

# **NEXCOM**

EtherCAT Master

CiA402 servo control API

用戶手冊

Manual Rev.: V1.4

Revision Date: February 24th, 2016

## Revise note:

Ver	Description
V0.1	2013/12/16: First beta version release
V1.0	2014/8/19: Enhance API descriptions. Add torque control APIs
V1.1	2015/5/15: Typo fix
V1.2	2015/7/13: Add description of moving direction to Jog and JogA
V1.3	2015/8/14: Add Ch2.4.11 NEC_CoE402SetTargetVelocity() API Add Ch2.4.12 NEC_CoE402GetActualVelocity() API
V1.4	2016/2/24: Add Ch2.7.4 NEC_CoE402ClearHomeStartBit() API Add Ch2.7.5 NEC_CoE402CheckHomeStatus() API

## 目錄

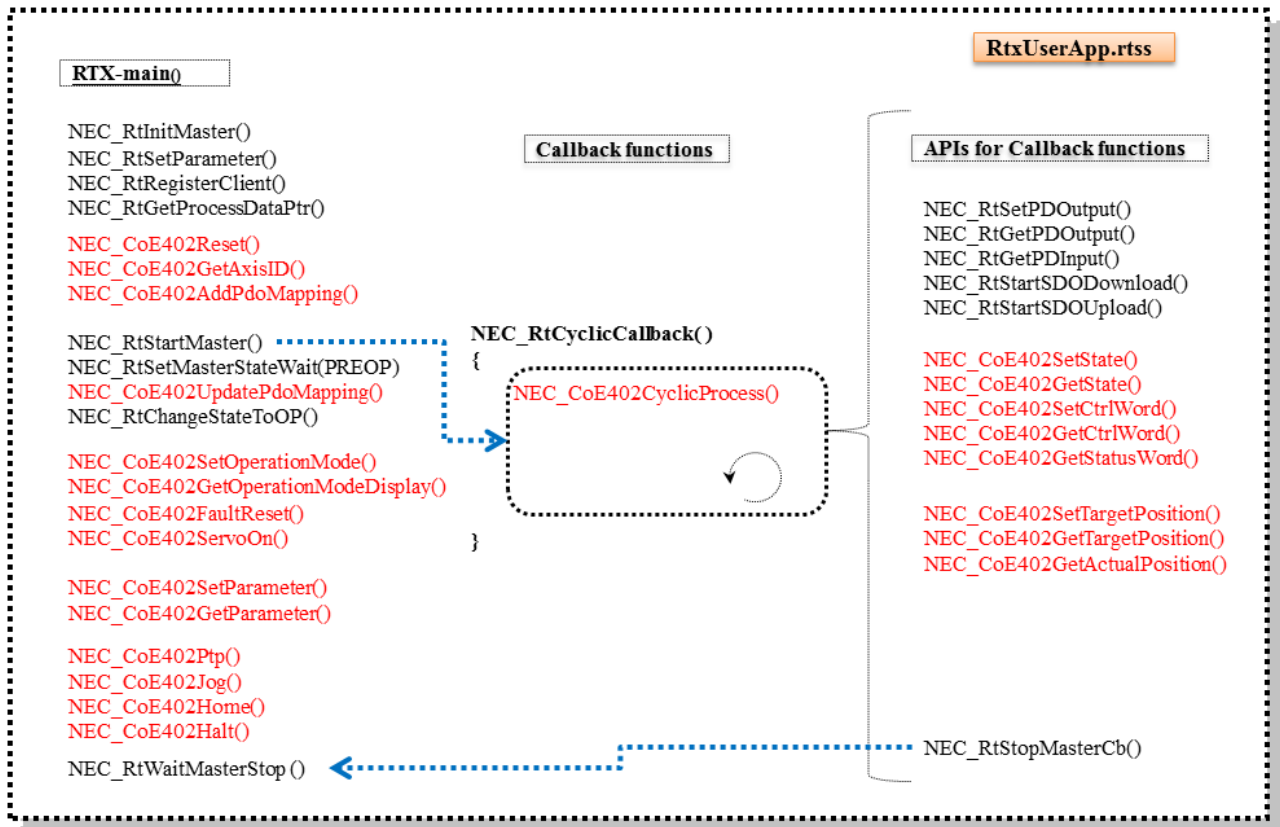
NEXCOM .....	1
Revise note: .....	2
目錄 .....	i
1. CoE CiA402 操作說明 .....	1
1.1. 流程圖 .....	1
1.2. 基本規格 .....	1
1.3. 注意事項 .....	1
2. CoE CiA402 函式庫 .....	2
2.1. API 總覽 .....	2
2.2. 初始化相關函式 .....	5
2.2.1. NEC_CoE402Reset .....	5
2.2.2. NEC_CoE402Close .....	6
2.2.3. NEC_CoE402GetAxisId .....	7
2.2.4. NEC_CoE402GetDefaultMapInfo .....	8
2.2.5. NEC_CoE402GetAxisIdEx .....	10
2.2.6. NEC_CoE402ResetPdoMapping .....	12
2.2.7. NEC_CoE402AddPdoMapping .....	13
2.2.8. NEC_CoE402UpdatePdoMapping .....	14
2.3. CiA402 狀態控制函式 .....	15
2.3.1. NEC_CoE402CyclicProcess .....	15
2.3.2. NEC_CoE402SetCtrlWord .....	17
2.3.3. NEC_CoE402GetCtrlWord .....	17
2.3.4. NEC_CoE402GetStatusWord .....	19
2.3.5. NEC_CoE402ChangeState .....	20
2.3.6. NEC_CoE402SetState .....	21

2.3.7.	NEC_CoE402GetState.....	22
2.3.8.	NEC_CoE402FaultReset.....	23
2.3.9.	NEC_CoE402ServoOn .....	24
2.4.	CiA402 驅動器基本操作函式 .....	25
2.4.1.	NEC_CoE402SetOperationMode.....	25
2.4.2.	NEC_CoE402GetOperationModeDisplay .....	26
2.4.3.	NEC_CoE402SetParameter.....	27
2.4.4.	NEC_CoE402GetParameter .....	28
2.4.5.	NEC_CoE402GetActualPosition.....	29
2.4.6.	NEC_CoE402SetTargetPosition .....	30
2.4.7.	NEC_CoE402GetTargetPosition.....	30
2.4.8.	NEC_CoE402SetQuickStopDec .....	31
2.4.9.	NEC_CoE402SetSoftPosLimit.....	32
2.4.10.	NEC_CoE402SetMaxVelLimit.....	33
2.4.11.	NEC_CoE402SetTargetVelocity .....	34
2.4.12.	NEC_CoE402GetActualVelocity.....	35
2.5.	點對點運動 (Profile Position mode) .....	36
2.5.1.	NEC_CoE402Ptp .....	36
2.5.2.	NEC_CoE402PtpV .....	36
2.5.3.	NEC_CoE402PtpA .....	36
2.5.4.	NEC_CoE402WaitTargetReached .....	38
2.6.	速度運動(Profile Velocity mode) .....	40
2.6.1.	NEC_CoE402Jog.....	40
2.6.2.	NEC_CoE402JogA .....	40
2.6.3.	NEC_CoE402Halt .....	42
2.7.	歸零運動 (Homing mode) .....	43
2.7.1.	NEC_CoE402Home .....	43

- 2.7.2. NEC\_CoE402HomeEx..... 43
- 2.7.3. NEC\_CoE402WaitHomeFinished ..... 45
- 2.7.4. NEC\_CoE402ClearHomeStartBit..... 46
- 2.7.5. NEC\_CoE402CheckHomeStatus ..... 47
- 2.8. 扭力控制 (Torque Control)..... 48
  - 2.8.1. NEC\_CoE402SetTargetTorque..... 48
  - 2.8.2. NEC\_CoE402GetTargetTorque ..... 48
  - 2.8.3. NEC\_CoE402GetActualTorque ..... 49
  - 2.8.4. NEC\_CoE402SetTorqueProfile..... 50

## 1. CoE CiA402 操作說明

### 1.1. 流程圖



### 1.2. 基本規格

1. 支援最高 32 軸
2. 支援標準 CoE402 EtherCAT 伺服驅動器

### 1.3. 注意事項

使用本函式庫前，下列幾點須注意：

1. ControlWord(0x6040)須設定為 RxPDO, Offset 不限制，但建議設定為 0
2. StatusWord(0x6041)須設定為 TxPDO, Offset 不限制，但建議設定為 0
3. TargetPosition(0x607A) 建議設定為 RxPDO, Offset 建議設定為 2
4. Position actual value(0x6064) 建議設定為 TxPDO, Offset 建議設定為 2

## 2. CoE CiA402 函式庫

### 2.1. API 總覽

下表列出 CiA402 函式庫 API 的列表，API 定義於 NexCoEMotion.h 標頭檔之中。

T: 欄位代表該函式可被呼叫的位置

C: 只能在 Callback 函式中呼叫

X: 不能再 Callback 函式中呼叫

B: 沒有限制

(T: Type → C: Callback only, X: Not for callback, B: Both)

函數名稱	說明	T
初始化相關函式		
NEC_CoE402Reset	重置 CiA402 函式庫	X
NEC_CoE402Close	關閉 CiA402 函式庫	X
NEC_CoE402GetAxisId	註冊並取得 CiA 伺服軸的識別碼 (單模組單軸)	X
NEC_CoE402GetDefaultMapInfo	初始化 MapInfo 資料結構	X
NEC_CoE402GetAxisIdEx	註冊並取得 CiA 伺服軸的識別碼 (單模組多軸)	X
NEC_CoE402ResetPdoMapping	重置 PDO mapping	X
NEC_CoE402AddPdoMapping	新增 PDO mapping 設定	X
NEC_CoE402UpdatePdoMapping	更新 PDO mapping 設定	X
CiA402 狀態控制函式		
NEC_CoE402CyclicProcess	周期狀態控制函式	C
NEC_CoE402SetCtrlWord	設定 CiA402 ControlWord (0x6040)	B
NEC_CoE402GetCtrlWord	讀取 CiA402 ControlWord (0x6040)	B
NEC_CoE402GetStatusWord	讀取 CiA402 StatusWord (0x6041)	B
NEC_CoE402ChangeState	改變控制軸的狀態	X
NEC_CoE402SetState	設定狀態	B
NEC_CoE402GetState	讀取驅動器狀態	B
NEC_CoE402FaultReset	清除/重置伺服驅動器錯誤	X
NEC_CoE402ServoOn	設定驅動器激磁/解激磁	X
CiA402 驅動器基本操作函式		
NEC_CoE402SetOperationMode	設定驅動器運動模式 (0x6060)	X
NEC_CoE402GetOperationModeDisplay	讀取實際驅動器運動模式 (0x6061)	X
NEC_CoE402SetParameter	設定 CiA402 驅動器物件參數	X
NEC_CoE402GetParameter	讀取 CiA402 驅動器物件參數	X
NEC_CoE402GetActualPosition	讀取驅動器馬達實際位置 (0x6064)	*B

NEC_CoE402SetTargetPosition	設定目標位置 (0x607A, PP, CSP)	*B
NEC_CoE402GetTargetPosition	讀取目標位置 (0x607A, PP, CSP)	*B
NEC_CoE402SetQuickStopDec	設定 Quick Stop 減速率 (0x6085)	X
NEC_CoE402SetSoftPosLimit	設定軟體位置極限 (0x607D)	X
NEC_CoE402SetMaxVelLimit	設定最大速度限制 (0x607F)	X
點對點運動 (Profile Position mode)		
NEC_CoE402Ptp	啟動單軸點對點運動	X
NEC_CoE402PtpV	啟動單軸點對點運動, 帶速度參數	X
NEC_CoE402PtpA	啟動單軸點對點運動, 帶速度、加速度參數	X
NEC_CoE402WaitTargetReached	等待目標位置到達	X
速度運動 (Profile Velocity mode)		
NEC_CoE402Jog	啟動單軸指定速度運轉	X
NEC_CoE402JogA	啟動單軸指定速度運轉, 帶加速度參數	X
NEC_CoE402Halt	停止/暫停運動	X
歸零運動 (Homing mode)		
NEC_CoE402Home	啟動單軸原點歸零運動 (Homing)	X
NEC_CoE402HomeEx	啟動單軸原點歸零運動 (Homing) 帶參數設定	X
NEC_CoE402WaitHomeFinished	等待歸零運動結束	X
NEC_CoE402ClearHomeStartBit	清除原點歸零運動 (Homing) 啟動位元	B
NEC_CoE402CheckHomeStatus	讀取原點歸零運動 (Homing) 狀態	B
扭力控制 (Torque Control)		
NEC_CoE402SetTargetTorque	設定目標扭力 (0x6071, PT, CST)	*B
NEC_CoE402GetTargetTorque	讀取目標扭力 (0x6071, PT, CST)	*B
NEC_CoE402GetActualTorque	讀取實際扭力值 (0x6077)	*B
NEC_CoE402SetTorqueProfile	設定目標扭力與變化率 (0x6071, 0x6087, PT)	X

(\*B 表示需將該資訊設定為 PDO mapping, 才能在 Callback 中呼叫, 否則回傳錯誤碼)

API 所使用的 C/C++ 資料型態定義於 nex\_type.h 中, 說明如下表:

型別	C/C++ 原型	說明	大小 byte	範圍
BOOL_T	int	布林型別	4	0:False, 1:True
U8_T	unsigned char	無號整數	1	0 ~ 255
U16_T	unsigned short	無號整數	2	0 ~ 65535
U32_T	unsigned int	無號整數	4	0 ~ 4294967295
U64_T	unsigned __int64	無號整數	8	0 ~ 18446744073709551615



I8_T	char	有號整數	1	-128 ~ 127
I16_T	short	有號整數	2	-32768 ~ 32767
I32_T	int	有號整數	4	-2147483648 ~ 2147483647
I64_T	__int64	有號整數	8	-9223372036854775808 ~ 9223372036854775807
F32_T	float	浮點數	4	IEEE-754, 有效小數後 7 位
F64_T	double	雙精浮點數	8	IEEE-754, 有效小數後 15 位
RTN_ERR	int	錯誤代碼	4	-2147483648 ~ 2147483647

## 2.2. 初始化相關函式

### 2.2.1. NEC\_CoE402Reset

重置 CiA402 函式庫

**C/C++語法:**

```
RTN_ERR NEC_CoE402Reset();
```

**參數:**

<無參數>

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

使用其他 CoE402 函式庫前，呼叫此函數進行函式庫內部初始化工作。

**注意!** 禁止於 Callback 函式中調用此函數

**參閱:**

```
NEC_CoE402Close ();
```

### 2.2.2. NEC\_CoE402Close

關閉 CiA402 函式庫

**C/C++語法:**

```
RTN_ERR NEC_CoE402Close ();
```

**參數:**

<無參數>

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

應用程式結束前，調用此函數釋放 CiA402 函式庫內部資源。

**注意!** 禁止於 Callback 函式中調用此函數

**參閱:**

```
NEC_CoE402Reset();
```

### 2.2.3. NEC\_CoE402GetAxisId

註冊並取得 CiA 伺服軸的識別碼

**C/C++語法:**

```
RTN_ERR NEC_CoE402GetAxisId( U16_T MasterId, U16_T SlaveAddr, CANAxis_T  
*pAxis );
```

**參數:**

U16\_T MasterId: 指定目標 EC-Master 代號，單一 EC-Master 請設為 0

U16\_T SlaveAddr: 指定目標 EC-Slave 代號，依網路配線順序，從 0 開始依序增號

CANAxis\_T \*pAxis: 回傳所對應的 CiA 伺服軸的控制識別碼(Identification)。用於稍後其他函式。

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”(0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

初始化 CiA402 函式庫後，使用本函式註冊 CiA402 模組並取得 CiA 伺服軸的控制代號。取得之軸代號用於爾後的其他 CiA402 函式庫的呼叫。

此 API 適用於只有單一 CiA402 伺服軸的 Slave 裝置，若單一 Slave 裝置具有多軸 (\*)需使用 NEC\_CoE402GetAxisIdEx()進行註冊的動作。

**注意!** 禁止於 Callback 函式中調用此函數

(\*)單一 Slave 裝置具有多軸 例如 NEXCOM AXE-5904 (4 axes CoE step/servo interface module)

**參閱:**

NEC\_CoE402CyclicProcess();NEC\_CoE402GetAxisIdEx()

#### 2.2.4. NEC\_CoE402GetDefaultMapInfo

初始化 MapInfo 資料結構

C/C++語法:

```
RTN_ERR NEC_CoE402GetDefaultMapInfo( U16_T TypeOfSlave, U16_T SlaveAddr,
EcCiA402AxisMapInfo_T *pMapInfo );
```

參數:

U16\_T TypeOfSlave: Slave 型態

EC\_CIA402\_SERVO\_DRIVE (0):標準 CiA402 單軸模組 (單模組單軸)

EC\_MULTIPLE\_CIA402\_SLAVE (1):標準 CiA402 多軸模組 (單模組多軸)

U16\_T SlaveAddr: 指定目標 EC-Slave 代號，依網路配線順序，從 0 開始依序增號  
EcCiA402AxisMapInfo\_T \*pMapInfo: CiA 軸 PDO 映射資料，此資料結構的內容將被初始化如下述:

```
typedef struct
{
    U16_T TypeOfSlave;
    U16_T SlaveAddr;
    U16_T SlaveSlotNum;
    U16_T Reserved;          // Reserved, set zero.
    U16_T CoeObjectOffset; // ex. 0x0800
    U16_T pdoMapOffset;    // ex. 0x0010
    U16_T MinRxPdoIndex;   // ex. 0x1600
    U16_T MaxRxPdoIndex;   // ex. 0x1603
    U16_T MinTxPdoIndex;   // ex. 0x1A00
    U16_T MaxTxPdoIndex;   // ex. 0x1A03
} EcCiA402AxisMapInfo_T;
```

U16\_T TypeOfSlave: Slave 型態,同上述並初始化為上述值。

U16\_T SlaveAddr: 目標 EC-Slave 代號，同上述並初始化為上述值。

U16\_T SlaveSlotNum: 單模組中之軸順序,由零遞增，初始化為 0

U16\_T Reserved: 保留，初始化為 0

U16\_T CoeObjectOffset: 軸對映的 CoE object 偏移量，初始化為 0x0800

U16\_T pdoMapOffset; 軸對映的 PDO map object 偏移量，初始化為 0x0010

U16\_T MinRxPdoIndex: 第一軸的最小 RxPDO 號碼，初始化為 0x1600

U16\_T MaxRxPdoIndex: 第一軸的最大 RxPDO 號碼，初始化為 0x1603

U16\_T MinTxPdoIndex: 第一軸的最小 TxPDO 號碼，初始化為 0x1A00

U16\_T MaxTxPdoIndex: 第一軸的最大 TxPDO 號碼，初始化為 0x1A03

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

在呼叫 *NEC\_CoE402GetAxisIdEx()* 註冊 CiA402 伺服軸時必須輸入該軸所對應的 PDO 映射資料” EcCiA402AxisMapInfo\_T”，可利用本 API 先將”

EcCiA402AxisMapInfo\_T” 進行基本初始化後，再將此資料結構參數帶入 *NEC\_CoE402GetAxisIdEx()*

**注意!** 禁止於 Callback 函式中調用此函數

**參閱:**

*NEC\_CoE402GetAxisIdEx()*

### 2.2.5. NEC\_CoE402GetAxisIdEx

註冊並取得 CiA 伺服軸的識別碼(單模組多軸)

C/C++語法:

```
RTN_ERR NEC_CoE402GetAxisIdEx( U16_T MasterId, EcCiA402AxisMapInfo_T *pInfo,
CANAxis_T *pAxis );
```

參數:

U16\_T MasterId: 指定目標 EC-Master 代號，單一 EC-Master 請設為 0

EcCiA402AxisMapInfo\_T \*pMapInfo: CiA 軸 PDO 映射資料

```
typedef struct
{
    U16_T TypeOfSlave;
    U16_T SlaveAddr;
    U16_T SlaveSlotNum;
    U16_T Reserved;           // Reserved, set zero.
    U16_T CoeObjectOffset;   // ex. 0x0800
    U16_T pdoMapOffset;      // ex. 0x0010
    U16_T MinRxPdoIndex;     // ex. 0x1600
    U16_T MaxRxPdoIndex;     // ex. 0x1603
    U16_T MinTxPdoIndex;     // ex. 0x1A00
    U16_T MaxTxPdoIndex;     // ex. 0x1A03
}EcCiA402AxisMapInfo_T;
```

U16\_T TypeOfSlave: Slave 型態

EC\_CIA402\_SERVO\_DRIVE (0):標準 CiA402 單軸模組 (單模組單軸)

EC\_MULTIPLE\_CIA402\_SLAVE (1):標準 CiA402 多軸模組 (單模組多軸)

U16\_T SlaveAddr: 目標 EC-Slave 代號，由 0 開始依接線遞增

U16\_T SlaveSlotNum: 單模組中之軸順序,由零遞增，初始化為 0

U16\_T Reserved: 保留，初始化為 0

U16\_T CoeObjectOffset: 軸對映的 CoE object 偏移量，初始化為 0x0800

U16\_T pdoMapOffset; 軸對映的 PDO map object 偏移量，初始化為 0x0010

U16\_T MinRxPdoIndex: 第一軸的最小 RxPDO 號碼，初始化為 0x1600

U16\_T MaxRxPdoIndex: 第一軸的最大 RxPDO 號碼，初始化為 0x1603

U16\_T MinTxPdoIndex: 第一軸的最小 TxPDO 號碼，初始化為 0x1A00

U16\_T MaxTxPdoIndex: 第一軸的最大 TxPDO 號碼，初始化為 0x1A03

CANAxis\_T \*pAxis: 回傳所對應的 CiA 伺服軸的控制識別碼(Identification)。用於稍後其他函式。

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

初始化 CiA402 函式庫後，使用本函式註冊 CiA402 模組並取得 CiA 伺服軸的控制代號。取得之軸代號用於爾後的其他 CiA402 函式庫的呼叫。

此 API 對於單一 CiA402 伺服軸的 Slave 裝置和單一 Slave 裝置具有多軸(\*)皆適用

**注意!** 禁止於 Callback 函式中調用此函數

(\*)單一 Slave 裝置具有多軸 例如 NEXCOM AXE-5904 (4 axes CoE step/servo interface module)

**參閱:**

NEC\_CoE402CyclicProcess(); NEC\_CoE402GetAxisId()



### 2.2.6. NEC\_CoE402ResetPdoMapping

重置 PDO mapping 設定

**C/C++語法:**

```
RTN_ERR NEC_CoE402ResetPdoMapping( CANAxis_T Axis );
```

**參數:**

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得。

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

重置函式庫內的 PDO mapping 設定，呼叫 NEC\_CoE402Reset()時會自動呼叫此函數。當 CiA402 Slave 的狀態再進入“OP”狀態之前。使用者必須正確的設定 PDO mapping 方式。其設定方式有下列 2 種方式:

1. 使用 NEC\_CoE402AddPdoMapping()將 PDO 的 Mapping 方式輸入置函式庫
2. 使用 NEC\_CoE402UpdatePdoMapping()，直接由 CiA402 模組端讀取 PDO mapping 的設定。

建議採用方法二叫簡潔不易發生人為錯誤。

**注意!** 禁止於 Callback 函式中調用此函數

**參閱:**

NEC\_CoE402AddPdoMapping()

### 2.2.7. NEC\_CoE402AddPdoMapping

新增 PDO mapping 設定

**C/C++語法:**

```
RTN_ERR NEC_CoE402AddPdoMapping( CANAxis_T Axis, U16_T Type, U16_T  
CANIndex, U8_T CANSubIndex, U8_T OffsetOfByte, U8_T LenOfByte );
```

**參數:**

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U16\_T Type, 指定新增的項目為 **TYPE\_RXPDO (0)**或 **TYPE\_TXPDO (1)**

U16\_T CANIndex: CAN 物件 Index，請參 CoE402 伺服馬達使用說明

U8\_T CANSubIndex: CAN 物件 sub-index，請參 CoE402 伺服馬達使用說明

U8\_T OffsetOfByte: 該物件在 Slave PDO Mapping 的位置偏移量，byte 為單位

U8\_T LenOfByte: 該物件的長度大小，byte 為單位

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

使用 *NEC\_CoE402SetParameter()* / *NEC\_CoE402GetParameter()* Ec-Master 會根據 PDO mapping 規則自動判斷以 SDO 或 PDO 的方式存取之。透過下列兩種方式讓 EC-Master 取得 CiA402 slave 模組的 PDO mapping 規則。

方法一:

使用本函數手動設定 EC-Slave 裝置的 PDO mapping，在使用前先呼叫 *NEC\_CoE402ResetPdoMapping()*清除 PDO mapping 的設定。上述方式是使用者建立該裝置的 PDO mapping 規則，基本上必須依照 ENI 的內容設定，若設定不一致可能出現異常。

方法二:

另外也可用 *NEC\_CoE402UpdatePdoMapping()*函數自動從 EC-Slave 上更新 PDO mapping 資料。

以上兩種方式請擇一使用。切勿混用。

**注意!** 禁止於 Callback 函式中調用此函數

**參閱:**

*NEC\_CoE402ResetPdoMapping()*; *NEC\_CoE402UpdatePdoMapping()*

### 2.2.8. NEC\_CoE402UpdatePdoMapping

更新 PDO mapping 設定

**C/C++語法:**

```
RTN_ERR NEC_CoE402UpdatePdoMapping( CANAxis_T Axis );
```

**參數:**

CANAxis\_T Axis, 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

使用 *NEC\_CoE402UpdatePdoMapping()* 函數自動從 CiA402 的 EC-Slave 上更新 PDO mapping 資料。注意在函數必須在 EC-Slave 狀態為“PREOP”，“SAFEOP”或 OP 狀態下方能呼叫此函數。

**注意!** 禁止於 Callback 函式中調用此函數

**參閱:**

NEC\_CoE402AddPdoMapping();

## 2.3. CiA402 狀態控制函式

### 2.3.1. NEC\_CoE402CyclicProcess

周期狀態控制函式

**C/C++語法:**

```
RTN_ERR NEC_CoE402CyclicProcess();
```

**參數:**

<無參數>

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

標準 CiA402 slave 模組可透過控制 ControlWord(0x6040) 和 StatusWord(0x6041) 來切換該模組的狀態。

使用者除了可以自行控制 ControlWord 來切換 CiA402 的狀態，亦可透過本函數來切換狀態。

透過 NEC\_CoE402GetAxisId()將 CoE 從站模組註冊到系統中，所有已註冊的模組的狀態將由 NEC\_CoE402CyclicProcess()管控。

本函數需週期性的執行，一般在 Callback 中呼叫執行本函數。本函數用於控制所有 CiA402 EC-Slaves 的狀態，內部實作狀態切換程序，用戶可直接下達欲切換的 CiA402 狀態，無須自行處理狀態切換。切換狀態相關函數請參考 NEC\_CoE402ChangeState(), NEC\_CoE402SetState()和 NEC\_CoE402ServoOn()

注意，本函數會控制 ControlWord(0x6060) 有關狀態切換相關的(Bit0~3)。若使用者使用本函數後又自行控制 ControlWord(0x6040)必須避免修改 bit0~3 的內容，否則可能造成競爭狀態(Race condition)。

Bit NO.	Function
0	Switch on
1	Enable voltage
2	Quick Stop

3	Enable operation
---	------------------

( CiA402 ControlWord bit0~3 定義 )

注意! ControlWord(0x6040)必須被設定為 RxPDO mapping , StatusWord (0x6041)必須被設定為 TxPDO mapping 。

參閱:

NEC\_CoE402ChangeState(); NEC\_CoE402SetState(); NEC\_CoE402GetState();  
NEC\_CoE402ServoOn()

### 2.3.2. NEC\_CoE402SetCtrlWord

### 2.3.3. NEC\_CoE402GetCtrlWord

設定/讀取 CiA402 ControlWord (0x6040)

#### C/C++語法:

```
RTN_ERR NEC_CoE402SetCtrlWord( CANAxis_T Axis, U16_T CtrlInBit );
```

```
RTN_ERR NEC_CoE402GetCtrlWord( CANAxis_T Axis, U16_T *CtrlInBit );
```

#### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U16\_T CtrlInBit: 設定 CiA402 ControlWord, Object index = 0x6040

U16\_T \*CtrlInBit: 讀取 CiA402 ControlWord, Object index = 0x6040

#### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

#### 用法:

本函數用於控制 CiA402 ControlWord，物件序號 0x6040，本函數可用於 Callback 函數之中。注意! ControlWord(0x6040)必須被設定為 RxPDO mapping。ControlWord 於 CiA402 中的定義如下:

Bit NO.	Function
0	Switch on
1	Enable voltage
2	Quick Stop
3	Enable operation
4 ~ 6	<Depend on operation mode> (*)
7	Fault reset
8	Halt
9	<Depend on operation mode> (*)
10	Reserved
11~15	<Vendor specific> (*)

(\*)請參與 CiA402 裝置使用手冊

若採用 NEC\_CoE402CyclicProcess()來控制 CiA402 狀態切換，該函式會控制 ControlWord(0x6040)的 Bit0~3。因此應避免 Bit0~3 的修改。若要同時使用，可先讀取 ControlWord 修改欲設定的 bit 後再寫入。

參閱：

NEC\_CoE402CyclicProcess()

### 2.3.4. NEC\_CoE402GetStatusWord

讀取 CiA402 StatusWord (0x6041)

**C/C++語法:**

```
RTN_ERR NEC_CoE402GetStatusWord( CANAxis_T Axis, U16_T *StatusInBit );
```

**參數:**

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U16\_T \*StatusInBit: 讀取 CiA402 StatusWord, Object index = 0x6041

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

本函數用於讀取 CiA402 StatusWord(物件序號 0x6041)，本函數可用於 Callback 函數之中。注意! StatusWord (0x6041)必須被設定為 TxPDO mapping。StatusWord 於 CiA402 中的定義如下:

Bit NO.	Function	Bit NO.	Function
0	Ready to switch on	8	<Vendor specific> (*)
1	switch on	9	Remote
2	Operatiion enable	10	<Depend on operation mode> (*)
3	Fault	11	Internal limit active
4	Voltage enable	12	<Depend on operation mode> (*)
5	Quick stop	13	
6	Swithec on disabled	14	<Vendor specific> (*)
7	Warning	15	

**參閱:**



### 2.3.5. NEC\_CoE402ChangeState

改變控制軸的狀態

**C/C++語法:**

```
RTN_ERR NEC_CoE402ChangeState( CANAxis_T Axis, U16_T TargetState, I32_T
TimeoutMs );
```

**參數:**

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U16\_T TargetState: 欲切換之 CiA402 狀態。

#define	CiA402 state	Note
COE_STA_DISABLE(0)	Switch on disable	Servo OFF
COE_STA_READY_TO_SWITCH_ON(1)	Ready to switch on	
COE_STA_SWITCH_ON (2)	Switched on	
COE_STA_OPERATION_ENABLE (3)	Operation enable	Servo ON
COE_STA_QUICK_STOP_ACTIVE (4)	Quick stop active	

I32\_T TimeoutMs: 切換狀態逾時等待時間，單位 millisecond。當 TimeoutMs 設定為 0 時，等同於呼叫 NEC\_CoE402SetState()函式。

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

設定馬達狀態，當函數成功返回表示已成功切換狀態。若需在 Callback 之中切換馬達狀態必須使用 NEC\_CoE402SetState()

本函式必須配合 NEC\_CoE402CyclicProcess()一起使用。

**注意!** 禁止於 Callback 函式中調用此函數

**參閱:**

NEC\_CoE402CyclicProcess(); NEC\_CoE402SetState();

### 2.3.6. NEC\_CoE402SetState

設定驅動器狀態

**C/C++語法:**

```
RTN_ERR NEC_CoE402SetState( CANAxis_T Axis, U16_T State );
```

**參數:**

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U16\_T State: 欲切換之 CiA402 狀態。

#define	CiA402 state	Note
COE_STA_DISABLE(0)	Switch on disable	
COE_STA_READY_TO_SWITCH_ON(1)	Ready to switch on	
COE_STA_SWITCH_ON (2)	Switched on	
COE_STA_OPERATION_ENABLE (3)	Operation enable	Servo ON
COE_STA_QUICK_STOP_ACTIVE (4)	Quick stop active	

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

本函數可於 Callback 函式中使用，用於設定欲切換之馬達狀態，此函數設定完後立即返回不會等待馬達狀態改變，用戶必須使用 NEC\_CoE402GetState()確定馬達的狀態。注意! ControlWord(0x6040)必須被設定為 RxPDO mapping。

本函式必須配合 NEC\_CoE402CyclicProcess()一起使用。

**參閱:**

NEC\_CoE402CyclicProcess(); NEC\_CoE402ChangeState();

### 2.3.7. NEC\_CoE402GetState

讀取驅動器狀態

**C/C++語法:**

```
RTN_ERR NEC_CoE402GetState( CANAxis_T Axis, U16_T *State );
```

**參數:**

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U16\_T \*State: 回傳馬達狀態。

#define	CiA402 state	Note
COE_STA_DISABLE(0)	Switch on disable	
COE_STA_READY_TO_SWITCH_ON(1)	Ready to switch on	
COE_STA_SWITCH_ON (2)	Switched on	
COE_STA_OPERATION_ENABLE (3)	Operation enable	Servo ON
COE_STA_QUICK_STOP_ACTIVE (4)	Quick stop active	
COE_STA_FAULT(5)	Fault	Error
COE_STA_FAULT_REACTION_ACTIVE (6)	Fault reaction active	Error

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

本函數可於 Callback 函式中使用，讀取馬達目前實際狀態。

**參閱:**

NEC\_CoE402SetState();

### 2.3.8. NEC\_CoE402FaultReset

清除/重置伺服驅動器錯誤

#### C/C++語法:

```
RTN_ERR NEC_CoE402FaultReset( CANAxis_T Axis, I32_T TimeoutMs );
```

#### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

I32\_T TimeoutMs: 逾時等待時間，單位 millisecond。當 TimeoutMs 設定為 0 或小於 0 時等待時間設定為 5000ms

#### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

#### 用法:

當驅動器狀態進入 Fault 狀態時，先確認驅動器的錯誤狀態並施以排除後，使用此函數解除 Fault 狀態。

**注意!** 禁止於 Callback 函式中調用此函數

#### 參閱:

### 2.3.9. NEC\_CoE402ServoOn

設定驅動器激磁/解激磁

#### C/C++語法:

```
RTN_ERR NEC_CoE402ServoOn( CANAxis_T Axis, U16_T OnOff );
```

#### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U16\_T OnOff:

0: Servo off (Switch on disable)

1 Servo ON (Operation enable)

#### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

#### 用法:

設定驅動器激磁(ServoOn)與解激磁(ServoOff)

ServoOn 相當於 CiA 狀態: Operation Enable

ServoOff 相當於 CiA 狀態: Switch on disable

本函式必須配合 NEC\_CoE402CyclicProcess()一起使用。

**注意!** 禁止於 Callback 函式中調用此函數

#### 參閱:

NEC\_CoE402CyclicProcess(); NEC\_CoE402SetState(); NEC\_CoE402ChangeState();

## 2.4. CiA402 驅動器基本操作函式

### 2.4.1. NEC\_CoE402SetOperationMode

設定驅動器運動模式(0x6060)

**C/C++語法:**

```
RTN_ERR NEC_CoE402SetOperationMode( CANAxis_T Axis, I8_T MotionMode,
I32_T CheckTimeoutMs );
```

**參數:**

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

I8\_T MotionMode: 設定運動模式。其支援的模式請參考驅動器使用手冊

Define	CiA402 運動模式
CiA402_OP_MODE_PROFILE_POSITION (1)	Profile Position mode
CiA402_OP_MODE_PROFILE_VELOCITY (3)	Profile Velocity mode
CiA402_OP_MODE_TORQUE_PROFILE (4)	Torque Profile mode
CiA402_OP_MODE_HOMING (6)	Homing mode
CiA402_OP_MODE_INTERPOLATED_POSITION (7)	Interpolated Position mode
CiA402_OP_MODE_CYCLIC_POSITION (8)	Cyclic Sync Position mode
CiA402_OP_MODE_CYCLIC_VELOCITY (9)	Cyclic Sync Velocity mode
CiA402_OP_MODE_CYCLIC_TORQUE (10)	Cyclic Sync Torque mode

I32\_T CheckTimeoutMs: 逾時等待時間，單位 millisecond。

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

設定驅動器運動模式(Mode of operation)，物件序號 0x6060。函數成功返回代表已成功切換驅動器的運動模式。

**注意!** 禁止於 Callback 函式中調用此函數

**參閱:**

## 2.4.2. NEC\_CoE402GetOperationModeDisplay

讀取實際驅動器運動模式(0x6061)

**C/C++語法:**

```
RTN_ERR NEC_CoE402GetOperationModeDisplay( CANAxis_T Axis, I8_T
*MotionMode );
```

**參數:**

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得  
I8\_T \*MotionMode: 回傳目前驅動器的運動狀態

Define	CiA402 運動模式
CiA402_OP_MODE_PROFILE_POSITION (1)	Profile Position mode
CiA402_OP_MODE_PROFILE_VELOCITY (3)	Profile Velocity mode
CiA402_OP_MODE_TORQUE_PROFILE (4)	Torque Profile mode
CiA402_OP_MODE_HOMING (6)	Homing mode
CiA402_OP_MODE_INTERPOLATED_POSITION (7)	Interpolated Position mode
CiA402_OP_MODE_CYCLIC_POSITION (8)	Cyclic Sync Position mode
CiA402_OP_MODE_CYCLIC_VELOCITY (9)	Cyclic Sync Velocity mode
CiA402_OP_MODE_CYCLIC_TORQUE (10)	Cyclic Sync Torque mode

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

讀回驅動器運動模式(Mode of operation display)，物件序號 0x6061。

**注意!** 禁止於 Callback 函式中調用此函數

**參閱:**

### 2.4.3. NEC\_CoE402SetParameter

設定 CiA402 驅動器物件參數

#### C/C++語法:

```
RTN_ERR NEC_CoE402SetParameter( CANAxis_T Axis, U16_T Index, U8_T SubIndex,  
U8_T LenOfByte, I32_T Value);
```

#### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U16\_T Index: CiA 物件參數序號，請參閱驅動器手冊

U8\_T SubIndex: CiA 物件參數子序號，請參閱驅動器手冊

U8\_T LenOfByte: 物件的大小(1~4 bytes)，單位 byte

I32\_T Value: 物件參數值

#### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”(0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

#### 用法:

設定 CiA402 驅動器參數。使用本函數存取 CiA402 模組參數，本函數會自動根據該模組目前 PDO mapping 的方式自動採用 PDO 或者 SDO 的方式存取。若物件參數為 PDO mapping 參數，本指令以 PDO 存取。若非 PDO mapping 參數則採用 SDO 方式存取。使用本函數必須前必須在程式初始化的過程中呼叫 NEC\_CoE402UpdatePdoMapping()。

驅動器的相關參數請參閱驅動器使用手冊。

**注意!** 禁止於 Callback 函式中調用此函數

#### 參閱:

NEC\_CoE402GetParameter (); NEC\_CoE402UpdatePdoMapping()



#### 2.4.4. NEC\_CoE402GetParameter

讀取 CiA402 驅動器物件參數

##### C/C++語法:

```
RTN_ERR NEC_CoE402GetParameter( CANAxis_T Axis, U16_T Index, U8_T SubIndex,  
U8_T LenOfByte, void *pRetVal );
```

##### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U16\_T Index: CiA 物件參數序號，請參閱驅動器手冊

U8\_T SubIndex: CiA 物件參數子序號，請參閱驅動器手冊

U8\_T LenOfByte: 物件的大小(1~4 bytes)，單位 byte

void \*pRetVal: 回傳物件參數值，給入指標的資料值的大小必需符合 LenOfByte 以避免指標存取錯誤。

##### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

##### 用法:

讀取 CiA402 驅動器的參數，驅動器的相關參數請參閱驅動器使用手冊。使用本函數存取 CiA402 模組參數，本函數會自動根據該模組目前 PDO mapping 的方式自動採用 PDO 或者 SDO 的方式存取。若物件參數為 PDO mapping 參數，本指令以 PDO 存取。若非 PDO mapping 參數則採用 SDO 方式存取。使用本函數必須前必須在程式初始化的過程中呼叫 NEC\_CoE402UpdatePdoMapping()。

**注意!** 禁止於 Callback 函式中調用此函數

##### 參閱:

NEC\_CoE402SetParameter (); NEC\_CoE402UpdatePdoMapping();

#### 2.4.5. NEC\_CoE402GetActualPosition

讀取驅動器馬達實際位置(0x6064)

##### C/C++語法:

```
RTN_ERR NEC_CoE402GetActualPosition( CANAxis_T Axis, I32_T *Position );
```

##### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

I32\_T \*Position: 回傳馬達實際位置(物件序號:0x6040)

##### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”(0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

##### 用法:

本函數可於 Callback 函式中使用，必須將物件 0x6064 設定為 TxPDO mapping 物件。

##### 參閱:

#### 2.4.6. NEC\_CoE402SetTargetPosition

#### 2.4.7. NEC\_CoE402GetTargetPosition

設定/讀取目標位置(0x607A,PP,CSP)

##### C/C++語法:

```
RTN_ERR NEC_CoE402SetTargetPosition( CANAxis_T Axis, I32_T TargetPos );
```

```
RTN_ERR NEC_CoE402GetTargetPosition( CANAxis_T Axis, I32_T *TargetPos );
```

##### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

I32\_T TargetPos: 設定目標位置(CoE: 0x607A)

I32\_T \*TargetPos: 回傳目標位置(CoE: 0x607A)

##### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”(0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

##### 用法:

本函數可於 Callback 函式中使用，必須將物件 0x607A 設定為 RxPDO mapping 物件。TargetPositon 被用於 Profile Position mode(PP)和 Cyclic Sync Position mode(CSP)運動模式，可用 NEC\_CoE402SetOperationMode()切換運動模式

##### 參閱:

NEC\_CoE402SetOperationMode()

#### 2.4.8. NEC\_CoE402SetQuickStopDec

設定 Quick Stop 減速率(0x6085)

##### C/C++語法:

```
RTN_ERR NEC_CoE402SetQuickStopDec( CANAxis_T Axis, U32_T QuickStopDec );
```

##### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U32\_T QuickStopDec: 設定 Quick Stop 減速率(0x6085)

##### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”(0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

##### 用法:

此函數等同呼叫 NEC\_CoE402SetParameter( Axis, 0x6085, 0, 4, (I32\_T) QuickStopDec );

**注意!** 禁止於 Callback 函式中調用此函數

##### 參閱:

NEC\_CoE402SetParameter ();

#### 2.4.9. NEC\_CoE402SetSoftPosLimit

設定軟體位置極限(0x607D)

##### C/C++語法:

```
RTN_ERR NEC_CoE402SetSoftPosLimit( CANAxis_T Axis, I32_T MinPositionLimit,  
I32_T MaxPositionLimit);
```

##### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

I32\_T MinPositionLimit: 負軟體極限位置 (0x607D : 01)

I32\_T MaxPositionLimit: 正軟體極限位置 (0x607D : 02)

##### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

##### 用法:

若要解除軟體正負極限的功能將正負極限同時設定為 0。

此函數等同呼叫 NEC\_CoE402SetParameter( Axis, 0x607D, 1, 4, MinPositionLimit )

設定負極限 and NEC\_CoE402SetParameter( Axis, 0x607D, 2, 4, MaxPositionLimit )

設定正極限.

**注意!** 禁止於 Callback 函式中調用此函數

##### 參閱:

NEC\_CoE402SetParameter ();

#### 2.4.10. NEC\_CoE402SetMaxVelLimit

設定最大速度限制(0x607F)

##### C/C++語法:

```
RTN_ERR NEC_CoE402SetMaxVelLimit( CANAxis_T Axis, U32_T MaxVelocityLimit );
```

##### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U32\_T MaxVelocityLimit: 設定最大速度限制(0x607F)

##### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

##### 用法:

在 Profile Position mode (P2P), Profile Velocity mode(Jog) 等模式中設定最大速度限制。此函數等同呼叫 NEC\_CoE402SetParameter( Axis, 0x607F, 0, 4, (I32\_T) MaxVelocityLimit );

**注意!** 禁止於 Callback 函式中調用此函數

##### 參閱:

NEC\_CoE402SetParameter ();

#### 2.4.11. NEC\_CoE402SetTargetVelocity

設定目標速度(0x60FF,PV,CSV)

##### C/C++語法:

```
RTN_ERR NEC_CoE402SetTargetVelocity ( CANAxis_T Axis, I32_T TargetVel );
```

##### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

I32\_T TargetVel: 設定目標速度(CoE: 0x60FF)

##### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

##### 用法:

本函數可於 Callback 函式中使用，必須將物件 0x60FF 設定為 RxPDO mapping 物件。TargetVelocity 被用於 Profile Velocity mode(PV)和 Cyclic Sync Velocity mode(CSV)運動模式，可用 NEC\_CoE402SetOperationMode()切換運動模式

##### 參閱:

NEC\_CoE402SetOperationMode()

#### 2.4.12. NEC\_CoE402GetActualVelocity

讀取當前速度(0x606C,PV,CSV)

##### C/C++語法:

```
RTN_ERR NEC_CoE402GetActualVelocity( CANAxis_T Axis, I32_T *ActualVel);
```

##### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

I32\_T \*ActualVel: 讀取當前速度(CoE: 0x606C)

##### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”(0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

##### 用法:

本函數可於 Callback 函式中使用，必須將物件 0x606C 設定為 TxPDO mapping 物件。ActualVelocity 被用於 Profile Velocity mode(PV)和 Cyclic Sync Velocity mode(CSV)運動模式，可用 NEC\_CoE402SetOperationMode()切換運動模式

##### 參閱:

NEC\_CoE402SetOperationMode()



## 2.5. 點對點運動 (Profile Position mode)

### 2.5.1. NEC\_CoE402Ptp

### 2.5.2. NEC\_CoE402PtpV

### 2.5.3. NEC\_CoE402PtpA

啟動單軸點對點運動(Profile Position mode)

**C/C++語法:**

```
RTN_ERR NEC_CoE402Ptp( CANAxis_T Axis, U32_T Option, I32_T TargetPos );
RTN_ERR NEC_CoE402PtpV( CANAxis_T Axis, U32_T Option, I32_T TargetPos, U32_T
MaxVel );
RTN_ERR NEC_CoE402PtpA( CANAxis_T Axis, U32_T Option, I32_T TargetPos, U32_T
MaxVel, U32_T Acc, U32_T Dec );
```

**參數:**

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U32\_T Option: 功能選項，採 bit 形式。兩個功能以上，以 OR 方式輸入

OPT\_ABS (0x00000000): TargetPos 為絕對位置座標

OPT\_REL( 0x00000040): TargetPos 為相對位置座標

OPT\_WMC (0x00010000 ): 等待 PTP 運動位置到達

OPT\_IMV( 0x10000000 ): 忽略 MaxVel 輸入參數

OPT\_IAC (0x20000000 ):忽略 Acc 輸入參數

OPT\_IDC (0x40000000): 忽略 Dec 輸入參數

I32\_T TargetPos: 絕對或相對目標位置(CoE: Target Position 0x607A), 由 Option 設定決定 TargetPos 代表絕對或相對位置。

U32\_T MaxVel: 最大速度 (CoE: Profile Velocity 0x6081)

U32\_T Acc: 加速率 (CoE: Profile Acceleration 0x6083)

U32\_T Dec: 減速率 (CoE: Profile Deceleration 0x6084)

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”(0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

本函式使用“Profile Position mode”來完成點對點運動，因此須確認 CiA402 模組是否支援 Profile Position (PP)模式。呼叫本函式時會檢查並自動切換 Operation Mode 是否為 PP 模式。而 Ptp, PtpV 和 PtpA 的差別在於輸入參數不同：

*NEC\_CoE402Ptp()*只設定目標位置 TargetPos (CoE: Target Position 0x607A) 並啟動點對點運動，最大速度 MaxVel (CoE: Profile Velocity 0x6081)及加減速度 Acc, Dec (CoE: Profile Acceleration 0x6083, Profile Deceleration 0x6084) 基本上是依照 CiA402 模組目前的設定值。

*NEC\_CoE402PtpV()* 設定目標位置 TargetPos (CoE: Target Position 0x607A) 和最大速度 MaxVel (CoE: Profile Velocity 0x6081)並啟動點對點運動，而加減速度 Acc, Dec (CoE: Profile Acceleration 0x6083, Profile Deceleration 0x6084)是依照 CiA402 模組目前的設定值。

*NEC\_CoE402PtpA()*設定所有相關參數包含目標位置 TargetPos (CoE: Target Position 0x607A)，最大速度 MaxVel (CoE: Profile Velocity 0x6081)，加速度 Acc (CoE: Profile Acceleration 0x6083) 和減速度 Dec (CoE: Profile Deceleration 0x6084)並啟動點對點運動。

Option “OPT\_REL ( 0x00000040)” 決定 Target Position 參數為相對位置或絕對位置

當 Option:OPT\_WMC (Wait motion complete) 至能(enable)，此函數會等待點對點運動完成後或錯誤發生時返回，其效果等同於呼叫 *NEC\_CoE402WaitTargetReached()*

OPT\_IMV( 0x10000000 ), OPT\_IAC (0x20000000 )和 OPT\_IDC (0x40000000)是適用於 *NEC\_CoE402PtpA()*，表示忽略輸入參數使用模組內部設定值。

呼叫 *NEC\_CoE402Halt()* 可以暫停或從啟 PTP 運動。

**注意!** 禁止於 Callback 函式中調用此函數

參閱：

*NEC\_CoE402WaitTargetReached();NEC\_CoE402Halt()*

#### 2.5.4. NEC\_CoE402WaitTargetReached

等待目標位置到達

##### C/C++語法:

```
RTN_ERR NEC_CoE402WaitTargetReached( CANAxis_T Axis );
```

##### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

##### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”（0），反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

##### 用法:

此函數為同步函數，通常配合 NEC\_CoE402Ptp(), NEC\_CoE402PtpV(), NEC\_CoE402PtpA(), NEC\_CoE402Jog() 和 NEC\_CoE402JogA() 函式使用。

若在 PTP 後呼叫，呼叫後直到 PTP 運動結束後才返回。

若在 Jog 後呼叫，呼叫後直到到達 Target Velocity 速度後才返回。

若返回錯誤代碼表示運動失敗或者暫停。

**注意!** 禁止於 Callback 函式中調用此函數

##### 參閱:

NEC\_CoE402Ptp(); NEC\_CoE402PtpV();

NEC\_CoE402PtpA(); NEC\_CoE402Jog(); NEC\_CoE402JogA();



## 2.6. 速度運動(Profile Velocity mode)

### 2.6.1. NEC\_CoE402Jog

### 2.6.2. NEC\_CoE402JogA

啟動單軸指定速度運轉(Profile velocity mode)

#### C/C++語法:

```
RTN_ERR NEC_CoE402Jog( CANAxis_T Axis, U32_T Option, I32_T MaxVel );  
RTN_ERR NEC_CoE402JogA( CANAxis_T Axis, U32_T Option, I32_T MaxVel, I32_T Acc,  
I32_T Dec );
```

#### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U32\_T Option: 功能選項，採 bit 形式。兩個功能以上，以 OR 方式輸入

OPT\_WMC (0x00010000): 等待到達所設定的最大速度後函式返回

OPT\_IAC (0x20000000): 忽略 Acc 輸入參數

OPT\_IDC (0x40000000): 忽略 Dec 輸入參數

I32\_T MaxVel: 目標速度 (CoE: Target Velocity 0x60FF)

數值正負決定運動方向。

U32\_T Acc: 加速率 (CoE: Profile Acceleration 0x6083)

U32\_T Dec: 減速率 (CoE: Profile Deceleration 0x6084)

#### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

#### 用法:

本函式使用“Profile Velocity mode”來完成 Jog 運動，因此須確認 CiA402 模組是否支援 Profile Velocity(PV)模式。呼叫本函式時會檢查並自動切換 Operation Mode 是否為 PV 模式。而 Jog 和 JogA 的差別在於輸入參數不同:

NEC\_CoE402Jog()只設定目標速度 “MaxVel” (CoE: Target Velocity 0x60FF) 並啟動 Jog 運動，加減速度(CoE: Profile Acceleration and Deceleration: 0x6083, 0x6084) 基本上是依照 CiA402 模組目前的設定值。

*NEC\_CoE402JogA()* 設定所有相關參數包含目標速度"MaxVel" (CoE: Target Velocity 0x60FF)，加速度 "Acc" (CoE: Profile Acceleration 0x6083) 和減速度"Dec" (CoE: Profile Deceleration 0x6084)並啟動 Jog 運動。

當 Option:OPT\_WMC (Wait motion complete) 至能(enable)，此函數會等待 Jog 運動到達設定的速度後或錯誤發生時返回，其效果等同於呼叫 *NEC\_CoE402WaitTargetReached()*

OPT\_IAC (0x20000000 )和 OPT\_IDC (0x40000000)是適用於 *NEC\_CoE402JogA ()*，表示忽略輸入參數使用模組內部設定值。

呼叫 *NEC\_CoE402Halt()* 可以暫停或從啟 Jog 運動。

**注意!** 禁止於 Callback 函式中調用此函數

**參閱:**

*NEC\_CoE402WaitTargetReached(); NEC\_CoE402Halt()*

### 2.6.3. NEC\_CoE402Halt

停止/暫停運動

**C/C++語法:**

```
RTN_ERR NEC_CoE402Halt( CANAxis_T Axis, I32_T OnOff );
```

**參數:**

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

I32\_T OnOff: 0:解除，1:停止或暫停

**回傳值:**

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”(0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

**用法:**

本含式主要控制 Controlword bit8:Halt (CoE: 0x6040)，用於停止或暫停目前的運動，停止的動作行為通常搭配 Halt option code (CoE:0x605D)參數的設定。詳細的動作規格需參閱該模組說明書。

NEC\_CoE402Halt()指令適用於所有操作模式(mode of operation)，當參數“OnOff”設定為 1 時表示停止或暫停運動，若要重啟運動必須先解除 Halt 狀態，須將參數“OnOff”設定為 0。

**注意!** 禁止於 Callback 函式中調用此函數

**參閱:**

NEC\_CoE402SetCtrlWord(); NEC\_CoE402GetCtrlWord()

## 2.7. 歸零運動 (Homing mode)

### 2.7.1. NEC\_CoE402Home

### 2.7.2. NEC\_CoE402HomeEx

啟動單軸原點歸零運動(Homing)

#### C/C++語法:

```
RTN_ERR NEC_CoE402Home( CANAxis_T Axis, U32_T Option );  
RTN_ERR NEC_CoE402HomeEx( CANAxis_T Axis, U32_T Option, I8_T Method, I32_T  
Offset, U32_T MaxVel, U32_T ZeroVel, U32_T Acc );
```

#### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U32\_T Option: 功能選項，採 bit 形式。兩個功能以上，以 OR 方式輸入

OPT\_WMC (0x00010000): 等待 Home 運動完成

OPT\_MTD (0x08000000): 忽略 Method 輸入參數

OPT\_IMV (0x10000000): 忽略 MaxVel 輸入參數

OPT\_IAC (0x20000000): 忽略 Acc 輸入參數

OPT\_IZV (0x40000000): 忽略 ZeroVel 輸入參數

OPT\_IOF (0x80000000): 忽略 Offset 輸入參數

I8\_T Method: 歸零模式(CoE: 0x6098)，請參與 CiA402 驅動器手冊

I32\_T Offset: 原點位置偏移量(CoE: 0x607C)

U32\_T MaxVel: 歸零運動最大速度(CoE: 0x6099:1)

U32\_T ZeroVel: 靠近尋找原點速度(CoE: 0x6099:2)

U32\_T Acc: 歸零運動加速率(CoE: 0x609A)

#### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”(0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

#### 用法:

本函式使用 CoE “Homing mode”來完成歸零運動(Homing)，因此須確認 CiA402 模組是否支援 Homing(HM)模式。呼叫本函式時會檢查並自動切換 Operation Mode 是否為 HM 模式。而 Home和 HomeEx 的差別在於輸入參數不同:



`NEC_CoE402Home()` 單純用來啟動歸零運動(Homing)，相關的 Homing 參數如歸零模式(CoE: 0x6098)，原點位置偏移量(CoE: 0x607C)等基本上是依照 CiA402 模組目前的設定值。可使用 `NEC_CoE402SetParameter()`單獨設定相關參數值。

`NEC_CoE402HomeEx()` 設定所有相關參數包含歸零模式 “Method “ (CoE: 0x6098)，原點位置偏移量 “Offset “ (CoE: 0x607C)，歸零運動最大速度 “MaxVel” (CoE: 0x6099:1)，靠近尋找原點速度 “ZeroVel” (CoE: 0x6099:2)和歸零運動加速率 “Acc” (CoE: 0x609A)並啟動 Homing 運動。

當 Option:OPT\_WMC (Wait motion complete) 至能(enable)，此函數會等待 Homing 運動到完成後或錯誤發生時返回，其效果等同於呼叫 `NEC_CoE402WaitHomeFinished()`

Option: OPT\_MTD ( 0x08000000 ), OPT\_IMV (0x10000000 ), OPT\_IAC (0x20000000 ), OPT\_IZV (0x40000000 ), OPT\_IOF (0x80000000)只適用於 `NEC_CoE402HomeEx()`，表示忽略輸入參數使用模組內部設定值。

呼叫 `NEC_CoE402Halt()` 可以暫停或從啟 Homing 運動。

### 注意!

禁止於 Callback 函式中調用此函數

除了引用“ OPT\_WMC”選項外，此函數返回時，不會清除歸零運動啟動位元。我們建議用戶在歸零運動程序完成後，調用 `NEC_CoE402ClearHomeStartBit` 函數清除歸零運動啟動位元。

### 參閱:

`NEC_CoE402WaitHomeFinished()`; `NEC_CoE402Halt()`;  
`NEC_CoE402ClearHomeStartBit()`;

### 2.7.3. NEC\_CoE402WaitHomeFinished

等待單軸原點歸零運動(Homing)結束

#### C/C++語法:

```
RTN_ERR NEC_CoE402WaitHomeFinished( CANAxis_T Axis );
```

#### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

#### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

#### 用法:

此函數為同步函數，呼叫後直到歸零運動結束後才返回。若返回錯誤代碼表示歸零運動失敗或者暫停。

**注意!** 禁止於 Callback 函式中調用此函數

#### 參閱:

```
NEC_CoE402Home();NEC_CoE402HomeEx();
```

#### 2.7.4. NEC\_CoE402ClearHomeStartBit

清除單軸原點歸零運動(Homing)啟動位元

##### C/C++語法:

```
RTN_ERR NEC_CoE402ClearHomeStartBit ( CANAxis_T Axis );
```

##### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

##### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

##### 用法:

此函數用於清除原點歸零運動啟動位元。當馬達正處在歸零運動種態下時，調用此函數，馬達的行為會依據驅動器的設計，立即停止或繼續執行歸零運動。

##### 參閱:

```
NEC_CoE402Home();NEC_CoE402HomeEx();
```

### 2.7.5. NEC\_CoE402CheckHomeStatus

檢查當前歸零運動狀態

#### C/C++語法:

```
RTN_ERR NEC_CoE402CheckHomeStatus( CANAxis_T Axis, U16_T *pStatus );
```

#### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

U16\_T \*pStatus: 傳回當前歸零運動狀態

#### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

#### 用法:

此函數用以取得當前歸零運動狀態。用戶可在執行歸零運動後，調用此函數已得知當前歸零運動狀態。此函數可能回傳的狀態如下所列：

#define HOMING_IN_PROGRESS	(0)
#define HOMING_TARGET_NOT_REACH	(1)
#define HOMING_COMPLETE	(2)
#define HOMING_INTERRUPTED	(3)
#define HOMING_ERR_VEL_ZERO	(4)
#define HOMING_ERR_VEL_NON_ZERO	(5)

除了歸零運動仍處在“HOMING\_IN\_PROGRESS”之外，此函數在返回前，會一併清除歸零運動啟動位元。

#### 參閱:

NEC\_CoE402Home(); NEC\_CoE402HomeEx();

## 2.8. 扭力控制 (Torque Control)

### 2.8.1. NEC\_CoE402SetTargetTorque

### 2.8.2. NEC\_CoE402GetTargetTorque

設定/讀取目標扭力 (CoE: 0x6071, PT, CST)

#### C/C++語法:

```
RTN_ERR NEC_CoE402SetTargetTorque( CANAxis_T Axis, I16_T TargetTorque );  
RTN_ERR NEC_CoE402GetTargetTorque( CANAxis_T Axis, I16_T *TargetTorque );
```

#### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

I16\_T TargetTorque: 設定目標扭力(CoE: 0x6071)

I16\_T \* TargetTorque: 回傳目標扭力(CoE: 0x6071)

#### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS”(0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

#### 用法:

本函數可於 Callback 函式中使用，但必須將物件 0x6071 設定為 RxPDO mapping 物件。

TargetTorque 通常被用於 Profile Torque mode(PT)和 Cyclic Sync Torque mode(CST)運動模式，可用 NEC\_CoE402SetOperationMode()切換運動模式。

Torque 單位視驅動器容量而定，通常一刻度為 0.1%，詳細請參閱相關手冊。

#### 參閱:

NEC\_CoE402SetOperationMode();NEC\_CoE402GetActualTorque()

### 2.8.3. NEC\_CoE402GetActualTorque

讀取實際扭力 (CoE: 0x6077, PT, CST)

#### C/C++語法:

```
RTN_ERR NEC_CoE402GetActualTorque( CANAxis_T Axis, I16_T  
*TorqueActualValue );
```

#### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

I16\_T \*TorqueActualValue: 回傳實際扭力 (CoE: 0x6077)

#### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

#### 用法:

本函數可於 Callback 函式中使用，但必須將物件 0x6077 設定為 TxPDO mapping 物件。

Torque 單位視驅動器容量而定，通常一刻度為 0.1%，詳細請參閱相關手冊。

#### 參閱:

NEC\_CoE402SetTargetTorque(); NEC\_CoE402GetTargetTorque();

#### 2.8.4. NEC\_CoE402SetTorqueProfile

設定目標扭力與變化率(0x6071, 0x6087)

##### C/C++語法:

```
RTN_ERR NEC_CoE402SetTorqueProfile( CANAxis_T Axis, I16_T TargetTorque, U32_T TorqueSlope );
```

##### 參數:

CANAxis\_T Axis: 指定控制軸代號。軸代號由 NEC\_CoE402GetAxisId()取得

I16\_T TargetTorque: 設定目標扭力 (CoE: 0x6071)

U32\_T TorqueSlope: 設定扭力變化率 (CoE: 0x6087)

##### 回傳值:

回傳錯誤代碼。

調用函數成功回傳“ECERR\_SUCCESS” (0)，反之函數調用失敗回傳錯誤代碼，錯誤代碼定義於EcErrors.h標頭檔中。

##### 用法:

本函式使用“Profile Torque mode”來完成 Torque 控制，因此須確認 CiA402 模組是否支援 Profile Torque (PT)模式。呼叫本函式時會檢查並自動切換 Operation Mode 是否為 PT 模式。

本函式會同時設定目標扭力 “TargetTorque” (CoE: 0x6071)與扭力變化率 “TorqueSlope” (CoE: 0x6087)，並自動判斷相關的 CoE 物件使用 PDO 或 SDO 存取。

Torque 單位視驅動器容量而定，通常一刻度為 0.1%，詳細請參閱相關手冊。

**注意!** 禁止於 Callback 函式中調用此函數

##### 參閱:

NEC\_CoE402SetOperationMode(); NEC\_CoE402SetTargetTorque();  
NEC\_CoE402GetActualTorque()