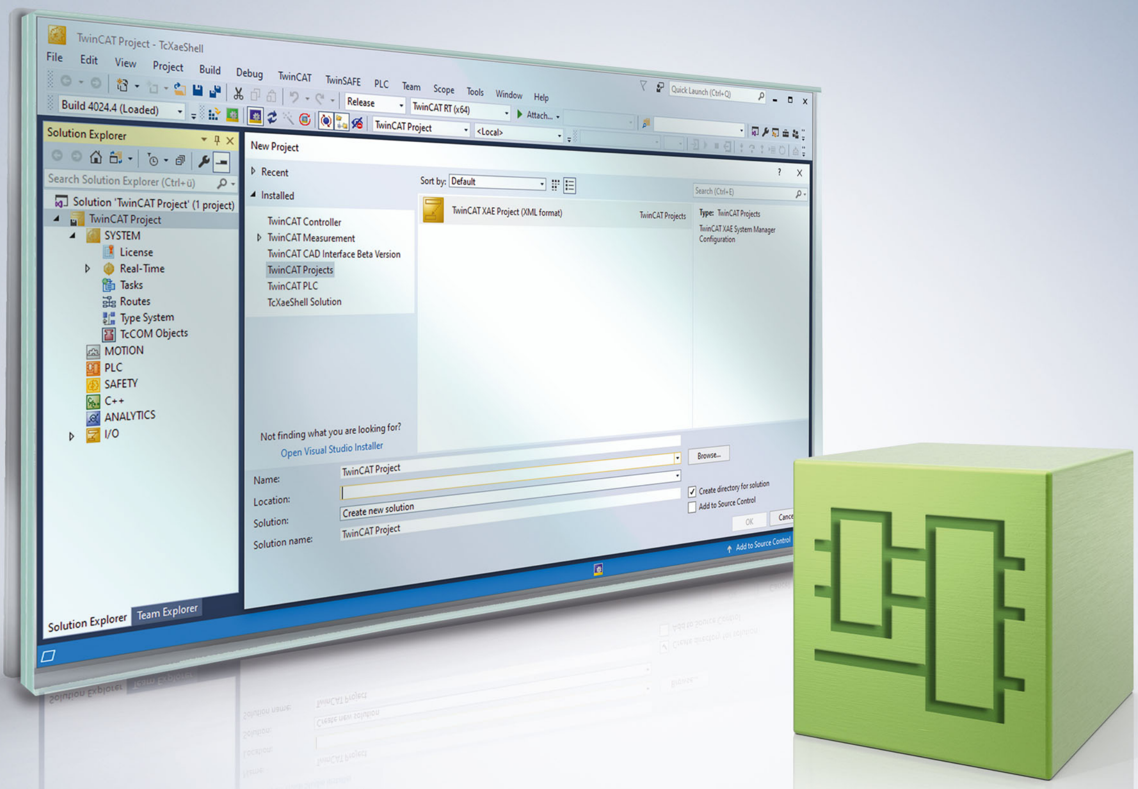
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | PLC Library: Tc2_EtherCAT



2025-06-27 | Version: 1.9.0

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.
The documentation and the following notes and explanations must be complied with when installing and commissioning the components.
The trained specialists must always use the current valid documentation.

The trained specialists must ensure that the application and use of the products described is in line with all safety requirements, including all relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been compiled 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.
Claims to modify products that have already been supplied may not be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

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

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.
The distribution and reproduction of this document, as well as the use and communication of its contents without express authorization, are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event that a patent, utility model, or design are registered.

Third-party trademarks

Trademarks of third parties may be used in this documentation. You can find the trademark notices here: https://www.beckhoff.com/trademarks.

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

ℹ 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.

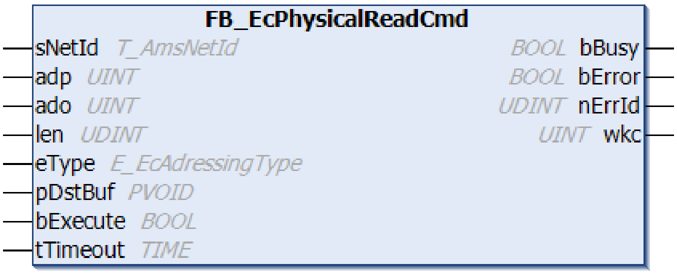
To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview

The PLC library Tc2_EtherCAT contains function blocks for executing services or functions on an EtherCAT master device and/or its slave devices.

3 EtherCAT Commands

3.1 FB_EcPhysicalReadCmd



The function block `FB_EcPhysicalReadCmd` can be used to send an EtherCAT read command (FPRD, APRD, BRD) to a particular EtherCAT slave or to all EtherCAT slaves. This command can be used by the PLC to read a register or the DPRAM of the EtherCAT slave controller.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
adp	UINT	This value determines which EtherCAT slave is to be addressed with this command. The meaning of this value depends on the addressing mode selected with eType. (See adp value)
ado	UINT	Physical memory (DPRAM) or register to be read.
len	UDINT	Number of bytes to be read.
eType	E_EcAdressingType	Different EtherCAT commands are sent, depending on value of eType. (See eType)
pDstBuf	PVOID	The address (pointer) of the receive buffer.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

adp value

This value determines which EtherCAT slave is to be addressed with this command. The meaning of this value depends on the addressing mode selected with eType:

eType	Description
eAdressingType_Fixed	The slave is addressed by means of its configured EtherCAT address. These EtherCAT addresses can be read via the function block FB_EcGetAllSlaveAddr.
eAdressingType_AutoInc	The slave is addressed based on its position in the ring. The first device has the address 0 (adp=0); adp is decremented by one for all subsequent slaves: 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. adp can be set to 0.

eType

Different EtherCAT commands are sent, depending on value of eType:

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

The individual commands differ in terms of addressing mode (see adp).

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
wkc	UINT	The working counter is incremented by each EtherCAT slave that has processed this command successfully. If only one EtherCAT slave was addressed by this command, this value should therefore be 1.

Example of an implementation in ST:

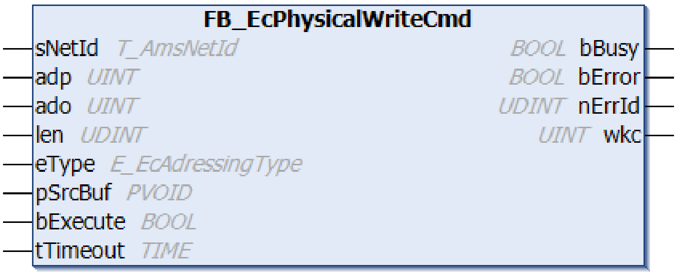
```
PROGRAM TEST_PhysicalReadCmd
VAR
    fbReadCmd  : FB_EcPhysicalReadCmd;
    bExecute   : BOOL;
    value      : UINT;
    adp        : UINT:=16#3E9;
    ado        : UINT:=16#1100;
    eType      : E_EcAdressingType := eAdressingType_Fixed;
    sNetId     : T_AmsNetId:='192.168.1.5.3.1';
    wkc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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

3.2 FB_EcPhysicalWriteCmd



The function block `FB_EcPhysicalWriteCmd` can be used to send an EtherCAT write command (FPWR, APWR, BWR) to a particular EtherCAT slave or to all EtherCAT slaves. This command can be used by the PLC to write to a register or the DPRAM of the EtherCAT slave controller.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
adp	UINT	This value determines which EtherCAT slave is to be addressed with this command. The meaning of this value depends on the addressing mode selected with eType. (See adp value)
ado	UINT	Physical memory (DPRAM) or register to be read.
len	UDINT	Number of bytes to be written.
eType	E_EcAdressingType	Different EtherCAT commands are sent, depending on the value of eType: (See eType)
pSrcBuf	PVOID	Address (pointer) of the transmit buffer.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

adp value

This value determines which EtherCAT slave is to be addressed with this command. The meaning of this value depends on the addressing mode selected with eType:

eType	Description
eAdressingType_Fixed	The slave is addressed by means of its configured EtherCAT address. These EtherCAT addresses can be read via the function block FB_EcGetAllSlaveAddr.
eAdressingType_AutoInc	The slave is addressed based on its position in the ring. The first device has the address 0 (adp=0); adp is decremented by one for all subsequent slaves: 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. adp should be set to 0.

eType

Different EtherCAT commands are sent, depending on the value of eType:

eType	Command
eAdressingType_Fixed	Configured Address Physical Write (FPWR)
eAdressingType_AutoInc	Auto Increment Physical Write (APWR)
eAdressingType_BroadCAST	Broadcast Write (BWR)

The individual commands differ in terms of addressing mode (see adp).

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
wkc	UINT	The working counter is incremented by each EtherCAT slave that has processed this command successfully. If only one EtherCAT slave was addressed by this command, this value should therefore be 1.

Example of an implementation in ST:

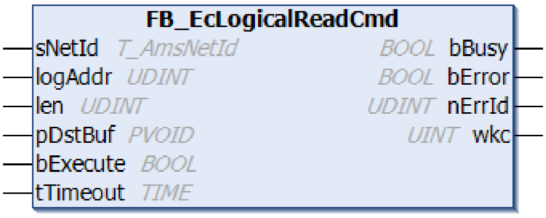
```
PROGRAM Test_PhysicalWriteCmd
VAR
    fbWriteCmd  : FB_EcPhysicalWriteCmd;
    bExecute    : BOOL;
    value       : UINT :=16#5555;
    adp         : UINT:=16#3E9;
    ado         : UINT:=16#1100;
    eType       : E_EcAdressingType := eAdressingType_Fixed;
    sNetId      : T_AmsNetId:='192.168.1.5.3.1';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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

3.3 FB_EcLogicalReadCmd



The master sends a logical EtherCAT read command (LRD) with the function block `FB_EcLogicalReadCmd`. In each slave, local address ranges (DPRAM) can be mapped to global logical address ranges. This command therefore addresses all EtherCAT slaves, which have mapping configured for the selected logical address range.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
logAddr	UDINT	Logical address
len	UDINT	Number of bytes to be read
pDstBuf	PVOID	Address (pointer) to the receive buffer
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
wkc	UINT	The working counter is incremented by each EtherCAT slave that has processed this command successfully. If only one EtherCAT slave was addressed by this command, this value should therefore be 1.

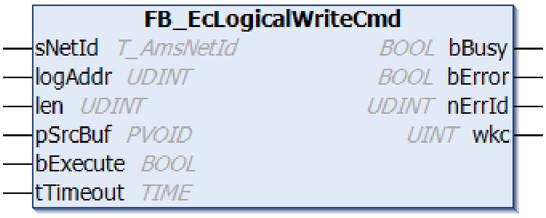
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

3.4 FB_EcLogicalWriteCmd



The master sends a logical EtherCAT write command (LWR) with the function block FB_EcLogicalWriteCmd. In each slave, local address ranges (DPRAM) can be mapped to global logical address ranges. This command therefore addresses all EtherCAT slaves, which have mapping configured for the selected logical address range.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
logAddr	UDINT	Logical address
len	UDINT	Number of bytes to be written
pSrcBuf	PVOID	Address (pointer) to the transmit buffer
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
wkc	UINT	The working counter is incremented by each EtherCAT slave that has processed this command successfully. If only one EtherCAT slave was addressed by this command, this value should therefore be 1.

Example of an implementation in ST:

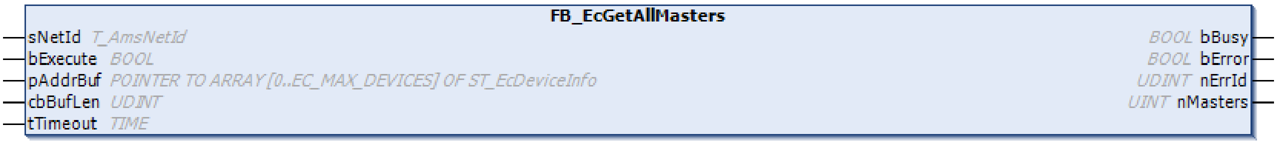
```
PROGRAM Test_LogicalWriteCmd
VAR
    fbWriteCmd : FB_EcLogicalWriteCmd;
    bExecute   : BOOL;
    value      : USINT :=16#55;
    logAddr    : UDINT :=16#10000;
    sNetId     : T_AmsNetId:='192.168.1.5.3.1';
    wkc        : UINT;
    bError     : BOOL;
    nErrId     : UDINT;
END_VAR

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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4 EtherCAT Diagnostic

4.1 FB_EcGetAllMasters



FB_EcGetAllMasters

sNetId *T_AmsNetId*
bExecute *BOOL*
pAddrBuf *POINTER TO ARRAY [0..EC_MAX_DEVICES] OF ST_EcDeviceInfo*
cbBufLen *UDINT*
tTimeout *TIME*

BOOL bBusy
BOOL bError
UDINT nErrId
UINT nMasters

The function block `FB_EcGetAllMasters` can be used to read information from all EtherCAT masters. If the call is successful, the buffer transferred in the pAddrBuf parameter contains the information (Device ID, AMS Net ID, name) of all masters as an array of ST_EcDeviceInfo.

If the buffer is too small for the information of all existing masters, the buffer is filled up to the specified size and the output nMasters outputs the total number of all existing masters.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the IPC on which the EtherCAT masters are to be read. default = local host
bExecute	BOOL	The function block is enabled by a positive edge at this input.
pAddrBuf	POINTER TO ARRAY [0..EC_MAX_DEVICES] OF ST_EcDeviceInfo	Address of an array of ST_EcDeviceInfo, in which the information of the individual masters is to be written. ST_EcDeviceInfo contains the Device ID, AMS Net ID and the name of an EtherCAT master.
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read. The size of the array of ST_EcDeviceInfo assigned to pAddrBuf must be specified here.
tTimeout	TIME	Maximum time that must not be exceeded during the execution of the ADS calls.

Outputs

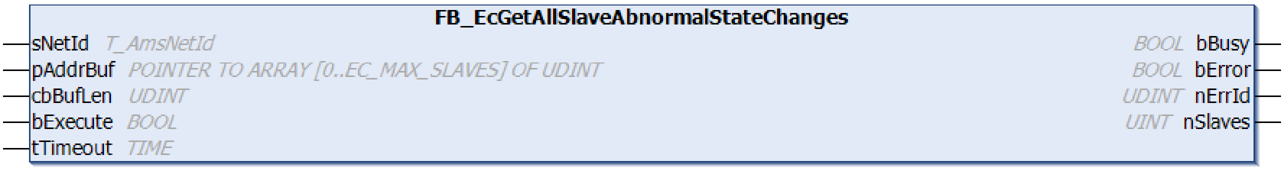
```
VAR_OUTPUT
    bBusy    : BOOL;
    bError   : BOOL;
    nErrId   : UDINT;
    nMasters : UINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set. Example: Error 1798 (0x706) indicates a null pointer at the buffer address.
nMasters	UINT	Number of EtherCAT masters present.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT >= v3.1.4024.62	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT >= v3.6.1.0

4.2 FB_EcGetAllSlaveAbnormalStateChanges



The function block `FB_EcGetAllSlaveAbnormalStateChanges` can be used to read the unexpected EtherCAT state changes of all the slaves connected to the master. If the call is successful, the buffer transferred in the parameter pBufAddr contains the number of unexpected state changes of all slaves as an array of UDINTs. EtherCAT state changes are unexpected if they were not requested by the EtherCAT master, e.g. if an EtherCAT slave spontaneously switches from OP state to SAFEOP state.

Inputs

```
VAR_INPUT
    sNetId   :  T_AmsNetId; (*AmsNetId of the EtherCAT master  device*)
    pAddrBuf :  POINTER TO ARRAY [0..EC_MAX_SLAVES] OF UDINT;
    (*Contains the address of the buffer the counters for the state changes f.i. Op to SafeOp-
Err are copied to.*)
    cbBufLen :  UDINT; (*Size of the buffer pAddrBuf.     The size of the buffer must be at least nS
lave *4 Bytes *)
    bExecute :  BOOL; Function Block execution is triggered by a rising    edge at this input*)
    tTimeout :  TIME; (*States the time before the function is cancelled.*)
END_VAR
```

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
pAddrBuf	POINTER TO ARRAY [0..EC_MAX_SLAVES] OF UDINT	Address of an array of UDINTs, into which the number of unexpected state changes of the individual slaves is to be written.
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

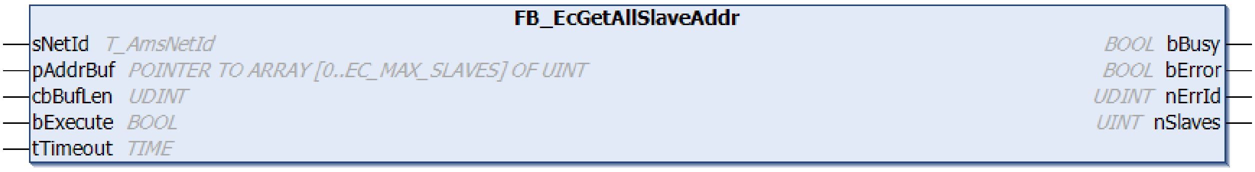
```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    nSlaves : UINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set. Error 1798 (0x706) indicates a null pointer at the buffer address. Error 1797 (0x705) indicates inadequate buffer size.
nSlaves	UINT	The number of slaves connected to the master.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.3 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.

Inputs

```
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
```

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
pAddrBuf	POINTER TO ARRAY [0..EC_MAX_SLAVES] OF UINT	Address of an array of UINTs into which the addresses of the individual slaves are to be written.
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set. Error 1798 (0x706) indicates a null pointer at the buffer address. Error 1797 (0x705) indicates inadequate buffer size.
nSlaves	UINT	The number of slaves connected to the master.

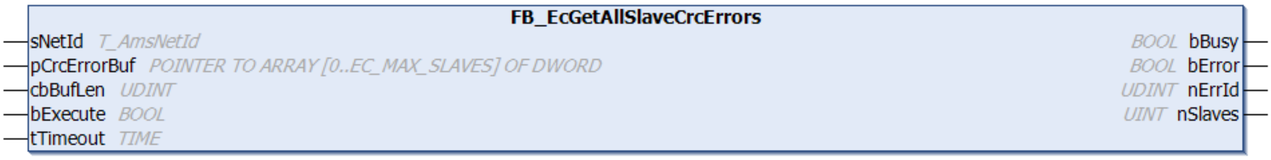
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.4 FB_EcGetAllSlaveCrcErrors



The `FB_EcGetAllSlaveCrcErrors` function block allows the CRC error counters of all the slaves connected to the master to be read. The CRC errors at the individual ports of a slave are added.

In order to read the CRC errors of the individual ports (A, B and C) of a slave, it is necessary to call the FB_EcGetSlaveCrcError [▶ 30] function block.

In order to read the CRC errors of the individual ports (A, B, C and D) of a slave, it is necessary to call the FB_EcGetSlaveCrcErrorEx [▶ 31] function block.

Inputs

```
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
```

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
pCrcErrorBuf	POINTER TO ARRAY [0..EC_MAX_SLAVES] OF DWORD	The address of an array of DWORDs into which the CRC error counter is to be written.
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set. Error 1798 (0x706) indicates a null pointer at the buffer address. Error 1797 (0x705) indicates inadequate buffer size.
nSlaves	UINT	The number of slaves connected to the master.

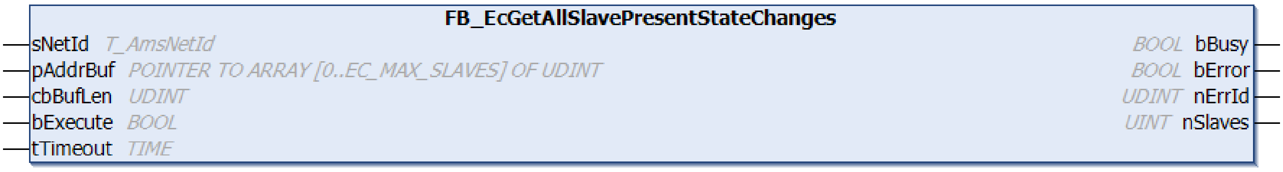
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.5 FB_EcGetAllSlavePresentStateChanges



The function block `FB_EcGetAllSlavePresentStateChanges` can be used to read the EtherCAT state changes from state “slave is present” to “INIT_NO_COMM” of all slaves connected to the master. If the call is successful, the buffer transferred in the parameter pBufAddr contains the number of state changes of all slaves as an array of UDINTs. The EtherCAT state change from state “slave is present” to “INIT_NO_COMM” means that the connection to the slave has been interrupted. For example by disconnecting the EtherCAT cable.

Inputs

```
VAR_INPUT
    sNetId    :  T_AmsNetId; (*AmsNetId of the EtherCAT master device*)
    pAddrBuf  :  POINTER TO ARRAY [0..EC_MAX_SLAVES] OF UDINT; (*Contains the address of the buffer
the counters for the state changes from Present to INIT_NO_COMM are copied to.*)
    cbBufLen  :  UDINT; (*Size of the buffer pAddrBuf. The size of the buffer must be at least nSlav
e *4 Bytes *)
    bExecute  :  BOOL; (*Function Block execution is triggered by a rising edge at this input*)
    tTimeout  :  TIME; (*States the time before the function is cancelled.*)
END_VAR
```

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
pAddrBuf	POINTER TO ARRAY [0..EC_MAX_SLAVES]OF UDINT	Address of an array of UDINTs, into which the number of state changes from "slave is present" to INIT_NO_COMM for the individual slaves is to be written.
bBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

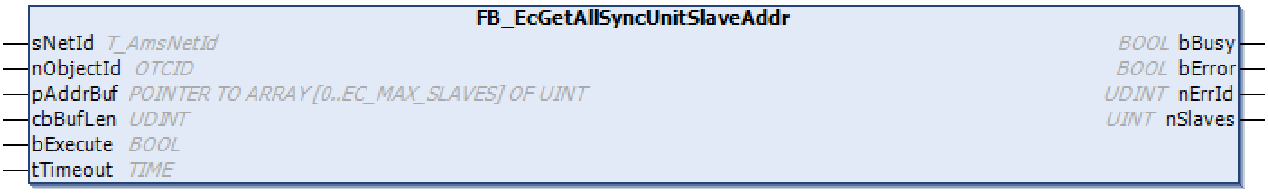
```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    nSlaves : UINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set. Error 1798 (0x706) indicates a null pointer at the buffer address. Error 1797 (0x705) indicates inadequate buffer size.
nSlaves	UINT	The number of slaves connected to the master.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.6 FB_EcGetAllSyncUnitSlaveAddr



The FB_EcGetAllSyncUnitSlaveAddr 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.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	The network address of the TwinCAT computer on which the function block is to be executed can be specified here. An empty string may be specified for the local computer.
nObjectId	OTCID	Object Id of the Sync Unit to be used, see Sync Unit
pAddrBuf	POINTER TO ARRAY [0..EC_MAX_SLAVES] OF UINT	Address of an array of UINTs into which the addresses of the individual slaves are to be written.
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read.
bExecute	BOOL	The function block is enabled by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

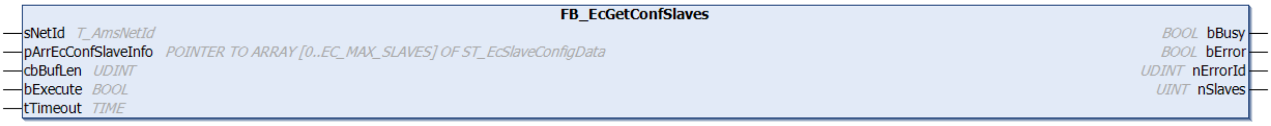
```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    nSlaves : UINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set. Error 1798 (0x706) indicates a null pointer at the buffer address. Error 1797 (0x705) indicates inadequate buffer size.
nSlaves	UINT	The number of slaves connected to the master.

Requirements

Development Environment	Target platform	PLC libraries to include
TwinCAT v3.2.4024.14	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT >= 3.3.17.0

4.7 FB_EcGetConfSlaves



The function block `FB_EcGetConfSlaves` can be used to read a list of configured slaves from the EtherCAT master object directory.

Inputs

```
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
```

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
pArrEcConfSlaveInfo	POINTER TO ARRAY [0..EC_MAX_SLAVES] OF ST_EcSlaveConfigData	Address of an array of structures of type ST_EcSlaveConfigData [▶ 123], into which data of each configured slave are to be written.
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

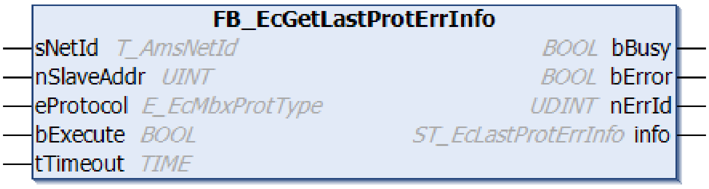
```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    nSlaves : UINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set. Error 1798 (0x706) indicates a null pointer at the buffer address. Error 1797 (0x705) indicates inadequate buffer size.
nSlaves	UINT	Returns the number of configured slaves.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.8 FB_EcGetLastProtErrInfo



The function block `FB_EcGetLastProtErrInfo` can be used to read additional error information relating to the most recent mailbox protocol error. An error-free mailbox command resets the last error every time.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave, whose error information is to be read.
eProtocol	E_EcMbxProtType	EtherCAT mailbox protocol type [▶ 121]
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
info	ST_EcLastProtErrInfo	Structure with additional error information [▶ 122]

Sample in ST:

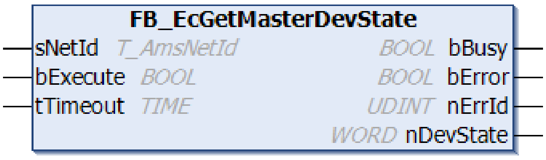
A rising edge at bGet triggers reading of additional error information relating to the most recent mailbox protocol error.

```
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 );
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.9 FB_EcGetMasterDevState



The function block `FB_EcGetMasterDevState` can be used to read the current state of the EtherCAT master.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

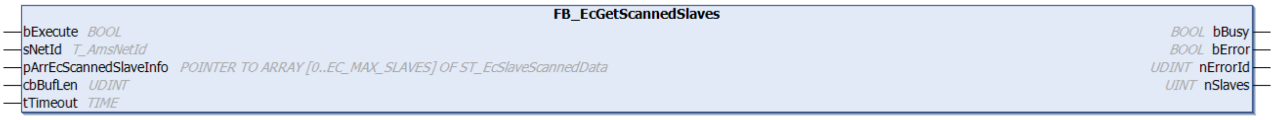
```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    nDevState  : WORD;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
nDevState	WORD	Current state of the master device

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.10 FB_EcGetScannedSlaves



The function block `FB_EcGetScannedSlaves` can be used to read a list of the currently available (scanned) slaves from the EtherCAT master object directory. To this end an online scan is executed, during which the EEPROMs of the EtherCAT slaves are read. The scanning process may take some time, depending on the number of connected slaves.

Inputs

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

Name	Type	Description
bExecute	BOOL	The function block is activated by a positive edge at this input.
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
pArrEcScannedSlaveInfo	POINTER TO ARRAY[0..EC_MAX_SLAVES] OF ST_EcSlaveScannedData	Address of an array of structures of type ST_EcSlaveScannedData [▶ 124], to which the data for each scanned slave are to be written.
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

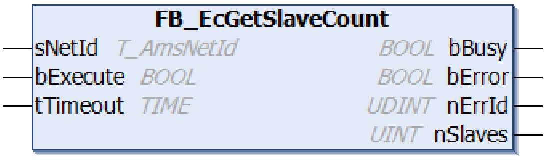
```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    nSlaves : UINT;
ND_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set. Error 1798 (0x706) indicates a null pointer at the buffer address. Error 1797 (0x705) indicates inadequate buffer size.
nSlaves	UINT	Returns the number of scanned slaves.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.11 FB_EcGetSlaveCount



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

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
nSlaves	UINT	The number of slaves connected to the master

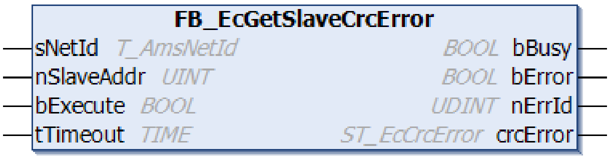
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.12 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 (e.g. EK1100). The function block `FB_EcGetSlaveCrcErrorEx` can also be used with slaves with up to 4 ports (e.g. EK1122).

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave whose CRC error counter is to be read.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
crcError	ST_EcCrcError	CRC error [▶ 121] counters for the individual ports

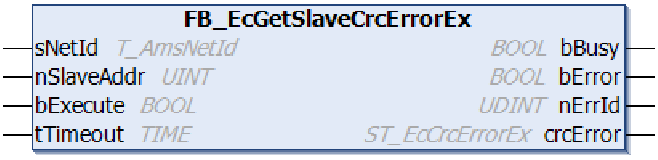
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.13 FB_EcGetSlaveCrcErrorEx



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

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

Inputs

```
VAR_INPUT
    sNetId      :  T_AmsNetId; (*AmsNetId of the EtherCAT master device*)
    nSlaveAddr  :  UINT; (*Address of the slave device*)
    bExecute    :  BOOL; (*Function block execution is triggered by a rising edge at this input.*)
    tTimeout    :  TIME; (*States the time before the function is cancelled.*)
END_VAR
```

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device (type: T_AmsNetId).
nSlaveAddr	UINT	Fixed address of the EtherCAT slaves whose CRC error counter is to be read.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

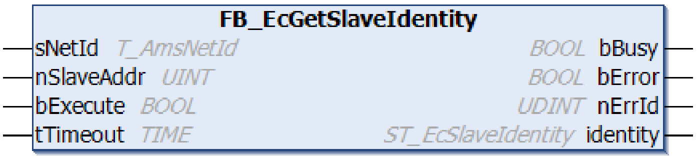
```
VAR_OUTPUT
    bBusy     :  BOOL;
    bError    :  BOOL;
    nErrId    :  UDINT;
    CrcError  :  ST_EcCrcErrorEx; (*Crc error of the EtherCAT slave device*)
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
CrcError	ST_EcCrcErrorEx	CRC error counter of the individual ports (type: ST_EcCrcErrorEx [▶ 122])

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.14 FB_EcGetSlaveIdentity



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

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
identity	ST_EcSlaveIdentity	CANopen Identity [▶ 124] of the EtherCAT device

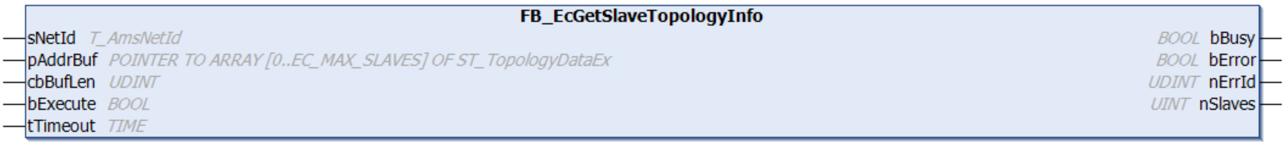
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.15 FB_EcGetSlaveTopologyInfo



The function block `FB_EcGetSlaveTopolgyInfo` can be used to determine topology information.

Inputs

```
VAR_INPUT
    sNetId    :  T_AmsNetId; (*AmsNetId of the EtherCAT master device*)
    pAddrBuf  :  POINTER TO ARRAY [0..EC_MAX_SLAVES] OF ST_TopologyDataEx; (*Contains the address of
 the buffer the topology data are copied to.*)
    cbBufLen  :  UDINT; (*Size of the buffer pAddrBuf. The size of the buffer must be at least nSlav
e * 64 Bytes*)
    bExecute  :  BOOL; (*Function block execution is triggered by a rising edge at this input*)
    tTimeout  :  TIME; (*States the time before the function is cancelled*)
END_VAR
```

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
pAddrBuf	POINTER TO ARRAY [0..EC_MAX_SLAVES] OF ST_TopologyDataEx	Address of an array of structures of type ST_TopologyDataEx [⏵ 129], which contains the topology data.
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

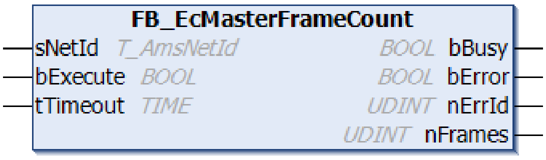
```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    nSlaves : UINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set. Error 1798 (0x706) indicates a null pointer at the buffer address. Error 1797 (0x705) indicates inadequate buffer size.
nSlaves	UINT	The number of slaves connected to the master.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.16 FB_EcMasterFrameCount



The function block FB_EcMasterFrameCount can be used to determine the number of EtherCAT frames configured in the master.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

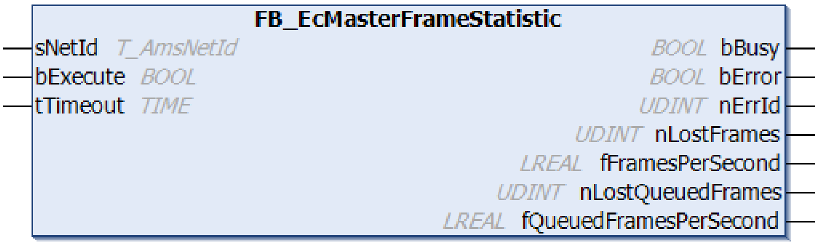
```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    nFrames : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
nFrames	UDINT	Number of EtherCAT frames

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.17 FB_EcMasterFrameStatistic



The function block `FB_EcMasterFrameStatistic` can be used to read the frame statistics of the EtherCAT master. A distinction is made between cyclic and acyclic (queued) frames. Acyclic frames are used for the initialization or for parameter access to EtherCAT slaves. Frames are regarded as lost if they fail to return to the master or are invalid.

The number of lost frames (i.e. lost or invalid cyclic frames), the number of cyclic frames per second, the number of lost queued frames (i.e. lost or invalid acyclic frames) and the number of queued frames per second is provided at the function block output.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

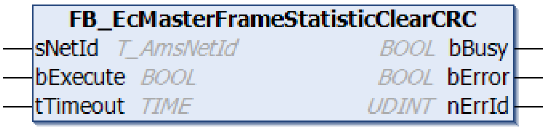
```
VAR_OUTPUT
    bBusy                   : BOOL;
    bError                  : BOOL;
    nErrId                  : UDINT;
    nLostFrames             : UDINT;
    fFramesPerSecond        : LREAL;
    nLostQueuedFrames       : UDINT;
    fQueuedFramesPerSecond : LREAL;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
nLostFrames	UDINT	Returns the current number of lost or invalid cyclic frames.
fFramesPer Second	LREAL	Returns the current number of cyclic frames per second.
nLostQueuedFra mes	UDINT	Returns the current number of lost or invalid queued (acyclic) frames.
fQueuedFramesP erSecond	LREAL	Returns the current number of queued (acyclic) frames per second.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.18 FB_EcMasterFrameStatisticClearCRC



The function block `FB_EcMasterFrameStatisticClearCRC` can be used to delete the CRC error counters of all EtherCAT slaves.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

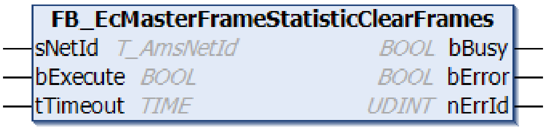
```
VAR_OUTPUT
    bBusy  : BOOL;
    bError : BOOL;
    nErrId : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.19 FB_EcMasterFrameStatisticClearFrames



The function block `FB_EcMasterFrameStatisticClearFrames` can be used to delete the lost frame counters.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

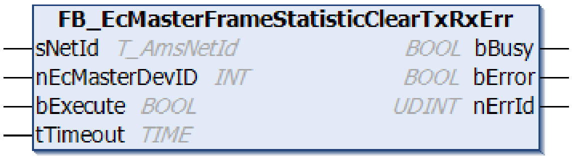
```
VAR_OUTPUT
    bBusy  : BOOL;
    bError : BOOL;
    nErrId : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.20 FB_EcMasterFrameStatisticClearTxRxErr



The function block `FB_EcMasterFrameStatisticClearTxRxErr` can be used to delete the error counters of the miniport driver of the network card.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the CPU (PC). (Type: T_AMSNetId)
nEcMasterDevID	INT	Device ID of the EtherCAT master.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

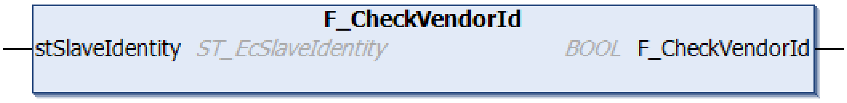
```
VAR_OUTPUT
    bBusy  : BOOL;
    bError : BOOL;
    nErrId : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.21 F_CheckVendorId



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

Return value

```
METHOD F_CheckVendorId : BOOL
```

Name	Type	Description
F_CheckVendorId	BOOL	TRUE if the VendorID is Beckhoff, otherwise FALSE.

Inputs

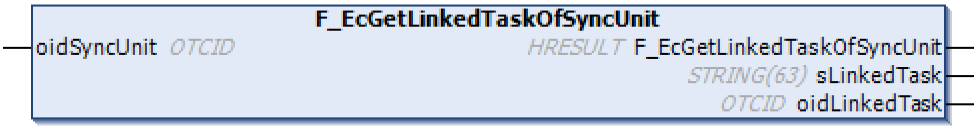
```
VAR_INPUT
    stSlaveIdentity : ST_EcSlaveIdentity;
END_VAR
```

Name	Type	Description
stSlaveIdentity	ST_EcSlaveIdentity	Slave Identity, which can be read with FB_EcGetSlaveIdentity [▶ 32].

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

4.22 F_EcGetLinkedTaskOfSyncUnit


F_EcGetLinkedTaskOfSyncUnit
oidSyncUnit OTCID
HRESULT F_EcGetLinkedTaskOfSyncUnit
STRING(63) sLinkedTask
OTCID oidLinkedTask

The name and object ID of the linked task of an EtherCAT Sync Unit can be read with this function. The return value of the function signals whether the call was successful and outputs the corresponding error code in case of an error.

Return value

```
METHOD F_EcGetLinkedTaskOfSyncUnit : HRESULT
```

Name	Type	Description
F_EcGetLinkedTaskOfSyncUnit	HRESULT	Signals whether the call was successful and outputs the corresponding error code in the event of an error.

Inputs

```
VAR_INPUT
    oidSyncUnit : OTCID; // object ID of sync unit
END_VAR
```

Name	Type	Description
oidSyncUnit	OTCID	The object ID of the Sync Unit is specified at this input. This can be found in the process image of the EtherCAT master.

Outputs

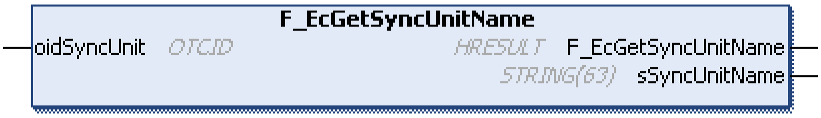
```
VAR_OUTPUT
    sLinkedTask   : STRING;
    oidLinkedTask : OTCID; // object ID of linked task
END_VAR
```

Name	Type	Description
sLinkedTask	STRING	Returns the name of the linked task.
oidLinkedTask	OTCID	Returns the object ID of the linked task.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.22	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT >= 3.3.17.0

4.23 F_EcGetSyncUnitName



The name of an EtherCAT Sync Unit can be read via its object ID using this function. The return value of the function signals whether the call was successful and outputs the relevant error code in the event of an error.

Return value

```
METHOD F_EcGetSyncUnitName : HRESULT
```

Name	Type	Description
F_EcGetSyncUnitName	HRESULT	Signals whether the call was successful and outputs the relevant error code in the event of an error.

Inputs

```
VAR_INPUT
    oidSyncUnit : OTCID; // object ID of sync unit
END_VAR
```

Name	Type	Description
oidSyncUnit	OTCID	The object ID of the Sync Unit is specified at this input. This can be found in the process image of the EtherCAT master.

Outputs

```
VAR_OUTPUT
    sSyncUnitName : STRING(63);
END_VAR
```

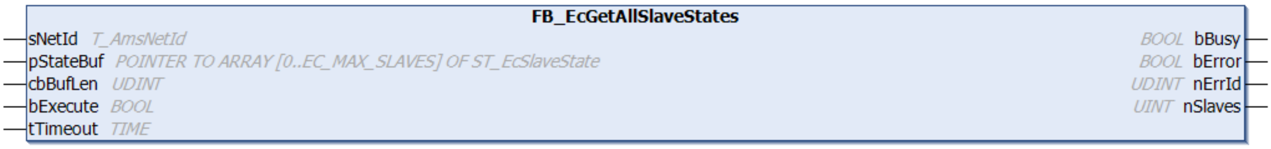
Name	Type	Description
sSyncUnitName	STRING	Returns the name of the sync unit.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.48	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT >= 3.4.2.0

5 EtherCAT State Machine

5.1 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.

Inputs

```
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
```

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
pStateBuf	POINTER TO ARRAY[0..EC_MAX_SLAVES] OF ST_EcSlaveState	The address of an array of ST_EcSlaveStates [▶ 126] into which the slave states are to be written.
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set. Error 1798 (0x706) indicates a null pointer at the buffer address. Error 1797 (0x705) indicates inadequate buffer size.
nSlaves	UINT	The number of slaves connected to the master

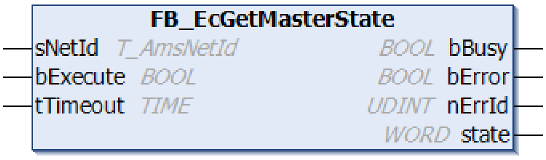
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

5.2 FB_EcGetMasterState



The function block `FB_EcGetMasterState` can be used to read the EtherCAT state of the master. If the call is successful, the State output variable of type WORD contains the requested status information.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
state	WORD	Current EtherCAT state of the master. (See State)

state

Current EtherCAT state of the master. The possible values are:

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

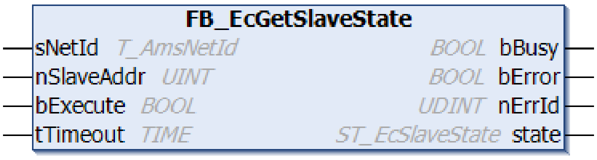
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

5.3 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.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave whose state is to be read.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
state	ST_EcSlaveState	Structure that contains the EtherCAT status and the Link status of the slave. (Type: ST_EcSlaveState [▶ 126])

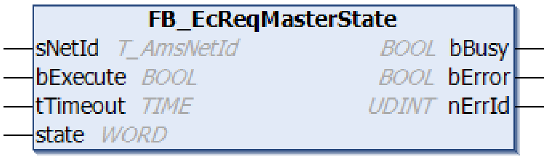
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

5.4 FB_EcReqMasterState



With this function block the EtherCAT state of a master device can be requested and set. The requested EtherCAT state is transferred in the state variable. The function block becomes inactive as soon as it has requested the EtherCAT state. Unlike the function block FB_EcSetMasterState it does not wait until the new state is set.

See also: FB_EcSetMasterState [▶ 48]

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.
state	WORD	EtherCAT state requested from the master. (See state)

State

EtherCAT state requested from the master. The possible State values are:

Constant	Value	Description
EC_DEVICE_STATE_INIT	0x01	Request Init state from master
EC_DEVICE_STATE_PREOP	0x02	Request Pre-operational state from master
EC_DEVICE_STATE_SAFEOP	0x04	Request Safe-operational state from master
EC_DEVICE_STATE_OP	0x08	Request Operational state from master

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.

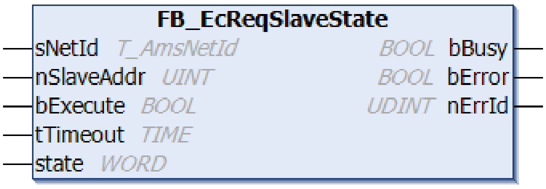
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

5.5 FB_EcReqSlaveState



With this function block a slave can be set to a specified EtherCAT state. The requested EtherCAT state is transferred in the state variable. The function block becomes inactive as soon as it has sent the command to change state. Unlike the function block FB_EcSetSlaveState it does not wait until the EtherCAT slave has attained the new state.

See also: FB_EcSetSlaveState [▶ 49]

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave whose EtherCAT state is to be set.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.
state	WORD	EtherCAT state requested from the master. (See State)

State

EtherCAT state to which the slave is to be set. The possible State values are:

Constant	Value	Description
EC_DEVICE_STATE_INIT	0x01	Set slave to Init state
EC_DEVICE_STATE_PREOP	0x02	Set slave to Pre-operational state
EC_DEVICE_STATE_BOOTSTRAP	0x03	Set slave to Bootstrap state. This state is used for firmware downloads.
EC_DEVICE_STATE_SAFEOP	0x04	Set slave to Safe-operational state
EC_DEVICE_STATE_OP	0x08	Set slave to Operational state
EC_DEVICE_STATE_ERROR	0x10	If the error bit in the status byte is set in the EtherCAT slave (state.deviceState & EC_DEVICE_STATE_ERROR = TRUE), the error bit can be reset by setting EC_DEVICE_STATE_ERROR.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.

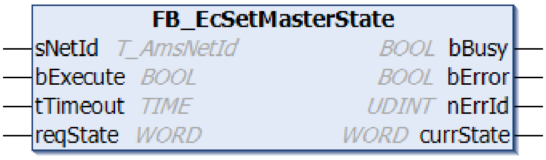
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

5.6 FB_EcSetMasterState



With this function block the EtherCAT state of a master device can be requested and set. The requested EtherCAT state is transferred with the reqState variable. The function block requests the EtherCAT state and, unlike the function block FB_EcReqMasterState, remains active until the new state is set or the maximum time tTimeout is exceeded. The current state is output in the currState variable.

See also: FB_EcReqMasterState [▶ 45]

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.
reqState	WORD	(See reqState)

reqState

EtherCAT state requested from the master. The possible values for reqState are:

Constant	Value	Description
EC_DEVICE_STATE_INIT	0x01	Request Init state from master
EC_DEVICE_STATE_PREOP	0x02	Request Pre-operational state from master
EC_DEVICE_STATE_SAFEOP	0x04	Request Safe-operational state from master
EC_DEVICE_STATE_OP	0x08	Request Operational state from master

Outputs

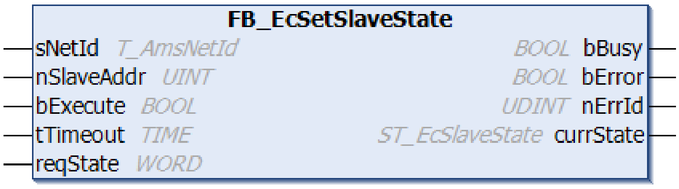
```
VAR_OUTPUT
    bBusy     : BOOL;
    bError    : BOOL;
    nErrId    : UDINT;
    currState : WORD;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
currState	WORD	Current EtherCAT state of the master

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

5.7 FB_EcSetSlaveState



With this function block a slave can be set to a specified EtherCAT state. The requested EtherCAT state is transferred with the reqState variable. The function block sends the command to change state and, unlike the function block FB_EcRegSlaveState, remains active until the EtherCAT slave has attained the new state or the maximum time tTimeout is exceeded. The current state is output in the currState variable.

See also: FB_EcReqSlaveState [▶ 47]

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave whose EtherCAT state is to be set.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.
reqState	WORD	EtherCAT state to which the slave is to be set. (See reqState)

reqState

EtherCAT state to which the slave is to be set. The possible values for reqState are:

Constant	Value	Description
EC_DEVICE_STATE_INIT	0x01	Set slave to Init state
EC_DEVICE_STATE_PREOP	0x02	Set slave to Pre-operational state
EC_DEVICE_STATE_BOOTSTRAP	0x03	Set slave to Bootstrap state. This state is used for firmware downloads.
EC_DEVICE_STATE_SAFEOP	0x04	Set slave to Safe-operational state
EC_DEVICE_STATE_OP	0x08	Set slave to Operational state
EC_DEVICE_STATE_ERROR	0x10	If the error bit in the status byte is set in the EtherCAT slave (currState.deviceState AND EC_DEVICE_STATE_ERROR = TRUE), the error bit can be reset by setting EC_DEVICE_STATE_ERROR.

Outputs

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

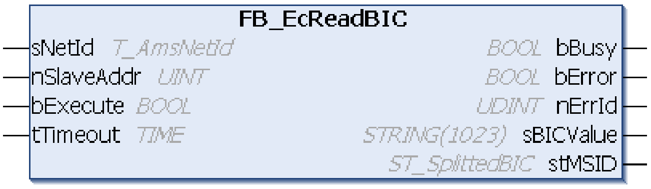
Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
currState	ST_EcSlaveState	Current EtherCAT state [▶ 126] of the slave

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

6 ADS Interface

6.1 FB_EcReadBIC



The function block FB_EcReadBIC can be used to read the BIC from the EEPROM of an EtherCAT slave via ADS. For this the slave must contain the BIC in the EEPROM.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave to which the SDO upload command should be sent.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    sBICValue  : STRING(1023)
    stMSID     : ST_SplittedBIC
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
sBICValue	STRING(1023)	This output contains (after an error-free run) the BIC of the EtherCAT slave after the bBusy output has been reset, e.g. "„1P193995SBTN0002agdw1KEL7411 Q1 2P112104020018".

stMSID	ST_SplittedBIC	This output contains (after an error-free run) the substrings of the BIC of the EtherCAT slave after the bBusy output has been reset.
		The following substrings are assigned for the above BIC:
		sItemNo = "193995"
		sBTN = "0002agdw"
		sDescription = "EL7411"
		sQuantity = "1"
		sBatchNo = "112104020018"

Sample of an implementation in ST

```
PROGRAM TEST_EcReadBIC
VAR
    fbEcBIC     : FB_EcReadBIC;
    sNetId      : T_AmsNetId := '172.16.2.131.2.1';
    bExecute    : BOOL := TRUE;
    nSlaveAddr  : UINT := 1006;
    sECBIC      : STRING(1023);
    stEcBIC     : ST_SplittedBIC;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR

fbEcBIC(sNetId:= sNetID, nSlaveAddr:= nPort, bExecute:= bExecute, tTimeout:= T#5s);;
IF NOT fbEcBIC.bBusy THEN
     bExecute := FALSE;
     IF NOT fbEcBIC.bError THEN
          stEcBIC    := fbEcBIC.stMSID;
          sECBIC     := fbEcBIC.sBICValue;
     END_IF
     fbEcBIC (bExecute:= bExecute);
END_IF
```

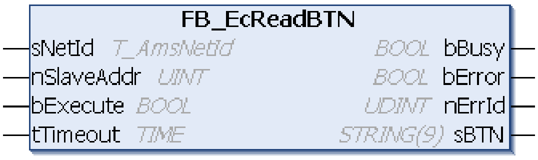
Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

Also see about this

FB_EcCoeSdoReadEx [▶ 56]

6.2 FB_EcReadBTN



The function block FB_EcReadBTN can be used to read the BTN from the EEPROM of an EtherCAT slave via ADS. For this the slave must contain the BIC in the EEPROM.

Inputs

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

Name	Type▲	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave to which the SDO upload command should be sent.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

```
VAR_OUTPUT
    bBusy  : BOOL;
    bError : BOOL;
    nErrId : UDINT;
    sBTN    : STRING()9
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
sBTN	STRING()9	This output contains (after an error-free run) the BTN of the EtherCAT slave after the bBusy output has been reset, e.g. "0002agdw".

Sample of an implementation in ST

```
PROGRAM TEST_EcReadBTN
VAR
    fbEcBTN     : FB_EcReadBTN;
    sNetId      : T_AmsNetId := '172.16.2.131.2.1';
    bExecute    : BOOL := TRUE;
    nSlaveAddr : UINT := 1006;
    sEcBTN      : STRING;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR

fbEcBTN(sNetId:= sNetID, nSlaveAddr:= nPort, bExecute:= bExecute, tTimeout:= T#5S);
IF NOT fbEcBTN.bBusy THEN
     bExecute := FALSE;
     IF NOT fbEcBTN.bError THEN
          sEcBTN := fbEcBTN.sBTN;
     END_IF
     fbEcBTN(bExecute:= bExecute);
END_IF
```

Prerequisites

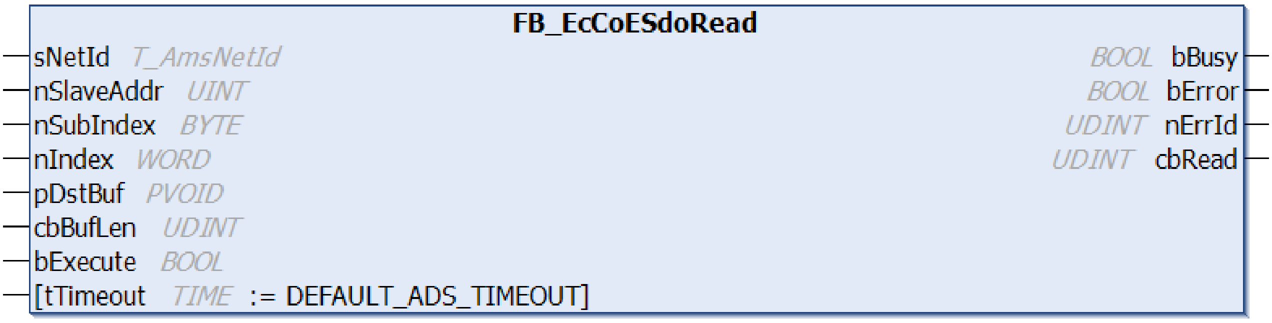
Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

Also see about this

- FB_EcCoeSdoReadEx [▸ 56]

7 CoE interface

7.1 FB_EcCoeSdoRead



The function block `FB_EcCoeSdoRead` allows data to be read from an object directory of an EtherCAT slave via 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 allow the object that is to be read to be selected. The function block FB_EcCoeSdoReadEx [▶ 56] must be used for access to the complete parameter, including subelements.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave to which the SDO upload command should be sent.
nSubIndex	BYTE	Subindex of the object that is to be read.
nIndex	WORD	Index of the object that is to be read.
pDstBuf	PVOID	Address (pointer) to the receive buffer.
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
cbRead	UDINT	Number of successfully read data bytes

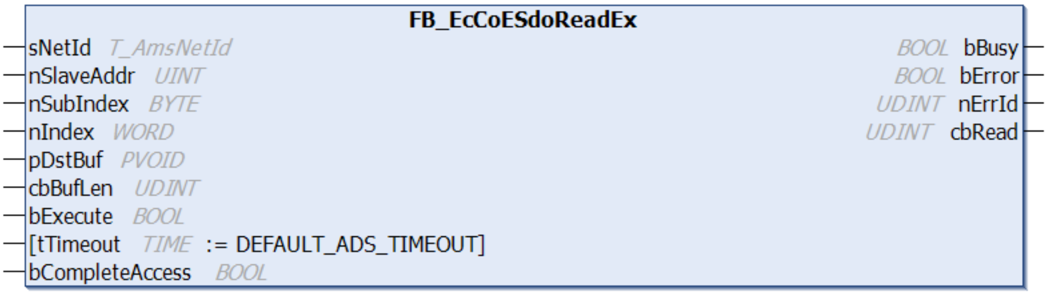
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

7.2 FB_EcCoeSdoReadEx



The function block `FB_EcCoeSdoReadEx` allows data to be read from an object directory of an EtherCAT slave via 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 allow the object that is to be read to be selected. Via `bCompleteAccess := TRUE` the parameter can be read with subelements.

Inputs

```
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         : PVOID; (* 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
```

Name	Type	Description
sNetId	T_Ams NetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlave Addr	UINT	Fixed address of the EtherCAT slave to which the SDO upload command should be sent.
nSubIndex	BYTE	Subindex of the object that is to be read.
nIndex	WORD	Index of the object that is to be read.
pDstBuf	PVOID	Address (pointer) to the receive buffer
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.
bComplete Access	BOOL	If bCompleteAccess is set, the whole parameter can be read in a single access.

Outputs

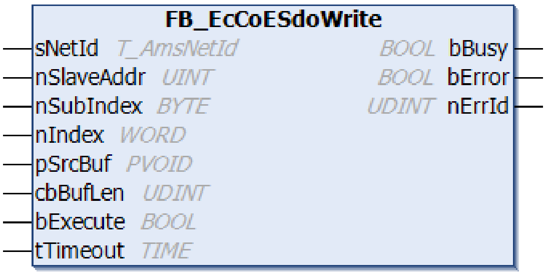
```
VAR_OUTPUT
    bBusy  : BOOL;
    bError : BOOL;
    nErrId : UDINT;
    cbRead : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
cbRead	UDINT	Number of successfully read data bytes

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

7.3 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 allow the object that is to be written to be selected. The function block FB_EcCoeSdoWriteEx [▶ 59] must be used for access to the complete parameter, including subelements.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave to which the SDO download command should be sent.
nSubIndex	BYTE	Subindex of the object that is supposed to be written.
nIndex	WORD	Index of the object that is supposed to be written.
pSrcBuf	PVOID	Address (pointer) to the transmit buffer
cbBufLen	UDINT	Number of date to be sent in bytes
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Example of an implementation in ST:

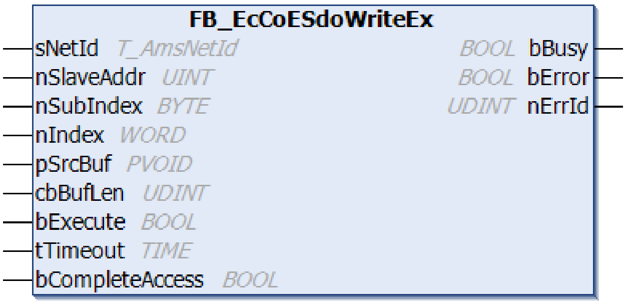
```
PROGRAM TEST_SdoWrite

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 fbSdoWrite.bError THEN
        (* write successful *)
        bError := FALSE;
        nErrId := 0;
    ELSE
        (* write failed *)
        bError := fbSdoWrite.bError;
        nErrId := fbSdoWrite.nErrId;
    END_IF

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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

7.4 FB_EcCoeSdoWriteEx



The `FB_EcCoeSdoWriteEx` 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 allow the object that is to be written to be selected. Via `bCompleteAccess := TRUE` the parameter can be written with sub-elements.

Inputs

```
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         : PVOID; (* 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
```

Name	Type	Description
sNetId	T_Ams NetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlave Addr	UINT	Fixed address of the EtherCAT slave to which the SDO download command should be sent.
nSubIndex	BYTE	Subindex of the object that is supposed to be written.
nIndex	WORD	Index of the object that is supposed to be written.
pSrcBuf	PVOID	Address (pointer) to the transmit buffer
cbBufLen	UDINT	Number of date to be sent in bytes
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.
bCompleteAccess	BOOL	If bCompleteAccess is set, the whole parameter can be written in a single access.

Outputs

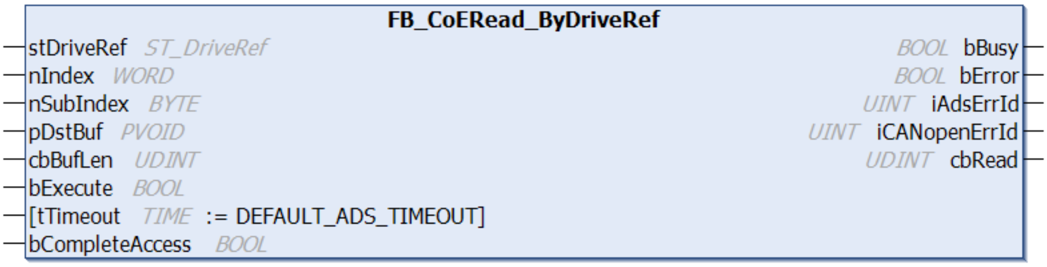
```
VAR_OUTPUT
    bBusy  : BOOL;
    bError : BOOL;
    nErrId : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

7.5 FB_CoERead_ByDriveRef



The function block `FB_CoERead_ByDriveRef` can be used to read drive parameters by means of the "CANopen over EtherCAT (CoE)" protocol. This requires the slave to have a mailbox and to support the “CANopen over EtherCAT” (CoE) protocol. The nSubIndex and nIndex parameters allow the object that is to be read to be selected. Via `bCompleteAccess := TRUE` the parameter can be read with subelements.

Inputs

```
VAR_INPUT
    stDriveRef        :  ST_DriveRef; (*Contains sNetID of EcMaster, nSlaveAddr of EcDrive, nDriveNo
of EcDrive, either preset or read from NC*)
    nIndex            :  WORD; (*SoE IDN: e.g. 'S_0_IDN+1' for S-0-0001 or 'P_0_IDN+23' for
P-0-0023*)
    nSubIndex         :  BYTE;
    pDstBuf           :  PVOID; (*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; (*States the time before the function is cancelled*)
    bCompleteAccess  :  BOOL;
END_VAR
```

Name	Type	Description
stDriveRef	ST_DriveRef	Structure containing the AMS network ID of the EtherCAT master device and the address of the slave device. The reference to the drive can be linked directly to the PLC in the System Manager. To this end an instance of ST_PlcDriveRef must be used and the NEtID of the Byte array converted to a string.
nIndex	WORD	Index of the object that is to be read.
nSubIndex	BYTE	Subindex of the object that is to be read.
pDstBuf	PVOID	Address (pointer) to the receive buffer.
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.
bCompleteAccess	BOOL	If bCompleteAccess is set, the whole parameter can be read in a single access.

Outputs

```
VAR_OUTPUT
    bBusy          : BOOL;
    bError         : BOOL;
    iAdsErrId      : UINT;
    iCANopenErrId  : UINT;
    cbRead         : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
iAdsErrId	UINT	Returns the ADS error code of the last executed command when the bError output is set.
iCANopenErrId	UINT	Returns the CANopen error code if the bError output is set.
cbRead	UDINT	Number of successfully read data bytes

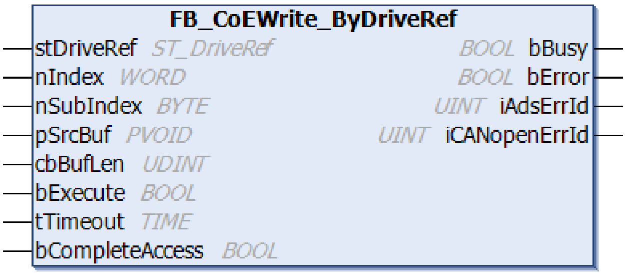
Example of an implementation in ST:

```
PROGRAM MAIN
VAR
    fbCoERead        : FB_CoERead_ByDriveRef;
    stDriveRef       : ST_DriveRef;
    nIndex           : WORD  := 16#1018;
    nSubIndex        : BYTE  := 1;
    bExecute         : BOOL  := TRUE;
    tTimeout         : TIME  := T#5S;
    bCompleteAccess  : BOOL  := TRUE;
    vendorId         : UDINT;
    bError           : BOOL;
    nAdsErrId        : UDINT;
    nCANopenErrId    : UDINT;
END_VAR

fbCoERead(
    stDriveRef:= stDriveRef,
    nIndex:= nIndex,
    nSubIndex:= nSubIndex,
    pDstBuf:= ADR(vendorId),
    cbBufLen:= SIZEOF(vendorId),
    bExecute:= bExecute,
    tTimeout:= tTimeout,
    bCompleteAccess:= bCompleteAccess,
);

IF NOT fbCoERead.bBusy THEN
    bError:=fbCoERead.bError;
    nAdsErrId:=fbCoERead.iAdsErrId;
    nCANopenErrId:=fbCoERead.iCANopenErrId;
    bExecute := FALSE;
    fbCoERead(bExecute := bExecute);
END_IF
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

7.6 FB_CoEWrite_ByDriveRef



The function block `FB_CoEWrite_ByDriveRef` can be used to write drive parameters based on the “CANopen over EtherCAT (CoE)” protocol. This requires the slave to have a mailbox and to support the “CANopen over EtherCAT” (CoE) protocol. The nSubIndex and nIndex parameters allow the object that is to be written to be selected. Via `bCompleteAccess := TRUE` the parameter can be written with sub-elements.

Inputs

```
VAR_INPUT
    stDriveRef       :  ST_DriveRef; (*Contains sNetID of EcMaster, nSlaveAddr EcDrive, nDriveNo of
EcDrive, either preset or read from NC*)
    nIndex           :  WORD; (*SoE IDN: e.g. 'S_0_IDN+1' for S-0-0001 or 'P_0_IDN+23' for
P-0-0023*)
    nSubIndex        :  BYTE; (*SoE element*)
    pSrcBuf          :  PVOID; (*Contains the address of the buffer containing the data to be sent*)
    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; (*States the time before the function is cancelled*)
    bCompleteAccess  :  BOOL;
END_VAR
```

Name	Type	Description
stDriveRef	ST_DriveRef	Structure containing the AMS network ID of the EtherCAT master device and the address of the slave device. The reference to the drive can be linked directly to the PLC in the System Manager. To this end an instance of ST_PlcDriveRef must be used and the NEtID of the Byte array converted to a string.
nIndex	WORD	Index of the object that is supposed to be written.
nSubIndex	BYTE	Subindex of the object that is supposed to be written.
pSrcBuf	PVOID	Address (pointer) to the transmit buffer
cbBufLen	UDINT	Maximum available buffer size for the data to be sent in bytes.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.
bComplete Access	BOOL	If bCompleteAccess is set, the whole parameter can be read in a single access.

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
iAdsErrId	UINT	Returns the ADS error code of the last executed command when the bError output is set.
iCANopenErrId	UINT	Returns the CANopen error code if the bError output is set.

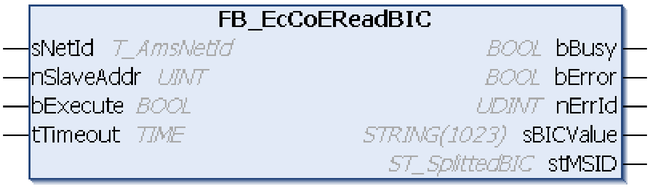
Example of an implementation in ST:

```
PROGRAM MAIN
VAR
    fbCoEWrite       : FB_CoEWrite_ByDriveRef;
    stDriveRef       : ST_DriveRef;
    nIndex           : WORD  := 16#1018;
    nSubIndex        : BYTE  := 1;
    bExecute         : BOOL  := TRUE;
    tTimeout         : TIME  := T#5S;
    bCompleteAccess : BOOL  := TRUE;
    vendorId         : UDINT := 2;
    bError           : BOOL;
    nAdsErrId        : UDINT;
    nCANopenErrId    : UDINT;
END_VAR

fbCoEWrite(
    stDriveRef:= stDriveRef,
    nIndex:= nIndex,
    nSubIndex:= nSubIndex,
    pSrcBuf:= ADR(vendorId),
    cbBufLen:= SIZEOF(vendorId),
    bExecute:= bExecute,
    tTimeout:= tTimeout,
    bCompleteAccess:= bCompleteAccess,
);

IF NOT fbCoEWrite.bBusy THEN
    bError:= fbCoEWrite.bError;
    nAdsErrId:= fbCoEWrite.iAdsErrId;
    nCANopenErrId:= fbCoEWrite.iCANopenErrId;
    bExecute := FALSE;
    fbCoEWrite(bExecute := bExecute);
END_IF
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

7.7 FB_EcCoeReadBIC



The function block FB_EcCoeReadBIC can be used to read the BIC from the object directory of an EtherCAT slave via SDO (Service Data Object) access. For this the slave must have a mailbox and support the "CANopen over EtherCAT (CoE)" protocol and the object directory must contain an object 0x10E2:01 with the BIC.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave to which the SDO upload command should be sent.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

```
VAR_OUTPUT
    bBusy     : BOOL;
    bError    : BOOL;
    nErrId    : UDINT;
    sBICValue : STRING
    stMSID    : ST_SplittedBIC
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
sBICValue	STRING(1023)	This output contains (after an error-free run) the BIC of the EtherCAT slave after the bBusy output has been reset, e.g. "„1P193995SBTN0002agdw1KEL7411 Q1 2P112104020018".

stMSID	ST_SplittedBIC	This output contains (after an error-free run) the substrings of the BIC of the EtherCAT slave after the bBusy output has been reset.
		The following substrings are assigned for the above BIC:
		sItemNo = "193995"
		sBTN = "0002agdw"
		sDescription = "EL7411"
		sQuantity = "1"
		sBatchNo = "112104020018"

Sample of an implementation in ST

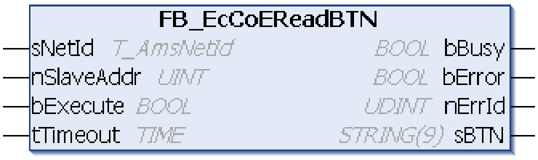
```
PROGRAM TEST_EcCoEReadBIC
VAR
    fbCoEBIC    : FB_EcCoEReadBIC;
    sNetId      : T_AmsNetId := '172.16.2.131.2.1';
    bExecute    : BOOL := TRUE;
    nSlaveAddr  : UINT := 1006;
    sCoEBIC     : STRING(1023);
    stCoEBIC    : ST_SplittedBIC;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

```
fbCoEBIC(sNetId:= sNetID, nSlaveAddr:= nPort, bExecute:= bExecute, tTimeout:= T#5s);;
IF NOT fbCoEBIC.bBusy THEN
     bExecute := FALSE;
     IF NOT fbCoEBIC.bError THEN
          stCoEBIC    := fbCoEBIC.stMSID;
          sCoEBIC     := fbCoEBIC.sBICValue;
     END_IF
     fbCoEBIC(bExecute:= bExecute);
END_IF
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

7.8 FB_EcCoeReadBTN



The function block FB_EcCoeReadBTN can be used to read the BTN from the object directory of an EtherCAT slave via SDO (Service Data Object) access. For this the slave must have a mailbox and support the "CANopen over EtherCAT (CoE)" protocol and the object directory must contain an object 0xF083 with the BTN.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave to which the SDO upload command should be sent.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

```
VAR_OUTPUT
    bBusy  : BOOL;
    bError : BOOL;
    nErrId : UDINT;
    sBTN   : STRING(9)
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
sBTN	STRING	This output contains (after an error-free run) the BTN of the EtherCAT slave after the bBusy output has been reset, e.g. "0002agdw".

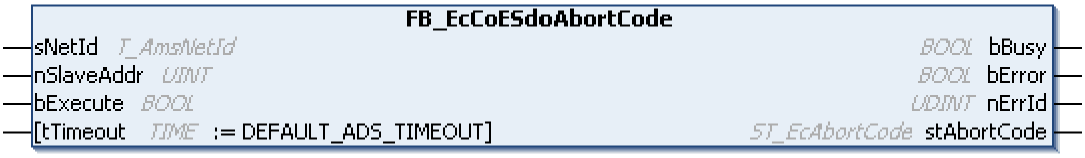
Sample of an implementation in ST

```
PROGRAM TEST_ EcCoEReadBtn
VAR
    fbCoEBTN   : FB_EcCoEReadBtn;
    sNetId     : T_AmsNetId := '172.16.2.131.2.1';
    bExecute   : BOOL := TRUE;
    nSlaveAddr : UINT := 1006;
    sCoEBTN    : STRING;
    bError     : BOOL;
    nErrId     : UDINT;
END_VAR

fbCoEBTN(sNetId:= sNetID, nSlaveAddr:= nPort, bExecute:= bExecute, tTimeout:= T#5S);
IF NOT fbCoEBTN.bBusy THEN
     bExecute  := FALSE;
     IF NOT fbCoEBTN.bError THEN
          sCoEBTN  := fbCoEBTN.sBTN;
     END_IF
     fbCoEBTN(bExecute:= bExecute);
END_IF
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

7.9 FB_EcCoESdoAbortCode



The function block

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device.
nSlave Addr	UINT	Fixed address of the EtherCAT slave to which the SDO download command should be sent.
bExecute	BOOL	The function block is enabled by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

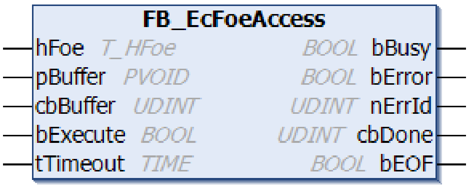
Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
stAbortCode	ST_EcAbortCode	Abort code

8 FoE interface

8.1 FB_EcFoeAccess



This function block writes or reads data via the communication port of the “File access over EtherCAT” mailbox protocol.

Inputs

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

Name	Type	Description
hFoe	T_HFoe [▶ 131]	"File access over EtherCAT" handle
pBuffer	DWORD	Contains the address of the buffer into which the data are to be read (read access) or the address of buffer containing the data to be written (write access). The buffer can be a single variable, an array or a structure, whose address can be found with the ADR operator.
cbBuffer	UDINT	Contains the number of data bytes to be written or read.
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

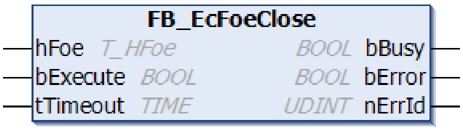
```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    cbDone  : UDINT;
    bEOF    : BOOL;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
cbDone	UDINT	Number of the most recent successfully written or read data bytes
bEOF	BOOL	End of File, this variable becomes TRUE if the end of the file is reached during read access. For write access this variable has no purpose.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

8.2 FB_EcFoeClose



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

Inputs

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

Name	Type	Description
hFoe	T_HFoe [▶ 131]	"File access over EtherCAT" handle
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

Outputs

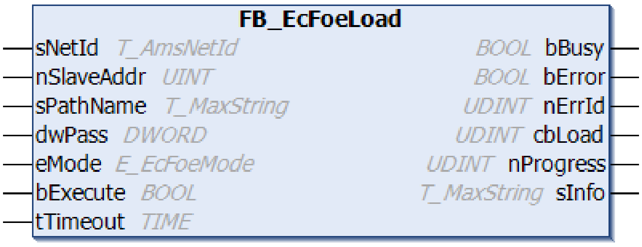
```
VAR_OUTPUT
    bBusy  : BOOL;
    bError : BOOL;
    nErrId : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

8.3 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.

> 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.

Inputs

```
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
```

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device.
nSlaveAddr	UINT	Fixed address of the EtherCAT slave whose file is to be uploaded or downloaded.
sPathName	T_MaxString	Contains the path and filenames of the file to be written or read. (e.g.: 'C:\FOE_Test\EL6751\ECATFW__EL6751_C6_V0030.efw')
dwPass	DWORD	Password (default: 0)
eMode	E_EcFoeMode [▶ 121]	"File access over EtherCAT" access mode (default: write access)
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time that must not be exceeded when the function block is executed (default: 200 s).

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
cbLoad	UDINT	Number of successfully written or read data bytes
nProgress	UDINT	Write access progress (range: 0 - 100%). This variable is currently not used for read access, in which case it is always 0.
sInfo	T_MaxString	Additional command information as string (reserved)

Sample in ST:

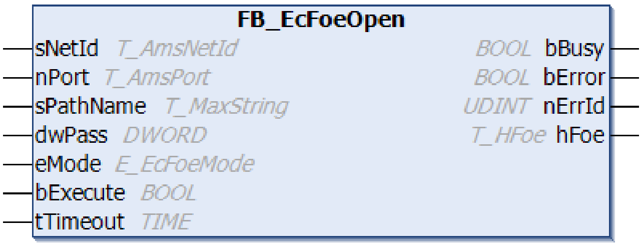
A rising edge at the bLoad variable triggers the firmware download via the “File access over EtherCAT” mailbox protocol.

```
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 );
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

8.4 FB_EcFoeOpen



This function block is used to open the communication port for the “File access over EtherCAT” mailbox protocol.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device.
nPort	UINT	Fixed address of the EtherCAT device
sPathName	T_MaxString	File path name (e.g.: 'c:\TwinCAT\FOE\Data.fwp') (See below for further explanations of sPathName.)
dwPass	DWORD	Password
eMode	E_EcFoeMode [▸ 121]	Access mode (write/read access)
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

sPathName

By default, only the filename (without filename extension) is extracted from the file path (e.g. 'c:\TwinCAT\FOE\Data.fwp') that was entered and used as the filename for the FoE protocol (in our example: 'Data'). From library version 3.3.12.0, filenames including the filename extension can also be used (in our example: 'Data.fwp').

Via the global boolean variable
`Tc2_EtherCAT.bEcFoeOpenFileNameWithFileExt`
the use of the filename extension can be enabled or disabled for all instances of the FB_EcFoeOpen function block. By default, the variable has the value FALSE (no filename extension). If you set the value to TRUE then the use of filename extensions is enabled.
Note that the FoE function blocks were originally used for firmware updates where no filename extension was used. If you want to update the firmware, you may have to make sure that the global variable has its original default value, i.e. FALSE.

Outputs

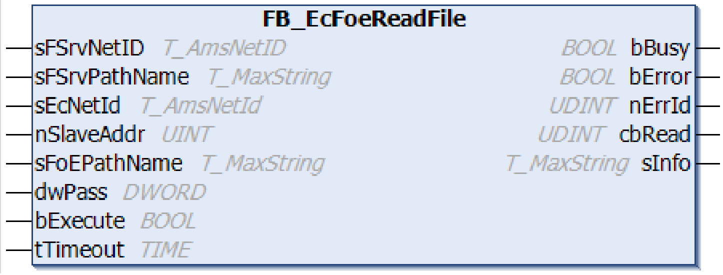
```
VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    hFoe    : T_HFoe;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
hFoe	T_HFoe [▸ 131]	"File access over EtherCAT" handle

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

8.5 FB_EcFoeReadFile



FB_EcFoeReadFile

sFSrvNetID	T_AmsNetID	BOOL	bBusy
sFSrvPathName	T_MaxString	BOOL	bError
sEcNetId	T_AmsNetId	UDINT	nErrId
nSlaveAddr	UINT	UDINT	cbRead
sFoEPathName	T_MaxString	T_MaxString	sInfo
dwPass	DWORD		
bExecute	BOOL		
tTimeout	TIME		

The FB_EcFoeReadFile function block can be used to download files from an EtherCAT device to the local data carrier via the "File access over EtherCAT" mailbox protocol.

The file path can only point to the local file system on the computer. This means that network paths cannot be used here.

Inputs

```
VAR_INPUT
    sFSrvNetId      : T_AmsNetId := '';
    sFSrvPathName   : T_MaxString;
    sEcNetId        : T_AmsNetId;
    nSlaveAddr      : UINT;
    sFoEPathName    : T_MaxString;
    dwPass          : DWORD := 0;
    bExecute        : BOOL;
    tTimeout        : TIME := T#200s;
END_VAR
```

Name	Type	Description
sFSrvNetId	T_AmsNetId	AMS network ID of the computer on which the file that was read is to be written. (Default: local computer)
sFSrvPathName	T_MaxString	Contains the path and filename of the file to be written (e.g. 'C:\Data\LogData.csv').
sEcNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device.
nSlaveAddr	UINT	Address of the EtherCAT slave
sFoEPathName	T_MaxString	Name of the file on the EtherCAT slave (e.g. 'LogData')
dwPass	DWORD	Password
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block. (Default: 200 s.)

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
cbRead	UDINT	Number of successfully read data bytes
sInfo	T_MaxString	Additional FoE error information (reserved)

Sample in ST:

A rising edge at the `bExecute` variable triggers reading of the specified file via the "File access over EtherCAT" mailbox protocol. The file named in `sFoEPathName` is read by the selected EtherCAT slave (`sEcNetId & nSlaveAddr`). The file is stored on the selected computer (`sFSrvNetID`) under the name specified in `sFSrvPathName`. If a password is required for reading the file from the EtherCAT slave, this can be specified via `dwPass`.

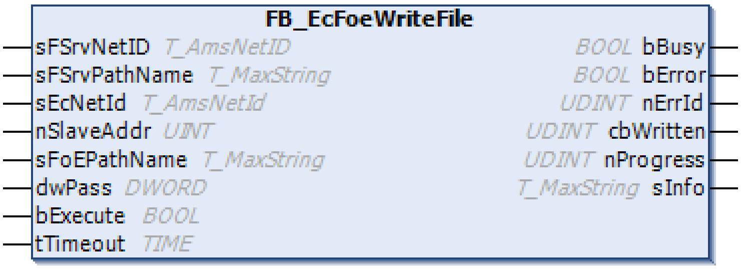
The read and write operation is not completed until `bBusy` switches to FALSE. Only then can the error information or the number of bytes read be evaluated.

```
PROGRAM MAIN
VAR
    fbEcReadFile : FB_EcFoeReadFile := (
        sFSrvNetID     := '5.0.34.38.1.1', (* NetID for target file *)
        sFSrvPathName  := 'C:\Data\LogData.csv', (* Pathname for target file *)
        sEcNetId       := '5.0.34.38.3.1', (* NetID of EtherCAT master *)
        nSlaveAddr     := 1004, (* EtherCAT slave address *)
        sFoEPathName   := 'LogData', (* Name of source file *)
        dwPass         := 0
    );
    bExecute   : BOOL := TRUE;
    bBusy      : BOOL;
    bError     : BOOL;
    nErrID     : UDINT;
    nBytesRead : UDINT;
END_VAR

fbEcReadFile (
        bExecute := bExecute,
        bBusy => bBusy,
        bError => bError,
        nErrId => nErrID
);
IF NOT bBusy THEN
    bExecute := FALSE;

    IF NOT bError THEN
         (* done, no error *)
        nBytesRead := fbEcReadFile.cbRead;
    ELSE
        (* evaluate error *)
        nBytesRead := 0;
    END_IF
    fbEcReadFile (bExecute := FALSE);
END_IF
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT >= 3.3.14

8.6 FB_EcFoeWriteFile



The function block FB_EcFoeWriteFile can be used to write files from a local data carrier to an EtherCAT device via the "File access over EtherCAT" mailbox protocol.

The file path can only point to the local file system on the computer. This means that network paths cannot be used here.

Inputs

```
VAR_INPUT
    sFSrvNetId      : T_AmsNetId := '';
    sFSrvPathName   : T_MaxString;
    sEcNetId        : T_AmsNetId;
    nSlaveAddr      : UINT;
    sFoEPathName    : T_MaxString;
    dwPass          : DWORD  := 0;
    bExecute        : BOOL;
    tTimeout        : TIME  := T#200s;
END_VAR
```

Name	Type	Description
sFSrvNetId	T_AmsNetId	AMS network ID of the computer from which the file to be written is to be read. (Default: local computer)
sFSrvPathName	T_MaxString	Contains the path and filename of the file to be read (e.g.: 'C:\Data\LogData.csv').
sEcNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device.
nSlaveAddr	UINT	Address of the EtherCAT slave
sFoEPathName	T_MaxString	Name of the file on the EtherCAT slave (e.g. 'LogData')
dwPass	DWORD	Password
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block. (Default: 200 s.)

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
cbWritten	UDINT	Number of successfully written data bytes
nProgress	UDINT	Write access progress (range: 0 - 100%).
sInfo	T_MaxString	Additional FoE error information (reserved)

Sample in ST:

A rising edge at the `bExecute` variable triggers writing of the specified file via the "File access over EtherCAT" mailbox protocol. The file named in `sFSrvPathName` is read from the selected computer (`sFSrvNetID`). The file is stored on the selected EtherCAT slave (`sEcNetId & nSlaveAddr`) under the name specified in `sFoEPathName`. If a password is required for writing the file to the EtherCAT slave, this can be specified via `dwPass`.

The read and write operation is not completed until `bBusy` switches to FALSE. Only then can the error information or the number of bytes read be evaluated.

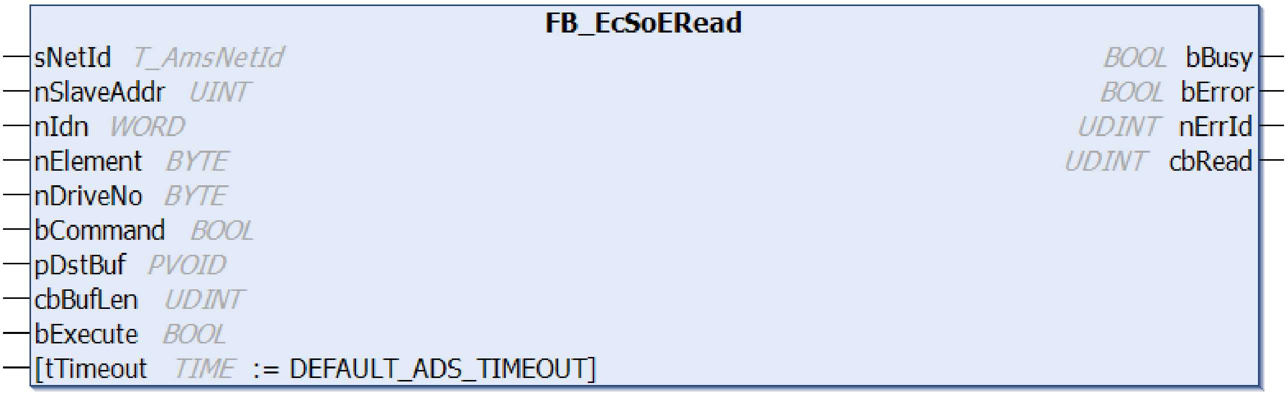
```
PROGRAM MAIN
VAR
    fbEcWriteFile : FB_EcFoeWriteFile := (
        sFSrvNetID     := '5.0.34.38.1.1', (* NetID for source file *)
        sFSrvPathName  := 'C:\Data\LogData.csv', (* Pathname for source file *)
        sEcNetId       := '5.0.34.38.3.1', (* NetID of EtherCAT master *)
        nSlaveAddr     := 1004, (* EtherCAT slave address *)
        sFoEPathName   := 'LogData', (* Name of target file *)
        dwPass         := 0
    );
    bExecute   : BOOL := TRUE;
    bBusy      : BOOL;
    bError     : BOOL;
    nErrID     : UDINT;
    nBytesWritten : UDINT;
END_VAR

fbEcWriteFile (
        bExecute := bExecute,
        bBusy => bBusy,
        bError => bError,
        nErrId => nErrID
);
IF NOT bBusy THEN
    bExecute := FALSE;

    IF NOT bError THEN
         (* done, no error *)
        nBytesWritten := fbEcWriteFile.cbWritten;
    ELSE
        (* evaluate error *)
        nBytesWritten := 0;
    END_IF
    fbEcWriteFile (bExecute := FALSE);
END_IF
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.56	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT >= 3.5.1.0

9 SoE interface

9.1 FB_EcSoeRead



FB_EcSoERead

sNetId *T_AmsNetId* — *BOOL* bBusy
nSlaveAddr *UINT* — *BOOL* bError
nIdn *WORD* — *UDINT* nErrId
nElement *BYTE* — *UDINT* cbRead
nDriveNo *BYTE*
bCommand *BOOL*
pDstBuf *PVOID*
cbBufLen *UDINT*
bExecute *BOOL*
[tTimeout *TIME* := DEFAULT_ADS_TIMEOUT]

The function block `FB_EcSoeRead` can be used to read drive parameters by means of the “Sercos over EtherCAT (SoE)” protocol. To this end the slave must have a mailbox and support the SoE protocol. The drive parameter to be read is specified with the parameters nIdn (identification number), nElement and nDriveNo.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave to which the SoE read command is to be sent.
nIdn	WORD	Identification number of the parameter to be read
nElement	BYTE	Element number of the parameter to be read (See nElement)
nDriveNo	BYTE	Drive number
bCommand	BOOL	This parameter should be set if internal command execution is to be used.
pDstBuf	PVOID	Address (pointer) to the receive buffer
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

nElement

Element number of the parameter to be read. The following values are permitted:

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

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
cbRead	UDINT	Number of successfully read data bytes

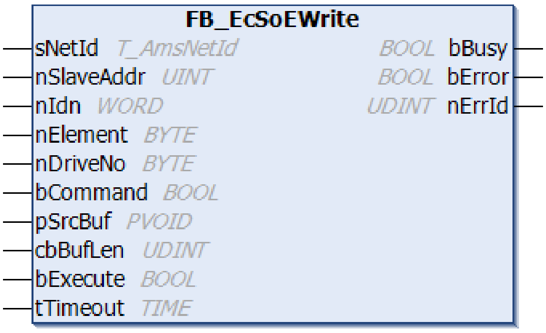
Example of an implementation in ST:

```
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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

9.2 FB_EcSoeWrite



The function block `FB_EcSoeWrite` can be used to write drive parameters by means of the “Sercos over EtherCAT (SoE)” protocol. To this end the slave must have a mailbox and support the SoE protocol. The drive parameter to be written is specified with the parameters nIdn (identification number), nElement and nDriveNo.

Inputs

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

Name	Type	Description
sNetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device. (Type: T_AmsNetId)
nSlaveAddr	UINT	Fixed address of the EtherCAT slave to which the SoE write command is to be sent.
nIdn	WORD	Identification number of the parameter to be written.
nElement	BYTE	Element number of the parameter to be written (See nElement)
nDriveNo	BYTE	Drive number
bCommand	BOOL	This parameter should be set if internal command execution is to be used.
pSrcBuf	PVOID	Address (pointer) to the transmit buffer
cbBufLen	UDINT	Number of date to be sent in bytes
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

nElement

Element number of the parameter to be written. The following values are permitted:

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

Outputs

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

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
nErrId	UDINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.

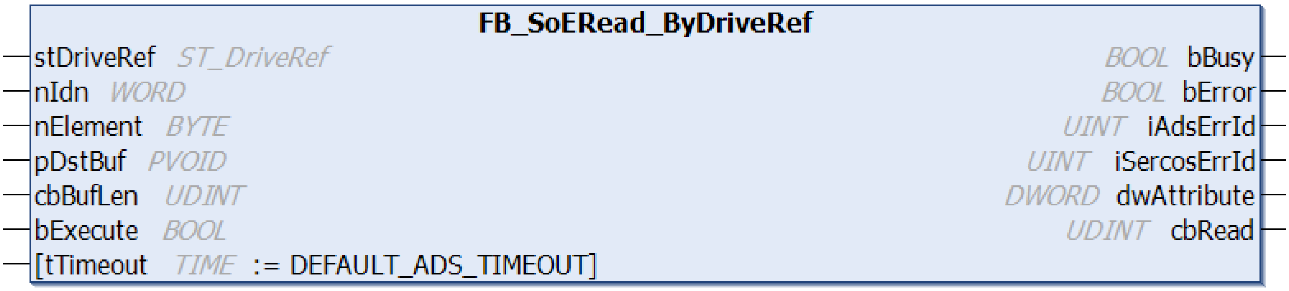
Example of an implementation in ST:

```
PROGRAM TEST_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;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

9.3 FB_SoERead_ByDriveRef



The `FB_SoeRead_ByRef` function block can be used to read drive parameters by means of the “Sercos over EtherCAT (SoE)” protocol. To this end the slave must have a mailbox and support the SoE protocol. The drive parameter to be read is specified with the parameters nIdn (identification number), nElement and stDriveRef.

The global variable `bSeqReadDrvAttrAndValue := TRUE` from the Tc2_EtherCAT library can be used to enforce sequential access to attribute and value. The default value of this variable is FALSE. Devices of the AX5xxx series enable parallel and sequential access to attribute and value. For third-party devices it may be necessary to separate access to attribute and value, which overall slows down access by several cycles.

Inputs

```
VAR_INPUT
    stDriveRef : ST_DriveRef; (* contains sNetID of EcMaster, nSlaveAddr of EcDrive, nDriveNo of EcD
rive, 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-0-0023*)
    nElement   : BYTE;  (* SoE element.*)
    pDstBuf    : PVOID; (* 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
```

Name	Type	Description
stDriveRef	ST_DriveRef	The reference to the drive can be linked directly to the PLC in the System Manager. To this end an instance of ST_PlcDriveRef must be used and the NetID of the Byte array converted to a string.
nIdn	WORD	Identification number of the parameter to be read
nElement	BYTE	Element number of the parameter to be read (See nElement)
pDstBuf	PVOID	Address (pointer) to the read buffer
cbBufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

nElement

Element number of the parameter to be read. The following values are permitted:

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

Outputs

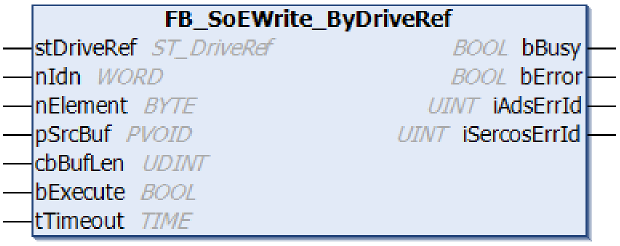
```
VAR_OUTPUT
    bBusy        : BOOL;
    bError       : BOOL;
    iAdsErrId    : UINT;
    iSercosErrId : UINT;
    dwAttribute  : DWORD;
    cbRead       : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
iAdsErrId	UINT	Returns the ADS error code of the last executed command when the bError output is set.
iSercosErrId	UINT	Returns the Sercos error of the last executed command in the event of a set bError output.
dwAttribute	DWORD	Returns the attributes of the Sercos parameter.
cbRead	UDINT	Number of successfully read data bytes

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

9.4 FB_SoEWrite_ByDriveRef



The function block `FB_SoeWrite_ByRef` can be used to write drive parameters by means of the “Sercos over EtherCAT (SoE)” protocol. To this end the slave must have a mailbox and support the SoE protocol. The drive parameter to be written is specified with the parameters nIdn (identification number), nElement and stDriveRef.

The global variable `bSeqReadDrvAttrAndValue := TRUE` from the Tc2_EtherCAT library can be used to enforce sequential access to attribute and value. The default value of this variable is FALSE. Devices of the AX5xxx series enable parallel and sequential access to attribute and value. For third-party devices it may be necessary to separate access to attribute and value, which overall slows down access by several cycles.

Inputs

```
VAR_INPUT
    stDriveRef : ST_DriveRef; (* contains sNetID of EcMaster, nSlaveAddr of EcDrive, nDriveNo of EcD
rive, 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-0-0023*)
    nElement   : BYTE;  (* SoE element.*)
    pSrcBuf    : PVOID; (* 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
```

Name	Type	Description
stDriveRef	ST_DriveRef	The reference to the drive can be linked directly to the PLC in the System Manager. To this end an instance of ST_PlcDriveRef must be used and the NetID of the Byte array converted to a string.
nIdn	WORD	Identification number of the parameter to be read
nElement	BYTE	Element number of the parameter to be read (See nElement)
pSrcBuf		Address (pointer) to the send buffer
cbBufLen		Maximum available buffer size (in bytes) for the data to be read
bExecute	BOOL	The function block is activated by a positive edge at this input.
tTimeout	TIME	Maximum time allowed for the execution of the function block.

nElement

Element number of the parameter to be read. The following values are permitted:

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

Outputs

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

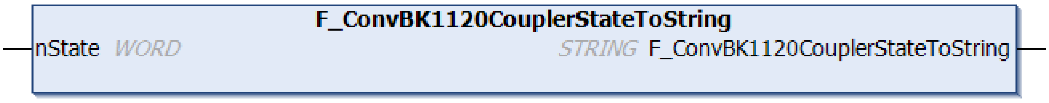
Name	Type	Description
bBusy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
iAdsErrId	UINT	Supplies the ADS error code associated with the most recently executed command if the bError output is set.
iSercos ErrId	UINT	Returns the Sercos error of the last executed command when the bError output is set.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

10 Conversion Functions

10.1 F_ConvBK1120CouplerStateToString



F_ConvBK1120CouplerStateToString
nState WORD
STRING F_ConvBK1120CouplerStateToString

The function `F_ConvBK1120CouplerStateToString` returns the coupler state of the BK1120/BK1150/BK1250 as string. For nState = 0 'No error' is returned, otherwise 'K-bus error' is returned, e.g. for nState = 1. If several errors are pending, they are separated by commas.

Inputs

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

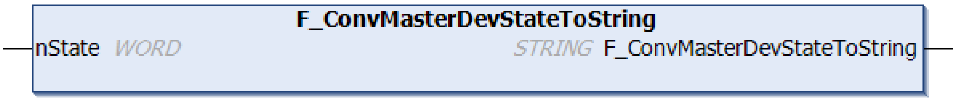
Name	Type	Description

nState	WORD	Coupler state; 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)'`

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

10.2 F_ConvMasterDevStateToString



F_ConvMasterDevStateToString
nState WORD
STRING F_ConvMasterDevStateToString

The function `F_ConvMasterDevStateToString` converts the device status of the EtherCAT master to string.

For nState = 0 'OK' is returned, otherwise, 'Not OK – Link error', e.g. for nState = 1. If several errors are pending, they are separated by hyphens.

Inputs

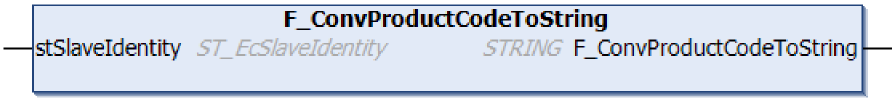
```
VAR_INPUT
    nState : WORD;
END_VAR
```

Name	Type	Description
nState	WORD	Device state 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'

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

10.3 F_ConvProductCodeToString



The function `F_ConvProductToString` returns the product code as string, e.g. 'EL6731-0000-0017'. From version 3.3.8.0 of the Tc2_EtherCAT library this function also supports ELM and EPP slaves such as 'EPP4374-0002-0018' and 'ELM3704-0001-0016'.

Inputs

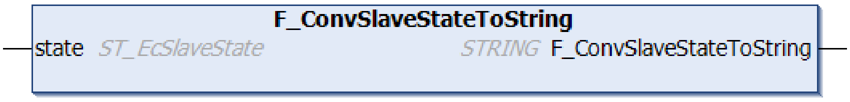
```
VAR_INPUT
    stSlaveIdentity : ST_EcSlaveIdentity;
END_VAR
```

Name	Type	Description
stSlaveIdentity	ST_EcSlaveIdentity	Slave Identity as it can be read with the FB_EcGetSlaveIdentity [▶ 32].

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

10.4 F_ConvSlaveStateToString



The function `F_ConvSlaveStateToString` returns the EtherCAT slave state as string. For conversion to the string see F_ConvStateToString [▶ 87].

Inputs

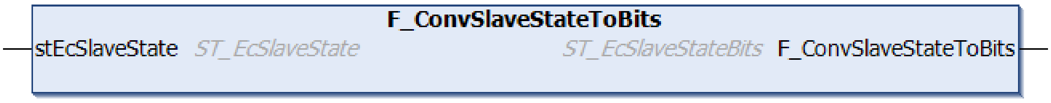
```
VAR_INPUT
    state : ST_EcSlaveState;
END_VAR
```

Name	Type	Description
state	ST_EcSlaveState	EtherCAT slave state structure (consisting of: deviceState : BYTE; linkState : BYTE;)

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

10.5 F_ConvSlaveStateToBits



The function `F_ConvSlaveStateToBits` returns the EtherCAT slave state as structure TYPE ST_EcSlaveStateBits [▶ 127].

Inputs

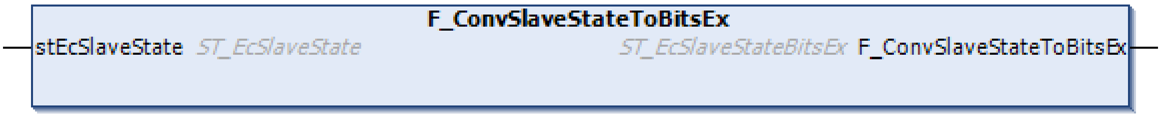
```
VAR_INPUT
    stEcSlaveState : ST_EcSlaveState;
END_VAR
```

Name	Type	Description
stEcSlaveState	ST_EcSlaveState	EtherCAT slave state structure (consisting of: deviceState : BYTE; linkState : BYTE;)

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

10.6 F_ConvSlaveStateToBitsEx



The function `F_ConvSlaveStateToBitsEx` returns the EtherCAT slave state as structureST_EcSlaveStateBitsEx [▶ 127].

Inputs

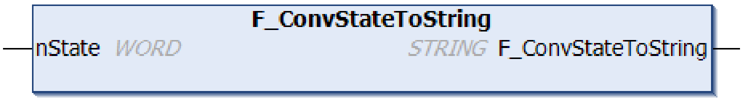
```
VAR_INPUT
    stEcSlaveState : ST_EcSlaveState;
END_VAR
```

Name	Type	Description
stEcSlaveState	ST_EcSlaveState	EtherCAT slave state structure (consisting of: deviceState : BYTE; linkState : BYTE;)

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

10.7 F_ConvStateToString



The function `F_ConvStateToString` returns the EtherCAT slave state as string. For nState = 0 ‘ ‘ is returned, otherwise, ‘INIT ‘ is returned, e.g. for nState = 1. If several messages are pending, they are separated by spaces.

Inputs

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

Name	Type	Description
nState	WORD	EtherCAT slave state as WORD 0x___1 = 'INIT' 0x___2 = 'PREOP' 0x___3 = 'BOOT' 0x___4 = 'SAFEOP' 0x___8 = 'OP' 0x001_ = 'Slave signals error' 0x002_ = 'Invalid vendorId, productCode... read' 0x004_ = 'Initialization error occurred' 0x008_ = 'Slave disabled' 0x010_ = 'Slave not present' 0x020_ = 'Slave signals link error' 0x040_ = 'Slave signals missing link' 0x080_ = 'Slave Slave signals unexpected link' 0x100_ = 'Communication port A' 0x200_ = 'Communication port B' 0x400_ = 'Communication port C' 0x800_ = 'Communication port D'

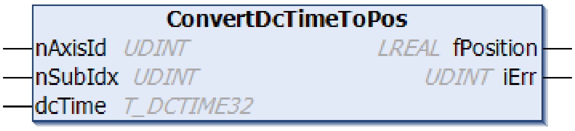
Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11 Distributed Clocks

11.1 DCTIME32

11.1.1 ConvertDcTimeToPos



This function block converts a 32-bit distributed clock system time variable of type T_DCTIME32 [▶ 130] to a corresponding NC axis position (i.e. the NC axis position at precisely this time).

Inputs

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

Name	Type	Description
nAxisId	UDINT	ID of the NC axis
nSubIdx	UDINT	This 32-bit input variable 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	T_DCTIME32	32-bit "Distributed Clock System Time" variable. This input variable is converted into the corresponding NC axis position.

The 32-bit time may only be used in the narrow range of ± 2,147 seconds around the current system time, to ensure that it is unambiguous. Within the function block this prerequisite cannot be checked.

Outputs

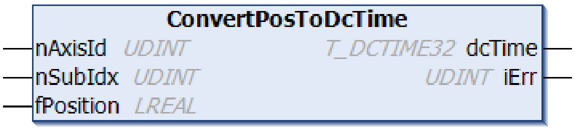
```
VAR_OUTPUT
    fPosition : LREAL;
    iErr      : UDINT;
END_VAR
```

Name	Type	Description
fPosition	LREAL	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 millimeters.
iErr	UDINT	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).

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.1.2 ConvertPosToDcTime



This function block converts an NC axis position to a corresponding 32-bit distributed clock system time variable of type T_DCTIME32 [▶ 130] (i.e. the time which precisely this NC axis position was or will be reached).

Inputs

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

Name	Type	Description
nAxisId	UDINT	ID of the NC axis
nSubIdx	UDINT	This 32-bit input magnitude is composed of two different items of information, and is 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.
fPosition	LREAL	NC axis position that is converted to the corresponding 32-bit "Distributed Clock System Time" variable. If the "Distributed Clock System Time" associated with the position is outside the expected time window of ± 2.147 seconds, this conversion is rejected with an error number.

Outputs

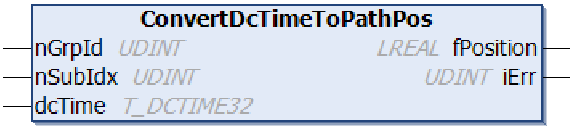
```
VAR_OUTPUT
    dcTime : T_DCTIME32;(* 32 bit distributed clock time *)
    iErr   : UDINT;
END_VAR
```

Name	Type	Description
dcTime	T_DCTIME32	Returns the 32-bit "Distributed Clock System Time" variable that corresponds to input fPosition.
iErr	UDINT	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 range exceeded (future) • Error 0x4362: time range exceeded (past) • Error 0x4363: position cannot be determined mathematically.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.1.3 ConvertDcTimeToPathPos



This function block converts a 32-bit distributed clock system time variable of type T_DCTIME32 [▶ 130] to a relative Nci path distance on the contour of the currently active Nci program (i.e. the function block returns a positive or negative relative interval, depending on the timing).

Inputs

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

Name	Type	Description
nGrpId	UDINT	Group ID of the corresponding Nci channel
nSubIdx	UDINT	This 32-bit input variable 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). 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	T_DCTIME32	32-bit "Distributed Clock System Time" variable. This input variable is converted to the corresponding relative Nci path distance on the contour.

The 32-bit time may only be used in the narrow range of ± 2,147 seconds around the current system time, to ensure that it is unambiguous. Within the function block this prerequisite cannot be checked.

Outputs

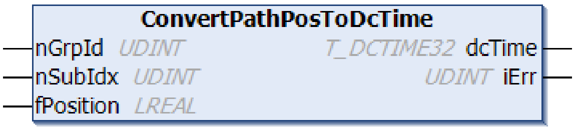
```
VAR_OUTPUT
    fPosition : LREAL;
    iErr      : UDINT;
END_VAR
```

Name	Type	Description
fPosition	LREAL	Returns the relative Nci path distance on the contour that corresponds to the dcTime.
iErr	UDINT	Returns an error number in the event of an error

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.1.4 ConvertPathPosToDcTime



This function block converts a relative Nci path distance to a corresponding 32-bit distributed clock system time variable of type T_DCTIME32 [▶ 130] (i.e. the time that corresponds or corresponded to the relative Nci path distance).

Inputs

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

Name	Type	Description
nGrpId	UDINT	Group ID of the corresponding Nci channel
nSubIdx	UDINT	This 32-bit input variable 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). 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	LREAL	Relative Nci path distance converted to the corresponding 32-bit "Distributed Clock System Time". If the "Distributed Clock System Time" associated with the relative Nci path distance is outside the expected time window of ± 2.147 seconds, this conversion is rejected with an error number.

Outputs

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

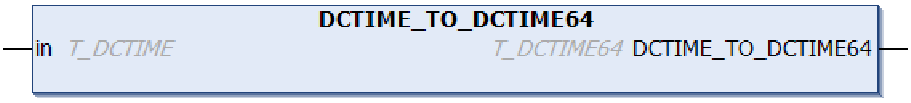
Name	Type	Description
dcTime	T_DCTIME32	Returns the 32-bit "Distributed Clock System Time" variable that corresponds to input fPosition.
iErr	UDINT	Supplies an error number if an error occurs, e.g. • Error 0x4361: time range exceeded (future) • Error 0x4362: time range exceeded (past) • Error 0x4363: position cannot be determined mathematically

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.2 DCTIME64

11.2.1 DCTIME_TO_DCTIME64



The function converts a distributed clock system time variable of type T_DCTIME [▶ 131] to a 64-bit distributed clock system time variable of type T_DCTIME64 [▶ 130].

FUNCTION DCTIME_TO_DCTIME64: T_DCTIME64

Inputs

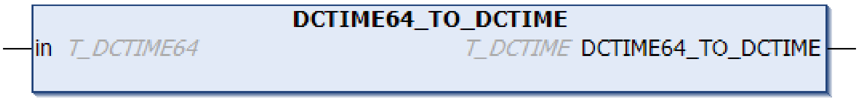
```
VAR_INPUT
    in : T_DCTIME;
END_VAR
```

Name	Type	Description
in	T_DCTIME	The "Distributed Clock System Time" variable to be converted

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.2.2 DCTIME64_TO_DCTIME



The function converts a distributed clock system time variable of type T_DCTIME [▶ 130] to a 64-bit distributed clock system time variable of type T_DCTIME64 [▶ 131].

FUNCTION DCTIME64_TO_DCTIME: T_DCTIME

Inputs

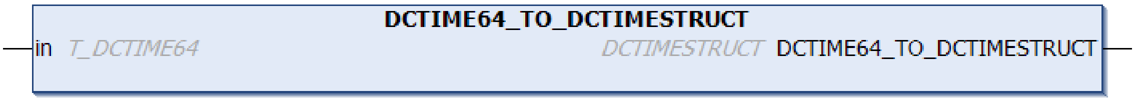
```
VAR_INPUT
    in : T_DCTIME64;
END_VAR
```

Name	Type	Description
in	T_DCTIME64	The "Distributed Clock System Time" variable to be converted

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.2.3 DCTIME64_TO_DCTIMESTRUCT



The function converts a 64-bit distributed clock system time variable of type T_DCTIME64 [▶ 130] to a structured variable of type DCTIMESTRUCT [▶ 129].

FUNCTION DCTIME64_TO_DCTIMESTRUCT

Inputs

```
VAR_INPUT
    in : T_DCTIME64;
END_VAR
```

Name	Type	Description
in	T_DCTIME64	The "Distributed Clock System Time" variable to be converted

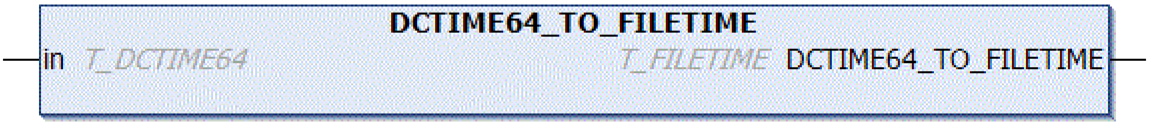
Sample:

```
PROGRAM P_TEST
VAR
    dcStruct : DCTIMESTRUCT;
    dcTime : T_DCTIME64;
END_VAR

dcTime : F_GetCurDcTickTime64();
dcStruct := DCTIME64_TO_DCTIMESTRUCT (dcTIME);
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.2.4 DCTIME64_TO_FILETIME64



The function converts a 64-bit "Distributed Clock System Time" variable of type T_DCTIME64 [▶ 130] to a 64-bit "Windows File Time" variable of type T_FILETIME64.

FUNCTION DCTIME64_TO_FILETIME64: T_FILETIME64

Inputs

```
VAR_INPUT
    in : T_DCTIME64:
END_VAR;
```

Name	Type	Description
in	T_DCTIME64	The "Distributed Clock System Time" variable to be converted

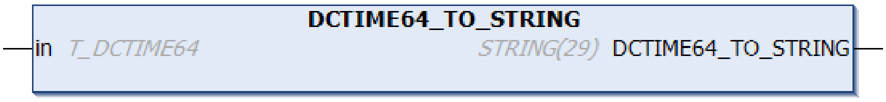
Sample:

```
PROGRAM P_TEST
VAR
    ft  : T_FILETIME64;
    dct : T_DCTIME64;
END_VAR

dct := F_GetCurDcTickTime64();
ft  := DCTIME64_TO_FILETIME64(dct);
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT >= 3.3.16.0

11.2.5 DCTIME64_TO_STRING



The function converts a 64-bit distributed clock system time variable of type T_DCTIME64 [▶ 130] to a string.

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

- YYYY: year;
- MM: month;
- DD: day;
- hh: hour;
- mm: minute;
- ss: second;
- nnnnnnnnn: nanoseconds

FUNCTION DCTIME64_TO_STRING: STRING (29)

Inputs

```
VAR_INPUT
    in : T_DCTIME64; (*Distributed Clock Time*)
END_VAR
```

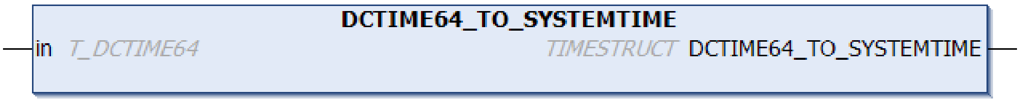
Name	Type	Description
in	T_DCTIME64	The "Distributed Clock System Time" variable to be converted

Example:
See description of the function: F_GetCurDcTickTime64 [▶ 102].

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.2.6 DCTIME64_TO_SYSTEMTIME



The function converts a 64-bit distributed clock system time variable of type T_DCTIME64 [▶ 130] to a structured Windows system time variable of type TIMESTRUCT.

DCTIME64_TO_SYSTEMTIME: TIMESTRUCT

Inputs

```
VAR_INPUT
    in : T_DCTIME64;
END_VAR
```

Name	Type	Description
in	T_DCTIME64	The "Distributed Clock System Time" variable to be converted

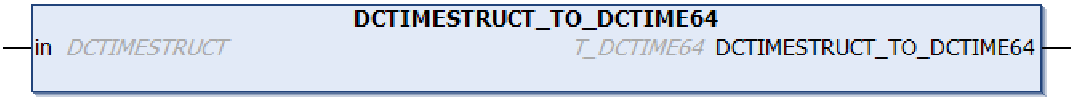
Sample:

```
PROGRAM P_TEST
VAR
    syst : TIMESTRUCT
END_VAR

syst := DCTIME64_TO_SYSTEMTIME ( F_GetCurDcTickTime64() )
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.2.7 DCTIMESTRUCT_TO_DCTIME64



The function converts a structured variable of type of type DCTIMESTRUCT [▶ 129] to a 64-bit distributed clock system time variable T_DCTIME64 [▶ 130]. The structure components wDayofWeek is ignored in the conversion. The structure components wYear must be greater than or equal to 2000 and less than 2584. For invalid values of the structure components the function returns the value zero.

FUNCTION DCTIMESTRUCT_TO_DCTIME64: T_DCTIME64

Inputs

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

Name	Type	Description
in	DCTIMESTRUCT	The structured variable to be converted

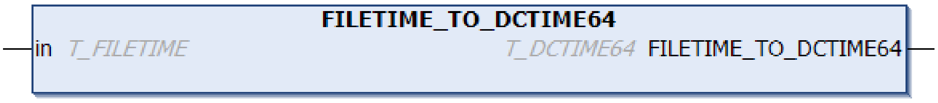
Sample:

```
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_DCTIME64;
END_VAR

dc64 := DCTIMESTRUCT_TO_DCTIME64( dcStruct );
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.2.8 FILETIME64_TO_DCTIME64



The function converts a 64-bit "Windows File Time" variable of type T_FILETIME64 to a 64-bit "Distributed Clock System Time" variable of type T_DCTIME64 [▸ 130]. In the event of a conversion error the function returns the value zero.

FUNCTION FILETIME64_TO_DCTIME64: T_DCTIME64

Inputs

```
VAR_INPUT
    in : T_FILETIME64;
END_VAR
```

Name	Type	Description
in	T_FILETIME64	The "Windows File Time" variable to be converted

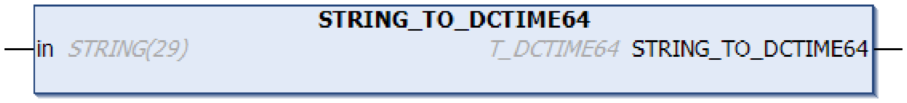
Sample:

```
PROGRAM P_TEST
VAR
    ft : T_FILETIME64;
    dct : T_DCTIME64;
END_VAR

ft := F_GetSystemTime();
dct := FILETIME64_TO_DCTIME64(ft);
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT >= 3.3.16.0

11.2.9 STRING_TO_DCTIME64



The function converts a string to a distributed clock system time variable of type T_DCTIME64 [▸ 130].

FUNCTION STRING_TO_DCTIME64: T_DCTIME64

Inputs

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

Name	Type	Description
in	STRING	The string to be converted The string must have the following format: **'YYYY-MM-DD-hh:mm:ss:nnnnnnnnn'** • YYYY: year; • MM: month; • DD: day; • hh: hour; • mm: minute; • ss: second; • nnnnnnnnn: nanoseconds

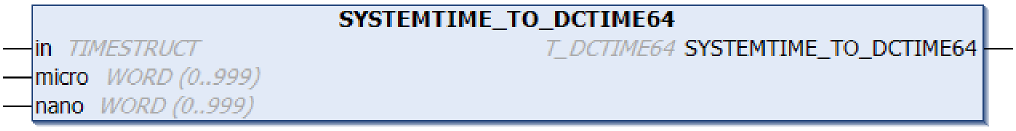
Sample:

See description of the function F_GetCurDcTickTime64 [▶ 102].

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.2.10 SYSTEMTIME_TO_DCTIME64



The function converts a structured Windows system time variable of type TIMESTRUCT to a 64-bit distributed clock system time variable of type T_DCTIME64 [▶ 130]. In the event of a conversion error the function returns the value zero.

FUNCTION SYSTEMTIME_TO_DCTIME64: T_DCTIME64

Inputs

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

Name	Type	Description
in	TIMESTRUCT	The "Windows System Time" variable to be converted
micro	WORD	Microseconds
nano	WORD	Nanoseconds

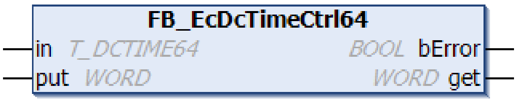
Sample:

```
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_DCTIME64( syst, 456, 789 );
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.2.11 FB_EcDcTimeCtrl64



This function block can be used to read the individual components such as year, month, day etc. of a 64-bit "Distributed Clock System Time" variable of type T_DCTIME64 [▶ 130]. The function block has several A_GETXYZ actions. Once the required action has been called, the value of the XYZ component is available in the “get” output variable. The “put” input variable is currently not used.

The function block features the following tasks:

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

Inputs

```
VAR_IN_OUT
    put : WORD;
END_VAR
```

Name	Type	Description
put	WORD	Input parameter (currently not used)

Inputs/outputs

```
VAR_IN_OUT
    in : T_DCTIME64;
END_VAR
```

Name	Type	Description
in	T_DCTIME64	The "Distributed Clock System Time" variable to be converted

Outputs

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

Name	Type	Description
bError	BOOL	This output is set if an error has occurred during the action call.
get	WORD	Output parameter (year, month, day, etc.)

Example of an implementation in ST:

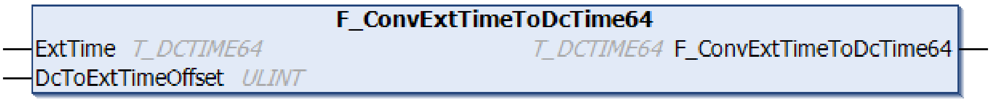
```
PROGRAM P_TEST
VAR
    dcStruct   : DCTIMESTRUCT;
    dcTime     : T_DCTIME64;
    fbCtrl     : FB_EcDcTimeCtrl;

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

dcTime := F_GetCurDcTickTime64();
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 );
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.3 DCTIME64 and ULINT

11.3.1 F_ConvExtTimeToDcTime64



The function `F_ConvExtTimeToDcTime64` converts an external time to the TwinCAT distributed clock system time.

FUNCTION F_ConvExtTimeToDcTime64: T_DCTIME64

Inputs

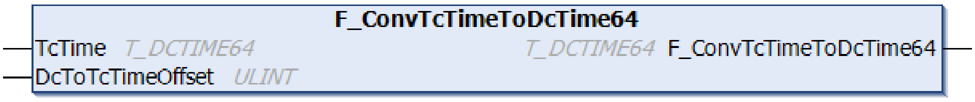
```
VAR_INPUT
   ExtTime           : T_DCTIME64;
   DcToExtTimeOffset : ULINT;
END_VAR
```

Name	Type	Description
ExtTime	T_DCTIME64	External time in TwinCAT "Distributed Clock" system time format
DcToExtTime Offset	ULINT	Time offset between the TwinCAT "Distributed Clock" system time and an external time

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.3.2 F_ConvTcTimeToDcTime64



The function `F_ConvTcTimeToDcTime64` converts the TwinCAT system time to the TwinCAT distributed clock system time.

FUNCTION F_ConvTcTimeToDcTime64: T_DCTIME64

Inputs

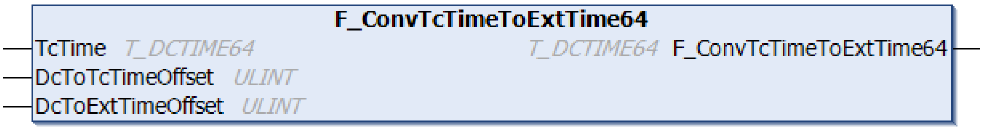
```
VAR_INPUT
   TcTime           : T_DCTIME64;
   DcToTcTimeOffset : ULINT;
END_VAR
```

Name	Type	Description
TcTime	T_DCTIME64	TwinCAT system time in TwinCAT "Distributed Clock" system time format
DcToTcTimeOffset	ULINT	Time offset between the TwinCAT "Distributed Clock" system time and the TwinCAT system time

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.3.3 F_ConvTcTimeToExtTime64



The function `F_ConvTcTimeToExtTime64` converts the TwinCAT distributed clock system time to an external time.

FUNCTION F_ConvTcTimeToExtTime64: T_DCTIME64

Inputs

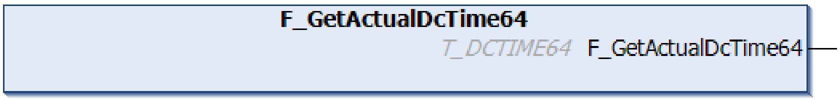
```
VAR_INPUT
   TcTime            : T_DCTIME64;
   DcToTcTimeOffset  : ULINT;
   DcToExtTimeOffset : ULINT;
END_VAR
```

Name	Type	Description
TcTime	T_DCTIME64	TwinCAT system time in "Distributed Clock" format
DcToTcTimeOffset	ULINT	Time offset between the TwinCAT "Distributed Clock" system time and the TwinCAT system time
DcToExtTimeOffset	ULINT	Time offset between the TwinCAT "Distributed Clock" system time and an external time

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.3.4 F_GetActualDcTime64



This function returns the current time in TwinCAT distributed clock system time format (T_DCTIME64 [▸ 130]).

FUNCTION F_GetActaulDCTime: T_DCTIME64

Inputs

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

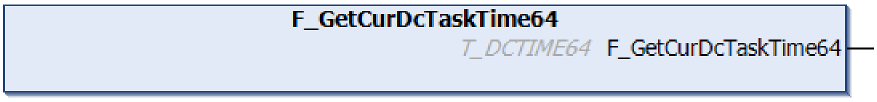
Sample in ST:

```
PROGRAM MAIN
VAR
    actDC  : T_DCTIME64;
    sAct   : STRING;
END_VAR

actDC := F_GetActualDcTime64();
sAct  := DCTIME64_TO_STRING( actDC );
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.3.5 F_GetCurDcTaskTime64



This function returns the task start time, the time at which the task should start, in TwinCAT "Distributed Clock" system time format (T_DCTIME64 [▸ 130]). The function always returns the start time of the task in which it was called.

FUNCTION F_GetCurDcTaskTime64: T_DCTIME64

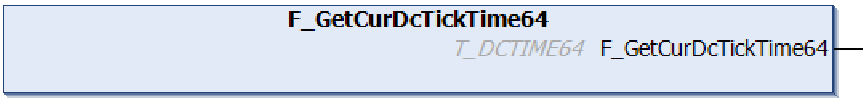
Inputs

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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.3.6 F_GetCurDcTickTime64



The function returns the time of the current (last) tick in TwinCAT distributed clock system time format (T_DCTIME64 [▸ 130]).

FUNCTION F_GetCurDcTickTime64: T_DCTIME64

Inputs

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

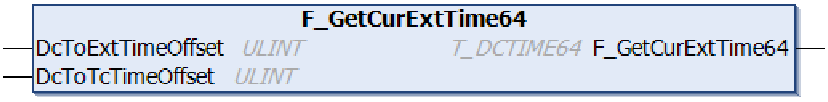
Sample:

```
PROGRAM MAIN
VAR
    tDC : T_DCTIME64;
    sDC : STRING;
    tDCBack : T_DCTIME64;

    sDCZero : STRING;(* DCTIME64 = zero time starts on 01.01.2000 *)
    tDCBackFromZero : T_DCTIME64;

    tDCFromString : T_DCTIME64;
    sDCBackFromString : STRING;
END_VAR

tDC := F_GetCurDcTickTime64();
sDC := DCTIME64_TO_STRING( tDC );
tDCBack := STRING_TO_DCTIME64( sDC );

sDCZero := DCTIME64_TO_STRING( ULARGE_INTEGER(0, 0) );
tDCBackFromZero := STRING_TO_DCTIME64( sDCZero );

tDCFromString := STRING_TO_DCTIME64( '2007-03-09-11:31:09.223456789' );
sDCBackFromString := DCTIME64_TO_STRING( tDCFromString );
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.3.7 F_GetCurExtTime64



The function returns the external time in TwinCAT distributed clock system time format (T_DCTIME64 [▸ 130]).

FUNCTION F_GetCurExtTime64: T_DCTIME64

Inputs

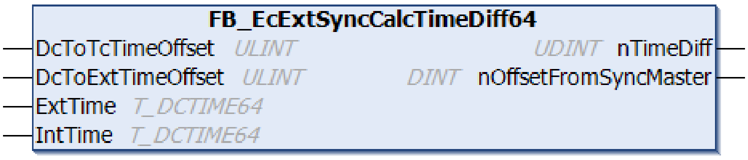
```
VAR_INPUT
   DcToExtTimeOffset : ULINT;
   DcToTcTimeOffset  : ULINT;
END_VAR
```

Name	Type	Description
DcToExtTimeOffset	ULINT	Time offset between the TwinCAT "Distributed Clock" system time and an external time
DcToTcTimeOffset	ULINT	Time offset between the TwinCAT "Distributed Clock" system time and the TwinCAT system time

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.3.8 FB_EcExtSyncCalcTimeDiff64



The function block `FB_EcExtSyncCalcTimeDiff64` calculates the difference between external and internal time, taking into account the time offsets.

Inputs/outputs

```
VAR_IN_OUT
   DcToTcTimeOffset  : ULINT;
   DcToExtTimeOffset : ULINT;
   ExtTime           : T_DCTIME64;
   IntTime           : T_DCTIME64;
END_VAR
```

Name	Type	Description
DcToTcTimeOffset	ULINT	Time offset between the TwinCAT "Distributed Clock" system time and the TwinCAT system time
DcToExtTimeOffset	ULINT	Time offset between the TwinCAT "Distributed Clock" system time and an external time
ExtTime	T_DCTIME64	External time in TwinCAT "Distributed Clock" system time format
IntTime	T_DCTIME64	Internal time in TwinCAT "Distributed Clock" system time format

Outputs

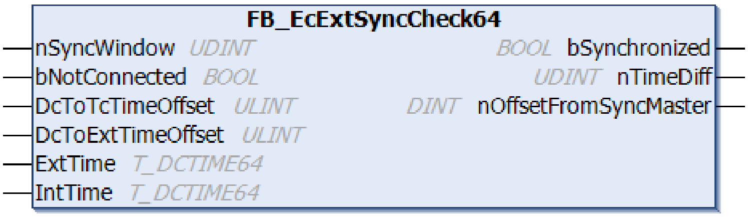
```
VAR_OUTPUT
   nTimeDiff             : UDINT; (*with difference greater than 32 bit timeDiff = 0xfffffff*)
   nOffsetFromSyncMaster : DINT; (*less than 32 bit int Offset = 0x80000000, greater than 32 bit int
Offset = 0x7FFFFFFF*)
END_VAR
```

Name	Type	Description
nTimeDiff	UDINT	If the difference is less than 32 bits, the time difference is returned. If the difference is greater than 32 bits, 16#FFFFFFFF is returned.
nOffsetFromSyncMaster	DINT	If the difference is greater than 32 bits and the offset between internal and DC Time is less than 32 bits, then 16#80000000 is returned. If the difference is greater than 32 bits and the offset between internal and DC Time is greater than 32 bits, then 16#7FFFFFFF is returned.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.3.9 FB_EcExtSyncCheck64



The function block `FB_EcExtSyncCheck64` checks whether the internal and external clocks are synchronous. See function block FB_EcExtSyncCalcTimeDiff64 [▶ 103].

Inputs

```
VAR_INPUT
   nSyncWindow    : UDINT;
   bNotConnected  : BOOL;
END_VAR
```

Name	Type	Description
nSyncWindow	UDINT	Time window within which the internal and external clock are regarded as synchronous.
bNotConnected	BOOL	TRUE = connection to external clock is interrupted.

Inputs/outputs

```
VAR_IN_OUT
   DcToTcTimeOffset  : T_LARGE_INTEGER;
   DcToExtTimeOffset : T_LARGE_INTEGER;
   ExtTime           : T_DCTIME64;
   IntTime           : T_DCTIME64;
END_VAR
```

Name	Type	Description
DcToTcTime Offset	T_LARGE_INTEG ER	Time offset between the TwinCAT "Distributed Clock" system time and the TwinCAT system time
DcToExtTime Offset	T_LARGE_INTEG ER	Time offset between the TwinCAT "Distributed Clock" system time and an external time
ExtTime	T_DCTIME64	External time in TwinCAT "Distributed Clock" system time format
IntTime	T_DCTIME64	Internal time in TwinCAT "Distributed Clock" system time format

Outputs

```
VAR_OUTPUT
   bSynchronized         : BOOL;
   nTimeDiff             : UDINT;
   nOffsetFromSyncMaster : DINT;
END_VAR
```

Name	Type	Description
bSynchronized	BOOL	TRUE = external and internal clock are synchronous
nTimeDiff	UDINT	Current time difference between the two clocks
nOffsetFrom SyncMaster	DINT	Offset to sync master

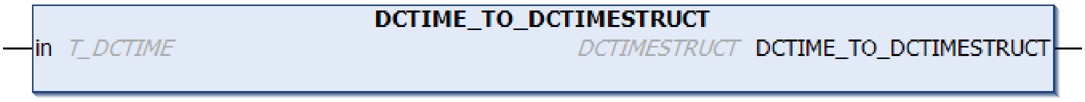
Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4 [obsolete]

11.4.1 [outdated DCTIME]

11.4.1.1 DCTIME_TO_DCTIMESTRUCT



Outdated function

This function is outdated. Use the function DCTIME64_TO_DCTIMESTRUCT [▶ 93] instead.

The function converts a 64-bit distributed clock system time variable of type T_DCTIME [▶ 131] to a structured variable of type DCTIMESTRUCT [▶ 129].

FUNCTION DCTIME_TO_DCTIMESTRUCT: DCTIMESTRUCT

Inputs

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

Name	Type	Description
in	T_DCTIME	The "Distributed Clock System Time" variable to be converted.

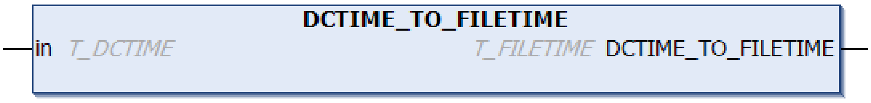
Sample:

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

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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.1.2 DCTIME_TO_FILETIME



Outdated function

This function is outdated. Use the function DCTIME64_TO_FILETIME [▸ 118] instead.

The function converts a 64-bit distributed clock system time variable of type T_DCTIME [▸ 131] to a 64-bit Windows file time variable of type T_FILETIME.

FUNCTION DCTIME_TO_FILETIME: T_FILETIME

Inputs

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

Name	Type	Description
in	T_DCTIME	The "Distributed Clock System Time" variable to be converted

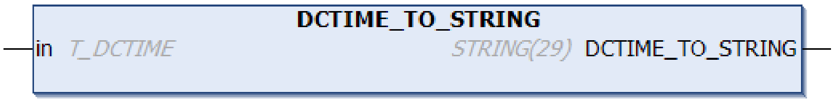
Sample:

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

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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.1.3 DCTIME_TO_STRING



Outdated function

This function is outdated. Use the function DCTIME64_TO_STRING [▸ 94] instead.

The function converts a string to a distributed clock system time variable of type T_DCTIME [▸ 131].

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

- YYYY: year;
- MM: month;
- DD: day;
- hh: hour;
- mm: minute;
- ss: second;
- nnnnnnnnn: nanoseconds;

FUNCTION DCTIME_TO_STRING: STRING(29)

Inputs

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

Name	Type	Description
in	T_DCTIME	The "Distributed Clock System Time" variable to be converted

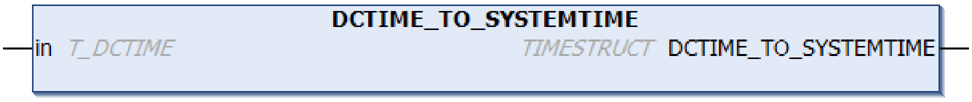
Sample:

See description of the function: F_GetCurDcTickTime [▶ 115].

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.1.4 DCTIME_TO_SYSTEMTIME



Outdated function

This function is outdated. Use the function DCTIME64_TO_SYSTEMTIME [▶ 95] instead.

The function converts a 64-bit distributed clock system time variable of type T_DCTIME [▶ 131] to a structured Windows system time variable of type TIMESTRUCT.

FUNCTION DCTIME_TO_SYSTEMTIME: TIMESTRUCT

Inputs

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

Name	Type	Description
in	T_DCTIME	The "Distributed Clock System Time" variable to be converted

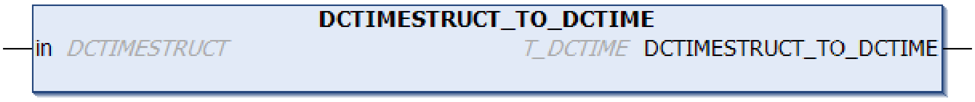
Sample:

```
PROGRAM P_TEST
VAR
    syst : TIMESTRUCT;
END_VAR

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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.1.5 DCTIMESTRUCT_TO_DCTIME



Outdated function

This function is outdated. Use the function DCTIMESTRUCT_TO_DCTIME64 [▶ 95] instead.

The function converts a structured variable of type DCTIMESTRUCT [▶ 129] to a 64-bit distributed clock system time variable of type T_DCTIME [▶ 131].
The structure components wDayofWeek is ignored in the conversion. The structure components wYear must be greater than or equal to 2000 and less than 2584. For invalid values of the structure components the function returns the value zero.

FUNCTION DCTIMESTRUCT_TO_DCTIME: T_DCTIME

Inputs

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

Name	Type	Description
in	DCTIMESTRUCT	The structured variable to be converted

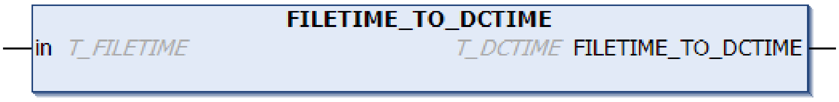
Sample:

```
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 );
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.1.6 FILETIME_TO_DCTIME



Outdated function

The function is outdated. Use the function FILETIME_TO_DCTIME64 [▶ 119] instead.

The function converts a 64-bit Windows file time variable of type T_FILETIME to a 64-bit distributed clock system time variable of type T_DCTIME [▶ 131]. In the event of a conversion error the function returns the value zero.

FUNCTION FILETIME_TO_DCTIME: T_DCTIME

Inputs

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

Name	Type	Description
in	T_FILETIME	The "Windows File Time" variable to be converted

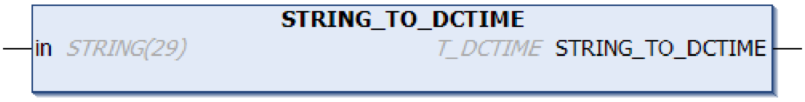
Sample:

```
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);
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.1.7 STRING_TO_DCTIME



Outdated function

This function is outdated. Use the function STRING_TO_DCTIME64 [▸ 96] instead.

The function converts a string to a distributed clock system time variable of type T_DCTIME [▸ 131].

FUNCTION STRING_TO_DCTIME: T_DCTIME

Inputs

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

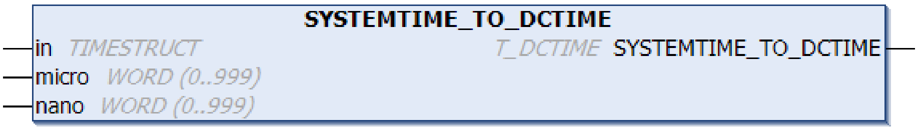
Name	Type	Description
in	STRING	The string to be converted The string must have the following format: **'YYYY-MM-DD-hh:mm:ss:nnnnnnnnn'** • YYYY: year; • MM: month; • DD: day; • hh: hour; • mm: minute; • ss: second; • nnnnnnnnn: nanoseconds;

See description of the function: F_GetCurDcTickTime [▶ 115].

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.1.8 SYSTEMTIME_TO_DCTIME



Outdated function

This function is outdated. Use the function SYSTEMTIME_TO_DCTIME64 [▶ 97] instead.

The function converts a structured Windows system time variable of type TIMESTRUCT to a 64-bit distributed clock system time variable of type T_DCTIME [▶ 131]. In the event of a conversion error the function returns the value zero.

FUNCTION SYSTEMTIME_TO_DCTIME: T_DCTIME

Inputs

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

Name	Type	Description
in	TIMESTRUCT	The "Windows System Time" variable to be converted
Micro	WORD	Microseconds
nano	WORD	Nanoseconds

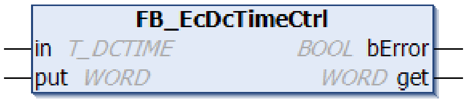
Sample:

```
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 );
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.1.9 FB_EcDcTimeCtrl



Outdated function

This function is outdated. Use the function block FB_EcDcTimeCtrl64 [▶ 98] instead.

This function block can be used to read the individual components such as year, month, day etc. of a 64-bit "Distributed Clock System Time" variable of type T_DCTIME [▶ 131]. The function block has several A_GetXYZ actions. Once the required action has been called, the value of the XYZ component is available in the “get” output variable. The “put” input variable is currently not used.

The function block currently has the following actions:

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

Inputs

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

Name	Type	Description
put	WORD	Input parameter (currently not used)

Inputs/outputs

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

Name	Type	Description
in	T_DCTIME	TwinCAT "Distributed Clock System Time" variable

Outputs

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

Name	Type	Description
bError	BOOL	This output is set if an error has occurred during the action call.
get	WORD	Output parameter (year, month, day, etc.)

Example of an implementation in ST:

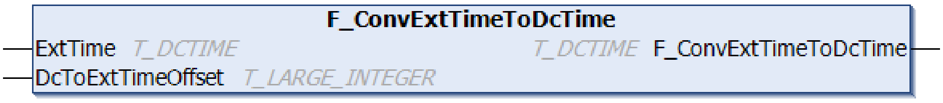
```
PROGRAM P_TEST
VAR
    dcStruct    : DCTIMESTRUCT;
    dcTime      : T_DCTIME;
    fbCtrl      : FB_EcDcTimeCtrl;

    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 );
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.2 [outdated DCTIME and T_LARGE_INTEGER]

11.4.2.1 F_ConvExtTimeToDcTime


F_ConvExtTimeToDcTime
ExtTime T_DCTIME
DcToExtTimeOffset T_LARGE_INTEGER
T_DCTIME F_ConvExtTimeToDcTime

Outdated function

This function is outdated. Use the function F_ConvExtTimeToDcTime64 [▶ 99] instead.

The function `F_ConvExtTimeToDcTime` converts an external time to the TwinCAT distributed clock system time.

FUNCTION F_ConvExtTimeToDcTime: T_DCTIME

Inputs

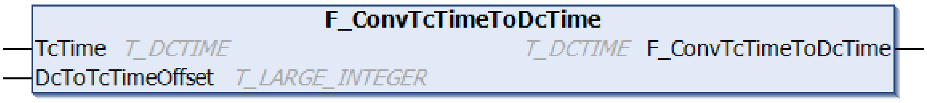
```
VAR_INPUT
   ExtTime           : T_DCTIME;
   DcToExtTimeOffset : T_LARGE_INTEGER;
END_VAR
```

Name	Type	Description
ExtTime	T_DCTIME	External time in TwinCAT "Distributed Clock" system time format
DcToExtTime Offset	T_LARGE_ INTEGER	Time offset between the TwinCAT "Distributed Clock" system time and an external time

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.2.2 F_ConvTcTimeToDcTime


F_ConvTcTimeToDcTime
TcTime T_DCTIME
DcToTcTimeOffset T_LARGE_INTEGER
T_DCTIME F_ConvTcTimeToDcTime

Outdated function

This function is outdated. Use the function F_ConvTcTimeToDcTime64 [▶ 100] instead.

The function F_ConvTcTimeToDcTime64 converts the TwinCAT system time to the TwinCAT distributed clock system time.

FUNCTION F_ConvTcTimeToDcTime: T_DCTIME

Inputs

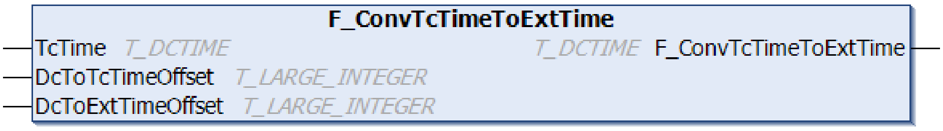
```
VAR_INPUT
   TcTime           : T_DCTIME;
   DcToTcTimeOffset : T_LARGE_INTEGER;
END_VAR
```

Name	Type	Description
TcTime	T_DCTIME	TwinCAT system time in TwinCAT "Distributed Clock" system time format
DcToTcTime Offset	T_LARGE_ INTEGER	Time offset between the TwinCAT "Distributed Clock" system time and the TwinCAT system time

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.2.3 F_ConvTcTimeToExtTime


F_ConvTcTimeToExtTime
TcTime T_DCTIME
DcToTcTimeOffset T_LARGE_INTEGER
DcToExtTimeOffset T_LARGE_INTEGER
T_DCTIME F_ConvTcTimeToExtTime

Outdated function

This function is outdated. Use the function F_ConvTcTimeToExtTime64 [▶ 100] instead.

The function `F_ConcTcTimeToExtTime` converts the TwinCAT distributed clock system time to an external time.

FUNCTION F_ConvTcTimeToExtTime: T_DCTIME

Inputs

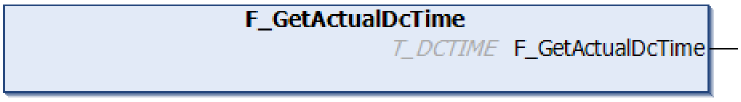
```
VAR_INPUT
   TcTime             :  T_DCTIME;
   DcToTcTimeOffset   :  T_LARGE_INTEGER;
   DcToExtTimeOffset :  T_LARGE_INTEGER;
END_VAR
```

Name	Type	Description
TcTime	T_DCTIME	TwinCAT system time in "Distributed Clock" format
DcToTcTime Offset	T_LARGE_ INTEGER	Time offset between the TwinCAT "Distributed Clock" system time and the TwinCAT system time
DcToExtTime Offset	T_LARGE_ INTEGER	Time offset between the TwinCAT "Distributed Clock" system time and an external time

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.2.4 F_GetActualDcTime



Outdated function

This function is outdated. Use the function F_GetActualDcTime64 [▶ 101] instead.

This function returns the current time in TwinCAT distributed clock system time format (T_DCTIME [▶ 131]).

FUNCTION F_GetActualDcTime: T_DCTIME

Inputs

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

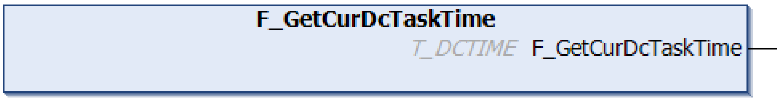
Sample:

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

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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.2.5 F_GetCurDcTaskTime



Outdated function

This function is outdated. Use the function F_GetCurDcTaskTime64 [▶ 101] instead.

This function returns the task start time (time at which the task should start) in TwinCAT distributed clock system time format (T_DCTIME [▸ 131]). The function always returns the start time of the task in which it was called.

FUNCTION F_GetCurDcTaskTime: T_DCTIME

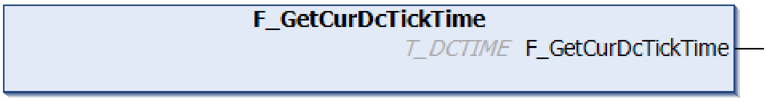
Inputs

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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.2.6 F_GetCurDcTickTime



Outdated function

The function is outdated. Use the function F_GetCurDcTickTime64 [▸ 102] instead.

The function returns the time of the current (last) tick in TwinCAT distributed clock system time format (T_DCTIME [▸ 131]).

FUNCTION F_GetCurDcTickTime: T_DCTIME

Inputs

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

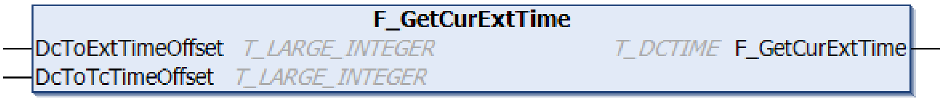
Sample:

```
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 );
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.2.7 F_GetCurExtTime


F_GetCurExtTime
DcToExtTimeOffset T_LARGE_INTEGER T_DCTIME F_GetCurExtTime
DcToTcTimeOffset T_LARGE_INTEGER

Outdated function

This function is outdated. Use the function F_GetCurExtTime64 [▶ 102] instead.

The function returns the external time in TwinCAT distributed clock system time format (T_DCTIME [▶ 131]).

FUNCTION F_GetCurExtTime: T_DCTIME

Inputs

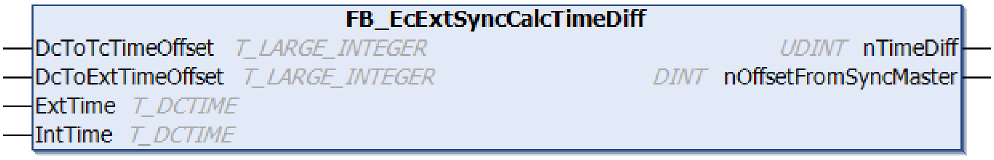
```
VAR_INPUT
   DcToExtTimeOffset : T_LARGE_INTEGER;
   DcToTcTimeOffset  : T_LARGE_INTEGER;
END_VAR
```

Name	Type	Description
DcToExtTimeOffset	T_LARGE_INTEGER	Time offset between the TwinCAT "Distributed Clock" system time and an external time
DcToTcTimeOffset	T_LARGE_INTEGER	Time offset between the TwinCAT "Distributed Clock" system time and the TwinCAT system time

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.2.8 FB_EcExtSyncCalcTimeDiff


FB_EcExtSyncCalcTimeDiff
DcToTcTimeOffset T_LARGE_INTEGER UDINT nTimeDiff
DcToExtTimeOffset T_LARGE_INTEGER DINT nOffsetFromSyncMaster
ExtTime T_DCTIME
IntTime T_DCTIME

Outdated function block

This function block is outdated. Use the function block FB_EcExtSyncCalcTimeDiff64 [▶ 103] instead.

The function block `FB_EcExtSyncCalcTimeDiff` calculates the difference between external and internal time, taking into account the time offsets.

Inputs/outputs

```
VAR_IN_OUT
   DcToTcTimeOffset  : T_LARGE_INTEGER;
   DcToExtTimeOffset : T_LARGE_INTEGER;
   ExtTime           : T_DCTIME;
   IntTime           : T_DCTIME;
END_VAR
```

Name	Type	Description
DcToTcTimeOffset	T_LARGE_INTEGER	Time offset between the TwinCAT "Distributed Clock" system time and the TwinCAT system time
DcToExtTimeOffset	T_LARGE_INTEGER	Time offset between the TwinCAT "Distributed Clock" system time and an external time
ExtTime	T_DCTIME	External time in TwinCAT "Distributed Clock" system time format
IntTime	T_DCTIME	Internal time in TwinCAT "Distributed Clock" system time format

Outputs

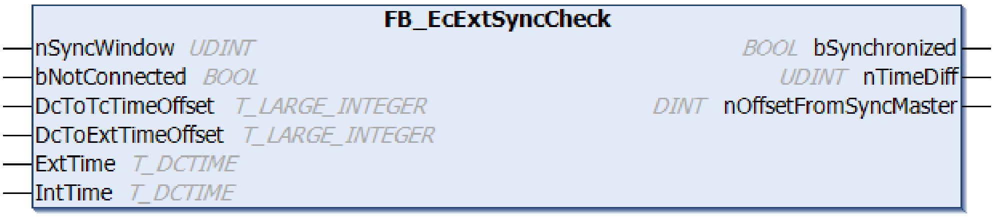
```
VAR_OUTPUT
   nTimeDiff             :  UDINT; (*with difference greater than 32 bit timeDiff = 0xffffffff*)
   nOffsetFromSyncMaster :  DINT; (*less than 32 bit int Offset = 0x80000000, greater than 32 bit
int Offset = 0x7FFFFFFF*)
END_VAR
```

Name	Type	Description
nTimeDiff	UDINT	If the difference is less than 32 bits, the time difference is returned. If the difference is greater than 32 bits, 16#FFFFFFFF is returned.
nOffsetFromSyncMaster	DINT	If the difference is greater than 32 bits and the offset between internal and DC Time is less than 32 bits, then 16#80000000 is returned. If the difference is greater than 32 bits and the offset between internal and DC Time is greater than 32 bits, then 16#7FFFFFFF is returned.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.2.9 FB_EcExtSyncCheck



Outdated function block

This function block is outdated. Use the function block FB_EcExtSyncCheck64 [▶ 104] instead.

The function block `FB_EcExtSyncCheck` checks whether the internal and external clocks are synchronous. See function block FB_EcExtSyncCalcTimeDiff [▶ 116].

Inputs

```
VAR_INPUT
   nSyncWindow    : UDINT;
   bNotConnected  : BOOL;
END_VAR
```

Name	Type	Description
nSyncWindow	UDINT	Time window within which the internal and external clock are regarded as synchronous.
bNotConnected	BOOL	TRUE = connection to external clock is interrupted.

Inputs/outputs

```
VAR_IN_OUT
   DcToTcTimeOffset  : T_LARGE_INTEGER;
   DcToExtTimeOffset : T_LARGE_INTEGER;
   ExtTime           : T_DCTIME;
   IntTime           : T_DCTIME;
END_VAR
```

Name	Type	Description
DcToTcTimeOffset	T_LARGE_INTEGER	Time offset between the TwinCAT "Distributed Clock" system time and the TwinCAT system time
DcToExtTimeOffset	T_LARGE_INTEGER	Time offset between the TwinCAT "Distributed Clock" system time and an external time
ExtTime	T_DCTIME	External time in TwinCAT "Distributed Clock" system time format
IntTime	T_DCTIME	Internal time in TwinCAT "Distributed Clock" system time format

Outputs

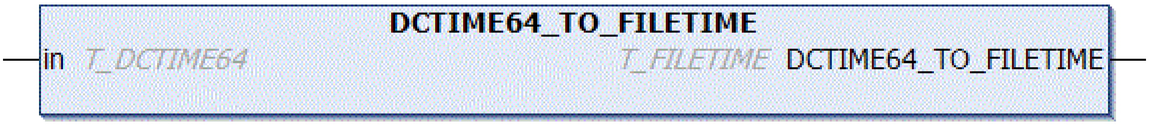
```
VAR_OUTPUT
   bSynchronized          : BOOL;
   nTimeDiff              : UDINT;
   nOffsetFromSyncMaster  : DINT;
END_VAR
```

Name	Type	Description
bSynchronized	BOOL	TRUE = external and internal clock are synchronous
nTimeDiff	UDINT	Current time difference between the two clocks
nOffsetFromSyncMaster	DINT	Offset to sync master

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.3 DCTIME64_TO_FILETIME



The function converts a 64-bit distributed clock system time variable of type T_DCTIME64 [▶ 130] to a 64-bit Windows file time variable of type T_FILETIME.

FUNCTION DCTIME64_TO_FILETIME: T_FILETIME

Inputs

```
VAR_INPUT
    in : T_DCTIME64:
END_VAR;
```

Name	Type	Description
in	T_DCTIME64	The "Distributed Clock System Time" variable to be converted

Sample:

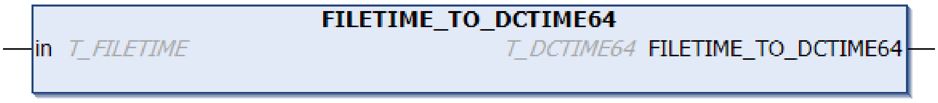
```
PROGRAM P_TEST
VAR
    ft  : T_FILETIME;
    dct : T_DCTIME64;
```

```
END_VAR

dct := F_GetCurDcTickTime64();
ft  := DCTIME64_TO_FILETIME(dct);
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

11.4.4 FILETIME_TO_DCTIME64



The function converts a 64-bit Windows file time variable of type T_FILETIME to a 64-bit distributed clock system time variable of type T_DCTIME64 [▶ 130]. In the event of a conversion error the function returns the value zero.

FUNCTION FILETIME_TO_DCTIME64: T_DCTIME64

Inputs

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

Name	Type	Description
in	T_FILETIME	The "Windows File Time" variable to be converted

Sample:

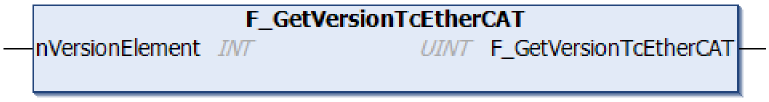
```
PROGRAM P_TEST
VAR
    fbSysFileTime : GETSYSTEMTIME;
    ft : T_FILETIME;
    dct : T_DCTIME64;
END_VAR

fbSysFileTime(timeLoDW=>ft.dwLowDateTime, timeHiDW=>ft.dwHighDateTime);
dct := FILETIME_TO_DCTIME64(ft);
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

12 [Obsolete]

12.1 F_GetVersionTcEtherCAT



Outdated function

This function is outdated. Use the global structure instance stLibVersion_Tc2_EtherCAT instead

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

FUNCTION F_GetVersionTcEtherCAT : UINT

Inputs

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

Name	Type	Description
nVersionElement	INT	Version element to be read. Possible parameters: • 1 : major number; • 2 : minor number; • 3 : revision number;

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13 Data types

13.1 E_EcAdressingType

Addressing in EtherCAT is either position-dependent (eAdressingType_AutoInc), based on a fixed, configured address (eAdressingType_Fixed) or applies to all slaves (eAdressingType_Broadcast).

```
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 address
) *)
                (* EtherCAT commands: FPRD, FPWR, FPRW *)
eAdressingType_Broadcast    (* Adress all slaves. *)
                (* EtherCAT commands: BRD, BWR, BRW *)
);
END_TYPE
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.2 E_EcFoeMode

Access mode for the “File access over EtherCAT” mailbox protocol.

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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.3 E_EcMbxProtType

Supported EtherCAT mailbox protocol types.

```
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
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.4 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
```

Name	Type	Description
portA	UDINT	CRC error counter of Port A
portB	UDINT	CRC error counter of Port B
portC	UDINT	CRC error counter of Port C

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.5 ST_EcCrcErrorEx

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

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

Name	Type	Description
portA	UDINT	CRC error counter of Port A
portB	UDINT	CRC error counter of Port B
portC	UDINT	CRC error counter of Port C
portD	UDINT	CRC error counter of Port D

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.6 ST_EcLastProtErrInfo

The structure ST_EcLastProtErrInfo contains additional error information relating to the most recent 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
```

Name	Type	Description
ownAddr	ST_AmsAddr	Own AMS address (address of the communication device that queries the error information)
orgAddr	ST_AmsAddr	AMS address of the error originator (address of communication device that has triggered or caused the protocol error)
errCode	UDINT	Mailbox protocol error number [▶ 134] (SoE, CoE, FoE error code)
binDesc	ARRAY[0..MAX_STRING_LENGTH] OF BYTE	Additional error information as binary data. The additional error information is device-specific and can include a string or binary data, for example.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.7 ST_EcMasterStatistic

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

Name	Type	Description
nSysTime	UDINT	System time in µs
nCycFrameCnt	UDINT	Number of cyclic EtherCAT frames
nCycFrameMissedCnt	UDINT	Number of lost cyclic EtherCAT frames
nQueuedFrameCnt	UDINT	Number of acyclic EtherCAT frames
nQueuedFrameMissedCnt	UDINT	Number of lost acyclic EtherCAT frames

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.8 ST_EcSlaveConfigData

The structure `ST_EcSlaveConfigData` contains the EtherCAT configuration data for an EtherCAT slave device.

```
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
```

Name	Type	Description
nEntries	WORD	Used internally
nAddr	WORD	Address of an EtherCAT slave
sType	STRING	EtherCAT type of a slave
sName	STRING	Name of an EtherCAT slave
nDevType	DWORD	EtherCAT device type of a slave
stSlaveIdentity	ST_EcSlaveIdentity	Identity of an EtherCAT slave (see ST_EcSlaveIdentity [▶ 124])
nMailboxOutSize	WORD	Mailbox OutSize of an EtherCAT slave
nMailboxInSize	WORD	Mailbox InSize of an EtherCAT slave
nLinkStatus	BYTE	Link status of an EtherCAT slave

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.9 ST_EcSlaveIdentity

The structure `ST_EcSlaveIdentity` contains the EtherCAT identity data for an EtherCAT slave device.

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

Name	Type	Description
vendorId	UDINT	Vendor-ID of the slave device
productCode	UDINT	Product code of the slave device
revisionNo	UDINT	Indicates the revision number of the slave device.
serialNo	UDINT	Indicates the serial number of the slave device.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.10 ST_EcSlaveScannedData

The `ST_EcSlaveScannedData` structure contains the EtherCAT configuration data of a scanned EtherCAT slave device.

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

Name	Type	Description
nEntries	WORD	Used internally
nAddr	WORD	Address of an EtherCAT slave
stSlaveIdentity	ST_EcSlaveIdentity	Identity of an EtherCAT slave (see ST_EcSlaveIdentity [▶ 124])
ndlStatusReg	WORD	Link status of an EtherCAT slave from ESC register 0110/0111$_{hex}$ or 272/273$_{dec}$. Status 0 is displayed if the slave cannot be reached or is offline. The "port number <=> socket/Ebus contact" assignment can be found in the respective device documentation. Unless described otherwise, port 0 is the left-hand Ebus contact of an EL/ES terminal or the RJ45 socket of an EP box, port 1 is the right-hand outgoing Ebus contact/RJ45 socket.

The bit meanings of ndlStatusReg are:

Bit	Meaning
0	internal use
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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.11 ST_EcSlaveState

The structure `ST_EcSlaveState` contains the EtherCAT state and the link state of an EtherCAT slave device.

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

Name	Type	Description
deviceState	BYTE	EtherCAT state of a slave (See deviceState)
linkState	BYTE	Link state of an EtherCAT slave (see linkState)

deviceState

EtherCAT state 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

In addition, the following bits can be set:

Constant	Value	Description
EC_DEVICE_STATE_ERROR	0x10	State machine error in the EtherCAT slave
EC_DEVICE_STATE_INVALID_VPRS	0x20	Invalid vendor ID, product code, revision number or serial number
EC_DEVICE_STATE_INITCMD_ERROR	0x40	Error during sending of initialization commands.
EC_DEVICE_STATE_DISABLED	0x80	Slave is disabled

linkState

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

Constant	Value	Description
EC_LINK_STATE_OK	0x00	
EC_LINK_STATE_NOT_PRESENT	0x01	No EtherCAT communication with the EtherCAT slave
EC_LINK_STATE_LINK_WITHOUT_COMM	0x02	Error at port X (specified through EC_LINK_STATE_PORT_A/B/C/D). The port has a link, but no communication is possible via this port.
EC_LINK_STATE_MISSING_LINK	0x04	Missing link at port X (specified through EC_LINK_STATE_PORT_A/B/C/D).
EC_LINK_STATE_ADDITIONAL_LINK	0x08	Additional link at port X (specified through 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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.12 ST_EcSlaveStateBits

The structure `ST_EcSlaveStateBits` contains the EtherCAT state and the link state 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
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.13 ST_EcSlaveStateBitsEx

The structure `ST_EcSlaveStateBitsEx` contains the EtherCAT state and the link state of an EtherCAT slave device.

```
TYPE ST_EcSlaveStateBitsEx:
STRUCT
    bInit             : BOOL;
    bPreop            : BOOL;
    bBootStrap        : BOOL;
    bSafeOp           : BOOL;
    bOp               : BOOL;
    bError            : BOOL;
    bInvVPRS          : BOOL;
    bInitCmdError     : BOOL;
    bDisabled         : BOOL;
    bLinkNotPresent   : BOOL;
    bLinkWithoutComm  : BOOL;
    bLinkMissing      : BOOL;
    bAdditionalLink   : BOOL;
    bPortA            : BOOL;
    bPortB            : BOOL;
    bPortC            : BOOL;
    bPortD            : BOOL;
END_STRUCT
END_TYPE
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.14 ST_EcAbortCode

```
TYPE ST_EcSlaveScannedData:
STRUCT
    sNetId        : T_AmsNetIdArr;
    nPort           : UINT;
    nAbortCode : UDINT;
END_STRUCT
END_TYPE
```

Name	Type	Description
sNetId	T_AmsNetIdArr	AmsNetId of the sender who aborted the command.
nPort	UINT	Port of the sender who aborted the command.
nAbortCode	UDINT	Abort code

13.15 ST_PortAddr

The structure `ST_PortAddr` contains EtherCAT topology information for EtherCAT slave device. EtherCAT slave devices typically have 2 to 4 ports.

```
TYPE ST_PortAddr:
STRUCT
    portA : UINT;
    portB : UINT;
    portC : UINT;
    portD : UINT;
END_STRUCT
END_TYPE
```

Name	Type	Description
portA	UINT	Address of the previous EtherCAT slave at port A of the current EtherCAT slave
portB	UINT	Address of the optional subsequent EtherCAT slave at port B of the current EtherCAT slave
portC	UINT	Address of the optional subsequent EtherCAT slave at port C of the current EtherCAT slave
portD	UINT	Address of the optional subsequent EtherCAT slave at port D of the current EtherCAT slave

13.16 ST_TopologyDataEx

The structure `ST_TopologyDataEx` contains information on EtherCAT topology and hot-connect groups.

```
TYPE ST_TopologyDataEx:
STRUCT
   nOwnPhysicalAddr   :  UINT;
   nOwnAutoIncAddr    :  UINT;
   stPhysicalAddr     :  ST_PortAddr;
   stAutoIncAddr      :  ST_PortAddr;
   aReserved1         :  ARRAY [0..3] OF UDINT;
   nStatusBits        :  DWORD;
   nHCSlaveCountCfg   :  UINT; (*nStatusBits.0 = TRUE: DcSupprt;.1 = TRUE: DC64Supprt; .2=TRUE: Slave
Present following hot connect info requires runtime >= TC 2.11 R3 B2246 nStatusBits.3 = TRUE: HotCon
nectGroupStart; .4 = HotConnectSlave; .5 = TRUE: HotCOnnectInvalidB; .6 = TRUE: HotConnectInvalidC;
.7 = TRUE: HotConnectInvalidD*)
   nHCSlaveCountAct   :  UINT;
   aReserved2         :  ARRAY [0..4] OF UDINT;
END_STRUCT
END_TYPE
```

Name	Type	Description
nOwnPhysicalAddr	UINT	Dedicated physical EtherCAT address of the EtherCAT slave device
nOwnAutoIncAddr	UINT	Dedicated auto-increment EtherCAT address of the EtherCAT slave device
stPhysicalAddr	ST_PortAddr	Physical address information of the EtherCAT slave devices at port A…D
stAutoIncAddr	ST_PortAddr	Auto-increment address information of the EtherCAT slave devices at port A…D
aReserved1	ARRAY [0..3] OF UDINT	Reserved
nStatusBits	DWORD	nStatusBits.0 = TRUE: Distributed clock is supported nStatusBits.1 = TRUE: Distributed Clock is supported (64-bit) nStatusBits.2 = TRUE: slave is present nStatusBits.3 = TRUE: slave is start node of a Hot Connect group nStatusBits.4 = TRUE: slave is in a Hot Connect group nStatusBits.5 = TRUE: Hot Connect is invalid at port B nStatusBits.6 = TRUE: Hot Connect is invalid at port C nStatusBits.7 = TRUE: Hot Connect is invalid at port D
nHCSlaveCountCfg	UINT	Configured number of Hot Connect group devices
nHCSlaveCountAct	UINT	Found number of Hot Connect group devices
aReserved2	ARRAY [0..4] OF UDINT	Reserved

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.17 DCTIMESTRUCT

Structured TwinCAT "Distributed Clock System Time" format. The smallest unit is a 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
```

Name	Type	Description
wYear	WORD	Year: 2000 ~ 2584
wMonth	WORD	Month: 1 ~ 12 (January = 1, February = 2 etc.)
wDayOfWeek	WORD	Day of the week: 0 ~ 6 (Sunday = 0, Monday = 1 etc.)
wDay	WORD	Day of the month: 1 ~ 31
wHour	WORD	Hour: 0 ~ 23
wMinute	WORD	Minute: 0 ~ 59
wSecond	WORD	Second: 0 ~ 59
wMilliseconds	WORD	Millisecond: 0 ~ 999
wMicroseconds	WORD	Microsecond: 0 ~ 999
wNanoseconds	WORD	Nanosecond: 0 ~ 999

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.18 T_DCTIME32

32-bit TwinCAT distributed clock system time format. The smallest unit is a nanosecond.

This 32-bit DC system time is formed from the full absolute 64-bit DC system time (T_DCTIME [▸ 131]) by using only the lowest-order 32 bits. This means the property of an absolute unique time is lost, and it is assumed that this 32-bit time is only used within a narrow time window of ± 2,147 seconds around the current system time, to ensure that it is unambiguous. There are many applications in which this assumption is possible.
If this assumption is violated, errors may occur in the interpretation and further processing of this time.

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

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.19 T_DCTIME64

The data type `T_DCTIME64` represents the distributed clock system time (abbreviated as DC time) as a linear 64-bit integer value. The time is expressed in nanoseconds since 1/1/2000 UTC. The smallest unit is a nanosecond.

```
TYPE T_DCTIME64 : ULINT;
END_TYPE
```

Useful distributed clock system time constants	Description
EC_DCTIME_DELTA_OFFSET64	Number of 100-nanosecond ticks between 1601-01-01 and 2000-01-01. This is the difference between the "Windows File Time" and the "Distributed Clock System Time".
EC_DCTIME_DATEDELTA_OFFSET	Number of days that have passed between the year zero and January 1, 2000
EC_DCTIME_TICKSPERMSEC64	Number of distributed clock system time nanoseconds per millisecond
EC_DCTIME_TICKSPERSEC64	Number of distributed clock system time nanoseconds per second
EC_DCTIME_TICKSPERDAY64	Number of distributed clock system time nanoseconds per day

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.20 T_DCTIME

Outdated data type

This data type is outdated. Use the data type T_DCTIME64 [▶ 130] instead.

The data type T_DCTIME represents the distributed clock system time (abbreviated as DC time) as a linear 64-bit integer value. The time is expressed in nanoseconds since 1.1.2000 UTC.
The data type is represented as two 32-bit DWORD variables, so that it can easily be processed in the PLC. Operations (addition and subtraction of times) can be executed with ui64 functions from the Tc2_Utilities library.

```
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 01.01.1601 and 01.01.2000. This is the difference between the Windows file time and the distributed clock system time.
EC_DCTIME_DATEDELTA_OFFSET	Number of days that have passed between the year zero and 1 January 2000
EC_DCTIME_TICKSPERMSEC	Number of distributed clock system time nanoseconds per millisecond
EC_DCTIME_TICKSPERSEC	Number of distributed clock system time nanoseconds per second
EC_DCTIME_TICKSPERDAY	Number of distributed clock system time nanoseconds per day

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

13.21 T_HFoe

“File access over EtherCAT” handle. Before the handle can be used, it must be initialized once with the function block FB_EcFoeOpen [▶ 71]. The variables of this structured type must 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
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

14 Constants

14.1 Global constants

VAR_GLOBAL CONSTANT

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

(*ethercat commands*)
EC_CMD_TYPE_APRD :BYTE :=1;
EC_CMD_TYPE_APWR :BYTE :=2;
EC_CMD_TYPE_APRW :BYTE :=3;
EC_CMD_TYPE_FPRD :BYTE :=4;
EC_CMD_TYPE_FPWR :BYTE :=5;
EC_CMD_TYPE_FPRW :BYTE :=6;
EC_CMD_TYPE_BRD  :BYTE :=7;
EC_CMD_TYPE_BWR  :BYTE :=8;
EC_CMD_TYPE_BRW  :BYTE :=9;
EC_CMD_TYPE_LRD  :BYTE :=10;
EC_CMD_TYPE_LWR  :BYTE :=11;
EC_CMD_TYPE_LRW  :BYTE :=12;

(* 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_MASTER_SENDCMD :UDINT :=16#00000008;
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;
EC_ADS_IGRP_MASTER_SLAVE_ABNORMAL_STATE_CHANGES :UDINT :=16#00000013;
EC_ADS_IGRP_MASTER_SLAVE_SETPRESENT_CHANGES :UDINT :=16#00000016;
EC_ADS_IGRP_MASTER_DEVICESTATE :UDINT :=16#00000045;
EC_ADS_IGRP_MASTER_COUNT_FRAME :UDINT :=16#00000048;

(* 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_DATEDELTA_OFFSET : DWORD := 730120; (* Number of past days since year zero until 1 January
 2000 *)
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_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 *)

EC_DCTIME_DELTA_OFFSET64 : ULINT := ULINT#16#01BF53EB_256D4000;
(* Number of 100ns ticks between 1.1.1601 and 1.1.2000 *)
EC_DCTIME_TICKSPERMSEC64 : ULINT := ULINT#16#00000000_000F4240;
(* Number of nanosecond ticks per millisecond *)
EC_DCTIME_TICKSPERSEC64 : ULINT := ULINT#16#00000000_3B9ACA00;
(* Number of nanosecond ticks per second *)
EC_DCTIME_TICKSPERDAY64 : ULINT := ULINT#16#00004E94_914F0000;
(* Number of nanosecond ticks per day *)

bSeqReadDrvAttrAndValue : BOOL := FALSE;
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

14.2 Library version

All libraries have a certain version. The version is indicated in the PLC library repository, for example. A global constant contains the information about the library version:

Global_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc2_EtherCAT : ST_LibVersion;
END_VAR
```

stLibVersion_Tc2_EtherCAT: Version information of the Tc2_EtherCAT library (type: ST_LibVersion)

To check whether the version you have is the version you need, use the function F_CmpLibVersion (defined in the Tc2_System library).

All other options for comparing library versions, which you may know from TwinCAT 2, are outdated!

14.3 EtherCAT 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 dictionary. *)
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 mapped
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 device. *
)
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 service para
meter does not match *)
EC_COE_PROTERR_DATA_SIZE1 : UDINT := 16#06070012; (* Data type does not match, length of service par
ameter too high *)
EC_COE_PROTERR_DATA_SIZE2 : UDINT := 16#06070013; (* Data type does not match, length of service par
ameter 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 (only for wri
te 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 the appli
cation. *)
EC_COE_PROTERR_TRANSFER1 : UDINT := 16#08000021; (* Data cannot be transferred or stored to the appl
ication because of local control. *)
EC_COE_PROTERR_TRANSFER2 : UDINT := 16#08000022; (* Data cannot be transferred or stored to the appl
ication because of the present device state. *)
EC_COE_PROTERR_DICTIONARY : UDINT := 16#08000023; (* Object dictionary dynamic generation fails or n
o object dictionary is present (e.g. object dictionary is generated from file and generation fails b
ecause of an file error). *)
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.0	PC or CX (x86, x64, Arm®)	Tc2_EtherCAT

15 Sample

Sample project and sample configuration for diagnostics

See https://infosys.beckhoff.com/content/1033/tcplclib_tc2_ethercat/Resources/2364613387.zip

Trademark statements

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Third-party trademark statements

Arm, Arm9 and Cortex are trademarks or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

More Information:
www.beckhoff.com/te1000

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