

Manual

# **TC3 EthernetIP Scanner**

TwinCAT 3

Version: 1.1 Date: 2019-01-09 Order No.: TF6281



# Table of contents

1	Fore	word	5						
	1.1	Notes on the documentation	5						
	1.2	Safety instructions	6						
2	Over	view	7						
3	Requ	uirements							
4	Licer	nsing	9						
5	Conf	iguration	14						
	5.1	EtherNet/IP	14						
	5.2	Sync Task	15						
	5.3	Settings dialog	15						
		5.3.1 Firewall setting	17						
	5.4	Diag History	18						
5.5 Connecting EtherNet/IP slaves									
5.6 PLC to PLC communication									
		5.6.1 Allen-Bradley CompactLogix	24						
	5.7	Data Table Read and Write	28						
	5.8	Diagnostics	34						
6	PLC	API	36						
	6.1	6.1 Function blocks							
		6.1.1 FB_GET_ATTRIBUTE_SINGLE	36						
		6.1.2 FB_SET_ATTRIBUTE_SINGLE	37						
		6.1.3 FB_CUSTOM_SERVICE	38						
		6.1.4 FB_CIP_DATA_TABLE_RDWR	40						
	6.2	Functions	41						
		6.2.1 RSL5KSTRING_TO_STRING	41						
		6.2.2 STRING_TO_RSL5KSTRING	42						
		6.2.3 F_GET_ETHERNETIP_ERROR_TEXT	42						
	6.3	Data types	42						
		6.3.1 RSL5K_STRING	42						
7	Арре	endix	43						
	7.1	Prepare Wireshark recording	43						
	7.2	Error Codes TF6281	44						
	7.3 Support and Service								

# 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 the 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<sup>®</sup>, TwinCAT<sup>®</sup>, EtherCAT<sup>®</sup>, Safety over EtherCAT<sup>®</sup>, TwinSAFE<sup>®</sup>, XFC<sup>®</sup> and XTS<sup>®</sup> 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, DE102004044764, DE102007017835

with corresponding applications or registrations in various other countries.

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

EP0851348, US6167425 with corresponding applications or registrations in various other countries.

### Ether**CAT**

EtherCAT<sup>®</sup> is 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.

### 

### **Risk of injury!**

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

### 

### **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 Overview

The function TF6281 is an EtherNet/IP scanner or master. Here you can connect EtherNet/IP slaves. TF6281 is a software extension that turns an Ethernet interface with Intel chipset into an EtherNet/IP scanner. The real-time driver for the Ethernet interface must be installed for this purpose. The driver is part of the TwinCAT system. This driver is pre-installed on Beckhoff IPCs and can be used on almost all hardware platforms with Intel Ethernet chipset. If you are using a third-party PC, you may need to check or install it.

### TC3 function: EtherNet/IP scanner TF6281

Technical data	TF6281							
Requires	TC1200 fro of the func	om build 40 tion	22.14, witho	out TC1200	) it is not po	essible to us	se the full fu	unctionality
Target system	Windows X	KP, Windov	vs 7/8, Wind	dows CE				
Performance class	20	30	40	50	60	70	80	90
(pp)	_	-	Х	Х	Х	Х	Х	Х

### Technical data of the EtherNet/IP scanner

TF6281	4022.0
Remote Nodes (Boxes) [Producer Object counts 1]	128
Client Connections	128
Server Connections	128
CIP Connections	256
Produced Tag	12
Consumed tag for each EtherNet/IP device	12

### **Ordering information**

TF6281-00pp TC3 EtherNet/IP s	canner
-------------------------------	--------

### EtherNet/IP

# EtherNet/IP<sup>®</sup>

EtherNet/IP (Ethernet Industrial Protocol, EIP) is a real-time Ethernet protocol, which was disclosed and standardized by the ODVA (Open DeviceNet Vendor Association). The protocol is based on TCP, UDP and IPv4.

Further information can be found at www.odva.org or https://en.wikipedia.org/wiki/Ethernet/IP.

# 3 Requirements

### Software

The TF6281 requires **TwinCAT** version **3.1** Build **4022.14** or higher. No further installation is required.

### Hardware

To use the TF6281, it is necessary that a real-time driver for the Ethernet interface is installed on the target system.

Beckhoff PC systems are usually preconfigured for the operation of EtherNet/IP devices.

# 4 Licensing

The TwinCAT 3 Function can be activated as a full version or as a 7-day test version. Both license types can be activated via the TwinCAT 3 development environment (XAE).

The licensing of a TwinCAT 3 Function is described below. The description is divided into the following sections:

- Licensing a 7-day test version [▶ 9]
- <u>Licensing a full version [▶ 10]</u>

Further information on TwinCAT 3 licensing can be found in the "Licensing" documentation in the Beckhoff Information System (TwinCAT 3 > Licensing).

### Licensing a 7-day test version

- 1. Start the TwinCAT 3 development environment (XAE).
- 2. Open an existing TwinCAT 3 project or create a new project.
- 3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the toolbar.
  - ⇒ The licensing settings always refer to the selected target system. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.
- 4. In the **Solution Explorer**, double-click **License** in the **SYSTEM** subtree.



- ⇒ The TwinCAT 3 license manager opens.
- 5. Open the **Manage Licenses** tab. In the **Add License** column, check the check box for the license you want to add to your project (e.g. "TF6420: TC3 Database Server").

T٧	TwinCAT SampleProject 🕫 🗙									
	Order Information (Runtime) Manage Licenses Project Licenses Online Licenses									
	Disable automatic detection of required licenses for project									
	Order No	License	Add License	•						
	TF6310	ТСЗ ТСР/ІР	cpu license							
	TF6311	TC3 TCP/UDP RT	cpu license							
	TF6340	TC3 Serial-Communication	cpu license							
	TF6350	TC3 SMS-SMTP	cpu license							
	TF6360	TC3 Virtual-Serial-COM	cpu license							
	TF6420	TC3 Database-Server	🗹 cpu license							

6. Open the Order Information (Runtime) tab.

⇒ In the tabular overview of licenses, the previously selected license is displayed with the status "missing".

inCAT SampleProject 😕 X									
der Information (Runtime) Manage Licenses Project Licenses Online Licenses									
License Devic	License Device Target (Hardware Id)								
System Id:		Platf	form:						
E13751F7-79E	39-4063-416C-615ED53FA7BD	othe	ər (90)	•					
License Reque	est								
Provider:	Beckhoff Automation	<b></b>	Generate File						
License Id:		Customer Id:							
Comment									
License Activat	ion								
71	Days Trial License	Licens	se Response File						
Order No         License         Instances         License TAN         Current Status         License Id         A									
TC1200	TC3 PLC	cpu license		expires on Jul 7, 2018 (trial license)	66689887-CCBD-452C-AC9A-039D997C6E66				
TF6420	TC3 Database-Server	cpu license		missing	92583661-35AE-45CE-BD4F-C35BFE16F07E				
TF6710	TC3 IoT Functions	cpu license		expires on Jul 7, 2018 (trial license)	2149932B-0B77-4004-B43F-E85CEEFF347D				

7. Click 7-Day Trial License... to activate the 7-day trial license.

winCAT SampleProject 🤫 🗙									
Order Information (Runtime) Manage Licenses Project Licenses Online Licenses									
License Device Target (Hardware Id)									
System Id:		PI	atform:						
E13751F7-79	B9-4063-416C-615ED53FA7BE	) 0	ther (90)	•					
License Requ	est								
Provider:	Beckhoff Automation	▼	Generate File						
License Id:		Customer Id:							
Comment:									
License Activa	tion								
7	Days Trial License	Lice	nse Response File						
Order No License Instances License TAN Current Status License Id									
TC1200	TC3 PLC	cpu license		expires on Jul 11, 2018 (trial license)	66689887-CCBD-452C-AC9A-039D997C6E66				
TF6420	TC3 Database-Server	cpu license		expires on Jul 11, 2018 (trial license)	92583661-35AE-45CE-BD4F-C35BFE16F07E				
TF6710	TC3 IoT Functions	cpu license		expires on Jul 11, 2018 (trial license)	2149932B-0B77-4004-B43F-E85CEEFF347D				

- ⇒ A dialog box opens, prompting you to enter the security code displayed in the dialog.
- 8. Enter the code exactly as it appears, confirm it and acknowledge the subsequent dialog indicating successful activation.
  - ⇒ In the tabular overview of licenses, the license status now indicates the expiration date of the license.
- 9. Restart the TwinCAT system.
- $\Rightarrow$  The 7-day trial version is enabled.

### Licensing a full version

- 1. Start the TwinCAT 3 development environment (XAE).
- 2. Open an existing TwinCAT 3 project or create a new project.
- 3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the toolbar.
  - ⇒ The licensing settings always refer to the selected target system. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.

4. In the Solution Explorer, double-click License in the SYSTEM subtree.



- $\Rightarrow$  The TwinCAT 3 license manager opens.
- 5. Open the **Manage Licenses** tab. In the **Add License** column, check the check box for the license you want to add to your project (e.g. "TE1300: TC3 Scope View Professional").

TwinCAT SampleProject 😕 🗙										
C	Order Information (Runtime) Manage Licenses Project Licenses Online Licenses									
	Disable automatic detection of required licenses for project									
	Order No	License	Add License	-						
	TE1110	TC3 Simulation Manager	cpu license							
	TE1111	TC3 EtherCAT Simulation	Cpu license							
	TE1120	TC3 XCAD Interface	Cpu license							
	TE1130	TC3 CAD Simulation Interface	cpu license							
	TE1200	TC3 PLC Static Analysis	cpu license							
	TE1210	TC3 PLC Profiler	cpu license							
	TE1300	TC3 Scope View Professional	🗹 cpu license							

### 6. Open the Order Information tab.

⇒ In the tabular overview of licenses, the previously selected license is displayed with the status "missing".

der Information (Runtime) Manage Licenses Project Licenses Online Licenses										
License Device Target (Hardware Id)										
System Id:		P	atform:							
E13751F7-79	B9-4063-416C-615ED53FA7BD	G	ther (90)	•						
License Requ	est									
Provider:	Beckhoff Automation	▼	Generate File							
License Id:		Customer Id:								
Comment										
License Activa	tion									
7	Days Trial License	Lice	ense Response File							
Order No	License	Instances	License TAN	Current Status	License Id					
TC1200	TC3 PLC	cpu license		expires on Jul 11, 2018 (trial license)	66689887-CCBD-452C-AC9A-039D997C6E6					
TE1300	TC3 Scope View Profe	cpu license		missing	4989A799-AEEA-4FEE-88EF-666B99EEFB45					
TF6420	TC3 Database-Server	cpu license		expires on Jul 11, 2018 (trial license)	92583661-35AE-45CE-BD4F-C35BFE16F07E					
TE6710	TC3 IoT Functions	cou license		expires on Jul 11, 2018 (trial license)	2149932B-0B77-4004-B43E-E85CEEEE347D					

A TwinCAT 3 license is generally linked to two indices describing the platform to be licensed: System ID: Uniquely identifies the device

Platform level: Defines the performance of the device

The corresponding System Id and Platform fields cannot be changed.

 Enter the order number (License Id) for the license to be activated and optionally a separate order number (Customer Id), plus an optional comment for your own purposes (Comment). If you do not know your Beckhoff order number, please contact your Beckhoff sales contact.

winCAT SampleProject 🕆 🗙								
Order Information (Runtime) Manage Licenses Project Licenses Online Licenses								
License Device Target (Hardware Id)								
System Id:		Platf	form:					
E13751F7-79B9-	-4063-416C-615ED53FA7BD	othe	er (90)	•				
License Request								
Provider:	Beckhoff Automation	•	Generate File					
License Id:		Customer Id:						
Comment								
License Activation	n							
7 Da	ays Trial License	Licens	se Response File					
		<b>.</b> .	I. TAN					
Order No	License	Instances	LICENSE TAIN	Current Status				
TC1200	TC3 PLC	cpu license		expires on Jul 11, 2018 (trial license)	66689887-CCBD-452C-AC9A-039D997C6E66			
TE1300	TC3 Scope View Profe	cpu license		missing	4989A799-AEEA-4FEE-88EF-666B99EEFB45			
TF6420	TC3 Database-Server	cpu license		expires on Jul 11, 2018 (trial license)	92583661-35AE-45CE-BD4F-C35BFE16F07E			
TF6710	TC3 IoT Functions	cpu license		expires on Jul 11, 2018 (trial license)	2149932B-0B77-4004-B43F-E85CEEFF347D			

- 8. Click the **Generate File**... button to create a License Request File for the listed missing license.
  - A window opens, in which you can specify where the License Request File is to be stored. (We recommend accepting the default settings.)
- 9. Select a location and click **Save**.
  - A prompt appears asking whether you want to send the License Request File to the Beckhoff license server for verification:

TwinCAT XAE	x
Send license request to Beckhoff	
Yes No	

- Click **Yes** to send the License Request File. A prerequisite is that an email program is installed on your computer and that your computer is connected to the internet. When you click **Yes**, the system automatically generates a draft email containing the License Request File with all the necessary information.
- Click No if your computer does not have an email program installed on it or is not connected to the internet. Copy the License Request File onto a data storage device (e.g. a USB stick) and send the file from a computer with internet access and an email program to the Beckhoff license server (tclicense@beckhoff.com) by email.
- 10. Send the License Request File.
  - ⇒ The License Request File is sent to the Beckhoff license server. After receiving the email, the server compares your license request with the specified order number and returns a License Response File by email. The Beckhoff license server returns the License Response File to the same email address from which the License Request File was sent. The License Response File differs from the License Request File only by a signature that documents the validity of the license file content. You can view the contents of the License Response File with an editor suitable for XML files (e.g. "XML Notepad"). The contents of the License Response File must not be changed, otherwise the license file becomes invalid.
- 11. Save the License Response File.

- 12. To import the license file and activate the license, click License Response File... in the Order Information tab.
- 13. Select the License Response File in your file directory and confirm the dialog.

🕶 Open			×
😋 💽 🗢 🚺 🕨 Computer 🕨	Downloads 👻 🍫	Search Downloads	Q
Organize 🔻 New folder		!≡ ▼ 🗔	0
☆ Favorites	▲ Name		Date m
🧮 Desktop	E LicenseResponseFile.tclrs		21.03.20
🚺 Downloads			
🖳 Recent Places			
Pictures			
🔁 Libraries			
Application			
Documents			
J Music			
Pictures			
🖶 Public	III		
File name	LicenseResponseFile.tclrs 🔹	TwinCAT License Response Fi	le 🔻
		Open	

⇒ The License Response File is imported and the license it contains is activated. Existing demo licenses will be removed.

- 14. Restart the TwinCAT system.
- ⇒ The license becomes active when TwinCAT is restarted. The product can be used as a full version. During the TwinCAT restart the license file is automatically copied to the directory ...\TwinCAT\3.1\Target \License on the respective target system.

# 5 Configuration

The following settings are possible for the EtherNet/IP scanner:

General Adapter EtherNet/IP Sync Task Settings Explicit Msg Diag History DPRAM (Online)

#### General:

Name and TwinCAT ID of the device

#### Adapter:

Setting for the Ethernet interface used

### EtherNet/IP:

Display of the software version and ADS address of the EtherNet/IP scanner

#### Sync Task:

Setting indicating which task triggers the EtherNet/IP scanner and the cycle time with which it operates

#### Settings:

Setting for IP address and other Ethernet-specific services

#### Explicit Msg:

Only required for Data Table Read/Write (see chapter Data Table Read and Write [> 28])

### Diag History:

All errors or notes regarding the EtherNet/IP scanner are logged.

#### DPRAM (online):

Not relevant for the user

### 5.1 EtherNet/IP

SW Version: Display of the driver version used for the EtherNet/IP scanner.

**NetId:** AMSNETID of the EtherNet/IP scanner. This is necessary if the EtherNet/IP-specific function blocks are required.

General Adapter	EtherNet/IP	Sync Task	Settings	Explicit Msg	Diag
SW Version:	01 (V01.00)				
Netld:	5.18.71.214.4	.1		Info Data Supp	port

**Info Data Support:** If this option is activated, the AMSNETID is also available in the TwinCAT tree and can then be linked accordingly.

Device 3 (TC3 EIP Scanner)
 Device 3 (TC3 EIP Scanner)-Image
 Image-Info
 Inputs
 Outputs
 InfoData
 DevId
 AmsNetId

### 5.2 Sync Task

The **Sync Task** starts the cyclic call of the EtherNet/IP driver. The Sync Time should be as short as possible, if the processor power allows this. 1 ms is the smallest time base that can be set. It is recommended to create the **Sync Task** via a **Special Sync Task**. If the Sync Task is performed via the mapping of the PLC, a breakpoint in the PLC also causes the EtherNet/IP Task to be stopped, so that the EtherNet/IP devices are no longer addressed. This results in a connection timeout.

eneral Adapter Et	therNet/IP Sync	Task Settings	Explicit Msg	Diag History	DPRAM (Online
Settings					
Standard (via M	Mapping)				
Special Sync T	ask				
Task 2		•	Cr	eate new I/O T	ask
0 T I					
Sync Task					
Name:	Task 2				
Cycle ticks:	1	•	1.000	m	s
	Adjusta	able by Protocol			
Priority:	1	* *			

Each slave can run with its own cycle time based on the Sync Task. The **Cycle Time Multiplier** setting is available on each device for this purpose. See chapter <u>Connection of EtherNet/IP slaves [> 18]</u>.

### 5.3 Settings dialog

The **Settings** dialog is required for settings such as the IP address and other basic settings. It is divided into two basic settings, which are indicated by the index numbers.

Index 0xF800 contains all the settings used on system startup.

Index 0xF900 contains the actual settings that are valid while the system is running. The actual valid settings are important if basic settings are not made via the **Settings** dialog but have been changed via the PLC.

The IP address is a virtual IP address. In the first step it is unrelated to the IP setting of the operating system (OS). It is recommended to use a different network class than the one selected in the OS. If the IP address of the EtherNet/IP scanner is nevertheless the same as that of the OS, the value 255.255.255.255 should be set under IP address (0xF800:21). (See also Firewall recommendation [ $\blacktriangleright$  17])

eral Adapter	EtherNet/IP Sync Task	Settings	Explicit M	sg Diag History	DPRAM (Or
laster Settings –					
Index	Name		Flags	Value	
<b>⊟ F800:0</b>	Master Settings		MRO	> 43 <	
F800:01	Number		MRO	0x0003 (3)	
F800:03	Product Name		MRW	Device 3 (TC3 E	IP Scanner)
F800:04	Device Type		MRO	0x000C (12)	
F800:05	Vendor ID		MRO	0x006C (108)	
F800:06	Product Code		MRO	0x1889 (6281)	
F800:07	Revision		MRO	3.1	
F800:08	Serial Number		MRO	0x00000000 (0)	
F800:20	MAC Address		MRO	02 01 05 12 47 D	6
F800:21	IP Address		MRW	192.168.1.10	
F800:22	Network Mask		MRW	255.255.255.0	
F800:23	Gateway Address		MRW	0.0.0.0	
F800:24	DHCP Max Retries		MRW	0	
F800:25	TCP/IP TTL		MRW	128	
F800:26	TCP/IP UDP Checksum	1	MRW	TRUE	
F800:27	TCP/IP TCP Timeout		MRW	300 Seconds	
F800:28	MultiCast TTL		MRW	1	
F800:29	MultiCast UDP Checksu	m	MRW	FALSE	
F800:2A	Forward Class3 to PLC		MRW	FALSE	
F800:2B	Advanced Options		MRW	0x0000 (0)	
	Master Info		RO	>43 <	

### Index 0xF800:0 Master Settings

Configuration parameters of the Ethernet/IP scanner

#### Index 0xF800:1 Number Box Id

Index 0xF800:3 Product Name Name of the device

Index 0xF800:4 Device Type Device type

Index 0xF800:5 Vendor ID Vendor number

Index 0xF800:6 Product Code Product code

Index 0xF800:7 Revision Version

Index 0xF800:8 Serial Number Serial number (see object 0xF900)

Index 0xF800:20 MAC Address MAC address (see object 0xF900)

Index 0xF800:21 IP Address Possible values:

- 0: The IP address is assigned dynamically by the DHCP service
- Otherwise: statically assigned IP address.

#### Index 0xF800:22 Network Mask

Possible values:

- 0: The subnet mask is assigned dynamically by the DHCP service
- Otherwise: statically assigned subnet mask.

#### Index 0xF800:23 Gateway Address

Possible values:

- 0: DHCP service is used,
- Otherwise: statically assigned gateway address.

### Index 0xF800:24 DHCP Max Retries

Possible values;

- 0: Continuous repetition of the DHCP addressing attempts.
- Currently only this mode is implemented, as of: 10-2017

#### Index 0xF800:25 TCP/IP TTL

"Time to live" value for unicast TCP/UDP communication

#### Index 0xF800:26 TCP/IP UDP Checksum

function (Unicast) Possible values:

- 0: UDP checksums disabled,
- 1: UDP checksums enabled

### Index 0xF800:27 TCP/IP TCP Timeout

Time switch for inactive TCP connection in seconds

• 0: Time switch disabled

### Index 0xF800:28 MultiCast TTL

"Time to live" value for multicast UDP communication

### Index 0xF800:29 MultiCast UDP Checksum

function (Multicast):

- 0: UDP checksums disabled
- 1: UDP checksums enabled

### Index 0xF800:2A Forward Class3 to PLC

Message forwarding to the PLC Currently not implemented, as of: 10-2017

### Index 0xF800:2B Advanced Slave Options

"Store Category" parameter:

- Bit9=Cat2
- Bit8=Cat1

### Index 0xF900 Scanner Info

The current valid settings are displayed here; these can differ from the object 0xF800.

The object 0xF900 displays the active parameters to you.

### 5.3.1 Firewall setting

The firewall must be enabled, if the EtherNet/IP address is to match the IP address of the operating system (OS). It is advisable to enable the firewall if the IP address of the EtherNet/IP scanner deviates from the IP setting of the operating system.

# 5.4 Diag History

The diagnostic history (**Diag History**) is a tool for monitoring the status of the Ethernet/IP interface and displaying the diagnostic messages with time stamp in plain text.

In addition, information / errors that occurred in the past are logged, in order to enable precise troubleshooting at a later stage. This also applies for errors that only occurred for such a short time that any corresponding messages were not visible.

The diagnostic history is part of the TwinCAT system, where it can be found under Devices, EtherNet/IP in the **Diag History** tab:

General Ac	lapter	EtherNet/IP Sync Ta	sk Settings Explicit Msg Diag History DPRAM (Online)	
Update	e History	y Auto Update	Ack. Messages Export Diag History Advanced	
Туре	Fla	Timestamp	Message	-
🕕 Info	Ν	19.5.2017 16:41:07	(0x8003) Device 3 (TC3 EIP Scanner): FwdOpen-Request sent to 192.168.1.220 ()	
🕕 Info	N	19.5.2017 16:41:06	(0x4001) Device 3 (TC3 EIP Scanner): MSG Connection Open (IN:0 OUT:0 API:7500ms) f	
<ol> <li>Info</li> </ol>	N	19.5.2017 16:41:06	(0x4001) Device 3 (TC3 EIP Scanner): MSG Connection Open (IN:0 OUT:0 API:7500ms) f	
<ol> <li>Info</li> </ol>	N	19.5.2017 16:41:06	(0x4002) Device 3 (TC3 EIP Scanner): MSG Connection Close (IN:0 OUT:0) from 192.168	
<ol> <li>Info</li> </ol>	N	19.5.2017 16:41:06	(0x4002) Device 3 (TC3 EIP Scanner): MSG Connection Close (IN:0 OUT:0) from 192.168	
🕕 Info	N	19.5.2017 16:41:06	(0x8003) Device 3 (TC3 EIP Scanner): FwdOpen-Request sent to 192.168.1.220 ()	
🕕 Info	N	19.5.2017 16:41:06	(0x2002) Device 3 (TC3 EIP Scanner): Network link detected	
🕕 Info	N	19.5.2017 16:41:01	(0x4003) Device 3 (TC3 EIP Scanner): IO Connection (IN:0 OUT:0) with 0.0.0.0 timed out	
<ol> <li>Info</li> </ol>	N	19.5.2017 16:41:01	(0x4003) Device 3 (TC3 EIP Scanner): IO Connection (IN:0 OUT:0) with 0.0.0.0 timed out	
🕕 Info	N	19.5.2017 16:41:00	(0x2001) Device 3 (TC3 EIP Scanner): Network link lost	
🕕 Info	N	19.5.2017 16:40:59	(0x8003) Device 3 (TC3 EIP Scanner): FwdOpen-Request sent to 192.168.1.220 ()	
🕕 Info	N	19.5.2017 16:40:59	(0x8002) Device 3 (TC3 EIP Scanner): Connection with 192.168.1.220 () timed out	
🕕 Info	N	19.5.2017 09:25:15	(0x4001) Device 3 (TC3 EIP Scanner): MSG Connection Open (IN:0 OUT:0 API:7500ms) f	
🕕 Info	N	19.5.2017 09:25:15	(0x4001) Device 3 (TC3 EIP Scanner): MSG Connection Open (IN:0 OUT:0 API:7500ms) f	
<ol> <li>Info</li> </ol>	N	19.5.2017 09:25:04	(0x4001) Device 3 (TC3 EIP Scanner): IO Connection Open (IN:0 OUT:0 API:20ms) from	
🕕 Info	N	19.5.2017 09:25:04	(0x4001) Device 3 (TC3 EIP Scanner): IO Connection Open (IN:0 OUT:0 API:2ms) from 1	
🕕 Info	N	19.5.2017 09:25:04	(0x8001) Device 3 (TC3 EIP Scanner): Connection with 192.168.1.220 () established	
🕕 Info	N	19.5.2017 09:25:04	(0x8003) Device 3 (TC3 EIP Scanner): FwdOpen-Request sent to 192.168.1.220 ()	
🕕 Info	Ν	19.5.2017 09:25:02	(0x2008) Device 3 (TC3 EIP Scanner): TCP handler initialized	
<ol> <li>Info</li> </ol>	N	19.5.2017 09:25:02	(0x2007) Device 3 (TC3 EIP Scanner): UDP handler initialized	-

### 5.5 Connecting EtherNet/IP slaves

An EtherNet/IP slave can be integrated as a generic node with EDS (Electronic Data Sheet), or without an EDS file. Not all EtherNet/IP slaves currently available on the market are supported. It should be possible to integrate Ethernet/IP devices that are delivered with an EDS file via the EDS import, provided they are supported by the TF6281. If this is not the case, you can send the EDS file to Beckhoff Support for verification.

If the EDS file can be integrated without errors, communication to the slave should be possible. If you use a slave that can only be integrated via the generic node (i.e. without an EDS file), it is to be assumed that it should also be usable.

The following slaves cannot be used:

- · Slaves that use CIP Sync, CIP Motion or CIP Safety
- · Slaves with modular EDS file

### Integrating EtherNet/IP slave without EDS file

Slaves that do not use an EDS file, or for which the manufacturer does not provide an EDS file, are integrated via a generic node. The following manufacturer information is required for this purpose:

• IP address of the slave

- Maximum RPI time, i.e. the maximum or minimum time with which the slave can work
- The Assembly Instance Number for config, input and output data and their length
- · Description of the data

Add a generic node under the EtherNet/IP scanner. As long as you have not specified an IP address, the

symbol is identified by a warning and question mark *A*. Enter the **IP address** under **Settings**.

Gei	neral Settings			
l c	Slave Settings			
	Index	Name	Flags	Value
	Ė <sup></sup> 8020:0	Remote Node Settings (Box 7)	MRO	> 43 <
	8020:01	Slave Number	MRO	0x0007 (7)
		IP Address	MRW	0.0.0.0

An "IO Connection" must first be created under the node. This IO Connection contains the inputs and outputs, which can now be created. The variable type is freely selectable, only the size has to match.

4	2	Box 7 (EtherNet/IP Slave (generic))
	Þ	🛁 Inputs
	Þ	Outputs
	4	Connection 5 (Input/Output)
		ᆜ Inputs
		📑 Outputs

Furthermore, the EtherNet/IP specific entries have to be made now.

Connection			
Default Connection (with	out eds)		
General			
Transport Trigger	Cyclic 🗸	Timeout Multiplier	4
Config Instance	0	Config Size	0 Add Config
Port	0	Slot	0
Inputs ( Data Length: 0 Byt	e)	Outputs (Data Length: 0 B	lyte)
Connection Point	0 Run/Idle	Connection Point	0 Run/Idle
Cycle Time Multiplier	10	Cycle Time Multiplier	10
Transport Type	Multicast 🔹	Transport Type	Point to Point
Priority	Scheduled -	Priority	Scheduled

It is sufficient to specify the values for **Config Instance** and **Config Size**. The **Connection Points** must be created for the inputs and outputs.

The data length results from the length you have previously created. You can verify it in this dialog. The **Cycle Time Multiplier** takes the cycle time of the **Sync Task** (-> see <u>Sync Task</u> [ $\blacktriangleright$  <u>15</u>]) and is multiplied by it. Example: If the Sync Task is set to 2 ms and the Cycle Time Multiplier is set to 10, the slave is operated with 20 ms.

### Integration of EtherNet/IP slave with EDS file

TwinCAT offers the option of integrating EDS files. The Import EDS File dialog is used for this purpose.

<ul> <li>Device I (ICS EIF Scattler)</li> <li>Image</li> <li>Inputs</li> <li>Outputs</li> <li>Mappings</li> </ul>	Add New Item Add Existing Item Remove	Ctrl+Shift+A Shift+Alt+A Del
	Import EDS File	
	Change NetId	
	Save Device 1 (TC3 EIP Scann	ner) As
	Online Reset	
	Online Reload	
	Online Delete	
1	Scan	
	Сору	Ctrl+C
*	Cut	Ctrl+X
	Paste	Ctrl+V
	Paste with Links	
	Independent Project File	
-	Disable	
	Change Id	

The files are checked and copied to the directory \TwinCAT\3.1\Config\lo\EtherNetIP after successful import.



EDS files must have an IO connection, otherwise this error message appears:

Microsoft	Visual Studio
<u> </u>	Invalid EDS-File. A valid EDS-File must contain at least one supported Connection, Device- and Assembly Section!
	ОК

These types of devices are not supported by the TF6871 Ethernet/IP scanner.

For EDS files that support symbols, the symbolism is ignored. The symbolism is therefore not usable:



After you have created the slave, the connection must be added. Only the connections described in the EDS file are displayed. Only one connection is allowed.

### 5.6 PLC to PLC communication

### **Consumed and Produced tags**

This type of communication is used for PLC – PLC communication. Data is exchanged in real-time between the two controllers. The data exchange takes place via the so-called Consumed and Produced tags. Tag stands for a variable name. The Consumed tag receives the data. The Produced tag provides the data. This

means that a Produced tag is created on one controller first, the opposite side that is supposed to receive the data "consumes" the data, hence Consumed tag. This type of communication always requires two EtherNet/IP scanners.

In the following paragraph this is explained by means of a TC3 controller (CX2020 in this case) with the function EtherNet/IP Scanner TF6281 and an Allen-Bradley CompactLogix from Rockwell (RSLogix5000 V20.03.00).

Both sides are described here to set up a communication as described above.

Produced Tag Name:TwinCA	T_IN_0	Consumed	d Tag Name:TwinCAT_IN	I_0
TwinCAT	EtherNet/I	<b>D</b> .	Allen-Bradley	
Consumed Tag Name:TwinC	AT_Out_0	Produced	Tag Name:TwinCAT_Ou	t_0

TwinCAT 3.1 Build 4022.x

### ProduceTag in TwinCAT

First, the EtherNet/IP scanner is created in TwinCAT (IP address and further settings can be found in the previous chapter <u>Settings dialog</u> [▶\_15] ). Right-clicking on the EtherNet/IP Scanner opens a dialog. Select **Add New Item...** 

### Then select Producer Object List:

Insert Box		1 - H 11, Make - 11, Mak
Type:	Beckhoff Auto Microscan Sy Parker Hanni PHOENIX CO Miscellaneou Generic E	omation GmbH /stems, fin, DNTACT Deutschland GmbH&Co is therNet/IP Slave <mark>Object List</mark>

A **Producer Object List** is then created below the scanner. This is available only once, even if the data is sent to more than one controller. Right-click on **Producer Object List** and select **Append Producer Connection**.



Now specify the name of the **Connection Tag**. This must be identical to the name of the consumer.

Then define the number and type of data. It is only possible to use DINT or larger variables.

For the further steps, the name **TwinCAT\_IN\_0** and a variable of type **DINT** were selected. To do this, navigate to the outputs of the **Producer Object** and insert a variable of type **DINT**.

Connection Tag	TwinCAT_IN_0
Transport Trigger	Cyclic

Set the Transport Trigger to Cyclic. Other operation modes are currently not supported.

#### **Consumer Tag in TwinCAT**

Next, create a **Consumer Tag**. To do this, create a **Generic EtherNet/IP Slave** in the EtherNet/IP Scanner. It requires the IP address of the Allen-Breadley CPU. Enter the address and add an **Append Consumer Connection** Consumer tag under the newly created slave. The name is important because it must later be specified as a Produced variable in the Allen-Breadley CPU.

The Port is the CPU port on which the variable will be used later. Usually this is 1.

General Settings			
Connection Tag			
TwinCAT_Out_0			
Conorol			
General			
Transport Trigger	Cyclic 🗸	Timeout Multiplier	4
Config Instance	0	Config Size	0 Add Config
Port	1	Slot	0
Inputs ( Data Length: 4 Byt	e)	Outputs (Data Length: 0 B	yte)
Connection Point	0 Run/Idle	Connection Point	0 Run/Idle
Cycle Time Multiplier	2	Cycle Time Multiplier	1
Transport Type	Point to Point	Transport Type	Point to Point 🔹
Priority	Scheduled	Priority	Low

Now you have created a producer in the TwinCAT tree and a consumer for the other EtherNet/IP controller.

4	+	Device 3 (TC3 EIP Scanner)
		Device 3 (TC3 EIP Scanner)-Image
	⊳	🕒 Inputs
	⊳	唱 Outputs
	4	2 Box 5 (Producer Object List)
		👂 🛄 Inputs
		Outputs
		Producer 2 (Output)
	4	🔜 Box 6 (EtherNet/IP Slave (generic))
		Inputs
		Outputs

Consumer 3 (Input)

### 5.6.1 Allen-Bradley CompactLogix

In order to enable PLC – PLC communication using the Consume and Produce tags, an EtherNet/IP controller must be installed at Allen-Bradley (AB). It is not possible to use a Beckhoff controller with AB, therefore an Allen-Bradley controller must be created in the configuration tool.

Click on Ethernet then right-click to create a new module. Select New Module...

### 🗄 🔄 I/O Configuration

Backplane, CompactLogix System 1769-L32E CPU\_Rockwell 1769-L32E Ethernet Port LocalENB Lthernet 1773 New Module... Discover Modules... Paste Ctrl+V Print ETHERNET MODUCE COL

Then select a controller, for example 1756-EN2T.

Enter Search Text for Module T	Clear Filters			Show Filters 🗧
Catalog Number	Description	Vendor	Category	4
1734-AENTR	1734 Ethemet Adapter, 2-Port, Twisted Pair Media	Allen-Bradley	Communication	
1738-AENT	1738 Ethernet Adapter, Twisted-Pair Media	Allen-Bradley	Communication	
1738-AENTR	1738 Ethernet Adapter, 2-Port, Twisted Pair Media	Allen-Bradley	Communication	
1747-AENTR	1747 Ethernet Adapter, 2-Port, Twisted-Pair Media	Allen-Bradley	Communication	
1753-L28BBBx GPLC 1600	GuardPLC 1600	Allen-Bradley	Programmable Logic Controller	
1753-L32BBBx-8A GPLC	GuardPLC 1800	Allen-Bradley	Programmable Logic Controller	
1756-EN2F	1756 10/100 Mbps Ethernet Bridge, Fiber Media	Allen-Bradley	Communication	
1756-EN2T	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley	Communication	
1756-EN2TR	1756 10/100 Mbps Ethernet Bridge, 2-Port, Twisted-P	Allen-Bradley	Communication	-
1756-EN2TSC	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Medi	Allen-Bradley	Communication	
1756-EN3TR	1756 10/100 Mbps Ethernet Bridge, 2-Port, Twisted-P	Allen-Bradley	Communication	
1756-ENBT	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley	Communication	
1756-ENET	1756 Ethernet Communication Interface	Allen-Bradley	Communication	
1756-EWEB	1756 10/100 Mbps Ethernet Bridge w/Enhanced Web	Allen-Bradley	Communication	
1767 EELD	1757 Foundation Fieldhus Linking Davias	Allon Desidlay	Communication	

۲

Now enter the IP address of the Beckhoff controller or the IP address of the Beckhoff EtherNet/IP Scanner. In addition, the controller requires a name. Select **Disable Keying** under **Module Definition**.

🔝 New Module							<b>-X</b> -
General <sup>×</sup> Com	antinu bladu	la lufa – lutamat Brataa	al Deat Configure	-tion Time Come			
Conn	ection   Modu	lie Inio   Intelnet Plotoci	or proteconligur	ation   Time Sync			_
Type:	1756-EN2T	1756 10/100 Mbps Ethe	met Bridge, Twis	ted-Pair Media		Change Type	÷
Vendor:	Allen-Bradley	, ,		- Ethernet Address			_
Parent:							1
Name:	TC3_PLC			Private Netw	ork: 192	2.168.1. 123	
Description:			*	IP Address:			
			Ŧ	O Host Name:			
Module Defin	ition						
		Char	nge	Slot:	0 🔻		
Revision:		5.1					
Electronic Ke	ying:	Disable Keying					
Rack Connel	ction:	None					
Time Sync C	onnection:	None					
Chassis Size	5.	17					
Status: Creating					OK	Cancel	Help

Now you have to create a PLC. Select 1756-L61, for example, and click Create:

a-Un-Delinea								
edefined	Select Module Type							
odule-Defined								
5	Catalog Module Discovery Fay	orites						
nfiguration								
ckplane. CompactLogix System								
1769-L32E CPU Rockwell	Enter Search Text for Module Type Clear Filters Show Filters							
1769-L32E Ethernet Port LocalENB								
- 🖧 Ethernet	Catalog Number	Description	Vendor	Category		*		
1734-AENT/A ABTest	1756-IT6I2	6 Channel Isolated Thermocouple Analog Input - Enha	Allen-Bradley	Analog				
1756-EN2F TwinCAT	1756-IV16	16 Point 10V-30V DC Input, Source	Allen-Bradley	Digital				
- 🗖 1756 Backplane, 1756-A17	1756-IV32	32 Point 10V-30V DC Input, Source	Allen-Bradley	Digital				
	1756-L1	ControlLogix5550 Controller	Allen-Bradley	Controller				
f [111756-161 Cou	1756-L53	ControlLogix5553 Controller	Allen-Bradley	Controller				
FTHERNET-MODULE test1	1756-L55	ControlLogix5555 Controller	Allen-Bradley	Controller		=		
1 1756-EN2T TC3 PLC	1756-L60M03SE	ControlLogix5560M03SE Controller	Allen-Bradley	Controller				
1756 Backplane 1756-017	1756-L61	ControlLogix5561 Controller	Allen-Bradley	Controller				
1750 Backplanc, 1750 AL7 1760-L32E Ethernet Port LocalENB	1756-L61S	ControlLogix5561S Safety Controller	Allen-Bradley	Controller				
CompactBur Local	1756-L62	ControlLogix5562 Controller	Allen-Bradley	Controller				
	1756-L62S	ControlLogix5562S Safety Controller	Allen-Bradley	Controller				
[] [1] 1709-IF4AOF2/A DI	1756-L63	ControlLogix5563 Controller	Allen-Bradley	Controller				
[] [2] 1709-1Q0XOW4/B Kelay	1756-L63S	ControlLogix5563S Safety Controller	Allen-Bradley	Controller				
17	1756-L64	ControlLogix5564 Controller	Allen-Bradley	Controller		-		
	1750105	Controll anivEECE Controllor	Allon Prodley	Controllor				
	117 of 117 Module Types Fou	ind				Add to Favorites		
	Close on Create				Create	Close Help		

Enter a name for the controller, e.g. **CPU\_2**; this name is still needed later when you create the Consumed tags.

Insert a new DINT variable under Controller Tags. Create it as type **Consumed**:

💕 Tag Properti	es - TwinCAT	
General		
Name:	TwinCAT	
Description:	~	
Туре:	Consumed Connection	
Alias For:	Base Alias	
Data Type:	Consumed	
Scope:	CPU_Rockwell	
External Access:	Read/Write	
Style:	Decimal	
Constant		
	OK Cancel Apply	Help

Now click **Connection**. Select the controller from which you want to receive the data. This requires the name that was assigned during configuration (in this example **CPU\_2**). Furthermore, the tag name, which was also assigned in the TwinCAT controller (here: **TwinCAT\_IN\_0**) and the RPI time are required. The RPI time should always be greater than or equal to the SyncTask of the EtherNet/IP Scanner in TwinCAT:

Consumed Tag Connection	×
Connection Status	
Producer: CPU_2	•
Remote Data: TwinCAT_IN_0	
(Tag Name or Instance Number)	
RPI: 2.0 🚔 ms	
☑ Use Unicast Connection over EtherNet/IP	
	OK Cancel Help

Now insert another **DINT** variable and configure it as **Produced**. It is only important to use the same name as in TwinCAT for the Consumed connection (here **TwinCAT\_Out\_0**).

👸 Tag Propertie	es - TwinCAT_Out_0	- • •
General		
Name	Tui-CAT Oct 0	
Name.	TWINCAT_OUT_0	
Description:	*	
	Ψ	
Туре:	Produced   Connection	
Alias For:		
	▼	
Data Type:	DINT	
Scope:	CPU_Rockwell	
Eternel		
Access:	Read/Write 🔹	
Style:	Decimal 🔹	
Constant		
	OK Cancel Apply	Help

### 5.7 Data Table Read and Write

### Please note the system requirements

Data Table Read and Write can only be used with the TC1200.

Like the Consumed and Produced tag, this function is used for communication between two EtherNet/IP controllers, with the difference that it is an acyclic communication. This enables data to be exchanged between two controllers which do not have to be transmitted cyclically, such as parameters, recipes or any other data. The data can be structures, arrays or a combination of both. TwinCAT enables data to be read from and written to a controller, and it is also possible to read or write data from TwinCAT using remote control. This is explained below by way of example:

Data that is to be sent or received via this service must be made known in the TwinCAT system. This data must be stored as a global variable in a folder ETHIP and in the flag area. The library Tc2\_EthernetIP must also be included. It contains a function block for the DataTable read/write. The data types must match in both PLCs.

### Creating the variables:

Create a global variable list with the name ETHIP. Now add two variables as shown in the image below. The variables must have a fixed address and lie within the flag area (%MBx, x address). For non-located variables, the internal address could change during an online change; such variables are currently not supported. Now compile the project and switch to the EtherNet/IP scanner.



Open the Explicit MSG tab:

						1	
General	Adapter	EtherNet/IP	Sync Task	Settings	Explicit Msg	<b>Diag History</b>	DPRAM (Online)

Move the mouse over the empty box, right-click and select **Add** to add the data:

General	Adapter	EtherNet/IP	Sync Task	Settings	Explicit Msg	Diag History	DPRAM (Online)	
	Symbolna	me			Msgr	name		Re
					Expe	ertMode		
					Abo	ut		
					Writ	e via ADS		
					Rese	et		
					Rem	ove		_
			No	moccaa	Add		, to add new m	0000000
			- NC	message	s conngureu.	ose context ii	iena lo adu new n	lessayes



The dialog Add Symbols ... appears Tick the data you want to use later:

Add Symbols	×
ETHIP.Test	
Cancel	Add

The data are now available in the dialog.

(	Genera	al Adapter EtherNet/IP Sync Task Settings Expl	icit Msg Diag History DPRAM (Online)		
		Symbolname	Msgname	Read	Write
	•	ETHIP.Test	Test	-	<ul> <li>Image: A start of the start of</li></ul>
		ETHIP.Test2	Test2	<b>v</b>	<ul> <li>Image: A set of the set of the</li></ul>

Next, recompile and restart the TwinCAT project. This is necessary if you change the data, e.g. the name, flag, address, type of variable, etc.

### Read a TwinCAT variable from the Allen-Bradley controller

First, enter the TwinCAT controller in the configuration, as for the Consumed and Produced tags; proceed in the same way.

Under **Controller Tags** enter variables **Test** and **iTest**, both as DINT. Now some code has to be written for the Allen-Bradley (AB) controller.

msg(msgTest); (\* Program language: Structured Text \*)

"msgTest "must be of type **MESSAGE**.

New Tag		<b>—</b> ×
Name:	msgTest	Create 🛛 🔻
Description:	A	Cancel
		Help
	Ψ.	
Usage:	<normal></normal>	
Туре:	Base   Connection	
Alias For:	· · · · · · · · · · · · · · · · · · ·	
Data Type:	MESSAGE	
Scope:	CPU_Rockwell 🗸 🗸	
External Access:	Read/Write 💌	
Style:	•	
Constant		
🔽 Open MES	SAGE Configuration	

Then click on the **msgTest** variable and configure the function block.

msg (msgT		Edit "msaTest" Properties	Alt+Enter
		Co <u>n</u> figure "msgTest"	
		<u>F</u> ind All "msgTest"	
		Go to Cross Reference for "msgTest"	Ctrl+E
		Monitor "msgTest"	
		T <u>r</u> end Tag "msgTest"	
		Browse Tag <u>s</u>	Ctrl+Space
	Ж	Cu <u>t</u>	Ctrl+X
	Ē	<u>С</u> ору	Ctrl+C
	Ê	<u>P</u> aste	Ctrl+V
		<u>G</u> o To	Ctrl+G
	•	<u>W</u> atch Tags	Alt+3

Set the message type to **CIP Data Table Read**. Under **Source Element** enter the name that you used in the TwinCAT project.

I	Message Configuration	- msgTest				×
	Configuration* Commun	nication Tag				
	Message Type:	CIP Data Ta	ble Read	•		
	Source Element:	Test				
	Number Of Elements:	1				
	Destination Element:	iTest		•	(	New Tag
	Enable	Waiting	Start	O Done	Done Length:	0
	O Error Code:	Extended	Error Code:		Timed Out	÷
	Error Path: Error Text:					
			ОК	Cancel	Apply	Help

Then open the **Communication** tab. Here you set the controller from which you want to read the variable **Test**.

Message Configuration - msgTest
Configuration* Communication Tag
Path: TwinCAT     Browse
TwinCAT  Broadcast:
Communication Method CIP DH+ Channel: 'A' Destination Link: 0 (Octal) CIP With Source Link: 0 Destination Node: 0 (Octal)
Source ID     Connected       Connected     Cache Connections
Enable
<ul> <li>◯ Error Code: Extended Error Code: □ Timed Out ◆</li> <li>Error Path:</li> <li>Error Text:</li> </ul>
OK Cancel Apply Help

Everything is now prepared for reading the variable.

The variable **Test** is read (on the Beckhoff side) and copied (on the AB side) to the variable **iTest**.

### Writing a TwinCAT variable from the Allen-Bradley controller

A similar procedure must be followed when writing. In this case, the MSG command must describe the Data Table Write. The source element is the variable in the Allen-Bradley controller. The **Destination Element** is the TwinCAT variable. Again, select the TwinCAT controller under **Communication**.

Message Configuration	- msgTestWrite		<b>X</b>
Configuration Communi	ication Tag		
Message Type:	CIP Data Table Write	•	]
Source Element:	Test	•	New Tag
Number Of Elements:	1		
Destination Element:	Test2		
Enable	Waiting 💿 Start	O Done	Done Length: 0
C Error Code:	Extended Error Code:		🔲 Timed Out 🗧
Error Path: Error Text:			
	ОК	Cancel	Apply Help

The variable **Test** (on the AB side) is copied to the variable **Test2** (on the Beckhoff side).

### Transferring STRING variables

On the Rockwell controller, STRINGS have a different data format than on the TwinCAT controller. The library Tc2\_EthernetIP features a data structure called **RSL5K\_STRING**, which facilitates the use of STRINGS. You must use this in order to be able to use STRINGS. The corresponding conversions are also available in the library. Only STRINGS with 82 characters or less may be used.

#### Data Table READ/WRITE from the Beckhoff controller

The PLC function block <u>FB CIP DATA TABLERDWR [> 40]</u> is used for DataTableRead/Write from the library Tc2\_EthernetIP (see DataTableRDWR). The usage is very similar to that of the AB controller and is shown here as an example:



As shown in the image above, a [\*] placeholder can also be used with ARRAYs on the TwinCAT side. To this end, the ARRAY value is entered with an \* in the variable name. The advantage is that only parts or just one element of an ARRAY is read or written. In other words, it is not necessary to read or write the complete ARRAY.

If you use an ARRAY in an ARRAY with \* in each case, the index is entered for all [\*] values. Example **DataARRAY**[\*].**ValueArray**[\*]: the index is entered for both.

### 5.8 Diagnostics

There are several diagnostic options for EtherNet/IP. The diagnosis is divided into two areas, i.e. diagnosis for the scanner (master), and diagnosis for the adapters (slaves) that are connected to the scanner. These are cyclic diagnostic data which can be linked to the PLC. A further diagnosis is available via DiagHistory. Errors in the EtherNet/IP system are logged and can be evaluated for diagnostic purposes.

### Diagnosis of the master (scanner)

The scanner diagnosis contains information about the status of the EtherNet/IP scanner. If the value is 0x0000, everything is OK and there is no error.



Values that the DevState can take:

0x0001 = Link error

```
0x0020 = Watchdog triggered
```

```
0x8000 = reserved
```

```
0x4000 = Master has no valid IP Addr - pending DHCP request
```

0x2000 = TCP server: unable to listen on local EtherNet/IP Port (44818)

<sup>0</sup>x0010 = Out of send resources (I/O reset required)

<sup>0</sup>x1000 = UDP server: unable to listen on local EtherNet/IP Port (44818)

### Diagnosis of the slave (adapter)

Each slave has a state and a Ctrl word.



The Ctrl word currently has no purpose. In an error-free state, the value of the state is 0x0000. The state has the following meaning:

- 0x8000 = Remote Node has no connections
- 0x4000 = Remote Node is not reachable
- 0x2000 = TCP Client: initialization failed
- 0x1000 = UDP Client: initialization failed
- 0x0X00 = reserved
- 0x0001 = 1st Connection disconnected
- 0x0002 = 2nd Connection disconnected
- 0x0004 = 3rd Connection disconnected

0x0080 = 8th Connection disconnected

#### **Producer State**

0x8000 = Producer has no valid Producer Objects configured

- 0x4000 = Producer has no valid IP Addr pending DHCP request
- 0x2000 = TCP server: unable to listen on local EtherNet/IP Port (44818)
- 0x1000 = UDP server: unable to listen on local EtherNet/IP Port (44818)
- 0x0001 = 1st Connection disconnected
- 0x0002 = 2nd Connection disconnected
- 0x0004 = 3rd Connection disconnected

0x0800 = 12th Connection disconnected

#### **Consumer State**

- 0x0X00 = reserved
- 0x0001 = 1st Connection disconnected
- 0x0002 = 2nd Connection disconnected
- 0x0004 = 3rd Connection disconnected

0x0800 = 12th Connection disconnected

# 6 PLC API

The TwinCAT function blocks can only be used in conjunction with the TC1200. The library Tc2\_EthernetIP can be found under **Communication**. It is part of the TC1200 TwinCAT installation.

### 6.1 Function blocks

### 6.1.1 FB\_GET\_ATTRIBUTE\_SINGLE

	FB_GET_ATTRIBUTE_SINGLE		
_	sNetId T_AmsNetID	BOOL	bBusy -
_	sIPv4Addr T_IPv4Addr	BOOL	bError-
_	bExecute BOOL	UDINT	nErrId
_	nClass WORD	WORD nD	ataLen -
_	nInstance WORD		
_	nAttribute WORD		
_	pDst POINTER TO BYTE		
_	nMaxLen WORD		
_	nSessionTimeoutMSec DWORD		
—	nCmdTimeoutMSec DWORD		
—	bRackComm BOOL		
—	nPort BYTE		
_	nSlot BYTE		

The function block FB\_GET\_ATTRIBUTE\_SINGLE enables reading of parameters from an EtherNet/IP device.

Service code: 0x0E

### VAR\_INPUT

VAR	INPUT		
-	sNetId	:	T AmsNetID;
	sIPv4Addr	:	T IPv4Addr;
	bExecute	:	BOOL;
	nClass	:	WORD;
	nInstance	:	WORD;
	nAttribute	:	WORD;
	pDst	:	POINTER TO BYTE;
	nMaxLen	:	WORD;
	nSessionTimeoutMSec	:	DWORD;
	nCmdTimeoutMSec	:	DWORD;
	bRackComm	:	BOOL;
	nPort	:	BYTE;
	nSlot	:	BYTE;
END	VAR		

sNetId: AMSNetId of the TwinCAT EtherNet/IP scanner through which the command is to run

sIPv4Addr: IP address of the target device

bExecute: A positive edge starts the command

nClasss: Class number of the CIP service

ninstance: Instance number of the CIP service

nAttribut: Attribute number of CIP service

pDst: Pointer to the variable to which the value is be copied (the pointer is determined with ADR)

nMaxLen: Size of the variable to which the pointer pDst points (determined with SizeOf)

nSessionTimeoutMSec: Timeout for the session; the default is 30 seconds

**nCmdTimeoutMSec:** Timeout for the command; the default is 7.5 seconds **bRackComm:** TRUE if the CPU is modular, i.e. a CPU with a rack design, for example a CompactLogix

nPort: Port number of the CPU (the TF6281 currently only supports port 1)

nSlot: Slot number if the CPU is not plugged into slot 0

#### VAR\_OUTPUT

```
VAR_OUTPUT

bBusy : BOOL;

bError : BOOL;

nErrId : UDINT;

nDataLen : WORD;

END VAR
```

**bBusy**: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

**bError**: If an error should occur during the transfer of the command, then this output is set once the bBusy output was reset.

nErrld: If an bError output is set, this parameter supplies an error number.

nDataLen: Returns the number of valid data (number of bytes).

### 6.1.2 FB\_SET\_ATTRIBUTE\_SINGLE

FB_SET_ATTRIBUTE_SINGLE		
 sNetId T_AmsNetID	BOOL	bBusy -
 sIPv4Addr T_IPv4Addr	BOOL	bError-
 bExecute BOOL	UDINT	nErrId-
 nClass WORD		
 nInstance WORD		
 nAttribute WORD		
 pSrc POINTER TO BYTE		
 nSrcDataLen WORD		
 nSessionTimeoutMSec DWORD		
 nCmdTimeoutMSec DWORD		
 bRackComm BOOL		
 nPort BYTE		
 nSlot BYTE		

The function block FB\_SET\_ATTRIBUTE\_SINGLE enables writing of parameters in an EtherNet/IP device.

Service code: 0x10

#### VAR\_INPUT

VAR_	INPUT		
	sNetId	:	T AmsNetID;
	sIPv4Addr	:	T IPv4Addr;
	bExecute	:	BOOL;
	nClass	:	WORD;
	nInstance	:	WORD;
	nAttribute	:	WORD;
	pSrc	:	POINTER TO BYTE;
	nSrcDataLen	:	WORD;
	nSessionTimeoutMSec	:	DWORD;
	nCmdTimeoutMSec	:	DWORD;



	bRackComm	:	BOOL;
	nPort	:	BYTE;
	nSlot	:	BYTE;
END	VAR		

**sNetId**: AMSNetId of the TwinCAT EtherNet/IP scanner through which the command is to run **sIPv4Addr:** IP address of the target device

**bExecute**: A positive edge starts the command **nClasss:** Class number of the CIP service

**ninstance:** Instance number of the CIP service

nAttribut: Attribute number of CIP service

**pSrc**: Pointer to the variable containing the value for sending the service (the pointer is determined with ADR)

nSrcDataLen: Size of the variable to which the pointer pSrc points (determined with SizeOf) nSessionTimeoutMSec: Timeout for the session; the default is 30 seconds nCmdTimeoutMSec: Timeout for the command; the default is 7.5 seconds bRackComm: TRUE if the CPU is modular, i.e. a CPU with a rack design, for example a CompactLogix nPort: Port number of the CPU (the TF6281 currently only supports port 1) nSlot: Slot number if the CPU is not plugged into slot 0

### VAR\_OUTPUT

VAR\_OUTPUT bBusy : BOOL; bError : BOOL; nErrId : UDINT; END\_VAR

**bBusy**: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

**bError**: If an error should occur during the transfer of the command, then this output is set once the bBusy output was reset.

nErrld: If an bError output is set, this parameter supplies an error number.

### 6.1.3 FB\_CUSTOM\_SERVICE

FB_CUSTOM_SERVICE	
 sNetId T_AmsNetID	BOOL bBusy
 sIPv4Addr T_IPv4Addr	BOOL bError
 bExecute BOOL	UDINT nErrId
 nServiceCode BYTE	WORD nDataLen
 nClass WORD	
 nInstance WORD	
 nAttribute WORD	
 pDst POINTER TO BYTE	
 nMaxLen WORD	
 pSrc POINTER TO BYTE	
 nSrcDataLen WORD	
 nSessionTimeoutMSec DWORD	
 nCmdTimeoutMSec DWORD	
 bRackComm BOOL	
 nPort BYTE	
 nSlot BYTE	

The function block FB\_CUSTOM\_SERVICE enables virtually any CIP services to be created by the user.

### VAR\_INPUT

VAR	INPUT		
	sNetId	:	T AmsNetID;
	sIPv4Addr	:	T IPv4Addr;
	bExecute	:	BOOL;
	nServiceCode	:	BYTE
	nClass	:	WORD;
	nInstance	:	WORD;
	nAttribute	:	WORD;
	pDst	:	POINTER TO BYTE;
	nMaxLen	:	WORD;
	pSrc	:	POINTER TO BYTE;
	nSrcDataLen	:	WORD;
	nSessionTimeoutMSec	:	DWORD;
	nCmdTimeoutMSec	:	DWORD;
	bRackComm	:	BOOL;
	nPort	:	BYTE;
	nSlot	:	BYTE;
END	VAR		

sNetId: AMSNetId of the TwinCAT EtherNet/IP scanner through which the command is to run

sIPv4Addr: IP address of the target device

**bExecute**: A positive edge starts the command

nServiceCode: Service code of the CIP service

nClasss: Class number of the CIP service

nInstance: Instance number of the CIP service

nAttribut: Attribute number of CIP service

pDst: Pointer to the variable to which the value is be copied (the pointer is determined with ADR)

nMaxLen: Size of the variable to which the pointer pDst points (determined with SizeOf)

**pSrc**: Pointer to the variable containing the value for sending the service (the pointer is determined with ADR)

**nSrcDataLen**: Size of the variable to which the pointer pSrc points (determined with SizeOf), or the number of bytes to be sent. Usually this is the size of the variable.

nCmdTimeoutMSec: Timeout for the command; the default is 7.5 seconds

bRackComm: TRUE if the CPU is modular, i.e. a CPU with a rack design, for example a CompactLogix

**nPort**: Port number of the CPU (the TF6281 currently only supports port 1)

**nSlot**: Slot number if the CPU is not plugged into slot 0

### VAR\_OUTPUT

```
VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
nDataLen : WORD;
END VAR
```

**bBusy**: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

**bError**: If an error should occur during the transfer of the command, then this output is set once the bBusy output was reset.

 $\ensuremath{\textbf{nErrld}}\xspace$  : If an bError output is set, this parameter supplies an error number.

**nDataLen**: Returns the number of valid data (number of bytes)

### 6.1.4 FB\_CIP\_DATA\_TABLE\_RDWR

	FB_CIP_DATA_TABLE_RDWR		
	sNetId T_AmsNetID	BOOL	bBusy -
	sIPv4Addr T_IPv4Addr	BOOL	bError -
	bExecute BOOL	UDINT	nErrId -
	bDataTableWrite BOOL		
	sSrcElementName STRING(82)		
	sDstElementName STRING(82)		
	nNumberOfElements INT		
	nLocalIndex INT		
	nRemoteIndex INT		
	nSessionTimeoutMSec DWORD		
	nCmdTimeoutMSec DWORD		
	bRackComm BOOL		
_	nPort BYTE		
_	nSlot BYTE		

Variables are read and written from TwinCAT via a function block that is part of the Tc2\_EthernetIP.

The function block FB\_CIP\_DATA\_TABLE\_RDWR can be used for reading and writing.

### VAR\_INPUT

VAR	INPUT		
	sNetId	:	T_AmsNetID;
	sIPv4Addr		T_IPv4Addr;
	bExecute	:	BOOL;
	bDataTableWrite	:	BOOL;
	sSrcElementName	:	WORD;
	sDstElementName	:	WORD;
	nNumberOfElements	:	POINTER TO BYTE;
	nLocalIndex	:	WORD;
	nRemoteIndex	:	DWORD;
	nSessionTimeoutMSec	:	DWORD
	nCmdTimeoutMSec	:	DWORD;
	bRackComm	:	BOOL;
	nPort	:	BYTE;
	nSlot	:	BYTE;

END VAR

sNetId: AMSNetId of the TwinCAT EtherNet/IP scanner through which the command is to run

sIPv4Addr: IP address of the target CPU

bExecute. A positive edge starts the command

bDataTableWrite: FALSE triggers a DataTableRead, TRUE a DataTableWrite

sSrcElementName: String for the source name

sDstElementName: String for the target name

nNumberOfElements: Number of elements

**nLocalIndex:** For ARRAYs the start index has to be set to indicate from which ARRAY index the data should be taken (local system)

**nRemoteIndex:** For ARRAYs the start index has to be set to indicate from which ARRAY index the data should be taken (remote system)

**nSessionTimeoutMSec**: Timeout for the session; the default is 30 seconds **nCmdTimeoutMSec**: Timeout for the command; the default is 7.5 seconds

**bRackComm**: TRUE if the CPU is modular, i.e. a CPU with a rack design, for example a CompactLogix

nPort: Port number of the CPU (usually 1)

nSlot: Slot number if the CPU is not plugged into slot 0

#### VAR\_OUTPUT

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

**bBusy**: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

**bError**: If an error should occur during the transfer of the command, then this output is set once the bBusy output was reset.

nErrld: If an bError output is set, this parameter supplies an error number.

Example

Removing test code

If you have already tested the communication from AB to Beckhoff, you should remove the function calls to DataTable Read/Write from the AB project

```
VAR
    FB CIP DATA TABLE RDWR: FB CIP DATA TABLE RDWR;
    SourceName: STRING := 'Test';
    DestName: STRING := 'ETHIP.Test';
END VAR
FB CIP DATA TABLE_RDWR(
  sNetId:='5.18.71.214.4.1'
  sIPv4Addr:='192.168.1.220',
  bExecute:=TRUE ,
  bDataTableWrite:=
  sSrcElementName:=(SourceName) ,
  sDstElementName:=(DestName) ,
  nNumberOfElements:=1 ,
 nLocalIndex:= ,
 nRemoteIndex:=
  nSessionTimeoutMSec:= ,
  nCmdTimeoutMSec:= ,
  bRackComm:=TRUE ,
  nPort:= ,
  nSlot:= ,
  bBusy=> ,
  bError=>
 nErrId=> );
IF NOT FB_CIP_DATA_TABLE_RDWR.bBusy THEN
  FB CIP DATA TABLE RDWR (bExecute:=FALSE);
  Error:=F GET ETHERNETIP ERROR HELPSTRING (FB CIP DATA TABLE RDWR.nErrId;
END IF
```

### 6.2 Functions

### 6.2.1 RSL5KSTRING\_TO\_STRING

The function converts an <u>RSL5KString value</u> [42] to a string value.

### FUNCTION RSL5KSTRING\_TO\_STRING : STRING(82)

### VAR\_INPUT

```
VAR_INPUT
in : RSL5K_STRING;
END VAR
```

### 6.2.2 STRING\_TO\_RSL5KSTRING



The function converts an <u>RSL5KString value</u> [ 42] to a string value

FUNCTION STRING\_TO\_RSL5KSTRING: RSL5K\_STRING

# VAR\_INPUT

```
in : STRING(82);
END_VAR
```

### 6.2.3 F\_GET\_ETHERNETIP\_ERROR\_TEXT

```
nErrorId UDINT F_GET_ETHERNETIP_ERROR_TEXT
```

This function returns a descriptive text based on an error number.

```
See list of <u>TF6281 error codes</u> [▶ <u>44</u>]
```

### FUNCTION F\_GET\_ETHERNETIP\_ERROR\_TEXT: STRING(80)

### VAR\_INPUT

```
VAR_INPUT
nErrorId : UDINT;
END VAR
```

### 6.3 Data types

### 6.3.1 RSL5K\_STRING

```
TYPE RSL5K_STRING
STRUCT
LENGTH : DINT;
DATA : ARRAY [0..81] OF SINT
END_STRUCT
END TYPE
```

Length: Length of char characters contained in the data (max. 82)

Data: Chat characters

# 7 Appendix

### 7.1 Prepare Wireshark recording

The Wireshark recording can be created with a network hub, a network switch with port mirroring, e.g. the <u>Beckhoff ET2000</u>, or with the **Promiscuous Mode** of the TwinCAT system. In **Promiscuous mode**, it can happen that the telegrams are not recorded in the correct order, depending on the system performance and traffic. It is recommended to use an ET2000 for the recording.

General	Adapter	Protocol	Sync Task	Diag History	DPRAM (C	Online)
0	Vetwork Ad	apter				
		۲	OS (NDIS)	O PC	I	O DPRAM
Des	scription:	L	AN-Verbindu	ng (Intel(R) Eth	ernet Conne	ection I218-LM - VirtualBox Bric
Dev	vice Name:	\[	\DEVICE\{C706CD25-DCCF-42A7-B4B7-81D7E66BD979}			
PC	Bus/Slot:					Search
MA	C Address:	е	c f4 bb 1f 7e 8	38		Compatible Devices
IP A	ddress:	1	69.254.254.51	(255.255.0.0)		
			Promiscuou	us Mode (use v	vith Wiresha	rk only)
			Virtual Devi	ce Names		
Adapter Reference						
Ada	apter:					•
Freerun Cycle (ms):		s): 4	•			

### 7.2 Error Codes TF6281

Error	Code hex / (deci- mal)	Description	Remedy/meaning
CN_ORC_ALREADY_USED	0x100 / (256)	Connection already in use	The connection is already established; use another connection or close this one.
CN_ORC_BAD_TRANSPORT	0x103 / (259)	Transport type not supported	The transport type is not supported
CN_ORC_OWNER_CONFLICT	0x106 / (262)	More than one guy configuring	A connection already exists; a further connection cannot be established
CN_ORC_BAD_CONNECTION	0x107 / (263)	Trying to close inactive connection	Faulty connection
CN_ORC_BAD_CONN_TYPE	0x108 / (264)	Unsupported connection type	The connection type is not supported; check your setting.
CN_ORC_BAD_CONN_SIZE	0x109 / (265)	Connection size mismatch	The connection size does not fit; check your setting.
CN_ORC_CONN_UNCONFIGURE	0x110 / (272)	Connection unconfigured	Connection was not configured
CN_ORC_BAD_RPI	0x111 / (273)	Unsupportable RPI	The task time usually doesn't match; make sure that the EL6652 operates internally with 1 ms and that you can adjust this with the Cycle Time Multiplier. Otherwise, adjust the task time.
CN_ORC_NO_CM_RESOURCES	0x113 / (275)	Conn Mgr out of connections	No further resources are available
CN_ORC_BAD_VENDOR_PRODU CT	0x114 / (276)	Mismatch in electronic key	Incorrect manufacturer number
CN_ORC_BAD_DEVICE_TYPE	0x115 / (277)	Mismatch in electronic key	Incorrect device type
CN_ORC_BAD_REVISION	0x116 / (278)	Mismatch in electronic key	Incorrect revision number
CN_ORC_BAD_CONN_POINT	0x117 / (279)	Non-existent instance number	Incorrect connection number
CN_ORC_BAD_CONFIGURATION	0x118 / (280)	Bad config instance number	Faulty configuration
CN_ORC_CONN_REQ_FAILS	0x119 / (281)	No controlling connection open	The connection could not be established
CN_ORC_NO_APP_RESOURCES	0x11A / (282)	App out of connections	No further free connections available.

If you cannot fix this error yourself, Support will require the following information:

- TwinCAT version and build number and a
- · Wireshark recording

### 7.3 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

### Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for <u>local support and service</u> on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages:

http://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

#### **Beckhoff Headquarters**

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20 33415 Verl Germany

Phone:

Fax: e-mail: +49(0)5246/963-0 +49(0)5246/963-198 info@beckhoff.com

#### Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- · design, programming and commissioning of complex automation systems
- · and extensive training program for Beckhoff system components

Hotline:	+49(0)5246/963-157
Fax:	+49(0)5246/963-9157
e-mail:	support@beckhoff.com

#### **Beckhoff Service**

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- · spare parts service
- · hotline service

Hotline: Fax: e-mail: +49(0)5246/963-460 +49(0)5246/963-479 service@beckhoff.com