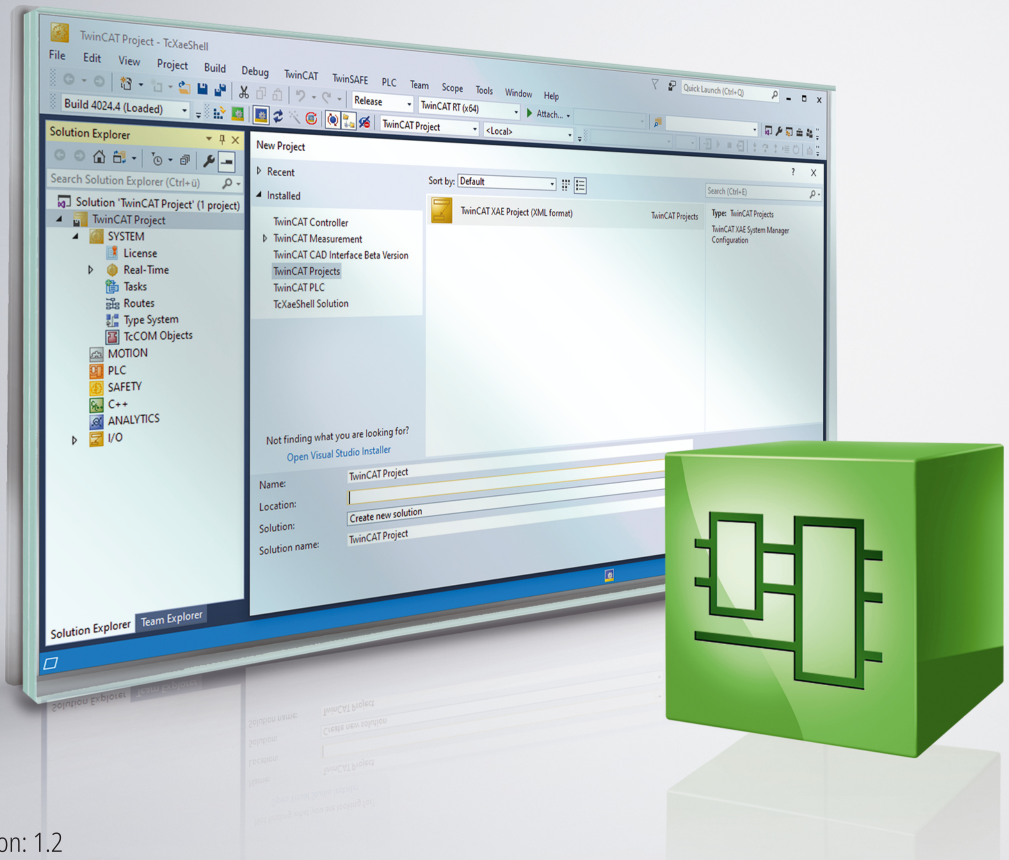
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | PLC Library: Tc2_DataExchange



2020-11-30 | Version: 1.2

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the 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 prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, 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.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

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

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

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations 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 engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

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 [▸ 8]	Reads a variable of any desired data type by variable name
FB_WriteAdsSymByName [▸ 9]	Writes a variable of any desired data type by variable name
FB_WriteBoolOnDelta [▸ 10]	Writes a variable of type BOOLEAN in response to an event.
FB_WriteByteOnDelta [▸ 12]	Writes a variable of type BYTE in response to an event.
FB_WriteWordOnDelta [▸ 13]	Writes a variable of type WORD in response to an event.
FB_WriteDWordOnDelta [▸ 14]	Writes a variable of type DWORD in response to an event.
FB_WriteRealOnDelta [▸ 16]	Writes a variable of type REAL in response to an event.
FB_WriteLRealOnDelta [▸ 17]	Writes a variable of type LREAL in response to an event.

Monitoring Blocks

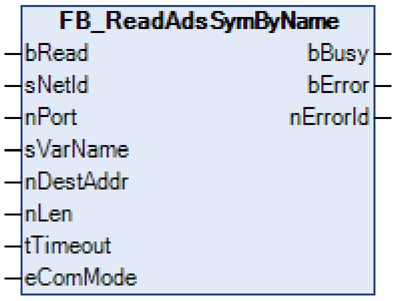
Name	Description
FB_WriteWatchdog [▸ 20]	Writes a watchdog signal (an incrementing counter) cyclically
FB_CheckWatchdog [▸ 19]	Monitors the received watchdog signal

3 Event driven data exchange

Write/Read Blocks

Name	Description
FB_ReadAdsSymByName [▶ 8]	Reads a variable of any desired data type by variable name
FB_WriteAdsSymByName [▶ 9]	Writes a variable of any desired data type by variable name
FB_WriteBoolOnDelta [▶ 10]	Writes a variable of type BOOLEAN in response to an event.
FB_WriteByteOnDelta [▶ 12]	Writes a variable of type BYTE in response to an event.
FB_WriteWordOnDelta [▶ 13]	Writes a variable of type WORD in response to an event.
FB_WriteDWordOnDelta [▶ 14]	Writes a variable of type DWORD in response to an event.
FB_WriteRealOnDelta [▶ 16]	Writes a variable of type REAL in response to an event.
FB_WriteLRealOnDelta [▶ 17]	Writes a variable of type LREAL in response to an event.

3.1 FB_ReadAdsSymByName



The function block 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:* Following each read procedure 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 *sVarName*, *sNetID* and *nPort* inputs do not change, the handle of the PLC variable will not be released after each read procedure. This mode should be used when values are exchanged very frequently.

VAR_INPUT

```
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
```

bRead: The function block reads the content of the variables *sVarName* of the selected ADS devices and writes it to the variable to which the pointer *nDestAddr* points.

sNetId: AMS-NetId of the ADS device from which the value is to be read. (Type T_AmsNetId)

nPort: AMS Port number of the ADS device from which the value is to be read. (Type T_AmsPort)

sVarName: Symbol name of the variable to be read on the selected ADS device (max. 255 characters).

nDestAddr: Address of the variable into which the read value is written.

nLen: Length of the variable to be read in bytes.

tTimeout: Time until processing is aborted.

eComMode: Enum used to specify whether the handle of the PLC variable is released again after each read procedure.

VAR_OUTPUT

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

bBusy: The transmission is active.

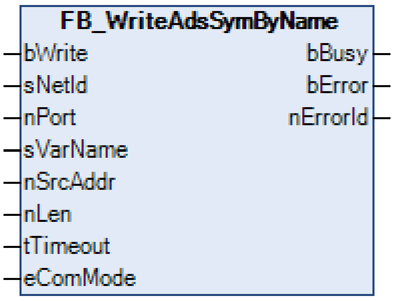
bError: An error occurred during the transmission.

nErrorId: 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 enables writing of any value to another controller using the symbol name.

On a positive edge at the *bWrite* input 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:* Following each write procedure 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 *sVarName*, *sNetID* and *nPort* inputs do not change, the handle of the PLC variable will not be released after each write procedure. This mode should be used when values are exchanged very frequently.

VAR_INPUT

```
VAR_INPUT
    bWrite           :  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
```

bWrite: This function block writes the contents of the variable to which pointer *nSrcAddr* points into the variable *sVarName* of the selected ADS device.

sNetId: AMS NetID of the ADS device to which the value is to be transmitted. (Type T_AmsNetId)

nPort: AMS port number of the ADS device to which the value is to be transmitted. (Type T_AmsPort)

sVarName: Symbol name of the variable to be written on the selected ADS device (max 255 characters).

nSrcAddr: Address of the variable in which the value to written is located.

nLen: Length in bytes of the variable to be written.

tTimeout: Time until processing is aborted.

eComMode: Enum used to specify whether the handle of the PLC variable is released again after each write procedure.

VAR_OUTPUT

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

bBusy: The transmission is active.

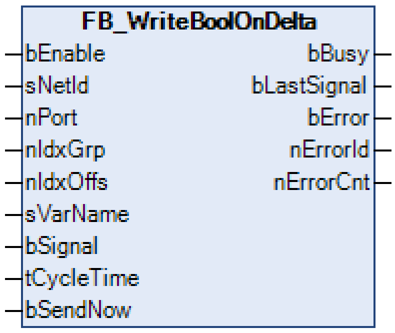
bError: An error occurred during the transmission.

nErrorId: 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 FB_WriteBoolOnDelta function block checks cyclically whether the value at the *bSignal* input 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 *bEnable* input is set to FALSE, no further signal transmission is carried out.

VAR_INPUT

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

bEnable: Enable function block.

sNetId: AMS NetID of the ADS device to which the value is to be transmitted. (Type T_AmsNetId)

nPort: AMS port number of the ADS device to which the value is to be transmitted. (Type T_AmsPort)

nIdxGrp: Index group within the ADS device into which the value is to be transmitted.

nIdxOffs: Index offset within the ADS device into which the value is to be transmitted.

sVarName: Symbol name within the ADS device into which the value is to be transmitted.

bSignal: Variable whose value is to be transferred.

tCycleTime: Cycle time in which the input signal is examined to see that it has changed.

bSendNow: The value is transmitted immediately in response to a positive edge.

VAR_OUTPUT

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

bBusy: The transmission is active.

bLastSignal: Most recently transmitted value.

bError: An error occurred during the transmission.

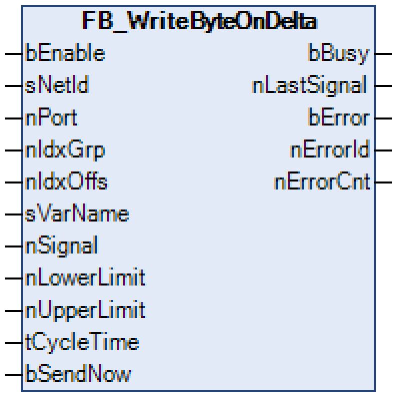
nErrorId: Error number if an error has occurred.

nErrorCnt: Number of transmission attempts that have returned faults.

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 FB_WriteByteOnDelta function block checks cyclically whether the value at the *nSignal* input 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 establishes that the current value is greater than the value *nUpperLimit* or lower than the value *nLowerLimit*, then the value of the signal is sent to the ADS device that is to be specified. 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 *bEnable* input is set to FALSE, no further signal transmission is carried out.

VAR_INPUT

```
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
```

bEnable: Enable function block.

sNetId: AMS NetID of the ADS device to which the value is to be transmitted. (Type T_AmsNetId)

nPort: AMS port number of the ADS device to which the value is to be transmitted. (Type T_AmsPort)

nIdxGrp: Index group within the ADS device into which the value is to be transmitted.

nIdxOffs: Index offset within the ADS device into which the value is to be transmitted.

sVarName: Symbol name within the ADS device into which the value is to be transmitted.

nSignal: Variable whose value is to be transmitted.

nLowerLimit: Lower limit value.

nUpperLimit: Upper limit value.

tCycleTime: Cycle time in which the input signal is examined to see whether it has exceeded the limit values.

bSendNow: The value is transmitted immediately in response to a positive edge.

VAR_OUTPUT

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

bBusy: The transmission is active.

nLastSignal: Most recently transmitted value.

bError: An error occurred during the transmission.

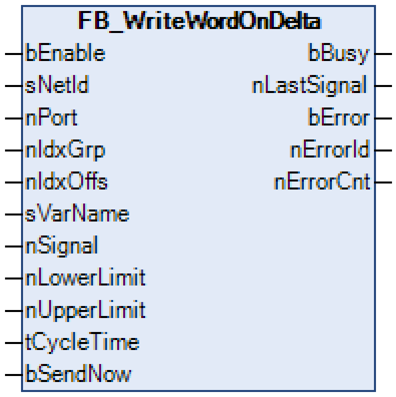
nErrorId: Error number if an error has occurred.

nErrorCnt: Number of transmission attempts that have returned faults.

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 FB_WriteWordOnDelta function block checks cyclically whether the value at the *nSignal* input 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 establishes that the current value is greater than the value *nUpperLimit* or lower than the value *nLowerLimit*, then the value of the signal is sent to the ADS device that is to be specified. 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 *bEnable* input is set to FALSE, no further signal transmission is carried out.

VAR_INPUT

```
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
```

bEnable: Enable function block.

sNetId: AMS NetID of the ADS device to which the value is to be transmitted. (Type T_AmsNetId)

nPort: AMS port number of the ADS device to which the value is to be transmitted. (Type T_AmsPort)

nIdxGrp: Index group within the ADS device into which the value is to be transmitted.

nIdxOffs: Index offset within the ADS device into which the value is to be transmitted.

sVarName: Symbol name within the ADS device into which the value is to be transmitted.

nSignal: Variable whose value is to be transmitted.

nLowerLimit: Lower limit value.

nUpperLimit: Upper limit value.

tCycleTime: Cycle time in which the input signal is examined to see whether it has exceeded the limit values.

bSendNow: The value is transmitted immediately in response to a positive edge.

VAR_OUTPUT

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

bBusy: The transmission is active.

nLastSignal: Most recently transmitted value.

bError: An error occurred during the transmission.

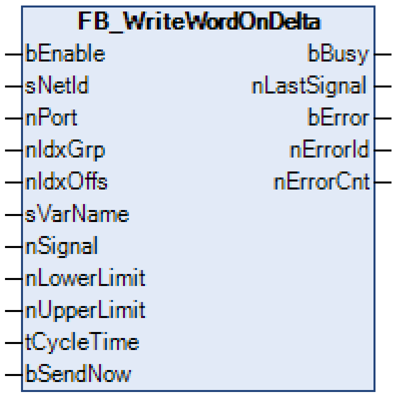
nErrorId: Error number if an error has occurred.

nErrorCnt: Number of transmission attempts that have returned faults.

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 FB_WriteDWordOnDelta function block checks cyclically whether the value at the *nSignal* input 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 establishes that the current value is greater than the value *nUpperLimit* or lower than the value *nLowerLimit*, then the value of the signal is sent to the ADS device that is to be specified. 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 *bEnable* input is set to FALSE, no further signal transmission is carried out.

VAR_INPUT

```
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
```

bEnable: Enable function block.

sNetId: AMS NetID of the ADS device to which the value is to be transmitted. (Type T_AmsNetId)

nPort: AMS port number of the ADS device to which the value is to be transmitted. (Type T_AmsPort)

nIdxGrp: Index group within the ADS device into which the value is to be transmitted.

nIdxOffs: Index offset within the ADS device into which the value is to be transmitted.

sVarName: Symbol name within the ADS device into which the value is to be transmitted.

nSignal: Variable whose value is to be transmitted.

nLowerLimit: Lower limit value.

nUpperLimit: Upper limit value.

tCycleTime: Cycle time in which the input signal is examined to see whether it has exceeded the limit values.

bSendNow: The value is transmitted immediately in response to a positive edge.

VAR_OUTPUT

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

bBusy: The transmission is active.

nLastSignal: Most recently transmitted value.

bError: An error occurred during the transmission.

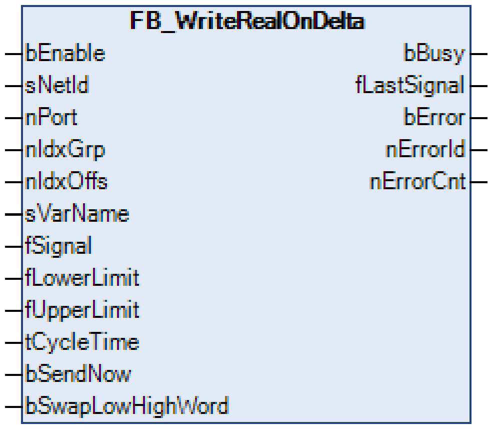
nErrorId: Error number if an error has occurred.

nErrorCnt: Number of transmission attempts that have returned faults.

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 FB_WriteRealOnDelta function block checks cyclically whether the value at the *fSignal* input 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 establishes that the current value is greater than the value *fUpperLimit* or lower than the value *fLowerLimit*, then the value of the signal is sent to the ADS device that is to be specified. 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 *bEnable* input is set to FALSE, no further signal transmission is carried out.

VAR_INPUT

```
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
```

bEnable: Enable function block.

sNetId: AMS NetID of the ADS device to which the value is to be transmitted. (Type T_AmsNetId)

nPort: AMS port number of the ADS device to which the value is to be transmitted. (T_AmsPort)

nIdxGrp: Index group within the ADS device into which the value is to be transmitted.

nIdxOffs: Index offset within the ADS device into which the value is to be transmitted.

sVarName: Symbol name within the ADS device into which the value is to be transmitted.

fSignal: Variable whose value is to be transmitted.

fLowerLimit: Lower limit value.

fUpperLimit: Upper limit value.

tCycleTime: Cycle time in which the input signal is examined to see whether it has exceeded the limit values.

bSendNow: The value is transmitted immediately in response to a positive edge.

bSwapLowHighWord: The least significant WORD and the most significant WORD are swapped.

VAR_OUTPUT

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

bBusy: The transmission is active.

fLastSignal: Most recently transmitted value.

bError: An error occurred during the transmission.

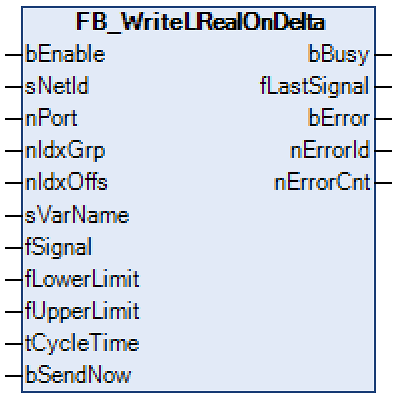
nErrorId: Error number if an error has occurred.

nErrorCnt: Number of transmission attempts that have returned faults.

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 FB_WriteLRealOnDelta function block checks cyclically whether the value at the *fSignal* input 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 establishes that the current value is greater than the value *fUpperLimit* or lower than the value *fLowerLimit*, then the value of the signal is sent to the ADS device that is to be specified. 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 *bEnable* input is set to FALSE, no further signal transmission is carried out.

VAR_INPUT

```
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
```

bEnable: Enable function block.

sNetId: AMS NetID of the ADS device to which the value is to be transmitted. (Type T_AmsNetId)

nPort: AMS port number of the ADS device to which the value is to be transmitted. (T_AmsPort)

nIdxGrp: Index group within the ADS device into which the value is to be transmitted.

nIdxOffs: Index offset within the ADS device into which the value is to be transmitted.

sVarName: Symbol name within the ADS device into which the value is to be transmitted.

fSignal: Variable whose value is to be transmitted.

fLowerLimit: Lower limit value.

fUpperLimit: Upper limit value.

tCycleTime: Cycle time in which the input signal is examined to see whether it has exceeded the limit values.

bSendNow: The value is transmitted immediately in response to a positive edge.

VAR_OUTPUT

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

bBusy: The transmission is active.

fLastSignal: Most recently transmitted value.

bError: An error occurred during the transmission.

nErrorId: Error number if an error has occurred.

nErrorCnt: Number of transmission attempts that have returned faults.

Requirements

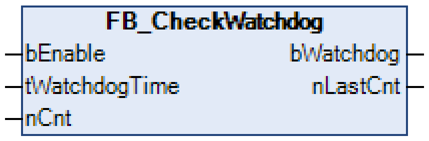
Development environment	required TC3 PLC library
TwinCAT v3.0.0	Tc2_DataExchange

4 Watchdog function blocks

Monitoring Blocks

Name	Description
FB_WriteWatchdog [▶ 20]	Writes a watchdog signal (an incrementing counter) cyclically
FB_CheckWatchdog [▶ 19]	Monitors the received watchdog signal

4.1 FB_CheckWatchdog



Monitoring of a watchdog signal, which is transferred with the function block FB_WriteWatchdog [▶ 20].

The device to be monitored regularly sends a variable counter value to the device that is to monitor the transmission. The function block FB_CheckWatchdog is used there to monitor the state of the counter. If this does not change within a specific period, the *bWatchdog* output 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 *tWachtdogTime* should be a multiple (5-10 times) of the time in which the monitoring signal is transmitted.

VAR_INPUT

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

bEnable: Enable function block.

tWatchdogTime: Duration during which *nCnt* has to change.

nCnt: Current counter value of the watchdog signal.

VAR_OUTPUT

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

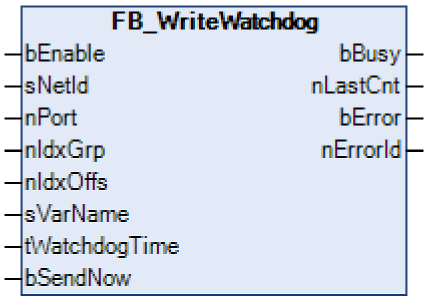
bWatchdog: FALSE indicates a valid monitoring signal. The output will become TRUE if no change is detected in *nCnt* during the period of time specified by *tWatchdogTime*.

nLastCnt: Most recent successfully transmitted counter state 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 *tWachtdogTime* 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 () [▶ 19].

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

VAR_INPUT

```
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
```

bEnable: Enable function block.

sNetId: AMS NetID of the ADS device to which the watchdog signal is to be transferred. (Type T_AmsNetId)

nPort: AMS port number of the ADS device to which the watchdog signal is to be transferred. (Type T_AmsPort)

nIdxGrp: Index group within the ADS device to which the watchdog signal is to be transferred.

nIdxOffs: Index offset within the ADS device to which the watchdog signal is to be transferred.

sVarName: Symbol name within the ADS device to which the watchdog signal is to be transferred.

tWatchdogTime: Cycle time in which the watchdog signal is transferred.

bSendNow: A positive edge triggers immediate transfer of the watchdog signal.

VAR_OUTPUT

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

bBusy: Transmission is active.

nLastCnt: Most recently transmitted counter state.

bError: An error occurred during the transmission.

nErrorId: 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 [▸ 22]	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
```

stLibVersion_Tc2_DataExchange: Version information of Tc2_DataExchange (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!

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