

ICP DAS ECATDAQ DLL

MANUAL

User manual

For ICP DAS EtherCAT SubDevice Module



English Version

V1.6

2025.Sep



Warning

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

Copyright

Copyright © 2025 by ICP DAS Co., Ltd. All rights are reserved.

Trademarks

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

About this Manual

This manual contains the information you need to get started with the ICP DAS ECATDAQ DLL software package. The DLL allow you to easily perform vital ICP DAS EtherCAT subdevice I/O and motion operations through the API, functions and structure.

The ECATDAQ DLL can be used to develop custom programs based on the VB, VC, BCB, Delphi, VB.NET, C#.NET, VC.NET, Console and other programming languages using Windows Systems. This manual also provides sample programs that can be modified to create custom applications that meet specific requirements.

If you have any questions, feel free to contact the ICP DAS Service Department via email at: service@icpdas.com

Table of contents

1.	Introducing the ECATDAQ DLL.....	6
	Feature	6
	Supported ICP DAS Products.....	7
	System Requirements	7
	Related Information	7
2.	Main Device Operation Flows	8
	ECDAQ_GetDeviceNoByID	9
3.	Main Device Operation Functions	12
	ECDAQ_GetDllVer.....	12
	ECDAQ_GetAdapterInfo	14
	ECDAQ_Init.....	16
	ECDAQ_Close	18
	ECDAQ_StartDeviceOpTask.....	20
	ECDAQ_StopDeviceOpTask	22
	ECDAQ_GetMasterState	24
4.	SubDevice Operation Functions.....	26
	ECDAQ_GetDeviceName.....	26
	ECDAQ_GetDeviceInfo	28
	ECDAQ_SetDeviceID.....	30
	ECDAQ_GetDeviceNoByID	31
	ECDAQ_SetSdoData	32
	ECDAQ_GetSdoData.....	34
	ECDAQ_SetSdoObject	35
	ECDAQ_GetSdoData.....	37
	ECDAQ_WriteDO	39
	ECDAQ_ReadDI.....	41
	ECDAQ_WriteAOVoltageH.....	43
	ECDAQ_WriteAOCurentH	44
	ECDAQ_ReadAIH	46
	ECDAQ_ReadAIScanH.....	48
5.	Motion Control Functions	50
	Axis Parameter Settings	50
	ECDAQ_MC_SetAxisServoON.....	50
	ECDAQ_MC_SetAxisMicroStep	52

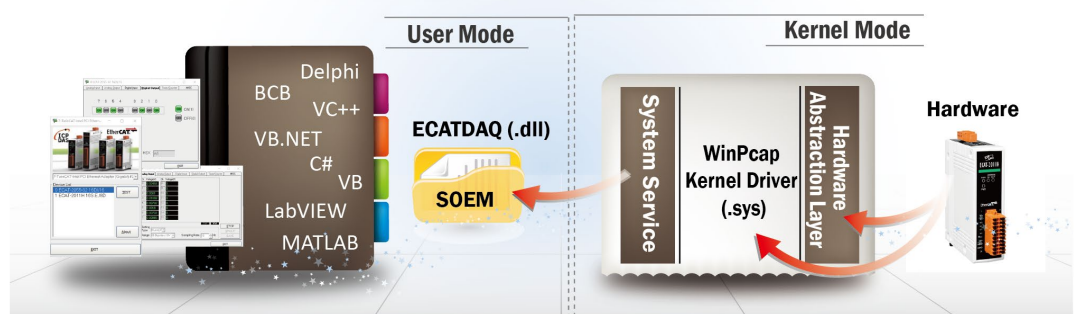
ECDAQ_MC_GetAxisMicroStep.....	54
ECDAQ_MC_SetAxisCurrent.....	56
ECDAQ_MC_GetAxisCurrent.....	58
Axis Status	60
ECDAQ_MC_GetAxisActualPos	60
ECDAQ_MC_GetAxisActualEnc	62
ECDAQ_MC_GetAxisActualVel	64
ECDAQ_MC_GetAxisActualTor	66
ECDAQ_MC_GetAxisState	68
ECDAQ_MC_GetAxisDI	71
ECDAQ_MC_GetAxisLastError.....	74
ECDAQ_MC_GetAxisDriverError	76
Axis Homing	78
ECDAQ_MC_AxisHomeStart	78
Axis Moving.....	81
ECDAQ_MC_ResetAxisError	81
ECDAQ_MC_ResetAxisCounter	83
ECDAQ_MC_AxisMoveAbs.....	85
ECDAQ_MC_AxisMoveRel.....	87
ECDAQ_MC_AxisChangePos	89
ECDAQ_MC_AxisChangeVel	92
ECDAQ_MC_AxisMoveVel.....	95
ECDAQ_MC_AxisMove_CiA402_PV	97
ECDAQ_MC_AxisMove_CiA402_PT	99
ECDAQ_MC_AxisStop.....	101
ECDAQ_MC_AxisQuickStop	104
StartUp Parameter Settings	107
ECDAQ_MC_SetLoadDefault.....	107
ECDAQ_MC_GetLoadDefaultState.....	109
Encoder Operation	111
ECDAQ_GetEncCountingMode	111
ECDAQ_SetEncCountingMode	113
ECDAQ_GetEncCount.....	115
ECDAQ_SetEncCount	117
ECDAQ_GetEncLatchCount	119
ECDAQ_GetEncLatchTriggerCount	121
ECDAQ_GetEncExLatchCount	123
ECDAQ_ResetEncCount.....	125

ECDAQ_SetEnableEncLatch.....	127
ECDAQ_GetEnableEncLatch.....	129
ECDAQ_SetEnableEncExLatch.....	131
ECDAQ_GetEnableEncExLatch.....	133
ECDAQ_ResetEncLatchCount.....	135
ECDAQ_ResetEncExLatchCount.....	137
ECDAQ_GetCmpTrigProperty.....	139
ECDAQ_SetCmpTrigProperty.....	141
ECDAQ_GetEncNextComparePos.....	143
ECDAQ_EnableEncEquidCompare.....	145
ECDAQ_SetArrCmpPosTrigger.....	147
ECDAQ_GetArrCmpPosTrigger.....	151
ECDAQ_SetArrCmpEndIdx.....	153
ECDAQ_GetArrCmpEndIdx.....	157
ECDAQ_EnableEncArrayCompare.....	159
ECDAQ_DisableEncCompare.....	162
ECDAQ_SetVelTimeInterval.....	164
ECDAQ_GetVelTimeInterval.....	166
ECDAQ_GetVeldP.....	168
ECDAQ_GetAbsEncState.....	169
ECDAQ_GetAbsEncCount.....	171
ECDAQ_GetAbsEncCheckCnt.....	173
ECDAQ_ResetAbsEncCheckCnt.....	175
ECDAQ_SetAbsEncBits.....	177
ECDAQ_GetAbsEncBits.....	179
ECDAQ_SetAbsEncMode.....	181
ECDAQ_GetAbsEncMode.....	183
Appendix.....	185
Error Codes.....	185
SDO Abort messages.....	189
Initial setup for CiA402 driver.....	191

1.Introducing the ECATDAQ DLL

The ICP DAS ECATDAQ DLL is built on the open source library SOEM (Simple Open EtherCAT Master).The ICP DAS ECATDAQ DLL provides complete hardware functions and maximum performance. With the ICP DAS ECATDAQ DLL, there is no need to use EtherCAT hardware-specific register commands thanks to the powerful API function that can be used with a variety of programming environments and languages.

The ICP DAS ECATDAQ DLL supports Windows both 32- and 64 bit versions Windows 7 and later version.



Feature

- ICP DAS Sub device can be controlled through a standard Ethernet interface card
- Provide DLL function library and VC/VB6/C#/Python/LabVIEW sample programs
- Cycle time 20ms (typical)
- Support 8 axes and 20 sub devices(Only support ICPDAS sub device)
- Support Free Run mode
- Support third-party CiA402 driver
- Supports single-axis linear interpolation and PP mode

Supported ICP DAS Products

ECAT-2045 series ECAT-2050 ECAT-2051 series ECAT-2052 series ECAT-2053 series ECAT-2055 series ECAT-2057 series ECAT-2060 ECAT-2061	ECAT-2024 series ECAT-2028 series ECAT-2011H ECAT-2012H ECAT-2016 series EC1-P16C16 EC1-P32 EC1-C32 EC2-P16C16 EC2-P32 EC2-C32	EC4-EP21 EC4-EP20E EC4-C16 EC4-A16 EC4-P16 EC4-AD2/AD4/AD8 EC4-DA2/DA4/DA4P EC4-TC8/TC8P EC4-RTD8 EC4-MP1U EC4-ENC2IT EC4-ENC2A/ENC4A	ECAT-2091 series ECAT-2094 series ECAT-2092 series ECAT-2093 series ECAT-2072A ECAT-2074A
---	--	--	--

System Requirements

ICP DAS XP-9000 series PAC or AXP-9000 series PAC(Must download and install ECATDAQ DLL Installer manually) or Person Computer(Personal computers need to meet the following requirements) :

- ECAT-2091 、ECAT-2094 or EC4 series EtherCAT subdevice
- Standard Ethernet Network Adapter
- 266 MHz 32-bit (x86) or 64-bit (x64) processor
- 512 MB of system memory
- Support for Super VGA graphics
- At least 100 MB of available space
- 32- or 64-bit Windows Operating System (Windows 7 or later)

Related Information

For details information about the ECATDAQ DLL user manual, installation program and sample programs, the quick download links are as follows:

<https://www.icpdas.com/en/product/guide+Software+Development+Tools+ECATDAQ>

2. Main Device Operation Flows

As the figure shown below, the user can call the *ECDAQ_GetAdapterInfo* function to find out how many network adapters can be used. Then, according to the adapters number, call *ECDAQ_Init* function to open that adapters. After this adapters 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 *ECDAQ_GetMasterState* to get the current states of the EtherCAT network. These states include the number of currently connected subdevices, the AL status, network link status, etc. Next, the *ECDAQ_GetDeviceInfo* function can be called for each subdevice to get the subdevice information. If some SDO objects need to be read/written, the *ECDAQ_GetSdoData* and *ECDAQ_SetDeviceID*

Description:

Set Device identification (Explicit Device ID or Alias ID)

Syntax:

WORD ECDAQ_SetSdoData(DWORD dwDeviceNo, WORD wIndex, UCHAR uSubIndex, DWORD dwVal)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
dwDeviceID	DWORD	IN	Set sub device ID

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

ECDAQ_GetDeviceNoByID

Description:

Get the sub device number(position) value through the explicit device identification or Alias ID.

Syntax:

WORD ECDAQ_SetSdoData(DWORD dwDeviceNo, WORD wIndex, UCHAR uSubIndex, DWORD dwVal)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD*	OUT	Device number
dwDeviceID	DWORD	IN	Explicit device identification or Alias ID

Return:

0: Success.

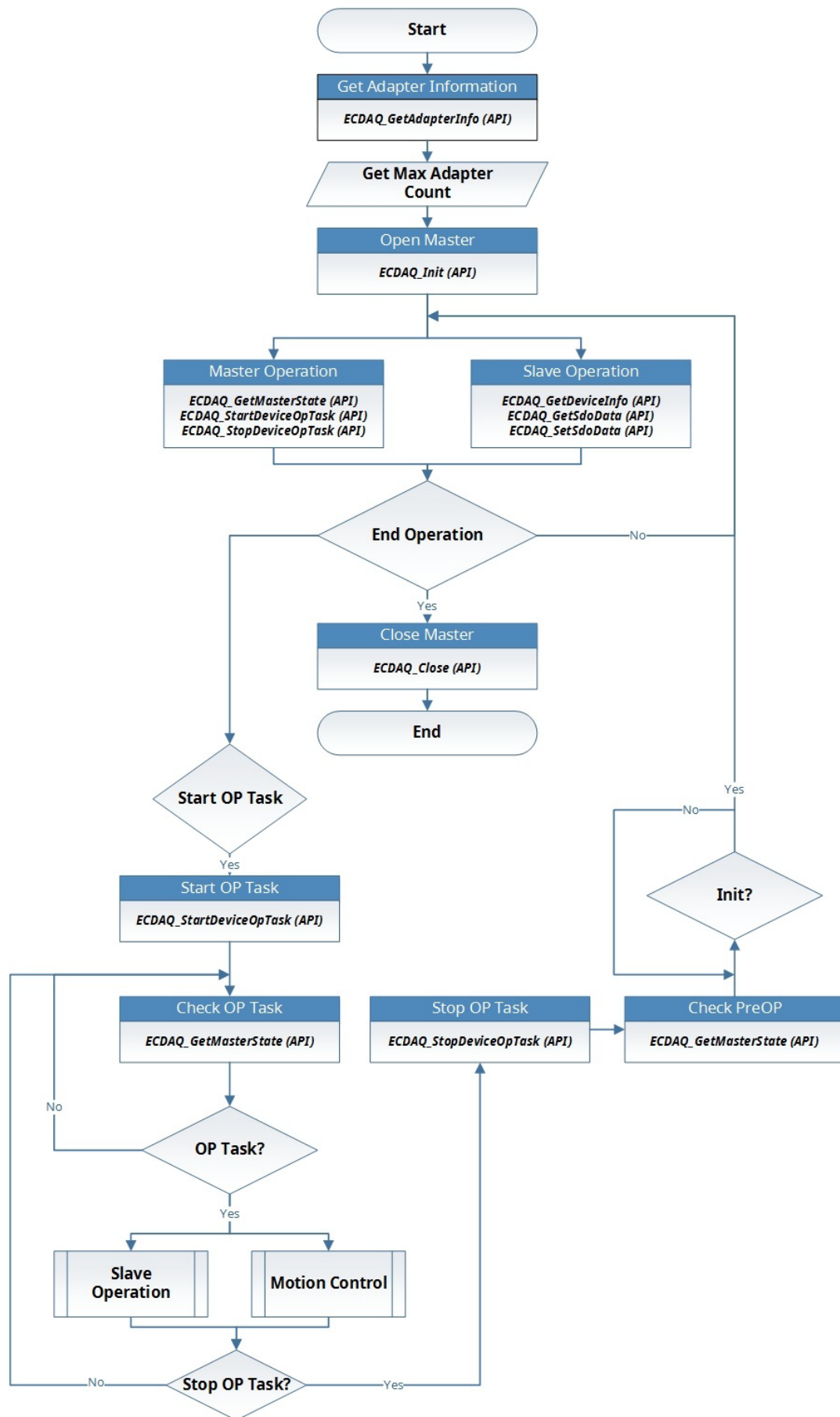
Others: Refer to Appendix "Error Codes".

ECDAQ_SetSdoData functions can be used for these purposes. These functions will do acyclic communication through EtherCAT Bus.

If all slaves are ready, call *ECDAQ_StartDeviceOpTask* function to start the EtherCAT operation task. This function will command slaves to enter into the OP state. The user can use *ECDAQ_GetMasterState* 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.

When master is in OP state, it will automatically initialize the motion axes of connected drivers. The user only needs to use *ECDAQ_MC_SetAxisServoON* function to enable the motion axis, and then the homing and motion control functions can be performed.

To stop the EtherCAT communication, *ECDAQ_StopDeviceOpTask* function must be called. To close the adapter, use *ECDAQ_Close* function to do it.



3. Main Device Operation

Functions

ECDAQ_GetDllVer

Description:

Get ECATDAQ SDK Version °

Syntax:

WORD ECDAQ_GetDllVer(DWORD *dwDllVer)

Parameters:

Name	Type	Direction	Description
dwDllVer	DWORD*	OUT	Version number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DLLVer= 0;

ret = ECDAQ_GetDllVer(&DLLVer);
if(ret < 0)
{
    printf("Failed to get dll version:%d\n",ret);
}
else
{
    printf("Get dll version successfully!\n");
}
```

ECDAQ_GetAdapterInfo

Description:

Get Network Adapter information °

Syntax:

```
WORD ECDAQ_GetAdapterInfo(WORD *wTotalNetworkNum, WORD  
wNetworkInfoNo, char* szNetworkInfoName)
```

Parameters:

Name	Type	Direction	Description
wTotalNetworkNum	WORD*	OUT	Total number of Network adapter
wNetworkInfoNo	WORD	IN	Network adapter number
szNetworkInfoName	char*	OUT	Network adapter name

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
WORD TotalNetworkNum= 0;
WORD NetworkInfoNo= 0;
char NetworkInfoName[128];

ret = ECDAQ_GetAdapterInfo (&TotalNetworkNum , NetworkInfoNo , NetworkInfoName);
if(ret < 0)
{
    printf("Failed to get adapter info:%d\n",ret);
}
else
{
    printf("Get adapter info successfully!\n");
}
```

ECDAQ_Init

Description:

Initilize the EtherCAT master ◦

Syntax:

```
WORD ECDAQ_Init(WORD wNetworkInfoNo, DWORD *dwTotalSlaveDevice);
```

Parameters:

Name	Type	Direction	Description
wNetworkInfoNo	WORD	IN	Network adapter number
dwTotalSlaveDevice	DWORD *	OUT	Read amount of the EtherCAT Device on Network adapter

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
WORD NetworkInfoNo= 0;
WORD TotalSlaveDevice = 0;

ret = ECDAQ_Init (NetworkInfoNo , & TotalSlaveDevice);
if(ret < 0)
{
    printf("Failed to init ECDAQ:%d\n",ret);
}
else
{
    printf("Init ECDAQ successfully!\n");
}
```

ECDAQ_Close

Description:

Close EtherCAT master °

Syntax:

WORD ECDAQ_Close ();

Parameters:

Name	Type	Direction	Description

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

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

ECDAQ_StartDeviceOpTask

Description:

Start the device EtherCAT operation task ◦

Syntax:

WORD ECDAQ_StartDeviceOpTask(WORD wNetworkInfoNo, WORD wEnumCycleTimeMs, DWORD dwWcErrCnt)

Parameters:

Name	Type	Direction	Description
wNetworkInfoNo	WORD	IN	Network adapter number
wEnumCycleTimeMs	WORD	IN	EtherCAT cycle time. Unit is ms.
dwWcErrCnt	DWORD	IN	Tolerant count of working counter error. The alarm is notified while the error count exceeds tolerant count.

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

WORD NetworkInfoNo= 0;

WORD EnumCycleTimeMs= 0;

DWORD WcErrCnt = 10;


ret = ECDAQ_StartDeviceOpTask (NetworkInfoNo, EnumCycleTimeMs, WcErrCnt);
if(ret < 0)
{
    printf("Failed to start device OP task:%d\n",ret);
}
else
{
    printf("Start device OP task successfully!\n");
}
```

ECDAQ_StopDeviceOpTask

Description:

Stop the EtherCAT operation task °

Syntax:

WORD ECDAQ_StopDeviceOpTask(WORD wNetworkInfoNo)

Parameters:

Name	Type	Direction	Description
wNetworkInfoNo	WORD	IN	Network adapter number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
```

```
WORD NetworkInfoNo= 0;
```

```
ret = ECDAQ_StopDeviceOpTask (NetworkInfoNo, EnumCycleTimeMs, WcErrCnt);
```

```
if(ret < 0)
```

```
{
```

```
    printf("Failed to stop device OP task:%d\n",ret);
```

```
}
```

```
else
```

```
{
```

```
    printf("Stop device OP task successfully!\n");
```

```
}
```

ECDAQ_GetMasterState

Description:

Get the EtherCAT network status of master.

Syntax:

```
WORD ECDAQ_GetMasterState(DWORD *dwLinkUp, DWORD *dwSlavesResp,  
DWORD *dwAlStates, DWORD *dwWkc)
```

Parameters:

Name	Type	Direction	Description
dwLinkUp	DWORD*	OUT	Network link status of Ethernet (EtherCAT) 0: Link Down 1: Link Up
dwSlavesResp	DWORD*	OUT	Sum of responding slaves on this EtherCAT network system
dwAlStates	DWORD*	OUT	AL state of EtherCAT master. AL states are defined shown below.
dwWkc	DWORD*	OUT	EtherCAT working counter value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Macro Definition	Value	Description
ECMST_SM_IDLE	0x00	Idle 狀態
ECMST_SM_INIT	0x01	Init 狀態
ECMST_SM_PRE_OP	0x02	Pre-Operational 狀態
ECMST_SM_BOOT	0x03	BOOT 狀態
ECMST_SM_SAFE_OP	0x04	Safe-Operational 狀態
ECMST_SM_OPERATIONAL	0x08	Operational 狀態

Example:

[C/C++]

```
int ret = 0;

DWORD LinkUp, SlavesResp, AlStates, Wkc;

char StrAlState[255];

ret = ECDAQ_GetMasterState(&LinkUp, &SlavesResp, &AlStates, &Wkc);
if(ret < 0)
{
    printf("Failed to stop device OP task:%d\n",ret);
}
else
{
    if(AlStates == ECMST_SM_IDLE)
        printf(StrAlState, "IDLE");
    else if(AlStates == ECMST_SM_INIT)
        printf(StrAlState, "INIT");
    else if(AlStates == ECMST_SM_PRE_OP)
        printf(StrAlState, "PREOP");
    else if(AlStates == ECMST_SM_BOOT)
        printf(StrAlState, "BOOT");
    else if(AlStates == ECMST_SM_SAFE_OP)
        printf(StrAlState, "SAFEOP");
    else if(AlStates == ECMST_SM_OPERATIONAL)
        printf(StrAlState, "OP");
    else
        printf(StrAlState, "err.0x%x", AlStates);
    printf("Slave(s):%u | AL States:%s | Link is :%s | Wc:%-u "
           ,SlavesResp, StrAlState, LinkUp? "up" : "down", Wc);
}
```

4.SubDevice Operation Functions

ECDAQ_GetDeviceName

Description:

Get device name of the EtherCAT slave.

Syntax:

WORD ECDAQ_GetDeviceName(DWORD dwDeviceNo, char *szDeviceName)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
szDeviceName	char*	OUT	Device name of the EtherCAT slave

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
char szDeviceName [128];

ret = ECDAQ_GetDeviceName (DeviceNo, szDeviceName);
if(ret < 0)
{
    printf("Failed to get device name:%d\n",ret);
}
else
{
    printf("Get device name successfully!\n");
}
```

ECDAQ_GetDeviceInfo

Description:

Get the EtherCAT device information °

Syntax:

```
WORD ECDAQ_GetDeviceInfo(DWORD dwDeviceNo, PECDAQ_DEVICE_INFO  
sDevInfo)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
sDevInfo	PECDAQ_DEVICE_INFO	OUT	Device information of the EtherCAT slave.

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
typedef struct _ECDAQ_DEVICE_INFO_
{
    DWORD dwSize;           //Structure Size

    DWORD dwVendorID;       //Device Vendor number
    DWORD dwProductID;      //Device Product number
    DWORD dwVersionID;      //Device Version number

    WORD  wOutputSize;      //Output Size(bit)
    WORD  wInputSize;       //Input Size(bit)

    WORD wOutputMapIndex;
    WORD wInputMapIndex;
    WORD wDIChNum;
    WORD wDOChNum;
    WORD wAIVoltageChNum;
    WORD wAICurrentChNum;
    WORD wAOVoltageChNum;
    WORD wAOCurrentChNum;
    WORD wCTRChNum;
    WORD wCOMNum;
}ECDAQ_DEVICE_INFO, *PECDQA_DEVICE_INFO;
PECDQA_DEVICE_INFO DevInfo;

ret = ECDAQ_GetDeviceInfo (DeviceNo, DevInfo);
if(ret < 0)
{
    printf("Failed to get device info:%d\n",ret);
}
else
{
    printf("Get device info successfully!\n");
}
```

ECDAQ_SetDeviceID

Description:

Set Device identification (Explicit Device ID or Alias ID)

Syntax:

WORD ECDAQ_SetSdoData(DWORD dwDeviceNo, WORD wIndex, UCHAR uSubIndex, DWORD dwVal)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
dwDeviceID	DWORD	IN	Set sub device ID

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

ECDAQ_GetDeviceNoByID

Description:

Get the sub device number(position) value through the explicit device identification or Alias ID.

Syntax:

WORD ECDAQ_SetSdoData(DWORD dwDeviceNo, WORD wIndex, UCHAR uSubIndex, DWORD dwVal)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD*	OUT	Device number
dwDeviceID	DWORD	IN	Explicit device identification or Alias ID

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

ECDAQ_SetSdoData

Description:

Set SDO data of a slave.

Syntax:

```
WORD ECDAQ_SetSdoData(DWORD dwDeviceNo, WORD wIndex, UCHAR uSubIndex,  
DWORD dwVal)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wIndex	WORD	IN	Object index
uSubIndex	UCHAR	IN	Object sub-index
dwVal	DWORD	IN	Data for writing

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD Index = 0x606C; //Home Offset Index

UCHAR SubIndex = 0x00; //Home Offset SubIndex

DWORD Val = 100;

ret = ECDAQ_SetSdoData (DeviceNo, Index, SubIndex, Val);

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

ECDAQ_GetSdoData

Description:

Get SDO data of a slave.

Syntax:

```
WORD ECDAQ_GetSdoData(DWORD dwDeviceNo, WORD wIndex, UCHAR uSubIndex,  
DWORD* dwVal)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wIndex	WORD	IN	Object index
uSubIndex	UCHAR	IN	Object sub-index
dwVal	DWORD*	OUT	Read-out data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

ECDAQ_SetSdoObject

Description:

Set SDO data of a subdevice and check for any operation errors.

Syntax:

```
WORD ECDAQ_SetSdoObject(DWORD dwDeviceNo, WORD wIndex, UCHAR  
uSubIndex, WORD wDataSize, DWORD dwVal, DWORD* AbortCode)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wIndex	WORD	IN	Object index
uSubIndex	UCHAR	IN	Object sub-index
wDataSize	WORD	IN	Object data size
dwVal	DWORD	IN	Data for writing
AbortCode	DWORD	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++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD Index = 0x607C; //Home Offset Index

UCHAR SubIndex = 0x00; //Home Offset SubIndex

WORD wDataSize = 4; //4 byte

DWORD Val = 100;

DWORD AbortCode = 0;

ret = ECDAQ_SetSdoDataObject (DeviceNo, Index, SubIndex, wDataSize, Val, &AbortCode);
if(ret < 0)
{
    if(AbortCode == 3) // EC_ERR_TYPE_PACKET_ERROR
        printf("Failed to set SDO object:%d, packet error\n",ret);
    else
        printf("Failed to set SDO object:%d, abortcode[0x%x]\n",ret, AbortCode);
}
else
{
    printf("Set SDO object successfully!\n");
}
```

ECDAQ_GetSdoData

Description:

Get SDO data of a subdevice and check for any operation errors.

Syntax:

WORD ECDAQ_GetSdoObject(DWORD dwDeviceNo, WORD wIndex, UCHAR
uSubIndex, WORD wDataSize, DWORD* dwVal, DWORD* AbortCode)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wIndex	WORD	IN	Object index
uSubIndex	UCHAR	IN	Object sub-index
wDataSize	WORD	IN	Object data size
dwVal	DWORD*	OUT	Read-out data
AbortCode	DWORD	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++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD Index = 0x607C; //Home Offset Index
UCHAR SubIndex = 0x00; //Home Offset SubIndex
WORD wDataSize = 4; //4 byte
DWORD Val = 0;
DWORD AbortCode = 0;

ret = ECDAQ_GetSdoDataObject (DeviceNo, Index, SubIndex, wDataSize, &Val, &AbortCode);
if(ret < 0)
{
    if(AbortCode == 3) // EC_ERR_TYPE_PACKET_ERROR
        printf("Failed to get SDO object:%d, packet error\n",ret);
    else
        printf("Failed to get SDO object:%d, abortcode[0x%x]\n",ret, AbortCode);
}
else
{
    printf("Get SDO object successfully!\n");
}
```

ECDAQ_WriteDO

Description:

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

Syntax:

WORD ECDAQ_WriteDO(DWORD dwDeviceNo, WORD wPortNo, char cValue)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wPortNo	WORD	IN	Port number
cValue	char	IN	Data for writing

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD PortNo = 0;
char Value = 0xff;

ret = ECDAQ_WriteDO(DeviceNo, PortNo, Value);
if(ret < 0)
{
    printf("Failed to Write DO data:%d\n",ret);
}
else
{
    printf("Write DO data successfully!\n");
}
```

ECDAQ_ReadDI

Description:

Get a port data of a slave's digital input.

Syntax:

WORD ECDAQ_ReadDI(DWORD dwDeviceNo, WORD wPortNo, char *cValue)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wPortNo	WORD	IN	Port number
cValue	char	IN	Read-out data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD PortNo = 0;
char Value;

ret = ECDAQ_ReadDI(DeviceNo, PortNo, &Value);
if(ret < 0)
{
    printf("Failed to Read DI data:%d\n",ret);
}
else
{
    printf("Read DI data successfully!\n");
}
```

ECDAQ_WriteAOVoltageH

Description:

Writes a binary value to one of the analog output channels, changing the voltage produced at the channel.

Syntax:

WORD ECDAQ_WriteAOVoltageH(DWORD dwDeviceNo, WORD wAOChannel, WORD wAORange, DWORD dwValue)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAOChannel	WORD	IN	Analog output channel number
wAORange	WORD	IN	Analog output range. The definition is shown below.
dwValue	DWORD	IN	Binary data to be written

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Macro Definition	Value	Description
ECDAQ_AO_UNI_5V	0	Voltage Range 0 ~ 5V
ECDAQ_AO_BI_5V	1	Voltage Range $\pm 5V$
ECDAQ_AO_UNI_10V	2	Voltage Range 0 ~ 10V
ECDAQ_AO_BI_10V	3	Voltage Range $\pm 10V$

ECDAQ_WriteAOCurrentH

Description:

Writes a binary value to one of the analog output channels, changing the current produced at the channel.

Syntax:

WORD ECDAQ_WriteAOCurrentH(DWORD dwDeviceNo, WORD wAOChannel, WORD wAORange, DWORD dwValue)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAOChannel	WORD	IN	Analog output channel number
wAORange	WORD	IN	Analog output range. The definition is shown below.
dwValue	DWORD	IN	Binary data to be written

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Macro Definition	Value	Description
ECDAQ_AO_UNI_10MA	4	Current Range 0 ~ 10mA
ECDAQ_AO_BI_10MA	5	Current Range ± 10 mA
ECDAQ_AO_UNI_20MA	6	Current Range 0 ~ 20mA
ECDAQ_AO_BI_20MA	7	Current Range ± 20 mA
ECDAQ_AO_UNI_4TO20MA	8	Current Range 4 ~ 20mA
ECDAQ_AO_BI_24MA	9	Current Range ± 24 mA

Example:

[C/C++]

```
int ret = 0;
```

```
DWORD DeviceNo = 0;
```

```
WORD AOChannel= 0;
WORD AORange = 6; // 0 ~ 20mA
DWORD Value = 255;

ret = ECDAQ_WriteAOCurrenH (DeviceNo, AOChannel, AORange, Value);
if(ret < 0)
{
    printf("Failed to Write AO current:%d\n",ret);
}
else
{
    printf("Write AO current successfully!\n");
}
```

ECDAQ_ReadAIH

Description:

Reads an analog input channel and returns one un-scaled result.

Syntax:

WORD ECDAQ_ReadAIH(DWORD dwDeviceNo, WORD wChannel, DWORD *dwValue)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wChannel	WORD	IN	Analog input channel number
dwValue	DWORD*	OUT	Raw data reading from sampled channel

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD Channel= 0;
DWORD Value;

ret = ECDAQ_ReadAIH(DeviceNo, Channel, &Value);
if(ret < 0)
{
    printf("Failed to Read AI raw data:%d\n",ret);
}
else
{
    printf("Read AI raw data successfully!\n");
}
```

ECDAQ_ReadAIScanH

Description:

Reads all analog input channels and returns one un-scaled result.

Syntax:

WORD ECDAQ_ReadAIALH(DWORD dwDeviceNo, WORD wTotalChannels, DWORD
*dwValue)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wTotalChannels	WORD	IN	Total analog input channel number
dwValue	DWORD*	OUT	The value of the sampled raw data from each channel

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD TotalChannels=4;

DWORD Value[4];

ret = ECDAQ_ReadAIALLH(DeviceNo, TotalChannels, Value);

if(ret < 0)
{
    printf("Failed to Read all AI raw data:%d\n",ret);
}
else
{
    printf("Read all AI raw data successfully!\n");
}
```

5. Motion Control Functions

Axis Parameter Settings

ECDAQ_MC_SetAxisServoON

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:

WORD ECDAQ_MC_SetAxisServoON(DWORD dwDeviceNo, WORD wAxisNo, UCHAR bServoOn)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
bServoOn	UCHAR	IN	Servo state 0: OFF 1: ON

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD AxisNo = 0;
UCHAR State = 1;
ret = ECDAQ_MC_SetAxisServoON (DeviceNo, AxisNo, State);
if(ret < 0)
{
    printf("Failed to set axis ServoOn:%d\n",ret);
}
else
{
    printf("Set axis ServoOn successfully!\n");
}
```

ECDAQ_MC_SetAxisMicroStep

Description:

Set the microstep value of an axis.

Note: This function only supports ECAT-2091S, ECAT-2094S, ECAT-2094DS and ECAT-2094P.

Syntax:

WORD ECDAQ_MC_SetAxisMicroStep(DWORD dwDeviceNo, WORD wAxisNo, WORD wMicroStep)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
wMicroStep	WORD	IN	Number of microsteps per full step. The corresponding microsteps of enumerated value 0 to 8 is: 1, 2, 4, 8, 16, 32, 64, 128, 256.

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD AxisNo = 0;
WORD MicroStep = 8;
ret = ECDAQ_MC_SetAxisMicroStep (DeviceNo, AxisNo, MicroStep);
if(ret < 0)
{
    printf("Failed to set axis MicroStep:%d\n",ret);
}
else
{
    printf("Set axis MicroStep successfully!\n");
}
```

ECDAQ_MC_GetAxisMicroStep

Description:

Get the microstep value of an axis.

Note: This function only supports ECAT-2091S, ECAT-2094S, ECAT-2094DS and ECAT-2094P.

Syntax:

```
WORD ECDAQ_MC_GetAxisMicroStep(DWORD dwDeviceNo, WORD  
wAxisNo, WORD *wMicroStep)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
wMicroStep	WORD*	OUT	Number of microsteps per full step. The corresponding microsteps of enumerated value 0 to 8 is: 1, 2, 4, 8, 16, 32, 64, 128, 256.

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD AxisNo = 0;
WORD MicroStep = 0;
ret = ECDAQ_MC_GetAxisMicroStep (DeviceNo, AxisNo, &MicroStep);
if(ret < 0)
{
    printf("Failed to get axis MicroStep:%d\n",ret);
}
else
{
    printf("Get axis MicroStep successfully!\n");
}
```

ECDAQ_MC_SetAxisCurrent

Description:

Set the current of an axis in each state.

Note: This function only supports ECAT-2091S and ECAT-2094S.

Syntax:

WORD ECDAQ_MC_SetAxisCurrent(DWORD dwDeviceNo, WORD wAxisNo, WORD wMax, WORD wHold, WORD wPowerOn, WORD wSafe)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
wMax	WORD	IN	Peak motor coil current for driving (unit: mA). When high-speed operation will cause loss of step, please increase this value
wHold	WORD	IN	Motor standstill current (unit: mA) When servo on and motor not running, the motor uses this current
wPowerOn	WORD	IN	Motor coil current output directly after power on (unit: mA) When not servo on, the motor uses this current
wSafe	WORD	IN	Set the safe motor coil current (will be applied if state changes from OP to a different state) (unit: mA)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

WORD MaxCurrent = 750;

WORD HoldCurrent = 500;

WORD PowerOnCurrent = 375;

WORD SafeCurrent = 375;


ret = MC_SetAxisCurrent(DeviceNo, AxisNo, MaxCurrent, HoldCurrent, PowerOnCurrent,
SafeCurrent);
if(ret < 0)
{
    printf("Failed to set axis current:%d\n",ret);
}
else
{
    printf("Set axis current successfully!\n");
}
```

ECDAQ_MC_GetAxisCurrent

Description:

Get the current of an axis in each state.

Note: This function only supports ECAT-2091S and ECAT-2094S.

Syntax:

```
WORD ECDAQ_MC_GetAxisCurrent(DWORD dwDeviceNo, WORD  
wAxisNo, WORD *wMax, WORD *wHold, WORD *wPowerOn, WORD *wSafe)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
wMax	WORD*	OUT	Peak motor coil current for driving (unit: mA). When high-speed operation will cause loss of step, please increase this value
wHold	WORD*	OUT	Motor standstill current (unit: mA) When servo on and motor not running, the motor uses this current
wPowerOn	WORD*	OUT	Motor coil current output directly after power on (unit: mA) When not servo on, the motor uses this current
wSafe	WORD*	OUT	Set the safe motor coil current (will be applied if state changes from OP to a different state) (unit: mA)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD AxisNo = 0;
WORD MaxCurrent, HoldCurrent, PowerOnCurrent, SafeCurrent;

ret = MC_SetAxisCurrent(DeviceNo, AxisNo, &MaxCurrent, &HoldCurrent, &PowerOnCurrent,
&SafeCurrent);
if(ret < 0)
{
    printf("Failed to get axis current:%d\n",ret);
}
else
{
    printf("Get axis current successfully!\n");
}
```

Axis Status

ECDAQ_MC_GetAxisActualPos

Description:

Get actual position of an axis.

Syntax:

WORD ECDAQ_MC_GetAxisActualPos (DWORD dwDeviceNo, WORD wAxisNo, long *lPos)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
lPos	long*	OUT	Actual position (Unit: pulse)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

long AxisPos;

ret = ECDAQ_MC_GetAxisActualPos(DeviceNo, AxisNo, & AxisPos);

if(ret < 0)
{
    printf("Failed to get axis position:%d\n",ret);
}
else
{
    printf("Axis position:%d\n", AxisPos );
}
```

ECDAQ_MC_GetAxisActualEnc

Description:

Get actual encoder value of an axis.

Syntax:

```
WORD ECDAQ_MC_GetAxisActualEnc(DWORD dwDeviceNo, WORD  
wAxisNo, long *lEnc)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
lEnc	long*	OUT	Actual encoder value (Unit: pulse)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

long AxisEnc;

ret = ECDAQ_MC_GetAxisActualEnc (DeviceNo, AxisNo, & AxisEnc);

if(ret < 0)
{
    printf("Failed to get axis encoder:%d\n",ret);
}
else
{
    printf("Axis encoder:%d\n", AxisEnc);
}
```

ECDAQ_MC_GetAxisActualVel

Description:

Get actual velocity of an axis.

Syntax:

WORD ECDAQ_MC_GetAxisActualVel(DWORD dwDeviceNo, WORD wAxisNo, INT *lVel)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
lVel	INT *	OUT	Actual velocity (Unit: pulse/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

long AxisVel;

ret = ECDAQ_MC_GetAxisActualVel(DeviceNo, AxisNo, & AxisVel);

if(ret < 0)
{
    printf("Failed to get axis velocity:%d\n",ret);
}
else
{
    printf("Axis velocity:%d\n", AxisVel);
}
```

ECDAQ_MC_GetAxisActualTor

Description:

Get actual torque of an axis.

Syntax:

```
WORD ECDAQ_MC_GetAxisActualTor(DWORD dwDeviceNo, WORD  
wAxisNo, SHORT *sTor)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
sTor	SHORT *	OUT	Actual Torque Please refer to the object "Torque Actual Value: 6077h" in the driver manual for the unit, generally 0.1% of the rated torque For rated torque, please refer to "Motor rated torque: 6076h"

Return:

0: Success.

Others: Refer to Appendix "Error Codes"

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

SHORT AxisTor;

ret = ECDAQ_MC_GetAxisActualTor(DeviceNo, AxisNo, &AxisTor);

if(ret < 0)
{
    printf("Failed to get axis velocity:%d\n",ret);
}
else
{
    printf("Axis velocity:%d\n", AxisVel);
}
```

ECDAQ_MC_GetAxisState

Description:

Get the state of an axis.

Syntax:

```
WORD ECDAQ_MC_GetAxisState(DWORD dwDeviceNo, WORD  
wAxisNo, UCHAR *bState)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
bState	UCHAR*	OUT	Axis state The definition is shown below.

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

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

Example:

[C/C++]

```
int ret = 0;
char buf[512];
DWORD DeviceNo = 0;
WORD AxisNo = 0;
UCHAR AxisState;
ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
if(ret < 0)
{
    printf("Failed to get axis state:%d\n",ret);
}
else
{
    switch(AxisState)
    {
        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);
}
```

ECDAQ_MC_GetAxisDI

Description:

Get digital inputs of an axis.

Syntax:

```
WORD ECDAQ_MC_GetAxisDI(DWORD dwDeviceNo, WORD wAxisNo,  
WORD *wDI)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
wDI	WORD *	OUT	Digital input status The definition is shown below. 0: OFF 1: ON

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

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	Busy (Command processing)
Bit 7~15	Reserved

Example:

[C/C++]

```
typedef struct axis_di{
    union
    {
        struct
        {
            char NOT      : 1;      //Negative limit switch
            char POT      : 1;      //Positive limit switch
            char ORG      : 1;      //home switch
            char ALM      : 1;      //alarm
            char WAN      : 1;      //warning
            char SVN      : 1;      //serve on status
            char Busy     : 1;      //processing cmd
            WORD reserved : 9;      //Reserved(bit7~bit15)
        };
        WORD DIs;
    };
}axis_di_t;

/*****/
int ret = 0;
axis_di_t AxisDI;
WORD DeviceNo = 0;
WORD 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"
"          |-Busy:%d\n"
"\n", AxisNo, AxisDI.NOT, AxisDI.POT, AxisDI.ORG
, AxisDI.ALM, AxisDI.WAN, AxisDI.SVN, AxisDI.Busy);
}
```

ECDAQ_MC_GetAxisLastError

Description:

Get last error of an axis.

Syntax:

```
WORD ECDAQ_MC_GetAxisLastError(DWORD dwDeviceNo, WORD  
wAxisNo, INT* iError)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
iError	INT*	OUT	Last error (Refer to Appendix "Error Codes")

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD AxisNo = 0;
UCHAR AxisState;
INT Error;
ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, &AxisState);
if(ret < 0)
{
    printf("Failed to get axis state:%d\n",ret);
}
else
{
    if(AxisState == MC_AS_ERRORSTOP)
    {
        ret = ECDAQ_MC_GetAxisLastError(DeviceNo, AxisNo, &Error);
        if(ret < 0)
        {
            printf("Failed to get axis last error:%d\n",ret);
        }
        else
        {
            printf("Axis last error:%d\n", Error);
        }
    }
}
```

ECDAQ_MC_GetAxisDriverError

Description:

Get the driver error of an axis.

Syntax:

WORD ECDAQ_MC_GetAxisDriverError(DWORD dwDeviceNo, WORD wAxisNo, WORD* wError)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
wError	WORD*	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++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

UCHAR AxisState;

INT Error;

WORD DriverError;

ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, &AxisState);

if(ret < 0)
{
    printf("Failed to get axis state:%d\n",ret);
}else{
    if(AxisState == MC_AS_ERRORSTOP)
    {
        ret = ECDAQ_MC_GetAxisLastError(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 == ECDAQ_ERR_MC_DRIVER_FAULT)
            {
                ret = ECDAQ_MC_GetAxisDriverError(DeviceNo, AxisNo, &DriverError);
                if(ret < 0)
                {
                    printf("Failed to get axis driver error:%d\n",ret);
                }else{
                    printf("Axis driver error:%d\n", DriverError);
                }
            }
        }
    }
}

}
```

Axis Homing

ECDAQ_MC_AxisHomeStart

Description:

Start home motion of an axis.

Syntax:

WORD MC_AxisHomeStart(DWORD dwDeviceNo, WORD wAxisNo, WORD wMethod, DWORD dwSearchSw, DWORD dwSearchZr, DWORD dwAcc, DWORD dwDec, long lPosOffset, UCHAR bUseEnc, double dEncMotorRatio)

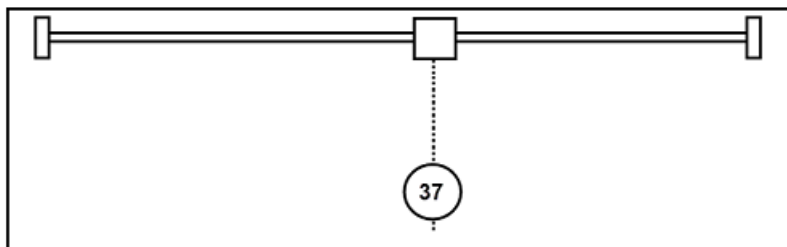
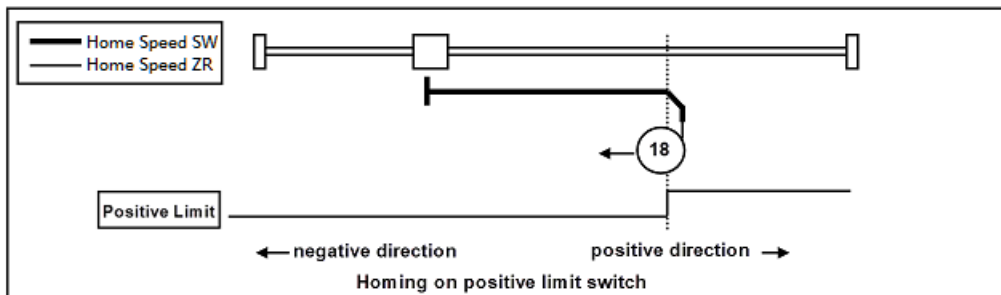
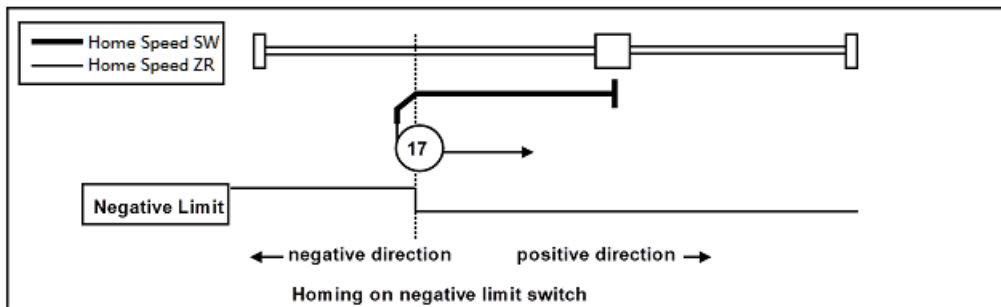
Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
wMethod	WORD	IN	Homing method ECAT-2091S/ECAT-2094S support: 17,18,37
dwSearchSw	DWORD	IN	Speed during search for switch (Unit: pulse/s)
dwSearchZr	DWORD	IN	Speed during search for z phase signal (Unit: pulse/s)
dwAcc	DWORD	IN	Homing Acceleration (Unit: pulse/s ²)
dwDec	DWORD	IN	Homing Deceleration (Unit: pulse/s ²)
lPosOffset	long	IN	Home offset (Unit: pulse)
bUseEnc	UCHAR	IN	Using external encoder 0: No, 1: Yes Note: Only for ECAT-2091S/ECAT-2094S
dEncMotorRatio	double	IN	The ratio between encoder and motor. For Example: The encoder's PPR (pulses per revolution) is 51200, and the motor's PPR is 4000, then the ratio between encoder and motor

Return:

0: Success.

Others: Refer to Appendix "Error Codes".



Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

WORD Method = 17;

WORD wSearchSw = 1000;

WORD wSearchZr = 100;

WORD Acc = 100;

WORD Dec = 100;

WORD PosOffset = 0;

UCHAR UseEnc = 1;

double EncMotorRatio = 4000.0/51200.0;

ret = MC_AxisHomeStart(DeviceNo, AxisNo, Method, SearchSw, SearchZr, Acc, Dec, PosOffset,
UseEnc, EncMotorRatio);

;

if(ret < 0)
{
    printf("Axis homeStart failed:%d\n",ret);
}
else
{
    printf("Axis homeStart success!\n");
}
```

Axis Moving

ECDAQ_MC_ResetAxisError

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:

```
WORD ECDAQ_MC_ResetAxisError(DWORD dwDeviceNo, WORD wAxisNo)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD AxisNo = 0;
ret = ECDAQ_MC_ResetAxisError (DeviceNo, AxisNo);
if(ret < 0)
{
    printf("Failed to reset axis error:%d\n",ret);
}
else
{
    printf("Reset axis error successfully!\n ");
}
```

ECDAQ_MC_ResetAxisCounter

Description:

Reset position or encoder counter of an axis.

Syntax:

```
WORD ECDAQ_MC_ResetAxisCounter(DWORD dwDeviceNo, WORD  
wAxisNo, UCHAR bResetPos, UCHAR bResetEnc)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
bResetPos	UCHAR	IN	0: Position not changed 1: Reset position
bResetEnc	UCHAR	IN	0: Encoder counter not changed 1: Reset encoder Note: Only for ECAT-2091S/ECAT-2094S

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

WORD ResetPos = 1;

WORD ResetEnc = 1;


ret = ECDAQ_MC_ResetAxisCounter (DeviceNo, AxisNo, ResetPos, ResetEnc);
if(ret < 0)
{
    printf("Failed to reset axis counter:%d\n",ret);
}
else
{
    printf("Reset axis counter successfully!\n ");
}
}
```

ECDAQ_MC_AxisMoveAbs

Description:

Start an absolute position motion of an axis.

Syntax:

WORD ECDAQ_MC_AxisMoveAbs(DWORD dwDeviceNo, WORD wAxisNo, DWORD dwVel, DWORD dwAcc, DWORD dwDec, long lPos)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
dwVel	DWORD	IN	Velocity (Unit: pulse/s)
dwAcc	DWORD	IN	Acceleration (Unit: pulse/s ²)
dwDec	DWORD	IN	Deceleration (Unit: pulse/s ²)
lPos	long	IN	Absolute position (Unit: pulse)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;
WORD AxisNo = 0;
UCHAR AxisState;
DWORD AxisVel = 1000;
WORD Acc = 100;
WORD Dec = 100;
WORD AxisPos = 10000;

ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
if(AxisState == MC_AS_STANDSTILL) //StandStill
{
    ret = ECDAQ_MC_AxisMoveAbs(DeviceNo, AxisNo, AxisVel, Acc, Dec, AxisPos);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
        }while(AxisState == MC_AS_DISCRETEMOTION) //DiscreteMotion

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

ECDAQ_MC_AxisMoveRel

Description:

Start a relative position motion of an axis

Syntax:

WORD ECDAQ_MC_AxisMoveRel(DWORD dwDeviceNo, WORD wAxisNo, DWORD dwVel, DWORD dwAcc, DWORD dwDec, long lDist)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
dwVel	DWORD	IN	Velocity (Unit: pulse/s)
dwAcc	DWORD	IN	Acceleration (Unit: pulse/s ²)
dwDec	DWORD	IN	Deceleration (Unit: pulse/s ²)
lDist	long	IN	Relative distance (Unit: pulse)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;
WORD AxisNo = 0;
UCHAR AxisState;
DWORD AxisVel = 1000;
WORD Acc = 100;
WORD Dec = 100;
WORD AxisDist = 10000;

ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
if(AxisState == MC_AS_STANDSTILL) //StandStill
{
    ret = ECDAQ_MC_AxisMoveRel(DeviceNo, AxisNo, AxisVel, Acc, Dec, AxisDist);
    if(ret < 0)
    {
        printf("Failed to start axis move rel:%d\n", ret);
    }
    else
    {
        do
        {
            sleep(1);
            ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
        }while(AxisState == MC_AS_DISCRETEMOTION) //DiscreteMotion

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


ECDAQ_MC_AxisChangePos

Description:

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

Syntax:

```
WORD ECDAQ_MC_AxisChangePos(DWORD dwDeviceNo, WORD  
wAxisNo, long lPos)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
lPos	long	IN	End position (Unit: pulse) It is an absolute position

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

UCHAR AxisState;

DWORD AxisVel = 1000;

WORD Acc = 100;

WORD Dec = 100;

WORD AxisPos = 10000;

ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
if(AxisState == MC_AS_STANDSTILL) //StandStill
{
    ret = ECDAQ_MC_AxisMoveAbs(DeviceNo, AxisNo, AxisVel, Acc, Dec, AxisPos);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        sleep(3);
        ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
        if (AxisState == MC_AS_DISCRETEMOTION) //DiscreteMotion
        {
            AxisPos = 20000;
            ret = ECDAQ_MC_AxisChangePos(DeviceNo, AxisNo, AxisPos);
            if(ret < 0)
            {
                printf("Failed to call axis cahnge position function:%d\n", ret);
            }
        }

        do
        {
            sleep(1);
```

```
    ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);  
    }while(AxisState == MC_AS_DISCRETEMOTION) //DiscreteMotion  
  
    if(AxisState == MC_AS_STANDSTILL) //StandStill  
        printf("Axis move successfully!\n");  
    else if(AxisState == MC_AS_ERRORSTOP) //ErrorStop  
    {  
        printf("Axis error stop\n");  
    }  
}  
}
```



ECDAQ_MC_AxisChangeVel

Description:

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

Syntax:

WORD ECDAQ_MC_AxisChangeVel(DWORD dwDeviceNo, WORD wAxisNo, INT iVel)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
iVel	INT	IN	Velocity (Unit: pulse/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;
WORD AxisNo = 0;
UCHAR AxisState;
DWORD AxisVel = 1000;
WORD Acc = 100;
WORD Dec = 100;
WORD AxisPos = 10000;

ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
if(AxisState == MC_AS_STANDSTILL) //StandStill
{
    ret = ECDAQ_MC_AxisMoveAbs(DeviceNo, AxisNo, AxisVel, Acc, Dec, AxisPos);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        sleep(3);
        ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
        if (AxisState == MC_AS_DISCRETEMOTION) //DiscreteMotion
        {
            AxisVel = 2000;
            ret = ECDAQ_MC_AxisChangeVel(DeviceNo, AxisNo, AxisVel);
            if(ret < 0)
            {
                printf("Failed to call axis cahnge position function:%d\n", ret);
            }
        }

        do
        {
            sleep(1);
```

```
    ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);  
    }while(AxisState == MC_AS_DISCRETEMOTION) //DiscreteMotion  
  
    if(AxisState == MC_AS_STANDSTILL) //StandStill  
        printf("Axis move successfully!\n");  
    else if(AxisState == MC_AS_ERRORSTOP) //ErrorStop  
    {  
        printf("Axis error stop\n");  
    }  
}  
}
```



ECDAQ_MC_AxisMoveVel

Description:

Start a never ending movement with a specified velocity.

Syntax:

WORD ECDAQ_MC_AxisMoveVel(DWORD dwDeviceNo, WORD wAxisNo, INT iVel, DWORD dwAcc, DWORD dwDec)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
iVel	INT	IN	Velocity (Unit: pulse/s)
dwAcc	DWORD	IN	Acceleration (Unit: pulse/s ²)
dwDec	DWORD	IN	Deceleration (Unit: pulse/s ²)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

UCHAR AxisState;

long AxisVel = -1000;

WORD Acc = 100;

WORD Dec = 100;

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

ECDAQ_MC_AxisMove_CiA402_PV

Description:

Start the single-axis motion in CiA402 profile velocity mode.

Syntax:

WORD ECDAQ_AxisMove_CiA402_PV(DWORD dwDeviceNo, WORD
wAxisNo, long lVel, DWORD dwAcc, DWORD dwDec)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
iVel	INT	IN	Velocity (Unit: pulse/s)
dwAcc	DWORD	IN	Acceleration (Unit: pulse/s ²)
dwDec	DWORD	IN	Deceleration (Unit: pulse/s ²)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".



Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

UCHAR AxisState;

long AxisVel = -1000;

WORD Acc = 100;

WORD Dec = 100;


ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, &AxisState);
if(AxisState == MC_AS_STANDSTILL) //StandStill
{
    ret = ECDAQ_AxisMove_CiA402_PV(DeviceNo, AxisNo, AxisVel, Acc, Dec);
    if(ret < 0)
    {
        printf("Failed to start axis move CiA402_PV:%d\n", ret);
    }
}
```

ECDAQ_MC_AxisMove_CiA402_PT

Description:

Start the single-axis motion in CiA402 profile torque mode.

Syntax:

WORD ECDAQ_MC_AxisMove_CiA402_PT(DWORD dwDeviceNo, WORD wAxisNo, SHORT sTor, DWORD dwSlpoe)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number
sTor	SHORT	IN	Target torque, 0.1% of the maximum rated torque, which setting range is 1~1000.
dwSlpoe	DWORD	IN	Torque slope (0.1%/s)

Return:

0: Success.

Others: Refer to Appendix "Error Codes".



Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

UCHAR AxisState;

SHORT Torque = 30;

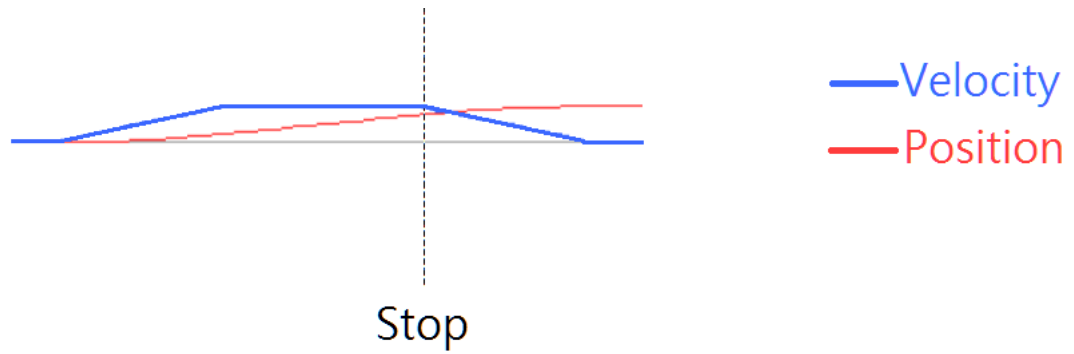
DWORD Slope = 10;


ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, &AxisState);
if(AxisState == MC_AS_STANDSTILL) //StandStill
{
    ret = ECDAQ_MC_AxisMove_CiA402_PT(DeviceNo, AxisNo, Torque, Slope);
    if(ret < 0)
    {
        printf("Failed to start axis move CiA402_PT:%d\n", ret);
    }
}
```

ECDAQ_MC_AxisStop

Description:

Stop an axis with deceleration.



Syntax:

```
WORD ECDAQ_MC_AxisStop(DWORD dwDeviceNo, WORD wAxisNo)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

UCHAR AxisState;

DWORD AxisVel = 1000;

WORD Acc = 500;

WORD Dec = 500;

WORD AxisPos = 10000;

ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
if(AxisState == MC_AS_STANDSTILL) //StandStill
{
    ret = ECDAQ_MC_AxisMoveAbs(DeviceNo, AxisNo, AxisVel, Acc, Dec, AxisPos);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        sleep(3);
        ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
        if (AxisState == MC_AS_DISCRETEMOTION) //DiscreteMotion
        {
            ret = ECDAQ_MC_AxisStop(DeviceNo, AxisNo);
            if(ret < 0)
            {
                printf("Failed to stop axis move:%d\n", ret);
            }
        }

        do
        {
            sleep(1);
            ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
```

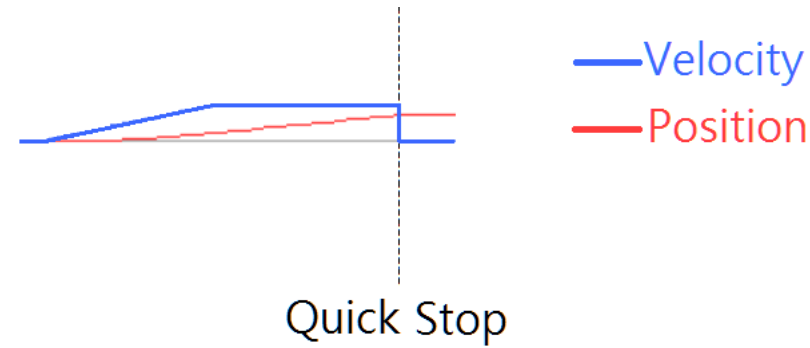
```
}while(AxisState == MC_AS_STOPPING) //Stopping

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

ECDAQ_MC_AxisQuickStop

Description:

Stop an axis quickly (immediately).



Syntax:

```
WORD ECDAQ_MC_AxisQuickStop(DWORD dwDeviceNo, WORD  
wAxisNo)
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wAxisNo	WORD	IN	Axis number

Return:

0: Success.
Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

UCHAR AxisState;

DWORD AxisVel = 1000;

WORD Acc = 500;

WORD Dec = 500;

WORD AxisPos = 10000;

ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
if(AxisState == MC_AS_STANDSTILL) //StandStill
{
    ret = ECDAQ_MC_AxisMoveAbs(DeviceNo, AxisNo, AxisVel, Acc, Dec, AxisPos);
    if(ret < 0)
    {
        printf("Failed to start axis move abs:%d\n", ret);
    }
    else
    {
        sleep(3);
        ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
        if (AxisState == MC_AS_DISCRETEMOTION) //DiscreteMotion
        {
            ret = ECDAQ_MC_AxisQuickStop(DeviceNo, AxisNo);
            if(ret < 0)
            {
                printf("Failed to stop axis move:%d\n", ret);
            }
        }

        do
        {
            sleep(1);
            ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, & AxisState);
```

```
}while(AxisState == MC_AS_STOPPING) //Stopping

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



StartUp Parameter Settings

ECDAQ_MC_SetLoadDefault

Description:

Set the startup SDO list data 0x8000 to 0x8321 of ECAT-2091S/ECAT-2094S for loading factory default or user configured setting. If set load factory default , the process completed immediately. If set load user configured setting, the device will take more than 10 seconds to save SDO list data to inter EEPROM.

After executing the ECDAQ_MC_SetLoadDefault funtion, user can call the function ECDAQ_MC_GetLoadDefaultState to get the process state. When the state is ECMST_ERR_NO_ERROR, it means the startup setting is completed.

It is recommend to power cycling the module after completing the startup parameters setting to ensure stable operation.

Note: This function only supports ECAT-2091S and ECAT-2094S.

Syntax:

WORD ECDAQ_MC_SetLoadDefault(DWORD dwDeviceNo, UCHAR bDefault)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
bDefault	UCHAR	IN	startup setting 0: Load user configured setting 1: Load factory default

Return:

0: Success.

Others: Refer to Appendix "Error Codes".



Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
UCHAR Default = 1;
ret = ECDAQ_MC_SetLoadDefault(DeviceNo, Default);
if(ret < 0)
{
    printf("Failed to set load default:%d\n",ret);
}
else
{
    printf("Set load default successfully!\n");
}
```

ECDAQ_MC_GetLoadDefaultState

Description:

Get the process state while the startup parameters setting of ECAT-2091S/ECAT-2094S is processing. When the state is ECMST_ERR_NO_ERROR, it means the startup setting is completed.

Note: This function only supports ECAT-2091S and ECAT-2094S.

Syntax:

WORD ECDAQ_MC_GetLoadDefaultState(DWORD dwDeviceNo)

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".



Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

ret = ECDAQ_MC_SetLoadDefault(DeviceNo, Default);

if(ret < 0)
{
    printf("Failed to set load default:%d\n",ret);
}
else
{
    do
    {
        sleep(1000);
        ret = ECDAQ_MC_GetLoadDefaultState (DeviceNo);
        printf("Busy in setting initial configuration!\n");
    }while(ret != ECMST_ERR_NO_ERR) //Busy
    printf("Set load default finished!\n");
}
```

Encoder Operation

ECDAQ_GetEncCountingMode

Description:

Get the encoder counting mode.

Note: This function only supports ECAT-2092T and ECAT-2093.

Syntax:

```
WORD ECDAQ_GetEncCountingMode(DWORD dwDeviceNo, WORD  
wEncNo, WORD* wMode);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wMode	WORD	OUT	Encoder Counting Mode 1: CW/CCW 2: Pulse/Dir 3: A/B Phase

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD Mode = 0;
ret = ECDAQ_GetEncCountingMode (DeviceNo, EncNo, & Mode);
if(ret < 0)
{
    printf("Failed to get encoder counting mode:%d\n",ret);
}
else
{
    printf("Get encoder counting mode successfully!\n");
}
```

ECDAQ_SetEncCountingMode

Description:

Set the encoder counting mode.

Note: This function only supports ECAT-2092T and ECAT-2093.

Syntax:

```
WORD ECDAQ_SetEncCountingMode(DWORD dwDeviceNo, WORD  
wEncNo, WORD wMode);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wMode	WORD	IN	Encoder Counting Mode 1: CW/CCW 2: Pulse/Dir 3: A/B Phase

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD Mode = 1; //CW/CCW
ret = ECDAQ_SetEncCountingMode (DeviceNo, EncNo, Mode);
if(ret < 0)
{
    printf("Failed to set encoder counting mode:%d\n",ret);
}
else
{
    printf("Set encoder counting mode successfully!\n");
}
```

ECDAQ_GetEncCount

Description:

Get the encoder counter value.

Note: This function only supports ECAT-2092T and ECAT-2093.

Syntax:

```
WORD ECDAQ_GetEncCount(DWORD dwDeviceNo, WORD wEncNo,  
INT* iCnt);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
iCnt	INT	OUT	Encoder counter value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo= 0;
INT Cnt= 0;
ret = ECDAQ_GetEncCount (DeviceNo, EncNo, & Cnt);
if(ret < 0)
{
    printf("Failed to get encoder count:%d\n",ret);
}
else
{
    printf("Get encoder count successfully!\n");
}
```

ECDAQ_SetEncCount

Description:

Set the encoder counter value.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_SetEncCount(DWORD dwDeviceNo, WORD wEncNo,  
INT iCnt);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
iCnt	INT	IN	Encoder counter value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo= 0;
INT Cnt= 0;
ret = ECDAQ_SetEncCount (DeviceNo, EncNo, Cnt);
if(ret < 0)
{
    printf("Failed to set encoder count:%d\n",ret);
}
else
{
    printf("Set encoder count successfully!\n");
}
```

ECDAQ_GetEncLatchCount

Description:

Get the index latch count of encoder.

Note: This function only supports ECAT-2092T and ECAT-2093.

Syntax:

```
WORD ECDAQ_GetEncLatchCount(DWORD dwDeviceNo, WORD  
wEncNo, INT* iCnt);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
iCnt	INT	OUT	Latch count

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo= 0;
INT Cnt= 0;
ret = ECDAQ_GetEncLatchCount (DeviceNo, EncNo, & Cnt);
if(ret < 0)
{
    printf("Failed to get encoder latch count:%d\n",ret);
}
else
{
    printf("Get encoder latch count successfully!\n");
}
```

ECDAQ_GetEncLatchTriggerCount

Description:

Get the index latch trigger count of encoder.

Note: This function only supports ECAT-2093.

Syntax:

```
WORD ECDAQ_GetEncLatchTriggerCount(DWORD dwDeviceNo, WORD  
wEncNo, WORD* wTrigCnt);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wTrigCnt	WORD	OUT	Counts the number of latch events. With each latch event the counter increases by one.

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD TrigCnt = 0;
ret = ECDAQ_GetEncLatchTriggerCount (DeviceNo, EncNo, & TrigCnt);
if(ret < 0)
{
    printf("Failed to get encoder latch trigger count:%d\n",ret);
}
else
{
    printf("Get encoder latch trigger count successfully!\n");
}
```

ECDAQ_GetEncExLatchCount

Description:

Get the external latch count of encoder.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_GetEncExLatchCount(DWORD dwDeviceNo, WORD  
wEncNo, INT* iRising, INT* iFalling);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
iRising	INT	OUT	Rising edge latched value
iFalling	INT	OUT	Falling edge latched value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo= 0;
INT CntR= 0;
INT CntF= 0;
ret = ECDAQ_GetEncExLatchCount (DeviceNo, EncNo, & CntR, & CntF);
if(ret < 0)
{
    printf("Failed to get encoder external latch count:%d\n",ret);
}
else
{
    printf("Get encoder external latch count successfully!\n");
}
```

ECDAQ_ResetEncCount

Description:

Clear the encoder counter value to 0.

Note: This function only supports ECAT-2092T and ECAT-2093.

Syntax:

```
WORD ECDAQ_ResetEncCount(DWORD dwDeviceNo, WORD wEncNo);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo= 0;
ret = ECDAQ_ResetEncCount (DeviceNo, EncNo);
if(ret < 0)
{
    printf("Failed to reset encoder count:%d\n",ret);
}
else
{
    printf("Reset encoder count successfully!\n");
}
```

ECDAQ_SetEnableEncLatch

Description:

Set the enable index latch parameter of encoder.

Note: (1) This function only supports ECAT-2093.

(2) The index latch of ECAT-2092T is always enabled and cannot be disabled.

Syntax:

```
WORD ECDAQ_SetEnableEncLatch(DWORD dwDeviceNo, WORD  
wEncNo, WORD wEnable);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wEnable	WORD	IN	Enable(1)/Disable(0) latch

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD Enable = 1;
ret = ECDAQ_SetEnableEncLatch (DeviceNo, EncNo, Enable);
if (ret < 0)
{
    printf("Failed to set enable encoder index latch:%d\n",ret);
}
else
{
    printf("Set enable encoder index latch successfully!\n");
}
```

ECDAQ_GetEnableEncLatch

Description:

Get the enable index latch parameter of encoder.

Note: (1) This function only supports ECAT-2093.

(2) The index latch of ECAT-2092T is always enabled and cannot be disabled.

Syntax:

```
WORD ECDAQ_GetEnableEncLatch(DWORD dwDeviceNo, WORD  
wEncNo, WORD* wEnable);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wEnable	WORD	OUT	Enable(1)/Disable(0) latch

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD Enable = 0;
ret = ECDAQ_GetEnableEncLatch (DeviceNo, EncNo, & Enable);
if(ret < 0)
{
    printf("Failed to get enable encoder index latch:%d\n",ret);
}
else
{
    printf("Get enable encoder index latch successfully!\n");
}
```

ECDAQ_SetEnableEncExLatch

Description:

Set the enable external latch parameter of encoder.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_SetEnableEncExLatch(DWORD dwDeviceNo, WORD  
wEncNo, WORD wEnable);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wEnable	WORD	IN	Enable(1)/Disable(0) latch

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD Enable = 1;
ret = ECDAQ_SetEnableEncExLatch (DeviceNo, EncNo, Enable);
if(ret < 0)
{
    printf("Failed to set enable encoder external latch:%d\n",ret);
}
else
{
    printf("Set enable encoder external latch successfully!\n");
}
```

ECDAQ_GetEnableEncExLatch

Description:

Get the enable external latch parameter of encoder.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_GetEnableEncExLatch(DWORD dwDeviceNo, WORD  
wEncNo, WORD* wEnable);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wEnable	WORD	OUT	Enable(1)/Disable(0) latch

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD Enable = 0;
ret = ECDAQ_SetEnableEncExLatch (DeviceNo, EncNo, & Enable);
if(ret < 0)
{
    printf("Failed to get enable encoder external latch:%d\n",ret);
}
else
{
    printf("Get enable encoder external latch successfully!\n");
}
```

ECDAQ_ResetEncLatchCount

Description:

Reset the index latch count of encoder.

Note: This function only supports ECAT-2092T and ECAT-2093.

Syntax:

```
WORD ECDAQ_ResetEncLatchCount(DWORD dwDeviceNo, WORD  
wEncNo);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo= 0;
ret = ECDAQ_ResetEncLatchCount (DeviceNo, EncNo);
if(ret < 0)
{
    printf("Failed to reset encoder index latch count:%d\n",ret);
}
else
{
    printf("Reset encoder index latch count successfully!\n");
}
```

ECDAQ_ResetEncExLatchCount

Description:

Reset the external latch count of encoder.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_ResetEncExLatchCount(DWORD dwDeviceNo, WORD  
wEncNo);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
ret = ECDAQ_ResetEncExLatchCount (DeviceNo, EncNo);
if(ret < 0)
{
    printf("Failed to reset encoder external latch count:%d\n",ret);
}
else
{
    printf("Reset encoder external latch count successfully!\n");
}
```

ECDAQ_GetCmpTrigProperty

Description:

Get the compare-trigger related properties of encoder.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_GetCmpTrigProperty(DWORD dwDeviceNo, WORD  
wEncNo, WORD* wPulseWidth);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wPulseWidth	WORD	OUT	Compare Trigger Pulse Width Range: 2 to 32767 (ms), Default: 50

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo= 0;
WORD PulseWidth= 0;
ret = ECDAQ_GetCmpTrigProperty (DeviceNo, EncNo, & PulseWidth);
if(ret < 0)
{
    printf("Failed to get encoder compare trigger property:%d\n",ret);
}
else
{
    printf("Get encoder compare trigger property successfully!\n");
}
```

ECDAQ_SetCmpTrigProperty

Description:

Set the compare-trigger related properties of encoder.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_SetCmpTrigProperty(DWORD dwDeviceNo, WORD  
wEncNo, WORD wPulseWidth);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wPulseWidth	WORD	IN	Compare Trigger Pulse Width Range: 2 to 32767 (ms), Default: 50

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD PulseWidth = 100;
ret = ECDAQ_SetCmpTrigProperty (DeviceNo, EncNo, PulseWidth);
if (ret < 0)
{
    printf("Failed to set encoder compare trigger property:%d\n",ret);
}
else
{
    printf("Set encoder compare trigger property successfully!\n");
}
```

ECDAQ_GetEncNextComparePos

Description:

Get the next compare position of encoder.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_GetEncNextComparePos(DWORD dwDeviceNo, WORD  
wEncNo, WORD* wValid, INT* iPos);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wValid	WORD	OUT	The next compare position is valid
iPos	INT	OUT	Next compare position

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD Valid = 0;
INT Pos = 0;
ret = ECDAQ_GetEncNextComparePos (DeviceNo, EncNo, & Valid, & Pos);
if (ret < 0)
{
    printf("Failed to get encoder next compare position:%d\n",ret);
}
else
{
    printf("Get encoder next compare position successfully!\n");
}
```

ECDAQ_EnableEncEquidCompare

Description:

Enable a single or continuous compare-trigger function of encoder.

Note: (1) This function only supports ECAT-2092T.

(2) The single compare-trigger function is enable when the auto increment compare value is assigned to 0.

Syntax:

```
WORD ECDAQ_EnableEncEquidCompare(DWORD dwDeviceNo, WORD wEncNo, WORD wDir, INT iFirstPos, INT iDistance);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wDir	WORD	IN	Compare direction 0: positive direction 1: negative direction
iFirstPos	INT	IN	Start position for this compare-trigger operation
iDistance	INT	IN	Auto increment compare value

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo= 0;
WORD Dir= 0; //0: Positive direction
INT StartPos= 100;
INT Distance= 50;
ret = ECDAQ_EnableEncEquidCompare (DeviceNo, EncNo, Dir, StartPos, Distance);
if(ret < 0)
{
    printf("Failed to enable encoder equidist compare:%d\n",ret);
}
else
{
    printf("Enable encoder equidist compare successfully!\n");
}
```

ECDAQ_SetArrCmpPosTrigger

Description:

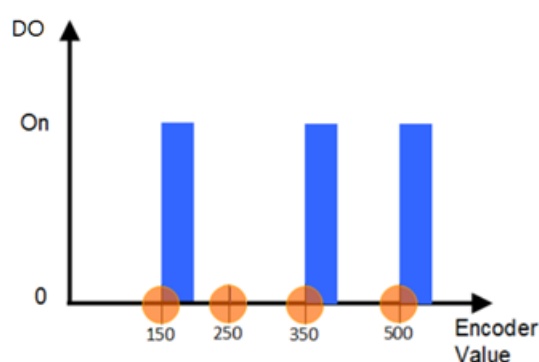
Set the array comparison position and the digital output trigger flag of encoder. After enabling the array comparison trigger function, the comparison will start from the comparison position stored in index 0. When the encoder reaches the comparison position, it will trigger the digital output if the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

Note: This function only supports ECAT-2092T.

Example:

In the following table 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

Index	0	1	2	3
Position	150	250	350	500
DO Trigger	1	0	1	1



Syntax:

```
WORD ECDAQ_SetArrCmpPosTrigger(DWORD dwDeviceNo, WORD  
wEncNo, WORD wIndex, INT iCmpPos, WORD wDoTrigger);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number

wIndex	WORD	IN	Index number Range: 0~199
iCmpPos	INT	IN	Compare position
wDoTrigger	WORD	IN	Enable(1)/Disable(0) digital output

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;

WORD Index = 0;
INT CmpPos = 150;
WORD DoTrigger = 1;
ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);
if(ret < 0)
{
    printf("Failed to set encoder array compare data:%d\n",ret);
}
else
{
    printf("Set encoder array compare data successfully!\n");
}

Index = 1;
CmpPos = 250;
DoTrigger = 0;
ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);
if(ret < 0)
{
    printf("Failed to set encoder array compare data:%d\n",ret);
}
else
{
    printf("Set encoder array compare data successfully!\n");
}

Index = 2;
CmpPos = 350;
DoTrigger = 1;
ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);
if(ret < 0)
```

```
{  
    printf("Failed to set encoder array compare data:%d\\n",ret);  
}  
else  
{  
    printf("Set encoder array compare data successfully!\\n");  
}  
  
Index = 3;  
CmpPos = 500;  
DoTrigger = 1;  
ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);  
if(ret < 0)  
{  
    printf("Failed to set encoder array compare data:%d\\n",ret);  
}  
else  
{  
    printf("Set encoder array compare data successfully!\\n");  
}  
}
```

ECDAQ_GetArrCmpPosTrigger

Description:

Get the array comparison position and the digital output trigger flag of encoder.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_GetArrCmpPosTrigger(DWORD dwDeviceNo, WORD  
wEncNo, WORD wIndex, INT* iCmpPos, WORD* wDoTrigger);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wIndex	WORD	IN	Index number Range: 0~199
iCmpPos	INT	OUT	Compare position
wDoTrigger	WORD	OUT	Enable(1)/Disable(0) digital output

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD EncNo= 0;


WORD Index = 0;
INT CmpPos = 0;
WORD DoTrigger = 0;
ret = ECDAQ_GetArrCmpPosTrigger (DeviceNo, EncNo, Index, &CmpPos, &DoTrigger);
if(ret < 0)
{
    printf("Failed to get encoder array compare data:%d\n",ret);
}
else
{
    printf("Get encoder array compare data successfully!\n");
}
```

ECDAQ_SetArrCmpEndIdx

Description:

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

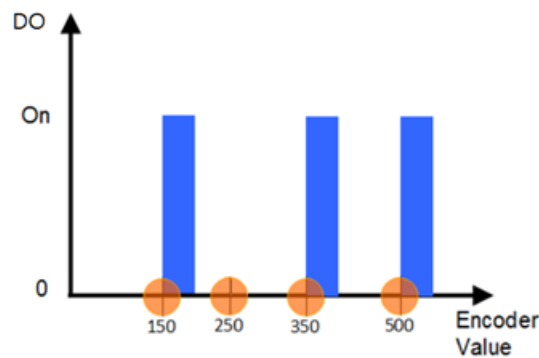
When the encoder position has reached the compare position stored in end index then the next compare position will start again with the position stored index 0.

Note: This function only supports ECAT-2092T.

Example:

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

Index	0	1	2	3
Position	150	250	350	500
DO Trigger	1	0	1	1
End Index	3			



Syntax:

```
WORD ECDAQ_SetArrCmpEndIdx(DWORD dwDeviceNo, WORD wEncNo, WORD wEndIndex);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wEndIndex	WORD	IN	索引值範圍

			範圍: 0~199
--	--	--	-----------

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;

WORD Index = 0;
INT CmpPos = 150;
WORD DoTrigger = 1;
ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);
if(ret < 0)
{
    printf("Failed to set encoder array compare data:%d\n",ret);
}
else
{
    printf("Set encoder array compare data successfully!\n");
}

Index = 1;
CmpPos = 250;
DoTrigger = 0;
ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);
if(ret < 0)
{
    printf("Failed to set encoder array compare data:%d\n",ret);
}
else
{
    printf("Set encoder array compare data successfully!\n");
}

Index = 2;
CmpPos = 350;
DoTrigger = 1;
ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);
if(ret < 0)
```

```
{
    printf("Failed to set encoder array compare data:%d\\n",ret);
}
else
{
    printf("Set encoder array compare data successfully!\\n");
}

Index = 3;
CmpPos = 500;
DoTrigger = 1;
ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);
if(ret < 0)
{
    printf("Failed to set encoder array compare data:%d\\n",ret);
}
else
{
    printf("Set encoder array compare data successfully!\\n");
}

WORD EndIndex = 3;
ret = ECDAQ_SetArrCmpEndIdx (DeviceNo, EncNo, EndIndex);
if(ret < 0)
{
    printf("Failed to set encoder array compare end index:%d\\n",ret);
}
else
{
    printf("Set encoder array compare array index successfully!\\n");
}
```

ECDAQ_GetArrCmpEndIdx

Description:

Get the end index to be used for the compare process of encoder.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_GetArrCmpEndIdx(DWORD dwDeviceNo, WORD  
wEncNo, WORD* wEndIndex);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wEndIndex	WORD	OUT	Index number Range: 0~199

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD EncNo= 0;

WORD EndIndex = 0;

ret = ECDAQ_GetArrCmpEndIdx (DeviceNo, EncNo, &EndIndex);

if(ret < 0)
{
    printf("Failed to get encoder array compare end index:%d\n",ret);
}
else
{
    printf("Get encoder array compare array index successfully!\n");
}
```

ECDAQ_EnableEncArrayCompare

Description:

Start Array compare-trigger function of encoder.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_EnableEncArrayCompare(DWORD dwDeviceNo, WORD  
wEncNo);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD EncNo= 0;


WORD Index = 0;
INT CmpPos = 150;
WORD DoTrigger = 1;

ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);
if(ret < 0)
    printf("Failed to set encoder array compare data:%d\n",ret);


Index = 1;
CmpPos = 250;
DoTrigger = 0;

ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);
if(ret < 0)
    printf("Failed to set encoder array compare data:%d\n",ret);


Index = 2;
CmpPos = 350;
DoTrigger = 1;

ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);
if(ret < 0)
    printf("Failed to set encoder array compare data:%d\n",ret);


Index = 3;
CmpPos = 500;
DoTrigger = 1;

ret = ECDAQ_SetArrCmpPosTrigger (DeviceNo, EncNo, Index, CmpPos, DoTrigger);
if(ret < 0)
    printf("Failed to set encoder array compare data:%d\n",ret);


WORD EndIndex = 3;

ret = ECDAQ_SetArrCmpEndIdx (DeviceNo, EncNo, EndIndex);
if(ret < 0)
```

```
printf("Failed to set encoder array compare end index:%d\n",ret);
```

```
ret = ECDAQ_EnableEncArrayCompare (DeviceNo, EncNo);
```

```
if(ret < 0)
```

```
printf("Failed to enable encoder array compare:%d\n",ret);
```

```
else
```

```
printf("Enable encoder array compare successfully!\n");
```

ECDAQ_DisableEncCompare

Description:

Disable position compare-trigger functions of encoder.

Note: This function only supports ECAT-2092T.

Syntax:

```
WORD ECDAQ_DisableEncCompare(DWORD dwDeviceNo, WORD  
wEncNo);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo= 0;
ret = ECDAQ_DisableEncCompare (DeviceNo, EncNo);
if(ret < 0)
{
    printf("Failed to disable encoder compare position:%d\n",ret);
}
else
{
    printf("Disable encoder compare position successfully!\n");
}
```

ECDAQ_SetVelTimeInterval

Description:

Set the time interval parameters of encoder for velocity calculation.

First, set the global time interval multiplier "Basic Multiplier" (0x8100: 01) of each channel, whose unit is 0.5 milliseconds.

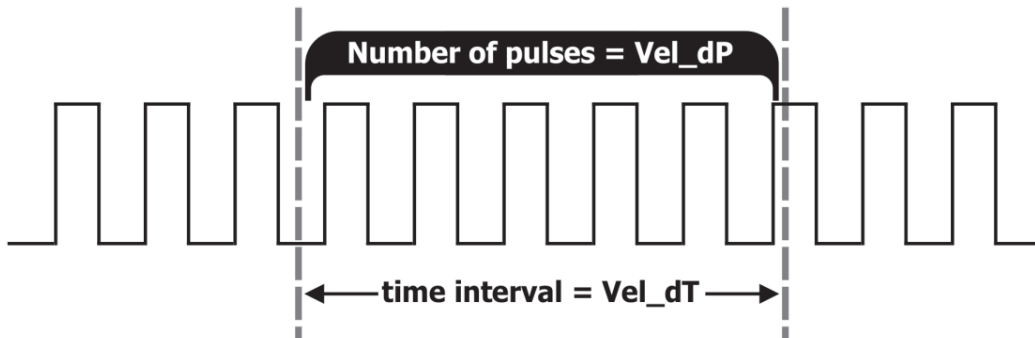
Second, set the time interval multiplier "CHn Multiplier" (0x8100 : 02+n, n = 0, 1, 2) of individual channels, then the time interval Vel_dT of each channel is configured as:

$$\text{Vel_dT} = T_{\text{basic}} \times T_{\text{chn}} \times 0.5 \times 10^{-3} \text{ sec}$$

Whereby T_{basic} represents "Basic Multiplier", T_{chn} represents "CHn Multiplier".

After that, the velocity V can be calculated from the position difference Vel_dP, and the unit is pulses/sec:

$$V = \text{Vel_dP} / \text{Vel_dT}$$



Note: This function only supports ECAT-2093, EC4-ENC2IT.

Syntax:

```
WORD ECDAQ_ECDAQ_SetVelTimeInterval (DWORD dwDeviceNo,  
BYTE bBaseMultiplier, BYTE bChnMultiplier[])
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
bBaseMultiplier	BYTE	IN	Base time multiplier
bChnMultiplier	BYTE[]	IN	Channel time multiplier

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

BYTE bBaseMultiplier; // unit: 0.5ms
BYTE bChnMultiplier[3];

// 1st device: ECAT-2093
DeviceNo = 0;
bBaseMultiplier = 1;
bChnMultiplier[0] = 1;
bChnMultiplier[1] = 2;
bChnMultiplier[2] = 4;
ret = ECDAQ_SetVelTimeInterval (DeviceNo, bBaseMultiplier, bChnMultiplier);
if(ret < 0)
{
    printf("Failed to set encoder velocity time interval:%d\n",ret);
}
else
{
    printf("Set encoder velocity time interval successfully!\n");
}

// 2nd device: EC4-ENC2IT
DeviceNo = 1;
bBaseMultiplier = 2;
bChnMultiplier[0] = 1;
bChnMultiplier[1] = 1;
ret = ECDAQ_SetVelTimeInterval (DeviceNo, bBaseMultiplier, bChnMultiplier);
if(ret < 0)
{
    printf("Failed to set encoder velocity time interval:%d\n",ret);
}
else
{
    printf("Set encoder velocity time interval successfully!\n");
}
```

ECDAQ_GetVelTimeInterval

Description:

Get the time interval parameters of encoder for velocity calculation.

Note: This function only supports ECAT-2093, EC4-ENC2IT.

Syntax:

```
WORD ECDAQ_ECDAQ_GetVelTimeInterval (DWORD dwDeviceNo,  
BYTE bBaseMultiplier, BYTE bChnMultiplier[])
```

Parameters:

Name	Type	IN or OUT	Description
dwDeviceNo	DWORD	IN	Device number
bBaseMultiplier	BYTE	OUT	Base time multiplier
bChnMultiplier	BYTE[]	OUT	Channel time multiplier

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

BYTE bBaseMultiplier; // unit: 0.5ms
BYTE bChnMultiplier[3];

// 1st device: ECAT-2093
DeviceNo = 0;
ret = ECDAQ_GetVelTimeInterval (DeviceNo, bBaseMultiplier, bChnMultiplier);
if(ret < 0)
{
    printf("Failed to get encoder velocity time interval:%d\n",ret);
}
else
{
    printf("Get encoder velocity time interval successfully!\n");
}

// 2nd device: EC4-ENC2IT
DeviceNo = 1;
ret = ECDAQ_GetVelTimeInterval (DeviceNo, bBaseMultiplier, bChnMultiplier);
if(ret < 0)
{
    printf("Failed to get encoder velocity time interval:%d\n",ret);
}
else
{
    printf("Get encoder velocity time interval successfully!\n");
}
```

ECDAQ_GetVeldP

Description:

Get the position difference of encoder for velocity calculation.

Note: This function only supports ECAT-2093, EC4-ENC2IT.

Syntax:

```
WORD ECDAQ_GetVeldP(DWORD dwDeviceNo, WORD wEncNo,  
SHORT* sVeldP)
```

Parameters:

Name	Type	IN or OUT	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
sVeldP	SHORT	OUT	Position difference per time interval

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

ECDAQ_GetAbsEncState

Description:

Get the absolute encoder state.

Note: This function only supports ECAT-2074A.

Syntax:

```
WORD ECDAQ_GetAbsEncState(DWORD dwDeviceNo, WORD wEncNo,  
WORD* wState);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wState	WORD	OUT	Encoder Counting Mode The definition is shown below. 0: OFF 1: ON

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Bit Number	Description
Bit 0	BiSS Warning
Bit 1	BiSS Error
Bit 2	Ready
Bit 3	CRC Error
Bit 4~7	Reserved

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD EncState = 0;
ret = ECDAQ_GetAbsEncState (DeviceNo, EncNo, &EncState);
if(ret < 0)
{
    printf("Failed to get absolute encoder state:%d\n",ret);
}
else
{
    printf("Get absolute encoder state successfully!\n");
}
```

ECDAQ_GetAbsEncCount

Description:

Get the absolute encoder counter value.

Note: This function only supports ECAT-2074A.

Syntax:

```
WORD ECDAQ_GetAbsEncCount(DWORD dwDeviceNo, WORD wEncNo,  
DWORD* dwMultiTurn, DWORD* dwSingleTurn);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
dwMultiTurn	DWORD	OUT	Encoder multiturn position data
dwSingleTurn	DWORD	OUT	Encoder singleturn position data

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
DWORD CntMT = 0;
DWORD CntST = 0;
ret = ECDAQ_GetAbsEncCount (DeviceNo, EncNo, &CntMT,&CntST);
if(ret < 0)
{
    printf("Failed to get absolute encoder count:%d\n",ret);
}
else
{
    printf("Get absolute encoder count successfully!\n");
}
```

ECDAQ_GetAbsEncCheckCnt

Description:

Get the absolute encoder data check count.

Note: This function only supports ECAT-2074A.

Syntax:

```
WORD ECDAQ_GetAbsEncCheckCnt(DWORD dwDeviceNo, WORD  
wEncNo, DWORD* dwCrcErrCnt, DWORD* dwNotReadyCnt);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
dwCrcErrCnt	DWORD	OUT	Count of CRC check error
dwNotReadyCnt	DWORD	OUT	Count of the data is not ready to read

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
DWORD CrcErrCnt = 0;
DWORD NotReadyCnt = 0;
ret = ECDAQ_GetAbsEncCheckCnt (DeviceNo, EncNo, &CrcErrCnt, &NotReadyCnt);
if(ret < 0)
{
    printf("Failed to get absolute encoder data check count:%d\n",ret);
}
else
{
    printf("Get absolute encoder data check count successfully!\n");
}
```

ECDAQ_ResetAbsEncCheckCnt

Description:

Reset the absolute encoder data check count.

Note: This function only supports ECAT-2074A.

Syntax:

```
WORD ECDAQ_ResetAbsEncCheckCnt(DWORD dwDeviceNo, WORD  
wEncNo);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
ret = ECDAQ_ResetAbsEncCheckCnt (DeviceNo, EncNo);
if(ret < 0)
{
    printf("Failed to reset encoder data check count:%d\n",ret);
}
else
{
    printf("Reset encoder data check count successfully!\n");
}
```

ECDAQ_SetAbsEncBits

Description:

Set the number of bits of the absolute encoder position data.

Note: This function only supports ECAT-2074A.

Syntax:

```
WORD ECDAQ_SetAbsEncBits(DWORD dwDeviceNo, WORD wEncNo,  
WORD wMultiTurn, WORD wSingleTurn);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wMultiTurn	WORD	IN	The number of bits of the multiturn position data of the encoder, the maximum setting is 32
wSingleTurn	WORD	IN	The number of bits of the singleturn position data of the encoder, the maximum setting is 32

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD BitsMT = 12;
WORD BitsST = 13;
ret = ECDAQ_SetAbsEncBits (DeviceNo, EncNo, BitsMT, BitsST);
if(ret < 0)
{
    printf("Failed to set the number of bits of absolute encoder position data:%d\n",ret);
}
else
{
    printf("Set the number of bits of absolute encoder position data successfully!\n");
}
```

ECDAQ_GetAbsEncBits

Description:

Get the number of bits of the absolute encoder position data.

Note: This function only supports ECAT-2074A.

Syntax:

```
WORD ECDAQ_GetAbsEncBits(DWORD dwDeviceNo, WORD wEncNo,  
WORD* wMultiTurn, WORD* wSingleTurn);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wMultiTurn	WORD	OUT	The number of bits of the multiturn position data of the encoder, the maximum setting is 32
wSingleTurn	WORD	OUT	The number of bits of the singleturn position data of the encoder, the maximum setting is 32

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
BYTE BitsMT = 0;
BYTE BitsST = 0;
ret = ECDAQ_GetAbsEncBits (DeviceNo, EncNo, &BitsMT, &BitsST);
if(ret < 0)
{
    printf("Failed to get the number of bits of absolute encoder position data:%d\n",ret);
}
else
{
    printf("Get the number of bits of absolute encoder position data successfully!\n");
}
```

ECDAQ_SetAbsEncMode

Description:

Set the absolute encoder transmission protocol.

Note: This function only supports ECAT-2074A.

Syntax:

```
WORD ECDAQ_SetAbsEncMode(DWORD dwDeviceNo, WORD wEncNo,  
WORD wMode);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wMode	WORD	IN	Transmission protocol used by the encoder: 0: BiSS 1: SSI

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD Mode = 0; //BiSS
ret = ECDAQ_SetAbsEncMode (DeviceNo, EncNo, Mode);
if(ret < 0)
{
    printf("Failed to set absolute encoder transmission protocol:%d\n",ret);
}
else
{
    printf("Set absolute encoder transmission protocol successfully!\n");
}
```

ECDAQ_GetAbsEncMode

Description:

Get the absolute encoder transmission protocol.

Note: This function only supports ECAT-2074A.

Syntax:

```
WORD ECDAQ_GetAbsEncMode(DWORD dwDeviceNo, WORD wEncNo,  
WORD* wMode);
```

Parameters:

Name	Type	Direction	Description
dwDeviceNo	DWORD	IN	Device number
wEncNo	WORD	IN	Encoder interface number
wMode	WORD	OUT	Transmission protocol used by the encoder: 0: BiSS 1: SSI

Return:

0: Success.

Others: Refer to Appendix "Error Codes".

Example:

[C/C++]

```
int ret = 0;
DWORD DeviceNo = 0;
WORD EncNo = 0;
WORD Mode = 0;
ret = ECDAQ_GetAbsEncMode (DeviceNo, EncNo, &Mode);
if(ret < 0)
{
    printf("Failed to get absolute encoder transmission protocol:%d\n",ret);
}
else
{
    printf("Get absolute encoder transmission protocol successfully!\n");
}
```

Appendix

Error Codes

Error ID	Error Code	Description
ECDAQ_GetDllVerErr	1	Failed to get ECATDAQ SDK Version
ECDAQ_ExceedDeviceNumber	2	Device number exceeds connected slave count
ECDAQ_ExceedAdapterNumber	3	Network adapter number exceeds installed adapter count
ECDAQ_FindDeviceErr	4	Failed to find device, no EtherCAT slave detected
ECDAQ_NoSocketConnect	5	NIC initialization failure, no device connected
ECDAQ_FindAdapterErr	6	Failed to find network adapter, no adapter installed
ECDAQ_CreateThreadErr	7	Failed to create thread
ECDAQ_WriteDataErr	8	
ECDAQ_StateErr	9	Not all slaves are in state OPERATIONAL
ECDAQ_UnSupportedMaster	10	ECATDAQ SDK does not support current PC platform
ECDAQ_ERR_ADAPTER_NOT_OPEN	51	Network adapter is not open. Please execute ECDAQ_Init first.
ECDAQ_ERR_ADAPTER_ALREADY_OPENED	52	Network adapter is already opened
ECDAQ_ERR_INVALID_ADAPTER	53	Invalid adapter number
ECDAQ_ERR_INVALID_SLAVE	54	Invalid slave number

ECDAQ_ERR_INVALID_CHANNEL	55	Invalid channel number
ECDAQ_ERR_WRONG_PARAMETER	56	Invalid parameter
ECDAQ_ERR_WRONG_STATE_MACHINE	57	Master is not in state OPERATIONAL
ECDAQ_ERR_WRONG_SLAVE_TYPE	58	Slave type error, the function is restricted to ECAT-2091S/94S or drivers which support CiA402 profile
ECDAQ_ERR_OPEN_FILE_FAILED	59	Failed to open *.bin file
ECDAQ_ERR_INCORRECT_FILE_SIZE	60	Incorrect file size
ECDAQ_ERR_WORKING_COUNTER	61	EtherCAT working counter error, please check slave connection status
ECDAQ_ERR_SDO_NO_RESPONSE	62	No SDO response received
ECDAQ_ERR_SDO_ACCESS_FAILED	63	Failed to access SDO. Please refer to Appendix "SDO Abort messages"
ECMST_ERR_209X_AXIS_BUSY	-101	Axis of ECAT-2091S/94S is in operation and cannot accept control commands. Please wait for the motion to complete, or execute ECDAQ_MC_AxisStop to stop the motion
ECMST_ERR_209X_AXIS_NOT_READY	-102	Axis of ECAT-2091S/94S is not enabled and cannot process motion command
ECMST_ERR_209X_AXIS_ERROR	-103	Axis error occurred in ECAT-2091S/94S. Please check the cause of error and execute ECDAQ_MC_ResetAxisError to clear error
ECMST_ERR_209X_INVALID_AXIS_STATE	-104	Invalid axis state for ECAT-2091S/94S
ECMST_ERR_209X_GET_INVALID_SDO_VALUE	-105	Invalid SDO value received

		from ECAT-2091S/94S
ECMST_ERR_209X_SET_INVALID_SDO_VALUE	-106	Invalid SDO value is set to ECAT-2091S/94S
ECMST_ERR_209X_SET_INVALID_AXIS_NO	-107	Invalid axis number for ECAT-2091S/94S
ECMST_ERR_209X_SET_PREOP_TIMEOUT	-108	ECAT-2091S/94S change EtherCAT state to preOP timeout while setting startup parameters
ECMST_ERR_209X_SET_OP_TIMEOUT	-109	ECAT-2091S/94S change EtherCAT state to OP timeout after setting startup parameters finished
ECMST_ERR_209X_SAVE_INTER_EEP_FAILED	-110	ECAT-2091S/94S set startup parameter failed
ECMST_ERR_209X_SAVE_INTER_EEP_CNT_ERR	-111	ECAT-2091S/94S save counter mismatch after setting startup parameters finished. Please power cycling ECAT-2091S/94S and check startup parameters, or set startup parameters again.
ECMST_ERR_209X_SET_SDO_FAILED	-112	Set SDO data to ECAT-2091S/94S failed while setting startup parameters, the setting process is uncompleted
ECMST_ERR_209X_SET_INVALID_PARAMETER	-113	Invalid motion parameter for ECAT-2091S/94S
ECMST_ERR_209X_HOME_METHOD_NOT_SUPPORT	-114	Unsupported home method for ECAT-2091S/94S
ECMST_ERR_402_INVALID_AXIS_STATE	-201	Invalid axis state for driver which support CiA402 profile
ECMST_209X_BUSY_SETTING_INITIAL_CFG	-255	ECAT-2091S/94S is busy setting startup parameters

		and cannot process motion command
--	--	--------------------------------------

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
------------	---

Initial setup for CiA402 driver

When the ECATDAQ is initialized, it will refer to the *DriveInfo.csv* file located in the same directory as the program to determine which operation modes can be enabled for the connected CiA402 drivers.

In this CSV file, the parameters of each CiA402 driver are listed in the following order: Vendor ID, Product ID, number of motion axes, default homing mode, whether torque value is read, whether PV mode is supported, whether PT mode is supported, and description (for user identification, optional), as shown in the table below.

VendorID	ProductID	AxisNum	HomeMode	ReadTorque	PV	PT	Description
0x00000539	0x02200301	1	35	1	1	1	YASKAWA / SGD7S
0x000005BC	0x00000001	1	35	1	1	1	SHIHLIN / SDP-E
0x0000066F	0x52510021	1	35	1	1	1	PANASONIC / MBDHT
0x0000066F	0x60380004	1	35	1	1	1	PANASONIC / MADLN05BE
0x000001DD	0x10305070	1	35	0	0	0	DELTA / A2
0x000001B9	0x00000006	1	35	1	1	1	SANYO / RS2
0x00494350	0x20944453	4	35	0	0	0	ICPDAS / ECAT-2094DS
0x00494350	0x00209450	4	35	0	0	0	ICPDAS / ECAT-2094P

Before using the CIA402 drive, it is necessary to confirm that applicable parameters have been set in this CSV file. Otherwise, after connecting this device, the master will not enter into the OP state.