

泓格 ECATDAQ 函式庫

使用手冊

僅支援泓格 EtherCAT 從屬模組



► 免責聲明

凡使用泓格產品除產品品質造成的損害，泓格科技股份有限公司不承擔任何法律責任。泓格科技股份有限公司有義務提供正確及詳細的資料，但保留修改權利，且不承擔使用者非法利用資料對第三方所造成侵害構成的法律責任。

► 版權

版權所有 © 2025 泓格科技股份有限公司,保留所有權利。

► 商標

手冊中所涉及所有公司商標，商標名稱及產品名稱分別屬於該商標或名稱的擁有者所有。

關於

本手冊說明如何透過泓格 ECATDAQ 函式庫在 Windows 下對泓格 EtherCAT 從站操作。本手冊提供了使用泓格 EtherCAT 從站的相關訊息。包含 I/O、運動控制操作的流程以及每個 API 函數的功能、參數、數據結構的說明。

使用者可以使用泓格 ECATDAQ 函式庫驅動在 Windows 系統下使用 VB、VC、BCB、Delphi、VB.NET、C#.NET、VC.NET、Console 等工具來實作開發。本手冊也提供範例程式，利用開發實例向使用者說明如何使用泓格 ECATDAQ 函式庫，提供給使用者參考進行應用開發。

如有本手冊未涵蓋之內容請來信諮詢泓格技術工程師。

Email: service@icpdas.com

Table of contents

1. 關於 ECATDAQ 函式庫	6
特色.....	6
支援的泓格產品.....	7
系統需求.....	7
參考資訊.....	7
2. 主站操作流程.....	8
3. 主站操作函式集說明.....	10
ECDAQ_GetDllVer.....	10
ECDAQ_GetAdapterInfo	12
ECDAQ_Init.....	14
ECDAQ_Close	16
ECDAQ_StartDeviceOpTask.....	18
ECDAQ_StopDeviceOpTask	20
ECDAQ_GetMasterState	22
4. 從站操作函式集說明.....	24
ECDAQ_GetDeviceName.....	24
ECDAQ_GetDeviceInfo	26
ECDAQ_SetDeviceID.....	28
ECDAQ_GetDeviceNoByID	29
ECDAQ_SetSdoData	30
ECDAQ_GetSdoData.....	32
ECDAQ_SetSdoObject	33
ECDAQ_GetSdoDataObject	35
ECDAQ_WriteDO	37
ECDAQ_ReadDI.....	39
ECDAQ_WriteAOVoltageH.....	41
ECDAQ_WriteAOCurrentH	43
ECDAQ_ReadAIH	45
ECDAQ_ReadAIScanH.....	47
5. 運動控制函式集說明.....	49
單軸運動控制設定.....	49
ECDAQ_MC_SetAxisServoON.....	49

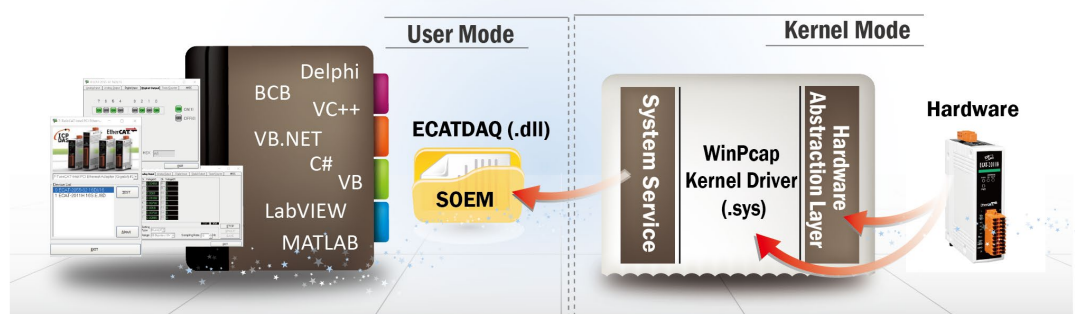
ECDAQ_MC_SetAxisMicroStep	51
ECDAQ_MC_GetAxisMicroStep.....	53
ECDAQ_MC_SetAxisCurrent.....	55
ECDAQ_MC_GetAxisCurrent.....	57
單軸運動狀態.....	59
ECDAQ_MC_GetAxisActualPos	59
ECDAQ_MC_GetAxisActualEnc	61
ECDAQ_MC_GetAxisActualVel	63
ECDAQ_MC_GetAxisActualTor	65
ECDAQ_MC_GetAxisState	66
ECDAQ_MC_GetAxisDI	69
ECDAQ_MC_GetAxisLastError.....	72
ECDAQ_MC_GetAxisDriverError	73
單軸自動原點復歸.....	74
ECDAQ_MC_AxisHomeStart	74
單軸運動控制.....	77
ECDAQ_MC_ResetAxisError	77
ECDAQ_MC_ResetAxisCounter	79
ECDAQ_MC_AxisMoveAbs	81
ECDAQ_MC_AxisMoveRel	84
ECDAQ_MC_AxisChangePos	86
ECDAQ_MC_AxisChangeVel	89
ECDAQ_MC_AxisMoveVel	92
ECDAQ_MC_AxisMove_CiA402_PV	94
ECDAQ_MC_AxisMove_CiA402_PT	96
ECDAQ_MC_AxisStop	98
ECDAQ_MC_AxisQuickStop	101
設置開機初始值.....	104
ECDAQ_MC_SetLoadDefault.....	104
ECDAQ_MC_GetLoadDefaultState	106
編碼器操作.....	108
ECDAQ_GetEncCountingMode	108
ECDAQ_SetEncCountingMode	110
ECDAQ_GetEncCount	112
ECDAQ_SetEncCount	114
ECDAQ_GetEncLatchCount	116
ECDAQ_GetEncLatchTriggerCount	118
ECDAQ_GetEncExLatchCount	120

ECDAQ_ResetEncCount.....	122
ECDAQ_SetEnableEncLatch.....	124
ECDAQ_GetEnableEncLatch.....	126
ECDAQ_SetEnableEncExLatch.....	128
ECDAQ_GetEnableEncExLatch.....	130
ECDAQ_ResetEncLatchCount.....	132
ECDAQ_ResetEncExLatchCount.....	134
ECDAQ_GetCmpTrigProperty.....	136
ECDAQ_SetCmpTrigProperty.....	138
ECDAQ_GetEncNextComparePos.....	140
ECDAQ_EnableEncEquidCompare.....	142
ECDAQ_SetArrCmpPosTrigger.....	144
ECDAQ_GetArrCmpPosTrigger.....	147
ECDAQ_SetArrCmpEndIdx.....	149
ECDAQ_GetArrCmpEndIdx.....	153
ECDAQ_EnableEncArrayCompare.....	155
ECDAQ_DisableEncCompare.....	158
ECDAQ_SetVelTimeInterval.....	160
ECDAQ_GetVelTimeInterval.....	162
ECDAQ_GetVeldP.....	164
ECDAQ_GetAbsEncCount.....	165
ECDAQ_GetAbsEncCheckCnt.....	167
ECDAQ_ResetAbsEncCheckCnt.....	169
ECDAQ_SetAbsEncBits.....	171
ECDAQ_GetAbsEncBits.....	173
ECDAQ_SetAbsEncMode.....	175
ECDAQ_GetAbsEncMode.....	177
附錄.....	179
函式錯誤回傳代碼.....	179
SDO 操作錯誤終止代碼.....	182
CiA402 驅動器初始設定說明.....	184

1. 關於 ECATDAQ 函式庫

泓格 ECATDAQ 函式庫建構於開源函式庫 SOEM(Simple Open EtherCAT Master)之上。ECATDAQ 函式庫提供完整的泓格從站控制 API，所以在泓格的 ECATDAQ 函式庫裡，不需使用特定的 EtherCAT 暫存器命令，即可輕鬆操作泓格的 EtherCAT 從站，而且泓格 ECATDAQ 函式庫整合出更適合泓格從站的專用函式，泓格 EtherCAT 的使用者可以在各種編程語言與環境下開發。

泓格 ECATDAQ 函式庫支援 Windows 7 之後所有的 32 位元及 64 位元作業系統，使用者不再需要擔心作業系統的相容性。



特色

- 透過標準乙太網路介面卡即可控制泓格從站
- 不需 ESI 檔案
- 提供 DLL 函式庫及 VC/VB6/C#/Python/LabVIEW 範例程式
- 控制週期 20ms(典型)
- 支援 8 軸及 20 個從站(僅支援泓格從站)
- 支援 Free Run 模式
- 支援第三方 CiA402 驅動器
- 支援單軸直線補間及 PP 模式

支援的泓格產品

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
---	--	--	--

系統需求

泓格 **XP-9000** 系列或 **AXP-9000** 系列控制器一台(需自行下載 ECATDAQ 函式庫安裝程式)或個人電腦上使用泓格 ECATDAQ 函式庫(個人電腦則需滿足下列需求)：

- 需擁有一台 ECAT-209x、ECAT-207x 或 EC4 系列從站模組
- 標準乙太網路介面卡
- 使用 266MHz 或更快的 32 位元(x86)或 64 位元(x64)處理器
- 至少 512 MB 的記憶體空間
- 相容於 VGA 的圖型顯示卡
- 至少 100 MB 磁碟空間
- Microsoft Windows 7 以上的 32 位元或 64 位元作業系統

參考資訊

詳細關於 ECATDAQ 函式庫使用手冊、安裝程式及範例程式，快速下載連結如下：

https://www.icpdas.com/tw/product/guide+Software+Development_Tools+ECATDAQ

2. 主站操作流程

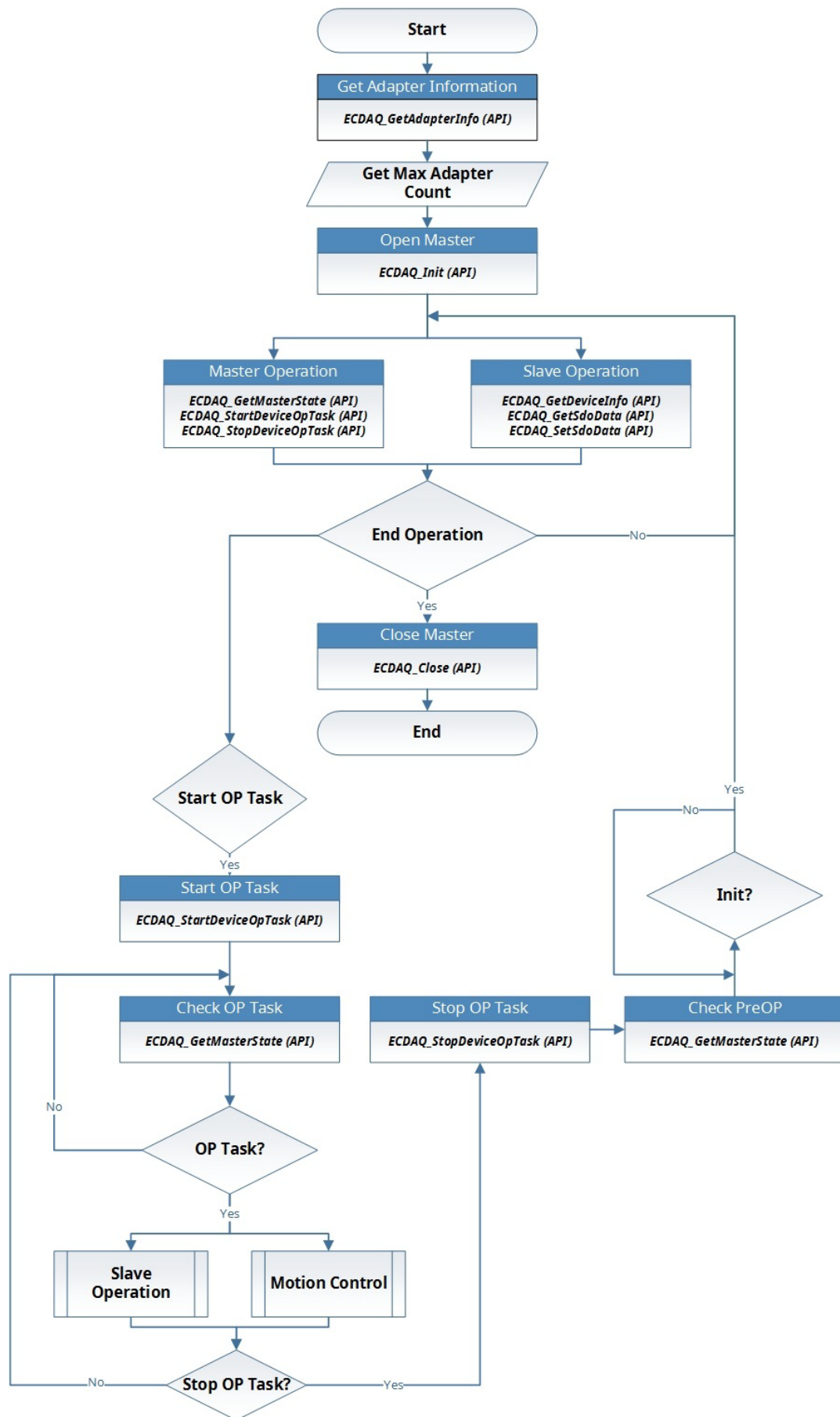
如下圖所示,使用者可調用 *GetAdapterInfo* 函式來取得網卡資訊，並根據網卡編號調用 *Init* 函式建立和網卡的通訊。於建立通訊後主站並未開始 EtherCAT 操作任務，此時可以進行基本的裝置操作。

首先可調用 *GetMasterState* 函式來取得目前整個 EtherCAT 網絡的各種狀態，包含目前連接從站數、主站的 AL 狀態 (INIT, PreOP, SafeOP, OP)、網路連接狀態(連線/斷線)等，或是對從站調用 *GetDeviceInfo* 函式取得從站相關資訊。若需讀寫 SDO 物件數值，可調用 *GetSdoData*, *SetSdoData* 函式來完成。

當使用者需要開始 EtherCAT 操作任務時，可以調用 *StartDeviceOpTask* 函式，在參數中指定已建立主站通訊的網卡來開始 EtherCAT 操作任務。然後，調用 *GetMasterState* 函式來取得 EtherCAT 網路狀態，等到主站進入到 OP 狀態 (代表所有從站也都進入 OP 狀態)後，使用者就可以進行從站 PDO 物件讀寫以及運動控制等等的操作。

當主站進入 OP 狀態後，程式即自動完成運動軸初始化，使用者只需呼叫 *MC_SetAxisServoON* 函式將運動軸啟用，即可進行單軸的原點復歸與運動控制。

調用 *StopDeviceOpTask* 函式可以主站離開 OP 模式，並且停止操作任務；而進一步調用 *Close* 函式則可以關閉與網卡的通訊。



3. 主站操作函式集說明

ECDAQ_GetDllVer

說明:

取得 ECATDAQ SDK 版本。

格式:

WORD ECDAQ_GetDllVer(DWORD *dwDllVer)

參數:

名稱	類型	方向	說明
dwDllVer	DWORD*	OUT	版本號碼

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得網卡資訊。

格式:

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

參數:

名稱	類型	方向	說明
wTotalNetworkNum	WORD*	OUT	實際網卡數量
wNetworkInfoNo	WORD	IN	指定網卡編號
szNetworkInfoName	char*	OUT	指定網卡的名稱

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

初始化 EtherCAT 主站。

格式:

WORD ECDAQ_Init(WORD wNetworkInfoNo, DWORD *dwTotalSlaveDevice);

參數:

名稱	類型	方向	說明
wNetworkInfoNo	WORD	IN	指定網卡編號
dwTotalSlaveDevice	DWORD *	OUT	指定網卡所連接的 EtherCAT 模組數量

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

關閉 EtherCAT 主站。

格式:

WORD ECDAQ_Close ();

參數:

名稱	類型	方向	說明

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

開始指定網卡的 EtherCAT 操作任務。

格式:

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

參數:

名稱	類型	方向	說明
wNetworkInfoNo	WORD	IN	指定網卡編號
wEnumCycleTimeMs	WORD	IN	EtherCAT 通訊週期時間，單位為毫秒 (ms) 參考設定為 20 ms。
dwWcErrCnt	DWORD	IN	設定網路狀態出錯時，可容許連續出錯次數，當發生次數到達設定值時才發出錯誤告知使用者端。

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

停止指定網卡的 EtherCAT 操作任務。

格式:

WORD ECDAQ_StopDeviceOpTask(WORD wNetworkInfoNo)

參數:

名稱	類型	方向	說明
wNetworkInfoNo	WORD	IN	指定網卡編號

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得主站 EtherCAT 網路狀態。

格式:

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

參數:

名稱	類型	方向	說明
dwLinkUp	DWORD*	OUT	顯示乙太網路孔連接狀態 0: 主站無連接任何從站 1: 主站已連接任何從站
dwSlavesResp	DWORD*	OUT	顯示目前已連接從站數量
dwAlStates	DWORD*	OUT	顯示目前整個網路 Slave EtherCAT 狀態 (如下表所示)
dwWkc	DWORD*	OUT	顯示 EtherCAT 工作計數器數值 (Working Counter)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

標頭檔定義名稱	數值	說明
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

使用範例:

[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. 從站操作函式集說明

ECDAQ_GetDeviceName

說明:

取得從站模組名稱。

格式:

WORD ECDAQ_GetDeviceName(DWORD dwDeviceNo, char *szDeviceName)

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
szDeviceName	char*	OUT	指定從站名稱

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得從站模組資訊。

格式:

WORD ECDAQ_GetDeviceInfo(DWORD dwDeviceNo, PECDAQ_DEVICE_INFO sDevInfo)

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
sDevInfo	PECDAQ_DEVICE_INFO	OUT	指定從站資訊

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

typedef struct _ECDAQ_DEVICE_INFO_
{
    WORD wSize;           //Structure Size
    WORD wExtDeviceID;     // Explicit Device Identification or Alias ID
    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, *PECDAQ_DEVICE_INFO;

PECDAQ_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

說明:

設置從站模組的識別碼或 Explicit Device ID。

格式:

WORD ECDAQ_GetDeviceInfo(DWORD dwDeviceNo, DWORD dwDeviceID)

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
dwDeviceID	DWORD	IN	設定從站識別碼

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

ECDAQ_GetDeviceNoByID

說明:

透過裝置識別碼或 Explicit Device ID 返查從站編號值。

格式:

WORD ECDAQ_GetDeviceNoByID (DWORD* dwDeviceNo, DWORD dwDeviceID)

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD*	OUT	取得從站編號
dwDeviceID	DWORD	IN	從站識別碼

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

ECDAQ_SetSdoData

說明:

設定從站模組 SDO 資料。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wIndex	WORD	IN	索引值
uSubIndex	UCHAR	IN	子索引值
dwVal	DWORD	IN	設定的 SDO 資料

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得從站模組 SDO 資料。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wIndex	WORD	IN	索引值
uSubIndex	UCHAR	IN	子索引值
dwVal	DWORD*	OUT	取得的 SDO 資料

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

ECDAQ_SetSdoObject

說明:

設定從站模組 SDO 資料，並檢查操作是否出錯。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wIndex	WORD	IN	索引值
uSubIndex	UCHAR	IN	子索引值
wDataSize	WORD	IN	資料大小
dwVal	DWORD	IN	設定的 SDO 資料
AbortCode	DWORD	OUT	SDO 操作錯誤終止代碼 (請參考附錄 "SDO 操作錯誤終止代碼"說明)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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_GetSdoDataObject

說明:

取得從站模組 SDO 資料，並檢查操作是否出錯。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wIndex	WORD	IN	索引值
uSubIndex	UCHAR	IN	子索引值
wDataSize	WORD	IN	資料大小
dwVal	DWORD*	OUT	取得的 SDO 資料
AbortCode	DWORD	OUT	SDO 操作錯誤終止代碼 (請參考附錄 "SDO 操作錯誤終止代碼"說明)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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");
}
else
{
    printf("Get SDO data successfully!\n");
}
```

ECDAQ_WriteDO

說明:

寫入從站模組指定位元的 DO 資料。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wPortNo	WORD	IN	位元資料編號
cValue	char	IN	寫入位元數值

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

讀取從站模組指定位元的 DI 資料。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wPortNo	WORD	IN	位元資料編號
cValue	char	IN	讀取位元數值

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

寫入二進制數值到從站模組的類比輸出通道，以改變輸出的電壓。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAOChannel	WORD	IN	類比輸出通道編號
wAORange	WORD	IN	類比輸出範圍 (如下表所示)
dwValue	DWORD	IN	寫入二進制數值

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

標頭檔定義名稱	數值	說明
ECDAQ_AO_UNI_5V	0	電壓範圍 0 ~ 5V
ECDAQ_AO_BI_5V	1	電壓範圍 $\pm 5V$
ECDAQ_AO_UNI_10V	2	電壓範圍 0 ~ 10V
ECDAQ_AO_BI_10V	3	電壓範圍 $\pm 10V$

使用範例:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AOChannel= 0;

WORD AORange = 0; // 0 ~ 5V

DWORD Value = 255;

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

ECDAQ_WriteAOCurrentH

說明:

寫入二進制數值到從站模組的類比輸出通道，以改變輸出的電流。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAOChannel	WORD	IN	類比輸出通道編號
wAORange	WORD	IN	類比輸出範圍 (如下表所示)
dwValue	DWORD	IN	寫入二進制數值

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

標頭檔定義名稱	數值	說明
ECDAQ_AO_UNI_10MA	4	電流範圍 0 ~ 10mA
ECDAQ_AO_BI_10MA	5	電流範圍 ± 10 mA
ECDAQ_AO_UNI_20MA	6	電流範圍 0 ~ 20mA
ECDAQ_AO_BI_20MA	7	電流範圍 ± 20 mA
ECDAQ_AO_UNI_4TO20MA	8	電流範圍 4 ~ 20mA
ECDAQ_AO_BI_24MA	9	電流範圍 ± 24 mA

使用範例:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AOChannel= 0;

WORD AORange = 6; // 0 ~ 20mA

DWORD Value = 255;

ret = ECDAQ_WriteAOCurH (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

說明:

讀取從站模組的類比輸入通道數值，並回傳未經校正的結果。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wChannel	WORD	IN	類比輸入通道編號
dwValue	DWORD*	OUT	讀取自指定採樣通道的原始數據

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

讀取從站模組的所有類比輸入通道數值，並回傳未經校正的結果。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wTotalChannels	WORD	IN	類比輸入通道總數
dwValue	DWORD*	OUT	讀取自每個採樣通道的原始數據

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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. 運動控制函式集說明

單軸運動控制設定

ECDAQ_MC_SetAxisServoON

說明:

設定運動軸的啟動訊號狀態。

設定 Servo On 之後，可以使用 ECDAQ_MC_GetAxisState 取得軸狀態，當軸狀態為 MC_AS_STANDSTILL 代表 Servo On 完成。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
bServoOn	UCHAR	IN	訊號狀態 0: OFF 1: ON

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

設置運動軸的微步進設定值。

注：此函式僅支援 ECAT-2091S、ECAT-2094S、ECAT-2094DS、ECAT-2094P

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
wMicroStep	WORD	IN	每全步的微步數, 可設定值由 0~8 分別對應: 1, 2, 4, 8, 16, 32, 64, 128, 256.

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得運動軸的微步進設定值。

注：此函式僅支援 ECAT-2091S、ECAT-2094S、ECAT-2094DS、ECAT-2094P

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
wMicroStep	WORD*	OUT	每全步的微步數, 可設定值由 0~8 分別對應: 1, 2, 4, 8, 16, 32, 64, 128, 256.

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

設定運動軸於各狀態下的電流大小。

注：此函式僅支援 ECAT-2091S、ECAT-2094S

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
wMax	WORD	IN	驅動馬達時的最大電流 (單位: mA) 當高速運轉會失步時，請加大此值
wHold	WORD	IN	馬達靜止電流 (單位: mA) 當激磁後且未運轉時，馬達使用此電流
wPowerOn	WORD	IN	驅動器上電後的馬達電流 (單位: mA) 當馬達未激磁時，使用此電流
wSafe	WORD	IN	馬達安全電流 (在狀態從 OP 改變 至其他狀態時應用) (單位: mA)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得運動軸於各狀態下的的電流大小。

注：此函式僅支援 ECAT-2091S、ECAT-2094S

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
wMax	WORD*	OUT	驅動馬達時的最大電流
wHold	WORD*	OUT	馬達靜止電流
wPowerOn	WORD*	OUT	驅動器上電後的馬達電流
wSafe	WORD*	OUT	馬達安全電流 (在狀態從 OP 改變至其他狀態時應用)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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");
}
```

單軸運動狀態

ECDAQ_MC_GetAxisActualPos

說明:

取得指定軸號當前位置。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
lPos	long*	OUT	指定軸號當前位置 (單位: pulse)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得指定軸號編碼器數值。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
lEnc	long*	OUT	指定軸號編碼器數值 (單位: pulse)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得指定軸號當前速度。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
lVel	long*	OUT	指定軸號速度 (單位: pulse/s)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得指定軸號當前轉矩。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
sTor	SHORT*	OUT	轉矩 單位請參考驅動器使用手冊的物件"Torque Actual Value: 6077h"，一般為額定轉矩的 0.1% 額定轉矩請參考"Motor rated torque: 6076h"

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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 torque:%d\n", ret);
else
    printf("Axis torque:%d\n", AxisTor)
```

ECDAQ_MC_GetAxisState

說明:

取得指定軸號當前狀態。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
bState	UCHAR*	OUT	指定軸號當前狀態編號(如下表所示)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

標頭檔定義名稱	數值	說明
MC_AS_DISABLED	0	軸尚未啟用
MC_AS_STANDSTILL	1	軸啟用且為停止狀態，準備接收新的運動命令
MC_AS_ERRORSTOP	2	軸目前出現錯誤且為停止狀態
MC_AS_STOPPING	3	軸目前在停止運動中
MC_AS_HOMING	4	軸目前在原點復歸中
MC_AS_DISCRETEMOTION	5	軸目前單軸運動中
MC_AS_CONTINUOUSMOTION	6	軸目前連續運動中

使用範例:

[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

說明:

取得指定軸號運動控制相關 I/O 訊號狀態。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
wDI	WORD *	OUT	軸 I/O 訊號狀態(如下表所示) 位元若為 0:表示對應的訊號狀態為 OFF 位元若為 1:表示對應的訊號狀態為 ON

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

位元	對應訊號
Bit 0	NOT (負方向位置極限)
Bit 1	POT (正方向位置極限)
Bit 2	ORG (原點)
Bit 3	ALM (錯誤)
Bit 4	WAN (警告)
Bit 5	SVN (伺服啟動)
Bit 6	Busy (命令處理中)
Bit 7~15	保留

使用範例:

[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

說明:

取得指定軸號最後出錯代碼。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
iError	INT*	OUT	指定軸號最後出錯代碼(請參考附錄"函式錯誤回傳代碼"說明)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

ECDAQ_MC_GetAxisDriverError

說明:

取得指定軸號驅動器出錯代碼。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
wError	WORD*	OUT	指定軸號驅動器出錯代碼(請根據驅動器使用手冊尋找相關錯誤代碼說明)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

單軸自動原點復歸

ECDAQ_MC_AxisHomeStart

說明:

開始執行自動原點復歸。

格式:

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

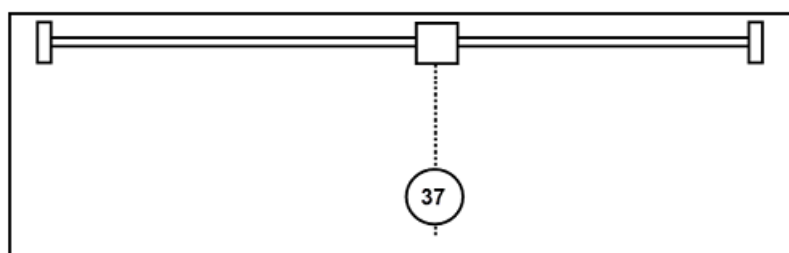
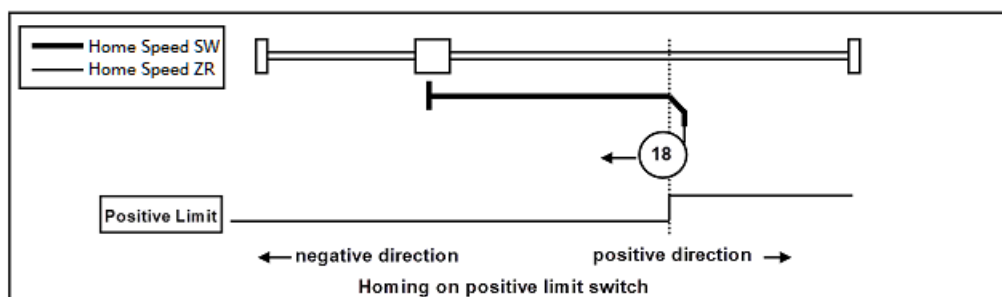
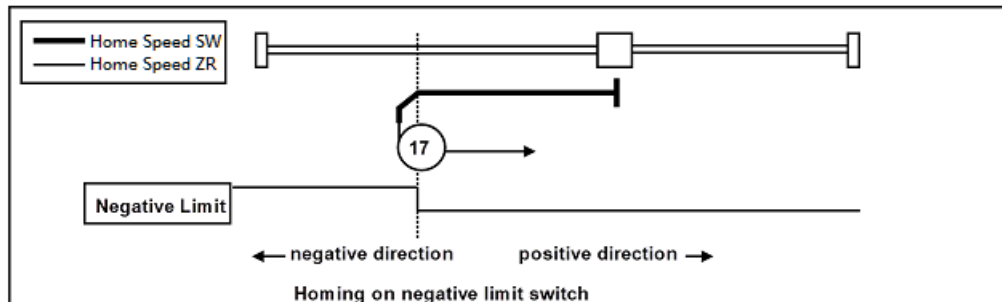
參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
wMethod	WORD	IN	原點復歸模式 ECAT-2091S/ECAT-2094S 支援模式包含 17. 18. 37
dwSearchSw	DWORD	IN	尋找原點訊號時使用的速度 (單位: pulse/s)
dwSearchZr	DWORD	IN	尋找馬達索引訊號時使用的速度 (單位: pulse/s)
dwAcc	DWORD	IN	執行自動原點復歸時使用的加速度 (單位: pulse/s ²)
dwDec	DWORD	IN	執行自動原點復歸時使用的減速度 (單位: pulse/s ²)
lPosOffset	long	IN	原點位置相對於零點位置的偏移量 (單位: pulse)
bUseEnc	UCHAR	IN	是否使用外部編碼器 0: 否, 1: 是
dEncMotorRatio	double	IN	編碼器對馬達的比例 例如: 馬達轉 1 圈 51200pulse, 編碼器轉一圈 4000pulse, 則比例為 4000/51200

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。



使用範例:

[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");
}
```

單軸運動控制

ECDAQ_MC_ResetAxisError

說明:

重置指定軸號的錯誤狀態，異常發生時軸將切換至錯誤狀態，請在異常排除後執行此函式重置軸的狀態。

使用此函式後之後，需要幾個 ms 才會完成重置軸狀態，可以使用 ECDAQ_MC_GetAxisState 取得軸狀態，軸狀態為 MC_AS_STANDSTILL(處於 Servo On 狀態)或 MC_AS_DISABLED(處於 Servo off 狀態)即重置軸狀態完成。

格式:

WORD ECDAQ_MC_ResetAxisError(DWORD dwDeviceNo, WORD wAxisNo)

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

重置指定軸號的位置或編碼器數值。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
bResetPos	UCHAR	IN	0: 不處理 1: 重置位置
bResetEnc	UCHAR	IN	0: 不處理 1: 重置編碼器

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

開始執行指定軸號單軸絕對位置運動控制。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
dwVel	DWORD	IN	移動速度(單位: pulse/s)
dwAcc	DWORD	IN	加速度(單位: pulse/s ²)
dwDec	DWORD	IN	減速度(單位: pulse/s ²)
lPos	long	IN	目標位置(單位: pulse)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

開始執行指定軸號單軸相對距離運動控制。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
dwVel	DWORD	IN	移動速度(單位: pulse/s)
dwAcc	DWORD	IN	加速度(單位: pulse/s ²)
dwDec	DWORD	IN	減速度(單位: pulse/s ²)
lDist	long	IN	相對於目前位置的距離(單位: pulse)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

當指定軸號處於移動中(僅限於絕對位置或相對距離模式)，該運動命令可用於改變終點位置。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
lPos	long	IN	終點位置 (單位: pulse)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

當指定軸號處於移動中(僅限於絕對位置或相對距離模式)，該運動命令可用於改變移動速度。

格式:

WORD ECDAQ_MC_AxisChangeVel(DWORD dwDeviceNo, WORD wAxisNo, long lVel)

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
lVel	long	IN	移動速度 (單位: pulse/s)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

開始執行指定軸號單軸等速連續運動控制，此運動持續移動，直到停止指令(執行函式或硬體訊號)觸發終止運動。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
iVel	INT	IN	移動速度(單位: pulse/s)
dwAcc	DWORD	IN	加速度(單位: pulse/s ²)
dwDec	DWORD	IN	減速度(單位: pulse/s ²)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

開始執行指定軸號單軸 CiA402 速度模式運動控制。

格式:

WORD ECDAQ_MC_AxisMove_CiA402_PV(DWORD dwDeviceNo, WORD wAxisNo, INT iVel, DWORD dwAcc, DWORD dwDec)

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
iVel	INT	IN	移動速度(單位: pulse/s)
dwAcc	DWORD	IN	加速度(單位: pulse/s ²)
dwDec	DWORD	IN	減速度(單位: pulse/s ²)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

UCHAR AxisState;

INT AxisVel = -1000;

WORD Acc = 100;

WORD Dec = 100;

ret = ECDAQ_MC_GetAxisState(DeviceNo, AxisNo, &AxisState);
if(AxisState == MC_AS_STANDSTILL) //StandStill
{
    ret = ECDAQ_MC_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

說明:

開始執行指定軸號單軸 CiA402 扭力模式運動控制。

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號
sTor	SHORT	IN	目標扭力值，最大扭力值的千分比，數值 1~1000
dwSlpoe	DWORD	IN	扭力上升率(單位: 0.1%/s)

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[C/C++]

```
int ret = 0;

DWORD DeviceNo = 0;

WORD AxisNo = 0;

UCHAR AxisState;

SHORT Torque = 30;

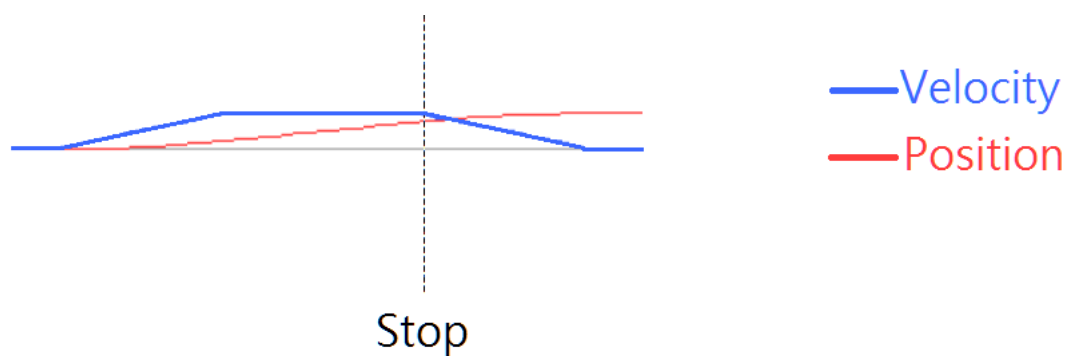
DWORD Slope = 10;


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

說明:

停止指定軸號單軸運動控制，執行此命令將減速後停止運動。



格式:

WORD ECDAQ_MC_AxisStop(DWORD dwDeviceNo, WORD wAxisNo)

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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);
```

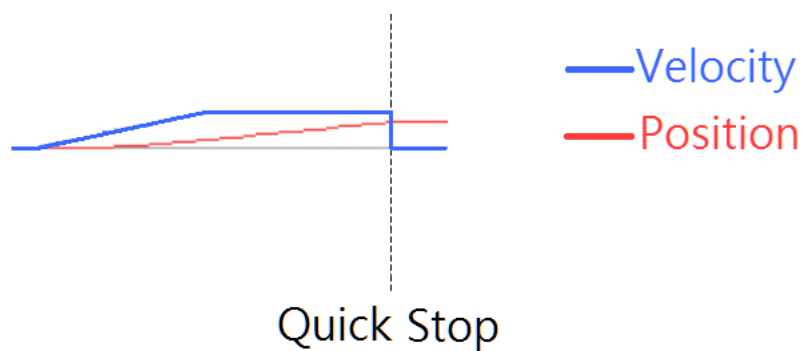
```
}while(AxisState == MC_AS_STOPPING) //Stopping

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

說明:

停止指定軸號單軸運動控制，執行此命令將立即停止運動。



格式:

WORD ECDAQ_MC_AxisQuickStop(DWORD dwDeviceNo, WORD wAxisNo)

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wAxisNo	WORD	IN	軸號

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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");
}
}
```



設置開機初始值

ECDAQ_MC_SetLoadDefault

說明:

設置開機時 SDO 列表 0x8000~0x8321 使用出廠或使用者設定值。當選用出廠設定時，很快就能完成設置；而當選用使用者設定時，需等待 10 多秒時間讓模組將 SDO 列表數值寫入內部 EEPROM。

執行 ECDAQ_MC_SetLoadDefault 後，使用者可以呼叫 ECDAQ_MC_GetLoadDefaultState 取得處理狀態，當狀態為 ECMST_ERR_NO_ERROR 即代表初始化選項設置完成

建議在完成初始化選項設置後，將模組重開新上電，以確保操作穩定。

注：此函式僅支援 ECAT-2091S、ECAT-2094S

格式:

WORD ECDAQ_MC_SetLoadDefault(DWORD dwDeviceNo, UCHAR bDefault)

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
bDefault	UCHAR	IN	初始化選項 0: 使用者設定 1: 出廠設定

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

在進行設定開機初始化選項時，取得初始化狀態。當狀態為 ECMST_ERR_NO_ERROR 即代表初始化選項設置完成

注：此函式僅支援 ECAT-2091S、ECAT-2094S

格式:

WORD ECDAQ_MC_GetLoadDefaultState(DWORD dwDeviceNo)

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。



使用範例:

[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");
}
```

編碼器操作

ECDAQ_GetEncCountingMode

說明:

取得編碼器從站模組的計數模式。

注：此函式僅支援 ECAT-2092T、ECAT-2093

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wMode	WORD	OUT	編碼器計數模式: 1: CW/CCW, 2: Pulse/Direction, 3: Quadrant

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

設置編碼器從站模組的計數模式。

注：此函式僅支援 ECAT-2092T、ECAT-2093

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wMode	WORD	IN	編碼器計數模式: 1: CW/CCW, 2: Pulse/Direction, 3: Quadrant

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得編碼器從站模組的計數器數值。

注：此函式僅支援 ECAT-2092T、ECAT-2093

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
iCnt	INT	OUT	編碼器計數器數值

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

設置編碼器從站模組的計數器數值。

注：此函式僅支援 ECAT-2092T

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
iCnt	INT	IN	編碼器計數器數值

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得編碼器從站模組的索引輸入鎖存器(Index Latch)數值

注: 此函式僅支援 ECAT-2092T

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
iCnt	INT	OUT	鎖存器數值

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得編碼器從站模組的計數器數值。

注：此函式僅支援 ECAT-2093

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wTrigCnt	WORD	OUT	編碼計數器的鎖存觸發次數，每當有鎖存觸發時此次數加 1

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得編碼器從站模組的外部輸入鎖存器(External Latch)數值。

注: 此函式僅支援 ECAT-2092T、ECAT-2093

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
iRising	INT	OUT	上緣觸發之鎖存器數值
iFalling	INT	OUT	下緣觸發之鎖存器數值

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

清除編碼器從站模組的計數器數值。

注：此函式僅支援 ECAT-2092T、ECAT-2093

格式:

WORD ECDAQ_ResetEncCount(DWORD dwDeviceNo, WORD wEncNo);

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

設置編碼器從站模組的啟用索引輸入鎖存器(Index Latch)參數。

注: (1)此函式僅支援 ECAT-2093

(2) ECAT-2092T 的索引輸入鎖存器始終保持啟用狀態且無法禁用

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wEnable	WORD	IN	啟用(1)/停用(0)鎖存器

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得編碼器從站模組的啟用索引輸入鎖存器(Index Latch)參數。

注: (1)此函式僅支援 ECAT-2093

(2) ECAT-2092T 的索引輸入鎖存器始終保持啟用狀態且無法禁用

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wEnable	WORD	OUT	啟用(1)/停用(0)鎖存器

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

設置編碼器從站模組的啟用外部輸入鎖存器(External Latch)參數。

注: 此函式僅支援 ECAT-2092T

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wEnable	WORD	IN	啟用(1)/停用(0)鎖存器

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得編碼器從站模組的啟用外部輸入鎖存器(External Latch)參數。

注: 此函式僅支援 ECAT-2092T

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wEnable	WORD	OUT	啟用(1)/停用(0)鎖存器

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

清除編碼器從站模組的索引輸入鎖存器(Index Latch)數值。

注: 此函式僅支援 ECAT-2092T、ECAT-2093

格式:

WORD ECDAQ_ResetEncLatchCount(DWORD dwDeviceNo, WORD wEncNo);

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

清除編碼器從站模組的外部輸入鎖存器(External Latch)數值。

注: 此函式僅支援 ECAT-2092T

格式:

WORD ECDAQ_ResetEncExLatchCount(DWORD dwDeviceNo, WORD wEncNo);

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得編碼器從站模組的比較觸發功能參數值。

注：此函式僅支援 ECAT-2092T

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wPulseWidth	WORD	OUT	比較觸發脈波寬度 範圍: 2~32767(微秒), 預設:50

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

設置編碼器從站模組的比較觸發功能參數值。

注：此函式僅支援 ECAT-2092T

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wPulseWidth	WORD	IN	比較觸發脈波寬度 範圍: 2~32767(微秒), 預設:50

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得編碼器從站模組的下一個比較位置。

注：此函式僅支援 ECAT-2092T

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wValid	WORD	OUT	下一個比較位置有效
iPos	INT	OUT	下一個比較位置

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

啟用編碼器從站模組的單點或連續比較觸發功能。

注: (1) 此函式僅支援 ECAT-2092T

(2) 當每次觸發的間隔數值設置為 0 時，啟用單點比較觸發功能

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wDir	WORD	IN	比較方向 0: 向上計數時比較 1: 向下計數時比較
iFirstPos	INT	IN	開始比較的數值
iDistance	INT	IN	每次觸發的間隔數值

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

設置編碼器從站模組的陣列比較位置與數位輸出觸發設定。啟用陣列比較功能後，會從 index 0 所儲存的比較位置開始比較，當編碼器達到比較位置時，會根據設定而決定是否要觸發數位輸出

注: (1) 此函式僅支援 ECAT-2092T

(2) 第一個要比較的值必須存放在 Index 0，第二個要比較的值必須存放在 Index 1，以此類推

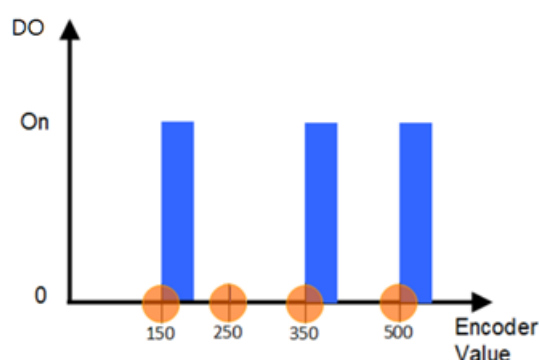
(3) 當 Index 0 的數值被比較過後，才會開始比較 Index 1 的數值，以此類推

(4) 即使該 Index 的數位輸出被禁用，其比較值還是需要被比較。例如以下範例，index 1 的數位輸出被禁用，但是 Index 1 的數值被比較過後(不觸發數位輸出)，才會開始比較 Index 2 的數值

例: 將下表設為陣列比較值及數位輸出設定

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

DO 實際輸出情形:



格式:

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

參數:

名稱	類型	方向	說明
----	----	----	----

dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wIndex	WORD	IN	索引值 範圍: 0~199
iCmpPos	INT	IN	比較的數值
wDoTrigger	WORD	IN	啟用(1)/禁用(0)數位輸出

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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);
```

ECDAQ_GetArrCmpPosTrigger

說明:

取得編碼器從站模組的陣列比較位置與數位輸出觸發設定。

注：此函式僅支援 ECAT-2092T

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wIndex	WORD	IN	索引值 範圍: 0~199
iCmpPos	INT	OUT	比較的數值
wDoTrigger	WORD	OUT	啟用(1)/禁用(0)數位輸出

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

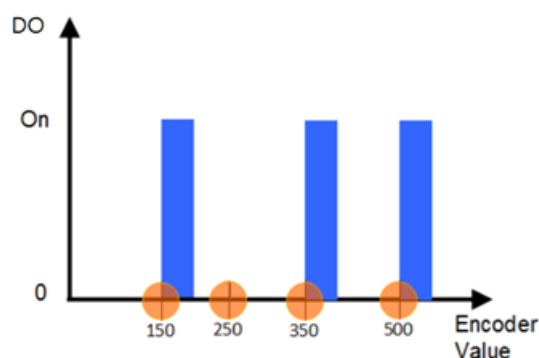
設定陣列比較範圍，啟用陣列比較觸發功能後，會從 index 0 所儲存的比較位置開始比較，比較至設定的 index 值後，下一個比較位置將會從 index 0 所儲存的比較位置重新開始比較。

注：此函式僅支援 ECAT-2092T

例：將下表設為陣列比較值、數位輸出設定及陣列比較範圍

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

DO 實際輸出情形:



格式:

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

參數:

名稱	類型	IN or OUT	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wEndIndex	WORD	IN	索引值範圍 範圍: 0~199

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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);
```

ECDAQ_GetArrCmpEndIdx

說明:

取得陣列比較範圍。

注：此函式僅支援 ECAT-2092T

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wEndIndex	WORD	OUT	索引值範圍 範圍: 0~199

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

啟用編碼器從站模組的陣列比較觸發功能。

格式:

WORD ECDAQ_EnableEncArrayCompare(DWORD dwDeviceNo, WORD wEncNo);

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號

回傳值:

0: 函式執行成功。

其他: 請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

禁用編碼器從站模組的比較觸發功能。

注：此函式僅支援 ECAT-2092T

格式:

WORD ECDAQ_DisableEncCompare(DWORD dwDeviceNo, WORD wEncNo);

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

設置編碼器從站模組用以計算速度的時間間隔參數。

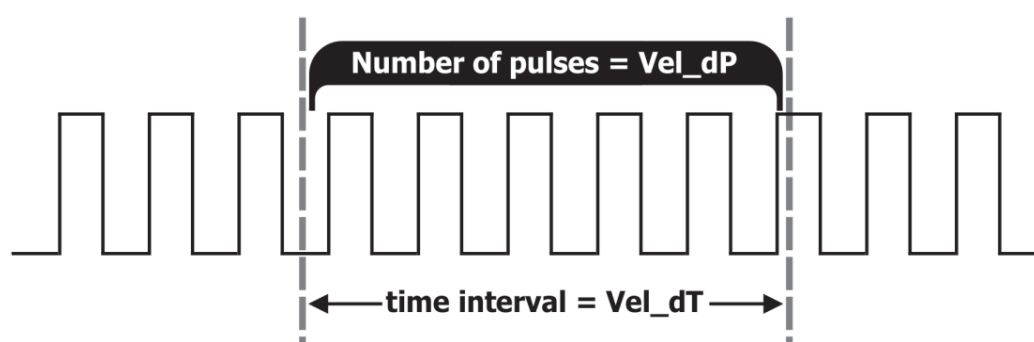
首先，設定各通道共用的時間間隔基礎乘數"Basic Multiplier"（0x8100：01），其單位時間為 0.5 毫秒(ms)；然後，再設置各通道的時間間隔乘數"CHn Multiplier"（0x8100：02+n, n = 0, 1, 2）。如此，各通道的時間間隔 Vel_dT 即設置如下：

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

其中， T_{basic} 表示"Basic Multiplier"， T_{chn} 表示"CHn Multiplier"。

之後，即可從位置差值 Vel_dP 算出速度 V，單位為脈波/秒 (pulses/sec):

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



注：此函式僅支援 ECAT-2093、EC4-ENC2IT

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
bBaseMultiplier	BYTE	IN	基礎時間乘數
bChnMultiplier	BYTE[]	IN	通道時間乘數

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得編碼器從站模組用以計算速度的時間間隔參數。

注：此函式僅支援 ECAT-2093、EC4-ENC2IT

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
bBaseMultiplier	BYTE	OUT	基礎時間乘數
bChnMultiplier	BYTE[]	OUT	通道時間乘數

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得編碼器從站模組用以計算速度的位置差值。

注：此函式僅支援 ECAT-2093、EC4-ENC2IT

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
sVeldP	SHORT	OUT	每間隔時間內的位置差值

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

ECDAQ_GetAbsEncCount

說明:

取得絕對型編碼器從站模組的計數器數值。

注：此函式僅支援 ECAT-2074A

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
dwMultiTurn	DWORD	OUT	編碼器的多圈位置資料
dwSingleTurn	DWORD	OUT	編碼器的單圈位置資料

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得絕對型編碼器從站模組的資料檢查計數值。

注：此函式僅支援 ECAT-2074A

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
dwCrcErrCnt	DWORD	OUT	CRC 檢查錯誤計數值
dwNotReadyCnt	DWORD	OUT	資料未準備讀取的計數值

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

重置絕對型編碼器從站模組的資料檢查計數值。

注：此函式僅支援 ECAT-2074A

格式:

WORD ECDAQ_ResetAbsEncCheckCnt(DWORD dwDeviceNo, WORD wEncNo);

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[C/C++]

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

ECDAQ_SetAbsEncBits

說明:

設置絕對型編碼器從站模組的位置資料位元數。

注：此函式僅支援 ECAT-2074A

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wMultiTurn	WORD	IN	編碼器的多圈位置資料位元數， 最大設置 32
wSingleTurn	WORD	IN	編碼器的單圈位置資料位元數， 最大設置 32

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得絕對型編碼器從站模組的位置資料位元數。

注：此函式僅支援 ECAT-2074A

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wMultiTurn	WORD	OUT	編碼器的多圈位置資料位元數， 最大設置 32
wSingleTurn	WORD	OUT	編碼器的單圈位置資料位元數， 最大設置 32

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

設置絕對型編碼器從站模組的傳輸協定。

注：此函式僅支援 ECAT-2074A

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wMode	WORD	IN	編碼器傳輸資料使用的傳輸協定： 0: BiSS 1: SSI

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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

說明:

取得絕對型編碼器從站模組的傳輸協定。

注：此函式僅支援 ECAT-2074A

格式:

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

參數:

名稱	類型	方向	說明
dwDeviceNo	DWORD	IN	從站編號
wEncNo	WORD	IN	編碼器編號
wMode	WORD	OUT	編碼器傳輸資料使用的傳輸協定： 0: BiSS 1: SSI

回傳值:

0: 函式執行成功。

其他：請參考附錄"函式錯誤回傳代碼"說明。

使用範例:

[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");
}
```

附錄

函式錯誤回傳代碼

標頭檔定義名稱	錯誤碼	說明
ECDAQ_GetDllVerErr	1	讀取 ECATDAQ SDK 版本失敗
ECDAQ_ExceedDeviceNumber	2	從站編號超出連接模組數量
ECDAQ_ExceedAdapterNumber	3	網卡編號超出安裝網卡數量
ECDAQ_FindDeviceErr	4	找尋模組失敗，未偵測到 ECAT 模組
ECDAQ_NoSocketConnect	5	網卡初始化失敗，未連接任何裝置
ECDAQ_FindAdapterErr	6	找尋網卡失敗，未裝設網卡
ECDAQ_CreateThreadErr	7	建立執行序失敗
ECDAQ_WriteDataErr	8	
ECDAQ_StateErr	9	從站模組 EtherCAT 狀態非 Operation
ECDAQ_UnSupportedMaster	10	ECATDAQ SDK 不支援當下主機平台
ECDAQ_ADAPTER_NOT_OPEN	51	未啟用網卡。請執行 ECDAQ_Init
ECDAQ_ADAPTER_ALREADY_OPENED	52	網卡已開啟
ECDAQ_INVALID_ADAPTER	53	無效的網卡編號
ECDAQ_INVALID_SLAVE	54	無效的模組編號
ECDAQ_INVALID_CHANNEL	55	無效的通道編號
ECDAQ_WRONG_PARAMETER	56	無效的參數
ECDAQ_WRONG_STATE_MACHINE	57	主站狀態錯誤，當下狀態未處於 OP
ECDAQ_WRONG_SLAVE_TYPE	58	模組類型錯誤，所使用 API 限制模組為 ECAT-2091S/94S 或支援 CiA402

		的驅動器
ECDAQ_OPEN_FILE_FAILED	59	開啟 bin 檔案失敗
ECDAQ_INCORRECT_FILE_SIZE	60	錯誤的檔案大小
ECDAQ_WORKING_COUNTER	61	EtherCAT 計數器數值異常，請檢查模組連線狀態
ECDAQ_SDO_NO_RESPONSE	62	存取 SDO 無回應，請稍後再試，或檢查連線狀態
ECDAQ_SDO_ACCESS_FAILED	63	存取 SDO 失敗，請參考附錄 SDO 操作錯誤終止代碼
ECMST_ERR_209X_AXIS_BUSY	-101	ECAT-2091S/94S 的運動軸運作中，無法接受此控制命令。請等待運動完成，或執行 ECDAQ_MC_AxisStop 停止運動
ECMST_ERR_209X_AXIS_NOT_READY	-102	ECAT-2091S/94S 運動軸未啟用，無法執行運動控制命令
ECMST_ERR_209X_AXIS_ERROR	-103	ECAT-2091S/94S 運動軸狀態錯誤，請檢查錯誤原因，然後執行 ECDAQ_MC_ResetAxisError 清除錯誤
ECMST_ERR_209X_INVALID_AXIS_STATE	-104	ECAT-2091S/94S 調用運動控制函式時，指定軸號目前狀態非核心可接受軸狀態。 請確認當前的軸狀態
ECMST_ERR_209X_GET_INVALID_SDO_VALUE	-105	在 ECAT-2091S/94S 取得無效的 SDO 數值
ECMST_ERR_209X_SET_INVALID_SDO_VALUE	-106	無效的 Channel 參數
ECMST_ERR_209X_SET_INVALID_AXIS_NO	-107	對 ECAT-2091S/94S 設置了無效的運動軸編號
ECMST_ERR_209X_SET_PREOP_TIMEOUT	-108	ECAT-2091S/94S 設置開機設定值時，切換狀態到 PreOP 逾時
ECMST_ERR_209X_SET_OP_TIMEOUT	-109	ECAT-2091S/94S 設置開機

		設定值完成後，切換狀態到 OP 逾時
ECMST_ERR_209X_SAVE_INTER_EEP_FAILED	-110	ECAT-2091S/94S 設置開機設定值失敗
ECMST_ERR_209X_SAVE_INTER_EEP_CNT_ERR	-111	ECAT-2091S/94S 設置開機設定值的計數器數值不符，請重新上電並確認是否設置成功，或再次設置開機設定值
ECMST_ERR_209X_SET_SDO_FAILED	-112	ECAT-2091S/94S 設置開機設定值時設置 SDO 失敗，未完成開機設定值設置
ECMST_ERR_209X_SET_INVALID_PARAMETER	-113	對 ECAT-2091S/94S 調用運動控制函式時，輸入了無效的參數
ECMST_ERR_209X_HOME_METHOD_NOT_SUPPORT	-114	對 ECAT-2091S/94S 調用原點賦歸命令時，輸入了不支援的模式
ECMST_ERR_402_INVALID_AXIS_STATE	-201	對支援 CiA402 的驅動器調用運動控制函式時，指定軸號目前狀態非核心可接受軸狀態。 請確認當前的軸狀態
ECMST_209X_BUSY_SETTING_INITIAL_CFG	-255	ECAT-2091S/94S 正在設置開機設定值，無法執行其他運動控制命令

SDO 操作錯誤終止代碼

錯誤碼	說明
0x05030000	分段傳輸時 toggle bit 無變化
0x05040000	SDO 傳輸逾時
0x05040001	命令碼無效或未知
0x05040005	記憶體不足
0x06010000	不支援存取的物件(object)
0x06010001	嘗試讀取一個唯寫的物件(object)
0x06010002	嘗試寫入一個唯讀的物件(object)
0x06020000	在物件字典(object directory)此物件(object)不存在
0x06040041	此物件(object)不能映射到 PDO 中
0x06040042	要映射的物件(object)數量和長度超過 PDO 數據長度
0x06040043	一般參數不相容
0x06040047	裝置一般內部不相容
0x06060000	由於硬體錯誤導致存取失敗
0x06070010	資料大小不匹配，服務參數長度不匹配
0x06070012	資料大小不匹配，服務參數長度過長
0x06070013	資料大小不匹配，服務參數長度過短
0x06090011	子索引不存在
0x06090030	寫入數據超出範圍
0x06090031	寫入數據太大
0x06090032	寫入數據太小
0x06090036	最大值小於最小值
0x08000000	一般性錯誤
0x08000020	數據無法傳輸或儲存到應用程序端
0x08000021	由於本地控制原因，數據無法傳輸或儲存到應用程序端
0x08000022	由於當前裝置狀態原因，數據無法傳輸或儲存到應用程序端
0x08000023	動態產生物件字典(object directory)失敗，或沒有找到物件字典(object directory)
0x05030000	分段傳輸時 toggle bit 無變化
0x05040000	SDO 傳輸逾時
0x05040001	命令碼無效或未知

0x05040005	記憶體不足
0x06010000	不支援存取的物件(object)
0x06010001	嘗試讀取一個唯寫的物件(object)
0x06010002	嘗試寫入一個唯讀的物件(object)
0x06020000	在物件字典(object directory)此物件(object)不存在
0x06040041	此物件(object)不能映射到 PDO 中
0x06040042	要映射的物件(object)數量和長度超過 PDO 數據長度
0x06040043	一般參數不相容
0x06040047	裝置一般內部不相容
0x06060000	由於硬體錯誤導致存取失敗
0x06070010	資料大小不匹配，服務參數長度不匹配
0x06070012	資料大小不匹配，服務參數長度過長
0x06070013	資料大小不匹配，服務參數長度過短
0x06090011	子索引不存在
0x06090030	寫入數據超出範圍
0x06090031	寫入數據太大
0x06090032	寫入數據太小
0x06090036	最大值小於最小值
0x08000000	一般性錯誤
0x08000020	數據無法傳輸或儲存到應用程序端
0x08000021	由於本地控制原因，數據無法傳輸或儲存到應用程序端
0x08000022	由於當前裝置狀態原因，數據無法傳輸或儲存到應用程序端
0x08000023	動態產生物件字典(object directory)失敗，或沒有找到物件字典(object directory)

CiA402 驅動器初始設定說明

ECATDAQ 在初始化時，會根據與程式同目錄下的 DriveInfo.csv 設定，以決定所連接的 CiA402 驅動器可啟用哪些操作模式。

在此 csv 檔中，各個 CiA402 驅動器所用參數依序為：廠商編號、產品編號、運動軸數、預設歸原點模式、是否讀取扭力值、是否支援 PV 模式、是否支援 PT 模式，與描述(方便使用者辨識，非必要)，如下表所示。

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

在使用 CiA402 驅動器之前，必須確認在此 csv 檔中已設置適用的參數，否則在連接此裝置後，將使主站無法進入 OP 狀態。