EtherCAT Master Software Manual

English Ver. 1.0.0, JUL. 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

Copyright © 2017 by ICP DAS. All rights are reserved.

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

Contens

1. Introduction 8
2. Software Installation 9
2.1. Obtaining the Driver Installer Package 9
2.2. Driver Installing Procedure 10
2.3. Uninstalling the Driver 17
2.4. Driver Installing Manually 19
2.4.1. Step1 19
2.4.2. Step2 22
2.4.3. Step3 23
3. EcatUtility 27
3.1. Device Operation Toolbar 28
3.1.1. Device Initialization Steps 28
3.1.2. Network Information Edit Steps 29
3.1.3. Start/Stop the EtherCAT Operation Task Steps 32
3.2. Message Panel 33
3.3. Device Network Status 33
3.4. Slave Operation Page 34
3.4.1. Basic Slave Operation Steps 35
3.4.2. Slave PDO and DI/DO LED Operation Steps 35
3.5. Motion Control Initialization Toolbar 36
3.5.1. Motion Control Parameter File Editing Steps 37
3.5.2. Motion Control Initialization Steps 39
3.6. Motion Control Page 40
3.6.1. Single-Axis Motion Control Page 40
3.6.2. Group Motion Control Page 43
3.6.3. Group Show Position Page 47
3.7. Device I/O Operation Page 48
3.7.1. Device DO control operation step 49
3.7.2. Device DI control operation step 49
3.8. PID Control Page 50
3.8.1. PID Control Page 50
4. Function Overview 52
4.1. Device Operation Flow 52
4.2. Slave Operation Flow 54
4.3. Motion Control Flow 55

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

4.3.1. Motion Control Initialization 55
4.3.2. Axis Motion Control 57
4.3.3. Axis Homing 59
4.3.4. Axis Error Process 61
4.3.5. Group Moving 63
4.4. Use motion Library in Windows 65
4.4.1. For Visual Studio 65
5. Device Operation Functions 67
5.1. ECAT_GetDeviceCnt 67
5.2. ECAT_OpenDevice 69
5.3. ECAT_CloseDevice 71
5.4. ECAT_GetDeviceDI 73
5.5. ECAT_ GetDeviceDIBit 75
5.6. ECAT_GetDeviceDO 77
5.7. ECAT_ GetDeviceDOBit 79
5.8. ECAT_SetDeviceDO 81
5.9. ECAT_ SetDeviceDOBit 83
5.10. ECAT_SetDeviceEncProperty 85
5.11. ECAT_GetDeviceEncProperty 87
5.12. ECAT_GetDeviceEncCount 89
5.13. ECAT_ResetDeviceEncCount 91
5.14. ECAT_SetDeviceCmpTrigProperty 93
5.15. ECAT_GetDeviceCmpTrigProperty 95
5.16. ECAT_SetDeviceCmpTrigData 97
5.17. ECAT_SetDeviceContCmpTrigData 99
5.18. ECAT_SetDeviceEmg 101
5.19. ECAT_GetDeviceEmg 103
5.20. ECAT_GetDeviceEmgStatus 105
5.21. ECAT_SetDeviceEmgSoftSig 107
5.22. ECAT_SetDeviceMPG 109
5.23. ECAT_GetDeviceMPG 111
5.24. ECAT_GetDeviceState 113
5.25. ECAT_StartDeviceOpTask 116
5.26. ECAT_StopDeviceOpTask 119
5.27. ECAT_SetTimer 121
5.28. ECAT_ SetTimerStop 123
5.29. ECAT_WaitforTimer 125
6. Slave Operation Functions 127

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.1. ECAT_GetSlaveInfo 127
6.2. ECAT_GetSlaveSdoObject 130
6.3. ECAT_SetSlaveSdoObject 132
6.4. ECAT_SetSlaveRxPdoData 134
6.5. ECAT_GetSlaveRxPdoData 136
6.6. ECAT_GetSlaveTxPdoData 138
6.7. ECAT_GetSlaveDI 140
6.8. ECAT_GetSlaveDIBit 142
6.9. ECAT_GetSlaveDO 144
6.10. ECAT_GetSlaveDOBit 146
6.11. ECAT_SetSlaveDO 148
6.12. ECAT_SetSlaveDOBit 150
6.13. ECAT_SetSlaveAoProperty 152
6.14. ECAT_GetSlaveAoProperty 154
6.15. ECAT_SetSlaveAoRawData 156
6.16. ECAT_GetSlaveAoRawData 158
6.17. ECAT_SetSlaveAoVoltData 160
6.18. ECAT_GetSlaveAoVoltData 162
6.19. ECAT_SetSlaveEncProperty 164
6.20. ECAT_GetSlaveEncProperty 166
6.21. ECAT_GetSlaveEncCount 168
6.22. ECAT_ResetSlaveEncCount 170
6.23. ECAT_SetSlaveEncLatch 172
6.24. ECAT_GetSlaveEncLatch 174
6.25. ECAT_GetSlaveEncLatchCnt 176
7. Motion Control Functions 178
7.1. Motion Control Initialization 178
7.1.1. ECAT_McInit 178
7.2. Axis Parameter Settings 181
7.2.1. ECAT_McSetAxisServoOn 181
7.2.2. ECAT_McSetAxisPPU 183
7.2.3. ECAT_McGetAxisPPU 185
7.2.4. ECAT_McSetProfileData 187
7.2.5. ECAT_McGetProfileData 189
7.2.6. ECAT_McSetAxisAccTime 191
7.2.7. ECAT_McGetAxisAccTime 193
7.2.8. ECAT_McSetAxisAccDecType 195
7.2.9. ECAT_McGetAxisAccDecType 197

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.10. ECAT_McSetAxisEncoderPPR 199
7.2.11. ECAT_McGetAxisEncoderPPR 201
7.2.12. ECAT_McSetAxisMotorPPR 203
7.2.13. ECAT_McGetAxisMotorPPR 205
7.2.14. ECAT_McSetEcamTable 207
7.2.15. ECAT_McGetEcamTable 209
7.2.16. ECAT_McConfigEcamTable 211
7.2.17. ECAT_McSetAxisTouchProbeProperty 213
7.2.18. ECAT_McGetAxisTouchProbeProperty 215
7.2.19. ECAT_McGetAxisTouchProbeValue 217
7.3. Axis Status 219
7.3.1. ECAT_McGetAxisActualPos 219
7.3.2. ECAT_McGetAxisActualVel 221
7.3.3. ECAT_McGetAxisState 223
7.3.4. ECAT_McGetAxisLastError 226
7.3.5. ECAT_McGetAxisDriveError 228
7.3.6. ECAT_McGetAxisDI 231
7.4. Axis Homing 234
7.4.1. ECAT_McSetAxisHomeMethod 234
7.4.2. ECAT_McGetAxisHomeMethod 236
7.4.3. ECAT_McSetAxisHomeSpeed 238
7.4.4. ECAT_McGetAxisHomeSpeed 240
7.4.5. ECAT_McSetAxisHomeAcc 242
7.4.6. ECAT_McGetAxisHomeAcc 244
7.4.7. ECAT_McSetAxisHomeOffset 246
7.4.8. ECAT_McGetAxisHomeOffset 248
7.4.9. ECAT_McAxisHome 250
7.5. Axis Moving 253
7.5.1. ECAT_McAxisErrorReset 253
7.5.2. ECAT_McAxisMoveAbs 255
7.5.3. ECAT_McAxisMoveRel 258
7.5.4. ECAT_McAxisChangePos 261
7.5.5. ECAT_McAxisChangeVel 264
7.5.6. ECAT_McAxisMoveVel 267
7.5.7. ECAT_McAxisGearIn 269
7.5.8. ECAT_McAxisGearOut 272
7.5.9. ECAT_McAxisMoveProfile 274
7.5.10. ECAT_McAxisCamIn 276

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.11. ECAT_McAxisStop 279
7.5.12. ECAT_McAxisQuickStop 282
7.6. Group Parameter Setting 285
7.6.1. ECAT_McAddAxisToGroup 285
7.6.2. ECAT_McRemoveAxisFromGroup 287
7.6.3. ECAT_McUngroupAllAxes 289
7.6.4. ECAT_McSetGroupCmdMode 291
7.6.5. ECAT_McGetGroupCmdMode 293
7.7. Group Status 295
7.7.1. ECAT_McGetGroupState 295
7.7.2. ECAT_McGetGroupCmdBuffer 298
7.8. Group Moving 300
7.8.1. ECAT_McGroupMoveLineAbs 300
7.8.2. ECAT_McGroupMoveLineRel 304
7.8.3. ECAT_McGroupMoveLineAbs_PT 308
7.8.4. ECAT_McGroupMoveLineRel_PT 311
7.8.5. ECAT_McGroupMoveLineAbs_PVT 314
7.8.6. ECAT_McGroupMoveLineRel_PVT 318
7.8.7. ECAT_McGroupMoveCircularAbs_CP_Angle 322
7.8.8. ECAT_McGroupMoveCircularRel_CP_Angle 325
7.8.9. ECAT_McGroupMoveCircularAbs_CP_EP 328
7.8.10. ECAT_McGroupMoveCircularRel_CP_EP 332
7.8.11. ECAT_McGroupMoveCircularAbs_BP_EP 335
7.8.12. ECAT_McGroupMoveCircularRel_BP_EP 339
7.8.13. ECAT_McGroupMove3DCircularAbs_CP_Angle 342
7.8.14. ECAT_McGroupMove3DCircularRel_CP_Angle 346
7.8.15. ECAT_McGroupMove3DCircularAbs_CP_EP 349
7.8.16. ECAT_McGroupMove3DCircularRel_CP_EP 353
7.8.17. ECAT_McGroupMove3DCircularAbs_BP_EP 356
7.8.18. ECAT_McGroupMove3DCircularRel_BP_EP 360
7.8.19. ECAT_McGroupMoveHelicalAbs 363
7.8.20. ECAT_McGroupMoveHelicalRel 366
7.8.21. ECAT_McGroupMove3DHelicalAbs_CP_Angle 370
7.8.22. ECAT_McGroupMove3DHelicalRel_CP_Angle 374
7.8.23. ECAT_McGroupMoveConicalHelixAbs 377
7.8.24. ECAT_ McGroupMoveConicalHelixRel 381
7.8.25. ECAT_McGroupMove3DConicalHelixAbs_CP_Angle 384
7.8.26. ECAT_McGroupMove3DConicalHelixRel_CP_Angle 388

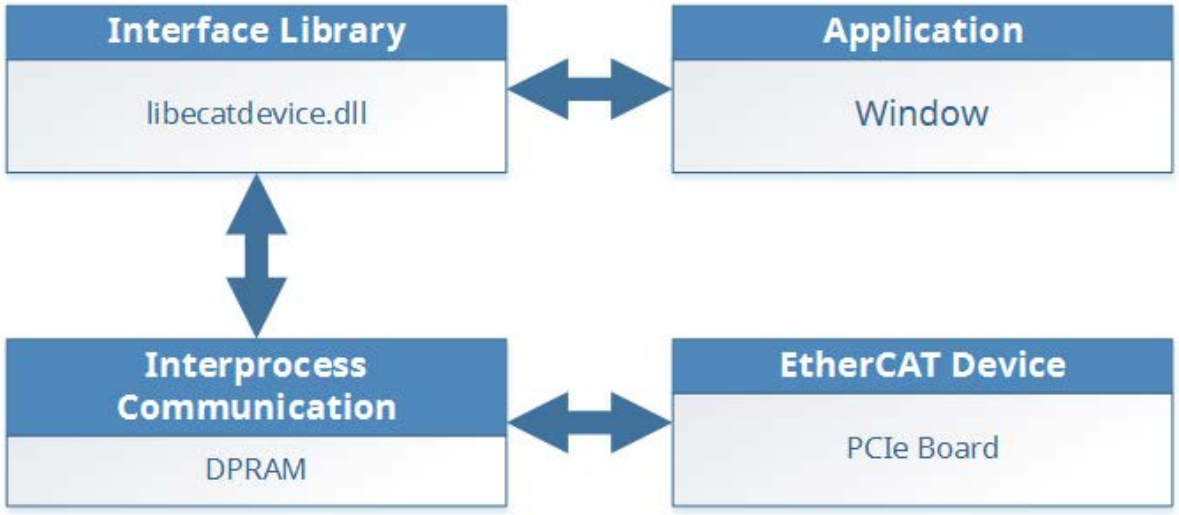
Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.27. ECAT_McGroupMoveProfile 391
7.8.28. ECAT_McGroupMoveDwell 394
7.8.29. ECAT_McGroupStop 397
7.8.30. ECAT_McGroupQuickStop 400
7.9. PID Controller 403
7.9.1. ECAT_PidGetSetPointValue 404
7.9.2. ECAT_PidSetSetPointValue 407
7.9.3. ECAT_PidGetProcessVariable 409
7.9.4. ECAT_PidGetSampleTime 412
7.9.5. ECAT_PidSetSampleTime 414
7.9.6. ECAT_PidGetStatus 416
7.9.7. ECAT_PidSetStatus 418
7.9.8. ECAT_PidGetSimulateMode 420
7.9.9. ECAT_PidSetSimulateMode 422
7.9.10. ECAT_PidGetParameter 424
7.9.11. ECAT_PidSetParameter 426
7.9.12. ECAT_PidGetProcessVariableModule 428
7.9.13. ECAT_PidSetProcessVariableModule 430
7.9.14. ECAT_PidGetControlOutputModule 433
7.9.15. ECAT_PidSetControlOutputModule 435
7.9.16. ECAT_PidGetControlOutputValue 438
7.9.17. ECAT_PidGetSimulateFeedback 441
7.9.18. ECAT_PidGet_Sp_Err_Op_Pv 444
8. Appendix 447
8.1. Error Codes 447
8.2. SDO Abort messages 451
8.3. Revision History 453

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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 subroutine for developing your application. To speed-up your developing process . The library architecture is shown in the following figure



Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

2. Software Installation

This chapter instructs how to get and install 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 obtained from ether ICP DAS FTP or web site. The location and addresses are in the table below:

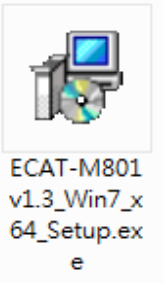
	http://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/master/ecat-m801/manual/
	ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/master/ecat-m801/manual/

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

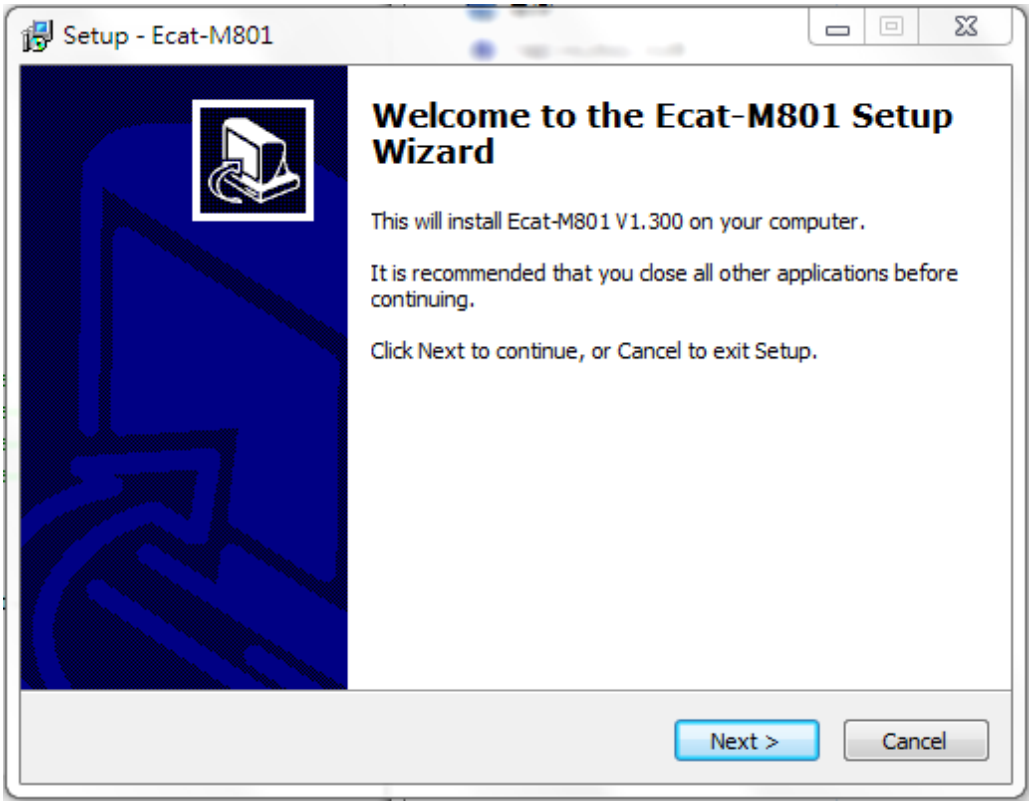
2.2. Driver Installing Procedure

To install drivers, follow the procedure described below:

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

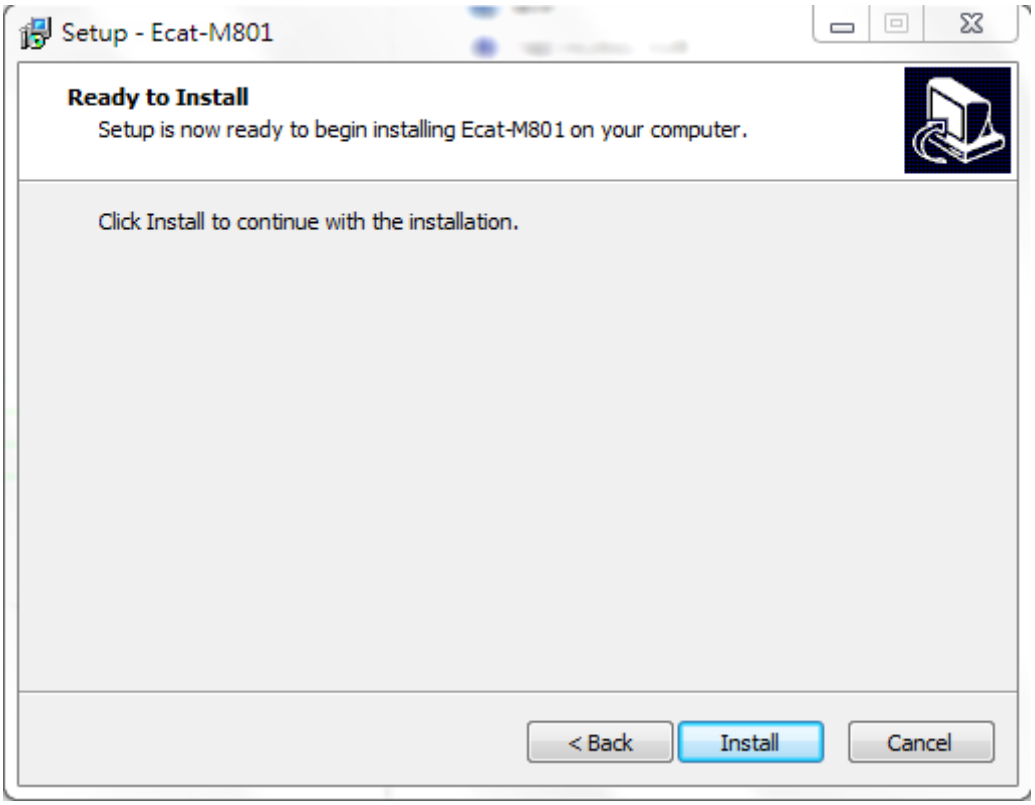


2. Click ther “Next >” button.

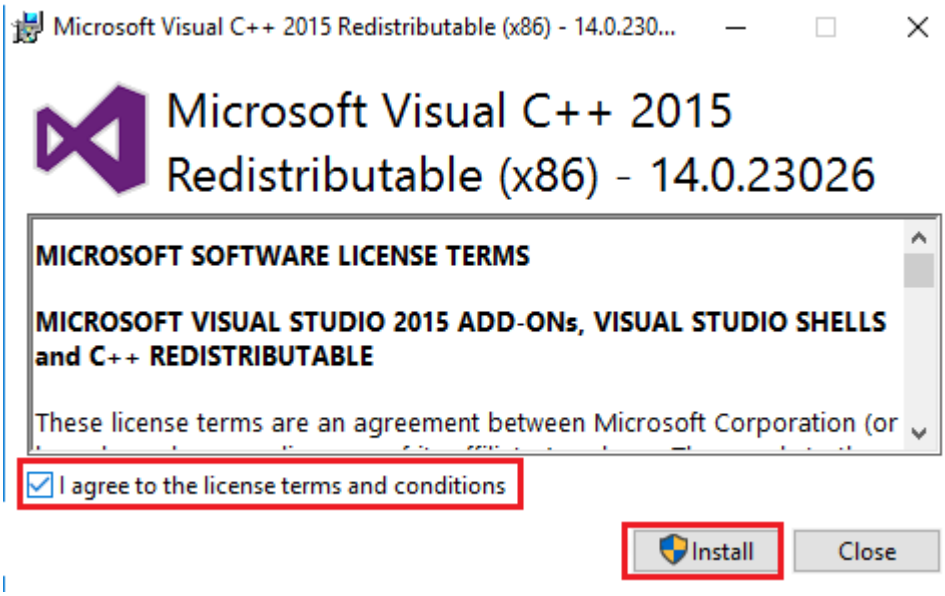


Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

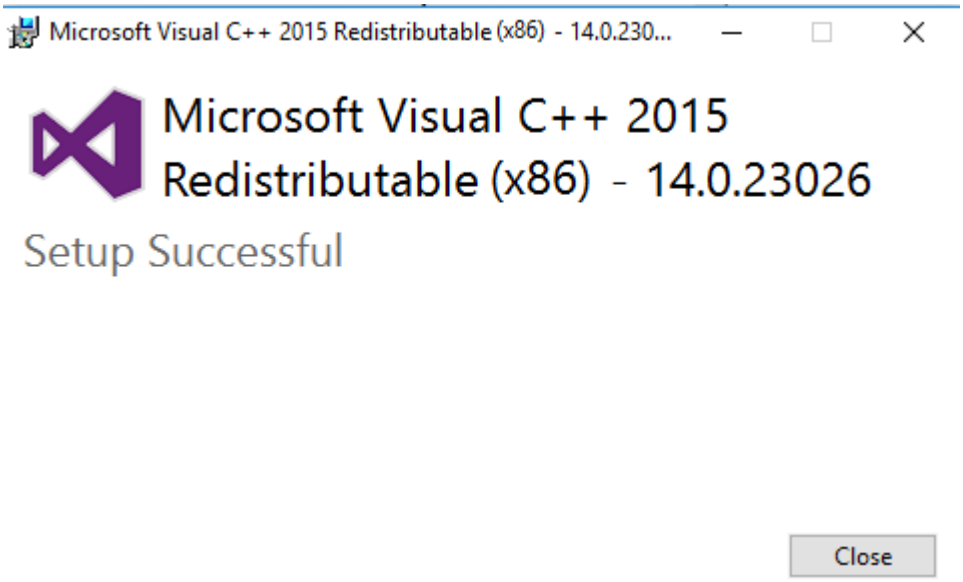


4. Check "I agree to the license terms and conditions" , Click the "Install" button to continue.



5. Click the "Close" button to continue.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

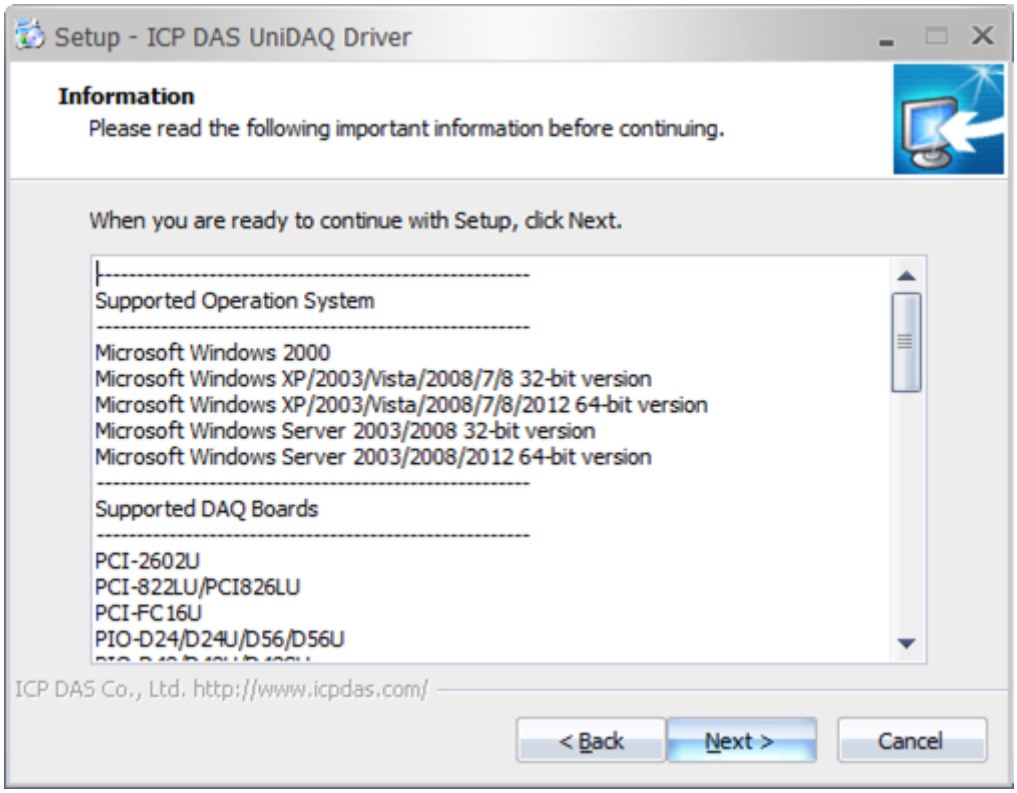


6. Click Next>

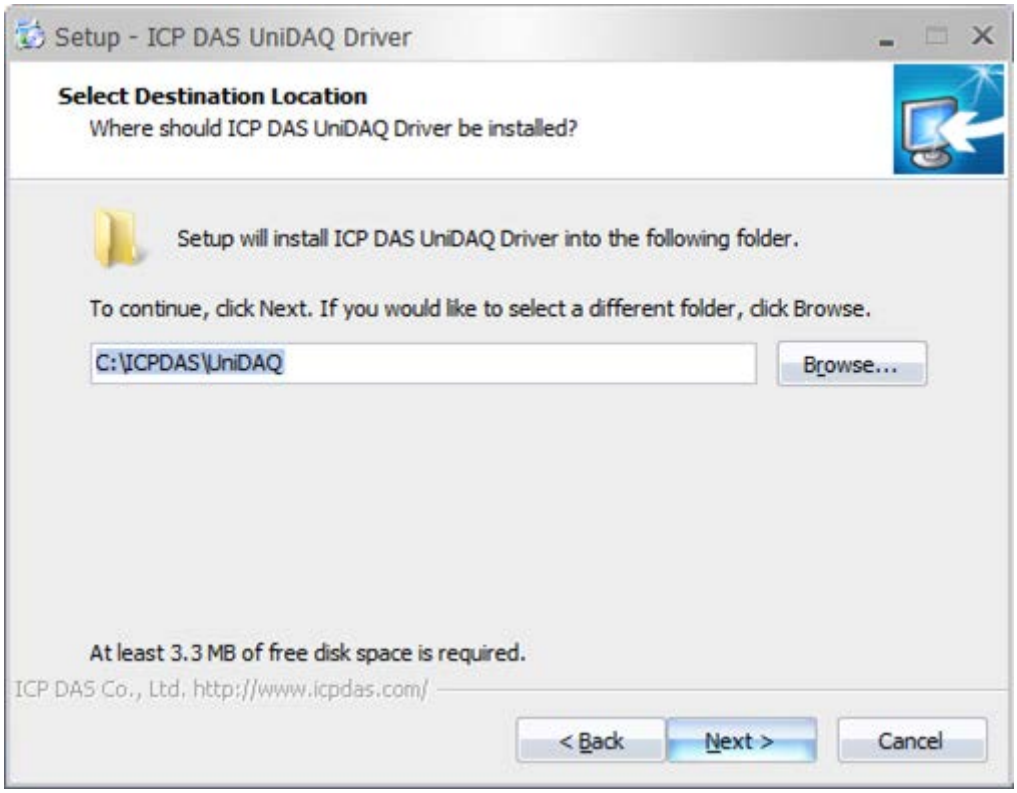


7. Click Next>

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

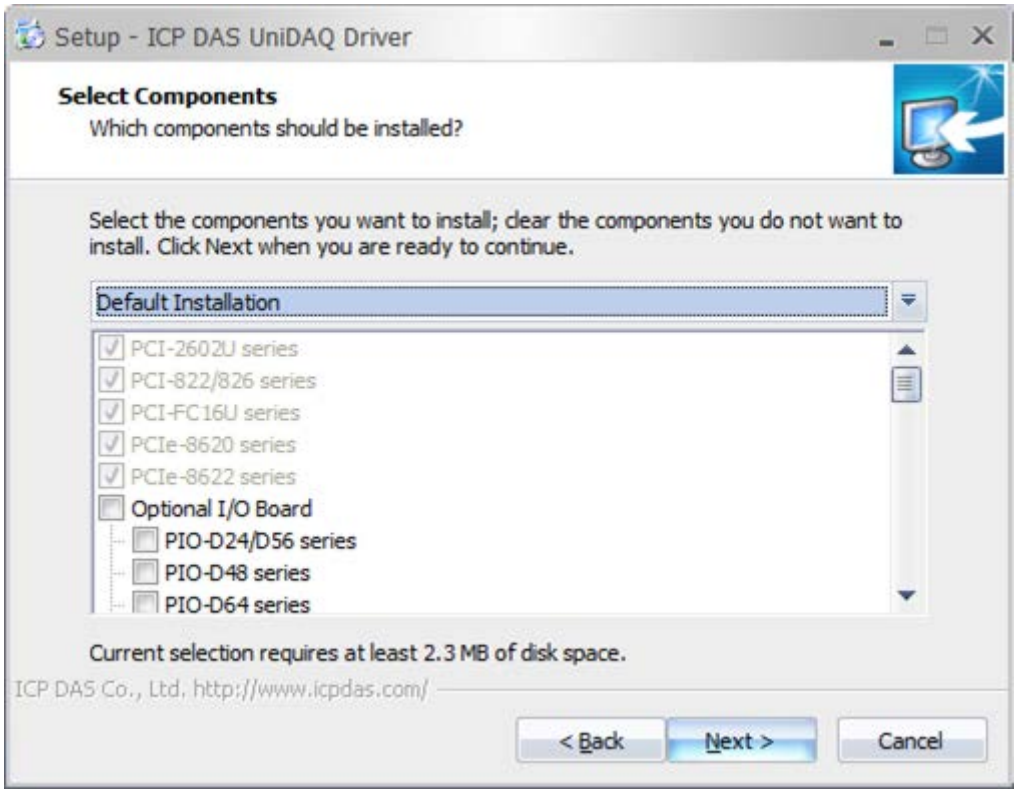


8. Click Next>

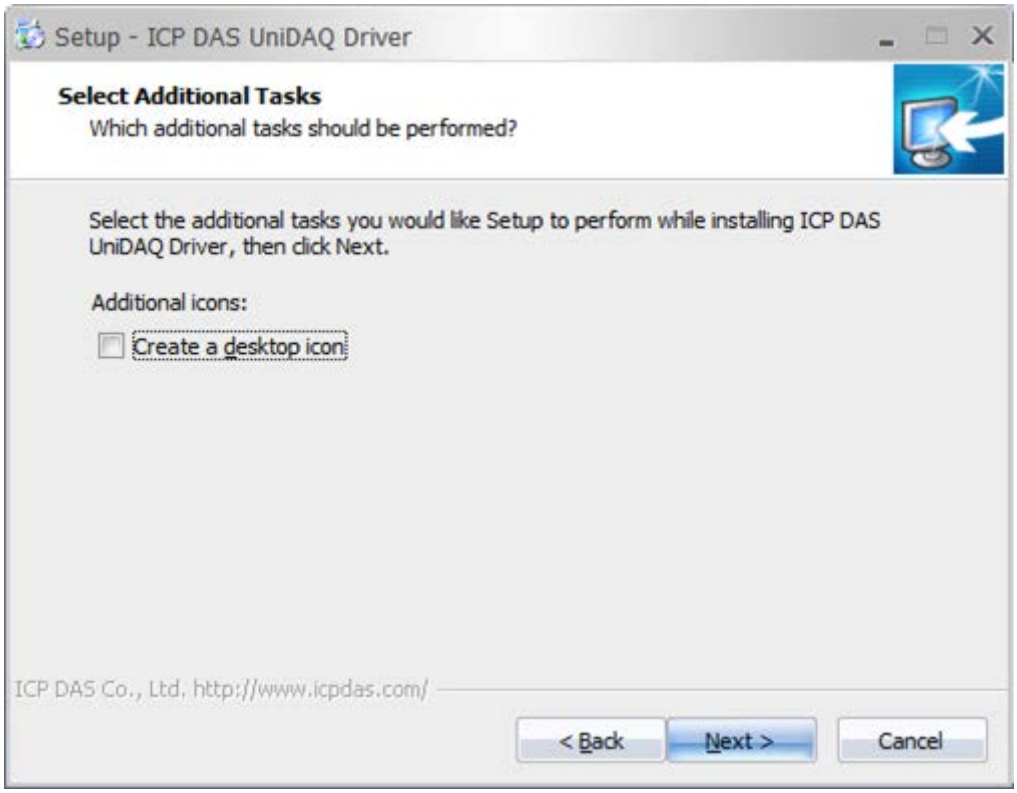


9. Click Next>

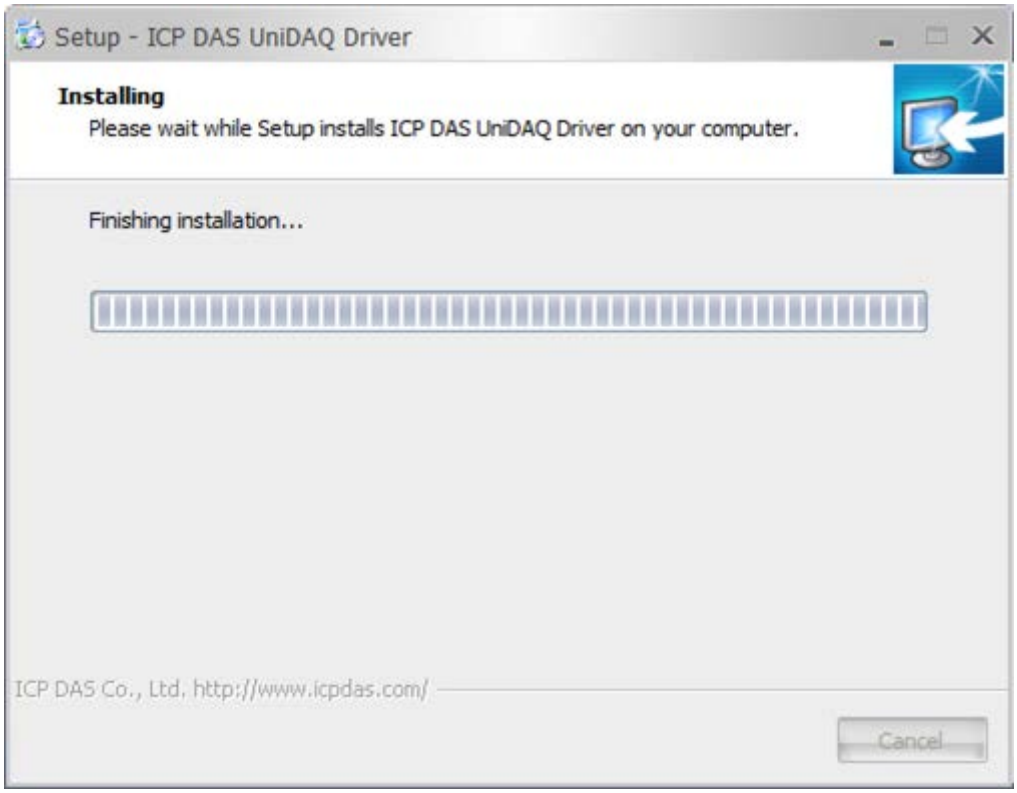
Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.



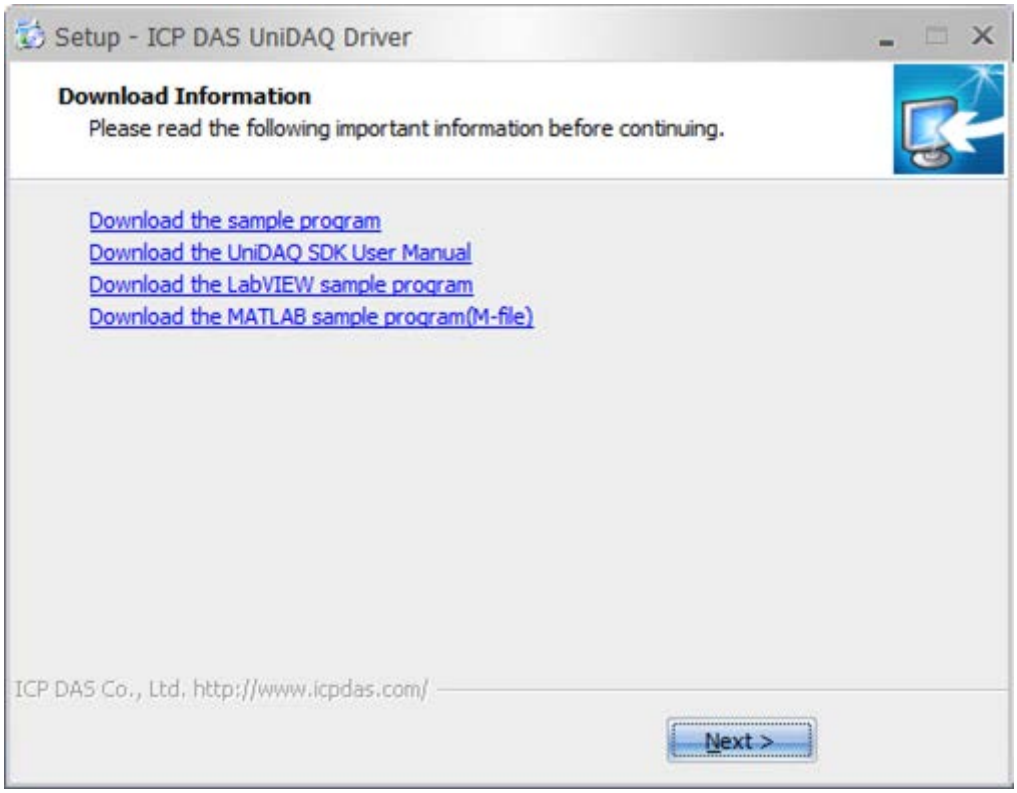
10. Click Next>



Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.



11. Click Next>

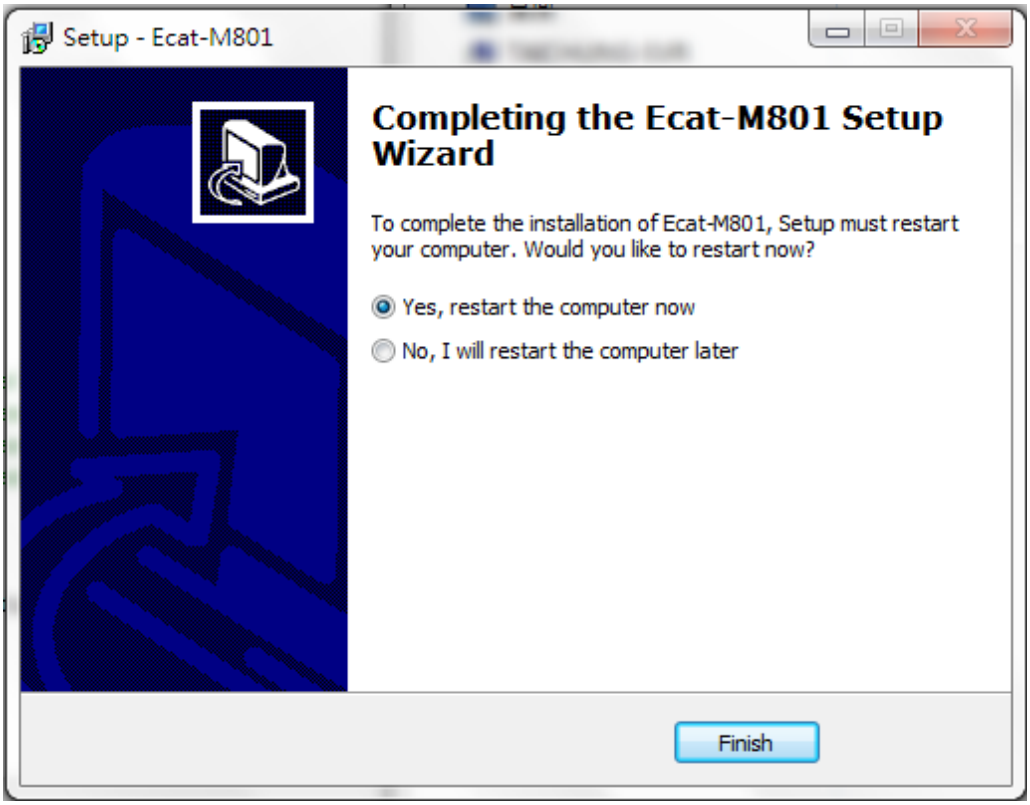


12. Click Finish

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.



13. Click the "Finish" button and restart the computer.

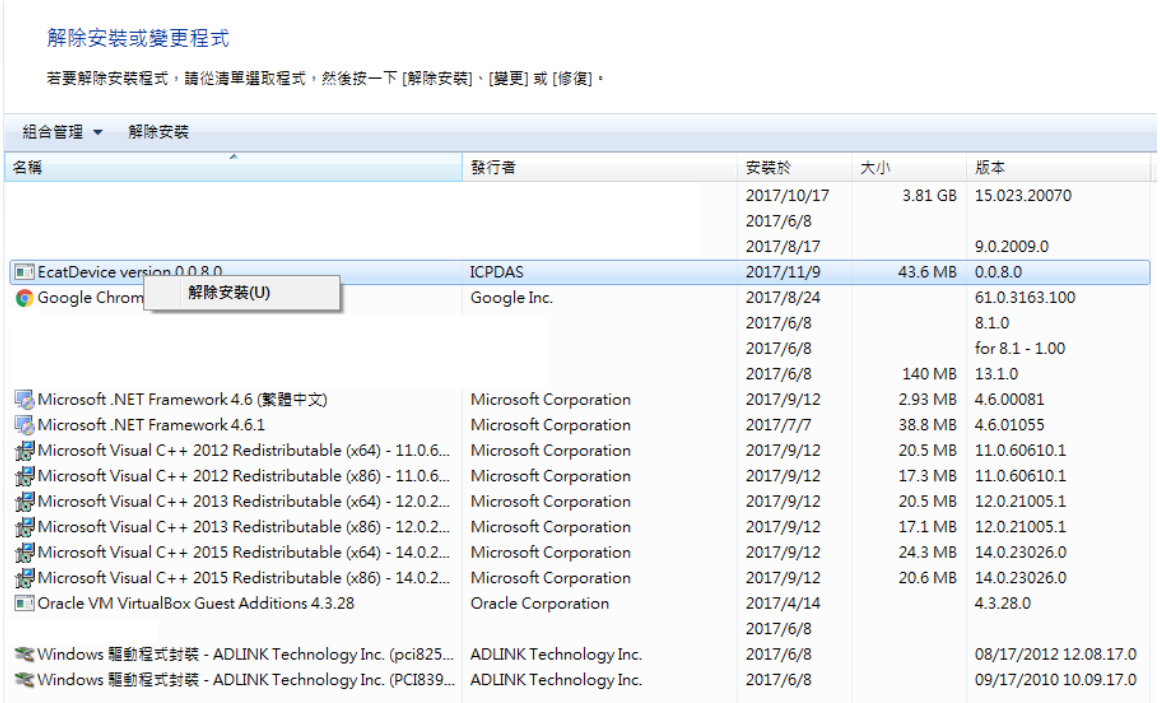


Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

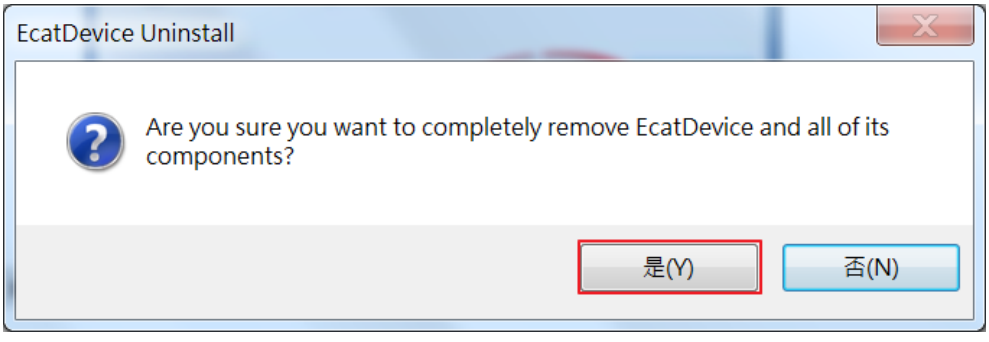
2.3. Uninstalling the Driver

ICPDAS driver includes an uninstall utility to help user 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.



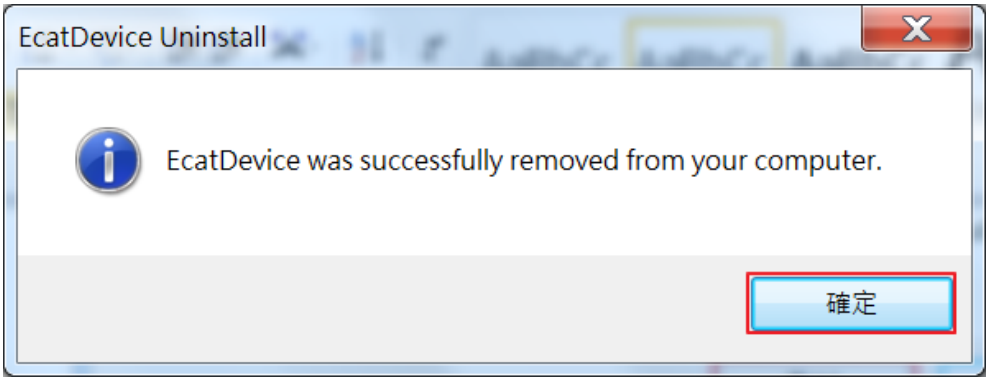
3. When the message box loads, click the Yes(Y) button to uninstall the software.



4. After the uninstallation process is complete, a dialog box will be displayed to you that

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

the driver was successfully removed. Click the “OK” button to finish the uninstallation process.

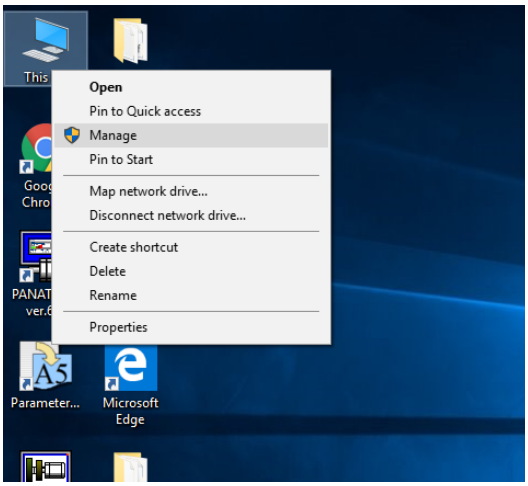


Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

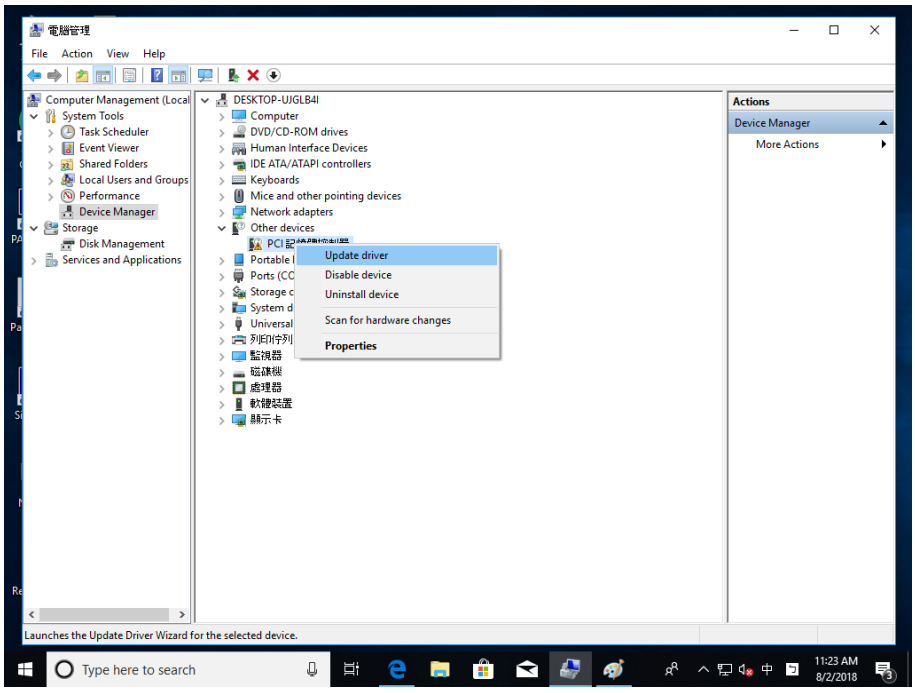
2.4. Driver Installing Manually

2.4.1. Step1

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

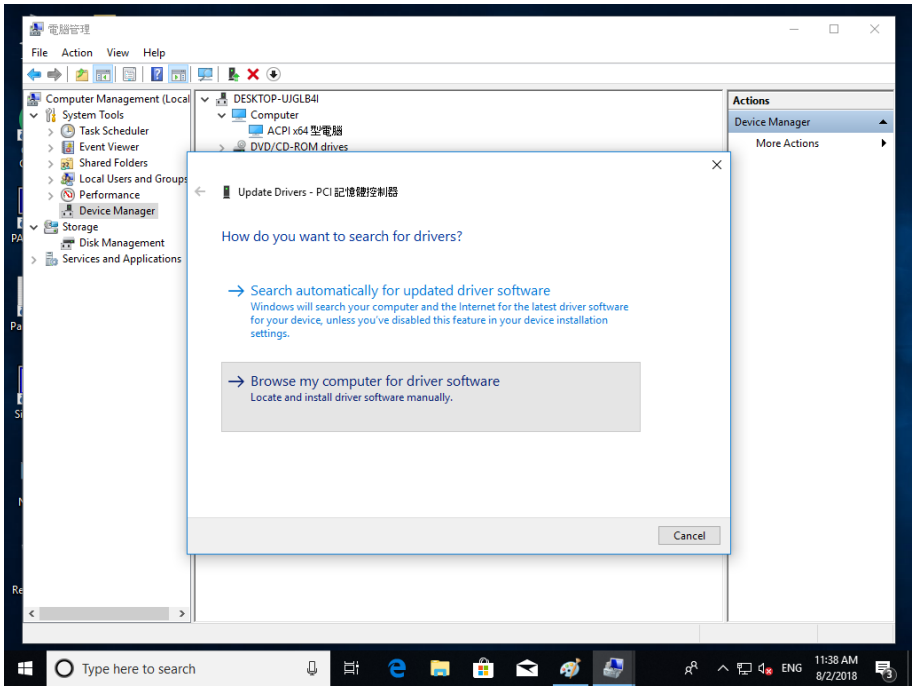


2. In the Computer Management window, on the left, click Device Manager.
3. Right-click the device, and select Update Driver.

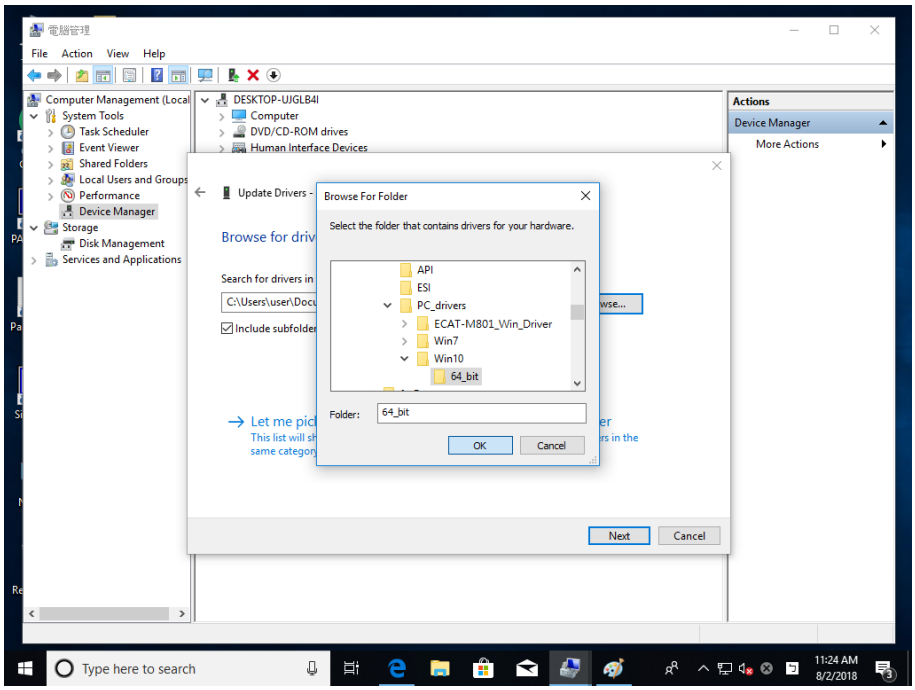


Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

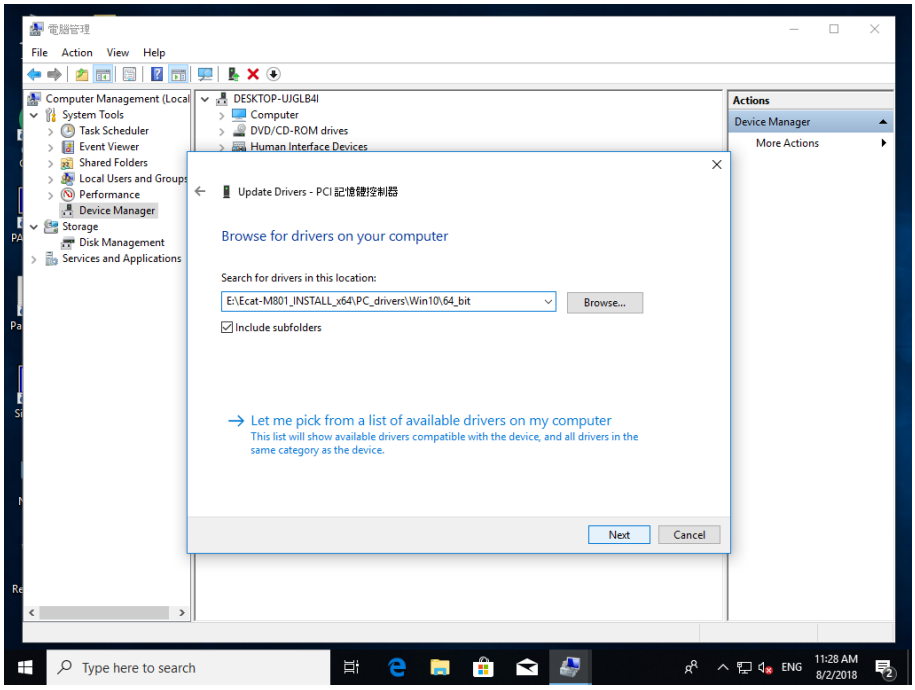


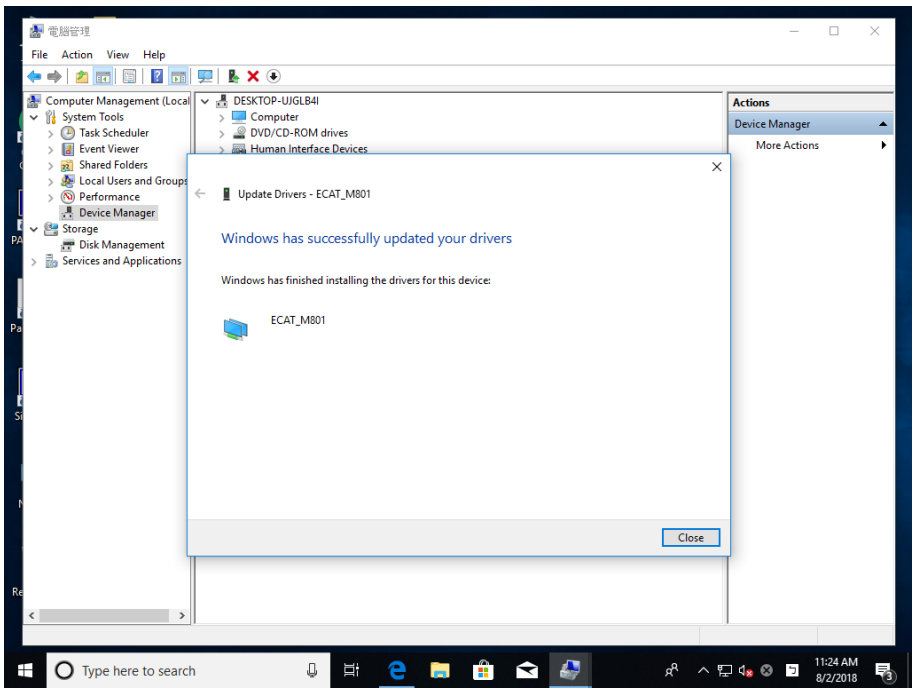
5. Click Browse and navigate to the folder that named 「Ecat-M801_INSTALL_x64\PC_drivers\Win10\64bit」. Click OK when this folder is selected.



Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

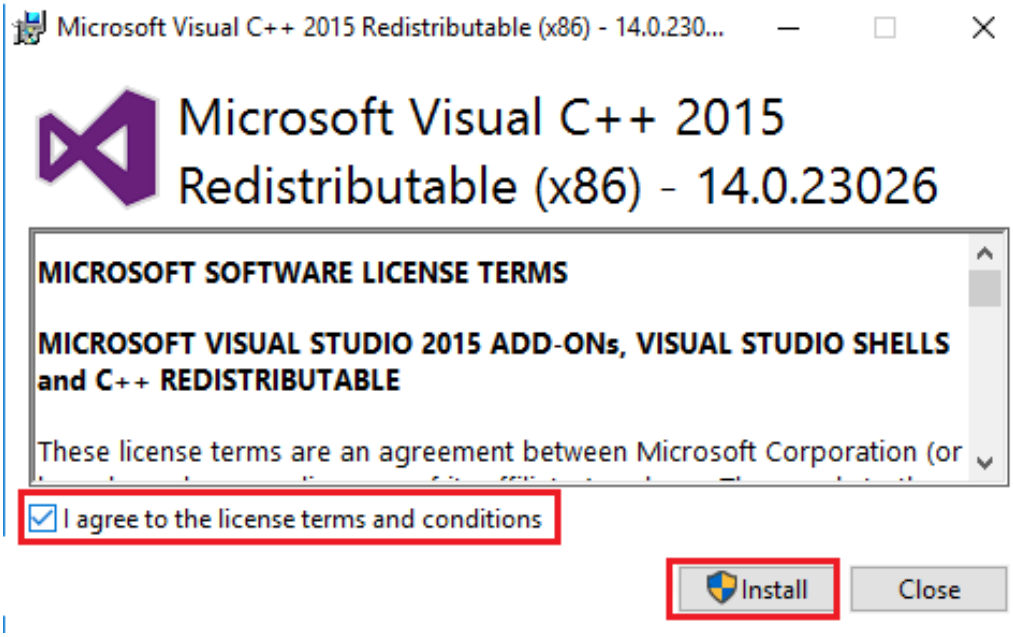




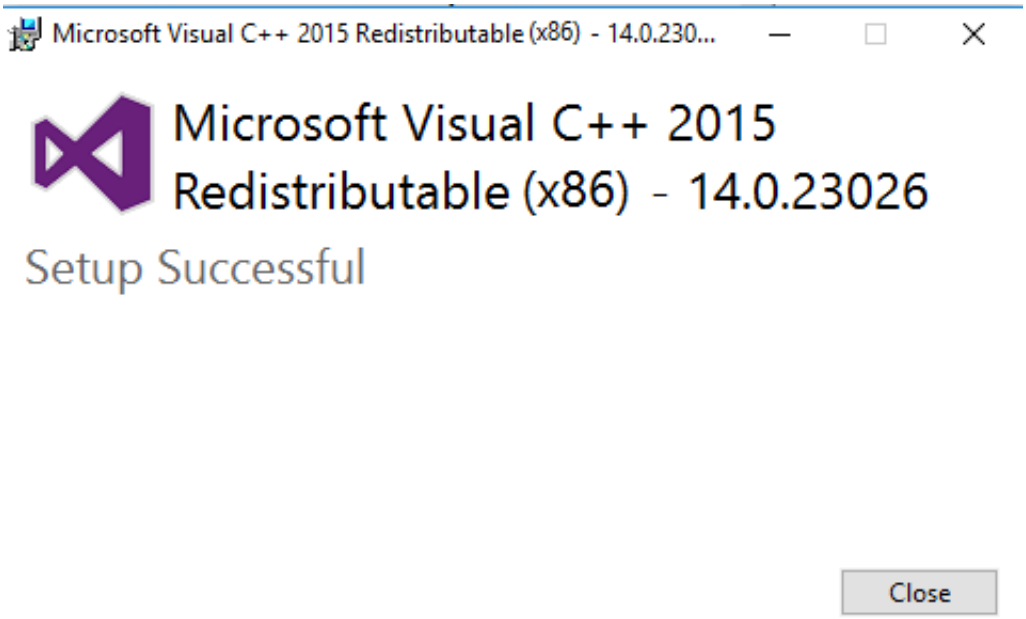
Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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.



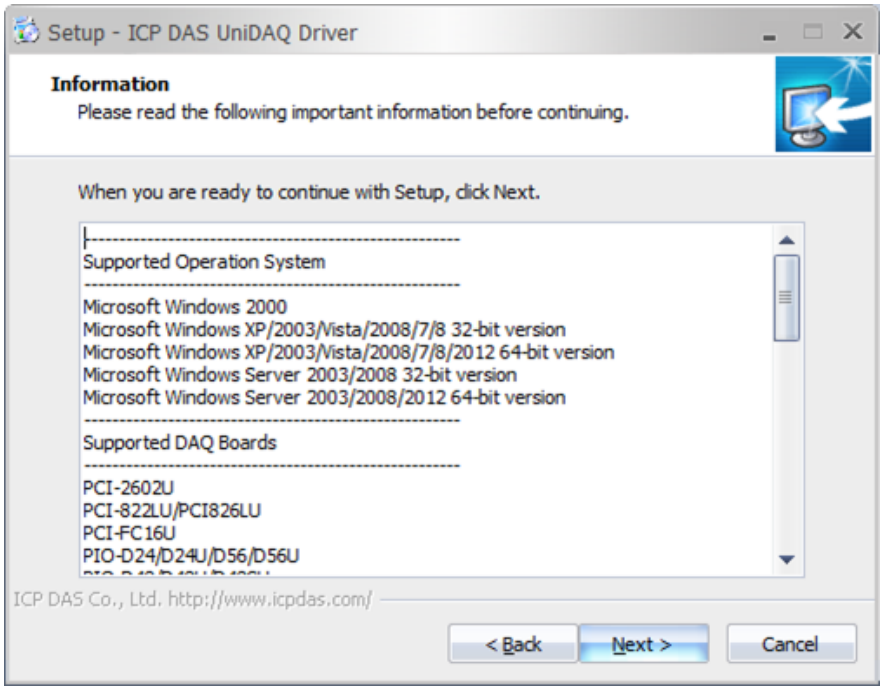
Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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>

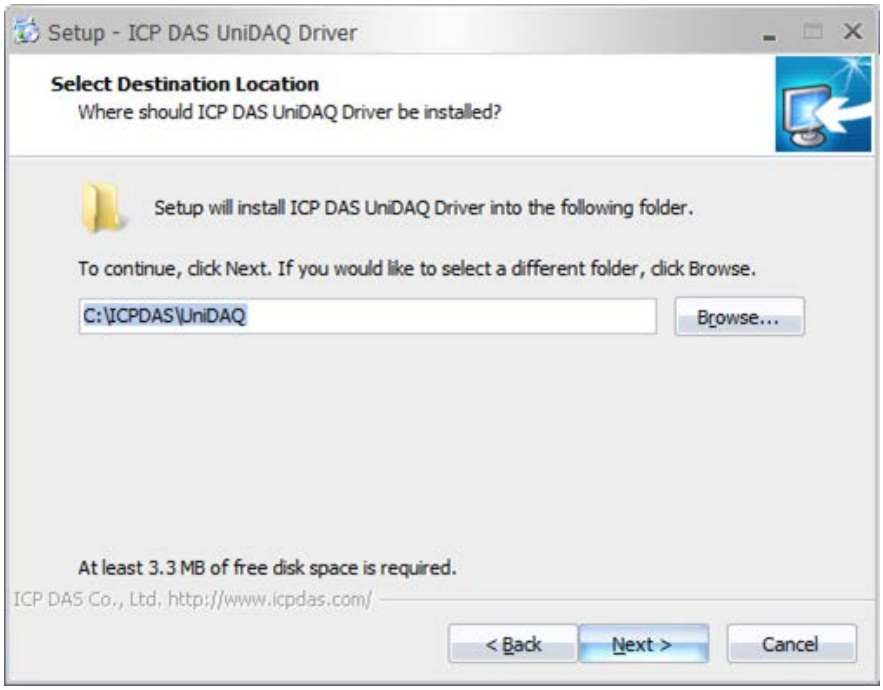


2. Click Next>

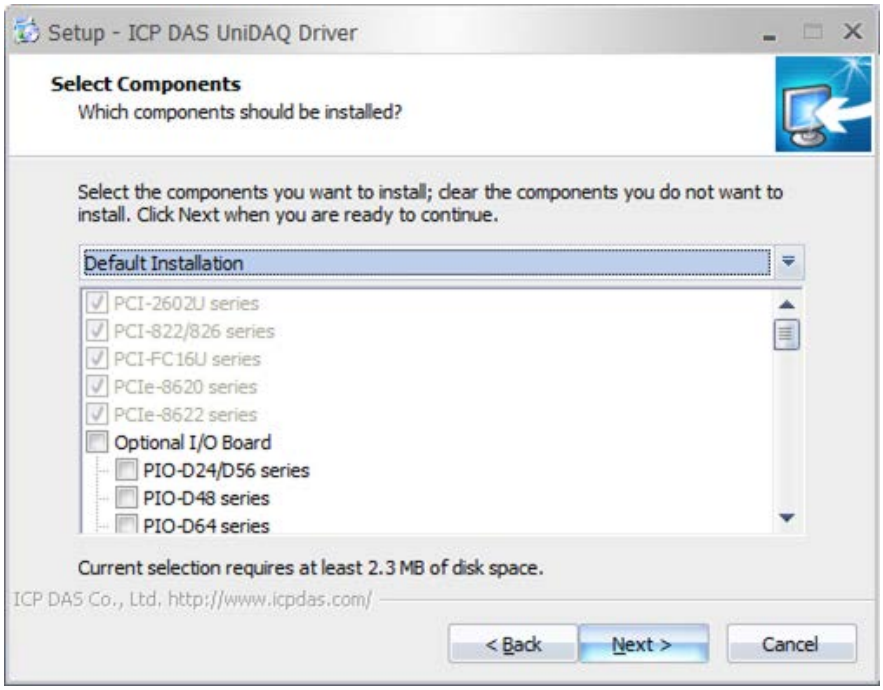


3. Click Next>

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

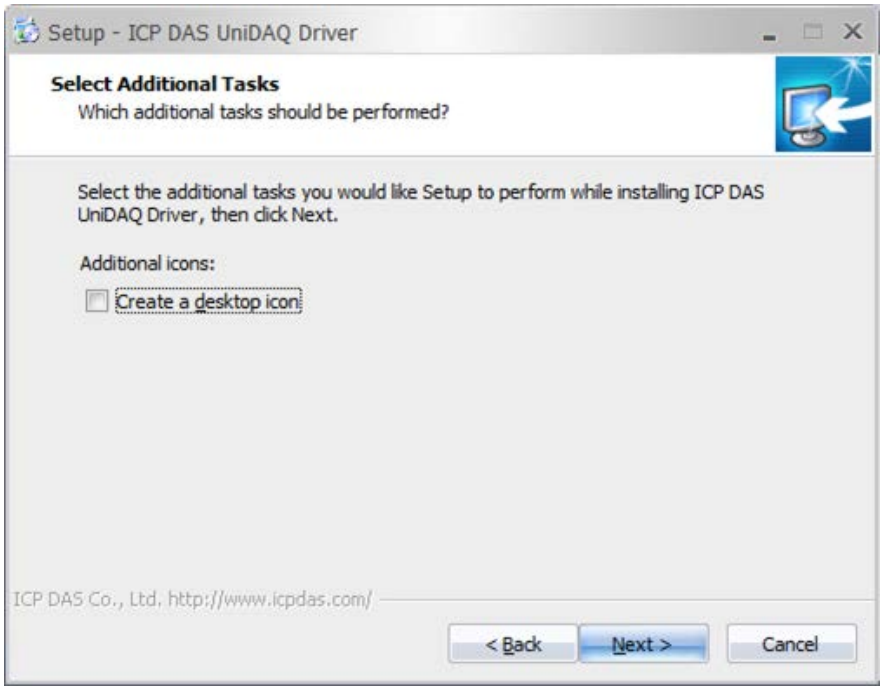


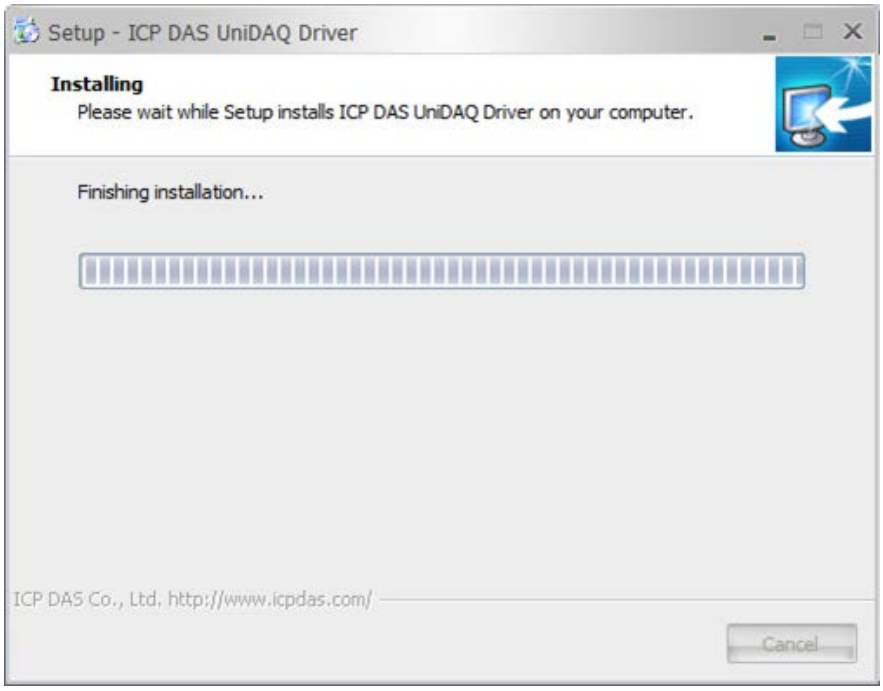
4. Click Next>



Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5. Click Next>





Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

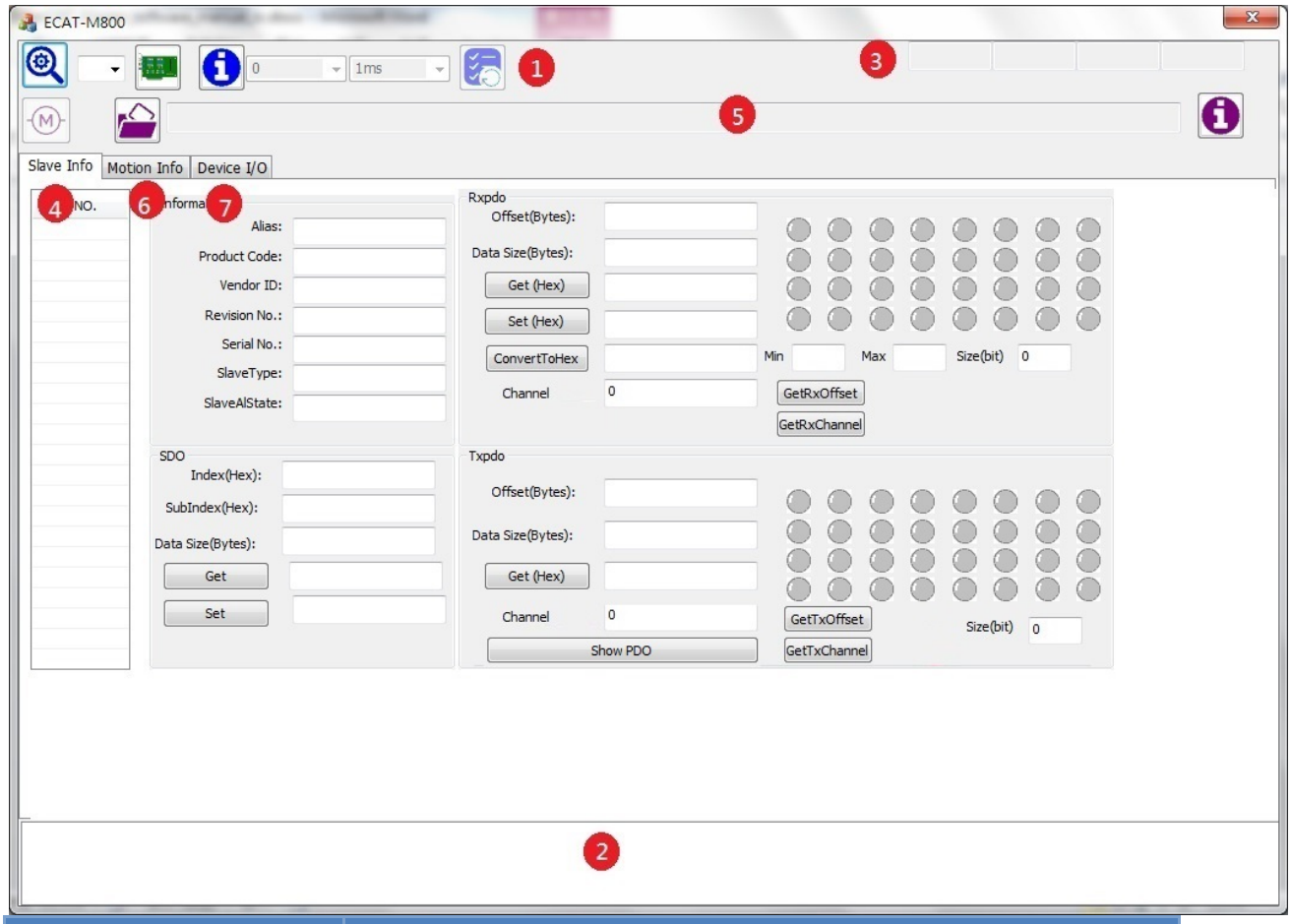
6. Click Finish, Driver Install Finish.



Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

3. EcatUtility

EcatUtility is a software tool provided by ICP DAS for users to edit device network information, test slave modules and do motion control function tests. The 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 initialization of Motion Control
(6)	Motion control page
(7)	Device I/O operation page

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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 the number of available devices
(2)	Open/Close device
(3) 0	Device number list
(4)	Network information edit page
(5) 0	Network information number list
(6) 1000000	Cycle time list
(7)	Start/Stop the device EtherCAT operation task

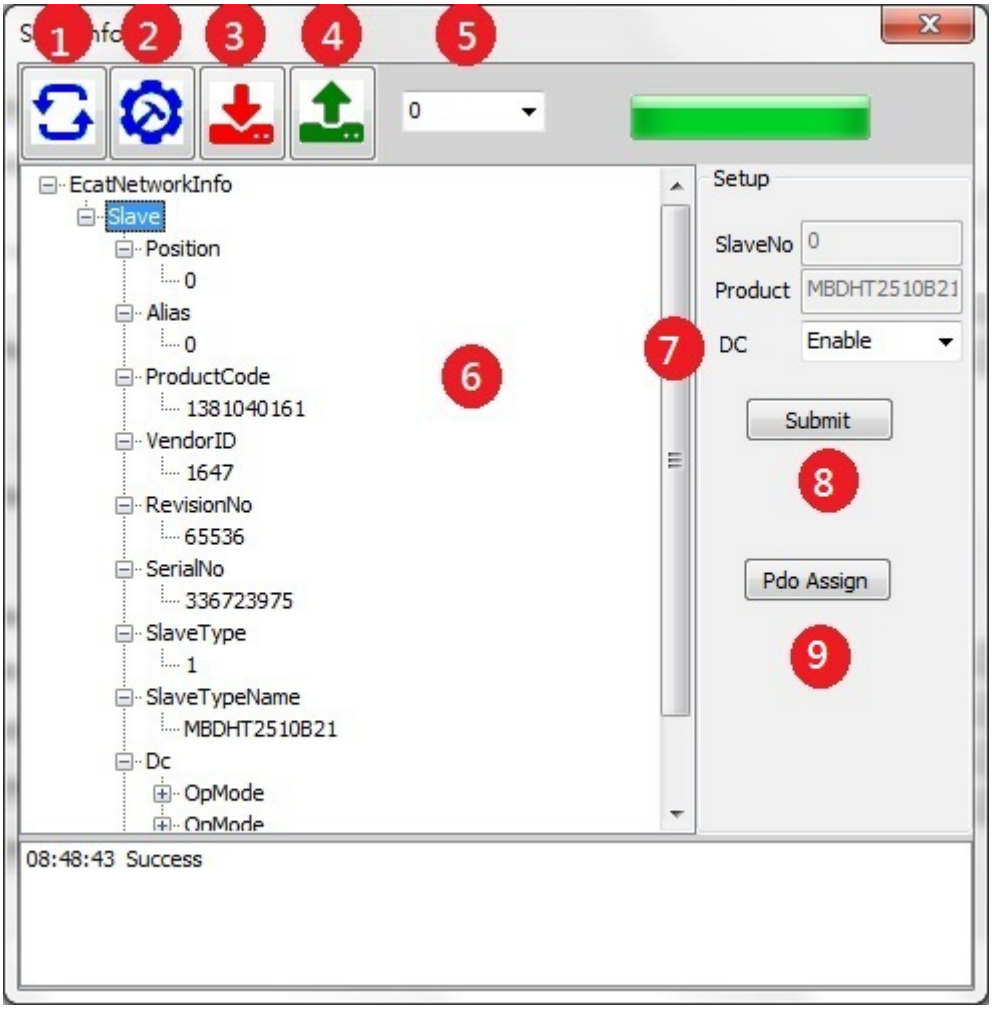
3.1.1. Device Initialization Steps

1. Click to get the number of devices (i.e. control cards).
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.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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.



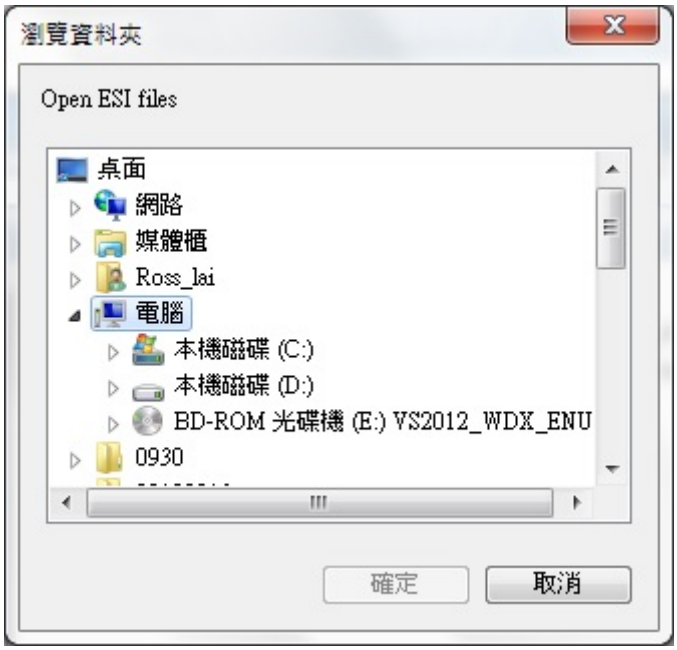
Item	Description
(1)	Scan EtherCAT network for finding slaves
(2)	Build network information from ESI files
(3)	Send the network information into the device (card)
(4)	Retrieve network information from the device (card)
(5) 0	The file number (file name) of network information for sending and retrieving.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

(6)	Network information panel
(7) DC Enable	DC setting of the selected slave
(8) Submit	Apply the configuration
(9) Pdo Assign	PDO Assign

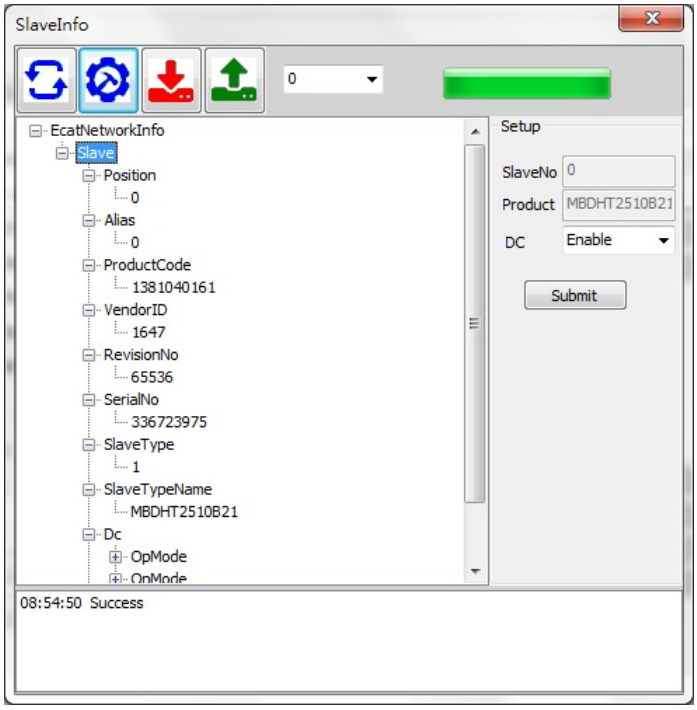
Descriptions:

1. Click to scan the connected slaves.
2. Click to select a ESI File Directory.



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

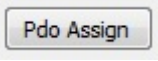
Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.



Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

3. Click 

to select a slave. User can select its DC setting on the right-hand side. Click 

to apply the setting.
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, 

provide user to assign Pdos.

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, wait for device network status to change the OP, then you can start the related EtherCAT operation.
4. If  is clicked again, it will stop the EtherCAT operation task.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

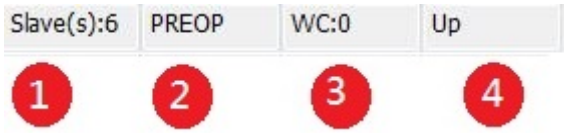
3.2. Message Panel

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 flollowing table.

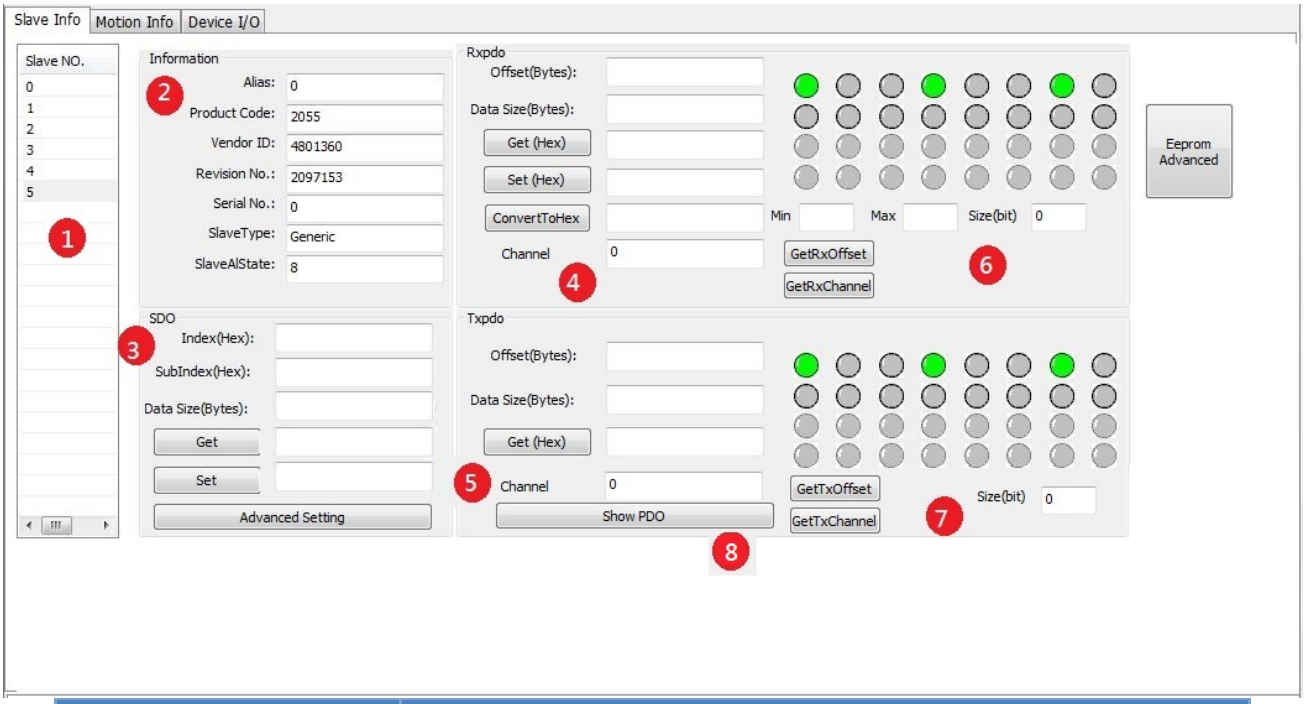


Item	Description
(1)	The total number of responding slaves
(2)	EtherCAT AL states of all slaves
(3)	EtherCAT working counter value
(4)	Network link status of EtherCAT (good wire connection or not) Down: Link Down Up: Link Up

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

3.4. Slave Operation Page

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

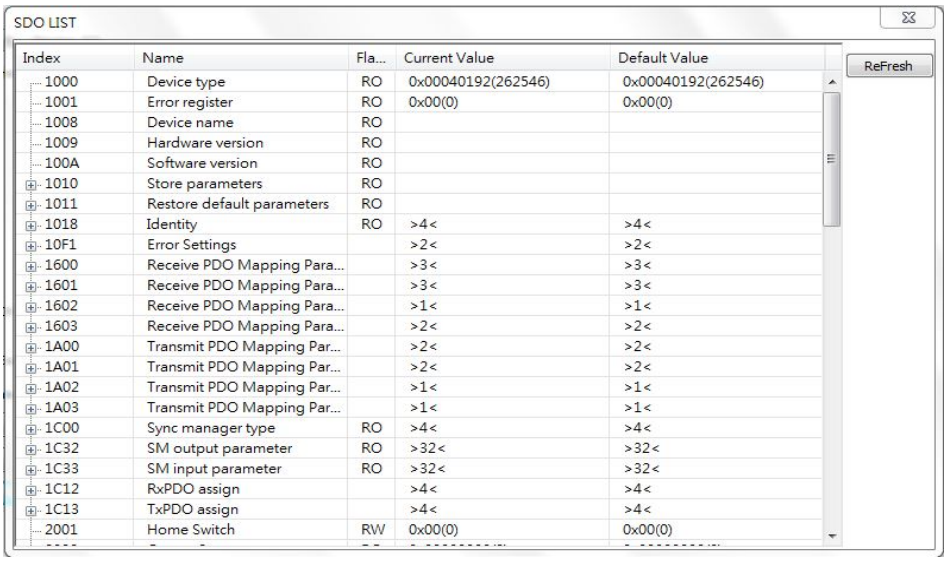


Item	Description
(1)	Slave list (all scanned slaves are listed here)
(2)	Selected Slave informations
(3)	SDO read/write for the selected slave
(4)	RxPDO read/write the output objects of the selected slave
(5)	TxPDO read input objects of the selected slave
(6)	DO control and status LED display of the selected slave
(7)	LED display for DI status of the selected slave
(8)	Show PDO

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

3.4.1. Basic Slave Operation Steps

1. After executing the device initialization steps, the user can select a slave from the slave list. 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.



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

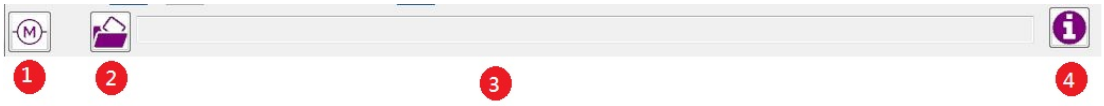
1. After the user completes the start EtherCAT operation task steps, he can access PDO by entering the Offset and Data Size in the PDO read/write group box and clicking "Get/Set" to read/write the slave PDO data. The data to be access are composed of bytes; and all the byte are separated by commas. For example, writing 2 bytes of data, 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 operations by further processing RxPDO and TxPDO data. Right now, this kind of functions only supports simple DI/DO slave modules.
3. If the module has multiple channels, you can use GetRxChannel or GetTxChannel to obtain the number and size of the channels.
4. If the module has multiple channels, enter channel and size, press GetRxOffset or GetTxOffset to get the offset of the channel.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5. ConvertToHex button function: Enter the value you want to convert and MIN, MAX, and Size (bit) to convert the value.
   EX: The value to be converted is: 10 MIN: 0 MAX: 10 Size (bit): 8, and the converted value is 7F.
6. Show PDO button function:Show RxPdo and TxPdo.

3.5. Motion Control Initialization Toolbar

Some parameters of axes must be initialized before starting motion control. The motion control initialization toolbar is shown below, and the description of each control item is shown in the following table.

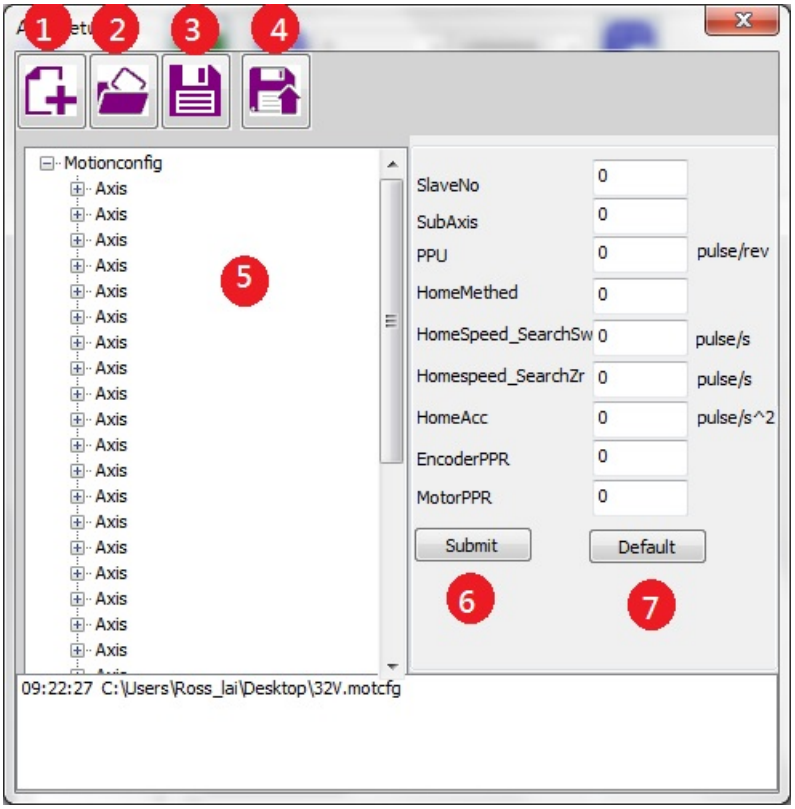


Item	Description
(1)	Start to initize axes for motion control
(2)	Open the file dialog for selecting a parameter file
(3)	File path of the motion control parameter
(4)	Open the edit page of motion control parameter file

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

3.5.1. Motion Control Parameter File Editing Steps

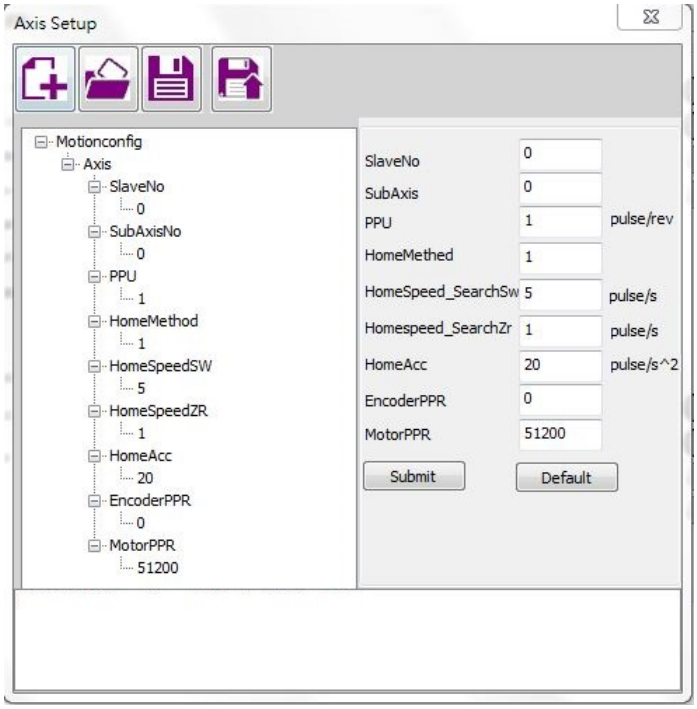
As shown below, click on the Motion Control Initialization Toolbar will open the Motion Control Parameter Edit page. The description of each control item is shown in the following table.



Item	Description
(1)	Create a new parameter file
(2)	Open a existed parameter file
(3)	Save to the current parameter file
(4)	Save as an new parameter file
(5)	Parameter information panel
(6)	Submit (apply) selected axis settings
(7)	Get the default values of the selected axis

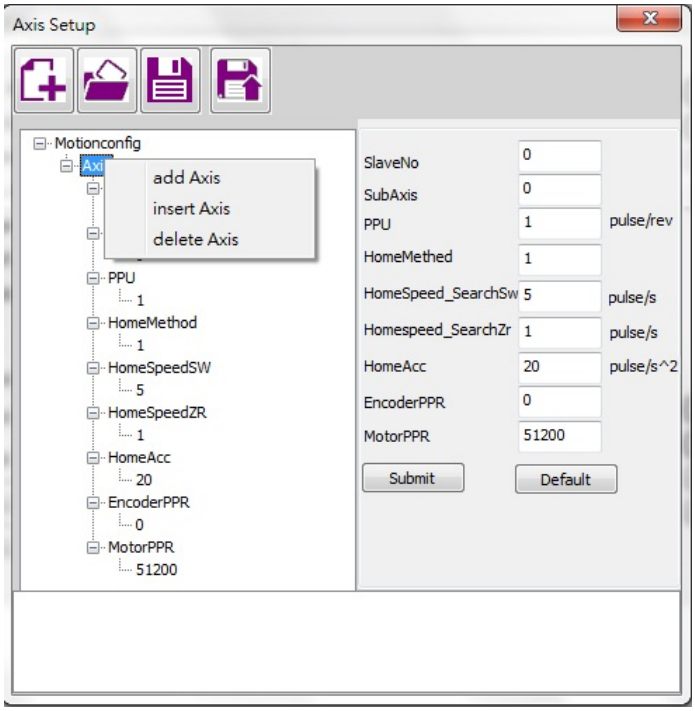
Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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. Then set values of PPU, HomeMethod, HomeSpeed_SearchSw, HomeSpeed_SearchZr, HomeAcc, EncoderPPR, MotorPPR parameters.
3. Choose an axis node. Click the right mouse button to display a small menu. Choose ″add Axis″ to append a set of axis parameters after the last node. Click ″insert Axis″ to add a set of axis parameters after the current node. Click ″delete Axis″ to delete the selected node.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.



4. After editing, click to save the file.
5. Click to save as a new parameter file.

3.5.2. Motion Control Initialization Steps

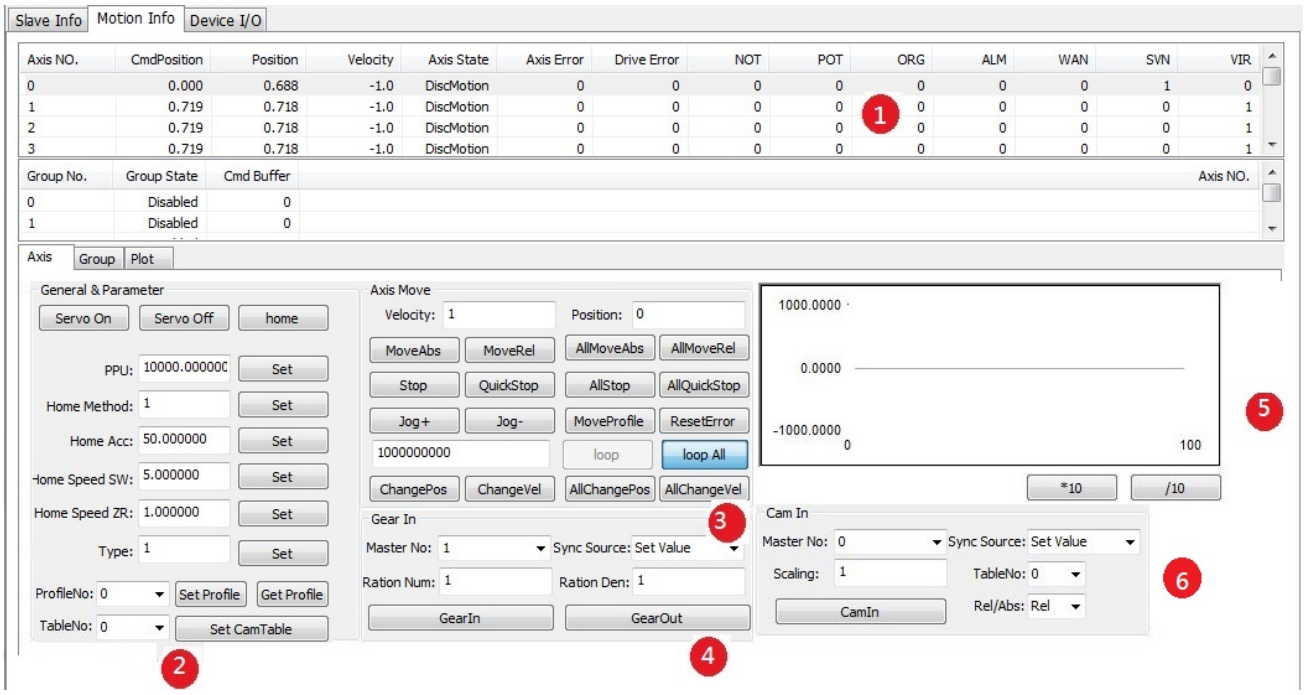
1. After the user has completed the motion control parameter file editing step, click on the motion control initialization toolbar to open the edited parameter file.
2. Click will use that parameter file for initialization.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

3.6. Motion Control Page

After the initialization of the motion control, the user can start the motion control operation. 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 Informations
(2)	Single-axis parameter settings
(3)	Single-axis motion control functions
(4)	Gear function settings and testings
(5)	Single-axis Position and Velocity
(6)	E-Cam function settings and testings

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

Single-axis parameter settings

1. Choose an axis number in the single-axis motion information panel.
2. Click the ″Servo ON/OFF″ button to enable or disable the motion. 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.

Single-axis motion control functions

1. Select the specified axis number in the single-axis motion information panel.
2. Set ″velocity″ and ″position″ parameters.
3. Click the ″MoveAbs″ or ″MoveRel″ button to start single-axis motion control.
4. To stop in single-axis motion control, click ″Stop″ or ″QuickStop″ to stop single-axis motion control.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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 never ending movement with a specified velocity. Click "Stop" or "QuickStop" to stop the motion.
7. Set the value first. When loop or loop All is clicked, the axis moves back and forth between the current position and the set position of the axis.

Gear function settings and testing

1. Select an axis number in the single-axis motion information panel.
2. Next, 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 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".
5. Next, click the "GearIn" button to start the gear motion. The state of the slave axis will change to SyncMotion. After that, the slave axis will follow any motion of the master axis with a 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.

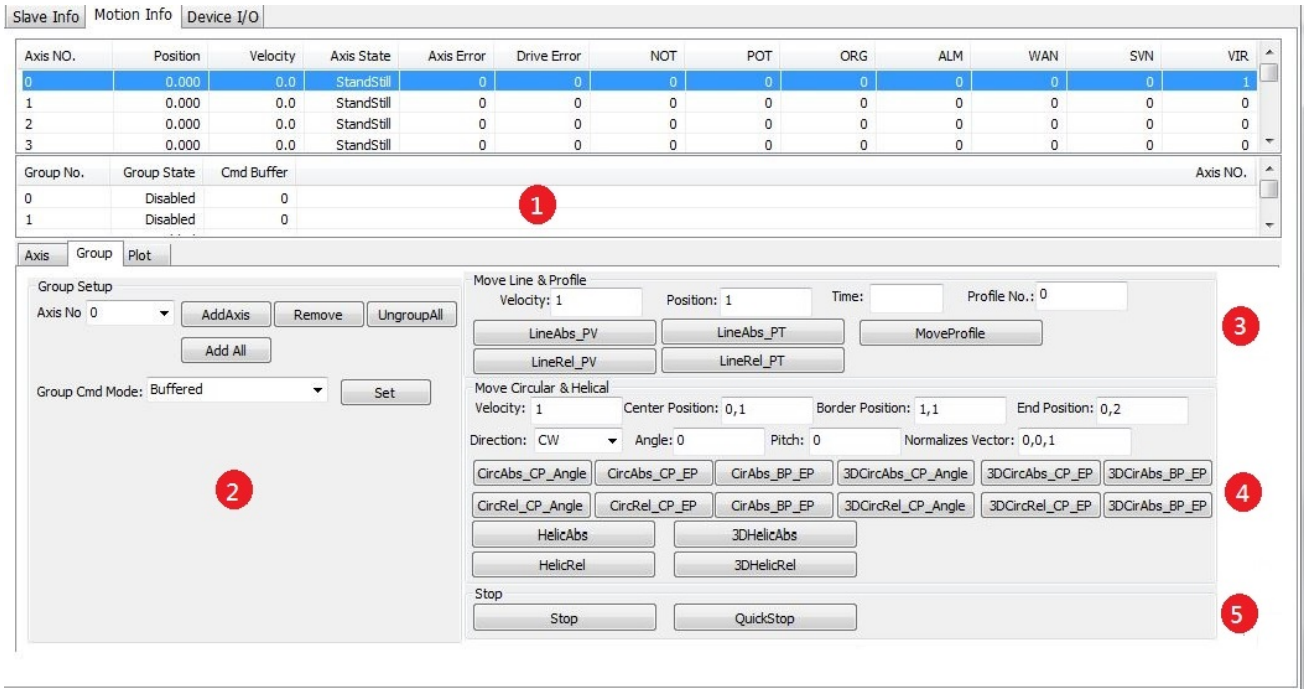
Gear function settings and testing

1. Select an axis number in the single-axis motion information panel.
2. Next, Set CamTable.
3. Next, Set its "Master No" parameter. This master axis will be the reference axis.
4. Next, set the slave reference source. The reference source can be the master's command set value or the master's actual position value.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5. Next, Set the Scaling,TableNo and Rel/Abs.
6. Next, click the "CamIn" button to start the ecam motion. The state of the slave axis will change to SyncMotion. After that, the slave axis will follow any motion of the master axis with CamTable before.
7. 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 informations 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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Group motion information panel

1. Click "Group" to display the group motion information panel.

Item	Description
Group No.	Group number
GroupState	Group state
Cmd Buffer	Number of commands still in the command buffer for future executing
Axis No.	All axis numbers of this group

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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 Absolute 2D circular interpolation motion by center position and angle.
CircRel_CP_Angle	Start group Relative 2D circular interpolation motion by center position and angle.
CircAbs_CP_EP	Start group absolute 2D circular interpolation motion by center position and end position.
CircRel _CP_EP	Start group Relative 2D circular interpolation motion by center position and end position.
CircAbs_BP_EP	Start group relative 2D circular interpolation motion by border position and end position.
CircRel _BP_EP	Start group Relative 2D circular interpolation motion by border position and end position.
3D CircAbs_CP_Angle	Start group Absolute 3D circular interpolation motion by center position and angle.
3D CircRel _CP_Angle	Start group Relative 3D circular interpolation motion by center position and angle.
3D CircAbs_CP_EP	Start group absolute 3D circular interpolation motion by center position and end position.
3D CircRel _CP_EP	Start group Relative 3D circular interpolation motion by center position and end position.
3D CircAbs_BP_EP	Start group relative 3D circular interpolation motion by border position and end position.
3D CircRel _BP_EP	Start group Relative 3D circular interpolation motion by border position and end position.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

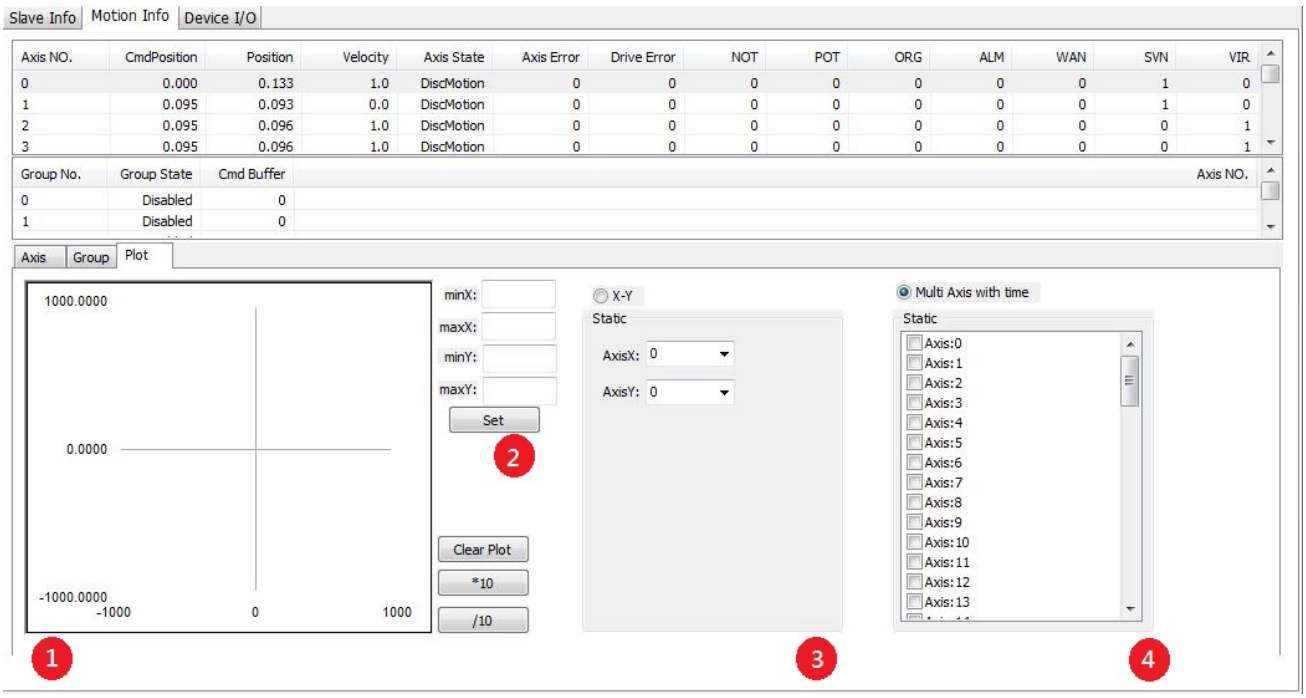
Group helical motion control

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

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

3.6.3. Group Show Position Page



Item	Description
(1)	Motion position display area
(2)	Motion position display area settings
(3)	X-Y display mode
(4)	multiple axis display mode

Motion position display area settings

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

X-Y display mode

1. Set the X axis and Y axis to observe the 2D position change.

multiple axis display mode

1. Check axes you want to observe ,then you can observe position of these axes

3.7. Device I/O Operation Page

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



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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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 bit number.

3.7.2. Device DI control operation step

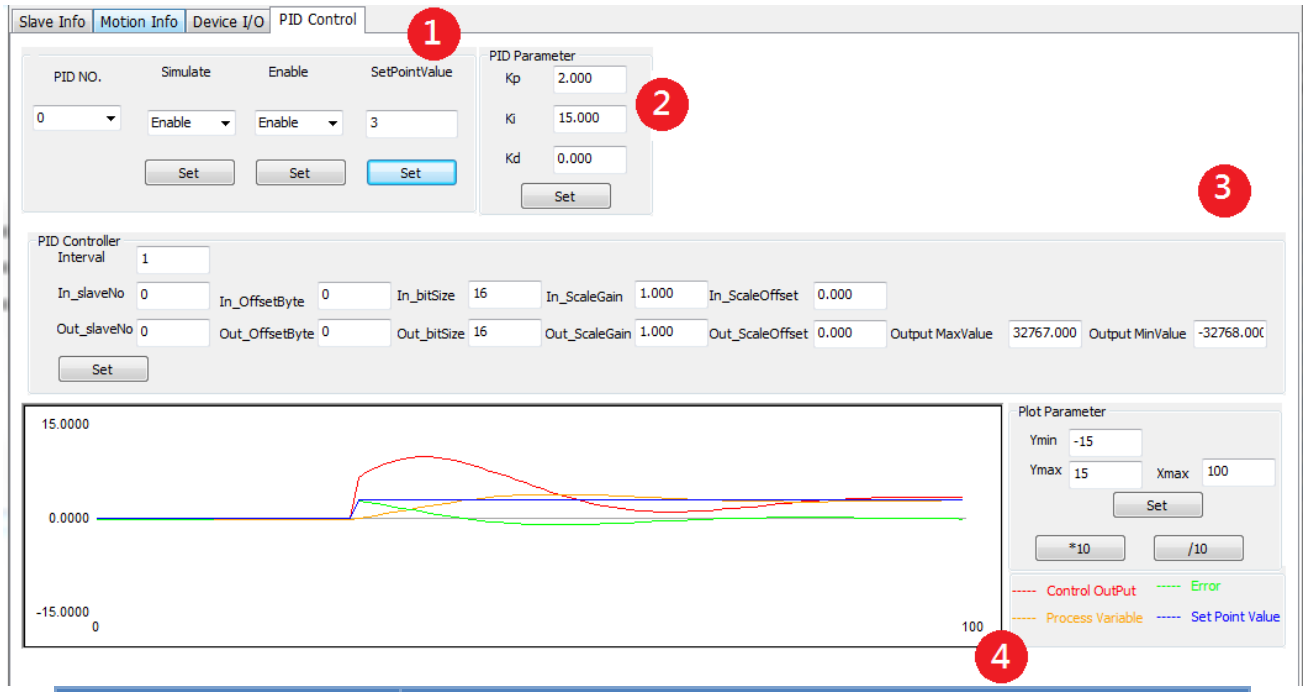
1. Click the "Get" button to get the DI data. If the "Bit" option is selected, enter the bit number in the edit box and click the "Get" button to get the bit value of the bit number.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

3.8. PID Control Page

After the user completes the start EtherCAT operation task steps,the user can operate Pid Controller.

3.8.1. PID Control Page



item	Description
(1)	PID Controller Status
(2)	PID Controller Parameter
(3)	PID Controller Input/Output
(4)	PID Controller Plot

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

PID Controller Status

item	Description
PID No.	PID Controller Number
Simulate	Activate simulate module or not
Enable	Activate PID Controller or not
SetPointValue	PID Controller Set Point

PID Controller Setting

1. Choose PID Controller Number.
2. Set PID Controller measure module and Output Module.
3. Set PID Parameter.
4. Set **Simulate** as Enable to activate simulate module,or set **Simulate** as disable to activate measure module and Output Module.
5. Set **Enable** as Enable to activate PID Controller .

PID Controller INPUT/OUTPUT

item	Description
Interval	PID Controller Interval，Unit:CycleTime
In_slaveNo	PID Controller measure module number
In_OffsetByte	PID Controller measure module TxPdoOffset
In_bitSize	PID Controller measure module DataSize，Unit:bit
In_ScaleGain	PID Controller measure module Scale gain
In_ScaleOffset	PID Controller measure module Scale offset
Out_slaveNo	PID Controller Output Module number
Out_OffsetByte	PID Controller Output Module RxPdo Offset
Out_bitSize	PID Controller Output Module DataSize，Unit:bit
Out_ScaleGain	PID Controller Output Module Output gain
Out_ScaleOffset	PID Controller Output Module Output offset
Output MaxValue	PID Controller Output Module Maximum Limitation of

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

	Control Output
Output MinValue	PID Controller Output Module 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 can be used. Then, according to the Card ID, call *OpenDevice* function to open that device. After device is opened, the EtherCAT communication does not start yet. At this moment, those basic device operation functions should be used to configure the communication.

First, the user can use *GetDeviceState* to get the current states of the EtherCAT network. The 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 theer is needs to read/write some SDO objects, *GetSlaveSdoObject* and *SetSlaveSdoObject* functions can be used for these purposes.

Before starting the operation task of EtherCAT, please use the utility program to create and edit EtherCAT network information file and write the those informations into the device. Next, execute *StartDeviceOpTask* function to start the EtherCAT operation task. After the device and all slaves enter into the OP state, the PDO data communication and motion control operations can be controlled. To stop the EtherCAT communication, *StopDeviceOpTask* function must be called. To close the device operation, use *CloseDevice* function to do it.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

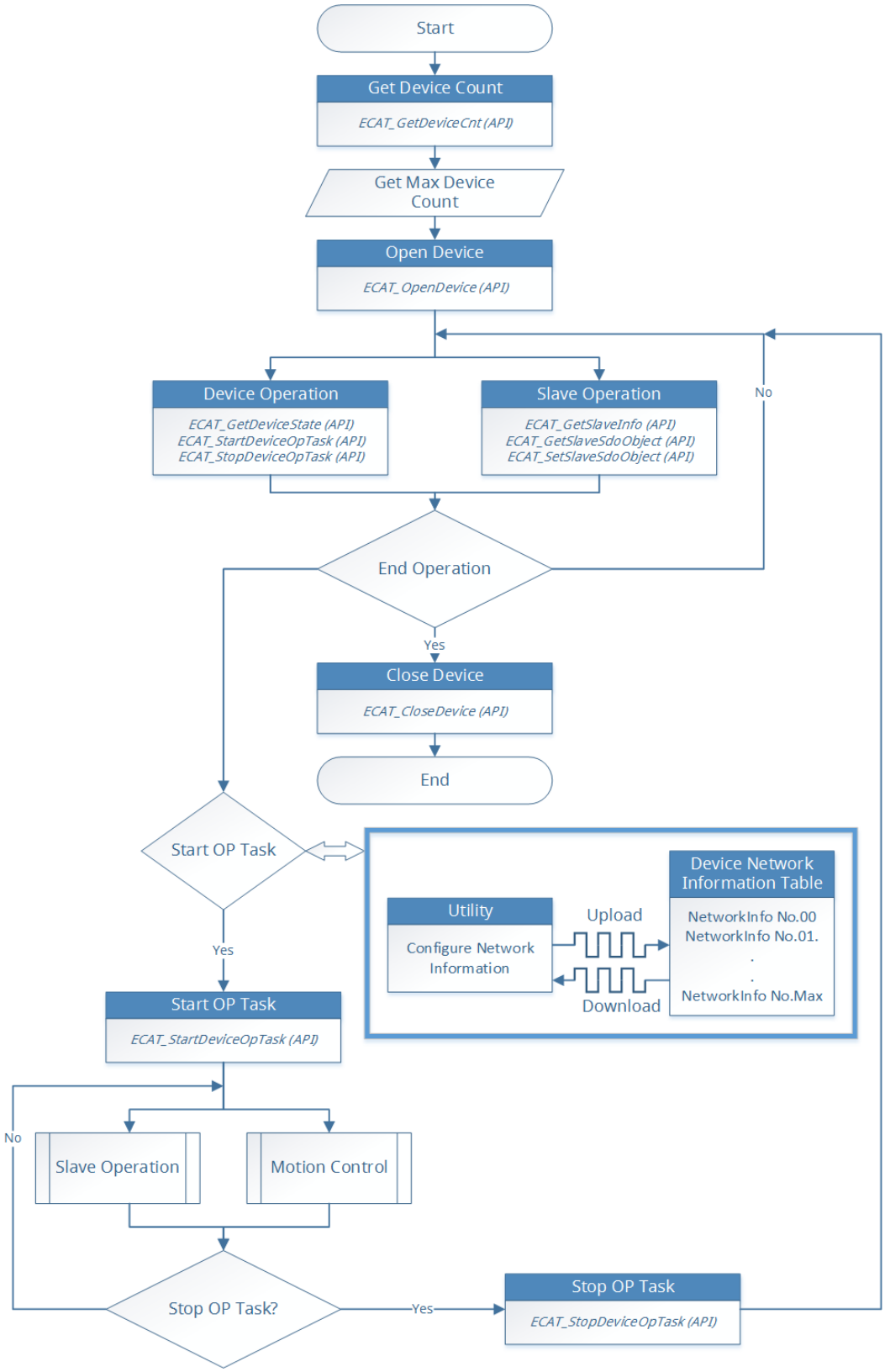


Figure 4-1

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

4.2. Slave Operation Flow

As show 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.

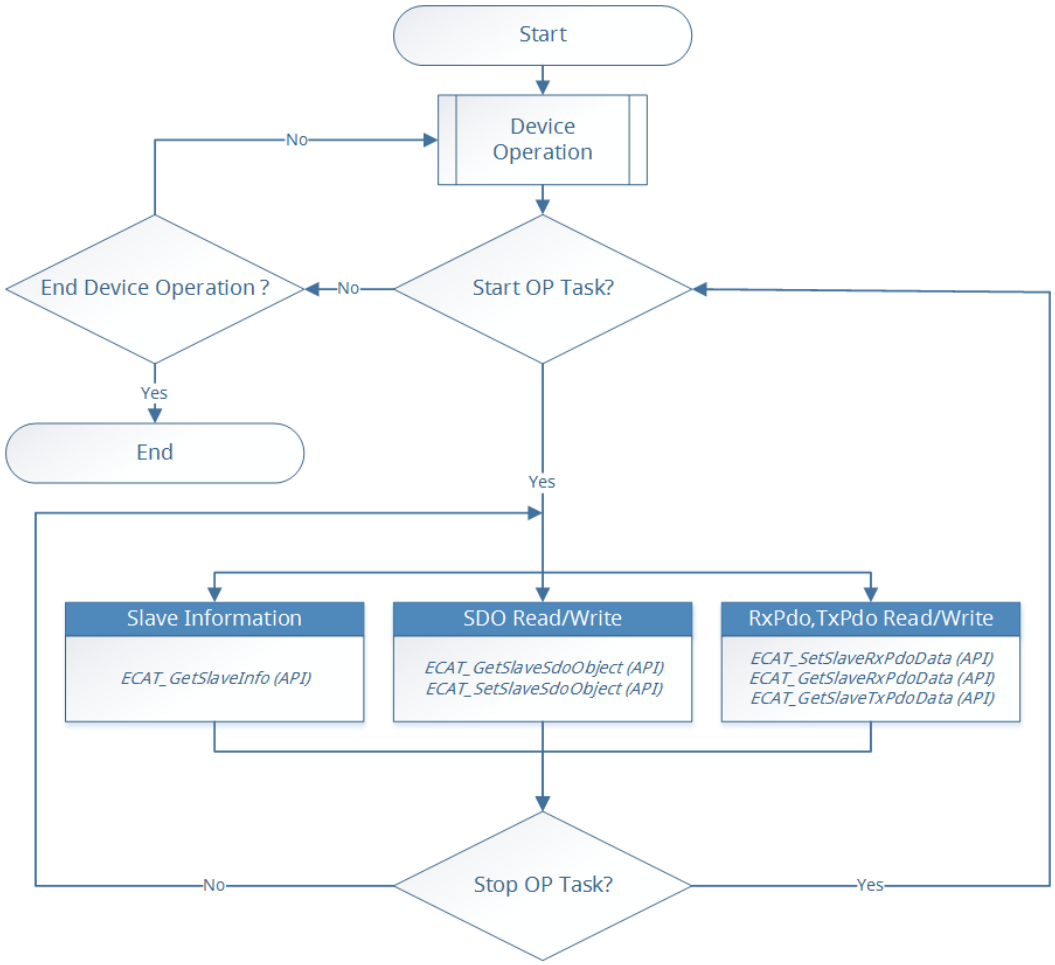


Figure 4-2

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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. After the initialization is successful, the user can start various motion related operations, such as axis homing, axis operation (single axis motion functions), axis error processing and group operation (multi-axis motion functions).

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

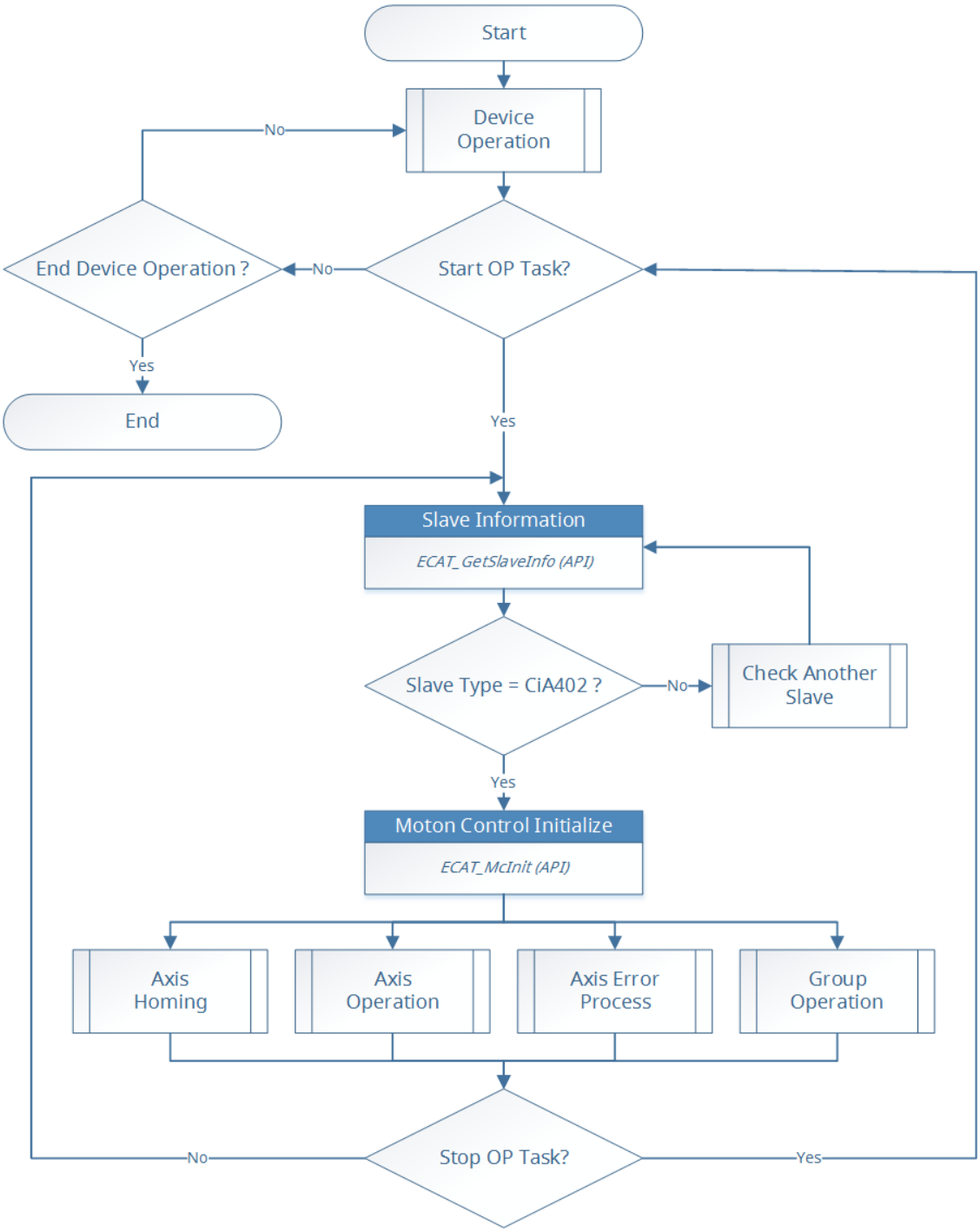


Figure 4-3

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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, that axis is currently stopped and ready to receive a new motion command. After successfully calling a motion related function, the axis state will change from the current Standstill state to Discrete Motion, Continuous Moiton, or Synchronized Moiton. In either of these three states, it indicates that axis is moving.

When an axis is moving, the user can call the stop function to stop its motion. 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. Then users need to deal with this error.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

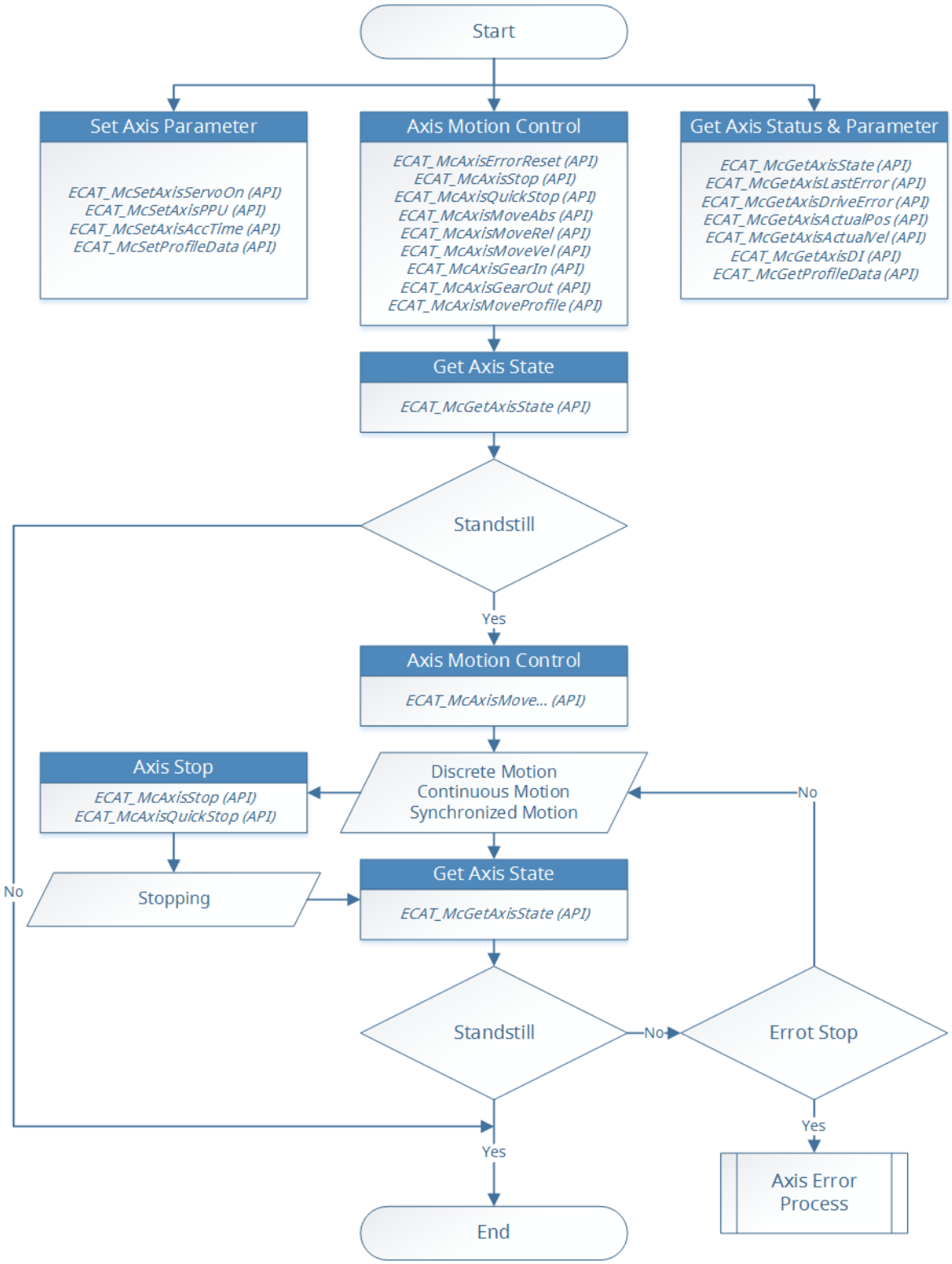


Figure 4-4

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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 its state from Homing to ErrorStop. The users need to deal with this error.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

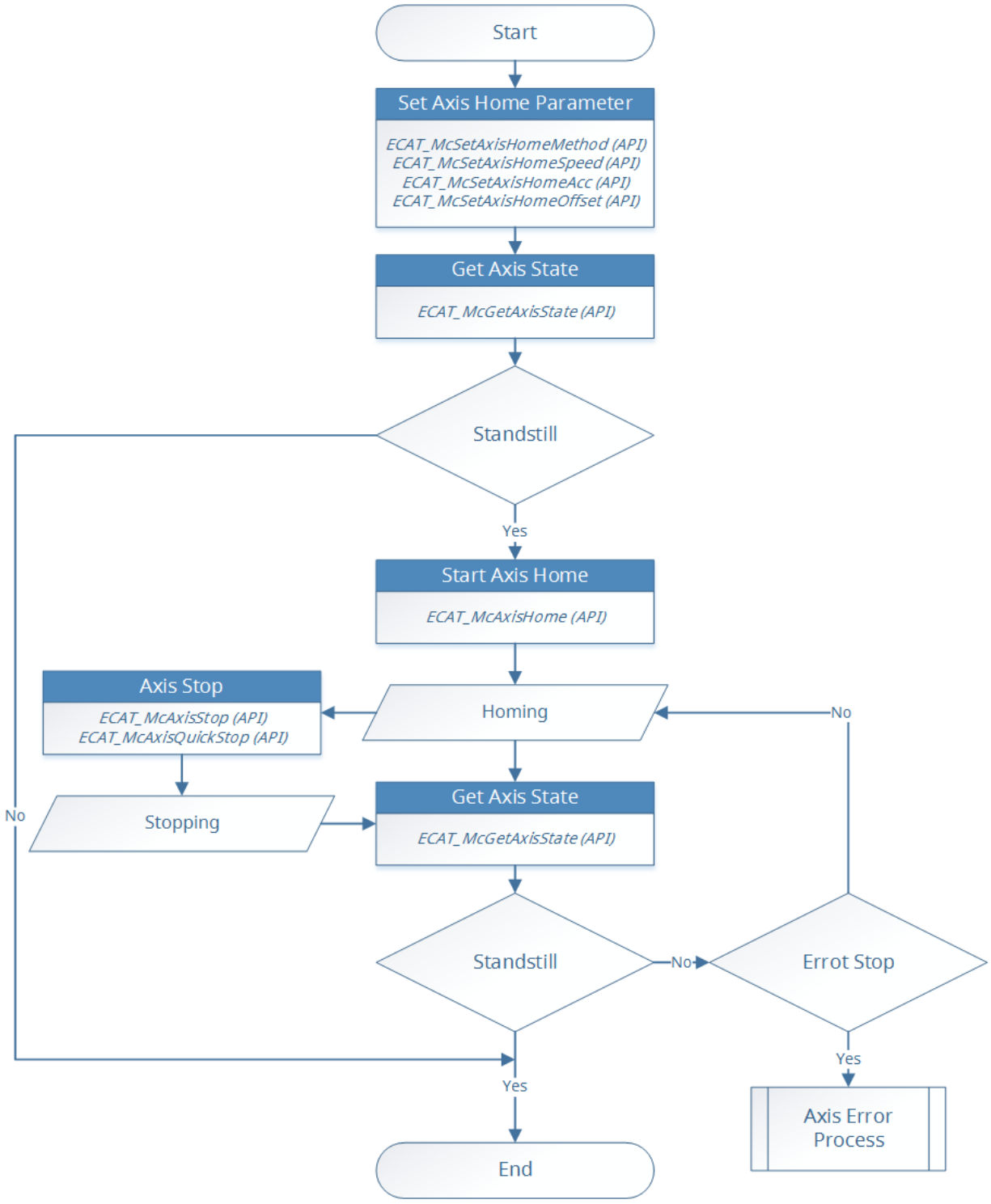


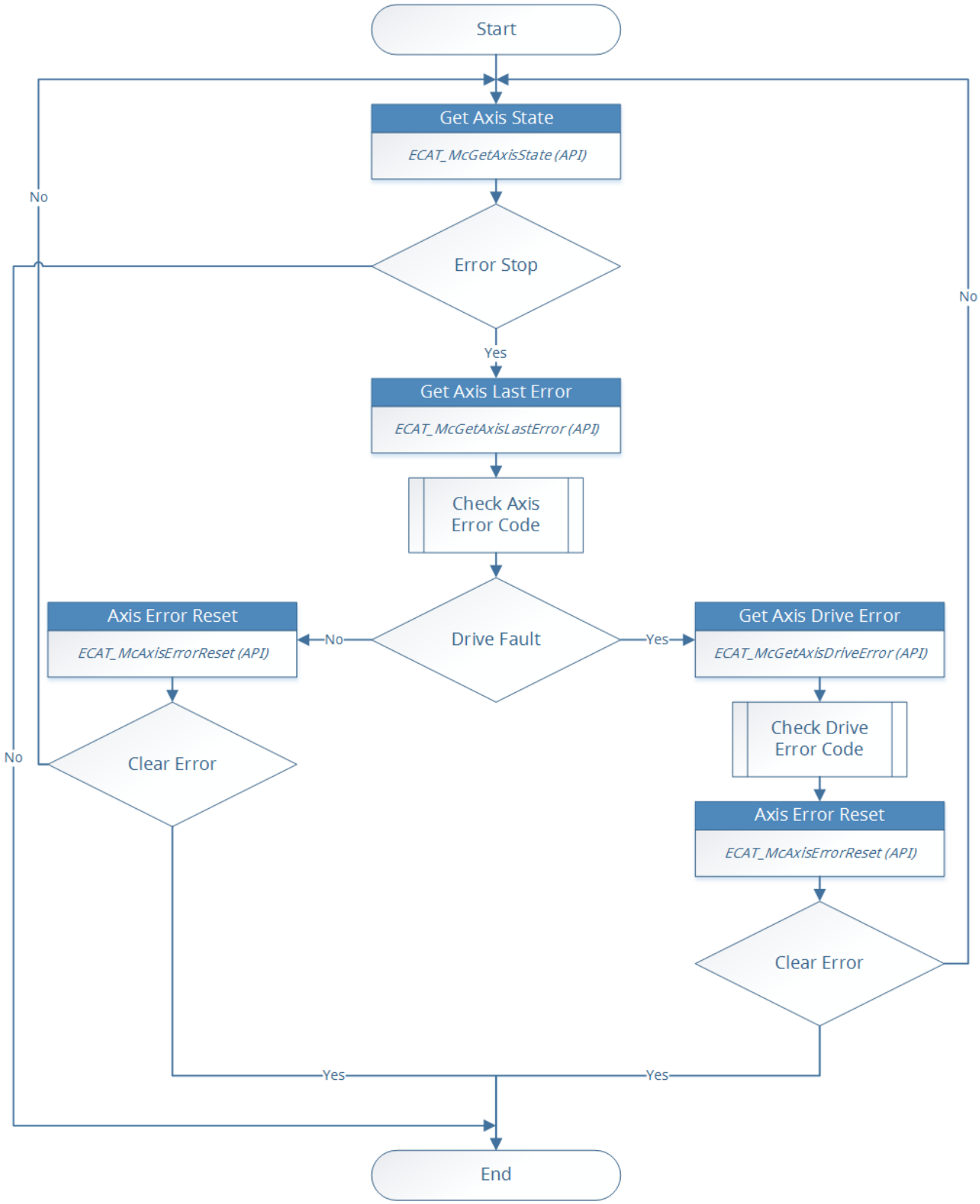
Figure 4-5

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

4.3.4. Axis Error Process

As show 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 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 driver, *McGetAxisDriveError* function can be called to get the drive's error code, and then call *McAxisErrorReset* to clear its error, Some drive errors can be cleared by the reset command; but some do not. If the reset command does not change the axis state back to Standstill, please restart the drive to clear the its error.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.



Figrue 4-6

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

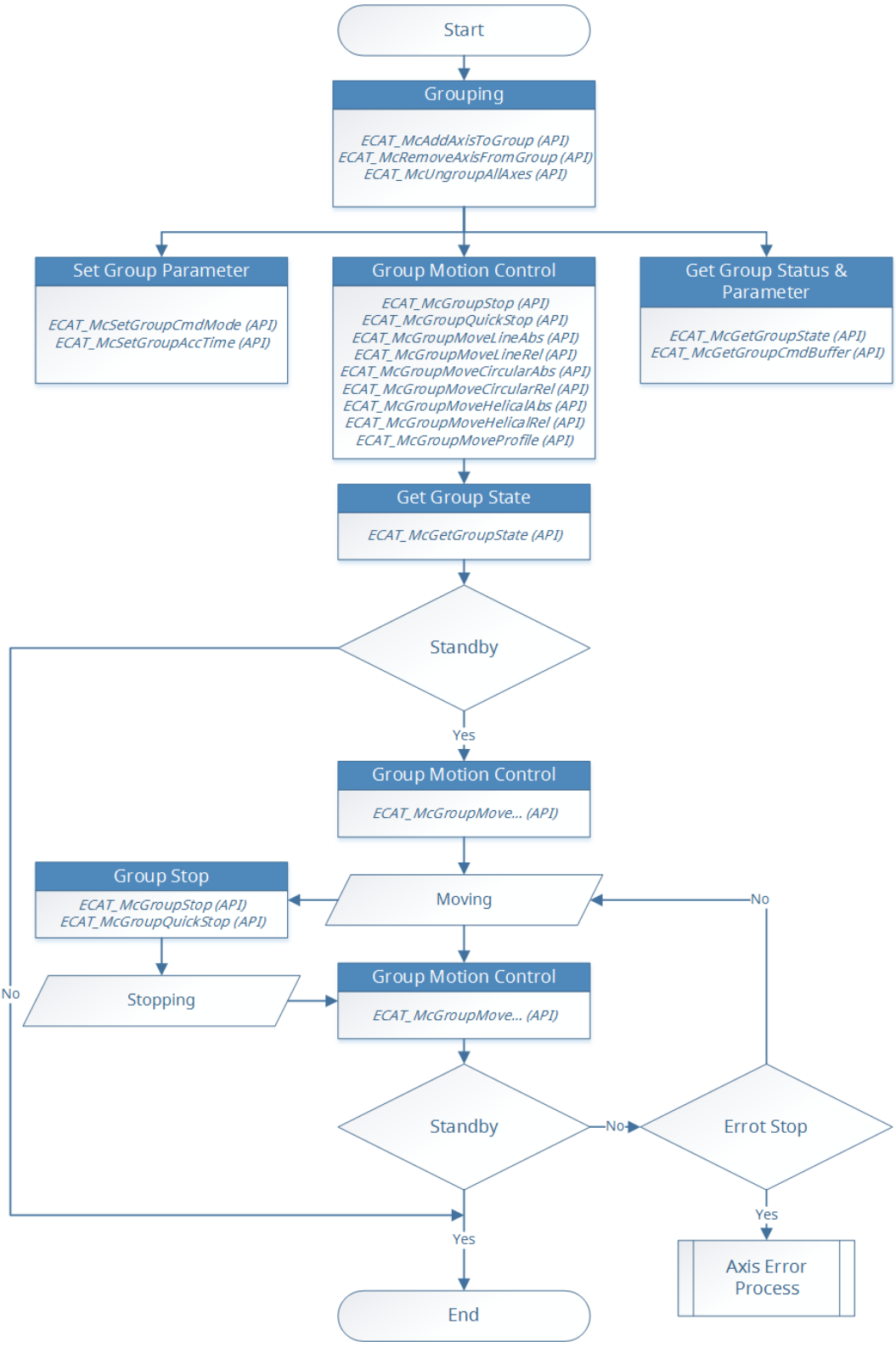
4.3.5. Group Moving

The user can use the group motion control to do the multu-axis interpolation motion. Before using the group motion, the user needs to group a group. *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 grouping axes in a group, users can use group motion commands, 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 stoppep. Users can issue a new motion command. After a motion function is called successfully, the group state changes from Standby to Moving.

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.



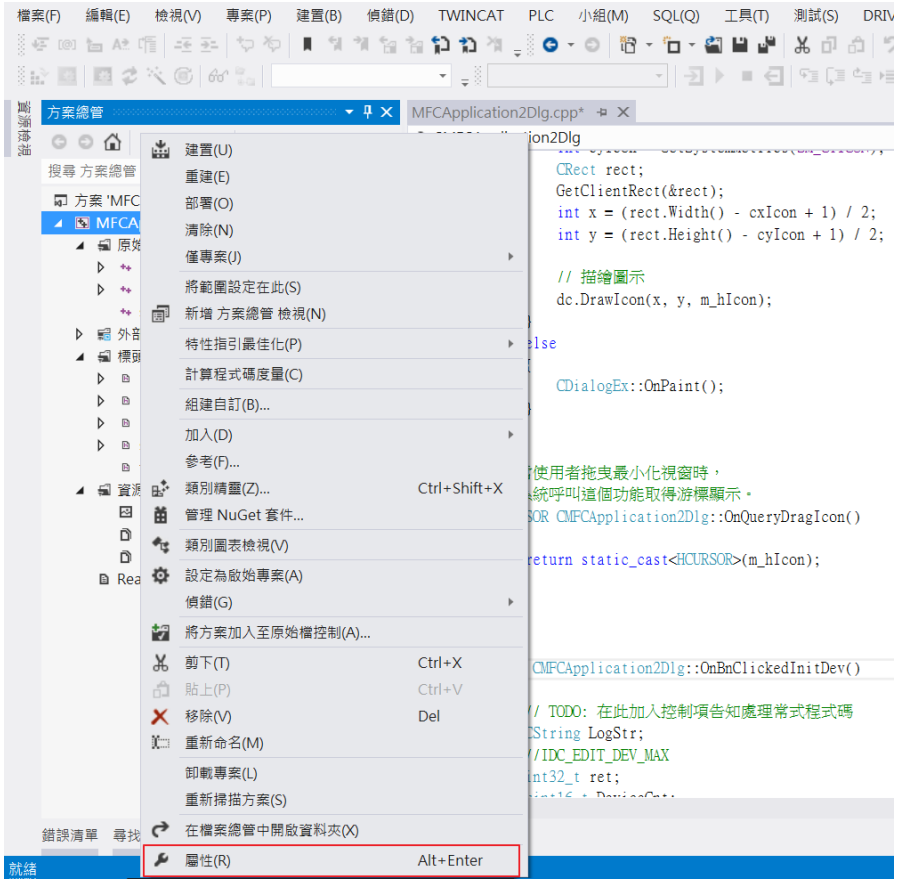
Figrue 4-7

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

4.4. Use motion Library in Windows

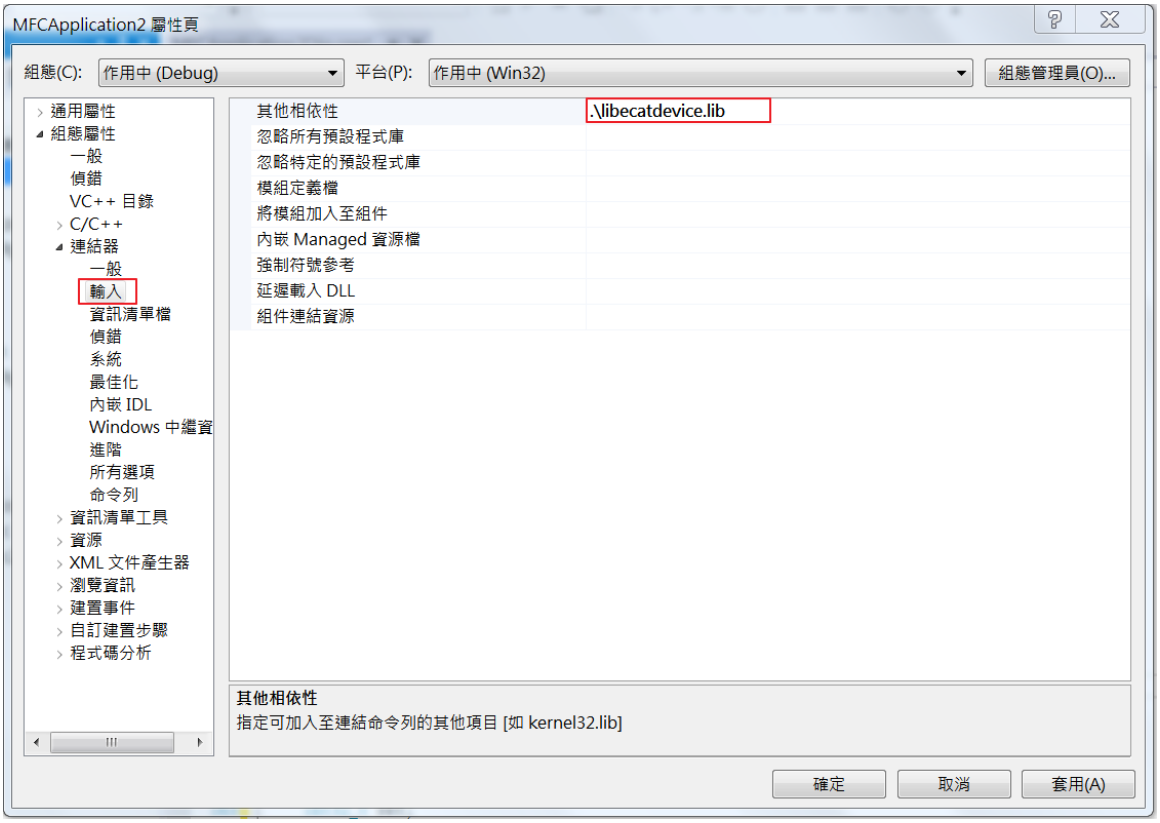
4.4.1. For Visual Studio

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.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.



Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.3. ECAT_CloseDevice

Description:

Close a device 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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.4. ECAT_GetDeviceDI

Description:

Get the digital input data of the specified device.

Syntax:

int32_t ECAT_GetDeviceDI(uint16_t DeviceNo, uint32_t *Value)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
Value	uint32*	OUT	Digital input data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.5. ECAT_ GetDeviceDIBit

Description:

Get a bit state of a device's 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
BitNo	uint16	IN	Bit number
Value	uint32*	OUT	Bit data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.6. ECAT_GetDeviceDO

Description:

Get the 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
Value	uint32*	OUT	Digital output data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.7. ECAT_ GetDeviceDOBit

Description:

Get a bit state of a device's 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
BitNo	uint16	IN	Bit number
Value	uint32*	OUT	Bit data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.8. ECAT_SetDeviceDO

Description:

Set the 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
Value	uint32	OUT	Digital output data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.9. ECAT_ SetDeviceDOBit

Description:

Set a bit data of a device's 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
BitNo	uint16	IN	Bit number
Value	uint32*	OUT	Bit data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.10. ECAT_SetDeviceEncProperty

Description:

Set the 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
EncNo	uint16_t	IN	Encoder number
Mode	uint8_t	IN	Encoder Mode 1: CW/CCW 2: Pulse/Dir 3: A/B Phase
InvertCnt	uint8_t	IN	Invert count
LPF	uint8_t	IN	Low pass filter (As show in Table 5-1)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.11. ECAT_GetDeviceEncProperty

Description:

Get the 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
EncNo	uint16_t	IN	Encoder number
Mode	uint8_t	OUT	Encoder Mode 1: CW/CCW 2: Pulse/Dir 3: A/B Phase
InvertCnt	uint8_t	OUT	Invert count
LPF	uint8_t	OUT	Low pass filter

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.12. ECAT_GetDeviceEncCount

Description:

Get the 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
EncNo	uint16_t	IN	Encoder number
Cnt	int32_t *	OUT	Encoder counter value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.13. ECAT_ResetDeviceEncCount

Description:

Clear an 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
EncNo	uint16_t	IN	Encoder number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.14. ECAT_SetDeviceCmpTrigProperty

Description:

Set the device compare-trigger related configurations.

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
EncNo	uint16_t	IN	Encoder number
PulseWidth	uint32_t	IN	Output Pulse width setting, each unit is 0.016us, the maximum value is 0x7fffffff x 0.016us

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.15. ECAT_GetDeviceCmpTrigProperty

Description:

Get the device compare-trigger related configurations.

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
EncNo	uint16_t	IN	Encoder number
PulseWidth	uint32_t *	OUT	Output Pulse width setting, each unit is 0.016us, the maximum value is 0x7fffffff x 0.016us

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.16. ECAT_SetDeviceCmpTrigData

Description:

Start single compare-trigger function.

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
EncNo	uint16_t	IN	Encoder number
CmpData	int32_t	IN	Single compare-trigger data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.17. ECAT_SetDeviceContCmpTrigData

Description:

Start 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
EncNo	uint16_t	IN	Encoder number
Start	int32_t	IN	Start compare-trigger data
Interval	uint32_t	IN	Trigger interval
Times	uint32_t	IN	Set 0 to continuous trigger, greater than 0 is the number of triggers

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.18. 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, uint16_t BitNo)

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.19. ECAT_GetDeviceEmg

Description:

Get the device emergency stop signal related configurations..

Syntax:

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

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
Source	uint8_t *	OUT	Emergency stop signal source (As show in 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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Example:

[C/C++]

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

ret = ECAT_GetDeviceEmg(DeviceNo, &Source, &Enable, &Logic, &SlaveNo, &BitNo)
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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.20. 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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.21. ECAT_SetDeviceEmgSoftSig

Description:

Trigger software emergency stop signal.

Syntax:

int32_t ECAT_SetDeviceEmgSoftSig (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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Example:

[C/C++]

```
int32_t ret;
uint32_t DeviceNo = 0;

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.22. ECAT_SetDeviceMPG

Description:

Set the device manual pulse generator related configurations.

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
Enable	uint8_t	IN	Enable/Disable MPG
AxisNo	uint16_t *	IN	Initialize the manual pulse generator control axis number
AxisCount	uint16_t	IN	Initialize the number of axes

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.23. ECAT_GetDeviceMPG

Description:

Get the device manual pulse generator related configurations.

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
Enable	uint8_t	OUT	Enable/Disable MPG
AxisNo	uint16_t *	OUT	Initialize the manual pulse generator control axis number
AxisCount	uint16_t	OUT	Initialize the number of axes

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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]);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.24. ECAT_GetDeviceState

Description:

Get the EtherCAT network status of a device.

Syntax:

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

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
LinkUp	uint32_t*	OUT	Network link status of Ethernet 0: Link Down 1: Link Up
SlavesResp	uint32_t*	OUT	Sum of responding slaves on all Ethernet devices
AlStates	uint32_t*	OUT	EtherCAT AL states of all slaves (As shown in Table 5-3)
Wc	uint32_t*	OUT	EtherCAT working counter value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Table 5-3: 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, AlStates, Wc;
ret = ECAT_OpenDevice(DeviceNo);
ret = ECAT_GetDeviceState(DeviceNo, &LinkUp, &SlavesResp, &AlStates, &Wc);
if(ret < 0)
    printf("Failed to get device state:%d\n", ret);
else
{
    if(AlStates == ECAT_AS_INIT)
        sprintf(StrAlState,"INIT");
    else if(AlStates == ECAT_AS_PREOP)
        sprintf(StrAlState,"PREOP");
    else if(AlStates == ECAT_AS_SAFEOP)
        sprintf(StrAlState,"SAFEOP");
    else if(AlStates == ECAT_AS_OP)
        sprintf(StrAlState,"OP");
    else
        sprintf(StrAlState,"Invalid");
    sprintf(buffer,"Slave(s):%u | AL States:%s | Link is :%s | Wc:%-u "
                    , SlavesResp, StrAlState, LinkUp? "up" : "down", Wc);
    printf("%s\n", buffer);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.25. ECAT_StartDeviceOpTask

Description:

Start the device EtherCAT operation task.

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
NetworkInfoNo	uint16_t	IN	Network information number (Configured by the EtherCAT utility)
EnumCycleTime	uint8_t	IN	Cycle time number (As show in Table 5-4)
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Table 5-4: 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;
ret = ECAT_OpenDevice(DeviceNo);

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.26. 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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.27. ECAT_SetTimer

Description:

Set Timer Interval.

Syntax:

int32_t ECAT_SetTimer(uint16_t DeviceNo, uint32_t Interval)

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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…
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.28. ECAT_ SetTimerStop

Description:

Disable Timer.

Syntax:

int32_t ECAT_ SetTimerStop(DeviceNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

5.29. ECAT_WaitforTimer

Description:

Wait until time's up.

Syntax:

int32_t ECAT_WaitforTimer(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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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…
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6. Slave Operation Functions

6.1. ECAT_GetSlaveInfo

Description:

Get slave information of a specific 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
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
SlaveType	uint32_t*	OUT	Slave Type (As show in Table 6-1)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Table 6-1: Slave Type

Macro Definition	Value	Description
SLAVE_TYPE_GENERIC	0	Generic
SLAVE_TYPE_CiA402	1	CiA 402
SLAVE_TYPE_STEPPER_MOTOR	2	Single Axis Stepper Motor
SLAVE_TYPE_4_AXIS_STEPPER_MOTOR	3	4-Axis Stepper Motor

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"
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
                    "           |-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);
            }
        }
    }
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.2. ECAT_GetSlaveSdoObject

Description:

Get SDO data from a specific slave.

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
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
AbortCode	uint32_t*	OUT	Abort code of the SDO (Refer to Appendix "SDO Abort messages")

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.3. ECAT_SetSlaveSdoObject

Description:

Set SDO data to a specific slave.

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
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
AbortCode	uint32_t*	OUT	Abort code of the SDO (Refer to Appendix "SDO Abort messages")

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.4. ECAT_SetSlaveRxPdoData

Description:

Set RxPDO data to a specific slave.

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
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)
Data	uint8_t*	IN	Data buffer

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.5. ECAT_GetSlaveRxPdoData

Description:

Get RxPDO data of a specific slave.

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
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)
Data	uint8_t*	IN	Data buffer

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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]);
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.6. ECAT_GetSlaveTxPdoData

Description:

Get TxPDO data of a specific slave.

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
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)
Data	uint8_t*	IN	Data buffer

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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]);
	}
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.7. ECAT_GetSlaveDI

Description:

Get the digital input data of a specific slave.

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
SlaveNo	uint16_t	IN	Slave number
Value	uint32_t*	OUT	Digital input data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
SlaveNo	uint16_t	IN	Slave number
BitNo	uint16_t	IN	Bit number
Value	uint32_t*	OUT	Bit data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.9. ECAT_GetSlaveDO

Description:

Get the digital output data of a specific slave.

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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
SlaveNo	uint16_t	IN	Slave number
BitNo	uint16_t	IN	Bit number
Value	uint32_t*	OUT	Bit data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.11. ECAT_SetSlaveDO

Description:

Set the digital output data of a specific slave.

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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
SlaveNo	uint16_t	IN	Slave number
BitNo	uint16_t	IN	Bit number
Value	uint32_t	IN	Bit data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.18. ECAT_GetSlaveAoVoltData

Description:

Get the voltage output 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.19. ECAT_SetSlaveEncProperty

Description:

Set the encoder property value.

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
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder number
Mode	uint8_t	IN	Encoder Mode 1: CW/CCW 2: Pulse/Dir 3: A/B Phase
InvertCnt	uint8_t	IN	Invert count
LPF	uint8_t	IN	Low pass filter

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.20. ECAT_GetSlaveEncProperty

Description:

Get the encoder property value.

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
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder number
Mode	uint8_t *	OUT	Encoder Mode 1: CW/CCW 2: Pulse/Dir 3: A/B Phase
InvertCnt	uint8_t *	OUT	Invert count
LPF	uint8_t *	OUT	Low pass filter

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.21. ECAT_GetSlaveEncCount

Description:

Get the encoder counter value.

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
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder number
Cnt	int32_t *	OUT	Encoder counter value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.22. ECAT_ResetSlaveEncCount

Description:

Clear the encoder counter value.

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
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.23. ECAT_SetSlaveEncLatch

Description:

Set the position latch function property value.

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
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder number
Enable	uint8_t	IN	Enable/Disable latch

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.24. ECAT_GetSlaveEncLatch

Description:

Get the position latch function property value.

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
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder number
Enable	uint8_t *	OUT	Enable/Disable latch

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

6.25. ECAT_GetSlaveEncLatchCnt

Description:

Get the latch count.

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
SlaveNo	uint16_t	IN	Slave number
EncNo	uint16_t	IN	Encoder number
Cnt	int32_t *	OUT	Latch count

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7. Motion Control Functions

7.1. Motion Control Initialization

7.1.1. ECAT_McInit

Description:

Motion control initialize.

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
SlaveNo	uint16_t[]	IN	A 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 which are individual defined.
AxisCount	uint16_t	IN	Initialize the number of axes (MC_AXIS_NO_MAX macro is the maximum number of axes)

Return:

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
AxisNo	uint16_t	IN	Axis number
PPU	double	IN	Pulse per unit

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.3. ECAT_McGetAxisPPU

Description:

Get pulse per unit of a specific 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
AxisNo	uint16_t	IN	Axis number
PPU	Double*	OUT	Pulse per unit

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.4. ECAT_McSetProfileData

Description:

Set a position array data into a profile 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
ProfileNo	uint16_t	IN	Profile buffer number (MC_PROFILE_NO_MAX macro is the maximum number)
Data	double*	IN	Data buffer (MC_PROFILE_DATA_MAX macro is the maximum size of the data)
DataSize	uint16_t	IN	Size of data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
ProfileNo	uint16_t	IN	Profile buffer number (MC_PROFILE_NO_MAX macro is the maximum number)
Data	double*	OUT	Data buffer (MC_PROFILE_DATA_MAX macro is the maximum size of the data)
DataSize	uint16_t	IN	Size of data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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]);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.6. ECAT_McSetAxisAccTime

Description:

Set acceleration time to a specific 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
AxisNo	uint16_t	IN	Axis number
Time_ms	uint16_t	IN	Acceleration time (Unit: millisecond)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.7. ECAT_McGetAxisAccTime

Description:

Get acceleration time of a specific 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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.8. ECAT_McSetAxisAccDecType

Description:

Set acceleration Type to a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.9. ECAT_McGetAxisAccDecType

Description:

Get acceleration Type of a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.10. ECAT_McSetAxisEncoderPPR

Description:

Set encoder pulse per revolution to a specific 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
AxisNo	uint16_t	IN	Axis number
PPR	Uint32_t	IN	Pulse per revolution

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.11. ECAT_McGetAxisEncoderPPR

Description:

Get encoder pulse per revolution of a specific 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
AxisNo	uint16_t	IN	Axis number
PPR	Uint32_t*	OUT	Pulse per revolution

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.12. ECAT_McSetAxisMotorPPR

Description:

Set motor pulse per revolution to a specific 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
AxisNo	uint16_t	IN	Axis number
PPR	Uint32_t	IN	Pulse per revolution

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.13. ECAT_McGetAxisMotorPPR

Description:

Get motor pulse per revolution of a specific 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
AxisNo	uint16_t	IN	Axis number
PPR	Uint32_t*	OUT	Pulse per revolution

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.14. ECAT_McSetEcamTable

Description:

Set the slave position data of the specified E-CAM table number.

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
TableNo	uint16_t	IN	E-CAM table number
Data	double*	IN	Slave position data (Unit:user unit)
DataSize	uint16_t	IN	Size of data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.15. ECAT_McGetEcamTable

Description:

Get the slave position data of the specified E-CAM table number.

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
TableNo	uint16_t	IN	E-CAM table number
Data	double*	OUT	Slave position data (Unit:user unit)
DataSize	uint16_t	IN	Size of data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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]);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.16. ECAT_McConfigEcamTable

Description:

Configure E-CAM table related parameters.

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
TableNo	uint16_t	IN	E-CAM table number
SlaveAbs	uint8_t	IN	Slave positon data type 0: Relative position 1: Absolute position

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.17. ECAT_McSetAxisTouchProbeProperty

Description:

Set the specified axis number Touch Probe function related configuration.

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.18. ECAT_McGetAxisTouchProbeProperty

Description:

Get the specified axis number Touch Probe function related configuration.

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.2.19. ECAT_McGetAxisTouchProbeValue

Description:

Get the specified axis number Touch Probe value.

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.3. Axis Status

7.3.1. ECAT_McGetAxisActualPos

Description:

Get actual position of a specific axis.

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
AxisNo	uint16_t	IN	Axis number
Pos	double*	OUT	Actual position (Unit:user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.3.2. ECAT_McGetAxisActualVel

Description:

Get actual velocity of a specific axis.

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
AxisNo	uint16_t	IN	Axis number
Vel	double*	OUT	Actual velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.3.3. ECAT_McGetAxisState

Description:

Get the state of a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
            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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.3.4. ECAT_McGetAxisLastError

Description:

Get last error of a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
        }
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.3.5. ECAT_McGetAxisDriveError

Description:

Get drive error of a specific 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
AxisNo	uint16_t	IN	Axis number
Error	int16_t *	OUT	drive error (Refer to the drive user manual to find the error code)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
                }
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
            }
        }
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.3.6. ECAT_McGetAxisDI

Description:

Get digital input of a specific axis.

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.4. Axis Homing

7.4.1. ECAT_McSetAxisHomeMethod

Description:

Set homing method to a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.4.2. ECAT_McGetAxisHomeMethod

Description:

Get homing method of a specific 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
AxisNo	uint16_t	IN	Axis number
Method	int32_t*	OUT	Homing method

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.4.3. ECAT_McSetAxisHomeSpeed

Description:

Set homing speed to a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.4.4. ECAT_McGetAxisHomeSpeed

Description:

Get homing speed of a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.4.5. ECAT_McSetAxisHomeAcc

Description:

Set homing acceleration to a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.4.6. ECAT_McGetAxisHomeAcc

Description:

Get homing acceleration of a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.4.7. ECAT_McSetAxisHomeOffset

Description:

Set home offset to a specific 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
AxisNo	uint16_t	IN	Axis number
Offset	double	IN	Home offset (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.4.8. ECAT_McGetAxisHomeOffset

Description:

Get home offset of a specific 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
AxisNo	uint16_t	IN	Axis number
Offset	Double*	OUT	Home offset (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.4.9. ECAT_McAxisHome

Description:

Start home motion of a specific 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
AxisNo	uint16_t	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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)
{
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
        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");
        }
    }
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5. Axis Moving

7.5.1. ECAT_McAxisErrorReset

Description:

Reset error state of a specific 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
AxisNo	uint16_t	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
        }
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.2. ECAT_McAxisMoveAbs

Description:

Start absolute position motion of a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.3. ECAT_McAxisMoveRel

Description:

Start relative position motion of a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
        {
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.4. ECAT_McAxisChangePos

Description:

When the specified axis number 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
AxisNo	uint16_t	IN	Axis number
Pos	double	IN	End positon (Unit: user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
            }
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
        }

        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");
        }
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.5. ECAT_McAxisChangeVel

Description:

When the specified axis number 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
AxisNo	uint16_t	IN	Axis number
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
            }
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
        }

        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");
        }
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.6. ECAT_McAxisMoveVel

Description:

Start a never ending movement with a specified velocity.

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
AxisNo	uint16_t	IN	Axis number
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.7. ECAT_McAxisGearIn

Description:

Start gear synchronization motion with a 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Table 7-3: Source for synchronization

Macro Definition	Value	Description
MC_AXIS_SYNC_SOURCE_SET_VALUE	0	Synchronization on master set value
MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE	1	Synchronization on master actual value

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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
{
    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");
        }
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.8. ECAT_McAxisGearOut

Description:

Disengages the slave axis from the master axis.

Syntax:

int32_t ECAT_McAxisGearOut(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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Example:

[C/C++]

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

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.9. ECAT_McAxisMoveProfile

Description:

Start profile position motion of a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
        }
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.10. 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 SlaveScaling)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
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)
SlaveScaling	double	IN	Slave position output ratio

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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[360];
uint16_t DataSize = 360;
uint8_t SlaveAbs = 0; //Relative
double SlaveScaling = 1.0
double AxisPos = 10.0;
double AxisVel = 4;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;

/**
***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);
    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
{
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
        ret = ECAT_McAxisCamIn(DeviceNo, MasterNo, SlaveNo, TableNo
                , SyncSource, 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");
            }
        }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.11. ECAT_McAxisStop

Description:

Stop motion of of a specific axis.

Syntax:

int32_t ECAT_McAxisStop(uint16_t DeviceNo, uint16_t AxisNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
AxisNo	uint16_t	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
            {
                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");
            }
        }
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.5.12. ECAT_McAxisQuickStop

Description:

Quick stop motion of of a specific axis.

Syntax:

int32_t ECAT_McAxisQuickStop(uint16_t DeviceNo, uint16_t AxisNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
AxisNo	uint16_t	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
            {
                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");
            }
        }
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.6. Group Parameter Setting

7.6.1. ECAT_McAddAxisToGroup

Description:

Add one axis to a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.6.2. ECAT_McRemoveAxisFromGroup

Description:

Remove one axis from a specific 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
GroupNo	uint16_t	IN	Group number
AxisNo	uint16_t	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.6.3. ECAT_McUngroupAllAxes

Description:

Remove all axes from a specific 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
GroupNo	uint16_t	IN	Group number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.6.4. ECAT_McSetGroupCmdMode

Description:

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

Aborting: this new command will abort the current executing command, the start this new command.

Buffered: this new command will wait until the current executing command finished, than start this new command.

Blending: this new command will start when the current executing command just starts to decelerate. Both outputs will blend together and send to drive as the axis command.

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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");
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.6.5. ECAT_McGetGroupCmdMode

Description:

Get the group command mode of a specific 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.7. Group Status

7.7.1. ECAT_McGetGroupState

Description:

Get the state of a specific 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
GroupNo	uint16_t	IN	Group number
State	uint32_t*	OUT	Group state (As show in Table 7-5)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Table 7-5: 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");
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
                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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.7.2. ECAT_McGetGroupCmdBuffer

Description:

Get the remaining number of commands in the 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8. Group Moving

7.8.1. ECAT_McGroupMoveLineAbs

Description:

Start group absolute linear interpolation motion by position and velocity.

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
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Position arrary of axis in group, each value of array elements represents the axis absolute position. (Unit:user unit)
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.2. ECAT_McGroupMoveLineRel

Description:

Start group relative linear interpolation motion by position and velocity.

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
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Distance arrary of axis in group, each value of array elements represents the axis relative position. (Unit:user unit)
Vel	double	IN	Velocity (Unit: user unit/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.3. ECAT_McGroupMoveLineAbs_PT

Description:

Start group absolute linear interpolation motion by position and time.

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
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Position arrary of axis in group, each value of array elements represents the axis absolute position. (Unit:user unit)
Time	double	IN	Time(Unit: second)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
{
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
    //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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.4. ECAT_McGroupMoveLineRel_PT

Description:

Start group relative linear interpolation motion by position and time.

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
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Distance arrary of axis in group, each value of array elements represents the axis relative position. (Unit:user unit)
Time	double	IN	Time(Unit: second)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
{
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
    //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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.5. ECAT_McGroupMoveLineAbs_PVT

Description:

Start group 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
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Position arrary of axis in group, each value of array elements represents the axis absolute position. (Unit:user unit)
Vel	double[]	IN	Velocity arrary of axis in group, each value of array elements represents the axis velocitiy. (Unit:user unit/s)
Time	double	IN	Time(Unit: second)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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;
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
GroupNo	uint16_t	IN	Group number
Pos	double[]	IN	Position arrary of axis in group, each value of array elements represents the axis relative position. (Unit:user unit)
Vel	double[]	IN	Velocity arrary of axis in group, each value of array elements represents the axis velocitiy. (Unit:user unit/s)
Time	double	IN	Time(Unit: second)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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;
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.7. ECAT_McGroupMoveCircularAbs_CP_Angle

Description:

Start group absolute 2D circular interpolation motion by center position and 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
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity(Unit: user unit/s)
Angle	double	IN	Angle of rotation(Unit:degree)
AuxPos	double[]	IN	Absolute positon of center point (Unit:user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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 = 90;
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.8. ECAT_McGroupMoveCircularRel_CP_Angle

Description:

Start group relative 2D circular interpolation motion by center position and 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
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity(Unit: user unit/s)
Angle	double	IN	Angle of rotation(Unit:degree)
AuxPos	double[]	IN	Axis relative distance of center point (Unit:user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 10.0; //Center Position
    CircAuxPos [1] = 0.0; //Center Position
    CircAngle = 90;
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.9. ECAT_McGroupMoveCircularAbs_CP_EP

Description:

Start group absolute 2D circular interpolation motion by center position and 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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] = 10.0 // End Position
    CircEndPos[1] = -10.0 // 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");
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.10. ECAT_McGroupMoveCircularRel_CP_EP

Description:

Start group relative 2D circular interpolation motion by center position and 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
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	Axis relative distance of center point (Unit:user unit)
EndPos	double[]	IN	Axis relative distance of end point (Unit:user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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] = 10.0 // End Position
    CircEndPos[1] = -10.0 // End Position

    GroupVel = 5;
    CircDir = 1; //CCW
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.11. ECAT_McGroupMoveCircularAbs_BP_EP

Description:

Start group absolute 2D circular interpolation motion by border position and 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
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 border point (Unit:user unit)
EndPos	double[]	IN	Absolute positon of end point (Unit:user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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] = 2.929; //Border Position
    CircAuxPos[1] = -7.071; //Border Position
    CircEndPos[0] = 10.0 // End Position
    CircEndPos[1] = -10.0 // End Position

    GroupVel = 5;
    CircDir = 1; //CCW
    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
    {
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.12. ECAT_McGroupMoveCircularRel_BP_EP

Description:

Start group relative 2D circular interpolation motion by border position and 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
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	Axis relative distance of border point (Unit:user unit)
EndPos	double[]	IN	Axis relative distance of end point (Unit:user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
ret = ECAT_McGetGroupState(DeviceNo,GroupNo,&State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos[0] = 2.929; //Border Position
    CircAuxPos[1] = -7.071; //Border Position
    CircEndPos[0] = 10.0 // End Position
    CircEndPos[1] = -10.0 // End Position

    GroupVel = 5;
    CircDir = 1; //CCW
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.13. ECAT_McGroupMove3DCircularAbs_CP_Angle

Description:

Start group absolute 3D circular interpolation motion by center position and 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
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity(Unit: user unit/s)
Angle	double	IN	Angle of rotation(Unit:degree)
AuxPos	double[]	IN	Absolute positon of center point (Unit:user unit)
NV	double[]	IN	Normal vector

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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] = 3.5355.0;   //Center Position
    CircAuxPos [1] = 3.5355.0;   //Center Position
    CircAuxPos [2] = 0.0;        //Center Position

    NV [0] = -0.7071.0;       //Normal Vector
    NV [1] = 0.7071.0;   //Normal Vector
    NV [2] = 0.0;        //Normal Vector

    CircAngle = 90;
    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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
            printf("Group move circular successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.14. ECAT_McGroupMove3DCircularRel_CP_Angle

Description:

Start group relative 3D circular interpolation motion by center position and 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: user unit/s)
Angle	double	IN	Angle of rotation(Unit:degree)
AuxPos	double[]	IN	Axis relative distance of center point (Unit:user unit)
NV	double[]	IN	Normal vector

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
ret = ECAT_McGetGroupState(DeviceNo,GroupNo,&State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 3.5355.0;   //Center Position
    CircAuxPos [1] = 3.5355.0;   //Center Position
    CircAuxPos [2] = 0.0;        //Center Position

    NV [0] = -0.7071.0;      //Normal Vector
    NV [1] = 0.7071.0;   //Normal Vector
    NV [2] = 0.0;        //Normal Vector

    CircAngle = 90;
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.15. ECAT_McGroupMove3DCircularAbs_CP_EP

Description:

Start group absolute 3D circular interpolation motion by center position and 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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
        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] = 3.5355.0;    //Center Position
        CircAuxPos [1] = 3.5355.0;    //Center Position
        CircAuxPos [2] = 0.0;         //Center Position

        CircEndPos[0] = 3.5355.0      // End Position
        CircEndPos[1] = 3.5355.0      // 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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.16. ECAT_McGroupMove3DCircularRel_CP_EP

Description:

Start group relative 3D circular interpolation motion by center position and 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
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	Axis relative distance of center point (Unit:user unit)
EndPos	double[]	IN	Axis relative distance of end point (Unit:user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
        return;
}

ret = ECAT_McGetGroupState(DeviceNo,GroupNo,&State);
if(State == MC_GS_STANDBY) //Standby
{
        CircAuxPos [0] = 3.5355.0;    //Center Position
        CircAuxPos [1] = 3.5355.0;    //Center Position
        CircAuxPos [2] = 0.0;         //Center Position

        CircEndPos[0] = 3.5355.0      // End Position
        CircEndPos[1] = 3.5355.0      // 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");
        }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.17. ECAT_McGroupMove3DCircularAbs_BP_EP

Description:

Start group absolute 3D circular interpolation motion by border position and 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
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 border point (Unit:user unit)
EndPos	double[]	IN	Absolute positon of end point (Unit:user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
        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] = 1.036.0;     //Border Position
        CircAuxPos [1] = 1.036.0;     //Border Position
        CircAuxPos [2] = 3.5355.0;    //Border Position

        CircEndPos[0] = 3.5355.0       // End Position
        CircEndPos[1] = 3.5355.0       // End Position
        CircEndPos[2] = 5.0         // End Position
        CircDir = 0; // CW
        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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.18. ECAT_McGroupMove3DCircularRel_BP_EP

Description:

Start group relative 3D circular interpolation motion by border position and 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
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	Axis relative distance of border point (Unit:user unit)
EndPos	double[]	IN	Axis relative distance of end point (Unit:user unit)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
        return;
}

ret = ECAT_McGetGroupState(DeviceNo,GroupNo,&State);
if(State == MC_GS_STANDBY) //Standby
{
        CircAuxPos [0] = 1.036.0;     //Border Position
        CircAuxPos [1] = 1.036.0;     //Border Position
        CircAuxPos [2] = 3.5355.0;    //Border Position

        CircEndPos[0] = 3.5355.0       // End Position
        CircEndPos[1] = 3.5355.0       // End Position
        CircEndPos[2] = 5.0         // End Position
        CircDir = 0; // CW
        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");
        }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.19. ECAT_McGroupMoveHelicalAbs

Description:

Start absolute helical interpolation motion of a specific group.

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
GroupNo	uint16_t	IN	Group number
Angle	double	IN	Angle of rotation, 360 indicates one full revolution, 720 will result in two full revolutions (Unit:degree)
AuxPos	double[]	IN	Absolute positon of 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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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 = 720;
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.20. ECAT_McGroupMoveHelicalRel

Description:

Start relative helical interpolation motion of a specific group.

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
GroupNo	uint16_t	IN	Group number
Angle	double	IN	Angle of rotation, 360 indicates one full revolution, 720 will result in two full revolutions (Unit:degree)
AuxPos	double[]	IN	Axis relative distance of 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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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 = 720;
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.21. ECAT_McGroupMove3DHelicalAbs_CP_Angle

Description:

Start absolute 3D helical interpolation motion of a specific group.

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
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, 720 will result in two full revolutions (Unit:degree)
Pitch	double	Pitch	Pitch(Unit: user unit)
AuxPos	double[]	AuxPos	Absolute positon of center point (Unit:user unit)
NV	double[]	IN	Normal vector

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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] = 3.5355.0;   //Center Position
    CircAuxPos [1] = 3.5355.0;   //Center Position
    CircAuxPos [2] = 0.0;        //Center Position

    NV [0] = -0.7071.0;      //Normal Vector
    NV [1] = 0.7071.0;   //Normal Vector
    NV [2] = 0.0;        //Normal Vector

    CircAngle = 720;
    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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.22. ECAT_McGroupMove3DHelicalRel_CP_Angle

Description:

Start relative 3D helical interpolation motion of a specific group.

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
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, 720 will result in two full revolutions (Unit:degree)
Pitch	double	Pitch	Pitch(Unit: user unit)
AuxPos	double[]	AuxPos	Axis relative distance of center point (Unit:user unit)
NV	double[]	IN	Normal vector

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
ret = ECAT_McGetGroupState(DeviceNo,GroupNo,&State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 3.5355.0;   //Center Position
    CircAuxPos [1] = 3.5355.0;   //Center Position
    CircAuxPos [2] = 0.0;        //Center Position

    NV [0] = -0.7071.0;      //Normal Vector
    NV [1] = 0.7071.0;   //Normal Vector
    NV [2] = 0.0;        //Normal Vector

    CircAngle = 720;
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.23. ECAT_McGroupMoveConicalHelixAbs

Description:

Start absolute conical helix interpolation motion of a specific group.

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
GroupNo	uint16_t	IN	Group number
Angle	double	IN	Angle of rotation, 360 indicates one full revolution, 720 will result in two full revolutions (Unit:degree)
AuxPos	double[]	IN	Absolute positon of center 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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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] = 5.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircAngle = 720;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 15;
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.24. ECAT_ McGroupMoveConicalHelixRel

Description:

Start relative conical helix interpolation motion of a specific group.

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
GroupNo	uint16_t	IN	Group number
Angle	double	IN	Angle of rotation, 360 indicates one full revolution, 720 will result in two full revolutions (Unit:degree)
AuxPos	double[]	IN	Axis relative distance of center 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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
{
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
    CircAuxPos[0] = 5.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircAngle = 720;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 15;
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.25. ECAT_McGroupMove3DConicalHelixAbs_CP_Angle

Description:

Start absolute 3D conical helix interpolation motion of a specific group.

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
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: user unit/s)
Angle	double	IN	Angle of rotation, 360 indicates one full revolution, 720 will result in two full revolutions (Unit:degree)
Pitch	double	IN	Pitch(Unit: user unit)
AuxPos	double[]	IN	Absolute positon 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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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] = 3.5355.0;   //Center Position
    CircAuxPos [1] = 3.5355.0;   //Center Position
    CircAuxPos [2] = 0.0;        //Center Position

    NV [0] = -0.7071.0;      //Normal Vector
    NV [1] = 0.7071.0;   //Normal Vector
    NV [2] = 0.0;        //Normal Vector

    CircAngle = 720;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 15;
    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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
            printf("Group move circular successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.26. ECAT_McGroupMove3DConicalHelixRel_CP_Angle

Description:

Start relative 3D conical helix interpolation motion of a specific group.

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
GroupNo	uint16_t	IN	Group number
Vel	double	IN	Velocity (Unit: user unit/s)
Angle	double	IN	Angle of rotation, 360 indicates one full revolution, 720 will result in two full revolutions (Unit:degree)
Pitch	double	IN	Pitch(Unit: user unit)
AuxPos	double[]	IN	Axis relative distance 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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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;
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
ret = ECAT_McGetGroupState(DeviceNo,GroupNo,&State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 3.5355.0;   //Center Position
    CircAuxPos [1] = 3.5355.0;   //Center Position
    CircAuxPos [2] = 0.0;        //Center Position

    NV [0] = -0.7071.0;      //Normal Vector
    NV [1] = 0.7071.0;   //Normal Vector
    NV [2] = 0.0;        //Normal Vector

    CircAngle = 720;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 15;
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.27. ECAT_McGroupMoveProfile

Description:

Start profile position motion of a specific group.

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
GroupNo	uint16_t	IN	Group number
ProfileNo	uint16_t[]	IN	Profile buffer number arrary Each element of array represents the axis profile buffer number
TotalStep	uint16_t	IN	Total moving steps

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
    }
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
    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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.28. ECAT_McGroupMoveDwell

Description:

Add dwell time to group continuous motion control..

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
GroupNo	uint16_t	IN	Group number
Cnt	Uint32_t	IN	Dwell time, multiple of cycle time

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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");
    }
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.29. ECAT_McGroupStop

Description:

Stop motion of of a specific group.

Syntax:

int32_t ECAT_McGroupStop(uint16_t DeviceNo, uint16_t GroupNo)

Parameters:

Name	Type	IN or OUT	Description
DeviceNo	uint16_t	IN	Device number
GroupNo	uint16_t	IN	Group number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
            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");
            }
        }
    }
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.8.30. ECAT_McGroupQuickStop

Description:

Quick stop motion of of a specific 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
GroupNo	uint16_t	IN	Group number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

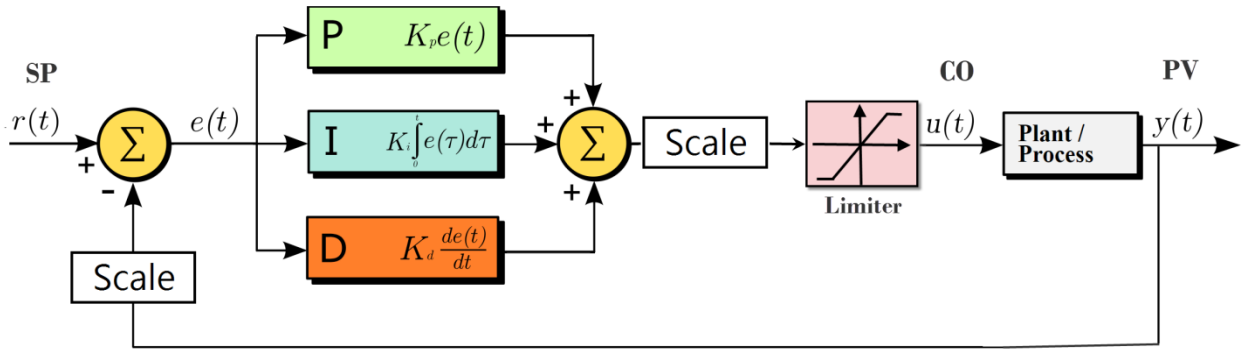
Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
            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");
            }
        }
    }
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9. PID Controller



SP:SetPoint

CO:Controller Output

PV:Process Variable

e(t):SP-PV

Simulate Plant:

$$G(s) = \frac{1}{s+1}$$

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.1. ECAT_PidGetSetPointValue

Description:

Get PID Controller 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
PidNo	uint32_t	IN	PID Controller number
SetPointValue	double *	OUT	PID Controller Set Point Value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.2. ECAT_PidSetSetPointValue

Description:

Set PID Controller 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
PidNo	uint32_t	IN	PID Controller number
SetPointValue	double	IN	PID Controller Set Point Value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.3. ECAT_PidGetProcessVariable

Description:

Get PID Controller 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
PidNo	uint32_t	IN	PID Controller number
ProcessVariable	double *	OUT	PID Controller Process Variable

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.4. ECAT_PidGetSampleTime

Description:

Get PID Controller Interval，Unit: Ethercat CycleTime.

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	Interval，Unit: Ethercat CycleTime

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.5. ECAT_PidSetSampleTime

Description:

Set PID Controller Interval，Unit: Ethercat CycleTime.

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
PidNo	uint32_t	IN	PID Controller number
Interval	int32_t	IN	Interval，Unit: Ethercat CycleTime

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.6. ECAT_PidGetStatus

Description:

Get PID Controller Status.

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
PidNo	uint32_t	IN	PID Controller number
status	uint8_t*	Output	Status 0:Not enable 1: enable

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.7. ECAT_PidSetStatus

Description:

Set PID Controller Status.Warn:Turn enable to disable will not set Control oputut module to 0; Turn disable to enable will set Control oputut module to 0.User can use ECAT_SetSlaveRxPdoData to set Control oputut module while Pid status is disabled.

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
PidNo	uint32_t	IN	PID Controller number
status	uint8_t	IN	Status 0:disable 1: enable

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.8. ECAT_PidGetSimulateMode

Description:

Get PID Controller Simulate.

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
PidNo	uint32_t	IN	PID Controller number
status	uint8_t*	Output	Status 0:Not enable 1: enable

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.9. ECAT_PidSetSimulateMode

Description:

Set PID Controller Simulate. Warn:Turn disable to enable will not set Control oputut module to 0; Turn enable to disable will set Control oputut module to 0.User can use ECAT_SetSlaveRxPdoData to set Control oputut module while Pid status is disabled or Pid Simulate is enabled.

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
PidNo	uint32_t	IN	PID Controller number
status	uint8_t	IN	Status 0:Not enable 1: enable

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.10. ECAT_PidGetParameter

Description:

Get PID Controller Parameter.

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
PidNo	uint32_t	IN	PID Controller number
kp	double *	Output	PID Parameter
ki	double *	Output	PID Parameter
kd	double *	Output	PID Parameter

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.11. ECAT_PidSetParameter

Description:

Set PID Controller Parameter.

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
PidNo	uint32_t	IN	PID Controller number
kp	double	IN	PID Parameter
ki	double	IN	PID Parameter
kd	double	IN	PID Parameter

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.12. ECAT_PidGetProcessVariableModule

Description:

Get PID Controller Process Variable Module information

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.13. ECAT_PidSetProcessVariableModule

Description:

Set PID Controller Process Variable Module information

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
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".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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,Scaleg
ain_input,Scaleoffset_input);
if(ret != 0)
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
{
    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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.14. ECAT_PidGetControlOutputModule

Description:

Get PID Controller Control Output Module information

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
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	Minimum

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
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,&Scaleoffs
et, &Max_Value, &Min_Value);
if(ret != 0)
{
    printf("Failed to Get Pid Output:%d\n",ret);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.15. ECAT_PidSetControlOutputModule

Description:

Set PID Controller Control Output Module information

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
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	Minimum

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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,Scaleg
ain_input,Scaleoffset_input);
if(ret != 0)printf("Failed to Set Pid Input:%d\n",ret);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
ret=ECAT_PidSetControlOutputModule(DeviceNo,PidNo,SlaveNo_output,Offset_output,Bitsize_output,Scale
gain_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);
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.16. ECAT_PidGetControlOutputValue

Description:

Get PID Controller Output Value.

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
PidNo	uint32_t	IN	PID Controller number
Output	double*	Output	PID Controller Output Value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.17. ECAT_PidGetSimulateFeedback

Description:

Get PID Controller Simulate model Output Value.

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	PID Controller Simulate model Output Value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

7.9.18. ECAT_PidGet_Sp_Err_Op_Pv

Description:

Get PID Controller Set Point、Error、Output、Process Variable

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
PidNo	uint32_t	IN	PID Controller number
SetPointValue	double*	Output	PID Controller Set Point Value
Error	double*	Output	PID Controller Error,defined as SP-PV
OutputValue	double*	Output	PID Controller Output Value
ProcessVariable	double*	Output	PID Controller ProcessVariable

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

```
{
    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);
}
```

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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 numer
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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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 positon parameter

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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_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	Faild 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	Faild to create driver handle
ECAT_ERR_DRV_IOCTL	-1303	Call driver IO control error
ECAT_ERR_DRV_DEVICE_NOT_FOUND	-1304	Device not found

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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	Gerneral 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	Subindex 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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

	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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.

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

Copyright © 2018 ICP DAS CO., Ltd. All Rights Reserved.