

BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | PLC Library: Tc2_DataExchange

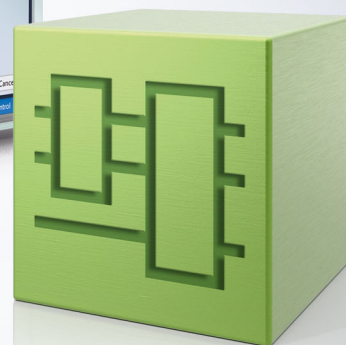
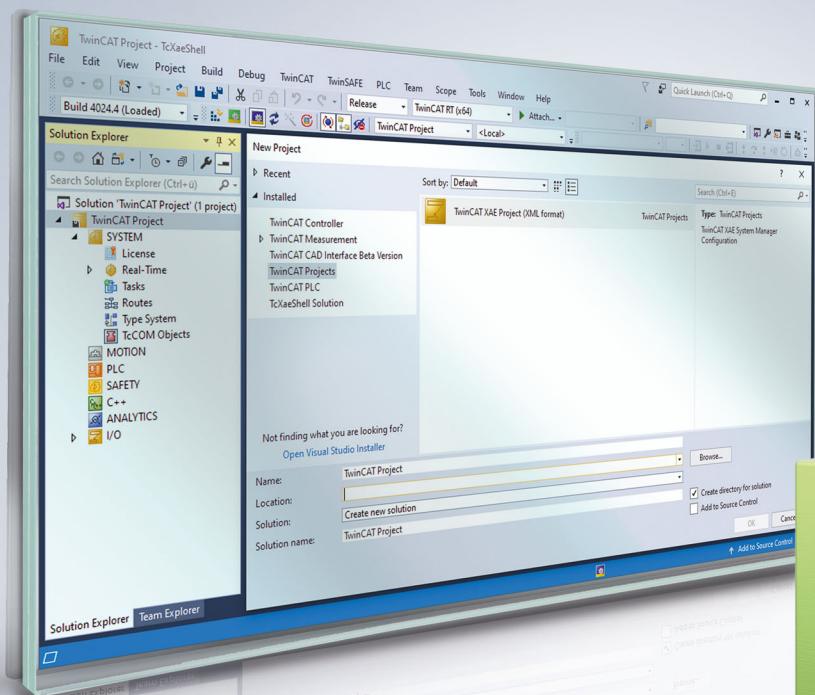


Table of contents

1 Foreword	5
1.1 Notes on the documentation	5
1.2 For your safety	5
1.3 Notes on information security	7
2 Introduction	8
3 Event driven data exchange	9
3.1 FB_ReadAdsSymByName	9
3.2 FB_WriteAdsSymByName	10
3.3 FB_WriteBoolOnDelta	11
3.4 FB_WriteByteOnDelta	13
3.5 FB_WriteWordOnDelta	14
3.6 FB_WriteDWordOnDelta	16
3.7 FB_WriteRealOnDelta	17
3.8 FB_WriteLRealOnDelta	19
4 Watchdog function blocks	21
4.1 FB_CheckWatchdog	21
4.2 FB_WriteWatchdog	22
5 Data types	24
5.1 E_AdsComMode	24
6 Global constants	25
6.1 Library version	25

1 Foreword

1.1 Notes on the documentation

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

For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.

The qualified personnel is obliged to always use the currently valid documentation.

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

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without notice.

No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

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

If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.



EtherCAT® is 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.

1.2 For your safety

Safety regulations

Read the following explanations for your safety.

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

Exclusion of liability

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

Personnel qualification

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

Signal words

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

Personal injury warnings

DANGER

Hazard with high risk of death or serious injury.

WARNING

Hazard with medium risk of death or serious injury.

CAUTION

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

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product



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

1.3 Notes on information security

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

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

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

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

2 Introduction

The present function blocks simplify event-driven data exchange between the TwinCAT PLC runtime system and/or other ADS devices (TwinCAT NC, Bus Terminal Controller, ...).

The FB_WriteXXXOnDelta() function block implements a write procedure when the input signal rises above or falls below a specified limit value. The frequency with which the input signal is examined can be set. Event-driven data writing minimizes the loading on the fieldbus. If an error occurs during transmission, the process is repeated until the connection is established once more. All data types supported in the TwinCAT PLC are permitted as source and destination variables. Symbol names are also supported.

Watchdog function blocks are available to monitor individual communication partners. The device that is to be monitored cyclically transmits an incrementing counter. A check is made at the receiver to see that the counter state changes within a specific time.

Write/Read Blocks

Name	Description
FB_ReadAdsSymByName [► 9]	Reads a variable of any desired data type by variable name
FB_WriteAdsSymByName [► 10]	Writes a variable of any desired data type by variable name
FB_WriteBoolOnDelta [► 11]	Writes a variable of type BOOLEAN in response to an event.
FB_WriteByteOnDelta [► 13]	Writes a variable of type BYTE in response to an event.
FB_WriteWordOnDelta [► 14]	Writes a variable of type WORD in response to an event.
FB_WriteDWordOnDelta [► 16]	Writes a variable of type DWORD in response to an event.
FB_WriteRealOnDelta [► 17]	Writes a variable of type REAL in response to an event.
FB_WriteLRealOnDelta [► 19]	Writes a variable of type LREAL in response to an event.

Monitoring Blocks

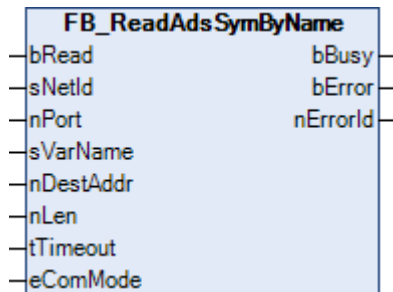
Name	Description
FB_WriteWatchdog [► 22]	Writes a watchdog signal (an incrementing counter) cyclically
FB_CheckWatchdog [► 21]	Monitors the received watchdog signal

3 Event driven data exchange

Write/Read Blocks

Name	Description
FB_ReadAdsSymByName [► 9]	Reads a variable of any desired data type by variable name
FB_WriteAdsSymByName [► 10]	Writes a variable of any desired data type by variable name
FB_WriteBoolOnDelta [► 11]	Writes a variable of type BOOLEAN in response to an event.
FB_WriteByteOnDelta [► 13]	Writes a variable of type BYTE in response to an event.
FB_WriteWordOnDelta [► 14]	Writes a variable of type WORD in response to an event.
FB_WriteDWordOnDelta [► 16]	Writes a variable of type DWORD in response to an event.
FB_WriteRealOnDelta [► 17]	Writes a variable of type REAL in response to an event.
FB_WriteLRealOnDelta [► 19]	Writes a variable of type LREAL in response to an event.

3.1 FB_ReadAdsSymByName



The function block FB_ReadAdsSymByName enables reading of any value from another controller using the symbol name.

On a positive edge at the *bRead* input the function block reads the value of the variable *sVarName* from the selected ADS device (e.g. PLC). The ADS device is indicated by the AMS-NetId (*sNetId*) and the AMS Port number (*nPort*). The value is written into the variable to which *nDestAddr* points.

The internal mode of operation of the function block can be changed with the aid of the *eComMode* input:

- *eComMode* := *eAdsComModeSecureCom*: After each read process, the handle of the PLC variable is released again. This mode should be used when values are exchanged very slowly.
- *eComMode* := *eAdsComModeFastCom*: As long as the inputs *sVarName*, *sNetId* and *nPort* do not change, the handle of the PLC variable will not be released after each read process. This mode should be used when values are exchanged very frequently.

Inputs

```

VAR_INPUT
    bRead          : BOOL;
    sNetId         : T_AmsNetId;
    nPort          : T_AmsPort := 851;
    sVarName       : STRING(255);
    nDestAddr      : PVOID;
    nLen           : UDINT;
    tTimeout       : TIME := DEFAULT_ADS_TIMEOUT;
    eComMode       : E_AdsComMode := eAdsComModeSecureCom;
END_VAR

```

Name	Type	Description
bRead	BOOL	The function block reads the content of the variables <i>sVarName</i> of the selected ADS device and writes it to the variable to which the pointer <i>nDestAddr</i> points.

sNetId	T_AmsNetID	AMS-NetId of the ADS device from which the value is to be read.
nPort	T_AmsNetID	AMS Port number of the ADS device from which the value is to be read.
sVarName	STRING(255)	Symbol name of the variable to be read on the selected ADS device (max. 255 characters).
nDestAddr	PVOID	Address of the variable into which the read value is written.
nLen	UDINT	Length of the variable to be read in bytes.
tTimeout	TIME	Time until processing is aborted.
eComMode	E_AdsComMode [► 24]	Enum used to specify whether the handle of the PLC variable is released again after each read process.

🔌 Outputs

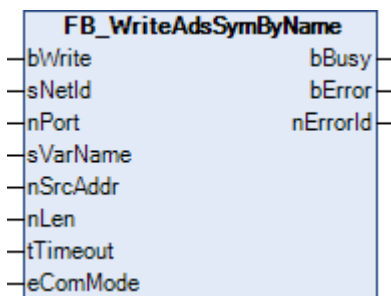
```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrorId   : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	The transmission is active.
bError	BOOL	An error occurred during the transmission.
nErrorId	UDINT	ADS error number if an error has occurred.

Requirements

Development environment	required TC3 PLC library
TwinCAT v3.1.0	Tc2_DataExchange

3.2 FB_WriteAdsSymByName



The function block FB_WriteAdsSymByName enables writing of any value to another controller using the symbol name.

On a positive edge at the input *bWrite* the function block writes the value to which the pointer *nSrcAddr* points into the variable *sVarName* of the selected ADS device (e.g. PLC). The ADS device is indicated by the AMS-NetId (*sNetId*) and the AMS Port number (*nPort*).

The internal mode of operation of the function block can be changed with the aid of the *eComMode* input:

- *eComMode* := *eAdsComModeSecureCom*: After each write process, the handle of the PLC variable is released again. This mode should be used when values are exchanged very slowly.
- *eComMode* := *eAdsComModeFastCom*: As long as the inputs *sVarName*, *sNetId* and *nPort* do not change, the handle of the PLC variable is not released after each write operation. This mode should be used when values are exchanged very frequently.

Inputs

```

VAR_INPUT
    bRead          : BOOL;
    sNetId         : T_AmsNetId;
    nPort          : T_AmsPort := 851;
    sVarName       : STRING(255);
    nSrcAddr       : PVOID;
    nLen           : UDINT;
    tTimeout       : TIME := DEFAULT_ADS_TIMEOUT;
    eComMode       : E_AdsComMode := eAdsComModeSecureCom;
END_VAR

```

Name	Type	Description
bWrite	BOOL	This function block writes the contents of the variable to which pointer <i>nSrcAddr</i> points into the variable <i>sVarName</i> of the selected ADS device.
sNetId	T_AmsNetID	AMS NetID of the ADS device to which the value is to be transmitted.
nPort	T_AmsNetID	AMS port number of the ADS device to which the value is to be transmitted.
sVarName	STRING(255)	Symbol name of the variable to be written on the selected ADS device (max. 255 characters).
nSrcAddr	PVOID	Address of the variable in which the value to written is located.
nLen	UDINT	Length in bytes of the variable to be written.
tTimeout	TIME	Time until processing is aborted.
eComMode	E_AdsComMode [► 24]	Enum used to specify whether the handle of the PLC variable is released again after each write process.

Outputs

```

VAR_OUTPUT
    bBusy          : BOOL;
    bError         : BOOL;
    nErrorId       : UDINT;
END_VAR

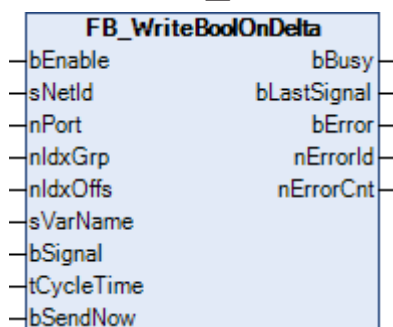
```

Name	Type	Description
bBusy	BOOL	The transmission is active.
bError	BOOL	An error occurred during the transmission.
nErrorId	UDINT	ADS error number if an error has occurred.

Requirements

Development environment	required TC3 PLC library
TwinCAT v3.1.0	Tc2_DataExchange

3.3 FB_WriteBoolOnDelta



The function block enables event-driven writing of a variable of type BOOLEAN.

The function block FB_WriteBoolOnDelta checks cyclically whether the value at the input *bSignal* has changed. The cycle time for checking is determined by the parameter *tCycleTime*. If 0 s is given for *tCycleTime*, the input signal is examined during every PLC cycle. If a change is detected, the value of the signal is sent to the specified ADS device. The receiver is addressed by means of the AMS-NetId and the port number (see also ADS Device Identification). The position within the receiver is specified by the index group/index offset or by the symbol name. Usually this is the input image or the flags area.

If the input *bEnable* is set to FALSE, no further signal transmission is carried out.

Inputs

```
VAR_INPUT
  bEnable      : BOOL := FALSE;
  sNetId       : T_AmsNetId;
  nPort        : T_AmsPort;
  nIndexGrp    : UDINT;
  nIndexOffs   : UDINT;
  sVarName     : STRING;
  bSignal      : BOOL;
  tCycleTime   : TIME := t#0s;
  bSendNow     : BOOL;
END_VAR
```

Name	Type	Description
bEnable	BOOL	Enable function block.
sNetId	T_AmsNetID	AMS NetID of the ADS device to which the value is to be transmitted.
nPort	T_AmsNetID	AMS port number of the ADS device to which the value is to be transmitted.
nIdxGrp	UDINT	Index group within the ADS device into which the value is to be transmitted.
nIdxOffs	UDINT	Index offset within the ADS device into which the value is to be transmitted.
sVarName	STRING	Symbol name within the ADS device into which the value is to be transmitted.
bSignal	BOOL	Variable whose value is to be transmitted.
tCycleTime	TIME	Cycle time in which the system checks if the input signal has changed.
bSendNow	BOOL	The value is transmitted immediately by a positive edge.

Outputs

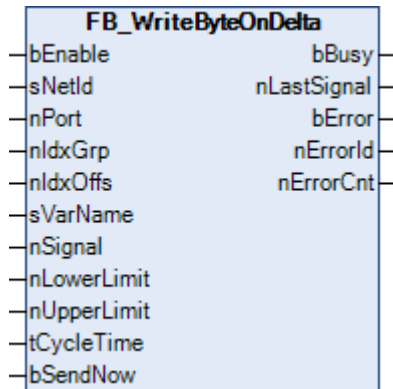
```
VAR_OUTPUT
  bBusy        : BOOL := FALSE;
  bLastSignal   : BOOL;
  bError       : BOOL := FALSE;
  nErrorId     : UDINT := 0;
  nErrorCnt    : UDINT := 0;
END_VAR
```

Name	Type	Description
bBusy	BOOL	The transmission is active.
bLastSignal	BOOL	Most recently transmitted value.
bError	BOOL	An error occurred during the transmission.
nErrorId	UDINT	ADS error number if an error has occurred.
nErrorCnt	UDINT	Number of failed transmission attempts.

Requirements

Development environment	required TC3 PLC library
TwinCAT v3.0.0	Tc2_DataExchange

3.4 FB_WriteByteOnDelta



The function block enables event-driven writing of a variable of type BYTE.

The function block `FB_WriteByteOnDelta` checks cyclically whether the value at the input `nSignal` has changed. The cycle time for checking is determined by the parameter `tCycleTime`. If 0 s is given for `tCycleTime`, the input signal is examined during every PLC cycle. If the comparison determines that the current value is greater by the value `nUpperLimit` or lower by the value `nLowerLimit`, the value of the signal is sent to the specified ADS device. The receiver is addressed by means of the AMS-NetId and the port number (see also ADS Device Identification). The position within the receiver is specified by the index group/index offset or by the symbol name. Usually this is the input image or the flags area.

If the input `bEnable` is set to FALSE, no further signal transmission is carried out.

Inputs

```

VAR_INPUT
    bEnable      : BOOL := FALSE;
    sNetId       : T_AmsNetId;
    nPort        : T_AmsPort;
    nIdxGrp      : UDINT;
    nIdxOffs     : UDINT;
    sVarName     : STRING;
    nSignal      : BYTE;
    nLowerLimit  : BYTE;
    nUpperLimit  : BYTE;
    tCycleTime   : TIME := t#0s;
    bSendNow     : BOOL;
END_VAR

```

Name	Type	Description
bEnable	BOOL	Enable function block.
sNetId	T_AmsNetID	AMS NetID of the ADS device to which the value is to be transmitted.
nPort	T_AmsNetID	AMS port number of the ADS device to which the value is to be transmitted.
nIdxGrp	UDINT	Index group within the ADS device into which the value is to be transmitted.
nIdxOffs	UDINT	Index offset within the ADS device into which the value is to be transmitted.
sVarName	STRING	Symbol name within the ADS device into which the value is to be transmitted.
nSignal	BYTE	Variable whose value is to be transmitted.
nLowerLimit	BYTE	Lower limit value.
nUpperLimit	BYTE	Upper limit value.
tCycleTime	TIME	Cycle time in which the input signal is checked to see whether it has exceeded the limit values.
bSendNow	BOOL	The value is transmitted immediately by a positive edge.

Outputs

```

VAR_OUTPUT
    bBusy          : BOOL := FALSE;
    nLastSignal    : BYTE;
    bError         : BOOL := FALSE;
    nErrorId       : UDINT := 0;
    nErrorCnt      : UDINT := 0;
END_VAR

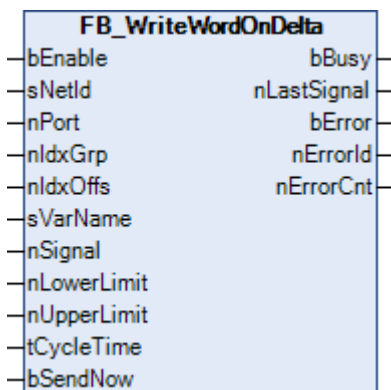
```

Name	Type	Description
bBusy	BOOL	The transmission is active.
nLastSignal	BYTE	Most recently transmitted value.
bError	BOOL	An error occurred during the transmission.
nErrorId	UDINT	ADS error number if an error has occurred.
nErrorCnt	UDINT	Number of failed transmission attempts.

Requirements

Development environment	required TC3 PLC library
TwinCAT v3.0.0	Tc2_DataExchange

3.5 FB_WriteWordOnDelta



The function block enables event-driven writing of a variable of type WORD.

The function block FB_WriteWordOnDelta checks cyclically whether the value at the input *nSignal* has changed. The cycle time for checking is determined by the parameter *tCycleTime*. If 0 s is given for *tCycleTime*, the input signal is examined during every PLC cycle. If the comparison determines that the current value is greater by the value *nUpperLimit* or lower by the value *nLowerLimit*, the value of the signal is sent to the specified ADS device. The receiver is addressed by means of the AMS-NetId and the port number (see also ADS Device Identification). The position within the receiver is specified by the index group/ index offset or by the symbol name. Usually this is the input image or the flags area.

If the input *bEnable* is set to FALSE, no further signal transmission is carried out.

Inputs

```
VAR_INPUT
    bEnable      : BOOL := FALSE;
    sNetId       : T_AmsNetId;
    nPort        : T_AmsPort;
    nIdxGrp      : UDINT;
    nIdxOffs     : UDINT;
    sVarName     : STRING;
    nSignal      : WORD;
    nLowerLimit  : WORD;
    nUpperLimit  : WORD;
    tCycleTime   : TIME := t#0s;
    bSendNow     : BOOL;
END_VAR
```

Name	Type	Description
bEnable	BOOL	Enable function block.
sNetId	T_AmsNetID	AMS NetID of the ADS device to which the value is to be transmitted.
nPort	T_AmsNetID	AMS port number of the ADS device to which the value is to be transmitted.
nIdxGrp	UDINT	Index group within the ADS device into which the value is to be transmitted.
nIdxOffs	UDINT	Index offset within the ADS device into which the value is to be transmitted.
sVarName	STRING	Symbol name within the ADS device into which the value is to be transmitted.
nSignal	WORD	Variable whose value is to be transmitted.
nLowerLimit	WORD	Lower limit value.
nUpperLimit	WORD	Upper limit value.
tCycleTime	TIME	Cycle time in which the input signal is checked to see whether it has exceeded the limit values.
bSendNow	BOOL	The value is transmitted immediately by a positive edge.

Outputs

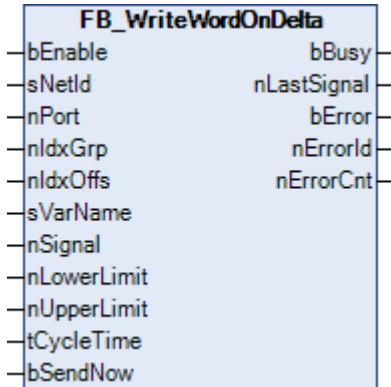
```
VAR_OUTPUT
    bBusy        : BOOL := FALSE;
    nLastSignal  : WORD;
    bError       : BOOL := FALSE;
    nErrorId     : UDINT := 0;
    nErrorCnt    : UDINT := 0;
END_VAR
```

Name	Type	Description
bBusy	BOOL	The transmission is active.
nLastSignal	WORD	Most recently transmitted value.
bError	BOOL	An error occurred during the transmission.
nErrorId	UDINT	ADS error number if an error has occurred.
nErrorCnt	UDINT	Number of failed transmission attempts.

Requirements

Development environment	required TC3 PLC library
TwinCAT v3.0.0	Tc2_DataExchange

3.6 FB_WriteDWordOnDelta



The function block enables event-driven writing of a variable of type DWORD.

The function block FB_WriteDWordOnDelta checks cyclically whether the value at the input *nSignal* has changed. The cycle time for checking is determined by the parameter *tCycleTime*. If 0 s is given for *tCycleTime*, the input signal is examined during every PLC cycle. If the comparison determines that the current value is greater by the value *nUpperLimit* or lower by the value *nLowerLimit*, the value of the signal is sent to the specified ADS device. The receiver is addressed by means of the AMS-NetId and the port number (see also ADS Device Identification). The position within the receiver is specified by the index group/index offset or by the symbol name. Usually this is the input image or the flags area.

If the input *bEnable* is set to FALSE, no further signal transmission is carried out.

Inputs

```
VAR_INPUT
    bEnable      : BOOL := FALSE;
    sNetId       : T_AmsNetId;
    nPort        : T_AmsPort;
    nIdxGrp      : UDINT;
    nIdxOffs     : UDINT;
    sVarName     : STRING;
    nSignal      : BYTE;
    nLowerLimit  : BYTE;
    nUpperLimit  : BYTE;
    tCycleTime   : TIME := t#0s;
    bSendNow     : BOOL;
END_VAR
```


Name	Type	Description
bEnable	BOOL	Enable function block.
sNetId	T_AmsNetID	AMS NetID of the ADS device to which the value is to be transmitted.
nPort	T_AmsNetID	AMS port number of the ADS device to which the value is to be transmitted.
nIdxGrp	UDINT	Index group within the ADS device into which the value is to be transmitted.
nIdxOffs	UDINT	Index offset within the ADS device into which the value is to be transmitted.
sVarName	STRING	Symbol name within the ADS device into which the value is to be transmitted.
nSignal	BYTE	Variable whose value is to be transmitted.
nLowerLimit	BYTE	Lower limit value.
nUpperLimit	BYTE	Upper limit value.
tCycleTime	TIME	Cycle time in which the input signal is checked to see whether it has exceeded the limit values.
bSendNow	BOOL	The value is transmitted immediately by a positive edge.

Outputs

```

VAR_OUTPUT
    bBusy          : BOOL := FALSE;
    nLastSignal    : DWORD;
    bError         : BOOL := FALSE;
    nErrorId       : UDINT := 0;
    nErrorCnt      : UDINT := 0;
END_VAR

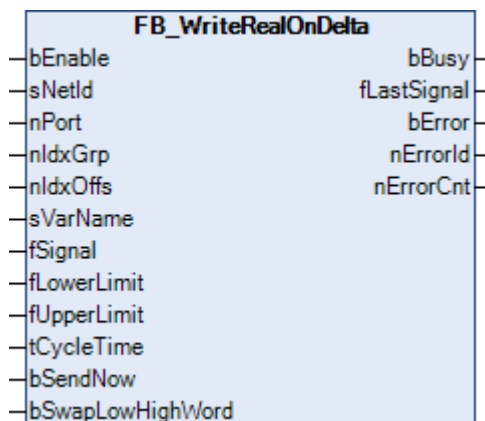
```

Name	Type	Description
bBusy	BOOL	The transmission is active.
nLastSignal	DWORD	Most recently transmitted value.
bError	BOOL	An error occurred during the transmission.
nErrorId	UDINT	ADS error number if an error has occurred.
nErrorCnt	UDINT	Number of failed transmission attempts.

Requirements

Development environment	required TC3 PLC library
TwinCAT v3.0.0	Tc2_DataExchange

3.7 FB_WriteRealOnDelta



The function block enables event-driven writing of a variable of type REAL.

The function block FB_WriteRealOnDelta checks cyclically whether the value at the input *fSignal* has changed. The cycle time for checking is determined by the parameter *tCycleTime*. If 0 s is given for *tCycleTime*, the input signal is examined during every PLC cycle. If the comparison determines that the current value is greater by the value *fUpperLimit* or less by the value *fLowerLimit*, the value of the signal is sent to the specified ADS device. The receiver is addressed by means of the AMS-NetId and the port number (see also ADS Device Identification). The position within the receiver is specified by the index group/index offset or by the symbol name. Usually this is the input image or the flags area.

The internal representation of floating point numbers differs according to the hardware being used. While Intel uses the "little endian" format, Motorola uses the "big endian" format. The input variable *bSwapLowHighWord* can be used to make the necessary adjustment in order to be able to exchange floating point numbers. This is necessary if floating point numbers are to be exchanged between the TwinCAT PLC on a PC and a BC9000, for example.

If the input *bEnable* is set to FALSE, no further signal transmission is carried out.

Inputs

```
VAR_INPUT
    bEnable          : BOOL := FALSE;
    sNetId           : T_AmsNetId;
    nPort            : T_AmsPort;
    nIdxGrp          : UDINT;
    nIdxOffs         : UDINT;
    sVarName         : STRING;
    fSignal          : REAL;
    fLowerLimit      : REAL;
    fUpperLimit      : REAL;
    tCycleTime       : TIME := t#0s;
    bSendNow         : BOOL;
    bSwapLowHighWord : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bEnable	BOOL	Enable function block.
sNetId	T_AmsNetID	AMS NetID of the ADS device to which the value is to be transmitted.
nPort	T_AmsNetID	AMS port number of the ADS device to which the value is to be transmitted.
nIdxGrp	UDINT	Index group within the ADS device into which the value is to be transmitted.
nIdxOffs	UDINT	Index offset within the ADS device into which the value is to be transmitted.
sVarName	STRING	Symbol name within the ADS device into which the value is to be transmitted.
fSignal	REAL	Variable whose value is to be transmitted.
nLowerLimit	REAL	Lower limit value.
nUpperLimit	REAL	Upper limit value.
tCycleTime	TIME	Cycle time in which the input signal is checked to see whether it has exceeded the limit values.
bSendNow	BOOL	The value is transmitted immediately by a positive edge.
bSwapLowHighWord	BOOL	The least significant WORD and the most significant WORD are swapped.

Outputs

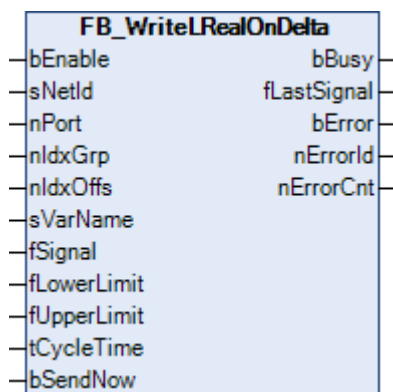
```
VAR_OUTPUT
    bBusy           : BOOL := FALSE;
    fLastSignal     : REAL;
    bError          : BOOL := FALSE;
    nErrorId        : UDINT := 0;
    nErrorCnt       : UDINT := 0;
END_VAR
```

Name	Type	Description
bBusy	BOOL	The transmission is active.
fLastSignal	REAL	Most recently transmitted value.
bError	BOOL	An error occurred during the transmission.
nErrorId	UDINT	ADS error number if an error has occurred.
nErrorCnt	UDINT	Number of failed transmission attempts.

Requirements

Development environment	required TC3 PLC library
TwinCAT v3.0.0	Tc2_DataExchange

3.8 FB_WriteLRealOnDelta



The function block enables event-driven writing of a variable of type LREAL.

The function block FB_WriteLRealOnDelta checks cyclically whether the value at the input *fSignal* has changed. The cycle time for checking is determined by the parameter *tCycleTime*. If 0 s is given for *tCycleTime*, the input signal is examined during every PLC cycle. If the comparison determines that the current value is greater by the value *fUpperLimit* or less by the value *fLowerLimit*, the value of the signal is sent to the specified ADS device. The receiver is addressed by means of the AMS-NetId and the port number (see also ADS Device Identification). The position within the receiver is specified by the index group/ index offset or by the symbol name. Usually this is the input image or the flags area.

If the input *bEnable* is set to FALSE, no further signal transmission is carried out.

Inputs

```

VAR_INPUT
    bEnable       : BOOL := FALSE;
    sNetId        : T_AmsNetId;
    nPort         : T_AmsPort;
    nIdxGrp       : UDINT;
    nIdxOffs      : UDINT;
    sVarName      : STRING;
    fSignal       : LREAL;
    fLowerLimit   : LREAL;
    fUpperLimit   : LREAL;
    tCycleTime    : TIME := t#0s;
    bSendNow      : BOOL;
END_VAR

```

Name	Type	Description
bEnable	BOOL	Enable function block.
sNetId	T_AmsNetID	AMS NetID of the ADS device to which the value is to be transmitted.
nPort	T_AmsNetID	AMS port number of the ADS device to which the value is to be transmitted.
nIdxGrp	UDINT	Index group within the ADS device into which the value is to be transmitted.
nIdxOffs	UDINT	Index offset within the ADS device into which the value is to be transmitted.
sVarName	STRING	Symbol name within the ADS device into which the value is to be transmitted.
fSignal	LREAL	Variable whose value is to be transmitted.
nLowerLimit	LREAL	Lower limit value.
nUpperLimit	LREAL	Upper limit value.
tCycleTime	TIME	Cycle time in which the input signal is checked to see whether it has exceeded the limit values.
bSendNow	BOOL	The value is transmitted immediately by a positive edge.

Outputs

```

VAR_OUTPUT
    bBusy          : BOOL := FALSE;
    fLastSignal    : LREAL;
    bError         : BOOL := FALSE;
    nErrorId       : UDINT := 0;
    nErrorCnt      : UDINT := 0;
END_VAR

```

Name	Type	Description
bBusy	BOOL	The transmission is active.
fLastSignal	LREAL	Most recently transmitted value.
bError	BOOL	An error occurred during the transmission.
nErrorId	UDINT	ADS error number if an error has occurred.
nErrorCnt	UDINT	Number of failed transmission attempts.

Requirements

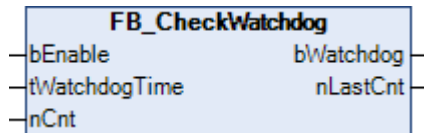
Development environment	required TC3 PLC library
TwinCAT v3.0.0	Tc2_DataExchange

4 Watchdog function blocks

Monitoring Blocks

Name	Description
FB_WriteWatchdog [► 22]	Writes a watchdog signal (an incrementing counter) cyclically
FB_CheckWatchdog [► 21]	Monitors the received watchdog signal

4.1 FB_CheckWatchdog



Monitoring of a watchdog signal, which is transmitted with the function block [FB_WriteWatchdog \[► 22\]](#).

The device to be monitored regularly sends a changing counter value to the device that is to monitor the transmission. The function block `FB_CheckWatchdog` is used there to monitor the counter value. If this does not change within a specific period, the output `bWatchdog` is set to TRUE. If a value of 0 s is specified for `tWatchdogTime`, the `bWatchdog` signal is set to FALSE. The period specified by `tWatchdogTime` should be a multiple (5-10 times) of the time in which the monitoring signal is transmitted.

Inputs

```

VAR_INPUT
    bEnable      : BOOL := FALSE;
    tWatchdogTime : TIME := t#0s;
    nCnt         : UDINT;
END_VAR

```

Name	Type	Description
bEnable	BOOL	Enable function block.
tWatchdogTime	TIME	Duration during which nCnt has to change.
nCnt	UDINT	Current counter value of the watchdog signal.

Outputs

```

VAR_OUTPUT
    bWatchdog : BOOL := FALSE;
    nLastCnt  : UDINT;
END_VAR

```

Name	Type	Description
bWatchdog	BOOL	FALSE indicates a valid monitoring signal. The output will become TRUE if no change is detected in <code>nCnt</code> during the period of time specified by <code>tWatchdogTime</code> .
nLastCnt	UDINT	Most recent successfully transmitted counter value of the monitoring signal.

Requirements

Development environment	required TC3 PLC library
TwinCAT v3.0.0	Tc2_DataExchange

4.2 FB_WriteWatchdog



Writing of a watchdog signal to another ADS device (TwinCAT PLC, Bus Terminal Controller, ...).

The FB_WriteWatchdog function block cyclically writes the contents of a 32-bit counter into another ADS device. The counter is incremented every time the transmission is successful. The FB_CheckWatchdog function block can be used at the receiver to evaluate this signal. The receiver is addressed by means of the AMS-NetId and the port number (see also ADS Device Identification). The position within the receiver is specified by the index group/index offset or by the symbol name. Usually this is the input image or the flags area.

The period for *tWatchdogTime* should not be shorter than 1 second, to avoid transmitting the counter state too frequently. If 0 s is given for *tWatchdogTime*, the signal is not transmitted. Please also note the description of the function block [FB_CheckWatchdog \(\)](#) [► 21].

If the input *bEnable* is set to FALSE, no further transfer of the watchdog signal takes place.

Inputs

```
VAR_INPUT
  bEnable      : BOOL := FALSE;
  sNetId       : T_AmsNetId;
  nPort        : T_AmsPort;
  nIdxGrp      : UDINT;
  nIdxOffs     : UDINT;
  sVarName     : STRING;
  tWatchdogTime : TIME := t#0s;
  bSendNow     : BOOL;
END_VAR
```

Name	Type	Description
bEnable	BOOL	Enable function block.
sNetId	T_AmsNetID	AMS NetID of the ADS device to which the watchdog signal is to be transmitted.
nPort	T_AmsNetID	AMS port number of the ADS device to which the watchdog signal is to be transmitted.
nIdxGrp	UDINT	Index group within the ADS device to which the watchdog signal is to be transmitted.
nIdxOffs	UDINT	Index offset within the ADS device to which the watchdog signal is to be transmitted.
sVarName	STRING	Symbol name within the ADS device to which the watchdog signal is to be transmitted.
tWatchdogTime	TIME	Cycle time in which the watchdog signal is transmitted.
bSendNow	BOOL	The watchdog signal is transmitted immediately by a positive edge.

Outputs

```
VAR_OUTPUT
  bBusy       : BOOL := FALSE;
  nLastCnt    : UDINT := 0;
  bError      : BOOL := FALSE;
  nErrorId    : UDINT := 0;
END_VAR
```

Name	Type	Description
bBusy	BOOL	The transmission is active.
nLastCnt	BOOL	Most recently transmitted counter value.
bError	BOOL	An error occurred during the transmission.
nErrorId	UDINT	ADS error number if an error has occurred.

Requirements

Development environment	required TC3 PLC library
TwinCAT v3.0.0	Tc2_DataExchange

5 Data types

Name	Description
E_ADSComMode [► 24]	Setting of the communication type: safe or fast.

5.1 E_AdsComMode

```
TYPE E_AdsComMode :  
(  
  eAdsComModeSecureCom := 0,  
  eAdsComModeFastCom   := 1  
);  
END_TYPE
```

Requirements

Development environment	required TC3 PLC library
TwinCAT v3.1.0	Tc2_DataExchange

6 Global constants

6.1 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_DataExchange : ST_LibVersion;
END_VAR
```

Name	Type	Description
stLibVersion_Tc2_DataExchange	ST_LibVersion	Version information of Tc2_DataExchange

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.

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

