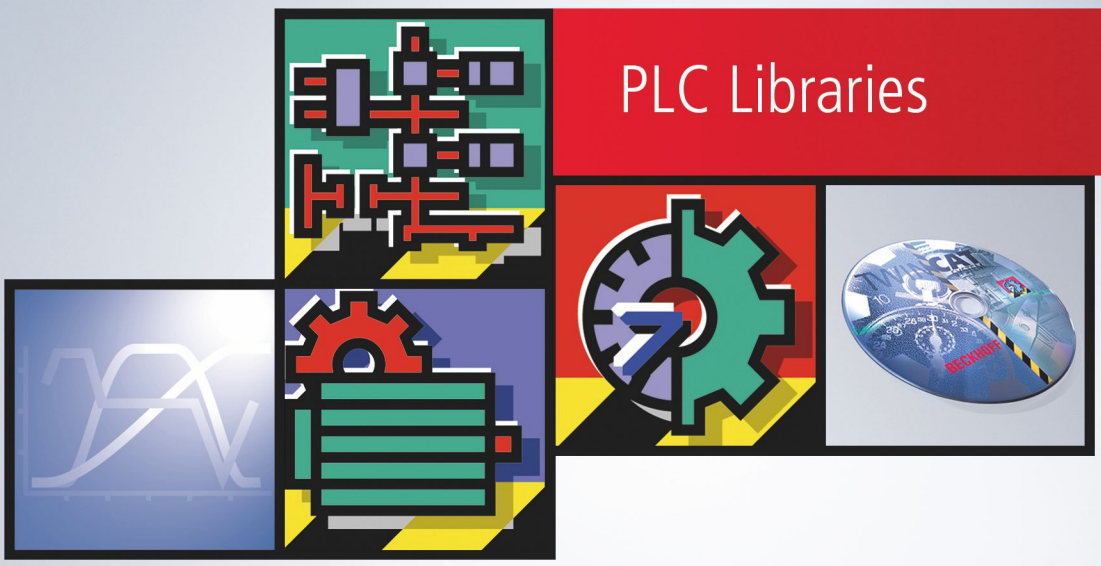
BECKHOFF New Automation Technology

Manual | EN

TX1200

TwinCAT 2 | PLC Library: TcEtherCAT



2023-10-24 | Version: 1.2

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without notice.
No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT®

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview

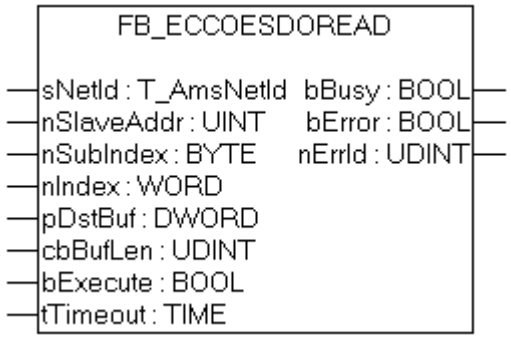
The PLC library: **TcEtherCAT.Lib** contains function blocks with which services and functions may be carried out on the EtherCAT master device and on the slave devices connected to it.

Sample project and configuration for drive diagnosis

See https://infosys.beckhoff.com/content/1033/tcplclibethercat/Resources/11934864011/.exe

3 CoE

3.1 FB_EcCoeSdoRead



The FB_EcCoeSdoRead function block allows data to be read from an EtherCAT slave through an SDO (Service Data Object) access. This requires the slave to have a mailbox and to support the "CANopen over EtherCAT" (CoE) protocol. The nSubIndex and nIndex parameters select the object that is to be read.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    nSubIndex   : BYTE;
    nIndex      : WORD;
    pDstBuf     : DWORD;
    cbBufLen    : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave to which the SDO upload command should be sent.

nSubIndex: Sub-index of the object that is to be read.

nIndex: Index of the object that is to be read.

pDstBuf: The address (pointer) of the receive buffer.

cbBufLen: The maximum available buffer size for the data to be read, in bytes.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Example of an implementation in ST:

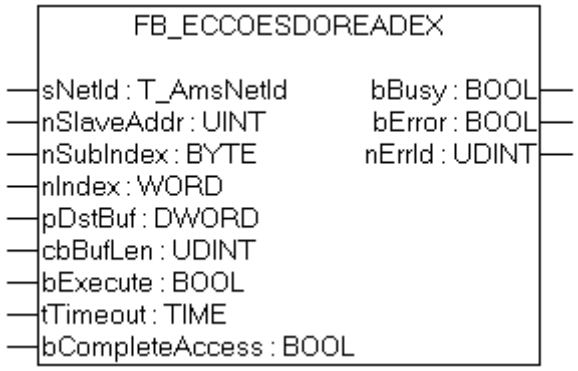
```
PROGRAM TEST_SdoRead
VAR
    fbSdoRead   : FB_EcCoESdoRead;
    sNetId      : T_AmsNetId := '172.16.2.131.2.1';
    bExecute    : BOOL;
    nSlaveAddr  : UINT := 1006;
    nIndex      : WORD := 16#1018;
    nSubIndex   : BYTE :=1;
    vendorId    : UDINT;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

```
fbSdoRead(sNetId:= sNetId,nSlaveAddr :=nSlaveAddr, nIndex:=nIndex, nSubIndex :=nSubIndex, pDstBuf:=
ADR(vendorId), cbBufLen:=SIZEOF(vendorId),bExecute:=bExecute);
bError:=fbSdoRead.bError;
nErrId:=fbSdoRead.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

3.2 FB_EcCoeSdoReadEx



The FB_EcCoeSdoReadEx function block allows data to be read from an EtherCAT slave through an SDO (Service Data Object) access. This requires the slave to have a mailbox and to support the "CANopen over EtherCAT" (CoE) protocol. The nSubIndex and nIndex parameters select the object that is to be read. The parameter can be read with all sub indexes via bCompleteAccess := TRUE.

VAR_INPUT

```
VAR_INPUT
    sNetId          : T_AmsNetId; (* AmsNetId of the EtherCAT master device.*)
    nSlaveAddr      : UINT; (* Address of the slave device.*)
    nSubIndex       : BYTE; (* CANopen Sdo subindex.*)
    nIndex          : WORD; (* CANopen Sdo index.*)
    pDstBuf         : DWORD; (* Contains the address of the buffer for the received data. *)
    cbBufLen        : UDINT; (* Contains the max. number of bytes to be received. *)
    bExecute        : BOOL; (* Function block execution is triggered by a rising edge at this input.
*)
    tTimeout        : TIME := DEFAULT_ADS_TIMEOUT;
(* States the time before the function is cancelled. *)
    bCompleteAccess : BOOL; (* access complete object*)
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave to which the SDO upload command should be sent.

nSubIndex: Sub-index of the object that is to be read.

nIndex: Index of the object that is to be read.

pDstBuf: The address (pointer) of the receive buffer.

cbBufLen: The maximum available buffer size for the data to be read, in bytes.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

bCompleteAccess: With bCompleteAccess := TRUE the complete parameter with all sub indexes is being read.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

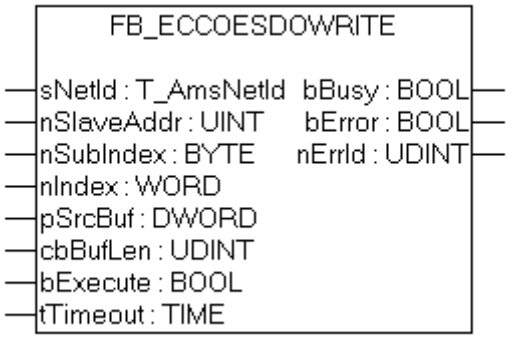
bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Requirements

Development environment	Target system type	Libraries to be linked
TwinCAT v2.10.0 Build >= 1319 or higher	PC or CX (x86)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1319 or higher	CX (ARM)	

3.3 FB_EcCoeSdoWrite



The FB_EcCoeSdoWrite function block permits an object from the object directory of an EtherCAT slave to be written by means of an SDO download. This requires the slave to have a mailbox and to support the "CANopen over EtherCAT" (CoE) protocol. The nSubIndex and nIndex parameters select the object the data should be written to.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    nSubIndex   : BYTE;
    nIndex      : WORD;
    pSrcBuf     : DWORD;
    cbBufLen    : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave to which the SDO download command should be sent.

nSubIndex: Sub-index of the object that should be written to.

nIndex: Index of the object that should be written to.

pSrcBuf: Address (pointer) of the send buffer.

cbBufLen: Number (in bytes) of data to be sent.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Example of an implementation in ST:

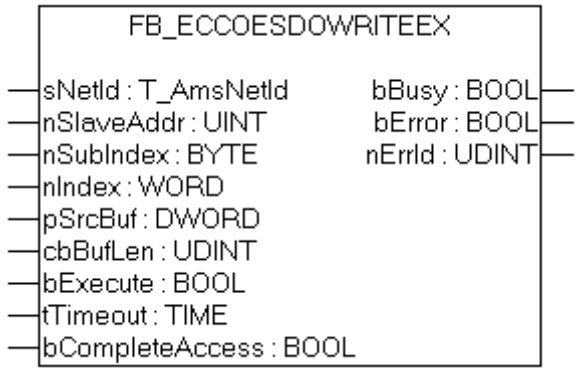
```
PROGRAM TEST_SdoWrite

VAR
    fbSdoWrite      : FB_EcCoESdoWrite;
    sNetId          : T_AmsNetId := '172.16.2.131.2.1'; (* NetId of EtherCAT Master *)
    nSlaveAddr      : UINT := 1005; (* Port Number of EtherCAT Slave *)
    nIndex          : WORD := 16#4062; (* CoE Object Index *)
    nSubIndex       : BYTE := 1; (* Subindex of CoE Object *)
    nValue          : UINT := 2; (* variable to be written to the CoE Object *)
    bExecute        : BOOL; (* rising edge starts writing to the CoE Object *)
    bError          : BOOL;
    nErrId          : UDINT;
END_VAR


fbSdoWrite(
    sNetId      := sNetId,
    nSlaveAddr  := nSlaveAddr,
    nIndex      := nIndex,
    nSubIndex   := nSubIndex,
    pSrcBuf     := ADR(nValue),
    cbBufLen    := SIZEOF(nValue),
    bExecute    := bExecute
);


IF NOT fbSdoWrite.bBusy THEN
    bExecute := FALSE;
```

```
    IF NOT bError THEN
        (* write successful *)
        bError := FALSE;
        nErrId := 0;
    ELSE
        (* write failed *)
        bError := fbSdoWrite.bError;
        nErrId := fbSdoWrite.nErrId;
    END_IF

    fbSdoWrite(bExecute := FALSE);
END_IF
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

3.4 FB_EcCoeSdoWriteEx



The FB_EcCoeSdoWrite function block permits an object from the object directory of an EtherCAT slave to be written by means of an SDO download. This requires the slave to have a mailbox and to support the "CANopen over EtherCAT" (CoE) protocol. The nSubIndex and nIndex parameters select the object the data should be written to. Via bCompleteAccess := TRUE the parameter is being written with all sub indexes.

VAR_INPUT

```
VAR_INPUT
    sNetId          : T_AmsNetId; (* AmsNetId of the EtherCAT master device.*)
    nSlaveAddr      : UINT; (* Address of the slave device.*)
    nSubIndex       : BYTE; (* CANopen Sdo subindex.*)
    nIndex          : WORD; (* CANopen Sdo index.*)
    pSrcBuf         : DWORD; (* Contains the address of the buffer containing the data to be send. *
)
    cbBufLen        : UDINT; (* Contains the max. number of bytes to be received. *)
    bExecute        : BOOL; (* Function block execution is triggered by a rising edge at this input.
*)
    tTimeout        : TIME := DEFAULT_ADS_TIMEOUT;
(* States the time before the function is cancelled. *)
    bCompleteAccess : BOOL; (* access complete object*)
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave to which the SDO download command should be sent.

nSubIndex: Sub-index of the object that should be written to.

nIndex: Index of the object that should be written to.

pSrcBuf: Address (pointer) of the send buffer.

cbBufLen: Number (in bytes) of data to be sent.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

bCompleteAccess: With bCompleteAccess := TRUE the complete parameter with all sub indexes is being written.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated, and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

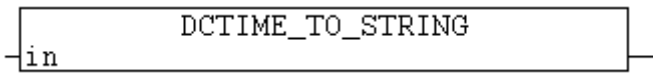
nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Requirements

Development environment	Target system type	Libraries to be linked
TwinCAT v2.10.0 Build >= 1319 or higher	PC or CX (x86)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1319 or higher	CX (ARM)	

4 Conversion Functions

4.1 DCTIME_TO_STRING



This function converts a "Distributed Clock System Time" variable to a string .

After conversion the resulting string has the following format: **'YYYY-MM-DD-hh:mm:ss.nnnnnnnnn'**

- YYYY: Year;
- MM: Month;
- DD: Day;
- hh: Hour;
- mm: Minute;
- ss: Second;
- nnnnnnnnn: Nanosecond;

FUNCTION DCTIME_TO_STRING: STRING(29)

```
VAR_INPUT
    in : T_DCTIME;
END_VAR
```

in: "Distributed Clock System Time [▶ 85]" variable to be converted.

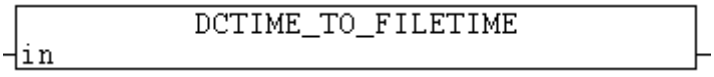
Example:

See description of: F_GetCurDcTickTime [▶ 27].

Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.10.0 Build > 1310 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

4.2 DCTIME_TO_FILETIME



This function converts the 64 bit "Distributed Clock System Time" variable to 64 bit "Windows File Time" variable.

FUNCTION DCTIME_TO_FILETIME: T_FILETIME

T_FILETIME

```
VAR_INPUT
    in : T_DCTIME;
END_VAR
```

in: "Distributed Clock System Time [▶ 85]" variable to be converted.

Example:

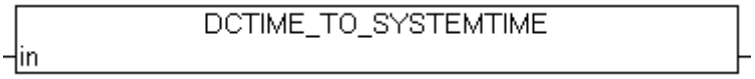
```
PROGRAM P_TEST
VAR
    ft      : T_FILETIME;
    dct     : T_DCTIME;
END_VAR
```

```
dct      := F_GetCurDcTickTime();
ft      := DCTIME_TO_FILETIME(dct);
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 Build >= 1324 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

4.3 DCTIME_TO_SYSTEMTIME



This function converts the 64 bit "Distributed Clock System Time" variable to structured "Windows System Time" variable.

FUNCTION DCTIME_TO_SYSTEMTIME: TIMESTRUCT

TIMESTRUCT

```
VAR_INPUT
    in : T_DCTIME;
END_VAR
```

in: "Distributed Clock System Time [▶ 85]" variable to be converted.

Example:

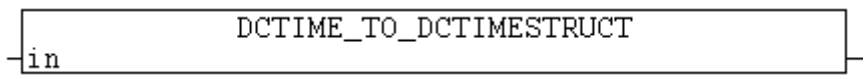
```
PROGRAM P_TEST
VAR
    syst : TIMESTRUCT;
END_VAR
```

```
syst := DCTIME_TO_SYSTEMTIME( F_GetCurDcTickTime() );
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 Build > 1340 TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

4.4 DCTIME_TO_DCTIMESTRUCT



This function converts the 64 bit "Distributed Clock System Time" variable to structured variable of type: DCTIMESTRUCT [▶ 76].

FUNCTION DCTIME_TO_DCTIMESTRUCT: DCTIMESTRUCT

DCTIMESTRUCT [▶ 76]

```
VAR_INPUT
    in : T_DCTIME;
END_VAR
```

in: "Distributed Clock System Time [▶ 85]" variable to be converted;

Example:

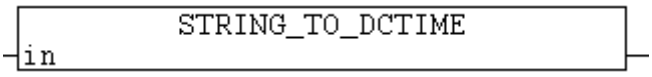
```
PROGRAM P_TEST
VAR
    dcStruct : DCTIMESTRUCT;
    dcTime   : T_DCTIME;
END_VAR
```

```
dcTime := F_GetCurDcTickTime();
dcStruct := DCTIME_TO_DCTIMESTRUCT(dcTime);
```

Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.10.0 Build > 1316 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included autimatically)

4.5 STRING_TO_DCTIME



This function converts a string to a "Distributed Clock System Time" variable.

FUNCTION STRING_TO_DCTIME: T_DCTIME

T_DCTIME [▶ 85]

```
VAR_INPUT
    in : STRING(29);
END_VAR
```

in: The string format is: **'YYYY-MM-DD-hh:mm:ss.nnnnnnnnn'**

- YYYY: Year;
- MM: Month;
- DD: Day;
- hh: Hour;
- mm: Minute;
- ss: Second;
- nnnnnnnnn: Nanosecond;

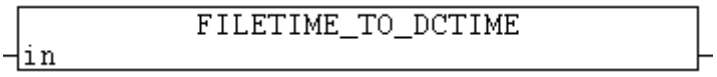
Example:

See Description of: F_GetCurDcTickTime [▶ 27].

Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.10.0 Build > 1310 oder höher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

4.6 FILETIME_TO_DCTIME



This function converts the 64 bit "Windows File Time" variable to the 64 bit "Distributed Clock System Time" variable. If conversion fails the return value is zero.

FUNCTION FILETIME_TO_DCTIME: T_DCTIME

T_DCTIME [▶ 85]

```
VAR_INPUT
    in : T_FILETIME;
END_VAR
```

in: The "Windows File Time" variable to be converted.

Example:

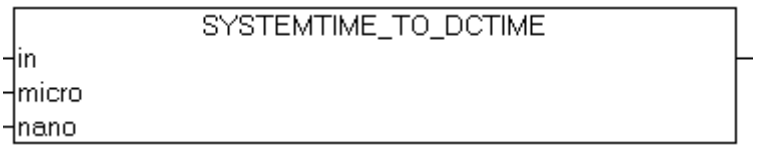
```
PROGRAM P_TEST
VAR
    fbSysFileTime    : GETSYSTEMTIME;
    ft               : T_FILETIME;
    dct              : T_DCTIME;
END_VAR
```

```
fbSysFileTime(timeLoDW=>ft.dwLowDateTime, timeHiDW=>ft.dwHighDateTime);
dct := FILETIME_TO_DCTIME(ft);
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 Build >= 1324 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

4.7 SYSTEMTIME_TO_DCTIME



This function converts the structured "Windows System Time" variable to the 64 bit "Distributed Clock System Time" variable. If conversion fails the return value is zero.

FUNCTION SYSTEMTIME_TO_DCTIME: T_DCTIME

T_DCTIME [▶ 85]

```
VAR_INPUT
    in    : TIMESTRUCT;
    micro : WORD(0..999); (* Microseconds: 0..999 *)
    nano  : WORD(0..999); (* Nanoseconds: 0..999 *)
END_VAR
```

in: The "Windows System Time" variable to be converted.

micro: Microseconds.

nano: Nanoseconds.

Example:

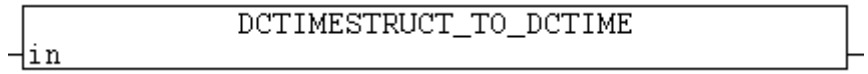
```
PROGRAM P_TEST
VAR
    syst : TIMESTRUCT := ( wYear := 2009, wMonth := 9, wDay := 16, wHour := 12, wMinute := 22, wSeco
nd := 44, wMilliseconds := 123 );
END_VAR
```

```
dct := SYSTEMTIME_TO_DCTIME( syst, 456, 789 );
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 Build > 1340 TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

4.8 DCTIMESTRUCT_TO_DCTIME



This function converts the structured variable of type: DCTIMESTRUCT [▶ 76] to the 64 bit "Distributed Clock System Time" variable.
The wDayOfWeek member of the structured variable is ignored. The wYear member must be greater or equal than 2000 and cannot be greater than 2585. If conversion fails the return value is zero.

FUNCTION DCTIMESTRUCT_TO_DCTIME: T_DCTIME

T_DCTIME [▶ 85]

```
VAR_INPUT
    in : DCTIMESTRUCT;
END_VAR
```

in: The variable [▶ 76] to be converted.

Example:

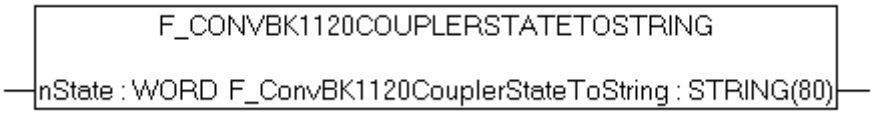
```
PROGRAM P_TEST
VAR
    dcStruct : DCTIMESTRUCT := (    wYear := 2008, wMonth := 3, wDay := 13,
                    wHour := 1, wMinute := 2, wSecond :=3,
                    wMilliseconds := 123, wMicroseconds := 456, wNanoseconds := 789 );
    dc64    : T_DCTIME;
END_VAR
```

```
dc64 := DCTIMESTRUCT_TO_DCTIME( dcStruct );
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 Build >= 1324 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

4.9 F_ConvBK1120CouplerStateToString



The function F_ConvBK1120CouplerStateToString supplies the coupler status of the BK1120/BK1250 as a string.

VAR_INPUT

```
VAR_INPUT
    nState : WORD;
END_VAR
```

nState: Coupler status (CouplerState) can be linked in the System Manager from the inputs of the BK1120/ BK1250 to the PLC.

```
    0x0000 = No error
    0x0001 = K-Bus error
    0x0002 = Configuration error
    0x0010 = Outputs disabled
    0x0020 = K-Bus overrun
    0x0040 = Communication error (Inputs)
    0x0080 = Communication error (Outputs)
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

4.10 F_ConvMasterDevStateToString

The function F_ConvMasterDevStateToString supplies the drive status of the EtherCAT Master as a string.

VAR_INPUT

```
VAR_INPUT
    nState : WORD;
END_VAR
```

nstate: Device status of the EtherCAT Master can be linked as DevState in the System Manager from the inputs of the EtherCAT Master to the PLC

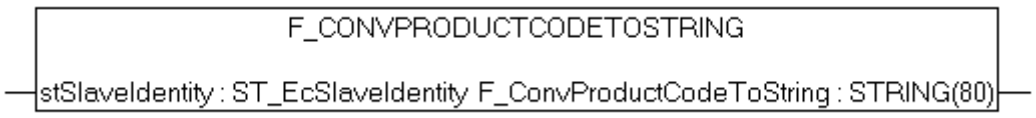
```
    0x0001 = Link error
    0x0002 = I/O locked after link error (I/O reset
required)
```

```
    0x0004 = Link error (redundancy adapter)
    0x0008 = Missing one frame (redundancy
mode)
    0x0010 = Out of send resources (I/O reset
required)
    0x0020 = Watchdog triggered
    0x0040 = Ethernet driver (miniport) not
found
    0x0080 = I/O reset active
    0x0100 = At least one device in 'INIT'
state
    0x0200 = At least one device in 'PRE-OP'
state
    0x0400 = At least one device in
'SAFE-OP' state
    0x0800 = At least one device indicates an error
state
    0x1000 = DC not in sync
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

4.11 F_ConvProductCodeToString



The function F_ConvProductToString supplies the Product Code as a string, i.e. 'EL6731-0000-0017'.

VAR_INPUT

```
VAR_INPUT
    stSlaveIdentity : ST_EcSlaveIdentity;
END_VAR
```

stSlaveIdentity: The SlaveIdentity can be read with FB_EcGetSlaveIdentity [▶ 45].

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

4.12 F_ConvSlaveStateToString

The function F_ConvSlaveStateToString supplies the EtherCAT slave state as a string.

VAR_INPUT

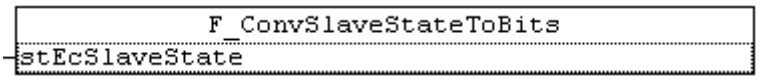
```
VAR_INPUT
    state : ST_EcSlaveState;
END_VAR
```

state: EtherCAT slave state structure (deviceState : BYTE; linkState : BYTE;)

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

4.13 F_ConvSlaveStateToString



The function F_ConvSlaveStateToString supplies the EtherCAT slave state as a structure TYPE ST_EcSlaveStateBits [▶ 83].

VAR_INPUT

```
VAR_INPUT
    stEcSlaveState : ST_EcSlaveState;
END_VAR
```

stEcSlaveState: EtherCAT slave state structure (deviceState : BYTE; linkState : BYTE;)

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

4.14 F_ConvStateToString

The function F_ConvStateToString supplies the EtherCAT slave state as a string.

VAR_INPUT

```
VAR_INPUT
    nState : WORD;
END_VAR
```

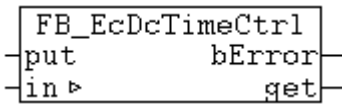
Name	Type	Description
nState	WORD	EtherCAT Slave State as WORD 0x___1 = Slave in'INIT' state 0x___2 = Slave in 'PREOP' state 0x___3 = Slave in 'BOOT' state 0x___4 = Slave in 'SAFEOP' state

Name	Type	Description
		0x___8 = Slave in 'OP' state 0x001_ = Slave signals error 0x002_ = Invalid vendorId, productCode...read 0x004_ = Initialization error occurred 0x010_ = Slave not present 0x020_ = Slave signals link error 0x040_ = Slave signals missing link 0x080_ = Slave signals unexpected link 0x100_ = Communication port A 0x200_ = Communication port B 0x400_ = Communication port C 0x800_ = Communication port D

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

4.15 FB_EcDcTimeCtrl



With this function block single components like year, month, day etc. of a 64 bit TwinCAT "Distributed Clock System Time" variable can be read. The function block contains several A_GetXYZ actions. After calling the wanted action the value of the XYZ component in the *get*-output variable is available. The put input parameter is not yet used.

The function block contains the following actions:

- A_GetYear;
- A_GetMonth;
- A_GetDay;
- A_GetDayOfWeek;
- A_GetHour;
- A_GetMinute;
- A_GetSecond;
- A_GetMilli;
- A_GetMicro;
- A_GetNano;

VAR_IN_OUT

```
VAR_IN_OUT
    in          : T_DCTIME;
END_VAR
```

in: TwinCAT "Distributed Clock System Time [▶ 85]" variable;

VAR_INPUT

```
VAR_INPUT
    put         : WORD;
END_VAR
```

put: Input parameter (not used);

VAR_OUTPUT

```
VAR_OUTPUT
    bError      : BOOL;
    get         : WORD;
END_VAR
```

bError: This output is set, if an error occurs at the action call;

get: Output parameter (year, month, day, etc.);

Example of an implementation in ST:

```
PROGRAM P_TEST
VAR
    dcStruct    : DCTIMESTRUCT;
    dcTime      : T_DCTIME;
    fbCtrl      : FB_EcDcTimeCtrl;

    wYear       : WORD;
    wMonth      : WORD;
    wDay        : WORD;
    wDayOfWeek  : WORD;
    wHour       : WORD;
    wMinute     : WORD;
    wSecond     : WORD;
    wMilli      : WORD;
    wMicro      : WORD;
    wNano       : WORD;
END_VAR
```

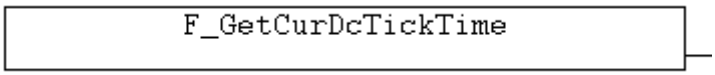
```
dcTime := F_GetCurDcTickTime();

fbCtrl.A_GetYear( in := dcTime, get => wYear );
fbCtrl.A_GetMonth( in := dcTime, get => wMonth );
fbCtrl.A_GetDay( in := dcTime, get => wDay );
fbCtrl.A_GetDayOfWeek( in := dcTime, get => wDayOfWeek );
fbCtrl.A_GetHour( in := dcTime, get => wHour );
fbCtrl.A_GetMinute( in := dcTime, get => wMinute );
fbCtrl.A_GetSecond( in := dcTime, get => wSecond );
fbCtrl.A_GetMilli( in := dcTime, get => wMilli );
fbCtrl.A_GetMicro( in := dcTime, get => wMicro );
fbCtrl.A_GetNano( in := dcTime, get => wNano );
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 Build >= 1316 or higher	PC or CX (x86, ARM)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
	CX (ARM)	

5 Distributed Clocks

5.1 F_GetCurDcTickTime



This function supplies the time of the current (last) tick in TwinCAT Distributed Clock time format.

FUNCTION F_GetCurDcTickTime : T_DCTIME

T_DCTIME [▶ 85]

```
VAR_INPUT
(*none*)
END_VAR
```

Example in ST:

```
PROGRAM MAIN
VAR
    tDC : T_DCTIME;
    sDC : STRING;
    tDCBack : T_DCTIME;

    sDCZero : STRING;(* DCTIME = zero time starts on 01.01.2000 *)
    tDCBackFromZero : T_DCTIME;

    tDCFromString : T_DCTIME;
    sDCBackFromString : STRING;
END_VAR
```

```
tDC := F_GetCurDcTickTime();
sDC := DCTIME_TO_STRING( tDC );
tDCBack := STRING_TO_DCTIME( sDC );

sDCZero := DCTIME_TO_STRING( ULARGE_INTEGER(0, 0) );
tDCBackFromZero := STRING_TO_DCTIME( sDCZero );

tDCFromString := STRING_TO_DCTIME( '2007-03-09-11:31:09.223456789' );
sDCBackFromString := DCTIME_TO_STRING( tDCFromString );
```

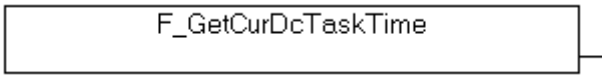
Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.10.0 Build > 1310 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

Also see about this

- DCTIME_TO_STRING [▶ 17]
- STRING_TO_DCTIME [▶ 19]

5.2 F_GetCurDcTaskTime



This function supplies the task target start time (in TwinCAT Distributed Clock system time format. The returned time is the target time of the task where the function is called.

FUNCTION F_GetCurDcTaskTime : T_DCTIME

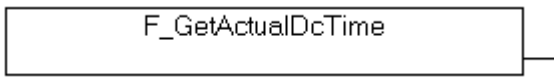
T_DCTIME [▶ 85]

```
VAR_INPUT
(*none*)
END_VAR
```

Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.11.0 Build >= 1524 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

5.3 F_GetActualDcTime



This function supplies the current time in TwinCAT Distributed Clock time format.

FUNCTION F_GetActualDcTime: T_DCTIME

T_DCTIME [▶ 85]

```
VAR_INPUT
(*none*)
END_VAR
```

Example in ST:

```
PROGRAM MAIN
VAR
    actDC : T_DCTIME;
    sAct : STRING;
END_VAR
```

```
actDC := F_GetActualDcTime();
sAct := DCTIME_TO_STRING( actDC );
```

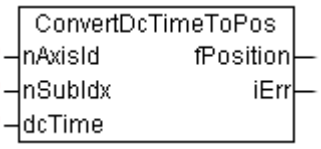
Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.11.0 Build > 1535 or higher	PC or CX (x86, ARM) PLC runtime: v2.11 Build > 1535	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

Also see about this

DCTIME_TO_STRING [▶ 17]

5.4 ConvertDcTimeToPos



This function block converts a 32 bit *Distributed Clock System Time* into an associated NC axis position (i.e. the axis position that was – or will be – present at exactly that time).

VAR_INPUT

```
VAR_INPUT
    nAxisId : UDINT;
    nSubIdx : UDINT;
    dcTime  : T_DCTIME32;(* 32 bit distributed clock time *)
END_VAR
```

nAxisId	ID of the NC axis.
nSubIdx	This 32-bit input value contains two different items of information and is therefore divided into two 16-bit values: The LowWord (the least significant 16 bits) contains the subindex for relative addressing of an encoder subelement at an axis. The subindex is counted upwards from zero. For the typical case of an axis that has just one encoder, the null subindex is correct. The HighWord (the most significant 16 bits) contains a control word (bit mask) that affects the way in which the position is calculated (e.g. the type of interpolation or extrapolation). A bit mask value of 0x0001 means that the set acceleration of the axis is to be included in the calculation.
dcTime	32-bit distributed clock system time (T_DCTIME32 [▶ 85]). This input value is converted into the corresponding NC axis position. **This 32-bit time may only be used in the close time range of ± 2.147 seconds around the current system time, as it is only unique here. This requirement cannot be checked within the FB.**

VAR_OUTPUT

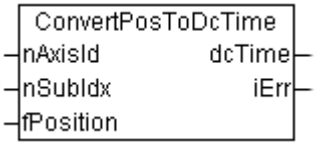
```
VAR_OUTPUT
    fPosition   : LREAL;
    iErr        : UDINT;
END_VAR
```

fPosition	Supplies the NC axis position corresponding to *dcTime*. This is an NC axis position that has been scaled and provided with an offset, having, for instance, physical units of degrees or of millimetres.
iErr	Returns the error number if an error occurs, e.g. error 0x4012 (axis ID is not allowed, or axis does not exist within the system).

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.11.0 Build >= 1524 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

5.5 ConvertPosToDcTime



This function block converts an NC axis position into an associated 32-bit *Distributed Clock System Time* (i.e. the time at which this NC axis position was or will be reached exactly).

VAR_INPUT

```
VAR_INPUT
    nAxisId         : UDINT;
    nSubIdx         : UDINT;
    fPosition       : LREAL;
END_VAR
```

nAxisId	ID of the NC axis.
nSubIdx	This 32 bit input magnitude is composed of two different items of information, and is divided into two 16-bit values: The low word (the 16 bits with the lowest value) contains the sub-index for relative addressing of an encoder sub-element at an axis. The sub-index is counted upwards from zero. For the typical case of an axis that has just one encoder, the null sub-index is correct. The high word (the 16 bits with the highest value) contains a control word (bit mask) that affects the way in which the position is calculated (e.g. the type of interpolation or extrapolation). A bit mask value of 0x0001 means that the set acceleration of the axis is to be included in the calculation.
fPosition	NC axis position that will be converted through calculation into the corresponding 32 bit distributed clock system time (T_DCTIME32 [▶ 85]). If the DC system time associated with the position is outside the expected time window of ± 2.147 seconds, the conversion is refused and an error number returned.

VAR_OUTPUT

```
VAR_OUTPUT
    dcTime   : T_DCTIME32;(* 32 bit distributed clock time *)
    iErr     : UDINT;
END_VAR
```

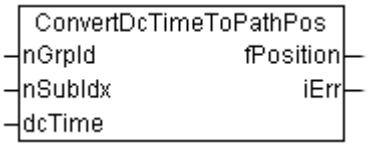
dcTime	Supplies the 32 bit Distributed Clock System Time (T_DCTIME32 [▶ 85]) associated with the input *fPosition*.
iErr	Supplies an error number if an error occurs, e.g. - error 0x4012: axis ID is not allowed, or axis is not present in the system, - error 0x4361: time window exceeded (future), - error 0x4362: time window exceeded (past), - error 0x4363: position cannot be found mathematically.

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.11.0 Build >= 1524 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

5.6 ConvertDcTimeToPathPos

Under construction...



This function block converts a 32 bit *Distributed Clock System Time* into an associated NC axis position (i.e. the axis position that was – or will be – present at exactly that time).

VAR_INPUT

```
VAR_INPUT
    nGrpId      : UDINT;
    nSubIdx     : UDINT;
    dcTime      : T_DCTIME32;(* 32 bit distributed clock time *)
END_VAR
```

nGrpId	Group ID of the corresponding Nci channel
nSubIdx	This 32-bit input value contains two different items of information and is therefore divided into two 16-bit values: The LowWord (the least significant 16 bits) contains the subindex for relative addressing of an encoder subelement at an axis. The subindex is counted upwards from zero. For the typical case of an axis with exactly one encoder the null subindex is correct. The HighWord (the most significant 16 bits) contains a control word (bit mask) that influences the type of position calculation (e.g. the interpolation or extrapolation type). The bit mask 0x0001 means that the set acceleration of the axis is to be included in the calculation. **Bit mask 0x0010 means that the calculation is relative and is currently mandatory. Otherwise the call is rejected with an error.**
dcTime	32-bit distributed clock system time (T_DCTIME32 [▶ 85]). This input value is converted to the corresponding relative Nci path distance on the contour. **This 32-bit time may only be used in the close time range of ± 2,147 seconds around the current system time, as it is only unique here. This requirement cannot be checked within the FB.**

VAR_OUTPUT

```
VAR_OUTPUT
    fPosition   : LREAL;
    iErr        : UDINT;
END_VAR
```

fPosition	Supplies the NC axis position corresponding to *dcTime*. This is an NC axis position that has been scaled and provided with an offset, having, for instance, physical units of degrees or of millimetres.
iErr	Returns the error number if an error occurs, e.g. error 0x4012 (axis ID is not allowed, or axis does not exist within the system).

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.11.0 Build > 2214 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

5.7 ConvertPathPosToDcTime

Under construction...

This function block converts an NC axis position into an associated 32-bit *Distributed Clock System Time* (i.e. the time at which this NC axis position was or will be reached exactly).

VAR_INPUT

```
VAR_INPUT
    nGrpId      : UDINT;
    nSubIdx     : UDINT;
    fPosition   : LREAL;
END_VAR
```

nGrpId	Group ID of the corresponding Nci channel
nSubIdx	This 32-bit input value contains two different items of information and is therefore divided into two 16-bit values: The LowWord (the least significant 16 bits) contains the subindex for relative addressing of an encoder subelement at an axis. The subindex is counted upwards from zero. For the typical case of an axis with exactly one encoder the null subindex is correct. The HighWord (the most significant 16 bits) contains a control word (bit mask) that influences the type of position calculation (e.g. the interpolation or extrapolation type). The bit mask 0x0001 means that the set acceleration of the axis is to be included in the calculation. **Bit mask 0x0010 means that the calculation is relative and is currently mandatory. Otherwise the call is rejected with an error.**
fPosition	Relative Nci path distance that is converted to the corresponding 32-bit distributed clock system time (T_DCTIME32 [▶ 85]). If the DC system time associated with the relative Nci path distance is outside the expected timeframe of ± 2.147 seconds, this conversion is rejected with an error number.

VAR_OUTPUT

```
VAR_OUTPUT
    dcTime  : T_DCTIME32;(* 32 bit distributed clock time *)
    iErr    : UDINT;
END_VAR
```

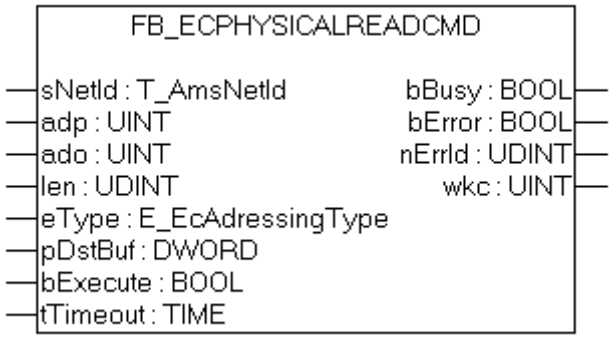
dcTime	Supplies the 32 bit Distributed Clock System Time (T_DCTIME32 [▶ 85]) associated with the input *fPosition*.
iErr	Supplies an error number if an error occurs, e.g. - error 0x4012: axis ID is not allowed, or axis is not present in the system, - error 0x4361: time window exceeded (future), - error 0x4362: time window exceeded (past), - error 0x4363: position cannot be found mathematically.

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.11.0 Build > 2214 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

6 EtherCAT Commands

6.1 FB_EcPhysicalReadCmd



The function block FB_EcPhysicalReadCmd sends an EtherCAT read command (FPRD, APRD, BRD) to a specific slave or to all EtherCAT slaves(Broadcast). This command can be sent by the PLC to read out a register or the DPRAM of an EtherCAT Slave Controller.

VAR_INPUT

```
VAR_INPUT
    sNetId   : T_AmsNetId;
    adp      : UINT;
    ado      : UINT;
    len      : UDINT;
    eType    : E_EcAdressingType := eAdressingType_Fixed;
    pDstBuf  : DWORD;
    bExecute : BOOL;
    tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

adp: This value defines which EtherCAT slave is addressed by this command. Which slave is addressed by this value depends on the addressing type defined by eType:

eType	Description
eAdressingType_Fixed	The slave is addressed by its configured EtherCAT address. The EtherCAT addresses of all slaves can be read out with FB_EcGetAllSlaveAddr.
eAdressingType_AutoInc	The slave is addressed by its position in the bus. The first slave has an address of 0(adp=0) and the adp of all following slaves is decremented by one. 1.Slave adp = 0 2.Slave adp = 16#ffff (-1) 3.Slave adp = 16#fffe(-2) 4.Slave adp = 16#fffd(-3) etc.
eAdressingType_BroadCAST	All slaves are addressed by this command. The value adp can be set to 0.

ado: Physical memory(DPRAM) or register one wants to read out.

len: Count of bytes to read.

eType: Depending on the value of eType different EtherCAT commands are sent:

eType	Command
eAdressingType_Fixed	Configured Address Physical Read (FPRD)
eAdressingType_AutoInc	Auto Increment Physical Read (APRD)
eAdressingType_BroadCAST	Broadcast Read (BRD)

The individual commands only differ in how the slaves are addressed (see adp).

pDestBuf: The address (pointer) of the receive buffer.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy    : BOOL;
    bError   : BOOL;
    nErrId   : UDINT;
    wkc      : UINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

wkc: The working counter is incremented by 1 for each EtherCAT slave that is addressed by this command. If only on EherCAT slave is addressed by this command, the working counter will be 1.

Sample for an implementation in ST:

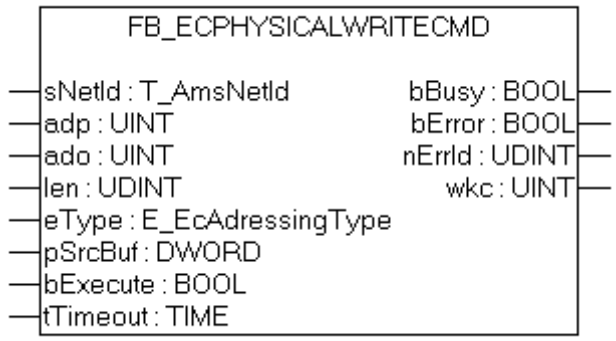
```
PROGRAM TEST_PhysicalReadCmd
VAR
    fbReadCmd       : FB_EcPhysicalReadCmd;
    bExecute    : BOOL;
    value    : UINT;
    adp      : UINT:=16#3E9;
    ado      : UINT:=16#1100;
    eType    : E_EcAdressingType := eAdressingType_Fixed;
    sNetId   : T_AmsNetId:='192.168.1.5.3.1';

    wkc      : UINT;
    bError   : BOOL;
    nErrId   : UDINT;
END_VAR
```

```
fbReadCmd (sNetId:=sNetID, ado:=ado, adp:=adp, eType:=eType, LEN := SIZEOF(value), pDstBuf:=ADR(valu
e), bExecute:=bExecute);
wkc := fbReadCmd.wkc;
bError:=fbReadCmd.bError;
nErrId:=fbReadCmd.nErrId;
```

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0 Build >= 1314 or higher	PC or CX (x86)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1314 or higher	CX (ARM)	

6.2 FB_EcPhysicalWriteCmd



The function block FB_EcPhysicalWriteCmd sends an EtherCAT write command (FPWR, APWR, BWR) to a specific slave or to all EtherCAT slaves (Broadcast). This command can be sent by the PLC to write to a register or the DPRAM of an EtherCAT Slave Controller.

VAR_INPUT

```
VAR_INPUT
    sNetId   : T_AmsNetId;
    adp      : UINT;
    ado      : UINT;
    len      : UDINT;
    eType    : E_EcAdressingType := eAdressingType_Fixed;
    pSrcBuf  : DWORD;
    bExecute : BOOL;
    tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

adp: This value defines which EtherCAT slave is addressed by this command. Which slave is addressed by this value depends on the addressing type defined by eType:

eType	Description
eAdressingType_Fixed	The slave is addressed by its configured EtherCAT address. The EtherCAT addresses of all slaves can be read out with FB_EcGetAllSlaveAddr.
eAdressingType_AutoInc	The slave is addressed by its position in the bus. The first slave has an address of 0(adp=0) and the adp of all following slaves is decremented by one. 1.Slave adp = 0 2.Slave adp = 16#ffff (-1) 3.Slave adp = 16#fffe(-2) 4.Slave adp = 16#fffd(-3) etc.
eAdressingType_BroadCAST	All slaves are addressed by this command. The value adp can be set to 0.

ado: Physical memory(DPRAM) or register one wants to write to.

len: Count of bytes to write.

eType: Depending on the value of eType different EtherCAT commands are sent:

eType	Command
eAdressingType_Fixed	Configured Address Physical Read (FPRD)
eAdressingType_AutoInc	Auto Increment Physical Read (APRD)

eType	Command
eAdressingType_BroadCAST	Broadcast Read (BRD)

The individual commands only differ in how the slaves are addressed (see adp).

pSrcBuf: The address (pointer) of the source buffer.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy    : BOOL;
    bError   : BOOL;
    nErrId   : UDINT;
    wkc      : UINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

wkc: The working counter is incremented by 1 for each EtherCAT slave that is addressed by this command. If only on EherCAT slave is addressed by this command, the working counter will be 1.

Sample for an implementation in ST:

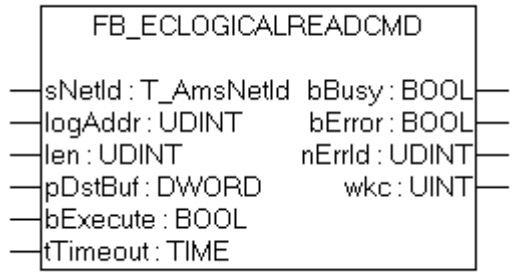
```
PROGRAM Test_PhysicalWriteCmd
VAR
    fbWriteCmd  : FB_EcPhysicalWriteCmd;
    bExecute    : BOOL;
    value       : UINT :=16#5555;
    adp         : UINT:=16#3E9;
    ado         : UINT:=16#1100;
    eType       : E_EcAdressingType := eAdressingType_Fixed;
    sNetId      : T_AmsNetId:='192.168.1.5.3.1';

    wkc         : UINT;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

```
fbWriteCmd (sNetId:=sNetID, ado:=ado, adp:=adp, eType:=eType, LEN := SIZEOF(value), pSrcBuf:=ADR(val
ue), bExecute:=bExecute);
wkc := fbWriteCmd.wkc;
bError:=fbWriteCmd.bError;
nErrId:=fbWriteCmd.nErrId;
```

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0 Build >= 1314 or higher	PC or CX (x86)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1314 or higher	CX (ARM)	

6.3 FB_EcLogicalReadCmd



The function block FB_EcLogicalReadCmd sends a logical read EtherCAT command(LRD). In every slave one can map local address areas (DPRAM of the EtherCAT Slave Controller) to global logical address areas. Therefore, this command addresses all EtherCAT slaves, that have a mapping configured for the selected logical address area.

VAR_INPUT

```
VAR_INPUT
    sNetId   : T_AmsNetId;
    logAddr  : UDINT;
    len      : UDINT;
    pDstBuf  : DWORD;
    bExecute : BOOL;
    tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

logAddr: Logical address.

len: Count of bytes to read.

pDestBuf: The address (pointer) of the receive buffer.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    wkc     : UINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

wkc: The working counter is incremented by 1 for each EtherCAT slave that is addressed by this command. If only on EherCAT slave is addressed by this command, the working counter will be 1.

Sample for an implementation in ST:

```
PROGRAM Test_LogicalReadCmd
VAR
    fbReadCmd   : FB_EcLogicalReadCmd;
    bExecute    : BOOL;
    value       : USINT;
```

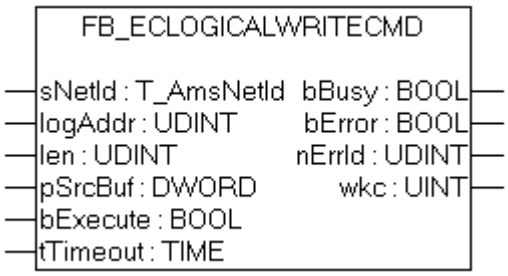
```
    logAddr     : UDINT :=16#10000;
    sNetId      : T_AmsNetId:='192.168.1.5.3.1';

    wkc         : UINT;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

```
fbReadCmd (sNetId:=sNetID, logAddr:=logAddr, LEN := SIZEOF(value), pDstBuf:=ADR(value), bExecute:=bE
xecute);
wkc := fbReadCmd.wkc;
bError:=fbReadCmd.bError;
nErrId:=fbReadCmd.nErrId;
```

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0 Build >= 1314 or higher	PC or CX (x86)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1314 or higher	CX (ARM)	

6.4 FB_EcLogicalWriteCmd



The function block FB_EcLogicalWriteCmd sends a logical write EtherCAT command (LWR). In every slave one can map local address areas (DPRAM of the EtherCAT Slave Controller) to global logical address areas. Therefore, this command addresses all EtherCAT slaves, that have a mapping configured for the selected logical address area.

VAR_INPUT

```
VAR_INPUT
    sNetId    : T_AmsNetId;
    logAddr   : UDINT;
    len       : UDINT;
    pSrcBuf   : DWORD;
    bExecute  : BOOL;
    tTimeout  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

logAddr: Logical address.

len: Count of bytes to write.

pSrcBuf: The address (pointer) of the source buffer.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    wkc     : UINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

wkc: The working counter is incremented by 1 for each EtherCAT slave that is addressed by this command. If only on EherCAT slave is addressed by this command, the working counter will be 1.

Sample for an implementation in ST:

```
PROGRAM Test_LogicalWriteCmd
VAR
    fbWriteCmd  : FB_EcLogicalWriteCmd;
    bExecute    : BOOL;
    value       : USINT :=16#55;
    logAddr     : UDINT :=16#10000;
    sNetId      : T_AmsNetId:='192.168.1.5.3.1';

    wkc         : UINT;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

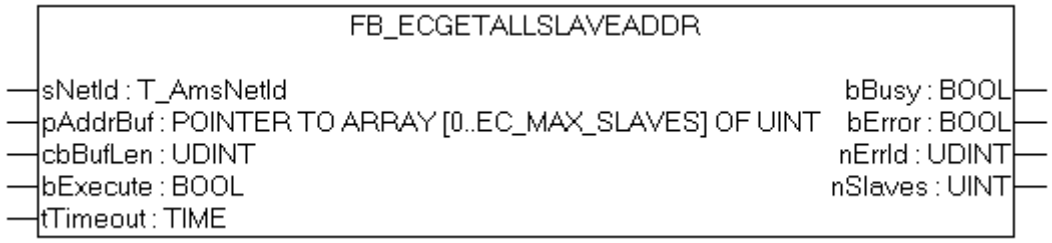
```
fbWriteCmd (sNetId:=sNetID, logAddr:=logAddr, LEN := SIZEOF(value), pSrcBuf:=ADR(value), bExecute:=b
Execute);
wkc := fbWriteCmd.wkc;
bError:=fbWriteCmd.bError;
nErrId:=fbWriteCmd.nErrId;
```

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0 Build >= 1314 or higher	PC or CX (x86)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1314 or higher	CX (ARM)	

7 EtherCAT Diagnostic

7.1 FB_EcGetAllSlaveAddr



The FB_EcGetAllSlaveAddr function block allows the addresses of all the slaves connected to the master to be read. When the call is successful, the buffer passed in the parameter pAddrBuf contains the addresses of all the slaves as an array of UINTs.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    pAddrBuf    : POINTER TO ARRAY[0..EC_MAX_SLAVES] OF UINT;
    cbBufLen    : UDINT
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

pAddrBuf: The address of an array of UINTs into which the addresses of the individual slaves are to be written.

cbBufLen: The maximum available buffer size for the data to be read, in bytes.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    nSlaves     : UINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

nSlaves: The number of slaves connected to the master.

Example of an implementation in ST:

```
PROGRAM TEST_GetAllSlaveAddresses
VAR
    fbGetAllSlaveAddr       : FB_EcGetAllSlaveAddr;
    sNetId          : T_AmsNetId := '172.16.2.131.2.1';
    bExecute        : BOOL;
```

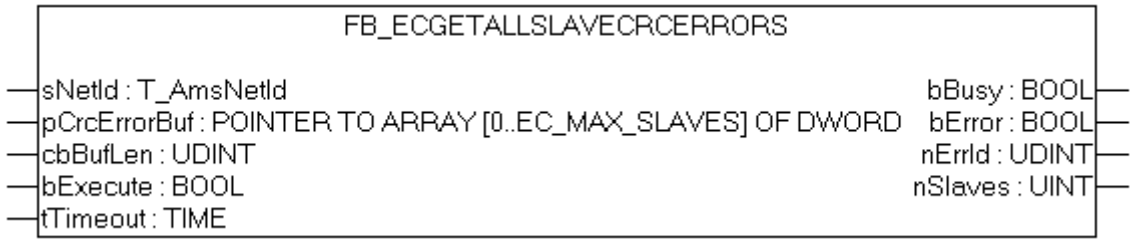
```
    slaveAddresses      : ARRAY[0..255] OF UINT;
    nSlaves             : UINT := 0;
    bError          : BOOL;
    nErrId          : UDINT;
END_VAR
```

```
fbGetAllSlaveAddr(sNetId:= sNetId,pAddrBuf := ADR(slaveAddresses), cbBufLen:= SIZEOF(slaveAddresses)
, bExecute:=bExecute);
nSlaves:=fbGetAllSlaveAddr.nSlaves;
bError:=fbGetAllSlaveAddr.bError;
nErrId:=fbGetAllSlaveAddr.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

7.2 FB_EcGetAllCrcErrors



The FB_EcGetAllSlaveCrcErrors function block allows the CRC error counters of all the slaves connected to the master to be read. The CRC errors of the individual ports of a slave are added up. To read the CRC errors of the individual ports (A, B and C) of a slave, it is necessary to call the FB_EcGetSlaveCrcError [▶ 42] function block.

VAR_INPUT

```
VAR_INPUT
    sNetId          : T_AmsNetId;
    pCrcErrorBuf    : POINTER TO ARRAY[0..EC_MAX_SLAVES] OF DWORD;
    cbBufLen        : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

pCrcErrorBuf : The address of an array of DWORDs into which the CRC error counter is to be written.

cbBufLen: The maximum available buffer size for the data to be read, in bytes.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    nSlaves     : UINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

nSlaves: The number of slaves connected to the master.

Example of an implementation in ST:

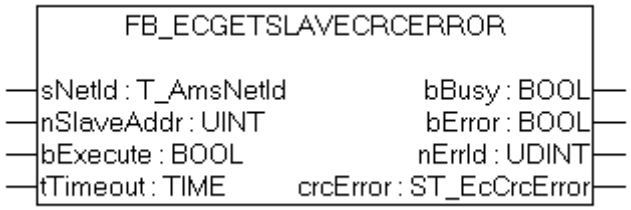
```
PROGRAM TEST_GetAllSlaveCrcErrors
VAR
    fbGetAllSlaveCrcErrors  : FB_EcGetAllSlaveCrcErrors;
    sNetId          : T_AmsNetId := '172.16.2.131.2.1';
    bExecute        : BOOL;
    crcErrors       : ARRAY[0..255] OF DWORD;
    nSlaves         : UINT := 0;
    bError          : BOOL;
    nErrId          : UDINT;
END_VAR
```

```
fbGetAllSlaveCrcErrors(sNetId:= sNetId, pCrcErrorBuf := ADR(crcErrors), cbBufLen:= SIZEOF(crcErrors)
, bExecute:=bExecute);
nSlaves:=fbGetAllSlaveCrcErrors.nSlaves;
bError:=fbGetAllSlaveCrcErrors.bError;
nErrId:=fbGetAllSlaveCrcErrors.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

7.3 FB_EcGetSlaveCrcError



The function block FB_EcGetSlaveCrcError allows the CRC error counters of the individual ports (A, B and C) of a slave to be read. If the call is successful, the output variable *crcError*, whose type is ST_EcCrcError, contains the requested CRC error counter.

The function block FB_EcGetSlaveCrcError can only be used with slaves with up to 3 ports (i.e. EK1100), the function block FB_EcGetSlaveCrcErrorEx can also be used with slaves with up to 4 ports (i.e. EK1122).

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave, whose CRC error counter should be read.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    crcError    : ST_EcCrcError;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

crcError: CRC error [▶ 78] counter of all ports.

Example of an implementation in ST:

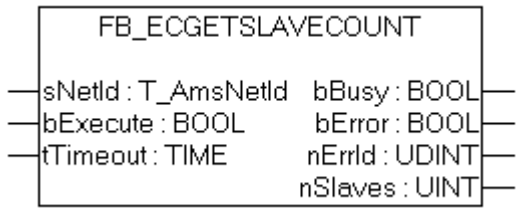
```
PROGRAM TEST_GetSlaveCrcError
VAR
    fbGetSlaveCrcError      : FB_EcGetSlaveCrcError;
    sNetId          : T_AmsNetId := '172.16.2.131.2.1';
    bExecute        : BOOL;
    crcError        : ST_EcCrcError;
    nSlaveAddr          : UINT := 1001;
    bError          : BOOL;
    nErrId          : UDINT;
END_VAR
```

```
fbGetSlaveCrcError(sNetId:= sNetId, nSlaveAddr:= nSlaveAddr, bExecute:=bExecute);
crcError:=fbGetSlaveCrcError.crcError;
bError:=fbGetSlaveCrcError.bError;
nErrId:=fbGetSlaveCrcError.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

7.4 FB_EcGetSlaveCount



The function block FB_EcGetSlaveCount can be used to determine the number of slaves that are connected to the master.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    nSlaves     : UINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

nSlaves: The number of slaves that are connected to the master.

Example of an implementation in ST:

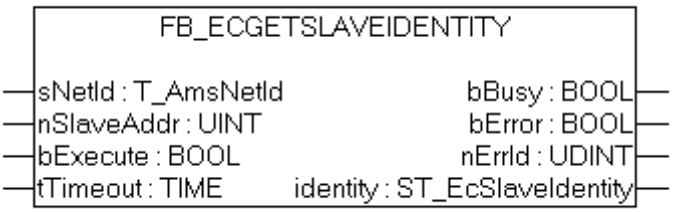
```
PROGRAM TEST_GetSlaveCount
VAR
    fbGetSlaveCount : FB_EcGetSlaveCount;
    sNetId          : T_AmsNetId := '172.16.2.131.2.1';
    bExecute        : BOOL;
    nSlaves         : UINT;
    bError          : BOOL;
    nErrId          : UDINT;
END_VAR
```

```
fbGetSlaveCount(sNetId:= sNetId, bExecute:=bExecute);
nSlaves:=fbGetSlaveCount.nSlaves;
bError:=fbGetSlaveCount.bError;
nErrId:=fbGetSlaveCount.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

7.5 FB_EcGetSlaveIdentity



The function block FB_EcGetSlaveIdentity allows the CANopen identity of an individual EtherCAT slave to be read. If the call is successful, the output variable *identity*, whose type is ST_EcSlaveIdentity, contains the requested identity information.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave whose CRC error counter is to be read

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    identity    : ST_EcSlaveIdentity;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

identity: CANopen Identity [▶ 80] object.

Example of an implementation in ST:

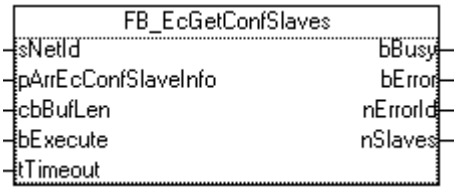
```
PROGRAM TEST_GetSlaveIdentity
VAR
    fbGetSlaveIdentity      : FB_EcGetSlaveIdentity;
    sNetId          : T_AmsNetId := '172.16.2.131.2.1';
    bExecute        : BOOL;
    identity        : ST_EcSlaveIdentity;
    nSlaveAddr          : UINT := 1001;
    bError          : BOOL;
    nErrId          : UDINT;
END_VAR
```

```
fbGetSlaveIdentity(sNetId:= sNetId, nSlaveAddr:= nSlaveAddr, bExecute:=bExecute);
identity:=fbGetSlaveIdentity.identity;
bError:=fbGetSlaveIdentity.bError;
nErrId:=fbGetSlaveIdentity.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

7.6 FB_EcGetConfSlaves



Function Block FB_EcGetConfSlaves generates a List of all configured Slaves from the Master object directory.

VAR_INPUT

```
VAR_INPUT
    sNetId              : T_AmsNetId;
    pArrEcConfSlaveInfo : POINTER TO ARRAY[0..EC_MAX_SLAVES] OF ST_EcSlaveConfigData;
    cbBufLen            : UDINT;
    bExecute            : BOOL;
    tTimeout            : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

pArrEcConfSlaveInfo : The address (pointer) of the receive buffer. Receive buffer is a array of struct with type ST_EcSlaveConfigData [▶ 79] with the data of each configured slave.

cbBufLen: The maximum available buffer size for the data to be read, in bytes.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    nSlaves     : UINT;
ND_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

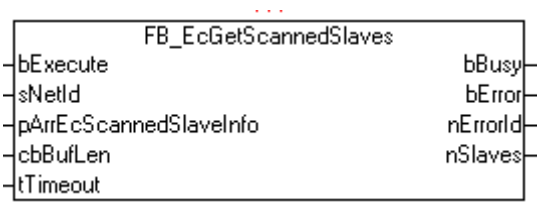
nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

nSlaves: Returns the number of configured slaves.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

7.7 FB_EcGetScannedSlaves



Function Block FB_EcGetConfSlaves generates a List of all scanned Slaves from the Master object directory. A scan of the EtherCAT network is automatically generated inside the FB.

VAR_INPUT

```
VAR_INPUT
    bExecute            : BOOL;
    sNetId              : T_AmsNetId;
    pArrEcConfSlaveInfo : POINTER TO ARRAY[0..EC_MAX_SLAVES] OF ST_EcSlaveScannedData;
    cbBufLen            : UDINT;
    tTimeout            : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

bExecute: The block is activated by a rising edge at this input.

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

pArrEcConfSlaveInfo : The address (pointer) of the receive buffer. Receive buffer is a array of struct with type ST_EcSlaveScannedData [▶ 81] with the data of each scanned slave.

cbBufLen: The maximum available buffer size for the data to be read, in bytes.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    nSlaves     : UINT;
ND_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

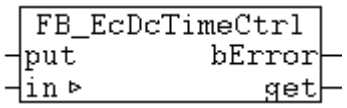
nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

nSlaves: Returns the number of configured slaves.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

7.8 FB_EcDcTimeCtrl



With this function block single components like year, month, day etc. of a 64 bit TwinCAT "Distributed Clock System Time" variable can be read. The function block contains several A_GetXYZ actions. After calling the wanted action the value of the XYZ component in the *get*-output variable is available. The put input parameter is not yet used.

The function block contains the following actions:

- A_GetYear;
- A_GetMonth;
- A_GetDay;
- A_GetDayOfWeek;
- A_GetHour;
- A_GetMinute;
- A_GetSecond;
- A_GetMilli;
- A_GetMicro;
- A_GetNano;

VAR_IN_OUT

```
VAR_IN_OUT
    in          : T_DCTIME;
END_VAR
```

in: TwinCAT "Distributed Clock System Time [▶ 85]" variable.

VAR_INPUT

```
VAR_INPUT
    put         : WORD;
END_VAR
```

put: Input parameter (not used);

VAR_OUTPUT

```
VAR_OUTPUT
    bError      : BOOL;
    get         : WORD;
END_VAR
```

bError: This output is set, if an error occurs at the action call;

get: Output parameter (year, month, day, etc.);

Example of an implementation in ST:

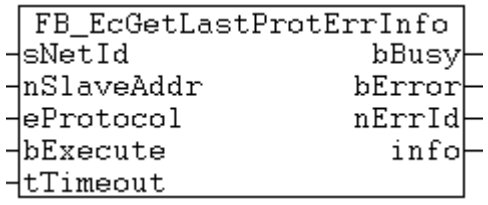
```
PROGRAM P_TEST
VAR
    dcStruct    : DCTIMESTRUCT;
    dcTime      : T_DCTIME;
    fbCtrl      : FB_EcDcTimeCtrl;

    wYear       : WORD;
    wMonth      : WORD;
    wDay        : WORD;
    wDayOfWeek  : WORD;
    wHour       : WORD;
    wMinute     : WORD;
    wSecond     : WORD;
    wMilli      : WORD;
    wMicro      : WORD;
    wNano       : WORD;
END_VAR
```

```
dcTime := F_GetCurDcTickTime();

fbCtrl.A_GetYear( in := dcTime, get => wYear );
fbCtrl.A_GetMonth( in := dcTime, get => wMonth );
fbCtrl.A_GetDay( in := dcTime, get => wDay );
fbCtrl.A_GetDayOfWeek( in := dcTime, get => wDayOfWeek );
fbCtrl.A_GetHour( in := dcTime, get => wHour );
fbCtrl.A_GetMinute( in := dcTime, get => wMinute );
fbCtrl.A_GetSecond( in := dcTime, get => wSecond );
fbCtrl.A_GetMilli( in := dcTime, get => wMilli );
fbCtrl.A_GetMicro( in := dcTime, get => wMicro );
fbCtrl.A_GetNano( in := dcTime, get => wNano );
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 Build >= 1316 or higher	PC or CX (x86, ARM)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
	CX (ARM)	

7.9 FB_EcGetLastProtErrInfo



FB_EcGetLastProtErrInfo gives additional error information about the last occurred mailbox protocol parameter. An error-free mailbox command resets the last error each time.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    eProtocol   : E_EcMbxProtType := eEcMbxProt_FoE;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave, whose error information should be read.

eProtocol: EtherCAT Mailbox protocol type [▶ 77].

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    info        : ST_EcLastProtErrInfo;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the *bBusy output* has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the *bError output* is set.

info: Structure [▶ 79] with additional error information.

Example in ST:

Additional error information about the last occurred mailbox protocol error is read at an rising edge at *bGet* .

```
PROGRAM MAIN
VAR
    fbGetInfo : FB_EcGetLastProtErrInfo := ( sNetID := '172.16.6.195.2.1',
                        nSlaveAddr := 1004,
                        eProtocol := eEcMbxProt_FoE,
                        tTimeout := DEFAULT_ADS_TIMEOUT );
    bGet    : BOOL;
    bBusy   : BOOL;
    bError  : BOOL;
    nErrID  : UDINT;
    sInfo   : T_MaxString;
END_VAR
```

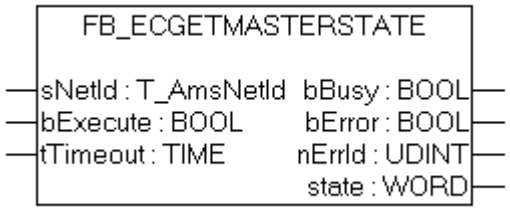
```
fbGetInfo(       bExecute:= bGet,
        bBusy=>bBusy,
        bError=>bError,
        nErrId=>nErrId );

sInfo := BYTEARR_TO_MAXSTRING( fbGetInfo.info.binDesc );
```

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0 Build > 1307 or higher	PC or CX (x86) CX (ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

8 EtherCAT State Machine

8.1 FB_EcGetMasterState



The function block FB_EcGetMasterState allows the EtherCAT state of the master to be read. If the call is successful, the output variable *state* of type word, contains the EtherCAT state of the master.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    state       : WORD;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

state: EtherCAT state of the master. Following values for **state** are possible:

Constant	Value	Description
EC_DEVICE_STATE_INIT	0x01	Master is in the state Init
EC_DEVICE_STATE_PREOP	0x02	Master is in the state Pre-Opereational
EC_DEVICE_STATE_SAFEOP	0x04	Master is in the state Safe-Operational
EC_DEVICE_STATE_OP	0x08	Master is in the state Operational

Example of an implementation in ST:

```
PROGRAM TEST_GetMasterState
VAR
    fbGetMasterState        : FB_EcGetMasterState;
    sNetId          : T_AmsNetId := '172.16.2.131.2.1';
    bExecute            : BOOL;
```

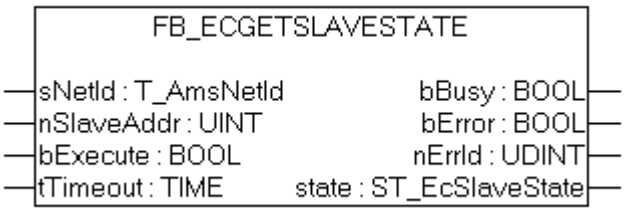
```
    state           : WORD;
    bError          : BOOL;
    nErrId          : UDINT;
END_VAR
```

```
fbGetMasterState(sNetId:= sNetId, bExecute:=bExecute);
state:=fbGetMasterState.state;
bError:=fbGetMasterState.bError;
nErrId:=fbGetMasterState.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

8.2 FB_EcGetSlaveState



The function block FB_EcGetSlaveState allows the EtherCAT status and the Link status of an individual EtherCAT slave to be read. If the call is successful, the output variable *state*, whose type is ST_EcSlaveState, contains the requested status information.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave whose status is to be read

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    state       : ST_EcSlaveState;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

state: Structure [▶ 82] that contains the EtherCAT status and the Link status of the slave.

Example of an implementation in ST:

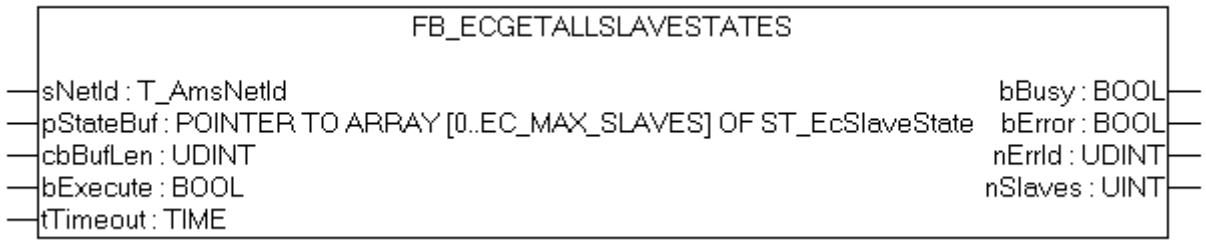
```
PROGRAM TEST_GetSlaveState
VAR
    fbGetSlaveState : FB_EcGetSlaveState;
    sNetId          : T_AmsNetId := '172.16.2.131.2.1';
    bExecute        : BOOL;
    state           : ST_EcSlaveState;
    nSlaveAddr      : UINT := 1001;
    bError          : BOOL;
    nErrId          : UDINT;
END_VAR
```

```
fbGetSlaveState(sNetId:= sNetId, nSlaveAddr:= nSlaveAddr, bExecute:=bExecute);
state:=fbGetSlaveState.state;
bError:=fbGetSlaveState.bError;
nErrId:=fbGetSlaveState.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

8.3 FB_EcGetAllSlaveStates



The FB_EcGetAllSlaveStates function block allows the EtherCAT status and the Link status of all the slaves connected to the master to be read. When the call is successful, the buffer passed in the parameter pStateBuf contains the requested status information as an array of ST_EcSlaveState.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    pStateBuf   : POINTER TO ARRAY[0..EC_MAX_SLAVES] OF ST_EcSlaveState;
    cbBufLen    : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

pStateBuf : The address of an array of ST_EcSlaveStates [▶ 82] into which the slave statuses are to be written.

cbBufLen: The maximum available buffer size for the data to be read, in bytes.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    nSlaves     : UINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

nSlaves: The number of slaves connected to the master.

Example of an implementation in ST:

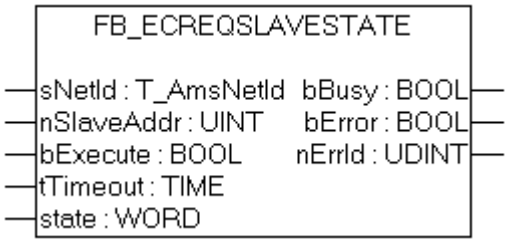
```
PROGRAM TEST_GetAllSlaveStates
VAR
    fbGetAllSlaveStates     : FB_EcGetAllSlaveStates;
    sNetId          : T_AmsNetId := '172.16.2.131.2.1';
    bExecute        : BOOL;
    devStates           : ARRAY[0..255] OF ST_EcSlaveState;
    nSlaves             : UINT := 0;
    bError          : BOOL;
    nErrId          : UDINT;
END_VAR
```

```
fbGetAllSlaveStates(sNetId:= sNetId, pStateBuf := ADR(devStates), cbBufLen:=SIZEOF(devStates), bExec
ute:=bExecute);
nSlaves:=fbGetAllSlaveStates.nSlaves;
bError:=fbGetAllSlaveStates.bError;
nErrId:=fbGetAllSlaveStates.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

8.4 FB_EcReqSlaveState



The function block FB_EcReqSlaveState requests the master to set an EtherCAT slave to the requested state. The parameter **state** specifies the EtherCAT state the slave should be set to.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
    state       : WORD;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

state: EtherCAT state the slave should be set to. Following values can be passed here:

Constant	Value	Description
EC_DEVICE_STATE_INIT	0x01	Set Slave to the init state
EC_DEVICE_STATE_PREOP	0x02	Set Slave to Pre-Operational
EC_DEVICE_STATE_BOOTSTRAP	0x03	Set Slave to the Bootstrap state. This state is used for firmware download.
EC_DEVICE_STATE_SAFEOP	0x04	Set Slave to Safe-Operational
EC_DEVICE_STATE_OP	0x08	Set Slave to Operational
EC_DEVICE_STATE_ERROR	0x10	If the Error-Bit is set in the EtherCAT Status-Byte (state.deviceState & EC_DEVICE_STATE_ERROR = TRUE), one can clear it by passing EC_DEVICE_STATE_ERROR.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Example of an implementation in ST:

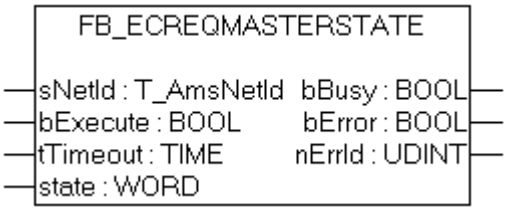
```
PROGRAM TEST_ReqSlaveState
VAR
    fbGetSlaveState     : FB_EcReqSlaveState;
    sNetId      : T_AmsNetId := '172.16.2.131.2.1';
    bExecute    : BOOL;
    state       : WORD := EC_DEVICE_STATE_INIT;
    nSlaveAddr      : UINT := 1001;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

```
fbGetSlaveState(sNetId:= sNetId, nSlaveAddr:= nSlaveAddr, bExecute:=bExecute, state:=state);
bError:=fbGetSlaveState.bError;
nErrId:=fbGetSlaveState.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

8.5 FB_EcReqMasterState



The function block FB_EcReqMasterState requests a new EtherCAT state from the master. The parameter **state** specifies the EtherCAT state that is requested from the master.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
    state       : WORD;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

state: EtherCAT state that is requested from the master. Following values can be passed here:

Constant	Value	Description
EC_DEVICE_STATE_INIT	0x01	Request the init state
EC_DEVICE_STATE_PREOP	0x02	Request the Pre-Operational state
EC_DEVICE_STATE_SAFEOP	0x04	Request the Safe-Operational state
EC_DEVICE_STATE_OP	0x08	Request the Operational state

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Example of an implementation in ST:

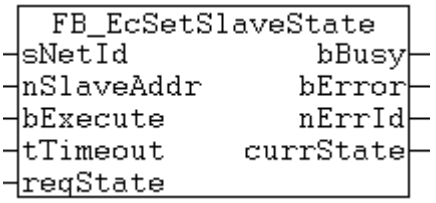
```
PROGRAM TEST_ReqMasterState
VAR
    fbReqMasterState    : FB_EcReqMasterState;
    sNetId      : T_AmsNetId := '172.16.2.131.2.1';
    bExecute    : BOOL;
    state       : WORD :=EC_DEVICE_STATE_INIT;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

```
fbReqMasterState(sNetId:= sNetId, bExecute:=bExecute, state:=state);
bError:=fbGetMasterState.bError;
nErrId:=fbGetMasterState.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

8.6 FB_EcSetSlaveState



The functionblock FB_EcSetSlaveState is used to set the EtherCAT slave in the state that is stated by the parameter *reqState*. The functionblock waits the maximum *tTimeout* time until the new state is set.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    bExecute    : BOOL;
    tTimeout    : TIME := T#10s;
    reqState    : WORD;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave whose state is to be set.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

reqState: EtherCAT state in which the slave is to be set. Possible values for *reqState* are:

Constant	Value	Description
EC_DEVICE_STATE_INIT	0x01	Set slave in Init state.
EC_DEVICE_STATE_PREOP	0x02	Set slave in pre-operational state.
EC_DEVICE_STATE_BOOTSTRAP	0x03	Set slave in bootstrap state. This state is used to make a firmware download.
EC_DEVICE_STATE_SAFEOP	0x04	Set slave in safe-operational state.

Constant	Value	Description
EC_DEVICE_STATE_OP	0x08	Set slave in operational state.
EC_DEVICE_STATE_ERROR	0x10	If the error bit at the EtherCAT Slave is set in the status byte (currState.deviceState AND EC_DEVICE_STATE_ERROR = TRUE), the error bit can be reset by setting EC_DEVICE_STATE_ERROR

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    currState   : ST_EcSlaveState;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

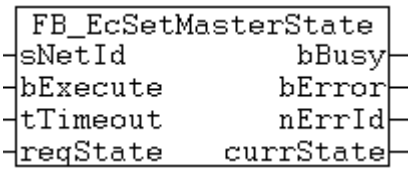
nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

currState: Current EtherCAT state [▶ 82] of the slave.

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0 Build > 1307 or higher	PC or CX (x86) CX (ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

8.7 FB_EcSetMasterState



The EtherCAT Master state, assigned in the variable **reqState,** can be requested by the master with the functionblock FB_EcSetMasterState
The functionblock waits for the maximum *tTimeout* time allowed until the new state is set.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    bExecute    : BOOL;
    tTimeout    : TIME := T#10s;
    reqState    : WORD;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

reqState: EtherCAT state requested by the master. Possible values for *reqState* are:

Constant	Value	Description
EC_DEVICE_STATE_INIT	0x01	Request init state from the master
EC_DEVICE_STATE_PREOP	0x02	Request pre-operational state from the master
EC_DEVICE_STATE_SAFEOP	0x04	Request safe-operational state from the master
EC_DEVICE_STATE_OP	0x08	Request operational state from the master

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    currState   : WORD;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

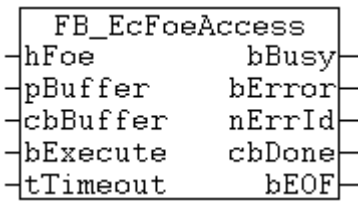
currState: Current EtherCAT state of the master.

Requirements

Development Environment	Target System	PLCLibraries to include
TwinCAT v2.10.0 Build > 1307 or higher	PC or CX (x86) CX (ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

9 FoE (File over EtherCAT)

9.1 FB_EcFoeAccess



This functionblock writes or reads data via the communication port of the "File access over EtherCAT" mailbox protocol.

VAR_INPUT

```
VAR_INPUT
    hFoe      : T_HFoe;
    pBuffer   : DWORD;
    cbBuffer  : UDINT;
    bExecute  : BOOL;
    tTimeout  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

hFoe: "File access over EtherCAT" handle [▶ 84].

pBuffer: Contains the data of the buffer in which the data is to be read (read access) or the address of the buffer which contains the data to be written (write access). The buffer can be a single variable, an array, or a structure, whose addresse can be determined with the ADR operator.

cbBuffer: Contains the number of the data to be written or read, in bytes

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    cbDone  : UDINT;
    bEOF    : BOOL;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

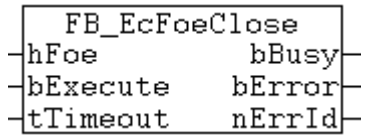
cbDone: Number of last successfully written or read data bytes.

bEOF: End of File. This output is switched to TRUE if the end of file is reached (for read access). This variable is irrelevant for write access

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0 or higher	PC or CX (x86) CX (ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib are included automatically)

9.2 FB_EcFoeClose



This function block closes the communication port for the "File access over EtherCAT" mailbox protocol.

VAR_INPUT

```
VAR_INPUT
    hFoe        : T_HFoe ;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

hFoe: "File access over EtherCAT" handle [▶ 84].

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

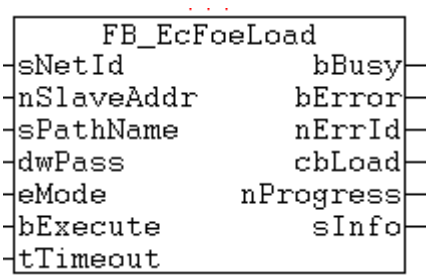
bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0 or higher	PC or CX (x86) CX (ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

9.3 FB_EcFoeLoad



The function block FB_EcFoeLoad can be used to download or upload files to or from an EtherCAT device via the "File access over EtherCAT" mailbox protocol (FoE).

The file path can only point to the local file system on the computer. This means that network paths cannot be used here! To upload or download files via the FoE protocol, the function block automatically resets the EtherCAT device to BOOTSTRAP mode. Finally, the function block tries to reset the device to the original state.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    sPathName   : T_MaxString;
    dwPass      : DWORD := 0;
    eMode       : E_EcFoeMode := eFoeMode_Write;
    bExecute    : BOOL;
    tTimeout    : TIME := T#200s;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT device.

nSlaveAddr: Fixed address of EtherCAT slaves, whose file is to be uploaded or downloaded.

sPathName: Contains the path name and the file name of file to be written or to be read. (e.g.: 'C:\FOE_Test\EL6751\ECATFW__EL6751_C6_V0030.efw').

dwPass: Password (default: 0).

eMode: "File access over EtherCAT" access mode [▶ 77] (default: write access).

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbLoad      : UDINT;
    nProgress   : UDINT;
    sInfo       : T_MaxString;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

cbLoad: Number of successfull written or read data bytes.

nProgress: Percentage progress at write access (Range: 0 - 100%). This variable is not used by read access, and always 0.

sInfo: Additional command information (reserved).

Example in ST:

The firmware download via the "File access over EtherCAT" mailbox protocol is started at a rising edge at the bLoad variable.

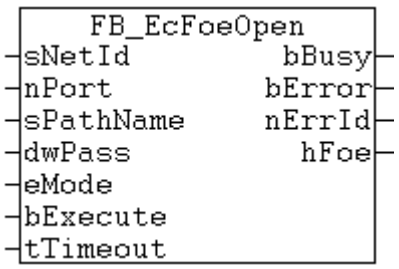
```
PROGRAM MAIN
VAR
    fbDownload : FB_EcFoeLoad := (
        sNetID := '5.0.34.38.3.1',
        nSlaveAddr := 1004,
        sPathName := 'C:\FOE_Test\EL6751\ECATFW__EL6751_C6_V0030.efw',
        dwPass := 0,
        eMode := eFoeMode_Write );

    bLoad : BOOL;
    bBusy : BOOL;
    bError : BOOL;
    nErrID : UDINT;
    nBytesWritten : UDINT;
    nPercent : UDINT;
END_VAR
```

```
fbDownload(      bExecute := bLoad,
        bBusy => bBusy,
        bError => bError,
        nErrId => nErrID,
        cbLoad => nBytesWritten,
        nProgress => nPercent );
```

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v2.10.0 Build > 1307	PC or CX (x86) CX (ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

9.4 FB_EcFoeOpen



This function block opens the communication port for the "File access over EtherCAT" mailbox protocol.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nPort       : UINT;
    sPathName   : T_MaxString;
    dwPass      : DWORD;
    eMode       : E_EcFoeMode;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT device.

nPort: Fixed address of the EtherCAT device.

sPathName: Path name (e.g.: 'c:\TwinCAT\FOE\Data.fwp').

dwPass: Password.

eMode: Access mode [▶ 77] (write/read access).

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    hFoe        : T_HFoe;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

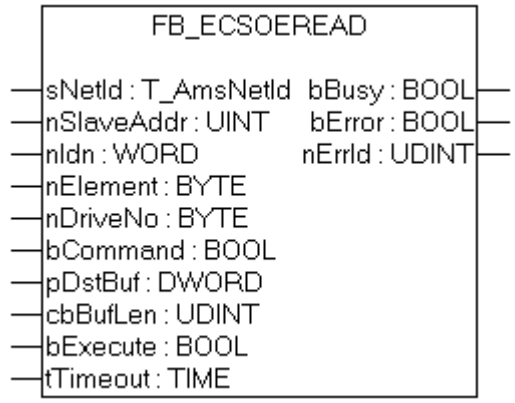
hFoe: "File access over EtherCAT"-Handle [▶ 84].

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0 or higher	PC or CX (x86) CX (ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

10 SoE

10.1 FB_EcSoeRead



The function block FB_EcSoeRead allows to read drive parameters with the "Servo Drive Profile over EtherCAT"(SoE) protocol. This requires the slave to have a mailbox and to support the SoE protocol. The parameter to be read is specified by the parameters nIdn (Identification number), nElement and nDriveNo .

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    nSubIndex   : BYTE;
    nIdn        : WORD;
    nElement    : BYTE;
    nDriveNo    : BYTE;
    bCommand    : BOOL
    pDstBuf     : DWORD;
    cbBufLen    : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave to which the SoE read command should be sent to.

nIdn: Identification number of the parameter that is to be read.

nElement: Element number of the parameter that is to be read. The following values are allowed:

Value	Description
0x01	Data Status
0x02	Name (read only)
0x04	Attribute
0x08	Unit
0x10	Minimum
0x20	Maximum
0x40	Value
0x80	Default

nDriveNo: Number of the drive.

bCommand: This parameter should be set, if the internal command execution should be used.

pDestBuf: The address (pointer) of the receive buffer.

cbBufLen: The maximum available buffer size for the data to be read, in bytes.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Example for an implementation in ST:

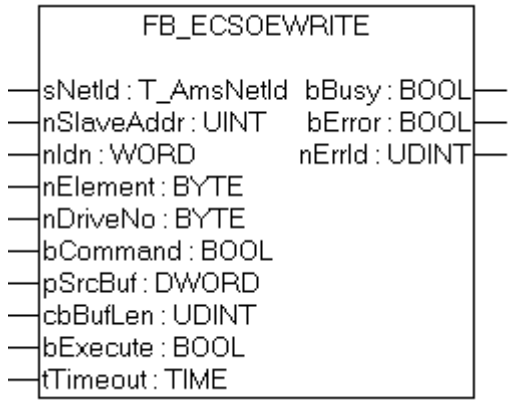
```
PROGRAM TEST_SoERead
VAR
    fbSoERead   : FB_EcSoERead;
    sNetId      : T_AmsNetId := '172.16.2.131.2.1';
    bExecute    : BOOL;
    nSlaveAddr  : UINT := 1006;
    nIdn        : WORD := 15;
    nElement    : BYTE := 0;
    nDriveNo    : BYTE := 0;
    bCommand    : BOOL := FALSE;
    val         : UINT;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

```
fbSoERead(sNetId:= sNetId,nSlaveAddr :=nSlaveAddr, nIdn := nIdn, nElement:=nElement, nDriveNo := nDr
iveNo, bCommand:=bCommand, pDstBuf:= ADR(val), cbBufLen:=SIZEOF(val),bExecute:=bExecute);
bError:=fbSoERead.bError;
nErrId:=fbSoERead.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

10.2 FB_EcSoeWrite



The function block FB_EcSoeWrite allows to write drive parameters with the "Servo Drive Profile over EtherCAT"(SoE) protocol. This requires the slave to have a mailbox and to support the SoE protocol. The parameter to be written is specified by the parameters nIdn (Identification number), nElement and nDriveNo .

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    nIdn        : WORD;
    nElement    : BYTE;
    nDriveNo    : BYTE;
    bCommand    : BOOL;
    pSrcBuf     : DWORD;
    cbBufLen    : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

nSlaveAddr: Fixed address of the EtherCAT slave to which the SoE write command should be sent to.

nIdn: Identification number of the parameter that is to be written.

nElement: Element number of the parameter that is to be written. The following values are allowed:

Value	Description
0x01	Data Status
0x02	Name (read only)
0x04	Attribute
0x08	Unit
0x10	Minimum
0x20	Maximum
0x40	Value
0x80	Default

nDriveNo: Number of the drive.

bCommand: This parameter should be set, if the internal command execution should be used.

pSrcBuf: The address (pointer) of the send buffer.

cbBufLen: The maximum available buffer size for the data to be sent, in bytes.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Example for an implementation in ST:

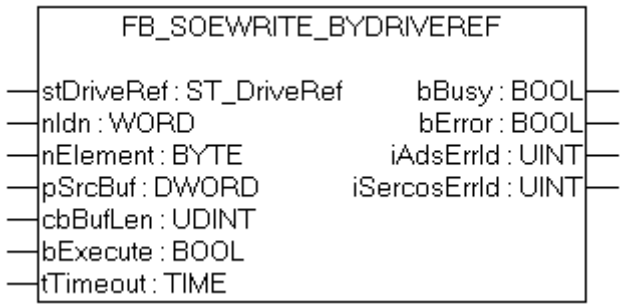
```
PROGRAM TEST_SoEWrite
VAR
    fbSoeWrite  : FB_EcSoEWrite;
    sNetId      : T_AmsNetId := '172.16.2.131.2.1';
    bExecute    : BOOL;
    nSlaveAddr  : UINT := 1006;
    nIdn        : WORD := 15;
    nElement    : BYTE := 0;
    nDriveNo    : BYTE := 0;
    bCommand    : BOOL := FALSE;
    val         : UINT;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

```
fbSoEWrite(sNetId:= sNetId,nSlaveAddr :=nSlaveAddr, nIdn := nIdn, nElement:=nElement, nDriveNo := nD
riveNo,bCommand:=bCommand, pSrcBuf:= ADR(val), cbBufLen:=SIZEOF(val),bExecute:=bExecute);
bError:=fbSoEWrite.bError;
nErrId:=fbSoEWrite.nErrId;
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

10.3 FB_SoEWrite_ByDriveRef



The function block FB_EcSoeWrite allows to write drive parameters with the "Servo Drive Profile over EtherCAT"(SoE) protocol. This requires the slave to have a mailbox and to support the SoE protocol. The parameter to be written is specified by the parameters nIdn (Identification number), nElement and stDriveRef .

VAR_INPUT

```
VAR_INPUT
    stDriveRef      : ST_DriveRef; (* contains sNetID of EcMaster, nSlaveAddr of EcDrive, nDriveNo o
f EcDrive, either preset or read from NC *)
    nIdn            : WORD; (* SoE IDN: e.g. "S_0_IDN + 1" for S-0-0001 or "P_0_IDN + 23" for P-0-00
23*)
    nElement        : BYTE; (* SoE element.*)
    pSrcBuf         : DWORD; (* Contains the address of the buffer containing the data to be send. *
)
    cbBufLen        : UDINT; (* Contains the max. number of bytes to be received. *)
    bExecute        : BOOL; (* Function block execution is triggered by a rising edge at this input.
*)
    tTimeout        : TIME := DEFAULT_ADS_TIMEOUT;
(* States the time before the function is cancelled. *)
END_VAR
```

stDriveRef: The drive reference can be linked in the System Manager between PLC and drive. The link can be done to an instance of the ST_PlcDriveRef. The structure ST_PlcDriveRef contains the NetID as byte array. The byte array can be converted to a string. See ST_DriveRef.

nIdn: Identification number of the parameter that is to be written.

nElement: Element number of the parameter that is to be written. The following values are allowed:

Value	Description
0x01	Data Status
0x02	Name (read only)
0x04	Attribute
0x08	Unit
0x10	Minimum
0x20	Maximum
0x40	Value
0x80	Default

pSrcBuf: The address (pointer) of the send buffer.

cbBufLen: The maximum available buffer size for the data to be sent, in bytes.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy           : BOOL;
    bError          : BOOL;
    iAdsErrId       : UINT;
    iSercosErrId    : UINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

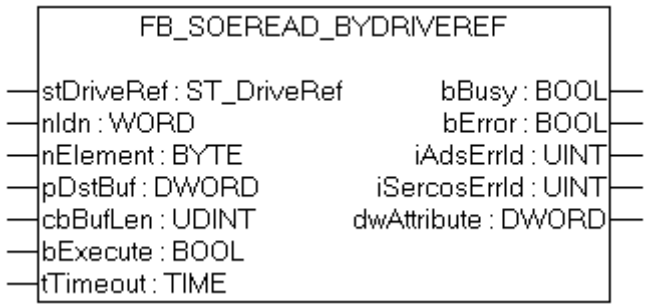
iAdsErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

iSercosErrId: Supplies the Sercos error code associated with the most recently executed command if the bError output is set.

Requirements

Development environment	Target system type	Libraries to be linked
TwinCAT v2.10.0 Build >= 1329 or higher	PC or CX (x86)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1329 or higher	CX (ARM)	

10.4 FB_SoERead_ByDriveRef



The function block FB_SoeRead_ByRef allows to read drive parameters with the "Servo Drive Profile over EtherCAT"(SoE) protocol. This requires the slave to have a mailbox and to support the SoE protocol. The parameter to be read is specified by the parameters nIdn (Identification number), nElement and stDriveRef .

VAR_INPUT

```
VAR_INPUT
    stDriveRef      : ST_DriveRef; (* contains sNetID of EcMaster, nSlaveAddr of EcDrive, nDriveNo o
f EcDrive, either preset or read from NC *)
    nIdn            : WORD; (* SoE IDN: e.g. "S_0_IDN + 1" for S-0-0001 or "P_0_IDN + 23" for P-0-00
23*)
    nElement        : BYTE; (* SoE element.*)
    pDstBuf         : DWORD; (* Contains the address of the buffer for the received data. *)
    cbBufLen        : UDINT; (* Contains the max. number of bytes to be received. *)
    bExecute        : BOOL; (* Function block execution is triggered by a rising edge at this input.
*)
    tTimeout        : TIME := DEFAULT_ADS_TIMEOUT;
(* States the time before the function is cancelled. *)
END_VAR
```

stDriveRef: The drive reference can be linked in the System Manager between PLC and drive. The link can be done to an instance of the ST_PlcDriveRef. The structure ST_PlcDriveRef contains the NetID as byte array. The byte array can be converted to a string. See ST_DriveRef.

nIdn: Identification number of the parameter that is to be read.

nElement: Element number of the parameter that is to be read. The following values are allowed:

Value	Description
0x01	Data Status
0x02	Name (read only)
0x04	Attribute
0x08	Unit
0x10	Minimum
0x20	Maximum
0x40	Value
0x80	Default

pDstBuf: The address (pointer) of the buffer to be read.

cbBufLen: The maximum available buffer size for the data to be read, in bytes.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy         : BOOL;
    bError        : BOOL;
    iAdsErrId     : UINT;
    iSercosErrId  : UINT;
    dwAttribute   : DWORD;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

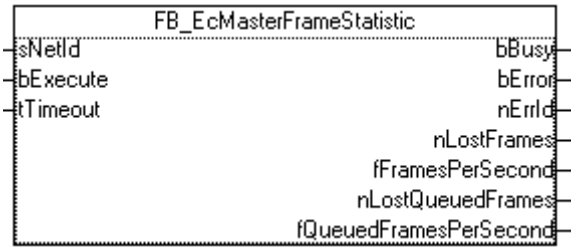
iSercosErrId: Supplies the Sercos error code associated with the most recently executed command if the bError output is set.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

11 Frame Statistic

11.1 FB_EcMasterFrameStatistic



The FB_EcMasterFrameStatistic function block reads the frame statistic of the EtherCAT Master. The number of 'lost frames', frames per second, the number of lost queued frames and queued frames per Second is shown at the outputs of the FB.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy                  : BOOL;
    bError                 : BOOL;
    nErrId                 : UDINT;
    nLostFrames            : UDINT;
    fFramesPerSecond       : LREAL;
    nLostQueuedFrames      : UDINT;
    fQueuedFramesPerSecond : LREAL;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

nLostFrames: Actual number of lost frames.

fFramesPerSecond: Actual number of frames per second.

nLostQueuedFrames: Actual number of queued frames.

fQueuedFramesPerSecond: Actual number of queued frames per Second.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

11.2 FB_EcMasterFrameStatisticClearCRC

The FB_EcMasterFrameStatisticClearCRC function block clears the CRC counters of all EtherCAT Slaves.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

11.3 FB_EcMasterFrameStatisticClearFrames

The FB_EcMasterFrameStatisticClearFrames function block clears the lost frames counter of the EtherCAT Master.

VAR_INPUT

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the EtherCAT master device.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

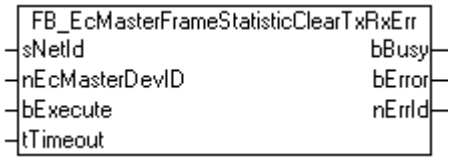
bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

11.4 FB_EcMasterFrameStatisticClearTxRxErr



The FB_EcMasterFrameStatisticClearTxRxErr function block clears the error counter of the Miniport-Driver (network adapter).

VAR_INPUT

```
VAR_INPUT
    sNetId          : T_AmsNetId;
    nEcMasterDevID  : INT;
    bExecute        : BOOL;
    tTimeout        : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: This is a string that contains the AMS network identifier of the CPU (PC).

nEcMasterDevID: Device ID of the EtherCAT Master.

bExecute: The block is activated by a rising edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: This output is set when the function block is activated and remains set until an acknowledgement is received.

bError: This output is set up after the bBusy output has been reset if there has been an error in transmission of the command.

nErrId: Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

12 Data types

12.1 DCTIMESTRUCT

Structured TwinCAT "Distributed Clock System Time" time format. The smallest unit is nanosecond. This data type represents the **number of nanoseconds since 01.01.2000 (GMT).**

```
TYPE DCTIMESTRUCT :
STRUCT
    wYear           : WORD;
    wMonth          : WORD;
    wDayOfWeek      : WORD;
    wDay            : WORD;
    wHour           : WORD;
    wMinute         : WORD;
    wSecond         : WORD;
    wMilliseconds   : WORD;
    wMicroseconds   : WORD;
    wNanoseconds    : WORD;
END_STRUCT
END_TYPE
```

wYear : Specifies the year: 2000 ~ 2584;

wMonth : Specifies the month: 1 ~ 12 (January = 1, February = 2 and so on);

wDayOfWeek : Specifies the day of the week: 0 ~ 6 (Sunday = 0, Monday = 1 and so on);

wDay : Specifies the day of the month: 1 ~ 31;

wHour : Specifies the hour: 0 ~ 23;

wMinute : Specifies the minute: 0 ~ 59;

wSecond : Specifies the second: 0 ~ 59;

wMilliseconds : Specifies the millisecond: 0 ~ 999;

wMicroseconds : Specifies the microsecond: 0 ~ 999;

wNanoseconds : Specifies the nanosecond: 0 ~ 999;

Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.10.0 Build > 1316 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

12.2 E_EcAdressingType

```
TYPE E_EcAdressingType :
(
    eAdressingType_AutoInc:=1, (* Adress slave by it's position. (adp = 1-
position, 1.Slave = 0, 2.Slave = 0xffff(-1) etc) *)
                (* EtherCAT commands: APRD, APWR, APRW *)
    eAdressingType_Fixed,   (* Adress slave by configured ethercat slave address (adp = configured a
ddress) *)
                (* EtherCAT commands: FPRD, FPWR, FPRW *)
    eAdressingType_Broadcast    (* Adress all slaves. *)
                (* EtherCAT commands: BRD, BWR, BRW *)
);
END_TYPE
```

12.3 E_EcFoeMode

"File access over EtherCAT" mailbox protokol access type.

```
TYPE E_EcFoeMode :
(
    eFoeMode_Write := 1,
    eFoeMode_Read
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 Build > 1307 or higher	PC or CX (x86) CX (ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.lib are included automatically)

12.4 E_EcMbxProtType

EtherCAT mailbox protocol type.

```
TYPE E_EcMbxProtType:
(
    eEcMbxProt_CoE := 3,(* CANopen over EtherCAT *)
    eEcMbxProt_FoE := 4,(* File over EtherCAT *)
    eEcMbxProt_SoE := 5 (* Servo Drive Profile over EtherCAT *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.10.0 Build > 1307	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.lib are included automatically)

12.5 E_EcScanSlavesCommandStatus

```
TYPE E_EcScanSlavesCommandStatus :
(
    eEcScanSlavesCommandState_Completed_NoErrors_NoReply := 0,(* completed, no errors, no reply *)
    eEcScanSlavesCommandState_Completed_NoErrors_Reply := 1, (* completed, no errors, reply *)
    eEcScanSlavesCommandState_Completed_Error_NoReply := 2, (* completed, errors, no reply *)
    eEcScanSlavesCommandState_Completed_Error_Reply := 3, (* completed, errors, reply *)
    eEcScanSlavesCommandState_Completed_Reserved := 4 (* reserved *)
);
END_TYPE
```

12.6 ProductCode

```
TYPE ProductCode :
    (
    PCTYPE_XXDDDD       := 0,
    PCTYPE_XXDDDD_DDDD      := 1,
    PCTYPE_XXDDDD_DDDD_DDDD     := 2,
    PCTYPE_XXDDDD_XDDD      := 3,
    PCTYPE_XXDDDD_XDDD_DDDD     := 4,
    PCTYPE_XXDDDD_DDDD_XDDD     := 5,
    PCTYPE_XXDDDD_XDDD_XDDD     := 6
    );
END_TYPE
```

12.7 ST_EcCrcError

Structure containing the Crc error counters of the individual ports (A, B and C) of an EtherCAT slave device.

```
TYPE ST_EcCrcError :
STRUCT
    portA : UDINT;
    portB : UDINT;
    portC : UDINT;
END_STRUCT
END_TYPE
```

portA: CRC error counter of Port A

portB: CRC error counter of Port B

portC: CRC error counter of Port C

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

12.8 ST_EcCrcErrorEx

Structure containing the Crc error counters of the individual ports (A, B, C and D) of an EtherCAT slave device.

```
YPE ST_EcCrcErrorEx :
STRUCT
    portA : UDINT;
    portB : UDINT;
    portC : UDINT;
    portD : UDINT;
END_STRUCT
END_TYPE
```

portA: CRC error counter of Port A

portB: CRC error counter of Port B

portC: CRC error counter of Port C

portD: CRC error counter of Port D

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 Build >= 1319 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1319 or higher	CX (ARM)	

12.9 ST_EcLastProtErrInfo

Th structure ST_EcLastProtErrInfo contains additional error information about the last occurred "EtherCAT mailbox protocol error".

```
TYPE ST_EcSlaveState:
STRUCT
    ownAddr : ST_AmsAddr;
    orgAddr : ST_AmsAddr;
    errCode : UDINT;
    binDesc : ARRAY[0..MAX_STRING_LENGTH] OF BYTE;
END_STRUCT
END_TYPE
```

ownAddr: Own AMS address (address of communication participant which calls the error information).

orgAddr: AMS address of the error causer (address of communication participant which caused the protocol error).

errCode: Mailbox protocol error number [▶ 88] (SoE, CoE, FoE error code).

binDesc: Additional error information as binary data. The additional error information is device specific and contains e.g. a string or binary data.

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.10.0 Build > 1307 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

12.1 ST_EcMasterStatistic 0

```
TYPE ST_EcMasterStatistic :
STRUCT
    nSysTime                : UDINT;
    nCycFrameCnt            : UDINT;
    nCycFrameMissedCnt      : UDINT;
    nQueuedFrameCnt         : UDINT;
    nQueuedFrameMissedCnt   : UDINT;
END_STRUCT
END_TYPE
```

nSysTime: System timestamp in µs

nCycFrameCnt: Amount of cyclic EtherCAT frames

nCycFrameMissedCnt: Amount of lost cyclic EtherCAT frames

nQueuedFrameCnt: Amount of acyclic (queued) EtherCAT frames

nQueuedFrameMissedCnt: Amount of lost acyclic (queued) EtherCAT frames

12.1 ST_EcSlaveConfigData 1

The structure ST_EcSlaveConfigData describes the EtherCAT configurationdata of one EtherCAT Slave.

```
TYPE ST_EcSlaveConfigData:
STRUCT
    nEntries        : WORD;
```

```
    nAddr           : WORD;
    sType           : STRING[15];
    sName           : STRING[31];
    nDevType        : DWORD;
    stSlaveIdentity : ST_EcSlaveIdentity;
    nMailboxOutSize : WORD;
    nMailboxInSize  : WORD;
    nLinkStatus     : BYTE;
END_STRUCT
END_TYPE
```

nEntries: intern used!

nAddr: Address of one EtherCAT Slave.

sType: EtherCAT Type of one Slave.

sName: Name of one EtherCAT Slave.

nDevType: EtherCAT Device Type of one Slave.

stSlaveIdentity: Identity of one EtherCAT Slave (s. ST_EcSlaveIdentity [▶ 80]).

nMailboxOutSize: Mailbox OutSize for one EtherCAT Slave.

nMailboxInSize: Mailbox InSize for one EtherCAT Slave.

nLinkStatus: Link Status for one EtherCAT Slave.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1307 or higher	CX (ARM)	

12.1 ST_EcSlaveIdentity 2

Structure with

```
TYPE ST_EcSlaveIdentity :
STRUCT
    vendorId        : UDINT;
    productCode     : UDINT;
    revisionNo      : UDINT;
    serialNo        : UDINT;
END_STRUCT
END_TYPE
```

vendorId: Vendor-ID of the slave device.

productCode: Product code of the slave device.

revisionNo: Indicates the revision number of the slave device.

serialNo: Indicates the serial number of the slave device.

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

12.1 ST_EcSlaveScannedData 3

The structure ST_EcSlaveScannedData describes the EtherCAT configuration data of one scanned EtherCAT Slave

```
TYPE ST_EcSlaveConfigData:
STRUCT
    nEntries        : WORD;
    nAddr           : WORD;
    stSlaveIdentity : ST_EcSlaveIdentity;
    ndlStatusReg    : WORD;
END_STRUCT
END_TYPE
```

nEntries: intern used!

nAddr: Address of one EtherCAT Slave.

stSlaveIdentity: Identity of one EtherCAT Slave (s. ST_EcSlaveIdentity).

ndlStatusReg: Link State of one EtherCAT Slave from ESC register $0110/0111_{hex}$ resp. $272/273_{dec}$. If the slave is not accessible/offline the sate is 0. The assignment „Port number <=> socket/Ebus contact“ can be seen in the device documentation. If not specified, Port0 is the Ebus contact on the left of an EL/ES terminal resp. RJ45 socket of an EP box and Port1 is the contact on the right/ RJ 45 socket.

Bits and description

Bit	Description
1	internal use
2	internal use
3	internal use
4	physical link on Port 0 0: no link, 1: Link detected
5	physical link on Port 1 0: no link, 1: Link detected
6	physical link on Port 2 0: no link, 1: Link detected
7	physical link on Port 3 0: no link, 1: Link detected
8	Loop Port 0 0: Open, 1:Closed
9	Communication on Port 0 0:no stable communication, 1:Communication established
10	Loop Port 1 0: Open, 1:Closed
11	Communication on Port 1 0:no stable communication, 1:Communication established
12	Loop Port 2 0: Open, 1:Closed
13	Communication on Port 2 0:no stable communication, 1:Communication established
14	Loop Port 3 0: Open, 1:Closed
15	Communication on Port 3 0:no stable communication, 1:Communication established

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1307 or higher	CX (ARM)	

12.1 ST_EcSlaveState 4

The structure ST_EcSlaveState contains the EtherCAT status and the Link status of an EtherCAT slave device.

```
TYPE ST_EcSlaveState:
STRUCT
    deviceState     :BYTE;
    linkState       :BYTE;
END_STRUCT
END_TYPE
```

deviceState: EtherCAT status of a slave. The status can adopt one of the following values:

Constant	Value	Description
EC_DEVICE_STATE_INIT	0x01	Init State
EC_DEVICE_STATE_PREOP	0x02	Pre-Operational State
EC_DEVICE_STATE_BOOTSTRAP	0x03	Bootstrap State
EC_DEVICE_STATE_SAFEOP	0x04	Safe-Operational State
EC_DEVICE_STATE_OP	0x08	Operational State

Additionally following bits can be set:

Konstante	Value	Description
EC_DEVICE_STATE_ERROR	0x10	Statemachine error in the EtherCAT slave
EC_DEVICE_STATE_INVALID_VPRS	0x20	Invalid VendorId, Product Code, RevisionsNo or SerialNo
EC_DEVICE_STATE_INITCMD_ERROR	0x40	Error occured while sending initialization commands.

linkState: Link status of an EtherCAT slave. The Link status can consist of an ORing of the following bits.

Konstante	Value	Description
EC_LINK_STATE_OK	0x00	Link status ok
EC_LINK_STATE_NOT_PRESENT	0x01	No communication with the EtherCAT-Slave
EC_LINK_STATE_LINK_WITHOUT_COMM	0x02	Error at port X(defined by EC_LINK_STATE_PORT_A/B/C/D). The port has a link, but no communication is possible.
EC_LINK_STATE_MISSING_LINK	0x04	Missing link at port X(defined by EC_LINK_STATE_PORT_A/B/C/D).
EC_LINK_STATE_ADDITIONAL_LINK	0x08	Additional link at port X(defined by EC_LINK_STATE_PORT_A/B/C/D).
EC_LINK_STATE_PORT_A	0x10	Port 0
EC_LINK_STATE_PORT_B	0x20	Port 1
EC_LINK_STATE_PORT_C	0x40	Port 2
EC_LINK_STATE_PORT_D	0x80	Port 3

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

12.1 ST_EcSlaveStateBits 5

The structure ST_EcSlaveStateBits contains the EtherCAT status and the Link status of an EtherCAT slave device.

```
TYPE ST_EcSlaveStateBits:
STRUCT
    bInit             : BOOL;
    bPreop            : BOOL;
    bBootStrap        : BOOL;
    bSafeOp           : BOOL;
    bOp               : BOOL;
    bError            : BOOL;
    bInvVPRS          : BOOL;
    bInitCmdError     : BOOL;

    bLinkNotPresent   : BOOL;
    bLinkWithoutComm  : BOOL;
    bLinkMissing      : BOOL;
    bAdditionalLink   : BOOL;
    bPortA            : BOOL;
    bPortB            : BOOL;
    bPortC            : BOOL;
    bPortD            : BOOL;
END_STRUCT
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

12.1 ST_TPCTYPE_CODE_XXDDD 6

```
TYPE ST_TPCTYPE_CODE_XXDDD :
STRUCT
    ty : UINT;
    c1 : UINT;
    c2 : UINT;
    d1 : UINT;
    d2 : UINT;
    d3 : UINT;
END_STRUCT
END_TYPE
```

12.1 ST_TPCTYPE_CODE_XXDDXD7

```
TYPE ST_TPCTYPE_CODE_XXDDXD :
STRUCT
    ty : UINT;
    c1 : UINT;
    c2 : UINT;
    d1 : UINT;
    c3 : UINT;
    d2 : UINT;
    d3 : UINT;
END_STRUCT
END_TYPE
```

12.1 ST_TPCTYPE_CODE_XXDXDD8

```
TYPE ST_TPCTYPE_CODE_XXDXDD :
STRUCT
    ty : UINT;
    c1 : UINT;
    c2 : UINT;
    d1 : UINT;
    d2 : UINT;
    c3 : UINT;
    d3 : UINT;
END_STRUCT
END_TYPE
```

12.1 ST_TPCTYPE_CODE_XXDXDXD9

```
TYPE ST_TPCTYPE_CODE_XXDXDXD :
STRUCT
    ty : UINT;
    c1 : UINT;
    c2 : UINT;
    d1 : UINT;
    c3 : UINT;
    d2 : UINT;
    c4 : UINT;
    d3 : UINT;
    END_STRUCT
END_TYPE
```

12.2 T_HFoe0

"File access over EtherCAT"-Handle.
The handle must be initialised once before using with the function block FB_EcFoeOpen [▶ 63]. The variables of this structured type can not be written directly.

```
TYPE T_HFoe :
STRUCT
    sNetID  : T_AmsNetId := '';
    nPort   : T_AmsPort := 0;
    handle  : UDINT := 0;
    eMode   : E_EcFoeMode := eFoeMode_Write;
END_STRUCT
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.10.0 Build > 1307 or higher	PC or CX (x86) CX (ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

Also see about this

- E_EcFoeMode [▶ 77]

12.2 T_DCTIME 1

The data type T_DCTIME represents the *Distributed Clock System Time* (known as DC Time for short) in the form of a linear, 64-bit integer value. The time is expressed in nanoseconds since 1.1.2000 UTC.
The data type is implemented as two 32-bit DWORD variables, which simplifies processing in the PLC. With the ui64-bit functions from the TcUtilities.Lib you can execute simple mathematical operations (addition and subratction of times) with this data type.

```
TYPE T_DCTIME : T_ULARGE_INTEGER;
END_TYPE
```

Useful "Distributed Clock System Time" constants	Description
EC_DCTIME_DELTA_OFFSET	Number of 100 nanosecond ticks between 1.1.1601 and 1.1.2000 It's the difference between "Windows File Time" and "Distributed Clock System Time".
EC_DCTIME_DATEDELTA_OFFSET	Number of past days since year zero until 1 January 2000
EC_DCTIME_TICKSPERMSEC	Number of nanosecond ticks per millisecond
EC_DCTIME_TICKSPERSEC	Number of nanosecond ticks per second
EC_DCTIME_TICKSPERDAY	Number of nanosecond ticks per day

Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.10.0 Build > 1310 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

12.2 T_DCTIME32 2

32 bit TwinCAT "Distributed Clock System Time" time format. The smallest unit is nanosecond.

Note:

This 32-bit *DC System Time* is formed from the complete, absolute 64-bit DC System Time (T_DCTIME [▶ 85]), by using only the lowest-value 32 bits. As a result, the property of an absolutely unambiguous time is lost, and the assumption that has to be made that this 32-bit time will only be used within a time window of ± 2.147 seconds on either side of the current system time, since it is only unambiguous within this range. There are many applications in which this assumption is possible.
If the assumption is not satisfied, errors can arise in the interpretation and the further processing of this time.

```
TYPE T_DCTIME32 : UDINT;
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.11.0 Build >= 1524 or higher	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

13 Constants

13.1 Global constants

VAR_GLOBAL CONSTANT

```
EC_AMSPORT_MASTER            :UINT   :=16#FFFF;
EC_MAX_SLAVES           :UINT   :=16#FFFF;

(* Device states *)
EC_DEVICE_STATE_MASK         :BYTE   :=16#0F;
EC_DEVICE_STATE_INIT        :BYTE   :=16#01;
EC_DEVICE_STATE_PREOP       :BYTE   :=16#02;
EC_DEVICE_STATE_BOOTSTRAP           :BYTE   :=16#03;
EC_DEVICE_STATE_SAFEOP      :BYTE   :=16#04;
EC_DEVICE_STATE_OP              :BYTE   :=16#08;
EC_DEVICE_STATE_ERROR       :BYTE   :=16#10;
EC_DEVICE_STATE_INVALID_VPRS    :BYTE   :=16#20;
EC_DEVICE_STATE_INITCMD_ERROR   :BYTE   :=16#40;

(* Link states *)
EC_LINK_STATE_OK         :BYTE   :=16#00;
EC_LINK_STATE_NOT_PRESENT           :BYTE   :=16#01;
EC_LINK_STATE_LINK_WITHOUT_COMM :BYTE   :=16#02;
EC_LINK_STATE_MISSING_LINK          :BYTE   :=16#04;
EC_LINK_STATE_ADDITIONAL_LINK   :BYTE   :=16#08;
EC_LINK_STATE_PORT_A        :BYTE   :=16#10;
EC_LINK_STATE_PORT_B        :BYTE   :=16#20;
EC_LINK_STATE_PORT_C        :BYTE   :=16#40;
EC_LINK_STATE_PORT_D        :BYTE   :=16#80;

(* Device/Link state IG/IO *)
EC_ADS_IGRP_MASTER_STATEMACHINE  :UDINT :=16#00000003;
EC_ADS_IOFFS_MASTER_CURSTATE    :UDINT :=16#00000100;
EC_ADS_IOFFS_MASTER_REQSTATE    :UDINT :=16#00000101;
EC_ADS_IOFFS_MASTER_INTERNALSTATE       :UDINT :=16#00000102;

EC_ADS_IGRP_MASTER_COUNT_SLAVE  :UDINT :=16#00000006;
EC_ADS_IOFFS_MASTER_COUNT_SLAVE     :UDINT :=16#00000000;
EC_ADS_IOFFS_MASTER_COUNT_PORT  :UDINT :=16#00000001;
EC_ADS_IOFFS_MASTER_COUNT_ROUTER    :UDINT :=16#00000002;

EC_ADS_IGRP_MASTER_SLAVE_ADDRESSES      :UDINT :=16#00000007;
EC_ADS_IGRP_SLAVE_STATEMACHINE  :UDINT :=16#00000009;
EC_ADS_IGRP_MASTER_SLAVE_IDENTITY       :UDINT :=16#00000011;
EC_ADS_IGRP_MASTER_SLAVE_CRC    :UDINT :=16#00000012;

(* SoE IG/IO *)
EC_ADS_IGRP_ECAT_SOE         :UDINT :=16#0000F420;
EC_ADS_IGRP_ECAT_SOE_LASTERROR  :UDINT :=16#0000F421;

EC_SOE_ELEMENT_DATASTATE        :BYTE :=16#01;
EC_SOE_ELEMENT_NAME         :BYTE :=16#02;
EC_SOE_ELEMENT_ATTRIBUTE        :BYTE :=16#04;
EC_SOE_ELEMENT_UNIT         :BYTE :=16#08;
EC_SOE_ELEMENT_MIN          :BYTE :=16#10;
EC_SOE_ELEMENT_MAX          :BYTE :=16#20;
EC_SOE_ELEMENT_VALUE        :BYTE :=16#40;
EC_SOE_ELEMENT_DEFAULT      :BYTE :=16#80;

(* FoE IG/IO *)
EC_ADS_IGRP_FOE_FOPENREAD           :UDINT :=16#0000F401;
EC_ADS_IGRP_FOE_FOPENWRITE          :UDINT :=16#0000F402;
EC_ADS_IGRP_FOE_FCLOSE      :UDINT :=16#0000F403;
EC_ADS_IGRP_FOE_FREAD       :UDINT :=16#0000F404;
EC_ADS_IGRP_FOE_FWRITE      :UDINT :=16#0000F405;
EC_ADS_IGRP_FOE_PROGRESSINFO    :UDINT :=16#0000F406;
EC_ADS_IGRP_FOE_LASTERROR           :UDINT :=16#0000F407;

(* CoE IG/IO *)
EC_ADS_IGRP_CANOPEN_SDO         :UDINT :=16#0000F302;
EC_ADS_IGRP_CANOPEN_SDO_LASTERROR       :UDINT :=16#0000F303;


EC_DCTIME_DELTA_OFFSET  : T_ULARGE_INTEGER := ( dwHighPart := 16#01BF53EB, dwLowPart := 16#256D4000
); (* Number of 100ns ticks between 1.1.1601 and 1.1.2000 *)
```

```
EC_DCTIME_DATEDELTA_OFFSET       : DWORD := 730120; (* Number of past days since year zero until 1 J
anuary 2000 *)

EC_DCTIME_TICKSPERMSEC  : T_ULARGE_INTEGER := ( dwHighPart := 16#00000000, dwLowPart := 16#000F4240)
;(* Number of nanosecond ticks per millisecond *)
EC_DCTIME_TICKSPERSEC    : T_ULARGE_INTEGER := ( dwHighPart := 16#00000000, dwLowPart := 16#3B9ACA00
);(* Number of nanosecond ticks per second *)
EC_DCTIME_TICKSPERDAY    : T_ULARGE_INTEGER := ( dwHighPart := 16#00004E94, dwLowPart := 16#914F0000
);(* Number of nanosecond ticks per day *)
```

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

13.2 EhterCAT mailbox protocol error codes

VAR_GLOBAL CONSTANT

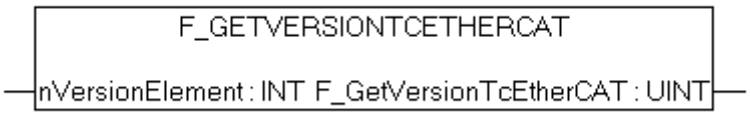
```
(* FoE mailbox protocol error codes *)
EC_FOE_PROTERR_NOTDEFINED        : UDINT := 0;
EC_FOE_PROTERR_NOTFOUND          : UDINT := 1;
EC_FOE_PROTERR_ACCESS       : UDINT := 2;
EC_FOE_PROTERR_DISKFULL          : UDINT := 3;
EC_FOE_PROTERR_ILLEAGAL          : UDINT := 4;
EC_FOE_PROTERR_PACKENO      : UDINT := 5;
EC_FOE_PROTERR_EXISTS       : UDINT := 6;
EC_FOE_PROTERR_NOUSER       : UDINT := 7;
EC_FOE_PROTERR_BOOTSTRAPONLY     : UDINT := 8;
EC_FOE_PROTERR_NOTINBOOTSTRAP    : UDINT := 9;
EC_FOE_PROTERR_INVALIDPASSWORD   : UDINT := 10;

(* CoE mailbox protocol error codes *)
EC_COE_PROTERR_TOGGLE        : UDINT := 16#05030000; (* Toggle bit not alternated. *)
EC_COE_PROTERR_TIMEOUT      : UDINT := 16#05040000; (* SDO protocol timed out. *)
EC_COE_PROTERR_CCS_SCS      : UDINT := 16#05040001; (* Client/
server command specifier not valid or unknown. *)
EC_COE_PROTERR_BLK_SIZE          : UDINT := 16#05040002; (* Invalid block size (block mode only). *)
EC_COE_PROTERR_SEQNO        : UDINT := 16#05040003; (* Invalid sequence number (block mode only). *)
EC_COE_PROTERR_CRC          : UDINT := 16#05040004; (* CRC error (block mode only). *)
EC_COE_PROTERR_MEMORY       : UDINT := 16#05040005; (* Out of memory. *)
EC_COE_PROTERR_ACCESS       : UDINT := 16#06010000; (* Unsupported access to an object. *)
EC_COE_PROTERR_WRITEONLY         : UDINT := 16#06010001; (* Attempt to read a write only object. *)
EC_COE_PROTERR_READONLY          : UDINT := 16#06010002; (* Attempt to write a read only object. *)
EC_COE_PROTERR_INDEX        : UDINT := 16#06020000; (* Object does not exist in the object dictionar
y. *)
EC_COE_PROTERR_PDO_MAP      : UDINT := 16#06040041; (* Object cannot be mapped to the PDO. *)
EC_COE_PROTERR_PDO_LEN      : UDINT := 16#06040042; (* The number and length of the objects to be ma
pped would exceed PDO length. *)
EC_COE_PROTERR_P_INCOMP          : UDINT := 16#06040043; (* General parameter incompatibility reason.
 *)
EC_COE_PROTERR_I_INCOMP          : UDINT := 16#06040047; (* General internal incompatibility in the d
evice. *)
EC_COE_PROTERR_HARDWARE          : UDINT := 16#06060000; (* Access failed due to an hardware error. *
)
EC_COE_PROTERR_DATA_SIZE         : UDINT := 16#06070010; (* Data type does not match, length of servi
ce parameter does not match *)
EC_COE_PROTERR_DATA_SIZE1           : UDINT := 16#06070012; (* Data type does not match, length of s
ervice parameter too high *)
EC_COE_PROTERR_DATA_SIZE2           : UDINT := 16#06070013; (* Data type does not match, length of s
ervice parameter too low *)
EC_COE_PROTERR_OFFSET       : UDINT := 16#06090011; (* Sub-index does not exist. *)
EC_COE_PROTERR_DATA_RANGE           : UDINT := 16#06090030; (* Value range of parameter exceeded (on
ly for write access). *)
EC_COE_PROTERR_DATA_RANGE1      : UDINT := 16#06090031; (* Value of parameter written too high. *)
EC_COE_PROTERR_DATA_RANGE2      : UDINT := 16#06090032; (* Value of parameter written too low. *)
EC_COE_PROTERR_MINMAX       : UDINT := 16#06090036; (* Maximum value is less than minimum value. *)
EC_COE_PROTERR_GENERAL      : UDINT := 16#08000000; (* general error *)
EC_COE_PROTERR_TRANSFER          : UDINT := 16#08000020; (* Data cannot be transferred or stored to t
he application. *)
```

```
EC_COE_PROTERR_TRANSFER1        : UDINT := 16#08000021; (* Data cannot be transferred or stored to t
he application because of local control. *)
EC_COE_PROTERR_TRANSFER2        : UDINT := 16#08000022; (* Data cannot be transferred or stored to t
he application because of the present device state. *)
EC_COE_PROTERR_DICTIONARY           : UDINT := 16#08000023; (* Object dictionary dynamic generation
fails or no object dictionary is present (e.g. object dictionary is generated from file and generati
on fails because of an file error). *)
```

Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v2.10.0 Build > 1307	PC or CX (x86, ARM)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)

14 F_GetVersionTcEtherCAT



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcEtherCAT : UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element to be read. Possible parameters:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are included automatically)
TwinCAT v2.10.0 Build >= 1301 or higher	CX (ARM)	

15 F_CheckVendorId

The function F_CheckVendorId returns a TRUE if the VendorID is Beckhoff otherwise it returns a FALSE.

VAR_INPUT

```
VAR_INPUT
    stSlaveIdentity : ST_EcSlaveIdentity;
END_VAR
```

stSlaveIdentity: The SlaveIdentity can be read by FB_EcGetSlaveIdentity [▶ 45].

Requirements

Development environment	Target system type	Libraries to be linked
TwinCAT v2.10.0 or higher	PC or CX (x86)	TcEtherCAT.Lib (Standard.Lib; TcBase.Lib; TcSystem.Lib, TcUtilities.Lib are automatically included)
TwinCAT v2.10.0 Build >= 1301or higher	CX (ARM)	

More Information:
www.beckhoff.com/tx1200

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com