# EtherCAT Master Software Manual

English　　　Ver. 1.0.24, Aug. 2021

## CONTACT US

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

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

## SUPPORT

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

# Contents

# 1. Introduction

For developing applications on EtherCAT Master series cards, ICP DAS provides users with a shared library libecatdevice (.dll) to support the use in Windows operating systems. It provides powerful, easy-to-use functions for developing applications and speed-up the developing process . The library architecture is shown in the following figure. The user programs are developed on PC. PC is communicated with ECAT-M801/ e-M901 via APIs which use DRPRM (dual-port RAM) as the bridge.
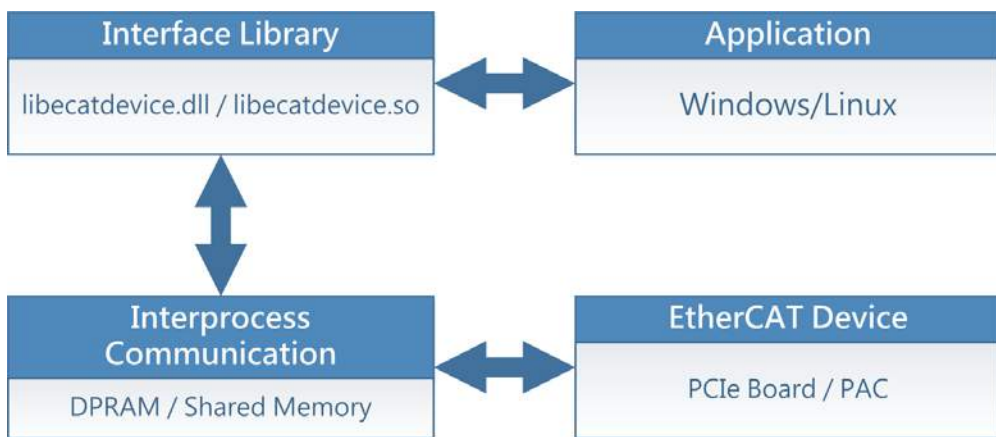


Chapter 2 is about installing software in a PC.

Chapter 3 introduces how to use utility for configuring the system and do some function tests.

Chapter 4 introduces some application developing concepts and settings for Visual studio.

Chapter 5 talking about how to open/close card and how to start an EtherCAT system and get its connection status. Local I/O operations are also mentioned.

Chapter 6 is about how to read/write objects of a slave. Both SDO and PDO communication methods are addressed. For simple I/O slaves, some simple APIs are provided. For complex slaves, such as AI/AO and Encoder interface modules, more functions are provided for configuration.

Chapter 7 includes a lot of functions. There are single axis motion functions, homing and group motion functions. Other functions are also provided here, such as PID control loops, Steward Platform Controller, motion data logger. A very power Event method in this system is also described here. Use Event method can let the PC loading reduce dramatically, and let the system respond faster.

## 1.1. Version update information

| Function modification | Version |
|---|---|

| | |
|---|---|
| 1. Cam Utility available | V1.0.08 or above |
| 1. Fixed the bug that when using ECAT2610, it is possible to enter the OP but not operate the PDO<br>2. Support Multi-axes driver<br>3. Support CiA402 Profile motion mode (PP, PV, PT)<br>4. Supports three second-order software filters (low-pass, high-pass, and notch)<br>5. Support Slave to Slave communication(topology independent) | V1.0.15 or above |
| 1. Modify the PVT algorithm<br>2. Added PVT deceleration stop<br>3. Support Motion Done Event<br>4. Supports Slave operation with Alias<br>5. Modify Gantry to cross-coupled control Gantry | V1.0.16 or above |
| 1. Advanced PDO Editing available<br>2. CiA402 Mapping Mode: "User Define" available | V1.0.17 or above |
| 1. Support ECAT_AbortWaitforEvent<br>2. Fixed the problem that some modules cannot read the module name using ECAT_GetSlaveInfo<br>3. Modify some APIs of ECAT-2092T and add related APIs of ECAT-2092T Array compare<br>4. Added coordinate conversion (rectangular coordinate command to polar coordinate mechanism) function<br>5. Added setting of DI active level API for CiA402 driver<br>6. Added API for reading DI (object 0x60fd) of CiA402 driver<br>7. Fixed the bug that the command is overwritten by the next command when writing DO continuously | V1.0.18 or above |
| 1. Added latch buffer for ECAT-2093<br>2. Added speed and acceleration unit conversion parameters<br>3. Add INP detection<br>4. Added asymmetric acceleration and deceleration (for ECAT-2091S, ECAT-2094S) | V1.0.19 or above |
| 1. Added DO operation of multiple slaves | V1.0.22 or above |

| 1. Support multi-domains | V1.0.23 or above |
|---|---|
| 1. Improve communication quality | V1.0.24 or above |

# 2. Software Installation

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

## 2.1. Obtaining the Driver Installer Package

How to search for drivers, manuals and spec information on ICP DAS website.

- For Mobile Web
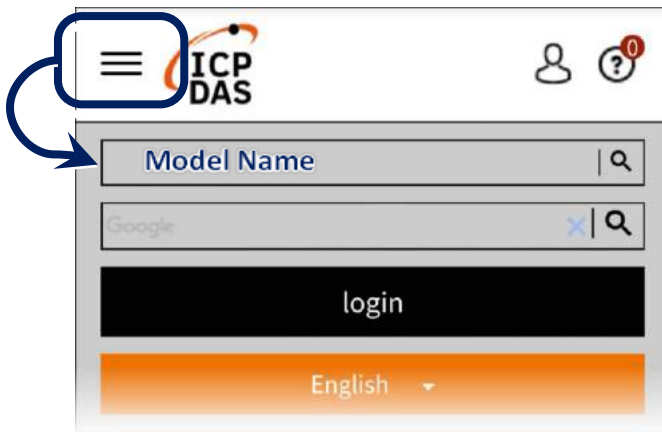


- For Desktop Web
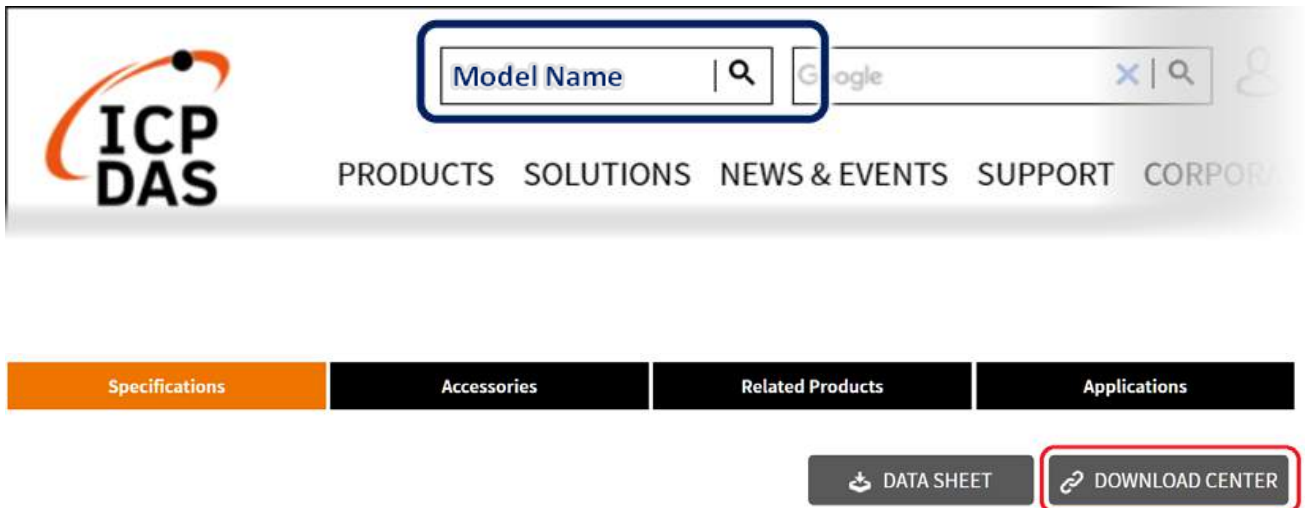
## 2.2. Driver Installing Procedure

This software includes API, Utility and driver installation

To install drivers, follow the procedure described below:

1. Double-Click "ECAT-Master_vx.x.xx_Windows_setup.exe" to install driver.

ECAT-M801
v1.3_Win7_x
64_Setup.ex
e

2. Choose whether to install dll to System32/ SysWOW64, Click the "Next >" button.

3. The default path is C:\icpdas\Ecat-M801 and C:\icpdas\e-M901, Click the "Install" button to continue.
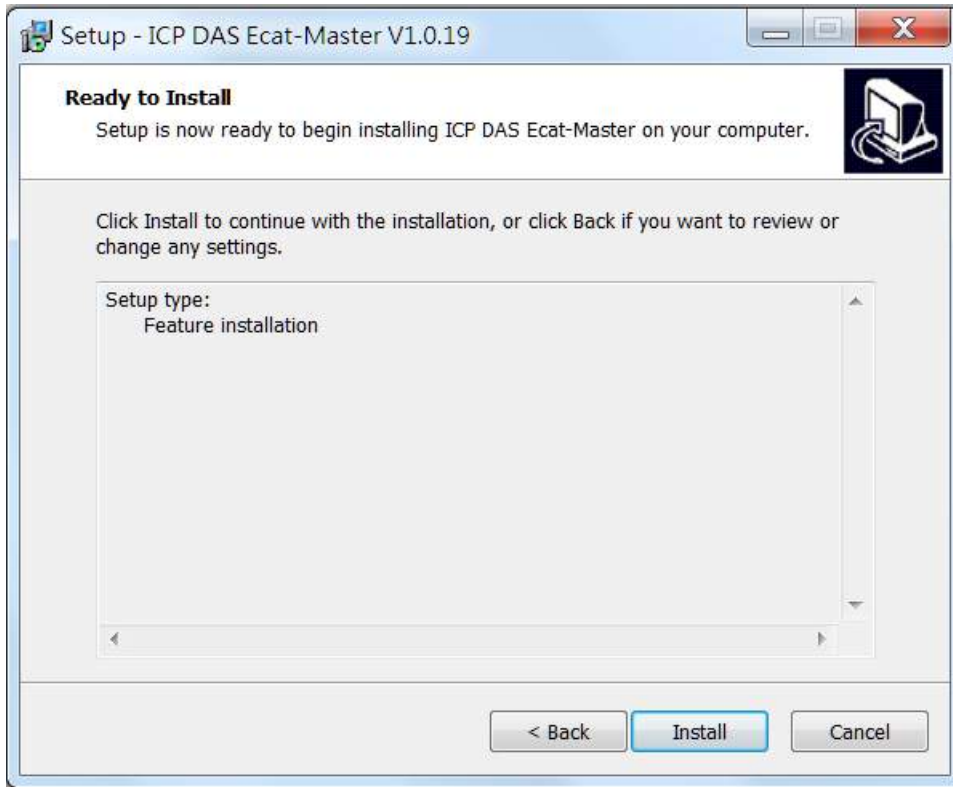


4. Check "I agree to the license terms and conditions", then click the "Install" button to continue. If the following screen does not appear, it means there is no need to install this part
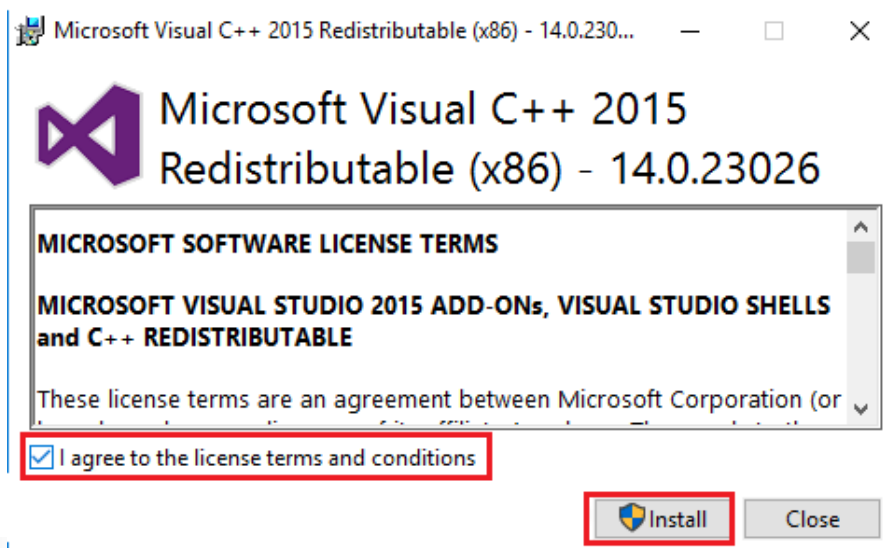
5. Click the "Close" button to continue.
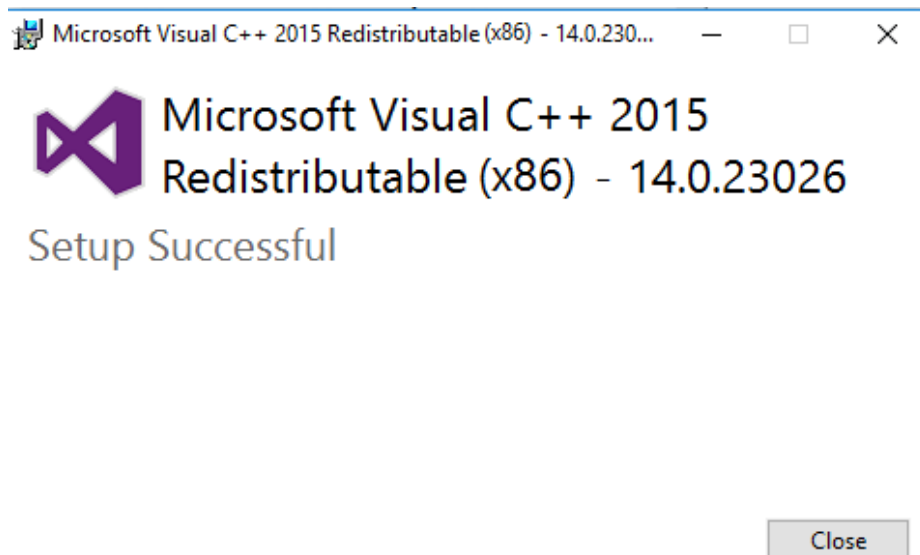


6. Click "Next >"

   Note: The version of V1.0.18 or below requires the UniDAQ driver

   If the following screen does not appear, it means there is no need to install this part

7.  Click "Next >"
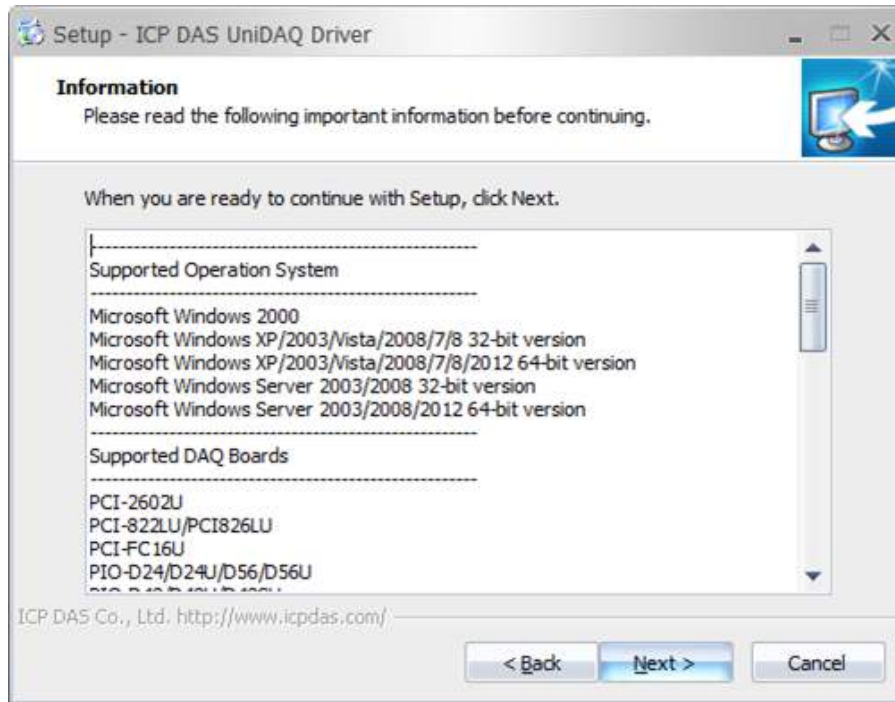


8.  Click "Next >"
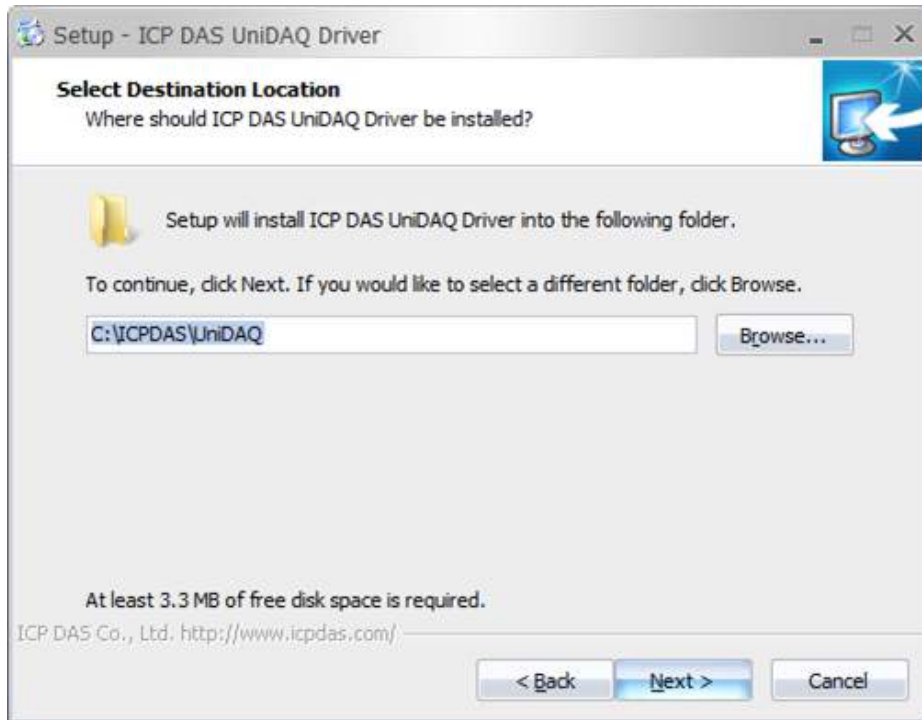
9.   Click "Next >"
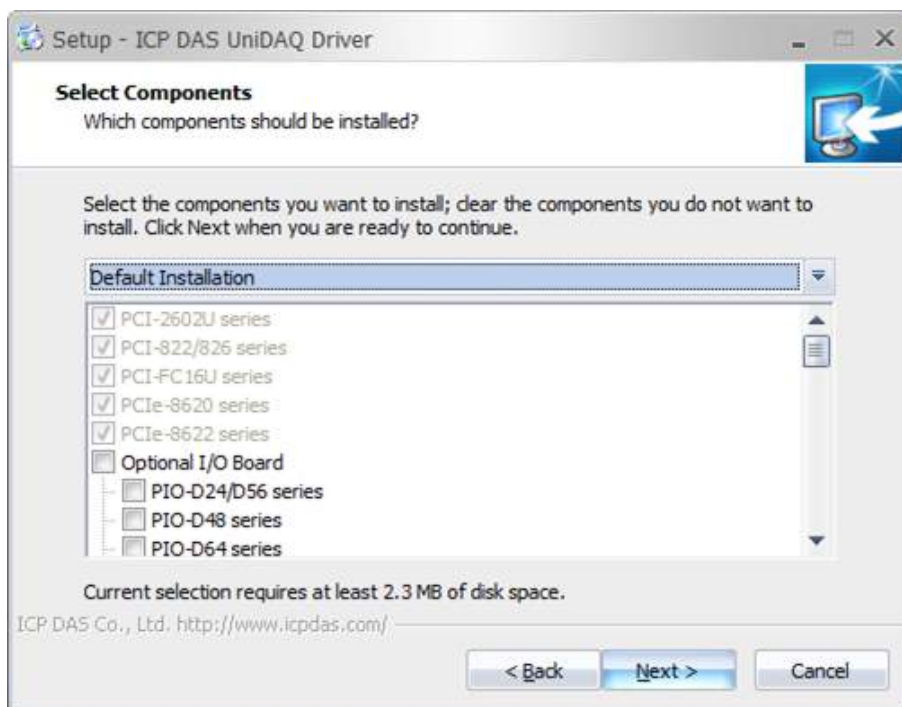


10.  Click "Next >"
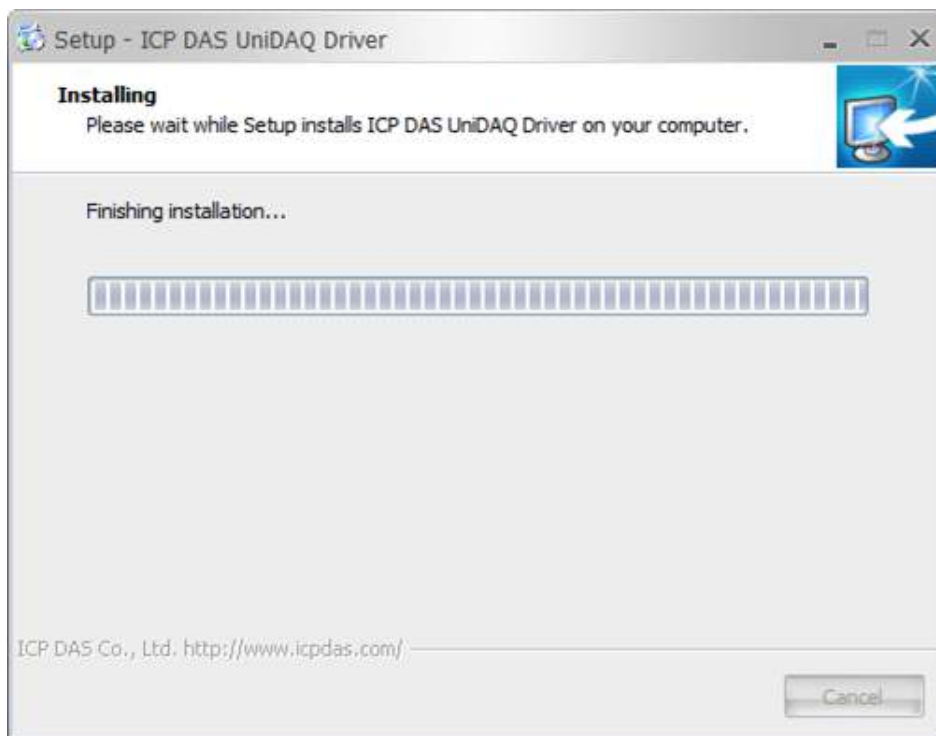
11. Click "Next >"
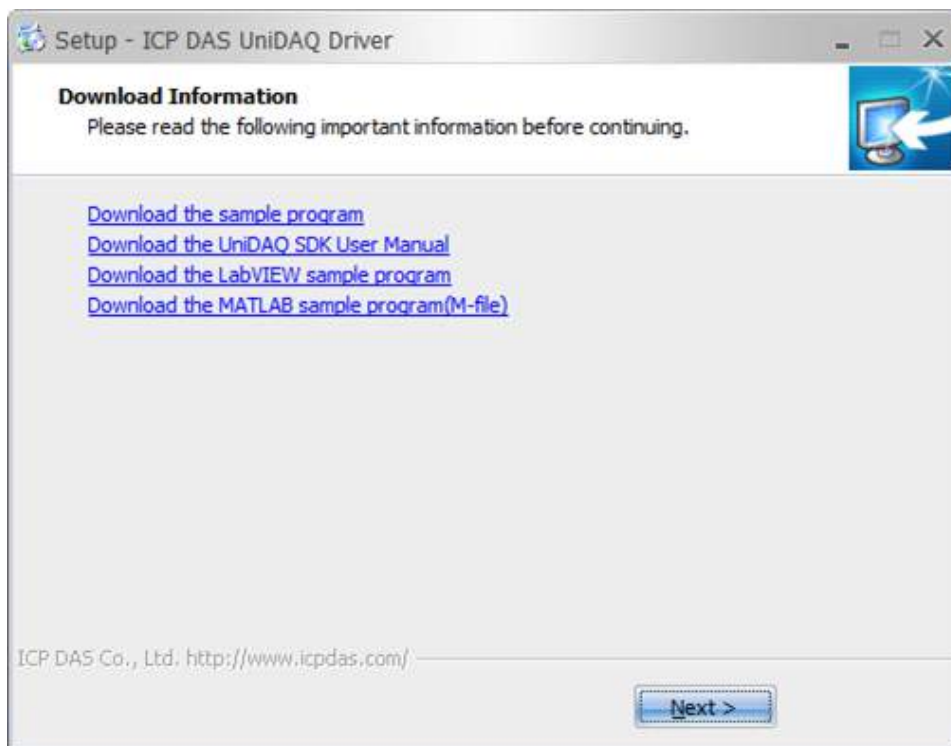
12. Click "Finish"



13. Click the "Finish" button and restart the computer.
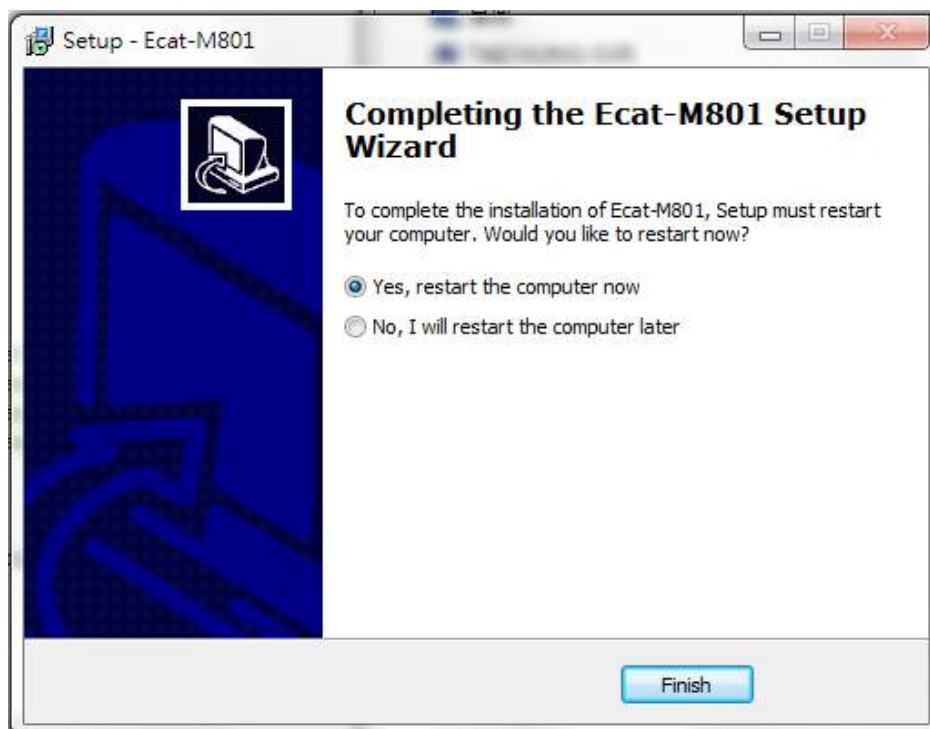
# 2.3.  Uninstalling the Driver

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

1.   Select **Settings >> Apps** from the Windows **Start** menu.
2.   Highlight the item ICP DAS Ecat-Master Vx.x.xx and then click Uninstall.
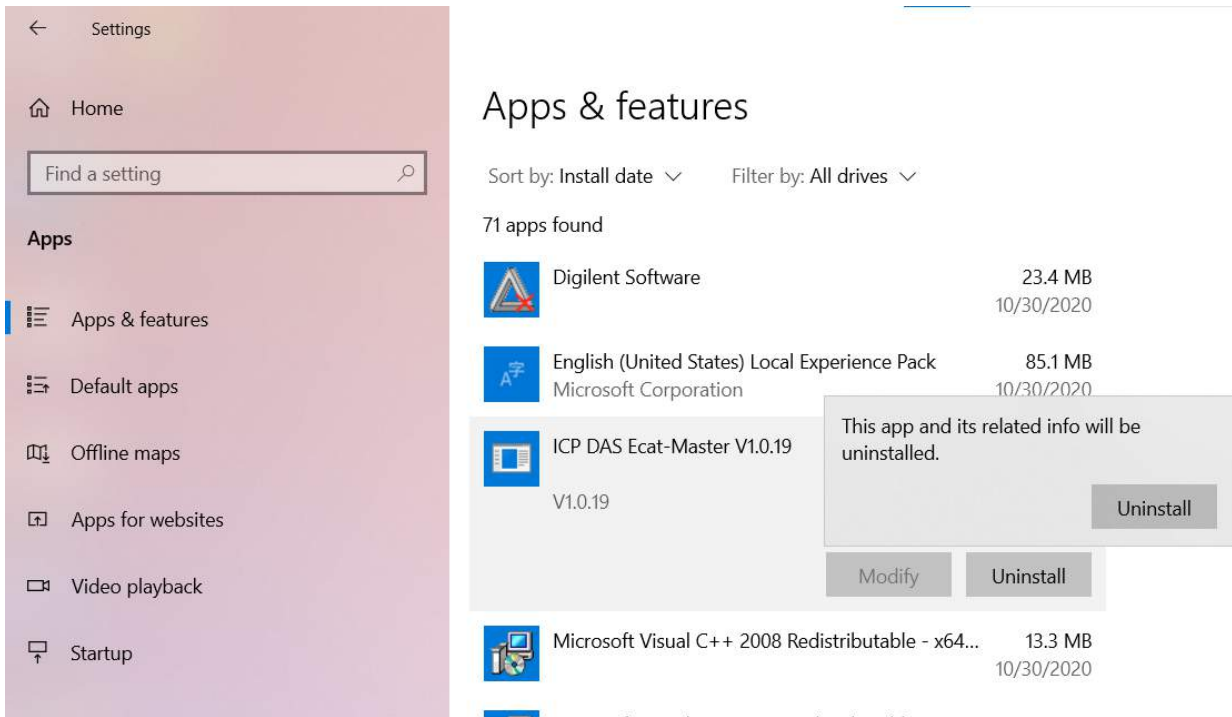


3.   When the message box loads, click the Yes(Y) button to uninstall the software.
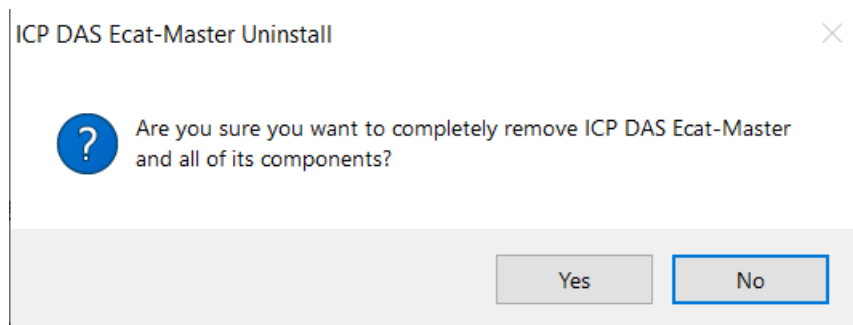
4.  After the uninstall process is complete, a dialog box will be displayed to you that your computer must be restarted. Click the "Yes" button to finish the uninstall process.

ICP DAS Ecat-Master Uninstall ✕

? To complete the uninstallation of ICP DAS Ecat-Master, your computer must be restarted.

Would you like to restart now?

Yes     No
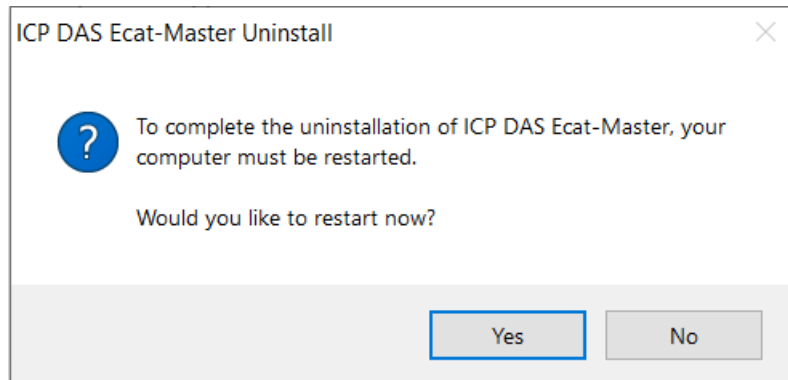
# 2.4. Installing the Linux driver

## 2.4.1. Installing the Linux driver

1. Extract the "ecat_m801_linux_setup_vx.xx.xx.tar.gz" file.

```
bryan@icpdas-mint-ibpc:~/workspace/gg$ tar xvf ecat_m801_linux_setup_v1.0.15.tar.gz
./ecat_m801_linux_setup/
./ecat_m801_linux_setup/libtool
./ecat_m801_linux_setup/.cproject
./ecat_m801_linux_setup/Makefile.am
./ecat_m801_linux_setup/drivers/
./ecat_m801_linux_setup/drivers/_ecat.c
./ecat_m801_linux_setup/drivers/Makefile.am
./ecat_m801_linux_setup/drivers/Makefile
./ecat_m801_linux_setup/drivers/ixecat.remove
./ecat_m801_linux_setup/drivers/ixecat.inst
./ecat_m801_linux_setup/drivers/Makefile.in
./ecat_m801_linux_setup/drivers/_proc.c
./ecat_m801_linux_setup/drivers/Kbuild
./ecat_m801_linux_setup/drivers/_pciecat.c
./ecat_m801_linux_setup/drivers/Kbuild.in
./ecat_m801_linux_setup/Makefile
./ecat_m801_linux_setup/COPYING
./ecat_m801_linux_setup/m4/
./ecat_m801_linux_setup/m4/ltversion.m4
./ecat_m801_linux_setup/m4/ltoptions.m4
./ecat_m801_linux_setup/m4/lt~obsolete.m4
./ecat_m801_linux_setup/m4/ltsugar.m4
./ecat_m801_linux_setup/m4/libtool.m4
./ecat_m801_linux_setup/aclocal.m4
./ecat_m801_linux_setup/README
./ecat_m801_linux_setup/.settings/
./ecat_m801_linux_setup/.settings/language.settings.xml
./ecat_m801_linux_setup/autom4te.cache/
```

2. Enter "configure" in the terminal.

```
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking for g++... g++
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
checking dependency style of g++... gcc3
checking for ar... ar
checking the archiver (ar) interface... ar
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking how to print strings... printf
checking for a sed that does not truncate output... /bin/sed
```

3. Enter "make modules" again in the terminal.

```
config.status: creating script/remove
config.status: creating script/Makefile
config.status: creating drivers/Kbuild
config.status: creating drivers/Makefile
config.status: creating include/Makefile
config.status: creating lib/Makefile
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ make modules
make -C "/lib/modules/`uname -r`/build" M="/home/bryan/workspace/gg/ecat_m801_linux_setup" m
odules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-20-generic'
Makefile:976: "Cannot use CONFIG_STACK_VALIDATION=y, please install libelf-dev, libelf-devel
 or elfutils-libelf-devel"
  CC [M]  /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/_ecat.o
  CC [M]  /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/_proc.o
  LD [M]  /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixecat.o
  CC [M]  /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/_pciecat.o
  LD [M]  /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixpciecat.o
  Building modules, stage 2.
  MODPOST 2 modules
  CC      /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixecat.mod.o
  LD [M]  /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixecat.ko
  CC      /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixpciecat.mod.o
  LD [M]  /home/bryan/workspace/gg/ecat_m801_linux_setup/drivers/ixpciecat.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-20-generic'
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$
```

4.  Finally enter "sudo make install" to install, the default installation path is in the "/opt/icpdas/ecat_m801" directory.

```
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ sudo make install
[sudo] password for bryan:
Making install in script
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
make[2]: Nothing to be done for 'install-exec-am'.
 /bin/mkdir -p '/opt/icpdas/ecat_m801/share/script'
 /usr/bin/install -c ecat_m801 ecat_m801.conf remove '/opt/icpdas/ecat_m801/share/script'
make  install-data-hook
make[3]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
cp /opt/icpdas/ecat_m801/share/script/ecat_m801 /etc/init.d/
update-rc.d ecat_m801 defaults
cp /opt/icpdas/ecat_m801/share/script/ecat_m801.conf /etc/ld.so.conf.d/
make[3]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/script'
Making install in drivers
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'
make  install-am
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'
make[3]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'
make[3]: Nothing to be done for 'install-exec-am'.
 /bin/mkdir -p '/opt/icpdas/ecat_m801/drivers'
 /usr/bin/install -c -m 644 ixecat.ko ixpciecat.ko '/opt/icpdas/ecat_m801/drivers'
 /bin/mkdir -p '/opt/icpdas/ecat_m801/drivers'
 /usr/bin/install -c ixecat.inst ixecat.remove '/opt/icpdas/ecat_m801/drivers'
make[3]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/drivers'
```

## 2.4.2.  Uninstalling the Linux driver

1.  Go to the "share/script" directory in the installation path.

```
more information, such as the ld(1) and ld.so(8) manual pages.
----------------------------------------------------------------
make  install-exec-hook
make[3]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
ldconfig
make[3]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
Making install in include
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[2]: Nothing to be done for 'install-exec-am'.
 /bin/mkdir -p '/opt/icpdas/ecat_m801/include'
 /usr/bin/install -c -m 644 EcatDeviceAPI.h '/opt/icpdas/ecat_m801/include'
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ cd /opt/icpdas/ecat_m801/share/sc
ript/^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ ^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ ^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ cd /home/
bryan@icpdas-mint-ibpc:/home$ cd /opt/icpdas/ecat_m801/share/script/
```

2.  Enter "sudo ./remove" in terminal to remove the driver and library. If there are no errors, the installation is successful.

```
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/lib'
Making install in include
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[2]: Nothing to be done for 'install-exec-am'.
 /bin/mkdir -p '/opt/icpdas/ecat_m801/include'
 /usr/bin/install -c -m 644 EcatDeviceAPI.h '/opt/icpdas/ecat_m801/include'
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup/include'
make[1]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[2]: Entering directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
make[1]: Leaving directory '/home/bryan/workspace/gg/ecat_m801_linux_setup'
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ cd /opt/icpdas/ecat_m801/share/sc
ript/^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ ^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ ^C
bryan@icpdas-mint-ibpc:~/workspace/gg/ecat_m801_linux_setup$ cd /home/
bryan@icpdas-mint-ibpc:/home$ cd /opt/icpdas/ecat_m801/share/script/
bryan@icpdas-mint-ibpc:/opt/icpdas/ecat_m801/share/script$ sudo remove
[sudo] password for bryan:
sudo: remove: command not found
bryan@icpdas-mint-ibpc:/opt/icpdas/ecat_m801/share/script$ sudo ./remove
bryan@icpdas-mint-ibpc:/opt/icpdas/ecat_m801/share/script$ ls
bryan@icpdas-mint-ibpc:/opt/icpdas/ecat_m801/share/script$ ls /opt/icpdas/
bryan@icpdas-mint-ibpc:/opt/icpdas/ecat_m801/share/script$
```

## 2.5. Update FirmWare

Warning: If the current ECAT-M801/ e-M901 FirmWare Ver. is 1.0.10 or below, Please follow the steps below and update twice.

(1) Install windows driver

(2) Open C:\icpdas\Ecat-M801\UpdateFirmware\Update_firmware.exe
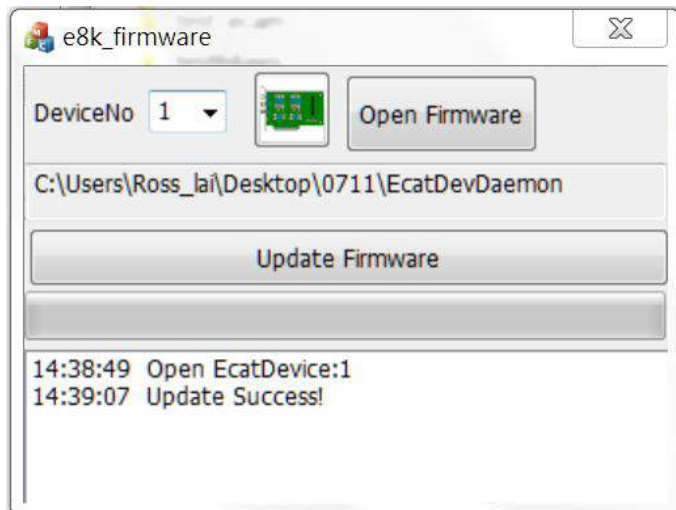


(2) Select Device [ 0 ▼ ], Clicked  to connect the device

(3) Clicked "Open Firmware", choose "EcatDevDaemon", Clicked "open file", If the file name is not "EcatDevDaemon", but the version number is added, such as "EcatDevDaemon_V1.0.17", please rename the file name to "EcatDevDaemon"
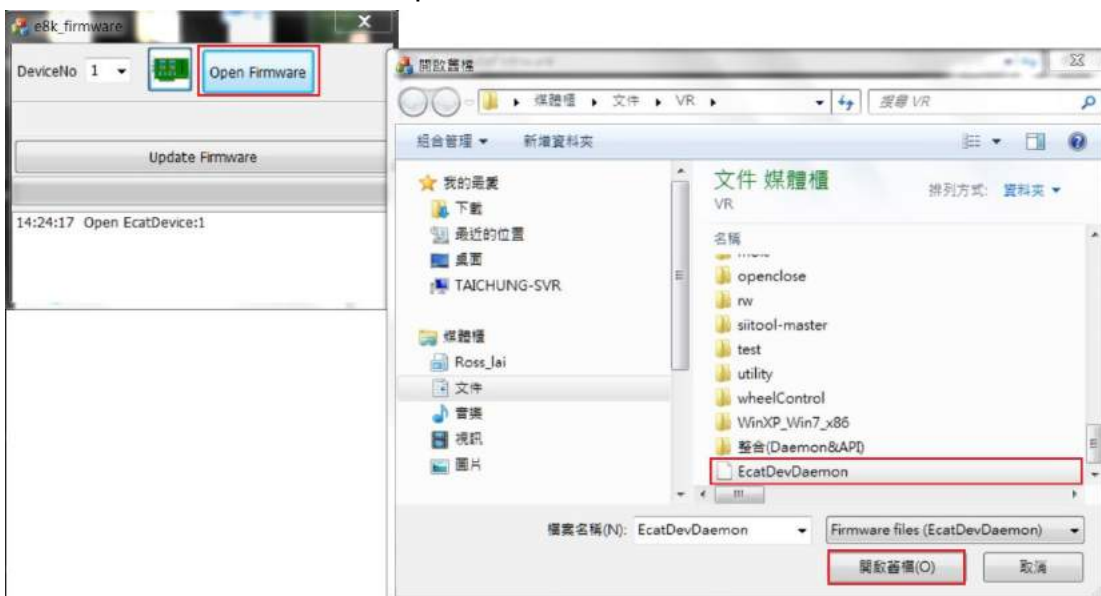
(4) Clicked "Update Firmware", update success

## 2.6. Auto Update

This software includes API, Utility and firmware updates

1.  Double-Click "ECAT-Master_vx.x.xx_Windows_AutoUpdate.exe" to install driver.

ECAT-Master-v
1.0.21_Window
s_AutoUpdate.e
xe
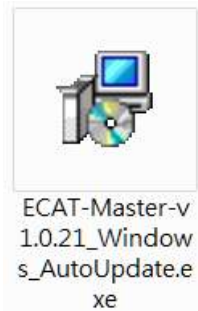
2.  The default path is C:\icpdas\Ecat-M801 and C:\icpdas\e-M901, Click the "Install" button

    to continue.
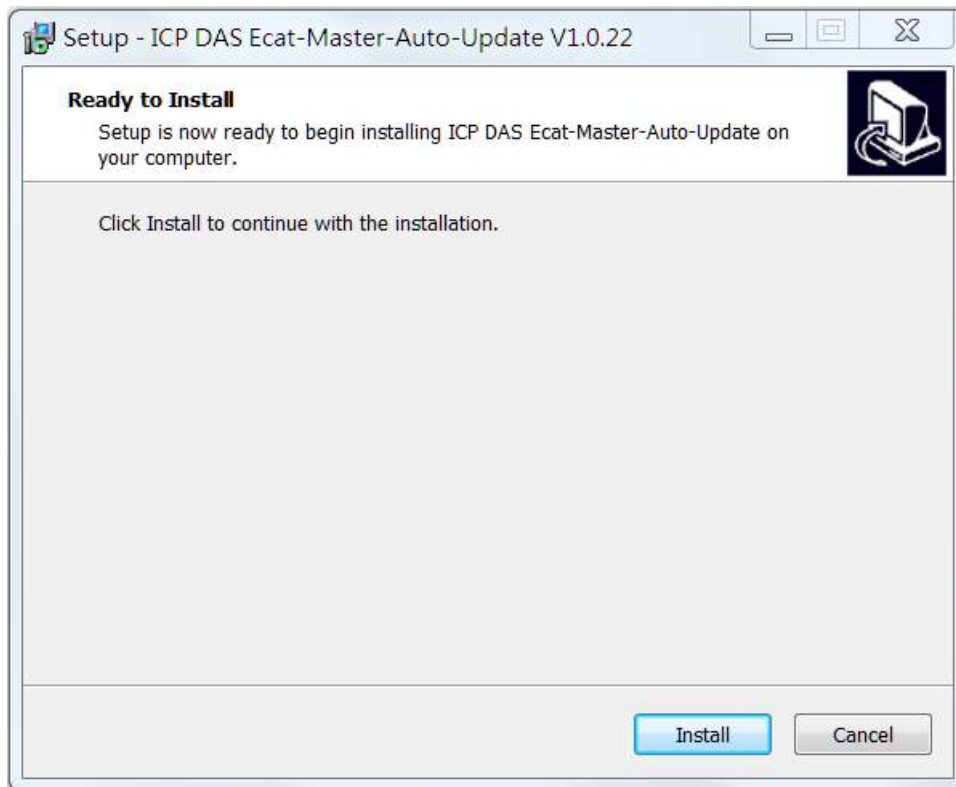
3.  After the installation is complete, the current version information will be displayed.。



4.  Click "Finish" to complete the installation
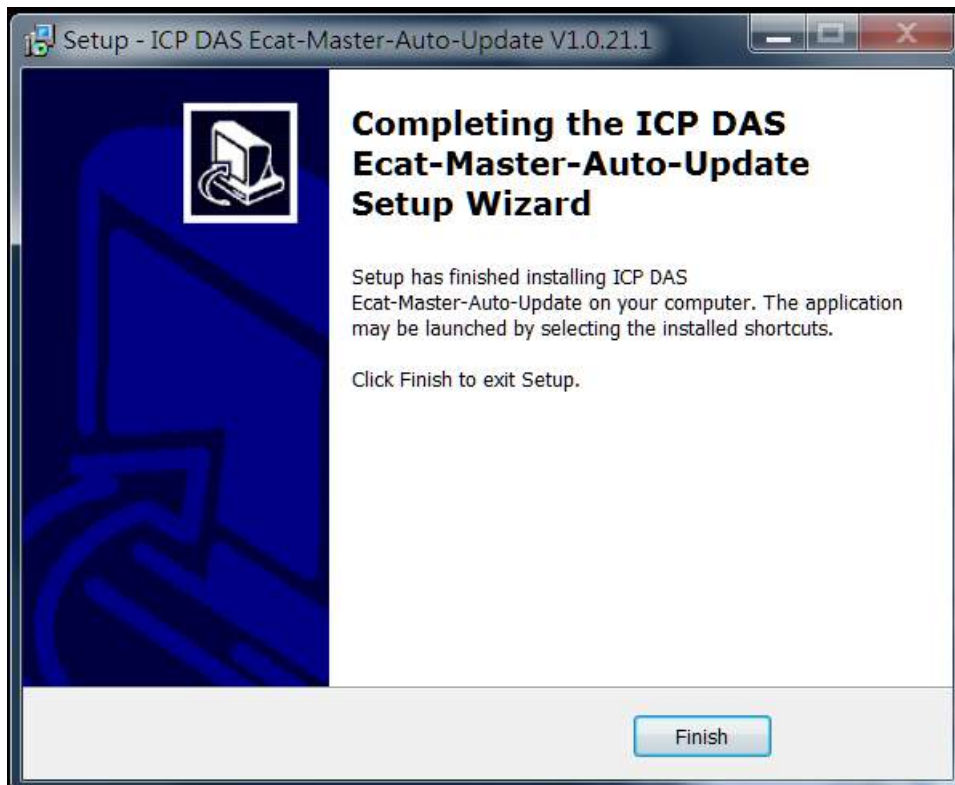
# 3. EcatUtility

Installed while installing driver.path: C:\icpdas\Ecat-M801\Utility\Utility.exe

EcatUtility is a software tool for users to use ECAT-M801/ e-M901 card on EtherCAT applications. It allows users to edit the device network information, to test slave modules, and to do motion control function tests. Start the Utility

## 3.0. Start the Utility

### 3.0.0. Connection

There are 2 network jacks of the module, namely IN and OUT. When connecting, please pay attention to:

The master station is connected to the IN of the first module; the OUT of the first module is connected to the IN of the second module, and so on.
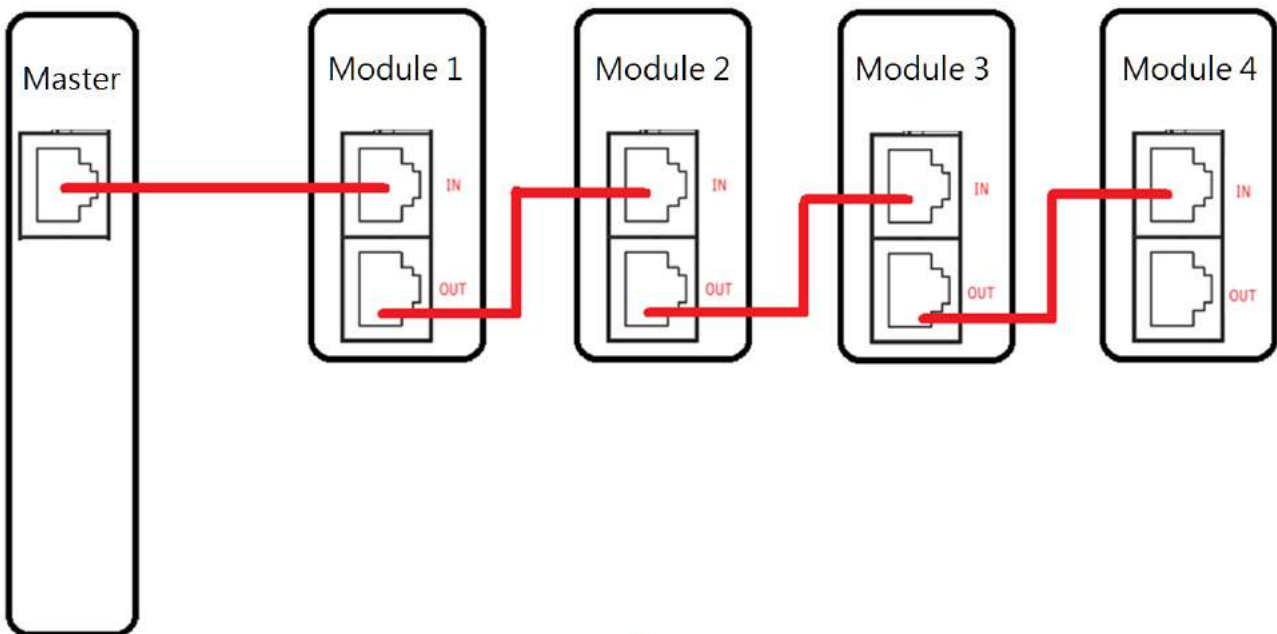
## 3.0.1.　Select Slave number definition

When starting Utility, you can choose the definition of the Slave number.
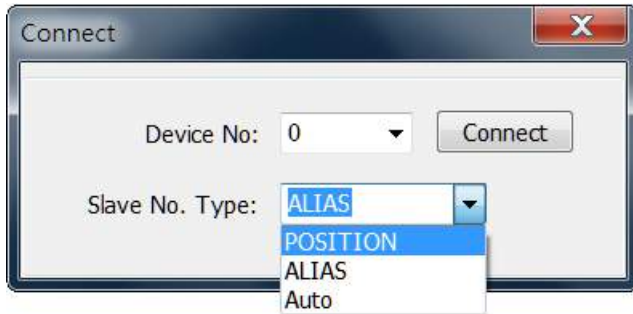


| Item | Description |
|---|---|
| (1) POSITION | The slave number is the position of the module.<br>Refers to the position of the module in the EtherCAT network architecture (Master-Module 0-Module 1...) |
| (2) ALIAS | The slave number is the module alias (Alias)<br>Not affected by module connection sequence, can be set by user<br>Range: 1~65534 |
| (3) Auto | The slave number is the module alias (Alias)<br>When "Auto" is selected, Network information(3.1.1 ~ 3.1.5) and Motion<br>Control Parameter File(3.5.1 ~ 3.5.2) will be created automatically |

Take Figure 3.1 as an example:

When the slave number type is POSITION, the slave number "1" refers to ECAT-2028

When the slave number type is ALIAS, the slave number "1" refers to ECAT-2011H
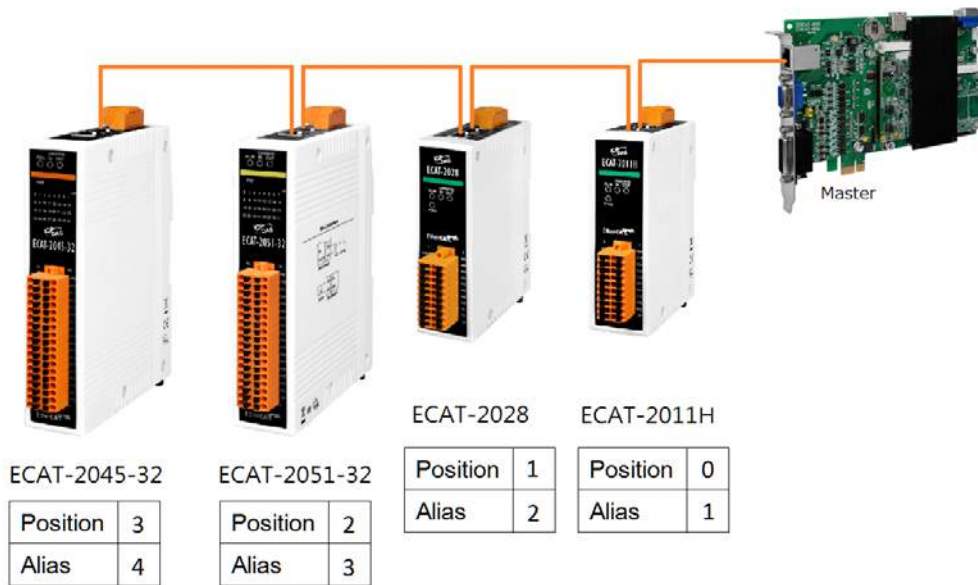
Figure 3.1

## 3.0.2. ALIAS Setting

When ALIAS mode is selected, it will enter the Alias Setting page.
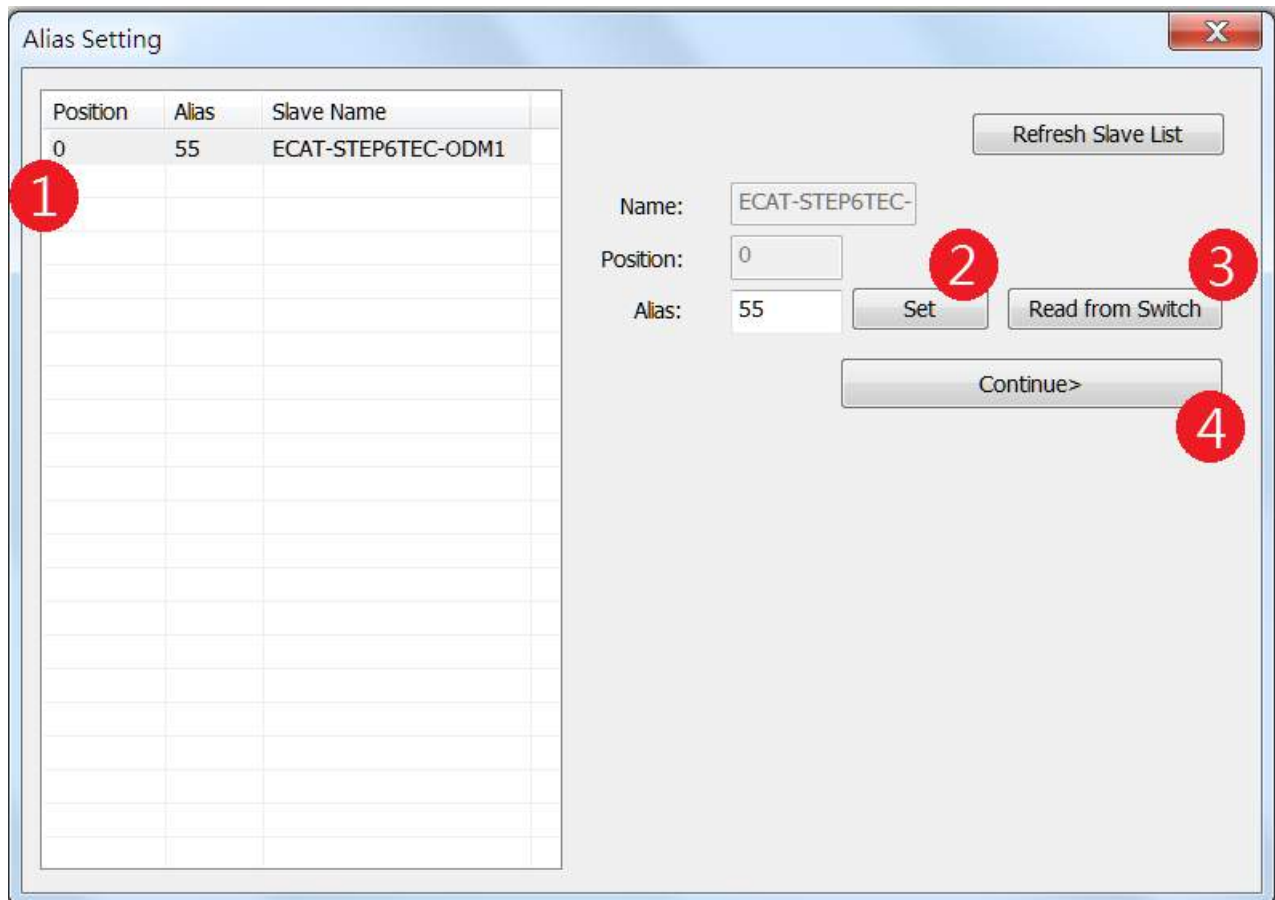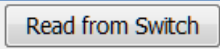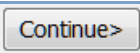


Figure 3.2

Table 3.1

| Item | Description |
|------|-------------|
| (1) | Slave list |
| (2) [Set] | Set Alias |
| (3) [Read from Switch] | Some modules have alias rotation switch, this button can read the switch value |
| (4) [Continue>] | After setting Alias, enter the Utility operation page |

### 3.0.3. ALIAS edit steps

When Alias conflicts or is out of the allowable range (1~65534), the following error will occur

## 3.0.4. Utility operation page

This software tool contains several parts shown in the following figure and table.
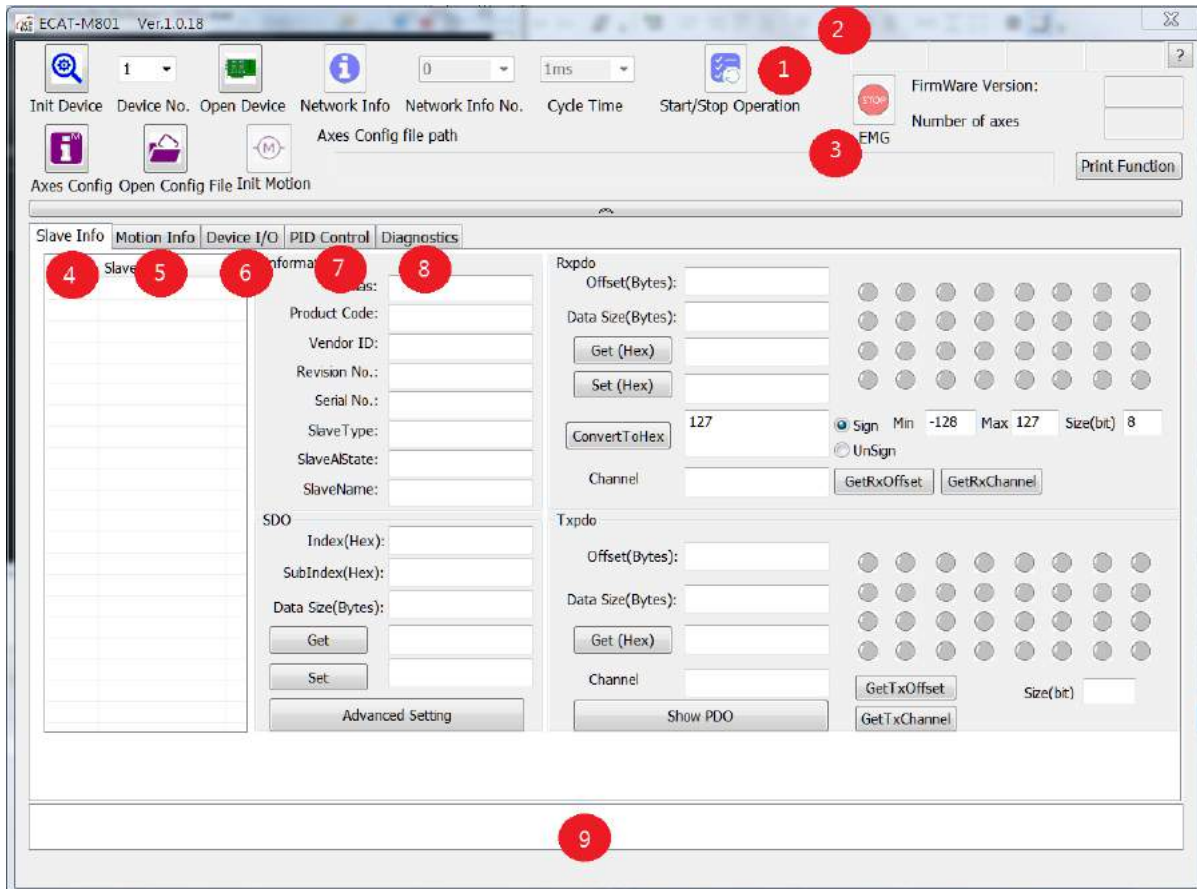


| Item | Description |
|------|-------------|
| (1) | Device operation toolbar |
| (2) | Device network status |
| (3) | Toolbar for the initialization of Motion Control |
| (4) | Slave Operation page |
| (5) | Motion control page |
| (6) | Device I/O operation page |
| (7) | PID operation page |
| (8) | EtherCAT diagnostics operation page |
| (9) | Message panel |

# 3.1.  Device Operation Toolbar

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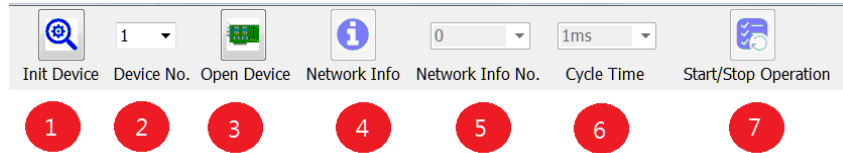


| Item | Description |
|---|---|
| (1) 🔍 | Get all the number of available devices and shown them in (3) |
| (2) [0 ▼] | Device number list (shown the chosen card number here) |
| (3) ▦ | Open or Close device which is chosen in (3) |
| (4) ⓘ | Network information edit page (for configuring slaves) |
| (5) [0 ▼] | Network information number list. The chosen file number will be used to store network information for downloading into ECAT-M801/ e-M901 card. |
| (6) [1000000 ▼] | Cycle time list. Unit is in micro-seconds. The chosen cycle time is used for cyclic communication when EtherCAT system goes into OP state. |
| (7) ▦ | Start or Stop the device EtherCAT operation task |

## 3.1.1.  Device Initialization Steps

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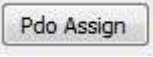
## 3.1.2.    Network Information Edit Steps

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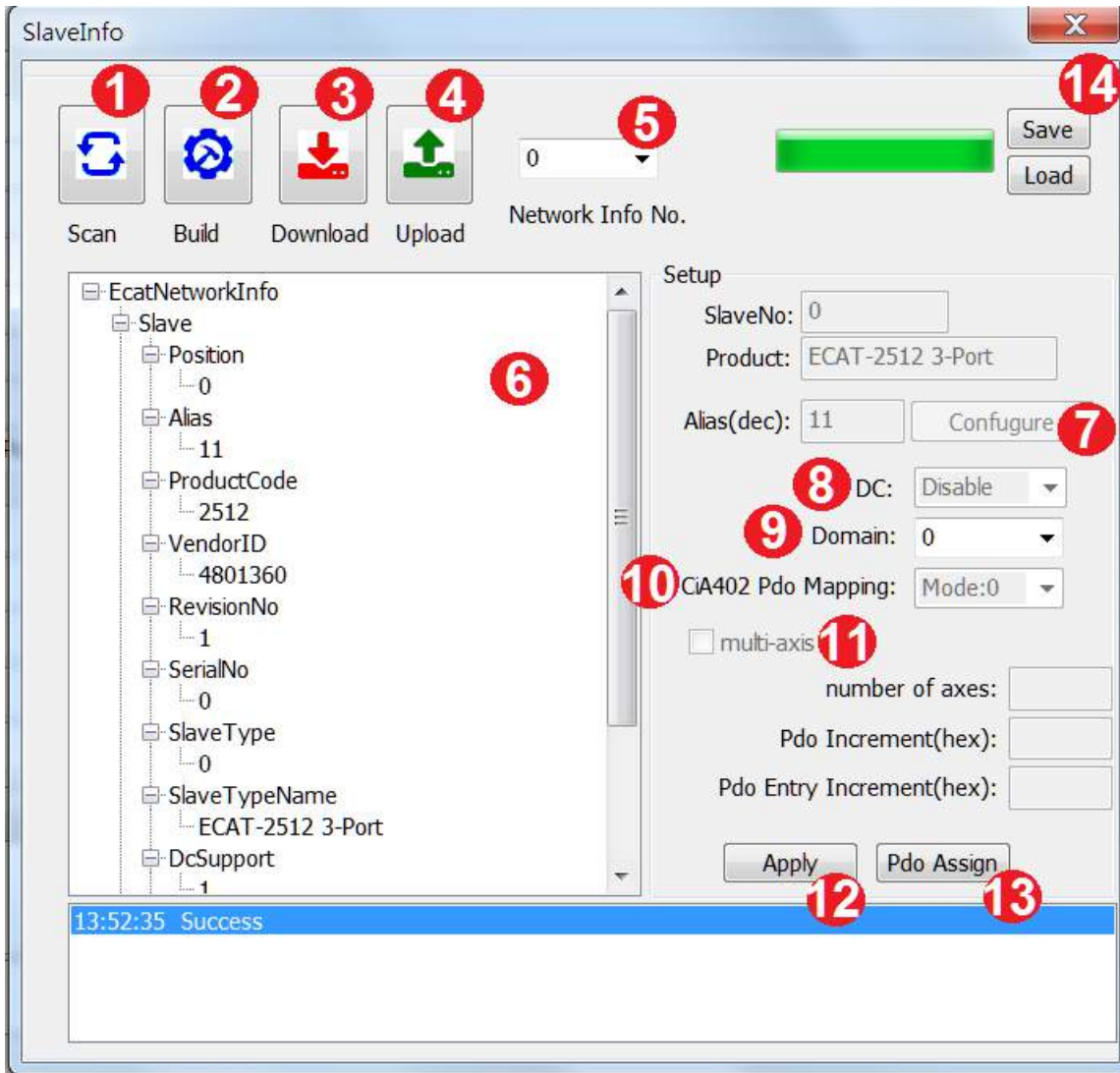


| Item | Description |
|---|---|
| (1) ⟳ | Scan EtherCAT network for finding slaves. |
| (2) ⚙ | Build network information from ESI files |
| (3) ⬇ | Send the network information into ECAT-M801/ e-M901 card. The network information file number is closed in (5). |

| | |
|---|---|
| (4) 🔼 | Retrieve network information from ECAT-M801/ e-M901 card. The network information file number is chose in (5). |
| (5) [0 ▾] | The file number (file name) of network information for sending and retrieving. |
| (6) | Network information panel |
| (7) [ Confugure ] | Open Alias Setting page. |
| (8) DC [Enable ▾] | DC setting of the selected slave. If the slave is capable of DC communication and meets the system cycle time setting, it can be set to **Enable**. |
| (9) Domain: [0 ▾] | Set the domain that the slave belongs to. |
| (10) CiA402 Pdo Mapping [Mode:0 ▾] | Do PDO assignment of CiA402 Model. |
| (11) □ multi-axis<br>number of axes: [ ]<br>Pdo Increment(hex): [ ]<br>Pdo Entry Increment(hex): [ ] | Do multi-axis settings of CiA402 Model.<br>Parameter Description:<br>➢ number of axes: Number of axes supported by the module<br>➢ Pdo Increment: Increment of each Pdos ,<br>Index of first axis is 1A00<br>While Index of second axis is 1A10,<br>then set to 10<br>While Index of second axis is 1A20,<br>then set to 10<br>➢ Pdo Entry Increment: Increment of each Pdo Entry,<br>Explain with Controlword<br>Index of Controlword of first axis is 6040h<br>While Index of Controlword of second axis is 6840h,<br>then set to 800<br>While Index of Controlword of second axis is7040h,<br>then set to 1000 |
| (12) [ Apply ] | Apply the configuration change of this slave. It must be |

| | |
|---|---|
| | apply before switching to edit other slaves; otherwise, system does not accept the change. |
| (13) Pdo Assign | Do PDO assignment. If users are familiar with EtherCAT technology and know how to do that, it is for them to change the default PDO assignment. |
| (14)<br>Save<br>Load | After editing the network information, you can save the current network information as a file. If you need to use the same network information on another card, you can directly read the previously saved file. While reading the file, please confirm whether to configure the Alias of the module or not. |
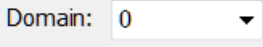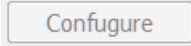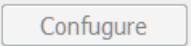
**Descriptions:**

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



Then the utility program will retrieve detailed slave information and show them on the info panel.
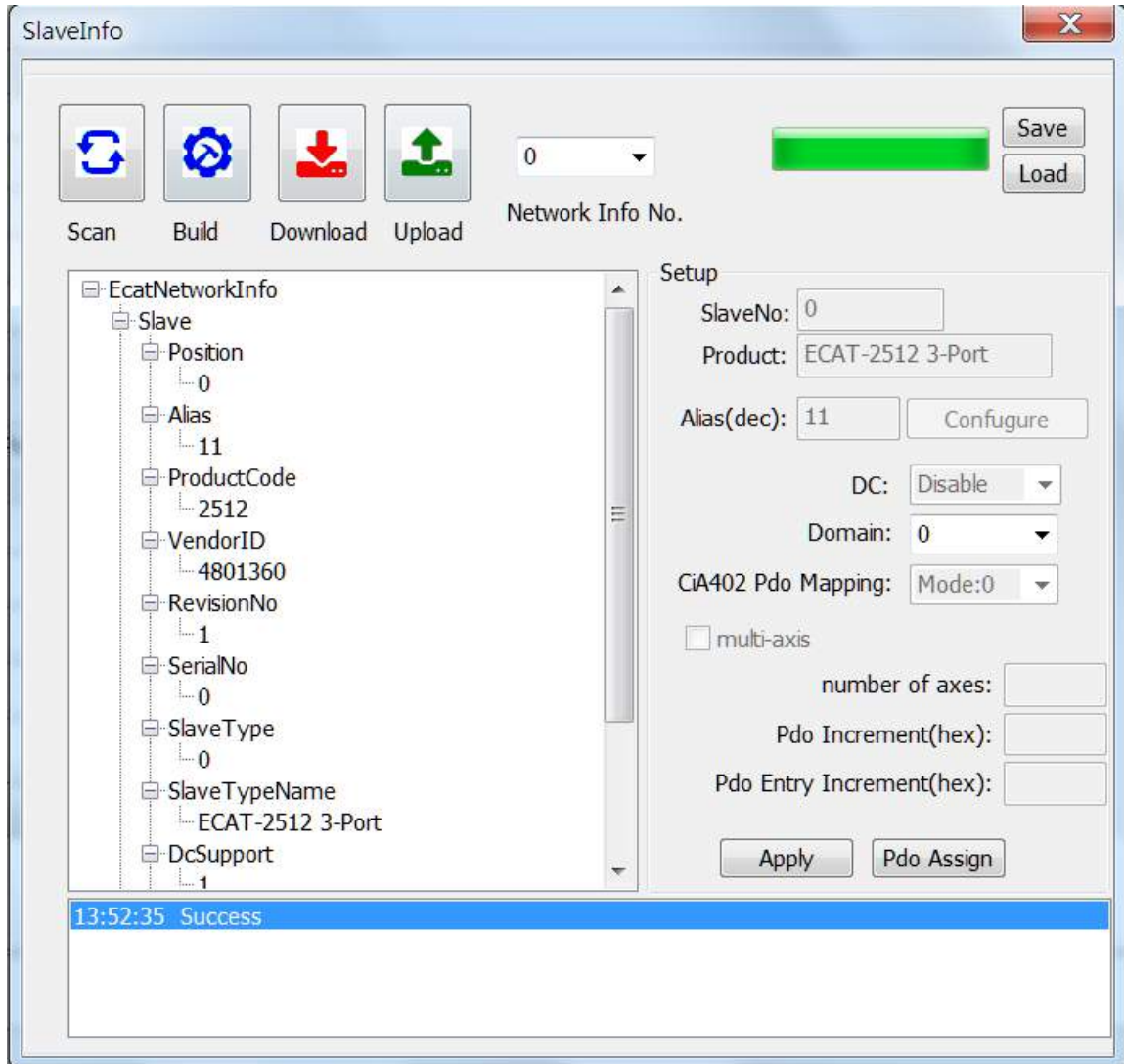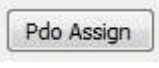
3.  Click  to select a slave. User can select its DC setting .

4.  Click  to select a slave. User can select its Domain setting 
    when setting the modules in different domains, when a working counter error occurs in
    one domain, the other domains will not be affected.

5.  Click  to open the ALIAS setting page. If the Alias value is modified,
    please power off/on the module with modified Alias first, and then clicks  again to
    scan the network information.
    When starting the Utility, the slave station number needs to be defined as "ALIAS" for
     to be effective.

6.  After setting all slaves, select a preferred network information number from the network

information list and click ![icon] to send it into the device (ECAT-M801/ e-M901 control card).

7.   If needed, a previous configuration file in the device can be retrieved by clicking ![icon].

### 3.1.3.   Network Information Edit Steps (PDO mapping)

Objects must be mapped for communications with process data objects (PDOs) to

exchange information in realtime with a fixed period. The above operations will use the

module's default PDO. If you only need to use the default PDO (such as general I / O), you

can skip this chapter.

Click ![Slave] to select a slave and ![Pdo Assign] provide user to assign PDO mapping. The descriptions of control items are shown in the following figure and table.
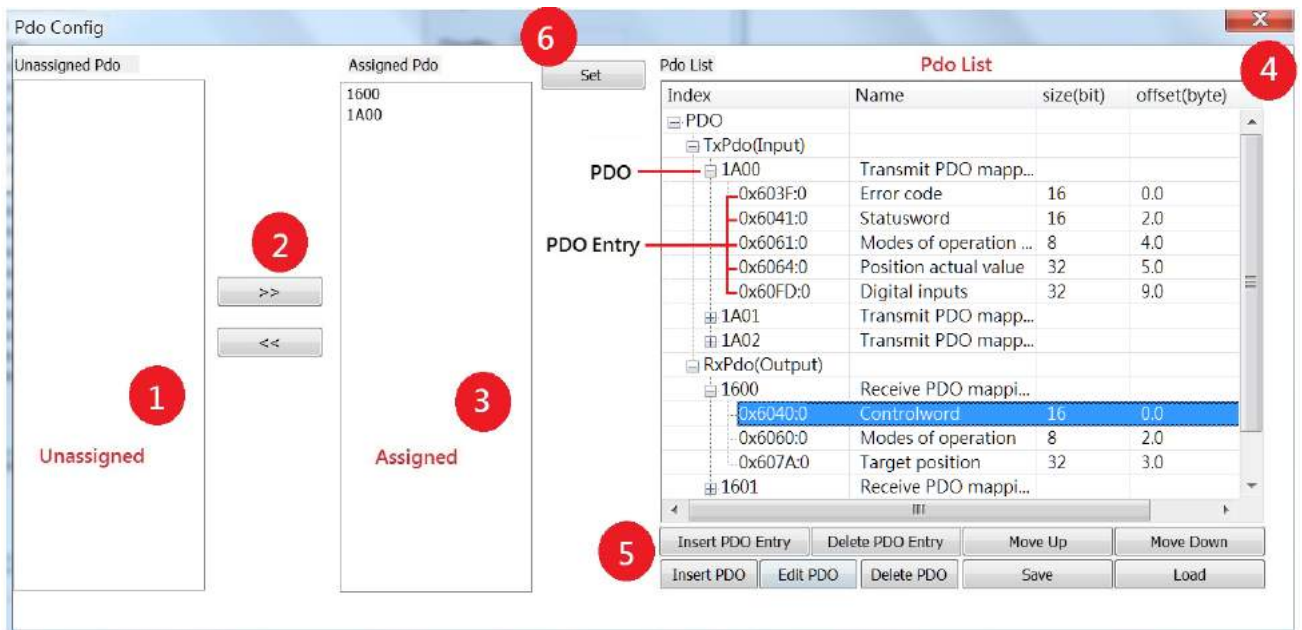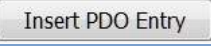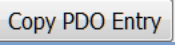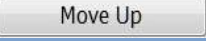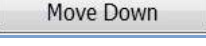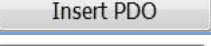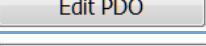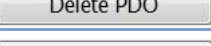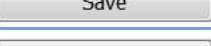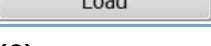


Figure 3.3

Table 3.2

| Item | Description |
|------|-------------|
| (1) | Unassigned Pdos |
| (2) | Assign Button |

| | |
|---|---|
| >> | Assign Pdo |
| << | Unassign Pdo |
| (3) | Assigned Pdos |
| (4) | Pdo List |
| (5) | Function Button |
| Insert PDO Entry | Insert a PDO Entry |
| Copy PDO Entry | Copy a PDO Entry |
| Delete PDO Entry | Delete a PDO Entry |
| Move Up | Move up a PDO Entry |
| Move Down | Move down a PDO Entry |
| Insert PDO | Insert a PDO |
| Edit PDO | Edit a PDO |
| Delete PDO | Delete a PDO |
| Save | Save Pdo List |
| Load | Load Pdo List |
| (6) | Set Button |

1.  Select the PDO to be configured in the "Unassigned Pdo" and "Assigned Pdo" areas, and click [>>] or [<<] to configure. The PDOs assigned to the "Assigned Pdo" area will be assigned to the module in order, and click "Set" to complete the setting of the module. After completing the setting, click ⬇.

2.  Before clicking the Set button, you can edit the content of the PDO. In the PDO List area, click the PDO or PDO Entry to be edited, and then click the "Function Button" to operate the selected PDO/PDO Entry.

3.  After editing the PDO List, you can click [Save] to save the edited PDO List. When you need to edit the PDO List again next time, you can click [Load] to read the edited PDO List and edit the PDO List, or use it for other same kind of modules.

Notice:

1.  If there is a display (Mandatory) or (Fixed) behind the PDO, the PDO cannot be edited,

for example: 1A01(Fixed)

2. Multiple PDOs are configured, please make sure that there is no duplicate PDO Entry in PDOs, otherwise you may not be able to enter Operatino Mode or work abnormally. The following figure is an example.

| | |
|---|---|
| ⊟ 1600 | 1st Receive PDO Ma... |
| ⊞ Exclude | |
| 0x6040:0 | Controlword |
| 0x607A:0 | Target position |
| 0x6060:0 | Modes of operation |
| ⊟ 1601 | 2nd Receive PDO M... |
| ⊞ Exclude | |
| 0x6040:0 | Controlword |

If 1600 and 1601 are configured in "Assigned Pdo", it can be found that 6040: 0 in 1600 and 6040: 0 in 1601 overlap, which may cause errors.

3. When adding PDO or PDO Entry, please make sure that the module supports the PDO / POD Entry added by the user

4. When adding a PDO Entry, please confirm that the PDO Entry belongs to TxPDO (input) or RxPDO (output)

## 3.1.4.   Network Information Edit Steps (PDO mapping for CiA402)

Provide 4 sets of PDO Mapping mode for CiA402 module. The default value is mode 0. Please refer to

1. If the module is a multi-axis CiA402 motor driver, make the following settings.

click ⊞ Slave , check ☑ multi-axis , enter Pdo Increment 、Pdo Entry Increment and

number of axes

Parameter Description:
 ➢ Pdo Increment: Increment of each Pdo,
   Index of first axis is 1A00h
   While Index of second axis is 1A10h, then set to 10
   While Index of second axis is 1A20h, then set to 20
 ➢ Pdo Entry Increment: Increment of each Pdo Entry, Explain with Controlword
   Index of Controlword of first axis is 6040h
   While Index of Controlword of second axis is 6840h, then set to 800

While Index of Controlword of second axis is7040h, then set to 1000

➢ number of axes: Number of axes supported by the module

2. Table 3.4 for the definition of each mode.

3. If you need to configure the PDO of the CiA402 module, click the module you want to change ⊞ Slave , and then click [CiA402 Pdo Mapping Mode:0 ▼] to configure it, and click Apply to confirm the change of the settings.
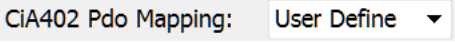
4. When the module is configured to unsupported PDO Entries, it may cause the module not to enter Operatino Mode or work abnormally. Please make sure that the module supports all PDO Entries in the selected PDO Mapping mode.If the module does not support the currently selected PDO Mapping Mode, please change to another PDO Mapping Mode, or change to [CiA402 Pdo Mapping: User Define ▼] and follow "Network Information Edit Steps (PDO mapping)" for advanced configuration.

5. When changing to, you must include the PDO Entries in Table 3.3; otherwise you will not be able to enter Operatino Mode.

Table 3.3

| RxPdo Entries | | TxPdo Entries | |
|---|---|---|---|
| 6040 | Controlword | 6041 | Statusword |
| 6060 | Modes of operation | 6061 | Modes of operation display |

6. If the module is a multi-axis CiA402 motor driver, make the following settings.

click ⊞ Slave , check ☑ multi-axis , enter Pdo Increment 、Pdo Entry Increment and

number of axes

Parameter Description:

➢ Pdo Increment: Increment of each Pdo,

Index of first axis is 1A00h

While Index of second axis is 1A10h, then set to 10

While Index of second axis is 1A20h, then set to 20

➢ Pdo Entry Increment: Increment of each Pdo Entry, Explain with Controlword

Index of Controlword of first axis is 6040h

While Index of Controlword of second axis is 6840h, then set to 800

While Index of Controlword of second axis is7040h, then set to 1000

➢ number of axes: Number of axes supported by the module

Table 3.4

|  | RxPDO | | TxPDO | |
|---|---|---|---|---|
| Mode0 | 6040 | Controlword | 6041 | Statusword |
|  | 6060 | Modes of operation | 603F | Error code |
|  | 607A | Target Position | 6061 | Modes of operation display |
|  | 60FF | Target Velocity | 6064 | Position actual value |
|  | 6071 | Target Torque | 606C | Velocity actual value |
|  | 60B8 | Touch probe function | 60FD | Digital inputs |
|  | 60B0 | Position offset | 6077 | Torque actual value |
|  | 60B1 | Velocity offset |  |  |
|  | 60B2 | Torque offset |  |  |
| Mode1 | 6040 | Controlword | 6041 | Statusword |
|  | 6060 | Modes of operation | 603F | Error code |
|  | 607A | Target Position | 6061 | Modes of operation display |
|  | 60FF | Target Velocity | 6064 | Position actual value |
|  | 6071 | Target Torque | 606C | Velocity actual value |
|  | 60B8 | Touch probe function | 60FD | Digital inputs |
|  | 60B0 | Position offset | 6077 | Torque actual value |
|  | 60B1 | Velocity offset |  |  |
|  | 60B2 | Torque offset |  |  |
| Mode2 | 6040 | Controlword | 6041 | Statusword |
|  | 6060 | Modes of operation | 603F | Error code |
|  | 607A | Target Position | 6061 | Modes of operation display |
|  | 60FF | Target Velocity | 6064 | Position actual value |
|  | 6071 | Target Torque | 606C | Velocity actual value |

| | 60B8 | Touch probe function | 60FD | Digital inputs |
| | 60B0 | Position offset | 6077 | Torque actual value |
| | 60B1 | Velocity offset | | |
| | 60B2 | Torque offset | | |
| Mode3 | 6040 | Controlword | 6041 | Statusword |
| | 6060 | Modes of operation | 603F | Error code |
| | 607A | Target Position | 6061 | Modes of operation display |
| | 60FF | Target Velocity | 6064 | Position actual value |
| | ~~6071~~ | ~~Target Torque~~ | 606C | Velocity actual value |
| | ~~60B8~~ | ~~Touch probe function~~ | 60FD | Digital inputs |
| | ~~60B0~~ | ~~Position offset~~ | ~~6077~~ | ~~Torque actual value~~ |
| | ~~60B1~~ | ~~Velocity offset~~ | | |
| | ~~60B2~~ | ~~Torque offset~~ | | |

### 3.1.5. Start/Stop the EtherCAT Operation Task Steps

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

## 3.2. Message Panel

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

```
09:08:54  User timeout
09:08:54  Failed to open EcatDevice:-1304
```

## 3.3.　Device Network Status

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



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

## 3.4. Slave Operation Page

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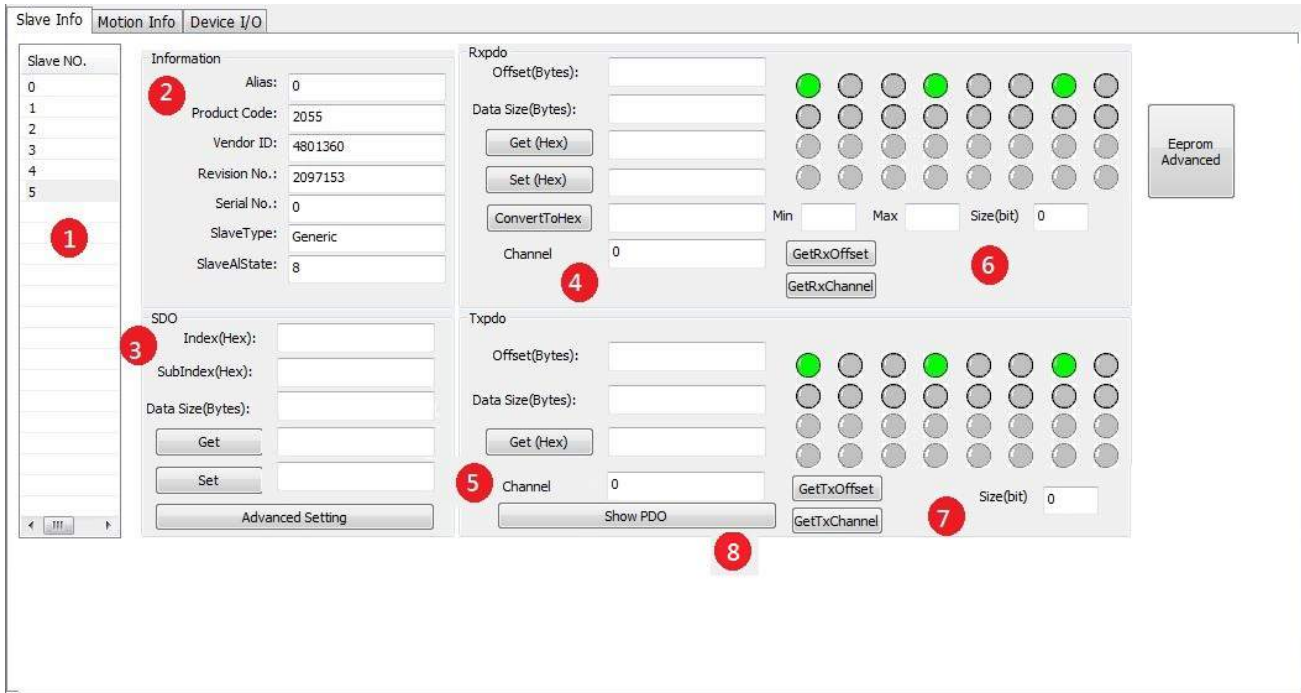


| Item | Description |
|------|-------------|
| (1) | Slave list (all the scanned slaves are listed here) |
| (2) | Slave information of the selected slave |
| (3) | SDO read/write for the selected slave |
| (4) | Read/Write output objects (RxPDO) of the selected slave |
| (5) | Read input objects (TxPDO)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

### 3.4.1. Basic Slave Operation Steps

1.  After executing the device initialization steps, the user can select a slave from the list of slaves. The related slave information will be displayed in the slave information group box.
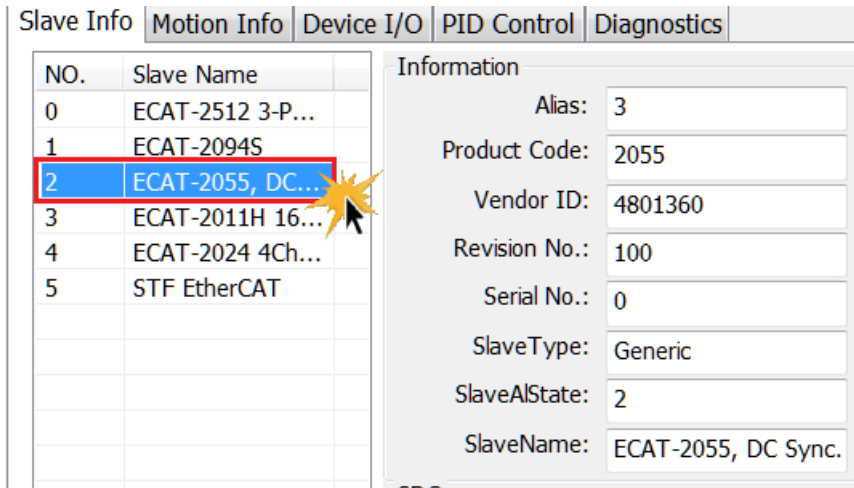


### 3.4.2. Slave SDO Operation Steps

1.  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.
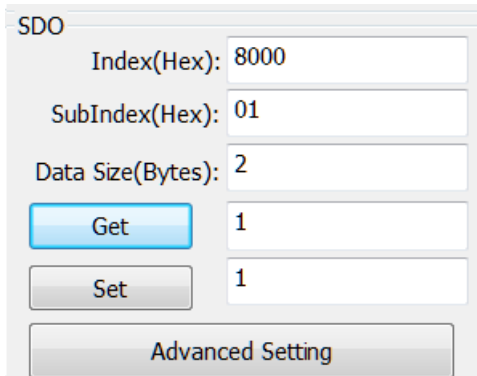


2.  After building (click ⚙) network information from ESI files, user can start using the advanced settings function.

3.  The Advance Setting button provides users with easy access to read and write SDO. Right click on sdo list to do write access.

4. When the selected module is ECAT-2091S/ ECAT-2094S, the following items will be shown(Table 3.5).

Table 3.5

| Item | Description |
|---|---|
| (1) Save As File | Save the SDO data of the current slave as a file (Index range 0x8000~0x8321) |
| (2) Load from File | Read the SDO data file and write to the current module, and confirm whether to burn EEPROM |
| (3) Load File for All | Read SDO data file and write to all matching modules connected to the Master station, and confirm whether to burn EEPROM |

### 3.4.3. Slave PDO and DI/DO LED Operation Steps

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

2.  DO/DI LED operations include some further processing on RxPDO and TxPDO data and show the status on LED display. A DO slave module has RxPDO objects mapping to digital outputs. A DI slave module has TxPDO objects mapping to digital inputs. Users can change digital outputs by writing data to RxPDO objects and get their values by reading them. In the same way, user can get the values of digital inputs by reading TxPDO objects.

3.  ConvertToHex button function: Enter the decimal value you want to convert to a hexadecimal number. The MIN and MAX define the range of that decimal value; and Size (bit) defines the range of the hexadecimal value. Note: here this hexadecimal value is a signed value.

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

4.  Show PDO button function: Show RxPDO and TxPDO objects.

    Right click on Output(RxPDO) list to do write access.
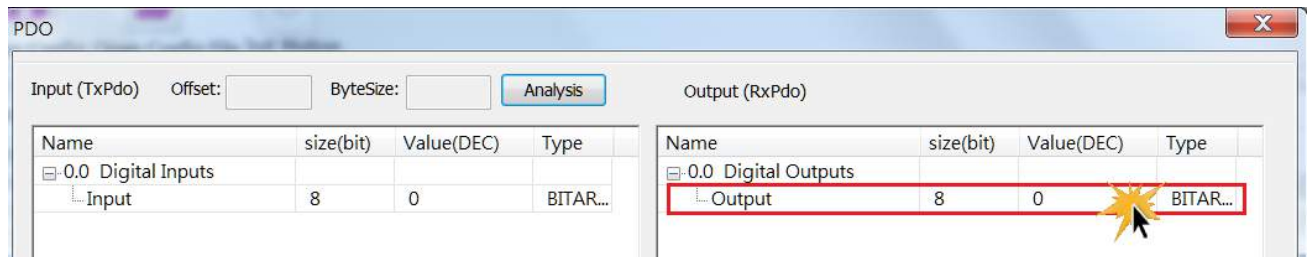
### 3.4.4. Slave PDO Analysis(Firmware Ver 1.0.15 or above)

1. Click on the data to be analyzed, and the Offset and ByteSize of the data are displayed above. If it is not displayed, the data cannot be analyzed. Click Analysis to start the analysis.
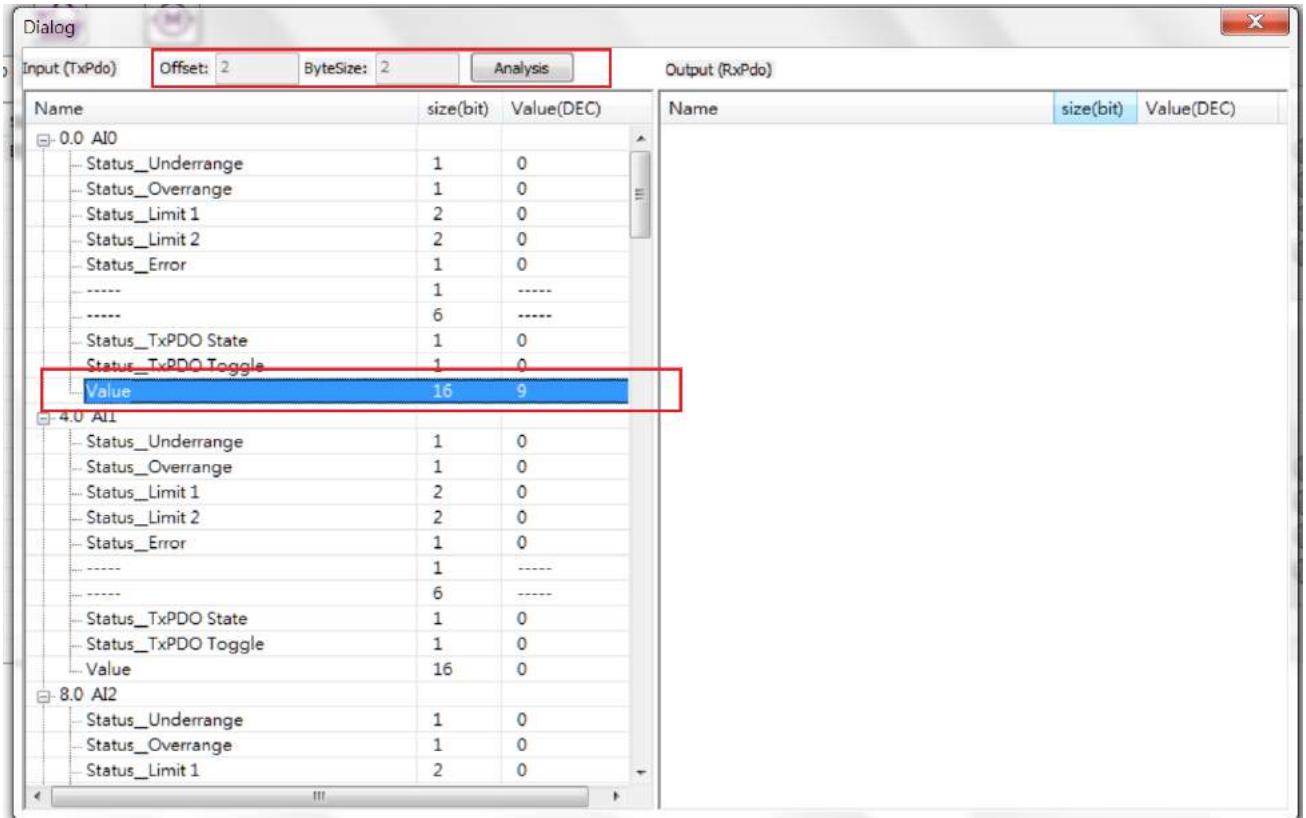


2.

The figure below shows the results of PDO analysis. The frequency domain is shown at the top and the time domain is shown below.

● Can be used to analyze whether the data has a specific noise frequency and noise intensity

● Software filter can be set to reduce noise interference

● Software filter uses the following API

➢ int32_t ECAT_SetAiFilterFreq(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t FilterType, double Frequency)

➢ int32_t ECAT_SetAiFilterParam(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, uint16_t Enable)

● Notch filter: Reduces noise at specific frequencies. Such as: 60hz noise.

- Low Pass filter: Reduces high frequency noise.
- High Pass filter: Reduces low frequency noise.
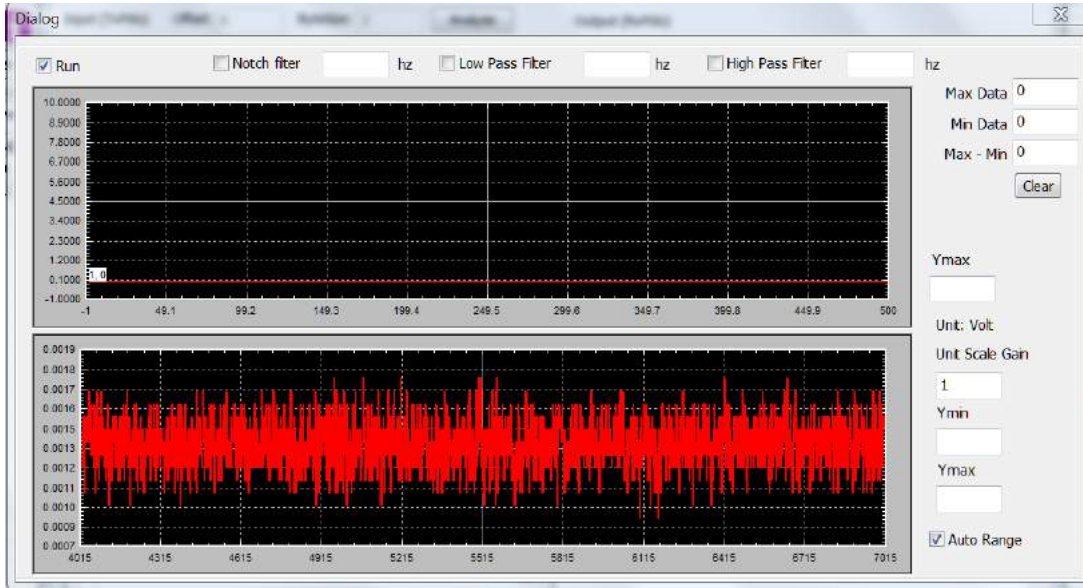


# 3.5. Motion Control Initialization Toolbar

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



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

## 3.5.1. Motion Control Parameter File Editing Steps

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

table.
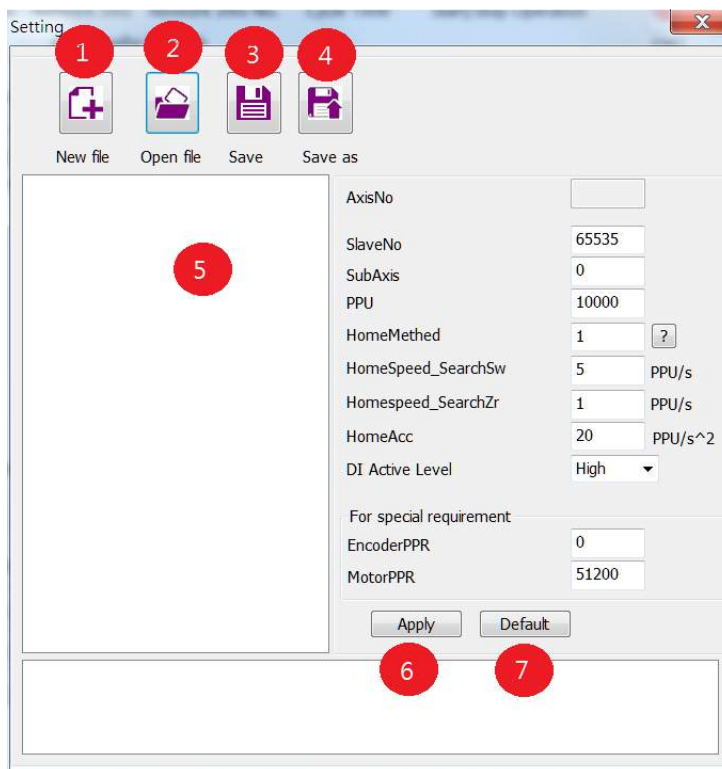


| Item | Description |
|------|-------------|
| (1) ⊞ | Create a new parameter file |
| (2) 📂 | Open an existed parameter file |
| (3) 💾 | Save the current parameter file |
| (4) 💾 | Save as another parameter file |
| (5) | Parameter information panel |
| (6) | Submit (apply) selected axis settings.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

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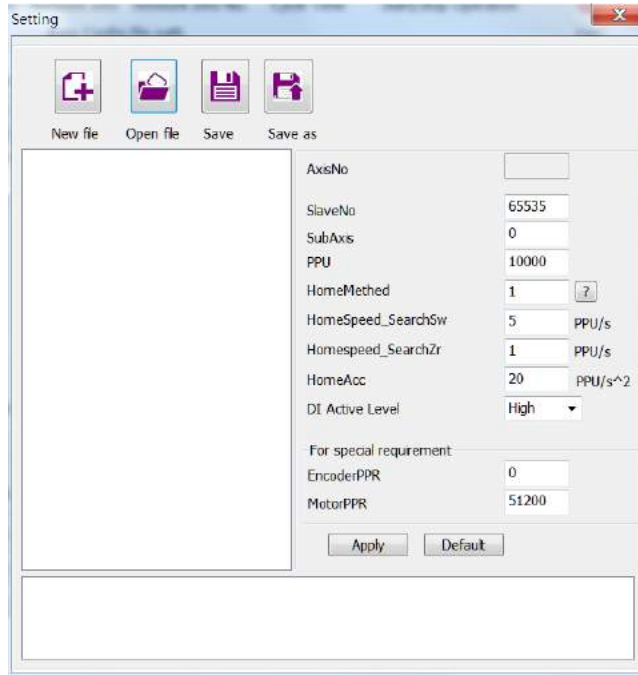


➢ Set the **SlaveNo** first. This is the axis number to be operated for this slave. **Note: When this SlaveNo is set to be 65535, it becomes a virtual axis.**

➢ SubAxis is for configuring multiple axes on one slave, such as some multi-axis motor drivers. ECAT-M801/ e-M901 FirmWare Version needs to be 1.0.15 or above, otherwise only ECAT-2094S is supported. Set 0 to Subaxis for the first axis of motor driver; set 1 to Subaxis for the second axis; and so on.

➢ PPU: Pulses Per Unit, pulses of each unit. If you want to set the unit to revolution, and every revolution requires 4194304 pulses, then set the PPU to 4194304.

It should be noted that the unit of all PDO Entry in Table 3.6 needs to be pulse to customize the PPU, and when the unit of PDO Entry (Table 3.6) of the driver is not pulse, the PPU needs to be set to 1.

| RxPDO | | TxPDO | |
|---|---|---|---|
| 607A | Target Position | 6064 | Position actual value |
| 60FF | Target Velocity | 606C | Velocity actual value |

Table 3.6

➢ HomeMethod

➢ HomeSpeed_SearchSw (speed for searching switch)

➢ HomeSpeed_SearchZr (speed for searching index)

➢ HomeAcc (acceleration)

➢ DI Active Level:

Most drives define the active level of three axial sensors, LSN (OT-), LSP (OT+) and HOME (ORG) as HIGH. Therefore, if the value of bit2 ~ bit0 of Digital Input Object 0x60FD is 000b, all these three sensors are not triggered. However, the Mitsubishi MR-JET-G-N1 servo drive takes a different definition as active LOW. It means that the motor can move only when the lowest two bits of object 0x60FD are '1'. In order to let all kinds of servo drives work together under this utility, users must set this as LOW for Mitsubishi drive and HIGH for other brands. For programming, there is an API for setting the active level for a specified axis.

If drive define the active level as LOW, you need to confirm that ECAT-M801/ e-M901 version is 1.0.18 or above.

➢ EncoderPPR (pulse per revolution of encoder which is defined for appending an encoder to a stepper motor), Support ECAT-2091S/ECAT-2094S, only need to set when Encoder is attatched.

➢ MotorPPR (pulse per revolution of motor), Support ECAT-2091S/ECAT-2094S, only need to set when Encoder is attatched.

➢ AccTime (acceleration time)

2. Choose an axis node by clicking the right-hand mouse button; a small menu will pop-up. Choose "**add Axis**" to add an axis after the last node. Click "**insert Axis**" to insert an axis right after the current node. Click "**delete Axis**" to delete the selected axis.
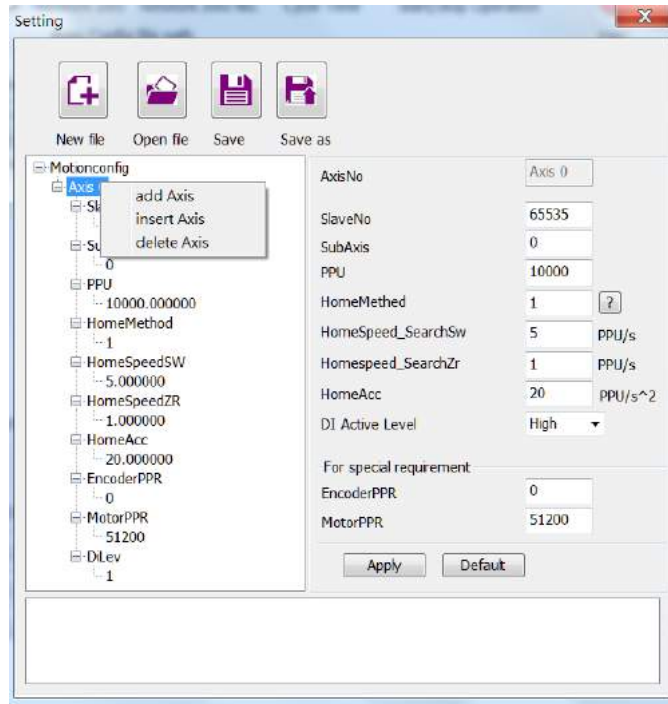
3.  After editing an axis, any time you click 📑, the changed contents are saving to file.

4.  Click 📑 to save the contents into a new parameter file.

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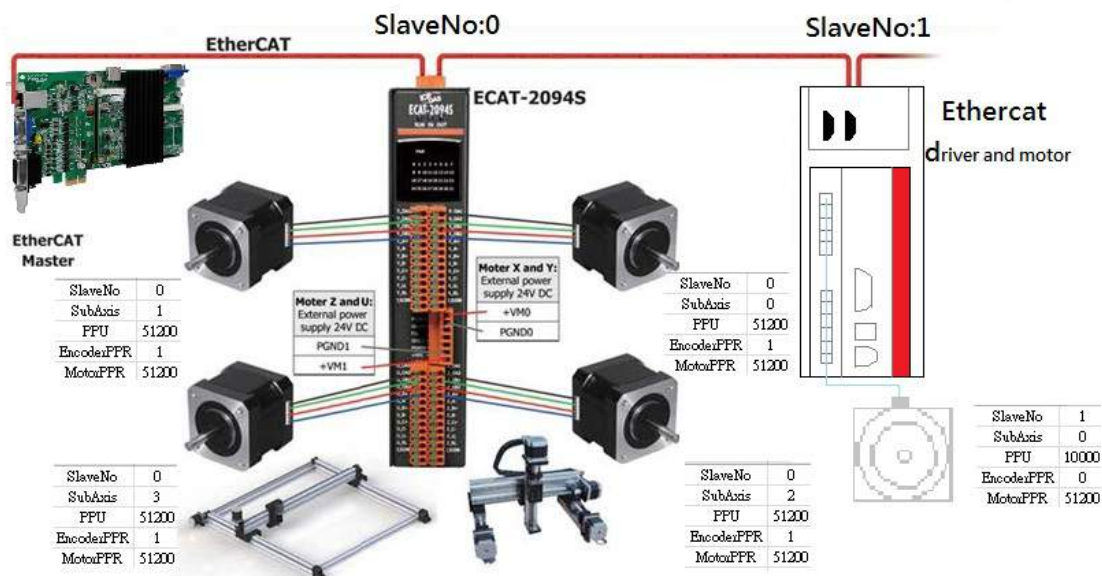


### 3.5.2. Motion Control Initialization Steps

1.  After the user has completed the motion control parameter file editing step, click 📂 on

the motion control initialization toolbar to open the edited parameter file.

2.  Click ⓜ will use this parameter file to initializing the every single-axis definition. To configure groups for motion control, further steps need to be implemented.

## 3.6. Motion Control Page

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

### 3.6.1. Single-Axis Motion Control Page
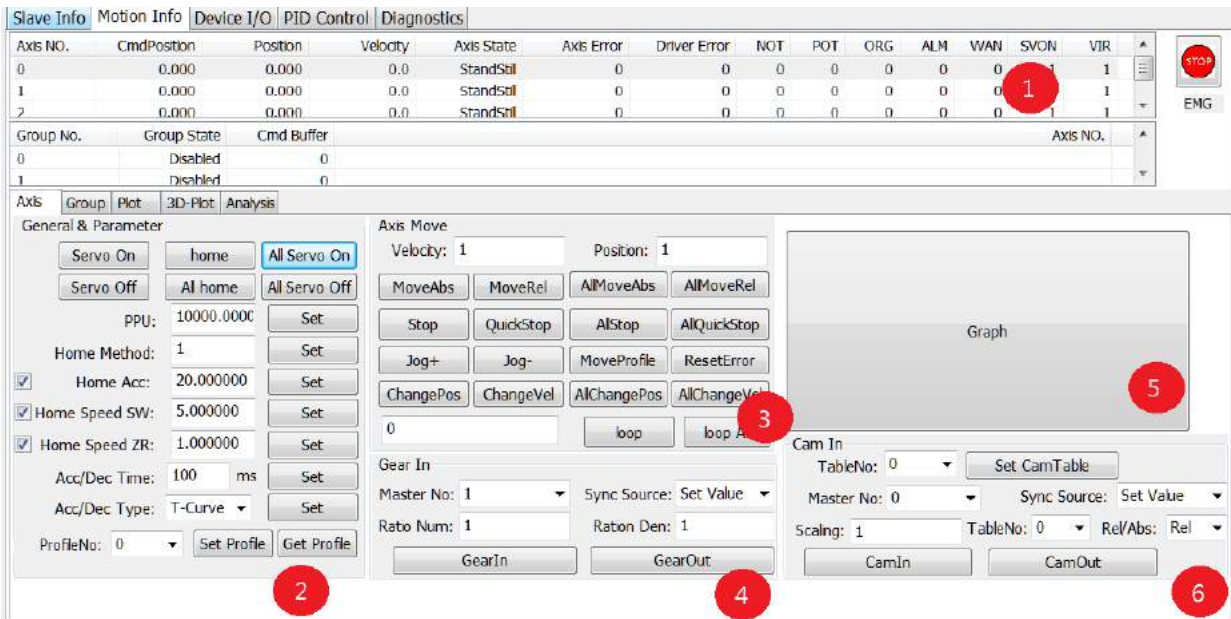


| Item | Description |
|------|-------------|
| (1) | Single-axis motion Information |
| (2) | Single-axis parameter settings |
| (3) | Single-axis motion control function tests |
| (4) | Gear function settings and testing |
| (5) | Display of single-axis Position and Velocity |
| (6) | E-Cam function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

## Single-axis motion information

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

## Single-axis parameter settings

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

## Single-axis motion control functions

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

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

4. Clicking "Stop" or "QuickStop" button can stop this single-axis motion control test.

5. To control of all axes, set the velocity "Velocity" and "Position" parameters. Then click the **"All MoveAbs"** or **"All MoveRel"** button to use the same parameter settings to perform single-axis motion control for all axes. Click the "All Stop" or "All QuickStop" button to stop all axes.

6. When the "Jog+" or "Jog-" button is clicked, the "Velocity" parameter is used to start a movement with a specified velocity. Click "Stop" or "QuickStop" to stop this motion.

7. The edit box beside the **loop** and **loop** buttons is used for enter a loop number. Set this value first. When loop or loop All is clicked, the axis or axes will move back and forth between the current position and the set position.

## Gear function settings and testing

1. An axis is selected in the single-axis motion information panel, and it is enabled.

2. Set its "Master No" parameter. This master axis will be the reference axis.

3. Next, set the slave reference source. The reference source can be either the master's command set value or the master's actual position value.

4. The electronic gear ratio is composed by a numerator and a denominator. Set the numerator in the edit box with label "Ratio Num", and set the denominator in the edit box with label "Ratio Den". Source value multiplied by the gear ratio will be the reference command of the slave axis.

5. Next, click the "GearIn" button to start the gear motion. The state of the slave axis will change into SyncMotion. After that, the slave axis will follow any motion of the master axis with the gear ratio defined before.

6. Click the "Gearout" button will stop the synchronized motion. The state of the slave axis will change from synchronized motion to be the continuous motion. If you want to stop the gear motion, click the "Stop" or "QuickStop" button to stop this following motion control.

## eCam function settings and testing

1. An axis is selected in the single-axis motion information panel, and it is enabled.

2. Set its "Master No" parameter. This master axis will be the reference axis.

3. Next, set the slave reference source. The reference source can be either the master's command set value or the master's actual position value.

4. Next, Set the Scaling, TableNo and Rel/Abs to define how slave following master axis.

5. Next, click the "CamIn" button to start the eCam motion. The state of the slave axis will change into SyncMotion. After that, the slave axis will follow any motion of the master axis by the definition in CamTable mentioned before.

6. If you want to stop the eCam motion, click the "Stop" or "QuickStop" button to stop this following motion control.

## Display of single-axis Position and Velocity

1. Choose an axis by clicking a axis number in the single-axis motion information panel.
2. Click the "Graph" button.
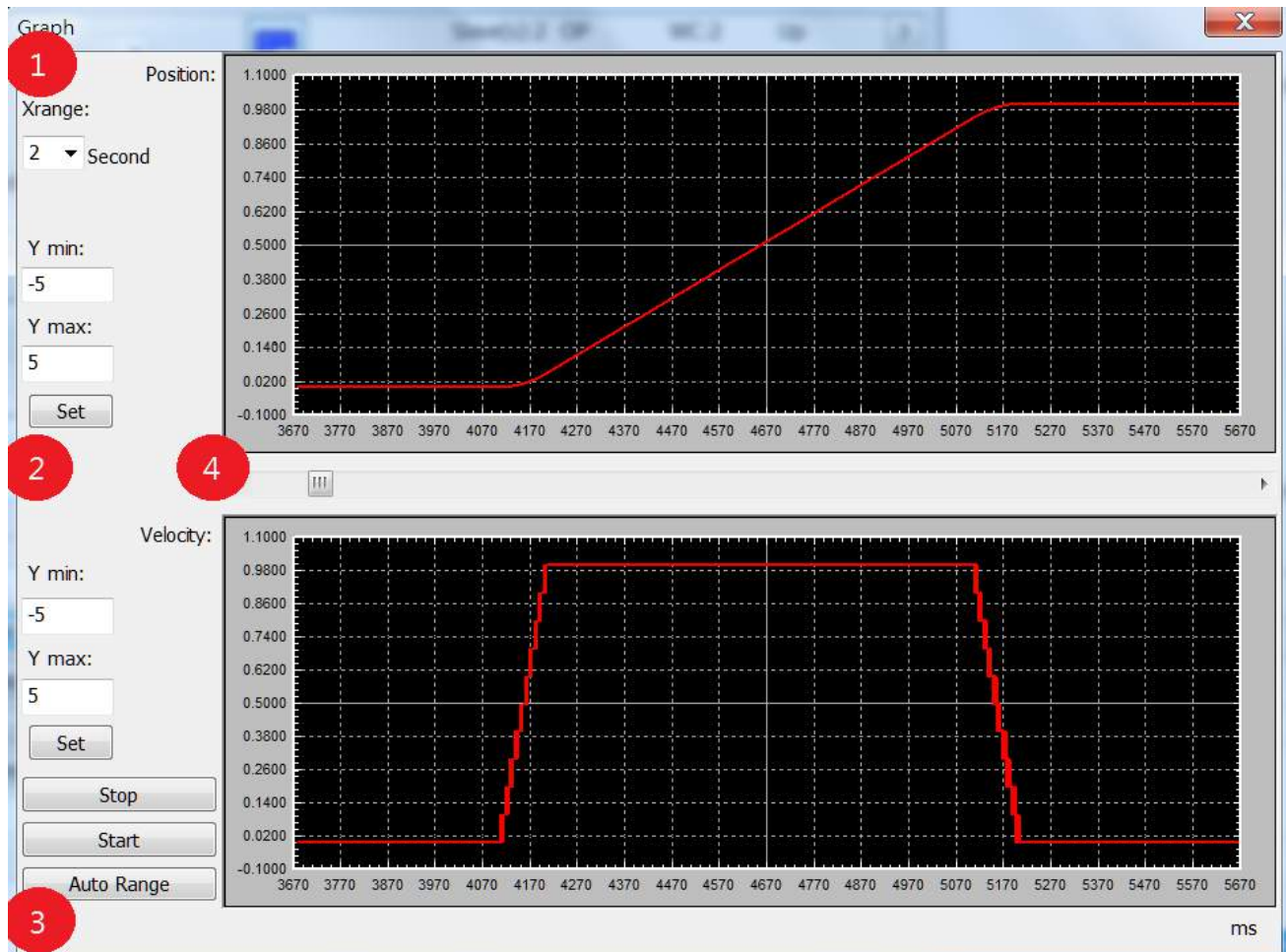


| Item | Description |
|------|-------------|
| (1) | X(Time)Axis Rane |
| (2) | Y(Position)Axis Range |
| (3) | Stop、Start and Auto Range(Y Axis) |
| (4) | Position、Velocity display box |

## 3.6.2. Group Motion Control Page
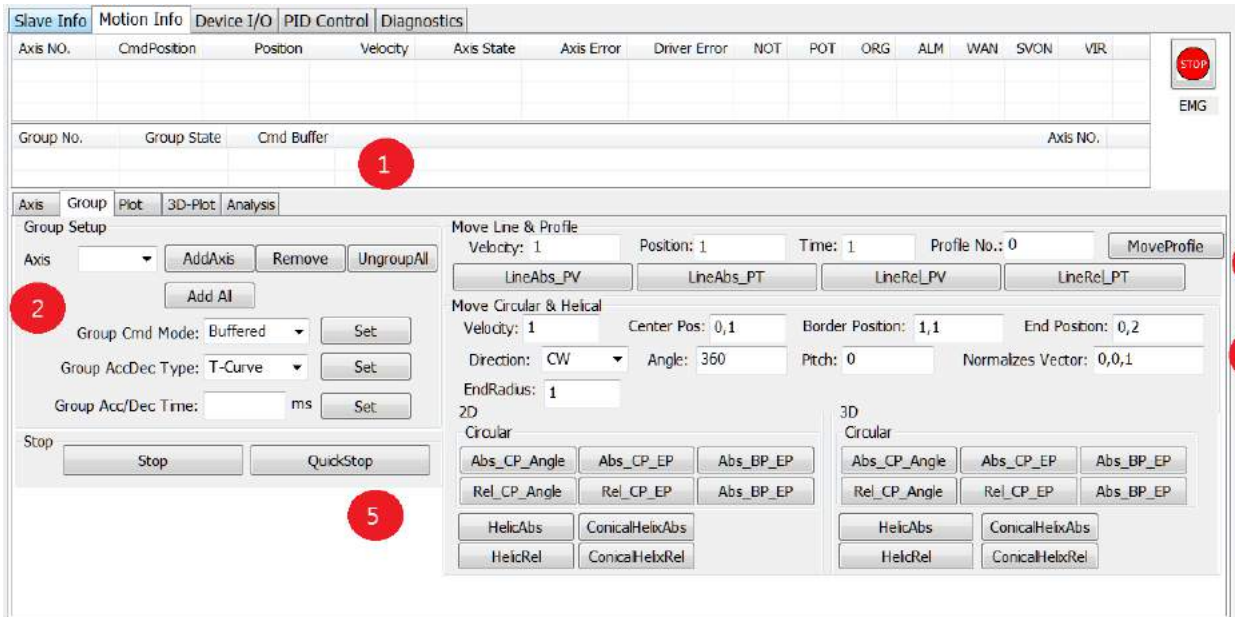


| Item | Description |
|------|-------------|
| (1) | Group motion information panel |
| (2) | Group motion parameter settings |
| (3) | Group linear motion and Profile motion tests |
| (4) | Group circular motion and helical motion tests |
| (5) | Group stop function tests |

## Group motion information panel

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

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

## Group motion parameter settings

1. Select a specified group number in the group motion information panel.
2. If there is not any axis number in the group, the group state is disabled. User can 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 absolute position method, users can input 50,100 in the position edit box to move the first axis to 50 and the second axis to 100.
3. Next, click the "Line Abs" or "Line Rel" button to start the multi-axis linear interpolation moving in absolute or relative mode.
4. While moving, click "Stop" or "QuickStop" to stop the group motion.

## Group circular motion control

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

| Item | Description |
|---|---|
| CircAbs_CP_Angle | Start group 2D circular interpolation motion by setting a center position and an angle in the absolute mode. |
| CircRel_CP_Angle | Start group 2D circular interpolation motion by setting a center position and an angle in the relative mode. |
| CircAbs_CP_EP | Start group 2D circular interpolation motion by setting a center position and an end position in the absolute mode. |
| CircRel_CP_EP | Start group 2D circular interpolation motion by setting a center position and an end position in the relative mode. |
| CircAbs_BP_EP | Start group 2D circular interpolation motion by setting a border position and an end position in the absolute mode. |
| CircRel_BP_EP | Start group 2D circular interpolation motion by setting a border position and an end position in the relative mode. |
| 3D CircAbs_CP_Angle | Start group 3D circular interpolation motion by setting a center position and an angle in the absolute mode. |
| 3D CircRel_CP_Angle | Start group 3D circular interpolation motion by setting a center position and an angle in the relative mode. |
| 3D CircAbs_CP_EP | Start group 3D circular interpolation motion by setting |

| | |
|---|---|
| | a center position and an end position in the absolute mode. |
| 3D CircRel_CP_EP | Start group 3D circular interpolation motion by setting a center position and an end position in the relative mode. |
| 3D CircAbs_BP_EP | Start group 3D circular interpolation motion by setting a border position and an end position in the absolute mode. |
| 3D CircRel_BP_EP | Start group 3D circular interpolation motion by setting a border position and an end position in the relative mode. |
| | |

## Group helical motion control

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

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

## 3.6.3.   Show Position Page



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

## Motion position display area

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

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

## X-Y display

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

## multiple axes display

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

these axes

### 3.6.4.    3D Show Position Page



Note: Only show data within an hour.

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

## Zoom In/Out

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

| | |
|---|---|
| Auto Zoom | Auto Zoom In/Out |
| Auto Focus | Auto Focus |

## Reset/Clear

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

## X-Y-Z Axis Setting

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

## Motion position display area

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

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

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

## 3.7.  Device I/O Operation Page

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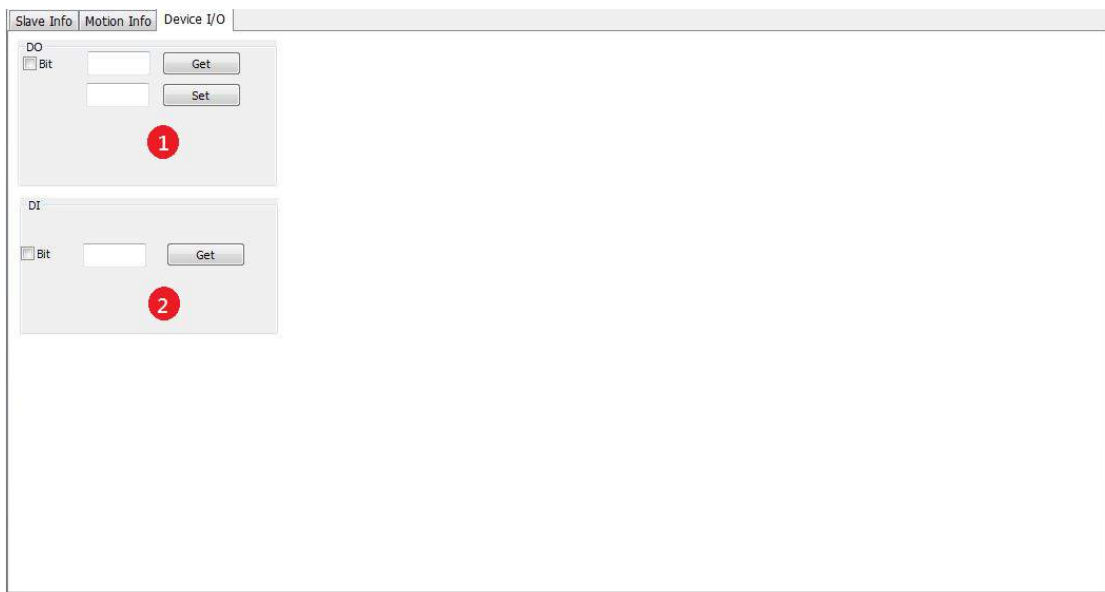


| Item | Description |
|------|-------------|
| (1) | Device DO control |
| (2) | Device DI control |

### 3.7.1.   Device DO control operation step

1. Click the "Set" button to write the data for all DO channels.

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

### 3.7.2.   Device DI control operation step

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

# 3.8.  PID Control Page

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

## 3.8.1.   PID Control Page



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

## Status of PID Controller

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

## Parameters of PID Controller

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

## Input/output module settings for PID Controller

| item | Description |
|------|-------------|
| Interval | Control Interval of PID Controller，Unit: EtherCAT CycleTime |
| In_slaveNo | Measuring channel is located in this slave module |
| In_OffsetByte | TxPDO Offset of the measuring channel |
| In_bitSize | Data size of this measuring channel, Unit: bit |
| In_ScaleGain | Scale gain for conversing digital value into physical value |
| In_ScaleOffset | Scale offset for conversing digital value into physical value |
| Out_slaveNo | Control output channel is located in this slave module |
| Out_OffsetByte | RxPDO Offset of this control output channel |
| Out_bitSize | Data size of this control output channel，Unit: bit |

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

# 3.9. EtherCAT Diagnostic

Show error counter after entering OP mode.

## 3.9.1. EtherCAT Diagnostic Page
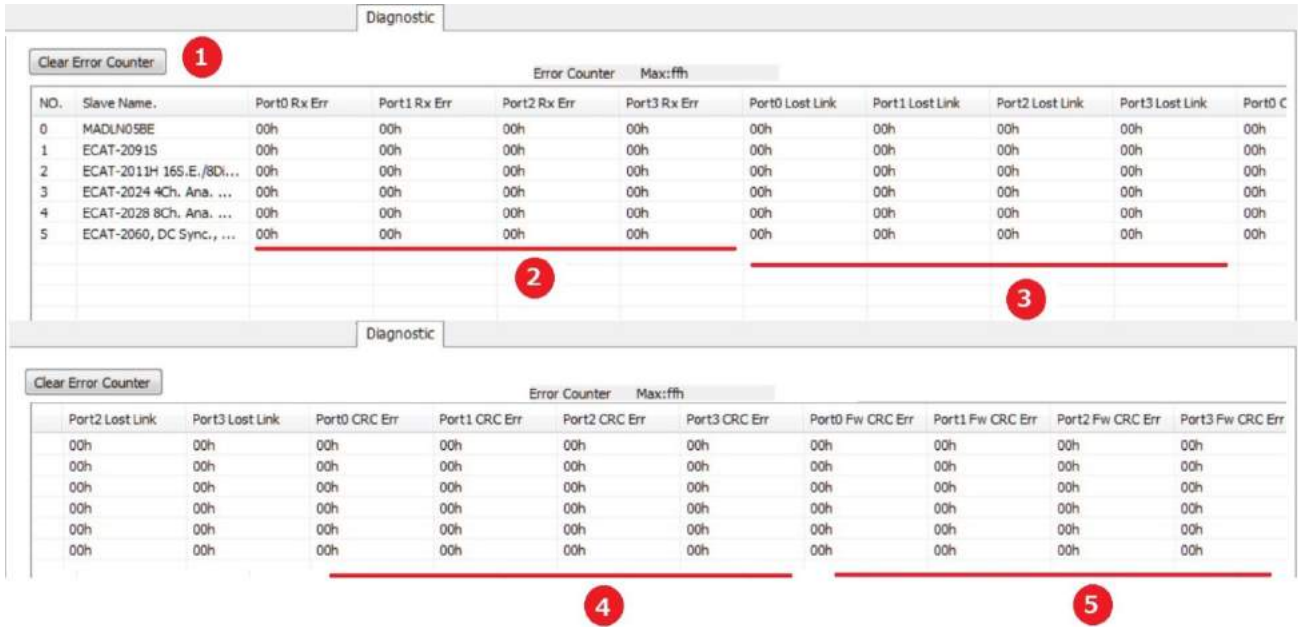


Fig. 3.4

Table 3.7

| item | Description |
|------|-------------|
| (1) | Clear error counter. Maximun value of error counter : 255(Dec) 0xff(Hex) |
| (2) | Invalid frame(Rx) error counter |
| (3) | Link lost error counter |
| (4) | Invalid frame (CRC) error counter |
| (5) | Forwarded CRC error counter |

## 3.9.1.   Hardware Diagnostic Procedure

Invalid frame(Rx)

Invalid frame (CRC)

A change of RX/CRC Error Counters indicates that the hardware signal received was

corrupted and that the carried data will be discarded.

Most likely reasons for signal corruption are:

• External EMC disturbances (usually sporadic counter increment)

• Damaged devices or interconnections (usually fast and systematic counter increment)

RX Errors:

• Correspond to individual invalid symbols

• Can occur both within and outside frames (when occurring within frames, they represent

   usually also Frame Errors)

CRC Errors:

• Correspond to frames whose overall bit sequence was corrupted

• Can occur only within frames

First port reporting RX/CRC Error Counter ≠ 0 → most likely problem location

| slave NO. | Port0 Rx Err | Port1 Rx Err | Port2 Rx Err | Port3 Rx Err |
|-----------|--------------|--------------|--------------|--------------|
| slave0 | 00h | 81h | 00h | 00h |
| slave1 | dah | 00h | 00h | 00h |
| slave2 | 00h | 00h | 00h | 00h |
| slave3 | 00h | 00h | 00h | 00h |
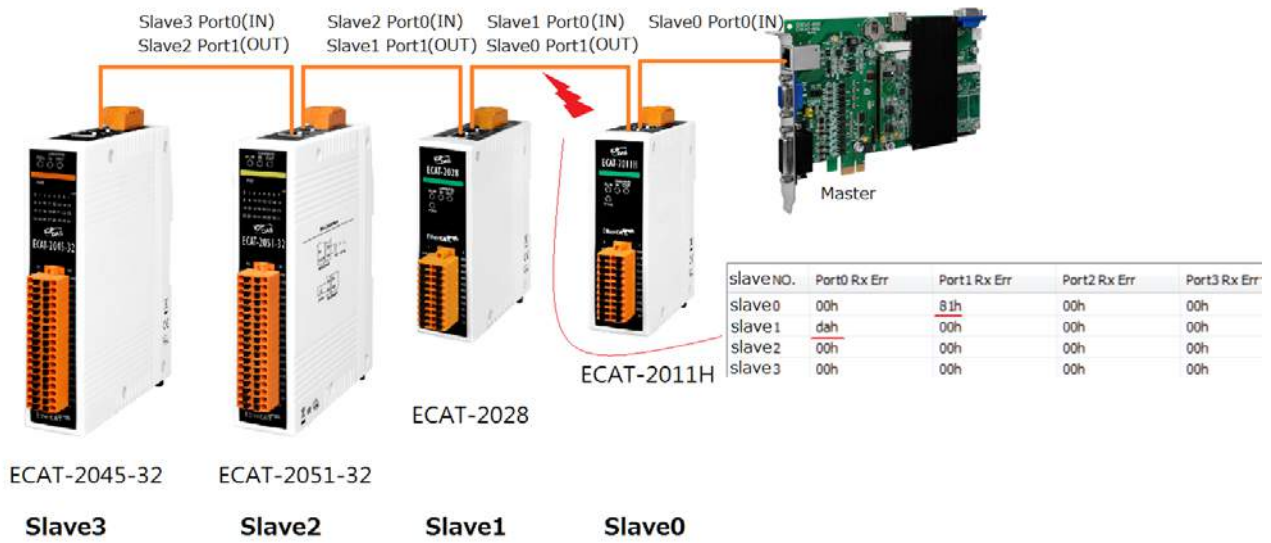
Check the following hardware aspects:

• Check cable between detected and previous slave:

  - EtherCAT cable is routed near to power cables or noise sources

  - Self-made cable connectors have been badly implemented

  - Cable is not properly shielded

• Check detected and previous device:

  - Not suitable power-supply (for example, low LVDS current)

  - Devices don´t share the same ground potential

• Try to replace/swap devices at two ends of the detected location, in order to check if errors

are related to a specific device part.

As external EMC disturbances are asynchronous with the communication, both Rx and CRC

Errors should be counted in this case (even if their ratio can vary).

Completely unbalanced counter values (many Physical Layer Errors with no Frame Errors,

or many Frame Errors with no Physical Layer Errors) could instead indicate an internal

device issue: replace the devices could be therefore the first suggested step in this case

## Link lost

An increment in a Lost Link Counter indicates an interruption in the hardware

communication.

Most likely reasons for link loss are:

• Temporary or permanent device power-supply loss, or device reset.

• Damaged cables or connectors or poor/oxidized contacts

• EMC disturbances

# 4. Function Overview

## 4.1. Device Operation Flow

As shown in Figure 4.1, the user can call the *GetDeviceCnt* function to find out how many devices (cards) can be used. Each device should have a unique Card ID. The Card ID is set by four-bit dip-switch on the ECAT-M801/ e-M901 control card. Then, according to the Card ID, call *OpenDevice* function to open that device. After this device is opened, the EtherCAT cyclic communication does not start yet. Some basic device operation functions should be used to configure the communication before the cyclic communication can be started.

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

Before starting the operation task of EtherCAT, please use the utility program to create and edit at least one EtherCAT network information file and write the system information into the device. Then, in your program, call *StartDeviceOpTask* function to start the EtherCAT operation task. This function will command slaves to enter into the OP state. The user can use *GetDeviceState* to get the current states. If there is no error and the AL state reaches OP, the PDO cyclic data communication is on. Motion control operations can be configured and started. To stop the EtherCAT communication, *StopDeviceOpTask* function must be called. To close the device operation (close a card), use *CloseDevice* function to do it.
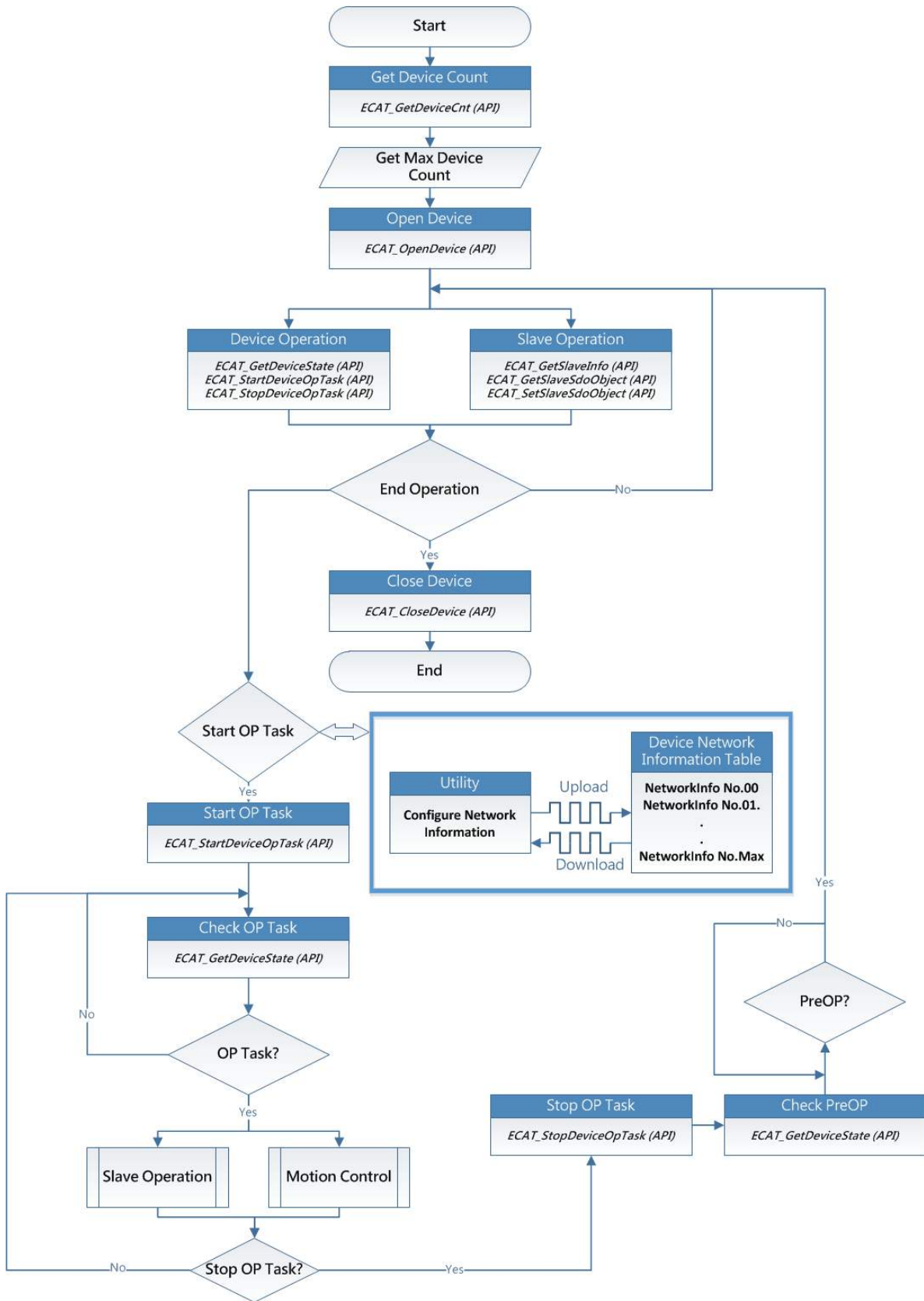
Figure 4.1

## 4.2. Slave Operation Flow

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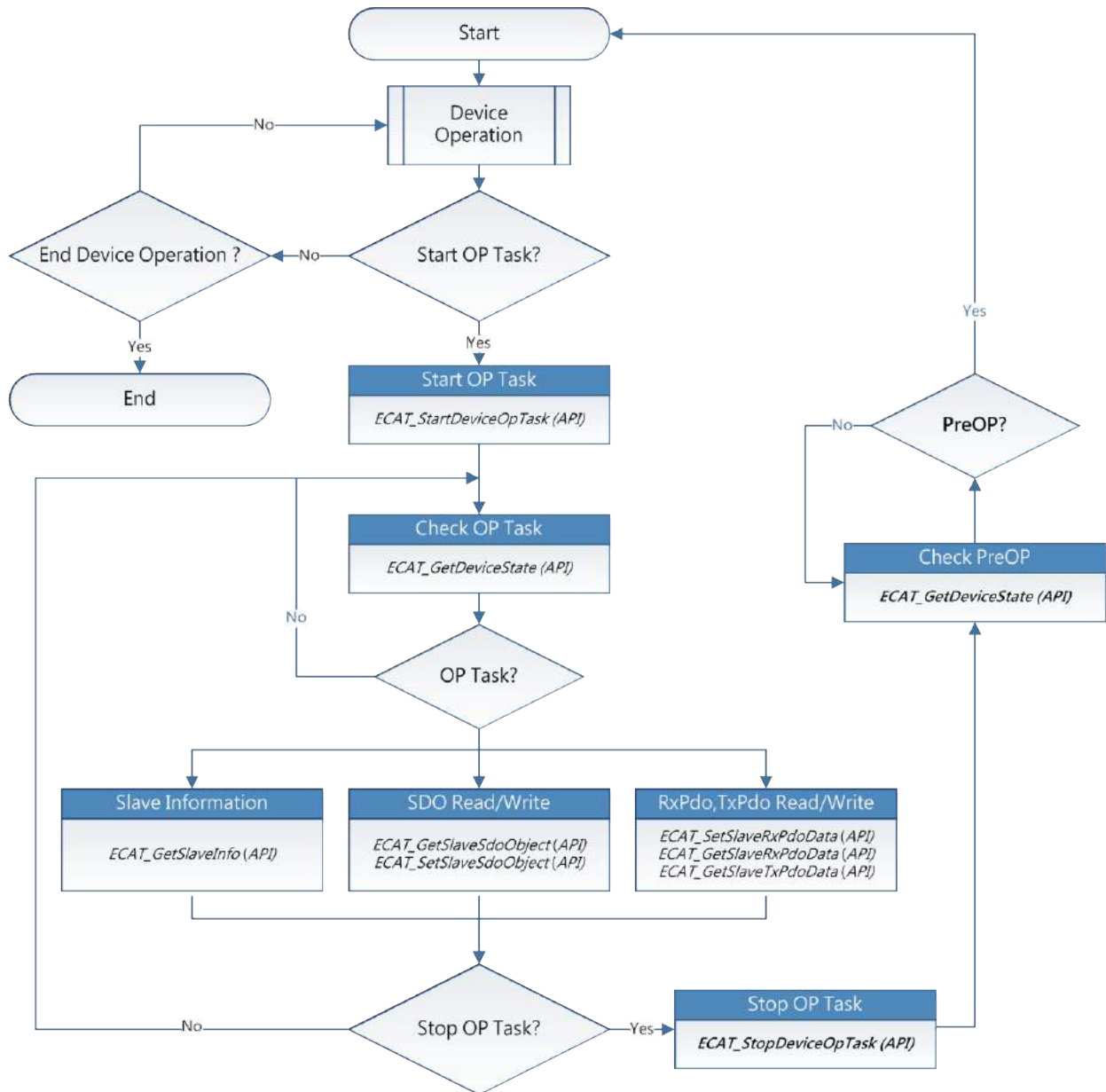


Figure 4.2

# 4.3. Motion Control Flow

## 4.3.1. Motion Control Initialization

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

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

Start

No → Device Operation

End Device Operation ? ← No → Start OP Task?

Yes → End

Yes → Start OP Task

ECAT_StartDeviceOpTask (API)

Check OP Task

ECAT_GetDeviceState (API)

No → OP Task?

Slave Information

ECAT_GetSlaveInfo (API)

*Stepper Refer to
STEPPER_MOTOR
4_AXIS_STEPPER_MOTOR

STEPPER_MOTOR refer to:
ECAT-2091S

4_AXIS_STEPPER_MOTOR refer to:
ECAT-2094S

Slave Type = CiA402 ?
Slave Type = *Stepper ? → No → Check Another Slave

Yes → Moton Control Initialize

ECAT_McInit (API)

No

Axis Operation | Axis Homing | Axis Error Process | Group Operation

Stop OP Task? → Yes → Stop OP Task

ECAT_StopDeviceOpTask (API)

PreOP? → No → Check PreOP

ECAT_GetDeviceState (API)

Yes

Figure 4.3

## 4.3.2. Axis Motion Control

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

When an axis is moving, the user can call stop functions to stop its motion. Only when the axis state changes to Standstill a new motion command can be issued again. If any error occurs while moving, the state of that axis will change to ErrorStop. In ErrorStop state, users need to deal with this error.
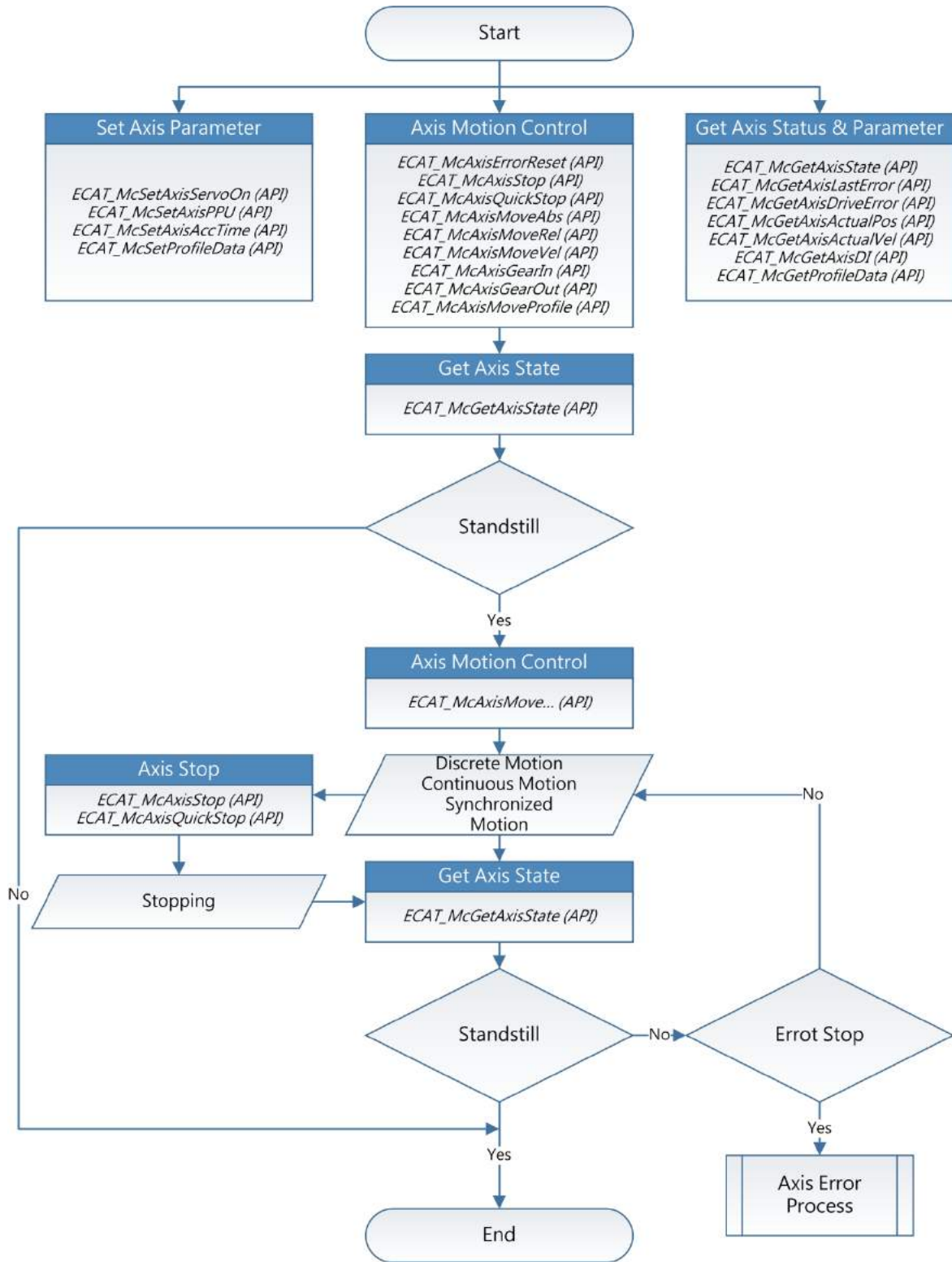
Figure 4.4

### 4.3.3. Axis Homing

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

The user can call the stop function to stop the axis homing. When the axis state changes from Homing to the Standstill, a new motion command can be issued. If any error occurs while homing, the state of the axis will be changed from Homing to ErrorStop. In this state, users need to deal with this error.
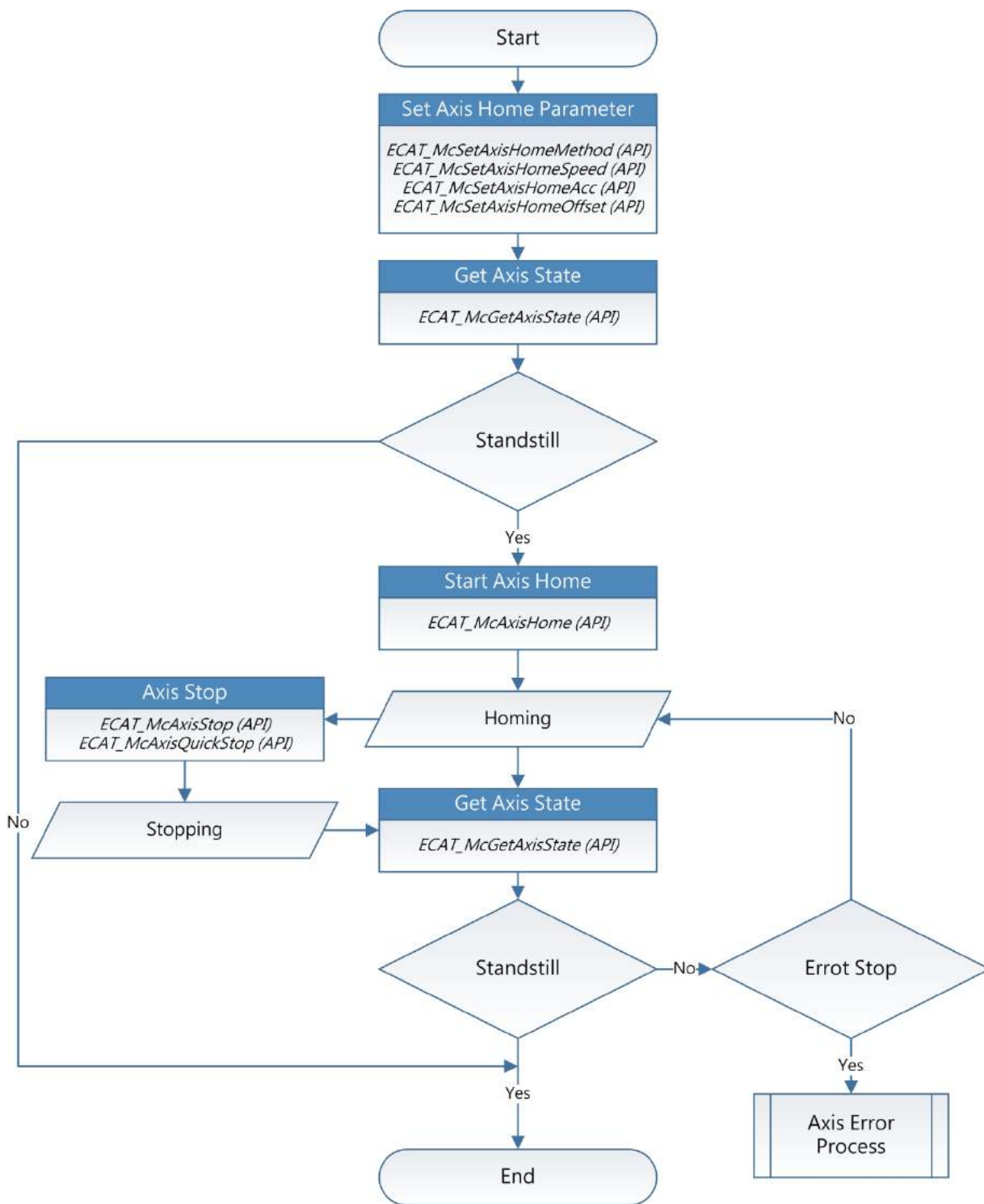
Figure 4.5

## 4.3.4.  Axis Error Process

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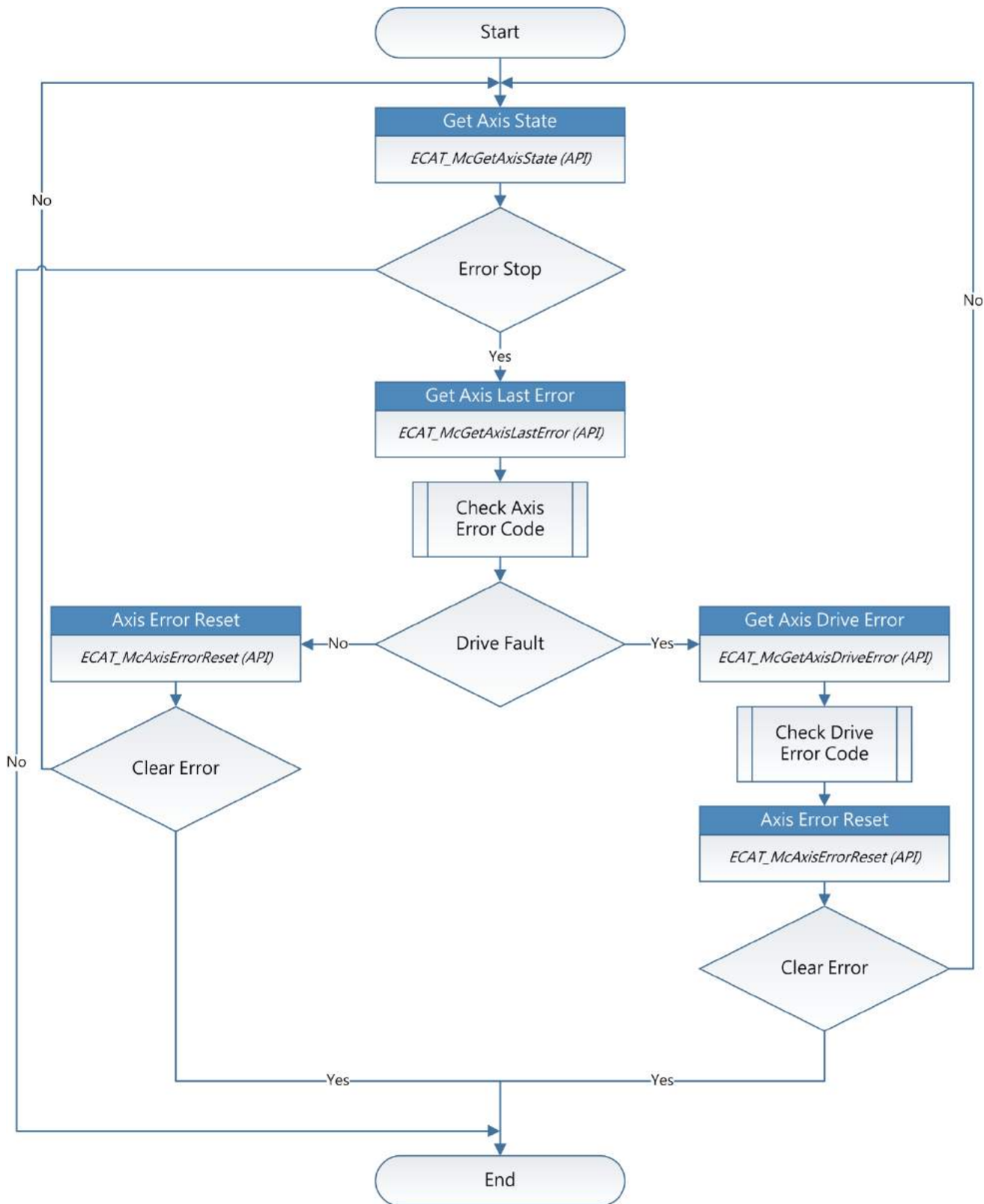
Figure 4.6

## 4.3.5. Group Moving

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

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

Users can call stop functions to stop the group motion. When the stop command is completed, group state will change from Moving to Standby. In Standby state, the group is ready for executing another motion command. If any error occurs while moving, the state of that group will change from Moving to ErrorStop. In this state, users have to deal with this error.
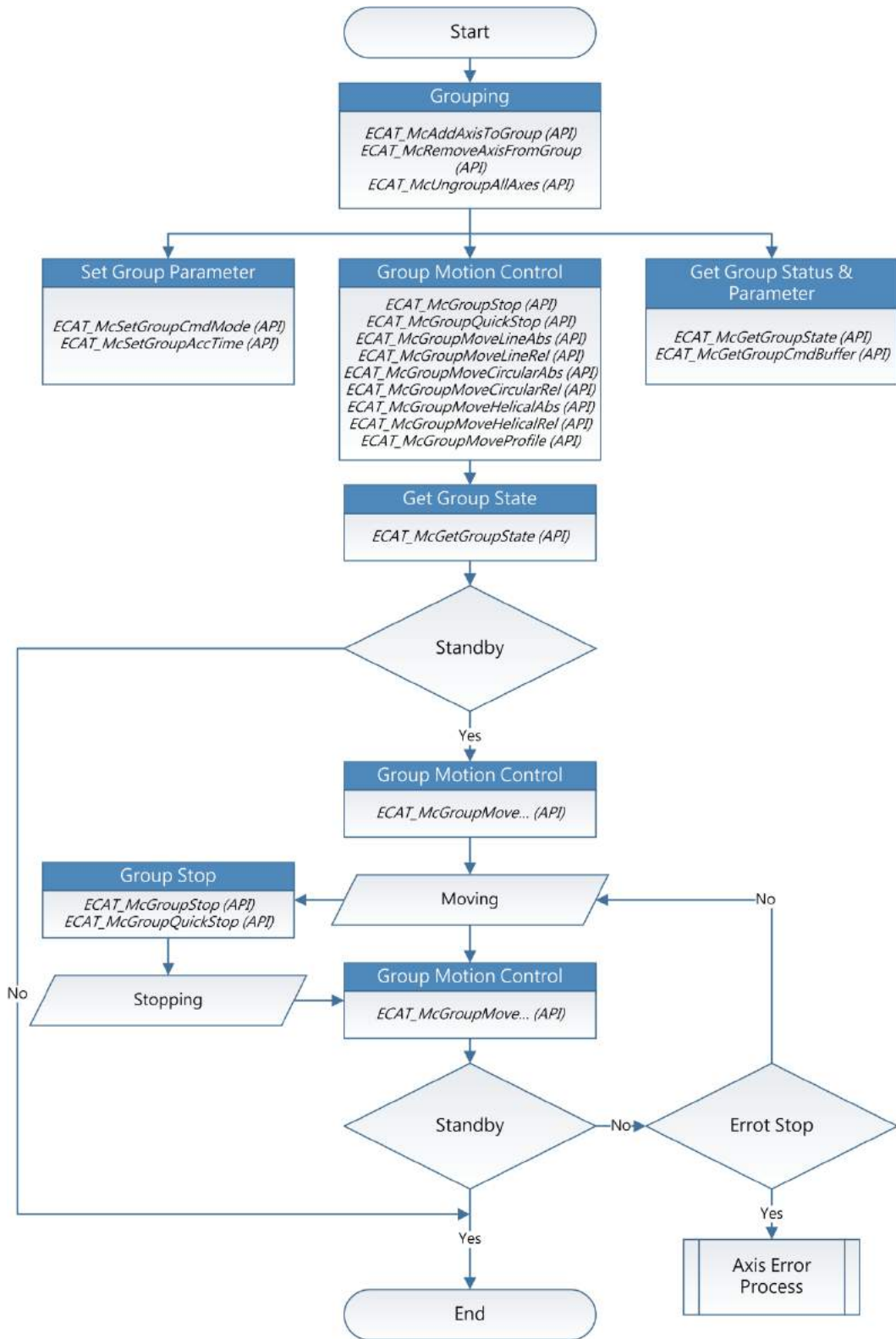
Figure 4.7

## 4.4. Communication error handling flow
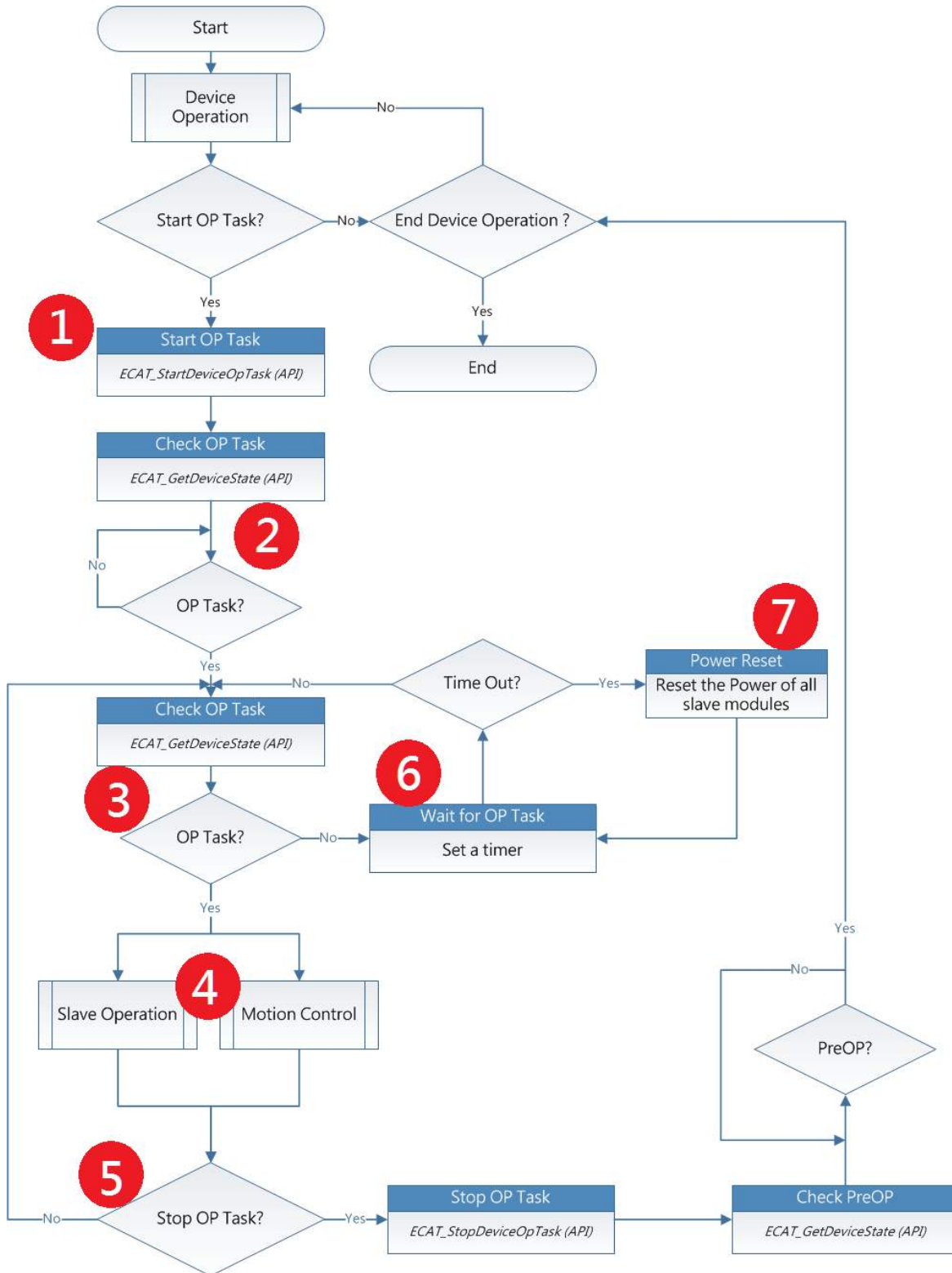


Figure 4.8

Step 1: Use ECAT_StartDeviceOpTask to enter the EtherCAT operation task.

Step 2: Use ECAT_GetDeviceState to read the current EtherCAT network status, and wait until the current EtherCAT network status is "OP" status. When the "OP" status is reached, record the current WC (Working Counter) and the number of slaves.

Step 3: After the network status is "OP", unless ECAT_StopDeviceOpTask is used to return to "PreOP" status, the network status should be "OP" status. Therefore, you need to use ECAT_GetDeviceState to periodically check whether the current EtherCAT network status is "OP" state, if the current state is not "OP", go to step 6.

- At the same time, compare whether the current WC is the same as the WC recorded in step 2.

- When the number of slave stations is different from the number of slave stations recorded in step 2, it means there is a disconnection. After the connection is restored, go to step 6.

Step 4: After the network status is "OP", users can start to read and write DIO and use motion control functions.

Step 5: Go back to step 3.

Step 6: When a communication problem occurs, the current network status deviates from "OP", or WC changes, or the number of slave stations changes. When communication returns to normal, the network status will change to "OP" again, and the WC will be the same as the WC recorded in step 2, go to step 3.

- However, some modules may fail to return to the "OP" state after communication fails, or the WC cannot return to the WC recorded in step 2. The user can set a timer, and the communication cannot be resumed when the state continues for a period of time, go to step 7.

- Users can use the ECAT-M801/ e-M901 application program to test the recovery time.

Step 7: Reset Power off all slave module, then go to step 6, wait for the network status to return to "OP".

# 4.5. Use motion Library in Windows

## 4.5.1. 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.
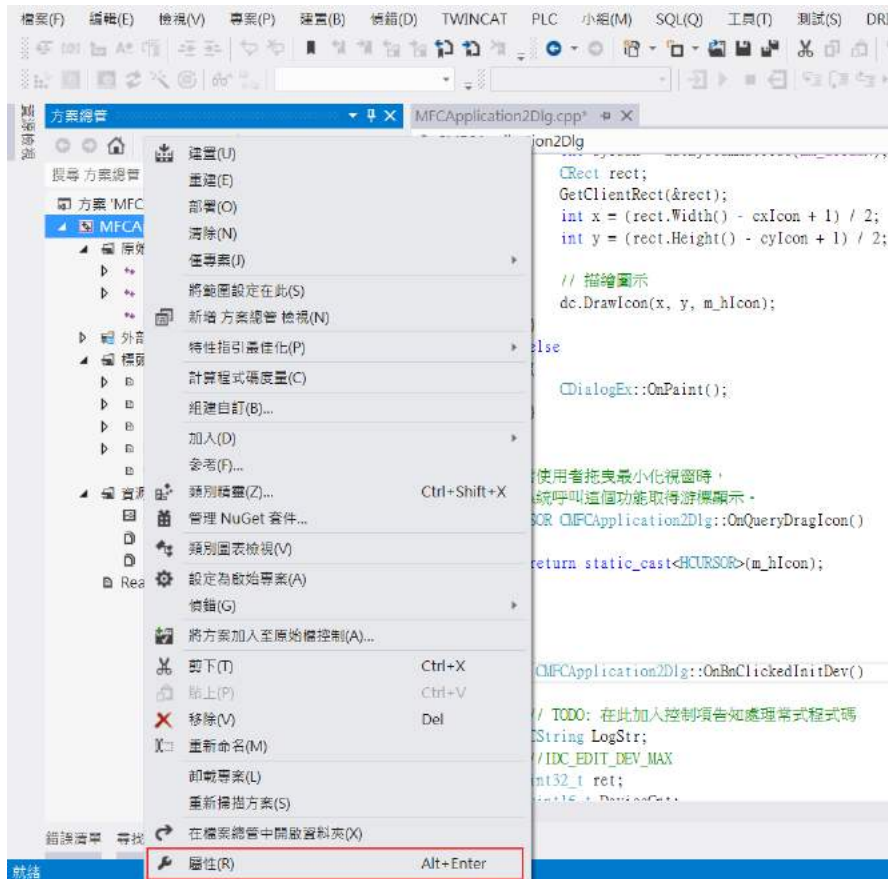


3.  Select **Configuration Properties->Linker->Input->Additional Dependencies**; enter **libecatdevice.lib** file in additional dependencies.
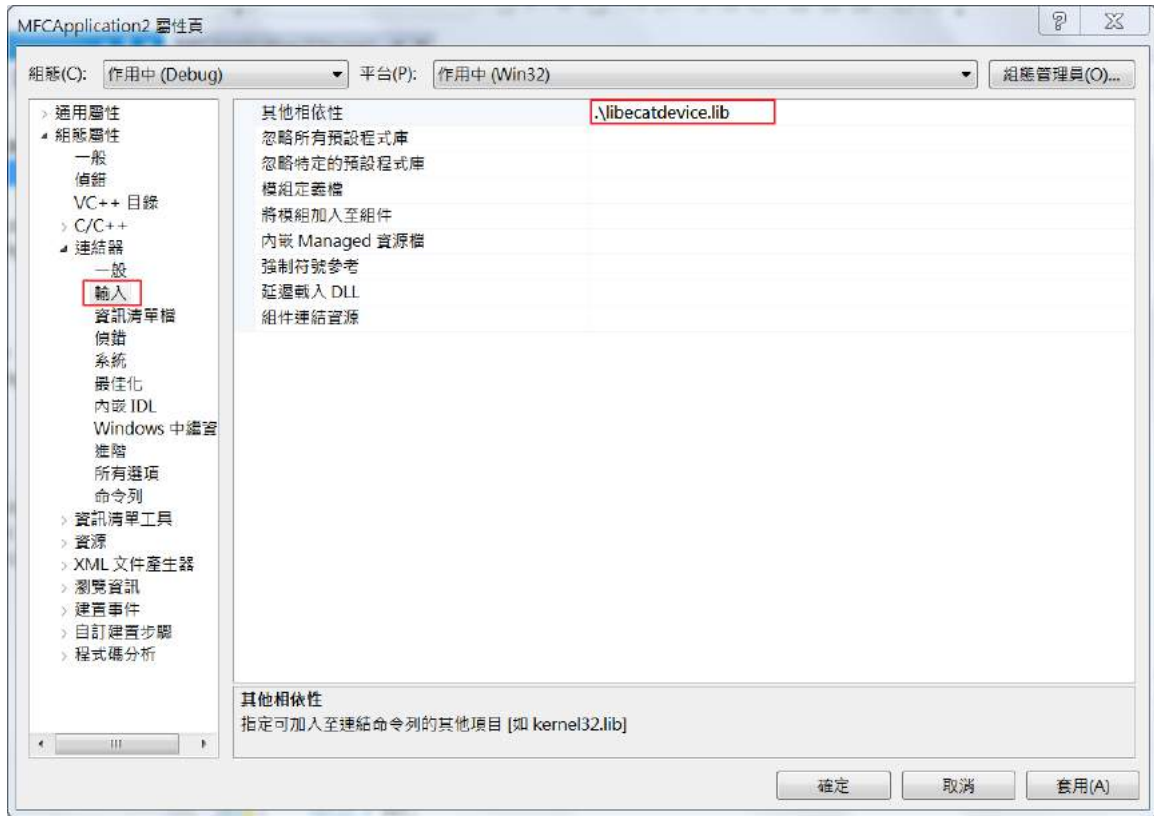
# 5. Device Operation Functions

## 5.1. ECAT_GetDeviceCnt

Description:

Get the number of available devices(ECAT-M801/ e-M901).
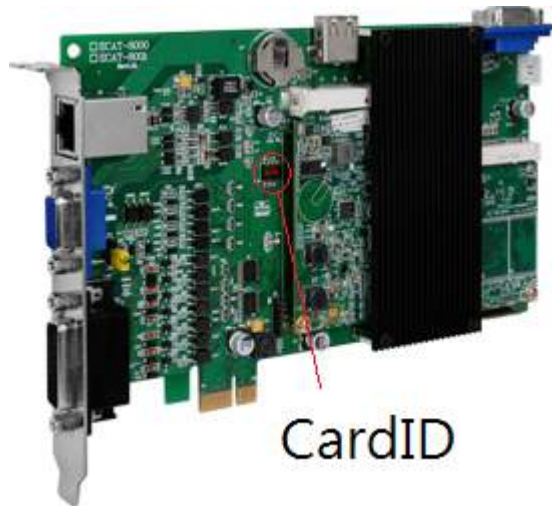


Syntax:

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

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```c
int32_t ret;
uint16_t DeviceCnt, i;
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);
    for(i=0;i< DeviceCnt;i++)
    {
        printf("CardId[%u] = %u \n", i, CardID[ i ]);
    }
}
```

# 5.2. ECAT_OpenDevice

**Description:**

Open a device with the specified Card ID.

Note: (1)A card can only be opened by one progress. If other progresses open the card while the card is opened, return -1304.

(2) If the -1211 is returned, it means that the PC may have gone to sleep, or the PC has turned on the fast boot, please do not sleep and

Turn the fast boot off, restart the PC and then open the card.

(3) If the return is -1206, it may be because the ECAT-M801/e-M901 has not been initialized yet, please open the card after the PC is turned on for 1 minute.

If you have been unable to open the card, please turn off the PC (please ensure that the shutdown process is completed, do not "restart" PC)After shut down, turn it on again, wait for 1 minute and open the card. If it still doesn't work (return -1206), please contact customer service staff.

**Syntax:**

int32_t ECAT_OpenDevice(uint16_t DeviceNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

# 5.3. ECAT_CloseDevice

**Description:**

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

**Syntax:**

int32_t ECAT_CloseDevice(uint16_t DeviceNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.4. ECAT_GetDeviceSerialNo

**Description:**

Get the hardware serial number.

**Syntax:**

int32_t ECAT_GetDeviceSerialNo(uint16_t DeviceNo, uint8_t *SerialNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

# 5.5. ECAT_GetDllVersion

**Description:**

Get the dll version.

**Syntax:**

int32_t ECAT_GetDllVersion(char *Version, uint16_t *Size);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| Version | char* | OUT | dll version |
| Size | uint16_t | OUT | size of Version Unit:byte |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

[C/C++]

```
int32_t ret;
char Version[512];
uint16_t size;
ret = ECAT_GetDllVersion(Version, &size);
if(ret < 0)
{
    printf("Failed to get dll version:%d\n", ret);
}
else
{
    printf("dll version = %s\n", Version);
}
```

# 5.6. ECAT_GetFirmwareVersion

**Description:**

Get the firmware version.

**Syntax:**

int32_t ECAT_GetFirmwareVersion(uint16_t DeviceNo, char *Version, uint16_t *Size);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| Version | char* | OUT | firmware version |
| Size | uint16_t | OUT | size of Version Unit:byte |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
char Version[255];
uint16_t size;
ret = ECAT_GetFirmwareVersion(DeviceNo ,Version, &size);
if(ret < 0)
{
    printf("Failed to get firmware version:%d\n", ret);
}
else
{
    printf("firmware version = %s\n", Version);
}
```

# 5.7. ECAT_GetDeviceDI

**Description:**

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

**Syntax:**

int32_t ECAT_GetDeviceDI(uint16_t DeviceNo, uint32_t *Value)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.8. ECAT_GetDeviceDIBit

Description:

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

Syntax:

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

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.9. ECAT_GetDeviceDO

Description:

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

Syntax:

int32_t ECAT_GetDeviceDO(uint16_t DeviceNo, uint32_t *Value)

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.10. ECAT_GetDeviceDOBit

Description:

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

Syntax:

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

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.11. ECAT_SetDeviceDO

Description:

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

Syntax:

int32_t ECAT_SetDeviceDO(uint16_t DeviceNo, uint32_t Value)

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.12. ECAT_SetDeviceDOBit

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.13. ECAT_SetDeviceEncProperty

**Description:**

　　Set the on-board encoder mode of a device.

**Syntax:**

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

**Parameters:**

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

**Return:**

　　0: Success.

　　Others: Refer to Appendix "Error Codes".

Table 5.1: Low Pass Filter Definition

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

| Example: |
|---|
| [C/C++] |

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

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

# 5.14. ECAT_GetDeviceEncProperty

**Description:**

Get the on-board encoder mode of a device.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.15. ECAT_GetDeviceEncCount

**Description:**

Get an on board encoder counter value of a device.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.16. ECAT_ResetDeviceEncCount

**Description:**

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

**Syntax:**

int32_t ECAT_ResetDeviceEncCount(uint16_t DeviceNo, uint16_t EncNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

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

# 5.17. ECAT_SetDeviceCmpTrigProperty

**Description:**

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

**Syntax:**

int32_t   ECAT_SetDeviceCmpTrigProperty(uint16_t   DeviceNo,   uint16_t   EncNo, uint32_t PulseWidth, uint8_t Source)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EncNo | uint16_t | IN | Encoder interface channel number (0 ~ 1) |
| PulseWidth | uint32_t | IN | Output Pulse width setting, the unit is 0.016us, and the maximum value is 0x7fffffff x 0.016us. |
| Source | uint8_t * | IN | DO output channel<br>0: Encoder channel 0 output to CMP1<br>    Encoder channel 1 output to CMP2<br>1: Encoder channel 0 output to CMP2<br>    Encoder channel 1 output to CMP1 |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

**[C/C++]**

```
int32_t ret;
uint16_t EncNo = 0;
uint32_t PulseWidth = 100000;
uint8_t Source = 0;
ret = ECAT_SetDeviceCmpTrigProperty(DeviceNo, EncNo, PulseWidth, Source);
if(ret != 0)
     printf("Failed to set compare triger property:%d\n", ret);
```

# 5.18. ECAT_GetDeviceCmpTrigProperty

**Description:**

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

**Syntax:**

int32_t ECAT_GetDeviceCmpTrigProperty(uint16_t DeviceNo, uint16_t EncNo, uint32_t *PulseWidth, uint8_t *Source)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EncNo | uint16_t | IN | Encoder interface channel number (0 ~ 1) |
| PulseWidth | uint32_t * | OUT | Output Pulse width setting value, the unit is 0.016us, and the maximum value is 0x7fffffff x 0.016us |
| Source | uint8_t * | OUT | DO output channel<br>0: Encoder channel 0 output to CMP1<br>   Encoder channel 1 output to CMP2<br>1: Encoder channel 0 output to CMP2<br>   Encoder channel 1 output to CMP1 |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t EncNo = 0;
uint32_t PulseWidth;
uint8_t Source;
ret = ECAT_GetDeviceCmpTrigProperty(DeviceNo, EncNo, &PulseWidth, &Source);
if(ret != 0)
{
    printf("Failed to get compare triger property:%d\n", ret);
}
else
{
    printf("Compare triger pulse width:%u\n", PulseWidth);
    printf("Compare triger source:%u\n", Source);
}
```

# 5.19. ECAT_SetDeviceCmpTrigData

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

**[C/C++]**

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

# 5.20. ECAT_SetDeviceContCmpTrigData

Description:

Start a continuous or a multiple compare-trigger function.

Syntax:

int32_t ECAT_SetDeviceContCmpTrigData(uint16_t DeviceNo, uint16_t EncNo, int32_t Start, uint32_t Interval, uint32_t Times, uint8_t Dir)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EncNo | uint16_t | IN | Encoder interface channel number (0 ~ 1) |
| Start | int32_t | IN | Start position for this compare-trigger operation |
| Interval | uint32_t | IN | Trigger interval (i.e. position increment) |
| Times | uint32_t | IN | Set 0 for continuous compare-trigger; a number greater than 0 is the number for multiple compare -trigger actions |
| Dir | uint8_t | IN | Compare direction 0: positive direction 1: negative direction |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

# 5.21. ECAT_SetDeviceEmg

**Description:**

Set the device emergency stop signal related configurations.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 5.2: Emergency stop signal source

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


| Example: |
|---|
| [C/C++] |

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

# 5.22. ECAT_GetDeviceEmg

**Description:**

Get the configurations of the device emergency stop signal.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.23. 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 (Card ID) |
| Status | uint8_t * | OUT | Emergency stop signal status |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint8_t Status;

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

# 5.24. ECAT_SetDeviceEmgSoftSig

Description:

Use this function to produce an emergency stop.

Syntax:

int32_t ECAT_SetDeviceEmgSoftSig (uint16_t DeviceNo)

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.25. ECAT_SetDeviceMPG

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 5.3: CON1 connector for MPG signal pin definitions

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

Table 5.4: CON2 connector MPG pin definitions

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

| Example: |
|---|
| [C/C++] |

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

```
AxisNo[1] = 1;

AxisNo[2] = 2;

AxisNo[3] = 3;


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

# 5.26. ECAT_GetDeviceMPG

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 5.27. ECAT_GetDeviceState

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 5.5: EtherCAT AL states

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

| Example: |
| --- |
| [C/C++] |

```
int32_t ret;
char buffer[1024];
char StrAlState[255];
uint16_t DeviceNo = 0;
uint32_t LinkUp, SlavesResp, AlState, Wc;
ret = ECAT_OpenDevice(DeviceNo);
ret = ECAT_GetDeviceState(DeviceNo, &LinkUp, &SlavesResp, &AlState, &Wc);
if(ret < 0)
    printf("Failed to get device state:%d\n", ret);
else
{
    if(AlState == ECAT_AS_INIT)
        sprintf(StrAlState, "INIT");
    else if(AlState == ECAT_AS_PREOP)
        sprintf(StrAlState, "PREOP");
    else if(AlState == ECAT_AS_SAFEOP)
        sprintf(StrAlState, "SAFEOP");
    else if(AlState == ECAT_AS_OP)
        sprintf(StrAlState, "OP");
    else
        sprintf(StrAlState, "Invalid");
    sprintf(buffer, "Slave(s):%u | AL State:%s | Link is :%s | Wc:%-u ",
                    SlavesResp, StrAlState, LinkUp? "up" : "down", Wc);
    printf("%s\n", buffer);
}
```

# 5.28. ECAT_StartDeviceOpTask

**Description:**

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

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| NetworkInfoNo | uint16_t | IN | Network information file number (Configured by the EtherCAT utility) |
| EnumCycleTime | uint8_t | IN | Cycle time number (Defined in Table 5.6) |
| WcErrCnt | uint32_t | IN | Counts of Working counter errors |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 5.6: Cycle time number

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

| Example: |
|---|
| [C/C++] |

int32_t ret;

uint16_t DeviceNo = 0;

uint16_t NetworkInfoNo = 0;

uint8_t EnumCycleTime = DEV_OP_CYCLE_TIME_1MS;

uint32_t WcErrCnt = 3;

int32_t flag = 1;

ret = ECAT_OpenDevice(DeviceNo);


ret = ECAT_StartDeviceOpTask(DeviceNo, NetworkInfoNo, EnumCycleTime, WcErrCnt);

if(ret < 0)

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

# 5.29. ECAT_StopDeviceOpTask

**Description:**

Stop the EtherCAT cyclic operation task.

**Syntax:**

int32_t ECAT_StopDeviceOpTask(uint16_t DeviceNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

# 5.30. ECAT_SetTimer

**Description:**

Set Timer Interval. This timer is only available when system is in OP state. In OP state, there is a cyclic task running inside ECAT_M801/ e-M901. This card takes advantage of this cyclic task to provide the host PC a timer function. *ECAT_SetTimer* function configures its time interval. A companion function *ECAT_WaitforTimer* is the one to wait for this interval. A thread calls *ECAT_WaitforTimer* will be suspended until time up.

**Syntax:**

int32_t ECAT_SetTimer(uint16_t DeviceNo, uint32_t Interval)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Interval = 100;


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

# 5.31. ECAT_SetTimerStop

**Description:**

Disable Timer. This timer is only available when system is in OP state. In OP state, there is a cyclic task running inside ECAT_M801/ e-M901. This card takes advantage of this cyclic task to provide the host PC a timer function. *ECAT_SetTimer* function configures its time interval. A companion function *ECAT_WaitforTimer* is the one to wait for this interval. A thread calls *ECAT_WaitforTimer* will be suspended until time up.

**Syntax:**

int32_t ECAT_SetTimerStop(uint16_t　DeviceNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
ret = ECAT_SetTimerStop(DeviceNo);
if(ret < 0)
{
    printf("Failed to Set Timer Stop:%d\n", ret);
}
else
{
    printf("Set Timer Stop successfully! \n");
}
```

# 5.32. ECAT_WaitforTimer

**Description:**

Wait until time up. This timer is only available when system is in OP state. In OP state, there is a cyclic task running inside ECAT_M801/ e-M901. This card takes advantage of this cyclic task to provide the host PC a timer function. *ECAT_SetTimer* function configures its time interval. A companion function *ECAT_WaitforTimer* is the one to wait for this interval. A thread calls *ECAT_WaitforTimer* will be suspended until time up.

**Syntax:**

int32_t ECAT_WaitforTimer(uint16_t DeviceNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Interval = 100;


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

# 5.33. ECAT_GetProcessTime

Description:

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

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

Syntax:

int32_t ECAT_GetProcessTime(uint16_t DeviceNo, double *Time)

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

# 5.34. ECAT_SetHeartBeat

Description:

Set heartbeat value.

After entering the OP and executing *ECAT_McInit*, If no command is executed for more than heartbeat value, the software emergency stop signal will be triggered.

Syntax:

int32_t ECAT_SetHeartBeat(uint16_t DeviceNo, uint32_t Value)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| Value | uint32_t | IN | heartbeat value<br>Unit: EtherCAT cycle time |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo   = 0;
uint32_t Value= 1000;

ret = ECAT_SetHeartBeat(DeviceNo, Value);
if(ret < 0)
{
    printf("Failed to Set heartbeat:%d\n", ret);
}
else
{
    printf("Set heartbeat successfully! \n");
}
```

# 5.35. ECAT_SetHeartBeatStatus

Description:

Set heartbeat function to be enabled or not.

After entering the OP and executing *ECAT_McInit*, If no command is executed for more than heartbeat value, the software emergency stop signal will be triggered.

Syntax:

int32_t ECAT_SetHeartBeatStatus(uint16_t DeviceNo, uint32_t Value)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| Value | uint32_t | IN | status<br>1: Enable<br>0: Disable |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t Value= 1000;

ret = ECAT_SetHeartBeatStatus(DeviceNo, Value);
if(ret < 0)
    printf("Failed to Set heartbeat:%d\n", ret);
else
    printf("Set heartbeat successfully! \n");

ret = ECAT_SetHeartBeat(DeviceNo, 1); //Enable
if(ret < 0)
{
    printf("Failed to Set heartbeat status:%d\n", ret);
}
else
{
    printf("Set heartbeat status successfully! \n");
}
```

# 5.36. ECAT_SetDeviceIgnoreWC

**Description:**

Enable/disable Ignore Working Counter check function.

After entering the OP, the master station and the slave station will start periodic communication. If the communication fails, the Working Counter will be missing. At this time, the DIO will not be able to control, and the motion axis will stop the current movement and change the status to MC_AS_ERRORSTOP, Last error is -1004.

When the status is OP and other modules are removed/connected, there will be communication failures for a short period of time (a few ms). This function is mainly used when it is necessary to remove/connect to other modules, first enable the ignore function, and then remove/connect the module. At this time, the axis status will not be switched to MC_AS_ERRORSTOP due to communication failure.

**Syntax:**

int32_t ECAT_SetDeviceIgnoreWC(uint16_t DeviceNo, int8_t Enable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| Enable | int8_t | IN | Enable/disable Ignore Working Counter check function<br>0: Disable (default)<br>1: Enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
int8_t Enable = 1;


ret = ECAT_SetDeviceIgnoreWC(DeviceNo, Enable);
if (ret != 0)
{
    printf("Failed to set device Ignore wc:%d\n", ret);
}
```

# 5.37. ECAT_GetDeviceIgnoreWC

**Description:**

Get the status of the Ignore Working Counter check function.

**Syntax:**

int32_t ECAT_GetDeviceIgnoreWC(uint16_t DeviceNo, int8_t *Enable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| Enable | int8_t | OUT | Enable/disable Ignore Working Counter check function<br>0: Disable (default)<br>1: Enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
int8_t Enable;

ret = ECAT_GetDeviceIgnoreWC(DeviceNo, &Enable);
if (ret != 0){
    printf("Failed to get ignore wc:%d\n", ret);
}
else{
    printf("ignore wc enable:%d\n", Enable);
}
```

# 6. Slave Operation Functions

## 6.1. ECAT_SetSlaveNoType

**Description:**

Define the slaveNo,

When the slaveNo type is SLAVE_NO_TYPE_POSITION, the slaveNo is the position of the module;

When the slaveNo type is SLAVE_NO_TYPE_ALIAS, the slaveNo is the module alias.

Take Figure 6.1 as an example:

When the slaveNo type is SLAVE_NO_TYPE_POSITION, the slaveNo "1" refers to ECAT-2028

When the slaveNo type is SLAVE_NO_TYPE_ALIAS, the slaveNo "1" refers to ECAT-2011H

Note:

(1) The position of the module refers to the position of the module in the EtherCAT network architecture (Master-Module 0-Module 1...)

(2) Module alias, which is not affected by the module connection order, can be set by Utility. For details, please refer to 3.1.2. Network Information Edit Steps.

Figure 6.1

**Syntax:**

int32_t ECAT_SetSlaveNoType(uint16_t DeviceNo, uint16_t Type)

**Parameters:**

| Name | Type | IN or OUT | Description |
| --- | --- | --- | --- |
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| Type | uint16_t | IN | SlaveNo type<br>Default: SLAVE_NO_TYPE_POSITION |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 6.1

| Macro Definition | Value | Description |
|---|---|---|
| SLAVE_NO_TYPE_POSITION | 0 | Position type |
| SLAVE_NO_TYPE_ALIAS | 1 | Alias Type |


| Example: |
|---|
| **[C/C++]** |

```
int32_t ret ;
uint16_t DeviceNo = 0;
uint16_t Type = SLAVE_NO_TYPE_ALIAS;
ret = ECAT_SetSlaveNoType(DeviceNo, Type);
if(ret < 0)
{
    printf("Failed to set slaveno type:%d\n", ret);
}
else
{
    printf("Set slaveno type successfully!\n");
}
```

# 6.2. ECAT_GetSlaveNoType

**Description:**

Get Definition of slaveNo,

When the slaveNo type is SLAVE_NO_TYPE_POSITION, the slaveNo is the position of the module;

When the slaveNo type is SLAVE_NO_TYPE_ALIAS, the slaveNo is the module alias.

Take Figure 6.1 as an example:

When the slaveNo type is SLAVE_NO_TYPE_POSITION, the slaveNo "1" refers to ECAT-2028

When the slaveNo type is SLAVE_NO_TYPE_ALIAS, the slaveNo "1" refers to ECAT-2011H

Note:

(1) The position of the module refers to the position of the module in the EtherCAT network architecture (Master-Module 0-Module 1...)

(2) Module alias, which is not affected by the module connection order, can be set by Utility. For details, please refer to 3.1.2. Network Information Edit Steps.

Figure 6.2

**Syntax:**

int32_t ECAT_GetSlaveNoType(uint16_t DeviceNo, uint16_t *Type)

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| Type | uint16_t* | OUT | SlaveNo type<br>Default: SLAVE_NO_TYPE_POSITION |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 6.2

| Macro Definition | Value | Description |
|---|---|---|
| SLAVE_NO_TYPE_POSITION | 0 | Position type |
| SLAVE_NO_TYPE_ALIAS | 1 | Alias Type |

| Example: |
|---|
| **[C/C++]** |
| int32_t ret ;<br>uint16_t DeviceNo = 0;<br>uint16_t Type;<br>ret = ECAT_GetSlaveNoType(DeviceNo, &Type);<br>if(ret < 0)<br>{<br>    printf("Failed to set slaveno type:%d\n", ret);<br>}<br>else<br>{<br>    printf("Get slaveno type successfully!\n");<br>} |

# 6.3. ECAT_GetSlaveInfo

## Description:

Get slave information of a slave.

When Slavename is blank, it means that the name cannot be read during PreOP, and the name obtained from ESI will be read after OP (V1.0.18 or above)

## 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, char *Slavename)

## Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| Alias | uint16_t* | OUT | Alias |
| ProductCode | uint32_t* | OUT | Product Code |
| VendorID | uint32_t* | OUT | Vendor ID |
| RevisionNo | uint32_t* | OUT | Revision number |
| SerialNo | uint32_t* | OUT | Serial number |
| AlState | uint8_t* | OUT | EtherCAT AL State of this slave |
| SlaveType | uint32_t* | OUT | Slave Type (Defined in Table 6.3) |
| Slavename | char* | OUT | Slave name |

## Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 6.3: Slave Type

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

| Example: |
|---|
| [C/C++] |

```
int32_t ret;
int16_t i;
uint16_t SlaveCnt;
uint16_t DeviceNo = 0;
uint16_t Alias;
uint32_t ProductCode, VendorID, RevisionNo, SerialNo, SlaveType;
uint8_t AlState;
ret = ECAT_OpenDevice(DeviceNo);
char Slavename[MAX_SLAVE_NAME_LENGTH];
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, &AlState, &SlaveType, Slavename);
        if(ret < 0)
        {
            printf("Failed to get slave information:%d\n",ret);
        }
```

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

# 6.4. ECAT_GetSlaveSdoObject

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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


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

# 6.5. ECAT_SetSlaveSdoObject

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.6. ECAT_SetSlaveRxPdoData

**Description:**

Set RxPDO data of a slave. Transfer process data to the RxPDO of a slave by means of cyclic communication. Digital outputs and analog outputs of slaves are set by RxPDO data.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret ;
uint16_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");
}
```

## Example:

### [C/C++]

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

//Example-ECAT-2055

typedef struct ecat_2055_t // 8DI    8DO
{
```

```
        uint8_t DI;

        uint8_t DO;

}ecat_2055_st;


ecat_2055_st    E2055;

OffsetByte = 0;

DataSize = 1; //1 bytes

E2055.DO = 0xFF;


ret = ECAT_SetSlaveRxPdoData(DeviceNo, SlaveNo, OffsetByte, DataSize, (uint8_t *)& E2055.DO);

if(ret < 0)

{

    printf("Failed to set RxPdo data:%d\n", ret);

}

else

{

    printf("Set RxPdo data successfully!\n");

}
```

# 6.7. ECAT_GetSlaveRxPdoData

**Description:**

Get RxPDO data of a slave. Read process data from the RxPDO of a slave by means of cyclic communication. Digital outputs or analog outputs of slaves are set by RxPDO data. To read the RxPDO data is to read back the status of these outputs.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret ;
uint16_t i;
uint16_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]);
    }
}
```

## Example:

### [C/C++]

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

//Example-ECAT-2024
```

```
typedef struct ecat_2024_t
{
    unsigned int Output            :16;
    unsigned int Gap               :16;
}ecat_2024_st;
ecat_2024_st   E2024;


OffsetByte = 0; / /VOUT 0
/ / OffsetByte = sizeof(E2024) * 1;   VOUT 1
DataSize = sizeof(E2024);


ret = ECAT_GetSlaveRxPdoData(DeviceNo, SlaveNo, OffsetByte, DataSize, (uint8_t *)&E2024);
if(ret < 0)
{
    printf("Failed to get RxPdo data:%d\n",ret);
}
else
{
    printf("AO Data: %u   \n", E2024.Output);
}
```

# 6.8. ECAT_GetSlaveTxPdoData

## Description:

Get TxPDO data of a slave. Read process data from the TxPDO of a slave by means of cyclic communication. TxPDO data are set by digital inputs or analog inputs of a slave. To read the TxPDO data is to read the status of these inputs.

## Syntax:

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

## Parameters:

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

## Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret ;
uint16_t i;
uint16_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]);
    }
}
```

## Example:

### [C/C++]

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

//Example-ECAT-2011-H
```

```
typedef struct ecat_2011h_t // V0
{
    unsigned int Underrange        : 1;
    unsigned int Overrange         : 1;
    unsigned int Limit1            : 2;
    unsigned int Limit2            : 2;
    unsigned int Error             : 1;
    unsigned int Gap1              : 1;
    unsigned int Gap2              : 6;
    unsigned int TxPDO_State       : 1;
    unsigned int TxPDO_Toggle      : 1;
    unsigned int Value             : 16;
}ecat_2011h_st;
ecat_2011h_st   E2011H;

OffsetByte = 0; // V0
/ / OffsetByte = sizeof(E2011H) * 1;   V 1
DataSize = sizeof( E2011H );

ret = ECAT_GetSlaveTxPdoData(DeviceNo, SlaveNo, OffsetByte, DataSize, (uint8_t *)&E2011H);
if(ret < 0)
{
    printf("Failed to get TxPdo data:%d\n", ret);
}
else
{
    printf("AI Data: %u \n", E2011H.Value);
}
```

# 6.9. ECAT_GetSlaveDI

## Description:

Get the digital input data of a slave. If a slave is a simple digital input slave, users can use this API to get DI values. Function *ECAT_GetSlaveTxPdoData* can also do it; but users have to enter more parameters for the same purpose. This function is limited to read up to 32 digital inputs of a 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 (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| Value | uint32_t* | OUT | Digital input data |

## Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.10. ECAT_GetSlaveDIBit

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

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

# 6.11. ECAT_GetSlaveDO

**Description:**

Get the digital output data of a slave. If a slave is a simple digital output slave, users can use this API to get DO states. Function *ECAT_GetSlaveRxPdoData* can also do it; but users have to enter more parameters for the same purpose. This function is limited to read up to 32 digital outputs of a 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".

## Example:

**[C/C++]**

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

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

# 6.12. ECAT_GetMultiSlaveDO

**Description:**

Get the digital output data of multiple slaves.

**Syntax:**

int32_t ECAT_GetMultiSlaveDO(uint16_t DeviceNo, uint16_t SlaveNo[], uint32_t *Value, uint16_t SlaveCnt)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t [] | IN | Slave number |
| Value | uint32_t* | OUT | Digital output data |
| SlaveCnt | uint16_t | IN | Number of modules |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo [MULTI_SLAVE_DO_MAX];
uint32_t Value[MULTI_SLAVE_DO_MAX];
uint16_t SlaveCnt;

SlaveNo[0] = 0;
SlaveNo[1] = 1;
SlaveNo[2] = 2;

SlaveCnt = 3;

ret = ECAT_GetMultiSlaveDO(DeviceNo, SlaveNo, Value, SlaveCnt);
if(ret < 0)
    printf("Failed to get slave DO:%d\n",ret);
else
{
    printf("DO[0]:%x, DO[1]:%x, DO[2]:%x \n", Value[0] , Value[1] , Value[2]);
}
```

# 6.13. ECAT_GetSlaveDOBit

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.14. ECAT_SetSlaveDO

**Description:**

Set the digital output data of a slave. If a slave is a simple digital input slave, users can use this API to set DO values. Function *ECAT_SetSlaveRxPdoData* can also do it; but users have to enter more parameters for the same purpose. This function is limited to set up to 32 digital outputs of a 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 (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| Value | uint32_t | IN | Digital output data |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.15. ECAT_SetMultiSlaveDO

**Description:**

Set the digital output data of multiple slaves at the same time.

**Syntax:**

int32_t ECAT_SetMultiSlaveDO(uint16_t DeviceNo, uint16_t SlaveNo[], uint32_t Value[], uint16_t SlaveCnt)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t [] | IN | Slave number |
| Value | uint32_t [] | IN | Digital output data |
| SlaveCnt | uint16_t | IN | Number of modules |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo [MULTI_SLAVE_DO_MAX];
uint32_t Value[MULTI_SLAVE_DO_MAX];
uint16_t SlaveCnt;

/*
Master
|- ECAT-2057
|- ECAT-2057
L ECAT-2057
*/

SlaveNo[0] = 0;
Value[0] = 0xFFFF;

SlaveNo[1] = 1;
Value[1] = 0xFFFF;

SlaveNo[2] = 2;
Value[2] = 0xFFFF;

SlaveCnt = 3;

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

# 6.16. ECAT_SetMultiSlaveDO_AutoOff

**Description:**

Set the digital output data of multiple slaves at the same time.

After the specified time (Width), the specified DO bit (Mask) will be turned off.

**Syntax:**

int32_t    ECAT_SetMultiSlaveDO_AutoOff(uint16_t    DeviceNo,    uint16_t    SlaveNo[], uint32_t Value[], uint32_t Width[], uint32_t Mask[], uint16_t SlaveCnt)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t [] | IN | Slave number |
| Value | uint32_t [] | IN | Digital output data |
| Width | uint32_t [] | IN | Time, multiple of EtherCAT cycle time |
| Mask | uint32_t [] | IN | Mask, only BIT with mask setting will be turned off |
| SlaveCnt | uint16_t | IN | Number of modules |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo [MULTI_SLAVE_DO_MAX];
uint32_t Value[MULTI_SLAVE_DO_MAX];
uint32_t Width[MULTI_SLAVE_DO_MAX];
uint32_t Mask[MULTI_SLAVE_DO_MAX];
uint16_t SlaveCnt;

/*
Master
|- ECAT-2057
|- ECAT-2057
L ECAT-2057
*/

SlaveNo[0] = 0;
Value[0] = 0xFFFF;
Width [0] = 30; //if EtherCAT cycle time is 1ms, then Pulse Width is 1ms*30 = 30ms
Mask [0] = 0xFFFF; //after 30ms(Width), 16 bits of DO will be truned off

SlaveNo[1] = 1;
Value[1] = 0xFFFF;
Width [1] = 30;
Mask [1] = 0x00FF; //after 30ms(Width), the first 8 bits of DO will be truned off, and the last 8bits of DO will
remain on

SlaveNo[2] = 2;
Value[2] = 0xFFFF;
Width [2] = 30;
Mask [2] = 0xFF00; //after 30ms(Width), the last 8 bits of DO will be truned off, and the first 8bits of DO will
remain on
SlaveCnt = 3;

ret = ECAT_SetMultiSlaveDO_AutoOff(DeviceNo, SlaveNo, Value, Width, Mask, SlaveCnt);
if(ret < 0)
```

```
        printf("Failed to set slave DO:%d\n", ret);
else
        printf("Set slave DO successfully! \n");
```

# 6.17. ECAT_SetSlaveDOBit

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.18. ECAT_SetSlaveAoProperty

**Description:**

Set the AO channel property value. Each AO channel can has different range setting from the others.

Note: It supports **ECAT-2024 and ECAT-2028.**

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| ChannelNo | uint16_t | IN | Channel number |
| Range | uint8_t | IN | AO range code (Defined in Table 6.4) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 6.4: 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 settings successfully! \n");
```

# 6.19. ECAT_GetSlaveAoProperty

**Description:**

Get the AO channel property value. Each AO channel can has different range setting from the others.

Note: It supports **ECAT-2024 and ECAT-2028.**

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| ChannelNo | uint16_t | IN | Channel number |
| Range | uint8_t * | OUT | AO range code (Defined in Table 6.4) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.20. ECAT_SetSlaveAoRawData

**Description:**

Set the 16-bit integer value of an analog output channel.

Note: It supports **ECAT-2024 and ECAT-2028.**

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.21. ECAT_GetSlaveAoRawData

**Description:**

Get the 16-bit integer value of an analog output channel.

Note: It supports **ECAT-2024 and ECAT-2028.**

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

**[C/C++]**

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

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

# 6.22. ECAT_SetSlaveAoVoltData

**Description:**

Set the floating-point voltage output value of a specified analog output channel.

Note: It supports **ECAT-2024 and ECAT-2028.**

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.23. ECAT_GetSlaveAoVoltData

**Description:**

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

Note: It supports **ECAT-2024 and ECAT-2028.**

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

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

# 6.24. ECAT_SetSlaveAiProperty

**Description:**

Set the AI channel property value. Each AI channel can has different range setting from others.

Note: It supports **ECAT-2011H、ECAT-2012H.**

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| ChannelNo | uint16_t | IN | Channel number |
| Range | uint8_t | IN | AI range code (Defined in Table 6.5) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 6.5: AI range codes

| Macro Definition | Value | Description |
|---|---|---|
| SLAVE_AI_BI_10V | 0 | ±10V |
| SLAVE_AI_BI_5V | 1 | ±5V |
| SLAVE_AI_BI_2_5V | 2 | ±2.5V |
| SLAVE_AI_UNI_10V | 3 | 0 ~ 10V |
| SLAVE_AI_UNI_20mA | 4 | 0 ~ 20mA |
| SLAVE_AI_UNI_4_20mA | 5 | 4 ~ 20mA |
| SLAVE_AI_BI_20mA | 6 | ±0 ~ 20mA |
| SLAVE_AI_BI_4_20mA | 7 | ±4 ~ 20mA |
| SLAVE_AI_BI_10V_UNI_20mA | 8 | CH0~3 ±10V, CH4~7 0 ~ 20mA |
| SLAVE_AI_BI_10V_UNI_4_20mA | 9 | CH0~3 ±10V, CH4~7 4 ~ 20mA |
| SLAVE_AI_BI_10V_BI_20mA | 10 | CH0~3 ±10V, CH4~7 ±0 ~ 20mA |
| SLAVE_AI_BI_10V_BI_4_20mA | 11 | CH0~3 ±10V, CH4~7 ±4 ~ 20mA |

**Example:**

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
uint8_t Range = SLAVE_AI_BI_10V;
ret = ECAT_SetSlaveAiProperty(DeviceNo, SlaveNo, ChannelNo, Range);
if(ret != 0)
    printf("Failed to set slave AI settings:%d\n", ret);
else
    printf("Set slave AI settings successfully! \n");
```

# 6.25. ECAT_GetSlaveAiProperty

**Description:**

Get the AI channel property value. Each AI channel can has different range setting from others.

Note: It supports **ECAT-2011H、ECAT-2012H.**

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| ChannelNo | uint16_t | IN | Channel number |
| Range | uint8_t * | OUT | AI range code (Defined in Table 6.5) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.26. ECAT_GetSlaveAiRawData

**Description:**

Get the 16-bit integer value of an analog output channel.

Note: It supports **ECAT-2011H、ECAT-2012H.**

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| ChannelNo | uint16_t | IN | Channel number |
| Data | int16_t * | OUT | AI raw value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.27. ECAT_GetSlaveAiVoltData

**Description:**

Get the floating-point voltage value of an analog input channel. Please make sure the channel property is set correctly.

Note: It supports **ECAT-2011H、ECAT-2012H.**

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| ChannelNo | uint16_t | IN | Channel number |
| Data | Double * | OUT | AI floating-point voltage value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.28. ECAT_GetSlaveAimAData

**Description:**

Get the floating-point milliampere value of an analog input channel. Please make sure the channel property is set correctly.

Note: It supports **ECAT-2011H、ECAT-2012H.**

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| ChannelNo | uint16_t | IN | Channel number |
| Data | Double * | OUT | A floating-point milliampere AI value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.29. ECAT_Set_ECAT2016_AiProperty

**Description:**

Set the AI channel property value. Each AI channel can has different range setting from others.

Note:

(1) It supports **ECAT-2016N.**

(2) Change any one channel property value will change the property values in the other remaining channels.

**Syntax:**

int32_t ECAT_Set_ECAT2016_AiProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, uint8_t Range, uint32_t *AbortCode)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| ChannelNo | uint16_t | IN | Channel number |
| Range | uint8_t | IN | AI range code (Defined in Table 6.6) |
| AbortCode | uint32_t* | OUT | Abort code of the SDO (Please refer to Appendix "SDO Abort messages") |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 6.6: AI Range Code

| Macro Definition | Value | Description |
|---|---|---|
| ECAT2016_AI_BI_10V | 0 | ±10V |
| ECAT2016_AI_BI_5V | 1 | ±5V |
| ECAT2016_AI_BI_2_5V | 2 | ±2.5V |
| ECAT2016_AI_BI_1V | 3 | ±1V |
| ECAT2016_AI_BI_500mV | 4 | ±500mV |
| ECAT2016_AI_BI_100mV | 5 | ±100mV |
| ECAT2016_AI_BI_50mV | 6 | ±50mV |
| ECAT2016_AI_BI_25mV | 7 | ±25mV |
| ECAT2016_AI_BI_20mV | 8 | ±20mV |
| ECAT2016_AI_BI_16mV | 9 | ±16mV |
| ECAT2016_AI_BI_15mV | 10 | ±15mV |

| Example: |
|---|
| [C/C++] |

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
uint8_t Range = ECAT2016_AI_BI_10V;
uint32_t Abortcode;
ret = ECAT_Set_ECAT2016_AiProperty(DeviceNo, SlaveNo, ChannelNo, Range, &Abortcode);
if(ret != 0)
    printf("Failed to set slave AI settings:%d\n", ret);
else
    printf("Set slave AI settings successfully! \n");
```

# 6.30. ECAT_Get_ECAT2016_AiProperty

**Description:**

Get the AI channel property value. Each AI channel can has different range setting from others.

Note: It supports **ECAT-2016N.**

**Syntax:**

int32_t ECAT_Get_ECAT2016_AiProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t *ChannelNo, uint8_t *Range, uint32_t *AbortCode)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| ChannelNo | uint16_t | IN | Channel number |
| Range | uint8_t * | OUT | AI range code (Defined in Table 6.6) |
| AbortCode | uint32_t* | OUT | Abort code of the SDO (Please refer to Appendix "SDO Abort messages") |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t ChannelNo = 0;
uint8_t Range;
uint32_t Abortcode;

ret = ECAT_Get_ECAT2016_AiProperty(DeviceNo, SlaveNo, ChannelNo, &Range, &Abortcode);
if(ret != 0)
    printf("Failed to get slave AI settings:%d\n", ret);
else
    printf("AI range:%d\n", Range);
```

# 6.31. ECAT_Get_ECAT2016_AiRawData

**Description:**

Get the 16-bit integer value of an analog input channel.

Note: It supports **ECAT-2016N.**

**Syntax:**

int32_t ECAT_Get_ECAT2016_AiRawData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, int16_t *Data)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| ChannelNo | uint16_t | IN | Channel number |
| Data | int16_t * | OUT | AI raw value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

ret = ECAT_Get_ECAT2016_AiRawData(DeviceNo, SlaveNo, ChannelNo, &Data);
if(ret != 0)
    printf("Failed to get slave AI raw data:%d\n", ret);
else
    printf("AI raw data:%d\n", Data);
```

# 6.32. ECAT_Get_ECAT2016_AiVoltData

**Description:**

Get the floating-point voltage value of an analog input channel. Please make sure the channel property is set correctly.

Note: It supports **ECAT-2016N.**

**Syntax:**

int32_t ECAT_Get_ECAT2016_AiVoltData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t ChannelNo, double *Data)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| ChannelNo | uint16_t | IN | Channel number |
| Data | Double * | OUT | AI floating-point voltage value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

ret = ECAT_Get_ECAT2016_AiVoltData(DeviceNo, SlaveNo, ChannelNo, &Data);
if(ret != 0)
    printf("Failed to get slave AI volt data:%d\n", ret);
else
    printf("AI volt data:%d\n", Data);
```

# 6.33. ECAT_SetSlaveEncProperty

**Description:**

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

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface channel number |
| Mode | uint8_t | IN | Encoder Mode<br>1: CW/CCW<br>2: Pulse/Dir<br>3: A/B Phase |
| InvertCnt | uint8_t | IN | change the counting direction |
| LPF | uint8_t | IN | Set low pass filter (Defined in Table 6.7) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 6.7: Low pass filter

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

| Example: |
| --- |
| [C/C++] |

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

# 6.34. ECAT_GetSlaveEncProperty

**Description:**

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

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| Mode | uint8_t * | OUT | Encoder Mode<br>1: CW/CCW<br>2: Pulse/Dir<br>3: A/B Phase |
| InvertCnt | uint8_t * | OUT | change counting direction |
| LPF | uint8_t * | OUT | Set low pass filter (Defined in Table 6.7) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

# 6.35. ECAT_GetSlaveEncCount

Description:

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

Syntax:

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

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

# 6.36. ECAT_ResetSlaveEncCount

**Description:**

Clear the encoder counter value to 0. This function is designed for encoder module ECAT-2093 and ECAT-2092T.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

# 6.37. ECAT_SetSlaveEncCount

**Description:**

Set the encoder counter value. This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t ECAT_SetSlaveEncCount(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, int32_t Value)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t Value = 100;

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

# 6.38. ECAT_SetSlaveEncIdxLatchProperty

**Description:**

Set the position index latch function property to be enabled or not. This function is designed for encoder module ECAT-2093 and ECAT-2092T.

Note: The index latch of ECAT-2092T is always enabled and can not be disabled.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.39. ECAT_GetSlaveEncIdxLatchProperty

Description:

Get the position index latch function setting. This function is designed for encoder module ECAT-2093 and ECAT-2092T.

Syntax:

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

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.40. ECAT_GetSlaveEncIdxLatchCnt

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

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

# 6.41. ECAT_ResetSlaveEncIdxLatchCnt

**Description:**

Reset the index latch count. This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t    ECAT_ResetSlaveEncIdxLatchCnt(uint16_t    DeviceNo,    uint16_t    SlaveNo, uint16_t EncNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

ret = ECAT_ResetSlaveEncIdxLatchCnt(DeviceNo, SlaveNo, EncNo);
if(ret != 0)
{
    printf("Failed to reset index latch count:%d\n", ret);
}
else
{
    printf("Index latch count:%u\n");
}
```

# 6.42. ECAT_SetSlaveEncIdxLatchBufferEnable

Description:

Set the position index latch buffer(FIFO) function property to be enabled or not. This function is designed for encoder module ECAT-2093

Note: The latch frequency is related to the EtherCAT communication cycle. For example: if the communication cycle is 1ms, the maximum latch frequency is 1kHz.

Syntax:

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

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.43. ECAT_GetSlaveEncIdxLatchBufferEnable

**Description:**

Get the position index latch buffer(FIFO) function setting. This function is designed for encoder module ECAT-2093.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.44. ECAT_GetSlaveEncIdxLatchBuffer

**Description:**

Read the index latch buffer(FIFO). This function is designed for encoder module ECAT-2093.

Note:(1) After reading the data, the data will be removed from the buffer

(2) When the buffer is full, the buffer will no longer update data until there is room in the buffer

**Syntax:**

int32_t ECAT_GetSlaveEncIdxLatchBuffer(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, int32_t GetCount, int32_t *ActualGetCount, int32_t *Data)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| GetCount | int32_t | IN | Quantity of records to get |
| ActualGetCount | int32_t * | OUT | Actual quantity of data gotten |
| Data | int32_t * | OUT | Latched data |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t GetCount = ENC_BUFFER_MAX;
int32_t ActualGetCount;
int32_t Data[ENC_BUFFER_MAX];


ret = ECAT_GetSlaveEncIdxLatchBuffer(DeviceNo, SlaveNo, EncNo, GetCount, &ActualGetCount, Data);
if(ret != 0)
{
    printf("Failed to get index latch buffer data:%d\n",ret);
}
else
{
    for(uint16_t i = 0, i < ActualGetCount, i++)
    {
        printf("Data[%u]:%f\n", i, Data[ i ]);
    }
}
```

# 6.45. ECAT_ResetSlaveEncIdxLatchBuffer

**Description:**

Reset the index latch buffer(FIFO). This function is designed for encoder module ECAT-2093.

**Syntax:**

int32_t  ECAT_ResetSlaveEncIdxLatchBuffer(uint16_t  DeviceNo,  uint16_t  SlaveNo, uint16_t EncNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

ret = ECAT_ResetSlaveEncIdxLatchBuffer(DeviceNo, SlaveNo, EncNo);
if(ret != 0)
{
    printf("Failed to reset index latch buffer:%d\n",ret);
}
else
{
    printf("Reset index latch buffer successfully!\n");
}
```

# 6.46. ECAT_SetSlaveEncExtLatchProperty

**Description:**

Set the position external latch function property. This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t  ECAT_SetSlaveEncExtLatchProperty(uint16_t  DeviceNo,  uint16_t  SlaveNo, uint16_t EncNo, uint8_t Enable, uint8_t Mode, uint8_t Logic)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| Enable | uint8_t | IN | Enable/Disable latch |
| Mode | uint8_t | IN | Latch mode<br>0: Reset encoder counter<br>1: Latch encoder counter |
| Logic | uint8_t | IN | Extern latch signal polarity<br>0: Active low<br>1: Active high |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.47. ECAT_GetSlaveEncExtLatchProperty

**Description:**

Get the position external latch function setting. This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t ECAT_GetSlaveEncExtLatchProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t *Enable, uint8_t *Mode, uint8_t *Logic)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| Enable | uint8_t * | OUT | Enable/Disable latch |
| Mode | uint8_t | IN | Latch mode<br>0: Reset encoder counter<br>1: Latch encoder counter |
| Logic | uint8_t | IN | Extern latch signal polarity<br>0: Active low<br>1: Active high |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Enable, Mode, Logic;

ret = ECAT_GetSlaveEncExtLatchProperty(DeviceNo, SlaveNo, EncNo, &Enable, &Mode, &Logic);
if(ret != 0)
{
    printf("Failed to get external latch property:%d\n",ret);
}
else
{
    printf("External latch enable:%u\n",    Enable);
    printf("External latch mode:%u\n",    Mode);
    printf("External latch logic:%u\n",    Logic);
}
```

# 6.48. ECAT_GetSlaveEncExtLatchCnt

**Description:**

Read the external latch count. This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t ECAT_GetSlaveEncExtLatchCnt(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, int32_t *Cnt_Rising, int32_t *Cnt_Falling)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| Cnt_Rising | int32_t * | OUT | Rising edge latched value |
| Cnt_Falling | int32_t * | OUT | Falling edge latched value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t Cnt_Rising;
int32_t Cnt_Falling;

ret = ECAT_GetSlaveEncExtLatchCnt(DeviceNo, SlaveNo, EncNo, &Cnt_Rising, &Cnt_Falling);
if(ret != 0)
{
    printf("Failed to get external latch count:%d\n",ret);
}
else
{
    printf("External latch count:%d, %d\n", Cnt_Rising, Cnt_Falling);
}
```

# 6.49. ECAT_ResetSlaveEncExtLatchCnt

**Description:**

Reset the external latch count. This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t ECAT_ResetSlaveEncExtLatchCnt(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

**[C/C++]**

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

ret = ECAT_ResetSlaveEncExtLatchCnt(DeviceNo, SlaveNo, EncNo, &Cnt);
if(ret != 0)
{
    printf("Failed to reset external latch count:%d\n", ret);
}
else
{
    printf("External latch count:%u\n");
}
```

# 6.50. ECAT_SetSlaveCmpTrigProperty

**Description:**

Set the compare-trigger related properties. This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t ECAT_SetSlaveCmpTrigProperty(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint16_t PulseWidth)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| PulseWidth | uint16_t | IN | Compare Trigger Pulse Width (Defined in Table 6.8) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 6.8: Compare Trigger Pulse Width

| Pulse Width Setting | Actual Pulse Width (µSec) |
|---|---|
| 2 | 2 |
| … | … |
| 50 (default) | 50 |
| … | … |
| 32766 | 32766 |
| 32767 | 32767 |

| Example: |
|---|
| [C/C++] |

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint16_t PulseWidth = 40;
ret = ECAT_SetSlaveCmpTrigProperty(DeviceNo, SlaveNo, EncNo, PulseWidth);
if(ret != 0)
    printf("Failed to set compare triger property:%d\n", ret);
```

# 6.51. ECAT_GetSlaveCmpTrigProperty

**Description:**

Get the compare-trigger related properties. This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t   ECAT_GetSlaveCmpTrigProperty(uint16_t   DeviceNo,   uint16_t   SlaveNo, uint16_t EncNo, uint16_t *PulseWidth)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| PulseWidth | uint16_t * | OUT | Compare Trigger Pulse Width (Defined in Table 6.8) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

# 6.52. ECAT_SetSlaveCmpTrigData

**Description:**

According to the setting value, start a single compare-trigger function for an on-board encoder interface channel. This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t ECAT_SetSlaveCmpTrigData(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, int32_t CmpData)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| CmpData | int32_t | IN | Single compare-trigger data |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

# 6.53. ECAT_SetSlaveContCmpTrigData

**Description:**

Start a continuous or a multiple compare-trigger function. This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t  ECAT_SetSlaveContCmpTrigData(uint16_t  DeviceNo,  uint16_t  SlaveNo, uint16_t EncNo, int32_t Start, uint32_t Interval, uint8_t Dir)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| Start | int32_t | IN | Start position for this compare-trigger operation |
| Interval | uint32_t | IN | Trigger interval (i.e. position increment) |
| Dir | uint8_t | IN | Compare direction<br>0: positive direction<br>1: negative direction |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
int32_t Start = 1000;
uint32_t Interval = 200;
uint8_t Dir = 0;
ret = ECAT_SetSlaveContCmpTrigData(DeviceNo, SlaveNo, EncNo, Start, Interval, Dir);
if(ret != 0)
    printf("Failed to set continus compare triger data:%d\n", ret);
```

# 6.54. ECAT_SetSlaveArrCmpPos

**Description:**

Set the array comparison position. After enabling the array comparison trigger function, the comparison will start from the comparison position stored in index 0. When the encoder reaches the comparison position, it will trigger the digital output.

Note: (1) This function is designed for encoder module ECAT-2092T.

(2) The first compare value has to be stored in index 0, the second in index 1, the third in index 2, etc..

(3) The value stored in index 0 as the first compare position and it remains in the compare register until the encoder counter reaches this position before using the index 1 value as the next compare position.

**Example:**

In Table 6.9 the compare values are set to 150, 250, 350 and 500. Digital output will be triggered for these positions (Figure 6.3).

Table 6.9

| Index | 0 | 1 | 2 | 3 |
|----------|-----|-----|-----|-----|
| Position | 150 | 250 | 350 | 500 |

Figure 6.3
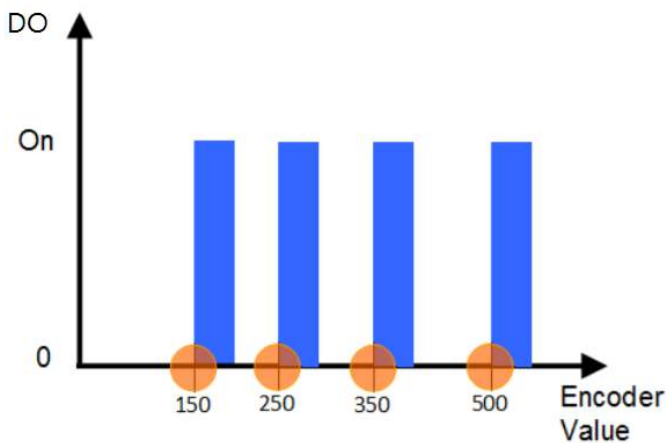


**Syntax:**

int32_t  ECAT_SetSlaveArrCmpPos(uint16_t  DeviceNo,  uint16_t  SlaveNo,  uint16_t

EncNo, uint8_t Index, int32_t Position)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| Index | uint8_t | IN | Index number<br>Range: 0~199 |
| Position | int32_t | IN | Compare position |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Index;
int32_t Position;

Index= 0;
Position= 150;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 1;
Position= 250;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 2;
Position= 350;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 3;
Position= 500;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);
```

# 6.55. ECAT_GetSlaveArrCmpPos

**Description:**

Get the array comparison position.

Note: This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t  ECAT_GetSlaveArrCmpPos(uint16_t  DeviceNo,  uint16_t  SlaveNo,  uint16_t EncNo, uint8_t Index, int32_t *Position)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| Index | uint8_t | IN | Index number<br>Range: 0~199 |
| Position | int32_t* | OUT | Compare position |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Index;
int32_t Position;

Index= 0;
ret = ECAT_GetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, &Position);
if(ret != 0)
    printf("Failed to get array compare position:%d\n", ret);
else
    printf("Index:%u, position:%d \n"  , Index, Position);
```

# 6.56. ECAT_SetSlaveArrCmpEnable

**Description:**

Set "DO trigger" flag, which indicates whether a digital output will be triggered if the encoder value reaches this position value.

Note: This function is designed for encoder module ECAT-2092T.

**Example:**

In Table 6.10 the compare values are set to 150, 250, 350 and 500. The "DO Trigger" flag for index 1 is disabled therefore no digital output will be triggered for this position (Figure 6.4).

Table 6.10

| Index | 0 | 1 | 2 | 3 |
|-------|-----|-----|-----|-----|
| Position | 150 | 250 | 350 | 500 |
| Enable | 1 | 0 | 1 | 1 |

Figure 6.4



**Syntax:**

int32_t ECAT_SetSlaveArrCmpEnable(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t Index, uint8_t Enable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|

| DeviceNo | uint16_t | IN | Device number (Card ID) |
|---|---|---|---|
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| Index | uint8_t | IN | Index number<br>Range: 0~199 |
| Enable | uint8_t | IN | Enable/Disable<br>0: Disable<br>1: Enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

Index= 0;
Position= 150;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 1;
Position= 250;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 2;
Position= 350;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 3;
Position= 500;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 0;
Enable= 1;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
```

```
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);


Index= 1;
Enable= 0;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);


Index= 2;
Enable= 1;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);


Index= 3;
Enable= 1;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);
```

# 6.57. ECAT_GetSlaveArrCmpEnable

**Description:**

Get "DO trigger" flag.

Note: This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t ECAT_GetSlaveArrCmpEnable(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint8_t Index, uint8_t *Enable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| Index | uint8_t | IN | Index number<br>Range: 0~199 |
| Enable | uint8_t* | OUT | Enable/Disable<br>0: Disable<br>1: Enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

**[C/C++]**

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

Index= 0;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, &Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);
else
    printf("Index:%u, enable:%u \n"  , Index, Enable);
```

# 6.58. ECAT_SetSlaveArrCmpEndIdx

**Description:**

Set the end index to be used for the compare process.

When the encoder position has reached the compare position stored in end index then the next compare position will start again with the position stored index 0.

Note: This function is designed for encoder module ECAT-2092T.

Example:

In Table 6.11 the compare values are set to 150, 250, 350 and 500. The "DO Trigger" flag for index 1 is disabled therefore no digital output will be triggered for this position Figure 6.5). End index set to 3.

Table 6.11

| Index | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Position | 150 | 250 | 350 | 500 |
| Enable | 1 | 0 | 1 | 1 |
| End index | 3 | | | |

Figure 6.5
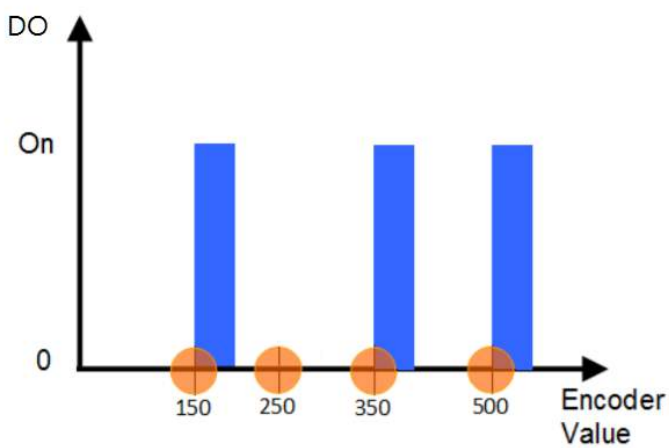


**Syntax:**

int32_t ECAT_SetSlaveArrCmpEndIdx(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint16_t EndIndex)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| EndIndex | uint16_t | IN | Index numberRange<br>Range: 0~199 |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint8_t Index;
int32_t Position;
uint8_t Enable;
uint16_t End_Index;

Index= 0;
Position= 150;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 1;
Position= 250;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 2;
Position= 350;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 3;
Position= 500;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);

Index= 0;
Enable= 1;
```

```
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);


Index= 1;
Enable= 0;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);


Index= 2;
Enable= 1;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);


Index= 3;
Enable= 1;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);


End_Index= 3;
ret = ECAT_SetSlaveArrCmpEndIdx(DeviceNo, SlaveNo, EncNo, End_Index);
if(ret != 0)
    printf("Failed to set array compare end index:%d\n", ret);
```

# 6.59. ECAT_GetSlaveArrCmpEndIdx

## Description:

Get the end index.

Note: This function is designed for encoder module ECAT-2092T.

## Syntax:

int32_t ECAT_GetSlaveArrCmpEndIdx(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo, uint16_t *EndIndex)

## Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |
| EncNo | uint16_t | IN | Encoder interface number |
| EndIndex | uint16_t* | OUT | Index numberRange<br>Range: 0~199 |

## Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
uint16_t EncNo = 0;
uint16_t End_Index;

ret = ECAT_GetSlaveArrCmpEndIdx(DeviceNo, SlaveNo, EncNo, &End_Index);
if(ret != 0)
    printf("Failed to set array compare end index:%d\n", ret);
else
    printf("End Index:%u\n" , End_Index);
```

# 6.60. ECAT_SetSlaveArrCmpTrig

**Description:**

Start Array compare-trigger function.

Note: This function is designed for encoder module ECAT-2092T.

**Syntax:**

int32_t ECAT_SetSlaveArrCmpTrig(uint16_t DeviceNo, uint16_t SlaveNo, uint16_t EncNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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


Index= 0;
Position= 150;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);


Index= 1;
Position= 250;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);


Index= 2;
Position= 350;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);


Index= 3;
Position= 500;
ret = ECAT_SetSlaveArrCmpPos(DeviceNo, SlaveNo, EncNo, Index, Position);
if(ret != 0)
    printf("Failed to set array compare position:%d\n", ret);


Index= 0;
Enable= 1;
```

```
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);


Index= 1;
Enable= 0;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);


Index= 2;
Enable= 1;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);


Index= 3;
Enable= 1;
ret = ECAT_SetSlaveArrCmpEnable(DeviceNo, SlaveNo, EncNo, Index, Enable);
if(ret != 0)
    printf("Failed to set array compare enable:%d\n", ret);


End_Index= 3;
ret = ECAT_SetSlaveArrCmpEndIdx(DeviceNo, SlaveNo, EncNo, End_Index);
if(ret != 0)
    printf("Failed to set array compare end index:%d\n", ret);


ret = ECAT_SetSlaveArrCmpTrig(DeviceNo, SlaveNo, EncNo);
if(ret != 0)
    printf("Failed to set array compare trigger:%d\n", ret);
```

# 6.61. ECAT_SetSlaveSaveArrCmpData

### Description:

Store the array comparison position and the "DO trigger" flag in the EEPROM of the module, and the data will not be lost when the power is turned off.

Note: (1) This function is designed for encoder module ECAT-2092T.

(2) It needs to be set in the PreOP state. If the module is in the OP state, an error will be returned.

(3) This function is a blocking function. It returns about 10 seconds after execution. If the return value is not zero, please re-power the module and set the data again.

### Syntax:

int32_t ECAT_SetSlaveSaveArrCmpData(uint16_t DeviceNo, uint16_t SlaveNo);

### Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave number |

### Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t SlaveNo = 0;
ret = ECAT_SetSlaveSaveArrCmpData(DeviceNo, SlaveNo);
if(ret != 0)
    printf("Failed to save array compare data:%d\n", ret);
```

# 6.62. ECAT_SetTxPdoBufParam

Description:

Set parameters of a TxPdo buffer

TxPdo buffer will store the values of the last PDO_BUFFER_DATA_MAX specified TxPdo. After reading the buffer, the data inside the buffer will be cleared.

Syntax:

int32_t　ECAT_SetTxPdoBufParam(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, uint16_t Enable);

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number Maximum: PDO_BUFFER_CHANNEL_MAX channels |
| SlaveNo | uint16_t | IN | Slave number |
| OffsetByte | uint16_t | IN | TxPdo offset (unit: byte) |
| DataSize | uint16_t | IN | Data size (Maxmum: 4bytes) |
| Enable | uint16_t | IN | 0:disable 1:enable |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte= 0;
uint16_t DataSize= 2;
uint16_t Enable= 0;
ret = ECAT_SetTxPdoBufParam(DeviceNo, ChannelNo, SlaveNo, OffsetByte, DataSize, Enable);
if(ret != 0)
    printf("Failed to set TxPdo Buffer parameters:%d\n", ret);
```

# 6.63. ECAT_GetTxPdoBufParam

Description:

Read parameters of a TxPdo buffer.


Syntax:

int32_t   ECAT_GetTxPdoBufParam(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t *SlaveNo, uint16_t *OffsetByte, uint16_t *DataSize, uint16_t *Enable);


Parameters:

| Name | Type | IN or OUT | Description |
| --- | --- | --- | --- |
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number<br>Maximum:<br>PDO_BUFFER_CHANNEL_MAX channels |
| SlaveNo | uint16_t* | OUT | Slave number |
| OffsetByte | uint16_t* | OUT | TxPdo offset (unit: byte) |
| DataSize | uint16_t* | OUT | Data size (Maxmum: 4bytes) |
| Enable | uint16_t* | OUT | 0:disable<br>1:enable |


Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte= 0;
uint16_t DataSize= 0;
uint16_t Enable= 0;
ret = ECAT_GetTxPdoBufParam(DeviceNo, ChannelNo, &SlaveNo, &OffsetByte, &DataSize, &Enable);
if(ret != 0)
    printf("Failed to get TxPdo Buffer parameters:%d\n", ret);
```

# 6.64. ECAT_SetTxPdoBufEnable

## Description:

Set enable/disable of a TxPdo buffer

## Syntax:

int32_t   ECAT_SetTxPdoBufEnable(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t Enable);

## Parameters:

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number<br>Maximum:<br>PDO_BUFFER_CHANNEL_MAX channels |
| Enable | uint16_t | IN | 0:disable<br>1:enable |

## Return:

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t Enable= 0;
ret = ECAT_SetTxPdoBufEnable(DeviceNo, ChannelNo, Enable);
if(ret != 0)
    printf("Failed to set TxPdo Buffer Eanble:%d\n", ret);
```

# 6.65. ECAT_GetTxPdoBufEnable

**Description:**

Get enable/disable of a TxPdo buffer

**Syntax:**

int32_t   ECAT_GetTxPdoBufEnable(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t *Enable);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number<br>Maximum:<br>PDO_BUFFER_CHANNEL_MAX channels |
| Enable | uint16_t* | OUT | 0:disable<br>1:enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t Enable= 0;
ret = ECAT_GetTxPdoBufEnable(DeviceNo, ChannelNo, &Enable);
if(ret != 0)
    printf("Failed to get TxPdo Buffer enable:%d\n", ret);
```

# 6.66. ECAT_GetTxPdoBufValue

**Description:**

Get Data of a TxPdo buffer

TxPdo buffer will store the values of the last PDO_BUFFER_DATA_MAX specified TxPdo. After reading the buffer, the data inside the buffer will be cleared.

**Syntax:**

int32_t  ECAT_GetTxPdoBufValue(uint16_t  DeviceNo,  uint16_t  ChannelNo,  float *Data, uint16_t Size, uint16_t *ActualGetSize);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number<br>Maximum:<br>PDO_BUFFER_CHANNEL_MAX channels |
| Data | float* | OUT | Data in the bufer ,<br>Max :PDO_BUFFER_DATA_MAX |
| Size | uint16_t | IN | Data Size |
| ActualGetSize | uint16_t* | OUT | Data sizeA ctual ge |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte= 0;
uint16_t DataSize= 2;
uint16_t Enable= 1;
float Data[PDO_BUFFER_DATA_MAX];
uint16_t Size = sizeof( Data );
uint16_t ActualGetSize;

ret = ECAT_SetTxPdoBufParam(DeviceNo, ChannelNo, SlaveNo, OffsetByte, DataSize, Enable);
if(ret != 0)
    printf("Failed to set TxPdo Buffer parameters:%d\n", ret);

ret = ECAT_GetTxPdoBufValue(DeviceNo, ChannelNo, Data, Size , &ActualGetSize);
if(ret != 0)
    printf("Failed to get TxPdo Buffer:%d\n", ret);
else
{
    for(uint16_t i = 0, i < ActualGetSize, i++)
    {
        printf("Data[%u]:%f\n", i, Data[ i ]);
    }
}
```

# 6.67. ECAT_SetAiFilterParam

**Description:**

Set a TxPdo filter, supporting notch filter 、high pass filter and low pass filter.

**Syntax:**

int32_t ECAT_SetAiFilterParam(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, uint16_t Enable);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number Maximum: AI_FILTER_CHANNEL_MAX channels |
| SlaveNo | uint16_t | IN | Slave number |
| OffsetByte | uint16_t | IN | TxPdo offset (unit: byte) |
| DataSize | uint16_t | IN | Data size (Maxmum: 4bytes) |
| Enable | uint16_t | IN | 0:disable 7:enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte= 0;
uint16_t DataSize= 2;
uint16_t Enable= 7;
ret = ECAT_SetAiFilterParam(DeviceNo, ChannelNo, SlaveNo, OffsetByte, DataSize, Enable);
if(ret != 0)
        printf("Failed to set TxPdo filter parameters:%d\n", ret);
```

# 6.68. ECAT_GetAiFilterParam

**Description:**

Get settings of a TxPdo filter

**Syntax:**

int32_t   ECAT_GetAiFilterParam(uint16_t  DeviceNo,  uint16_t  ChannelNo,  uint16_t *SlaveNo, uint16_t *OffsetByte, uint16_t *DataSize, uint16_t *Enable);

**Parameters:**

| Name | Type | IN or OUT | Description |
| --- | --- | --- | --- |
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number<br>Maximum:<br>AI_FILTER_CHANNEL_MAX channels |
| SlaveNo | uint16_t* | OUT | Slave number |
| OffsetByte | uint16_t* | OUT | TxPdo offset (unit: byte) |
| DataSize | uint16_t* | OUT | Data size (Maxmum: 4bytes) |
| Enable | uint16_t* | OUT | 0:disable<br>7:enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte= 0;
uint16_t DataSize= 0;
uint16_t Enable= 0;
ret = ECAT_GetAiFilterParam(DeviceNo, ChannelNo, &SlaveNo, &OffsetByte, &DataSize, &Enable);
if(ret != 0)
    printf("Failed to get TxPdo filter parameters:%d\n", ret);
```

# 6.69. ECAT_SetAiFilterEnable

**Description:**

Set enable/disable of a TxPdo filter

**Syntax:**

int32_t ECAT_SetAiFilterEnable(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t Enable);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number Maximum: AI_FILTER_CHANNEL_MAX channels |
| Enable | uint16_t | IN | 0:disable 7:enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t Enable= 7;
ret = ECAT_SetAiFilterEnable(DeviceNo, ChannelNo, Enable);
if(ret != 0)
    printf("Failed to set TxPdo filter Eanble:%d\n", ret);
```

# 6.70. ECAT_GetAiFilterEnable

**Description:**

Get enable/disable of a TxPdo filter

**Syntax:**

int32_t   ECAT_GetAiFilterEnable(uint16_t  DeviceNo,  uint16_t  ChannelNo,  uint16_t *Enable);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number<br>Maximum:<br>AI_FILTER_CHANNEL_MAX channels |
| Enable | uint16_t* | OUT | 0:disable<br>7:enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t Enable= 0;
ret = ECAT_GetAiFilterEnable(DeviceNo, ChannelNo, &Enable);
if(ret != 0)
        printf("Failed to get TxPdo filter enable:%d\n", ret);
```

# 6.71. ECAT_SetAiFilterFreq

**Description:**

Set frequency of a TxPdo filter

**Syntax:**

int32_t   ECAT_SetAiFilterFreq(uint16_t   DeviceNo,   uint16_t   ChannelNo,   uint16_t FilterType, double Frequency);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number<br>Maximum:<br>AI_FILTER_CHANNEL_MAX channels |
| FilterType | uint16_t | IN | 1: notch filter<br>2: low pass filter<br>4: high pass filter |
| Frequency | float | IN | frequency<br>notch filter:center frequency(hz)<br>low pass filter:cut off frequency(hz)<br>high pass filter:cut off frequency(hz)<br>Set 0 means disable the filter<br>ex: FilterType = notch filter, Frequency = 0, FilterType = low pass filter, Frequency = 100(hz),<br>means enable low pass filter, disable notch filter |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t FilterType= FILTER_LOW_PASS; // 2
double Frequency = 60;

ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
if(ret != 0)
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);


FilterType = FILTER_NOTCH ;//1
Frequency = 0;//disable
ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
if(ret != 0)
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);


FilterType =   FILTER_HIGH_PASS;//4
Frequency = 0;//disable
ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
if(ret != 0)
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);
```

# 6.72. ECAT_GetAiFilterFreq

**Description:**

Get frequency of a TxPdo filter

**Syntax:**

int32_t  ECAT_GetAiFilterFreq(uint16_t  DeviceNo,  uint16_t  ChannelNo,  uint16_t FilterType, double *Frequency);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number<br>Maximum:<br>AI_FILTER_CHANNEL_MAX channels |
| FilterType | uint16_t | IN | 1: notch filter<br>2: low pass filter<br>4: high pass filter |
| Frequency | float* | OUT | frequency<br>notch filter:center frequency<br>low pass filter:cut off frequency<br>high pass filter:cut off frequency |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t FilterType= FILTER_LOW_PASS;
double Frequency;
ret = ECAT_GetAiFilterFreq(DeviceNo, ChannelNo, FilterType, &Frequency);
if(ret != 0)
    printf("Failed to get TxPdo filter Frequency:%d\n", ret);
```

# 6.73. ECAT_GetAiFilterOutput

**Description:**

Get output of a TxPdo filter

**Syntax:**

int32_t  ECAT_GetAiFilterOutput(uint16_t  DeviceNo,  uint16_t  ChannelNo,  int32_t *Output);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number<br>Maximum:<br>AI_FILTER_CHANNEL_MAX channels |
| Output | int32_t* | OUT | output of a TxPdo filter |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte= 0;
uint16_t DataSize= 2;
uint16_t Enable= 7;
uint16_t FilterType;
double Frequency;
int32_t Output;

FilterType= FILTER_LOW_PASS;
Frequency = 60;
ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
if(ret != 0)
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);

FilterType= FILTER_NOTCH;
Frequency = 0;
ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
if(ret != 0)
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);

FilterType= FILTER_HIGH_PASS;
Frequency = 0;
ret = ECAT_SetAiFilterFreq(DeviceNo, ChannelNo, FilterType, Frequency);
if(ret != 0)
    printf("Failed to set TxPdo filter Frequency:%d\n", ret);

ret = ECAT_SetAiFilterParam(DeviceNo, ChannelNo, SlaveNo, OffsetByte, DataSize, Enable);
if(ret != 0)
    printf("Failed to set TxPdo filter parameters:%d\n", ret);

ret = ECAT_GetAiFilterOutput(DeviceNo, ChannelNo, &Output);
if(ret != 0)
```

```
        printf("Failed to set TxPdo filter parameters:%d\n", ret);
    else
        printf("Failed to set TxPdo filter parameters:%d\n", ret);
```

# 6.74. ECAT_SetPdoInToOutParam

**Description:**

Set a Pdo input Output. Write TxPDO data (Input) to RxPDO (Output) every EtherCAT cycle.

**Syntax:**

int32_t ECAT_SetPdoInToOutParam(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t SlaveNoIn, uint16_t OffsetByteIn, uint16_t DataSizeIn, uint16_t SlaveNoOut, uint16_t OffsetByteOut, uint16_t DataSizeOut)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number Maximum: PDO_INTOOUT_CHANNEL_MAX channels |
| SlaveNoIn | uint16_t | IN | Slave number |
| OffsetByteIn | uint16_t | IN | TxPdo offset (unit: byte) |
| DataSizeIn | uint16_t | IN | Data size (Maxmum: 4bytes) |
| SlaveNoOut | uint16_t | IN | Slave number |
| OffsetByteOut | uint16_t | IN | RxPdo offset (unit: byte) |
| DataSizeOut | uint16_t | IN | Data size (Maxmum: 4bytes) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNoIn = 0;
uint16_t OffsetByteIn= 0;
uint16_t DataSizeIn= 0;
uint16_t SlaveNoOut = 0;
uint16_t OffsetByteOut= 0;
uint16_t DataSizeOut= 0;
ret = ECAT_SetPdoInToOutParam(DeviceNo, ChannelNo, SlaveNoIn, OffsetByteIn, DataSizeIn,
SlaveNoOut, OffsetByteOut, DataSizeOut);
if(ret != 0)
    printf("Failed to set PdoInToOut parameters:%d\n", ret);
```

# 6.75. ECAT_GetPdoInToOutParam

**Description:**

Get settings of a Pdo input Output

**Syntax:**

int32_t ECAT_GetPdoInToOutParam(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t *SlaveNoIn, uint16_t *OffsetByteIn, uint16_t *DataSizeIn, uint16_t *SlaveNoOut, uint16_t *OffsetByteOut, uint16_t *DataSizeOut)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number Maximum: PDO_INTOOUT_CHANNEL_MAX channels |
| SlaveNoIn | uint16_t* | OUT | Slave number |
| OffsetByteIn | uint16_t* | OUT | TxPdo offset (unit: byte) |
| DataSizeIn | uint16_t* | OUT | Data size (Maxmum: 4bytes) |
| SlaveNoOut | uint16_t* | OUT | Slave number |
| OffsetByteOut | uint16_t* | OUT | RxPdo offset (unit: byte) |
| DataSizeOut | uint16_t* | OUT | Data size (Maxmum: 4bytes) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t SlaveNoIn = 0;
uint16_t OffsetByteIn= 0;
uint16_t DataSizeIn= 0;
uint16_t SlaveNoOut = 0;
uint16_t OffsetByteOut= 0;
uint16_t DataSizeOut= 0;
ret = ECAT_GetPdoInToOut(DeviceNo, ChannelNo, &SlaveNoIn, &OffsetByteIn, &DataSizeIn,
&SlaveNoOut, &OffsetByteOut, &DataSizeOut);
if(ret != 0)
    printf("Failed to get PdoInToOut parameters:%d\n", ret);
```

# 6.76. ECAT_SetPdoInToOutCoeff

**Description:**

Set coefficient of a Pdo input Output

Output = input * gain + offset

**Syntax:**

int32_t   ECAT_SetPdoInToOutCoeff(uint16_t  DeviceNo,  uint16_t  ChannelNo,  float gain, float offset)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number Maximum: PDO_INTOOUT_CHANNEL_MAX channels |
| gain | float | IN | gain |
| offset | float | IN | offset |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
float gain= 1;
float offset = 0;

ret = ECAT_SetPdoInToOutCoeff(DeviceNo, ChannelNo, gain, offset);
if(ret != 0)
    printf("Failed to set PdoInToOut Coeff:%d\n", ret);
```

# 6.77. ECAT_GetPdoInToOutCoeff

**Description:**

Get coefficient of a Pdo input Output

**Syntax:**

int32_t   ECAT_GetPdoInToOutCoeff(uint16_t  DeviceNo,  uint16_t  ChannelNo,  float *gain, float *offset)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number Maximum: PDO_INTOOUT_CHANNEL_MAX channels |
| gain | float* | OUT | gain |
| offset | float* | OUT | offset |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
float gain;
float offset;
ret = ECAT_GetPdoInToOutCoeff(DeviceNo, ChannelNo, &gain, &offset);
if(ret != 0)
    printf("Failed to get PdoInToOut coeff:%d\n", ret);
```

# 6.78. ECAT_SetPdoInToOutEnable

Description:

Set enable/disable of a Pdo input Output

Syntax:

int32_t ECAT_SetPdoInToOutEnable(uint16_t DeviceNo, uint16_t ChannelNo, uint16_t Enable)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number Maximum: PDO_INTOOUT_CHANNEL_MAX channels |
| Enable | uint16_t | IN | 0:disable 1:enable |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t Enable= 0;
ret = ECAT_SetPdoInToOutEnable(DeviceNo, ChannelNo, Enable);
if(ret != 0)
    printf("Failed to set PdoInToOutEanble:%d\n", ret);
```

# 6.79. ECAT_GetPdoInToOutEnable

**Description:**

Get enable/disable of a Pdo input Output

**Syntax:**

int32_t   ECAT_GetPdoInToOutEnable(uint16_t   DeviceNo,   uint16_t   ChannelNo, uint16_t *Enable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ChannelNo | uint16_t | IN | Channel number<br>Maximum:<br>PDO_INTOOUT_CHANNEL_MAX<br>channels |
| Enable | uint16_t* | OUT | 0:disable<br>1:enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ChannelNo= 0;
uint16_t Enable= 0;
ret = ECAT_GetPdoInToOutEnable(DeviceNo, ChannelNo, &Enable);
if(ret != 0)
    printf("Failed to get PdoInToOut enable:%d\n", ret);
```

# 7. Motion Control Functions

## 7.1. Motion Control Initialization

### 7.1.1. ECAT_McInit

Description:

Initialize parameters for motion control.

Syntax:

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

Parameters:

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisCount=0;
uint16_t McSlaveNo[MC_AXIS_NO_MAX];
uint16_t McSubAxisNo[MC_AXIS_NO_MAX];
  // Ex: The network system is composed of two CiA402 servo drives and an ECAT-2094S slave.
McSlaveNo[0] = 0;    // axis 0 is slave 0
McSlaveNo[1] = 1;    // axis 1 is slave 1
McSlaveNo[2] = 2;    // axis 2 is one axis of slave 2 (Note: slave 2 is a 4-axis slave)
McSlaveNo[3] = 2;    // axis 3 is one axis of slave 2
McSlaveNo[4] = 2;    // axis 4 is one axis of slave 2
McSlaveNo[5] = 2;    // axis 5 is one axis of slave 2
McSubAxisNo [0] = 0;    // axis 0 is a single axis slave
McSubAxisNo [1] = 0;    // axis 1 is a single axis slave
McSubAxisNo [2] = 0;    // axis 2 is the local axis0 of a 4-axis slave
McSubAxisNo [3] = 1;    // axis 3 is the local axis1 of a 4-axis slave
McSubAxisNo [4] = 2;    // axis 4 is the local axis2 of a 4-axis slave
McSubAxisNo [5] = 3;    // axis 5 is the local axis3 of a 4-axis slave

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

## 7.1.2. ECAT_McInit_Ex

**Description:**

Use Motion Control Parameter File to initialize motion control, please refer to "Motion Control Parameter File Editing Steps" for file creation.

**Syntax:**

int32_t ECAT_McInit_Ex(uint16_t DeviceNo, char* FileName, uint16_t SlaveNo[], uint16_t SubAxisNo[], uint16_t* AxisCnt)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
char* FileName=" File.motcfg"; // Local path
//char* FileName="C:\\ File.motcfg"; // Absolute path
uint16_t SlaveNo[MC_AXIS_NO_MAX];
uint16_t SubAxisNo[MC_AXIS_NO_MAX];
uint16_t AxisCnt;

ret = ECAT_McInit_Ex(DeviceNo, FileName, SlaveNo, SubAxisNo, &AxisCnt);
if(ret < 0)
{
    printf("Failed to initialize motion control:%d\n",ret);
}
else
{
    uint16_t i;
    printf("Initialize motion control successfully\n");
    printf("Axis Count: %u\n",   AxisCnt);
    for(i = 0; i < AxisCnt; i++)
    {
        printf("Axis[%u].SlaveNo: %u\n", i, SlaveNo[i]);
    }
}
```

# 7.2. Axis Parameter Settings

## 7.2.1. ECAT_McSetAxisDefaultMode

**Description:**

Set the default motion mode of the specified axis.

The initial default motion mode is CSP mode. When ServoOn or ErrorReset, the motion mode will be switched to the default motion mode. Table 7.1 shows the PDO Entries required by each motion mode.

Table 7.1

|  | RxPdo Entries | | TxPdo Entries | |
|---|---|---|---|---|
| MC_MODE_CSP | 607A | Target Position | 6064 | Position actual value |
| MC_MODE_CSV | 60FF | Target Velocity | 6064 | Position actual value |
| MC_MODE_CST | 6071 | Target Torque | 6077 | Torque actual value |

**Syntax:**

int32_t ECAT_McSetAxisDefaultMode(uint16_t DeviceNo, uint16_t AxisNo, int8_t Mode)

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Mode | int8_t | IN | Motion mode<br>MC_MODE_CSP(CSP mode)<br>MC_MODE_CSV(CSV mode)<br>MC_MODE_CST(CST mode)<br>Initial value: MC_MODE_CSP |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
int8_t Mode = MC_MODE_CSP;
ret = ECAT_McSetAxisDefaultMode(DeviceNo, AxisNo, Mode);
if(ret < 0)
{
    printf("Failed to set axis default mode:%d\n",ret);
}
else
{
    printf("Set axis default mode successfully!\n");
}
```

## 7.2.2. ECAT_McGetAxisDefaultMode

**Description:**

Get default motion mode of an axis.

**Syntax:**

int32_t   ECAT_McGetAxisDefaultMode(uint16_t   DeviceNo,   uint16_t   AxisNo,   int8_t *Mode)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Mode | int8_t * | OUT | Motion mode |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
int8_t Mode;
ret = ECAT_McGetAxisDefaultMode(DeviceNo, AxisNo, &Mode);
if(ret < 0)
{
    printf("Failed to get axis default mode:%d\n", ret);
}
else
{
    printf("Axis[%u] default mode:%f\n", AxisNo, Mode);
}
```

### 7.2.3. ECAT_McSetAxisServoOn

**Description:**

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

After setting Servo On, it takes several ms to complete Servo On. You can use ECAT_McGetAxisState to get the axis status. When the axis status is MC_AS_STANDSTILL, it means Servo On is completed.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.4. ECAT_McSetAxisPPU

**Description:**

Set Pulses Per Unit (PPU) value for an axis. Motion command is based on Unit. Inside the control card, pulses are used for control motors.

Pulses Per Unit, pulses of each unit. For example, If you want to set the "user unit" as mm and a mm requires 1000 pulses, then set the PPU to be 1000. If the unit of PDO Entry (Table 7.2) of the driver is not pulse、pulse/s and pulse/s^2, the PPU needs to be set to 1, at this time, "user unit" is the original unit of the PDO, such as rpm, rpm/s... Etc.

Table 7.2

| RxPDO | | TxPDO | | unit |
|---|---|---|---|---|
| 607A | Target Position | 6064 | Position actual value | pulse |
| 60FF | Target Velocity | 606C | Velocity actual value | pulse/s |
| 60B0 | Position offset | | | pulse |
| 60B1 | Velocity offset | | | pulse/s |
| 6099 | Speed during search for switch | | | pulse/s |
| 6099 | Speed during search for zero | | | pulse/s |
| 609A | Homing acceleration | | | pulse/s^2 |

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.5.  ECAT_McGetAxisPPU

**Description:**

Get pulses per unit setting of an axis.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.6. ECAT_McSetAxisVelAccScale

### Description:

Set the velocity/acceleration scaling parameters of an axis.

When the unit of PDO Entry (Table 7.2) of the drive is not pulse/s or pulse/s^2, this parameter can be used for unit scaling.

Example:

The original velocity unit of PDO is rpm. To convert pulse/s to rpm, the formula is (1/1 revolution) * 60. Assuming that the number of pulses required for one revolution is 10000, the velocity scaling parameter is set as 1/10000*60 = 0.006

The original acceleration unit of PDO is rpm. To convert pulse/s to rpm, the formula is (1/1 revolution) * 60. Assuming that the number of pulses required for one revolution is 10000, the acceleration scaling parameter is set as 1/10000*60 = 0.006

### Syntax:

int32_t ECAT_McSetAxisVelAccScale(uint16_t DeviceNo, uint16_t AxisNo, double VelScal, double AccScal)

### Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| VelScal | double | IN | Velocity scale parameter default:1 |
| AccScal | double | IN | Acceleration scale parameter default:1 |

### Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double VelScal = 1;
double AccScal = 1;
ret = ECAT_McSetAxisVelAccScale(DeviceNo, AxisNo, VelScal, AccScal);
if(ret < 0)
{
    printf("Failed to set axis vel acc scale:%d\n",ret);
}
else
{
    printf("Set axis vel acc scale successfully!\n");
}
```

## 7.2.7. ECAT_McGetAxisVelAccScale

**Description:**

Get velocity/acceleration scaling parameters of an axis.

**Syntax:**

int32_t  ECAT_McGetAxisVelAccScale(uint16_t  DeviceNo,  uint16_t  AxisNo,  double *VelScal, double *AccScal)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| VelScal | double* | OUT | Velocity scale parameter |
| AccScal | double* | OUT | Acceleration scale parameter |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double VelScal;
double AccScal;
ret = ECAT_McGetAxisVelAccScale(DeviceNo, AxisNo, &VelScal, &AccScal);
if(ret < 0)
{
    printf("Failed to get axis vel acc scale:%d\n", ret);
}
else
{
    printf("Axis[%u] VelScal:%f AccScal:%f, \n", AxisNo, VelScal, AccScal);
}
```

## 7.2.8.   ECAT_McSetProfileData

**Description:**

Set   a   position   array   data   into   a   buffer   number   for   profile motion(*ECAT_McAxisMoveProfile*).

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

## 7.2.9.   ECAT_McGetProfileData

**Description:**

Get a position array data from a profile buffer number

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

## 7.2.10. ECAT_McSetProfileInterval

**Description:**

Set interval of position array data for profile motion. It controls the data consuming speed as well as the motion speed. For example, if the interval value is 2, the system will consume each position value for every 2 cycles, i.e., the increment for each cycle is half of the original defined value.

**Syntax:**

int32_t ECAT_McSetProfileInterval(uint16_t DeviceNo, uint16_t ProfileNo, uint16_t Value)

**Parameters:**

| Name | Type | IN or OUT | Description |
| --- | --- | --- | --- |
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| ProfileNo | uint16_t | IN | Profile number, available number range 0~15 |
| Value | uint16_t | IN | Interval<br>For example:<br>1: read position array data for motion every cycletime (default setting)<br>2: read position array data for motion every two cycletimes. |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t ProfileNo = 0;
uint16_t Value = 2;
ret = ECAT_McSetProfileInterval(DeviceNo, ProfileNo, Value);
if(ret < 0)
{
    printf("Failed to set profile Interval:%d\n", ret);
}
else
{
    printf("Set profile Interval successfully!\n");
}
```

Interval = 1(default)
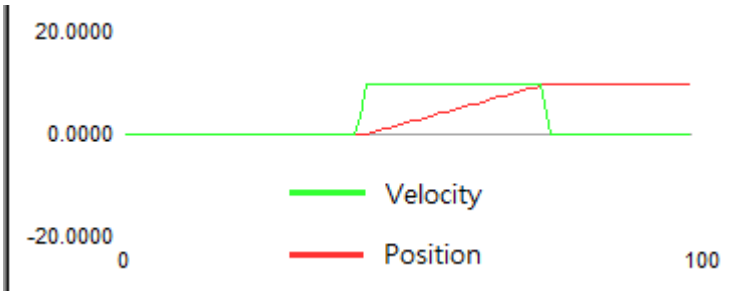


Interval = 2
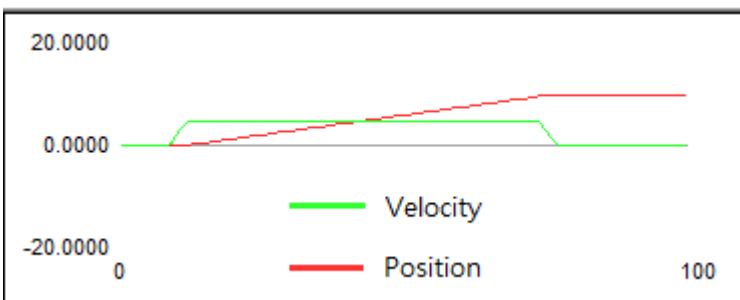
## 7.2.11. ECAT_McSetProfileCSV

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Figure 7.1

## Example:

### [C/C++]

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

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

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

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


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


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


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


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

## 7.2.12. ECAT_McGetProfileCSV

**Description:**

Read out position data from a CSV file. This file is used for a profile motion. The format is shown in Figure 7.1.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

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

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

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

```
    }
};
```

## 7.2.13.  ECAT_McSetAxisAccTime

**Description:**

Set acceleration time of an axis.



**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Time_ms | uint16_t | IN | Acceleration time (Unit: millisecond) default:100 |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

## 7.2.14. ECAT_McGetAxisAccTime

**Description:**

Get acceleration time of an axis.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

## 7.2.15. ECAT_McSetAxisAccDecTime_Stepper

**Description:**

Set acceleration time and deceleration time of an axis.

Note: This function is designed for ECAT-2091S/ ECAT-2094S.
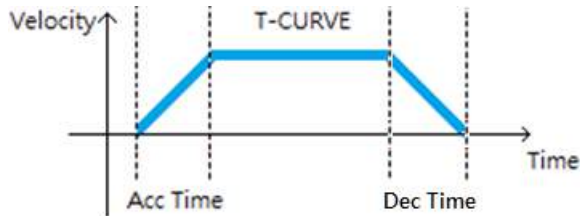


**Syntax:**

int32_t ECAT_McSetAxisAccDecTime_Stepper(uint16_t DeviceNo, uint16_t AxisNo, uint16_t AccTime_ms, uint16_t DecTime_ms)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number |
| AxisNo | uint16_t | IN | Axis number |
| AccTime_ms | uint16_t | IN | Acceleration time (Unit: millisecond) default:100 |
| DecTime_ms | uint16_t | IN | deceleration time (Unit: millisecond) default:100 |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t AccTime_ms = 500;
uint16_t DecTime_ms = 500;
ret = ECAT_McSetAxisAccDecTime_Stepper(DeviceNo, AxisNo, AccTime_ms , DecTime_ms);
if(ret < 0)
{
    printf("Failed to set axis acc dec time:%d\n", ret);
}
else
{
    printf("Set axis acc dec time successfully!\n");
}
```

## 7.2.16. ECAT_McGetAxisAccDecTime_Stepper

**Description:**

Get acceleration time and deceleration time of an axis.

Note: This function is designed for ECAT-2091S/ ECAT-2094S.

**Syntax:**

int32_t ECAT_McGetAxisAccDecTime_Stepper(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *AccTime_ms, uint16_t *DecTime_ms);

大

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number |
| AxisNo | uint16_t | IN | Axis number |
| AccTime_ms | uint16_t* | OUT | Acceleration time (Unit: millisecond) default:100 |
| DecTime_ms | uint16_t* | OUT | deceleration time (Unit: millisecond) default:100 |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t AccTime_ms;
uint16_t DecTime_ms;
ret = McGetAxisAccDecTime_Stepper(DeviceNo, AxisNo, &AccTime_ms, &DecTime_ms);
if(ret < 0)
{
    printf("Failed to get axis acc dec time:%d\n", ret);
}
else
{
    printf("Axis[%u] acc dec Time(ms):%f, %f\n", AxisNo, AccTime_ms, DecTime_ms);
}
```

## 7.2.17.　ECAT_McSetAxisAccUnit_Stepper

**Description:**

　　Set the acc unit of an axis.This setting is only to correctly convert the acc unit to milliseconds.

　　　　Note:　(1)Support ECAT-2094S、ECAT-2091S

　　　　　　(2) No matter how much this value is set, when using ECAT_McSetAxisAccDecTime_Stepper or ECAT_McSetAxisAccTime to set acceleration and deceleration, the unit is milliseconds

　　　　　　(3) Don't set this value unless necessary

　　　　　　(4) This setting will modify the value of 8x20:08

| 8020:0 | POS Settings X | | > 9 < |
|---|---|---|---|
| 8020:01 | Velocity min | RW | 0x00000000 (0) |
| 8020:02 | Velocity max | RW | 0x000F4240 (1000000) |
| 8020:03 | Acceleration pos | RW | 0x03E8 (1000) |
| 8020:04 | Acceleration neg | RW | 0x03E8 (1000) |
| 8020:05 | Deceleration pos | RW | 0x03E8 (1000) |
| 8020:06 | Deceleration neg | RW | 0x03E8 (1000) |
| 8020:07 | Emergency deceleration | RW | 0x0000 (0) |
| 8020:08 | Acceleration unit | RW | Acceleration time from Vmin to Vmax [ms] (0) |
| 8020:09 | Acc-Dec parameter definition | RW | Acceleration-->Start phase & Deceleration-->Stop Phase (0) |

**Syntax:**

　　int32_t ECAT_McSetAxisAccUnit_Stepper(uint16_t DeviceNo, uint16_t AxisNo, uint16_t Unit, uint32_t *AbortCode)

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Unit | uint16_t | IN | Acc Unit |
| AbortCode | uint32_t * | OUT | Abort code of the SDO (Please refer to Appendix "SDO Abort messages") |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Unit = 0;
uint32_t AbortCode;
ret = ECAT_McSetAxisAccUnit_Stepper(DeviceNo, AxisNo, Unit, &AbortCode);
if(ret < 0)
{
    printf("Failed to set axis acc unit:%d\n", ret);
}
else
{
    printf("Set axis acc unit successfully!\n");
}
```

## 7.2.18. ECAT_McGetAxisAccUnit_Stepper

**Description:**

Get acc unit of an axis.

Note: Support ECAT-2094S、ECAT-2091S



**Syntax:**

int32_t ECAT_McGetAxisAccUnit_Stepper(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *Unit, uint32_t *AbortCode)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Unit | uint16_t* | OUT | Acc Unit |
| AbortCode | uint32_t* | OUT | Abort code of the SDO (Please refer to Appendix "SDO Abort messages") |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Unit;
uint32_t AbortCode;
ret = ECAT_McGetAxisAccUnit_Stepper(DeviceNo, AxisNo, &Unit, &AbortCode);
if(ret < 0)
{
    printf("Failed to get axis acc unit:%d\n", ret);
}
else
{
    printf("Axis[%u] acc unit:%u\n", AxisNo, Unit);
}
```

## 7.2.19. ECAT_McSetAxisAccDecType
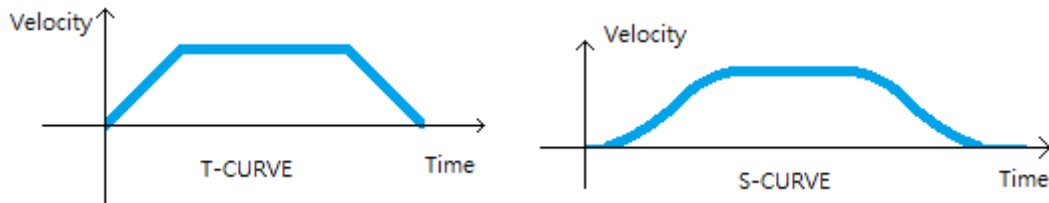
**Description:**

Set acceleration type of an axis.



**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.20. ECAT_McGetAxisAccDecType

**Description:**

Get acceleration type of an axis.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.21. ECAT_McSetAxisEncoderPPR

**Description:**

Set encoder pulses per revolution value of an axis, For encoder and motor scaling.

Note: For Encoder module / Stepper motor controller

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| PPR | Uint32_t | IN | Pulses per revolution<br>default:1 |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.22. ECAT_McGetAxisEncoderPPR

**Description:**

Get encoder pulses per revolution of an axis.

Note: For Encoder module / Stepper motor controller

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;

uint16_t DeviceNo = 0;

uint16_t AxisNo = 0;

uint32_t PPR ;

ret = ECAT_McGetAxisEncoderPPR(DeviceNo, AxisNo, &PPR);

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

## 7.2.23.  ECAT_McSetAxisMotorPPR

**Description:**

Set motor pulses per revolution of an axis, For encoder and motor scaling.

Note: For Encoder module / Stepper motor controller

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| PPR | Uint32_t | IN | Pulses per revolution default:1 |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

## 7.2.24.  ECAT_McGetAxisMotorPPR

**Description:**

Get motor pulses per revolution of an axis.

Note: For Encoder module / Stepper motor controller

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.25. ECAT_McSetEcamTable

Description:

Set the slave position data for an E-CAM table.

Users can use Cam Utility to create E -CAM table

To download the software and manual, please refer to the chapter "Software Installation"

Syntax:

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

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| TableNo | uint16_t | IN | E-CAM table number |
| Data | double* | IN | Slave position data (Unit: user unit) |
| DataSize | uint16_t | IN | Size of data (Up to 1000) |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t TableNo = 0;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double Data[3];
uint16_t DataSize = 3;
uint8_t SlaveAbs = 0; //Relative
double MasterInterval = 0.5;
double SlaveScaling = 1.0;
double AxisPos = 3.0;
double AxisVel = 1;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;

//Write E-CAM Table data to Data[3]
Data[0] = 0;
Data[1] = 1;
Data[2] = 0;

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
```

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

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

E-CAM Table:



E-CAM synchronization motion diagram:

MasterInterval = 0.25
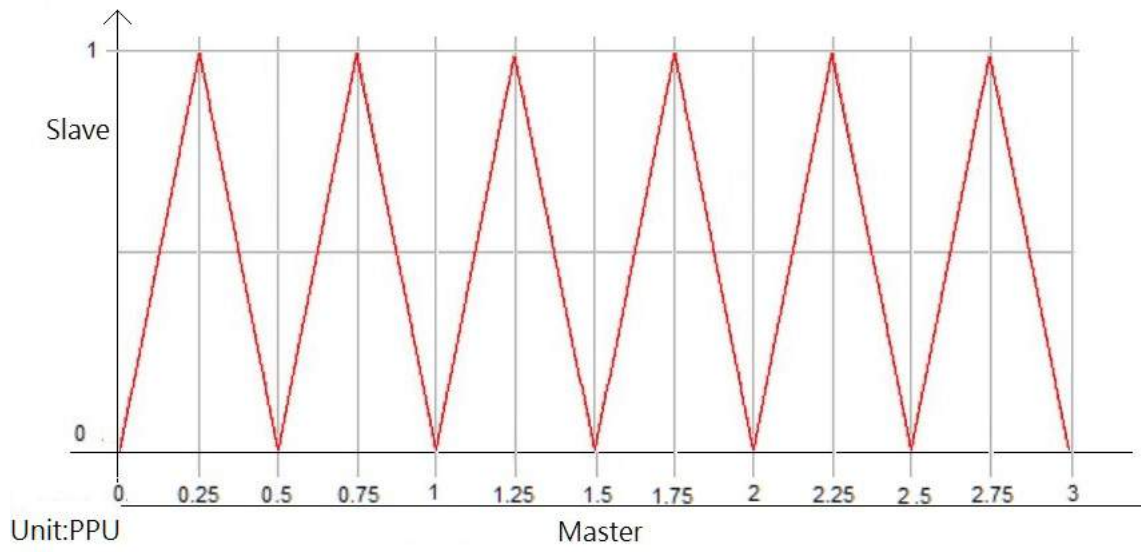
## 7.2.26. ECAT_McGetEcamTable

**Description:**

Get the slave position data from an E-CAM table.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| TableNo | uint16_t | IN | E-CAM table number |
| Data | double* | OUT | Slave position data (Unit: user unit) |
| DataSize | uint16_t | IN | Size of data (Up to 1000) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t TableNo = 0;
double Data[1000];
uint16_t DataSize = 1000;
ret = ECAT_McGetEcamTable(DeviceNo, TableNo, Data, DataSize);


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

## 7.2.27. ECAT_McConfigEcamTable

**Description:**

Set relative/absolute position property of an E-CAM table.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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


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

## 7.2.28.  ECAT_McSetAxisTouchProbeProperty

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.29. ECAT_McGetAxisTouchProbeProperty

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.30.  ECAT_McGetAxisTouchProbeValue

**Description:**

Get the Touch Probe value of an axis.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| ProbeNo | uint16_t | IN | Touch Probe number<br>1: Touch Probe 1 input<br>2: Touch Probe 2 input |
| Value | double * | OUT | Touch Probe Value (Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.31. ECAT_McSetAxisVelocityFeedForwardGain

### Description:

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



Velocity Feedforward OFF       Velocity Feedforword ON 100%

------Command Position Value
------Actual Position Value
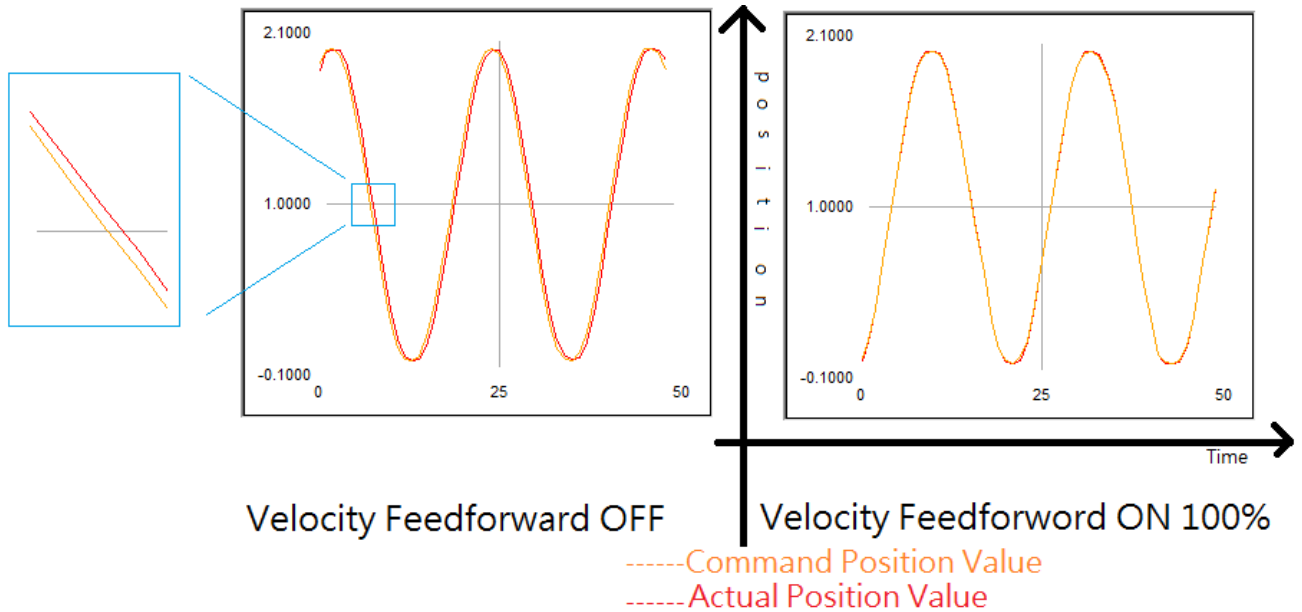
### Syntax:

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

### Parameters:

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

### Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.32. ECAT_McGetAxisVelocityFeedForwardGain

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.33. ECAT_McSetAxisPosSoftwareLimit

**Description:**

Set position software limits to a specific axis.

Notice: Only for CiA402 and Virtual axis.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.34. ECAT_McGetAxisPosSoftwareLimit

**Description:**

Get position software limits to an axis.

Notice: Only for CiA402 and Virtual axis.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.35. ECAT_McSetAxisPosSoftwareLimitStatus

**Description:**

Set position software limit status to be enabled or not for an axis.

Notice: Only for CiA402 and Virtual axis.

| Case 1: | axis | | | | | |
|---|---|---|---|---|---|---|
| Status: 0 (disabled) | | Axis NO. | CmdPosition | Position | Velocity | Axis State | Axis Error |
| | | 0 | 10.000 | 10.000 | 0.0 | StandStill | 0 |
| | Group |  | | | | | |

| Case 2: | axis | | | | | |
|---|---|---|---|---|---|---|
| Status:1 ErrorStop: 0 | | Axis NO. | CmdPosition | Position | Velocity | Axis State | Axis Error |
| | | 0 | 10.000 | 8.000 | 0.0 | StandStill | -1134 |
| Limits of X-Axis: Maxmum:8 Minimum:-8 | Group |  | | | | | |

| Case 3: | axis | | | | | |
|---|---|---|---|---|---|---|
| Status:1 ErrorStop: 1 | | Axis NO. | CmdPosition | Position | Velocity | Axis State | Axis Error |
| | | 0 | 8.000 | 7.999 | 0.0 | ErrorStop | -1134 |
| Limit of X-Axis: Maxmum:8 Minimum:-8 | Group |  | | | | | |

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Status | uint16_t | IN | Software Limit Status:<br>0: disabled(default)<br>1: enabled |
| ErrorStop | uint16_t | IN | Error handling method when software limit is triggered.<br>**0:**<br>providing a message "Axis Last error: -1134" when software limit is triggered, but system does not stop.<br>**1:**<br>ErrorStop and clear group buffer when software limit triggered. |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
uint16_t DeviceNo = 0;

uint16_t AxisNo = 0;

uint16_t Status = 0;

uint16_t ErrorStop = 0;

ret = ECAT_McSetAxisPosSoftwareLimitStatus(DeviceNo,AxisNo, Status, ErrorStop);

if(ret < 0)

{

    printf("Failed to set axis position software limit status :%d\n", ret);

}

else

{

    printf("Set axis position software limit status successfully!\n");

}
```

## 7.2.36. ECAT_McGetAxisPosSoftwareLimitStatus

**Description:**

Get position software limit status to a specific axis.

Notice: Only for CiA402 and Virtual axis.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Status | uint16_t | IN | Software Limit Status: <br> 0: disabled <br> 1: enabled |
| ErrorStop | uint16_t | IN | Error handling method when software limit is triggered. <br> **0:** <br> providing a message "Axis Last error: -1134" when software limit is triggered, but system does not stop. <br> **1:** <br> ErrorStop and clear group buffer when software limit triggered. |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
uint16_t DeviceNo = 0;

uint16_t AxisNo = 0;

uint16_t Status = 0;

uint16_t ErrorStop = 0;

ret = ECAT_McGetAxisPosSoftwareLimitStatus(DeviceNo, AxisNo, &Status, &ErrorStop);

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

## 7.2.37. ECAT_OpenMotionConfig

**Description:**

Read a file which is created by Axis configuration in the utility program and save to variables provided in the arguments. These settings can be further transferred into a control card by calling several different functions. Since this function uses the COM technique of Microsoft to process data, it is not supported in a Linux OS system.

**Syntax:**

int32_t ECAT_ OpenMotionConfig(char* bstrFileName, uint16_t *AxisCnt

, uint16_t SlaveNo[], uint16_t SubAxisNo[], double PPU[], int32_t HomeMethod[]

, double HomeSpeedSeachSw[], double HomeSpeedSeachZr[], double HomeAcc[]

, uint32_t EncoderPPR[], uint32_t MotorPPR[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| bstrFileName | char* | IN | File name of this axis configuration file |
| AxisCnt | uint16_t | OUT | Number of axes |
| SlaveNo | uint16_t * | OUT | An array of Slave number. Each index of this array is a slave number. |
| SubAxisNo | uint16_t * | OUT | Sub-axis number. In general, a slave only has one axis. But for some slaves, each one has several axes. Therefore, several sub-axis numbers are provided for axes inside this kind of slaves. With the combination of slave number and sub-axis number, the system can have all axes be defined and used individually. |
| PPU | Double* | OUT | Pulses Per Unit |

| HomeMethod | int32_t * | OUT | Homing method (Refer to the drive user manual) |
|---|---|---|---|
| HomeSpeedSeachSw | Double* | OUT | Speed during search for Home switch (Unit: user unit/s) |
| HomeSpeedSeachZr | Double* | OUT | Speed during search for z phase signal (Unit: user unit/s) |
| HomeAcc | Double* | OUT | Homing Acceleration (Unit: user unit/s^2) |
| EncoderPPR | uint32_t * | OUT | Pulses per revolution |
| MotorPPR | uint32_t * | OUT | Pulses per revolution |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

## 7.2.38. ECAT_McSetAxisMaxVelocity

**Description:**

Set maximum velocity of an axis for the following functions.

*ECAT_McAxisMoveAbs_P2P*

*ECAT_McAxisMoveRel_P2P*

*ECAT_McGroupMoveLineAbs_P2P*

*ECAT_McGroupMoveLineRel_P2P*

**Syntax:**

int32_t ECAT_McSetAxisMaxVelocity(uint16_t DeviceNo, uint16_t AxisNo, double MaxVelocity)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| MaxVelocity | double | IN | Maximum velocity(Unit: user unit/s) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double MaxVelocity = 100;
ret = ECAT_McSetAxisMaxVelocity(DeviceNo, AxisNo, MaxVelocity);
if(ret < 0)
{
    printf("Failed to set axis MaxVelocity:%d\n",ret);
}
else
{
    printf("Set axis MaxVelocity successfully!\n");
}
```

## 7.2.39. ECAT_McGetAxisMaxVelocity

**Description:**

Get maximum velocity of an axis.

**Syntax:**

int32_t ECAT_McGetAxisMaxVelocity(uint16_t DeviceNo, uint16_t AxisNo, double *MaxVelocity)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| MaxVelocity | Double* | OUT | Maximum velocity(Unit: user unit/s) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.40.    ECAT_McSetAxisDIActiveLevel

**Description:**

Set limit active level and home active level of an axis.

For CiA402 driver, it is usually "active high", means that the di bit is set to high when the switch is logically active.

When the settings of drive is "active low ", user need to use this function to inform the EtherCAT Master that the active level is "active low" .

Note: Regardless of the setting of "active high " or "active high", when using ECAT_McGetAxisDI to read the signal, 1 means that the signal is active.

**Syntax:**

int32_t  ECAT_McSetAxisDIActiveLevel(uint16_t  DeviceNo,  uint16_t  AxisNo,  uint8_t POT, uint8_t NOT, uint8_t ORG)


**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| POT | uint8_t | IN | Positive limit switch<br>0: Low Dectect<br>1: High Dectect (default) |
| NOT | uint8_t | IN | Negative limit switch<br>0: Low Dectect<br>1: High Dectect (default) |
| ORG | uint8_t | IN | Home switch<br>0: Low Dectect<br>1: High Dectect (default) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint8_t POT = 0;
uint8_t NOT = 0;
uint8_t ORG = 0;
ret = ECAT_McSetAxisDIActiveLevel(DeviceNo, AxisNo, POT, NOT, ORG);
if(ret < 0)
{
    printf("Failed to set axis di active level:%d\n",ret);
}
else
{
    printf("Set axis di active level successfully!\n");
}
```

# 7.2.41. ECAT_McGetAxisDIActiveLevel

Description:

Get limit active level and home active level of an axis.

Syntax:

int32_t  ECAT_McGetAxisDIActiveLevel(uint16_t  DeviceNo,  uint16_t  AxisNo,  uint8_t *POT, uint8_t *NOT, uint8_t *ORG)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| POT | uint8_t* | OUT | Positive limit switch<br>0: Low Dectect<br>1: High Dectect (default) |
| NOT | uint8_t* | OUT | Negative limit switch<br>0: Low Dectect<br>1: High Dectect (default) |
| ORG | uint8_t* | OUT | Home switch<br>0: Low Dectect<br>1: High Dectect (default) |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint8_t POT;
uint8_t NOT;
uint8_t ORG;
ret = ECAT_McGetAxisDIActiveLevel(DeviceNo, AxisNo, &POT, &NOT, &ORG);
if(ret < 0)
{
    printf("Failed to get axis di active level:%d\n",ret);
}
else
{
    printf("POT: %u, NOT: %u, ORG: %u\n ", POT, NOT, ORG);
}
```

## 7.2.42.　ECAT_McSetAxisActualPosition

**Description:**

Set actual position of an axis.

Note: This function is designed for ECAT-2091S/ ECAT-2094S.

**Syntax:**

int32_t  ECAT_McSetAxisActualPosition(uint16_t  DeviceNo,  uint16_t  AxisNo,  double Position)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double AxisPos = 1000;
ret = ECAT_McSetAxisActualPos(DeviceNo, AxisNo, AxisPos);
if(ret < 0)
{
    printf("Failed to set axis actual position:%d\n", ret);
}
```

## 7.2.43.    ECAT_McSetAxisCommandPosition

**Description:**

Set command position of an axis.

Note: This function is designed for ECAT-2091S/ ECAT-2094S.

**Syntax:**

int32_t    ECAT_McSetAxisCommandPosition(uint16_t    DeviceNo,    uint16_t    AxisNo, double Position)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
double AxisPos = 1000;
ret = ECAT_McSetAxisCommandPosition(DeviceNo, AxisNo, AxisPos);
if(ret < 0)
{
    printf("Failed to set axis command position:%d\n", ret);
}
```

## 7.2.44.  ECAT_McSetAxisInpSignal

**Description:**

Set the INP signal source of the specified axis.

When this function is enabled, MC_AS_STANDSTILL and MC_GS_STANDBY represent the completion of positioning.

When group commands are used and there are commands in the command buffer (BUFFERED mode), the next command will not start until the positioning is completed.

Note: (1) It can be used together with ECAT_McSetAxisInpCompare.

(2) When the motion command speed is too high and the time is too short, it may cause the motion command has been finished but the motor has not started to run. In this case, it may cause the misjudgment of the positioning signal. It is recommended to use it with ECAT_McSetAxisInpCompare.

Example: Table 7.3 is the default PDO Mapping. Take a Panasonic drive as an example, the INP signal is the 24th bit of the Digital inputs, so the Offset is set to 13, and the Bit is set to 24.

Table 7.3

|  | TxPdo Entries | Size(byte) | Offset(byte) |
|---|---|---|---|
| Mode0 | Statusword | 2 | 0 |
|  | Error code | 2 | 2 |
|  | Modes of operation display | 1 | 4 |
|  | Position actual value | 4 | 5 |
|  | Velocity actual value | 4 | 9 |
|  | Digital inputs | 4 | 13 |

**Syntax:**

int32_t ECAT_McSetAxisInpSignal(uint16_t DeviceNo, uint16_t AxisNo, uint16_t Offset, uint16_t Bit, int32_t Enable)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Offset | uint16_t | IN | Byte offset |
| Bit | uint16_t | IN | Bit number |
| Enable | uint32_t | IN | 0: Disable(Default)<br>1: Enable |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Offset= 13;
uint16_t Bit= 24;
uint32_t Enable = 1;
ret = ECAT_McSetAxisInpSignal(DeviceNo, AxisNo, Offset, Bit, Enable);
if(ret < 0)
{
    printf("Failed to set axis inp signal:%d\n", ret);
}
else
{
    printf("Set axis inp signal successfully!\n");
}
```

## 7.2.45. ECAT_McGetAxisInpSignal

Description:

Get INP signal source of an axis

Syntax:

int32_t ECAT_McGetAxisInpSignal(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *Offset, uint16_t *Bit, int32_t *Enable)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Offset | uint16_t* | OUT | Byte offset |
| Bit | uint16_t* | OUT | Bit number |
| Enable | uint32_t* | OUT | 0: Disable<br>1: Enable |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint16_t Offset;
uint16_t Bit;
uint32_t Enable;
ret = ECAT_McGetAxisInpSignal(DeviceNo, AxisNo, &Offset, &Bit, &Enable);
if(ret < 0)
{
    printf("Failed to get axis inp signal:%d\n", ret);
}
else
{
    printf("Axis[%u] Offset:%u, Bit: %u, Enable: %u\n", AxisNo, Offset, Bit, Enable);
}
```

## 7.2.46.　　ECAT_McSetAxisInpCompare

**Description:**

　　Set the INP signal conditions of the specified Axis.

　　When this function is enabled, MC_AS_STANDSTILL and MC_GS_STANDBY represent the completion of positioning.

　　When group commands are used and there are commands in the command buffer (BUFFERED mode), the next command will not start until the positioning is completed.

　　When a command is executed, the difference between the command position and the actual position is within the position window, and the stabilization time has elapsed, the status will be changed to MC_AS_STANDSTILL/ MC_GS_STANDBY.

　　Note: It can be used together with ECAT_McSetAxisInpSignal.

**Syntax:**

　　int32_t  ECAT_McSetAxisInpCompare(uint16_t  DeviceNo,  uint16_t  AxisNo,  uint32_t Time_ms, double PosWindow, int32_t Enable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Time_ms | uint32_t | IN | Stabilization time<br>Unit: ms |
| PosWindow | double | IN | Position Window<br>unit:user unit |
| Enable | uint32_t | IN | 0: Disable(Default)<br>1: Enable |

**Return:**

　　0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t Time_ms = 3;
double PosWindow = 0.1;
uint32_t Enable = 1;
ret = ECAT_McSetAxisInpCompare(DeviceNo, AxisNo, Time_ms, PosWindow, Enable);
if(ret < 0)
{
    printf("Failed to set axis inp compare:%d\n", ret);
}
else
{
    printf("Set axis inp compare successfully!\n");
}
```

## 7.2.47.　ECAT_McGetAxisInpCompare

Description:

Get the INP conditions of the specified Axis

Syntax:

int32_t  ECAT_McGetAxisInpCompare(uint16_t  DeviceNo,  uint16_t  AxisNo,  uint32_t *Time_ms, double *PosWindow, int32_t *Enable)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Time_ms | uint32_t* | OUT | Stabilization time<br>Unit: ms |
| PosWindow | double* | OUT | Position Window<br>unit:user unit |
| Enable | uint32_t* | OUT | 0: Disable<br>1: Enable |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.48. ECAT_McSetAxisInpTimeOut

**Description:**

Set the INP timeout of the specified axis.

When the INP function is enabled, if the motion command has ended, but no INP signal is detected after a period of time, the axis state will change to MC_AS_ERRORSTOP.

**Syntax:**

int32_t ECAT_McSetAxisInpTimeOut(uint16_t DeviceNo, uint16_t AxisNo, uint32_t Time_ms)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Time_ms | uint32_t | IN | Timeout<br>unit: ms<br>Default: 2000ms |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t Time_ms = 30;
ret = ECAT_McSetAxisInpTimeOut(DeviceNo, AxisNo, Time_ms);
if(ret < 0)
{
    printf("Failed to set axis inp timeout:%d\n", ret);
}
else
{
    printf("Set axis inp timeout successfully!\n");
}
```

## 7.2.49.  ECAT_McGetAxisInpTimeOut

**Description:**

Get the INP timeout of the specified axis.

**Syntax:**

int32_t  ECAT_McGetAxisInpTimeOut(uint16_t  DeviceNo,  uint16_t  AxisNo,  uint32_t *Time_ms)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Time_ms | uint32_t* | OUT | Timeout<br>unit: ms |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t Time_ms;
ret = ECAT_McGetAxisInpTimeOut(DeviceNo, AxisNo, &Time_ms);
if(ret < 0)
{
    printf("Failed to get axis inp timeout:%d\n", ret);
}
else
{
    printf("Axis[%u] Time_ms:%u\n", AxisNo, Time_ms);
}
```

## 7.2.50. ECAT_McSetAxisWanErrEnable

**Description:**

Set whether the specified Axis will change the status to MC_AS_ERRORSTOP when encountering a WAN (warning) signal.

Note: WAN (Warning) signal can be read using ECAT_McGetAxisDI

**Syntax:**

int32_t ECAT_McSetAxisWanErrEnable(uint16_t DeviceNo, uint16_t AxisNo, uint32_t Enable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Enable | uint32_t | IN | 0: Disable<br>1: Enable(Default) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.2.51.　ECAT_McGetAxisWanErrEnable

**Description:**

Get the WAN (warning) signal function setting of the specified Axis number.

**Syntax:**

int32_t　ECAT_McGetAxisWanErrEnable(uint16_t　DeviceNo,　uint16_t　AxisNo,　uint32_t *Enable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Enable | uint32_t* | OUT | 0: Disable<br>1: Enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t Enable;
ret = ECAT_McGetAxisWanErrEnable(DeviceNo, AxisNo, &Enable);
if(ret < 0)
{
    printf("Failed to get WanErr enable:%d\n", ret);
}
else
{
    printf("Axis[%u] WanErr enable:%u\n", AxisNo, Enable);
}
```

# 7.3. Axis Status

## 7.3.1. ECAT_McGetAxisActualPos

Description:

Get actual position of an axis.

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

Syntax:

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

Parameters:

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

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## Example:

### [C/C++]

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

## 7.3.2.　ECAT_McGetAxisCommandPos

**Description:**

Get command position of an axis.

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## Example:

### [C/C++]

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

### 7.3.3. ECAT_McGetAxisActualVel

**Description:**

Get actual velocity of an axis.

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## Example:

### [C/C++]

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

## 7.3.4. ECAT_McGetAxisActualPosVel

**Description:**

Get actual position and velocity of an axis.

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

## Example:

### [C/C++]

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

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

```
    }
}
```

## 7.3.5.   ECAT_McGetAxisState

### Description:

Get the state of an axis.

### Syntax:

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

### Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| State | uint32_t* | OUT | Axis state (Defined in Table 7.4) |

### Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.4: 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 discrete motion |
| MC_AS_CONTINUOUSMOTION | 6 | Axis is continuous motion |
| MC_AS_SYNCHRONIZEDMOTION | 7 | Axis is synchronized motion |

**Example:**

**[C/C++]**

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

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

## 7.3.6.　ECAT_McGetAxisLastError

### Description:

Get last error of an axis.

### Syntax:

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

### Parameters:

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

### Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.3.7.　ECAT_McGetAxisDriveError

### Description:

Get the drive error of an axis.

### Syntax:

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

### Parameters:

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

### Return:

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

**[C/C++]**

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

```
                }
            }
        }
    }
```

## 7.3.8.   ECAT_McGetAxisDI

**Description:**

Get digital inputs of an axis. Most of digital inputs are available in the drive.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| DI | uint32_t * | OUT | Digital input status (Defined in Table 7.5) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.5: Axis I/O status

| Bit Number | Description |
|---|---|
| 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 | INP (In position) Note: You need to set INP signal first, and then this Bit is valid |
| Bit 8~31 | Reserved |

| Example: |
|---|
| [C/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
            uint8_t INP      : 1;         //in position
            uint32_t    reserved : 24;   //Reserved(bit8~bit31)
        };
        uint32_t DIs;
    };
```

```
}axis_di_t;
/**************************************************/
int32_t ret;
axis_di_t AxisDI;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
ret = ECAT_McGetAxisDI(DeviceNo, AxisNo, &AxisDI.DIs);
if(ret < 0)
{
    printf("Failed to get axis DI:%d\n", ret);
}
else
{
    printf("AxisNo[%u]-+-AxisDI\n"
            "            |-NOT:%d\n"
            "            |-POT:%d\n"
            "            |-ORG:%d\n"
            "            |-ALM:%d\n"
            "            |-WAN:%d\n"
            "            |-SVN:%d\n"
            "            |-VIR:%d\n"
            "            |-INP:%d\n"
            "\n", AxisNo, AxisDI.NOT, AxisDI.POT, AxisDI.ORG
            , AxisDI.ALM, AxisDI.WAN, AxisDI.SVN, AxisDI.VIR, AxisDI.INP);
}
```

## 7.3.1. ECAT_McGetAxisDI_60FD

**Description:**

Get digital inputs(object 0x60FD) of an axis.

**Syntax:**

int32_t ECAT_McGetAxisDI_60FD(uint16_t DeviceNo, uint16_t AxisNo, uint32_t *DI)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| DI | uint32_t * | OUT | Digital input status |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t DI = 0;
ret = ECAT_McGetAxisDI_60FD(DeviceNo, AxisNo, &DI);
if(ret < 0)
{
    printf("Failed to get axis DI:%d\n", ret);
}
else
{
    printf("AxisNo[%u], AxisDI: %u "\n", AxisNo, DI);
}
```

## 7.3.2. ECAT_McGetAxisHomeState

**Description:**

Get Home state of an axis. Check if this axis has already executed home action successfully.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| State | uint16_t * | OUT | *ECAT_McAxisHome* is executed successfully after *ECAT_McInit* <br> 0: N <br> 1: Y |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

# 7.4. Axis Homing

## 7.4.1. ECAT_McSetAxisHomeMethod

**Description:**

Set the homing method of an axis.

Note: (1) homing method 38、39

The homing mode 38, 39 are torque homing mode, not in the CiA402 specification, supporting the CiA402 module (requires 6072h (Max torque), 6077h (Torque actual value), need to support homing mode 37)

Home 38
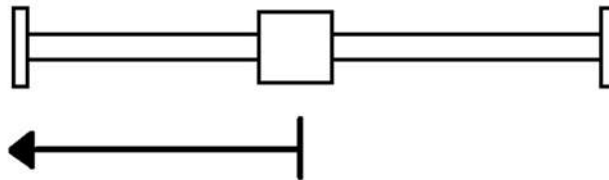
Torque actual value > Home torque

Home 39
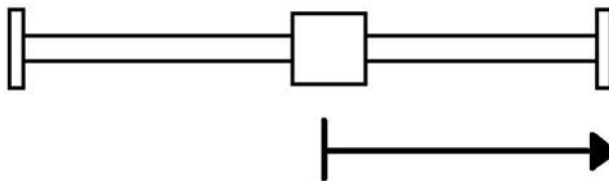
Torque actual value > Home torque

Note:(2) ECAT-2091S/ ECAT-2094S support home method 17、18、37
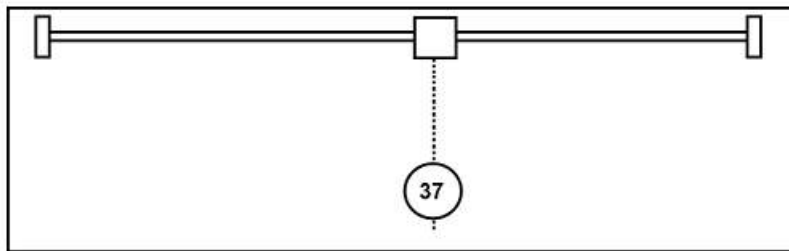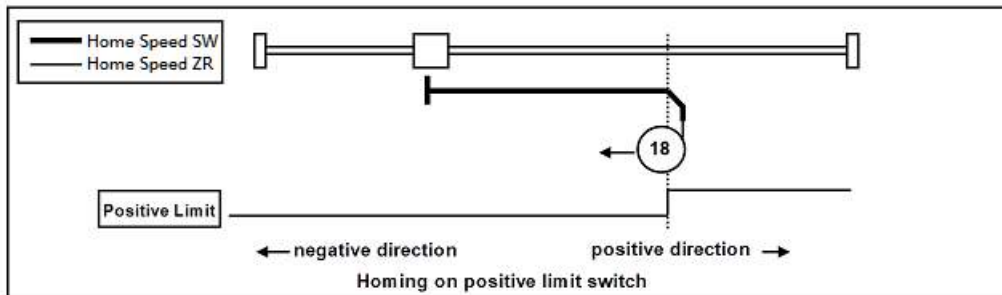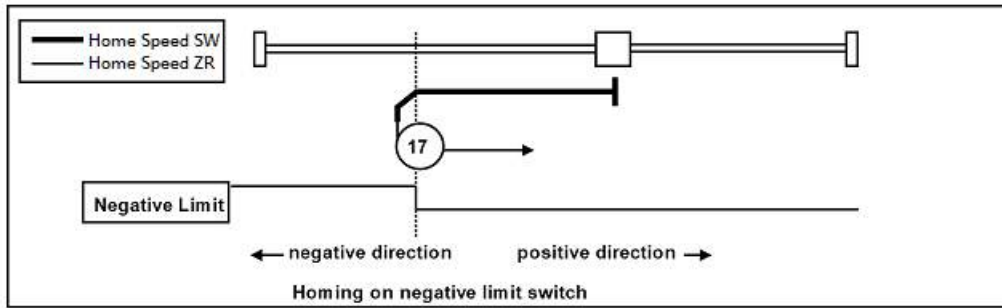


Homing on negative limit switch



Homing on positive limit switch

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Method | int32_t | IN | Homing method<br>(Refer to the drive's user manual) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.4.2.　ECAT_McGetAxisHomeMethod

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

### 7.4.3. ECAT_McSetAxisHomeSpeed

**Description:**

Set the homing speed settings of an axis. **SeachSw** speed is used for searching the home sensor; **SeachZr** speed is used for searching the encoder index Z signal.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**[C/C++]**

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

## 7.4.4.　ECAT_McGetAxisHomeSpeed

**Description:**

　　Get the homing speed settings of an axis. **SeachSw** speed is used for searching the home sensor; **SeachZr** speed is used for searching the encoder index Z signal.

**Syntax:**

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

**Parameters:**

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

**Return:**

　　0: Success.

　　Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
uint16_t DeviceNo = 0;

uint16_t AxisNo = 0;

double SeachSw;

double SeachZr;

...

ret = ECAT_McGetAxisHomeSpeed(DeviceNo, AxisNo, &SeachSw, &SeachZr);

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

## 7.4.5. ECAT_McSetAxisHomeAcc

**Description:**

Set homing acceleration of an axis.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.4.6.　ECAT_McGetAxisHomeAcc

**Description:**

Get homing acceleration of an axis.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

## 7.4.7. ECAT_McSetAxisHomeOffset

**Description:**

Set home offset to an axis.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

[C/C++]

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

## 7.4.8. ECAT_McGetAxisHomeOffset

**Description:**

Get home offset of an axis.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.4.9. ECAT_McSetAxisHomeTorque

**Description:**

Set homing torque of an axis.

Note:(1)for homing mode 38、39

(2) supporting the CiA402 module (requires 6072h (Max torque), 6077h (Torque actual value), need to support homing mode 37)

**Syntax:**

int32_t ECAT_McSetAxisHomeTorque(uint16_t DeviceNo, uint16_t AxisNo, uint16_t Torque);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Torque | uint16_t | IN | Home torque (unit: 0.1% of (rated torque 6076h)) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
int16_t Torque= 500;
ret = ECAT_McSetAxisHomeTorque(DeviceNo, AxisNo, Torque);
if(ret < 0)
{
    printf("Failed to set axis home torque:%d\n", ret);
}
```

## 7.4.10. ECAT_McGetAxisHomeTorque

**Description:**

Set homing torque of an axis.

**Syntax:**

int32_t ECAT_McGetAxisHomeTorque(uint16_t DeviceNo, uint16_t AxisNo, uint16_t *Torque);

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Torque | int16_t* | OUT | Home torque (unit: 0.1% of (rated torque 6076h)) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
int16_t Torque;
ret = ECAT_McGetAxisHomeTorque (DeviceNo, AxisNo, &Torque);
if(ret < 0)
{
    printf("Failed to get axis home torque:%d\n", ret);
}
else
{
    printf("Axis[%u] Home torque:%d\n", AxisNo, Torque);
}
```

## 7.4.11. ECAT_McAxisHome

**Description:**

Start home motion of an axis.

Note: Since a few servo drives do not support the dynamic settings of some home-related parameters, such as the home acceleration setting, an error may occur. Another function *ECAT_McAxisHomeEx* is provided for dealing with this kind of drives.

**Syntax:**

int32_t ECAT_McAxisHome(uint16_t DeviceNo, uint16_t AxisNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

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

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

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

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

        return;

    }


    ret = ECAT_McAxisHome(DeviceNo, AxisNo);

    if(ret < 0)

    {

        printf("Failed to start axis home:%d\n", ret);

    }

    else

    {

        do

        {

            sleep(1);

            ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);

        }while(State == MC_AS_HOMING) //Homing


        if(State == MC_AS_STANDSTILL) //StandStill

            printf("Axis homing successfully!\n");

        else if(State == MC_AS_ERRORSTOP) //ErrorStop

        {

            printf("Axis error stop\n");

        }

    }
```

## 7.4.12. ECAT_McAxisHomeEx

**Description:**

Start home motion of an axis.

Note: When this command is executed, some home-related parameters will be set via SDO communication. Please check parameters in Table 7.7 and the user manual of this CiA402 servo drive to defined as this axis to ensure SDOs related to Homing are exist.

**Syntax:**

int32_t ECAT_McAxisHomeEx(uint16_t DeviceNo, uint16_t AxisNo, uint16_t Settings)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Settings | uint16_t | IN | Home settings (As shown in Table 7.6) Each value represents a setting via SDO communication for that object in the slave. This value is obtained by adding values corresponding to those settings which are needed. |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.7 Home Settings

| Macro Definition | Value | Description | SDO  Index |
|---|---|---|---|
| MC_AS_HOME_SPEED_SW | 1 | Speed during search for switch | 6099h:01h |
| MC_AS_HOME_SPEED_ZR | 2 | Speed during search for z phase signal | 6099h:02h |
| MC_AS_HOME_ACC | 4 | Homing Acceleration | 609Ah:00h |
| MC_AS_HOME_OFFSET | 8 | Home offset | 607Ch:00h |

| Example: |
|---|
| [C/C++] |

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

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

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

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

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

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

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

# 7.5.  Axis Moving
## 7.5.1.  ECAT_McAxisErrorReset
**Description:**

Reset the error state of an axis.

After using this function, it takes a few ms to complete the reset axis state, you can use ECAT_McGetAxisState to get the axis state, when the axis state is MC_AS_STANDSTILL(In Servo On state) or MC_AS_DISABLED(In Servo Off state), it means that the reset axis state is completed.

**Syntax:**

int32_t ECAT_McAxisErrorReset(uint16_t DeviceNo, uint16_t AxisNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

## 7.5.2.   ECAT_McAxisMoveAbs

**Description:**

Start an absolute position motion of an axis.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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


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

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

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

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

### 7.5.3.    ECAT_McAxisMoveRel

**Description:**

Start a relative position motion of an axis.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

[C/C++]

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

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

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

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

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

## 7.5.4. ECAT_McAxisMoveAbs_P2P

**Description:**

Start a point-to-point absolute position motion of an axis.

Note: This motion will use the maximum velocity of the spcified axis, which is defined by *ECAT_McSetAxisMaxVelocity*.

**Syntax:**

int32_t ECAT_McAxisMoveAbs_P2P(uint16_t DeviceNo, uint16_t AxisNo, double Pos)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double MaxVelocity = 100;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo,Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}

ret = ECAT_McSetAxisMaxVelocity(DeviceNo, AxisNo, MaxVelocity);
if(ret < 0)
{
    printf("Failed to set axis MaxVelocity:%d\n",ret);
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs_P2P(DeviceNo, AxisNo, AxisPos);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {

        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
```

```
        }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion


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

## 7.5.5.　ECAT_McAxisMoveRel_P2P

**Description:**

Start a point-to-point relative position motion of an axis.

Note: This motion will use the maximum velocity of the spcified axis, which is defined by *ECAT_McSetAxisMaxVelocity*.

**Syntax:**

int32_t ECAT_McAxisMoveRel_P2P(uint16_t DeviceNo, uint16_t AxisNo, double Pos)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 500;
double AxisPos = 10.0;
double MaxVelocity = 100;
ret = ECAT_McSetAxisAccTime(DeviceNo, AxisNo,Time_ms);
if(ret < 0)
{
    printf("Failed to set axis acceleration time:%d\n", ret);
    return;
}
ret = ECAT_McSetAxisMaxVelocity(DeviceNo, AxisNo, MaxVelocity);
if(ret < 0)
{
    printf("Failed to set axis MaxVelocity:%d\n",ret);
}

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveRel_P2P(DeviceNo, AxisNo, AxisPos);
    if(ret < 0)
    {
        printf("Failed to start axis move rel:%d\n", ret);
    }
    else
    {

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

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

## 7.5.6. ECAT_McAxisChangePos

**Description:**

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

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Pos | double | IN | End positon<br>It is an absolute position.<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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 change position function:%d\n", ret);
            }
```

```
        }

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

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

## 7.5.7. ECAT_McAxisChangeVel

**Description:**

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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 relatively:%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 change velocity function:%d\n", ret);
            }
```

```
        }

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

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

## 7.5.8.   ECAT_McAxisMoveSuperimposed

**Description:**

Start a relative position motion of an axis additional to an existing motion.

Note:

ECAT_McAxisChangePos and ECAT_McAxisChangeVel cannot be used during this command execution.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
uint16_t Time_ms = 100;
double AxisPos = 6.0;
double AxisVel = 3.0;
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(1);
        ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
        if (State == MC_AS_DISCRETEMOTION) //DiscreteMotion
        {
            AxisPos = 2.0;
            AxisVel = 2.0;
            ret = ECAT_McAxisMoveSuperimposed(DeviceNo, AxisNo, AxisPos, AxisVel);
            if(ret < 0)
            {
                printf("Failed to call AxisMoveSuperimposed function:%d\n", ret);
            }
        }
```

```
        }

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

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

Command 1
AxisMoveAbs

velocity = 3,
position = 6

Command 2
AxisMoveSuperimposed

velocity = 2,
position = 2

Command 1+
Command 2

position = 8

Velocity
Position

## 7.5.9. ECAT_McAxisHaltSuperimposed

**Description:**

Stop a Superimposed motion of an axis with deceleration.

**Syntax:**

int32_t ECAT_McAxisHaltSuperimposed(uint16_t DeviceNo, uint16_t AxisNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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_McAxisMoveSuperimposed(DeviceNo, AxisNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move superimposed:%d\n", ret);
    }
    else
    {
        sleep(1);
        ret = ECAT_McAxisHaltSuperimposed(DeviceNo, AxisNo);
        if(ret < 0)
        {
            printf("Failed to stop axis move:%d\n", ret);
            return;
        }
        else
        {
            do
            {
```

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

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

## 7.5.10. ECAT_McAxisMoveVel

**Description:**

Start a never ending movement with a specified velocity.

Note: A velocity control mode (CSV) is used.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

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

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

## 7.5.11. ECAT_McAxisMoveVelByPos

**Description:**

Start a velocity motion with a specified velocity.

Note: A position control mode (CSP) is used.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

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

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

## 7.5.12.  ECAT_McAxisGearIn

**Description:**

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

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| MasterNo | uint16_t | IN | Master axis (just for reference) |
| SlaveNo | uint16_t | IN | Slave axis<br>Its speed will be changed! |
| RatioNum | int32_t | IN | Gear ratio numerator |
| RationDen | uint32_t | IN | Gear ratio denominator |
| SyncSource | uint16_t | IN | Slave reference source for Synchronization<br>0: command position of master axis<br>1: real position of master axis<br>(Defined in Table 7.8) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.9: Source for synchronization

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

| Example: |
|---|
| [C/C++] |

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

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

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

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

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

## 7.5.13.  ECAT_McAxisGearOut

**Description:**

Disengages the slave axis from the master axis. After disengagement, the slave axis can either keep moving with the last velocity, stop slowly, or stop immediately.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave axis |
| Stop | uint16_t | IN | 0: Constant velocity motion<br>1: Stop<br>2: Quick stop |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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


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

## 7.5.14. ECAT_McAxisMoveProfile

**Description:**

Start a profile position motion of an axis. A profile buffer is an array that contains a lot of pre-defined motion points. Up to 3000 points can be defined for a single profile. If more than 3000 points are required, please use function *ECAT_McAxisMoveProfileCSV*. Function *ECAT_McSetProfileInterval* will affect the data consuming rate.

Note: Set profile by using *ECAT_McSetProfileData.*

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| ProfileNo | uint16_t | IN | Profile buffer number<br>Available number range: 0~15 |
| TotalStep | uint16_t | IN | Total moving steps (Maximum: 3000) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

## 7.5.15. ECAT_McAxisMoveProfileCSV

**Description:**

Start a profile position motion of an axis. A file contains all the position data for a profile motion. Its format is shown in Figure 7.1. Function *ECAT_McSetProfileInterval* will affect the data consuming rate.

Note: Set profile by using *ECAT_McSetProfileCSV*.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| ProfileNo | uint16_t | IN | Profile buffer number<br>Available number range: 0~15 |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

## 7.5.16.  ECAT_McAxisCamIn

**Description:**

Start E-CAM synchronization motion with a table defining the relationship of a slave axis and its master axis.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| MasterNo | uint16_t | IN | Master axis number |
| SlaveNo | uint16_t | IN | Slave axis number |
| TableNo | int16_t | IN | E-CAM table number |
| SyncSource | uint16_t | IN | Slave reference source for Synchronization<br>0: command position of master axis<br>1: real position of master axis<br>(Defined in Table 7.9) |
| MasterInterval | double | IN | Master Interval (unit: User Unit)<br>It is a distance for the master axis corresponding to a distance between two continuous positions defined for the slave axis. |
| SlaveScaling | double | IN | Slave position output ratio |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t TableNo = 0;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double Data[1000];
uint16_t DataSize = 1000;
uint8_t SlaveAbs = 0;    // set the data type to be the Relative type
double MasterInterval = 0.001;
double SlaveScaling = 1.0;
double AxisPos = 3.0;
double AxisVel = 1;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;
/**
***Write E-CAM Table data to Data[1000]
**/
ret = ECAT_McSetEcamTable(DeviceNo, TableNo, Data, DataSize); // fill the data into a table
if(ret < 0)
{
    printf("Failed to set E-CAM table data:%d\n", ret);
    return;
}

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

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

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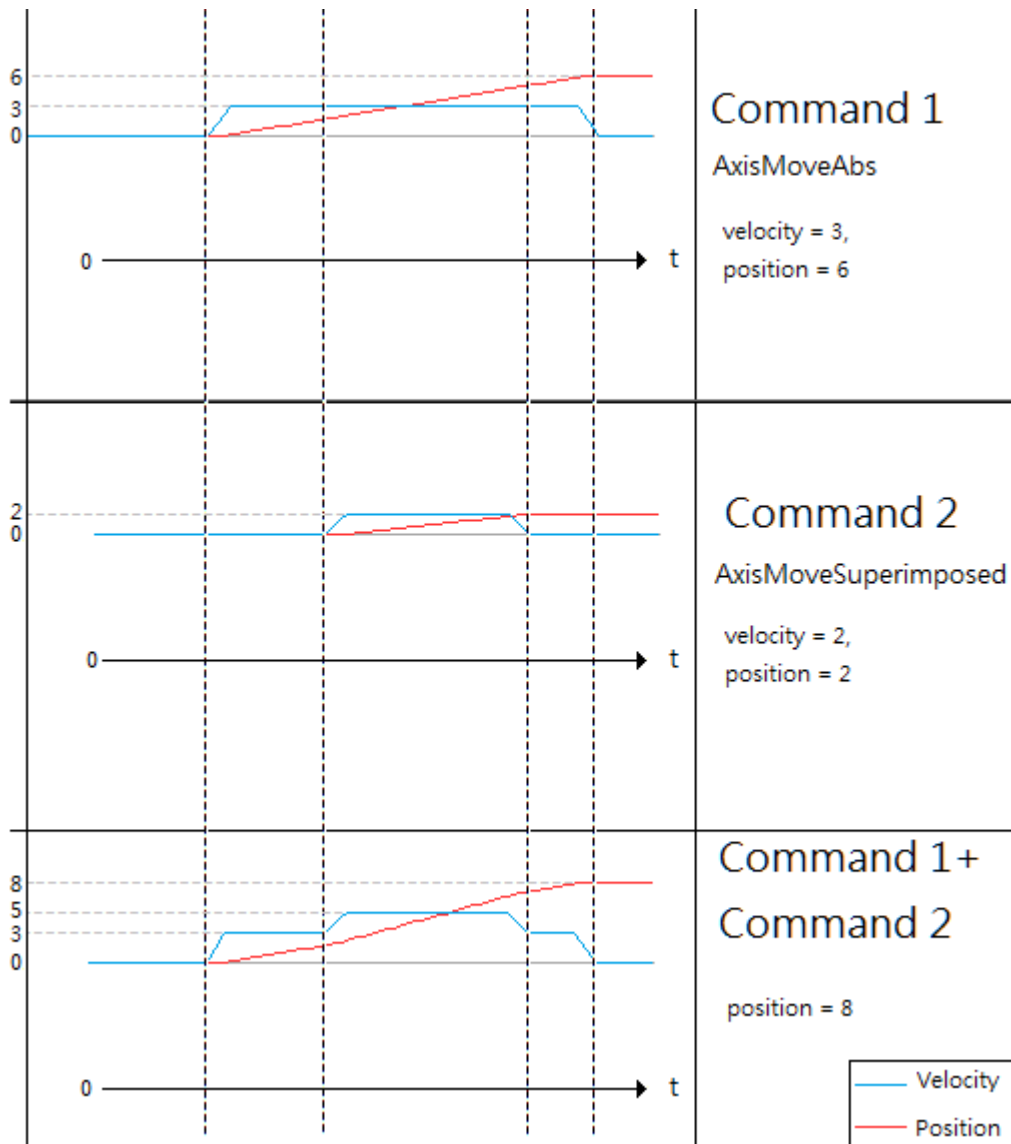
E-CAM Table:



E-CAM synchronization motion diagram:

## 7.5.17. ECAT_McAxisCamPhaseShift

**Description:**

Set the phase shift between the master axis and the slave axis for an E-CAM synchronization motion. Phase shift changes the starting point of the master axis relative to the slave axis in the CAM table.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave axis number |
| PhaseShift | double | IN | Master phase shift (unit: User Unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t TableNo = 0;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double Data[1000];
uint16_t DataSize = 1000;
uint8_t SlaveAbs = 0; //Relative
double MasterInterval = 0.001;
double SlaveScaling = 1.0;
double AxisPos = 3.0;
double AxisVel = 1;
double PhaseShift = -0.5;
uint16_t SyncSource = MC_AXIS_SYNC_SOURCE_ACTUAL_VALUE;
/**
***Write E-CAM Table data to Data[1000]
**/
ret = ECAT_McSetEcamTable(DeviceNo, TableNo, Data, DataSize);
if(ret < 0)
{
    printf("Failed to set E-CAM table data:%d\n", ret);
    return;
}

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

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

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

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

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

E-CAM Table:



E-CAM synchronization motion:



After setting the phase shift of the master:
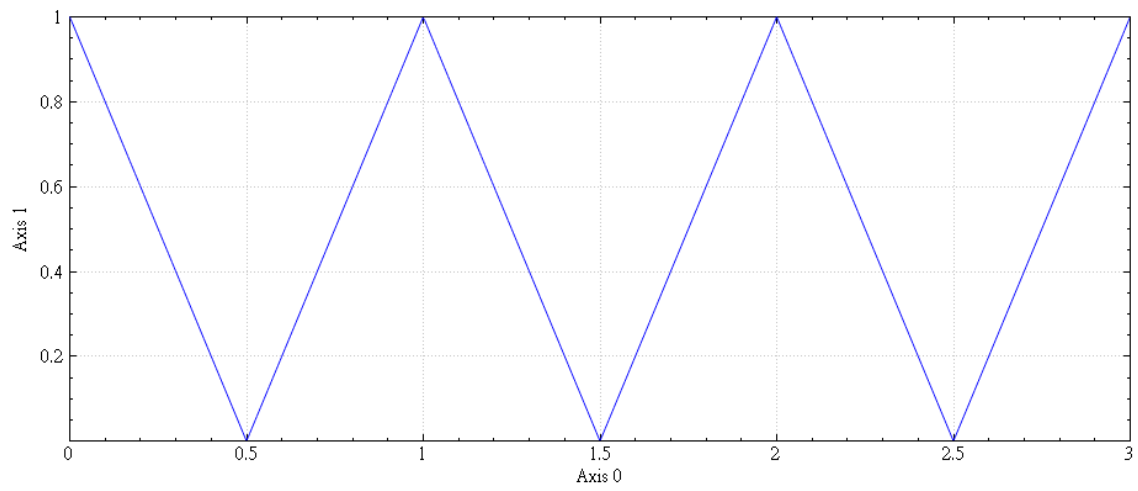
## 7.5.18. ECAT_McAxisCamOut

**Description:**

Stop a slave axis for performing an E-CAM synchronization motion. After an axis is set to Cam out, it stops immediately.

**Syntax:**

int32_t ECAT_McAxisCamOut(uint16_t DeviceNo, uint16_t SlaveNo)

**Parameters:**

| Name | Type | IN or OUT | Description |
| --- | --- | --- | --- |
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave axis number |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

## 7.5.19.  ECAT_McAxisTangentInGroup

**Description:**

Start a tangent motion. Tangent motion is a name simplified from tangential following motion. It defines an axis to rotate by following the tangential direction of a continuous profile which is generated by a two-axis group motion. If the vector direction is not continuous for a new group motion, the rotating axis is assigned a new angle to match with the new direction by calling this tangent-in function.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number (a single rotary tool axis) |
| GroupNo | uint16_t | IN | Group number |
| Angle | int16_t | IN | Tangent angle (Unit: degrees) This is the desired angle when this tangent-in command is issued. The rotary motion is executed with the velocity defined by Vel parameter. |
| Vel | uint16_t | IN | Rotate to tangent angle with this velocity (Unit: $degrees/s$) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

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

```
        }

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

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

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

```
            } else
                task_index++;
            break;
        case 3:
            ret = check_axis_tangent_in();
            if (ret == -1)
                task_stop = true;
            else if (ret == 0)
                task_index++;
            break;
        case 4:
            GroupPos[0] = 0.0;
            GroupPos[1] = 10.0;
            GroupVel = 5;
            ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
            if(ret != 0) {
                printf("Failed to add group move line command:%d\n", ret);
                task_stop = true;
            } else
                task_index++;
            break;
        case 5:
            ret = check_grp_state();
            if (ret == -1)
                task_stop = true;
            else if (ret == 0)
                task_index++;
            break;
        case 6:
            AxisAngle = 0;
            AxisVel = 90;
            ret = ECAT_McAxisTangentInGroup(DeviceNo, TangentInAxisNo, GroupNo, AxisAngle,
AxisVel);
            if (ret != 0) {
                printf("Axis tangent in failed:%d\n", ret);
                task_stop = true;
            } else
                task_index++;
```

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

            }

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

## Tangent motion path of example:
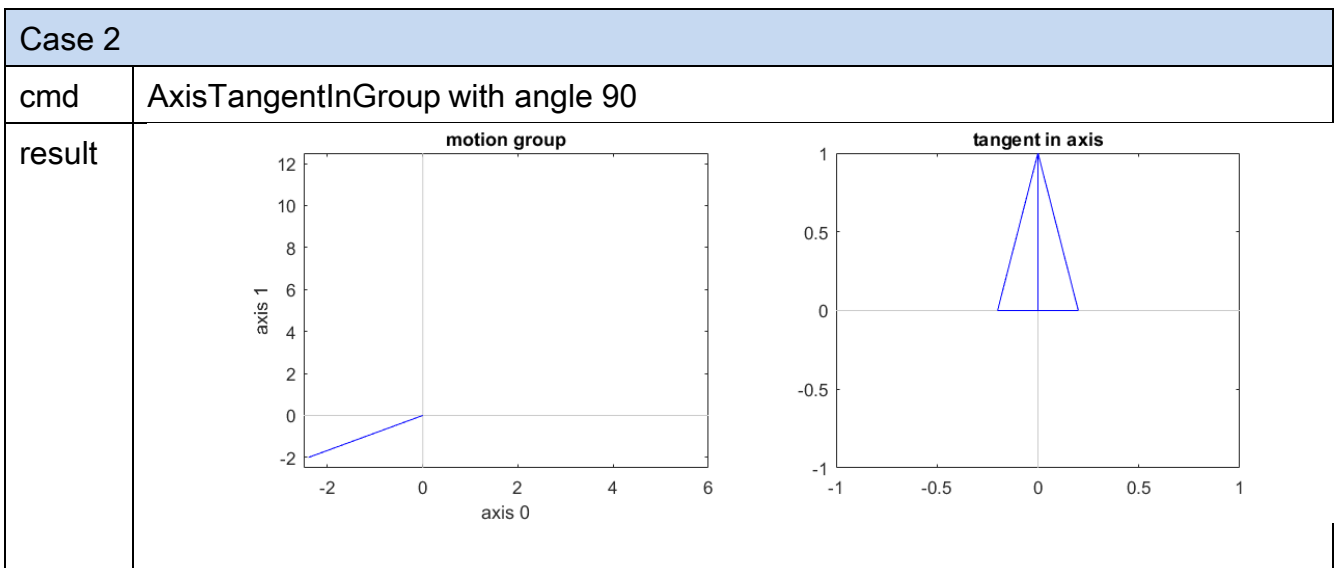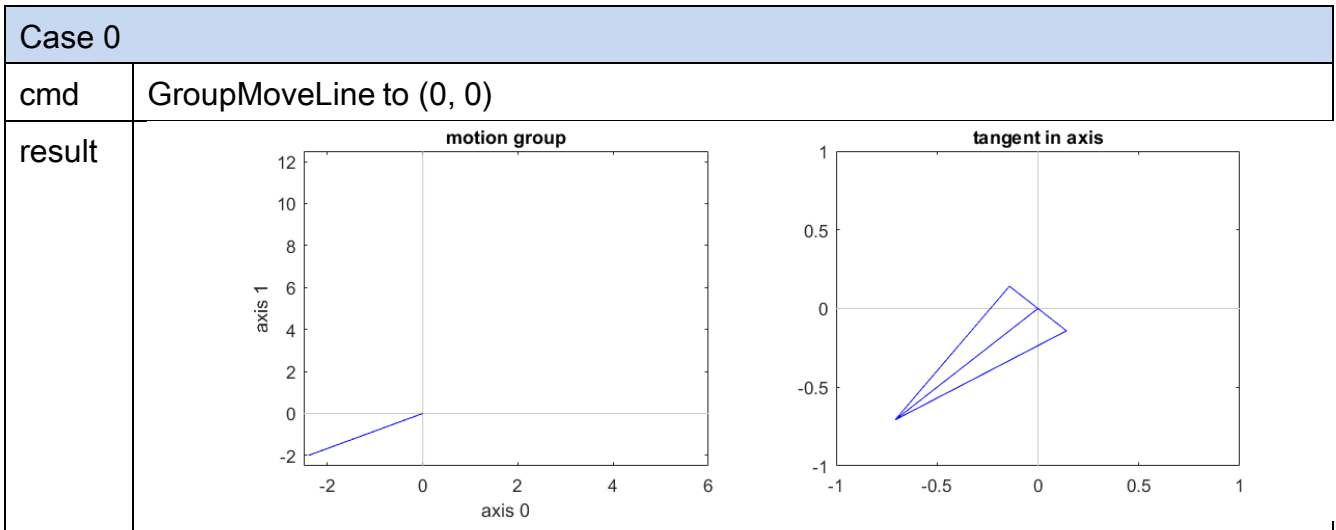
| Case 0 | |
|---|---|
| cmd | GroupMoveLine to (0, 0) |
| result |  |

| Case 2 | |
|---|---|
| cmd | AxisTangentInGroup with angle 90 |
| result |  |

| Case 4 | |
|---|---|
| cmd | GroupMoveLine to (0, 10) |
| result |  |

| Case 6 | |
|---|---|
| cmd | AxisTangentInGroup with angle 0 |
| result |  |

| Case 8 | |
|--------|---|
| cmd | GroupMoveCircular to angle -180 |
| result |  |
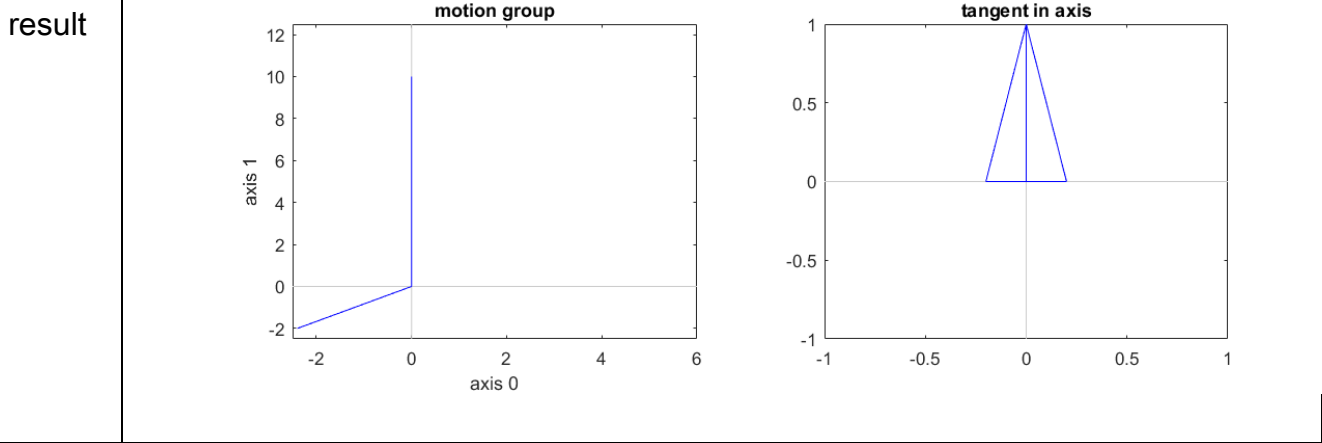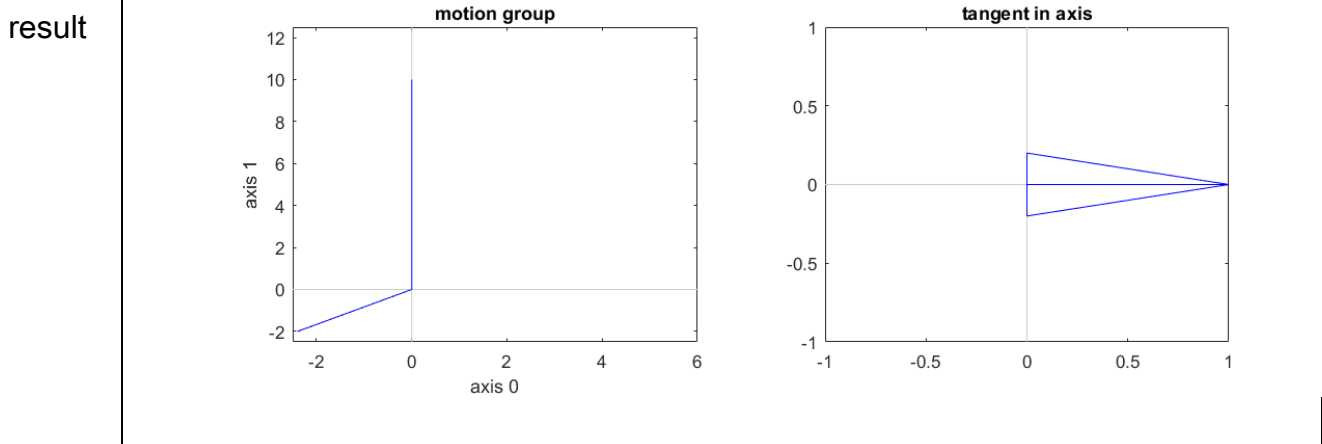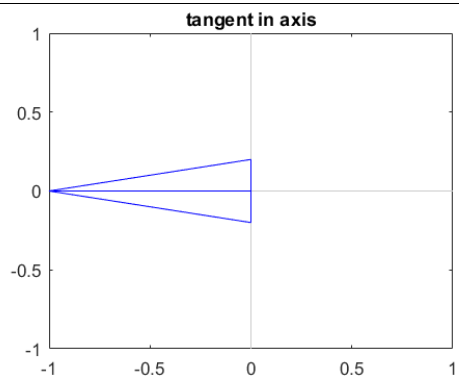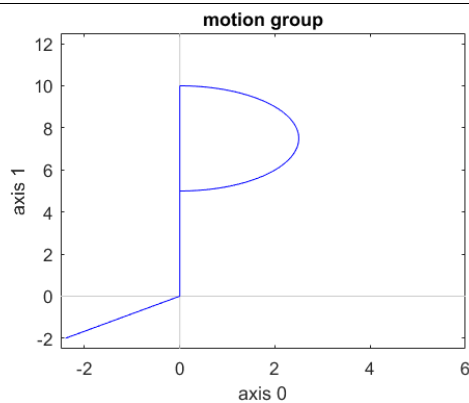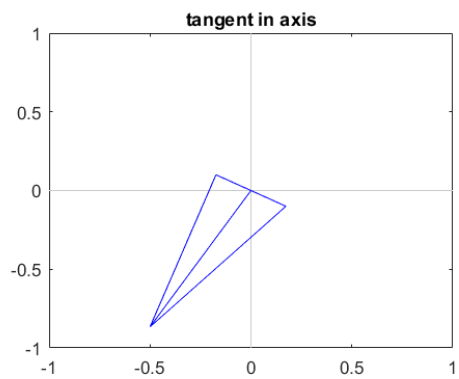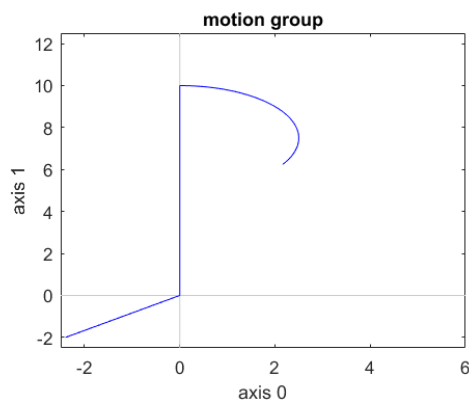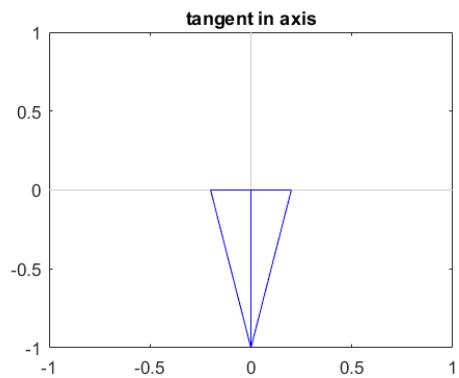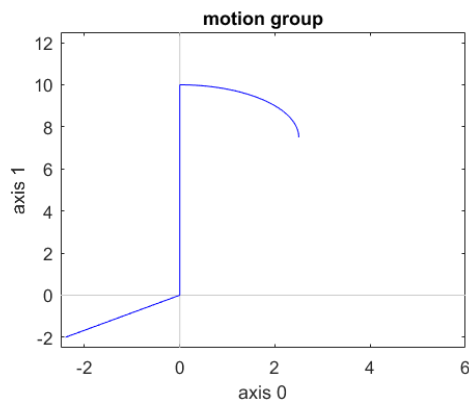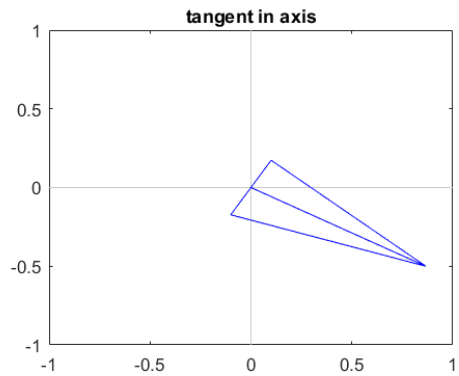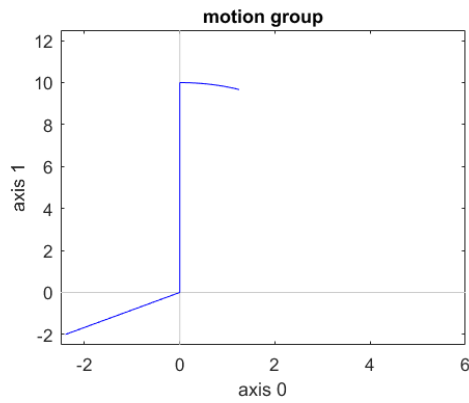
## 7.5.20. ECAT_McAxisGantryIn

**Description:**

Start a gantry control synchronization motion. Similar to an electrical gear control, the slave axis will follow command position of master axis motion with gear ratio 1 or -1. Enable the feedforward velocity gain will make the servo response for both axes faster; thus, improve the synchronization performance.

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

(2) Use ECAT_McSetAxisVelocityFeedForwardGain to adjust the feedforward gain to get better synchronization performance.

(3) The Master axis number must be smaller than the slave axis number, otherwise the position command of Slave axis will lag behind a CycleTime.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| MasterNo | uint16_t | IN | Master axis |
| SlaveNo | uint16_t | IN | Slave axis |
| Direction | int32_t | IN | Direction of both axes are the same or not<br>1: the same<br>-1: different |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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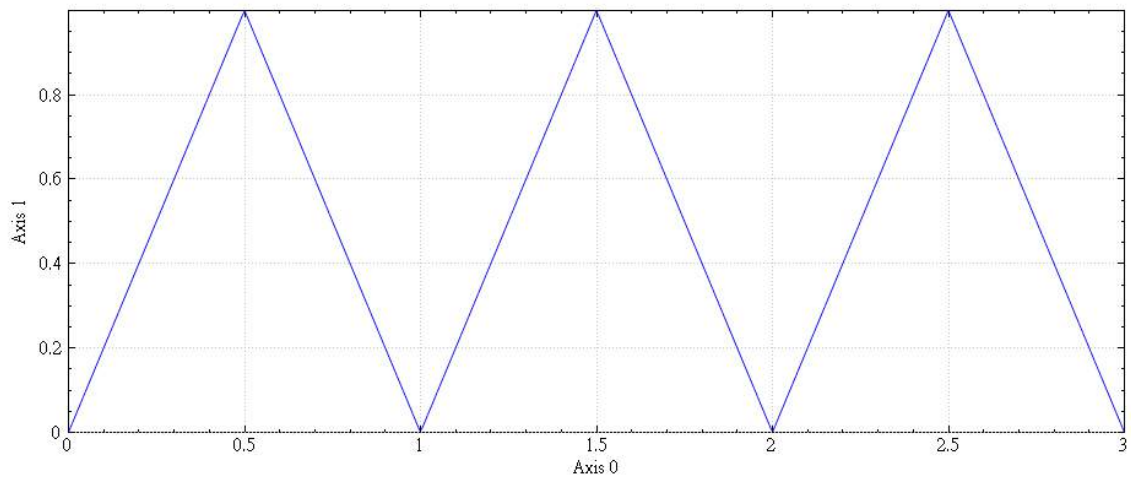
```
        if(State == MC_AS_STANDSTILL) //StandStill
            printf("Axis move successfully!\n");
        else if(State == MC_AS_ERRORSTOP) //ErrorStop
        {
            printf("Axis error stop\n");
        }
    }
}
```

## 7.5.21. ECAT_McAxisGantryMaxPosDiff

**Description:**

Set the limitation of position deviation of the master axis and the slave axis. If the position deviation is greater than the set value, it will trigger the error stop.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave axis |
| Value | double | IN | Maximum position deviation (positive or zero) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

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

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

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

## 7.5.22. ECAT_McAxisGantryMaxPosDiffStatus

**Description:**

Enable or disable the checking of maximum position deviation status for gantry control.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave axis |
| Status | uint16_t | IN | Enable the checking or not<br>0: Disable<br>1: Enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

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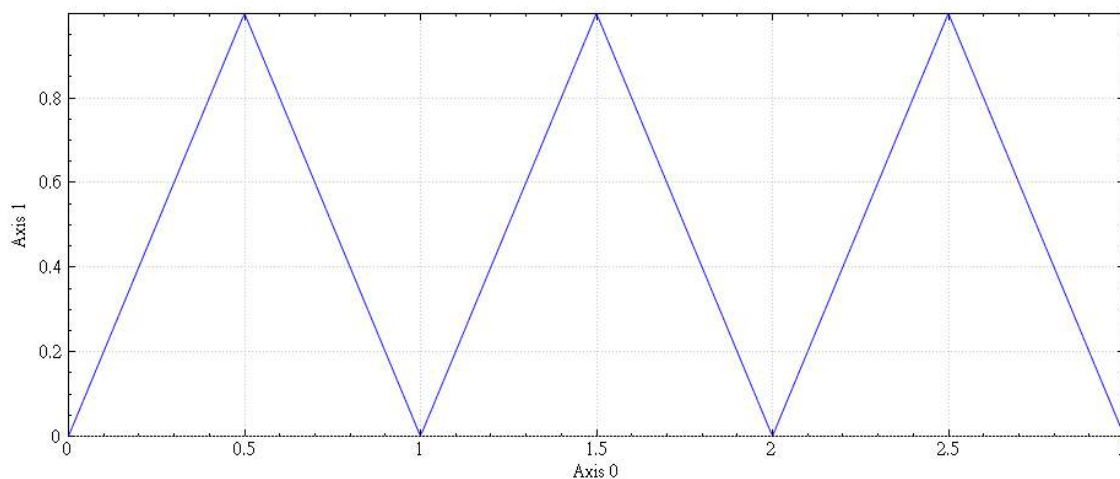
```
if(State == MC_AS_STANDSTILL) //StandStill
{
    ret = ECAT_McAxisMoveAbs(DeviceNo, MasterNo, AxisPos, AxisVel);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetAxisState(DeviceNo,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

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

## 7.5.1.　ECAT_McAxisGantryGain

**Description:**

　　　Please use McSetAxisVelocityFeedForwardGain to adjust the feedforward gain first. If you are not satisfied with the result, use this function. If the parameter is set badly, it may cause oscillation. Please set it carefully. You can use ECAT_McAxisGantryMaxPosDiff to set the maximum position error or use emergency stop. When the position is oscillating/diverge, you can stop it in time.

　　　Users can use Gantry Utility to tune gantry gain (Firmware vir. Must be 1.0.16 or above),

　　　To download the software and manual, please refer to the chapter "Software Installation"
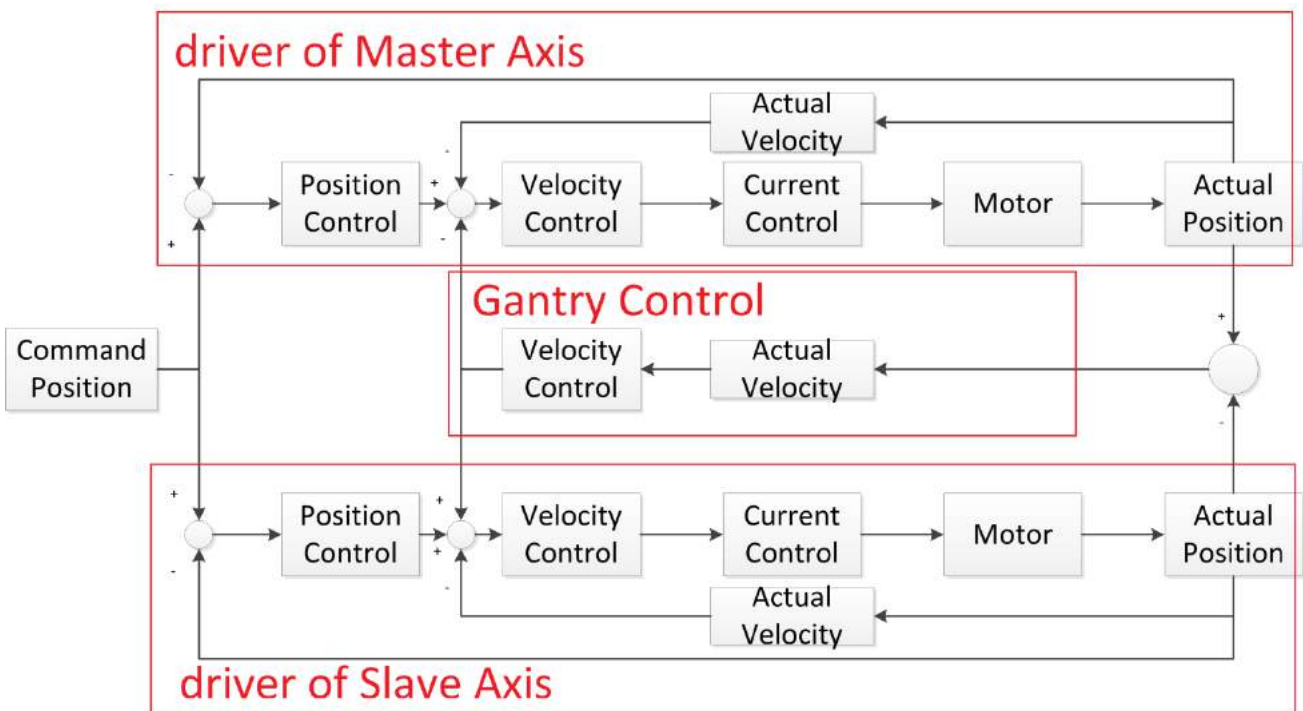


Set the PI gain of the gantry control. The servo driver needs to support the 60B1h velocity offset.

Note: (1) The parameter starts from zero and increases by 0.1 each time. If the parameter is too large, it may cause oscillation. Please set it carefully.

**Syntax:**

int32_t ECAT_McAxisGantryGain(uint16_t DeviceNo, uint16_t SlaveNo, double Kp, double Ki)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| SlaveNo | uint16_t | IN | Slave axis |
| Kp | double | IN | proportional gain for velocity loop PID controller |
| Ki | double | IN | integral gain for velocity loop PID controller |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t MasterNo = 0;
uint16_t SlaveNo = 1;
uint32_t State;
double AxisPos = 10.0;
double AxisVel = 4;
int32_t Direction= 1;
double Value = 1;
uint16_t Status = 1;
double Kp = 0.1;
double Ki = 0.1;


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

```
        ret = ECAT_McAxisGantryGain(DeviceNo, SlaveNo, Kp, Ki)
        if(ret < 0)
        {
            printf("Set Axis gantry Gain is falied:%d\n", ret);
            return;
        }

    }

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

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

## 7.5.2.　ECAT_McAxisMoveAbsAdv

**Description:**

　　Start an absolute position motion of an axis.

**Syntax:**

　　int32_t ECAT_McAxisMoveAbsAdv(uint16_t DeviceNo, uint16_t AxisNo, double EndPos, double StartVel, double ReqVel, double FinalVel, double Accel, double Decel, uint8_t AccDecMode)

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| EndPos | double | IN | Absolute position (Unit: user unit) |
| StartVel | double | IN | Start velocity (Unit: $\text{user unit}/s$) |
| ReqVel | double | IN | Target velocity (Unit: $\text{user unit}/s$) |
| FinalVel | double | IN | Final velocity (Unit: $\text{user unit}/s$) |
| Accel | double | IN | Acceleration rate ($\text{user unit}/s^2$) or acceleration time (second) |
| Decel | double | IN | Deceleration rate ($\text{user unit}/s^2$) or deceleration time (second) |
| AccDecMode | uint8_t | IN | Acceleration and deceleration input mode: <br> 0: acceleration and deceleration rate ($\text{user unit}/s^2$) <br> 1: acceleration and deceleration time (second) |

**Return:**

　　0: Success.

　　Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
double start_vel;
double req_vel;
double final_vel;
double accel;
double decel;
double end_pos;

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    //Command 1
    start_vel = 0;
    req_vels = 2;
    final_vel = 2;
    accel = 4;
    decel = 4;
    end_pos = 2;
    ret = ECAT_McAxisMoveAbsAdv(DeviceNo, AxisNo, end_pos,
                    start_vel, req_vel, final_vel, accel, decel, 0);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    //Command 2
    start_vel = 2;
    req_vels = 2.5;
    final_vel = 1.5;
    accel = 6;
    decel = 6;
    end_pos = 5;
    ret = ECAT_McAxisMoveAbsAdv(DeviceNo, AxisNo, end_pos,
```

```
                    start_vel, req_vel, final_vel, accel, decel, 0);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);


    //Command 3
    start_vel = 1.5;
    req_vels = 1.5;
    final_vel = 0;
    accel = 2;
    decel = 2;
    end_pos = 7;
    ret = ECAT_McAxisMoveAbsAdv(DeviceNo, AxisNo, end_pos,
                    start_vel, req_vel, final_vel, accel, decel, 0);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);


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

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

}
```
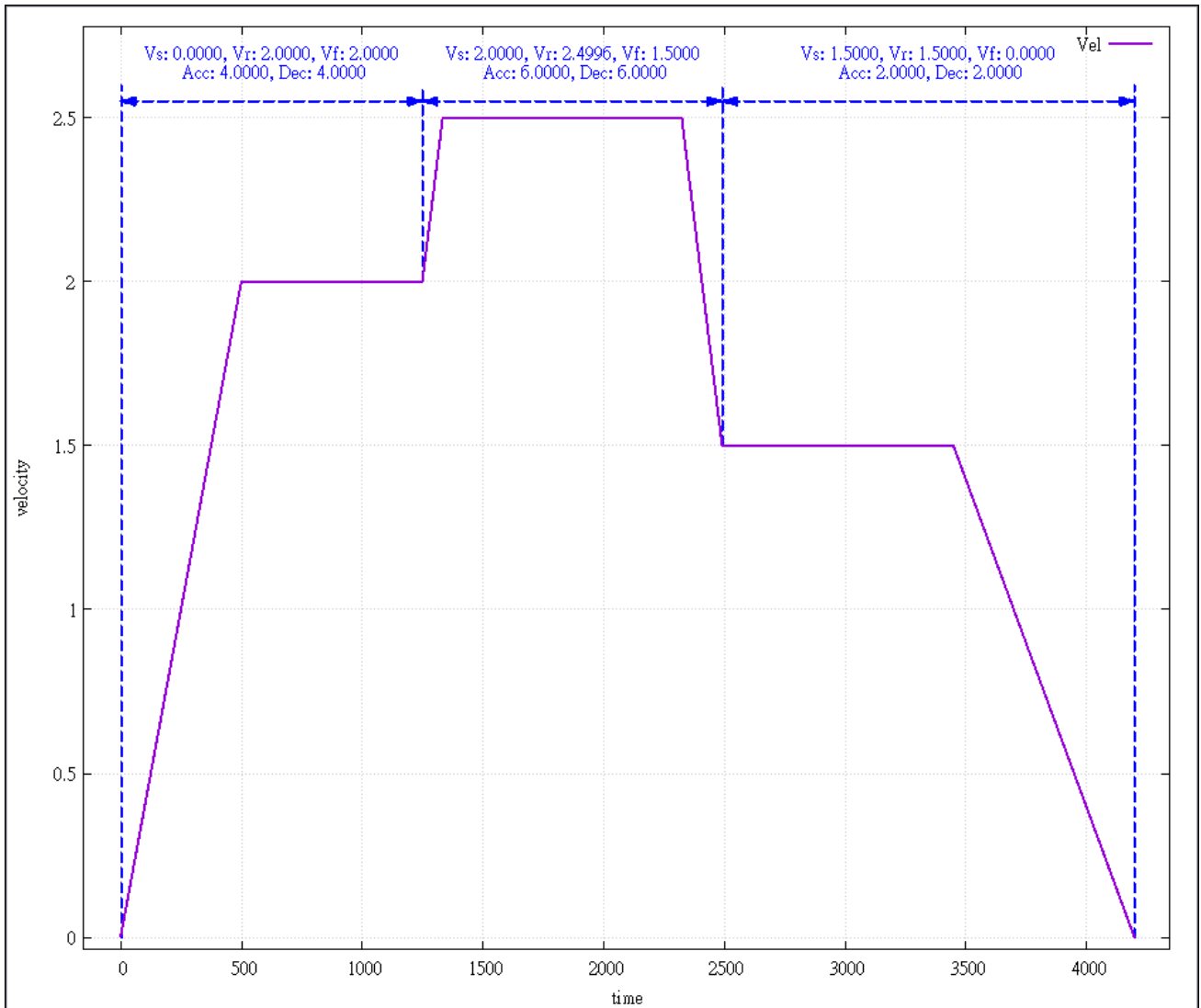
## 7.5.3.　ECAT_McAxisMoveRelAdv

**Description:**

Start a relative position motion of an axis.

**Syntax:**

int32_t ECAT_McAxisMoveRelAdv(uint16_t DeviceNo, uint16_t AxisNo, double EndPos, double StartVel, double ReqVel, double FinalVel, double Accel, double Decel, uint8_t AccDecMode)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| EndPos | double | IN | Relative distance (Unit: user unit) |
| StartVel | double | IN | Start velocity (Unit: $\text{user unit}/\text{s}$) |
| ReqVel | double | IN | Target velocity (Unit: $\text{user unit}/\text{s}$) |
| FinalVel | double | IN | Final velocity (Unit: $\text{user unit}/\text{s}$) |
| Accel | double | IN | Acceleration rate ($\text{user unit}/s^2$) or acceleration time (second) |
| Decel | double | IN | Deceleration rate ($\text{user unit}/s^2$) or deceleration time (second) |
| AccDecMode | uint8_t | IN | Acceleration and deceleration input mode: <br> 0: acceleration and deceleration rate ($\text{user unit}/s^2$) <br> 1: acceleration and deceleration time (second) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
double start_vel;
double req_vel;
double final_vel;
double accel;
double decel;
double end_pos;

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    //Command 1
    start_vel = 0;
    req_vels = 2;
    final_vel = 2;
    accel = 4;
    decel = 4;
    end_pos = 2;
    ret = ECAT_McAxisMoveAbsAdv(DeviceNo, AxisNo, end_pos,
                    start_vel, req_vel, final_vel, accel, decel, 0);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    //Command 2
    start_vel = 2;
    req_vels = 2.5;
    final_vel = 1.5;
    accel = 6;
    decel = 6;
    end_pos = 3;
    ret = ECAT_McAxisMoveRelAdv(DeviceNo, AxisNo, end_pos,
```

```
                        start_vel, req_vel, final_vel, accel, decel, 0);
        if (ret != 0)
                printf("Failed to add move command:%d\n", ret);


        //Command 3
        start_vel = 1.5;
        req_vels = 1.5;
        final_vel = 0;
        accel = 2;
        decel = 2;
        end_pos = 2;
        ret = ECAT_McAxisMoveRelAdv(DeviceNo, AxisNo, end_pos,
                        start_vel, req_vel, final_vel, accel, decel, 0);
        if (ret != 0)
                printf("Failed to add move command:%d\n", ret);


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

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

}
```
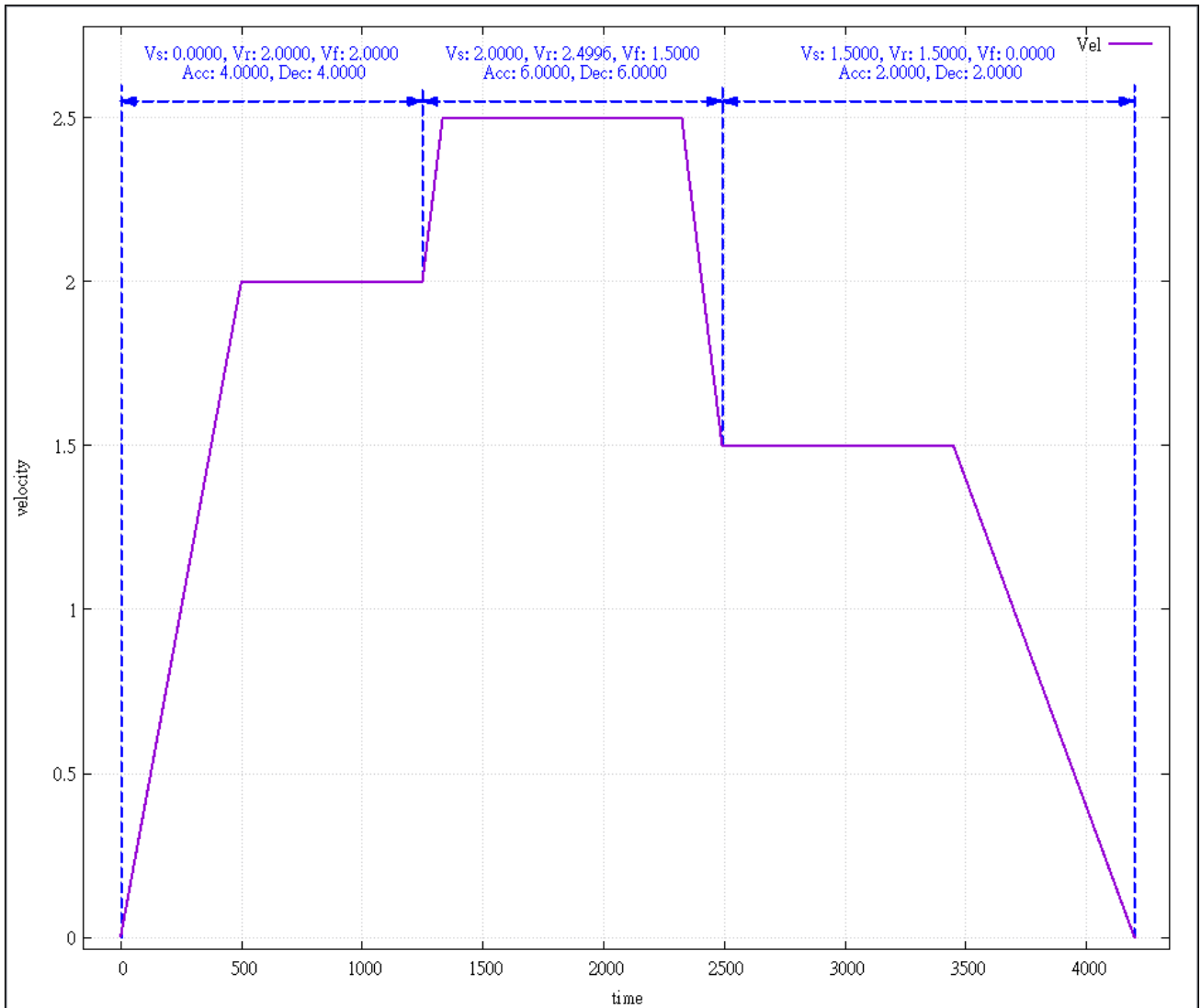
## 7.5.1.　ECAT_McAxisMove_CiA402_PP

**Description:**

Start the single-axis motion in CiA402 profile position mode.

Note: This function contains SDO commands

**Syntax:**

int32_t ECAT_McAxisMove_CiA402_PP(uint16_t DeviceNo, uint16_t AxisNo, uint8_t Abort, uint8_t AbsMove, double EndPos, double Vel, double Accel, double Decel)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Abort | uint8_t | IN | Aborting current command |
| AbsMove | uint8_t | IN | Absolute move mode:<br>0: relative<br>1: absolute |
| EndPos | double | IN | Absolute position or relative distance (Unit: user unit) |
| Vel | double | IN | Target velocity (Unit: user unit/s) |
| Accel | double | IN | Acceleration rate (user unit/$s^2$) |
| Decel | double | IN | Deceleration rate (user unit/$s^2$) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
double req_vel;
double accel;
double decel;
double end_pos;

ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    //Command 1
    req_vels = 2;
    accel = 4;
    decel = 4;
    end_pos = 2;
    ret = ECAT_McAxisMove_CiA402_PP(DeviceNo, AxisNo, 0, 1,
                    end_pos, req_vel, accel, decel);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    //Command 2
    req_vels = 2.5;
    accel = 6;
    decel = 6;
    end_pos = 3;
    ret = ECAT_McAxisMove_CiA402_PP(DeviceNo, AxisNo, 0, 1,
                    end_pos, req_vel, accel, decel);
    if (ret != 0)
        printf("Failed to add move command:%d\n", ret);

    do
    {
        sleep(1);
```

```
            ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
      }while(State == MC_AS_DISCRETEMOTION) //DiscreteMotion

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


   }
```

## 7.5.1.　ECAT_McAxisMove_CiA402_PV

**Description:**

Start the single-axis motion in CiA402 profile vecolity mode.

Note: This function contains SDO commands

**Syntax:**

int32_t ECAT_McAxisMove_CiA402_PV(uint16_t DeviceNo, uint16_t AxisNo, double Vel, double Accel, double Decel)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| EndPos | double | IN | Absolute position or relative distance (Unit: user unit) |
| Vel | double | IN | Target velocity (Unit: $\text{user unit}/s$) |
| Accel | double | IN | Acceleration rate ($\text{user unit}/s^2$) |
| Decel | double | IN | Deceleration rate ($\text{user unit}/s^2$) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
double req_vel;
double accel;
double decel;


ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    //Command 1
    req_vels = 2;
    accel = 4;
    decel = 4;
    ret = ECAT_McAxisMove_CiA402_PV(DeviceNo, AxisNo, req_vel, accel, decel);
    if (ret != 0)
        printf("Failed to start move:%d\n", ret);

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

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

}
```

## 7.5.1.   ECAT_McAxisMove_CiA402_PT

**Description:**

Start the single-axis motion in CiA402 profile torque mode.

Note: This function contains SDO commands

**Syntax:**

int32_t ECAT_McAxisMove_CiA402_PT(uint16_t DeviceNo, uint16_t AxisNo,    double Torque, double Slope)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| AxisNo | uint16_t | IN | Axis number |
| Torque | double | IN | Target torque, 0.1% of the maximum rated torque, which setting range is 1~1000. |
| Slope | double | IN | Torque slope (0.1%/s) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t AxisNo = 0;
uint32_t State;
double torque, double slope;



ret = ECAT_McGetAxisState(DeviceNo, AxisNo, &State);
if(State == MC_AS_STANDSTILL) //StandStill
{
    //Command 1
    torque = 50;
    slope = 10;

    ret = ECAT_McAxisMove_CiA402_PT(DeviceNo, AxisNo, torque, slope);
    if (ret != 0)
        printf("Failed to start move:%d\n", ret);

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

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

}
```

## 7.5.2.   ECAT_McAxisStop

**Description:**

Stop an axis with deceleration.

**Syntax:**

int32_t ECAT_McAxisStop(uint16_t DeviceNo, uint16_t AxisNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_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(1);
        ret = ECAT_McAxisStop(DeviceNo, AxisNo);
        if(ret < 0)
        {
            printf("Failed to stop axis move:%d\n", ret);
            return;
        }
        else
        {
            do
```

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

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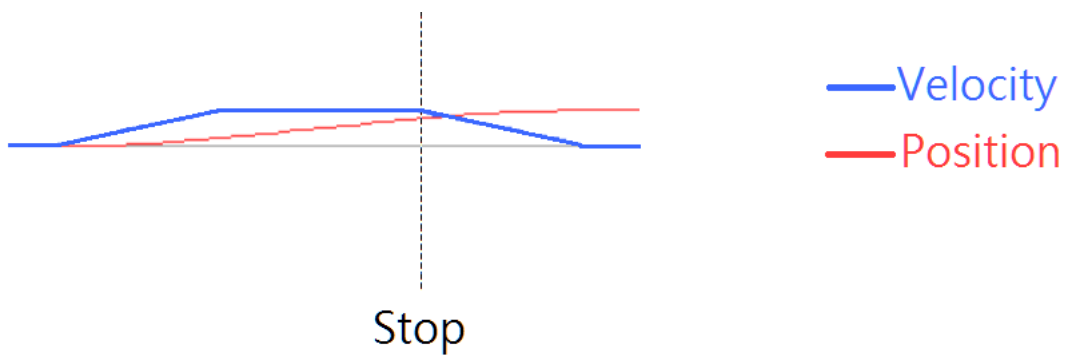


Stop

### 7.5.3. ECAT_McAxisQuickStop

**Description:**

Stop an axis quickly (immediately).

**Syntax:**

int32_t ECAT_McAxisQuickStop(uint16_t DeviceNo, uint16_t AxisNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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(1);
        ret = ECAT_McAxisQuickStop(DeviceNo, AxisNo);
        if(ret < 0)
        {
            printf("Failed to quickstop the axis move:%d\n", ret);
            return;
        }
        else
        {
            do
```

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

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



Quick Stop
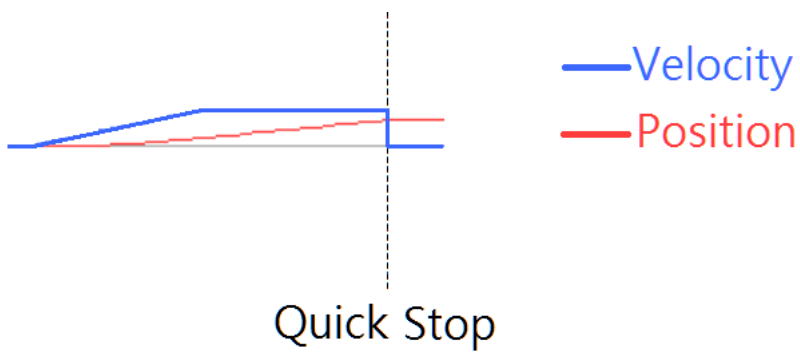
# 7.6. Group Parameter Setting

## 7.6.1. ECAT_McAddAxisToGroup

**Description:**

Add one axis to a group.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number (under 16) (MC_GROUP_NO_MAX macro defines the maximum number) |
| AxisNo | uint16_t | IN | Axis number |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

## 7.6.2. ECAT_McRemoveAxisFromGroup

**Description:**

Remove one axis from a group.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

## 7.6.3.　ECAT_McUngroupAllAxes

**Description:**

Remove all axes from a group. This group no longer owns any axis.

**Syntax:**

int32_t ECAT_McUngroupAllAxes(uint16_t　DeviceNo, uint16_t　GroupNo)

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.6.4.   ECAT_McSetGroupCmdMode

**Description:**

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

**Aborting**: A new command will abort the current executing command; then the new command executes immediately. However, the motion kernel still provides a smooth velocity transition for this mode.

**Buffered**: A new command will be pushed into the group command buffer and wait for being executed. The motion kernel program will execute all commands in this command buffer sequentially. Each command is executed until finished, then another one is loaded from the buffer for next execution by the motion kernel.

**Blending**: A new command will be pushed into a command buffer and wait for being executed. The motion kernel program will execute all commands in the buffer sequentially. While a command is executing, at the beginning of deceleration the motion kernel will load next command from the buffer and executed both commands at the same time. Therefore, the previous motion is partially blending into next one. In this way, a smooth velocity transition is provided.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| CmdMode | uint16_t | IN | Group command mode (As show in Table 7.10) default: BUFFERED Mode |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.10: Group command mode

| Macro Definition | Value | Description |
|---|---|---|
| MS_GRP_CM_ABORTING | 0 | Aborting |
| MS_GRP_CM_BUFFERED | 1 | Buffered |
| MS_GRP_CM_BLENDING | 2 | Blending |

## Example:

### [C/C++]

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

## 7.6.5.  ECAT_McSetGroupCmdModeEx

**Description:**

This function is a little different from *ECAT_McSetGroupCmdMode* at the timing for setting group command mode. It will be pushed into the command buffer first and wait for executing if the current command mode in Buffered mode or Blending mode. However, in Aborting mode it will change command mode immediately. A group command mode decides how a motion command is processed by motion kernel. There are three command modes: Aborting, Buffered, and Blending.

**Aborting**: A new command will abort the current executing command; then the new command executes immediately. However, the motion kernel still provides a smooth velocity transition for this mode.

**Buffered**: A new command will be pushed into the group command buffer and wait for being executed. The motion kernel program will execute all commands in this command buffer sequentially. Each command is executed until finished, then another one is loaded from the buffer for next execution by the motion kernel.

**Blending**: A new command will be pushed into a command buffer and wait for being executed. The motion kernel program will execute all commands in the buffer sequentially. While a command is executing, at the beginning of deceleration the motion kernel will load next command from the buffer and executed both commands at the same time. Therefore, the previous motion is partially blending into next one. In this way, a smooth velocity transition is provided.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| CmdMode | uint16_t | IN | Group command mode |

| | | | (As show in Table 7.10) |
|---|---|---|---|
| | | | |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.11: Group command mode

| Macro Definition | Value | Description |
|---|---|---|
| MS_GRP_CM_ABORTING | 0 | Aborting |
| MS_GRP_CM_BUFFERED | 1 | Buffered |
| MS_GRP_CM_BLENDING | 2 | Blending |

## Example:

**[C/C++]**

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

## 7.6.6.    ECAT_McGetGroupCmdMode

**Description:**

Get the group command mode of a group.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| CmdMode | uint16_t* | OUT | Group command mode (As show in Table 7.10) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.6.7.　ECAT_McSetGroupAccTime

**Description:**

Set the acceleration time of a group.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Time_ms | uint16_t | IN | Acceleration time (Unit: millisecond) default:100 |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

## 7.6.8.　ECAT_McSetGroupAccTimeEx

**Description:**

　　Set the acceleration time of a group. This command will be pushed into command buffer and wait for execution if group command mode is in Buffered mode or Blending mode. The motion kernel will wait the previous motion to be finished and then set the acceleration time.

**Syntax:**

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

**Parameters:**

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

**Return:**

　　0: Success.

　　Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.6.9. ECAT_McGetGroupAccTime

**Description:**

Get the acceleration time of a group.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.6.10. ECAT_McSetGroupAccDecType

**Description:**

Set the acceleration type of a group. There are two acceleration types: T-Curve (linear acceleration) and S-Curve.

Note: The T-curve acceleration time is set by function *ECAT_McSetGroupAccTime.* However, the S-curve acceleration time is twice the acceleration time set by that function.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Type | uint16_t | IN | Acceleration Type<br>1: T-Curve (linear acceleration)(default)<br>2: S-Curve |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

## 7.6.11. ECAT_McGetGroupAccDecType

**Description:**

Get the acceleration type of a group.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

## 7.6.12. ECAT_McSetGroupBlendingPercent

**Description:**

Set the blending percent of a group. In the Blending mode, a "100" blending percent means to blend the next motion command from the starting of deceleration of the pervious motion command. A "0" blending percent means no blending part; and the behavior is similar to the Buffered command mode. Blending will introduce a smooth transition from one command to another; however, it will produce corner error.

**Syntax:**

int32_t ECAT_McSetGroupBlendingPercent(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Value)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Value | uint16_t | IN | Percent<br>range: 0 ~ 100<br>default: 100 |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Value = 50;
ret = ECAT_McSetGroupBlendingPercent(DeviceNo, GroupNo, Value);
if(ret < 0)
{
    printf("Failed to set group blending percent:%d\n", ret);
}
else
{
    printf("Set group blending percent successfully!\n");
}
```

0%
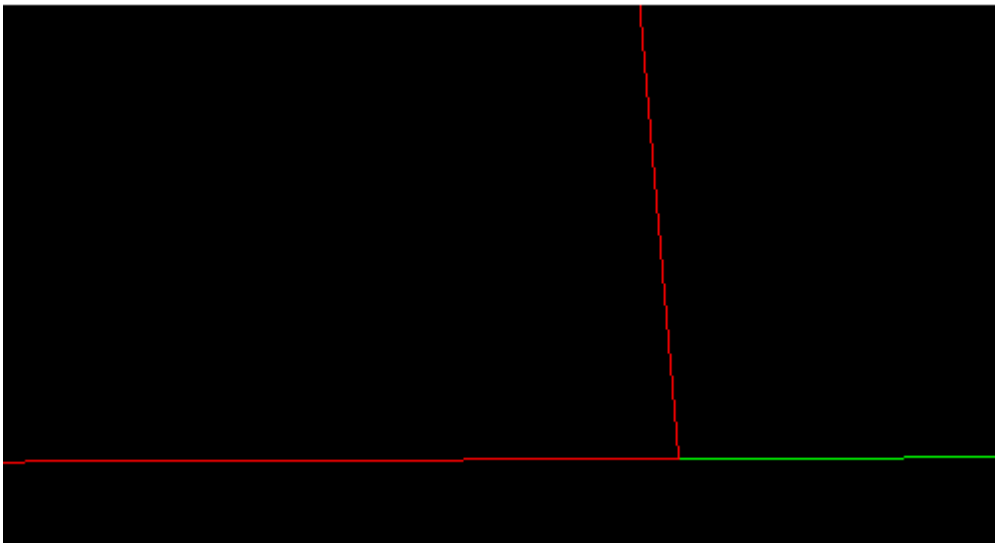
50%
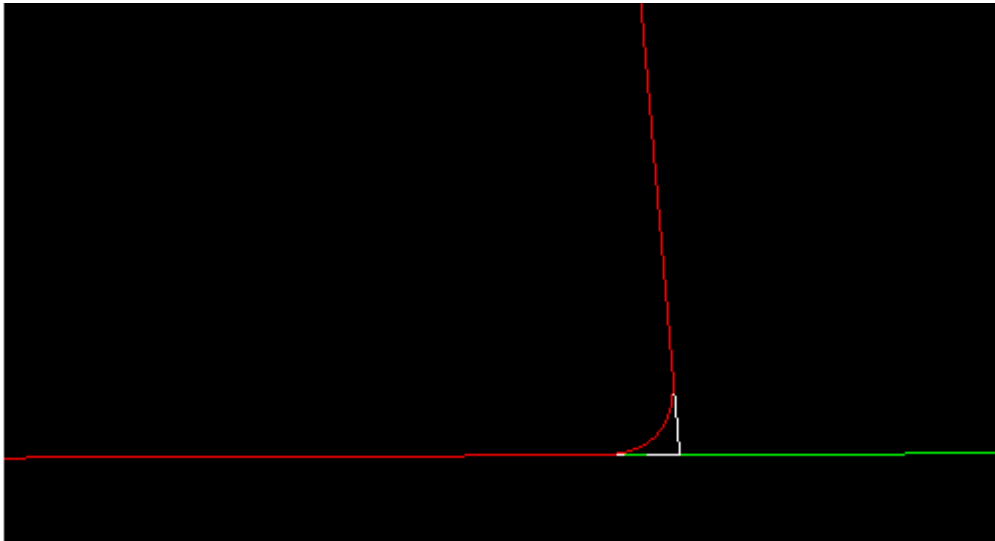


100%
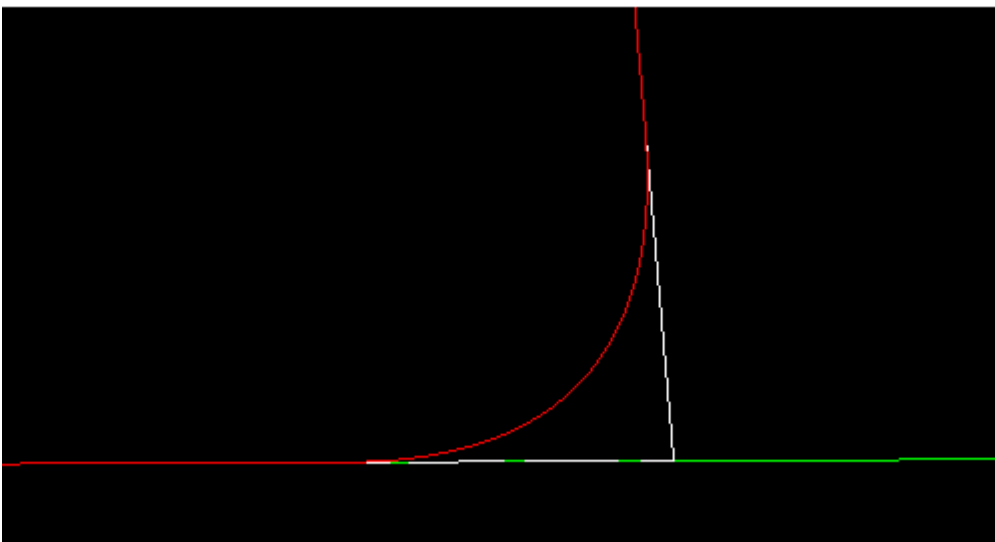
## 7.6.13. ECAT_McSetGroupBlendingPercentEx

**Description:**

Set the blending percent of a group. It is different from *ECAT_McSetGroupBlendingPercent* at the executing time. This command will be pushed into command buffer first in Buffered mode or Blending mode and wait for execution.

**Syntax:**

int32_t ECAT_McSetGroupBlendingPercentEx(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Value)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Value | uint16_t | IN | Percent<br>range: 0 ~ 100<br>default: 100 |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint16_t Time_ms = 999;
double Pos[MC_AXIS_NO_MAX]={ 0};
double Vel = 5;


// Add Axis
AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
    printf("Failed to add axis to group:%d\n", ret);
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
    printf("Failed to add axis to group:%d\n", ret);
// Set Acctime
ret = ECAT_McSetGroupAccTime(DeviceNo, GroupNo, Time_ms);
if(ret < 0)
    printf("Failed to set group acceleration time:%d\n", ret);
// Set blending mode
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, MS_GRP_CM_BLENDING);
if(ret < 0)
    printf("Failed to set group command mode:%d\n", ret);



// Start
ret = ECAT_McSetGroupBlendingPercentEx(DeviceNo,GrpNo, 80);
if(ret < 0)      printf("Failed to set group blending percent:%d\n", ret);
Pos[0] = 2;
Pos[1] = 0;
ret = ECAT_McGroupMoveLineAbs(DeviceNo,GrpNo,Pos,Vel);
if(ret < 0)      printf("Failed to group move abs%d\n", ret);
```

```
ret = ECAT_McSetGroupBlendingPercentEx(DeviceNo,GrpNo, 60);
if(ret < 0)        printf("Failed to set group blending percent:%d\n", ret);
Pos[0] = 2;
Pos[1] = 2;
ret = ECAT_McGroupMoveLineAbs(DeviceNo, GrpNo, Pos, Vel);
if(ret < 0)        printf("Failed to group move absolutely %d\n", ret);



ret = ECAT_McSetGroupBlendingPercentEx(DeviceNo,GrpNo, 40);
if(ret < 0)        printf("Failed to set group blending percent:%d\n", ret);
Pos[0] = 0;
Pos[1] = 2;
ret = ECAT_McGroupMoveLineAbs(DeviceNo, GrpNo, Pos, Vel);
if(ret < 0)        printf("Failed to group move absolutely %d\n", ret);


ret = ECAT_McSetGroupBlendingPercentEx(DeviceNo, GrpNo, 20);
if(ret < 0)        printf("Failed to set group blending percent: %d\n", ret);
Pos[0] = 0;
Pos[1] = 0;
ret = ECAT_McGroupMoveLineAbs(DeviceNo, GrpNo, Pos, Vel);
if(ret < 0)        printf("Failed to group move absolutely %d\n", ret);


ret = ECAT_McSetGroupBlendingPercentEx(DeviceNo, GrpNo, 0);
if(ret < 0)        printf("Failed to set group blending percent:%d\n", ret);
Pos[0] = -2;
Pos[1] = 0;
ret = ECAT_McGroupMoveLineAbs(DeviceNo,GrpNo,Pos,Vel);
if(ret < 0)        printf("Failed to group move absolutely %d\n", ret);


Pos[0] = -2;
Pos[1] = 2;
ret = ECAT_McGroupMoveLineAbs(DeviceNo,GrpNo,Pos,Vel);
if(ret < 0)        printf("Failed to group move absolutely %d\n", ret);
```
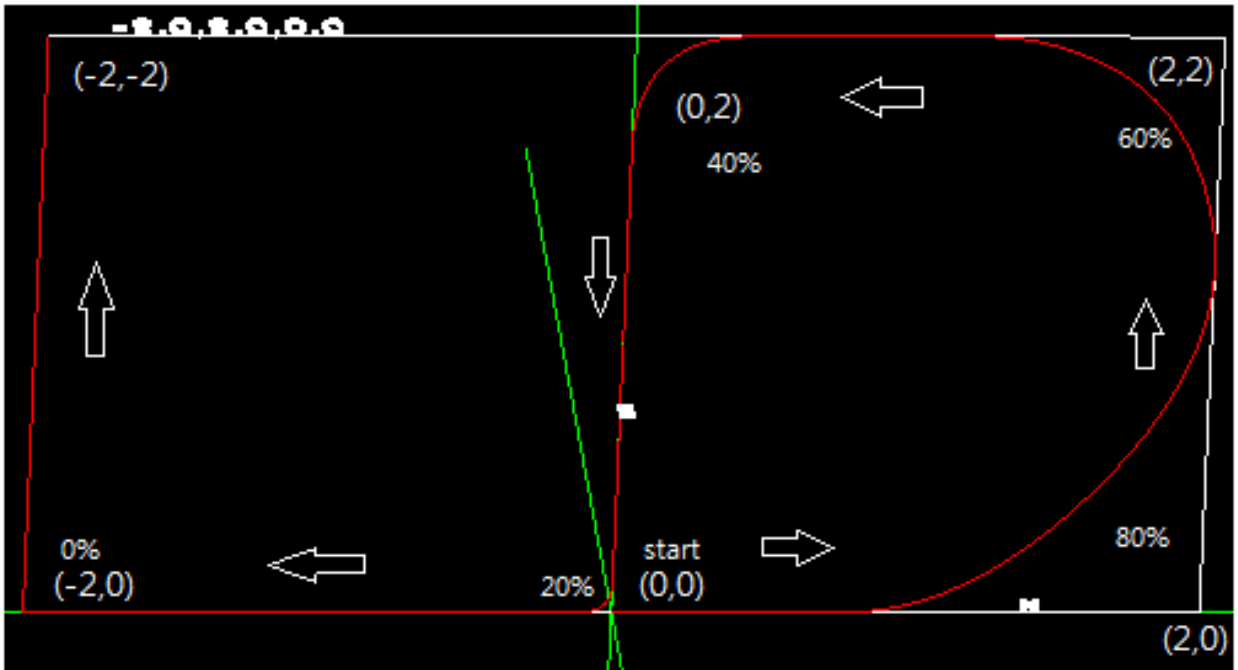
## 7.6.14. ECAT_McSetGroupPvtDecEnable

**Description:**

Set whether to decelerate or not after the PVT motion is finished of a group.

**Syntax:**

int32_t ECAT_McSetGroupPvtDecEnable(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Enable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Enable | uint16_t | IN | 0: no deceleration (default) <br> 1: deceleration |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Enable= 1;
ret = ECAT_McSetGroupPvtDecEnable(DeviceNo, GroupNo, Enable);
if(ret < 0)
{
    printf("Failed to set group PvtDecEnable:%d\n", ret);
}
else
{
    printf("Set group PvtDecEnable successfully!\n");
}
```
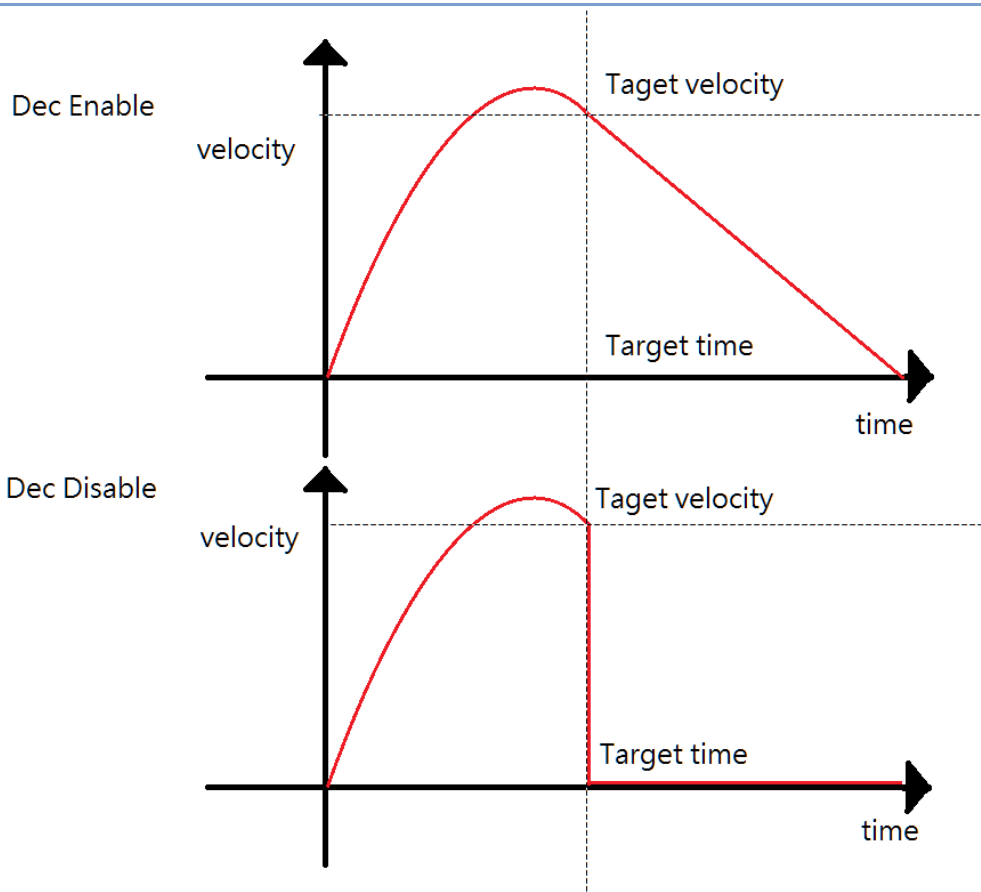
## 7.6.15. ECAT_McGetGroupPvtDecEnable

**Description:**

Get whether to decelerate or not after the PVT motion is finished of a group.

**Syntax:**

int32_t    ECAT_McGetGroupPvtDecEnable(uint16_t    DeviceNo,    uint16_t    GroupNo, uint16_t *Enable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Enable | uint16_t* | OUT | 0: no deceleration<br>1: deceleration |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.6.16. ECAT_McSetGroupCoordinate

Description:

Set whether the group motion control performs coordinate conversion.

When performing coordinate conversion, the functions of Table 7.9 cannot be used. To use it, set the coordinate conversion mode to MC_DEFAULT_INTERPOLATION.

Table7.12

| Name | Type |
|------|------|
| ECAT_McSetAxisServoOn | Axis Parameter Settings |
| ECAT_McSetAxisPPU | Axis Parameter Settings |
| ECAT_McAxisHome | Axis Homing |
| 錯誤！找不到參照來源。 | Axis Homing |
| 7.5Axis Moving | Axis Moving |
| ECAT_McAddAxisToGroup | Group Parameter Setting |
| 錯誤！找不到參照來源。 | Group Parameter Setting |
| 錯誤！找不到參照來源。 | Group Parameter Setting |
| 錯誤！找不到參照來源。 | Group Moving |

Syntax:

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

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Type | uint16_t | IN | Table7.13 default: MC_DEFAULT_INTERPOLATION |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Table7.13

| Name | Value | Description |
|---|---|---|
| MC_DEFAULT_INTERPOLATION | 0 | No coordinate conversion |
| MC_POLAR_INTERPOLATION | 1 | The command is a rectangular coordinate and the output is a polar coordinate<br>note:<br>➢ In this mode, the mechanism should be a linear axis and a rotating axis.<br>➢ Avoid linear axis length less than zero.<br>➢ The unit of the rotary axis needs to be set to rad, that is, the PPU needs to be set to the number of pulses required to rotate 1rad, and the PPU of the linear axis can be set by the user.<br>➢ In this mode, the command position is the rectangular coordinate mode, and the unit is the same as the linear axis.<br>➢ The linear axis needs to be added to the group first, then add the rotation axis to the group, please see the following example. |

| Example: |
|---|
| [C/C++] |

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Type;
uint16_t AxisNo;
double PPU;

//add linear axis first
```

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

ret = ECAT_McSetGroupCoordinate(DeviceNo, GroupNo, Type);
if(ret < 0)
{
    printf("Failed to set group Coordinate:%d\n",ret);
}
```

## 7.6.17.    ECAT_McSetGroupCoordinateLimit

**Description:**

Set the position software limit of a group.

**Syntax:**

int32_t   ECAT_McSetGroupCoordinateLimit(uint16_t   DeviceNo,   uint16_t   GroupNo, uint16_t Type, uint16_t Enable, double MIN_Value, double MAX_Value)


**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Type | uint16_t | IN | Table7.13 |
| Enable | uint16_t | IN | Enable/Disable position software limit default: disable 0: disable 1: enable |
| MIN_Value | double | IN | minimum value |
| MAX_Value | double | IN | maximum value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Type = MC_POLAR_INTERPOLATION;
uint16_t Enable = 1;
double Min = 0;
double Max = 10;
ret = ECAT_McSetGroupCoordinateLimit(DeviceNo, GroupNo, Type, Enable, Min, Max);
if(ret < 0)
{
    printf("Failed to set group coordinate limit:%d\n",ret);
}
```

## 7.6.18. ECAT_McGetGroupCoordinateLimit

**Description:**

Get the position software limit of a group.

**Syntax:**

int32_t ECAT_McGetGroupCoordinateLimit(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Type, uint16_t *Enable, double *MIN_Value, double *MAX_Value)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Type | uint16_t | IN | Table7.13 |
| Enable | uint16_t* | OUT | Enable/Disable position software limit<br>0: disable<br>1: enable |
| MIN_Value | double* | OUT | minimum value |
| MAX_Value | double* | OUT | maximum value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t Type = MC_POLAR_INTERPOLATION;
uint16_t Enable;
double Min;
double Max;
ret = ECAT_McGetGroupCoordinateLimit(DeviceNo, GroupNo, Type, &Enable, &Min, &Max);
if(ret < 0)
{
    printf("Failed to get group coordinate limit:%d\n",ret);
}
else
{
    printf("Enable: %d, min: %f, max: %f\n", Enable, Min, Max);
}
```

# 7.7. Group Status

## 7.7.1. ECAT_McGetGroupState

**Description:**

Get the state of a group.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| State | uint32_t* | OUT | Group state (As show in Table 7.14) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.14: Group state

| Macro Definition | Value | Description |
|------------------|-------|-------------|
| MC_GS_DISABLED | 0 | Group is disabled |
| MC_GS_STANDBY | 1 | Group is standby |
| 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++]

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

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

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

## 7.7.2.  ECAT_McGetGroupCmdBuffer

**Description:**

Get the number of commands buffered inside a group buffer.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Buffer | uint16_t* | OUT | Number of commands in the group command buffer |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

## 7.7.3.    ECAT_McSetGroupVelLimitStatus

**Description:**

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

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

## 7.7.4.  ECAT_McGetGroupVelLimitStatus

**Description:**

Get the setting of enabling or disabling the velocity limit of a group.

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

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Status | uint16_t* | OUT | Setting of velocity limit checking of a group<br>0: disable<br>1: enable |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.7.5. ECAT_McSetGroupVelLimitValue

**Description:**

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

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

## 7.7.6.  ECAT_McGetGroupVelLimitValue

**Description:**

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

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

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

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

# 7.8.  Group Moving

## 7.8.1.  ECAT_McGroupMoveLineAbs

**Description:**

Start an absolute linear interpolation motion of a group. An array of position data of axes and a velocity are requested to enter.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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


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

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

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

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

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

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

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

    }while(State == MC_GS_MOVING) //Moving

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

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

## 7.8.2. ECAT_McGroupMoveLineRel

**Description:**

Start a relative linear interpolation motion of a group. An array of distance data of axes and a velocity are requested to enter.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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


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

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

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

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

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

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

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

    }while(State == MC_GS_MOVING) //Moving

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

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

### 7.8.3.　ECAT_McGroupMoveLineAbs_PT

**Description:**

Start an absolute linear interpolation motion of a group. An array of position data of axes and action time are requested to enter. The command speed of each axis is calculated according to the position data and the time value.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_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
{
```

```c
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupTime = 5;
    ret = ECAT_McGroupMoveLineAbs_PT (DeviceNo, GroupNo, GroupPos, GroupTime);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }
    //Command 2
    GroupPos[0] = 30.0;
    GroupPos[1] = 50.0;
    GroupTime = 10;
    ret = ECAT_McGroupMoveLineAbs_PT(DeviceNo, GroupNo, GroupPos, GroupTime);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }

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

    }while(State == MC_GS_MOVING) //Moving

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

## 7.8.4.   ECAT_McGroupMoveLineRel_PT

Description:

Start a relative linear interpolation motion of a group. An array of distance data of axes and action time are requested to enter. The command speed of each axis is calculated according to the distance data and the time value.

Syntax:

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

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Pos | double[] | IN | A distance array of a group<br> Each array element is the relative<br> position of a corresponding axis.<br>(Unit: user unit) |
| Time | double | IN | Time (Unit: second) |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

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

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

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
```

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

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

}while(State == MC_GS_MOVING) //Moving

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

## 7.8.5.  ECAT_McGroupMoveLineAbs_PVT

**Description:**

Start an absolute PVT motion.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Pos | double[] | IN | A position data array of a group<br>Each array element is the absolute position of a corresponding axis.<br>(Unit: user unit) |
| Vel | double[] | IN | A velocity data array of a group<br>Each array element is the velocity data of a corresponding axis.<br>(Unit: user unit/s) |
| Time | double | IN | Time (Unit: second) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double AxisPos[MC_AXIS_NO_MAX];
double AxisVel[MC_AXIS_NO_MAX];
double Time;

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

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
```

```
AxisPos[0] = 0.0;
AxisPos[1] = 0.0;
AxisVel[0] = 0.0;
AxisVel[1] = 0.0;
Time = 0;
ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 2
AxisPos[0] = 5.0;
AxisPos[1] = 0.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 1.5;
ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 3
AxisPos[0] = 0.0;
AxisPos[1] = 10.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 3.0;
ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 4
AxisPos[0] = 5.0;
AxisPos[1] = 10.0;
```

```
    AxisVel[0] = 0.0;

    AxisVel[1] = 0.0;

    Time = 4.5;

    ret = ECAT_McGroupMoveLineAbs_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);

    if(ret < 0)

    {

        printf("Failed to add group move line command:%d\n", ret);

        return;

    }

    do

    {

        sleep(1);

        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);


    }while(State == MC_GS_MOVING) //Moving


    if(State == MC_GS_STANDBY) //Standby

        printf("Group move line successfully!\n");

    else if(State == MC_GS_ERRORSTOP) //ErrorStop

    {

        printf("Group error stop\n");

    }

}
```

## 7.8.6.　ECAT_McGroupMoveLineRel_PVT

**Description:**

Start a relative PVT motion of a group.

**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Pos | double[] | IN | A position data array of a group<br> Each array element is the relative<br> displacement of a corresponding axis.<br>(Unit: user unit) |
| Vel | double[] | IN | A velocity data array of a group<br> Each array element is the velocity data<br> of a corresponding axis.<br>(Unit: user unit/s) |
| Time | double | IN | Time (Unit: second) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double AxisPos[MC_AXIS_NO_MAX];
double AxisVel[MC_AXIS_NO_MAX];
double Time;

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

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
```

```
AxisPos[0] = 0.0;
AxisPos[1] = 0.0;
AxisVel[0] = 0.0;
AxisVel[1] = 0.0;
Time = 0;
ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 2
AxisPos[0] = 5.0;
AxisPos[1] = 0.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 1.5;
ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 3
AxisPos[0] = -5.0;
AxisPos[1] = 10.0;
AxisVel[0] = 20.0;
AxisVel[1] = 0.0;
Time = 3.0;
ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
    return;
}
//Command 4
AxisPos[0] = 5.0;
AxisPos[1] = 0.0;
```

```
        AxisVel[0] = 0.0;
        AxisVel[1] = 0.0;
        Time = 4.5;
        ret = ECAT_McGroupMoveLineRel_PVT(DeviceNo, GroupNo, AxisPos, AxisVel, Time);
        if(ret < 0)
        {
            printf("Failed to add group move line command:%d\n", ret);
            return;
        }
        do
        {
            sleep(1);
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

        }while(State == MC_GS_MOVING) //Moving

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

## 7.8.7. ECAT_McGroupMoveLineAbs_P2P

**Description:**

Start an absolute position motion of each axis in a group.

Note: Use Maximum velocity of each axis (*ECAT_McSetAxisMaxVelocity*)

**Syntax:**

int32_t ECAT_McGroupMoveLineAbs_P2P(uint16_t DeviceNo, uint16_t GroupNo, double Pos[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Pos | double[] | IN | Position array of a group<br>Each array element is the absolute position of an axis.<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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 MaxVelocity = 100;

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

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

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


ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    ret = ECAT_McGroupMoveLineAbs_P2P(DeviceNo, GroupNo, GroupPos);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }

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

    }while(State == MC_GS_MOVING) //Moving

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

## 7.8.8. ECAT_McGroupMoveLineRel_P2P

**Description:**

Start a relative position motion of each axis in a group.

Note: Use Maximum velocity of each axis (*ECAT_McSetAxisMaxVelocity*)

**Syntax:**

int32_t ECAT_McGroupMoveLineRel_P2P(uint16_t DeviceNo, uint16_t GroupNo, double Pos[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Pos | double[] | IN | Distance array of a group<br>Each array element is the relative position of an axis.<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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 MaxVelocity = 100;

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

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

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


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

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

    }while(State == MC_GS_MOVING) //Moving

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

## 7.8.9. ECAT_McGroupMoveLineAbs_PTexT

**Description:**

This function is for factory use only

Start an absolute linear interpolation motion of a group. An array of position data of axes and action time are requested to enter. The command speed of each axis is calculated according to the position data and the time value.

Use command speed to extend action time by extension time.

Ex: Current position of axis is 0, command position is 1, Time is 1, extension Time = 0.5

command speed = command position / Time => 1/1 = 1

totoal action time = Time + extension Time = 1 + 0.5 = 1.5

Actual moving distance = command speed * totoal action time = 1*1.5 = 1.5
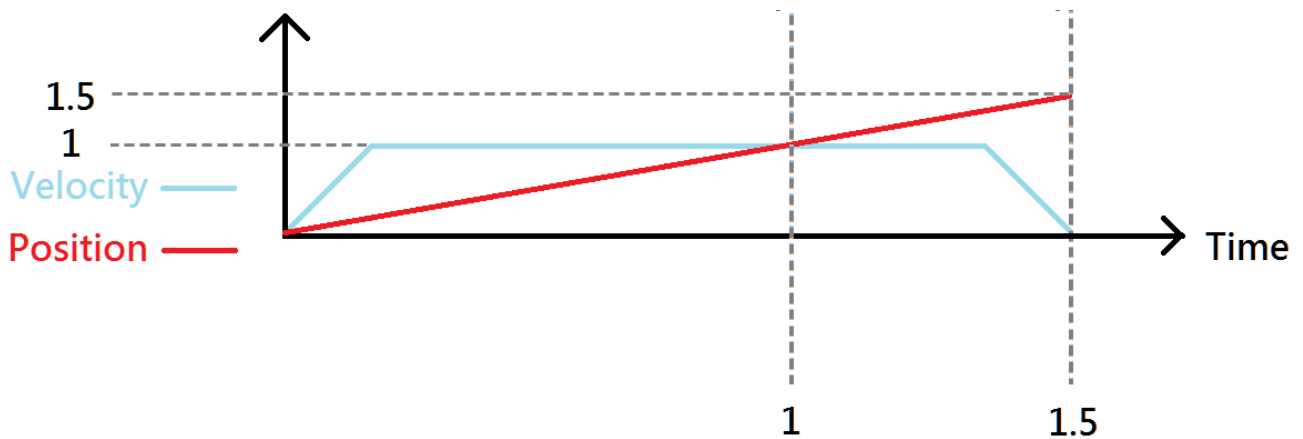
**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Pos | double[] | IN | A position array for a group Each array element is the absolute position of a corresponding axis. (Unit: user unit) |
| Time | double | IN | action Time (Unit: second) |
| exTime | double | IN | extension Time (Unit: second) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

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

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

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
```

```
//Command 1
GroupPos[0] = 1;
GroupPos[1] = 0;
GroupTime = 1;
extendTime = 0.5;

ret = ECAT_McGroupMoveLineAbs_PTexT(DeviceNo, GroupNo, GroupPos, GroupTime, extendTime);
if(ret < 0)
{
    printf("Failed to add group move line command:%d\n", ret);
}
do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

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

## 7.8.10. ECAT_McGroupMoveLineRel_PTexT

**Description:**

This function is for factory use only

Start a relative linear interpolation motion of a group. An array of distance data of axes and action time are requested to enter. The command speed of each axis is calculated according to the distance data and the time value.

Use command speed to extend action time by extension time.

Ex: Current position of axis is 0, command position is 1, Time is 1, extension Time = 0.5

command speed = command position / Time => 1/1 = 1

totoal action time = Time + extension Time = 1 + 0.5 = 1.5

Actual moving distance = command speed * totoal action time = 1*1.5 = 1.5
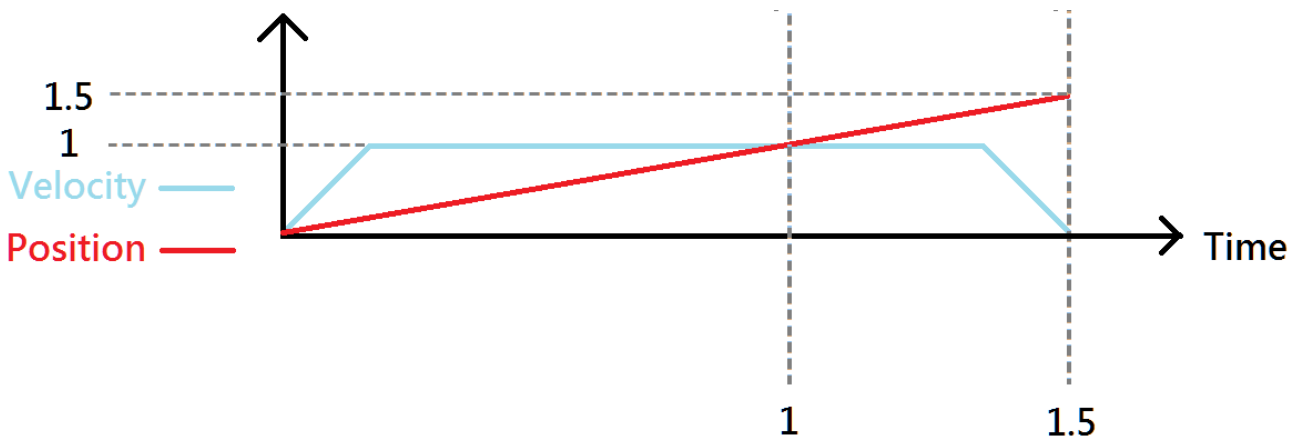
**Syntax:**

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

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Pos | double[] | IN | A distance array of a group<br> Each array element is the relative<br> position of a corresponding axis.<br>(Unit: user unit) |
| Time | double | IN | Time (Unit: second) |
| exTime | double | IN | extension Time (Unit: second) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_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;
double extendTime;

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

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

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

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
```

```
{
    //Command 1
    GroupPos[0] = 1;
    GroupPos[1] = 0;
    GroupTime = 1;
    extendTime = 0.5;

    ret = ECAT_McGroupMoveLineRel_PTexT(DeviceNo, GroupNo, GroupPos, GroupTime, extendTime);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }

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

    }while(State == MC_GS_MOVING) //Moving

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

## 7.8.11. ECAT_McGroupMoveLineAbs_PPT

**Description:**

Start an absolute curve interpolation motion of a group. An array of position data of axes and action time are requested to enter.

This command is a cubic smooth curve and will pass through the target point.

**Syntax:**

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

**Parameters:**

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

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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
{
```

```c
//Command 1
GroupPos[0] = 10.0;
GroupPos[1] = 20.0;
GroupTime = 5;
ret = ECAT_McGroupMoveLineAbs_PPT (DeviceNo, GroupNo, GroupPos, GroupTime);
if(ret < 0)
{
    printf("Failed to add group move curve command:%d\n", ret);
}
//Command 2
GroupPos[0] = 30.0;
GroupPos[1] = 50.0;
GroupTime = 10;
ret = ECAT_McGroupMoveLineAbs_PPT(DeviceNo, GroupNo, GroupPos, GroupTime);
if(ret < 0)
{
    printf("Failed to add group move curve command:%d\n", ret);
}

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

}while(State == MC_GS_MOVING) //Moving

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

## 7.8.12. ECAT_McGroupMoveLineRel_PPT

Description:

Start a relative linear interpolation motion of a group. An array of distance data of axes and action time are requested to enter.

This command is a cubic smooth curve and will pass through the target point.

Syntax:

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

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Pos | double[] | IN | A distance array of a group  Each array element is the relative  position of a corresponding axis.  (Unit: user unit) |
| Time | double | IN | Time (Unit: second) |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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
{
```

```c
//Command 1
GroupPos[0] = 10.0;
GroupPos[1] = 20.0;
GroupTime = 5;
ret = ECAT_McGroupMoveLineRel_PPT(DeviceNo, GroupNo, GroupPos,   GroupTime);
if(ret < 0)
{
    printf("Failed to add group move curve command:%d\n", ret);
}
//Command 2
GroupPos[0] = 30.0;
GroupPos[1] = 50.0;
GroupTime = 10;
ret = ECAT_McGroupMoveLineRel_PPT(DeviceNo,GroupNo,GroupPos, GroupTime);
if(ret < 0)
{
    printf("Failed to add group move curve command:%d\n", ret);
}

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

}while(State == MC_GS_MOVING) //Moving

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

## 7.8.13.  ECAT_McGroupMoveCircularAbs_CP_Angle

Description:

Start an absolute 2D circular interpolation motion by providing the center position and its angle.

Syntax:

int32_t ECAT_McGroupMoveCircularAbs_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double AuxPos[])

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| Angle | double | IN | Angle from start point to end point<br>(Unit: degree) |
| AuxPos | double[] | IN | Absolute position data of the center point<br>(Unit: user unit) |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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


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


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


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


ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```

```
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);

    //Command 2
    CircAuxPos [0] = 10.0; //Center Position
    CircAuxPos [1] = 0.0; //Center Position
    CircAngle = 135;
    GroupVel = 5;
    ret = ECAT_McGroupMoveCircularAbs_CP_Angle(DeviceNo, GroupNo, GroupVel, CircAngle
                            , CircAuxPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n", ret);
    }

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

    }while(State == MC_GS_MOVING) //Moving

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

**2D circular interpolation motion path of example:**

## 7.8.14.  ECAT_McGroupMoveCircularRel_CP_Angle

**Description:**

Start a relative 2D circular interpolation motion by providing the center position and its angle.

**Syntax:**

int32_t ECAT_McGroupMoveCircularRel_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double AuxPos[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| Angle | double | IN | Angle from start point to end point<br>(Unit: degree) |
| AuxPos | double[] | IN | Relative distance data from the center point to the start point<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

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

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

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```

```
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 10.0; //Center Position
    CircAuxPos [1] = 0.0; //Center Position
    CircAngle = 135;
    GroupVel = 5;
    ret = ECAT_McGroupMoveCircularRel_CP_Angle(DeviceNo, GroupNo, GroupVel, CircAngle
                        , CircAuxPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n", ret);
    }

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

    }while(State == MC_GS_MOVING) //Moving

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

## 7.8.15. ECAT_McGroupMoveCircularAbs_CP_EP

**Description:**

Start an absolute 2D circular interpolation motion by providing the center position and the end position.

**Syntax:**

int32_t ECAT_McGroupMoveCircularAbs_CP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| Dir | uint8_t | IN | Direction<br>0: CW<br>1: CCW |
| AuxPos | double[] | IN | Absolute position data of center point<br>(Unit: user unit) |
| EndPos | double[] | IN | Absolute position data of end point<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

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

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

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

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);

    //Command 2
    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircEndPos[0] = 17.071 // End Position
    CircEndPos[1] = 7.071 // End Position

    GroupVel = 5;
    CircDir = 0; //CW
    ret = ECAT_McGroupMoveCircularAbs_CP_EP(DeviceNo, GroupNo, GroupVel, CircDir
            , CircAuxPos, CircEndPos);
    if(ret < 0)
        printf("Failed to add group move circular command:%d\n", ret);


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

    }while(State == MC_GS_MOVING) //Moving

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

```
    }
}
```

## 2D circular interpolation motion path of example:

## 7.8.16.  ECAT_McGroupMoveCircularRel_CP_EP

**Description:**

Start a relative 2D circular interpolation motion by providing the center position and the end position.

**Syntax:**

int32_t ECAT_McGroupMoveCircularRel_CP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity (Unit: user unit/s) |
| Dir | uint8_t | IN | Direction 0: CW 1: CCW |
| AuxPos | double[] | IN | Relative distance data from the center point to the start point (Unit: user unit) |
| EndPos | double[] | IN | Relative distance data from the end point to the start point (Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

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

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

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

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{

    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircEndPos[0] = 17.071 // End Position
    CircEndPos[1] = 7.071 // End Position

    GroupVel = 5;
    CircDir = 0; //CW
    ret = ECAT_McGroupMoveCircularRel_CP_EP(DeviceNo, GroupNo, GroupVel, CircDir
            , CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n", ret);
    }

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

    }while(State == MC_GS_MOVING) //Moving

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

## 7.8.17. ECAT_McGroupMoveCircularAbs_BP_EP

**Description:**

Start an absolute 2D circular interpolation motion by providing a border position and the end position.

**Syntax:**

int32_t ECAT_McGroupMoveCircularAbs_BP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| Dir | uint8_t | IN | Direction<br>0: CW<br>1: CCW |
| AuxPos | double[] | IN | Absolute position data of the border point<br>(Unit: user unit) |
| EndPos | double[] | IN | Absolute position data of the end point<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

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

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

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

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);

    //Command 2
    CircAuxPos[0] = 17.071; //Border Position
    CircAuxPos[1] = 7.071; //Border Position
    CircEndPos[0] = 17.071 // End Position
    CircEndPos[1] = -7.071 // End Position

    GroupVel = 5;
    CircDir = 0; //CW
    ret = ECAT_McGroupMoveCircularAbs_BP_EP(DeviceNo, GroupNo, GroupVel, CircDir
            , CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n", ret);
    }

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

    }while(State == MC_GS_MOVING) //Moving

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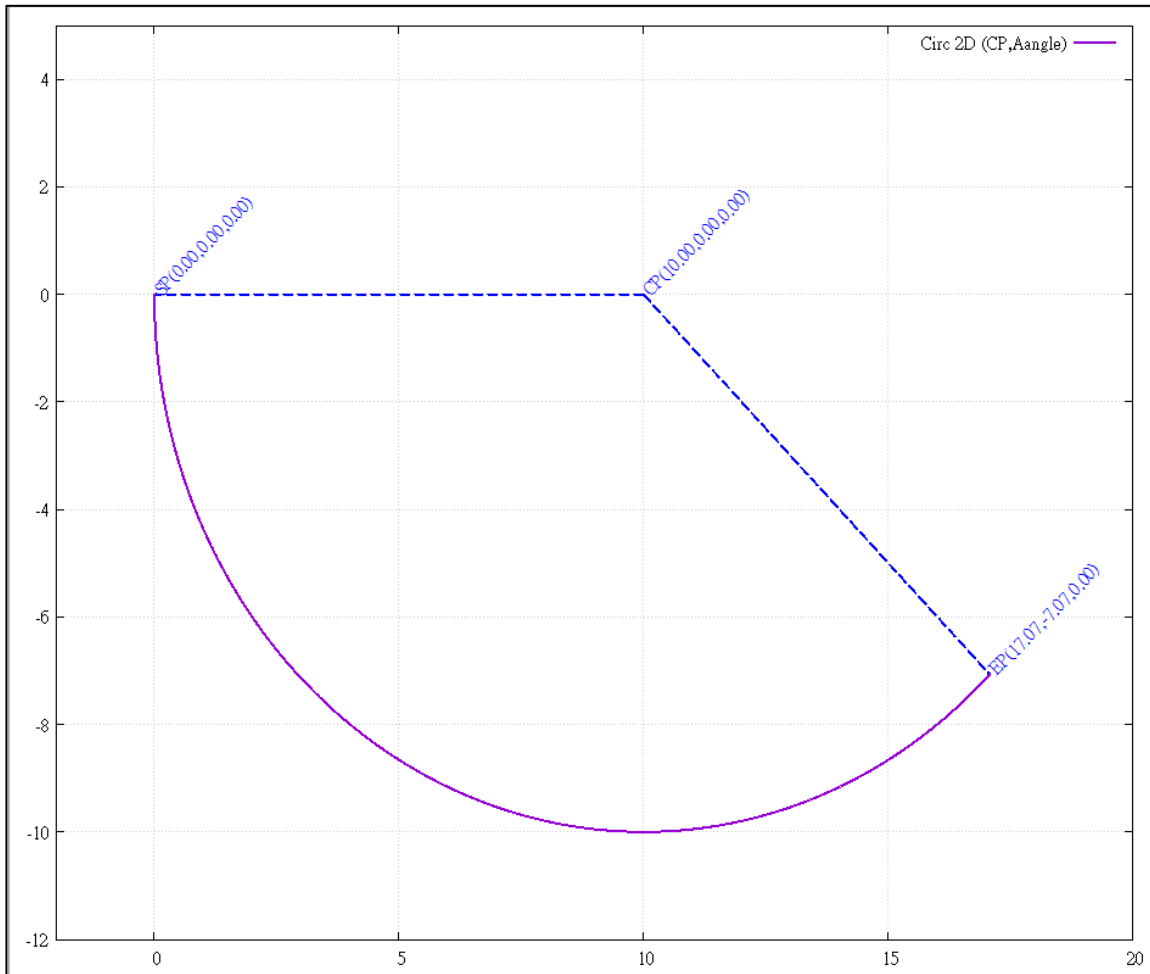
```
        printf("Group error stop\n");

    }

}
```

**2D circular interpolation motion path of example:**

## 7.8.18.  ECAT_McGroupMoveCircularRel_BP_EP

**Description:**

Start a relative 2D circular interpolation motion by providing a border position and the end position.

**Syntax:**

int32_t ECAT_McGroupMoveCircularRel_BP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| Dir | uint8_t | IN | Direction<br>0: CW<br>1: CCW |
| AuxPos | double[] | IN | Relative distance data from the border point to the start point<br>(Unit: user unit) |
| EndPos | double[] | IN | Relative distance data from the end point to the start point<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];

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

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

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

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos[0] = 17.071; //Border Position
    CircAuxPos[1] = 7.071; //Border Position
    CircEndPos[0] = 17.071 // End Position
    CircEndPos[1] = -7.071 // End Position

    GroupVel = 5;
    CircDir = 0; //CW
    ret = ECAT_McGroupMoveCircularRel_BP_EP(DeviceNo, GroupNo, GroupVel, CircDir
            , CircAuxPos, CircEndPos);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n", ret);
    }

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

    }while(State == MC_GS_MOVING) //Moving

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

## 7.8.19.  ECAT_McGroupMove3DCircularAbs_CP_Angle

**Description:**

Start an absolute 3D circular interpolation motion by providing the center position and an angle.

**Syntax:**

int32_t ECAT_McGroupMove3DCircularAbs_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double AuxPos[], double NV[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Tangent velocity of the motion (Unit: user unit/s) |
| Angle | double | IN | Angle between the end point and the start point (right-hand rule) (Unit: degree) |
| AuxPos | double[] | IN | Absolute position data of center point (Unit: user unit) |
| NV | double[] | IN | Normal vector of the circle |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;

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

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

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

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupPos[2] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);

    //Command 2
    CircAuxPos [0] = 3.5355;    //Center Position
    CircAuxPos [1] = 3.5355;    //Center Position
    CircAuxPos [2] = 0.0;            //Center Position

    NV [0] = -0.7071;        //Normal Vector
    NV [1] = 0.7071;    //Normal Vector
    NV [2] = 0.0;            //Normal Vector

    CircAngle = 135;
    GroupVel = 5;
    ret = ECAT_McGroupMove3DCircularAbs_CP_Angle(DeviceNo, GroupNo, GroupVel
            , CircAngle, CircAuxPos, NV);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n", ret);
    }

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

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
```

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

## 3D circular interpolation motion path of example:

## 7.8.20.  ECAT_McGroupMove3DCircularRel_CP_Angle

**Description:**

Start a relative 3D circular interpolation motion by providing the center position and an angle.

**Syntax:**

int32_t ECAT_McGroupMove3DCircularRel_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double AuxPos[], double NV[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity (Unit: user unit/s) |
| Angle | double | IN | Angle between the end point and the start point (right-hand rule) (Unit: degree) |
| AuxPos | double[] | IN | Relative distance data from the center point to the start point (Unit: user unit) |
| NV | double[] | IN | Normal vector of the circle |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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;
}
```

```c
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 3.5355;    //Center Position
    CircAuxPos [1] = 3.5355;    //Center Position
    CircAuxPos [2] = 0.0;           //Center Position

    NV [0] = -0.7071;        //Normal Vector
    NV [1] = 0.7071;    //Normal Vector
    NV [2] = 0.0;            //Normal Vector

    CircAngle = 135;
    GroupVel = 5;
    ret = ECAT_McGroupMove3DCircularRel_CP_Angle(DeviceNo, GroupNo, GroupVel
            , CircAngle, CircAuxPos, NV);
    if(ret < 0)
    {
        printf("Failed to add group move circular command:%d\n", ret);
    }

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

    }while(State == MC_GS_MOVING) //Moving

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

## 7.8.21. ECAT_McGroupMove3DCircularAbs_CP_EP

**Description:**

Start an absolute 3D circular interpolation motion by providing the center position and the end position.

**Syntax:**

int32_t ECAT_McGroupMove3DCircularAbs_CP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

**Parameters:**

| Name | Type | IN or OUT | Description |
| --- | --- | --- | --- |
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| Dir | uint8_t | IN | Direction<br>0: CW<br>1: CCW |
| AuxPos | double[] | IN | Absolute position data of the center point<br>(Unit: user unit) |
| EndPos | double[] | IN | Absolute position data of the end point<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];


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

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

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
```

```
        return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
        //Command 1
        GroupPos[0] = 0.0;
        GroupPos[1] = 0.0;
        GroupPos[2] = 0.0;
        GroupVel = 5;
        ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
        if(ret < 0)
                printf("Failed to add group move line command:%d\n", ret);

        //Command 2
        CircAuxPos [0] = 3.5355;   //Center Position
        CircAuxPos [1] = 3.5355;   //Center Position
        CircAuxPos [2] = 0.0;            //Center Position

        CircEndPos[0] = 3.5355       // End Position
        CircEndPos[1] = 3.5355       // End Position
        CircEndPos[2] = 5.0         // End Position
        CircDir = 0; // CW
        GroupVel = 5;
        ret = ECAT_McGroupMove3DCircularAbs_CP_EP(EcatDeviceID, GroupNo, GroupVel
                , CircDir, CircAuxPos, CircEndPos);
        if(ret < 0)
        {
                printf("Failed to add group move circular command:%d\n", ret);
        }

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

        }while(State == MC_GS_MOVING) //Moving
```

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

## 3D circular interpolation motion path of example:

## 7.8.22. ECAT_McGroupMove3DCircularRel_CP_EP

**Description:**

Start a relative 3D circular interpolation motion by providing the center position and the end position.

**Syntax:**

int32_t ECAT_McGroupMove3DCircularRel_CP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| Dir | uint8_t | IN | Direction<br>0: CW<br>1: CCW |
| AuxPos | double[] | IN | Relative distance data from the center point to the start point<br>(Unit: user unit) |
| EndPos | double[] | IN | Relative distance data from the end point to the start point<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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);
```

```c
        return;
    }

    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    if(State == MC_GS_STANDBY) //Standby
    {
        CircAuxPos [0] = 3.5355;    //Center Position
        CircAuxPos [1] = 3.5355;    //Center Position
        CircAuxPos [2] = 0.0;            //Center Position

        CircEndPos[0] = 3.5355       // End Position
        CircEndPos[1] = 3.5355       // End Position
        CircEndPos[2] = 5.0          // End Position
        CircDir = 0; // CW
        GroupVel = 5;
        ret = ECAT_McGroupMove3DCircularRel_CP_EP(EcatDeviceID, GroupNo, GroupVel
                , CircDir, CircAuxPos, CircEndPos);
        if(ret < 0)
        {
            printf("Failed to add group move circular command:%d\n", ret);
        }

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

        }while(State == MC_GS_MOVING) //Moving

        if(State == MC_GS_STANDBY) //Standby
            printf("Group move circular successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
    }
```
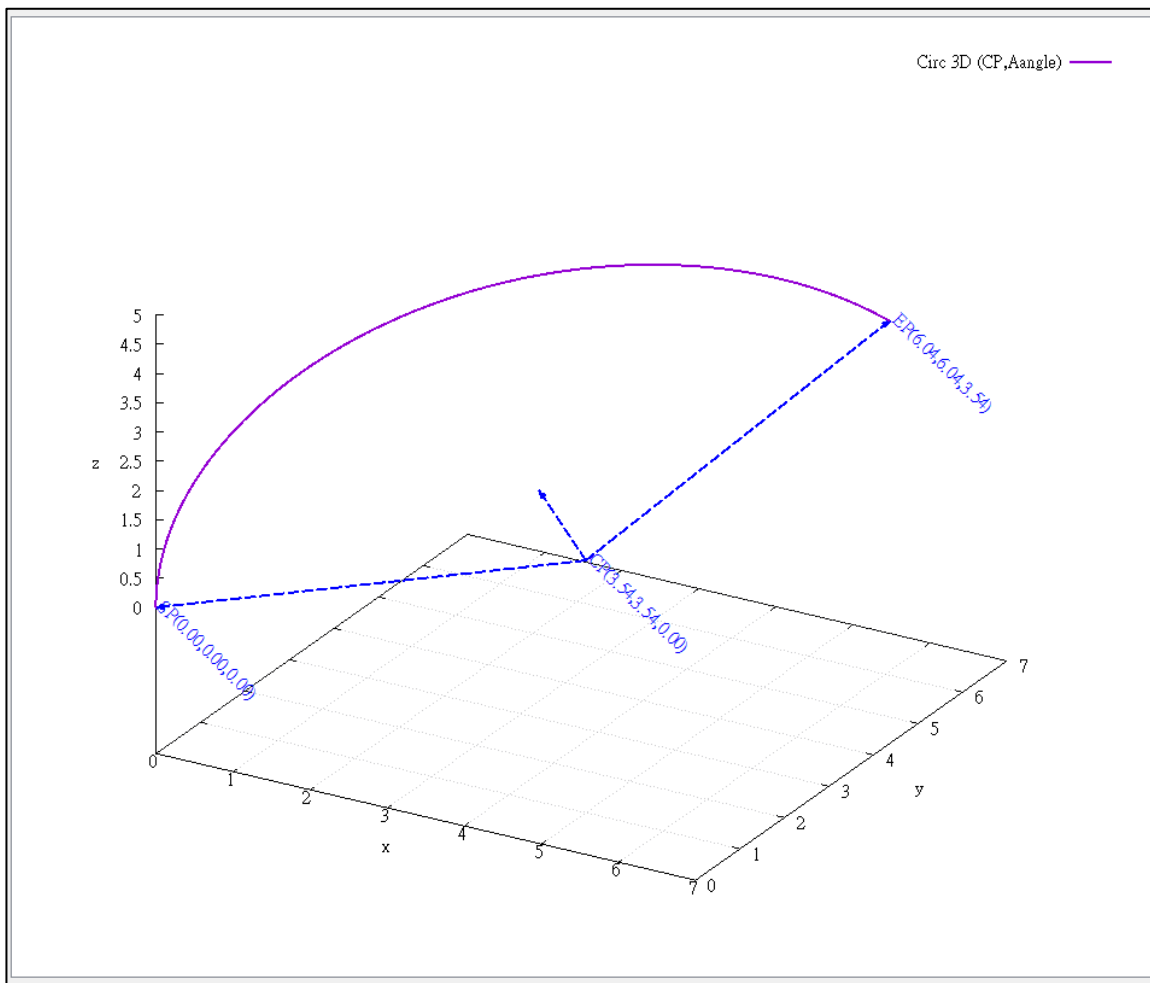
## 7.8.23.  ECAT_McGroupMove3DCircularAbs_BP_EP

**Description:**

Start an absolute 3D circular interpolation motion by providing a border position and the end position.

**Syntax:**

int32_t ECAT_McGroupMove3DCircularAbs_BP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity (Unit: user unit/s) |
| Dir | uint8_t | IN | Direction 0: CW 1: CCW |
| AuxPos | double[] | IN | Absolute position data of the border point (Unit: user unit) |
| EndPos | double[] | IN | Absolute position data of the end point (Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];


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

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

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
```

```
        return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
        //Command 1
        GroupPos[0] = 0.0;
        GroupPos[1] = 0.0;
        GroupPos[2] = 0.0;
        GroupVel = 5;
        ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
        if(ret < 0)
                printf("Failed to add group move line command:%d\n", ret);

        //Command 2
        CircAuxPos [0] = 1.036;     //Border Position
        CircAuxPos [1] = 1.036;     //Border Position
        CircAuxPos [2] = 3.5355;   //Border Position

        CircEndPos[0] = 8.53656       // End Position
        CircEndPos[1] = 8.53656       // End Position
        CircEndPos[2] = 5.0         // End Position
        CircDir = 1; // CCW
        GroupVel = 5;
        ret = ECAT_McGroupMove3DCircularAbs_BP_EP(EcatDeviceID, GroupNo, GroupVel
                , CircDir, CircAuxPos, CircEndPos);
        if(ret < 0)
        {
                printf("Failed to add group move circular command:%d\n", ret);
        }

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

        }while(State == MC_GS_MOVING) //Moving
```

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

## 3D circular interpolation motion path of example:

## 7.8.24. ECAT_McGroupMove3DCircularRel_BP_EP

**Description:**

Start a relative 3D circular interpolation motion by providing a border position and the end position.

**Syntax:**

int32_t ECAT_McGroupMove3DCircularRel_BP_EP(uint16_t DeviceNo, uint16_t GroupNo, double Vel, uint8_t Dir, double AuxPos[], double EndPos[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity (Unit: user unit/s) |
| Dir | uint8_t | IN | Direction 0: CW 1: CCW |
| AuxPos | double[] | IN | Relative distance data from the border point to the start point (Unit: user unit) |
| EndPos | double[] | IN | Relative distance data from the end point to the start point (Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
uint8_t CircDir;
double CircEndPos[MC_AXIS_NO_MAX];


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

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

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
```

```
        return;
    }

    ret = ECAT_McGetGroupState(DeviceNo, GroupNo,& State);
    if(State == MC_GS_STANDBY) //Standby
    {
        CircAuxPos [0] = 1.036;     //Border Position
        CircAuxPos [1] = 1.036;     //Border Position
        CircAuxPos [2] = 3.5355;    //Border Position

        CircEndPos[0] = 8.53656       // End Position
        CircEndPos[1] = 8.53656       // End Position
        CircEndPos[2] = 5.0         // End Position
        CircDir = 1; // CCW
        GroupVel = 5;
        ret = ECAT_McGroupMove3DCircularRel_BP_EP(EcatDeviceID, GroupNo, GroupVel
                , CircDir, CircAuxPos, CircEndPos);
        if(ret < 0)
        {
            printf("Failed to add group move circular command:%d\n", ret);
        }

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

        }while(State == MC_GS_MOVING) //Moving

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

## 7.8.25. ECAT_McGroupMoveHelicalAbs

**Description:**

Start a helical interpolation motion.

**Syntax:**

int32_t ECAT_McGroupMoveHelicalAbs(uint16_t DeviceNo,uint16_t GroupNo, double Angle, double AuxPos[], double Pitch, double Vel)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Angle | double | IN | Angle of rotation 360 indicates one full revolution; and 720 will produce two full revolutions (Unit: degree) |
| AuxPos | double[] | IN | Absolute position data of the center point (Unit: user unit) |
| Pitch | double | IN | Pitch (Unit: user unit) |
| Vel | double | IN | Velocity (Unit: user unit/s) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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);
```

```c
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);

    //Command 2
    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    ret = ECAT_McGroupMoveHelicalAbs(DeviceNo, GroupNo
            , CircAngle, CircAuxPos, HelicalPitch, GroupVel);
    if(ret < 0)
        printf("Failed to add group move helical command:%d\n", ret);

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

    }while(State == MC_GS_MOVING) //Moving

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

**helical interpolation motion path of example (right-handed):**
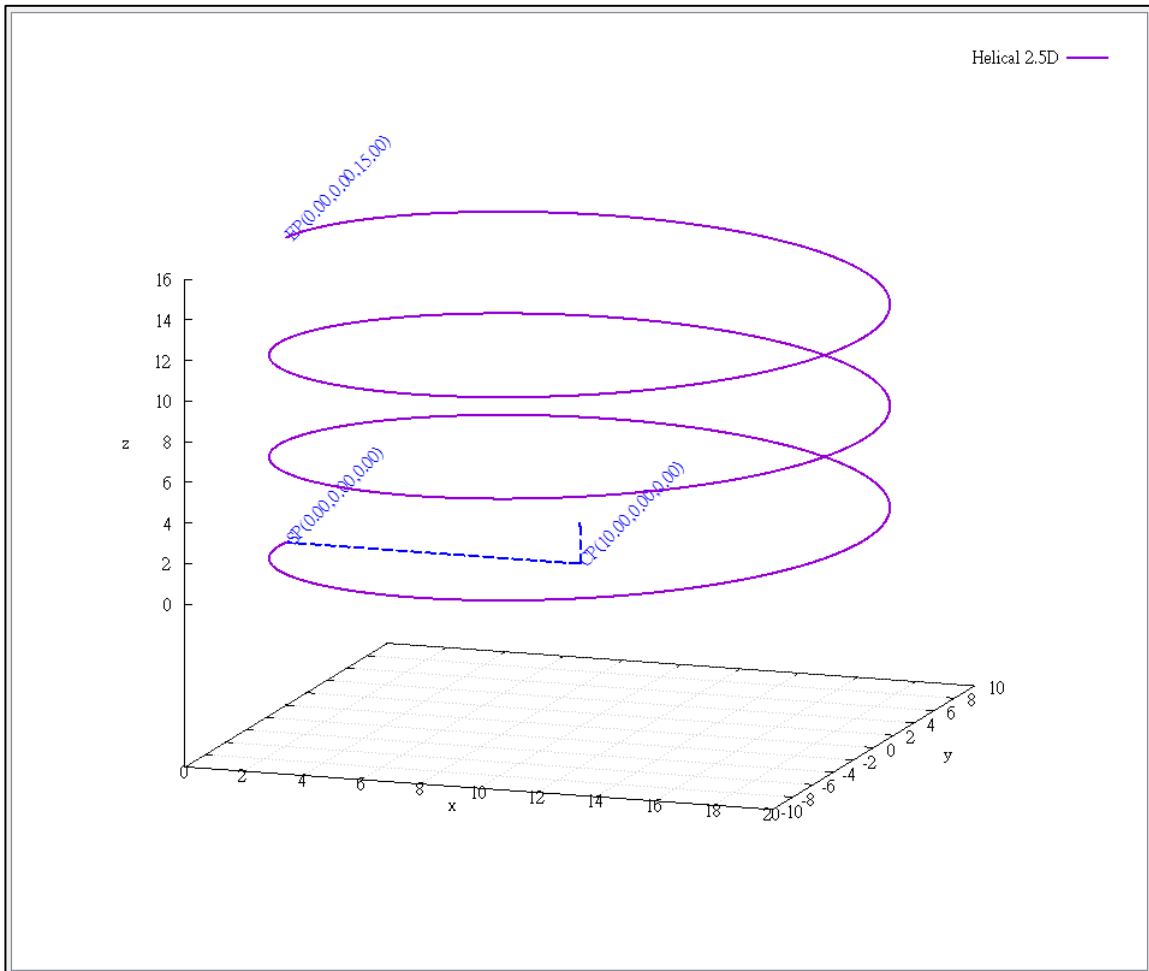
**helical interpolation motion path of example (left-handed):**

If the rotation angle parameter is set to negative value, the helical motion path is left-handed.
**CircAngle = -1080;**
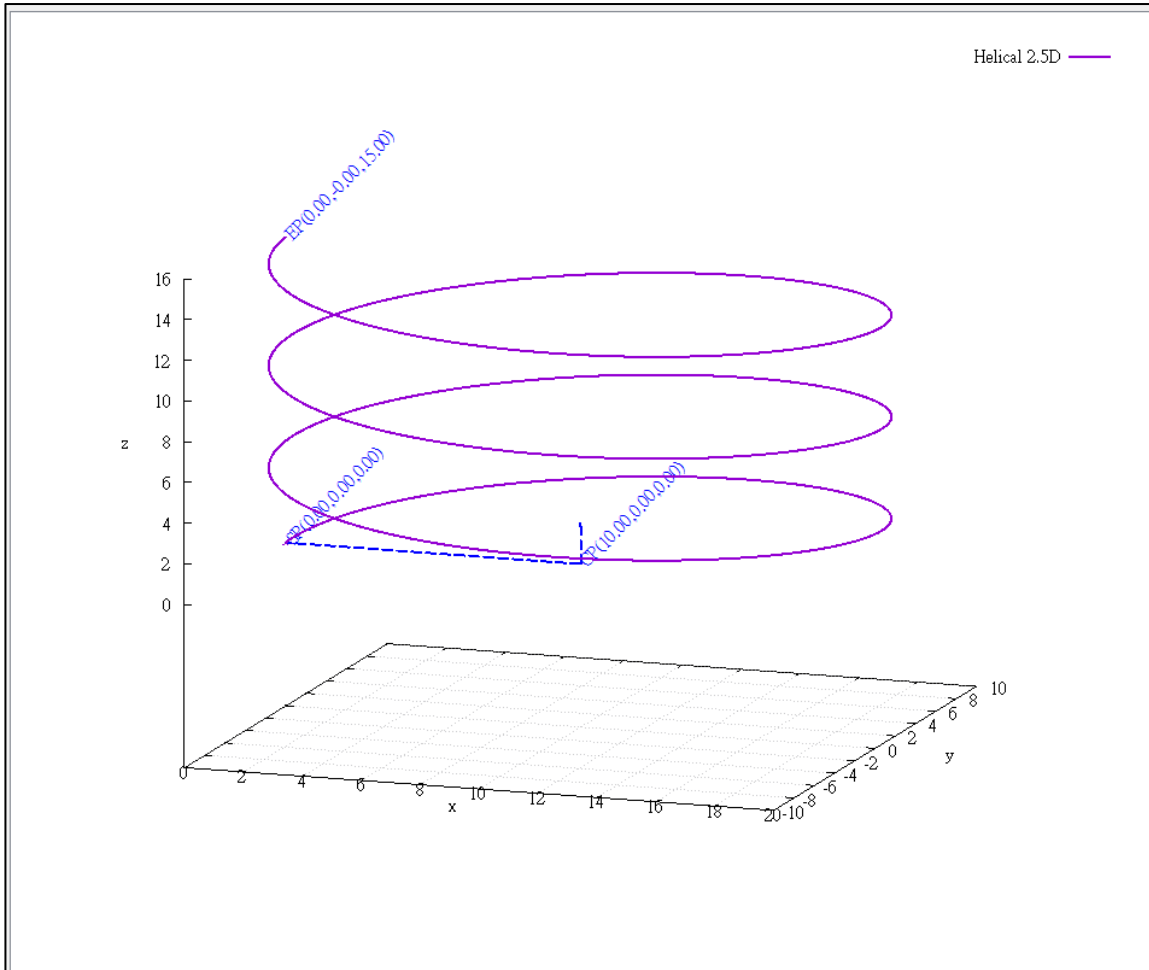
## 7.8.26. ECAT_McGroupMoveHelicalRel

**Description:**

Start a relative helical interpolation motion.

**Syntax:**

int32_t ECAT_McGroupMoveHelicalRel(uint16_t DeviceNo,uint16_t GroupNo, double Angle, double AuxPos[], double Pitch, double Vel)

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Angle | double | IN | Angle of rotation<br>360 indicates one full revolution and 720 will produce two full revolutions<br>(Unit: degree) |
| AuxPos | double[] | IN | Axis relative distance data from the center point to the start point<br>(Unit: user unit) |
| Pitch | double | IN | Pitch<br>(Unit: user unit) |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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


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


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


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

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


ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0; //Center Position
    GroupPos[1] = 20.0; //Center Position
    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    ret = ECAT_McGroupMoveHelicalRel (EcatDeviceID, GroupNo,
            CircAngle, GroupPos, HelicalPitch, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move helical command:%d\n", ret);
    }

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

    }while(State == MC_GS_MOVING) //Moving

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

## 7.8.27. ECAT_McGroupMove3DHelicalAbs_CP_Angle

**Description:**

Start an absolute 3D helical interpolation motion.

**Syntax:**

int32_t ECAT_McGroupMove3DHelicalAbs_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[])

**Parameters:**

| Name | Type | IN or OUT | Description |
| --- | --- | --- | --- |
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| Angle | double | IN | Angle of rotation<br>360 indicates one full revolution and 720 will produce two full revolutions<br>(Unit: degree) |
| Pitch | double | IN | Pitch<br>(Unit: user unit) |
| AuxPos | double[] | IN | Absolute position data of the center point of the base circle<br>(Unit: user unit) |
| NV | double[] | IN | Normal vector of the base circle |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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;
}
```

```c
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);


    //Command 2
    CircAuxPos [0] = 3.5355;   //Center Position
    CircAuxPos [1] = 3.5355;   //Center Position
    CircAuxPos [2] = 0.0;          //Center Position

    NV [0] = -0.7071;       //Normal Vector
    NV [1] = 0.7071;    //Normal Vector
    NV [2] = 0.0;           //Normal Vector

    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    ret = ECAT_McGroupMove3DHelicalAbs_CP_Angle(DeviceNo, GroupNo, GroupVel
            , CircAngle, HelicalPitch, CircAuxPos, NV);
    if(ret < 0)
        printf("Failed to add group move helical command:%d\n", ret);

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

    }while(State == MC_GS_MOVING) //Moving

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

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

## 3D helical interpolation motion path of example (right-handed):
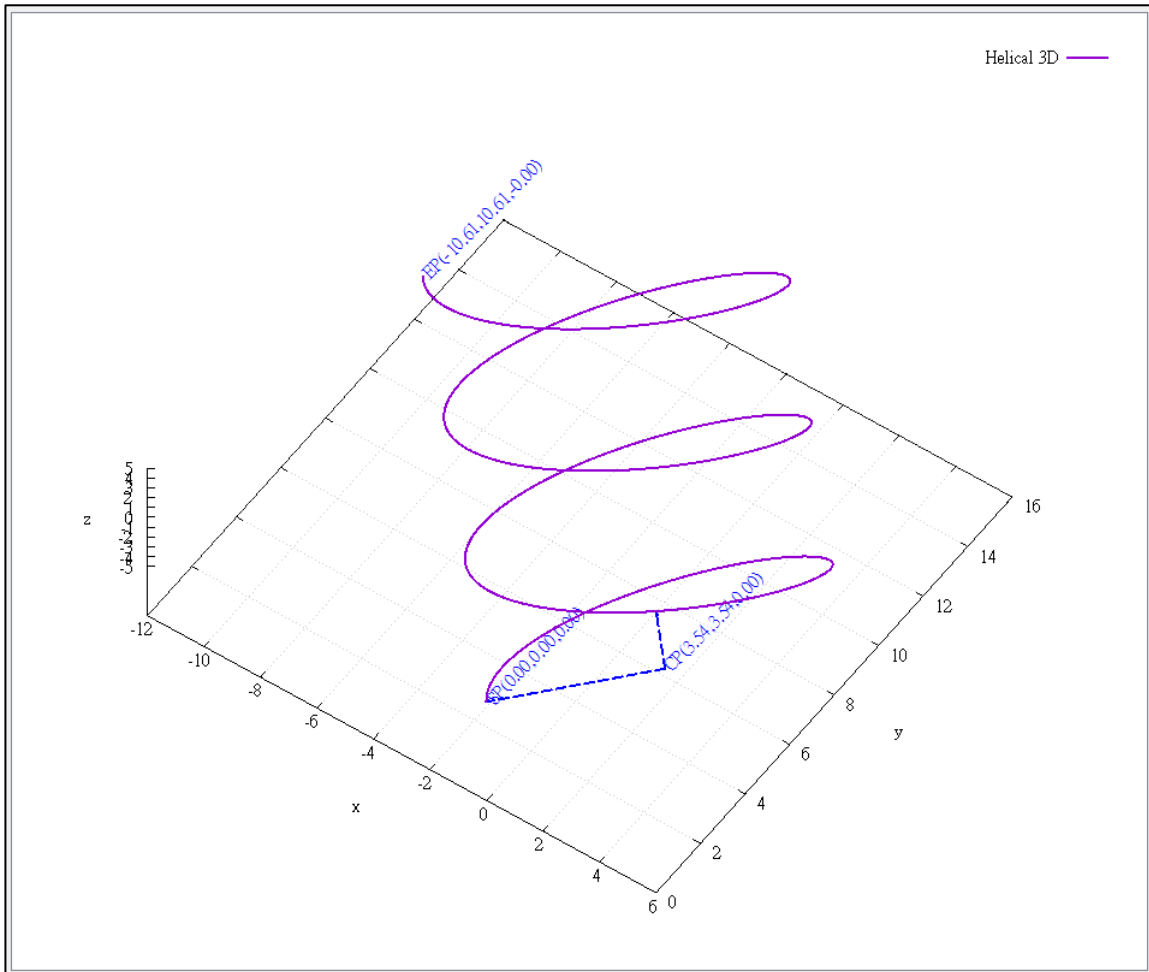
## 3D helical interpolation motion path of example (left-handed):

If the rotation angle parameter is set to negative value, the helical motion path is left-handed.
CircAngle = -1080;
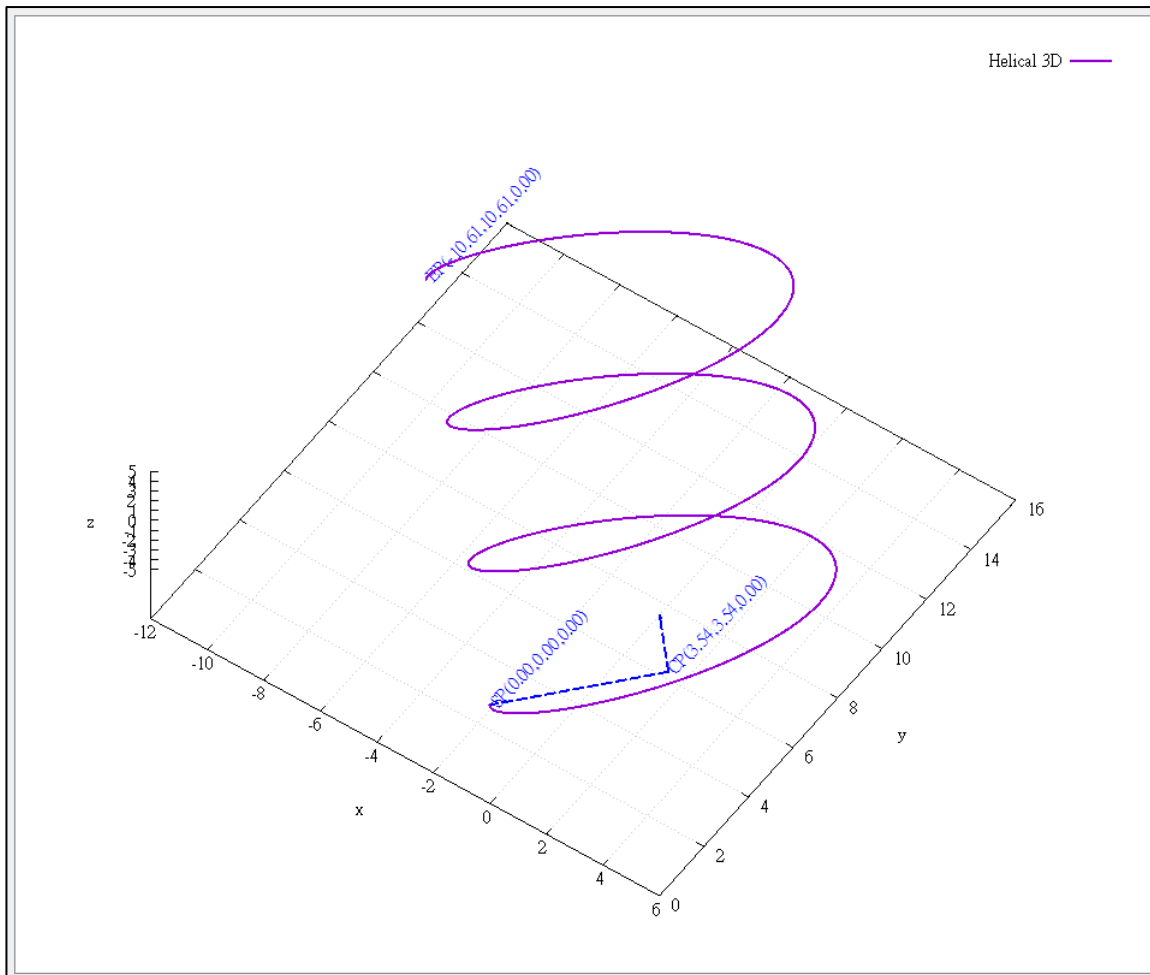
## 7.8.28. ECAT_McGroupMove3DHelicalRel_CP_Angle

**Description:**

Start a relative 3D helical interpolation motion.

**Syntax:**

int32_t ECAT_McGroupMove3DHelicalRel_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| Angle | double | IN | Angle of rotation<br>360 indicates one full revolution and 720 will produce two full revolutions<br>(Unit: degree) |
| Pitch | double | IN | Pitch<br>(Unit: user unit) |
| AuxPos | double[] | IN | Relative distance data from the center point of the base circle to its start point<br>(Unit: user unit) |
| NV | double[] | IN | Normal vector of the base circle |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;

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

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

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

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 3.5355;   //Center Position
    CircAuxPos [1] = 3.5355;   //Center Position
    CircAuxPos [2] = 0.0;          //Center Position

    NV [0] = -0.7071;       //Normal Vector
    NV [1] = 0.7071;   //Normal Vector
    NV [2] = 0.0;           //Normal Vector

    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    ret = ECAT_McGroupMove3DHelicalRel_CP_Angle(DeviceNo, GroupNo, GroupVel
            , CircAngle, HelicalPitch, CircAuxPos, NV);
    if(ret < 0)
        printf("Failed to add group move helical command:%d\n", ret);

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

    }while(State == MC_GS_MOVING) //Moving

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

## 7.8.29. ECAT_McGroupMoveConicalHelixAbs

**Description:**

Start an absolute conical helix interpolation motion.

**Syntax:**

int32_t ECAT_McGroupMoveConicalHelixAbs(uint16_t DeviceNo, uint16_t GroupNo, double Angle, double AuxPos[], double Pitch, double Vel, double EndRadius)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Angle | double | IN | Angle of rotation<br>360 indicates one full revolution and 720 will produce two full revolutions<br>(Unit: degree) |
| AuxPos | double[] | IN | Absolute position data of the center point<br>"Start Radius" is the distance between the center point and the start point<br>(Unit: user unit) |
| Pitch | double | IN | Pitch<br>(Unit: user unit) |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| EndRadius | double | IN | End Radius<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

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

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

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

```
}


ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);


    //Command 1
    CircAuxPos[0] = 10.0; //Center Position
    CircAuxPos[1] = 0.0; //Center Position
    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 20;
    ret = ECAT_McGroupMoveConicalHelixAbs(DeviceNo, GroupNo
            , CircAngle, CircAuxPos, HelicalPitch, GroupVel, EndRadius);
    if(ret < 0)
        printf("Failed to add group move conical helix command:%d\n", ret);


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

    }while(State == MC_GS_MOVING) //Moving


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

}

conical helical interpolation motion path of example:
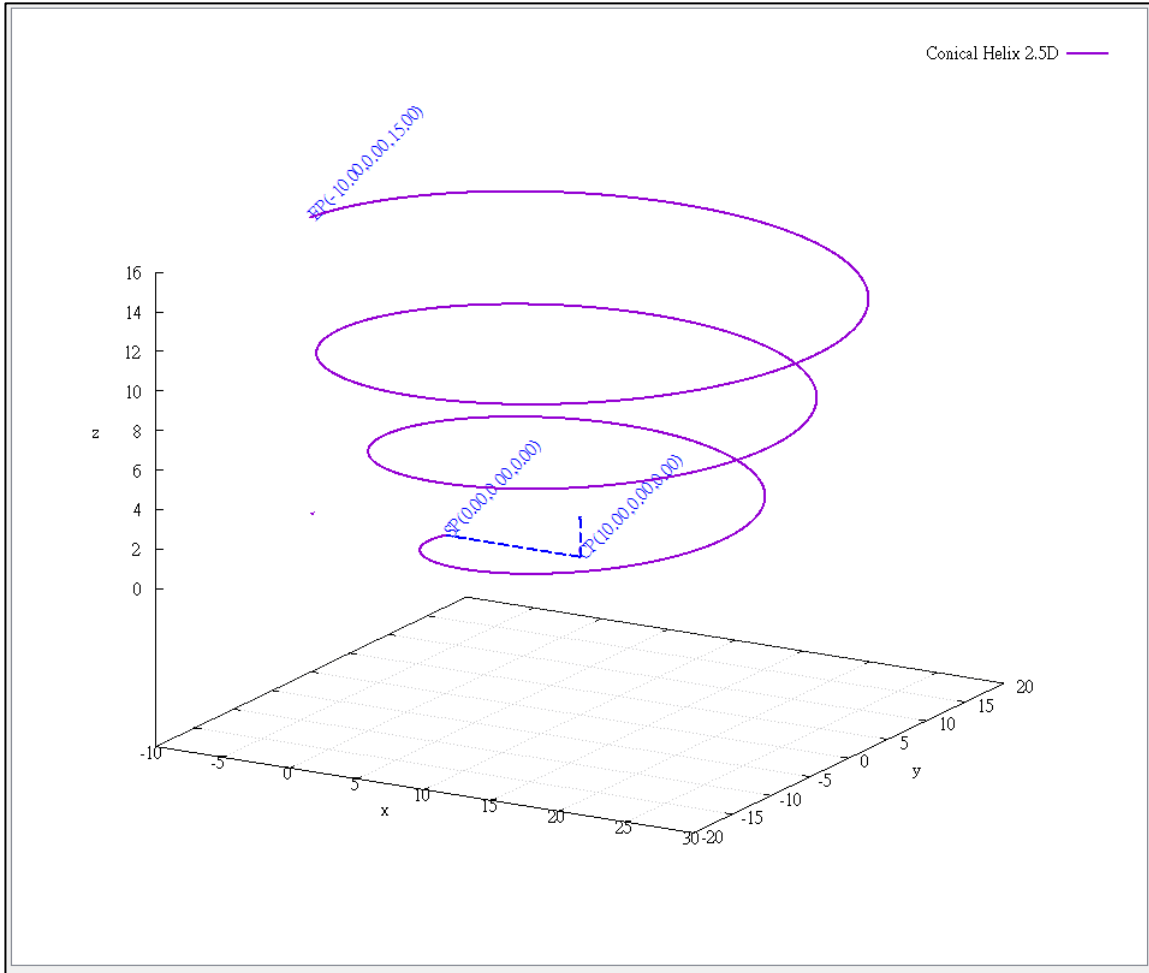
## 7.8.30. ECAT_McGroupMoveConicalHelixRel

Description:

Start a relative conical helix interpolation motion.

Syntax:

int32_t ECAT_McGroupMoveConicalHelixRel(uint16_t DeviceNo, uint16_t GroupNo, double Angle, double AuxPos[], double Pitch, double Vel, double EndRadius)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Angle | double | IN | Angle of rotation<br>360 indicates one full revolution and 720 will produce two full revolutions<br>(Unit: degree) |
| AuxPos | double[] | IN | Relative position data of the center point<br>"Start Radius" is the distance between the center point and the start point<br>(Unit: user unit) |
| Pitch | double | IN | Pitch<br>(Unit: user unit) |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| EndRadius | double | IN | End Radius<br>(Unit: user unit) |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

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

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```

```
if(State == MC_GS_STANDBY) //Standby
{
     CircAuxPos[0] = 10.0; //Center Position
     CircAuxPos[1] = 0.0; //Center Position
     CircAngle = 1080;
     HelicalPitch = 5;
     GroupVel = 5;
     EndRadius = 20;
     ret = ECAT_McGroupMoveConicalHelixRel(DeviceNo, GroupNo
               , CircAngle, CircAuxPos, HelicalPitch, GroupVel, EndRadius);
     if(ret < 0)
          printf("Failed to add group move conical helix command:%d\n", ret);

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

     }while(State == MC_GS_MOVING) //Moving

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

## 7.8.31. ECAT_McGroupMove3DConicalHelixAbs_CP_Angle

**Description:**

Start an absolute 3D conical helix interpolation motion.

**Syntax:**

int32_t ECAT_McGroupMove3DConicalHelixAbs_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[], double EndRadius)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| Angle | double | IN | Angle of rotation<br>360 indicates one full revolution and 720 will result in two full revolutions<br>(Unit: degree) |
| Pitch | double | IN | Pitch<br>(Unit: user unit) |
| AuxPos | double[] | IN | Absolute position data of the center point<br>(Unit: user unit) |
| NV | double[] | IN | Normal vector |
| EndRadius | double | IN | End Radius<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
double EndRadius;

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

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
```

```
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 0.0;
    GroupPos[1] = 0.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n", ret);

    //Command 2
    CircAuxPos [0] = 3.5355;    //Center Position
    CircAuxPos [1] = 3.5355;    //Center Position
    CircAuxPos [2] = 0.0;            //Center Position

    NV [0] = -0.7071;        //Normal Vector
    NV [1] = 0.7071;    //Normal Vector
    NV [2] = 0.0;            //Normal Vector

    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 20;
    ret = ECAT_McGroupMove3DConicalHelixAbs _CP_Angle(DeviceNo, GroupNo, GroupVel
            , CircAngle, HelicalPitch, CircAuxPos, NV, EndRadius);
    if(ret < 0)
        printf("Failed to add group move conical helix command:%d\n", ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving
```

```
        if(State == MC_GS_STANDBY) //Standby

            printf("Group move circular successfully!\n");

        else if(State == MC_GS_ERRORSTOP) //ErrorStop

        {

            printf("Group error stop\n");

        }

}
```

**conical helical interpolation motion path of example:**

## 7.8.32.  ECAT_McGroupMove3DConicalHelixRel_CP_Angle

**Description:**

Start a relative 3D conical helix interpolation motion.

**Syntax:**

int32_t ECAT_McGroupMove3DConicalHelixRel_CP_Angle(uint16_t DeviceNo, uint16_t GroupNo, double Vel, double Angle, double Pitch, double AuxPos[], double NV[], double EndRadius)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Vel | double | IN | Velocity<br>(Unit: user unit/s) |
| Angle | double | IN | Angle of rotation<br>360 indicates one full revolution and 720 will result in two full revolutions.<br>(Unit: degree) |
| Pitch | double | IN | Pitch<br>(Unit: user unit) |
| AuxPos | double[] | IN | Relative position data of center point<br>(Unit: user unit) |
| NV | double[] | IN | Normal vector |
| EndRadius | double | IN | End Radius<br>(Unit: user unit) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double CircAuxPos[MC_AXIS_NO_MAX];
double NV[3];
double CircAngle;
double GroupVel;
double EndRadius;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}
```

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    CircAuxPos [0] = 3.5355;   //Center Position
    CircAuxPos [1] = 3.5355;   //Center Position
    CircAuxPos [2] = 0.0;          //Center Position

    NV [0] = -0.7071;        //Normal Vector
    NV [1] = 0.7071;    //Normal Vector
    NV [2] = 0.0;            //Normal Vector

    CircAngle = 1080;
    HelicalPitch = 5;
    GroupVel = 5;
    EndRadius = 20;
    ret = ECAT_McGroupMove3DConicalHelixRel_CP_Angle(DeviceNo, GroupNo, GroupVel
            , CircAngle, HelicalPitch, CircAuxPos, NV, EndRadius);
    if(ret < 0)
        printf("Failed to add group move conical helix command:%d\n", ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move circular successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

## 7.8.33. ECAT_McGroupMoveProfile

**Description:**

Start a profile position motion.

**Syntax:**

int32_t ECAT_McGroupMoveProfile(uint16_t DeviceNo, uint16_t GroupNo, uint16_t ProfileNo[], uint16_t TotalStep)

**Parameters:**

| Name | Type | IN or OUT | Description |
| --- | --- | --- | --- |
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| ProfileNo | uint16_t[] | IN | An array contains several profile buffer numbers. Each element in this array is a profile buffer number. |
| TotalStep | uint16_t | IN | Total moving steps |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t ProfileNo[MC_AXIS_NO_MAX];
uint16_t TotalStep = 1000;

...
AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    ProfileNo[0] = 0;
    ProfileNo[1] = 1;
    ret = ECAT_McGroupMoveProfile(DeviceNo, GroupNo, ProfileNo, TotalStep);
    if(ret < 0)
    {
        printf("Failed to start group move profile:%d\n", ret);
    }
```

```
do
{
    sleep(1);
    ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

}while(State == MC_GS_MOVING) //Moving

if(State == MC_GS_STANDBY) //Standby
    printf("Group move profile successfully!\n");
else if(State == MC_GS_ERRORSTOP) //ErrorStop
{
    printf("Group error stop\n");
}
}
```

## 7.8.34. ECAT_McGroupMoveProfileCSV

**Description:**

Start a profile position motion. The profile data are read from a CSV file. The file format is shown in Figure 7.1.

**Syntax:**

int32_t ECAT_McGroupMoveProfileCSV(uint16_t DeviceNo, uint16_t GroupNo, uint16_t ProfileNo[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| ProfileNo | uint16_t | IN | File number of Profile data This file contains profile data for all axes in the group. |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t ProfileNo = 0;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}
AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    ret = ECAT_McGroupMoveProfileCSV(DeviceNo, GroupNo, ProfileNo);
    if(ret < 0)
    {
        printf("Failed to start group move profile CSV:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
```

```
    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move profile successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

## 7.8.35. ECAT_McGroupMoveDwell

**Description:**

The motion kernel will make a group to wait for the dwell time; after time is up, continue to load and execute the next command. This command can be used for adjusting the blending distance between two motion commands in continuous blending motion. This command behaves just like any other motion commands and is sequentially executed. In Buffered or Blending mode, if a motion command is being executed, it will be pushed into the command buffer. In Aborting mode, motion kernel will stop executing the current command by deceleration and start to wait for the dwell time.

**Syntax:**

int32_t ECAT_McGroupMoveDwell(uint16_t DeviceNo, uint16_t GroupNo, uint32_t Cnt)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Cnt | Uint32_t | IN | Dwell time<br>Unit: EtherCAT cycle time |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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;
}

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] = 9.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
```
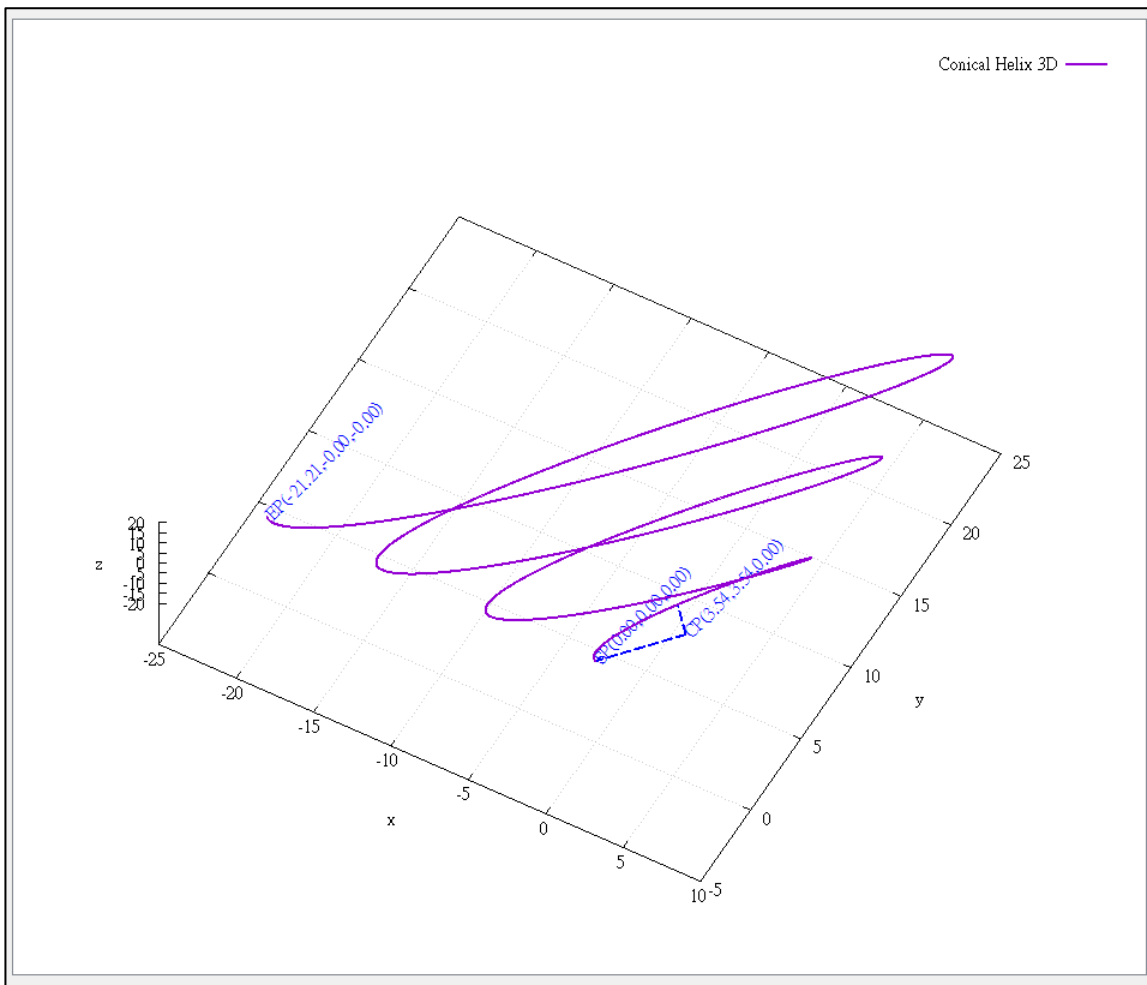
```
        printf("Failed to add group move line command:%d\n",ret);
    }
    DwellTime = 500; //Wait 500ms, If cycletime = 1ms
    ret = ECAT_McGroupMoveDwell(DeviceNo, GroupNo, DwellTime);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n",ret);
    }


    //Command 2
    GroupPos[0] = 0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }



    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```
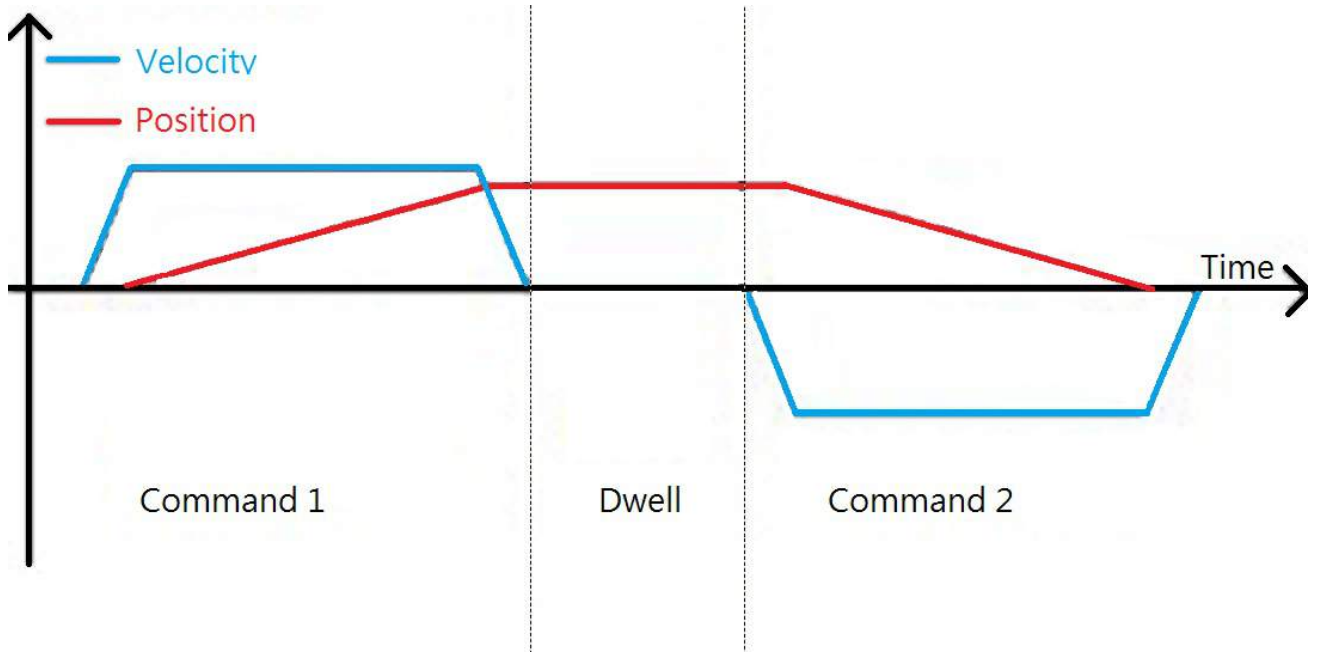
## 7.8.36.  ECAT_McGroupMoveDO

**Description:**

Add a slave DO output command in the group motion. This command will not execute immediately. It will be put into command buffer and wait for execution.

**Syntax:**

int32_t ECAT_McGroupMoveDO(uint16_t DeviceNo,uint16_t GroupNo, uint16_t SlaveNo, uint16_t BitNo, uint32_t Value)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| SlaveNo | uint16_t | IN | Slave number |
| BitNo | uint16_t | IN | Bit number |
| Value | uint32_t | IN | Bit data (0 or 1) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint16_t SlaveNo, BitNo, Value;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
```

```
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }
    //Command 2
    SlaveNo = 3;
    BitNo = 1;
    Value = 1;
    DwellTime = 1000; //Wait 1s, If cycletime = 1ms
    ret = ECAT_McGroupMoveDO(DeviceNo, GroupNo, SlaveNo, BitNo, Value);
    if(ret < 0)
    {
        printf("Failed to add group move DO command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

## 7.8.37. ECAT_McGroupMoveAO

**Description:**

Add a slave AO output command in the group motion. This command will not execute immediately. It will be put into command buffer and wait for execution.

Note:   Please use ECAT_SetSlaveAoProperty to configure an AO slave before setting its value.

**Syntax:**

int32_t ECAT_McGroupMoveAO(uint16_t DeviceNo, uint16_t GroupNo, uint16_t SlaveNo, uint32_t RunMode, uint16_t ChannelNo, uint16_t RawData, double VoltData)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| SlaveNo | uint16_t | IN | Slave number |
| RunMode | uint32_t | IN | RunMode<br>0: Use the binary value to set AO<br>1: Use the voltage output value to set AO |
| ChannelNo | uint16_t | IN | Channel number |
| RawData | uint16_t | IN | AO integer value (an unsigned 16-bit integer value) |
| VoltData | double | IN | AO voltage value (an floating-point value) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint16_t SlaveNo, ChannelNo, RawData;
double VoltData = 0;
uint32_t RunMode = 0;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;


AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

AxisNo = 1;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}
```

```
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }
    //Command 2
    SlaveNo = 3;
    ChannelNo = 0;
    RawData = 32767;
    DwellTime = 1000; //Wait 1s, If cycletime = 1ms
    ret = ECAT_McGroupMoveAO(DeviceNo, GroupNo, SlaveNo, RunMode, ChannelNo, RawData,
    VoltData);
    if(ret < 0)
    {
        printf("Failed to add group move AO command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

## 7.8.38.  ECAT_McGroupMoveBlendingSync

**Description:**

When the group is in the blending mode, this command will make the motion kernel to wait until the current command is finished before executing the next motion command. After that, the group is still in blending mode and a new current motion command will blend with its next motion command.

**Syntax:**

int32_t ECAT_McGroupMoveBlendingSync(uint16_t DeviceNo, uint16_t GroupNo)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BLENDING; //0: Aborting, 1: Buffered, 2: Blending
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;
uint32_t DwellTime;


AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n",ret);
    return;
}

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] = 5.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
```

```
            printf("Failed to add group move line command:%d\n",ret);
        }
        //Command 2
        ret = ECAT_McGroupMoveBlendingSync(DeviceNo, GroupNo);
        if(ret < 0)
        {
            printf("Failed to add group move line command:%d\n",ret);
        }

        //Command 3
        GroupPos[0] = 10.0;
        GroupVel = 5;
        ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
        if(ret < 0)
        {
            printf("Failed to add group move line command:%d\n",ret);
        }

        do
        {
            sleep(1);
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

        }while(State == MC_GS_MOVING) //Moving

        if(State == MC_GS_STANDBY) //Standby
            printf("Group move line successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
    }
```
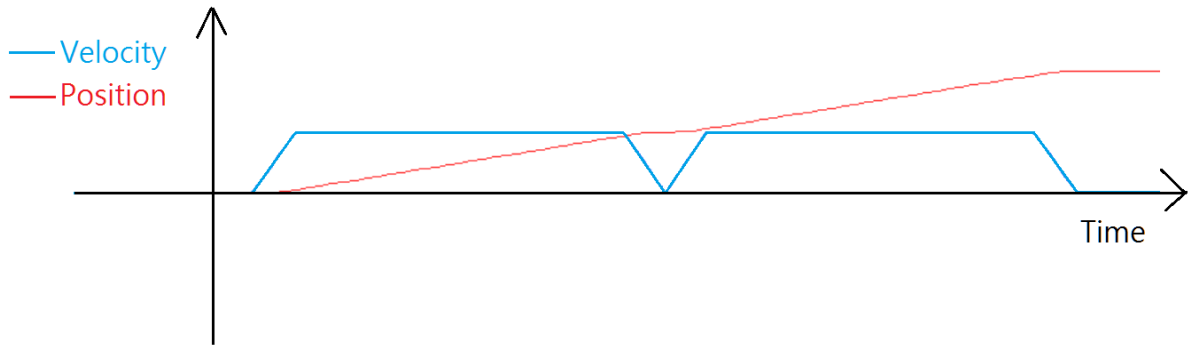
## 7.8.39. ECAT_McGroupStop

**Description:**

Stop the motion of a group with deceleration.

**Syntax:**

int32_t ECAT_McGroupStop(uint16_t DeviceNo, uint16_t GroupNo)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
nt32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint32_t State;
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;


...
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_DISABLED) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }
    //Command 2
    GroupPos[0] = 30.0;
    GroupPos[1] = 50.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }

    sleep(1000);
    ret = ECAT_McGroupStop(DeviceNo, GroupNo);
    if(ret < 0)
    {
        printf("Failed to stop group move:%d\n", ret);
```

```
        return;
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECAT_McGetGroupState(DeviceNo,GroupNo, &State);
        }while(State == MC_GS_STOPPING) //Stopping

        if(State == MC_GS_STANDBY) //Standby
            printf("Group move stop successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
    }
}
```

## 7.8.40.  ECAT_McGroupQuickStop

**Description:**

Stop the motion of a group immediately.

**Syntax:**

int32_t ECAT_McGroupQuickStop(uint16_t DeviceNo, uint16_t GroupNo)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
nt32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint32_t State;
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;


...
ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_DISABLED) //Standby
{
    //Command 1
    GroupPos[0] = 10.0;
    GroupPos[1] = 20.0;
    GroupVel = 5;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }
    //Command 2
    GroupPos[0] = 30.0;
    GroupPos[1] = 50.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineRel(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }

    sleep(1000);
    ret = ECAT_McGroupQuickStop(DeviceNo, GroupNo);
    if(ret < 0)
    {
        printf("Failed to stop group move:%d\n", ret);
```

```
            return;
        }
        else
        {
            do
            {
                sleep(1);
                ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
            }while(State == MC_GS_STOPPING) //Stopping

            if(State == MC_GS_STANDBY) //Standby
                printf("Group move stop successfully!\n");
            else if(State == MC_GS_ERRORSTOP) //ErrorStop
            {
                printf("Group error stop\n");
            }
        }
    }
}
```

## 7.8.41.  ECAT_McSetGroupHold

Description:

The group state becomes **MC_GS_HOLD** and the motion kernel will stop loading new commands after current command is done. After disable the holding, the motion kernel will load a new command from command buffer and execute commands sequentially.

Notice: The PVT motion command and other group motion commands cannot be used together.

Syntax:

int32_t ECAT_McSetGroupHold(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Status)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Status | uint16_t | IN | Do "hold command" or not<br>0: disable hold state<br>1: hold |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint32_t State;
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;
uint16_t CmdMode = MS_GRP_CM_BUFFERED; //0: Aborting, 1: Buffered, 2: Blending

AxisNo = 0;
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
{
    ret = ECAT_McSetGroupHold(DeviceNo, GroupNo, 1 ); // hold
    if(ret < 0)
    {
        printf("Failed to set group hold:%d\n",ret);
    }

    //Command 1
    GroupPos[0] = 5.0;
    GroupVel = 10;
```

```c
    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] = 10.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);
    usleep(200000); //sleep 200 ms

    ret = ECAT_McSetGroupHold(DeviceNo, GroupNo, 1 ); //Hold
    if(ret < 0)
    {
        printf("Failed to set group hold:%d\n",ret);
    }
    usleep(800000); //sleep 800 ms

    ret = ECAT_McSetGroupHold(DeviceNo, GroupNo, 0 ); //Resume
    if(ret < 0)
    {
        printf("Failed to set group resume:%d\n",ret);
    }
    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```
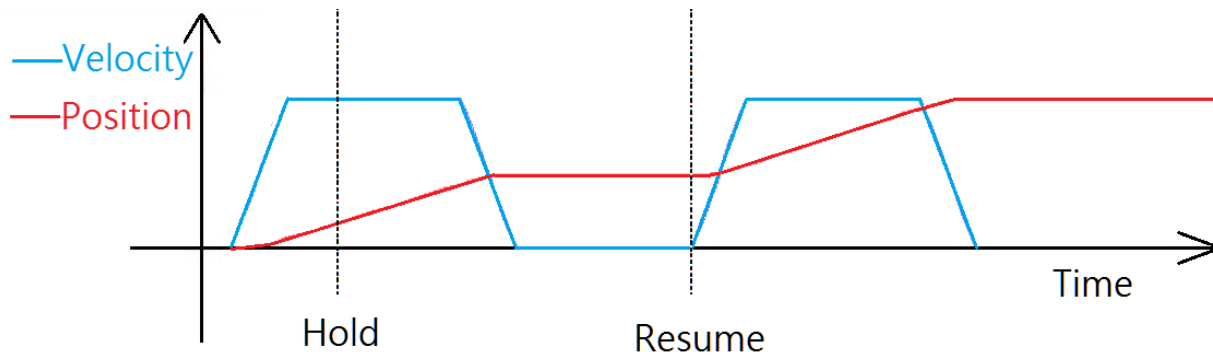
## 7.8.42.  ECAT_McSetGroupPause

**Description:**

The group state becomes MC_GS_PAUSE and the motion kernel will pause the current group motion with deceleration immediately. The current command is just partially done, and some remaining part is held. After the pause state becomes disabled, the motion kernel will execute the remaining part of the unfinished command and other commands in the command buffer sequentially.

**Syntax:**

int32_t ECAT_McSetGroupPause(uint16_t DeviceNo, uint16_t GroupNo, uint16_t Status)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Status | uint16_t | IN | To pause or not<br>0: disable the pause<br>1: enable the pause |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint32_t State;
double GroupPos[MC_AXIS_NO_MAX];
double GroupVel;
uint16_t CmdMode = MS_GRP_CM_ABORTING; //0: Aborting, 1: Buffered, 2: Blending

AxisNo = 0;
ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, AxisNo);
if(ret < 0)
{
    printf("Failed to add axis to group:%d\n", ret);
    return;
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    GroupPos[0] = 5.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel);
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);
    usleep(200000); //sleep 200 ms
    ret = ECAT_McSetGroupPause(DeviceNo, GroupNo, 1 ); //pause
    if(ret < 0)
```

```
    {
        printf("Failed to set group pause:%d\n",ret);
    }
    usleep(200000); //sleep 200 ms
    ret = ECAT_McSetGroupPause (DeviceNo, GroupNo, 0 ); //resume
    if(ret < 0)
        printf("Failed to set group resume:%d\n",ret);
    usleep(200000); //sleep 200 ms
    ret = ECAT_McSetGroupPause(DeviceNo, GroupNo, 1 ); //pause
    if(ret < 0)
    {
        printf("Failed to set group pause:%d\n",ret);
    }
    usleep(50000); //sleep 50ms

    GroupPos[0] = 10.0;
    GroupVel = 10;
    ret = ECAT_McGroupMoveLineAbs(DeviceNo, GroupNo, GroupPos, GroupVel); //Abort
    if(ret < 0)
        printf("Failed to add group move line command:%d\n",ret);

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```
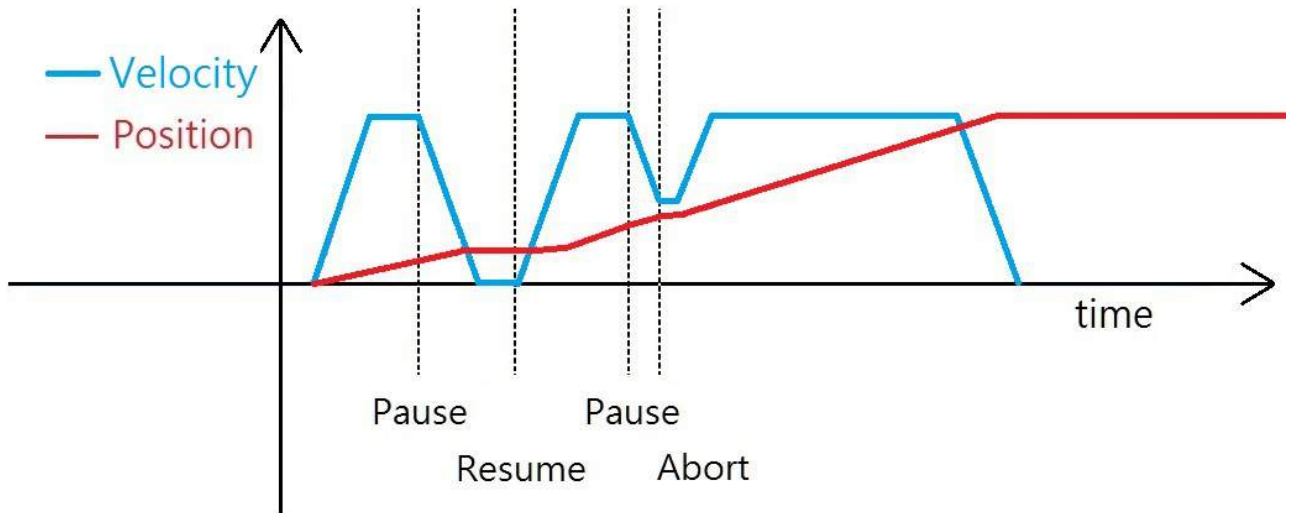
# 7.8.43.   ECAT_McAddPathData

**Description:**

Add a Path data to the queue.

**Syntax:**

int32_t ECAT_McAddPathData(uint16_t DeviceNo, uint16_t PathDataNo, uint32_t CmdType, uint8_t AbsMove, double EndPos[], double AuxPos[], double Args[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PathDataNo | uint16_t | IN | Queue number |
| CmdType | uint32_t | IN | Command type(Defined in Table 7.15) |
| AbsMove | uint8_t | IN | Absolute motion setting<br>0: Relative<br>1: Absolute |
| EndPos | double [] | IN | End position |
| AuxPos | double [] | IN | Auxiliary position |
| Args | double [] | IN | Parameters |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

Table 7.15: Path Data command type

| Macro Definition | Value | Description |
|---|---|---|
| MC_PATH_CMD_TYPE_MOVE_LINE | 1 | Linear interpolation motion. |
| MC_PATH_CMD_TYPE_MOVE_CIRC_CP_ANGLE | 2 | 2D circular interpolation motion by providing the center position and an angle. |
| MC_PATH_CMD_TYPE_MOVE_CIRC_CP_EP | 3 | 2D circular interpolation motion by providing the center position and the end position. |
| MC_PATH_CMD_TYPE_MOVE_CIRC_BP_EP | 4 | 2D circular interpolation motion by providing a border position and the end position. |
| MC_PATH_CMD_TYPE_MOVE_3D_CIRC_CP_ANGLE | 5 | 3D circular interpolation motion by providing the center position and an angle. |
| MC_PATH_CMD_TYPE_MOVE_3D_CIRC_CP_EP | 6 | 3D circular interpolation motion by providing the center position and the end position. |
| MC_PATH_CMD_TYPE_MOVE_3D_CIRC_BP_EP | 7 | 3D circular interpolation motion by providing a border position and the end position. |
| MC_PATH_CMD_TYPE_MOVE_HELICAL | 8 | Helical interpolation motion. |
| MC_PATH_CMD_TYPE_MOVE_3D_HELICAL | 9 | 3D helical interpolation |

| | | motion. |
|---|---|---|
| MC_PATH_CMD_TYPE_MOVE_CONICAL_HELIX | 10 | Conical helix interpolation motion. |
| MC_PATH_CMD_TYPE_MOVE_3D_CONICAL_HELIX | 11 | 3D Conical helix interpolation motion. |
| MC_PATH_CMD_TYPE_SET_ACCDEC_TIME | 12 | Set acceleration time |
| MC_PATH_CMD_TYPE_SET_ACCDEC_TYPE | 13 | Set the type of acceleration |
| MC_PATH_CMD_TYPE_SET_CMD_MODE | 14 | Set the blend mode |
| MC_PATH_CMD_TYPE_SET_BLEND_PERCENT | 15 | Set the percentage of blending |
| MC_PATH_CMD_TYPE_DWELL | 16 | Wait for the dwell time. |
| MC_PATH_CMD_TYPE_SET_DO | 17 | Output DO |
| MC_PATH_CMD_TYPE_SET_AO_VOLT | 18 | Output AO volt |

**Example:**

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double    EndPos[MC_AXIS_NO_MAX];
double    AuxPos[3];
double   Args[MC_PATH_DATA_ARGS_MAX];

/***************************Move Line*************************/
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                        AbsMove, EndPos, NULL, Args);
if(ret != 0)
```

```
        printf("Failed to add path data:%d\n",ret);


/*************************Move Circular (CP, ANGLE)*************************/
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
Args[0] = 4; //Velocity
Args[1] = 315; //Angle
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CIRC_CP_ANGLE,
AbsMove, NULL, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);


/*************************Move Circular (CP, EP)*************************/
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
EndPos[0] = 8.535533; //End Position
EndPos[1] = 3.535533; //End Position
Args[0] = 4; //Velocity
Args[1] = 1; //Dir
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CIRC_CP_EP,
                                AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);


/*************************Move Circular (BP, EP)*************************/
AuxPos[0] = 5; //Border Position
AuxPos[1] = 5; //Border Position
EndPos[0] = 8.535533; //End Position
EndPos[1] = 3.535533; //End Position
Args[0] = 4; //Velocity
Args[1] = 0; //Dir
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CIRC_BP_EP,
                                AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);
```

```
/**************************Move 3D Circular (CP, ANGLE)************************/
AuxPos[0] = 3.5355; //Center Position
AuxPos[1] = 3.5355; //Center Position
AuxPos[2] = 0.0; //Center Position
EndPos[0] = -0.7071; //Normal vector
EndPos[1] = 0.7071; //Normal vector
EndPos[2] = 0; //Normal vector
Args[0] = 4; //Velocity
Args[1] = 315; //Angle
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo,
MC_PATH_CMD_TYPE_MOVE_3D_CIRC_CP_ANGLE, AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);


/**************************Move 3D Circular (CP, EP)************************/
AuxPos[0] = 3.5355; //Center Position
AuxPos[1] = 3.5355; //Center Position
AuxPos[2] = 0.0; //Center Position
EndPos[0] = 1.03552; //End Position
EndPos[1] = 1.03552; //End Position
EndPos[2] = -3.53547; //End Position
Args[0] = 4; //Velocity
Args[1] = 0; //Dir
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_3D_CIRC_CP_EP,
AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);


/**************************Move 3D Circular (CP, EP)************************/
AuxPos[0] = 3.5355; //Center Position
AuxPos[1] = 3.5355; //Center Position
AuxPos[2] = 0.0; //Center Position
EndPos[0] = 1.03552; //End Position
EndPos[1] = 1.03552; //End Position
EndPos[2] = -3.53547; //End Position
```

```
Args[0] = 4; //Velocity
Args[1] = 0; //Dir
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_3D_CIRC_CP_EP,
AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);


/**************************Move 3D Circular (BP, EP)*************************/
AuxPos[0] = 1.036; //Border Position
AuxPos[1] = 1.036; //Border Position
AuxPos[2] = 3.5355; //Border Position
EndPos[0] = 6.035; //End Position
EndPos[1] = 6.035; //End Position
EndPos[2] = -3.53547; //End Position
Args[0] = 4; //Velocity
Args[1] = 1; //Dir
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_3D_CIRC_BP_EP,
AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);


/**************************Move Helical*************************/
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
Args[0] = 4; //Velocity
Args[1] = 720; //Angle
Args[2] = 5; //Pitch
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_HELICAL,
                                AbsMove, NULL, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);


/**************************Move 3D Helical*************************/
AuxPos[0] = 3.5355; //Center Position
AuxPos[1] = 3.5355; //Center Position
```

```
AuxPos[2] = 0.0; //Center Position
EndPos[0] = -0.7071; //Normal vector
EndPos[1] = 0.7071; //Normal vector
EndPos[2] = 0; //Normal vector
Args[0] = 4; //Velocity
Args[1] = 720; //Angle
Args[2] = 5; //Pitch
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_3D_HELICAL,
                                        AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);


/*************************Move Conical Helix*************************/
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
Args[0] = 4; //Velocity
Args[1] = 720; //Angle
Args[2] = 5; //Pitch
Args[3] = 10; //End Radius
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CONICAL_HELIX,
AbsMove, NULL, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);


/*************************Move 3D Conical Helix*************************/
AuxPos[0] = 3.5355; //Center Position
AuxPos[1] = 3.5355; //Center Position
AuxPos[2] = 0.0; //Center Position
EndPos[0] = -0.7071; //Normal vector
EndPos[1] = 0.7071; //Normal vector
EndPos[2] = 0; //Normal vector
Args[0] = 4; //Velocity
Args[1] = 720; //Angle
Args[2] = 5; //Pitch
Args[3] = 10; //End Radius
AbsMove = 0;
```

```
ret = ECAT_McAddPathData(DeviceNo, PathDataNo,
MC_PATH_CMD_TYPE_MOVE_3D_CONICAL_HELIX, AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
     printf("Failed to add path data:%d\n",ret);


/*************************Set Acc Time*************************/
rgs[0] = 900; //ms
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_ACCDEC_TIME,
                                    0, NULL, NULL, Args);
if(ret != 0)
     printf("Failed to add path data:%d\n",ret);


/*************************Set Acc Type*************************/
Args[0] = 2; // 1:T-Curve, 2:S-Curve
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_ACCDEC_TYPE,
                                    0, NULL, NULL, Args);
if(ret != 0)
     printf("Failed to add path data:%d\n",ret);


/*************************Set Cmd Mode*************************/
Args[0] = MS_GRP_CM_BLENDING; //MS_GRP_CM_BUFFERED or MS_GRP_CM_BLENDING
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_CMD_MODE,
                                    0, NULL, NULL, Args);
if(ret != 0)
     printf("Failed to add path data:%d\n",ret);


/*************************Set Bledn Percent*************************/
Args[0] = 50; //50%
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_BLEND_PERCENT,
                                    0, NULL, NULL, Args);
if(ret != 0)
     printf("Failed to add path data:%d\n",ret);


/*************************Execute Dwell*************************/
Args[0] = 2000; //Cycle time tick
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_DWELL,
                                    0, NULL, NULL, Args);
if(ret != 0)
```

```
        printf("Failed to add path data:%d\n",ret);


/*************************Output DO**************************/
Args[0] = 3; //SlaveNo
Args[1] = 1; //BitNo
Args[2] = 0; //Do Value
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_DO,
                                    0, NULL, NULL, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);


/*************************Output AO**************************/
Args[0] = 4; //SlaveNo
Args[1] = 1; //ChannelNo
Args[2] = 7.5; //Volt
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_SET_AO_VOLT,
                                    0, NULL, NULL, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);
```

# 7.8.44.    ECAT_McSetPathData

**Description:**

Modify the specified index Path data.

**Syntax:**

int32_t ECAT_McSetPathData(uint16_t DeviceNo, uint16_t PathDataNo, uint16_t DataIndex, uint32_t CmdType, uint8_t AbsMove, double EndPos[], double AuxPos[], double Args[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PathDataNo | uint16_t | IN | Queue number |
| DataIndex | uint16_t | IN | Data index |
| CmdType | uint32_t | IN | Command type(Defined in Table 7.15) |
| AbsMove | uint8_t | IN | Absolute motion setting<br>0: Relative<br>1: Absolute |
| EndPos | double [] | IN | End position |
| AuxPos | double [] | IN | Auxiliary position |
| Args | double [] | IN | Parameters |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double    EndPos[MC_AXIS_NO_MAX];
double    AuxPos[3];
double  Args[MC_PATH_DATA_ARGS_MAX];


/*************************Move Line*************************/
//data index 0
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                                   AbsMove, EndPos, NULL, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);


/*************************Move Circular *************************/
//data index 1
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
Args[0] = 4; //Velocity
Args[1] = 315; //Angle
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CIRC_CP_ANGLE,
AbsMove, NULL, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);



//modify data index 1
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
```

```
EndPos[0] = 8.535533; //End Position
EndPos[1] = 3.535533; //End Position
Args[0] = 4; //Velocity
Args[1] = 1; //Dir
AbsMove = 0;
ret = ECAT_McSetPathData(DeviceNo, PathDataNo, 1, MC_PATH_CMD_TYPE_MOVE_CIRC_CP_EP,
                                       AbsMove, EndPos, AuxPos, Args);
if(ret != 0)
       printf("Failed to set path data:%d\n",ret);
```

# 7.8.45. ECAT_McGetPathData

**Description:**

Get the specified index Path data.

**Syntax:**

int32_t ECAT_McGetPathData(uint16_t DeviceNo, uint16_t PathDataNo, uint16_t DataIndex, uint32_t *CmdType, uint8_t *AbsMove, double EndPos[], double AuxPos[], double Args[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PathDataNo | uint16_t | IN | Queue number |
| DataIndex | uint16_t | IN | Data index |
| CmdType | uint32_t * | OUT | Command type(Defined in Table 7.15) |
| AbsMove | uint8_t * | OUT | Absolute motion setting<br>0: Relative<br>1: Absolute |
| EndPos | double [] | OUT | End position |
| AuxPos | double [] | OUT | Auxiliary position |
| Args | double [] | OUT | Parameters |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double    EndPos[MC_AXIS_NO_MAX];
double    AuxPos[3];
double   Args[MC_PATH_DATA_ARGS_MAX];
uint32_t CmdType;
uint8_t AbsMove;
/**************************Move Line************************/
//data index 0
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                                  AbsMove, EndPos, NULL, Args);
if(ret != 0)
   printf("Failed to add path data:%d\n",ret);

ret = ECAT_McGetPathData(DeviceNo, PathDataNo, 0, &CmdType, &AbsMove, EndPos, AuxPos, Args);
if(ret != 0) {
       printf("Failed to get path data:%d\n",ret);
} else {
       printf("Cmd Type:%u\n", CmdType);
       printf("Abs. Move:%u\n", AbsMove);
       printf("EndPos[0]:%u\n", EndPos[0]);
       printf("EndPos[1]:%u\n", EndPos[1]);
       printf("Args[0]:%u\n", Args[0]);
}
```

# 7.8.46.　ECAT_McClearPathData

**Description:**

Clear the path data in the queue.

**Syntax:**

int32_t ECAT_McClearPathData(uint16_t DeviceNo, uint16_t PathDataNo)


**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PathDataNo | uint16_t | IN | Queue number |


**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double      EndPos[MC_AXIS_NO_MAX];
double      AuxPos[3];
double   Args[MC_PATH_DATA_ARGS_MAX];
/**************************Move Line*************************/
//data index 0
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                                   AbsMove, EndPos, NULL, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

ret = McClearPathData(DeviceNo, PathDataNo);
if(ret != 0) {
        printf("Failed to clear path data:%d\n",ret);
}
```

## 7.8.47.　ECAT_McGetPathDataSize

**Description:**

Get the number of Path data in the queue.

**Syntax:**

int32_t ECAT_McGetPathDataSize(uint16_t DeviceNo, uint16_t PathDataNo, uint16_t *Size)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PathDataNo | uint16_t | IN | Queue number |
| Size | uint16_t * | OUT | Number of Path data |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double     EndPos[MC_AXIS_NO_MAX];
double     AuxPos[3];
double   Args[MC_PATH_DATA_ARGS_MAX];
uint16_t Size;

/*************************Move Line*************************/
//data index 0
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                                    AbsMove, EndPos, NULL, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

ret = ECAT_McGetPathDataSize(DeviceNo, PathDataNo, &Size);
if(ret != 0) {
        printf("Failed to get path data size:%d\n",ret);
} else {
        printf("path data size:%u\n", Size);
}
```

# 7.8.48.  ECAT_McGroupMovePath

Description:

Start Path Motion Control.

Syntax:

int32_t ECAT_McGroupMovePath(uint16_t DeviceNo, uint16_t GroupNo, uint16_t PathDataNo, uint8_t Restart, uint16_t DataIndex, uint8_t Repeat)

Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PathDataNo | uint16_t | IN | Queue number |
| Restart | uint8_t | IN | Restart<br>0: If DataIndex is 0, it will be executed from the last stop index value. If DataIndex is not 0, it will be executed from the specified index value.<br>1: Execute from index 0 |
| DataIndex | uint16_t | IN | Data index |
| Repeat | uint8_t | IN | Repeat |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t PathDataNo = 0;
uint8_t AbsMove;
double     EndPos[MC_AXIS_NO_MAX];
double     AuxPos[3];
double   Args[MC_PATH_DATA_ARGS_MAX];

/*************************Move Line************************/
EndPos[0] = 10;
EndPos[1] = 10;
Args[0] = 4; //Velocity
AbsMove = 1;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_LINE,
                                    AbsMove, EndPos, NULL, Args);
if(ret != 0)
    printf("Failed to add path data:%d\n",ret);

/*************************Move Circular (CP, ANGLE)*************************/
AuxPos[0] = 5; //Center Position
AuxPos[1] = 0; //Center Position
Args[0] = 4; //Velocity
Args[1] = 315; //Angle
AbsMove = 0;
ret = ECAT_McAddPathData(DeviceNo, PathDataNo, MC_PATH_CMD_TYPE_MOVE_CIRC_CP_ANGLE,
AbsMove, NULL, AuxPos, Args);
if(ret != 0)
        printf("Failed to add path data:%d\n",ret);

/*************************Group Move Path************************/
ret = ECAT_McGroupMovePath(DeviceNo, GroupNo, PathDataNo, 1, 0, 1);
if(ret != 0)
        printf("Group move path failed:%d\n",ret);
```

## 7.8.49.  ECAT_McGroupMoveLineAbsAdv

**Description:**

Start an absolute linear interpolation motion of a group.

**Syntax:**

int32_t ECAT_McGroupMoveLineAbsAdv(uint16_t DeviceNo, uint16_t GroupNo, double EndPos[], double StartVel, double ReqVel, double FinalVel, double Accel, double Decel, uint8_t AccDecMode)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| EndPos | Double * | IN | Position array of a group<br>Each array element is the absolute position of an axis. (Unit: user unit) |
| StartVel | double | IN | Start velocity (Unit: user unit/s) |
| ReqVel | double | IN | Target velocity (Unit: user unit/s) |
| FinalVel | double | IN | Final velocity (Unit: user unit/s) |
| Accel | double | IN | Acceleration rate ($user\ unit/s^2$) or acceleration time (second) |
| Decel | double | IN | Deceleration rate ($user\ unit/s^2$) or deceleration time (second) |
| AccDecMode | uint8_t | IN | Acceleration and deceleration input mode:<br>0: acceleration and deceleration rate ($user\ unit/s^2$)<br>1: acceleration and deceleration time (second) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

| Example: |
| --- |
| [C/C++] |

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
double start_vel;
double req_vel;
double final_vel;
double accel;
double decel;
double end_pos[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_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
```

```
        start_vel = 0;
        req_vels = 2;
        final_vel = 2;
        accel = 4;
        decel = 4;
    end_pos[0] = 1.5;
    end_pos[1] = 1.5;


        ret = ECAT_McGroupMoveLineAbsAdv(DeviceNo, GroupNo, end_pos,
                        start_vel, req_vel, final_vel, accel, decel, 0);
        if (ret != 0)
            printf("Failed to add group move line command:%d\n", ret);


        //Command 2
        start_vel = 2;
        req_vels = 2.5;
        final_vel = 1.5;
        accel = 6;
        decel = 6;
    end_pos[0] = 3.5;
    end_pos[1] = 3.5;


        ret = ECAT_McGroupMoveLineAbsAdv(DeviceNo, GroupNo, end_pos,
                        start_vel, req_vel, final_vel, accel, decel, 0);
        if (ret != 0)
            printf("Failed to add group move line command:%d\n", ret);


        //Command 3
        start_vel = 1.5;
        req_vels = 1.5;
        final_vel = 0;
        accel = 2;
        decel = 2;
    end_pos[0] = 5;
    end_pos[1] = 5;


        ret = ECAT_McGroupMoveLineAbsAdv(DeviceNo, GroupNo, end_pos,
                        start_vel, req_vel, final_vel, accel, decel, 0);
```

```
    if (ret != 0)
            printf("Failed to add group move line command:%d\n", ret);



        do
        {
            sleep(1);
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

        }while(State == MC_GS_MOVING) //Moving

        if(State == MC_GS_STANDBY) //Standby
            printf("Group move line successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
}
```
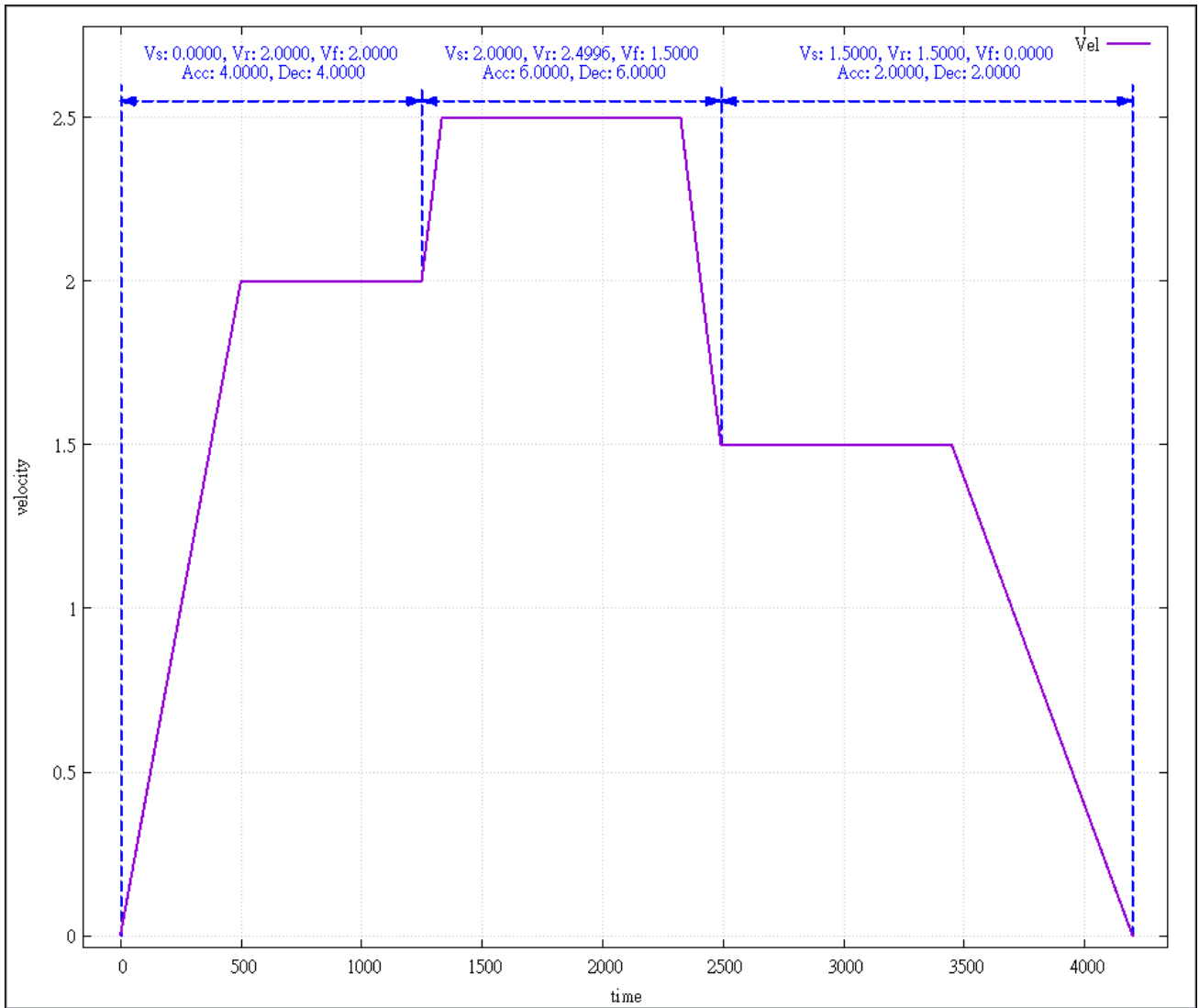
## 7.8.50.  ECAT_McGroupMoveLineRelAdv

**Description:**

Start a relative linear interpolation motion of a group.

**Syntax:**

int32_t ECAT_McGroupMoveLineRelAdv(uint16_t DeviceNo, uint16_t GroupNo, double EndPos[], double StartVel, double ReqVel, double FinalVel, double Accel, double Decel, uint8_t AccDecMode)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| EndPos | Double * | IN | Distance array of a group Each array element is the relative position of an axis. (Unit: user unit) |
| StartVel | double | IN | Start velocity (Unit: user unit/s) |
| ReqVel | double | IN | Target velocity (Unit: user unit/s) |
| FinalVel | double | IN | Final velocity (Unit: user unit/s) |
| Accel | double | IN | Acceleration rate ($user\ unit/s^2$) or acceleration time (second) |
| Decel | double | IN | Deceleration rate ($user\ unit/s^2$) or deceleration time (second) |
| AccDecMode | uint8_t | IN | Acceleration and deceleration input mode: 0: acceleration and deceleration rate ($user\ unit/s^2$) 1: acceleration and deceleration time (second) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t AxisNo;
uint32_t State;
double start_vel;
double req_vel;
double final_vel;
double accel;
double decel;
double end_pos[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_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    //Command 1
    start_vel = 0;
    req_vels = 2;
    final_vel = 2;
```

```
        accel = 4;
        decel = 4;
    end_pos[0] = 1.5;
    end_pos[1] = 1.5;

        ret = ECAT_McGroupMoveLineAbsAdv(DeviceNo, GroupNo, end_pos,
                        start_vel, req_vel, final_vel, accel, decel, 0);
        if (ret != 0)
                printf("Failed to add group move line command:%d\n", ret);

        //Command 2
        start_vel = 2;
        req_vels = 2.5;
        final_vel = 1.5;
        accel = 6;
        decel = 6;
    end_pos[0] = 3;
    end_pos[1] = 3;

        ret = ECAT_McGroupMoveLineRelAdv(DeviceNo, GroupNo, end_pos,
                        start_vel, req_vel, final_vel, accel, decel, 0);
        if (ret != 0)
                printf("Failed to add group move line command:%d\n", ret);

        //Command 3
        start_vel = 1.5;
        req_vels = 1.5;
        final_vel = 0;
        accel = 2;
        decel = 2;
    end_pos[0] = 1.5;
    end_pos[1] = 1.5;

        ret = ECAT_McGroupMoveLineRelAdv(DeviceNo, GroupNo, end_pos,
                        start_vel, req_vel, final_vel, accel, decel, 0);
        if (ret != 0)
                printf("Failed to add group move line command:%d\n", ret);
```

```
        do
        {
            sleep(1);
            ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);


        }while(State == MC_GS_MOVING) //Moving


        if(State == MC_GS_STANDBY) //Standby
            printf("Group move line successfully!\n");
        else if(State == MC_GS_ERRORSTOP) //ErrorStop
        {
            printf("Group error stop\n");
        }
    }
```
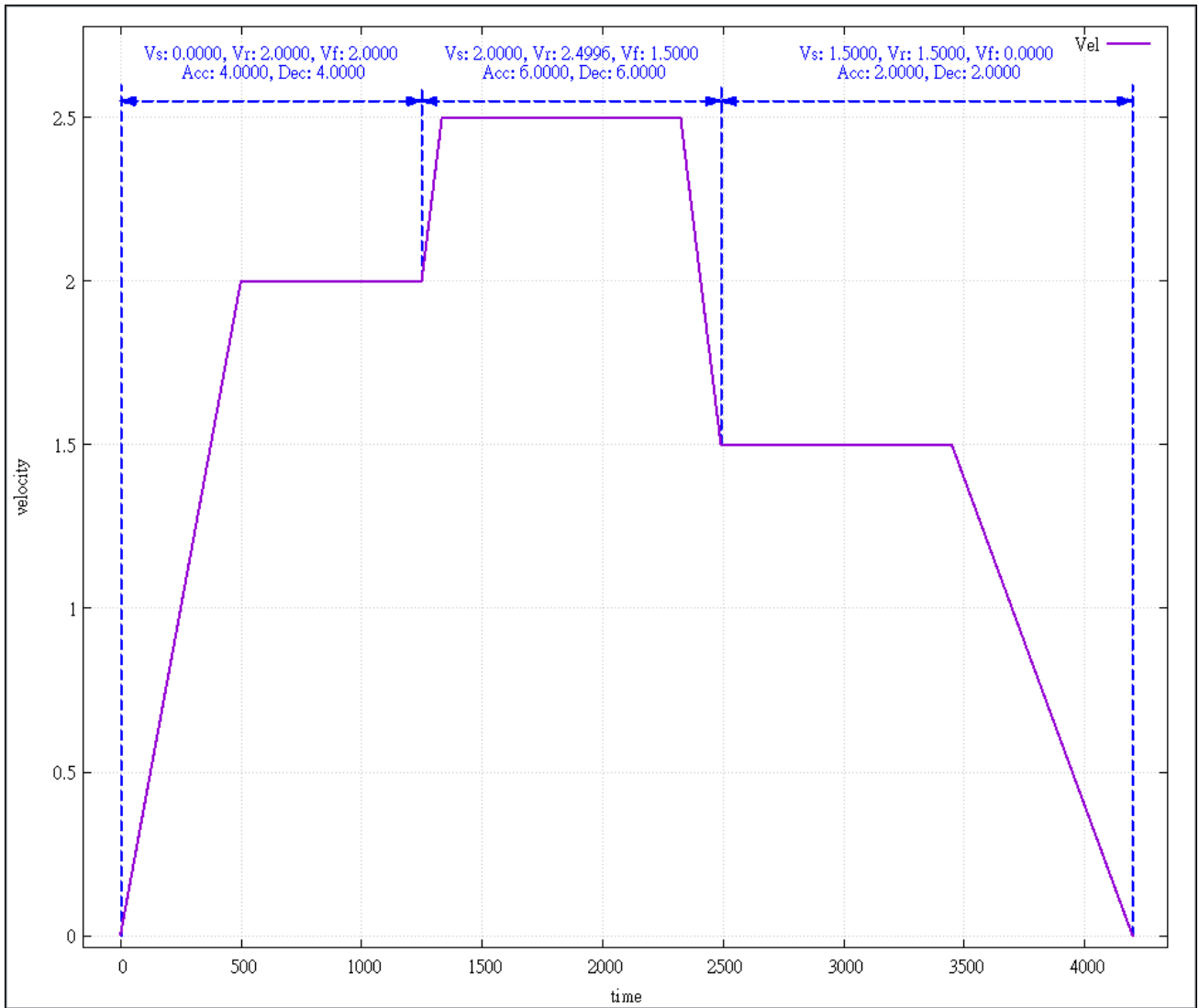
# 7.8.51.    ECAT_McGroupMoveShaker

**Description:**

This function is for factory use only


Start a relative sine wave motion of a group.

Note:(1) Can be used with some group commands

(2) Blending or buffer mode is not supported, it can be used in blending or buffer mode, but will abort relative sine wave movement in progress.

Y = Amp * sin(2 * pi * Freq * t + phase)

t = 0 to Time


**Syntax:**

int32_t  ECAT_McGroupMoveShaker(uint16_t  DeviceNo,  uint16_t  GroupNo,  double Amp, double phase[], double Freq, double Time);


**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Amp | double | IN | amplitude (Unit:user unit) |
| phase | double[] | IN | phase (Unit:degree) |
| Freq | double | IN | Frequency(Unit:hz) |
| Time | double | IN | Moving time (Unit:second) |


**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_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 Amp;
double phase[MC_AXIS_NO_MAX];
double Freq;
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
    Amp = 0.001;
    phase[0] =180;
    Freq = 20;
    Time = 1;
    ret = ECAT_McGroupMoveShaker(DeviceNo, GroupNo, Amp, phase, Freq, Time);
    if(ret < 0)
    {
        printf("Failed to add group move shaker command:%d\n", ret);
    }

    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

# 7.9. PID Controller
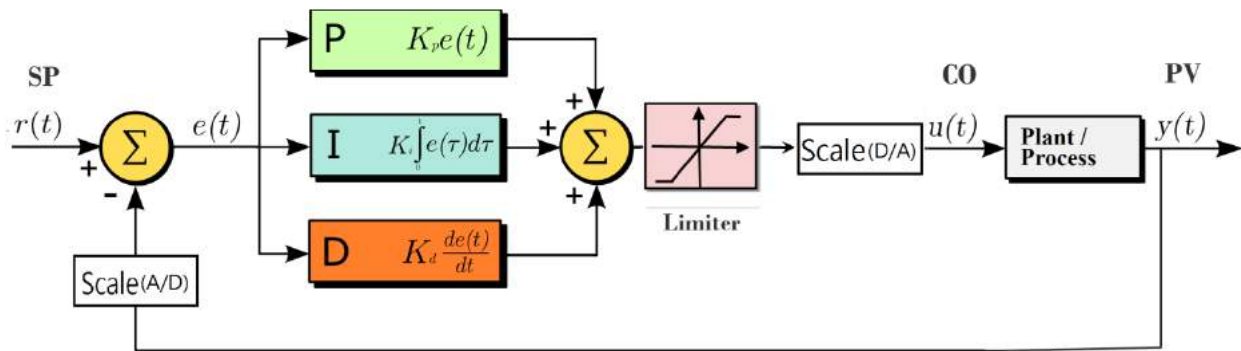


SP: SetPoint

CO: Controller Output

PV: Process Variable

e(t): SP-PV

Simulate Plant Model:

$$G\left(s\right)=\frac{1}{s+1}$$

Scale:



b = a* ScaleGain + ScaleOffset

## 7.9.1.   ECAT_PidGetSetPointValue

**Description:**

Get the Set Point Value.

**Syntax:**

int32_t ECAT_PidGetSetPointValue(uint16_t DeviceNo, uint32_t PidNo, double* SetPointValue)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| SetPointValue | double* | OUT | Set Point Value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0) printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0) printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0) printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_ PidSetStatus (DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);

ret = ECAT_PidGetSetPointValue(DeviceNo, PidNo, &SetPointValue);
if(ret != 0)
{
    printf("Failed to get Pid Set Point Value:%d\n", ret);
}
else
{
    printf("Pid Set Point Value %d\n", SetPointValue);
}
```

## 7.9.2.   ECAT_PidSetSetPointValue

**Description:**

Set the Set Point Value.

**Syntax:**

int32_t ECAT_PidSetSetPointValue(uint16_t DeviceNo, uint32_t PidNo, double SetPointValue)

**Parameters:**

| Name | Type | IN or OUT | Description |
| --- | --- | --- | --- |
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| SetPointValue | double | IN | Set Point Value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0) printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0) printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0)
{
    printf("Failed to set Pid Set Point Value:%d\n", ret);
}
```

## 7.9.3. ECAT_PidGetProcessVariable

**Description:**

Get the Process Variable.

**Syntax:**

int32_t ECAT_PidGetProcessVariable(uint16_t DeviceNo, uint32_t PidNo, double* ProcessVariable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| ProcessVariable | double* | OUT | Process Variable (or Process Value) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0) printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0) printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0) printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);

ret = ECAT_PidGetProcessVariable(DeviceNo, PidNo, &SetPointValue);
if(ret != 0)
{
    printf("Failed to get Pid Set Point Value:%d\n", ret);
}
else
{
    printf("Pid Set Point Value %d\n", SetPointValue);
}
```

## 7.9.4.   ECAT_PidGetSampleTime

**Description:**

Get the sampling time.

**Syntax:**

int32_t ECAT_PidGetSampleTime(uint16_t DeviceNo, uint32_t PidNo, uint32_t* Interval)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number |
| PidNo | uint32_t | IN | PID Controller number |
| Interval | int32_t* | Output | Sampling time<br>Unit: EtherCAT Cycle Time |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
int32_t Interval = 0;

ret = ECAT_PidGetSampleTime(DeviceNo, PidNo, &Interval);
if(ret != 0)
{
    printf("Failed to Get Pid Controller:%d\n", ret);
}
else
{
    printf("Pid Interval %d\n", Interval);
}
```

## 7.9.5. ECAT_PidSetSampleTime

**Description:**

Set the sampling time.

**Syntax:**

int32_t ECAT_PidSetSampleTime(uint16_t DeviceNo, uint32_t PidNo, uint32_t Interval)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| Interval | int32_t | IN | Sampling time<br>Unit: EtherCAT Cycle Time |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
int32_t Interval = 1;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0)
{
    printf("Failed to set Pid Controller:%d\n", ret);
}

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0) printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0) printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);
```

## 7.9.6.  ECAT_PidGetStatus

**Description:**

Get the controller status. It can be enabled or disabled.

**Syntax:**

int32_t ECAT_PidGetStatus(uint16_t DeviceNo, uint32_t PidNo, uint8_t *status)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| status | uint8_t* | Output | Status<br>0: disabled<br>1: enabled |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint8_t Status= 0;

ret = ECAT_PidGetStatus(DeviceNo, PidNo, &Status);
if(ret != 0)
{
    printf("Failed to Get Pid Status:%d\n", ret);
}
else
{
    printf("Pid Status Value %d\n", Status);
}
```

## 7.9.7. ECAT_PidSetStatus

**Description:**

Set PID Controller Status.

Note: Changing the status from **Enabled** to **Disabled** will not clear the output of the control output module. Users can set control output to whatever they like by using function _ECAT_SetSlaveRxPdoData_ if PID Controller Status is disabled. However, if the status is changed from **Disabled** to **Enabled, it** will set the output of the control output module to 0; then the controller start to work.

**Syntax:**

int32_t ECAT_PidSetStatus(uint16_t DeviceNo, uint32_t PidNo, uint8_t status)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| status | uint8_t | IN | Status<br>0: disabled<br>1: enabled |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0) printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0) printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0) printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)
{
    printf("Failed to Set Pid Status:%d\n", ret);
}
```

## 7.9.8.   ECAT_PidGetSimulateMode

**Description:**

Get simulation status. Use it to know whether the system is set for simulation or not.

**Syntax:**

int32_t ECAT_PidGetSimulateMode(uint16_t DeviceNo, uint32_t PidNo, uint8_t *status)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| status | uint8_t* | Output | Status<br>0: disabled<br>1: enabled |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint8_t Status= 0;

ret = ECAT_PidGetSimulateMode(DeviceNo, PidNo, &Status);
if(ret != 0)
{
    printf("Failed to Get Pid Simulate:%d\n", ret);
}
else
{
    printf("Pid Simulate %d\n", Status);
}
```

## 7.9.9.　ECAT_PidSetSimulateMode

**Description:**

Set simulation status. Use it to set whether the system is set for simulation or not.

Note: Changing the status from **Disable** to **Enable simulation** will clear the output of the control output module which is used for this PID controller. Users can set control output value by using function *ECAT_SetSlaveRxPdoData* if simulation is disabled.

**Syntax:**

int32_t ECAT_PidSetSimulateMode(uint16_t DeviceNo, uint32_t PidNo, uint8_t status)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| status | uint8_t | IN | Status<br>0: Disable simulation<br>1: Enable simulation |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0)
{
    printf("Failed to Set Pid Simulate:%d\n", ret);
}

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0) printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0) printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);
```

# 7.9.10. ECAT_PidGetParameter

**Description:**

Get the control parameters of a PID Controller.

**Syntax:**

int32_t ECAT_PidGetParameter(uint16_t DeviceNo, uint32_t PidNo, double *kp, double *ki, double *kd)

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number (0 ~ 9) |
| kp | double * | Output | Proportional control gain |
| ki | double * | Output | Integral control gain |
| kd | double * | Output | Derivative control gain |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double kp= 0;
double ki= 0;
double kd= 0;
ret = ECAT_PidGetParameter(DeviceNo, PidNo, &kp, &ki, &kd)
if(ret != 0)
{
    printf("Failed to Get Pid Parameter:%d\n", ret);
}
else
{
    printf("Pid Parameter : kp:%f , ki:%f , kd:%f \n", kp, ki, kd);
}
```

## 7.9.11. ECAT_PidSetParameter

**Description:**

Set the control parameters of a PID Controller.

**Syntax:**

int32_t ECAT_PidSetParameter(uint16_t DeviceNo, uint32_t PidNo, double kp, double ki, double kd)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number (0 ~ 9) |
| kp | double | IN | Proportional control gain |
| ki | double | IN | Integral control gain |
| kd | double | IN | Derivative control gain |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0) printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0)
{
    printf("Failed to Set Pid Parameter:%d\n", ret);
}

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0) printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);
```

## 7.9.12.  ECAT_PidGetProcessVariableModule

**Description:**

A Process Variable in a PID control loop is measured by an analog input channel in a module. In order to convert an analog input value to a physical value, users need to set scaling parameters. **SlaveNo**, **OffsetByte** and **Bitlength** parameters are used for the assignment of the module and its analog input channel.  **ScaleGain** and **ScaleOffset** paramters are used for data conversion. This function can get the settings of these parameters.

**Syntax:**

int32_t ECAT_PidGetProcessVariableModule(uint16_t DeviceNo, uint32_t PidNo, uint16_t* SlaveNo, uint16_t* OffsetByte, uint16_t* Bitlength, double* ScaleGain, double* ScaleOffset)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| SlaveNo | uint16* | OUT | Slave number |
| OffsetByte | uint16* | OUT | Byte offset |
| Bitlength | uint16* | OUT | Data Size, Unit: bit |
| ScaleGain | double* | OUT | Input Gain |
| ScaleOffset | double* | OUT | Input Offset |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint16_t SlaveNo = 0;
uint16_t Offset = 0;
uint16_t Bitsize = 16;
double Scalegain = 1;
double Scaleoffset = 0;

ret=ECAT_PidGetProcessVariableModule(DeviceNo, PidNo, &SlaveNo, &Offset, &Bitsize, &Scalegain,
&Scaleoffset);
if(ret != 0)
{
    printf("Failed to Get Pid Input:%d\n", ret);
}
```

## 7.9.13.  ECAT_PidSetProcessVariableModule

**Description:**

A Process Variable in a PID control loop is measured by an analog input channel in a module. In order to convert an analog input value to a physical value, users need to set scaling parameters. **SlaveNo**, **OffsetByte** and **Bitlength** paramters are used for the assignment of the module and its analog input channel.  **ScaleGain** and **ScaleOffset** paramters are used for data conversion. This function can set these settings.

**Syntax:**

int32_t   ECAT_PidSetProcessVariableModule(uint16_t   DeviceNo,   uint32_t   PidNo, uint16_t   SlaveNo,   uint16_t   OffsetByte,   uint16_t   Bitlength,double   ScaleGain,double ScaleOffset)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| SlaveNo | uint16 | IN | Slave number |
| OffsetByte | uint16 | IN | Byte offset |
| Bitlength | uint16 | IN | Data Size, Unit: bit |
| ScaleGain | double | IN | Input Gain |
| ScaleOffset | double | IN | Input Offset |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint16_t SlaveNo_input = 0;
uint16_t Offset_input = 0;
uint16_t Bitsize_input = 16;
double Scalegain_input = 1;
double Scaleoffset_input = 0;
uint16_t SlaveNo_output = 0;
uint16_t Offset_output = 2;
uint16_t Bitsize_output = 16;
double Scalegain_output = 1;
double Scaleoffset_output = 0;
double Max_Value = 0;
double Min_Value = 0;
int32_t Input = 5;
uint8_t Simulate= 0;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;

ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0) printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0) printf("Failed to Set Pid Parameter:%d\n", ret);

ret=ECAT_PidSetProcessVariableModule(DeviceNo, PidNo, SlaveNo_input, Offset_input, Bitsize_input, Scalegain_input, Scaleoffset_input);
if(ret != 0)
```

```
{
    printf("Failed to Set Pid Input:%d\n", ret);
}

ret= ECAT_PidSetControlOutputModule(DeviceNo, PidNo, SlaveNo_output, Offset_output, Bitsize_output,
Scalegain_output, Scaleoffset_output, Max_Value, Min_Value);
if(ret != 0) printf("Failed to Set Pid Output:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0)printf("Failed to Set Pid Status:%d\n", ret);
```

## 7.9.14.  ECAT_PidGetControlOutputModule

**Description:**

A Control Output in a PID control loop is sent to an analog output channel in an AO module. In order to convert a physical value to an analog output value, users need to set scaling parameters. **SlaveNo**, **OffsetByte** and **Bitlength** parameters are used for assignment of the module and its analog output channel.   **ScaleGain** and **ScaleOffset** parameters are used for data conversion. **Output_Max_Value** and **Output_Min_Value** parameters are used to limit the control output value. This function can get these settings.

**Syntax:**

int32_t ECAT_PidGetControlOutputModule(uint16_t DeviceNo, uint32_t PidNo, uint16_t* SlaveNo, uint16_t* OffsetByte, uint16_t* Bitlength, double* ScaleGain, double* ScaleOffset, double* Output_Max_Value, double* Output_Min_Value)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| SlaveNo | uint16* | OUT | Slave number |
| OffsetByte | uint16* | OUT | Byte offset |
| Bitlength | uint16* | OUT | Data Size, Unit: bit |
| ScaleGain | double * | OUT | Output Gain |
| ScaleOffset | double * | OUT | Output Offset |
| Output_Max_Value | double * | OUT | Output Maximum Value |
| Output_Min_Value | double * | OUT | Output Maximum Value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

## [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint16_t SlaveNo = 0;
uint16_t Offset = 0;
uint16_t Bitsize = 16;
double Scalegain = 1;
double Scaleoffset = 0;
double Max_Value = 0;
double Min_Value = 0;

ret=ECAT_PidGetControlOutputModule(DeviceNo, PidNo, &SlaveNo, &Offset, &Bitsize, &Scalegain,
&Scaleoffset, &Max_Value, &Min_Value);
if(ret != 0)
{
    printf("Failed to Get Pid Output:%d\n", ret);
}
```

## 7.9.15. ECAT_PidSetControlOutputModule

Description:

A Control Output in a PID control loop is sent to an analog output channel in an AO module. In order to convert a physical value to an analog output value, users need to set scaling parameters. **SlaveNo**, **OffsetByte** and **Bitlength** parameters are used for assignment of the module and its analog output channel. **ScaleGain** and **ScaleOffset** parameters are used for data conversion. **Output_Max_Value** and **Output_Min_Value** parameters are used to limit the control output value. This function can set these settings.

Syntax:

int32_t ECAT_PidSetControlOutputModule(uint16_t DeviceNo, uint32_t PidNo, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t Bitlength, double ScaleGain, double ScaleOffset, double Output_Max_Value, double Output_Min_Value)

Parameters:

| Name | Type | IN or OUT | Description |
| --- | --- | --- | --- |
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| SlaveNo | uint16 | IN | Slave number |
| OffsetByte | uint16 | IN | Byte offset |
| Bitlength | uint16 | IN | Data Size, Unit: bit |
| ScaleGain | double | IN | Output Gain |
| ScaleOffset | double | IN | Output Offset |
| Output_Max_Value | double | IN | Output Maximum Value |
| Output_Min_Value | double | IN | Output Maximum Value |

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
uint16_t SlaveNo_input = 0;
uint16_t Offset_input = 0;
uint16_t Bitsize_input = 16;
double Scalegain_input = 1;
double Scaleoffset_input = 0;
uint16_t SlaveNo_output = 0;
uint16_t Offset_output = 2;
uint16_t Bitsize_output = 16;
double Scalegain_output = 1;
double Scaleoffset_output = 0;
double Max_Value = 0;
double Min_Value = 0;
int32_t Input = 5;
uint8_t Simulate= 0;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;


ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);


ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0) printf("Failed to get Pid Simulate:%d\n", ret);


ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0) printf("Failed to Set Pid Parameter:%d\n", ret);


ret=ECAT_PidSetProcessVariableModule(DeviceNo, PidNo, SlaveNo_input, Offset_input, Bitsize_input,
Scalegain_input, Scaleoffset_input);
if(ret != 0) printf("Failed to Set Pid Input:%d\n", ret);
```

ret=ECAT_PidSetControlOutputModule(DeviceNo, PidNo, SlaveNo_output, Offset_output, Bitsize_output, Scalegain_output, Scaleoffset_output, Max_Value, Min_Value);
if(ret != 0)
{
    printf("Failed to Set Pid Output:%d\n", ret);
}
ret = ECAT_ PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);

## 7.9.16. ECAT_PidGetControlOutputValue

**Description:**

Get Control Output Value in a PID control loop.

**Syntax:**

int32_t ECAT_PidGetControlOutputValue(uint16_t DeviceNo, uint32_t PidNo, double* Output)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| Output | double* | OUT | Control Output Value |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Value= 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0) printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0) printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0) printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidGetControlOutputValue(DeviceNo, PidNo, &Value);
if(ret != 0)
{
    printf("Failed to Get Pid Output Value:%d\n", ret);
}
else
{
    printf("Pid OutputValue :%f \n", Value);
}
```

## 7.9.17. ECAT_PidGetSimulateFeedback

**Description:**

If the simulation is enabled for a PID control loop, this function can get the Control Output Value of this loop.

**Syntax:**

int32_t ECAT_PidGetSimulateFeedback(uint16_t DeviceNo, uint32_t PidNo, double* Feedback)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number |
| PidNo | uint32_t | IN | PID Controller number |
| Feedback | double* | Output | Control Output Value in a PID control loop with a simulation model as the process. |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```cpp
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Feedback = 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);


ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0) printf("Failed to get Pid Simulate:%d\n", ret);


ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0) printf("Failed to Set Pid Parameter:%d\n", ret);


ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);


ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0) printf("Failed to set Pid Set Point Value:%d\n", ret);


ret = ECAT_PidGetSimulateFeedback(DeviceNo, PidNo, &Feedback);
if(ret != 0)
{
    printf("Failed to Get Pid Simulate Feedback:%d\n", ret);
}
else
{
    printf("Pid Simulate Feedback:%f \n", Feedback);
}
```

## 7.9.18.  ECAT_PidGet_Sp_Err_Op_Pv

**Description:**

Get the Set Point Value, Error, Control Output, and Process Variable of a PID control system. Users can use this function to get these values back efficiently.

**Syntax:**

int32_t   ECAT_PidGet_Sp_Err_Op_Pv(uint16_t   DeviceNo,   uint32_t   PidNo,   double *SetPointValue, double *Error, double *OutputValue, double *ProcessVariable)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| PidNo | uint32_t | IN | PID Controller number |
| SetPointValue | double* | OUT | Set Point Value (SP) |
| Error | double* | OUT | Error ( = SP-PV) |
| OutputValue | double* | OUT | Control Output Value (CO) |
| ProcessVariable | double* | OUT | ProcessVariable (PV) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t PidNo = 0;
double Error= 0;
double ProcessVariable= 0;
double OutputValue= 0;
double SetPointValue = 5;
uint8_t Simulate= 1;
int32_t Interval = 1;
double kp = 1;
double ki = 1;
double kd = 0;
uint8_t enable = 1;
ret = ECAT_PidSetSampleTime(DeviceNo, PidNo, Interval);
if(ret != 0) printf("Failed to set Pid Controller:%d\n", ret);

ret = ECAT_PidSetSimulateMode(DeviceNo, PidNo, Simulate);
if(ret != 0) printf("Failed to get Pid Simulate:%d\n", ret);

ret = ECAT_PidSetParameter(DeviceNo, PidNo, kp, ki, kd)
if(ret != 0) printf("Failed to Set Pid Parameter:%d\n", ret);

ret = ECAT_PidSetStatus(DeviceNo, PidNo, enable);
if(ret != 0) printf("Failed to Set Pid Status:%d\n", ret);

ret = ECAT_PidSetSetPointValue(DeviceNo, PidNo, SetPointValue);
if(ret != 0) printf("Failed to set Pid Set Point Value:%d\n", ret);

ret = ECAT_PidGet_Sp_Err_Op_Pv(DeviceNo, PidNo, &SetPointValue, &Error, &OutputValue,
&ProcessVariable);
if(ret != 0)
{
    printf("Failed to Get Pid Sp_Err_Op_Pv:%d\n", ret);
}
else
```

```
{
    printf("Pid Set Point Value :%f \n", Setpoint);
    printf("Pid Error :%f \n", Error);
    printf("Pid OutputValue :%f \n", OutputValue);
    printf("Pid ProcessVariable:%f \n", ProcessVariable);
}
```

# 7.10. Stewart Platform
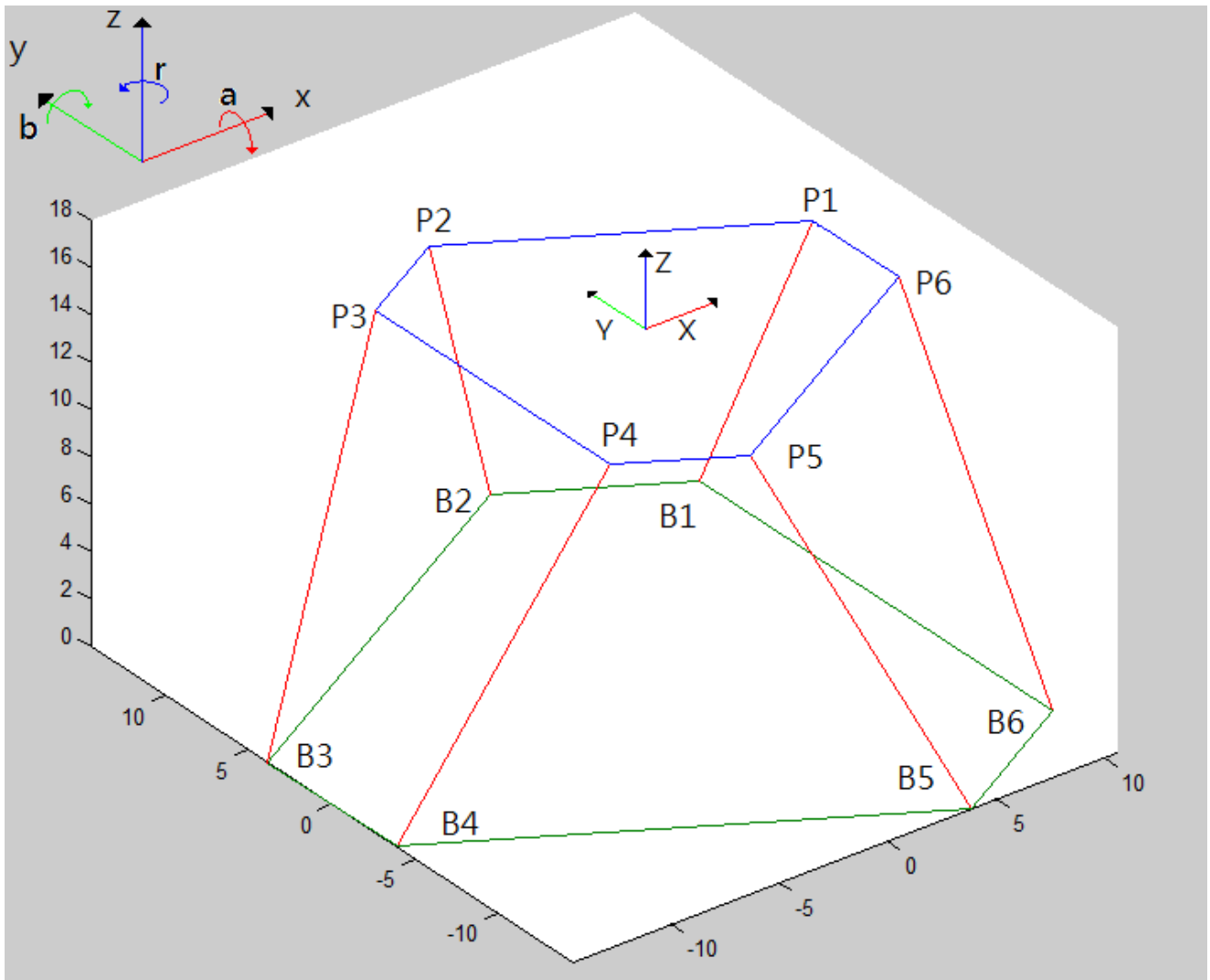


Figure 7.2

Top platform: the plane is formed by 6 Knots, P1 ~ P6

Base platform: the plane is formed by 6 Knots, B1 ~ B6
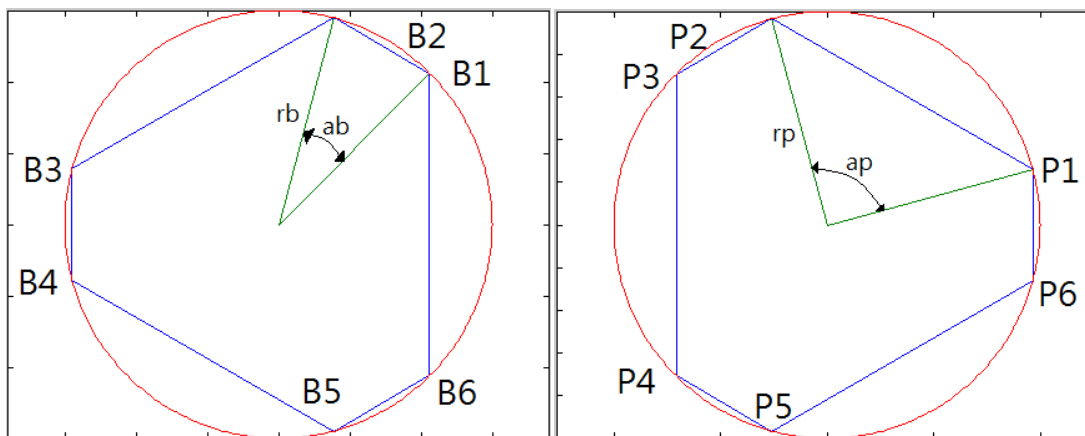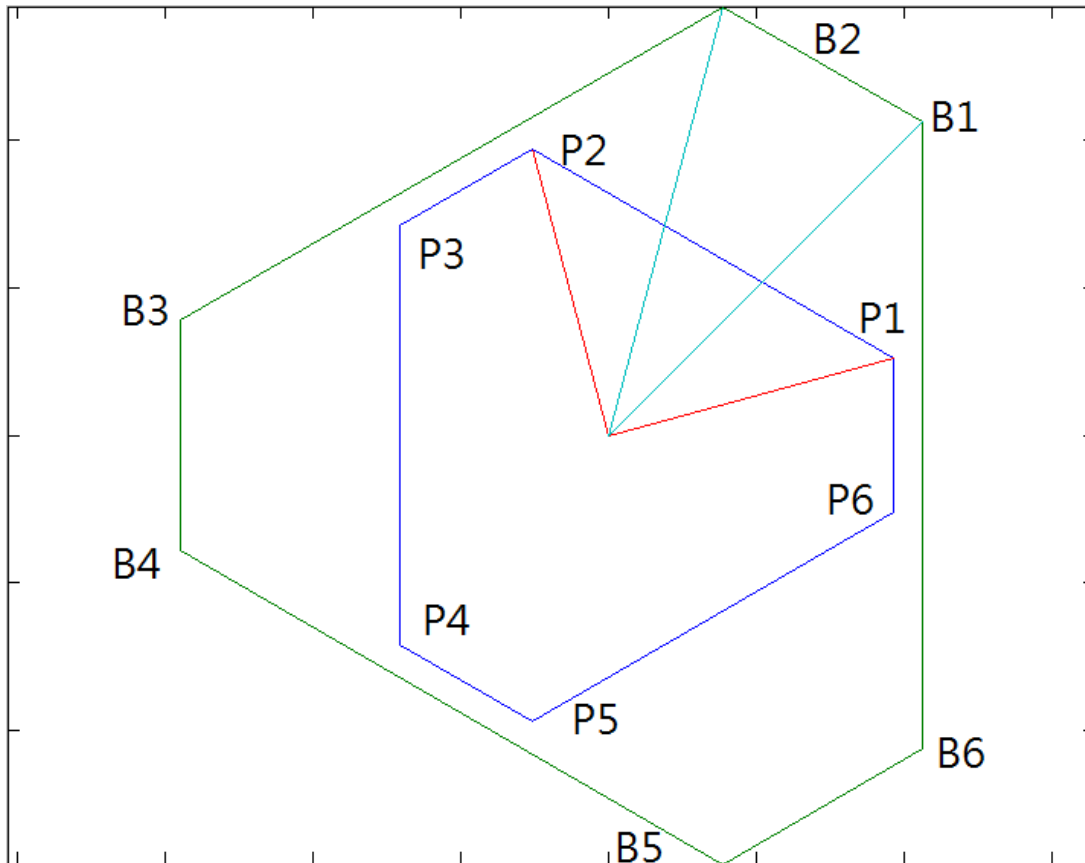
rb: Radius of the base platform

ab: The angle between B1, the center point of the base platform, and B2

rp: Radius of the top platform

ap: The angle between P1, the center point of the top platform, and P2

## 7.10.1. ECAT_McSetStewartPlatform_M1

**Description:**

Set geometric parameters for a Stewart platform (method 1).

**Syntax:**

int32_t ECAT_McSetStewartPlatform_M1(uint16_t DeviceNo, double radiusB, double angleB, double radiusP, double angleP, double RodLength, double Max_RodLength)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| radiusB | double | IN | Radius of the base platform. Unit: mm |
| angleB | double | IN | The angle formed by B1, the center point of the base platform, and B2 Unit: degree |
| radiusP | double | IN | Radius of the top platform, Unit: mm |
| angleP | double | IN | The angle formed by P1, the center point of the top platform, and P2 Unit: degree |
| RodLength | double | IN | Minimum length of rod connecting base and top platforms. Unit: mm |
| Max_RodLength | double | IN | Maximum length of rod connecting base and top platforms. Unit: mm |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
double radiusB = 15;
double angleB = 30;
double radiusP = 10;
double angleP = 90;
double RodLength = 15;
double Max_RodLength = 30;

ret = ECAT_McSetStewartPlatform_M1(DeviceNo, radiusB, angleB, radiusP, angleP, RodLength,
Max_RodLength);
if(ret < 0)
{
    printf("Failed to Set Stewart Platform:%d\n", ret);
    return;
}
```

## 7.10.2. ECAT_McSetStewartPlatform_M1

**Description:**

Get geometric parameters of a Stewart platform (method 1).

**Syntax:**

int32_t ECAT_McGetStewartPlatform_M1(uint16_t DeviceNo, double* radiusB, double* angleB, double* radiusP, double* angleP, double* RodLength, double* Max_RodLength)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| radiusB | double* | OUT | Radius of the base platform. Unit: mm |
| angleB | double* | OUT | The angle formed by B1, the center point of the base platform, and B2 Unit: degree |
| radiusP | double* | OUT | Radius of the top platform, Unit: mm |
| angleP | double* | OUT | The angle formed by P1, the center point of the top platform, and P2 Unit: degree |
| RodLength | double* | OUT | The minimum length of rod connecting base and top platforms. Unit: mm |
| Max_RodLength | double* | OUT | The maximum length of rod connecting base and top platforms. Unit: mm |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
double radiusB = 0;
double angleB = 0;
double radiusP = 0;
double angleP = 0;
double RodLength = 0;
double Max_RodLength = 0;

ret = ECAT_McGetStewartPlatform_M1(DeviceNo, &radiusB, &angleB, &radiusP, &angleP, &RodLength,
&Max_RodLength);
if(ret < 0)
{
    printf("Failed to Get Stewart Platform:%d\n", ret);
    return;
}
```

## 7.10.3.  ECAT_McSetStewartPlatform_M2

**Description:**

Set geometric parameters of a Stewart platform (method 2).

**Syntax:**

int32_t ECAT_McSetStewartPlatform_M2(uint16_t DeviceNo, double Bx[], double By[], double Px[], double Py[], double Z0, double RodLength[], double Max_RodLength[])

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| Bx | double[] | IN | An array contains 6 elements. Each value is the X Coordinate of Bi, i = 1~6, Unit: mm |
| By | double[] | IN | An array contains 6 elements. Each value is the Y Coordinate of Bi, i = 1~6, Unit: mm |
| Px | double[] | IN | An array contains 6 elements. Each value is the X Coordinate of Pi, i = 1~6, Unit: mm |
| Py | double[] | IN | An array contains 6 elements. Each value is the Y Coordinate of Pi, i = 1~6, Unit: mm |
| Z0 | double | IN | The vertical height of the top platform relative to the base platform. Unit: mm |
| RodLength | double[] | IN | The minimum length of rod connecting base and top platform. Unit: mm |
| Max_RodLength | double[] | IN | The maximum length of rod connecting base and top platform, Unit: mm |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
double Bx[6]= {10.6066,   3.8823,   -14.4889,   -14.4889,   3.8823,   10.6066};
double By[6]= {10.6066,   14.4889,   3.8823,   -3.8823,   -14.4889,   -10.6066};
double Px[6] = {9.6593,   -2.5882,   -7.0711,   -7.0711,   -2.5882,   9.6593};
double Py[6] = {2.5882,   9.6593,   7.0711,   -7.0711,   -9.6593,   -2.5882};
double Z0 = 14.1421;
double RodLength[6] = {15,   15,   15,   15,   15,   15 };
double Max_RodLength[6] = {30,   30,   30,   30,   30,   30};

ret = ECAT_McSetStewartPlatform_M2(DeviceNo, Bx, By, Px, Py, Z0, RodLength, Max_RodLength);
if(ret < 0)
{
    printf("Failed to Set Stewart Platform:%d\n", ret);
    return;
}
```

## 7.10.4.  ECAT_McGetStewartPlatform_M2

**Description:**

Get geometric parameters for a Stewart platform (method 2).

**Syntax:**

int32_t ECAT_McGetStewartPlatform_M2(uint16_t DeviceNo, double* Bx, double* By, double* Px, double* Py, double* Z0, double* RodLength, double* Max_RodLength)

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| Bx | double* | OUT | An array contains 6 elements. Each value is the X coordinate value of $B_i$, i = 1~6,    Unit: mm |
| By | double* | OUT | An array contains 6 elements. Each value is the Y coordinate value of $B_i$, i = 1~6,    Unit: mm |
| Px | double* | OUT | An array contains 6 elements. Each value is the X coordinate value of $P_i$, i = 1~6,    Unit: mm |
| Py | double* | OUT | An array contains 6 elements. Each value is the Y coordinate value of $P_i$, i = 1~6,    Unit: mm |
| Z0 | double* | OUT | The initial distance between the center of base platform and the center of top platform. Unit: mm |
| RodLength | double* | OUT | The minimum length of rod connecting the base platform and the top platform. Unit: mm |
| Max_RodLength | double* | OUT | The maximum length of rod connecting |

|  |  |  | the base platform and the top platform, Unit: mm |
| --- | --- | --- | --- |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
double Bx[6]= {0};
double By[6]= {0};
double Px[6] = {0};
double Py[6] = {0};
double Z0 = 0;
double RodLength[6] = {0};
double Max_RodLength[6] = {0};

ret = ECAT_McGetStewartPlatform_M2(DeviceNo, &Bx, &By, &Px, &Py, &Z0, &RodLength,
Max_RodLength);
if(ret < 0)
{
    printf("Failed to Get Stewart Platform:%d\n", ret);
    return;
}
```

## 7.10.5. ECAT_McStewartPlatformMoveAbs_PT

**Description:**

Start an absolute linear interpolation motion by providing world coordinate space positions and time for executing this motion command. This is a group motion command. The pose includes the 6-axis world coordinate space positions. A long-distance linear motion or circular motion can be approximated by many of these short-distance commands. ECAT-M801/ e-M901 has a 3000-depth command buffer. Users can send commands continuously to this card. If the command mode is Blending, this card will smoothly execute every desired motion command.

Note: At first, this card will process pose command obtain the targeted joint space positions by processing the inverse kinematics. Then a 6-axis linear interpolation motion in joint space is implemented for this motion. Actually, the linear interpolation is not implemented in the world coordinate system. Only continuous short-distance commands can approach nearly linear commands.

**Syntax:**

int32_t ECAT_McStewartPlatformMoveAbs_PT(uint16_t DeviceNo, uint16_t GroupNo, double Pose[], double* Pos, double Time)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| GroupNo | uint16_t | IN | Group number |
| Pose | double[] | IN | Requested pose in world coordinate system of the Stewart platform<br>x: the displacement along X-axis.<br>Unit: mm<br>y: the displacement along Y-axis.<br>Unit: mm<br>z: the displacement along Z-axis.<br>Unit: mm |

| | | | |
|---|---|---|---|
| | | | a: the rotating angle around the X-axis. Unit: degree b: the rotating angle around the Y-axis. Unit: degree r : the rotating angle around the Z-axis. Unit: degree Please refer to Figure 7.2 for the direction definitions for displacement and rotation. |
| Pos | double* | OUT | This array contains the targeting 6-axis joint space positions. Each element in this array is an absolute position. Unit: user unit |
| Time | double | IN | Time Unit: second |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t GroupNo = 0;
uint16_t i;
uint32_t State;
uint16_t CmdMode = MS_GRP_CM_BLENDING; //0: Aborting, 1: Buffered, 2: Blending
double StewartPlatformPose[6]; //x  y  z  a  b  r
double Pos[6]; //position of axis0~axis5
double GroupTime;
double radiusB = 15;
double angleB = 30;
double radiusP = 10;
double angleP = 90;
double RodLength = 15;
double Max_RodLength = 30;

ret = ECAT_McSetStewartPlatform_M1(DeviceNo, radiusB, angleB, radiusP, angleP, RodLength,
Max_RodLength);
if(ret < 0)
{
    printf("Failed to Set Stewart Platform:%d\n", ret);
    return;
}

for(i=0;i<6;i++)//6-axis Stewart Platform
{
    ret = ECAT_McAddAxisToGroup(DeviceNo, GroupNo, i);
    if(ret < 0)
    {
        printf("Failed to add axis to group:%d\n", ret);
        return;
    }
}

ret = ECAT_McSetGroupCmdMode(DeviceNo, GroupNo, CmdMode);
```

```
if(ret < 0)
{
    printf("Failed to set group command mode:%d\n", ret);
    return;
}

ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);
if(State == MC_GS_STANDBY) //Standby
{
    StewartPlatformPose [0] = 0; // x
    StewartPlatformPose [1] = 0; // y
    StewartPlatformPose [2] = 1; // z
    StewartPlatformPose [3] = 0; // a
    StewartPlatformPose [4] = 0; // b
    StewartPlatformPose [5] = 0; // r
    GroupTime = 1;
    ret = ECAT_McStwartPlatformMoveAbs_PT(DeviceNo, GroupNo, StewartPlatformPose, &Pos,
GroupTime);
    if(ret < 0)
    {
        printf("Failed to add group move line command:%d\n", ret);
    }
    do
    {
        sleep(1);
        ret = ECAT_McGetGroupState(DeviceNo, GroupNo, &State);

    }while(State == MC_GS_MOVING) //Moving

    if(State == MC_GS_STANDBY) //Standby
        printf("Group move line successfully!\n");
    else if(State == MC_GS_ERRORSTOP) //ErrorStop
    {
        printf("Group error stop\n");
    }
}
```

# 7.11. Motion Data Recorder

## 7.11.1. ECAT_McSetMotionRecord

**Description:**

This function can start or stop an ECAT-M801/ e-M901 to record the position and/or velocity of axes. Inside the ECAT-M801/ e-M901, the program can save a record for each cycle time. Up to 100,000 records can be saved.

Note: This function will not clear record count to 0. Users can clear record count with function *ECAT_McClearMotionRecord*.

**Syntax:**

int32_t ECAT_McSetMotionRecord(uint16_t DeviceNo, uint16_t state)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| state | uint16_t | IN | 1: Start recording data<br>0: Stop recording data |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Example:**

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
ret = ECAT_McSetMotionRecord(DeviceNo, 1);
if(ret < 0)
{
    printf("Failed to Set Motion Record:%d\n", ret);
}
else
{
    printf("Set Motion Record successfully! \n");
}
```

## 7.11.2. ECAT_McGetMotionRecordState

**Description:**

Get the recording status.

**Syntax:**

int32_t ECAT_McGetMotionRecordState(uint16_t DeviceNo, uint16_t *state, uint32_t *count)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| state | uint16_t* | OUT | Recording or not<br>1: Recording<br>0: Not recording |
| count | uint32_t* | OUT | Count of recorded data |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t state;
uint32_t cnt;
ret = ECAT_McGetMotionRecordState(DeviceNo, &state, &cnt);
if(ret < 0)
{
    printf("Failed to Get Motion Record State: %d\n", ret);
}
else
{
    printf("State: %u , Count: %u \n", state, cnt);
}
```

### 7.11.3.  ECAT_McClearMotionRecord

**Description:**

Clear the counting index to 0. If recording is enabled, the counting number is started from the current counting index instead of always counting from 0.

**Syntax:**

int32_t ECAT_McClearMotionRecord(uint16_t DeviceNo)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
ret = ECAT_McClearMotionRecord(DeviceNo);
if(ret < 0)
{
    printf("Failed to Clear Motion Record:%d\n", ret);
}
```

## 7.11.4. ECAT_McSetMotionRecordParam

**Description:**

　　Set parameters for deciding which two out of four values are going to be recorded. Please refer to Table 7.16, the candidated four values are Actual Position, Actual Velocity, Command Position, and Command Velocity.

**Syntax:**

　　int32_t ECAT_McSetMotionRecordParam(uint16_t DeviceNo, uint16_t Value1, uint16_t Value2)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| Value1 | uint16_t | IN | The first motion parameter for recording (Refer to Table 7.16) |
| Value2 | uint16_t | IN | The second motion parameter for recording (Refer to Table 7.16) |

**Return:**

　　0: Success.

　　Others: Refer to Appendix "Error Codes".

Table 7.16 Motion parameters for recording

| Macro Definition | Value | Description |
|---|---|---|
| MC_RECORD_POSITION | 0 | Actual Position of Axis (Unit: user unit) |
| MC_RECORD_VELOCITY | 1 | Actual Velocity of Axis (Unit: user unit/second) |
| MC_RECORD_COMMAND_POSITION | 2 | Command Position of Axis (Unit: user unit) |
| MC_RECORD_COMMAND_VELOCITY | 3 | Command Velocity of Axis (Unit: user unit/second) |

**Example:**

**[C/C++]**

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t Value1= MC_RECORD_POSITION;
uint16_t Value2= MC_RECORD_VELOCITY;

ret = ECAT_McSetMotionRecordParam(DeviceNo, Value1, Value2);
if(ret < 0)
{
    printf("Failed to set motion record parameters:%d\n", ret);
}
ret = ECAT_McSetMotionRecord(DeviceNo, 1);
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n", ret);
```

## 7.11.5. ECAT_McGetMotionRecordParam

**Description:**

Get the settings of the recorded parameters.

**Syntax:**

int32_t ECAT_McGetMotionRecordParam(uint16_t DeviceNo, uint16_t *Value1, uint16_t *Value2)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| Value1 | uint16_t* | OUT | The first motion parameter for recording (Refer to Table 7.16) |
| Value2 | uint16_t* | OUT | The second motion parameter for recording (Refer to Table 7.16) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t *Value1;
uint16_t *Value2;

ret = ECAT_McGetMotionRecordParam(DeviceNo, &Value1, &Value2);
if(ret < 0)
{
    printf("Failed to get motion record parameters:%d\n", ret);
}
else
{
    printf("Value1:%u , Value2:%u \n", Value1, Value2);
}
```

## 7.11.6. ECAT_McGetMotionRecordValue

**Description:**

Get parameter values of an assigned axis at an assigned index number.

Note: When the AxisNo is set to 65535, values of all axes at the assigned index number are returned by Value1and Value2 pointers.

**Syntax:**

int32_t ECAT_McGetMotionRecordValue(uint16_t DeviceNo, uint32_t CountNo, uint16_t AxisNo, float *Value1, float *Value2)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| CountNo | uint32_t | IN | Count Number (an index number) |
| AxisNo | uint16_t | IN | Axis Number |
| Value1 | float* | OUT | Value of the first parameter recorded at the specified Count Number |
| Value2 | float* | OUT | Value of the second parameter recorded at the specified Count Number |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t CountNo= 0;
uint16_t AxisNo = 0;
uint16_t state;
uint32_t cnt;
int i;
uint16_t Param1= MC_RECORD_POSITION;
uint16_t Param2= MC_RECORD_VELOCITY;
float Value1;
float Value2;

ret = ECAT_McSetMotionRecordParam(DeviceNo, Param1, Param2);
if(ret < 0)
    printf("Failed to set motion record parameters:%d\n", ret);
ret = ECAT_McSetMotionRecord(DeviceNo,1); // start record
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n", ret);

Sleep(1000); //wait for record something…

ret = ECAT_McSetMotionRecord(DeviceNo, 0); // stop record
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n", ret);

ret = ECAT_McGetMotionRecordState(DeviceNo, &state, &cnt); // get record count
if(ret < 0)
    printf("Failed to Get Motion Record State:%d\n", ret);
for(i=0;i< cnt;i++)
{
    ret = ECAT_McGetMotionRecordValue(DeviceNo, i , AxisNo, &Value1, &Value2);
    if(ret < 0)
    {
        printf("Failed to get motion record value:%d\n", ret);
    }
```

```
        else
        {
            printf("Axis Value1:%f , Value2:%f \n", Value1, Value2);
        }
    }
    ret = ECAT_McClearMotionRecord(DeviceNo);
    if(ret < 0)
        printf("Failed to Clear Motion Record:%d\n", ret);


}
```

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t CountNo= 0;
uint16_t AxisNo = 65535;
uint16_t state;
uint32_t cnt;
int i;
uint16_t Param1= MC_RECORD_POSITION;
uint16_t Param2= MC_RECORD_VELOCITY;
float Value1[MC_AXIS_NO_MAX];
float Value2[MC_AXIS_NO_MAX];

ret = ECAT_McSetMotionRecordParam(DeviceNo, Param1, Param2);
if(ret < 0)
    printf("Failed to set motion record parameters:%d\n", ret);
ret = ECAT_McSetMotionRecord(DeviceNo,1); // start record
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n", ret);

Sleep(1000); //wait for record something…

ret = ECAT_McSetMotionRecord(DeviceNo, 0); // stop record
if(ret < 0)
```

```
            printf("Failed to Set Motion Record:%d\n", ret);


    ret = ECAT_McGetMotionRecordState(DeviceNo, &state, &cnt); // get record count
    if(ret < 0)
            printf("Failed to Get Motion Record State:%d\n", ret);
    for( i=0; i< cnt; i++)
    {
            ret = ECAT_McGetMotionRecordValue(DeviceNo, i , AxisNo, Value1, Value2);
            if(ret < 0)
            {
                    printf("Failed to get motion record value:%d\n", ret);
            }
            else
            {
                    for( j=0; j< MC_AXIS_NO_MAX; j++)
                    {
                            printf("Axis Value1:%f , Value2:%f \n", Value1[ j ] , Value2[ j ]);
                    }
            }
    }
    ret = ECAT_McClearMotionRecord(DeviceNo);
    if(ret < 0)
            printf("Failed to Clear Motion Record:%d\n", ret);


    }
```

## 7.11.7. ECAT_McGetMotionRecordValueEx

**Description:**

Get parameter values of an axis starting from an assigned index number. This function is able to get more records than *ECAT_McGetMotionRecordValue*. This function can get up to 64 records each time rather than only one record by *ECAT_McGetMotionRecordValue*.

**Syntax:**

int32_t ECAT_McGetMotionRecordValueEx(uint16_t DeviceNo, uint32_t CountNo, uint16_t Count, uint16_t AxisNo, float *Value1, float *Value2, uint16_t *ActualCount)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| CountNo | uint32_t | IN | Starting Count Number |
| Count | uint16_t | IN | Quantity of records to get, Max: 64 |
| AxisNo | uint16_t | IN | Axis Number |
| Value1 | float* | OUT | Array values of the first parameter recorded starting from the specified Count Number |
| Value2 | float* | OUT | Array values of the second parameter recorded starting from the specified Count Number |
| ActualCount | uint16_t* | OUT | Actual quantity of records gotten |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint32_t CountNo= 0;
uint16_t AxisNo = 0;
uint16_t state;
uint32_t cnt;
int I,j;
uint16_t Param1= MC_RECORD_POSITION;
uint16_t Param2= MC_RECORD_VELOCITY;
float Value1[RECORDDATA_GET_COUNT_MAX];
float Value2[RECORDDATA_GET_COUNT_MAX];
uint16_t *ActualGetCount;

ret = ECAT_McSetMotionRecordParam(DeviceNo, Param1, Param2);
if(ret < 0)
    printf("Failed to set motion record parameters:%d\n", ret);
ret = ECAT_McSetMotionRecord(DeviceNo,1); // start record
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n", ret);

Sleep(1000); //wait for record something…

ret = ECAT_McSetMotionRecord(DeviceNo, 0); // stop record
if(ret < 0)
    printf("Failed to Set Motion Record:%d\n", ret);

ret = ECAT_McGetMotionRecordState(DeviceNo, &state, &cnt); // get record count
if(ret < 0)
    printf("Failed to Get Motion Record State:%d\n", ret);
for(i=0; i< cnt; i+= RECORDDATA_GET_COUNT_MAX)
{
    ret = ECAT_McGetMotionRecordValueEx(DeviceNo, i , RECORDDATA_GET_COUNT_MAX, AxisNo,
Value1, Value2, &ActualGetCount);
    if(ret < 0)
    {
```

```
                printf("Failed to get motion record value:%d\n", ret);

        }
        else
        {
            for(j=0; j< ActualGetCount; j++)
            {
                printf("Axis Value1:%f , Value2:%f \n", Value1[ j], Value2[ j]);
            }
        }
    }
    ret = ECAT_McClearMotionRecord(DeviceNo);
    if(ret < 0)
        printf("Failed to Clear Motion Record:%d\n", ret);


}
```

# 7.12. Event

The way to check for state changes in the control card on the PC is to read the data back for inspection. But this way will consume a lot of CPU time of the PC. In order to reduce the burden on the PC and speed up the response of the system, there is a method for providing an event notification to the PC in the ECAT-M801/ e-M901. The programmer sets the conditions for the event in advance, and then allows the program to enter a wait state. While waiting, the program (or thread) does not occupy the CPU resources of the PC. The system will wake up the waiting program after specified event occurs.

Currently, the conditions for triggering events have position comparison, single DI changes, multiple DI changes, and motion status checks. Up to 32 trigger events can be set. Basically, the trigger condition is automatically disabled (disabled). If the event trigger is going to be used again, it must be set to enabled again in the event processing program.

*ECAT_SetTimer* API is actually a timer event; but this event is somewhat different from the events mentioned above. Once a timer event is enabled, it will continue to fire periodically, no need to reset it again. However, the events here must be set again in order to be used continuously.

An event can be used in the program to set or enable another event of different properties when the triggered event is processing. This allows a system to perform a series of complex actions.

## 7.12.1. ECAT_EvEnableEvent

**Description:**

Enable an event.

Note: After an event is triggered, it will become disabled.

**Syntax:**

int32_t ECAT_EvEnableEvent(uint16_t DeviceNo, uint16_t EventID)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EventID | uint16_t | IN | Event ID. It can be 0 ~ 31. |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
{
    printf("Failed to enable event:%d\n", ret);
}
else
{
    printf("Enable event successfully!\n");
}
```

## 7.12.2. ECAT_EvDisableEvent

**Description:**

Disable an event.

**Syntax:**

int32_t ECAT_EvDisableEvent(uint16_t DeviceNo, uint16_t EventID)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EventID | uint16_t | IN | Event ID. It can be 0 ~ 31. |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

[C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
ret = ECAT_EvDisableEvent(DeviceNo, EventID);
if(ret < 0)
{
    printf("Failed to disable event:%d\n", ret);
}
else
{
    printf("Disable event successfully!\n");
}
```

## 7.12.3. ECAT_WaitforEvent

**Description:**

Program is blocked until the specified event is triggered or time out occurs.

**Syntax:**

int32_t ECAT_WaitforEvent(uint16_t DeviceNo, uint32_t TimeOut, uint32_t
*TriggeredEvent)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| TimeOut | uint16_t | IN | TimeOut , Unit: ms |
| TriggeredEvent | uint32_t* | IN | Events are triggered Note: There may be multiple events triggered at the same time. |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID0 = 0;
uint16_t EventID1 = 1;
uint32_t Value= 0;
ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n", ret);
}
else
{
    if(((Value>>(int(EventID0))) & (0x01)) == 1) //EventID 0 triggered
    {
        // do something…
    }
    if(((Value>>(int(EventID1))) & (0x01)) == 1) //EventID 1 triggered
    {
        // do something…
    }

}
```

## 7.12.4. ECAT_AbortWaitforEvent

**Description:**

Use this function to force ECAT_WaitforEvent to return 0 directly.

**Syntax:**

int32_t ECAT_AbortWaitforEvent(uint16_t DeviceNo)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

[C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
ret = ECAT_AbortWaitforEvent(DeviceNo);
if(ret != 0)
{
    printf("Failed to abort wait event:%d\n",ret);
}
```

## 7.12.5.  ECAT_EvSetComparePositionParameters

**Description:**

Set event parameters for a position comparison event.

**Syntax:**

int32_t ECAT_EvSetComparePositionParameters(uint16_t DeviceNo, uint16_t EventID, uint16_t AxisNo, uint16_t Operator, double ComparePosition)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EventID | uint16_t | IN | Event ID. It can be 0 ~ 31. |
| AxisNo | uint16_t | IN | Axis number |
| Operator | uint16_t | IN | Operator number (defined below) |
| ComparePosition | double | IN | Real position for the comparison |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Operator number:**

| Macro Definition | Value | Description |
|------------------|-------|-------------|
| GREATER_THAN | 0 | position greater than compare position |
| GREATER_THAN_OR_EQUAL_TO | 1 | position greater than or equal to compare position |
| LESS_THAN | 2 | position less than compare position |
| LESS_THAN_OR_EQUAL_TO | 3 | position less than or equal to compare position |

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t AxisNo = 0;
uint16_t Operator = GREATER_THAN;
double ComparePosition = 100;
uint32_t Value;

ret = ECAT_EvSetComparePositionParameters(DeviceNo, EventID, AxisNo, Operator, ComparePosition);
if(ret < 0)
{
    printf("Failed to set compare position parameters:%d\n", ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n", ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n", ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something…
    }
}
```

## 7.12.6. ECAT_EvSetCompareCmdPositionParameters

**Description:**

Set event parameters for a command position comparison event.

**Syntax:**

int32_t ECAT_EvSetCompareCmdPositionParameters(uint16_t DeviceNo, uint16_t EventID, uint16_t AxisNo, uint16_t Operator, double ComparePosition)

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EventID | uint16_t | IN | Event ID. It can be 0 ~ 31. |
| AxisNo | uint16_t | IN | Axis number |
| Operator | uint16_t | IN | Operator number (defined below) |
| ComparePosition | double | IN | Real position for the comparison |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Operator number:**

| Macro Definition | Value | Description |
|---|---|---|
| GREATER_THAN | 0 | position greater than compare position |
| GREATER_THAN_OR_EQUAL_TO | 1 | position greater than or equal to compare position |
| LESS_THAN | 2 | position less than compare position |
| LESS_THAN_OR_EQUAL_TO | 3 | position less than or equal to compare position |

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t AxisNo = 0;
uint16_t Operator = GREATER_THAN;
double ComparePosition = 100;
uint32_t Value;

ret = ECAT_EvSetCompareCmdPositionParameters(DeviceNo, EventID, AxisNo, Operator, ComparePosition);
if(ret < 0)
{
    printf("Failed to set compare command position parameters:%d\n", ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n", ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.// It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n", ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something…
    }
}
```

## 7.12.7. ECAT_EvSetCompareDIBitParameters

**Description:**

Set event parameters for a DI-BIT comparison event.

**Syntax:**

int32_t ECAT_EvSetCompareDIBitParameters(uint16_t DeviceNo, uint16_t EventID, uint16_t SlaveNo, uint16_t BitNo, uint32_t CompareValue)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EventID | uint16_t | IN | Event ID. It can be 0 ~ 31. |
| SlaveNo | uint16_t | IN | Slave number |
| BitNo | uint16_t | IN | bit number |
| CompareValue | uint32_t | IN | Compare Value<br>Event is triggered according to following definition.<br>0: DI bit value from 1 to 0 (falling edge)<br>1: DI bit value from 0 to 1 (rising edge)<br>2: DI bit value from 1 to 0 or from 0 to 1 (both falling and rising edges) |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t SlaveNo = 0;
uint16_t BitNo = 0;
uint32_t CompareValue = 1;
uint32_t Value;

ret = ECAT_EvSetCompareDIBitParameters(DeviceNo, EventID, SlaveNo, BitNo, CompareValue);
if(ret < 0)
{
    printf("Failed to set compare DI Bit parameters:%d\n", ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n", ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n", ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something…
    }
}
```

## 7.12.8. ECAT_EvSetCompareDIParameters

**Description:**

Set event parameters for multiple DI comparison event.

**Syntax:**

int32_t ECAT_EvSetCompareDIParameters(uint16_t DeviceNo, uint16_t EventID, uint16_t SlaveNo, uint16_t OffsetByte, uint32_t CompareValue, uint32_t Mask)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EventID | uint16_t | IN | Event ID. It can be 0 ~ 31. |
| SlaveNo | uint16_t | IN | Slave number |
| OffsetByte | uint16_t | IN | Byte offset |
| CompareValue | uint32_t | IN | Compare Value<br>Event is triggered while the specified DI value is changed from not this **CompareValue** to this value. |
| Mask | uint32_t | IN | Mask of DI value for comparison<br>The real DI value for comparison is defined as (DI & Mask). |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t SlaveNo = 0;
uint16_t OffsetByte = 0;
uint32_t CompareValue = 1;
uint32_t Mask = 1;
uint32_t Value;

ret = ECAT_EvSetCompareDIParameters(DeviceNo, EventID, SlaveNo, OffsetByte, CompareValue, Mask);
if(ret < 0)
{
    printf("Failed to set compare DI parameters:%d\n", ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n", ret);
ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n", ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something…
    }
}
```

## 7.12.9. ECAT_EvSetCompareAxisStateParameters

### Description:

Set event parameters as for checking an Axis state.

### Syntax:

int32_t ECAT_EvSetCompareAxisStateParameters(uint16_t DeviceNo, uint16_t EventID, uint16_t AxisNo, uint32_t CompareState)

### Parameters:

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EventID | uint16_t | IN | Event ID. It can be 0 ~ 31. |
| AxisNo | uint16_t | IN | Axis number |
| CompareState | uint32_t | IN | Compare Axis State<br>Please refer to Table 7.17 for axis state definitions. |

### Return:

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t AxisNo = 0;
uint32_t CompareState = MC_AS_STANDSTILL;
uint32_t Value;

ret = ECAT_EvSetCompareAxisStateParameters(DeviceNo, EventID, AxisNo, CompareState);
if(ret < 0)
{
    printf("Failed to set compare status parameters:%d\n", ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n", ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n", ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something…
    }
}
```

## 7.12.10.    ECAT_EvSetMotionCompleteParameters

**Description:**

Set event parameters as for checking an Axis motion done.

**Syntax:**

int32_t ECAT_EvSetMotionCompleteParameters(uint16_t DeviceNo, uint16_t EventID, uint16_t AxisNo)

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EventID | uint16_t | IN | Event ID. It can be 0 ~ 31. |
| AxisNo | uint16_t | IN | Axis number |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t AxisNo = 0;
uint32_t Value;

ret = ECAT_EvSetMotionCompleteParameters(DeviceNo, EventID, AxisNo);
if(ret < 0)
{
    printf("Failed to set motion complete parameters:%d\n", ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n", ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n", ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something…
    }
}
```
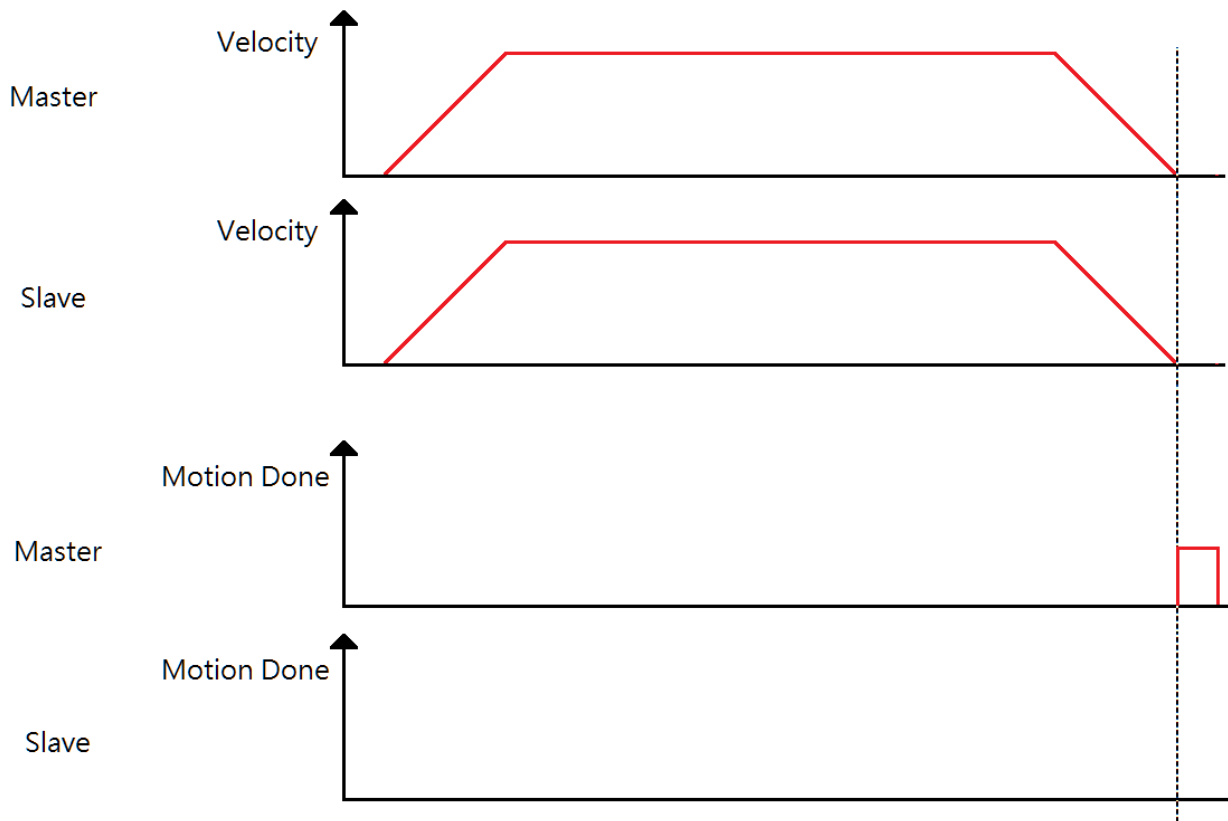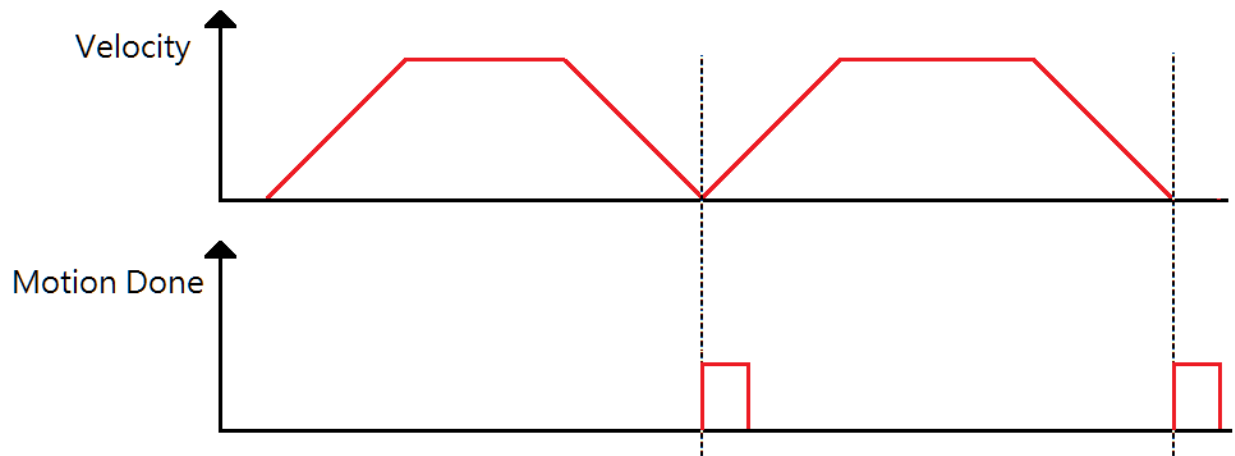
Single Axis motion:
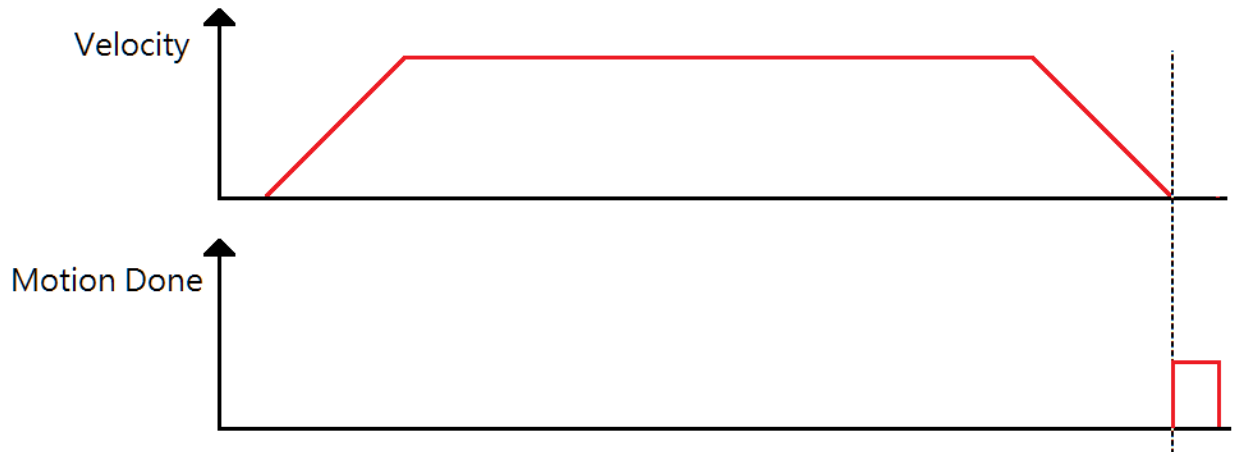
gear/ cam/ gantry:



Single axis motion: buffer mode

## 7.12.11.　ECAT_EvSetMotionCompleteParameters_Grp

**Description:**

Set event parameters as for checking a Group motion done.

**Syntax:**

int32_t ECAT_EvSetMotionCompleteParameters_Grp(uint16_t DeviceNo, uint16_t EventID, uint16_t GrpNo)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EventID | uint16_t | IN | Event ID. It can be 0 ~ 31. |
| GrpNo | uint16_t | IN | Group number |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t GroupNo = 0;
uint32_t Value;

ret = ECAT_EvSetMotionCompleteParameters_Grp(DeviceNo, EventID, GroupNo);
if(ret < 0)
{
    printf("Failed to set motion complete parameters:%d\n",ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n",ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something…
    }
}
```
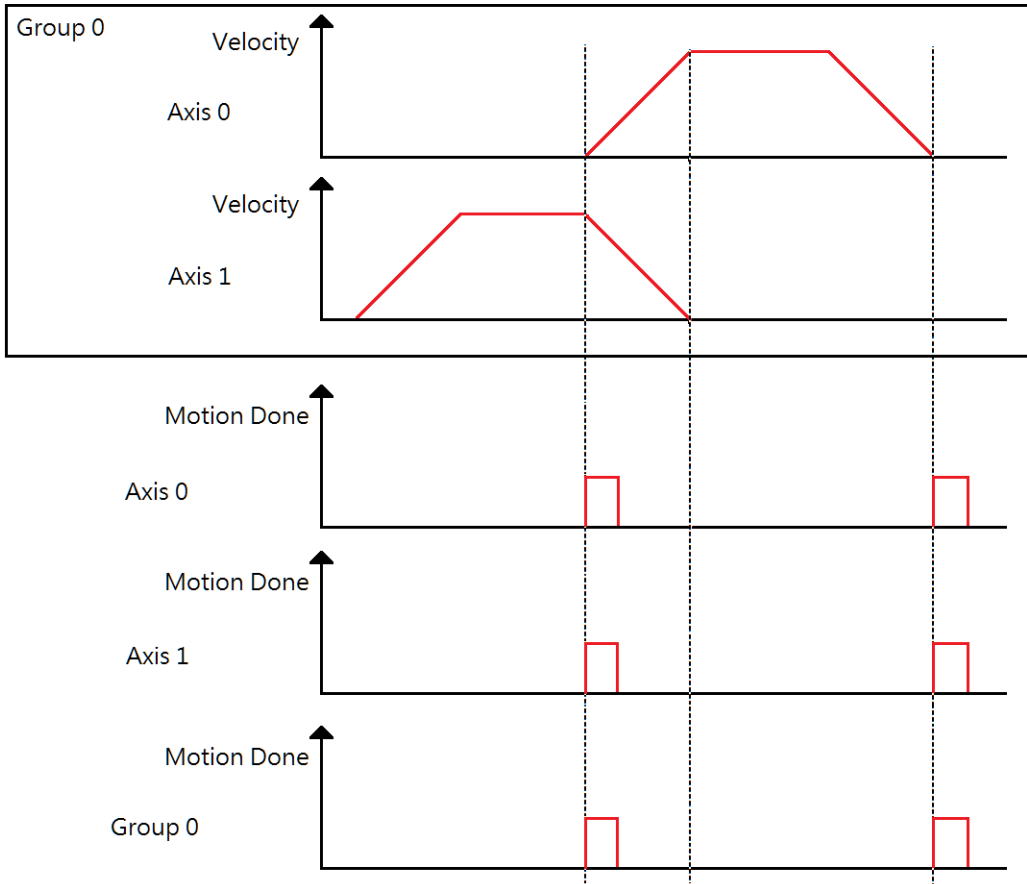
Group motion: Buffer mode
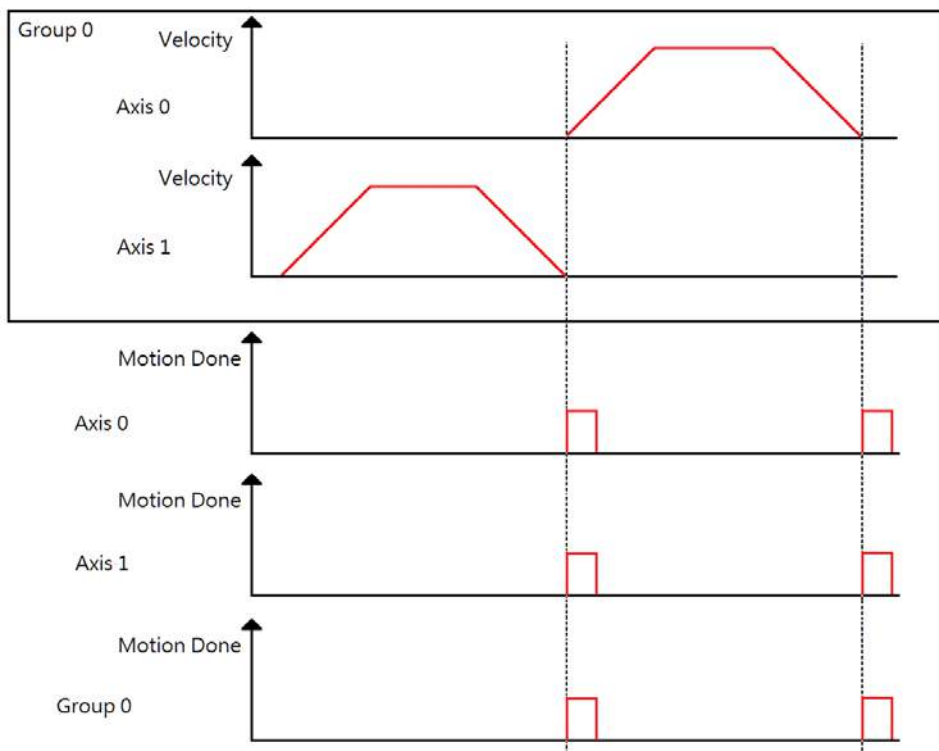
Group motion: Blending mode

## 7.12.12.　ECAT_EvSetCompareAxisVelStateParameters

**Description:**

Set event parameters as for checking an Axis velocity state.

**Syntax:**

int32_t ECAT_EvSetCompareAxisVelStateParameters(uint16_t DeviceNo, uint16_t EventID, uint16_t AxisNo, uint32_t CompareState)

**Parameters:**

| Name | Type | IN or OUT | Description |
|------|------|-----------|-------------|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EventID | uint16_t | IN | Event ID. It can be 0 ~ 31. |
| AxisNo | uint16_t | IN | Axis number |
| CompareState | uint32_t | IN | Compare Axis velocity State |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

**Axis velocity State**

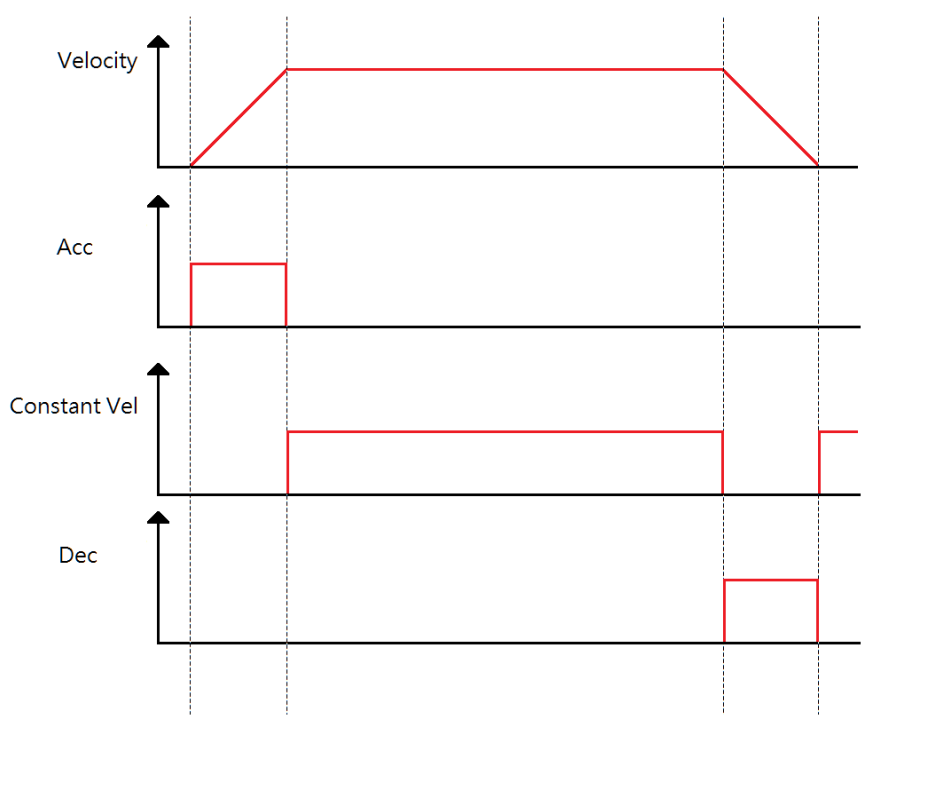| Macro Definition | Value | Description |
|------------------|-------|-------------|
| MC_AS_CONSTANT_VEL | 0 | Constant velocity section |
| MC_AS_ACC | 1 | Acceleration section |
| MC_AS_DEC | 2 | Deceleration section |

## Example:

### [C/C++]

```c
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t AxisNo = 0;
uint32_t CompareState = MC_AS_STANDSTILL;
uint32_t Value;

ret = ECAT_EvSetCompareAxisVelStateParameters(DeviceNo, EventID, AxisNo, CompareState);
if(ret < 0)
{
    printf("Failed to set compare status parameters:%d\n", ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n", ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n", ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something…
    }
}
```
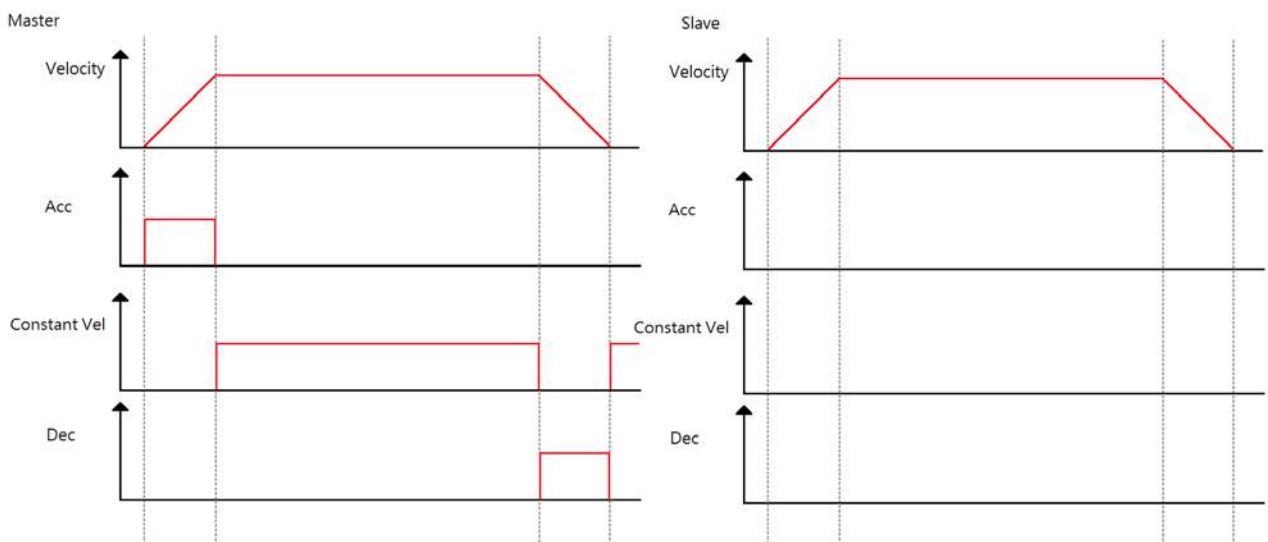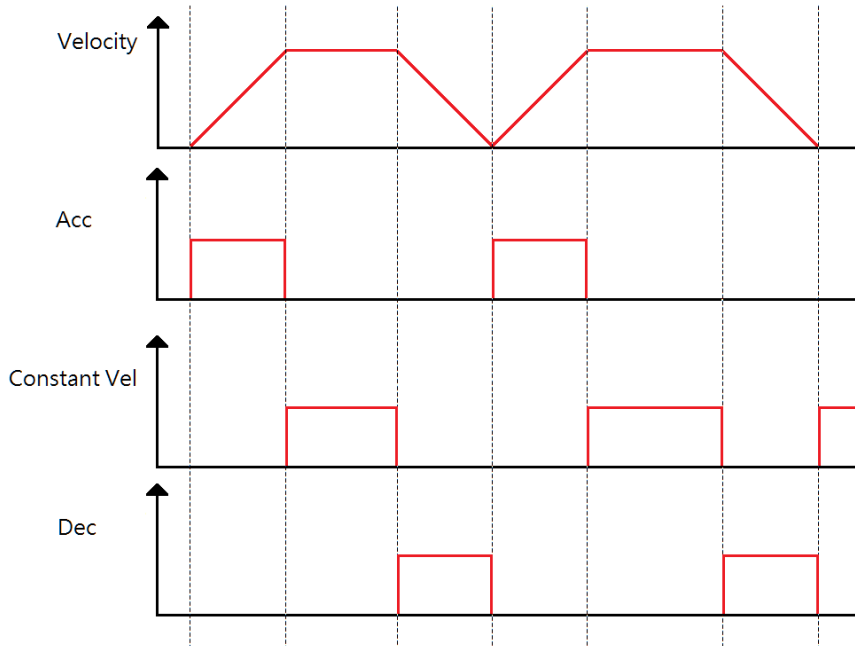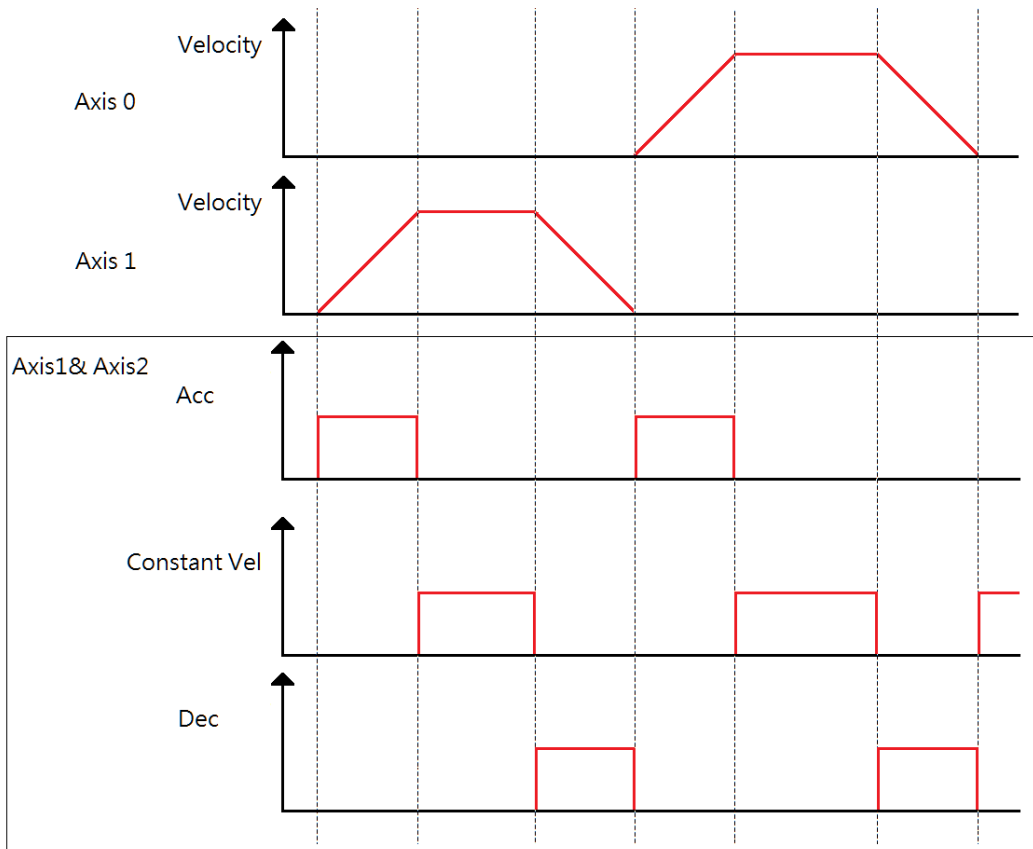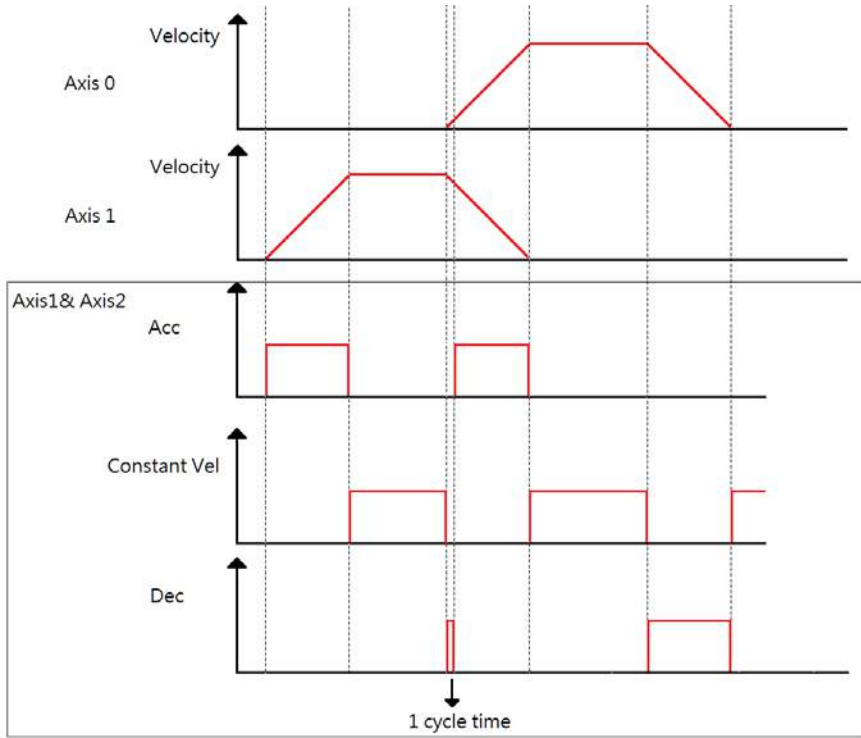
Single Axis:

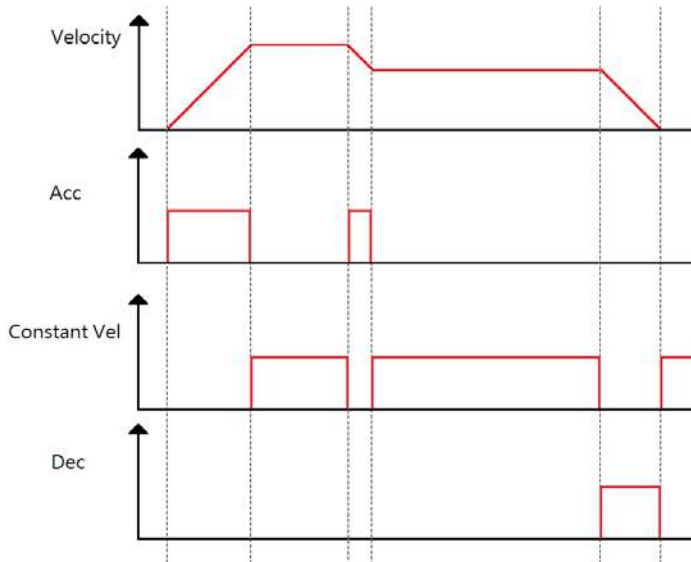## Single Axis: Gear/ Cam/ Gantry



## Single Axis: Buffer

## Group: Buffer

Group: Blending



Group: Abort

## 7.12.13.　ECAT_EvSetCompareAiParameters

**Description:**

Set event parameters for Ai comparison event.

Get TxPDO data of a slave by using **OffsetByte** and **DataSize**.

Compare Value to "**CompareValue**" after Scaling (Value = int(TxPDO data) * **ScaleGain** + **ScaleOffset**).

**Syntax:**

int32_t ECAT_ EvSetCompareAiParameters(uint16_t DeviceNo, uint16_t EventID, uint16_t Operator, uint16_t SlaveNo, uint16_t OffsetByte, uint16_t DataSize, double ScaleGain, double ScaleOffset, double CompareValue)

**Parameters:**

| Name | Type | IN or OUT | Description |
|---|---|---|---|
| DeviceNo | uint16_t | IN | Device number (Card ID) |
| EventID | uint16_t | IN | Event ID. It can be 0 ~ 31. |
| Operator | uint16_t | IN | Operator number (defined as 錯誤! 找不到參照來源。 ) |
| SlaveNo | uint16_t | IN | Slave number |
| OffsetByte | uint32_t | IN | Byte offset |
| DataSize | uint32_t | IN | Size of data (RW_PDO_DATA_SIZE_MAX macro is the maximum size of the data. It is 512.) |
| ScaleGain | uint32_t | IN | Scale Gain |
| ScaleOffset | uint32_t | IN | Scale Offset |
| CompareValue | uint32_t | IN | value for the comparison |

**Return:**

0: Success.

Others: Refer to Appendix "Error Codes".

## Example:

### [C/C++]

```
int32_t ret;
uint16_t DeviceNo = 0;
uint16_t EventID = 0;
uint16_t Operator = GREATER_THAN;
uint16_t SlaveNo = 0;
uint16_t OffsetByte = 0;
uint16_t DataSize = 2; //2 bytes
double ScaleGain = 10.0 / 0x7FFF;
double ScaleOffset = 0;
double CompareValue = 3.3;
uint32_t Value;
ret = ECAT_EvSetCompareAiParameters(DeviceNo, EventID, Operator, SlaveNo, OffsetByte, DataSize, ScaleGain, ScaleOffset, CompareValue);
if(ret < 0)
{
    printf("Failed to set compare Ai parameters:%d\n",ret);
}
ret = ECAT_EvEnableEvent(DeviceNo, EventID);
if(ret < 0)
    printf("Failed to enable event:%d\n",ret);

ret = ECAT_WaitforEvent(DeviceNo, INFINITE, &Value); // blocks until event triggered or failed or timeout.//
It is recommended to put it in the thread
if(ret != 0)
{
    printf("Failed to wait event:%d\n",ret);
}
else
{
    if(((Value>>(int(EventID))) & (0x01)) == 1)
    {
        // do something…
    }
}
```

# 8. Appendix

## 8.1. Error Codes

| Error ID | Error Code | Description |
|---|---|---|
| ECAT_ERR_REQUEST_MASTER | -1001 | Failed to request master please retry later |
| ECAT_ERR_ETHERNET_LINK_DOWN | -1002 | Ethernet network link status is down Please connect the module and try again |
| ECAT_ERR_SLAVES_STATE | -1003 | Not all slaves are in state OPERATIONAL Please startOP and try again |
| ECAT_ERR_WORKING_COUNTER | -1004 | Working counter mismatch please retry later please refer to "EtherCAT Diagnostic" |
| ECAT_ERR_SLAVE_CNT_EXCEEDED | -1005 | Connected slave count exceeds maximum support slave count |
| ECAT_ERR_CREATE_DOMAIN | -1006 | Failed to create domain data |
| ECAT_ERR_ALLOCATE_SLAVE_DATA | -1007 | Failed to allocate slave data |
| ECAT_ERR_CONFIG_SLAVE | -1008 | Failed to configure slaves |
| ECAT_ERR_NETWORK_MISMATCH | -1009 | Currently connected bus topology does not match configured one |
| ECAT_ERR_MASTER_ACTIVATE | -1010 | Failed to activate master |
| ECAT_ERR_GET_PROCESS_DATA | -1011 | Failed to get domain process data |
| ECAT_ERR_CONFIG_CYCLIC_TASK | -1012 | Failed to configure cyclic task |
| ECAT_ERR_RUN_CYCLIC_TASK | -1013 | Failed to run cyclic task |
| ECAT_ERR_INVALID_SLAVE_TYPE | -1014 | Invalid slave type |
| ECAT_ERR_SAME_SLAVE_NO | -1015 | Same slave number |
| ECAT_ERR_INVALID_SLAVE_NO | -1016 | Invalid slave number |
| ECAT_ERR_INVALID_PARAM | -1017 | Invalid parameter |
| ECAT_ERR_INVALID_DATA_SIZE | -1018 | Invalid size of data |
| ECAT_ERR_SDO_REQUEST_BUSY | -1019 | SDO request is being processed |
| ECAT_ERR_SDO_REQUEST_ERROR | -1020 | SDO request processing failed |
| ECAT_ERR_ALLOCATE_PDO_QUEUE | -1021 | Failed to allocate PDO queue data |
| ECAT_ERR_INVALID_OFFSET | -1022 | Invalid data offset |
| ECAT_ERR_INIT_MOTION | -1023 | Failed to initialize motion |

| ECAT_ERR_GET_SLAVE_INFO | -1024 | Failed to get slave information |
|---|---|---|
| ECAT_ERR_OPEN_FILE | -1025 | Failed to open file |
| ECAT_ERR_WRITE_FILE | -1026 | Failed to write data to file |
| ECAT_ERR_READ_FILE | -1027 | Failed to read data from file |
| ECAT_ERR_FUNC_NOT_SUPPORT | -1028 | Function is not supported |
| ECAT_ERR_INVALID_CHANNEL | -1029 | Invalid channel parameter |
| ECAT_ERR_EMG_HAPPENED | -1030 | Emergency happened |
| ECAT_ERR_INVALID_PID_NO | -1031 | Invalid PID number |
| ECAT_ERR_TIMER_NOT_ACTIVATED | -1032 | Timer is not activated |
| ECAT_ERR_ALL_EVENT_CREATE | -1033 | All event created |
| ECAT_ERR_EVENT_NOT_CREATE | -1034 | Event is not created |
| ECAT_ERR_INVALID_EVENTID | -1035 | Invalid event id |
| ECAT_ERR_INVALID_FILTER_TYPE | -1036 | Invalid filter type |
| ECAT_ERR_SLAVES_ALIAS | -1037 | repeating alias or alias == 0 |
| ECAT_ERR_SLAVES_ALIAS_NOT_EXIST | -1038 | alias is not exist |
| ECAT_ERR_OPTASK | -1039 | Master are in state OPERATIONAL |
| ECAT_ERR_ALL_BUFFER_USED | -1042 | all buffers are used |
| ECAT_ERR_BUFFER_NOT_ENABLE | -1043 | buffer is not enabled |
| ECAT_ERR_DEACTIVATE_SLAVES | -1046 | Changing state from OP to PreOP |
| ECAT_ERR_MASTER_BUSY_SCANNING | -1047 | Busy scanning modules |
| ECAT_ERR_MC_NOT_ENABLE_DC | -1100 | Not enable DC |
| ECAT_ERR_MC_TIME_OUT | -1101 | Call motion function time out |
| ECAT_ERR_MC_AXIS_CNT_EXCEEDED | -1102 | Initialized axis count exceeds maximum support axis count |
| ECAT_ERR_MC_NOT_INITIALIZED | -1103 | Motion is not initialized |
| ECAT_ERR_MC_INVALID_AXIS_NO | -1104 | Invalid axis number |
| ECAT_ERR_MC_NOT_AXIS_SERVO_ON | -1105 | Axis is not servo-on |
| ECAT_ERR_MC_INVALID_AXIS_STATE | -1106 | Invalid axis state |
| ECAT_ERR_MC_DRIVE_FAULT | -1107 | Drive fault |
| ECAT_ERR_MC_DRIVE_WARNING | -1108 | Drive warning |
| ECAT_ERR_MC_INVALID_PARAM | -1109 | Invalid motion parameter |
| ECAT_ERR_MC_HOMING | -1110 | An error occurs when the homing |
| ECAT_ERR_MC_LIMIT_ACTIVE | -1111 | Limit switch is active |
| ECAT_ERR_MC_INVALID_ACC_TIME | -1112 | Invalid acceleration time |
| ECAT_ERR_MC_INVALID_GROUP_NO | -1113 | Invalid group number |
| ECAT_ERR_MC_INVALID_GROUP_STATE | -1114 | Invalid group state |
| ECAT_ERR_MC_AXIS_WAS_IN_GROUP | -1115 | Axis is already in group |
| ECAT_ERR_MC_AXIS_IN_OTHER_GROUP | -1116 | Axis is already in other group |
| ECAT_ERR_MC_GROUP_CMD_ALLOCATE | -1117 | Failed to allocate group command |

| ECAT_ERR_MC_GROUP_CMD_BUFFER_OVERFLOW | -1118 | Group command is overflow |
|---|---|---|
| ECAT_ERR_MC_INVALID_AXIS_SYNC_MODE | -1119 | Invalid axis synchronization mode |
| ECAT_ERR_MC_INVALID_PROFILE_NO | -1120 | Invalid profile number |
| ECAT_ERR_MC_INVALID_GROUP_MOVE_CMD | -1121 | Invalid group command |
| ECAT_ERR_MC_GROUP_CMD_MODE_NOT_SUPPORT | -1122 | The function does not support the current group command mode |
| ECAT_ERR_MC_INVALID_ACC_DEC_TYPE | -1123 | Invalid acceleration type parameter |
| ECAT_ERR_MC_INVALID_VEL | -1124 | Invalid velocity parameter |
| ECAT_ERR_MC_INVALID_ANGLE | -1125 | Invalid angle parameter |
| ECAT_ERR_MC_INVALID_RADIUS | -1126 | Invalid radius parameter |
| ECAT_ERR_MC_INVALID_END_POS | -1127 | Invalid end position parameter |
| ECAT_ERR_MC_INVALID_ECAM_TABLE_NO | -1128 | Invalid E-CAM table number |
| ECAT_ERR_MC_INVALID_NORMAL_VECTOR | -1129 | Invalid normal vector parameter |
| ECAT_ERR_MC_NOT_SETUP | -1130 | Not setup |
| ECAT_ERR_MC_GREATER_THAN_MAX_RODLENGTH | -1131 | Calculated value is greater than maximum rod length |
| ECAT_ERR_MC_LESS_THAN_RODLENGTH | -1132 | Calculated value is less than rod length |
| ECAT_ERR_MC_GREATER_THAN_RECORD_COUNT | -1133 | Exceed maximum record count |
| ECAT_ERR_MC_SOFTWARE_LIMIT_ACTIVATE | -1134 | Software limit is active |
| ECAT_ERR_MC_GANTRY_POS_EXCESSIVE_DEVIATION | -1135 | Position excessive deviation of gantry control |
| ECAT_ERR_MC_GROUP_NO_NOT_SUPPORT | -1136 | Group number not support |
| ECAT_ERR_MC_INVALID_MOVE_CMD | -1137 | Invalid move command |
| ECAT_ERR_MC_QUEUE_IS_FULL | -1138 | Queue is full |
| ECAT_ERR_MC_COORDINATE_TRANS_ON | -1139 | Coordinate conversion is active |
| ECAT_ERR_MC_HAVE_NOT_BEEN_SET | -1140 | Have not been set yet |
| ECAT_ERR_MC_HOMIE_NOT_DONE | -1141 | Homing is not done |
| ECAT_ERR_MC_INHIBITED_FUNCTION | -1142 | Inhibited function |
| ECAT_ERR_MC_LACK_PDOS | -1143 | Lack PDOs |
| ECAT_ERR_MC_SAFETY_STOP | -1144 | Stop for safety |
| ECAT_ERR_IPC_INVALID_DEVICE_NO | -1201 | Invalid device number |
| ECAT_ERR_IPC_DEVICE_IS_OPEN | -1202 | Device is open |
| ECAT_ERR_IPC_DEVICE_NOT_OPEN | -1203 | Device is not open |
| ECAT_ERR_IPC_CREATE_HANDLE | -1204 | Failed to create IPC handle |
| ECAT_ERR_IPC_BUSY | -1205 | IPC is busy |
| ECAT_ERR_IPC_TIME_OUT | -1206 | IPC is time out |

| ECAT_ERR_IPC_INVALID_CMD | -1207 | Invalid IPC command |
|---|---|---|
| ECAT_ERR_IPC_WRITE_SHM | -1208 | Failed to write data to shard memory |
| ECAT_ERR_IPC_READ_SHM | -1209 | Failed to read data from shard memory |
| ECAT_ERR_IPC_RUN_DOWN_UP_LOAD | -1210 | Failed to process download / upload data |
| ECAT_ERR_IPC_INVALID_SHM | -1211 | Invalid shard memory |
| ECAT_ERR_IPC_DEVICE_NOT_READY | -1212 | Device is not ready |
| ECAT_ERR_DRV_GET_INFO | -1301 | Failed to get driver information |
| ECAT_ERR_DRV_CREATE_HANDLE | -1302 | Failed to create driver handle |
| ECAT_ERR_DRV_IOCTL | -1303 | Call driver IO control error |
| ECAT_ERR_DRV_DEVICE_NOT_FOUND | -1304 | Device not found |

## 8.2.  SDO Abort messages

| Abort code | Description |
| --- | --- |
| 0x05030000 | Toggle bit not changed |
| 0x05040000 | SDO protocol timeout |
| 0x05040001 | Client/Server command specifier not valid or unknown |
| 0x05040005 | Out of memory |
| 0x06010000 | Unsupported access to an object |
| 0x06010001 | Attempt to read a write-only object |
| 0x06010002 | Attempt to write a read-only object |
| 0x06020000 | This object does not exist in the object directory |
| 0x06040041 | The object cannot be mapped into the PDO |
| 0x06040042 | The number and length of the objects to be mapped would exceed the PDO length |
| 0x06040043 | General parameter incompatibility reason |
| 0x06040047 | General internal incompatibility in device |
| 0x06060000 | Access failure due to a hardware error |
| 0x06070010 | Data type does not match, length of service parameter does not match |
| 0x06070012 | Data type does not match, length of service parameter too high |
| 0x06070013 | Data type does not match, length of service parameter too low |
| 0x06090011 | Sub-index does not exist |
| 0x06090030 | Value range of parameter exceeded |
| 0x06090031 | Value of parameter written too high |
| 0x06090032 | Value of parameter written too low |
| 0x06090036 | Maximum value is less than minimum value |
| 0x08000000 | General error |
| 0x08000020 | Data cannot be transferred or stored to the application |
| 0x08000021 | Data cannot be transferred or stored to the application because of local control |
| 0x08000022 | Data cannot be transferred or stored to the application because of the present device state |
| 0x08000023 | Object dictionary dynamic generation fails or no object dictionary is present |

## 8.3. Revision History

This chapter provides revision history information to this document

The table below shows the revision history.

| Revision | Date | Description |
|----------|------|-------------|
| 1.0.24 | 2021 | Please refers to 1.1 |