareLimit(uint16_t DeviceNo, uint16_t AxisNo, double Maximum, double Minimum);

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
AxisNo	uint16_t	IN	Axis number
Maximum	double	IN	Position maximum value(unit: user unit)
Minimum	double	IN	Position minimum value(unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Maximum = 100.0;
double Minimum = -100.0;
ret = ECAT_McSetAxisPosSoftwareLimit(DeviceNo,AxisNo, Maximum, Minimum);
if(ret < 0)
{
    printf("Failed to set axis position software limit :%d\n",ret);
}
else
{
    printf("Set axis position software limit successfully!\n");
}
```


7.2.25. ECAT_McGetAxisPosSoftwareLimit

Description:

Get position software limit to a specific axis. Notice:Only for CiA402 and Virtual axis.

Syntax:

int32_t ECAT_McGetAxisPosSoftwareLimit(uint16_t DeviceNo, uint16_t AxisNo, double *Maximum, double *Minimum)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
AxisNo	uint16_t	IN	Axis number
Maximum	Double*	OUT	Position maximum value(unit: user unit)
Minimum	Double*	OUT	Position minimum value(unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

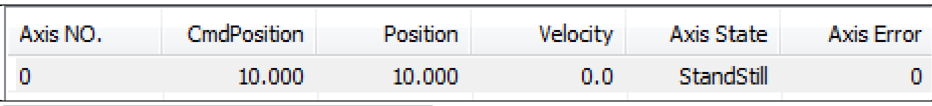
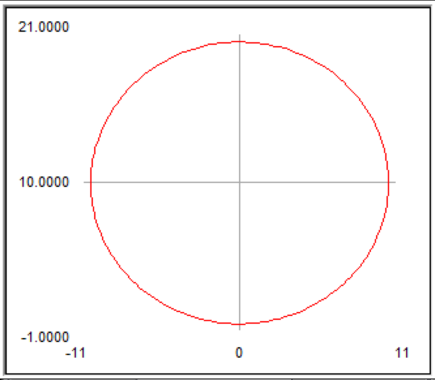
[C/C++]

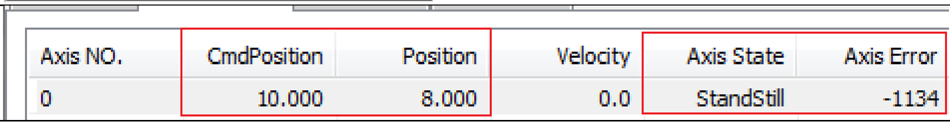
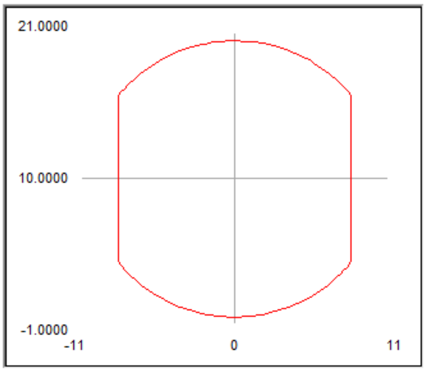
```
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Maximum;
double Minimum;
ret = ECAT_McGetAxisPosSoftwareLimit(DeviceNo,AxisNo,&Maximum,&Minimum);
if(ret < 0)
{
	printf("Failed to get axis position software limit:%d\n",ret);
}
else
{
	printf("Axis[%u] position software limit [Maximin:%f] , [Minimum:%f] \n", AxisNo, Maximum, Minimum);
}
```

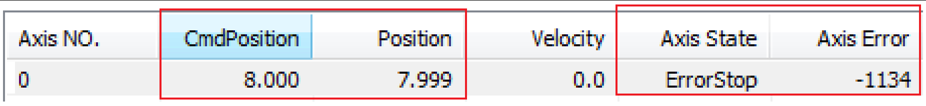
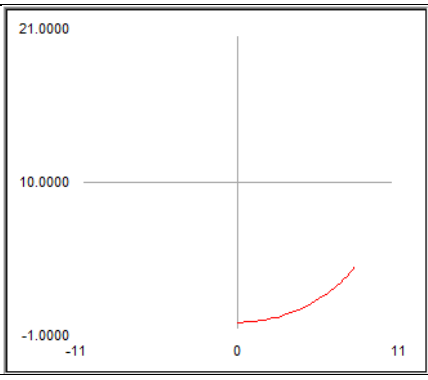
7.2.26. ECAT_McSetAxisPosSoftwareLimitStatus

Description:

Set position software limit status to a specific axis. Notice:Only for CiA402 and Virtual axis.

<table>
<tr><td rowspan="2">example:
Status: 0
(disable)</td><td>axis</td><td>
</td></tr>
<tr><td>Group</td><td>
</td></tr>
<tr><td rowspan="2">Status:1
ErrorStop: 0

Limit of X-Axis:
Maxmum:8
Minimum:-8</td><td>axis</td><td>
</td></tr>
<tr><td>Group</td><td>
</td></tr>
<tr><td rowspan="2">Status:1
ErrorStop: 1

Limit of X-Axis:
Maxmum:8
Minimum:-8</td><td>axis</td><td>
</td></tr>
<tr><td>Group</td><td>
</td></tr>
</table>

Syntax:

int32_t ECAT_McSetAxisPosSoftwareLimitStatus(uint16_t DeviceNo, uint16_t AxisNo, uint16_t Status, uint16_t ErrorStop)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
AxisNo	uint16_t	IN	Axis number
Status	uint16_t	IN	0:disable 1:enable
ErrorStop	uint16_t	IN	0: Axis Last error: -1134 when software limit triggered. 1: ErrorStop and clear group buffer when software limit triggered.

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Status = 0;
uint16_t ErrorStop = 0;
ret = ECAT_McSetAxisPosSoftwareLimitStatus(DeviceNo,AxisNo, Status, ErrorStop);
if(ret < 0)
{
    printf("Failed to set axis position software limit status :%d\n",ret);
}
else
{
    printf("Set axis position software limit status successfully!\n");
}
```


7.2.27. ECAT_McGetAxisPosSoftwareLimitStatus

Description:

Get position software limit status to a specific axis. Notice:Only for CiA402 and Virtual Axis.

Syntax:

int32_t ECAT_McGetAxisPosSoftwareLimitStatus(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *Status, uint16_t *ErrorStop)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
AxisNo	uint16_t	IN	Axis number
Status	uint16_t	IN	0:disable 1:enable
ErrorStop	uint16_t	IN	0: Axis Last error: -1134 when software limit triggered. 1: ErrorStop and clear group buffer when software limit triggered.

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Status = 0;
uint16_t ErrorStop = 0;
ret = ECAT_McGetAxisPosSoftwareLimitStatus(DeviceNo, AxisNo, &Status, &ErrorStop);
if(ret < 0)
{
    printf("Failed to get axis position software limit status:%d\n",ret);
}
else
{
    printf("Axis[%u] position software limit [Status:%f] , [ErrorStop:%f] \n", AxisNo, Status, ErrorStop);
}
```


7.2.28. ECAT_OpenMotionConfig

Description:

Read file of Axis configuration.

Syntax:

int32_t ECAT_ OpenMotionConfig(char* bstrFileName, uint16_t *AxisCnt
, uint16_t SlaveNo[], uint16_t SubAxisNo[], double PPU[], int32_t HomeMethod[]
, double HomeSpeedSeachSw[], double HomeSpeedSeachZr[], double HomeAcc[]
, uint32_t EncoderPPR[], uint32_t MotorPPR[])

Parameters:

Name	Type	IN or OUT	Description
bstrFileName	char*	IN	file name of axis configuration
AxisCnt	uint16_t	OUT	the number of axes
SlaveNo	uint16_t *	OUT	An array of Slave number. Each index of this array is an slave number.
SubAxisNo	uint16_t *	OUT	Sub-axis number. In general, a slave only has an axis. But some slave has several axes. Several sub-axis numbers are provided for this kind of slave. With the combination of save number and sub-axis number, the system can have all axes be defined and used individually.
PPU	Double*	OUT	Pulse Per Unit
HomeMethod	int32_t *	OUT	Homing method (Refer to the drive user manual)
HomeSpeedSeachSw	Double*	OUT	Speed during search for switch (Unit: user unit/s)

HomeSpeedSeachZr	Double*	OUT	Speed during search for z phase signal (Unit: user unit/s)
HomeAcc	Double*	OUT	Homing Acceleration (Unit: user unit/s^2)
EncoderPPR	uint32_t *	OUT	Pulse per revolution
MotorPPR	uint32_t *	OUT	Pulse per revolution

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
char* Filename = "MotionConfig.motcfg";
uint16_t Axiscnt;
uint16_t SlaveNo[MC_AXIS_NO_MAX];
uint16_t SubAxisNo[MC_AXIS_NO_MAX];
double PPU[MC_AXIS_NO_MAX];
int32_t HomeMethod[MC_AXIS_NO_MAX];
double HomeSpeedSeachSw [MC_AXIS_NO_MAX];
double HomeSpeedSeachZr[MC_AXIS_NO_MAX];
double HomeAcc[MC_AXIS_NO_MAX];
uint32_t EncoderPPR [MC_AXIS_NO_MAX];
uint32_t MotorPPR [MC_AXIS_NO_MAX];

CoInitialize(NULL);
ret = ECAT_OpenMotionConfig(FileName, &AxisCnt
, SlaveNo, SubAxisNo, PPU, HomeMethod
, HomeSpeedSeachSw, HomeSpeedSeachZr, HomeAcc
, EncoderPPR, MotorPPR);
CoUninitialize();
if(ret < 0)
{
	printf("Failed to Open Motion Config file:%d\n",ret);
}
```

7.3. Axis Status

7.3.1. ECAT_McGetAxisActualPos

Description:

Get actual position of an axis.

Note: When AxisNo is set to 65535, actual positions of all axes are read back in Pos array pointer.

Syntax:

int32_t ECAT_McGetAxisActualPos(uint16_t DeviceNo, uint16_t AxisNo, double *Pos)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Pos	double*	OUT	Actual position (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double AxisPos;
...
ret = ECAT_McGetAxisActualPos(DeviceNo, AxisNo, &AxisPos);
if(ret < 0)
{
    printf("Failed to get axis actual position:%d\n", ret);
}
else
{
    printf("Axis Actual Position:%f\n", AxisPos);
}
```

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 65535;
double AxisPos[MC_AXIS_NO_MAX];
ret = ECAT_McGetAxisActualPos(DeviceNo, AxisNo, AxisPos);
if(ret < 0)
{
    printf("Failed to get axis actual position:%d\n", ret);
}
else
{
    int i;
    for(i=0; i< MC_AXIS_NO_MAX; i++)
    {
        printf("Axis[%d] Actual Position:%f\n", i, AxisPos[ i ] );
    }
}
```


7.3.2. ECAT_McGetAxisCommandPos

Description:

Get command position of an axis.

Note: When AxisNo is set to 65535, command positions of all axes are read back in Pos array pointer.

Syntax:

int32_t ECAT_McGetAxisCommandPos(uint16_t DeviceNo, uint16_t AxisNo, double *Pos)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Pos	double*	OUT	Command position (Unit:user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double AxisPos;
ret = ECAT_McGetAxisCommandPos(DeviceNo, AxisNo, &AxisPos);
if(ret < 0)
{
    printf("Failed to get axis command position:%d\n", ret);
}
else
{
    printf("Axis command Position:%f\n", AxisPos);
}
```

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 65535;
double AxisPos[MC_AXIS_NO_MAX];
ret = ECAT_McGetAxisCommandPos(DeviceNo, AxisNo, AxisPos);
if(ret < 0)
{
    printf("Failed to get axis command position:%d\n", ret);
}
else
{
    int i;
    for(i=0;i< MC_AXIS_NO_MAX;i++)
    {
        printf("Axis[%d] CommandPosition:%f\n", i, AxisPos[ i ] );
    }
}
```


7.3.3. ECAT_McGetAxisActualVel

Description:

Get actual velocity of an axis.

Note: When AxisNo is set to 65535, the actual velocities of all axes are read back in Vel array pointer.

Syntax:

int32_t ECAT_McGetAxisActualVel(uint16_t DeviceNo, uint16_t AxisNo, double *Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Vel	double*	OUT	Actual velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double AxisVel;
ret = ECAT_McGetAxisActualVel(DeviceNo, AxisNo, &AxisVel);
if(ret < 0)
{
	printf("Failed to get axis actual velocity:%d\n", ret);
}
else
{
	printf("Axis Actual Velocity:%f\n", AxisVel);
}
```

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 65535;
double AxisVel[MC_AXIS_NO_MAX];
ret = ECAT_McGetAxisActualVel(DeviceNo, AxisNo, AxisVel);
if(ret < 0)
{
	printf("Failed to get axis actual velocity:%d\n", ret);
}
else
{
	int i;
	for(i=0;i< MC_AXIS_NO_MAX;i++)
	{
		printf("Axis[%d] Actual velocity:%f\n", i, AxisVel[ i ] );
	}
}
```


7.3.4. ECAT_McGetAxisActualPosVel

Description:

Get actual position and velocity of an axis.

Note: When AxisNo is set to 65535, the actual positions and velocities of all axes are read back in Pos and Vel array pointers, respectively.

Syntax:

int32_t ECAT_McGetAxisActualPosVel(uint16_t DeviceNo, uint16_t AxisNo, float *Pos, float *Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
AxisNo	uint16_t	IN	Axis number
Pos	float*	OUT	Actual position (Unit: user unit)
Vel	float*	OUT	Actual velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
float AxisPos;
float AxisVel;

ret = ECAT_McGetAxisActualPos(DeviceNo,AxisNo, &AxisPos, &AxisVel);
if(ret < 0)
{
      printf("Failed to get axis actual position and velocity:%d\n", ret);
}
else
{
      printf("Axis Actual Position:%f , Velocity:%f \n", AxisPos, AxisVel);
}
```

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 65535;
float AxisPos[MC_AXIS_NO_MAX];
float AxisVel[MC_AXIS_NO_MAX];

ret = ECAT_McGetAxisActualPosVel(DeviceNo, AxisNo, AxisPos, AxisVel);
if(ret < 0)
{
      printf("Failed to get axis actual position and velocity:%d\n", ret);
}
else
{
      int i;
      for(i=0; i< MC_AXIS_NO_MAX; i++)
      {
          printf("Axis[%d] Actual Position:%f , Velocity:%f   \n", i, AxisPos[ i ], AxisVel [ i ] );
```



```
    }
}
```

7.3.5. ECAT_McGetAxisState

Description:

Get the state of an axis.

Syntax:

int32_t ECAT_McGetAxisState(uint16_t DeviceNo, uint16_t AxisNo, uint32_t *State)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
State	uint32_t*	OUT	Axis state (As shown in Table 7.1)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.1: Axis State

Macro Definition	Value	Description
MC_AS_DISABLED	0	Axis is disabled
MC_AS_STANDSTILL	1	Axis is standstill, and no motion command active
MC_AS_ERRORSTOP	2	Axis is stopped because of error
MC_AS_STOPPING	3	Axis is stopping
MC_AS_HOMING	4	Axis is homing
MC_AS_DISCRETEMOTION	5	Axis is dicrete motion
MC_AS_CONTINUOUSMOTION	6	Axis is continuous motion
MC_AS_SYNCHRONIZEDMOTION	7	Axis is synchronized motion

Example:

[C/C++]

```
int32_t ret;
char buf[512];
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
...
ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(ret < 0)
{
        printf("Failed to get axis state:%d\n",ret);
}
else
{
        switch(State)
        {
                case MC_AS_DISABLED:
                        printf(buf,"Disabled");
                        break;
                case MC_AS_STANDSTILL:
                        sprintf(buf,"StandStill");
```



```
                break;
            case MC_AS_ERRORSTOP:
                sprintf(buf,"ErrorStop");
                break;
            case MC_AS_STOPPING:
                sprintf(buf,"Stopping");
                break;
            case MC_AS_HOMING:
                sprintf(buf,"Homing");
                break;
            case MC_AS_DISCRETEMOTION:
                sprintf(buf,"DiscMotion");
                break;
            case MC_AS_CONTINUOUSMOTION:
                sprintf(buf,"ContMotion");
                break;
            case MC_AS_SYNCHRONIZEDMOTION:
                sprintf(buf,"SyncMotion");
                break;
            default:
                sprintf(buf,"Invalid");
        }
        printf("Axis State:%s\n", buf);
    }
```


7.3.6. ECAT_McGetAxisLastError

Description:

Get last error of an axis.

Syntax:

int32_t ECAT_McGetAxisLastError(uint16_t DeviceNo, uint16_t AxisNo, int32_t *Error)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Error	int32_t *	OUT	Last error (Refer to Appendix "Error Codes")

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
int32_t Error;
...
ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(ret < 0)
{
    printf("Failed to get axis state:%d\n", ret);
}
else
{
    if(State == MC_AS_ERRORSTOP)
    {
        ret = ECAT_McGetAxisLastError(DeviceNo, AxisNo, &Error);
        if(ret < 0)
        {
            printf("Failed to get axis last error:%d\n", ret);
        }
        else
        {
            printf("Axis Last Error:%d\n", Error);
        }
    }
}
```


7.3.7. ECAT_McGetAxisDriveError

Description:

Get drive error of an axis.

Syntax:

int32_t ECAT_McGetAxisDriveError(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *Error)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Error	int16_t *	OUT	drive error (Refer to the user manual of a servo drive to find the error code)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
int32_t Error;
int16_t DriveError;
...
ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(ret < 0)
{
    printf("Failed to get axis state:%d\n", ret);
}
else
{
    if(State == MC_AS_ERRORSTOP)
    {
        ret = ECAT_McGetAxisLastError(DeviceNo, AxisNo, &Error);
        if(ret < 0)
        {
            printf("Failed to get axis last error:%d\n", ret);
        }
        else
        {
            printf("Axis Last Error:%d\n", Error);
            if(Error == ECAT_ERR_MC_DRIVE_FAULT) //Drive fault
            {
                ret = ECAT_McGetAxisDriveError(EcatDeviceID, AxisNo, &DriveError);
                if(ret < 0)
                {
                    printf("Failed to get axis drive error:%d\n", ret);
                }
                else
                {
                    printf("Axis Drive Error:%d\n", DriveError);
                }
```



```
                }
            }
        }
    }
```

7.3.8. ECAT_McGetAxisDI

Description:

Get digital inputs of an axis. These digital inputs are available in this drive.

Syntax:

int32_t ECAT_McGetAxisDI(uint16_t DeviceNo, uint16_t AxisNo, uint32_t *DI)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
DI	uint32_t *	OUT	Digital input status (As show in Table 7.2)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.2: Axis I/O status

位元	對應訊號
Bit 0	NOT (Negative limit switch)
Bit 1	POT (Positive limit switch)
Bit 2	ORG (Home switch)
Bit 3	ALM (Alarm)
Bit 4	WAN (Warning)
Bit 5	SVN (Servo-ON state)
Bit 6	VIR (Virtual Axis)
Bit 7~31	Reserved

Example:

[C/C++]

```
typedef struct axis_di{
    union
    {
        struct
        {
            uint8_t NOT       : 1;       //Negative limit switch
            uint8_t POT       : 1;       //Positive limit switch
            uint8_t ORG       : 1;       //home switch
            uint8_t ALM       : 1;       //alarm
            uint8_t WAN       : 1;       //warning
            uint8_t SVN       : 1;       //serve on status
            uint8_t VIR       : 1;       //virtual axis
            uint32_t     reserved : 25;  //Reserved(bit7~bit31)
        };
        uint32_t DIs;
    };
}axis_di_t;
/****************************************************/
int32_t ret;
axis_di_t AxisDI;
uint32_t DeviceNo = 0;
```



```
uint16_t AxisNo = 0;
ret = ECAT_McGetAxisDI(DeviceNo, AxisNo, &AxisDI.DIs);
if(ret < 0)
{
    printf("Failed to get axis DI:%d\n", ret);
}
else
{
    printf("AxisNo[%u]-+-AxisDI\n"
            "           |-NOT:%d\n"
            "           |-POT:%d\n"
            "           |-ORG:%d\n"
            "           |-ALM:%d\n"
            "           |-WAN:%d\n"
            "           |-SVN:%d\n"
            "           |-VIR:%d\n"
            "\n", AxisNo, AxisDI.NOT, AxisDI.POT, AxisDI.ORG,
            AxisDI.ALM, AxisDI.WAN, AxisDI.SVN, AxisDI.VIR);
}
```


7.3.9. ECAT_McGetAxisHomeState

Description:

Get Home state of a specific axis.

Syntax:

int32_t ECAT_McGetAxisHomeState(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *State)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
AxisNo	uint16_t	IN	Axis number
State	uint16_t *	OUT	ECAT_McAxisHome succeed after ECAT_McInit 0:N 1:Y

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t State;
ret = ECAT_McGetAxisHomeState(DeviceNo, AxisNo, &State);
if(ret < 0)
{
	printf("Failed to get axis home state:%d\n",ret);
}
```


7.4. Axis Homing

7.4.1. ECAT_McSetAxisHomeMethod

Description:

Set the homing method of an axis.

Syntax:

int32_t ECAT_McSetAxisHomeMethod(uint16_t DeviceNo, uint16_t AxisNo, int32_t Method)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Method	int32_t	IN	Homing method (Refer to the drive user manual)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
int32_t Method = 1;
...
ret = ECAT_McSetAxisHomeMethod(DeviceNo, AxisNo, Method);
if(ret < 0)
{
    printf("Failed to set axis home method:%d\n", ret);
}
else
{
    printf("Set axis home method successfully!\n");
}
```

7.4.2. ECAT_McGetAxisHomeMethod

Description:

Get the homing method of an axis. Please refer to the user manual of this CiA402 servo drive defined as this axis.

Syntax:

int32_t ECAT_McGetAxisHomeMethod(uint16_t DeviceNo, uint16_t AxisNo, int32_t *Method)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Method	int32_t*	OUT	Homing method

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
int32_t Method;
...
ret = ECAT_McGetAxisHomeMethod(DeviceNo, AxisNo, &Method);
if(ret < 0)
{
        printf("Failed to get axis home method:%d\n", ret);
}
else
{
        printf("Axis[%u] Home Method:%d\n", AxisNo, Method);
}
```


7.4.3. ECAT_McSetAxisHomeSpeed

Description:

Set the homing speed settings of an axis.

Syntax:

int32_t ECAT_McSetAxisHomeSpeed(uint16_t DeviceNo, uint16_t AxisNo, double SeachSw, double SeachZr)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
SeachSw	double	IN	Speed during search for switch (Unit: user unit/s)
SeachZr	double	IN	Speed during search for z phase signal (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double SeachSw = 100.0;
double SeachZr = 10.0;
...
ret = ECAT_McSetAxisHomeSpeed(DeviceNo, AxisNo, SeachSw, SeachZr);
if(ret < 0)
{
    printf("Failed to set axis home speed:%d\n", ret);
}
else
{
    printf("Set axis home speed successfully!\n");
}
```

7.4.4. ECAT_McGetAxisHomeSpeed

Description:

Get the homing speed settings of an axis.

Syntax:

int32_t ECAT_McGetAxisHomeSpeed(uint16_t DeviceNo, uint16_t AxisNo, double *SeachSw, double *SeachZr)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
SeachSw	Double*	OUT	Speed during search for switch (Unit: user unit/s)
SeachZr	Double*	OUT	Speed during search for z phase signal (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double SeachSw;
double SeachZr;
...
ret = ECAT_McGetAxisHomeSpeed(DeviceNo, AxisNo, &SeachSw, &SeachZr);
if(ret < 0)
{
	printf("Failed to get axis home speed:%d\n", ret);
}
else
{
	printf("Axis[%u] Home Speed [Search Switch:%f] / [Search Zero:%f] \n", AxisNo, SeachSw, SeachZr);
}
```


7.4.5. ECAT_McSetAxisHomeAcc

Description:

Set homing acceleration of an axis.

Syntax:

int32_t ECAT_McSetAxisHomeAcc(uint16_t DeviceNo, uint16_t AxisNo, double Acc)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Acc	double	IN	Homing Acceleration (Unit: user unit/s^2)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Acc = 1000.0;
...
ret = ECAT_McSetAxisHomeAcc(DeviceNo, AxisNo, Acc);
if(ret < 0)
{
	printf("Failed to set axis home acceleration:%d\n", ret);
}
else
{
	printf("Set axis home acceleration successfully!\n");
}
```


7.4.6. ECAT_McGetAxisHomeAcc

Description:

Get homing acceleration of an axis.

Syntax:

int32_t ECAT_McGetAxisHomeAcc(uint16_t DeviceNo, uint16_t AxisNo, double *Acc)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Acc	Double*	OUT	Homing Acceleration (Unit: user unit/s^2)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double Acc;
...
ret = ECAT_McSetAxisHomeAcc(DeviceNo, AxisNo, &Acc);
if(ret < 0)
{
    printf("Failed to get axis home acceleration:%d\n", ret);
}
else
{
    printf("Axis[%u] Home Acceleration:%f\n", AxisNo, Acc);
}
```


7.4.7. ECAT_McSetAxisHomeOffset

Description:

Set home offset to an axis.

Syntax:

int32_t ECAT_McSetAxisHomeOffset(uint16_t DeviceNo, uint16_t AxisNo, double Offset)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Offset	double	IN	Home offset (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double HomeOffset = 5.0;
...
ret = ECAT_McSetAxisHomeOffset(DeviceNo, AxisNo, HomeOffset);
if(ret < 0)
{
	printf("Failed to set axis home offset:%d\n", ret);
}
else
{
	printf("Set axis home offset successfully!\n");
}
```

7.4.8. ECAT_McGetAxisHomeOffset

Description:

Get home offset of an axis.

Syntax:

int32_t ECAT_McGetAxisHomeOffset(uint16_t DeviceNo, uint16_t AxisNo, double *Offset)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Offset	Double*	OUT	Home offset (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
double HomeOffset;
ret = ECAT_McGetAxisHomeOffset(DeviceNo, AxisNo, &HomeOffset);
if(ret < 0)
{
    printf("Failed to get axis home offset:%d\n", ret);
}
else
{
    printf("Axis[%u] Home Offset:%f\n", AxisNo, HomeOffset);
}
```


7.4.9. ECAT_McAxisHome

Description:

Start home motion of an axis.

Syntax:

int32_t ECAT_McAxisHome(uint16_t DeviceNo, uint16_t AxisNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
int32_t Method = 1;
double SeachSw = 100.0;
double SeachZr = 10.0;
double Acc = 1000.0;
double HomeOffset = 5.0;

...
ret = ECAT_McSetAxisHomeMethod(DeviceNo, AxisNo, Method);
if(ret < 0)
{
    printf("Failed to set axis home method:%d\n", ret);
    return;
}

ret = ECAT_McSetAxisHomeSpeed(DeviceNo,AxisNo, SeachSw, SeachZr);
if(ret < 0)
{
    printf("Failed to set axis home speed:%d\n", ret);
    return;
}

ret = ECAT_McSetAxisHomeAcc(DeviceNo, AxisNo, Acc);
if(ret < 0)
{
    printf("Failed to set axis home acceleration:%d\n", ret);
    return;
}

ret = ECAT_McSetAxisHomeOffset(DeviceNo, AxisNo, HomeOffset);
if(ret < 0)
{
```



```
        printf("Failed to set axis home offset:%d\n", ret);
        return;
}

ret = ECAT_McAxisHome(DeviceNo, AxisNo);
if(ret < 0)
{
        printf("Failed to start axis home:%d\n", ret);
}
else
{
        do
        {
                sleep(1);
                ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        }while(State == MC_AS_HOMING) //Homing

        if(State == MC_AS_STANDSTILL) //StandStill
                printf("Axis homing successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
                printf("Axis error stop\n");
        }
}
```


7.5. Axis Moving

7.5.1. ECAT_McAxisErrorReset

Description:

Reset the error state of an axis.

Syntax:

int32_t ECAT_McAxisErrorReset(uint16_t DeviceNo, uint16_t AxisNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(ret < 0)
{
	printf("Failed to get axis state:%d\n", ret);
}
else
{
	if(State == MC_AS_ERRORSTOP) //ErrorStop
	{
		ret = ECAT_McAxisErrorReset(DeviceNo, AxisNo);
		if(ret < 0)
		{
			printf("Failed to reset axis error:%d\n", ret);
		}
		else
		{
			printf("Reset axis error successfully!\n");
		}
	}
}
```


7.5.2. ECAT_McAxisMoveAbs

Description:

Start an absolute position motion of an axis.

Syntax:

int32_t ECAT_McAxisMoveAbs(uint16_t DeviceNo, uint16_t AxisNo, double Pos, double Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Pos	double	IN	Absolute position (Unit: user unit)
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;
...
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
	printf("Failed to set axis acceleration time:%d\n", ret);
	return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
	ret = ECAT_McAxisMoveAbs(DeviceNo, AxisNo, AxisPos, AxisVel);
	if(ret < 0)
	{
		printf("Failed to start axis move abs:%d\n", ret);
	}
	else
	{

		do
		{
			sleep(1);
			ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
		}while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

		if(State == MC_AS_STANDSTILL) //StandStill
			printf("Axis move successfully!\n");
		else if(State == MC_AS_ERRORSTOP) //ErrorStop
```

```
            {
                printf("Axis error stop\n");
            }
        }
    }
```

7.5.3. ECAT_McAxisMoveRel

Description:

Start a relative position motion of an axis.

Syntax:

int32_t ECAT_McAxisMoveRel(uint16_t DeviceNo, uint16_t AxisNo, double Pos, double Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Pos	double	IN	Relative distance (Unit: user unit)
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;

ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
        printf("Failed to set axis acceleration time:%d\n", ret);
        return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
        ret = ECAT_McAxisMoveRel(DeviceNo, AxisNo, AxisPos, AxisVel);
        if(ret < 0)
        {
                printf("Failed to start axis move rel:%d\n", ret);
        }
        else
        {
                do
                {
                        sleep(1);
                        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
                }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

                if(State == MC_AS_STANDSTILL) //StandStill
                        printf("Axis move successfully!\n");
                else if(State == MC_AS_ERRORSTOP) //ErrorStop
                {
```



```
                printf("Axis error stop\n");
            }
        }
    }
```


7.5.4. ECAT_McAxisChangePos

Description:

When the specified axis is in motion, this motion command can be used to change the end position.

Syntax:

int32_t ECAT_McAxisChangePos (uint16_t DeviceNo, uint16_t AxisNo, double Pos)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Pos	double	IN	End positon (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;

ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
	printf("Failed to set axis acceleration time:%d\n", ret);
	return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
	ret = ECAT_McAxisMoveRel(DeviceNo, AxisNo, AxisPos, AxisVel);
	if(ret < 0)
	{
		printf("Failed to start axis move rel:%d\n", ret);
	}
	else
	{
		sleep(3);
		ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
		if (State == MC_AS_DISCRETEMOTION) //DiscreteMotion
		{
			AxisPos = 20.0;
			ret = ECAT_McAxisChangePos(DeviceNo, AxisNo, AxisPos);
			if(ret < 0)
			{
				printf("Failed to call axis cahnge position function:%d\n", ret);
			}
```



```
        }

        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
```


7.5.5. ECAT_McAxisChangeVel

Description:

When the specified axis is in motion, this motion command can be used to change the velocity.

Syntax:

int32_t ECAT_McAxisChangeVel (uint16_t DeviceNo,uint16_t AxisNo,double Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Vel	double	IN	Velocity (Unit: $\text{user unit}/s$)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;
...
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo,Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveRel(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move rel:%d\n", ret);
    }
    else
    {
        sleep(3);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        if (State == MC_AS_DISCRETEMOTION) //DiscreteMotion
        {
            AxisVel = 5.0;
            ret = ECAT_McAxisChangeVel(DeviceNo, AxisNo, AxisVel);
            if(ret < 0)
            {
                printf("Failed to call axis cahnge velocity function:%d\n", ret);
            }
```



```
            }

            do
            {
                sleep(1);
                ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
            }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

            if(State == MC_AS_STANDSTILL) //StandStill
                printf("Axis move successfully!\n");
            else if(State == MC_AS_ERRORSTOP) //ErrorStop
            {
                printf("Axis error stop\n");
            }
        }
    }
```

7.5.6. ECAT_McAxisMoveVel

Description:

Start a never ending movement with a specified velocity.

Note:Velocity control mode.

Syntax:

int32_t ECAT_McAxisMoveVel(uint16_t DeviceNo, uint16_t AxisNo, double Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisVel = 2;

ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveVel(DeviceNo, AxisNo, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move vel:%d\n", ret);
    }
}
```


7.5.7. ECAT_McAxisMoveVelByPos

Description:

Start a never ending movement with a specified velocity.

Note:Position control mode.

Syntax:

int32_t ECAT_McAxisMoveVel(uint16_t DeviceNo, uint16_t AxisNo, double Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisVel = 2;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveVelByPos(DeviceNo, AxisNo, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move vel:%d\n", ret);
    }
}
```

7.5.8. ECAT_McAxisGearIn

Description:

Start a gear synchronization motion with a speed ratio between a slave axis and a master axis.

Syntax:

int32_t ECAT_McAxisGearIn(uint16_t DeviceNo, uint16_t MasterNo, uint16_t SlaveNo, int32_t RatioNum, uint32_t RationDen, uint16_t SyncSource)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
MasterNo	uint16_t	IN	Master axis
SlaveNo	uint16_t	IN	Slave axis
RatioNum	int32_t	IN	Gear ratio numerator
RationDen	uint32_t	IN	Gear ratio denominator
SyncSource	uint16_t	IN	Slave reference source for synchronization (As show in Table 7.3)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.3: Source for synchronization

Macro Definition	Value	Description
MC_AXIS_SYNC_SOURCE_SET_VALUE	0	Synchronization on command value of the master
MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE	1	Synchronization on actual value of the master

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double AxisPos = 10.0;
double AxisVel = 4;
int32_t RatioNum = 1;
uint32_t RationDen = 2;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;

...
ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisGearIn(DeviceNo, MasterNo, SlaveNo, RatioNum, RationDen, SyncSource)
    if(ret < 0)
    {
        printf("Axis gearin is falied:%d\n",ret);
        return;
    }
}

ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
```



```
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
```

7.5.9. ECAT_McAxisGearOut

Description:

Disengages the slave axis from the master axis.

Syntax:

int32_t ECAT_McAxisGearOut(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t Stop)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
SlaveNo	uint16_t	IN	Slave axis
Stop	uint16_t	IN	0: Constant velocity motion 1: Stop 2: Quick stop

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t SlaveNo = 1;
uint16_t Stop = 0;
uint32_t State;

...
ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_SYNCHRONIZEDMOTION)
{
    ret = ECAT_McAxisGearOut(DeviceNo, SlaveNo, Stop)
    if(ret < 0)
    {
        printf("Axis gearout is falied:%d\n", ret);
        return;
    }
}
```

7.5.10. ECAT_McAxisMoveProfile

Description:

Start profile position motion of an axis.

Syntax:

int32_t ECAT_McAxisMoveProfile(uint16_t DeviceNo, uint16_t AxisNo, uint16_t ProfileNo, uint16_t TotalStep)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
ProfileNo	uint16_t	IN	Profile buffer number
TotalStep	uint16_t	IN	Total moving steps

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProfileNo = 0;
uint32_t State;
uint16_t TotalStep = 1000;
...
ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
	ret = ECAT_McAxisMoveProfile(DeviceNo, AxisNo, ProfileNo, TotalStep);
	if(ret < 0)
	{
		printf("Failed to start axis move profile:%d\n", ret);
	}
	else
	{
		do
		{
			sleep(1);
			ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
		}while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion
		if(State == MC_AS_STANDSTILL) //StandStill
			printf("Axis move successfully!\n");
		else if(State == MC_AS_ERRORSTOP) //ErrorStop
		{
			printf("Axis error stop\n");
		}
	}
}
```


7.5.11. ECAT_McAxisMoveProfileCSV

Description:

Start profile position motion of an axis. A file contains all the position data. Its format is shown in Figure 7.1.

Syntax:

int32_t ECAT_McAxisMoveProfileCSV(uint16_t DeviceNo, uint16_t AxisNo, uint16_t ProfileNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number
ProfileNo	uint16_t	IN	Profile buffer number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t ProfileNo = 0;
uint32_t State;

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveProfileCSV(DeviceNo, AxisNo, ProfileNo);
    if(ret < 0)
    {
        printf("Failed to start axis move profile CSV:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion
        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
```

7.5.12. ECAT_McAxisCamIn

Description:

Start E-CAM synchronization motion.

Syntax:

int32_t ECAT_McAxisCamIn(uint16_t DeviceNo, uint16_t MasterNo, uint16_t SlaveNo, uint16_t TableNo, uint16_t SyncSource, double MasterInterval, double SlaveScaling)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
MasterNo	uint16_t	IN	Master axis number
SlaveNo	uint16_t	IN	Slave axis number
TableNo	int16_t	IN	E-CAM table number
SyncSource	uint16_t	IN	Slave reference source for synchronization (As show in Table 7.3)
MasterInterval	double	IN	Master Interval
SlaveScaling	double	IN	Slave position output ratio

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

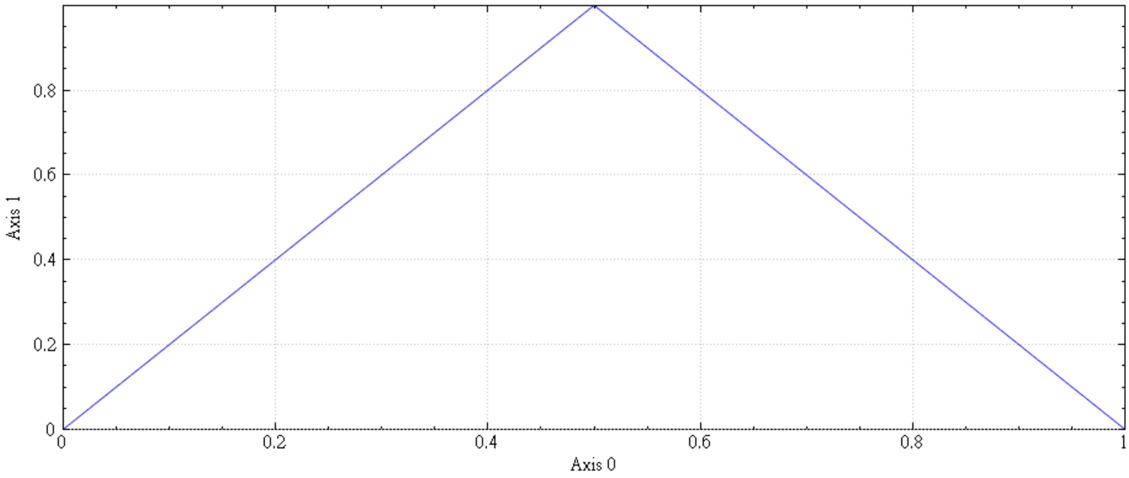
[C/C++]

```
int32_t ret;
uint16_t TableNo = 0;
uint32_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double Data[100];
uint16_t DataSize = 1000;
uint8_t SlaveAbs = 0; //Relative
double MasterInterval = 0.002;
double SlaveScaling = 1.0;
double AxisPos = 3.0;
double AxisVel = 1;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;
/**
***Write E-CAM Table data to Data[1000]
**/
ret = ECAT_McSetEcamTable(DeviceNo, TableNo, Data, DataSize);
if(ret < 0)
{
	printf("Failed to set E-CAM table data:%d\n", ret);
	return;
}

ret = ECAT_McConfigEcamTable(DeviceNo, TableNo, SlaveAbs);
if(ret < 0)
{
	printf("Failed to configure E-CAM table parameter:%d\n", ret);
	return;
}
ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
	ret = ECAT_McAxisCamIn(DeviceNo, MasterNo, SlaveNo, TableNo
			, SyncSource, MasterInterval, SlaveScaling)
```

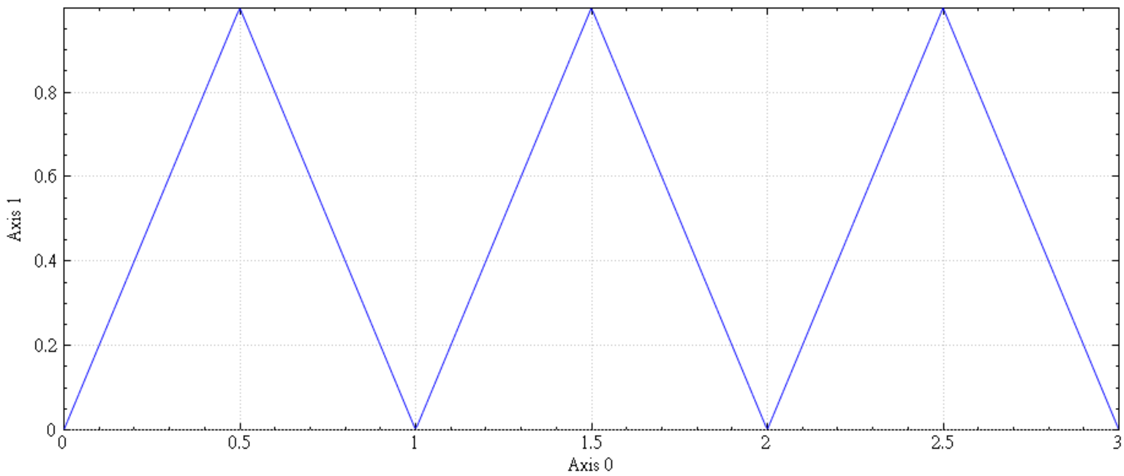


```
        if(ret < 0)
        {
                printf("Axis camin is falied:%d\n", ret);
                return;
        }
}
ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
        ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
        if(ret < 0)
        {
                printf("Failed to start axis move abs:%d\n", ret);
        }   else   {
                do
                {
                        sleep(1);
                        ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
                }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

                if(State == MC_AS_STANDSTILL) //StandStill
                        printf("Axis move successfully!\n");
                else if(State == MC_AS_ERRORSTOP) //ErrorStop
                {
                        printf("Axis error stop\n");
                }
        }
}
```

E-CAM Table:



E-CAM synchronization motion diagram:



7.5.13. ECAT_McAxisCamPhaseShift

Description:

Set the E-CAM synchronization motion master phase shift.

Syntax:

int32_t ECAT_McAxisCamPhaseShift(uint16_t DeviceNo, uint16_t SlaveNo, double PhaseShift)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave axis number
PhaseShift	double	IN	Master phase shift

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

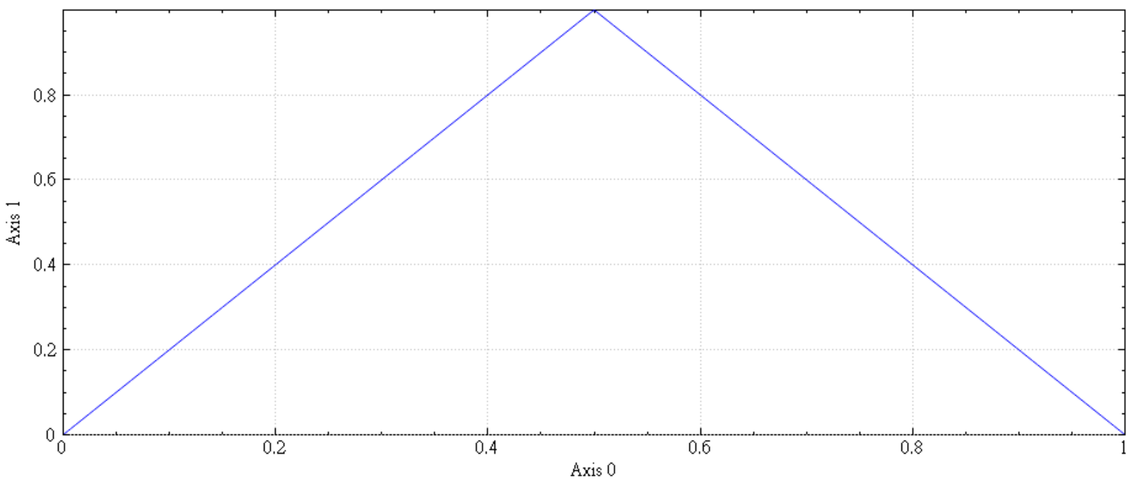
[C/C++]

```
int32_t ret;
uint16_t TableNo = 0;
uint32_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double Data[100];
uint16_t DataSize = 1000;
uint8_t SlaveAbs = 0; //Relative
double MasterInterval = 0.002;
double SlaveScaling = 1.0;
double AxisPos = 3.0;
double AxisVel = 1;
double PhaseShift = -0.5;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;
/**
***Write E-CAM Table data to Data[1000]
**/
ret = ECAT_McSetEcamTable(DeviceNo, TableNo, Data, DataSize);
if(ret < 0)
{
      printf("Failed to set E-CAM table data:%d\n", ret);
      return;
}

ret = ECAT_McConfigEcamTable(DeviceNo, TableNo, SlaveAbs);
if(ret < 0)
{
      printf("Failed to configure E-CAM table parameter:%d\n", ret);
      return;
}
ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
      ret = ECAT_McAxisCamIn(DeviceNo, MasterNo, SlaveNo, TableNo
```

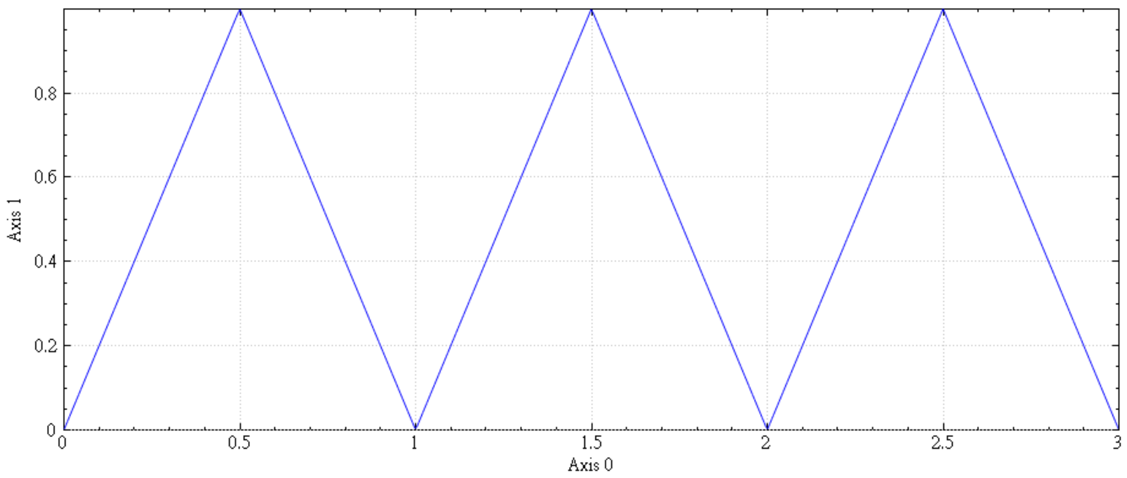


```
                , SyncSource, MasterInterval, SlaveScaling)
        if(ret < 0)
        {
            printf("Axis camin is falied:%d\n", ret);
            return;
        }
}

ret = ECAT_McAxisCamPhaseShift(DeviceNo, SlaveNo, PhaseShift)
if(ret < 0)
{
        printf("Failed to set cam phase shift:%d\n", ret);
        return;
}

ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
        ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
        if(ret < 0)
        {
            printf("Failed to start axis move abs:%d\n", ret);
        }  else  {
            do
            {
                sleep(1);
                ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
            }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

            if(State == MC_AS_STANDSTILL) //StandStill
                printf("Axis move successfully!\n");
            else if(State == MC_AS_ERRORSTOP) //ErrorStop
            {
                printf("Axis error stop\n");
            }
        }
}
```

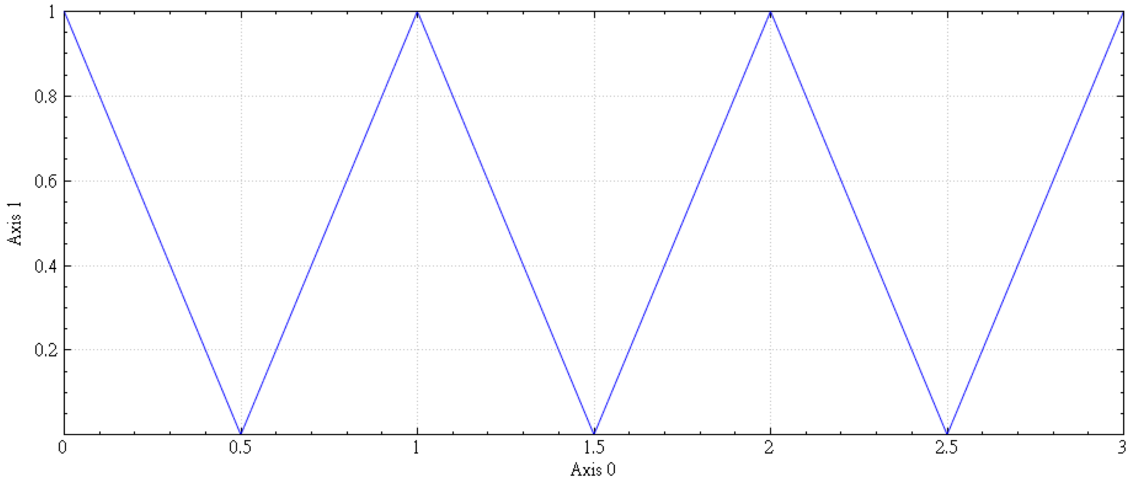

E-CAM Table:



E-CAM synchronization motion:



After setting the phase shift of the master:



1
0.8
0.6
0.4
0.2
Axis 1
0
0.5
1
1.5
2
2.5
3
Axis 0

7.5.14. ECAT_McAxisCamOut

Description:

Stop E-CAM synchronization motion.

Syntax:

int32_t ECAT_McAxisCamOut(uint16_t DeviceNo, uint16_t SlaveNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
SlaveNo	uint16_t	IN	Slave axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t SlaveNo = 0;

ret = ECAT_McAxisCamOut(DeviceNo, SlaveNo);
if (ret != 0) {
	printf("Failed to cam out:%d\n", ret);
}
```


7.5.15. ECAT_McAxisTangentInGroup

Description:

Start tangent motion.

Syntax:

int32_t ECAT_McAxisTangentInGroup(uint16_t DeviceNo, uint16_t AxisNo, uint16_t GroupNo, double Angle, double Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
AxisNo	uint16_t	IN	Axis number
GroupNo	uint16_t	IN	Group number
Angle	int16_t	IN	Tangent angle (Unit:degree)
Vel	uint16_t	IN	Rotate to tangent angle velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint16_t TangentInAxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;
double AxisAngle, AxisVel, CircAngle;
uint32_t task_index;
bool task_stop;

/*****************************************************************************/
int32_t check_grp_state(void)
{
    int32_t ret;
    uint32_t State;
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    if (ret != 0) {
        printf("Failed to get group state:%d\n").arg(ret);
        return -1;
    } else {
        if(State == MC_GS_ERRORSTOP) {
            printf("Group error stop\n");
            return -1;
        } else if(State == MC_GS_STANDBY) {
            return 0;
        } else
            return 1;
    }
}
/*****************************************************************************/
int32_t check_axis_tangent_in(void)
{
```



```
    int32_t ret;
    uint32_t State;
    ret = ECAT_McGetAxisState(DeviceNo, TangentInAxisNo, &State);
    if (ret != 0) {
        printf("Failed to get axis state:%d\n").arg(ret);
        return -1;
    } else {
        if(State == MC_AS_ERRORSTOP) {
            printf("Group error stop\n");
            return -1;
        } else if(State == MC_AS_SYNCHRONIZEDMOTION) {
            return 0;
        } else
            return 1;
    }
}
/**********************************************************************************/
int main()
{
    AxisNo = 0;
    ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
    if(ret < 0)
    {
        printf("Failed to add axis to group:%d\n", ret);
        return -1;
    }
    AxisNo = 1;
    ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
    if(ret < 0)
    {
        printf("Failed to add axis to group:%d\n", ret);
        return -1;
    }
    ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
    if(ret < 0)
    {
        printf("Failed to set group command mode:%d\n", ret);
        return -1;
```



```
    }

    ret = check_grp_state();
    if (ret == -1)
        return -1

    task_index = 0;
    task_stop = false;
    TangentInAxisNo = 2;

    while(!task_stop) {
        switch(task_index) {
            case 0:
                GroupPos[0] = 0.0;
                GroupPos[1] = 0.0;
                GroupVel = 5;
                ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
                if(ret != 0) {
                    printf("Failed to add group move line command:%d\n", ret);
                    task_stop = true;
                } else
                    task_index++;
                break;
            case 1:
                ret = check_grp_state();
                if (ret == -1)
                    task_stop = true;
                else if (ret == 0)
                    task_index++;
                break;
            case 2:
                AxisAngle = 90;
                AxisVel = 0.25;
                ret = ECAT_McAxisTangentInGroup(DeviceNo, TangentInAxisNo, GroupNo, AxisAngle,
AxisVel);
                if (ret != 0) {
                    printf("Axis tangent in failed:%d\n", ret);
                    task_stop = true;
```

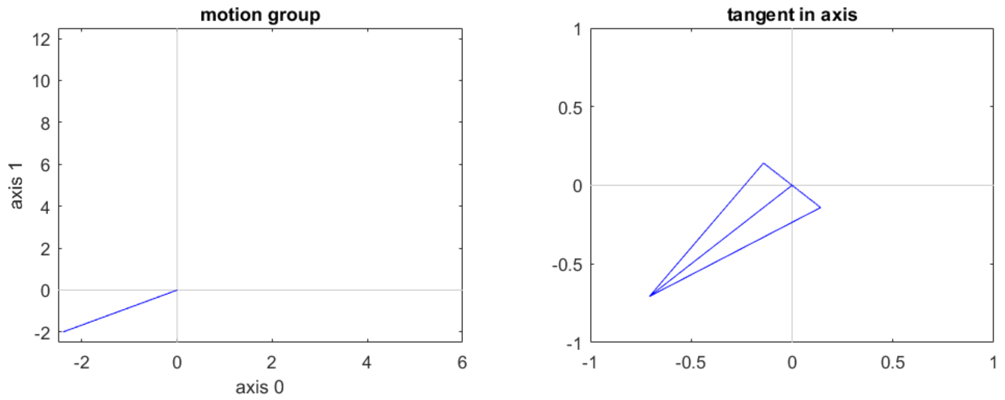


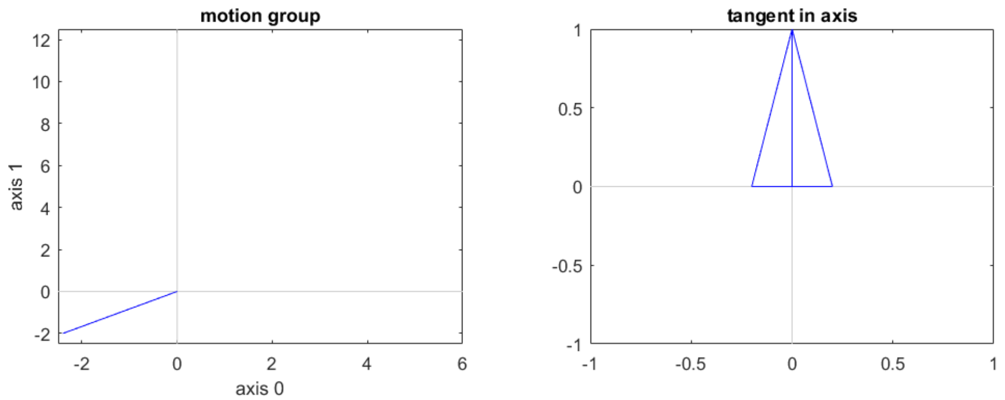
```
            } else
                task_index++;
            break;
        case 3:
            ret = check_axis_tangent_in();
            if (ret == -1)
                task_stop = true;
            else if (ret == 0)
                task_index++;
            break;
        case 4:
            GroupPos[0] = 0.0;
            GroupPos[1] = 10.0;
            GroupVel = 5;
            ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
            if(ret != 0) {
                printf("Failed to add group move line command:%d\n", ret);
                task_stop = true;
            } else
                task_index++;
            break;
        case 5:
            ret = check_grp_state();
            if (ret == -1)
                task_stop = true;
            else if (ret == 0)
                task_index++;
            break;
        case 6:
            AxisAngle = 0;
            AxisVel = 0.25;
            ret = ECAT_McAxisTangentInGroup(DeviceNo, TangentInAxisNo, GroupNo, AxisAngle,
AxisVel);
            if (ret != 0) {
                printf("Axis tangent in failed:%d\n", ret);
                task_stop = true;
            } else
                task_index++;
```

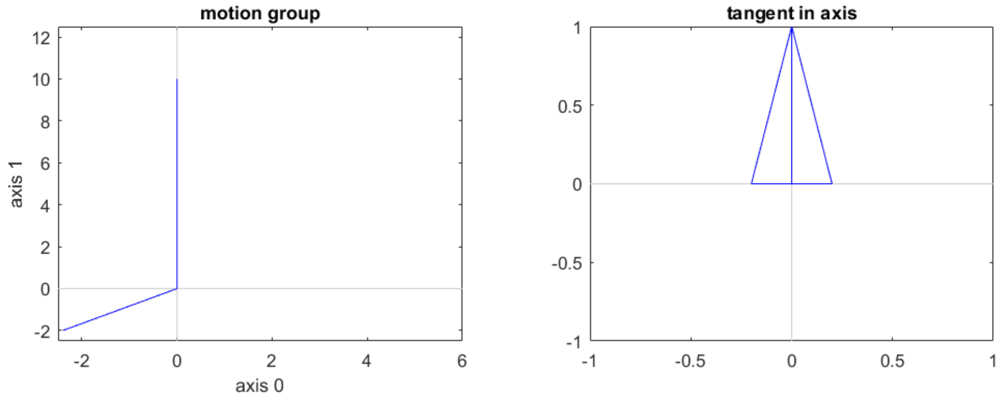


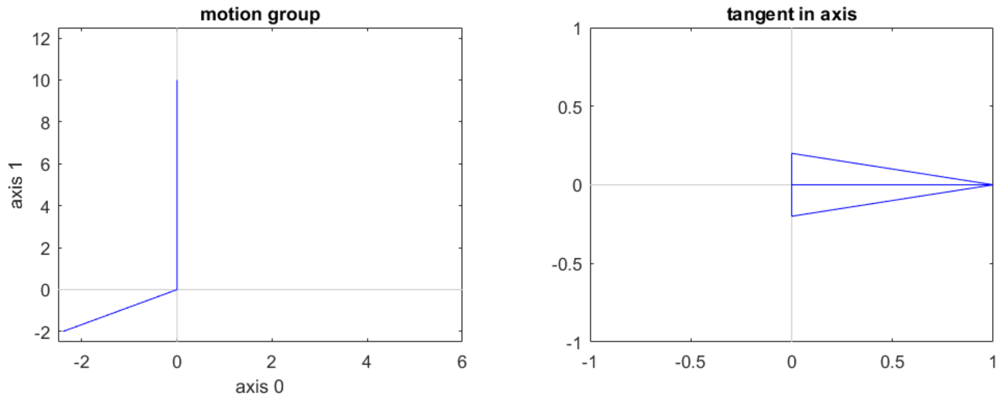
```
                break;
            case 7:
                ret = check_axis_tangent_in();
                if (ret == -1)
                    task_stop = true;
                else if (ret == 0)
                    task_index++;
                break;
            case 8:
                GroupPos[0] = 0.0;
                GroupPos[1] = -2.5;
                GroupVel = 0.5;
                CircAngle = -180;
                ret = ECAT_McGroupMoveCircularRel_CP_Angle(DeviceNo, GroupNo, GroupVel,
CircAngle, GroupPos);
                if (ret != 0) {
                    printf"Group move circular failed:%d\n").arg(ret);
                    task_stop = true;
                } else
                    task_index++;
                break;
            case 9:
                ret = check_grp_state();
                if (ret == -1)
                    task_stop = true;
                else if (ret == 0)
                    task_index++;
                break;
            default:
                task_stop = true;
                break;

        }

        msleep(1);
    }
    return 0;
}
```

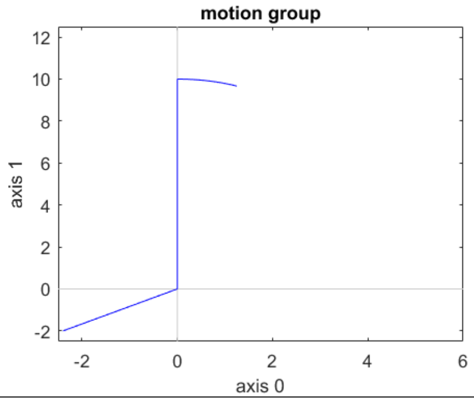
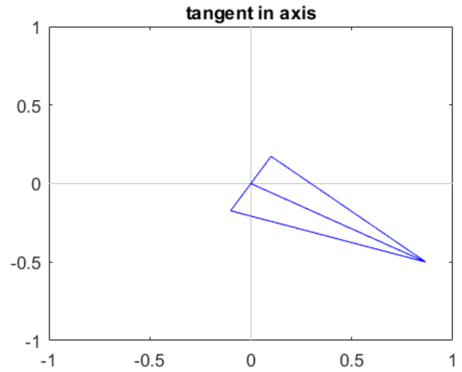
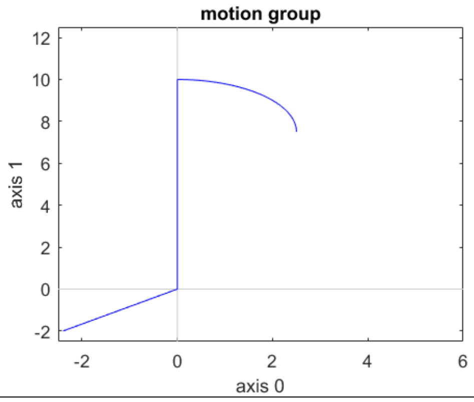
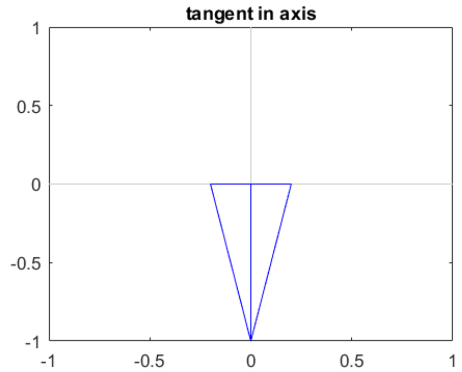
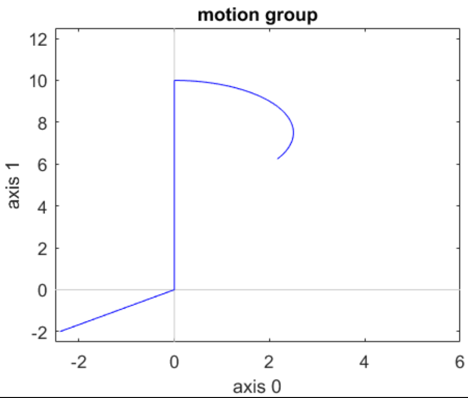
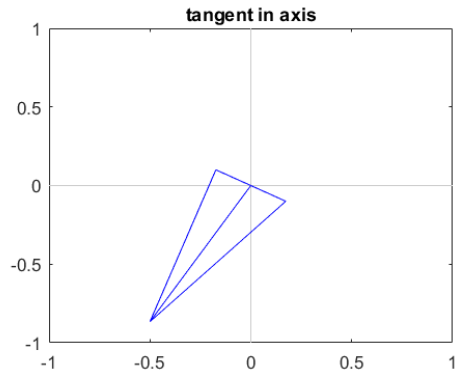
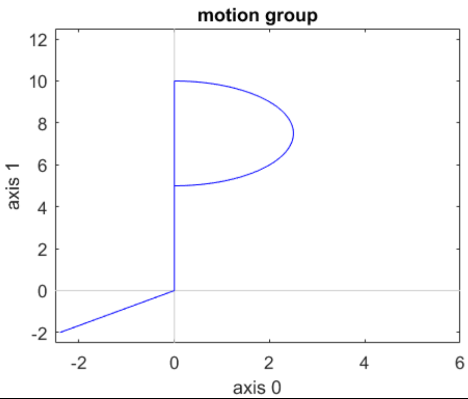
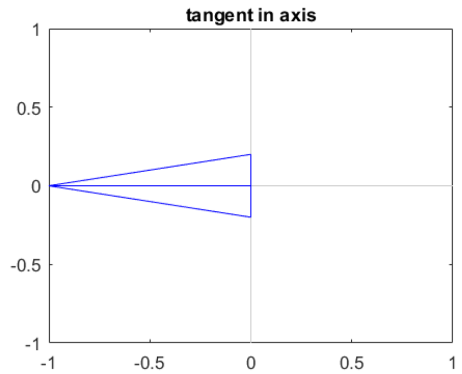

Tangent motion path of example:

Case 0	
cmd	GroupMoveLine to (0, 0)
result	

Case 2	
cmd	AxisTangentInGroup with angle 90
result	

<table>
<tr><th colspan="2">Case 4</th></tr>
<tr><td>cmd</td><td>GroupMoveLine to (0, 10)</td></tr>
<tr><td>result</td><td>
</td></tr>
</table>

<table>
<tr><th colspan="2">Case 6</th></tr>
<tr><td>cmd</td><td>AxisTangentInGroup with angle 0</td></tr>
<tr><td>result</td><td>
</td></tr>
</table>

Case 8	
cmd	GroupMoveCircular to angle -180
result	       

7.5.16. ECAT_McAxisGantryIn

Description:

Start gantry control synchronization motion.

Note:(1) Use ECAT_McAxisStop to stop gantry control synchronization motion

(2)Use ECAT_McSetAxisVelocityFeedForwardGain to adjust feedforward gain to make better result.

Syntax:

int32_t ECAT_McAxisGantryIn(uint16_t DeviceNo, uint16_t MasterNo, uint16_t SlaveNo, int32_t Direction)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
MasterNo	uint16_t	IN	Master axis
SlaveNo	uint16_t	IN	Slave axis
Direction	int32_t	IN	Direction of movement 1:same -1:different

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double AxisPos = 10.0;
double AxisVel = 4;
int32_t Direction = 1;
...
ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisGantryIn(DeviceNo, MasterNo, SlaveNo, Direction)
    if(ret < 0)
    {
        printf("Axis gearin is falied:%d\n",ret);
        return;
    }
}

ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion
```

```
            if(State == MC_AS_STANDSTILL) //StandStill
                printf("Axis move successfully!\n");
            else if(State == MC_AS_ERRORSTOP) //ErrorStop
            {
                printf("Axis error stop\n");
            }
        }
    }
```


7.5.17. ECAT_McAxisGantryMaxPosDiff

Description:

Set maximum position deviation of Master axis and Slave axis,it will trigger error stop while position deviation greater than set value.

Syntax:

int32_t ECAT_McAxisGantryMaxPosDiff(uint16_t DeviceNo, uint16_t SlaveNo, double Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave axis
Value	double	IN	maximum position deviation

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double AxisPos = 10.0;
double AxisVel = 4;
int32_t Direction= 1;
double Value = 1;
uint16_t Status = 1;

ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisGantryIn(DeviceNo, MasterNo, SlaveNo, Direction)
    if(ret < 0)
    {
        printf("Axis gantryin is falied:%d\n", ret);
        return;
    }
    ret = ECAT_McAxisGantryMaxPosDiff(DeviceNo, SlaveNo, Value)
    if(ret < 0)
    {
        printf(" Set Axis gantry Max position deviation is falied:%d\n", ret);
        return;
    }
    ret = ECAT_McAxisGantryMaxPosDiffStatus(DeviceNo, SlaveNo, Status)
    if(ret < 0)
    {
        printf(" Set Axis gantry Max position deviation Status is falied:%d\n", ret);
        return;
    }
}

ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
```



```
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
```


7.5.18. ECAT_McAxisGantryMaxPosDiffStatus

Description:

Set maximum position deviation status.

Syntax:

int32_t ECAT_McAxisGantryMaxPosDiffStatus(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t Status)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
SlaveNo	uint16_t	IN	Slave axis
Status	uint16_t	IN	0:disable 1:enable

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double AxisPos = 10.0;
double AxisVel = 4;
int32_t Direction= 1;
double Value = 1;
uint16_t Status = 1;

ret = ECAT_McGetAxisState(DeviceNo, SlaveNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
	ret = ECAT_McAxisGantryIn(DeviceNo, MasterNo, SlaveNo, Direction)
	if(ret < 0)
	{
		printf("Axis gantryin is falied:%d\n", ret);
		return;
	}
	ret = ECAT_McAxisGantryMaxPosDiff(DeviceNo, SlaveNo, Value)
	if(ret < 0)
	{
		printf(" Set Axis gantry Max position deviation is falied:%d\n", ret);
		return;
	}
	ret = ECAT_McAxisGantryMaxPosDiffStatus(DeviceNo, SlaveNo, Status)
	if(ret < 0)
	{
		printf(" Set Axis gantry Max position deviation Status is falied:%d\n", ret);
		return;
	}
}

ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
```



```
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, MasterNo, &State);
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
```


7.5.19. ECAT_McAxisStop

Description:

Stop an axis with deceleration.

Syntax:

int32_t ECAT_McAxisStop(uint16_t DeviceNo, uint16_t AxisNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;
...
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n",ret);
    }
    else
    {
        sleep(1000);
        ret = ECAT_McAxisStop(DeviceNo, AxisNo);
        if(ret < 0)
        {
            printf("Failed to stop axis move:%d\n", ret);
            return;
        }
        else
        {
            do
```



```
                {
                    sleep(1);
                    ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
                }while(State == MC_AS_STOPPING) //Stopping

                if(State == MC_AS_STANDSTILL) //StandStill
                    printf("Axis move stop successfully!\n");
                else if(State == MC_AS_ERRORSTOP) //ErrorStop
                {
                    printf("Axis error stop\n");
                }
            }
        }
}
```

7.5.20. ECAT_McAxisQuickStop

Description:

Stop an axis quickly.

Syntax:

int32_t ECAT_McAxisQuickStop(uint16_t DeviceNo, uint16_t AxisNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
AxisNo	uint16_t	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double AxisVel = 2;
...
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        sleep(1000);
        ret = ECAT_McAxisQuickStop(DeviceNo, AxisNo);
        if(ret < 0)
        {
            printf("Failed to quickstop axis move:%d\n", ret);
            return;
        }
        else
        {
            do
```

```
                {
                    sleep(1);
                    ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
                }while(State == MC_AS_STOPPING) //Stopping

                if(State == MC_AS_STANDSTILL) //StandStill
                    printf("Axis move stop successfully!\n");
                else if(State == MC_AS_ERRORSTOP) //ErrorStop
                {
                    printf("Axis error stop\n");
                }
            }
        }
    }
```


7.6. Group Parameter Setting

7.6.1. ECAT_McAddAxisToGroup

Description:

Add one axis to a group.

Syntax:

int32_t ECAT_McAddAxisToGroup(uint16_t DeviceNo, uint16_t GroupNo, uint16_t AxisNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number (MC_GROUP_NO_MAX macro is the maximum number)
AxisNo	uint16_t	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
...
AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
```

7.6.2. ECAT_McRemoveAxisFromGroup

Description:

Remove one axis from a group.

Syntax:

int32_t ECAT_McRemoveAxisFromGroup(uint16_t DeviceNo, uint16_t GroupNo, uint16_t AxisNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
AxisNo	uint16_t	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
...
AxisNo = 1;
ret = ECAT_McRemoveAxisFromGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to remove axis from group:%d\n", ret);
}
else
{
	printf("Remove axis from group successfully!\n");
}
```


7.6.3. ECAT_McUngroupAllAxes

Description:

Remove all axes from a group.

Syntax:

int32_t ECAT_McUngroupAllAxes(uint16_t DeviceNo, uint16_t GroupNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
...
ret = ECAT_McUngroupAllAxes(DeviceNo, GroupNo);
if(ret < 0)
{
	printf("Failed to ungroup all axes:%d\n", ret);
}
else
{
	printf("Ungroup all axes successfully!\n");
}
```

7.6.4. ECAT_McSetGroupCmdMode

Description:

Set the group command mode of a group immediately. The command mode will decide how a motion commnad is processed. There three command modes: aborting, buffered, and blending.

Aborting: A new command will abort the current executing command; then the new command executes immediately.

Buffered: A new command will be push into a command buffer and wait for being executed. The motion kernel program will execute all commands in the command buffer sequentially. Each command is executed and waits until stop, then another command is loaded from the command buffer and is executed by the motion kernel next.

Blending: A new command will be push into a command buffer and wait for being executed. The motion kernel program will execute all commands in the command buffer sequentially. Each command is executed and waits until the begining of deceleration, then another command is loaded from the command buffer and is executed by the motion kernel at the same time. Therefore, previous motion command is partially blending into next one.

Syntax:

int32_t ECAT_McSetGroupCmdMode(uint16_t DeviceNo, uint16_t GroupNo, uint16_t CmdMode)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
CmdMode	uint16_t	IN	Group command mode (As show in Table 7.4)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.4: Group command mode

Macro Definition	Value	Description
MS_GRP_CM_ABORTING	0	Aboring
MS_GRP_CM_BUFFERED	1	Buffered
MS_GRP_CM_BLENDING	2	Blending

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t CmdMode = MS_GRP_CM_BUFFERED;
...
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
}
else
{
    printf("Set group command mode successfully!\n");
}
```


7.6.5. ECAT_McSetGroupCmdModeEx

Description:

Set the group command mode of a group, this command will be push into command buffer in Buffered mode or Blending mode, In Aborting mode, it will change command mode immediately. The command mode will decide how a motion commnad is processed. There three command modes: aborting, buffered, and blending.

Aborting: A new command will abort the current executing command; then the new command executes immediately.

Buffered: A new command will be push into a command buffer and wait for being executed. The motion kernel program will execute all commands in the command buffer sequentially. Each command is executed and waits until stop, then another command is loaded from the command buffer and is executed by the motion kernel next.

Blending: A new command will be push into a command buffer and wait for being executed. The motion kernel program will execute all commands in the command buffer sequentially. Each command is executed and waits until the begining of deceleration, then another command is loaded from the command buffer and is executed by the motion kernel at the same time. Therefore, previous motion command is partially blending into next one.

Syntax:

int32_t ECAT_McSetGroupCmdMode(uint16_t DeviceNo, uint16_t GroupNo, uint16_t CmdMode)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
CmdMode	uint16_t	IN	Group command mode (As show in Table 7.4)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.5: Group command mode

Macro Definition	Value	Description
MS_GRP_CM_ABORTING	0	Aboring
MS_GRP_CM_BUFFERED	1	Buffered
MS_GRP_CM_BLENDING	2	Blending

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t CmdMode = MS_GRP_CM_BUFFERED;
...
ret = ECAT_McSetGroupCmdModeEx(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
	printf("Failed to set group command mode:%d\n", ret);
}
else
{
	printf("Set group command mode successfully!\n");
}
```


7.6.6. ECAT_McGetGroupCmdMode

Description:

Get the group command mode of a group.

Syntax:

int32_t ECAT_McGetGroupCmdMode(uint16_t DeviceNo, uint16_t GroupNo, uint16_t *CmdMode)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
CmdMode	uint16_t*	OUT	Group command mode (As show in Table 7.4)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t CmdMode;
ret = ECAT_McGetGroupCmdMode(DeviceNo, GroupNo, &CmdMode);
if(ret < 0)
{
    printf("Failed to get group command mode:%d\n", ret);
}
else
{
    printf("Group[%u] Command Mode:%u\n", GroupNo, CmdMode);
}
```

7.6.7. ECAT_McSetGroupAccTime

Description:

Set acceleration time of a group.

Syntax:

int32_t ECAT_McSetGroupAccTime(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Time_ms)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Time_ms	uint16_t	IN	Acceleration time (Unit: millisecond)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Time_ms = 500;
ret = ECAT_McSetGroupAccTime(DeviceNo, GroupNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set group acceleration time:%d\n", ret);
}
else
{
    printf("Set group acceleration time successfully!\n");
}
```


7.6.8. ECAT_McGetGroupAccTime

Description:

Get acceleration time of a group.

Syntax:

int32_t ECAT_McGetGroupAccTime(uint16_t DeviceNo, uint16_t GroupNo, uint16_t *Time_ms)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
GroupNo	uint16_t	IN	Group number
Time_ms	uint16_t*	OUT	Acceleration time (Unit: millisecond)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Time_ms;
ret = ECAT_McGetGroupAccTime(DeviceNo, GroupNo, &Time_ms);
if(ret < 0)
{
    printf("Failed to get gruop acceleration time:%d\n", ret);
}
else
{
    printf("group[%u] Acceleration Time(ms):%f\n", GroupNo, Time_ms);
}
```


7.6.9. ECAT_McSetGroupAccDecType

Description:

Set acceleration type of a group.

Syntax:

int32_t ECAT_McSetGroupAccDecType(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Type)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Type	uint16_t	IN	Acceleration Type 1:T-Curve 2:S-Curve

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Type = 1; //T-Curve
ret = ECAT_McSetGroupAccDecType (DeviceNo, GroupNo, Type);
if(ret < 0)
{
	printf("Failed to set group AccDecType:%d\n", ret);
}
else
{
	printf("Set group AccDecType successfully!\n");
}
```


7.6.10. ECAT_McGetGroupAccDecType

Description:

Get acceleration type of a group.

Syntax:

int32_t ECAT_McGetGroupAccDecType(uint16_t DeviceNo, uint16_t GroupNo, uint16_t * Type)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Type	uint16_t*	OUT	Acceleration Type 1:T-Curve 2:S-Curve

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Type;
ret = ECAT_McGetGroupAccDecType(DeviceNo, GroupNo, &Type);
if(ret < 0)
{
    printf("Failed to get group AccDecType:%d\n",ret);
}
else
{
    printf("group[%u] AccDecType:%f\n", GroupNo, Type);
}
```

7.7. Group Status

7.7.1. ECAT_McGetGroupState

Description:

Get the state of a group.

Syntax:

int32_t ECAT_McGetGroupState(uint16_t DeviceNo, uint16_t GroupNo, uint32_t *State)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
State	uint32_t*	OUT	Group state (As show in Table 7.6)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.6: Group state

Macro Definition	Value	Description
MC_GS_DISABLED	0	Group is disabled
MC_GS_STANDBY	1	Group is stanby
MC_GS_ERRORSTOP	2	Group is stopped because of error
MC_GS_STOPPING	3	Group is stopping
MC_GS_HOMING	4	Reserved
MC_GS_MOVING	5	Group is in motion

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
char buf[512];
uint32_t State;

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(ret < 0)
{
    printf("Failed to get group state:%d\n",ret);
}
else
{
    switch(State)
    {
        case MC_GS_DISABLED:
            sprintf(buf,"Disabled");
            break;
        case MC_GS_STANDBY:
            sprintf(buf,"Standby");
            break;
        case MC_GS_ERRORSTOP:
            sprintf(buf,"ErrorStop");
            break;
        case MC_GS_STOPPING:
            sprintf(buf,"Stopping");
            break;
        case MC_GS_HOMING:
            sprintf(buf,"Homing");
            break;
        case MC_GS_MOVING:
            sprintf(buf,"Moving");
            break;
        default:
            sprintf(buf,"Invalid");
```

```
    }
    printf("Group State:%s\n", buf);
}
```

7.7.2. ECAT_McGetGroupCmdBuffer

Description:

Get the number of commands buffered inside a group buffer.

Syntax:

int32_t ECAT_McGetGroupCmdBuffer(uint16_t DeviceNo, uint16_t GroupNo, uint16_t *Buffer)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Buffer	uint16_t*	OUT	Number of commands to be executed

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t BufferCnt;

ret = ECAT_McGetGroupCmdBuffer(DeviceNo, GroupNo, &BufferCnt);
if(ret < 0)
{
	printf("Failed to get group command buffer:%d\n", ret);
}
else
{
	printf("Group command buffer:%u\n", BufferCnt);
}
```


7.7.3. ECAT_McSetGroupVelLimitStatus

Description:

Set the velocity limit state of a group. This function is used to enable or disable a velocity limit check. If state is "Enable", each axis speed in this group will be checked for not over a defined maximum value. If one of these axes is over the speed limit value, this group speed will be recalculated to meet the speed limit requirement.

Syntax:

int32_t ECAT_McSetGroupVelLimitStatus(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Status)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Status	uint16_t	IN	Velocity limit state of a group. 0: disable 1: enable

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t State = 1;
ret = ECAT_McSetGroupVelLimitStatus(DeviceNo, GroupNo, State);
if(ret < 0)
{
	printf("Failed to Set group velocity limit status:%d\n", ret);
}
```

7.7.4. ECAT_McGetGroupVelLimitStatus

Description:

Get the velocity limit state of a group.

Syntax:

int32_t ECAT_McGetGroupVelLimitStatus(uint16_t DeviceNo, uint16_t GroupNo, uint16_t *Status)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Status	uint16_t*	OUT	velocity limit state of a specific group. 0: disable 1: enable

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t State;
ret = ECAT_McGetGroupVelLimitStatus(DeviceNo, GroupNo, &State);
if(ret < 0)
{
    printf("Failed to get group group velocity limit status:%d\n", ret);
}
else
{
    printf("Group velocity limit status:%u\n", State);
}
```


7.7.5. ECAT_McSetGroupVelLimitValue

Description:

Set the velocity limit value of each axis in a group.

Note: This velocity limit requirement right now is valid only for two functions: *ECAT_McGroupMoveLineAbs_PT* and *ECAT_McGroupMoveLineRel_PT*.

Syntax:

int32_t ECAT_McSetGroupVelLimitValue(uint16_t DeviceNo, uint16_t GroupNo, double Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Value	double	IN	velocity limit of each of axis in a group

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
double Value = 100;
ret = ECAT_McSetGroupVelLimitValue(DeviceNo, GroupNo, Value);
if(ret < 0)
{
    printf("Failed to Set group velocity limit value:%d\n", ret);
}
```

7.7.6. ECAT_McGetGroupVelLimitValue

Description:

Get the velocity limit of each of axis in a group.

Note: This velocity limit requirement right now is valid only for two functions: *ECAT_McGroupMoveLineAbs_PT* and *ECAT_McGroupMoveLineRel_PT*.

Syntax:

int32_t ECAT_McGetGroupVelLimitValue(uint16_t DeviceNo, uint16_t GroupNo, double *Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Value	double*	OUT	velocity limit of each of axis in a group

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
double Value;
ret = ECAT_McGetGroupVelLimitValue(DeviceNo, GroupNo, &Value);
if(ret < 0)
{
    printf("Failed to get group group velocity limit value:%d\n", ret);
}
else
{
    printf("Group velocity limit value:%f\n", Value);
}
```

7.8. Group Moving

7.8.1. ECAT_McGroupMoveLineAbs

Description:

Start an absolute linear interpolation motion of a group. Positions and a velocity are requested to enter.

Syntax:

int32_t ECAT_McGroupMoveLineAbs(uint16_t DeviceNo, uint16_t GroupNo, double Pos[], double Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Position arrary of a group Each array element is the absolute position of a axis. (Unit: user unit)
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t Time_ms = 500;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

...
AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n",ret);
	return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n",ret);
	return;
}

ret = ECAT_McSetGroupAccTime(DeviceNo, GroupNo, Time_ms);
if(ret < 0)
{
	printf("Failed to set group acceleration time:%d\n",ret);
	return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
```



```
if(ret < 0)
{
	printf("Failed to set group command mode:%d\n", ret);
	return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
	//Command 1
	GroupPos[0] = 10.0;
	GroupPos[1] = 20.0;
	GroupVel = 5;
	ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
	if(ret < 0)
	{
		printf("Failed to add group move line command:%d\n", ret);
	}
	//Command 2
	GroupPos[0] = 30.0;
	GroupPos[1] = 50.0;
	GroupVel = 10;
	ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
	if(ret < 0)
	{
		printf("Failed to add group move line command:%d\n", ret);
	}

	do
	{
		sleep(1);
		ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

	}while(State == MC_GS_MOVING) //Moving

	if(State == MC_GS_STANDBY) //Standby
		printf("Group move line successfully!\n");
	else if(State == MC_GS_ERRORSTOP) //ErrorStop
```

```
        {
            printf("Group error stop\n");
        }
}
```

7.8.2. ECAT_McGroupMoveLineRel

Description:

Start an relative linear interpolation motion of a group. Positions and a velocity are requested to enter.

Syntax:

int32_t ECAT_McGroupMoveLineRel(uint16_t DeviceNo, uint16_t GroupNo, double Pos[], double Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Distance arrary of a group Each array element is the relative position of a axis. (Unit: user unit)
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t Time_ms = 500;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

...
AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupAccTime(DeviceNo, GroupNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set group acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
```



```
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }
    //Command 2
    GroupPos[0] = 30.0;
    GroupPos[1] = 50.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
```

```
    {
        printf("Group error stop\n");
    }
}
```


7.8.3. ECAT_McGroupMoveLineAbs_PT

Description:

Start an absolute linear interpolation motion of a group. Positions and a time value are requested to enter. The command speed of each axis is calculated according to positions and the time value.

Syntax:

int32_t ECAT_McGroupMoveLineAbs_PT(uint16_t DeviceNo, uint16_t GroupNo, double Pos[], double Time)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Position arrary for a group Each array element is the absolute position of each axis. (Unit:user unit)
Time	double	IN	Time (Unit: second)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupTime;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
	printf("Failed to set group command mode:%d\n",ret);
	return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
```



```
        //Command 1
        GroupPos[0] = 10.0;
        GroupPos[1] = 20.0;
        GroupTime = 5;
        ret = ECAT_McGroupMoveLineAbs_PT (DeviceNo, GroupNo, GroupPos, GroupTime);
        if(ret < 0)
        {
            printf("Failed to add group move line command:%d\n",ret);
        }
        //Command 2
        GroupPos[0] = 30.0;
        GroupPos[1] = 50.0;
        GroupTime = 10;
        ret = ECAT_McGroupMoveLineAbs_PT(DeviceNo, GroupNo, GroupPos, GroupTime);
        if(ret < 0)
        {
            printf("Failed to add group move line command:%d\n", ret);
        }

        do
        {
            sleep(1);
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

        }while(State == MC_GS_MOVING) //Moving

        if(State == MC_GS_STANDBY) //Standby
            printf("Group move line successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
    }
```

7.8.4. ECAT_McGroupMoveLineRel_PT

Description:

Start a relative linear interpolation motion of a group. Positions and a time value are requested to enter. The command speed of each axis is calculated according to positions and the time value.

Syntax:

int32_t ECAT_McGroupMoveLineRel_PT(uint16_t DeviceNo, uint16_t GroupNo, double Pos[], double Time)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Distance array for a group Each array element is the relative position of a axis. (Unit: user unit)
Time	double	IN	Time (Unit: second)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupTime;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n",ret);
	return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
	printf("Failed to set group command mode:%d\n", ret);
	return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
```



```
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupTime = 5;
    ret = ECAT_McGroupMoveLineRel_PT(DeviceNo, GroupNo, GroupPos,   GroupTime);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }
    //Command 2
    GroupPos[0] = 30.0;
    GroupPos[1] = 50.0;
    GroupTime = 10;
    ret = ECAT_McGroupMoveLineRel_PT(DeviceNo,GroupNo,GroupPos, GroupTime);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

7.8.5. ECAT_McGroupMoveLineAbs_PVT

Description:

Start an absolute PVT motion.

Syntax:

int32_t ECAT_McGroupMoveLineAbs_PVT(uint16_t DeviceNo, uint16_t GroupNo, double Pos[], double Vel[], double Time)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Position data arrary of a group Each array element is the absolute position of an axis. (Unit: user unit)
Vel	double[]	IN	Velocity data arrary of a group Each array element is the veloctiy data of an axis. (Unit: user unit/s)
Time	double	IN	Time (Unit: second)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double AxisPos[MC_AXIS_NO_MAX];
double AxisVel[MC_AXIS_NO_MAX];
double Time;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
```



```
AxisPos[0] = 0.0;
AxisPos[1] = 0.0;
AxisVel[0] = 0.0;
AxisVel[1] = 0.0;
Time = 0;
ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 2
AxisPos[0] = 5.0;
AxisPos[1] = 0.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 1.5;
ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 3
AxisPos[0] = 0.0;
AxisPos[1] = 10.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 3.0;
ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 4
AxisPos[0] = 5.0;
AxisPos[1] = 10.0;
```

```
    AxisVel[0] = 0.0;
    AxisVel[1] = 0.0;
    Time = 4.5;
    ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
        return;
    }
    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```


7.8.6. ECAT_McGroupMoveLineRel_PVT

Description:

Start group relative PVT motion.

Syntax:

int32_t ECAT_McGroupMoveLineRel_PVT(uint16_t DeviceNo, uint16_t GroupNo, double Pos[], double Vel[], double Time)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Position data arrary of a group Each array element is the relative displacement of an axis. (Unit: user unit)
Vel	double[]	IN	Velocity data arrary of a group Each array element is the veloctiy data of an axis. (Unit: user unit/s)
Time	double	IN	Time (Unit: second)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double AxisPos[MC_AXIS_NO_MAX];
double AxisVel[MC_AXIS_NO_MAX];
double Time;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo,GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
```



```
AxisPos[0] = 0.0;
AxisPos[1] = 0.0;
AxisVel[0] = 0.0;
AxisVel[1] = 0.0;
Time = 0;
ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n",ret);
    return;
}
//Command 2
AxisPos[0] = 5.0;
AxisPos[1] = 0.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 1.5;
ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 3
AxisPos[0] = -5.0;
AxisPos[1] = 10.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 3.0;
ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 4
AxisPos[0] = 5.0;
AxisPos[1] = 0.0;
```



```
    AxisVel[0] = 0.0;
    AxisVel[1] = 0.0;
    Time = 4.5;
    ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
        return;
    }
    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```


7.8.7. ECAT_McGroupMoveCircularAbs_CP_Angle

Description:

Start an absolute 2D circular interpolation motion by providing the center position and its angle.

Syntax:

int32_t ECAT_McGroupMoveCircularAbs_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double AuxPos[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: user unit/s)
Angle	double	IN	Angle from start point to end point (Unit: degree)
AuxPos	double[]	IN	Absolute positon of the center point (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

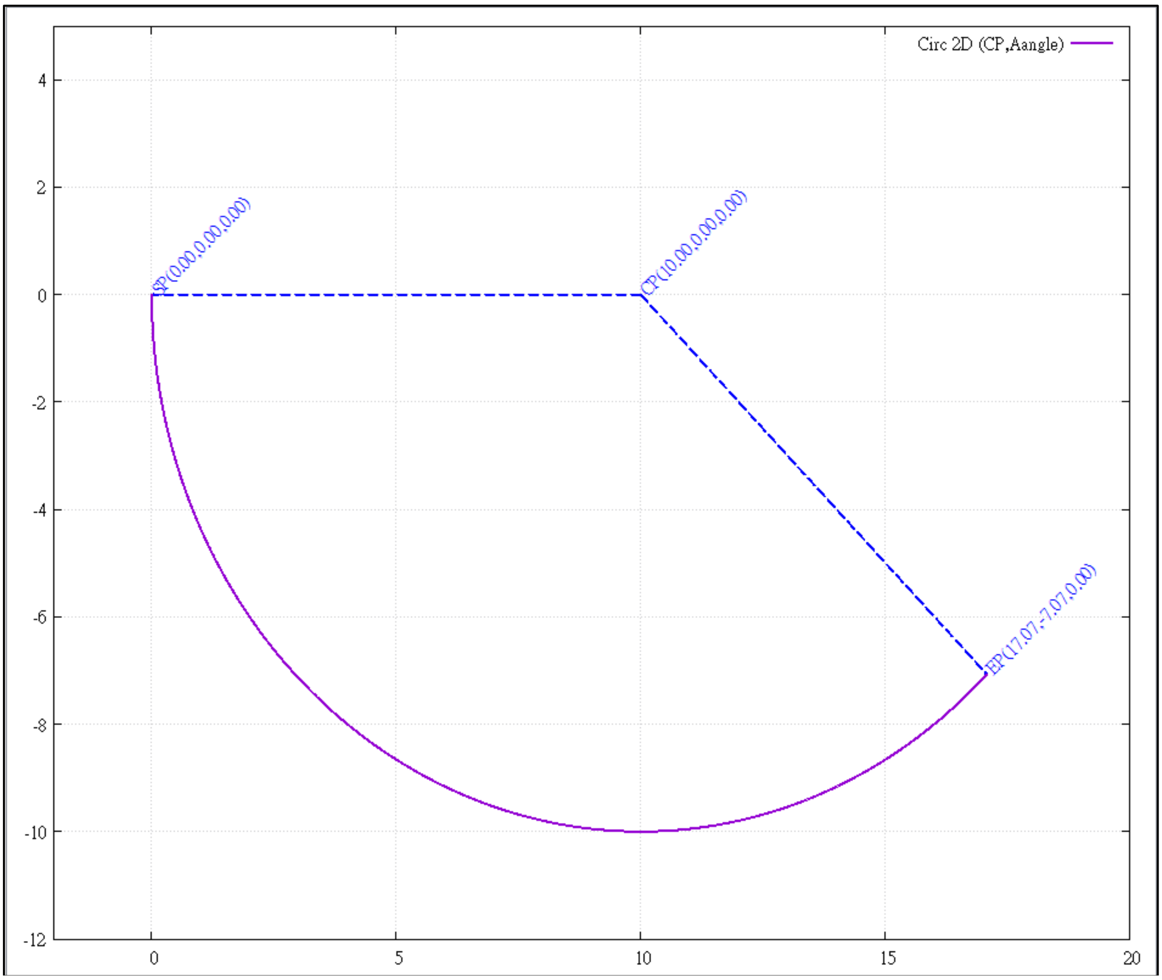
Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```

```
if(State == MC_GS_STANDBY) //Standby
{
        //Command 1
        GroupPos[0] = 0.0;
        GroupPos[1] = 0.0;
        GroupVel = 5;
        ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
        if(ret < 0)
                printf("Failed to add group move line command:%d\n", ret);

        //Command 2
        CircAuxPos [0] = 10.0; //Center Position
        CircAuxPos [1] = 0.0; //Center Position
        CircAngle = 135;
        GroupVel = 5;
        ret = ECAT_McGroupMoveCircularAbs_CP_Angle(DeviceNo, GroupNo, GroupVel, CircAngle
                                            , CircAuxPos);
        if(ret < 0)
        {
                printf("Failed to add group move circular command:%d\n", ret);
        }

        do
        {
                sleep(1);
                ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

        }while(State == MC_GS_MOVING) //Moving

        if(State == MC_GS_STANDBY) //Standby
                printf("Group move circular successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
                printf("Group error stop\n");
        }
}
```


2D circular interpolation motion path of example:



7.8.8. ECAT_McGroupMoveCircularRel_CP_Angle

Description:

Start a relative 2D circular interpolation motion by providing the center position and its angle.

Syntax:

int32_t ECAT_McGroupMoveCircularRel_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double AuxPos[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: $\text{user unit}/s$)
Angle	double	IN	Angle from start point to end point (Unit: degree)
AuxPos	double[]	IN	Relative distance from the center point to the start point (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```



```
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 10.0; //Center Position
    CircAuxPos [1] = 0.0; //Center Position
    CircAngle = 135;
    GroupVel = 5;
    ret = ECAT_McGroupMoveCircularRel_CP_Angle(DeviceNo, GroupNo, GroupVel, CircAngle
                                            , CircAuxPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```


7.8.9. ECAT_McGroupMoveCircularAbs_CP_EP

Description:

Start an absolute 2D circular interpolation motion by providing the center position and the end position.

Syntax:

int32_t ECAT_McGroupMoveCircularAbs_CP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: user unit/s)
Dir	uint8_t	IN	Direction 0: CW 1: CCW
AuxPos	double[]	IN	Absolute positon of center point (Unit: user unit)
EndPos	double[]	IN	Absolute positon of end point (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}
```

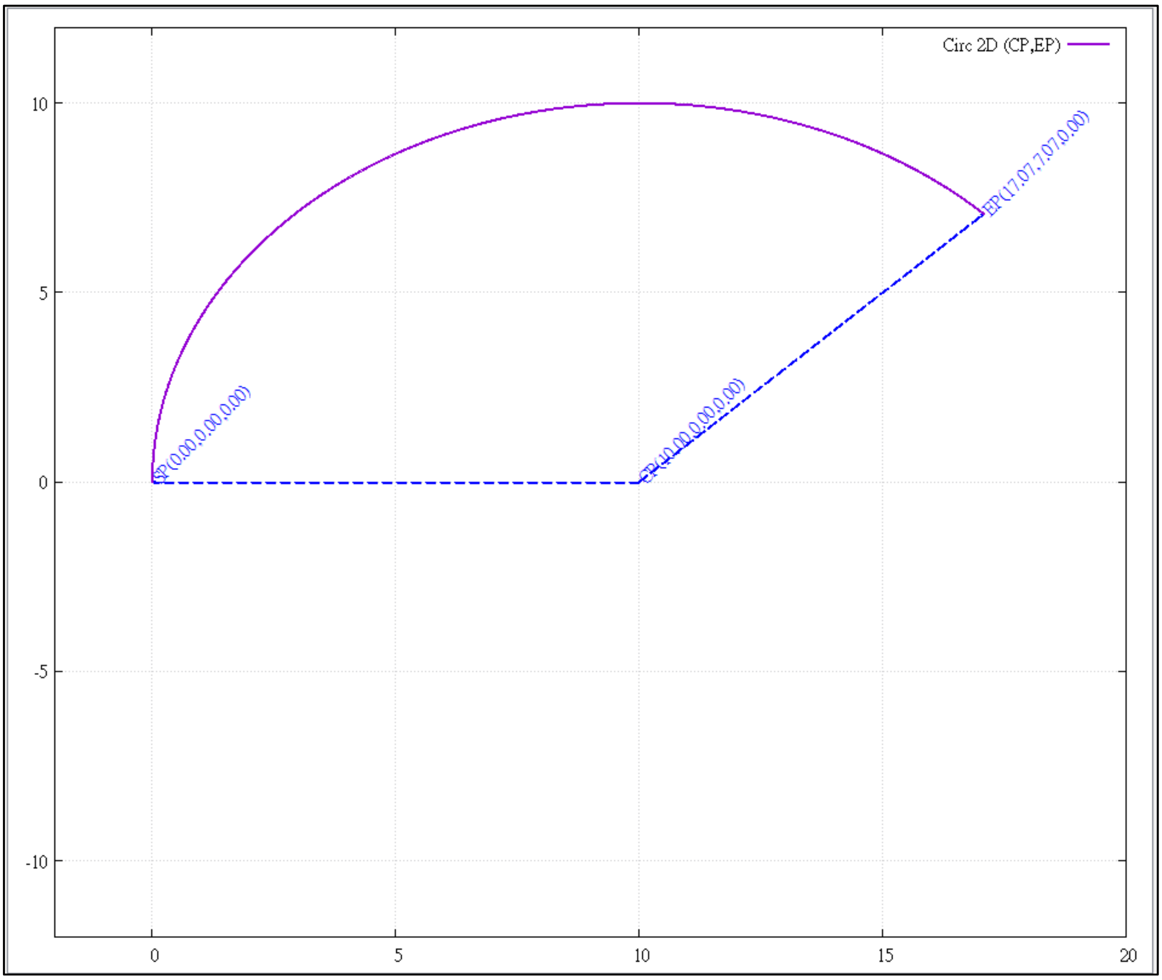


```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);

    //Command 2
    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircEndPos[0] = 17.071 // End Position
    CircEndPos[1] = 7.071 // End Position

    GroupVel = 5;
    CircDir = 0; //CW
    ret = ECAT_McGroupMoveCircularAbs_CP_EP(DeviceNo, GroupNo, GroupVel, CircDir
            , CircAuxPos, CircEndPos);
    if(ret < 0)
        printf("Failed to add group move circular command:%d\n", ret);


    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
```



```
    }
}
```

2D circular interpolation motion path of example:



7.8.10. ECAT_McGroupMoveCircularRel_CP_EP

Description:

Start a relative 2D circular interpolation motion by providing the center position and the end position.

Syntax:

int32_t ECAT_McGroupMoveCircularRel_CP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: user unit/s)
Dir	uint8_t	IN	Direction 0: CW 1: CCW
AuxPos	double[]	IN	Relative distance from the center point to the start point (Unit: user unit)
EndPos	double[]	IN	Relative distance from the end point to the start point (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n",ret);
	return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n",ret);
	return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
	printf("Failed to set group command mode:%d\n", ret);
	return;
}
```



```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{

    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircEndPos[0] = 17.071 // End Position
    CircEndPos[1] = 7.071 // End Position

    GroupVel = 5;
    CircDir = 0; //CW
    ret = ECAT_McGroupMoveCircularRel_CP_EP(DeviceNo, GroupNo, GroupVel, CircDir
            , CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

7.8.11. ECAT_McGroupMoveCircularAbs_BP_EP

Description:

Start an absolute 2D circular interpolation motion by providing a border position and the end position.

Syntax:

int32_t ECAT_McGroupMoveCircularAbs_BP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: $\text{user unit}/s$)
Dir	uint8_t	IN	Direction 0: CW 1: CCW
AuxPos	double[]	IN	Absolute positon of the border point (Unit: user unit)
EndPos	double[]	IN	Absolute positon of the end point (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
	printf("Failed to set group command mode:%d\n", ret);
	return;
}
```

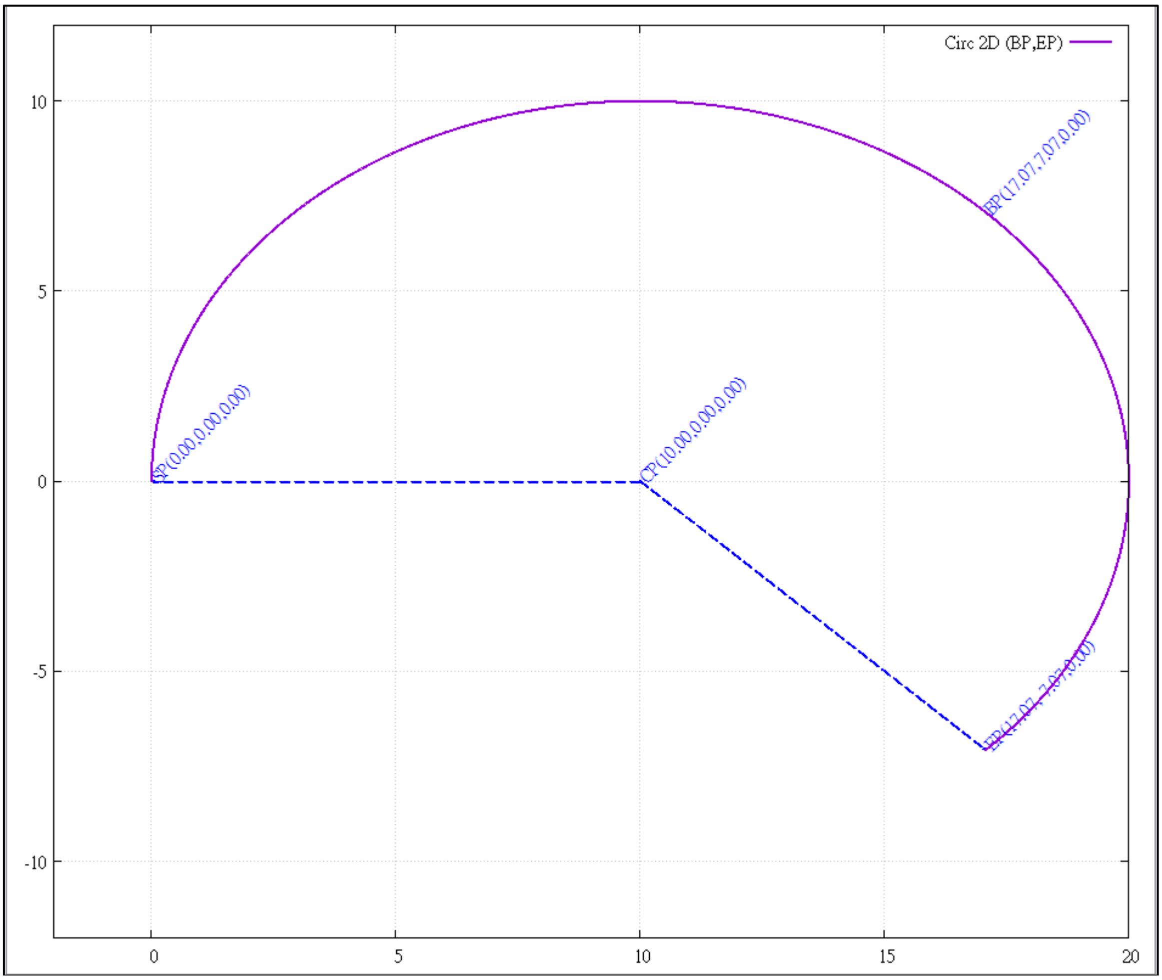


```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
	//Command 1
	GroupPos[0] = 0.0;
	GroupPos[1] = 0.0;
	GroupVel = 5;
	ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
	if(ret < 0)
		printf("Failed to add group move line command:%d\n", ret);

	//Command 2
	CircAuxPos[0] = 17.071; //Border Position
	CircAuxPos[1] = 7.071; //Border Position
	CircEndPos[0] = 17.071 // End Position
	CircEndPos[1] = -7.071 // End Position

	GroupVel = 5;
	CircDir = 0; //CW
	ret = ECAT_McGroupMoveCircularAbs_BP_EP(DeviceNo, GroupNo, GroupVel, CircDir
			, CircAuxPos, CircEndPos);
	if(ret < 0)
	{
		printf("Failed to add group move circular command:%d\n", ret);
	}

	do
	{
		sleep(1);
		ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

	}while(State == MC_GS_MOVING) //Moving

	if(State == MC_GS_STANDBY) //Standby
		printf("Group move circular successfully!\n");
	else if(State == MC_GS_ERRORSTOP) //ErrorStop
	{
```



```
            printf("Group error stop\n");
        }
    }
```

2D circular interpolation motion path of example:



7.8.12. ECAT_McGroupMoveCircularRel_BP_EP

Description:

Start a relative 2D circular interpolation motion by providing a border position and the end position.

Syntax:

int32_t ECAT_McGroupMoveCircularRel_BP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: $\mathrm{user\ unit}/s$)
Dir	uint8_t	IN	Direction 0: CW 1: CCW
AuxPos	double[]	IN	Relative distance from the border point to the start point (Unit: user unit)
EndPos	double[]	IN	Relative distance from the end point to the start point (Unit:user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}
```



```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos[0] = 17.071; //Border Position
    CircAuxPos[1] = 7.071; //Border Position
    CircEndPos[0] = 17.071 // End Position
    CircEndPos[1] = -7.071 // End Position

    GroupVel = 5;
    CircDir = 0; //CW
    ret = ECAT_McGroupMoveCircularRel_BP_EP(DeviceNo, GroupNo, GroupVel, CircDir
              , CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```


7.8.13. ECAT_McGroupMove3DCircularAbs_CP_Angle

Description:

Start an absolute 3D circular interpolation motion by providing the center position and an angle.

Syntax:

int32_t ECAT_McGroupMove3DCircularAbs_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double AuxPos[], double NV[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Tangent velocity of the motion (Unit: $\text{user unit}/s$)
Angle	double	IN	Angle between the end point and the start point (right-hand rule) (Unit: degree)
AuxPos	double[]	IN	Absolute positon of center point (Unit: user unit)
NV	double[]	IN	Normal vector of the circle

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}
```

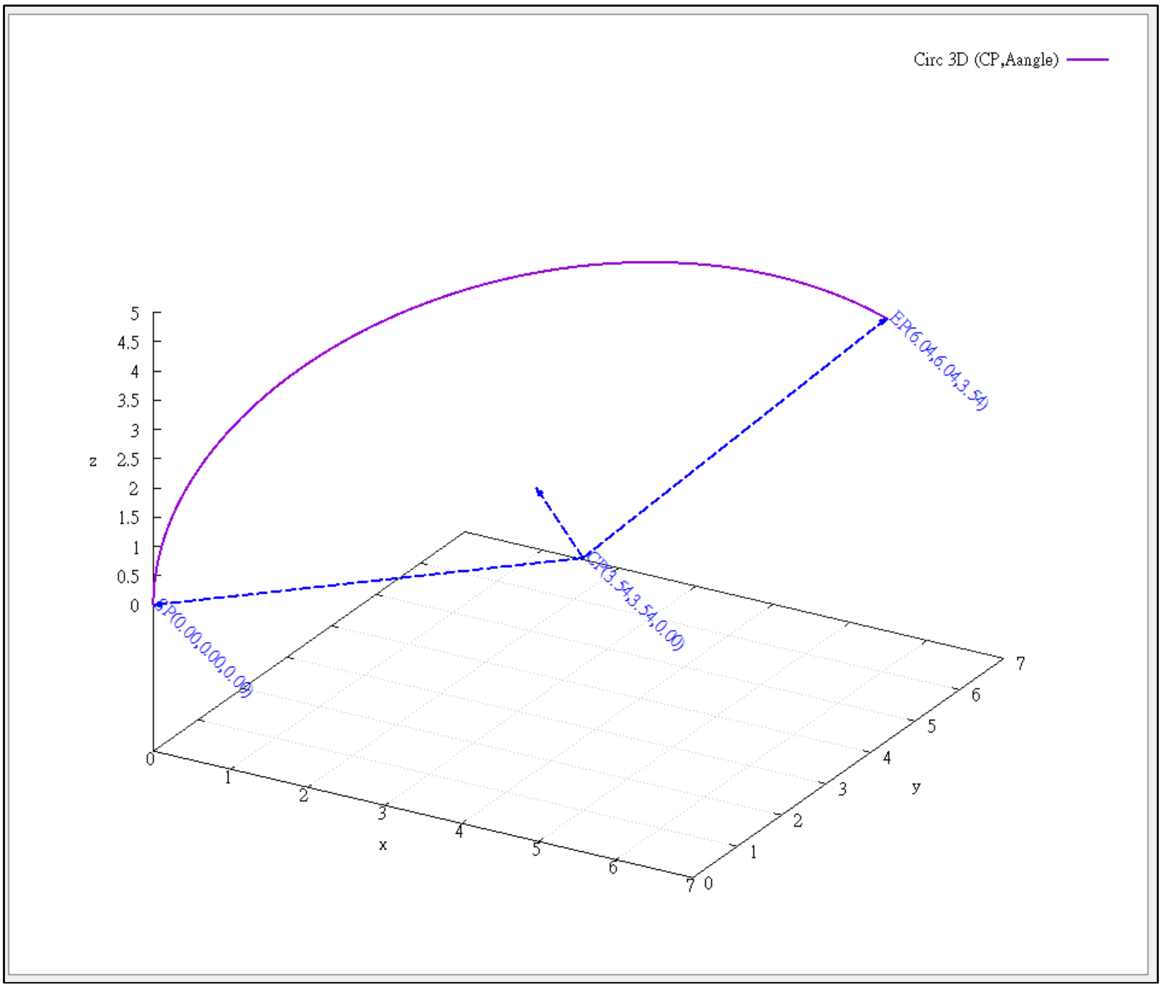


```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupPos[2] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);

    //Command 2
    CircAuxPos [0] = 3.5355;    //Center Position
    CircAuxPos [1] = 3.5355;    //Center Position
    CircAuxPos [2] = 0.0;           //Center Position

    NV [0] = -0.7071;        //Normal Vector
    NV [1] = 0.7071;    //Normal Vector
    NV [2] = 0.0;           //Normal Vector

    CircAngle = 135;
    GroupVel = 5;
    ret = ECAT_McGroupMove3DCircularAbs_CP_Angle(DeviceNo, GroupNo, GroupVel
            , CircAngle, CircAuxPos, NV);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
```



```
            printf("Group move circular successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
    }
```

3D circular interpolation motion path of example:



7.8.14. ECAT_McGroupMove3DCircularRel_CP_Angle

Description:

Start a relative 3D circular interpolation motion by providing the center position and an angle.

Syntax:

int32_t ECAT_McGroupMove3DCircularRel_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double AuxPos[], double NV[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: $\text{user unit}/s$)
Angle	double	IN	Angle between the end point and the start point (right-hand rule) (Unit: degree)
AuxPos	double[]	IN	Relative distance from the center point to the start point (Unit: user unit)
NV	double[]	IN	Normal vector of the circle

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}
```



```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
	CircAuxPos [0] = 3.5355;	//Center Position
	CircAuxPos [1] = 3.5355;	//Center Position
	CircAuxPos [2] = 0.0;		//Center Position

	NV [0] = -0.7071;	//Normal Vector
	NV [1] = 0.7071;	//Normal Vector
	NV [2] = 0.0;		//Normal Vector

	CircAngle = 135;
	GroupVel = 5;
	ret = ECAT_McGroupMove3DCircularRel_CP_Angle(DeviceNo, GroupNo, GroupVel
			, CircAngle, CircAuxPos, NV);
	if(ret < 0)
	{
		printf("Failed to add group move circular command:%d\n", ret);
	}

	do
	{
		sleep(1);
		ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

	}while(State == MC_GS_MOVING) //Moving

	if(State == MC_GS_STANDBY) //Standby
		printf("Group move circular successfully!\n");
	else if(State == MC_GS_ERRORSTOP) //ErrorStop
	{
		printf("Group error stop\n");
	}
}
```


7.8.15. ECAT_McGroupMove3DCircularAbs_CP_EP

Description:

Start an absolute 3D circular interpolation motion by providing the center position and the end position.

Syntax:

int32_t ECAT_McGroupMove3DCircularAbs_CP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: $\mathrm{user\ unit/s}$)
Dir	uint8_t	IN	Direction 0: CW 1: CCW
AuxPos	double[]	IN	Absolute positon of the center point (Unit: user unit)
EndPos	double[]	IN	Absolute positon of the end point (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];


AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
```

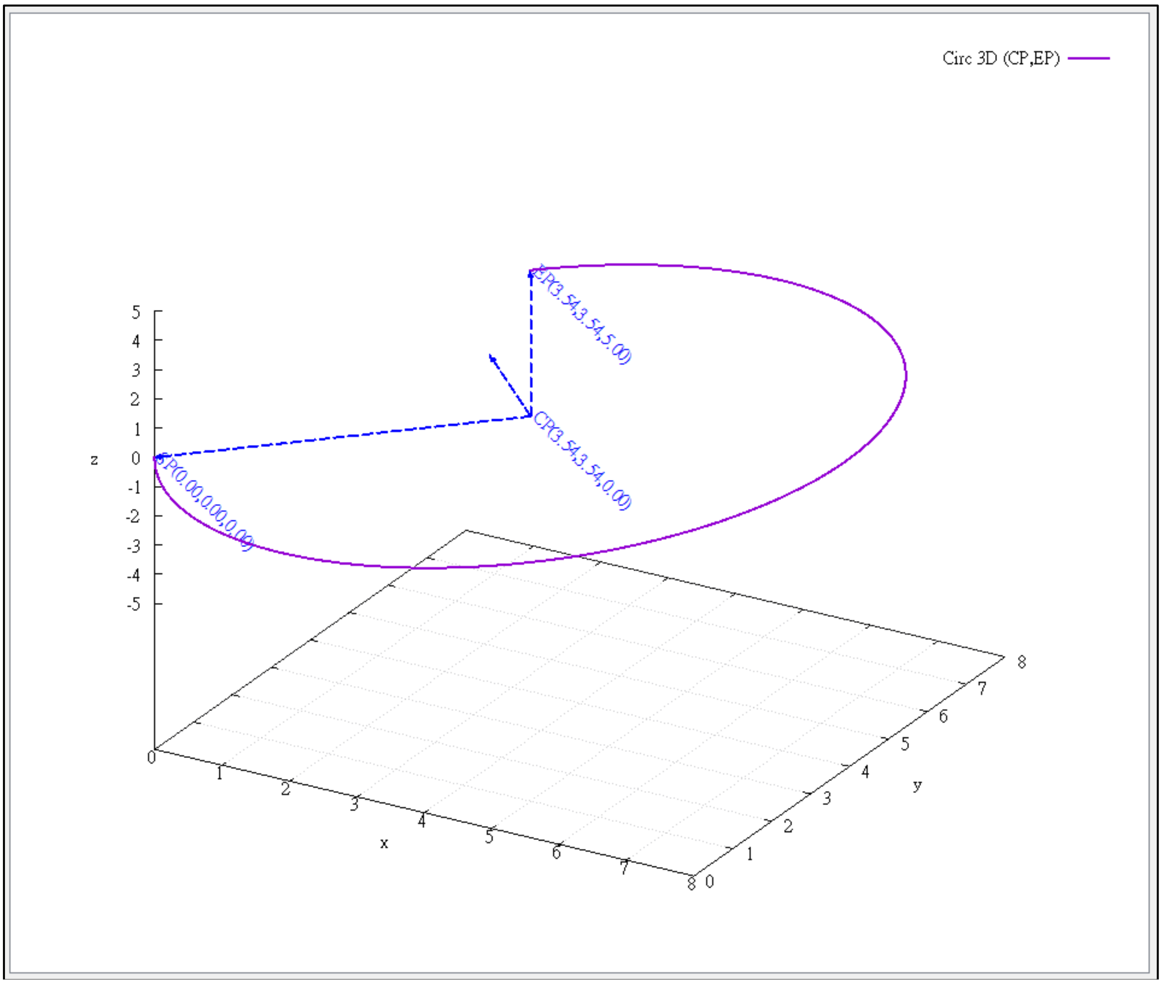


```
        return;
    }

    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    if(State == MC_GS_STANDBY) //Standby
    {
        //Command 1
        GroupPos[0] = 0.0;
        GroupPos[1] = 0.0;
        GroupPos[2] = 0.0;
        GroupVel = 5;
        ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
        if(ret < 0)
            printf("Failed to add group move line command:%d\n", ret);

        //Command 2
        CircAuxPos [0] = 3.5355;    //Center Position
        CircAuxPos [1] = 3.5355;    //Center Position
        CircAuxPos [2] = 0.0;           //Center Position

        CircEndPos[0] = 3.5355       // End Position
        CircEndPos[1] = 3.5355       // End Position
        CircEndPos[2] = 5.0        // End Position
        CircDir = 0; // CW
        GroupVel = 5;
        ret = ECAT_McGroupMove3DCircularAbs_CP_EP(EcatDeviceID, GroupNo, GroupVel
                , CircDir, CircAuxPos, CircEndPos);
        if(ret < 0)
        {
            printf("Failed to add group move circular command:%d\n", ret);
        }

        do
        {
            sleep(1);
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

        }while(State == MC_GS_MOVING) //Moving
```



```
    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

3D circular interpolation motion path of example:



7.8.16. ECAT_McGroupMove3DCircularRel_CP_EP

Description:

Start a relative 3D circular interpolation motion by providing the center position and the end position.

Syntax:

int32_t ECAT_McGroupMove3DCircularRel_CP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: user unit/s)
Dir	uint8_t	IN	Direction 0: CW 1: CCW
AuxPos	double[]	IN	Relative distance from the center point to the start point (Unit: user unit)
EndPos	double[]	IN	Relative distance from the end point to the start point (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];


AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
```



```
        return;
    }

    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    if(State == MC_GS_STANDBY) //Standby
    {
        CircAuxPos [0] = 3.5355;    //Center Position
        CircAuxPos [1] = 3.5355;    //Center Position
        CircAuxPos [2] = 0.0;           //Center Position

        CircEndPos[0] = 3.5355      // End Position
        CircEndPos[1] = 3.5355      // End Position
        CircEndPos[2] = 5.0         // End Position
        CircDir = 0; // CW
        GroupVel = 5;
        ret = ECAT_McGroupMove3DCircularRel_CP_EP(EcatDeviceID, GroupNo, GroupVel
                , CircDir, CircAuxPos, CircEndPos);
        if(ret < 0)
        {
            printf("Failed to add group move circular command:%d\n", ret);
        }

        do
        {
            sleep(1);
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

        }while(State == MC_GS_MOVING) //Moving

        if(State == MC_GS_STANDBY) //Standby
            printf("Group move circular successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
    }
```


7.8.17. ECAT_McGroupMove3DCircularAbs_BP_EP

Description:

Start an absolute 3D circular interpolation motion by providing a border position and the end position.

Syntax:

int32_t ECAT_McGroupMove3DCircularAbs_BP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: user unit/s)
Dir	uint8_t	IN	Direction 0: CW 1: CCW
AuxPos	double[]	IN	Absolute positon of the border point (Unit: user unit)
EndPos	double[]	IN	Absolute positon of the end point (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];


AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
```

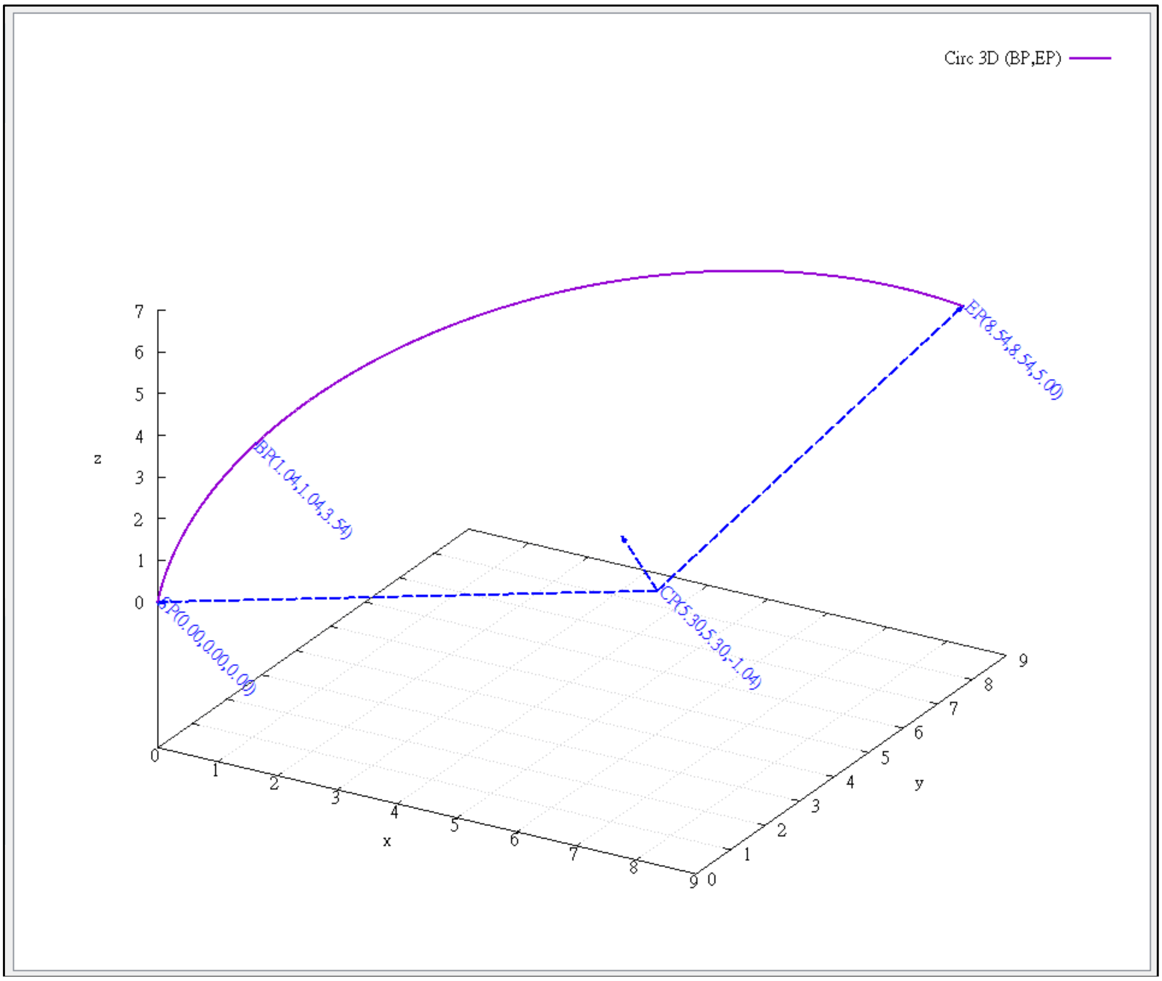


```
        return;
    }

    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    if(State == MC_GS_STANDBY) //Standby
    {
        //Command 1
        GroupPos[0] = 0.0;
        GroupPos[1] = 0.0;
        GroupPos[2] = 0.0;
        GroupVel = 5;
        ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
        if(ret < 0)
            printf("Failed to add group move line command:%d\n", ret);

        //Command 2
        CircAuxPos [0] = 1.036;    //Border Position
        CircAuxPos [1] = 1.036;    //Border Position
        CircAuxPos [2] = 3.5355;   //Border Position

        CircEndPos[0] = 8.53656      // End Position
        CircEndPos[1] = 8.53656      // End Position
        CircEndPos[2] = 5.0        // End Position
        CircDir = 1; // CCW
        GroupVel = 5;
        ret = ECAT_McGroupMove3DCircularAbs_BP_EP(EcatDeviceID, GroupNo, GroupVel
                  , CircDir, CircAuxPos, CircEndPos);
        if(ret < 0)
        {
            printf("Failed to add group move circular command:%d\n", ret);
        }

        do
        {
            sleep(1);
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

        }while(State == MC_GS_MOVING) //Moving
```



```
    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

3D circular interpolation motion path of example:



7.8.18. ECAT_McGroupMove3DCircularRel_BP_EP

Description:

Start a relative 3D circular interpolation motion by providing a border position and the end position.

Syntax:

int32_t ECAT_McGroupMove3DCircularRel_BP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: user unit/s)
Dir	uint8_t	IN	Direction 0: CW 1: CCW
AuxPos	double[]	IN	Relative distance from the border point to the start point (Unit: user unit)
EndPos	double[]	IN	Relative distance form the end point to the start point (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];


AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
	printf("Failed to set group command mode:%d\n", ret);
```



```
        return;
    }

    ret = ECAT_McGetGroupState(DeviceNo, GroupNo,& State);
    if(State == MC_GS_STANDBY) //Standby
    {
        CircAuxPos [0] = 1.036;     //Border Position
        CircAuxPos [1] = 1.036;     //Border Position
        CircAuxPos [2] = 3.5355;    //Border Position

        CircEndPos[0] = 8.53656       // End Position
        CircEndPos[1] = 8.53656       // End Position
        CircEndPos[2] = 5.0         // End Position
        CircDir = 1; // CCW
        GroupVel = 5;
        ret = ECAT_McGroupMove3DCircularRel_BP_EP(EcatDeviceID, GroupNo, GroupVel
                , CircDir, CircAuxPos, CircEndPos);
        if(ret < 0)
        {
            printf("Failed to add group move circular command:%d\n", ret);
        }

        do
        {
            sleep(1);
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

        }while(State == MC_GS_MOVING) //Moving

        if(State == MC_GS_STANDBY) //Standby
            printf("Group move circular successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
    }
```


7.8.19. ECAT_McGroupMoveHelicalAbs

Description:

Start a helical interpolation motion.

Syntax:

int32_t ECAT_McGroupMoveHelicalAbs(uint16_t DeviceNo,uint16_t GroupNo, double Angle, double AuxPos[], double Pitch, double Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Angle	double	IN	Angle of rotation 360 indicates one full revolution; and 720 will produce two full revolutions (Unit: degree)
AuxPos	double[]	IN	Absolute positon of the center point (Unit: user unit)
Pitch	double	IN	Pitch (Unit: user unit)
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

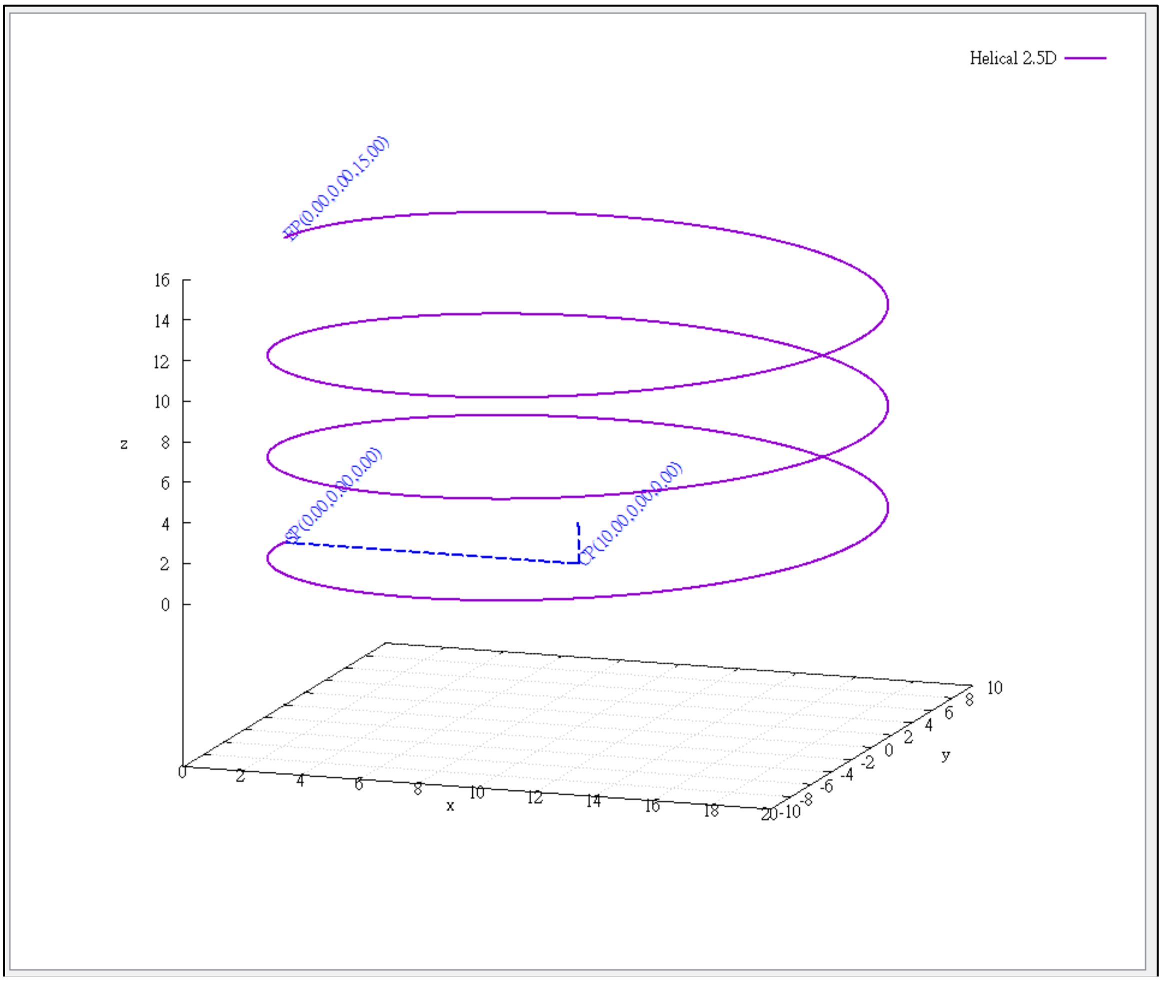
[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```



```
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);

    //Command 2
    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    ret = ECAT_McGroupMoveHelicalAbs(DeviceNo, GroupNo
            , CircAngle, CircAuxPos, HelicalPitch, GroupVel);
    if(ret < 0)
        printf("Failed to add group move helical command:%d\n", ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

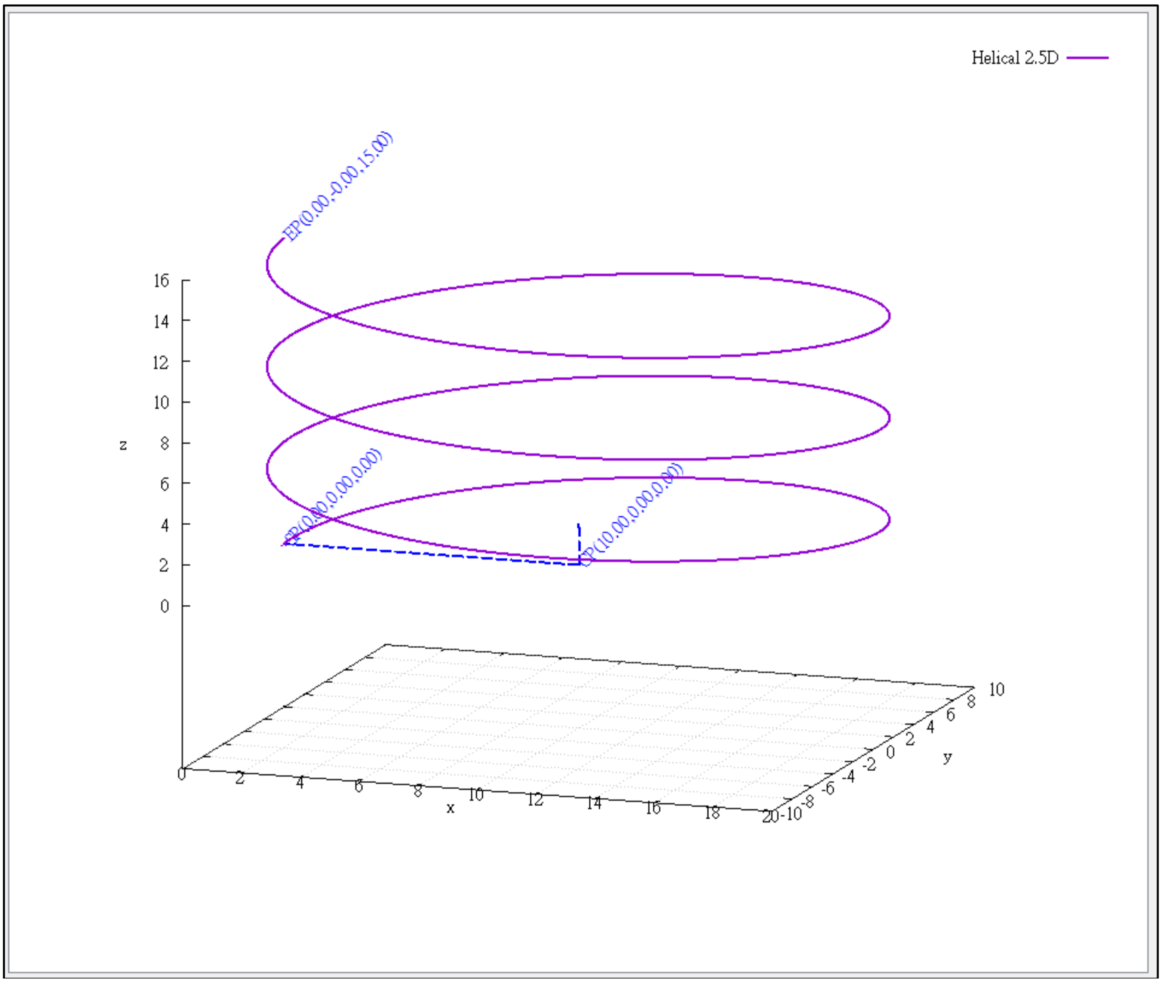

helical interpolation motion path of example (right-handed):



helical interpolation motion path of example (left-handed):

If the rotation angle parameter is set to negative value, the helical motion path is left-handed.

CircAngle = -1080;



7.8.20. ECAT_McGroupMoveHelicalRel

Description:

Start a relative helical interpolation motion.

Syntax:

int32_t ECAT_McGroupMoveHelicalRel(uint16_t DeviceNo,uint16_t GroupNo, double Angle, double AuxPos[], double Pitch, double Vel)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Angle	double	IN	Angle of rotation 360 indicates one full revolution and 720 will produce two full revolutions (Unit: degree)
AuxPos	double[]	IN	Axis relative distance from the center point to the start point (Unit: user unit)
Pitch	double	IN	Pitch (Unit: user unit)
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t Time_ms = 500;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAngle;
double HelicalPitch;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupAccTime(DeviceNo, GroupNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set group acceleration time:%d\n", ret);
    return;
}
```



```
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
	printf("Failed to set group command mode:%d\n", ret);
	return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
	//Command 1
	GroupPos[0] = 0.0; //Center Position
	GroupPos[1] = 20.0; //Center Position
	CircAngle = 1080;
	HelicalPitch = 5;
	GroupVel = 5;
	ret = ECAT_McGroupMoveHelicalRel (EcatDeviceID, GroupNo,
			CircAngle, GroupPos, HelicalPitch, GroupVel);
	if(ret < 0)
	{
		printf("Failed to add group move helical command:%d\n", ret);
	}

	do
	{
		sleep(1);
		ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

	}while(State == MC_GS_MOVING) //Moving

	if(State == MC_GS_STANDBY) //Standby
		printf("Group move helical successfully!\n");
	else if(State == MC_GS_ERRORSTOP) //ErrorStop
	{
		printf("Group error stop\n");
	}
}
```


7.8.21. ECAT_McGroupMove3DHelicalAbs_CP_Angle

Description:

Start an absolute 3D helical interpolation motion.

Syntax:

int32_t ECAT_McGroupMove3DHelicalAbs_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	DeviceNo	Device number (Card ID)
GroupNo	uint16_t	GroupNo	Group number
Vel	double	Vel	Velocity (Unit: $\text{user unit}/s$)
Angle	double	Angle	Angle of rotation 360 indicates one full revolution and 720 will produce two full revolutions (Unit: degree)
Pitch	double	Pitch	Pitch (Unit: user unit)
AuxPos	double[]	AuxPos	Absolute positon of the center point of the base circle (Unit:user unit)
NV	double[]	IN	Normal vector of the base circle

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo,GroupNo,AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo,GroupNo,AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo,GroupNo,CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}
```

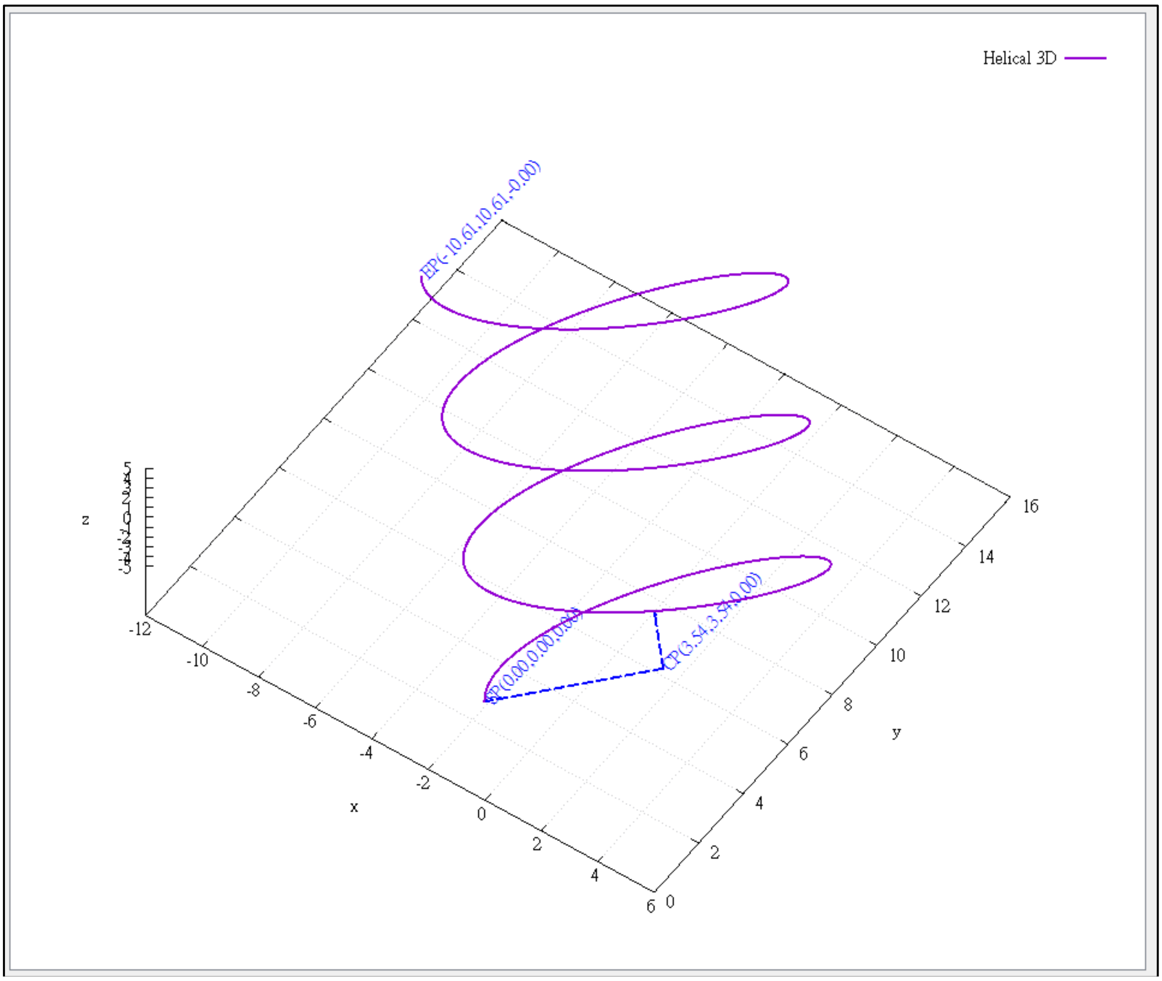


```
ret = ECAT_McGetGroupState(DeviceNo,GroupNo,&State);
if(State == MC_GS_STANDBY) //Standby
{
	//Command 1
	GroupPos[0] = 0.0;
	GroupPos[1] = 0.0;
	GroupVel = 5;
	ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
	if(ret < 0)
		printf("Failed to add group move line command:%d\n",ret);

	//Command 2
	CircAuxPos [0] = 3.5355;	//Center Position
	CircAuxPos [1] = 3.5355;	//Center Position
	CircAuxPos [2] = 0.0;		//Center Position

	NV [0] = -0.7071;	//Normal Vector
	NV [1] = 0.7071;	//Normal Vector
	NV [2] = 0.0;		//Normal Vector

	CircAngle = 1080;
	HelicalPitch = 5;
	GroupVel = 5;
	ret = ECAT_McGroupMove3DHelicalAbs_CP_Angle(DeviceNo, GroupNo, GroupVel
			, CircAngle, HelicalPitch, CircAuxPos, NV);
	if(ret < 0)
		printf("Failed to add group move helical command:%d\n",ret);

	do
	{
		sleep(1);
		ret = ECAT_McGetGroupState(DeviceNo,GroupNo,&State);

	}while(State == MC_GS_MOVING) //Moving

	if(State == MC_GS_STANDBY) //Standby
		printf("Group move circular successfully!\n");
	else if(State == MC_GS_ERRORSTOP) //ErrorStop
```



```
    {
        printf("Group error stop\n");
    }
}
```

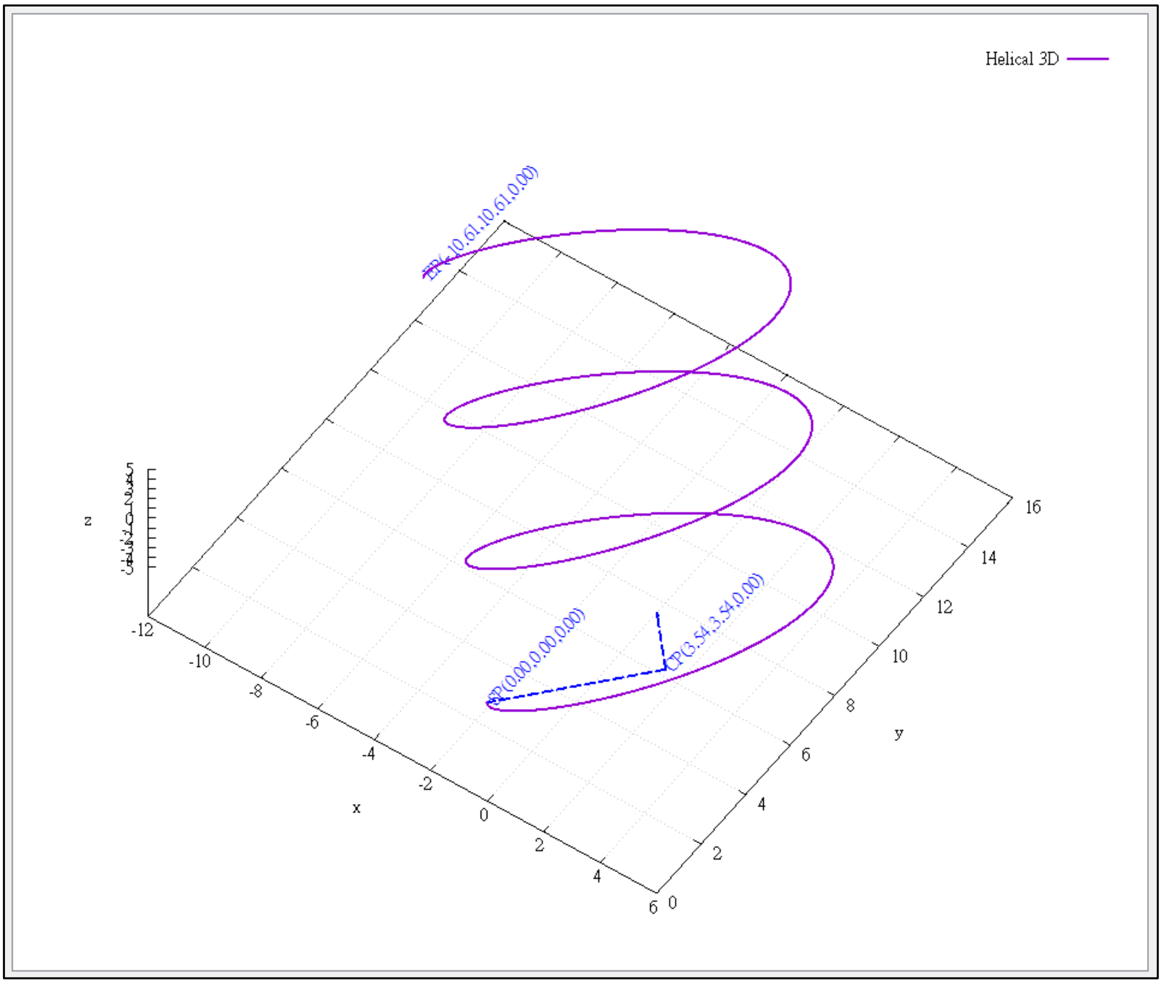
3D helical interpolation motion path of example (right-handed):



3D helical interpolation motion path of example (left-handed):

If the rotation angle parameter is set to negative value, the helical motion path is left-handed.

```
CircAngle = -1080;
```



7.8.22. ECAT_McGroupMove3DHelicalRel_CP_Angle

Description:

Start a relative 3D helical interpolation motion.

Syntax:

int32_t ECAT_McGroupMove3DHelicalRel_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	DeviceNo	Device number (Card ID)
GroupNo	uint16_t	GroupNo	Group number
Vel	double	Vel	Velocity (Unit: user unit/s)
Angle	double	Angle	Angle of rotation 360 indicates one full revolution and 720 will produce two full revolutions (Unit: degree)
Pitch	double	Pitch	Pitch (Unit: user unit)
AuxPos	double[]	AuxPos	Relative distance from the center point of the base circle to its start point (Unit:user unit)
NV	double[]	IN	Normal vector of the base circle

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}
```



```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
	CircAuxPos [0] = 3.5355;	//Center Position
	CircAuxPos [1] = 3.5355;	//Center Position
	CircAuxPos [2] = 0.0;		//Center Position

	NV [0] = -0.7071;	//Normal Vector
	NV [1] = 0.7071;	//Normal Vector
	NV [2] = 0.0;		//Normal Vector

	CircAngle = 1080;
	HelicalPitch = 5;
	GroupVel = 5;
	ret = ECAT_McGroupMove3DHelicalRel_CP_Angle(DeviceNo, GroupNo, GroupVel
			, CircAngle, HelicalPitch, CircAuxPos, NV);
	if(ret < 0)
		printf("Failed to add group move helical command:%d\n", ret);

	do
	{
		sleep(1);
		ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

	}while(State == MC_GS_MOVING) //Moving

	if(State == MC_GS_STANDBY) //Standby
		printf("Group move circular successfully!\n");
	else if(State == MC_GS_ERRORSTOP) //ErrorStop
	{
		printf("Group error stop\n");
	}
}
```

7.8.23. ECAT_McGroupMoveConicalHelixAbs

Description:

Start an absolute conical helix interpolation motion.

Syntax:

int32_t ECAT_McGroupMoveConicalHelixAbs(uint16_t DeviceNo, uint16_t GroupNo, double Angle, double AuxPos[], double Pitch, double Vel, double EndRadius)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Angle	double	IN	Angle of rotation 360 indicates one full revolution and 720 will produce two full revolutions (Unit: degree)
AuxPos	double[]	IN	Absolute positon of the center point "Start Radius" is the distance between the center point and the start point (Unit: user unit)
Pitch	double	IN	Pitch (Unit: user unit)
Vel	double	IN	Velocity (Unit: user unit/s)
EndRadius	double	IN	End Radius (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

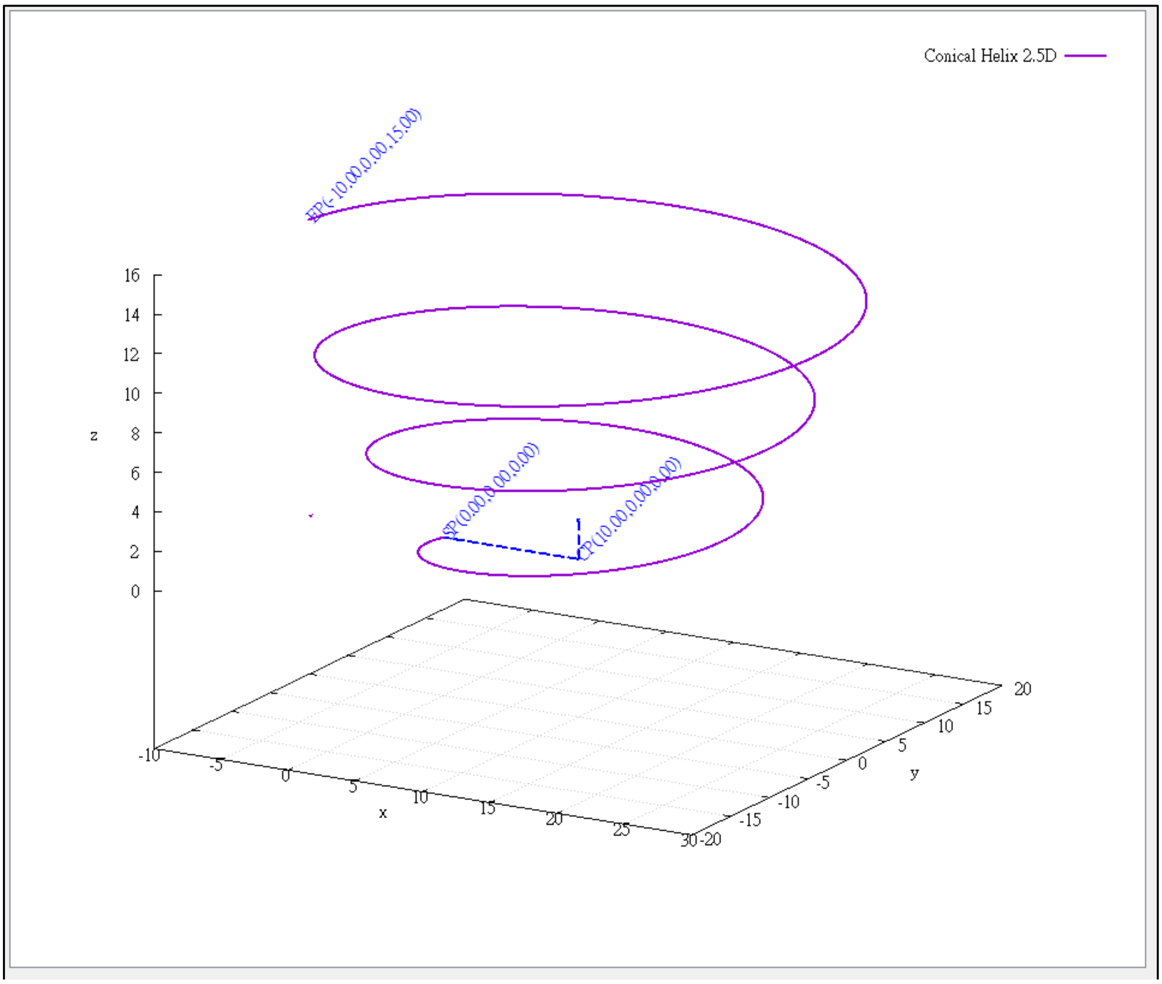
[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
double EndRadius;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
```

```
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);

    //Command 1
    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 20;
    ret = ECAT_McGroupMoveConicalHelixAbs(DeviceNo, GroupNo
            , CircAngle, CircAuxPos, HelicalPitch, GroupVel, EndRadius);
    if(ret < 0)
        printf("Failed to add group move conical helix command:%d\n", ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
```

```
}
```

conical helical interpolation motion path of example:



7.8.24. ECAT_McGroupMoveConicalHelixRel

Description:

Start a relative conical helix interpolation motion.

Syntax:

int32_t ECAT_McGroupMoveConicalHelixRel(uint16_t DeviceNo, uint16_t GroupNo, double Angle, double AuxPos[], double Pitch, double Vel, double EndRadius)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Angle	double	IN	Angle of rotation 360 indicates one full revolution and 720 will produce two full revolutions (Unit: degree)
AuxPos	double[]	IN	Relative positon of the center point "Start Radius" is the distance between the center point and the start point (Unit: user unit)
Pitch	double	IN	Pitch (Unit: user unit)
Vel	double	IN	Velocity (Unit: user unit/s)
EndRadius	double	IN	End Radius (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:
[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
double EndRadius;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
	printf("Failed to set group command mode:%d\n", ret);
	return;
}
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```



```
if(State == MC_GS_STANDBY) //Standby
{
	CircAuxPos[0] = 10.0; //Center Position
	CircAuxPos[1] = 0.0; //Center Position
	CircAngle = 1080;
	HelicalPitch = 5;
	GroupVel = 5;
	EndRadius = 20;
	ret = ECAT_McGroupMoveConicalHelixRel(DeviceNo, GroupNo
			, CircAngle, CircAuxPos, HelicalPitch, GroupVel, EndRadius);
	if(ret < 0)
		printf("Failed to add group move conical helix command:%d\n", ret);

	do
	{
		sleep(1);
		ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

	}while(State == MC_GS_MOVING) //Moving

	if(State == MC_GS_STANDBY) //Standby
		printf("Group move circular successfully!\n");
	else if(State == MC_GS_ERRORSTOP) //ErrorStop
	{
		printf("Group error stop\n");
	}
}
```


7.8.25. ECAT_McGroupMove3DConicalHelixAbs_CP_Angle

Description:

Start an absolute 3D conical helix interpolation motion.

Syntax:

int32_t ECAT_McGroupMove3DConicalHelixAbs_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[], double EndRadius)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: $\text{user unit}/s$)
Angle	double	IN	Angle of rotation 360 indicates one full revolution and 720 will result in two full revolutions (Unit: degree)
Pitch	double	IN	Pitch (Unit: user unit)
AuxPos	double[]	IN	Absolute positon of the center point (Unit:user unit)
NV	double[]	IN	Normal vector
EndRadius	double	IN	End Radius (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

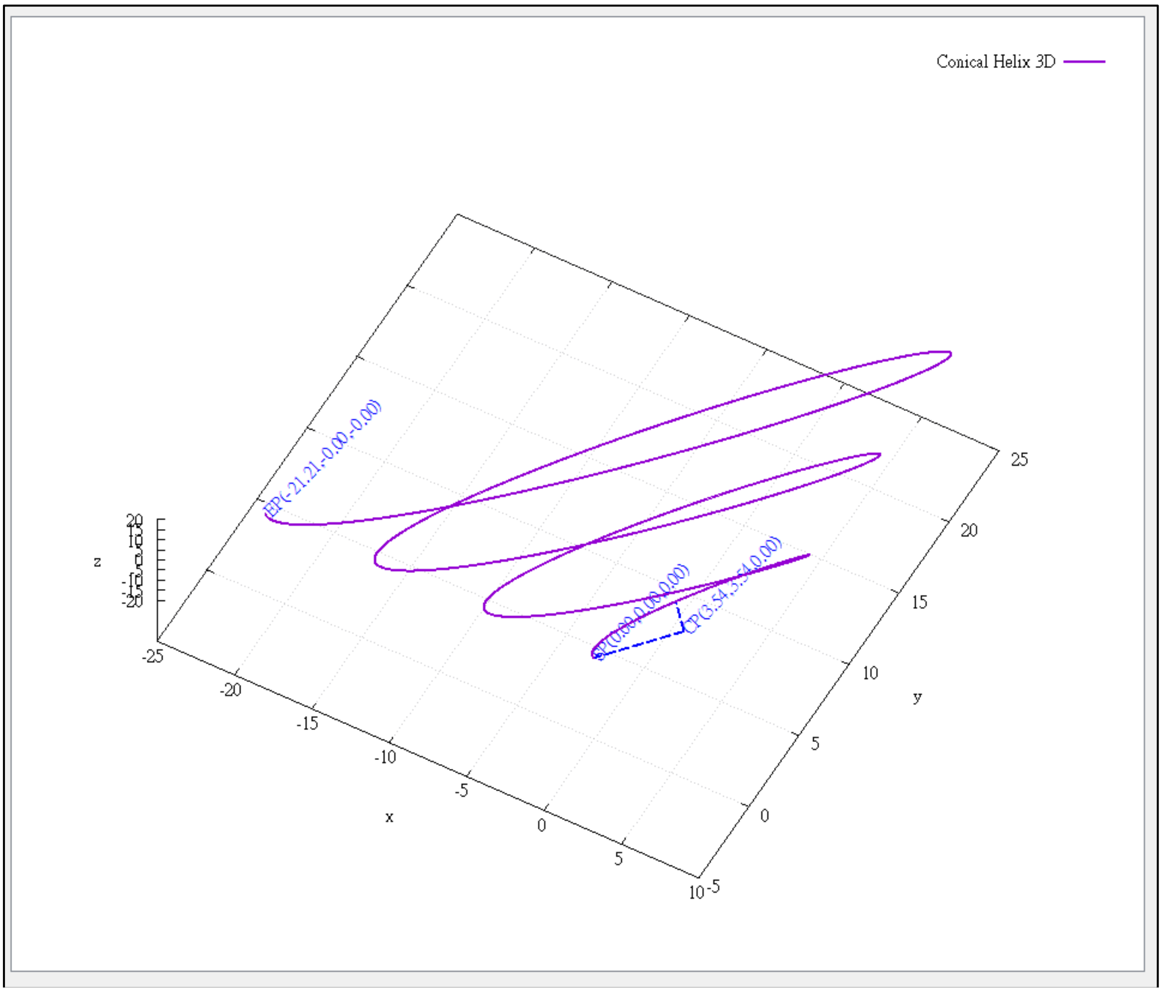
```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
double EndRadius;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}
```



```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);

    //Command 2
    CircAuxPos [0] = 3.5355;    //Center Position
    CircAuxPos [1] = 3.5355;    //Center Position
    CircAuxPos [2] = 0.0;           //Center Position

    NV [0] = -0.7071;        //Normal Vector
    NV [1] = 0.7071;    //Normal Vector
    NV [2] = 0.0;           //Normal Vector

    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 20;
    ret = ECAT_McGroupMove3DConicalHelixAbs _CP_Angle(DeviceNo, GroupNo, GroupVel
                , CircAngle, HelicalPitch, CircAuxPos, NV, EndRadius);
    if(ret < 0)
        printf("Failed to add group move conical helix command:%d\n", ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
```

```
            printf("Group move circular successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
}
```

conical helical interpolation motion path of example:



7.8.26. ECAT_McGroupMove3DConicalHelixRel_CP_Angle

Description:

Start a relative 3D conical helix interpolation motion.

Syntax:

int32_t ECAT_McGroupMove3DConicalHelixRel_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[], double EndRadius)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: $\text{user unit}/s$)
Angle	double	IN	Angle of rotation 360 indicates one full revolution and 720 will result in two full revolutions. (Unit: degree)
Pitch	double	IN	Pitch (Unit: user unit)
AuxPos	double[]	IN	Relative position of center point (Unit: user unit)
NV	double[]	IN	Normal vector
EndRadius	double	IN	End Radius (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
double EndRadius;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}
```



```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 3.5355;   //Center Position
    CircAuxPos [1] = 3.5355;   //Center Position
    CircAuxPos [2] = 0.0;          //Center Position

    NV [0] = -0.7071;        //Normal Vector
    NV [1] = 0.7071;    //Normal Vector
    NV [2] = 0.0;          //Normal Vector

    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 20;
    ret = ECAT_McGroupMove3DConicalHelixRel_CP_Angle(DeviceNo, GroupNo, GroupVel
            , CircAngle, HelicalPitch, CircAuxPos, NV, EndRadius);
    if(ret < 0)
        printf("Failed to add group move conical helix command:%d\n", ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```


7.8.27. ECAT_McGroupMoveProfile

Description:

Start a profile position motion.

Syntax:

int32_t ECAT_McGroupMoveProfile(uint16_t DeviceNo, uint16_t GroupNo, uint16_t ProfileNo[], uint16_t TotalStep)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
ProfileNo	uint16_t[]	IN	An array contains serval profile buffer numbers. Each element in this array is an profile buffer number.
TotalStep	uint16_t	IN	Total moving steps

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t ProfileNo[MC_AXIS_NO_MAX];
uint16_t TotalStep = 1000;

...
AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
	ProfileNo[0] = 0;
	ProfileNo[1] = 1;
	ret = ECAT_McGroupMoveProfile(DeviceNo, GroupNo, ProfileNo, TotalStep);
	if(ret < 0)
	{
		printf("Failed to start group move profile:%d\n", ret);
	}
```



```
        do
        {
            sleep(1);
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

        }while(State == MC_GS_MOVING) //Moving

        if(State == MC_GS_STANDBY) //Standby
            printf("Group move profile successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
    }
```


7.8.28. ECAT_McGroupMoveProfileCSV

Description:

Start a profile position motion. The profile data are read from a CSV file. The file format is shown in Figure 7.1.

Syntax:

int32_t ECAT_McGroupMoveProfileCSV(uint16_t DeviceNo, uint16_t GroupNo, uint16_t ProfileNo[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
ProfileNo	uint16_t	IN	File number of Profile data This file contains profile data for all axes in the group.

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t ProfileNo = 0;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n", ret);
	return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
	ret = ECAT_McGroupMoveProfileCSV(DeviceNo, GroupNo, ProfileNo);
	if(ret < 0)
	{
		printf("Failed to start group move profile CSV:%d\n", ret);
	}

	do
	{
		sleep(1);
		ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```

```
    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move profile successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```


7.8.29. ECAT_McGroupMoveDwell

Description:

The motion will wait for the dwell time; then continue to execute the next command. This command can be used for adjusting the blending distance for two motion commands in continuous blending motion. This command behaves just like any other motion commands and is sequentially executed. In Buffered or Blending mode, if a motion command is being executed, it will be pushed into the command buffer. In Aborting mode, it will stop the current command; and then the system starts to wait for the dwell time.

Syntax:

int32_t ECAT_McGroupMoveDwell(uint16_t DeviceNo, uint16_t GroupNo, uint32_t Cnt)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Cnt	Uint32_t	IN	Dwell time Unit: EtherCAT cycle time

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;
uint32_t DwellTime;


AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```



```
if(State == MC_GS_STANDBY) //Standby
{
	//Command 1
	GroupPos[0] = 10.0;
	GroupPos[1] = 20.0;
	GroupVel = 5;
	ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
	if(ret < 0)
	{
		printf("Failed to add group move line command:%d\n", ret);
	}
	//Command 2
	GroupPos[0] = 30.0;
	GroupPos[1] = 50.0;
	DwellTime = 1000; //Wait 1s, If cycletime = 1ms
	ret = ECAT_McGroupMoveDwell(DeviceNo, GroupNo, DwellTime);
	if(ret < 0)
	{
		printf("Failed to add group move line command:%d\n", ret);
	}

	do
	{
		sleep(1);
		ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

	}while(State == MC_GS_MOVING) //Moving

	if(State == MC_GS_STANDBY) //Standby
		printf("Group move line successfully!\n");
	else if(State == MC_GS_ERRORSTOP) //ErrorStop
	{
		printf("Group error stop\n");
	}
}
```

7.8.30. ECAT_McGroupMoveDO

Description:

Add the specified slave DO output command in the group motion.

Syntax:

int32_t ECAT_McGroupMoveDO(uint16_t DeviceNo,uint16_t GroupNo, uint16_t SlaveNo, uint16_t BitNo, uint32_t Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
SlaveNo	uint16_t	IN	Slave number
BitNo	uint16_t	IN	Bit number
Value	uint32_t	IN	Bit data (0 or 1)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint16_t SlaveNo, BitNo, Value;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n",ret);
	return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
	printf("Failed to add axis to group:%d\n",ret);
	return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
	printf("Failed to set group command mode:%d\n",ret);
	return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
```

```
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }
    //Command 2
    SlaveNo = 3;
    BitNo = 1;
    Value = 1;
    DwellTime = 1000; //Wait 1s, If cycletime = 1ms
    ret = ECAT_McGroupMoveDO(DeviceNo, GroupNo, SlaveNo, BitNo, Value);
    if(ret < 0)
    {
        printf("Failed to add group move DO command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```


7.8.31. ECAT_McGroupMoveAO

Description:

Add the specified slave AO output command in the group motion.

Note:RunMode as 0 : Set the binary value of an analog output channel.

RunMode as 1: Set the voltage output value of a specified analog output channel.(Use ECAT_SetSlaveAoProperty to configure ao slave)。

Syntax:

int32_t ECAT_McGroupMoveAO(uint16_t DeviceNo, uint16_t GroupNo, uint16_t SlaveNo, uint32_t RunMode, uint16_t ChannelNo, uint16_t RawData, double VoltData)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
SlaveNo	uint16_t	IN	Slave number
RunMode	uint32_t	IN	RunMode
ChannelNo	uint16_t	IN	Channel number
RawData	uint16_t	IN	AO binary value
VoltData	double	IN	AO voltage value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint16_t SlaveNo, ChannelNo, RawData;
double VoltData = 0;
uint32_t RunMode = 0;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}
```



```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }
    //Command 2
    SlaveNo = 3;
    ChannelNo = 0;
    RawData = 32767;
    DwellTime = 1000; //Wait 1s, If cycletime = 1ms
    ret = ECAT_McGroupMoveAO(DeviceNo, GroupNo, SlaveNo, RunMode, ChannelNo, RawData,
    VoltData);
    if(ret < 0)
    {
        printf("Failed to add group move AO command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```


7.8.32. ECAT_McGroupMoveBlendingSync

Description:

When the group motion command is the blending mode, adding this command will wait for the deceleration stop to complete before executing the next motion command.

Syntax:

int32_t ECAT_McGroupMoveBlendingSync(uint16_t DeviceNo, uint16_t GroupNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BLENDING; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;
uint32_t DwellTime;


AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n",ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```

```
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }
    //Command 2
    ret = ECAT_McGroupMoveBlendingSync(DeviceNo, GroupNo);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }

    //Command 3
    GroupPos[0] = 30.0;
    GroupPos[1] = 40.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
```

```
    {
        printf("Group error stop\n");
    }
}
```

7.8.33. ECAT_McGroupStop

Description:

Stop motion of a group with deceleration.

Syntax:

int32_t ECAT_McGroupStop(uint16_t DeviceNo, uint16_t GroupNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
nt32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint32_t State;
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

...
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_DISABLED) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }
    //Command 2
    GroupPos[0] = 30.0;
    GroupPos[1] = 50.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }

    sleep(1000);
    ret = ECAT_McGroupStop(DeviceNo, GroupNo);
    if(ret < 0)
    {
        printf("Failed to stop group move:%d\n", ret);
```



```
            return;
        }
        else
        {
            do
            {
                sleep(1);
                ret = ECAT_McGetGroupState(DeviceNo,GroupNo, &State);
            }while(State == MC_GS_STOPPING) //Stopping

            if(State == MC_GS_STANDBY) //Standby
                printf("Group move stop successfully!\n");
            else if(State == MC_GS_ERRORSTOP) //ErrorStop
            {
                printf("Group error stop\n");
            }
        }
    }
```

7.8.34. ECAT_McGroupQuickStop

Description:

Immediately stop the motion of a group.

Syntax:

int32_t ECAT_McGroupQuickStop(uint16_t DeviceNo, uint16_t GroupNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
nt32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint32_t State;
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

...
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_DISABLED) //Standby
{
	//Command 1
	GroupPos[0] = 10.0;
	GroupPos[1] = 20.0;
	GroupVel = 5;
	ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
	if(ret < 0)
	{
		printf("Failed to add group move line command:%d\n", ret);
	}
	//Command 2
	GroupPos[0] = 30.0;
	GroupPos[1] = 50.0;
	GroupVel = 10;
	ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
	if(ret < 0)
	{
		printf("Failed to add group move line command:%d\n", ret);
	}

	sleep(1000);
	ret = ECAT_McGroupQuickStop(DeviceNo, GroupNo);
	if(ret < 0)
	{
		printf("Failed to stop group move:%d\n", ret);
```



```
            return;
        }
        else
        {
            do
            {
                sleep(1);
                ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
            }while(State == MC_GS_STOPPING) //Stopping

            if(State == MC_GS_STANDBY) //Standby
                printf("Group move stop successfully!\n");
            else if(State == MC_GS_ERRORSTOP) //ErrorStop
            {
                printf("Group error stop\n");
            }
        }
}
```


7.8.35. ECAT_McSetGroupHold

Description:

Group state become **MC_GS_HOLD**, hold group motion after current command is done, Hold until disable hold state.

Notice:PVT motion command and another group motion command can not be used together.

Syntax:

int32_t ECAT_McSetGroupHold(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Status)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
GroupNo	uint16_t	IN	Group number
Status	uint16_t	IN	0:disable hold state 1:hold

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

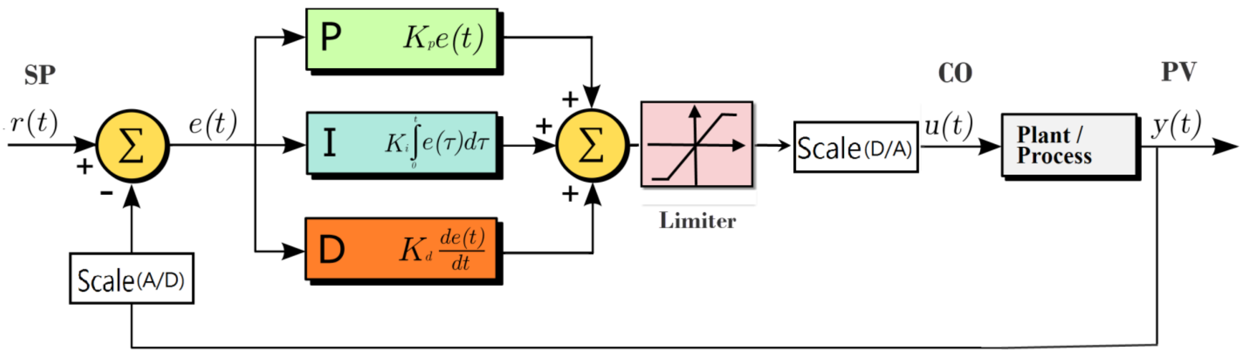
Example:

[C/C++]

```
nt32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint32_t State;
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
	ret = ECAT_McSetGroupHold(DeviceNo, GroupNo, 1 ); // hold
	if(ret < 0)
	{
		printf("Failed to set group hold:%d\n",ret);
	}

	//Command 1
	GroupPos[0] = 10.0;
	GroupPos[1] = 20.0;
	GroupVel = 5;
	ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
	if(ret < 0)
	{
		printf("Failed to add group move line command:%d\n",ret);
	}
	//Command 2
	GroupPos[0] = 30.0;
	GroupPos[1] = 50.0;
	GroupVel = 10;
	ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
	if(ret < 0)
	{
		printf("Failed to add group move line command:%d\n",ret);
	}
	ret = ECAT_McSetGroupHold(DeviceNo, GroupNo, 0 );
```

```
    if(ret < 0)
    {
        printf("Failed to set group hold:%d\n",ret);
    }
    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

7.9. PID Controller



SP: SetPoint

CO: Controller Output

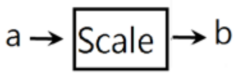
PV: Process Variable

e(t): SP-PV

Simulate Plant Model:

$$G(s) = \frac{1}{s+1}$$

Scale:



b = a* ScaleGain + ScaleOffset

7.9.1. ECAT_PidGetSetPointValue

Description:

Get the Set Point Value.

Syntax:

int32_t ECAT_PidGetSetPointValue(uint16_t DeviceNo, uint32_t PidNo, double* SetPointValue)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
SetPointValue	double*	OUT	Set Point Value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n",ret);

ret = ECAT_ PidSetStatus (DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);

ret = ECAT_PidGetSetPointValue(DeviceNo, PidNo, &SetPointValue);
if(ret != 0)
{
	printf("Failed to get Pid Set Point Value:%d\n",ret);
}
else
{
	printf("Pid Set Point Value %d\n", SetPointValue);
}
```


7.9.2. ECAT_PidSetSetPointValue

Description:

Set the Set Point Value.

Syntax:

int32_t ECAT_PidSetSetPointValue(uint16_t DeviceNo, uint32_t PidNo, double SetPointValue)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
SetPointValue	double	IN	Set Point Value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0)
{
	printf("Failed to set Pid Set Point Value:%d\n", ret);
}
```


7.9.3. ECAT_PidGetProcessVariable

Description:

Get the Process Variable.

Syntax:

int32_t ECAT_PidGetProcessVariable(uint16_t DeviceNo, uint32_t PidNo, double* ProcessVariable)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
ProcessVariable	double*	OUT	Process Variable (or Process Value)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n", ret);

ret = ECAT_PidGetProcessVariable(DeviceNo, PidNo, &SetPointValue);
if(ret != 0)
{
	printf("Failed to get Pid Set Point Value:%d\n", ret);
}
else
{
	printf("Pid Set Point Value %d\n", SetPointValue);
}
```


7.9.4. ECAT_PidGetSampleTime

Description:

Get the sampling time.

Syntax:

int32_t ECAT_PidGetSampleTime(uint16_t DeviceNo, uint32_t PidNo, uint32_t* Interval)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
PidNo	uint32_t	IN	PID Controller number
Interval	int32_t*	Output	Sampling time Unit: EtherCAT Cycle Time

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
int32_t Interval = 0;

ret = ECAT_PidGetSampleTime(DeviceNo, PidNo, &Interval);
if(ret != 0)
{
    printf("Failed to Get Pid Controller:%d\n", ret);
}
else
{
    printf("Pid Interval %d\n", Interval);
}
```


7.9.5. ECAT_PidSetSampleTime

Description:

Set the sampling time.

Syntax:

int32_t ECAT_PidSetSampleTime(uint16_t DeviceNo, uint32_t PidNo, uint32_t Interval)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
Interval	int32_t	IN	Sampling time Unit: EtherCAT Cycle Time

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
int32_t Interval = 1;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0)
{
    printf("Failed to set Pid Controller:%d\n", ret);
}

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n",ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n",ret);
```


7.9.6. ECAT_PidGetStatus

Description:

Get the controller status. It can be enabled or disabled.

Syntax:

int32_t ECAT_PidGetStatus(uint16_t DeviceNo, uint32_t PidNo, uint8_t *status)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
status	uint8_t*	Output	Status 0: disabled 1: enabled

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint8_t Status= 0;

ret = ECAT_PidGetStatus(DeviceNo, PidNo, &Status);
if(ret != 0)
{
	printf("Failed to Get Pid Status:%d\n", ret);
}
else
{
	printf("Pid Status Value %d\n", Status);
}
```

7.9.7. ECAT_PidSetStatus

Description:

Set PID Controller Status.

Note: Changing the status from **Enabled** to **Disabled** will not clear the output of the control output module. Users can set control oputut to whatever they like by using function *ECAT_SetSlaveRxPdoData* if PID Controller Status is disabled. However, if the status is changed from **Disabled** to **Enabled, it** will set the output of the control oputut module to 0; then the controller start to work.

Syntax:

int32_t ECAT_PidSetStatus(uint16_t DeviceNo, uint32_t PidNo, uint8_t status)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
status	uint8_t	IN	Status 0: disabled 1: enabled

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)
{
	printf("Failed to Set Pid Status:%d\n", ret);
}
```


7.9.8. ECAT_PidGetSimulateMode

Description:

Get simulation status. Use it to know whether the system is set for simulaiton or not.

Syntax:

int32_t ECAT_PidGetSimulateMode(uint16_t DeviceNo, uint32_t PidNo, uint8_t *status)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
status	uint8_t*	Output	Status 0: disabled 1: enabled

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint8_t Status= 0;

ret = ECAT_PidGetSimulateMode(DeviceNo, PidNo, &Status);
if(ret != 0)
{
	printf("Failed to Get Pid Simulate:%d\n", ret);
}
else
{
	printf("Pid Simulate %d\n", Status);
}
```


7.9.9. ECAT_PidSetSimulateMode

Description:

Set simulation status. Use it to set whether the system is set for simulaiton or not.

Note: Changing the status from **Disable** to En**able simulation** will clear the output of the control output module which is used for this PID controller. Users can set control oputut value by using function *ECAT_SetSlaveRxPdoData* if simulation is disabled.

Syntax:

int32_t ECAT_PidSetSimulateMode(uint16_t DeviceNo, uint32_t PidNo, uint8_t status)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
status	uint8_t	IN	Status 0: Disable simulation 1: Enable simulation

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)
{
	printf("Failed to Set Pid Simulate:%d\n", ret);
}

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n", ret);
```


7.9.10. ECAT_PidGetParameter

Description:

Get the control parameters of a PID Controller.

Syntax:

int32_t ECAT_PidGetParameter(uint16_t DeviceNo, uint32_t PidNo, double *kp, double *ki, double *kd)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number (0 ~ 9)
kp	double *	Output	Proporational control gain
ki	double *	Output	Integral control gain
kd	double *	Output	Derivative control gain

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double kp= 0;
double ki= 0;
double kd= 0;
ret = ECAT_PidGetParameter(DeviceNo, PidNo, &kp, &ki, &kd)
if(ret != 0)
{
    printf("Failed to Get Pid Parameter:%d\n", ret);
}
else
{
    printf("Pid Parameter : kp:%f , ki:%f , kd:%f \n", kp, ki, kd);
}
```


7.9.11. ECAT_PidSetParameter

Description:

Set the control parameters of a PID Controller.

Syntax:

int32_t ECAT_PidSetParameter(uint16_t DeviceNo, uint32_t PidNo, double kp, double ki, double kd)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number (0 ~ 9)
kp	double	IN	Proporational control gain
ki	double	IN	Integral control gain
kd	double	IN	Derivative control gain

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0) printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)
{
	printf("Failed to Set Pid Parameter:%d\n", ret);
}

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);
```


7.9.12. ECAT_PidGetProcessVariableModule

Description:

A Process Variable in a PID control loop is measured by an analog input channel in a module. In order to convert a analog input value to a physical value, users need to set scaling parameters. SlaveNo, OffsetByte and Bitlength are used for assignment of the module and its analog input channel. ScaleGain and ScaleOffset are used for data converion. This function can get these settings.

Syntax:

int32_t ECAT_PidGetProcessVariableModule(uint16_t DeviceNo, uint32_t PidNo, uint16_t* SlaveNo, uint16_t* OffsetByte, uint16_t* Bitlength, double* ScaleGain, double* ScaleOffset)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
SlaveNo	uint16*	Output	Slave number
OffsetByte	uint16*	Output	Byte offset
Bitlength	uint16*	Output	Data Size, Unit: bit
ScaleGain	double*	Output	Input Gain
ScaleOffset	double*	Output	Input Offset

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint16_t SlaveNo = 0;
uint16_t Offset = 0;
uint16_t Bitsize = 16;
double Scalegain = 1;
double Scaleoffset = 0;

ret=ECAT_PidGetProcessVariableModule(DeviceNo, PidNo, &SlaveNo, &Offset, &Bitsize, &Scalegain,
&Scaleoffset);
if(ret != 0)
{
    printf("Failed to Get Pid Input:%d\n", ret);
}
```

7.9.13. ECAT_PidSetProcessVariableModule

Description:

A Process Variable in a PID control loop is measured by an analog input channel in a module. In order to convert a analog input value to a physical value, users need to set scaling parameters. **SlaveNo**, **OffsetByte** and **Bitlength** are used for assignment of the module and its analog input channel. **ScaleGain** and **ScaleOffset** are used for data converion. This function can set these settings.

Syntax:

int32_t ECAT_PidSetProcessVariableModule(uint16_t DeviceNo, uint32_t PidNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t Bitlength,double ScaleGain,double ScaleOffset)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
SlaveNo	uint16	IN	Slave number
OffsetByte	uint16	IN	Byte offset
Bitlength	uint16	IN	Data Size, Unit: bit
ScaleGain	double	IN	Input Gain
ScaleOffset	double	IN	Input Offset

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint16_t SlaveNo_input = 0;
uint16_t Offset_input = 0;
uint16_t Bitsize_input = 16;
double Scalegain_input = 1;
double Scaleoffset_input = 0;
uint16_t SlaveNo_output = 0;
uint16_t Offset_output = 2;
uint16_t Bitsize_output = 16;
double Scalegain_output = 1;
double Scaleoffset_output = 0;
double Max_Value = 0;
double Min_Value = 0;
int32_t Input = 5;
uint8_t Simulate= 0;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;

ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n", ret);

ret=ECAT_PidSetProcessVariableModule(DeviceNo, PidNo, SlaveNo_input, Offset_input, Bitsize_input,
Scalegain_input, Scaleoffset_input);
if(ret != 0)
```

```
{
	printf("Failed to Set Pid Input:%d\n", ret);
}

ret= ECAT_PidSetControlOutputModule(DeviceNo, PidNo, SlaveNo_output, Offset_output, Bitsize_output,
Scalegain_output, Scaleoffset_output, Max_Value, Min_Value);
if(ret != 0) printf("Failed to Set Pid Output:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n", ret);
```


7.9.14. ECAT_PidGetControlOutputModule

Description:

A Control Output in a PID control loop is sent to an analog output channel in a AO module. In order to convert a physical value to an analog output value, users need to set scaling parameters. **SlaveNo**, **OffsetByte** and **Bitlength** are used for assignment of the module and its analog output channel. **ScaleGain** and **ScaleOffset** are used for data converion. **Output_Max_Value** and **Output_Min_Value** are used to limit the control output value. This function can get these settings.

Syntax:

int32_t ECAT_PidGetControlOutputModule(uint16_t DeviceNo, uint32_t PidNo, uint16_t* SlaveNo, uint16_t* OffsetByte, uint16_t* Bitlength, double* ScaleGain, double* ScaleOffset, double* Output_Max_Value, double* Output_Min_Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
SlaveNo	uint16*	Output	Slave number
OffsetByte	uint16*	Output	Byte offset
Bitlength	uint16*	Output	Data Size, Unit: bit
ScaleGain	double *	Output	Output Gain
ScaleOffset	double *	Output	Output Offset
Output_Max_Value	double *	Output	Output Maximum Value
Output_Min_Value	double *	Output	Output Maximum Value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint16_t SlaveNo = 0;
uint16_t Offset = 0;
uint16_t Bitsize = 16;
double Scalegain = 1;
double Scaleoffset = 0;
double Max_Value = 0;
double Min_Value = 0;

ret=ECAT_PidGetControlOutputModule(DeviceNo, PidNo, &SlaveNo, &Offset, &Bitsize, &Scalegain,
&Scaleoffset, &Max_Value, &Min_Value);
if(ret != 0)
{
    printf("Failed to Get Pid Output:%d\n", ret);
}
```


7.9.15. ECAT_PidSetControlOutputModule

Description:

A Control Output in a PID control loop is sent to an analog output channel in a AO module. In order to convert a physical value to an analog output value, users need to set scaling parameters. **SlaveNo**, **OffsetByte** and **Bitlength** are used for assignment of the module and its analog output channel. **ScaleGain** and **ScaleOffset** are used for data converion. **Output_Max_Value** and **Output_Min_Value** are used to limit the control output value. This function can set these settings.

Syntax:

int32_t ECAT_PidSetControlOutputModule(uint16_t DeviceNo, uint32_t PidNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t Bitlength, double ScaleGain, double ScaleOffset, double Output_Max_Value, double Output_Min_Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
SlaveNo	uint16	IN	Slave number
OffsetByte	uint16	IN	Byte offset
Bitlength	uint16	IN	Data Size, Unit: bit
ScaleGain	double	IN	Output Gain
ScaleOffset	double	IN	Output Offset
Output_Max_Value	double	IN	Output Maximum Value
Output_Min_Value	double	IN	Output Maximum Value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint16_t SlaveNo_input = 0;
uint16_t Offset_input = 0;
uint16_t Bitsize_input = 16;
double Scalegain_input = 1;
double Scaleoffset_input = 0;
uint16_t SlaveNo_output = 0;
uint16_t Offset_output = 2;
uint16_t Bitsize_output = 16;
double Scalegain_output = 1;
double Scaleoffset_output = 0;
double Max_Value = 0;
double Min_Value = 0;
int32_t Input = 5;
uint8_t Simulate= 0;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;

ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n", ret);

ret=ECAT_PidSetProcessVariableModule(DeviceNo, PidNo, SlaveNo_input, Offset_input, Bitsize_input,
Scalegain_input, Scaleoffset_input);
if(ret != 0) printf("Failed to Set Pid Input:%d\n", ret);
```



```
ret=ECAT_PidSetControlOutputModule(DeviceNo, PidNo, SlaveNo_output, Offset_output, Bitsize_output,
Scalegain_output, Scaleoffset_output, Max_Value, Min_Value);
if(ret != 0)
{
    printf("Failed to Set Pid Output:%d\n", ret);
}
ret = ECAT_ PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);
```


7.9.16. ECAT_PidGetControlOutputValue

Description:

Get Control Output Value in a PID control loop.

Syntax:

int32_t ECAT_PidGetControlOutputValue(uint16_t DeviceNo, uint32_t PidNo, double* Output)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
Output	double*	Output	Control Output Value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Value= 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n",ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n",ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n",ret);

ret = ECAT_PidGetControlOutputValue(DeviceNo, PidNo, &Value);
if(ret != 0)
{
    printf("Failed to Get Pid Output Value:%d\n", ret);
}
else
{
    printf("Pid OutputValue :%f \n", Value);
}
```


7.9.17. ECAT_PidGetSimulateFeedback

Description:

If the simulation is enabled for a PID control loop, this function can get the Control Output Value of this loop.

Syntax:

int32_t ECAT_PidGetSimulateFeedback(uint16_t DeviceNo, uint32_t PidNo, double* Feedback)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
PidNo	uint32_t	IN	PID Controller number
Feedback	double*	Output	Control Output Value in a PID control loop with a simulation model as the process.

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Feedback = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidGetSimulateFeedback(DeviceNo, PidNo, &Feedback);
if(ret != 0)
{
    printf("Failed to Get Pid Simulate Feedback:%d\n",ret);
}
else
{
    printf("Pid Simulate Feedback:%f \n", Feedback);
}
```


7.9.18. ECAT_PidGet_Sp_Err_Op_Pv

Description:

Get the Set Point Value, Error, Control Output, and Process Variable of a PID control system. Users can use this function to get these values back efficiently.

Syntax:

int32_t ECAT_PidGet_Sp_Err_Op_Pv(uint16_t DeviceNo, uint32_t PidNo, double *SetPointValue, double *Error, double *OutputValue, double *ProcessVariable)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
PidNo	uint32_t	IN	PID Controller number
SetPointValue	double*	Output	Set Point Value (SP)
Error	double*	Output	Error (= SP-PV)
OutputValue	double*	Output	Control Output Value (CO)
ProcessVariable	double*	Output	ProcessVariable (PV)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Error= 0;
double ProcessVariable= 0;
double OutputValue= 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0)printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidGet_Sp_Err_Op_Pv(DeviceNo, PidNo, &SetPointValue, &Error, &OutputValue,
&ProcessVariable);
if(ret != 0)
{
    printf("Failed to Get Pid Sp_Err_Op_Pv:%d\n", ret);
}
else
```



```
{
        printf("Pid Set Point Value :%f \n", Setpoint);
        printf("Pid Error :%f \n", Error);
        printf("Pid OutputValue :%f \n", OutputValue);
        printf("Pid ProcessVariable:%f \n", ProcessVariable);
}
```


7.10. Stewart Platform

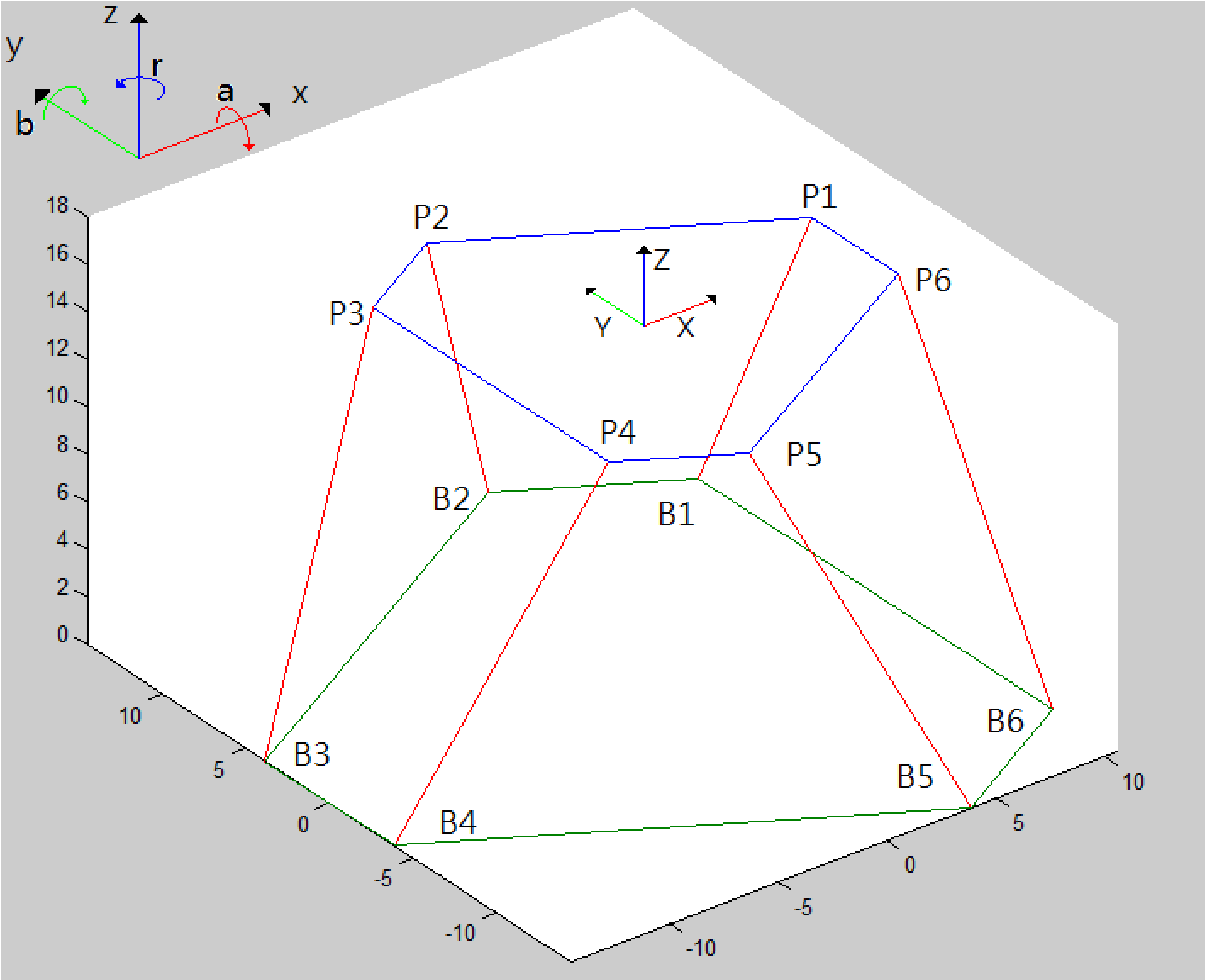
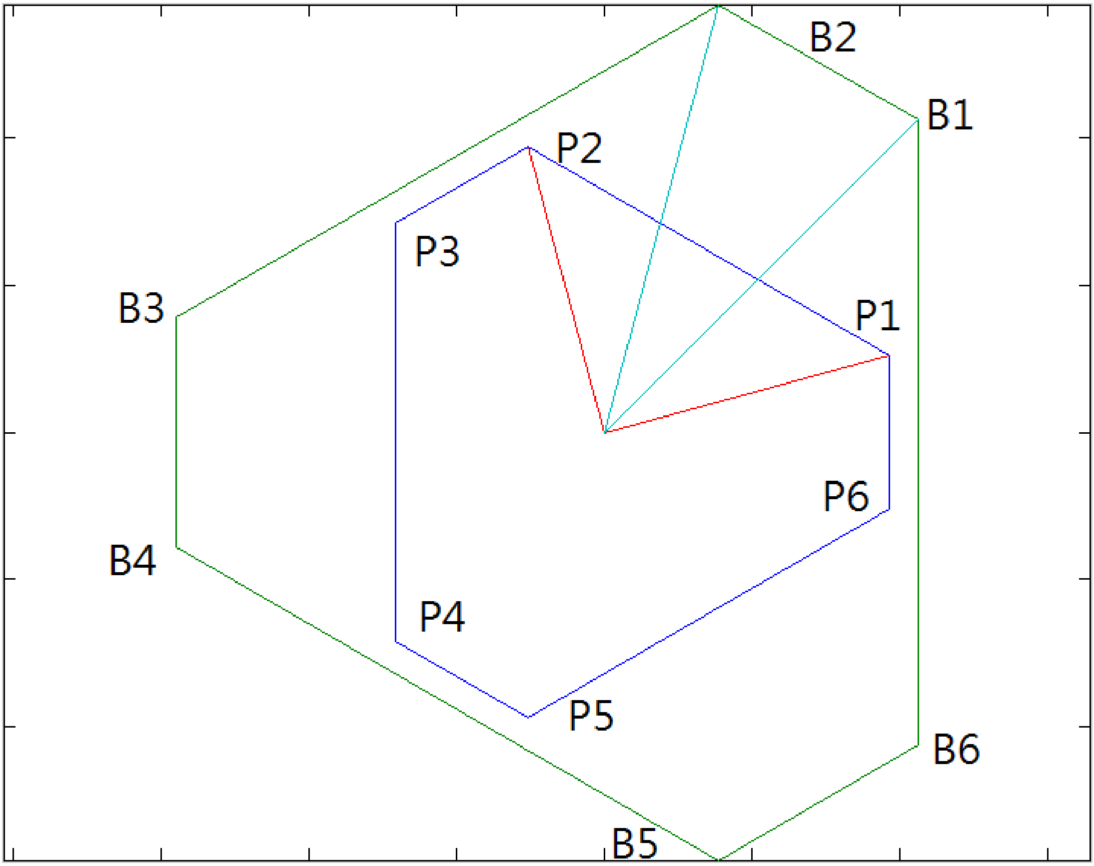
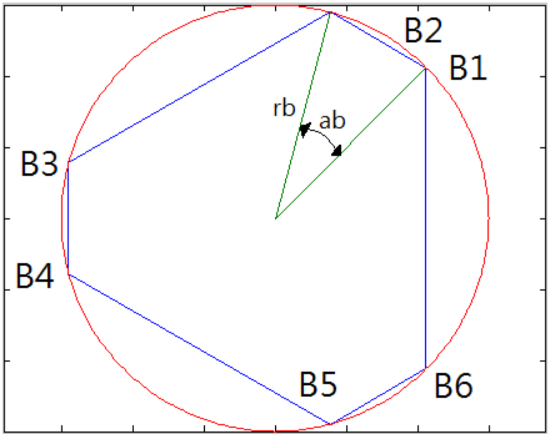


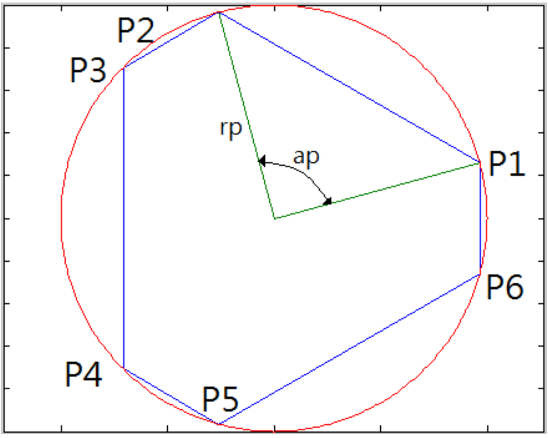
Figure 7.2

Top platform: the plane is formed by 6 Knots, P1 ~ P6

Base platform: the plane is formed by 6 Knots, B1 ~ B6







rb: Radius of the base platform

ab: The angle between B1, the center point of the base platform, and B2

rp: Radius of the top platform

ap: The angle between P1, the center point of the top platform, and P2

7.10.1. ECAT_McSetStewartPlatform_M1

Description:

Set geometric parameters for a stewart platform (method 1).

Syntax:

int32_t ECAT_McSetStewartPlatform_M1(uint16_t DeviceNo, double radiusB, double angleB, double radiusP, double angleP, double RodLength, double Max_RodLength)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
radiusB	double	IN	Radius of the base platform. Unit: mm
angleB	double	IN	The angle formed by B1, the center point of the base platform, and B2 Unit: degree
radiusP	double	IN	Radius of the top platform, Unit: mm
angleP	double	IN	The angle formed by P1, the center point of the top platform, and P2 Unit: degree
RodLength	double	IN	Minimum length of rod connecting base and top platforms. Unit: mm
Max_RodLength	double	IN	Maximum length of rod connecting base and top platforms. Unit: mm

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
double radiusB = 15;
double angleB = 30;
double radiusP = 10;
double angleP = 90;
double RodLength = 15;
double Max_RodLength = 30;

ret = ECAT_McSetStewartPlatform_M1(DeviceNo, radiusB, angleB, radiusP, angleP, RodLength,
Max_RodLength);
if(ret < 0)
{
    printf("Failed to Set Stewart Platform:%d\n", ret);
    return;
}
```


7.10.2. ECAT_McSetStewartPlatform_M1

Description:

Get geometric parameters of a stewart platform (method 1).

Syntax:

int32_t ECAT_McGetStewartPlatform_M1(uint16_t DeviceNo, double* radiusB, double* angleB, double* radiusP, double* angleP, double* RodLength, double* Max_RodLength)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
radiusB	double*	OUT	Radius of the base platform. Unit: mm
angleB	double*	OUT	The angle formed by B1, the center point of the base platform, and B2 Unit: degree
radiusP	double*	OUT	Radius of the top platform, Unit: mm
angleP	double*	OUT	The angle formed by P1, the center point of the top platform, and P2 Unit: degree
RodLength	double*	OUT	The minimum length of rod connecting base and top platforms. Unit: mm
Max_RodLength	double*	OUT	The maximum length of rod connecting base and top platforms. Unit: mm

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
double radiusB = 0;
double angleB = 0;
double radiusP = 0;
double angleP = 0;
double RodLength = 0;
double Max_RodLength = 0;

ret = ECAT_McGetStewartPlatform_M1(DeviceNo, &radiusB, &angleB, &radiusP, &angleP, &RodLength,
&Max_RodLength);
if(ret < 0)
{
    printf("Failed to Get Stewart Platform:%d\n", ret);
    return;
}
```

7.10.3. ECAT_McSetStewartPlatform_M2

Description:

Set geometric parameters of a stewart platform (method 2).

Syntax:

int32_t ECAT_McSetStewartPlatform_M2(uint16_t DeviceNo, double Bx[], double By[], double Px[], double Py[], double Z0, double RodLength[], double Max_RodLength[])

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Bx	double[]	IN	An array contains 6 elements. Each value is the X Coordinate of Bi, i = 1~6, Unit: mm
By	double[]	IN	An array contains 6 elements. Each value is the Y Coordinate of Bi, i = 1~6, Unit: mm
Px	double[]	IN	An array contains 6 elements. Each value is the X Coordinate of Pi, i = 1~6, Unit: mm
Py	double[]	IN	An array contains 6 elements. Each value is the Y Coordinate of Pi, i = 1~6, Unit: mm
Z0	double	IN	The vertical height of the top platform relative to the base platform. Unit: mm
RodLength	double[]	IN	The minimum length of rod connecting base and top platform. Unit: mm
Max_RodLength	double[]	IN	The maximum length of rod connecting base and top platform, Unit: mm

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
double Bx[6]= {10.6066,   3.8823,   -14.4889,   -14.4889,   3.8823,   10.6066};
double By[6]= {10.6066,   14.4889,   3.8823,   -3.8823,   -14.4889,   -10.6066};
double Px[6] = {9.6593,   -2.5882,   -7.0711,   -7.0711,   -2.5882,   9.6593};
double Py[6] = {2.5882,   9.6593,   7.0711,   -7.0711,   -9.6593,   -2.5882};
double Z0 = 14.1421;
double RodLength[6] = {15,   15,   15,   15,   15,   15 };
double Max_RodLength[6] = {30,   30,   30,   30,   30,   30};

ret = ECAT_McSetStewartPlatform_M2(DeviceNo, Bx, By, Px, Py, Z0, RodLength, Max_RodLength);
if(ret < 0)
{
    printf("Failed to Set Stewart Platform:%d\n", ret);
    return;
}
```

7.10.4. ECAT_McGetStewartPlatform_M2

Description:

Get geometric parameters for a stewart platform (method 2).

Syntax:

int32_t ECAT_McGetStewartPlatform_M2(uint16_t DeviceNo, double* Bx, double* By, double* Px, double* Py, double* Z0, double* RodLength, double* Max_RodLength)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Bx	double*	OUT	An array contains 6 elements. Each value is the X coordinate value of Bi, i = 1~6, Unit: mm
By	double*	OUT	An array contains 6 elements. Each value is the Y coordinate value of Bi, i = 1~6, Unit: mm
Px	double*	OUT	An array contains 6 elements. Each value is the X coordinate value of Pi, i = 1~6, Unit: mm
Py	double*	OUT	An array contains 6 elements. Each value is the Y coordinate value of Pi, i = 1~6, Unit: mm
Z0	double*	OUT	The initial distance between the center of base platform and the center of top platform. Unit: mm
RodLength	double*	OUT	The minimum length of rod connecting the base platform and the top platform. Unit: mm
Max_RodLength	double*	OUT	The maximum length of rod connecting

			the base platform and the top platform, Unit: mm

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
double Bx[6]= {0};
double By[6]= {0};
double Px[6] = {0};
double Py[6] = {0};
double Z0 = 0;
double RodLength[6] = {0};
double Max_RodLength[6] = {0};

ret = ECAT_McGetStewartPlatform_M2(DeviceNo, &Bx, &By, &Px, &Py, &Z0, &RodLength,
Max_RodLength);
if(ret < 0)
{
    printf("Failed to Get Stewart Platform:%d\n", ret);
    return;
}
```


7.10.5. ECAT_McStwartPlatformMoveAbs_PT

Description:

Start an absolute linear interpolation motion by providing world coordinate space positions and time for executing this motion command. This is a group motion command. The pose includes the 6-axis world coordinate space positions. A long-distance linear motion or circular motion can be approximated by many of these short-distance commands. ECAT-M801 has a 3000-depth command buffer. Users can send commands continuously to this card. If the command mode is Blending, this card will smoothly execute every desired motion command.

Note: At first, this card will process pose command obtain the targeted joint space positions by processing the inverse kinematics. Then a 6-axis linear interpolation motion in joint space is implemented for this motion. Actually, the linear interpolation is not implemented in the world coordinate system. Only continuous short-distance commands can approach nearly linear commands.

Syntax:

int32_t ECAT_McStwartPlatformMoveAbs_PT(uint16_t DeviceNo, uint16_t GroupNo, double Pose[], double* Pos, double Time)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
GroupNo	uint16_t	IN	Group number
Pose	double[]	IN	Requested pose in world coordinate system of the Stewart platform x: the displacement along X-axis. Unit: mm y: the displacement along Y-axis. Unit: mm z: the displacement along Z-axis. Unit: mm

			a: the rotating angle around the X-axis. Unit: degree b: the rotating angle around the Y-axis. Unit: degree r : the rotating angle around the Z-axis. Unit: degree Please refer to Figure 7.2 for the direction definitions for displacement and rotation.
Pos	double*	OUT	This array contains the targeting 6-axis joint space positions. Each element in this array is an absolute position. Unit: user unit
Time	double	IN	Time Unit: second

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t i;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BLENDING; //0: Aborting, 1: Buffered, 2: Blending
double StewartPlatformPose[6]; //x   y   z   a   b   r
double Pos[6]; //position of axis0~axis5
double GroupTime;
double radiusB = 15;
double angleB = 30;
double radiusP = 10;
double angleP = 90;
double RodLength = 15;
double Max_RodLength = 30;

ret = ECAT_McSetStewartPlatform_M1(DeviceNo, radiusB, angleB, radiusP, angleP, RodLength,
Max_RodLength);
if(ret < 0)
{
    printf("Failed to Set Stewart Platform:%d\n", ret);
    return;
}

for(i=0;i<6;i++)//6-axis Stewart Platform
{
    ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, i);
    if(ret < 0)
    {
        printf("Failed to add axis to group:%d\n", ret);
        return;
    }
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
```



```
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    StewartPlatformPose [0] = 0; // x
    StewartPlatformPose [1] = 0; // y
    StewartPlatformPose [2] = 1; // z
    StewartPlatformPose [3] = 0; // a
    StewartPlatformPose [4] = 0; // b
    StewartPlatformPose [5] = 0; // r
    GroupTime = 1;
    ret = ECAT_McStwartPlatformMoveAbs_PT(DeviceNo, GroupNo, StewartPlatformPose, &Pos,
GroupTime);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }
    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```


7.11. Motion Data Recorder

7.11.1. ECAT_McSetMotionRecord

Description:

This function can start or stop an ECAT-M801 to record the position and/or velocity of axes. Inside the ECAT-M801, the program can save a record for each cycle time. Up to 100,000 records can be saved.

Note: This function will not clear record count to 0. Users can clear record count with function *ECAT_McClearMotionRecord*.

Syntax:

int32_t ECAT_McSetMotionRecord(uint16_t DeviceNo, uint16_t state)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
state	uint16_t	IN	1: Start recording data 0: Stop recording data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
ret = ECAT_McSetMotionRecord(DeviceNo, 1);
if(ret < 0)
{
	printf("Failed to Set Motion Record:%d\n", ret);
}
else
{
	printf("Set Motion Record successfully! \n");
}
```


7.11.2. ECAT_McGetMotionRecordState

Description:

Get the recording status.

Syntax:

int32_t ECAT_McGetMotionRecordState(uint16_t DeviceNo, uint16_t *state, uint32_t *count)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
state	uint16_t*	OUT	Recording or not 1: Recording 0: Not recording
count	uint32_t*	OUT	Count of recorded data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t state;
uint32_t cnt;
ret = ECAT_McGetMotionRecordState(DeviceNo, &state, &cnt);
if(ret < 0)
{
    printf("Failed to Get Motion Record State: %d\n", ret);
}
else
{
    printf("State: %u , Count: %u \n", state, cnt);
}
```


7.11.3. ECAT_McClearMotionRecord

Description:

Clear the counting index to 0. If recording is enabled, the counting number is started from the current counting index instead of always counting from 0.

Syntax:

int32_t ECAT_McClearMotionRecord(uint16_t DeviceNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
ret = ECAT_McClearMotionRecord(DeviceNo);
if(ret < 0)
{
	printf("Failed to Clear Motion Record:%d\n", ret);
}
```


7.11.4. ECAT_McSetMotionRecordParam

Description:

Set parameters for deciding which two out of four values are goning to be recorded. Please refer to Table 7.7, the candidated four values are Actual Position, Actual Velocy, Command Position, and Command Velocity.

Syntax:

int32_t ECAT_McSetMotionRecordParam(uint16_t DeviceNo, uint16_t Value1, uint16_t Value2)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Value1	uint16_t	IN	The first motion parameter for recording (Refer to Table 7.7)
Value2	uint16_t	IN	The second motion parameter for recording (Refer to Table 7.7)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.7 Motion parameters for recording

Macro Definition	Value	Description
MC_RECORD_POSITION	0	Actual Position of Axis (Unit: user unit)
MC_RECORD_VELOCITY	1	Actual Velocity of Axis (Unit: user unit/second)
MC_RECORD_COMMAND_POSITION	2	Command Position of Axis (Unit: user unit)
MC_RECORD_COMMAND_VELOCITY	3	Command Velocity of Axis (Unit: user unit/second)

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t Value1= MC_RECORD_POSITION;
uint16_t Value2= MC_RECORD_VELOCITY;

ret = ECAT_McSetMotionRecordParam(DeviceNo, Value1, Value2);
if(ret < 0)
{
	printf("Failed to set motion record parameters:%d\n",ret);
}
ret = ECAT_McSetMotionRecord(DeviceNo, 1);
if(ret < 0)
	printf("Failed to Set Motion Record:%d\n", ret);
```

7.11.5. ECAT_McGetMotionRecordParam

Description:

Get the settings of the recorded parameters.

Syntax:

int32_t ECAT_McGetMotionRecordParam(uint16_t DeviceNo, uint16_t *Value1, uint16_t *Value2)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
Value1	uint16_t*	OUT	The first motion parameter for recording (Refer to Table 7.7)
Value2	uint16_t*	OUT	The second motion parameter for recording (Refer to Table 7.7)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint16_t *Value1;
uint16_t *Value2;

ret = ECAT_McGetMotionRecordParam(DeviceNo, &Value1, &Value2);
if(ret < 0)
{
	printf("Failed to get motion record parameters:%d\n", ret);
}
else
{
	printf("Value1:%u , Value2:%u \n", Value1, Value2);
}
```


7.11.6. ECAT_McGetMotionRecordValue

Description:

Get parameter values of an assigned axis at an assigned index number.

Note: When the AxisNo is set to 65535, values of all axes at the assigned index number are returned by Value1and Value2 pointers.

Syntax:

int32_t ECAT_McGetMotionRecordValue(uint16_t DeviceNo, uint32_t CountNo, uint16_t AxisNo, float *Value1, float *Value2)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number (Card ID)
CountNo	uint32_t	IN	Count Numer (an index number)
AxisNo	uint16_t	IN	Axis Number
Value1	float*	OUT	Value of the first parameter recorded at the specified Count Numer
Value2	float*	OUT	Value of the second parameter recorded at the specified Count Numer

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint32_t CountNo= 0;
uint16_t AxisNo = 0;
uint16_t state;
uint32_t cnt;
int i;
uint16_t Param1= MC_RECORD_POSITION;
uint16_t Param2= MC_RECORD_VELOCITY;
float Value1;
float Value2;

ret = ECAT_McSetMotionRecordParam(DeviceNo, Param1, Param2);
if(ret < 0)
	printf("Failed to set motion record parameters:%d\n", ret);
ret = ECAT_McSetMotionRecord(DeviceNo,1); // start record
if(ret < 0)
	printf("Failed to Set Motion Record:%d\n", ret);

Sleep(1000); //wait for record something…

ret = ECAT_McSetMotionRecord(DeviceNo, 0); // stop record
if(ret < 0)
	printf("Failed to Set Motion Record:%d\n", ret);

ret = ECAT_McGetMotionRecordState(DeviceNo, &state, &cnt); // get record count
if(ret < 0)
	printf("Failed to Get Motion Record State:%d\n", ret);
for(i=0;i< cnt;i++)
{
	ret = ECAT_McGetMotionRecordValue(DeviceNo, i , AxisNo, &Value1, &Value2);
	if(ret < 0)
	{
		printf("Failed to get motion record value:%d\n", ret);
	}
```



```
        else
        {
              printf("Axis Value1:%f , Value2:%f \n", Value1, Value2);
        }
    }
    ret = ECAT_McClearMotionRecord(DeviceNo);
    if(ret < 0)
        printf("Failed to Clear Motion Record:%d\n",ret);

}
```

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;
uint32_t CountNo= 0;
uint16_t AxisNo = 65535;
uint16_t state;
uint32_t cnt;
int i;
uint16_t Param1= MC_RECORD_POSITION;
uint16_t Param2= MC_RECORD_VELOCITY;
float Value1[MC_AXIS_NO_MAX];
float Value2[MC_AXIS_NO_MAX];

ret = ECAT_McSetMotionRecordParam(DeviceNo, Param1, Param2);
if(ret < 0)
    printf("Failed to set motion record parameters:%d\n", ret);
ret = ECAT_McSetMotionRecord(DeviceNo,1); // start record
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n", ret);

Sleep(1000); //wait for record something…

ret = ECAT_McSetMotionRecord(DeviceNo, 0); // stop record
if(ret < 0)
```



```
        printf("Failed to Set Motion Record:%d\n", ret);

ret = ECAT_McGetMotionRecordState(DeviceNo, &state, &cnt); // get record count
if(ret < 0)
        printf("Failed to Get Motion Record State:%d\n", ret);
for( i=0; i< cnt; i++)
{
        ret = ECAT_McGetMotionRecordValue(DeviceNo, i , AxisNo, Value1, Value2);
        if(ret < 0)
        {
                printf("Failed to get motion record value:%d\n", ret);
        }
        else
        {
                for( j=0; j< MC_AXIS_NO_MAX; j++)
                {
                        printf("Axis Value1:%f , Value2:%f \n", Value1[ j ] , Value2[ j ]);
                }
        }
}
ret = ECAT_McClearMotionRecord(DeviceNo);
if(ret < 0)
        printf("Failed to Clear Motion Record:%d\n", ret);

}
```

8. Appendix

8.1. Error Codes

Error ID	Error Code	Description
ECAT_ERR_REQUEST_MASTER	-1001	Failed to request master
ECAT_ERR_ETHERNET_LINK_DOWN	-1002	Ethernet network link status is down
ECAT_ERR_SLAVES_STATE	-1003	Not all slaves are in state OPERATIONAL
ECAT_ERR_WORKING_COUNTER	-1004	Working counter mismatch
ECAT_ERR_SLAVE_CNT_EXCEEDED	-1005	Connected slave count exceeds maximum support slave count
ECAT_ERR_CREATE_DOMAIN	-1006	Failed to create domain data
ECAT_ERR_ALLOCATE_SLAVE_DATA	-1007	Failed to allocate slave data
ECAT_ERR_CONFIG_SLAVE	-1008	Failed to configure slaves
ECAT_ERR_NETWORK_MISMATCH	-1009	Currently connected bus topology does not match configured one
ECAT_ERR_MASTER_ACTIVATE	-1010	Failed to activate master
ECAT_ERR_GET_PROCESS_DATA	-1011	Failed to get domain process data
ECAT_ERR_CONFIG_CYCLIC_TASK	-1012	Failed to configure cyclic task
ECAT_ERR_RUN_CYCLIC_TASK	-1013	Failed to run cyclic task
ECAT_ERR_INVALID_SLAVE_TYPE	-1014	Invalid slave type

ECAT_ERR_SAME_SLAVE_NO	-1015	Same slave number
ECAT_ERR_INVALID_SLAVE_NO	-1016	Invalid slave number
ECAT_ERR_INVALID_PARAM	-1017	Invalid parameter
ECAT_ERR_INVALID_DATA_SIZE	-1018	Invalid size of data
ECAT_ERR_SDO_REQUEST_BUSY	-1019	SDO request is being processed
ECAT_ERR_SDO_REQUEST_ERROR	-1020	SDO request processing failed
ECAT_ERR_ALLOCATE_PDO_QUEUE	-1021	Failed to allocate PDO queue data
ECAT_ERR_INVALID_OFFSET	-1022	Invalid data offset
ECAT_ERR_INIT_MOTION	-1023	Failed to initialize motion
ECAT_ERR_GET_SLAVE_INFO	-1024	Failed to get slave information
ECAT_ERR_OPEN_FILE	-1025	Failed to open file
ECAT_ERR_WRITE_FILE	-1026	Failed to write data to file
ECAT_ERR_READ_FILE	-1027	Failed to read data from file
ECAT_ERR_FUNC_NOT_SUPPORT	-1028	Function is not supported
ECAT_ERR_INVALID_CHANNEL	-1029	Invalid channel parameter
ECAT_ERR_EMG_HAPPENED	-1030	Emergency happened
ECAT_ERR_INVALID_PID_NO	-1031	Invalid PID number
ECAT_ERR_TIMER_NOT_ACTIVATED	-1032	Timer is not activated
ECAT_ERR_MC_NOT_ENABLE_DC	-1100	Not enable DC
ECAT_ERR_MC_TIME_OUT	-1101	Call motion function time out
ECAT_ERR_MC_AXIS_CNT_EXCEEDED	-1102	Initialized axis count exceeds maximum support axis count

ECAT_ERR_MC_NOT_INITIALIZED	-1103	Motion is not initialized
ECAT_ERR_MC_INVALID_AXIS_NO	-1104	Invalid axis number
ECAT_ERR_MC_NOT_AXIS_SERVO_ON	-1105	Axis is not servo-on
ECAT_ERR_MC_INVALID_AXIS_STATE	-1106	Invalid axis state
ECAT_ERR_MC_DRIVE_FAULT	-1107	Drive fault
ECAT_ERR_MC_DRIVE_WARNING	-1108	Drive warning
ECAT_ERR_MC_INVALID_PARAM	-1109	Invalid motion parameter
ECAT_ERR_MC_HOMING	-1110	An error occurs when the homing
ECAT_ERR_MC_LIMIT_ACTIVE	-1111	Limit switch is active
ECAT_ERR_MC_INVALID_ACC_TIME	-1112	Invalid acceleration time
ECAT_ERR_MC_INVALID_GROUP_NO	-1113	Invalid group number
ECAT_ERR_MC_INVALID_GROUP_STATE	-1114	Invalid group state
ECAT_ERR_MC_AXIS_WAS_IN_GROUP	-1115	Axis is already in group
ECAT_ERR_MC_AXIS_IN_OTHER_GROUP	-1116	Axis is already in other group
ECAT_ERR_MC_GROUP_CMD_ALLOCATE	-1117	Failed to allocate group command
ECAT_ERR_MC_GROUP_CMD_BUFFER_OVERFLOW	-1118	Group command is overflow
ECAT_ERR_MC_INVALID_AXIS_SYNC_MODE	-1119	Invalid axis synchronization mode
ECAT_ERR_MC_INVALID_PROFILE_NO	-1120	Invalid profile number
ECAT_ERR_MC_INVALID_GROUP_MOVE_CMD	-1121	Invalid group command
ECAT_ERR_MC_GROUP_CMD_MODE_NOT_SUPPORT	-1122	The function does not support the current group command mode
ECAT_ERR_MC_INVALID_ACC_DEC_TYPE	-1123	Invalid acceleration type parameter
ECAT_ERR_MC_INVALID_VEL	-1124	Invalid velocity parameter

ECAT_ERR_MC_INVALID_ANGLE	-1125	Invalid angle parameter
ECAT_ERR_MC_INVALID_RADIUS	-1126	Invalid radius parameter
ECAT_ERR_MC_INVALID_END_POS	-1127	Invalid end position parameter
ECAT_ERR_MC_INVALID_ECAM_TABLE_NO	-1128	Invalid E-CAM table number
ECAT_ERR_MC_INVALID_NORMAL_VECTOR	-1129	Invalid normal vector parameter
ECAT_ERR_MC_NOT_SETUP	-1130	not setup
ECAT_ERR_MC_GREATER_THAN_MAX_RODLENGTH	-1131	calculated value is greater than maximum rod length
ECAT_ERR_MC_LESS_THAN_RODLENGTH	-1132	calculated value is less than rod length
ECAT_ERR_MC_GREATER_THAN_RECORD_COUNT	-1133	exceede maximum record count
ECAT_ERR_MC_SOFTWARE_LIMIT_ACTIVATE	-1134	software limit is active
ECAT_ERR_MC_GANTRY_POS_EXCESSIVE_DEVIATION	-1135	position excessive deviation of gantry control
ECAT_ERR_IPC_INVALID_DEVICE_NO	-1201	Invalid device number
ECAT_ERR_IPC_DEVICE_IS_OPEN	-1202	Device is open
ECAT_ERR_IPC_DEVICE_NOT_OPEN	-1203	Device is not open
ECAT_ERR_IPC_CREATE_HANDLE	-1204	Failed to create IPC handle
ECAT_ERR_IPC_BUSY	-1205	IPC is busy
ECAT_ERR_IPC_TIME_OUT	-1206	IPC is time out
ECAT_ERR_IPC_INVALID_CMD	-1207	Invalid IPC command
ECAT_ERR_IPC_WRITE_SHM	-1208	Failed to write data to shard memory
ECAT_ERR_IPC_READ_SHM	-1209	Failed to read data from

		shard memory
ECAT_ERR_IPC_RUN_DOWN_UP_LOAD	-1210	Failed to process download/upload data
ECAT_ERR_IPC_INVALID_SHM	-1211	Invalid shard memory
ECAT_ERR_IPC_DEVICE_NOT_READY	-1212	Device is not ready
ECAT_ERR_DRV_GET_INFO	-1301	Failed to get driver information
ECAT_ERR_DRV_CREATE_HANDLE	-1302	Failed to create driver handle
ECAT_ERR_DRV_IOCTL	-1303	Call driver IO control error
ECAT_ERR_DRV_DEVICE_NOT_FOUND	-1304	Device not found

8.2. SDO Abort messages

Abort code	Description
0x05030000	Toggle bit not changed
0x05040000	SDO protocol timeout
0x05040001	Client/Server command specifier not valid or unknown
0x05040005	Out of memory
0x06010000	Unsupported access to an object
0x06010001	Attempt to read a write-only object
0x06010002	Attempt to write a read-only object
0x06020000	This object does not exist in the object directory
0x06040041	The object cannot be mapped into the PDO
0x06040042	The number and length of the objects to be mapped would exceed the PDO length
0x06040043	General parameter incompatibility reason
0x06040047	General internal incompatibility in device
0x06060000	Access failure due to a hardware error
0x06070010	Data type does not match, length of service parameter does not match
0x06070012	Data type does not match, length of service parameter too high
0x06070013	Data type does not match, length of service parameter too low
0x06090011	Sub-index does not exist
0x06090030	Value range of parameter exceeded
0x06090031	Value of parameter written too high
0x06090032	Value of parameter written too low
0x06090036	Maximum value is less than minimum value
0x08000000	General error
0x08000020	Data cannot be transferred or stored to the application
0x08000021	Data cannot be transferred or stored to the application

	because of local control
0x08000022	Data cannot be transferred or stored to the application because of the present device state
0x08000023	Object dictionary dynamic generation fails or no object dictionary is present

8.3. Revision History

This chapter provides revision history information to this document

The table below shows the revision history.

Revision	Date	Description
1.0	2018	Initial issue