

Manual | EN

TS8036

TwinCAT 2 | Crestron Server

Supplement |
Building Automation



Table of contents

1 Foreword	5
1.1 Notes on the documentation	5
1.2 Safety instructions	6
1.3 Notes on information security.....	7
2 Introduction	8
3 Hardware requirements	9
4 User requirements	10
5 Functioning	11
6 Performance Test	13
7 Installation	15
7.1 Installation under Windows CE (WEC)	15
7.2 Installation under Windows Embedded Standard (WES).....	15
7.3 Installation Crestron User Modules	15
8 Beckhoff PLC Function Blocks	17
8.1 FB_CrestronReadByte	17
8.2 FB_CrestronWriteByte	18
8.3 FB_CrestronWriteByteArray.....	19
8.4 FB_CrestronReadInt	19
8.5 FB_CrestronWriteInt	20
8.6 FB_CrestronWriteIntArray.....	21
8.7 FB_CrestronReadReal	22
8.8 FB_CrestronWriteReal	23
8.9 FB_CrestronGetClientConnectionState	24
9 Crestron SIMPL Windows User Modules	25
9.1 Beckhoff Communication	25
9.2 Beckhoff Communication 400	26
9.3 Beckhoff Read Bit	27
9.4 Beckhoff Read Bit By Name.....	28
9.5 Beckhoff Write Bit.....	29
9.6 Beckhoff Write Bit By Name	30
9.7 Beckhoff Read Byte	30
9.8 Beckhoff Read Byte By Name.....	31
9.9 Beckhoff Write Byte.....	32
9.10 Beckhoff Write Byte By Name.....	33
9.11 Beckhoff Read Int.....	34
9.12 Beckhoff Read Int By Name	35
9.13 Beckhoff Write Int.....	36
9.14 Beckhoff Write Int By Name	36
9.15 Beckhoff Switch.....	37
9.16 Beckhoff Switch By Name	38
10 Error Codes	39

11 Samples	40
11.1 Beckhoff PLC programming examples.....	40
11.2 Crestron programming examples	40

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®, 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.

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

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

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.

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

Requirements

Crestron Electronics with head office in Rockleigh (www.crestron.com), New Jersey (USA) is one of the market leaders for AV control systems. In addition to AV control units, building management components can also be integrated. The following article describes data exchange between Crestron controllers and Beckhoff hardware via Ethernet.



For the Crestron control centres, blocks (user modules) are available for data exchange via TCP/IP. The TwinCAT Crestron Server must be installed on the Beckhoff controller. A Crestron controller can call write and read commands and a Beckhoff controller can write data to the Crestron controller or read data from it.

3 Hardware requirements

Beckhoff hardware requirements:

The TwinCAT Crestron Server can be used on any PC controller that is compatible to TwinCAT (except for the CX9000).

The following are required / necessary for the installation of the TwinCAT Crestron Server on a CX x86 or CX ARM processor-based hardware platform:

- TwinCAT PLC Runtime
- Microsoft .NET Framework / Compact .NET Framework.
- Ethernet interface

Please refer to the Beckhoff Information System as to which hardware with which image fulfils these requirements.

Crestron hardware requirements:

- Crestron Series-2 controller with network interface (e.g. CP2E, MP2E etc.)
- Firmware version 4.001.10.12 (Feb 17 2009) or higher.

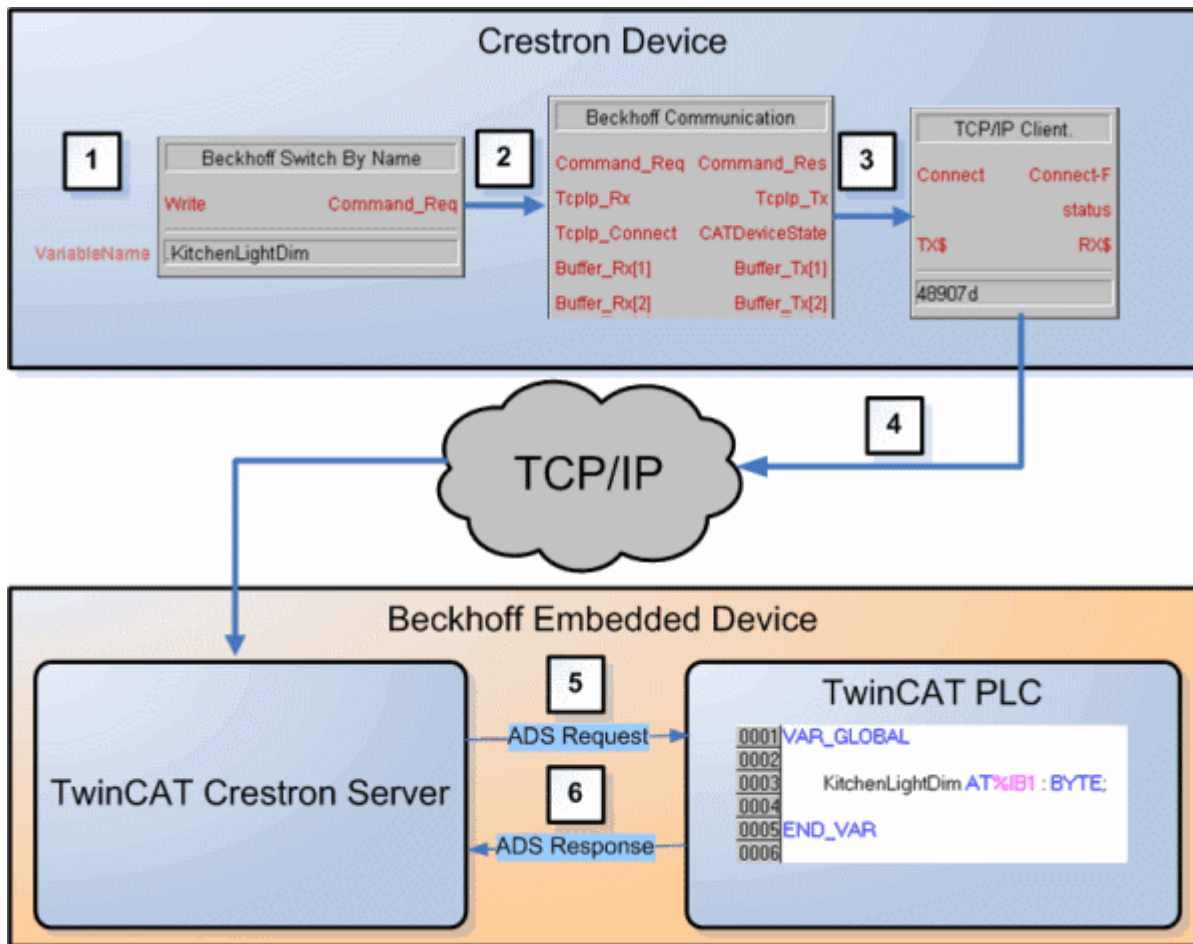
4 User requirements

The user of this ADS server / this library requires basic knowledge of the following:

- Handling the TwinCAT PLC Control
- Handling the TwinCAT System Manager
- PC and network knowledge
- Relevant safety regulations for building technical equipment

5 Functioning

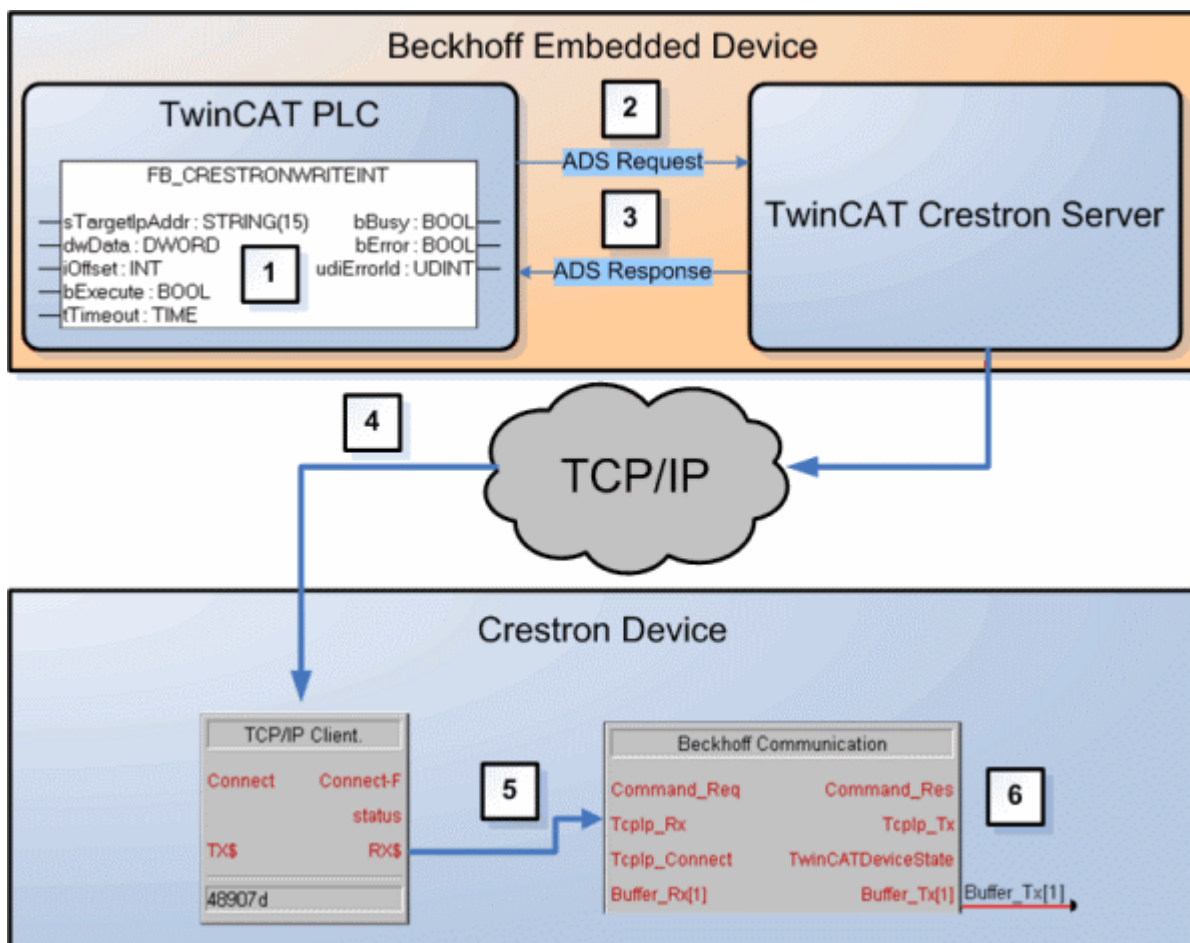
The flow diagram describes how the individual work steps are executed in a write command from the Crestron controller to the TwinCAT PLC.



Work steps:

1. The write command is initiated by a positive edge on *Beckhoff Switch By Name*.
2. It is forwarded to the *Beckhoff Communication* module for further supplements to the command parameters.
3. The expanded write command is passed to the TCP/IP client for sending.
4. The data is sent via TCP/IP.
5. The *TwinCAT Crestron Server* receives the write command, converts it to an ADS request telegram and sends the data to the TwinCAT PLC.
6. The PLC replies with an ADS response, which is received by the *TwinCAT Crestron Server* in order to log any errors that may have occurred.

The flow diagram describes how the individual work steps are executed in a write command from the TwinCAT PLC to the Crestron controller.



Work steps:

1. The write command is initiated by a positive edge on *FB_CrestronWriteInt*.
2. It is sent to the *TwinCAT Crestron Server* as an *ADS request*. This converts the data received into a TCP/IP telegram that is understandable to Crestron and sends it to the Crestron controller via TCP/IP.
3. An *ADS Response*, which carries information on the progress of the transmission, is generated in reply to the *ADS Request*. This ensures an appropriate reaction to any errors that have occurred.
4. The data is sent via TCP/IP.
5. The Crestron *TCP/IP Client* receives the data and forwards it to the *Beckhoff Communication* module for processing.
6. The *Beckhoff Communication* module reacts to the data from the TCP/IP client and filters the information in order to then write the addressed signal with the transferred value.

6 Performance Test

The results of the test should give an indication of the relationship of the PLC cycle time to the employed hardware. This may be helpful when selecting the necessary hardware.



The test took place under optimum conditions in a test environment specially set up for the purpose. Therefore, the results obtained cannot be carried over to other environments without restriction.

Structure

A read command was transmitted on a 16-bit value for one minute with PLC cycle times of **50**, **10** and **1** millisecond. The Beckhoff PLC was thereby the reading device. The number of successfully transmitted read commands was recorded by a counter. This produced the following results:

Results

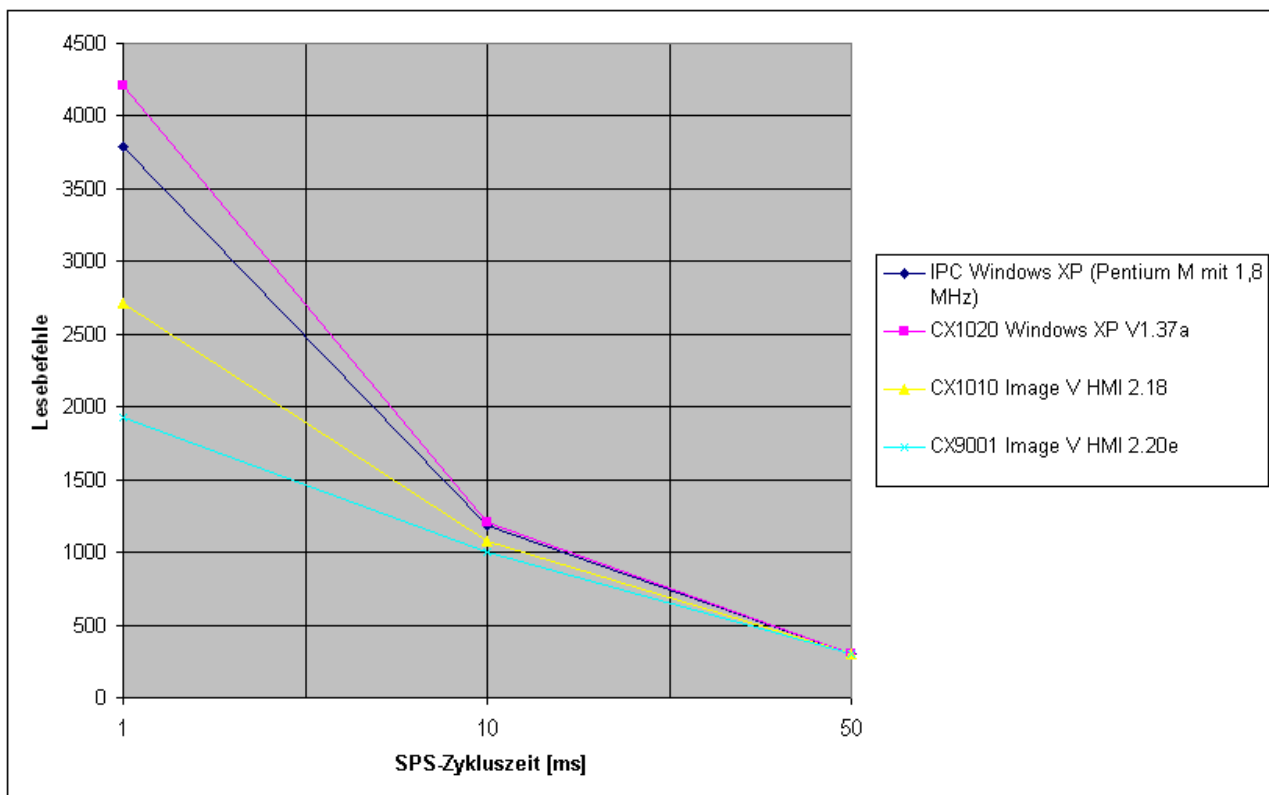


Fig. 1:

Device description	PLC cycle time			Value description
	50ms	10ms	1ms	
IPC Windows XP (Pentium M at 1.8 MHz) MP2E Firmware V4.001.1012 (Feb 17 2009)	0	0	0	Error
	299	1180	3785	Read commands
	299	1180	3785	Successful read commands
CX1020 Windows XP V1.37a MP2E Firmware V4.001.1012 (Feb 17 2009)	0	0	0	Error
	298	1208	4208	Read commands
	298	1208	4208	Successful read commands

CX1010 Image V HMI 2.18 MP2E Firmware V4.001.1012 (Feb 17 2009)	0	0	0	Error
	298	1078	2709	Read commands
	298	1078	2709	Successful read commands
CX9001 Image V HMI 2.20e MP2E Firmware V4.001.1012 (Feb 17 2009)	0	0	0	Error
	299	997	1928	Read commands
	299	997	1928	Successful read commands


The test shows that reducing the PLC cycle time significantly increases the number of read commands. The higher the PLC cycle time is, the smaller the influence of the hardware capability on the data transfer rate.

Select the hardware such that the PLC cycle time can be set as low as possible. The performance can also be improved using an additional, faster background task in which communication with the Crestron controller is executed.



The TwinCAT System Manager offers aids for optimizing the PLC cycle time. Further information on this can be found in the documentation for the TwinCAT System Manager.

Both the PLC and the SIMPL Windows program are available to you as downloads.

Description	Download
Beckhoff PLC for cyclic reading of a value / SIMPL Windows program for providing a value.	https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364625547/.zip 

7 Installation

7.1 Installation under Windows CE (WEC)

Transfer of the installation to the CE device

If the TwinCAT Crestron Server was successfully installed, two subfolders will be found in the folder "**C:\TwinCAT\CE\Crestron Server**".

- **\ARMV4I**
 - This folder contains the CE installation:
TcCrestronServerCe.ARMV4I.CAB for ARMV4 CPUs (e.g. CX900x)
- **\I586**
 - This folder contains the CE installation:
TcCrestronServerCe.I586.CAB for X86 CPUs (e.g. CX10xx)

Transfer the file "**TcCrestronServerCe.ARMV4I.CAB**" or "**TcCrestronServerCe.I586.CAB**" to the Windows CE device. This can take place via

- a shared folder (public folder)
- FTP
- ActiveSync
- Compact Flash (CF) adaptors

Installation on the Windows CE device

The installation can be executed with a normal double click on the CAB file. Confirm the preselected (default) directory "**hard disk\System\TC-Crestron-Server**" with "OK".

Following the installation, the CAB file will be deleted automatically. Restart the device after the installation has finished.

7.2 Installation under Windows Embedded Standard (WES)

Starting the installation

Start the installation and follow the instructions in the dialogs.

After the installation

After the installation has finished, you will find two new directories in your TwinCAT folder: "**..\TwinCAT\CE\Crestron Server**" and "**..\TwinCAT\Crestron Server**".

The installation files (.CAB) for the different Windows CE platforms ARM and X86 can be found in the directory "**..\TwinCAT\CE\Crestron Server**".

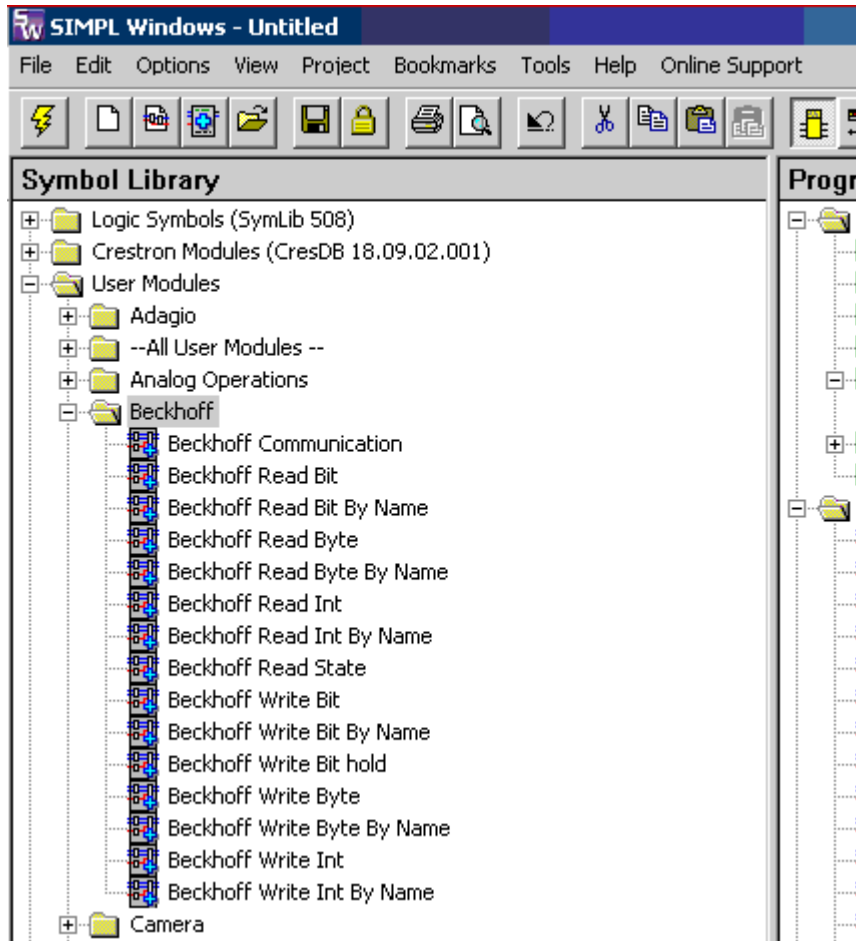
7.3 Installation Crestron User Modules

The installation consists of several user modules. These are in turn subdivided into SIMPL+ modules and UserMacros. The SIMPL+ modules have the extension USH or USP. The UserMacros have the extension UMC.

After installation of the TwinCAT Crestron Server, the user modules are located in the following folder: **..\TwinCAT\Crestron Server\CrestronModule\BeckhoffCommunication.zip**. Unzip the file and copy it to the following directory:

Files	Directory
*.umc	<Crestron-Program-Folder>\Simpl\Usrmacro
*.ush und *.usp	<Crestron-Program-Folder>\Simpl\Usrplus

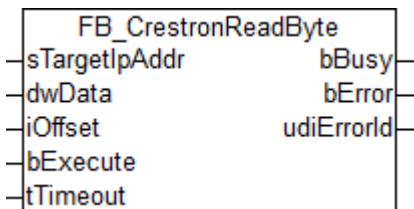
The individual user modules are listed in the SIMPL window under *Symbol Library -> User Modules -> Beckhoff*.



8 Beckhoff PLC Function Blocks

Name	Function
FB_CrestronWriteByte [▶ 18]	Write a value of type byte (8 Bit).
FB_CrestronWriteByteArray [▶ 19]	Write several values (array) of type byte (8 bit).
FB_CrestronReadByte [▶ 17]	Read a value of type byte (8 Bit).
FB_CrestronWriteInt [▶ 20]	Write a value of type integer (16 Bit).
FB_CrestronWriteIntArray [▶ 21]	Write several values (array) of type integer (16 bit).
FB_CrestronReadInt [▶ 19]	Read a value of type integer (16 Bit).
FB_CrestronWriteReal [▶ 23]	Write a value of type real (32 Bit).
FB_CrestronReadReal [▶ 22]	Read a value of type real (32 Bit).
FB_CrestronGetClientConnectionState [▶ 24]	Read the Client connection state

8.1 FB_CrestronReadByte



This function block enables a byte value (8 bit) to be read from the Crestron controller.

VAR_INPUT

```
sTargetIpAddr : STRING(15);
dwData       : DWORD;
iOffset      : INT;
bExecute     : BOOL;
tTimeout     : TIME := T#5s;
```

sTargetIpAddr: IP address of the Crestron controller.

dwData: Address pointer to a PLC variable of data type BYTE (8 bit), in which the read value is stored.

iOffset: Indicates the offset in the buffer *Buffer_RX* of the Crestron controller from which the data are to be read.

bExecute: A positive edge starts the block.

tTimeout: Specifies the interval for awaiting a response.

VAR_OUTPUT

```
bBusy       : BOOL;
bError      : BOOL;
udiErrorId  : UDINT;
```

bBusy: During the read process *bBusy* is set to True and remains True until the reading process is complete. During this time no further read commands can be activated via this block.

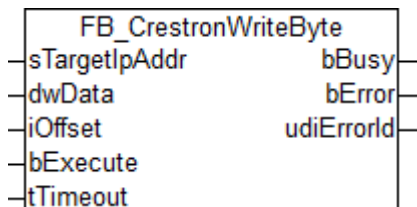
bError: A positive flank indicates that an error has occurred.

udiErrorId: Contains the [error code](#) [[▶ 39](#)] that may occur during block execution. If no error has occurred, the value is 0.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64	PC/CX	TcCrestron library from V1.0.0

8.2 FB_CrestronWriteByte



This function block enables a byte value (8 bit) to be written to the Crestron controller.

VAR_INPUT

```
sTargetIpAddr : STRING(15);
dwData       : DWORD;
iOffset      : INT;
bExecute     : BOOL;
tTimeout     : TIME := T#5s;
```

sTargetIpAddr: IP address of the Crestron controller.

dwData: Address pointer to a PLC variable of data type INT (16 bit), in which the value to be written is stored.

iOffset: Indicates the offset in the buffer *Buffer_TX* of the Crestron controller from which the data are to be written.

bExecute: A positive edge starts the block

tTimeout: Specifies the interval for awaiting a response.

VAR_OUTPUT

```
bBusy       : BOOL;
bError      : BOOL;
udiErrorId  : UDINT;
```

bBusy: During the write process *bBusy* is set to True and remains True until the writing process is complete. During this time no further read commands can be activated via this block.

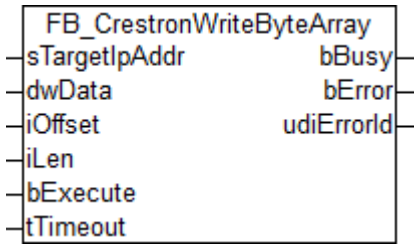
bError: A positive flank indicates that an error has occurred.

udiErrorId: Contains the [error code](#) [► 39] that may occur during block execution. If no error has occurred, the value is 0.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64	PC/CX	TcCrestron library from V1.0.0

8.3 FB_CrestronWriteByteArray



This function block allows a byte array with a maximum length of 40 fields to be written to the Crestron controller.

VAR_INPUT

```

sTargetIpAddr : STRING(15);
dwData       : DWORD;
iOffset      : INT;
iLen        : INT;
bExecute     : BOOL;
tTimeout     : TIME := T#5s;
  
```

sTargetIpAddr: IP address of the Crestron controller.

dwData: Address pointer to the byte array.

iOffset: Indicates the position in the *Buffer_Rx* from which writing to the Crestron controller should take place.

iLen: States the length of the array to be written. The length is restricted to 40.

bExecute: A positive edge starts the block.

tTimeout: Specifies the interval for awaiting a response.

VAR_OUTPUT

```

bBusy       : BOOL;
bError      : BOOL;
udiErrorId  : UDINT;
  
```

bBusy: During the write procedure, the *bBusy* output is set to TRUE and remains TRUE until writing has been completed. During this time no further commands can be activated via this block.

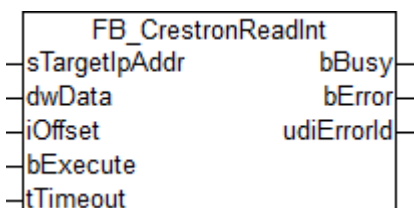
bError: The output is TRUE as soon as an error occurs during the execution of the command.

udiErrorId: Contains the error code [► 39] that may occur during execution of the block. The value is 0 if no error has occurred.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64	PC/CX	TcCrestron library from V1.0.0

8.4 FB_CrestronReadInt



This function block enables an integer value (16 bit) to be read from the Crestron controller.

VAR_INPUT

```
sTargetIpAddr : STRING(15);
dwData        : DWORD;
iOffset       : INT;
bExecute      : BOOL;
tTimeout      : TIME := T#5s;
```

sTargetIpAddr: IP address of the Crestron controller.

dwData: Address pointer to a PLC variable of data type INT (16 bit), in which the read value is stored.

iOffset: Indicates the offset in the buffer *Buffer_RX* of the Crestron controller from which the data are to be read.

bExecute: A positive edge starts the block.

tTimeout: Specifies the interval for awaiting a response.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
udiErrorId   : UDINT;
```

bBusy: During the read process *bBusy* is set to True and remains True until the read process is complete. During this time no further read commands can be activated via this block.

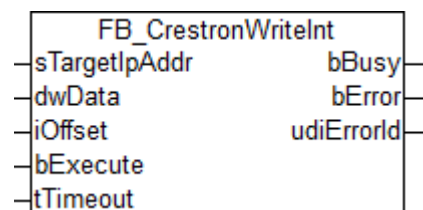
bError: A positive flank indicates that an error has occurred.

udiErrorId: Contains the [error code](#) [▶ 39] that may occur during block execution. If no error has occurred, the value is 0.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64	PC/CX	TcCrestron library from V1.0.0

8.5 FB_CrestronWriteInt



This function block enables an integer value (16 bit) to be written to the Crestron controller.

VAR_INPUT

```
sTargetIpAddr : STRING(15);
dwData        : DWORD;
iOffset       : INT;
bExecute      : BOOL;
tTimeout      : TIME := T#5s;
```

sTargetIpAddr: IP address of the Crestron controller.

dwData: Address pointer to a PLC variable of data type INT (16 bit), in which the value to be written is stored.

iOffset: Indicates the offset in the buffer *Buffer_TX* of the Crestron controller from which the data are to be written.

bExecute: A positive edge starts the block.

tTimeout: Specifies the interval for awaiting a response.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
udiErrorId : UDINT;
```

bBusy: During the write process *bBusy* is set to True and remains True until the writing process is complete. During this time no further read commands can be activated via this block.

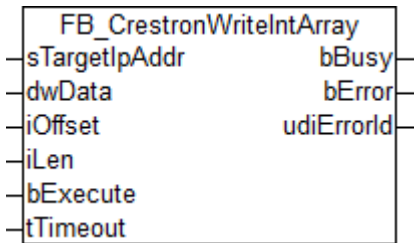
bError: A positive flank indicates that an error has occurred.

udiErrorId: Contains the error code [► 39] that may occur during block execution. If no error has occurred, the value is 0.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64	PC/CX	TcCrestron library from V1.0.0

8.6 FB_CrestronWriteIntArray



This function block allows an integer array with a maximum length of 40 fields to be written to the Crestron controller.

VAR_INPUT

```
sTargetIpAddr : STRING(15);
dwData        : DWORD;
iOffset       : INT;
iLen          : INT;
bExecute      : BOOL;
tTimeout      : TIME := T#5s;
```

sTargetIpAddr: IP address of the Crestron controller.

dwData: Address pointer to the integer array.

iOffset: Indicates the position in the *Buffer_Rx* from which writing to the Crestron controller should take place.

iLen: States the length of the array to be written. The length is restricted to 40.

bExecute: A positive edge starts the block.

tTimeout: Specifies the interval for awaiting a response.

VAR_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
udiErrorId : UDINT;
```

bBusy: During the write procedure, the *bBusy* output is set to TRUE and remains TRUE until writing has been completed. During this time no further commands can be activated via this block.

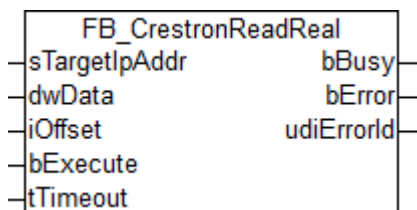
bError: The output is TRUE as soon as an error occurs during the execution of the command.

udiErrorId: Contains the error code [► 39] that may occur during execution of the block. The value is 0 if no error has occurred.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64	PC/CX	TcCrestron library from V1.0.0

8.7 FB_CrestronReadReal



This function block enables a Real (32 bit) to be read from the Crestron controller.



The value is distributed over 4 bytes in the Crestron controller. This means that four placeholders have to be used in the *Buffer_Tx* array of the Beckhoff Communication [► 25] or Beckhoff Communication 400 [► 26] module in order to suitably map a variable of type Real.

VAR_INPUT

```
sTargetIpAddr : STRING(15);
dwDat         : DWORD;
iOffset       : INT;
bExecute      : BOOL;
tTimeout      : TIME := T#5s;
```

sTargetIpAddr: IP address of the Crestron controller.

dwData: Address pointer to the Real (32-bit) in which the read value is stored.

iOffset: Indicates the position in the *Buffer_Tx* from which reading out of the Crestron controller should take place.

bExecute: A positive edge starts the block

tTimeout: Specifies the interval for awaiting a response.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
udiErrorId   : UDINT;
```

bBusy: During the read procedure, the *bBusy* output is set to TRUE and remains TRUE until reading has been completed. During this time no further commands can be activated via this block.

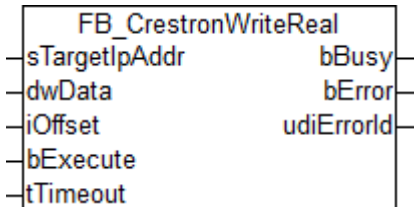
bError: The output is TRUE as soon as an error occurs during the execution of the command.

udiErrorId: Contains the error code [► 39] that may occur during execution of the block. The value is 0 if no error has occurred.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64	PC/CX	TcCrestron library from V1.0.0

8.8 FB_CrestronWriteReal



This function block enables a real (32 bit) to be written to the Crestron controller.

i The value is distributed over 4 bytes in the Crestron controller. Four placeholders are required in the Buffer_Rx of the [Beckhoff Communication \[► 25\]](#) or [Beckhoff Communication 400 \[► 26\]](#) module. The iOffset input describes the position in the Buffer_Rx array at which the writing of the 4 bytes begins.

VAR_INPUT

```
sTargetIpAddr : STRING(15);
dwData       : DWORD;
iOffset      : INT;
bExecute     : BOOL;
tTimeout     : TIME := T#5s;
```

sTargetIpAddr: IP address of the Crestron controller.

dwData: Address pointer to the Real (32-bit) from which the value to be written is read.

iOffset: Indicates the position in the *Buffer_Rx* from which writing to the Crestron controller should take place.

bExecute: A positive edge starts the block.

tTimeout: Specifies the interval for awaiting a response.

VAR_OUTPUT

```
bBusy       : BOOL;
bError      : BOOL;
udiErrorId  : UDINT;
```

bBusy: During the write procedure, the *bBusy* output is set to TRUE and remains TRUE until writing has been completed. During this time no further commands can be activated via this block.

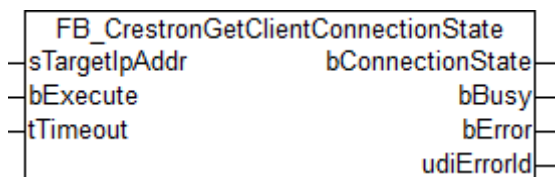
bError: The output is TRUE as soon as an error occurs during the execution of the command.

udiErrorId: Contains the [error code \[► 39\]](#) that may occur during execution of the block. The value is 0 if no error has occurred.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64	PC/CX	TcCrestron library from V1.0.0

8.9 FB_CrestronGetClientConnectionState



This function block queries the connection status of a Crestron controller.

VAR_INPUT

```
sTargetIpAddr : STRING(15);
bExecute      : BOOL;
tTimeout      : TIME := T#5s;
```

sTargetIpAddr: IP address of the Crestron controller.

bExecute: A positive edge starts the block

tTimeout: Specifies the interval for awaiting a response.

VAR_OUTPUT

```
bConnectionState : BOOL;
bBusy            : BOOL;
bError           : BOOL;
udiErrorId       : UDINT;
```

bConnectionState: The output is TRUE if a Crestron controller with the IP address entered at the input is connected to the Crestron Server.

bBusy: During the query procedure, the *bBusy* output is set to TRUE and remains TRUE until query has been completed. During this time no further commands can be activated via this block.

bError: The output is TRUE as soon as an error occurs during the execution of the command.

udiErrorId: Contains the error code [► 39] that may occur during execution of the block. The value is 0 if no error has occurred.

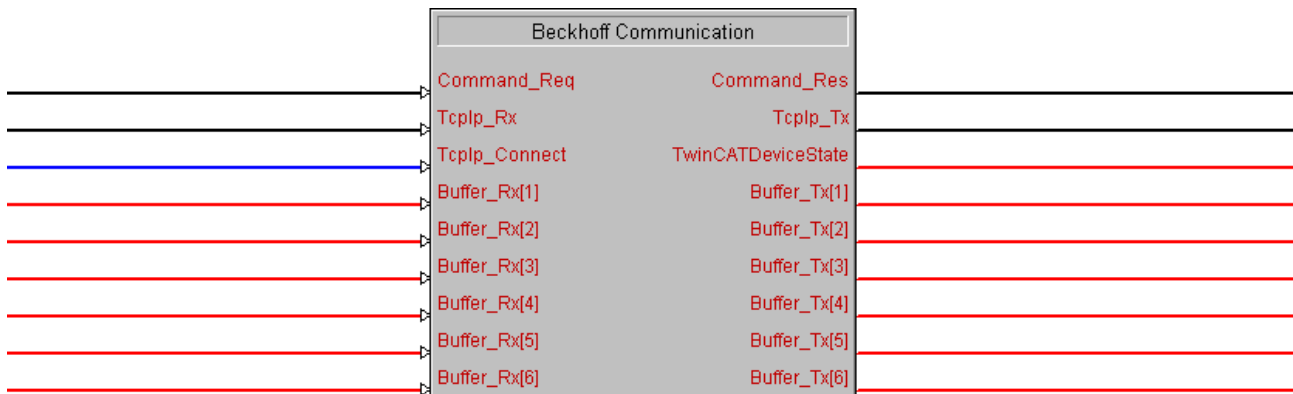
Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64	PC/CX	TcCrestron library from V1.0.0

9 Crestron SIMPL Windows User Modules

Name	Function
Beckhoff Communication [▶ 25]	User Module, that receives read/write data from TCP/IP-Client (up to 100 signals).
Beckhoff Communication 400 [▶ 26]	User Module, that receives read/write data from TCP/IP-Client (up to 400 signals).
Beckhoff Read Bit [▶ 27]	Read a value of type bit by Index Group and Index Offset.
Beckhoff Read Bit By Name [▶ 28]	Read a value of type bit by variable name.
Beckhoff Read Byte [▶ 30]	Read a value of type byte by Index Group and Index Offset.
Beckhoff Read Byte By Name [▶ 31]	Read a value of type byte by variable name.
Beckhoff Read Int [▶ 34]	Read a value of type integer by Index Group and Index Offset.
Beckhoff Read Int By Name [▶ 35]	Read a value of type integer by variable name.
Beckhoff Switch [▶ 37]	Write the value '1' for a positive edge and the value '0' for a negative edge of type bit by Index Group and Index Offset.
Beckhoff Switch By Name [▶ 38]	Write the value '1' for a positive edge and the value '0' for a negative edge of type bit by variable name.
Beckhoff Write Bit [▶ 29]	Write a value of type bit by Index Group and Index Offset.
Beckhoff Write Bit By Name [▶ 30]	Write a value of type bit by variable name.
Beckhoff Write Byte [▶ 32]	Write a value of type byte by Index Group and Index Offset.
Beckhoff Write Byte By Name [▶ 33]	Write a value of type byte by variable name.
Beckhoff Write Int [▶ 36]	Write a value of type integer by Index Group and Index Offset.
Beckhoff Write Int By Name [▶ 36]	Write a value of type integer by variable name.

9.1 Beckhoff Communication



The macro forms the basis for the communication between Beckhoff and Crestron via the Crestron controller. It receives read/write commands from Beckhoff via TCP/IP and converts them to signals Crestron can understand. In addition, Crestron signals are transformed such that the TwinCAT Crestron server can process them. Furthermore, the macro cyclically reads the AdsState of the TwinCAT device (see FAQ).

The macro can handle up to 100 signals.

Inputs

```

Command_Req      : SERIAL;
TcpIp_Rx         : SERIAL;
TcpIp_Connect    : DIGITAL;
Buffer_Rx[1..100] : ANALOG;
    
```

Command_Req: Signal from the individual write and read blocks. This signal contains the ADS command that is transferred to the TCP/IP client.

Tcplp_Rx: This signal is connected with the Rx-Output of the TCP/IP client.

Tcplp_Connect: The block establishes the connection as long as this input is "high".

Buffer_Rx[1..100]: Data with external read access.

Outputs

```
Command_Res      : SERIAL;
TcpIp_Tx         : SERIAL;
TwinCATDeviceState : ANALOG;
Buffer_Tx[1..100] : ANALOG;
```

Command_Res: Feedback signal for transfer to the individual write and read blocks.

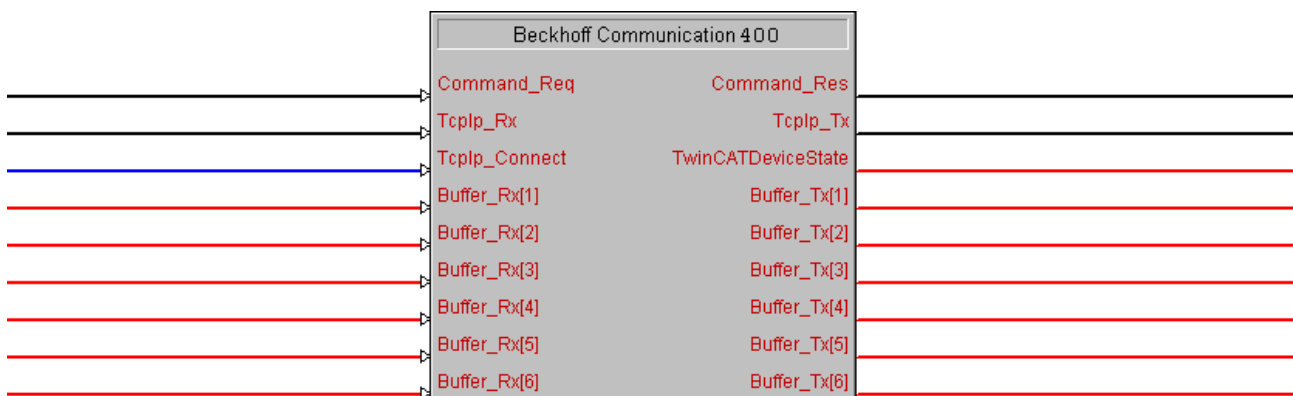
Tcplp_Tx: This signal is connected with the Tx-Input of the TCP/IP client.

TwinCATDeviceState: This analog signal contains the AdsState of TwinCAT PLC.

Buffer_Tx[1..100]: Data with external write access.

Development environment	Target platform
MP2E V4.001.1012 (Feb 17 2009) Build 7	2 -Series with network interface

9.2 Beckhoff Communication 400



The macro forms the basis for the communication between Beckhoff and Crestron via the Crestron controller. It receives read/write commands from Beckhoff via TCP/IP and converts them to signals Crestron can understand. In addition, Crestron signals are transformed such that the TwinCAT Crestron server can process them. Furthermore, the macro cyclically reads the AdsState of the TwinCAT device (see FAQ).

The macro can handle up to 400 signals.

Inputs

```
Command_Req      : SERIAL;
TcpIp_Rx         : SERIAL;
TcpIp_Connect    : DIGITAL;
Buffer_Rx[1..400] : ANALOG;
```

Command_Req: Signal from the individual write and read blocks. This signal contains the ADS command that is transferred to the TCP/IP client.

Tcplp_Rx: This signal is connected with the Rx-Output of the TCP/IP client.

Tcplp_Connect: The block establishes the connection as long as this input is "high".

Buffer_Rx[1..400]: Data with external read access.

Outputs

```
Command_Res      : SERIAL;
TcpIp_Tx        : SERIAL;
TwinCATDeviceState : ANALOG;
Buffer_Tx[1..400] : ANALOG;
```

Command_Res: Feedback signal for transfer to the individual write and read blocks.

Tcplp_Tx: This signal is connected with the Tx-Input of the TCP/IP client.

TwinCATDeviceState:This analog signal contains the AdsSate of TwinCAT PLC.

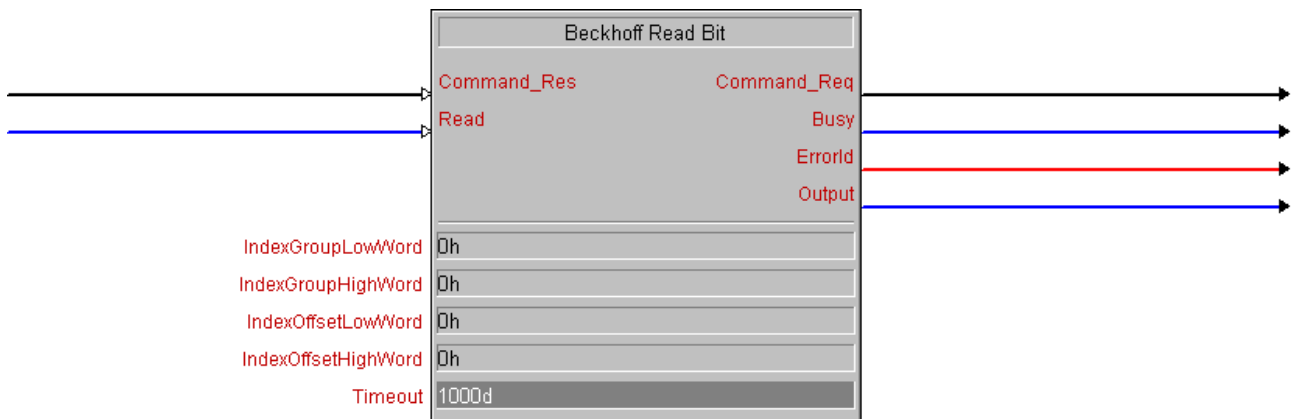
Buffer_Tx[1..400]: Data with external write access.



The module is available with Version 3.1.0 of the TwinCAT Crestron Server.

Development environment	Target platform
MP2E V4.001.1012 (Feb 17 2009) Build 7	2 -Series with network interface

9.3 Beckhoff Read Bit



This block reads a Boolean variable from a TwinCAT PLC and makes it available as a digital signal at the *output*.

Inputs

```
Commande_Res    : SERIAL;
Read            : DIGITAL;
```

Command_Res: This signal is connected to the *Command_Res* output of the Beckhoff Communication [▶ 25] or Beckhoff Communication 400 [▶ 26] block. It is used for transferring feedback.

Read: The read command is activated via a rising edge at this input.

Outputs

```
Command_Req     : SERIAL;
Busy            : DIGITAL;
ErrorId         : ANALOG;
Output          : DIGITAL;
```

Command_Req: This signal is connected to the *Command_Req* input of the Beckhoff Communication [▶ 25] or Beckhoff Communication 400 [▶ 26] block. It is used for transferring commands.

Busy: During the transfer the output is set and remains active until the signal has been read. During this time no further read commands can be activated via this block.

ErrorId: This analog signal contains the error code that may occur during command execution. If no error has occurred, this signal has the value 0.

Output: The value that was read is output via this digital signal.

Parameters

```
IndexGroupLowWord : ANALOG;
IndexGroupHighWord : ANALOG;
IndexOffsetLowWord : ANALOG;
IndexOffsetHighWord : ANALOG;
Timeout : ANALOG;
```

IndexGroupLowWord: The low-order word of the Index Group.

IndexGroupHighWord: The high-order word of the Index Group.

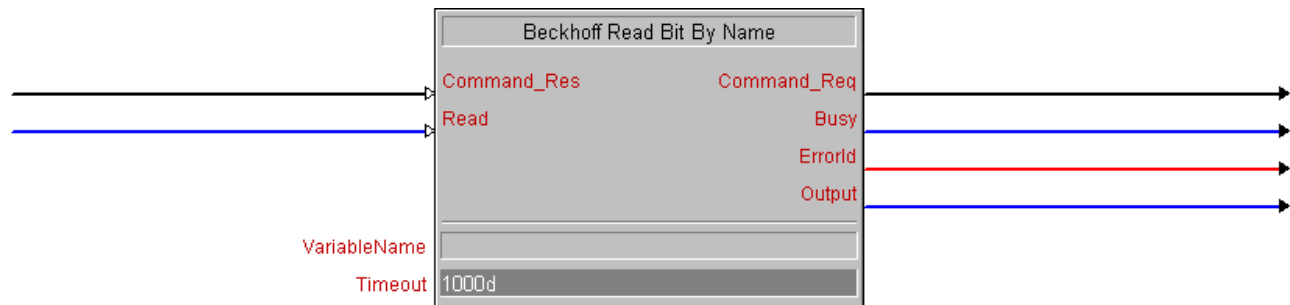
IndexOffsetLowWord: The low-order word of the Index Offset.

IndexOffsetHighWord: The high-order word of the Index Offset.

Timeout: Specifies the interval for awaiting a response. The unit is ms.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.4 Beckhoff Read Bit By Name



This block reads a Boolean variable from a TwinCAT PLC and makes it available as a digital signal at the *output*.

Inputs

```
Command_Res : SERIAL;
Read : DIGITAL;
```

Command_Res: This signal is connected to the *Command_Res* output of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring feedback.

Read: The read command is activated via a rising edge at this input.

Outputs

```
Command_Req : SERIAL;
Busy : DIGITAL;
ErrorId : ANALOG;
Output : DIGITAL;
```

Command_Req: This signal is connected to the *Command_Req* input of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring commands.

Busy: During the transfer the output is set and remains active until the signal has been read. During this time no further read commands can be activated via this block.

ErrorId: This analog signal contains the error code that may occur during command execution. If no error has occurred, this signal has the value 0.

Output: The value that was read is output via this digital signal.

Parameters

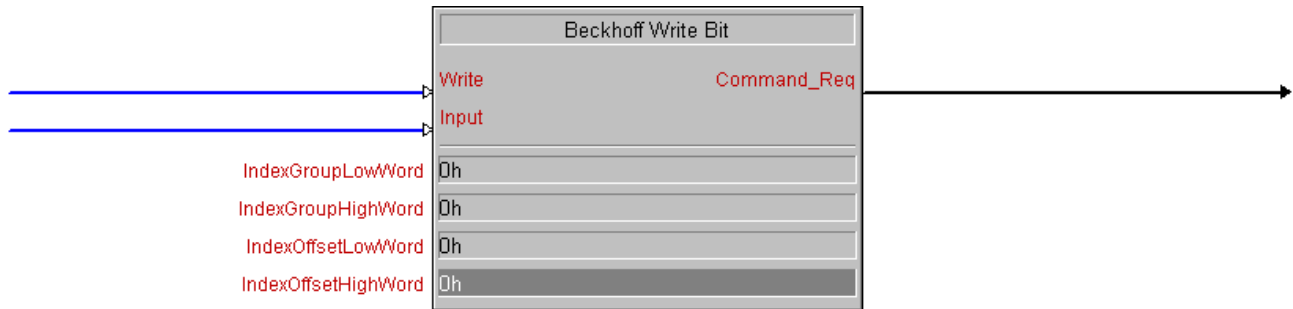
```
VariableName : SERIAL;
Timeout      : ANALOG;
```

VariableName: Specifies the name of the variable to be read in the TwinCAT PLC.

Timeout: Specifies the interval for awaiting a response. The unit is ms.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.5 Beckhoff Write Bit



This block writes the value of the Boolean variable that is present at the *Input* to the TwinCAT PLC.

Inputs

```
Command_Res : SERIAL;
Write       : DIGITAL;
Input       : DIGITAL;
```

Write: The write command is activated by a positive edge at this input.

Input: This value will be transferred.

Outputs

```
Command_Req : SERIAL;
```

Command_Req: This signal is connected to the *Command_Req* input of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring commands.

Parameters

```
IndexGroupLowWord : ANALOG;
IndexGroupHighWord : ANALOG;
IndexOffsetLowWord : ANALOG;
IndexOffsetHighWord : ANALOG;
```

IndexGroupLowWord: The low-order word of the Index Group.

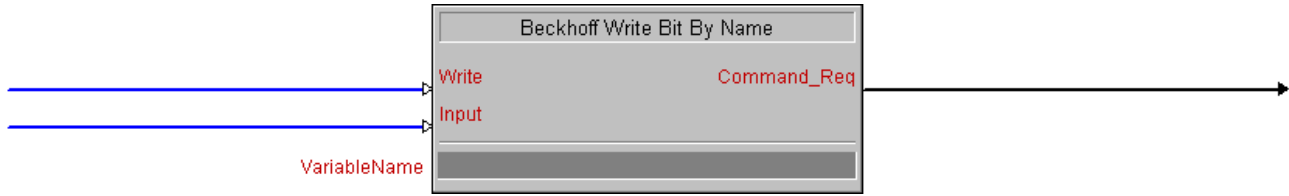
IndexGroupHighWord: The high-order word of the Index Group.

IndexOffsetLowWord: The low-order word of the Index Offset.

IndexOffsetHighWord: The high-order word of the Index Offset.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.6 Beckhoff Write Bit By Name



This block writes the value of the Boolean variable that is present at the *Input* to the TwinCAT PLC.

Inputs

```
Write : DIGITAL;
Input : DIGITAL;
```

Write: The write command is activated by a positive edge at this input.

Input: This value will be transferred.

Outputs

```
Command_Req : SERIAL;
```

Command_Req: This signal is connected to the *Command_Req* input of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring commands.

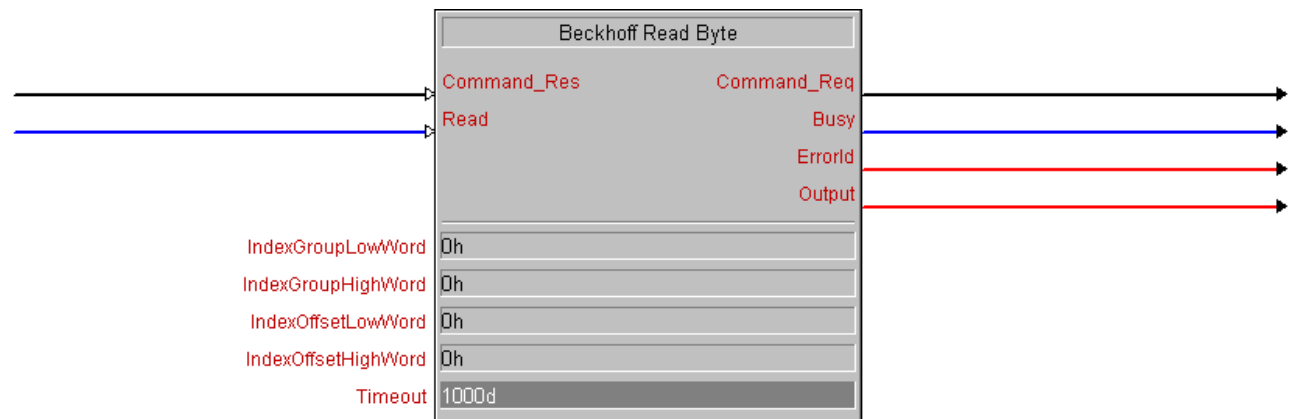
Parameters

```
VariableName : SERIAL;
```

VariableName: Specifies the name of the variable to be described in the TwinCAT PLC.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.7 Beckhoff Read Byte



This block reads an 8-bit variable from a TwinCAT PLC and makes it available as an analog signal at the *output*.

Inputs

```
Command_Res : SERIAL;
Read : DIGITAL;
```

Command_Res: This signal is connected to the *Command_Res* output of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring feedback.

Read: The read command is activated via a rising edge at this input.

Outputs

```
Command_Req : SERIAL;
Busy        : DIGITAL;
ErrorId     : ANALOG;
Output      : ANALOG;
```

Command_Req: This signal is connected to the *Command_Req* input of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring commands.

Busy: During the transfer the output is set and remains active until the signal has been read. During this time no further read commands can be activated via this block.

ErrorId: This analog signal contains the error code that may occur during command execution. If no error has occurred, this signal has the value 0.

Output: The value that was read is output via this analog signal.

Parameters

```
IndexGroupLowWord : ANALOG;
IndexGroupHighWord : ANALOG;
IndexOffsetLowWord : ANALOG;
IndexOffsetHighWord : ANALOG;
Timeout           : ANALOG;
```

IndexGroupLowWord: The low-order word of the Index Group.

IndexGroupHighWord: The high-order word of the Index Group.

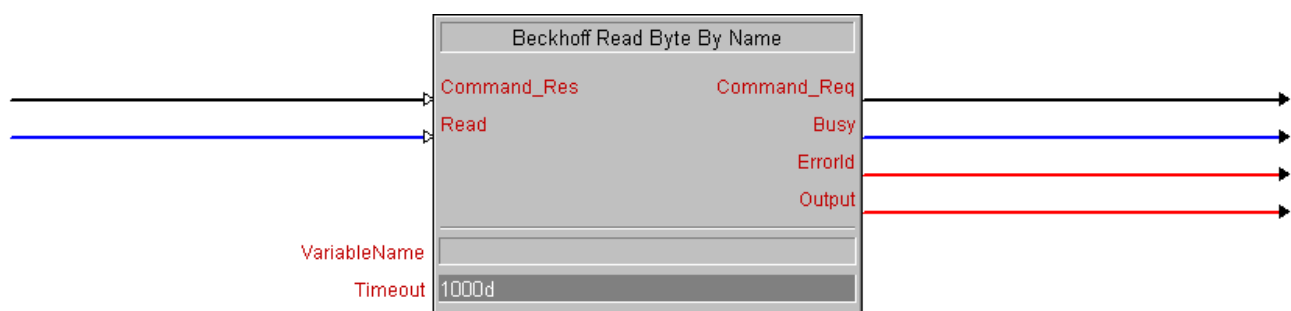
IndexOffsetLowWord: The low-order word of the Index Offset.

IndexOffsetHighWord: The high-order word of the Index Offset.

Timeout: Specifies the interval for awaiting a response. The unit is ms.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.8 Beckhoff Read Byte By Name



This block reads an 8-bit variable from a TwinCAT PLC and makes it available as an analog signal at the *output*.

Inputs

```
Command_Res : SERIAL;
Read        : DIGITAL;
```

Command_Res: This signal is connected to the *Command_Res* output of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring feedback.

Read: The read command is activated via a rising edge at this input.

Outputs

```
Command_Req : SERIAL;
Busy        : DIGITAL;
ErrorId     : ANALOG;
Output      : ANALOG;
```

Command_Req: This signal is connected to the *Command_Req* input of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring commands.

Busy: During the transfer the output is set and remains active until the signal has been read. During this time no further read commands can be activated via this block.

ErrorId: This analog signal contains the error code that may occur during command execution. If no error has occurred, this signal has the value 0.

Output: The value that was read is output via this analog signal.

Parameters

```
VariableName : SERIAL;
Timeout      : ANALOG;
```

VariableName: Specifies the name of the variable to be read in the TwinCAT PLC.

Timeout: Specifies the interval for awaiting a response. The unit is ms.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.9 Beckhoff Write Byte

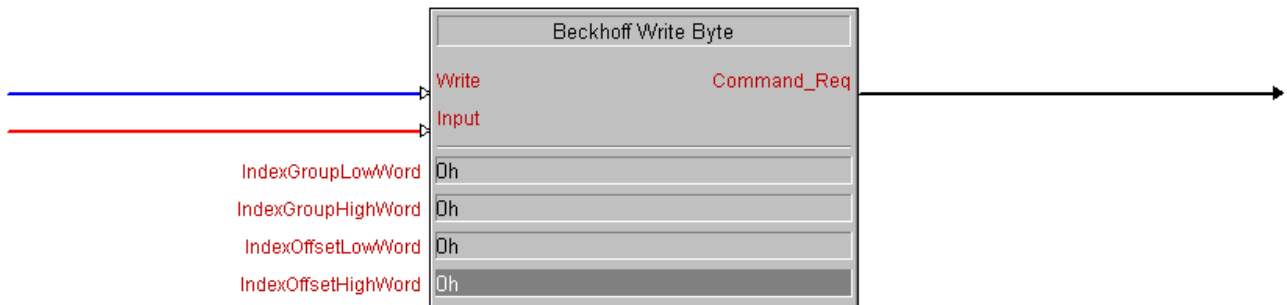


Fig. 2: Beckhoff_Write_Byte_Symbol

This block writes the value of the 8-bit variable that is present at the *Input* to a TwinCAT PLC.

Inputs

```
Write : DIGITAL;
Input  : ANALOG;
```

Write: The write command is activated by a positive edge at this input.

Input: This value will be transferred.

Outputs

```
Command_Req : SERIAL;
```

Command_Req: This signal is connected to the *Command_Req* input of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring commands.

Parameters

```
IndexGroupLowWord : ANALOG;
IndexGroupHighWord : ANALOG;
IndexOffsetLowWord : ANALOG;
IndexOffsetHighWord : ANALOG;
```

IndexGroupLowWord: The low-order word of the Index Group.

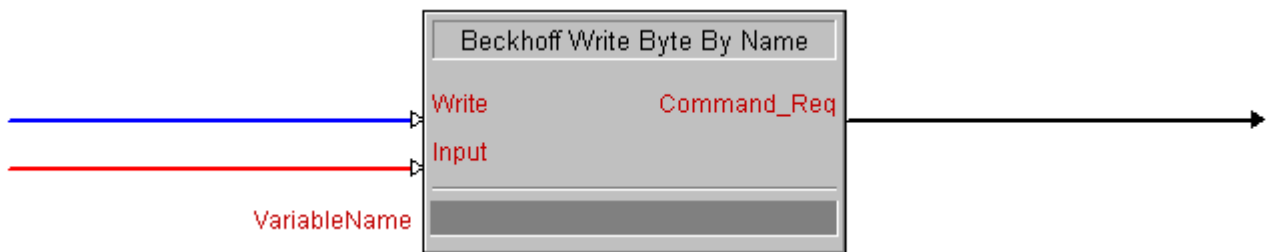
IndexGroupHighWord: The high-order word of the Index Group.

IndexOffsetLowWord: The low-order word of the Index Offset.

IndexOffsetHighWord: The high-order word of the Index Offset.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.10 Beckhoff Write Byte By Name



This block writes the value of the 8-bit variable that is present at the *Input* to a TwinCAT PLC.

Inputs

```
Write : DIGITAL;
Input : ANALOG;
```

Write: The write command is activated by a positive edge at this input.

Input: This value will be transferred.

Outputs

```
Command_Req : SERIAL;
```

Command_Req: This signal is connected to the *Command_Req* input of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring commands.

Busy: During the transfer the output is set and remains active until the signal has been read. During this time no further read commands can be activated via this block.

ErrorId: This analog signal contains the error code that may occur during command execution. If no error has occurred, this signal has the value 0.

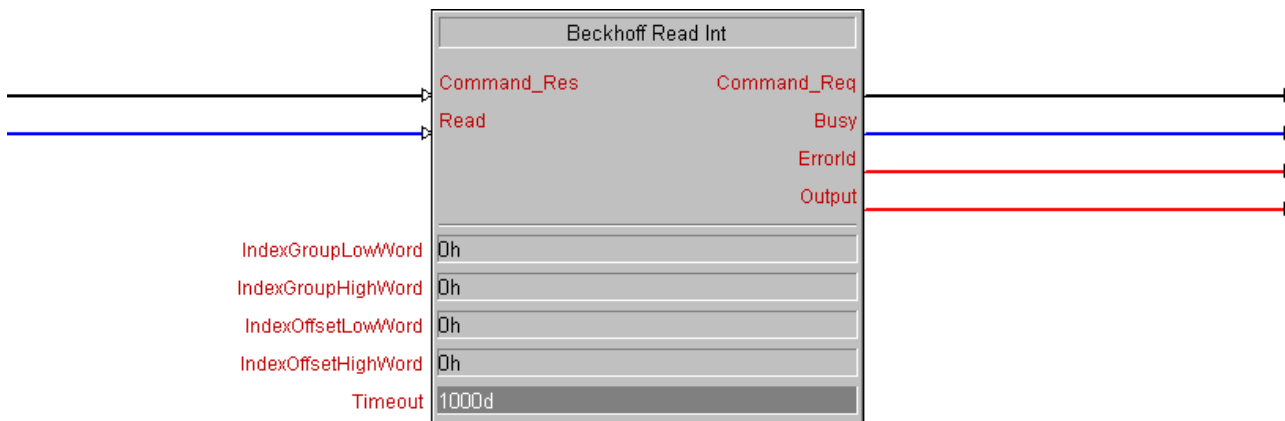
Parameters

```
VariableName : SERIAL;
```

VariableName: Specifies the name of the variable to be described in the TwinCAT PLC.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.11 Beckhoff Read Int



This block reads an 16-bit variable from a TwinCAT PLC and makes it available as an analog signal at the *output*.

Inputs

```
Command_Res : SERIAL;
Read        : DIGITAL;
```

Command_Res: This signal is connected to the *Command_Req* output of the [Beckhoff Communication \[▶ 25\]](#) or [Beckhoff Communication 400 \[▶ 26\]](#) block. It is used for transferring feedback.

Read: The read command is activated via a rising edge at this input.

Outputs

```
Command_Req : SERIAL;
Busy        : DIGITAL;
ErrorId     : ANALOG;
Output      : ANALOG;
```

Command_Req: This signal is connected to the *Command_Res* input of the [Beckhoff Communication \[▶ 25\]](#) or [Beckhoff Communication 400 \[▶ 26\]](#) block. It is used for transferring commands.

Busy: During the transfer the output is set and remains active until the signal has been read. During this time no further read commands can be activated via this block.

ErrorId: This analog signal contains the error code that may occur during command execution. If no error has occurred, this signal has the value 0.

Output: The value that was read is output via this analog signal.

Parameters

```
IndexGroupLowWord : ANALOG;
IndexGroupHighWord : ANALOG;
IndexOffsetLowWord : ANALOG;
IndexOffsetHighWord : ANALOG;
Timeout           : ANALOG;
```

IndexGroupLowWord: The low-order word of the Index Group.

IndexGroupHighWord: The high-order word of the Index Group.

IndexOffsetLowWord: The low-order word of the Index Offset.

IndexOffsetHighWord: The high-order word of the Index Offset.

Timeout: Specifies the interval for awaiting a response. The unit is ms.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.12 Beckhoff Read Int By Name



This block reads an 16-bit variable from a TwinCAT PLC and makes it available as an analog signal at the *output*.

Inputs

```
Command_Res      : SERIAL;
Read             : DIGITAL;
```

Command_Res: This signal is connected to the *Command_Res* output of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring feedback.

Read: The read command is activated via a rising edge at this input.

Outputs

```
Command_Req     : SERIAL;
Busy           : DIGITAL;
ErrorId        : ANALOG;
Output         : ANALOG;
```

Command_Req: This signal is connected to the *Command_Req* input of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring commands.

Busy: During the transfer the output is set and remains active until the signal has been read. During this time no further read commands can be activated via this block.

ErrorId: This analog signal contains the error code that may occur during command execution. If no error has occurred, this signal has the value 0.

Output: The value that was read is output via this analog signal.

Parameters

```
VariableName    : SERIAL;
Timeout         : ANALOG;
```

VariableName: Specifies the name of the variable to be read in the TwinCAT rPLC.

Timeout: Specifies the interval for awaiting a response. The unit is ms.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.13 Beckhoff Write Int

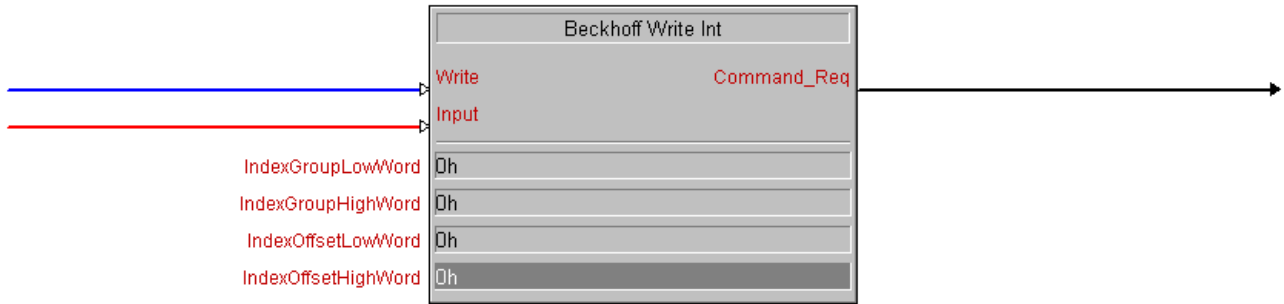


Fig. 3: Beckhoff_Write_Int_Symbol

This block writes the value of the 16-bit variable that is present at the *Input* to a TwinCAT PLC.

Inputs

```
Write : DIGITAL;
Input : ANALOG;
```

Write: The write command is activated by a positive edge at this input.

Input: This value will be transferred.

Outputs

```
Command_Req : SERIAL;
```

Command_Req: This signal is connected to the *Command_Req* input of the [Beckhoff Communication](#) [▶ 25] or [Beckhoff Communication 400](#) [▶ 26] block. It is used for transferring commands.

Parameters

```
IndexGroupLowWord : ANALOG;
IndexGroupHighWord : ANALOG;
IndexOffsetLowWord : ANALOG;
IndexOffsetHighWord : ANALOG;
```

IndexGroupLowWord: The low-order word of the Index Group.

IndexGroupHighWord: The high-order word of the Index Group.

IndexOffsetLowWord: The low-order word of the Index Offset.

IndexOffsetHighWord: The high-order word of the Index Offset.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.14 Beckhoff Write Int By Name

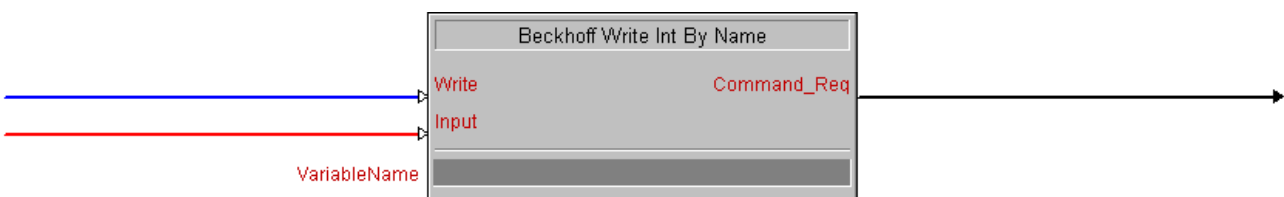


Fig. 4: Beckhoff_Write_Int_by_Name_Symbol

This block writes the value of the 16-bit variable that is present at the *Input* to a TwinCAT PLC.

Inputs

```
Write : DIGITAL;
Input : ANALOG;
```

Write: The write command is activated by a positive edge at this input.

Input: This value will be transferred.

Outputs

```
Command_Req : SERIAL;
```

Command_Req: This signal is connected to the *Command_Res* input of the [Beckhoff Communication \[► 25\]](#) or [Beckhoff Communication 400 \[► 26\]](#) block. It is used for transferring commands.

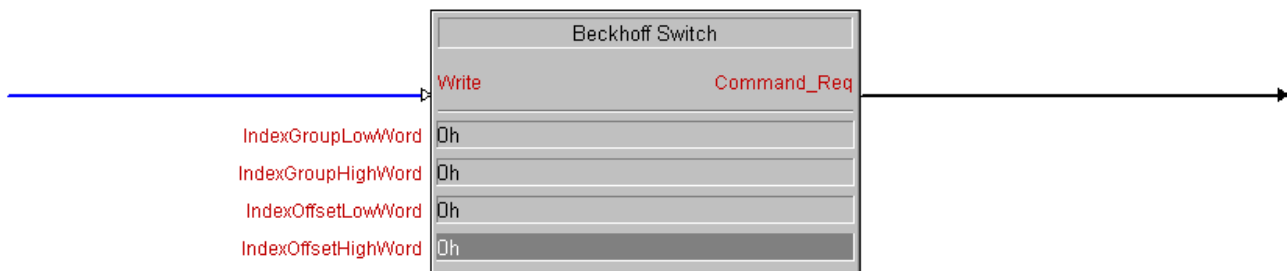
Parameters

```
VariableName : SERIAL;
```

VariableName: Specifies the name of the variable to be described in the TwinCAT PLC.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.15 Beckhoff Switch



This block writes the value '1' for a positive edge at input *Write* and the value '0' for a negative edge at input *Write* to a TwinCAT PLC.

Inputs

```
Write : DIGITAL;
```

Write: Value '1' is written via a positive edge at this input, value '0' is written via a negative edge.

Outputs

```
Command_Req : SERIAL;
```

Command_Req: This signal is connected to the *Command_Res*-input of the [Beckhoff Communication \[► 25\]](#) or [Beckhoff Communication 400 \[► 26\]](#) block. It is used for transferring commands.

Parameters

```
IndexGroupLowWord : ANALOG;
IndexGroupHighWord : ANALOG;
IndexOffsetLowWord : ANALOG;
IndexOffsetHighWord : ANALOG;
```

IndexGroupLowWord: The low-order word of the Index Group.

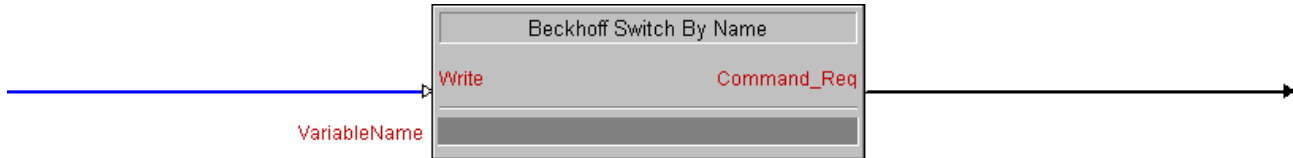
IndexGroupHighWord: The high-order word of the Index Group.

IndexOffsetLowWord: The low-order word of the Index Offset.

IndexOffsetHighWord: The high-order word of the Index Offset.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

9.16 Beckhoff Switch By Name



This block writes the value '1' for a positive edge at input *Write* and the value '0' for a negative edge at input *Write* to a TwinCAT PLC.

Inputs

Write : DIGITAL;

Write: Value '1' is written via a positive edge at this input, value '0' is written via a negative edge.

Outputs

Command_Req : SERIAL;

Command_Req: This signal is connected to the *Command_Req*-input of the [Beckhoff Communication I ▶ 25](#) or [Beckhoff Communication 400 ▶ 26](#) block. It is used for transferring commands.

Parameters

VariableName : SERIAL;

VariableName: Specifies the name of the variable to be described in the TwinCAT PLC.

Development environment	Target platform
MP2E V4.001.1012(Feb 17 2009) Build 7	2 -Series with network interface

10 Error Codes

Hex code	Dec code	Description
0x8501	34049	Input parameter error on a PLC function block.
0x8502	34050	The configuration file TcCrestronServer.cfg was not found
0x8503	34051	Internal server error while processing the configuration file TcCrestronServer.cfg .
0x8504	34052	Error in the configuration file TcCrestronServer.cfg . The XML node " TcpPort " was not found.
0x8505	34053	Error in the configuration file TcCrestronServer.cfg . The XML node " AmsPort " was not found.
0x8506	34054	The client was not found.
0x8507	34055	Error while processing a read request.
0x8508	34056	Error while processing a write request.
0x8509	34057	Error while processing an array read request.
0x850A	34058	Error while processing an array write request.
0x850B	34059	Error while reading a client status. The specified client was not found.
0x850C	34060	Time exceeded while reading from a PLC value.
0x850D	34061	PLC function block input parameter sTargetIpAddr is wrong.
0x850E	34062	PLC function block input parameter dwData is wrong.
0x850F	34063	PLC function block input parameter iOffset is wrong.
0x8510	34064	PLC function block input parameter iLen is wrong.
0x8511	34065	PLC function block input parameter iLen is wrong. iLen has exceeded the maximum value of 40.
0x8512	34066	PLC function block input parameter error. iOffset plus iLen is greater than the maximum allowable size of 100 fields. The function module writes beyond the allowed fields, values are lost.
0x8F07	36615	Error in the configuration file TcCrestronServer.cfg . The XML node " AdsMessageCycleTime " was not found.

11 Samples

11.1 Beckhoff PLC programming examples

Requirements

The following components are required:

- PC with TwinCAT from V2.10 Build 1330 or newer.
- CX device with WinCE + **.NET Compact Framework** or CX device / IPC with CE or Windows XP Embedded + **.NET Framework**.
- Beckhoff supplement: TwinCAT Crestron Server.
- SIMPL Windows V2.08.44 with Crestron Data Base V18.09.02.001 or newer.

Description	Download
Write Int: This example demonstrates how an integer (16-bit) is sent to a Crestron controller and writes an input signal.	https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364626955/.zip https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364626955/.zip
Write Int Array: This example demonstrates how an integer (16-bit) array of ten fields is sent to a Crestron controller and writes ten input signals there.	https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364628363/.zip
Read Int: This example demonstrates how an integer (16-bit) value is read from the output signal of a Crestron controller.	https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364629771/.zip
Get Client Connection State: This example demonstrates how the information is read out regarding whether a Crestron controller with a certain IP address is connected to the TwinCAT Crestron Server.	https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364631179/.zip

11.2 Crestron programming examples

Requirements

The following components are required:

- PC with TwinCAT from V2.10 Build 1330 or higher.
- CX device with WinCE + **.NET Compact Framework** or CX device / IPC with CE or Windows XP Embedded + **.NET Framework**.
- Beckhoff supplement: TwinCAT Crestron Server.

- SIMPL Windows V2.08.44 with Crestron Data Base V18.09.02.001 or higher.



The following Crestron examples were created on a Crestron MP2E V4.001.1012 (Feb 17 2009) Build 7. Please bear this in mind when using the examples.

Description	Download
<p>Write Int: This example demonstrates how an integer (16-bit) value is written to the Beckhoff PLC using the Beckhoff module “<i>Write Int By Name</i> and <i>Write Int By Index Group / Index Offset</i>” for Crestron.</p>	<p>https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364632587/.zip</p> <p>https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364632587/.zip</p>
<p>Read Int: This example demonstrates how an integer (16-bit) value is read out of the Beckhoff PLC using the Beckhoff module “<i>Read Int By Name</i> and <i>Read Int By Index Group / Index Offset</i>” for Crestron.</p>	<p>https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364632587/.zip</p> <p>https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364633995/.zip</p>
<p>Switch modules: This example describes how the Beckhoff module “<i>Switch By Name</i> and <i>Switch By Index Group / Index Offset</i>” accesses a toggle variable in the Beckhoff PLC for writing.</p>	<p>https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364632587/.zip</p> <p>https://infosys.beckhoff.com/content/1033/tcsample_crestron/Resources/11364635403/.zip</p>

More Information:
www.beckhoff.com/ts8036

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

