Operating Instruction Manual

netSLAVE DTM for Hilscher netX Slave Devices

Configuration of Hilscher Slave Devices

V1.1100

Hilscher Gesellschaft für Systemautomation mbH

www.hilscher.com

DOC080801OI21EN | Revision 21 | English | 2020-03 | Released | Public

Table of Contents

1 Introduction

1.1 About this Manual

This manual describes how to set up and configure Slave devices, which are described with GSDML files. These devices can be configured by use of the netSLAVE DTM within a FDT Framework.

The manual provides information on how to set and configure the device parameters of a Slave device using the netSLAVE DTM, and what can be read from the diagnosis windows.

Descriptions for Gateway Devices

For gateway applications this manual provides descriptions of the configuration and diagnosis of the primary network and secondary network protocols of the different protocol conversions, which also can include Master functionalities.

For the description of the configuration of the gateway and the driver settings as well as of the device assignment procedure for the respective gateway application refer to the netGateway help.

Dialog Windows

The table below gives an overview for the individual dialog windows descriptions:

Section	Subsection	Manual Page
Settings	*Overview Settings*	24
	Driver	27
	Device Assignment	37
	Firmware Download	43
Configuration	*Overview Configuration*	50
	Parameters of the Protocol – Overview Configuration Parameters	52
	Real-Time Ethernet Systems	53
	Fieldbus Systems	69
	Serial Protocols	79
	Overview Command Table and Signal Configuration	52
	Command Table (Gateway)	105
	Signal Configuration (Gateway)	99
Diagnosis	*Overview Diagnosis*	122
	General Diagnosis	123
	Firmware Diagnosis	125
Extended Diagnosis	*Overview Extended Diagnosis*	126
Tools	*Overview Tools*	224
	Packet Monitor	225
	IO Monitor	228

Table 1: Descriptions Dialog Windows

1.1.1 Online Help

The netSLAVE DTM contains an integrated online help facility.

- To open the online help, click on **Help** or press the **F1** key.

1.1.2 List of Revisions

Index	Date	Version	Chapter	Revision
21	20-03-11	1.1100	5.5	Section *Signal Configuration* added.
			1, 2, 11 All	Chapters Introduction, Saftey and Appendix adapted. Terminology correction “pane” -> “window”.

Table 2: List of Revisions

1.2 About netSLAVE DTM

You can use the netSLAVE DTM to configure the Slave device within a FDT Framework.

Devices and Protocols

Using netSLAVE DTM devices can be configured for the following protocols:

As a Slave device:

- Open-Modbus/TCP
- POWERLINK
- VARAN
- CC-Link
- CompoNet

As a gateway device:

- EtherCAT
- EtherNet/IP
- Open-Modbus/TCP
- POWERLINK
- PROFINET IO
- Sercos
- VARAN
- PROFIBUS DP
- CANopen
- DeviceNet
- CC-Link
- CompoNet
- Modbus-RTU
- 3964R
- ASCII
- netSCRIPT

1.2.1 Requirements

System Requirements

- PC with 1 GHz processor or higher
- Windows® XP SP3,
  Windows® Vista (32-Bit) SP2,
  Windows® 7 (32-Bit and 64-Bit) SP1,
  Windows® 8 (32-Bit and 64-Bit),
  Windows® 8.1 (32-Bit and 64-Bit),
  Windows® 10 (32-Bit and 64-Bit)
- Administrator privilege required for installation
- Internet Explorer 5.5 or higher
- RAM: min. 512 MByte, recommended 1024 MByte
- Graphic resolution: min. 1024 x 768 pixel
- Keyboard and Mouse
- Restriction: Touch screen is not supported.

Note: If the project file is used on a further PC,

- this PC must also comply with the above system requirements,
- the device description files of the devices used in the project must be imported into the configuration software SYCON.net on the new PC,
- and the DTMs of the devices used in the project must also be installed on that further PC.

Requirements netSLAVE DTM

To configure a Slave device with the DTM the following requirements have to be accomplished:

- Completed hardware installation of a DTM-compatible cifX Slave device, inclusive loaded firmware and loaded cifX configuration file
- Installed FDT/DTM V1.2 compatible Frame application
- Loaded DTM in the Device Catalog of the FDT frame application

For more information to the hardware installation, please refer to the corresponding **User Manual** of your device.

1.3 Dialog Structure of the netSLAVE DTM

The graphical user interface of the DTM is composed of different areas and elements listed hereafter:

1. A header area containing the **General Device Information**,
2. The **Navigation Area** (area on the left side),
3. The **Dialog Window** (main area on the right side),
4. **OK**, **Cancel**, **Apply**, **Help**,
5. The **Status Line** containing information e. g. the online-state of the utility.

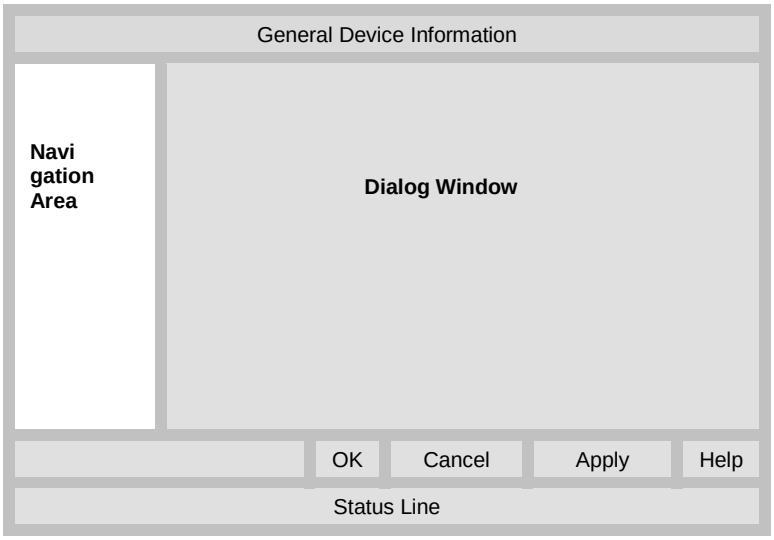


Figure 1: Dialog Structure of netSLAVE DTM

1.3.1 General Device Information

Parameter	Meaning
IO Device	Name of the device
Vendor	Vendor name of the device
Device ID	Identification number of the device
Vendor ID	Identification number of the vendor

Table 3: General Device Information

1.3.2 Navigation Area

The **Navigation Area** contains folders and subfolders to open the dialog windows of the DTM.

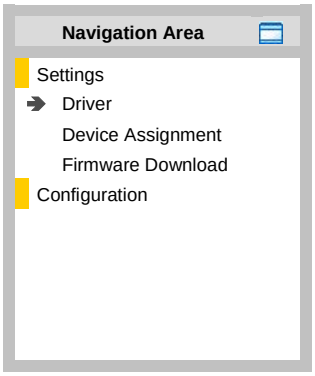


Figure 2: Navigation Area

➢ Select the required folder and subfolder.

➢ The corresponding Dialog window is displayed.

Hide / display Navigation

	Hiding the navigation area (above right side).
Show navigation area	**Opening** the navigation area (below left side).

Gateway Devices

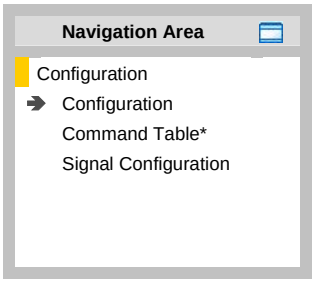


*Figure 3: Navigation Area - Configuration for Protocol Conversions (*for devices with Master functionality)*

1.3.3 Dialog Window

At the dialog window the setting, the configuration or the diagnostic window is opened via the corresponding folder in the navigation area.

Settings	
Driver	On the window **Driver** you can select a driver from the driver list. For further information, refer to section *Driver* on page *27*.
Device Assignment	On the window **Device Assignment** you select the device and assign the device to the driver. For further information, refer to section *Device Assignment* on page *37*.
Firmware Download	The dialog on the window Firmware Download is used to load a new firmware into the device. A detailed description can be found in section *Firmware Download* on page *43*.
Configuration	
Configuration	In the **Configuration** window the parameters of the currently loaded firmware are displayed and can be edited there, provided that a firmware has already been loaded. For further information see section *Overview Configuration* on page 50.
Command Table	For Gateway applications with Master or Client functionality in the **Command Table** window you can define which data of the connected Slave or Server devices you want to read or write automatically. For further information see section *Command Table* on page 105.
Signal Configuration	At the window **Signal Configuration** the data structure of the in- and output signals is provided. For further information see section *Signal* Configuration on page 99.
Diagnosis	
Diagnosis/ Extended Diagnosis	At the **Diagnosis** windows diagnosis information can be read. For further information, refer to section *Overview Diagnosis* on page 122 or section *Overview Extended Diagnosis* on page 126.
Tools	
Packet Monitor/ IO Monitor	Under **Tools** the Packet Monitor and the IO Monitor are provided for test and diagnosis purposes. For further information, refer to section *Packet Monitor* on page *225* or section *IO Monitor* on page 228.

Table 4: Overview Dialog Windows

Note: Accessing the **Diagnosis** windows of the netSLAVE DTM requires an online connection from the netSLAVE DTM to the Slave device .

For further information, refer to section *Connecting/Disconnecting Device* on page 118.

1.3.4 OK, Cancel, Apply and Help

OK, **Cancel**, **Apply** and **Help** you can use as described hereafter.

	Meaning
OK	To confirm your latest settings, click **OK**. All changed values will be applied on the frame application database. *The dialog then closes.*
Cancel	To cancel your latest changes, click **Cancel**. Answer to the safety query **Configuration data has been changed. Do you want to save the data?** by **Yes**, **No** or **Cancel**. **Yes**: The changes are saved or the changed values are applied on the frame application database. *The dialog then closes.* **No**: The changes are not saved or the changed values are not applied on the frame application database. *The dialog then closes.* **Cancel**: *Back to the DTM.*
Apply	To confirm your latest settings, click **Apply**. All changed values will be applied on the frame application database. *The dialog remains opened.*
Help	To open the DTM online help, click **Help**.

Table 5: OK, Cancel, Apply and Help

1.3.5 Table Lines

In the DTM dialog window table lines can be selected, inserted or deleted.

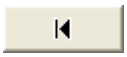
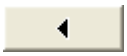
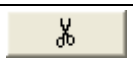
	Meaning
	To select the first line of a table use **First Line**.
	To select the previous line of a table use **Previous Line**.
	To select the next line of a table use **Next Line**.
	To select the last line of a table use **Last Line**.
	**Create a new Line** inserts new lines into the table.
	**Delete selected Line** deletes the selected line from the table.

Table 6: Selecting, inserting, deleting Table Line

1.3.6 Status Bar

The **Status Bar** displays information about the current state of the DTM. The current activity, e.g. download, is signaled graphically via icons in the status bar.

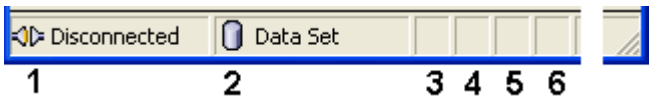
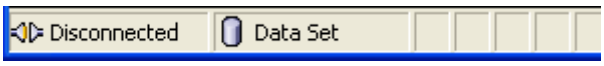


Figure 4: Status Bar – Status Fields 1 to 6

Status Field	Icon / Meaning	
1	**DTM Connection States**	
		Connected: Icon closed = Device is online
		Disconnected: Icon opened = Device is offline
2	**Data Source States**	
		Data set: The displayed data are read out from the instance data set (database).
		Device: The displayed data are read out from the device.
3	**States of the instance Date Set**	
		Valid Modified: Parameter is changed (not equal to data source).
4	**Changes directly made on the Device**	
		Load/configure diagnosis parameters: Diagnosis is activated.
6	**Device Diagnosis Status**	
		Save operation succeeded: The save operation has been successful.
		Further messages due to successful handling of device data.
		Firmware Download: Firmware Download is running
		Save operation failed: The save operation has failed.
		Further fail operation messages due to incorrect communication due to malfunction in the field device or its peripherals.

Table 7: Status Bar Icons [1]

Offline State



Save operation succeeded

Firmware Download

Firmware Download successful

Online State and Diagnosis

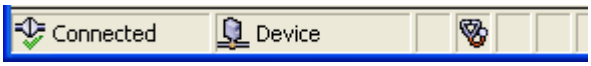


Figure 5: Status Bar Display Examples

2 Safety

2.1 General Note

The documentation in the form of a user manual, an operating instruction manual or other manual types, as well as the accompanying texts have been created for the use of the products by educated personnel. When using the products, all Safety Messages, Integrated Safety Messages, Property Damage Messages and all valid legal regulations must be obeyed. Technical knowledge is presumed. The user has to assure that all legal regulations are obeyed.

2.2 Intended Use

The netSLAVE DTM serves for configuration and diagnosis of Slave devices.

2.3 Personnel Qualification

Personnel responsible for the application of the network system shall be aware of the system behavior and shall be trained in using the system.

2.4 Safety Instructions

To ensure your own personal safety and to avoid personal injury, you necessarily must read, understand, and comply with the safety instructions and safety messages in this manual before you install and operate your system.

For cases if both, personal injury as well as property damage (damage of equipment or device) may occur together, you find the safety instructions in this section.

2.4.1 Communication Stop during Firmware or Configuration Download

If you want to perform either a firmware update (as a download) or a configuration download, both via the netSLAVE DTM, be aware of the following:

- Together with the firmware download, an automated reset to the device is performed that will interrupt all network communication and all established connections will drop.
- If you download the configuration during bus operation, the communication between master and slaves is stopped.

Possible faulty System Operation

- An unpredictable and unexpected behavior of machines and plant components may cause personal injury and property damage.

➢ Stop the application program, before starting the firmware update or before downloading the configuration.

➢ Make sure that your equipment operates under conditions that prevent personal injury or property damage. All network devices should be placed in a fail-safe mode, before starting the firmware update or before downloading a configuration.

Loss of Device Parameters, Overwriting of Firmware

- Both the firmware download and the configuration download erase the configuration data base. The firmware download overwrites the existing firmware in the network device.

➢ To complete the firmware update and to make the device operable again, re-load the configuration after the firmware update has been finished.

For devices with Ethernet technology

- Device parameters that have been saved volatile, e. g. as the temporarily set IP address parameters, are getting lost during the reset.

➢ In order to prevent loss of configuration data, make sure that your project configuration data are saved non-volatile, before you initiate a firmware update or download the configuration.

2.4.2 Mismatching System Configuration

Mismatching system configuration loaded into the device could result in faulty data mapping in the application program and thus unexpected equipment operation may cause personal injury or damage of equipment.

➢ In the device use only a configuration suitable for the system.

2.5 Property Damage

To avoid property damage and damage to your system or to your equipment, you necessarily must read, understand, and comply with the safety instructions and safety messages in this manual before you configure your system.

2.5.1 Power Disconnect while dowloading Firmware or Configuration

If during the process of downloading a firmware or configuration

- the power supply to a PC with the software application is interrupted,
- or the power supply to the netSLAVE Slave device is interrupted,
- or a reset to the device is performed,

this may lead to the following consequences:

Loss of Device Parameters, Firmware Corruption

- The firmware download or the configuration download will be interrupted and remains incomplete.
- The firmware or the configuration database will be corrupted and device parameters will be lost.
- Device damage may occur as the device cannot be rebooted.

Whether these consequences occur depends on when the power disconnect occurs during the download.

Power Drop during Write and Delete Accesses in the File System

The FAT file system in the netX firmware is subject to certain limitations in its operation. Write and delete accesses in the file system (firmware update, configuration download etc.) can destroy the FAT (File Allocation Table) if the accesses cannot be completed if the power drops. Without a proper FAT, a firmware may not be found and cannot be started.

Make sure that the power supply to the device is not interrupted during write and delete accesses in the file system (firmware update, configuration download, etc.).

2.5.2 Invalid Firmware

Loading invalid firmware files could render your device unusable.

➢ Only download firmware files to the device that are valid for this device.

Otherwise you might be forced to return your device for repair.

2.6 Safety Messages on Firmware or Configuration Download

If you perform a firmware download or a configuration download via the netSLAVE DTM be aware of the following:

WARNING

Communication Stop caused by Firmware or Configuration Download
Initiating a firmware or configuration dowload process during bus operation will stop the communication and a subsequent plant stop may cause unpredictable and unexpected behavior of machines and plant components, possibly resulting in personal injury and damage to your equipment.
The firmware download overwrites the existing firmware. The communication stop may cause loss of device parameters and possible device damage may occur.

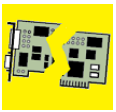
- Stop the application program, before you start the firmware or configuration dowload.
- Make sure that all network devices are placed in a fail-safe condition.

WARNING

Mismatching System Configuration
Mismatching system configuration loaded into the device could result in faulty data mapping in the application program and thus unexpected equipment operation may cause personal injury or damage of equipment.

- In the device use only a configuration suitable for the system.

NOTICE

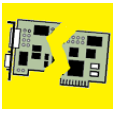


Power Disconnect while dowloading Firmware or Configuration

If the power supply to the PC or device is interrupted while the firmware or configuration is being downloaded, the download will be aborted, the firmware may be corrupted, the device parameters may be lost, and the device may be damaged.

- During firmware or configuration download process do not interrupt the power supply to the PC, or to the device and do not perform a reset to the device!

NOTICE



Invalid Firmware
Loading invalid firmware files could render your device unusable.

- Only proceed with a firmware version valid for your device.

3 Getting started

3.1 Configuration Steps

The following table describes the steps to configure a device with netSLAVE DTM as it is typical for many cases. At this time it is presupposed that the hardware installation was done.

#	Step	Short Description	For detailed information see section	Page
1	Load device catalog	Depending of the FDT Container: For netDevice: - select **Network > Device Catalog**, - select **Reload Catalog**.	*(See Operating Instruction Manual netDevice and netProject)*	-
2	Create new project / Open existing project	Depending of the frame application. For the configuration software: - select **File > New** or **File > Open**.	*(See Operating Instruction Manual netFrame)*	-
3	Insert Slave device into configuration	Depending of the FDT Container: For netDevice: - in the Device Catalog under **Gateway / Stand-Alone Slave** click to the Slave device, - and insert the device via drag and drop **to the root line** in the network view.	*(See Operating Instruction Manual netFrame)*	-
4	Open the netSLAVE DTM configuration dialog	Open the netSLAVE DTM configuration dialog. - Double click to the device icon of the Slave. - The netSLAVE DTM configuration dialog is displayed.	-	-
5	Verify or adapt Driver Settings	In the netSLAVE DTM configuration dialog: - select **Settings > Driver**, • **Note!** For PC cards cifX the **cifX Device Driver** is preset as a default driver. For all the other Hilscher devices the **netX Driver** is preset as a default driver.Use the **cifX Device Driver** if the netSLAVE DTM is installed on the same PC as the Slavedevice. • Use the **netX Driver** to establish a USB, Serial (RS232) or TCP/IP connection from the netSLAVE DTM to the Slave device. • The **3SGateway Driver for netX (V3.x)** is used only in relationship with CODESYS. To search for devices you can check one or multiple drivers simultaneously. - Verify that the default driver is checked. - If necessary, check another driver or multiple drivers.	*Settings for Driver and Device Assignment* and *Driver*	25 27

#	Step	Short Description	For detailed information see section	Page
6	Configure Driver	If you use the **netX Driver**, you respectively must configure it. For **netX Driver and communication via TCP/IP** set the IP address of the device: - Select **Settings > Driver > netX Driver > TCP Connection**. - Via  add an IP range. - Under **IP Address** enter the IP Address of the device or an IP range. - Click **Save**. Adjust the driver parameters **netX Driver USB/RS232** only if they differ from the default settings.  **Note!** • The **cifX Device Driver** requires no configuration. • The configuration of the **3SGateway Driver for netX (V3.x)** is carried out via the CODESYS surface.	*netX Driver*	29
7	Assign Slave device (with or without firmware)	Assign the device to this driver. In the netSLAVE DTM configuration dialog: - select **Settings > Device Assignment**, - select a Slave device (with or without firmware), - therefore check the appropriate checkbox, - select **Apply**.	*Selecting the Device (with or without firmware)*	40
8	Select and download firmware	If not yet a firmware was loaded to the device: - Adhere to the necessary safety precautions to prevent personnel injury and property damage. In the netSLAVE DTM configuration dialog: - select **Settings > Firmware Download**, - select **Browse..**, - select a firmware file, - select **Open**, - select **Download** and **Yes**.	*Safety Messages on Firmware or Configuration Download* *Firmware Download*	18 43
9	Assign Slave device once more (with firmware and system chanal) *For repeated download this step is omitted.*	In the netSLAVE DTM configuration dialog: - select **Settings > Device Assignment**, - select **Scan**, - select the Slave device (with loaded firmware and defined system channel), - therefore check the appropriate checkbox, - select **Apply**, - close the netSLAVE DTM configuration dialog via **OK**.	*Selecting the Device once more (with Firmware)*	41
10	Configure Slave device	In the netSLAVE DTM configuration dialog: - select **Configuration**, - set the configuration parameters for the Slave to be used, - close the netSLAVE DTM configuration dialog via **OK**.	*Overview Configuration* and *Parameters of the Protocol – Overview Configuration Parameters*	50 52

#	Step	Short Description	For detailed information see section	Page
11	Create Signal Configuration	In the netSLAVE DTM configuration dialog: - select **Configuration** > **Signal Configuration** - Uncheck **Disable advanced editing**. - Adjust Signal Names, Define Data Types, Add Module or Signal. - select **Apply**, - close the netSLAVE DTM configuration dialog via **OK**.	*Signal Configuration*	99
12	Save project	Depending of the frame application. For the configuration software: - select **File > Save**.	*(See Operating Instruction Manual netFrame)*	-
13	Connect device	Depending of the FDT Container. For netDevice: - right click to the device icon, - select **Connect**.	*Connecting/Disconnecting Device*	118
14	Download Configuration	- Adhere to the necessary safety precautions to prevent personnel injury and property damage. Depending of the FDT Container. For netDevice: - right click to the device icon,, - select **Download**.	*Safety Messages on Firmware or Configuration Download* *Download Configuration*	18 120
15	Diagnosis	Depending of the FDT Container. For netDevice: - right click to the device icon, - select **Diagnosis**. - The netSLAVE DTM diagnosis dialog is displayed. (1) Check whether the communication is OK: **Diagnosis** > **General Diagnosis** > **Device status "Communication"** must be green! (2) **"Communication"** is green: Open thel IO Monitor and test the input or output data. (3) **"Communication"** is not green: Use Diagnosis and Extended diagnosis for troubleshooting. - close the netSLAVE DTM diagnosis dialog via **OK**.	*Overview Diagnosis*	122
16	IO Monitor	Depending of the FDT Container: For netDevice: - right click to the device icon of the Slave, - select **Diagnosis**, - select **Tools > IO Monitor**. - Check the input or output data, - close the IO Monitor dialog via **OK**.	*IO Monitor*	*228*
17	Disconnect	Depending of the FDT Container. For netDevice: - right click to the device icon, - select **Disconnect**.	*Connecting/Disconnecting Device*	118

Table 8: Getting started - Configuration Steps

3.2 Configuration Steps for the Gateway

The following table describes the steps how to configure the primary network and the secondary network protocols of the different protocol conversions for gateway applications as it is typical for many cases. This configuration must be done in the DTM dialog of the respective Slave or Master device.

At this time it is presupposed that the hardware installation was done and the gateway application is configured in the netGateway DTM dialog (configuration of the gateway, driver settings and device assignment).

#	Step	Short Description	For detailed information see section	Page
1	Create new project / Open existing project	Depending of the frame application. For the configuration software: - select **File > New** or **File > Open**.	*(See Operating Instruction Manual netFrame)*	-
2	Insert Gateway device into configuration	Depending of the FDT Container: For netDevice: - in the Device Catalog under **Gateway / Stand-Alone Slave** click to the Gateway device, - and insert the device via drag and drop **to the line** in the network view.	*(See Operating Instruction Manual netFrame)*	-
3	Open the netGateway DTM Configuration Dialog	Depending of the FDT Container: For netDevice: - Double click to the device icon of the gateway. - The netGateway DTM configuration dialog is displayed.	-	-
4	Select Primary Network and Secondary Network Protocols and apply	netGateway-DTM-Dialog: - select **Configuration > Settings**. - under **Protocol Combination > Primary network (Port X2)** select the primary network protocol. - under **Protocol Combination > Secondary network (Port X3)** select the secondary network protocol. - close the netGateway DTM configuration dialog via **OK**.	*(See Operating Instruction Manual netGateway and User Manual for the Gateway device)*	-
5	If necessary enter Slave into Configuration at the Primary Network	If the gateway works as Master at the primary network, insert the Slave device as follows. Depending of the FDT Container: For netDevice: - in the Device Catalog under **Slave** click to the Slave device, - and insert the device via drag and drop **to the gateway line** in the network view.	*(See Operating Instruction Manual netFrame)*	-
6	Open the DTM Configuration Dialog for Primary Network	Depending of the FDT Container: For netDevice: - right click to the device icon of the gateway, - select **Configuration >** [*Protocol Name Primary Network*]. - The DTM configuration dialog is displayed.	-	-
7	Configure Primary Network Protocol	In the DTM configuration dialog: - select **Configuration > Configuration**, - set the configuration parameters for the respective primary network protocol. If the gateway works as Modbus/TCP Client at the primary network: - select **Bus** > **Mode** > „Message Mode“. - select **Configuration > Command Table**, - define which data of the connected Server devices shall be read or written automatically, - close the DTM configuration dialog via **OK**.	*Overview Configuration* and *Parameters of the Protocol – Overview Configuration Parameters,* *Open Modbus/TCP Parameters, Command Table Open Modbus/TCP*	50, 52, 56, 111

#	Step	Short Description	For detailed information see section	Page
8	If necessary enter Slave into Configuration at the Secondary Network	If the gateway works as Master at the secondary network, insert the Slave device as follows. Depending of the FDT Container: For netDevice: - in the Device Catalog under **Slave** click to the Slave device, - and insert the device via drag and drop **to the gateway line** in the network view.	*(See Operating Instruction Manual netFrame)*	-
9	Open the DTM Configuration Dialog for Secondary Network	For netDevice: - right click to the device icon of the gateway, - select **Configuration >** [*Protocol Name Secondary Network*]. - The DTM configuration dialog is displayed.	-	-
10	Configure Secondary Network Protocol	In the DTM configuration dialog: - select **Configuration > Configuration**, - set the configuration parameters for the respective secondary network protocol. If the gateway works as Modbus RTU Master at the secondary network: - select **Bus** > **Mode** > „Master“, - select **Configuration > Command Table**, - define which data of the connected Slave devices shall be read or written automatically, - close the DTM configuration dialog via **OK**.	*Parameters of the Protocol – Overview Configuration Parameters,* *Modbus RTU Parameters, Command Table Modbus RTU*	52, 79, 105
11	Open the netGateway DTM Configuration Dialog	For netDevice: - Double click to the device icon of the gateway. - The netGateway DTM configuration dialog is displayed.	-	-
12	Configuration Gateway	- Proceed the Signal Mapping, - Set the driver, - Proceed the device assignment, - close the netGateway DTM configuration dialog via **OK**.	*(See Operating Instruction Manual netGateway and User Manual for the Gateway device)* *Signal Configuration*	- 99
13	Save project	Depending of the frame application. For the configuration software: - select **File > Save**.	*(See Operating Instruction Manual netFrame)*	-
14	Connect device	Depending of the FDT Container. For netDevice: - right click to the device icon of the gateway, - select **Connect**.	*Connecting/Disconnecting Device*	118
15	Download Configuration	- Adhere to the necessary safety precautions to prevent personnel injury and property damage. Depending of the FDT Container. For netDevice: - right click to the device icon of the gateway, - select **Download**.	*Safety Messages on Firmware or Configuration Download* *Download Configuration*	18 120
16	Diagnosis	Depending of the FDT Container: For netDevice: - right click to the device icon of the gateway, - select context menu **Diagnosis > Gateway, [*Protocol Name Primary Network*]** or **[*Protocol Name Secondary Network*]**. - The DTM diagnosis dialog is displayed. - Continue with further device diagnosis, - close the DTM diagnosis dialog via **OK**.	*Overview Diagnosis*	122
17	Disconnect	Depending of the FDT Container. For netDevice: - right click to the device icon of the gateway, - select **Disconnect**.	*Connecting/Disconnecting Device*	118

Table 9: Getting started - Configuration Steps Gateway

4 Settings

4.1 Overview Settings

Settings Dialog Windows

The table below gives an overview for the individual **Settings** dialog windows descriptions:

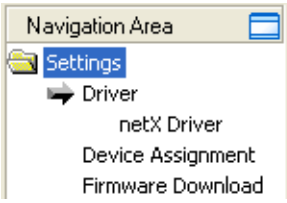
netSLAVE DTM	Folder Name / Section	Subsection	Manual Page
 *Navigation Area – Settings (Example)* *Additional drivers can be displayed.*	*Driver*		27
		Verify or adapt Driver Settings	27
		cifX Device Driver	29
		netX Driver	29
		Configuring netX Driver	30
	Device Assignment		37
		Scanning for Devices	37
		Scanning for all Devices or for suitable only	39
		Selecting the Device (with or without firmware)	40
		Selecting the Device once more (with Firmware)	41
	Firmware Download		43

Table 10: Descriptions of the Dialog Windows Settings

Note: To edit the **Settings** dialog windows you need *User Rights* for “Maintenance”.

Notice the descriptions in the section *Settings for Driver and Device Assignment* on page 25.

To access to the online help with the descriptions of the drivers:

- Select **Settings > Driver >** [*Name of the assigned driver*].
- Press the **F1** key.

For information on the **Settings** for gateway applications refer to the *Operating Instruction Manual netGateway* and the **User Manual** for the Gateway device.

4.2 Settings for Driver and Device Assignment

The following steps are needed to establish a connection from the netSLAVE DTM to the Slave device:

Verifying or adapting Driver Settings

Verify the Driver Settings and adapt them if necessary.

1. Open the DTM configuration dialog.

➢ In the FDT container **netDevice** double click to the Slave device icon.

2. Verify that the default driver is checked and respectively check another or multiple drivers.

➢ Select **Settings > Driver**.

Note! For PC cards cifX the **cifX Device Driver** is preset as a default driver. For all the other Hilscher devices the **netX Driver** is preset as a default driver.

- Use the **cifX Device Driver** if the netSLAVE DTM is installed on the same PC as the Slave device.
- Use the **netX Driver** to establish an USB, Serial (RS232) or TCP/IP connection from the netSLAVE DTM to the Slave device.
- The **3SGateway Driver for netX (V3.x)** is used only in relationship with CODESYS.

To search for devices on the network you can check one or multiple drivers simultaneously.

➢ Verify that the default driver for your device is checked.

➢ If necessary, check another driver or multiple drivers.

Configuring Driver

Note!

- The **cifX Device Driver** requires no configuration.
- The configuration of the **3SGateway Driver for netX (V3.x)** is carried out via the CODESYS surface.

If you use the **netX Driver**, you respectively must configure it.

3. Configure the **netX Driver** if necessary.

For the driver **netXDriver** an individual driver dialog window can be opened where you can configure the driver.

➢ Select **Settings > Driver** > **netX Driver**.

➢ For netX Driver and communication via TCP/IP set the IP address of the device.

Adjust the driver parameters **netX Driver USB/RS232** only if they differ from the default settings.

Assigning the Slave device to the DTM

4. Scan for and select the devices (with or without firmware).

➢ Select **Settings > Device Assignment**.

➢ Under **Device selection** select *suitable only* or *all* and then **Scan**.

➢ In the table check the required devices.

➢ Select **Apply**.

Selecting and downloading the Firmware

5. If not yet a firmware was loaded to the device, select and download the firmware.

➢ Select **Settings > Firmware Download**.

➢ Select and download the firmware via **Download**.

➢ Select **Apply**.

6. Scan for and select the devices (with firmware and defined system channel) once more.

   For repeated download this step is omitted.

➢ Select **Settings > Device Assignment**.

➢ Select **Scan**.

➢ In the table check the required device.

7. Close the DTM configuration dialog via **OK**.

Connecting the Device

8. In **netDevice** put a right-click on the Slave device icon.
9. Select the **Connect** command from the context menu.

↬ In the network view the device description at the device icon of the Slave is displayed with a green colored background. The Slave device now is connected to the netSLAVE DTM via an online connection.

Further Information

For descriptions about these steps refer to the sections following hereafter.

4.3 Driver

The **Driver** dialog window displays the drivers to be used for a netSLAVE DTM to establish a device communication connection.

Note! A **default driver** is set in the configuration software.

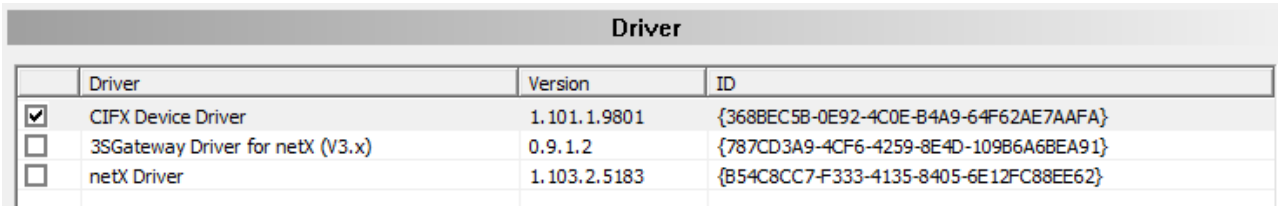


Figure 6: Default Driver ‚cifX Device Driver' for PC cards cifX

Parameter	Meaning
Driver	Name of the driver (for more details see descriptions hereafter)
Version	ODMV3 Version of the respective driver
ID	ID of the driver (driver identification)

Table 11: Driver Selection List Parameters

To establish a connection from the netSLAVE DTM to the Slave device, verify if the default driver is checked and respectively check another driver or multiple drivers.

4.3.1 Verify or adapt Driver Settings

Proceed as follows:

1. Select **Settings > Driver** in the navigation area.

➮ The **Driver** dialog window is displayed with the available drivers and the setting for the default driver.

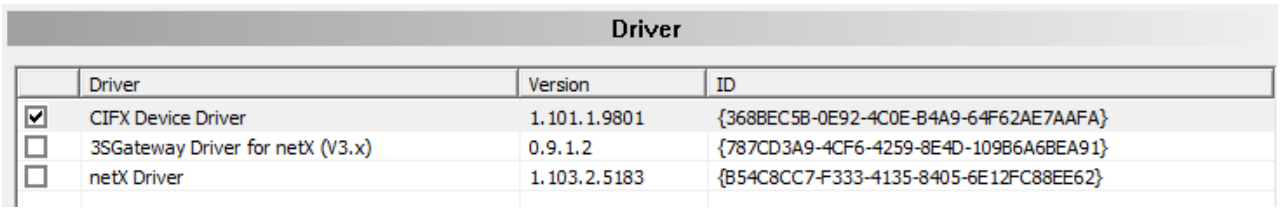


Figure 7: Default Driver ‚cifX Device Driver' for PC cards cifX (example)

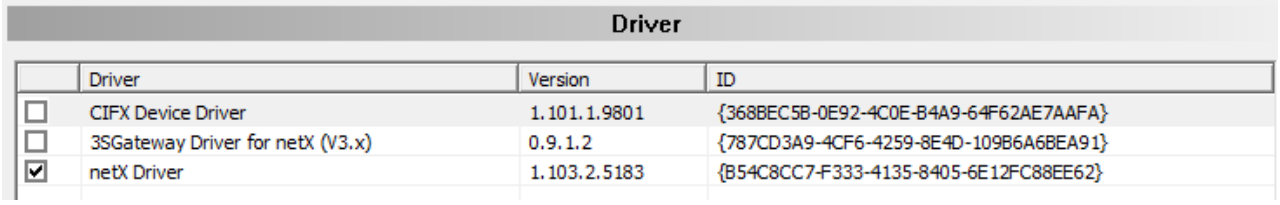


Figure 8: Default Driver ‚netX Driver' for Hilscher devices except for PC cards cifX (example)

2. Verify that the default driver is checked.

➢ Verify that the default driver for your device is checked.

Default Driver (Pre-settings in the Configuration Software): For PC cards cifX the **cifX Device Driver** is preset as a default driver. For all the other Hilscher devices the **netX Driver** is preset as a default driver.

3. Respectively check another driver.

Note! The driver used for the connection from the netSLAVE DTM to the Slave device must be supported by the device and must be available for the device.

- Use the **cifX Device Driver** if the netSLAVE DTM is installed on the same PC as the Slave device.
- Use the **netX Driver** to establish a USB, Serial (RS232) or TCP/IP connection from the netSLAVE DTM to the Slave device.
- The **3SGateway Driver for netX (V3.x)** is used only in relationship with CODESYS. The version V3.x refers to the driver version defined by 3S-Smart Software Solutions GmbH.

➢ Check the checkbox for the driver in the selection list.

4. Respectively check multiple drivers.

To search for devices on the network you can check multiple drivers simultaneously.

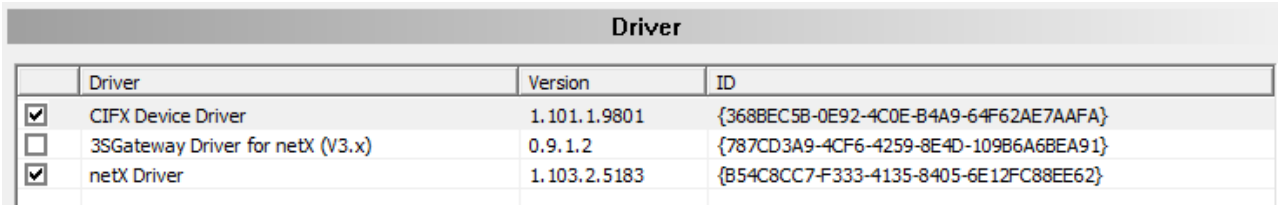


Figure 9: Manual Selection of multiple drivers (Example)

4.3.2 cifX Device Driver

In the netSLAVE DTM for the **cifX Device Driver** no driver dialog window is available, since for the **cifX Device Driver** no driver settings are required.

The **cifX Device Driver** will be used if the netSLAVE DTM is installed in the same PC as the Slave device.

Note: To establish a connection from a DTM to a Slave device via the **cifX Device Driver**, the **cifX Device Driver** must be installed and the driver must have access to the Slave device.

4.3.3 netX Driver

The **netX Driver** is used to connect the DTM to the device via different connection types. The DTM communicates with the device via an USB connection, a serial (RS232) connection or a TCP/IP connection. The **netX Driver** establishes

- via the USB interface of the device and the USB port of the PC an USB connection to the device,
- via the RS232 interface of the device and the COM port of the PC a serial connection (RS232) to the device
- and via Ethernet a TCP/IP connection to the device.

To connect the DTM to the physical layer of the device the **netX Driver** software works in combination with the software components:

- “USB/COM connector” for the USB connection and for the serial connection (RS232) and
- “TCP/IP connector” for the Ethernet connection.

4.3.4 Configuring netX Driver

The following steps are required to configure the netX Driver:

USB/RS232 Connection

To set the driver parameters for an USB/RS232 connection note:

Note: Adjust the driver parameters netX Driver USB/RS232 only if they differ from the default settings. After saving the changed driver parameters, these parameters are used for the device assignment when scanning devices.

For setting the driver parameters for an USB connection or a serial connection:

1. Select **Settings > Driver > netX Driver > USB/RS232 Connection**.

➢ Set the driver netX Driver USB/RS232 parameters.

TCP/IP Connection

For setting the driver parameters for a TCP/IP connection:

1. Select **Settings > Driver > netX Driver > TCP Connection**.
2. Set IP Address of the device:

➢ Add an IP Range via **Select IP Range**  .

3. Under **IP Range Configuration** > **IP Address** enter the IP Address of the device (**Use IP Range** is unchecked).

Or

4. Set IP Range:

➢ Check **Use IP Range**.

➢ Under **IP Range Configuration** > **IP Address** enter the start address (left side) and the ending address of the IP scanning range (right side).

5. Click **Save**, to save the IP address or the IP range.

➢ After saving the changed driver parameters, these parameters are used for the device assignment when scanning devices.

4.3.5 netX Driver - USB/RS232 Connection

The communication from the DTM to the device via an **USB/RS232 Connection** is used when the DTM is installed on a PC and between the PC and the device

- an USB connection
- or a serial connection (RS232) exists.

The DTM accesses the device via the USB interface or via the RS232 interface. This requires either to connect an USB port of the PC to the USB interface of the device using an USB cable or to connect a physical COM port of the PC to the RS232 interface of the device via a serial cable.

The **netX Driver** / **USB/RS232 Connection** supports all physical and virtual COM ports available on the PC.

Via the RS232 interface or USB interface, the device is configured or diagnosis is performed.

4.3.5.1 Driver Parameters for netX Driver - USB/RS232 Connection

The settings of the driver parameters for the USB/RS232 connection are made via the **netX Driver / USB/RS232 Connection** configuration dialog.

➢ Open the **USB/RS232 Connection** dialog via navigation area **Settings > Driver > netX Driver**.

➢ The **USB/RS232 Connection** dialog is displayed:

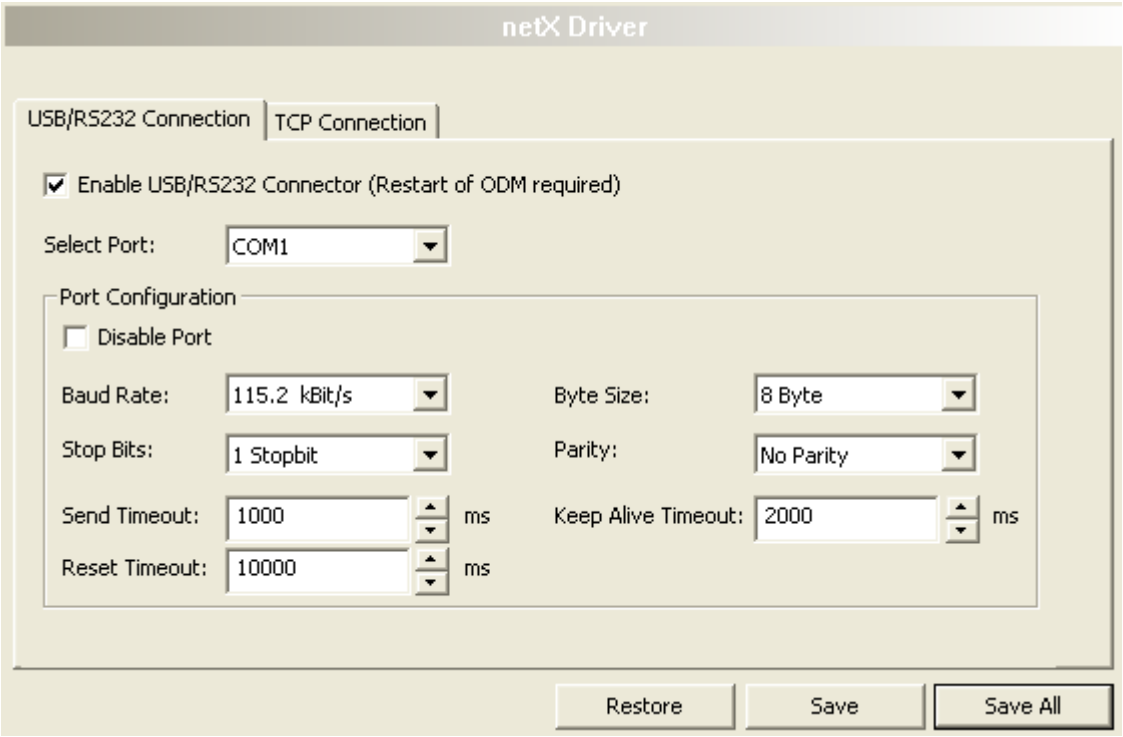


Figure 10: netX Driver > USB/RS232 Connection

Parameter	Meaning	Range of Value / Default Value
Enable USB/RS232 Connector (Restart of ODM required)	checked: The netX Driver can communicate via the USB/RS232 interface. unchecked: The netX Driver can not communicate via the USB/RS232 interface. If the check mark for **Enable USB/RS232 Connector** is set or removed, then the ODM server must be restarted[1], to make the new setting valid. _____ [1] Restart the ODM server via the **ODMV3 Tray Application**: -In the foot line click on  using the right mouse key. - In the context menu select **Service > Start**.	checked, unchecked; Default: unchecked
Select Port	Depending on the COM ports (interfaces) available on the PC, they will be listed under **Select Port**.	COM 1 to COM N
Port Configuration		
Disable Port	checked: No connection. unchecked: The netX Driver tries to establish a connection using the configured USB/RS232 interface.	checked, unchecked (Default)
Baud rate	Transfer rate: number of bits per second. The device must support the baud rate.	9.6, 19.2, 38.4, 57.6 or 115.2 [kBit/s]; Default (RS232): 115.2 [kBit/s]

Parameter	Meaning	Range of Value / Default Value
Stop bits	Number of stop bits sent after the transfer of the send data for synchronization purposes to the receiver.	Stop bit: 1, 1.5, 2; Default (RS232): 1
Send Timeout	Maximum time before the transfer of the transmission data is canceled, when the send process fails, for example, because of the transfer buffer is full.	100 … 60.000 [ms]; Default (RS232 and USB): 1000 ms
Reset Timeout	Maximum time for a device reset, including the re-initialization of the physical interface used for the communication.	100 … 60.000 [ms]; Default (RS232 and USB): 5000 ms
Byte size	Number of bits per byte by byte specification	7 Bit, 8 Bit; Default (RS232): 8 Bit
Parity	In the error detection in data transmission using parity bits, "parity" describes the number of bits occupied with 1 in the transmitted information word. No Parity: no parity bit Odd Parity: The parity is "odd" if the number of bits occupied with 1 in the transmitted information word will be odd. Even parity: The parity is "even" if the number of bits occupied with 1 in the transmitted information word will be even. Mark Parity: if the parity bit is always 1, this will be named mark-parity (the bit does not contain any information). Space Parity: if the parity bit always 0, this will be named space-parity (the bit represents an empty space).	No Parity, Odd Parity, Even Parity, Mark Parity, Space Parity; Default (RS232): No Parity
Keep Alive Timeout	The "Keep Alive" mechanism is used to monitor whether the connection to the device is active. Connection errors are detected using a periodic heartbeat mechanism. The heartbeat mechanism will be initiated after the set time has elapsed if the communication has failed.	100 … 60.000 [ms]; Default (RS232 and USB): 2000 ms
Restore	Resets all settings in the configuration dialog to the default values.	
Save	Saving all settings made in the configuration dialog **netX Driver > Save USB/RS232 Connection**, i. e. only for the selected connection type.	
Save All	Saving all settings made in the configuration dialog **netX Driver**, i. e. for all connection types.	

Table 12: Parameters netX Driver > USB/RS232 Connection

4.3.6 netX Driver - TCP/IP Connection

The communication from the DTM to the device via a **TCP/IP Connection** is used in the following two typical applications:

Application 1: The device has its own Ethernet interface. The DTM is installed on a PC and the TCP/IP connection is established from this PC to the stand-alone device. The IP address of the device is used.

Application 2: The device is installed in a remote PC. The DTM is installed on an additional PC and the TCP/IP connection is established from this PC to the remote PC. The IP address of the remote PC is used. For the TCP/IP connection is made, on the remote PC the cifX TCP/IP server must be started. The cifX TCP/IP server allows the remote access to the device via a TCP/IP connection.

Note: An exe file for the cifXTCP/IP server is provided on the product CD in the *Tools* directory.

Via the TCP/IP interface of the device or of the remote PC, the device is configured or diagnosis is performed.

4.3.6.1 Driver Parameters for netX Driver - TCP/IP Connection

The settings of the driver parameters for the TCP/IP connection are made via the **netX Driver / TCP Connection** configuration dialog.

➢ Open the **TCP Connection** dialog via navigation area **Settings > Driver > netX Driver**.

↬ The dialog **netX Driver** is displayed:

➢ Select **TCP Connection**.

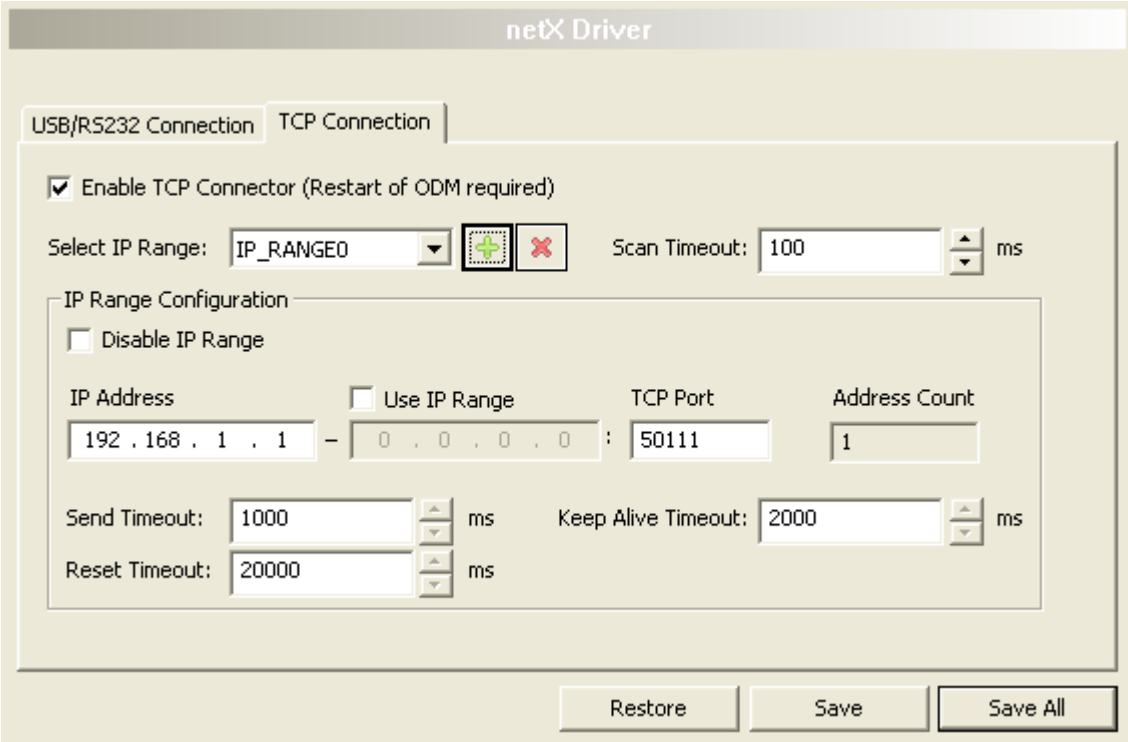


Figure 11: netX Driver > TCP Connection

Parameter	Meaning	Range of Value / Default Value
Enable TCP Connector (Restart of ODM required)	checked: The netX Driver can communicate via the TCP/IP interface. unchecked: The netX Driver can not communicate via the TCP/IP interface. If the check mark for **Enable TCP Connector** is set or removed, then the ODM server must be restarted[1], to make the new setting valid. _____ [1] Restart the ODM server via the **ODMV3 Tray Application**: -In the foot line click on  using the right mouse key. - In the context menu select **Service > Start**.	checked, unchecked; Default: unchecked
Select IP Range	Via **Select IP Range** already created IP ranges can be selected. Via  an additional IP range can be added. Via  an IP range can be deleted.	
Scan Timeout [ms]	With **Scan Timeout** can be set, how long to wait for a response while a connection is established.	10 … 10000 [ms]; Default: 100 ms

Parameter	Meaning	Range of Value / Default Value
IP Range Configuration		
Disable IP Range	checked: No connection. unchecked: The netX Driver tries to establish a connection using the configured TCP/IP interface.	checked, unchecked (Default)
IP Address (left)	Enter the IP address of the device, (if **Use IP Range** is not checked). Enter the start address of the IP scanning range, (if **Use IP Range** is checked).	valid IP address; Default: 192.168.1.1
Use IP Range	checked: An IP address range is used. unchecked: Only one IP address is used.	checked, unchecked; Default: unchecked
IP Address (right)	Enter the ending address of the IP scanning range, (only if **Use IP Range** is checked).	valid IP address; Default: 0.0.0.0
Address Count	Displays the scanning range address count, depending on the selected IP-start or IP-end address. (For this read the note given below.)	recommended: 10
TCP Port	Identifies the endpoint of a logical connection or addresses a specific endpoint on the device or PC.	0 - 65535; Default Hilscher device: 50111
Send Timeout [ms]	Maximum time before the transfer of the transmission data is canceled, when the send process fails, for example, because of the transfer buffer is full.	100 … 60.000 [ms]; Default (TCP/IP): 1000 ms
Reset Timeout [ms]	Maximum time for a device reset, including the re-initialization of the physical interface used for the communication.	100 … 60.000 [ms]; Default (TCP/IP): 2000 ms
Keep Alive Timeout [ms]	The "Keep Alive" mechanism is used to monitor whether the connection to the device is active. Connection errors are detected using a periodic heartbeat mechanism. The heartbeat mechanism will be initiated after the set time has elapsed if the communication has failed.	100 … 60.000 [ms]; Default (TCP/IP): 2000 ms
Restore	Resets all settings in the configuration dialog to the default values.	
Save	Saving all settings made in the configuration dialog **netX Driver > Save TCP/IP Connection**, i. e. only for the selected connection type.	
Save All	Saving all settings made in the configuration dialog **netX Driver**, i. e. for all connection types.	

Table 13: Parameters netX Driver > TCP Connection

Note: Do not use large IP ranges in combination with a low scan timeout. Microsoft introduced in Windows® XP SP2 a limit of concurrent half-open outbound TCP/IPconnections (connection attempts) to slow the spread of virus and malware from system to system. This limit makes it impossible to have more than 10 concurrent half-open outbound connections. Every further connection attempt is put in a queue and forced to wait. Due to this limitation a large IP range used in combination with a low scan timeout could prevent the connection establishment to a device.

4.4 Device Assignment

Note: In the **Device Assignment** dialog window you first must assign the Slave device to the netSLAVE DTM by checking the check box. This is essential to establish an online connection from the netSLAVE DTM to the Slave device later, as described in section *Connecting/Disconnecting Device* on page 118.

Therefore in the **Device Assignment** dialog window you scan for the Slave device and select it.

If the device did not get a firmware or shall get a new firmware:

1. first you scan for the device (with or without firmware) and select the device,
2. then you download a firmware to the device and
3. subsequently you scan for the device (with firmware) once more and select the device again.

4.4.1 Scanning for Devices

1. Select **Settings > Device Assignment** in the navigation area.

↬ The dialog window **Device Assignment** is displayed.

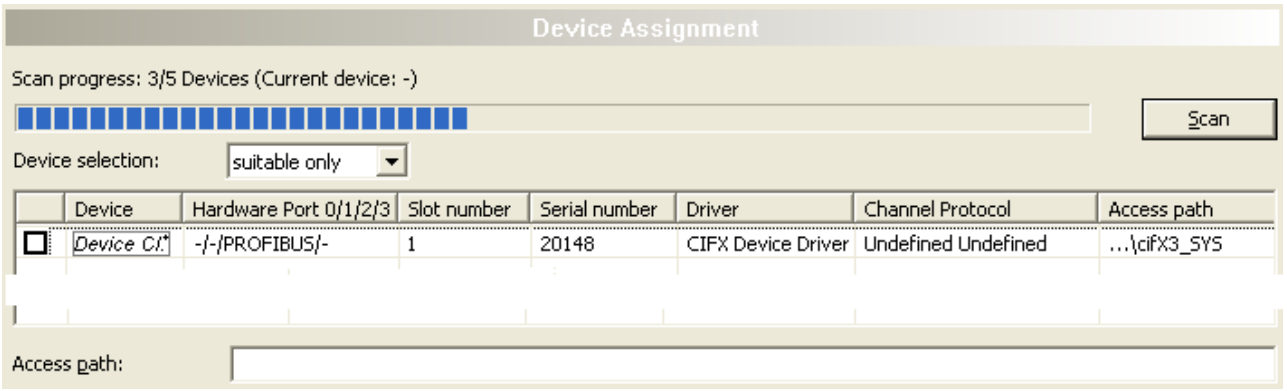


Figure 12: Device Assignment - detected Devices (The name of the device class is displayed.) – Example for a device without firmware*

2. Under **Device Selection** select *suitable only*.
3. Select **Scan**, to start the scanning process.

↬ In the table all devices are displayed, which can be connected to the netSLAVE DTM via the preselected driver.

Note: For devices, which have been found via the **cifX Device Driver** in the column **Access path** the indication …\cifX[*0toN*]**_SYS** is displayed. This is correct, as long as a device did not get a firmware. After the firmware download has been completed, in the column **Access path** the indication …\cifX[*0toN*]**_Ch**[*0to3*] is displayed.

Parameter	Meaning	Range of Value / Default Value
Device selection	Selecting **suitable only** or **all** devices.	suitable only, all
Device	Device class of the Slave devices.	
Hardware Port 0/1/2/3	Shows, which hardware is assigned to which communication interface.	
Slot number	Shows the **Slot Number (Card ID)** preset at the PC card cifX via the **Rotary Switch Slot Number (Card ID)**. The indication **n/a** means that no **Slot-Nummer (Card ID)** exists. This will occur if the PC card cifX is not equipped with a **Rotary Switch Slot Number (Card ID)** or for PC cards cifX equipped with a **Rotary Switch Slot Number (Card ID)** if the rotary switch is set to the value 0 (zero).	1 to 9, n/a
Serial number	Serial number of the device	
Driver	Name of the driver	
Channel Protocol	Shows, which firmware is loaded to which device channel. The data for the used channel consists of the protocol class and the communication class. a.) For devices without firmware: Undefined Undefined, b.) For devices with firmware: Protocol name corresponding to the used Firmware	
Access path (last column on the right)	Depending on the used driver in the column **Access path** different data to the device are displayed. For the **cifX Device Driver** the following data are displayed: a.) For devices without firmware: …\cifX[*0toN*]**_SYS**, b.) For devices with firmware: …\cifX[*0toN*]_**Ch**[*0to3*]. cifX[*0toN*] = Board number 0 to N Ch[*0to3*] = Channel number 0 to 3	Depending on the device and on the driver: board or channel number, IP address or COM interface
Access path (at the lower side of the dialog window)	If in the table a device is checked, under **Access path** (at the lower side of the dialog window) the driver identification or depending on the used driver additional data to the device will be displayed. For the **cifX Device Driver** the following data are displayed: a.) For devices without firmware: …\cifX[*0toN*]**_SYS**, b.) For devices with firmware: …\cifX[*0toN*]_**Ch**[*0to3*]. cifX[*0toN*] = Board number 0 to N Ch[*0to3*] = Channel number 0 to 3	driver identification (ID) depending on the device and on the driver: board or channel number, IP address or COM interface

Table 14: Parameters of the Device Assignment

4.4.1.1 Scanning for all Devices or for suitable only

all

1. Under **Device Selection** select *all*.
2. Select **Scan**.

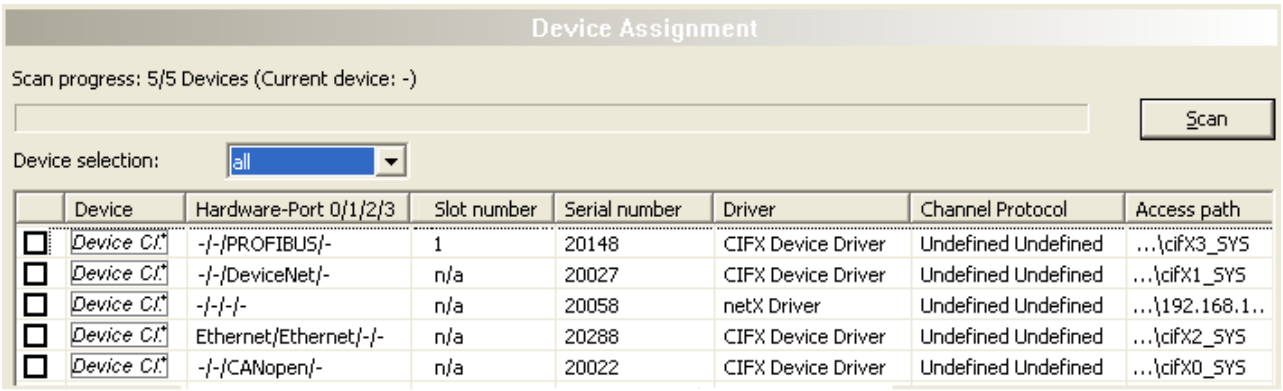


Figure 13: Device Assignment - detected Devices (The name of the device class is displayed.) Example for Devices without Firmware*

↬ In the table all devices are displayed, which are attainable in the network and which can be connected to a single DTM each via the preselected drivers.

Note: During a subsequent firmware download in the selection window **Select Firmware File** all files from the selected folder are displayed, under **Files of Type** „*All Files (*.*)*“ is displayed and the check box **Validate the selected firmware file.** is unchecked.

suitable only

1. Under **Device Selection** select *suitable only*.
2. Select **Scan**.

↬ In the table all devices are displayed, which can be connected to the netSLAVE DTM via the preselected drivers.

Note: During a subsequent firmware download in the selection window **Select Firmware File** only firmware files from the selected folder are displayed, under **Files of Type** „*Firmware Files (*.nxm)*“ or „*Firmware Files (*.nxf)*“ is displayed and the check box **Validate the selected firmware file.** is checked.

4.4.2 Selecting the Device (with or without firmware)

Note: A connection with the netSLAVE DTM can only be established with one Slave device.

To select the physical Slave device (with or without firmware):

1. Check the appropriate device.

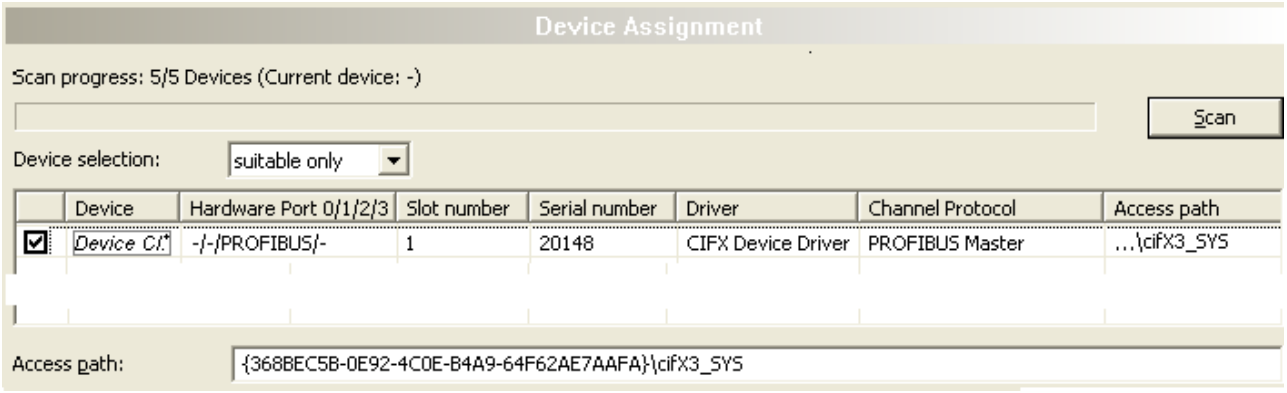


Figure 14: Device Assignment - Selecting the Device (The name of the device class is displayed.) – Example for a device without firmware / one Device is selected*

- Under **Access path** (below in the dialog window) the access path to the device, e. g. the driver identification, or depending on the used driver additional access data of the device are displayed.

2. Select **Apply**, to apply the selection.

Note: Before an online connection from the netSLAVE DTM to the Slave device can be established, a firmware must be loaded to the device and the device must be selected once more.

For further information refer to section to section *Firmware Download* on page 43 or to section *Selecting the Device once more (with Firmware)* on page 41.

4.4.3 Selecting the Device once more (with Firmware)

Note: For repeated download this step is omitted.

To select the Slave device (with firmware and defined system channel) once more, proceed as described hereafter:

all

1. Under **Device Selection** select *all*.
2. Select **Scan**.

↳ In the table all devices are displayed, which are attainable in the network and which can be connected to a DTM via the preselected drivers.

3. Check the appropriate device.

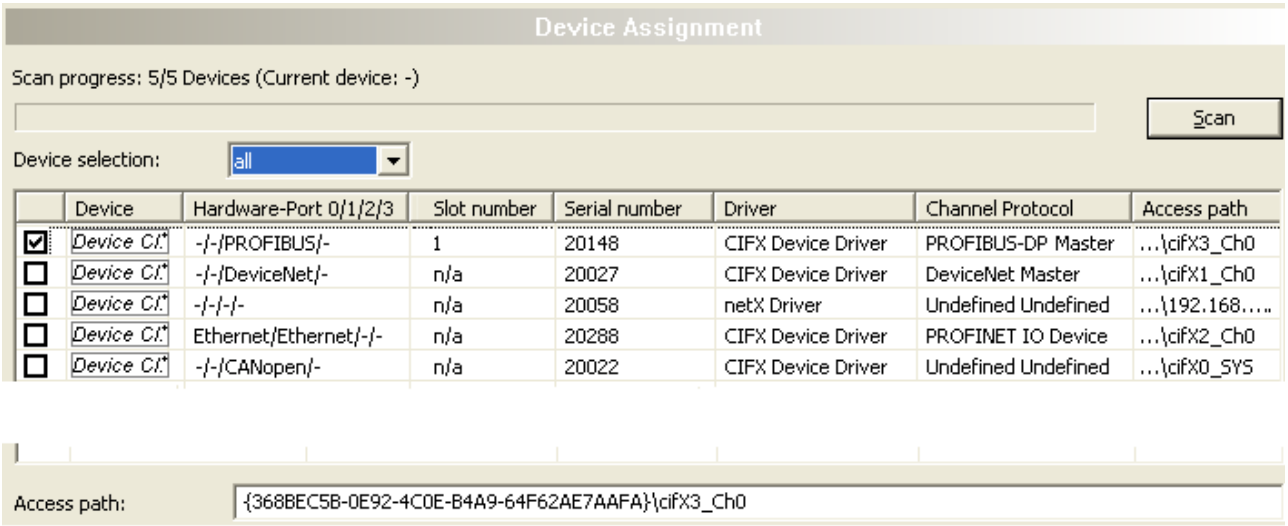


Figure 15: Device Assignment - Selecting the Device (The name of the device class is displayed.) – Example for Devices with and without Firmware / one Device is selected*

Note: After the firmware download has been completed, for the devices which have been detected via the **cifX Device Driver** the following data are displayed:

- In the column **Channel Protocol**: the data for the firmware for the used channel
- In the column **Access path** or under **Access path** (below in the dialog window): the data: …\cifX[*0toN*]_Ch[*0to3*].

  cifX[*0toN*] = board number 0 to N
  Ch[*0to3*] = channel number 0 to 3

4. Select **Apply**, to apply the selection.
5. Or select **OK**, to apply the selection and to close the DTM interface dialog.
6. Connect the DTM to the device using the context menu (right mouse click).

Or:

suitable only

1. Under **Device Selection** select *suitable only*.
2. Select **Scan**.

↬ In the table all devices are displayed, which can be connected to the netSLAVE DTM via the preselected drivers.

3. Check the appropriate device.

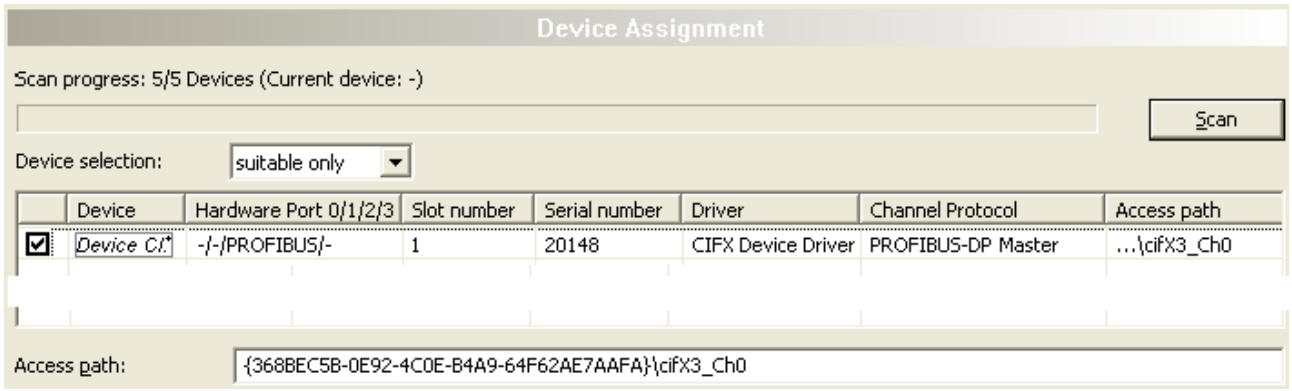


Figure 16: Device Assignment - Selecting the Device (The name of the device class is displayed.) – Example for a device with firmware / one Device is selected*

Note: After the firmware download has been completed, for the devices which have been detected via the **cifX Device Driver** the following data are displayed:

- In the column **Channel Protocol**: the data for the firmware for the used channel
- In the column **Access path** or under **Access path** (below in the dialog window): the data: …\cifX[*0toN*]_Ch[*0to3*].

  cifX[*0toN*] = board number 0 to N
  Ch[*0to3*] = channel number 0 to 3

4. Select **Apply**, to apply the selection.
5. Or select **OK**, to apply the selection and to close the DTM interface dialog.
6. Connect the DTM to the device using the context menu (right mouse click).

For further information how to establish an online connection from the netSLAVE DTM to the Slave device, refer to section *Connecting/Disconnecting Device* on page 118.

4.5 Firmware Download

Using the **Firmware Download** dialog a firmware can be transferred to the device.

Note: Prior to the firmware download, you must select the driver and the Slave device (with or without firmware) and the device must be assigned to the hardware.

For further information refer to section *Overview Settings* on page 24.

To load the firmware to the device:

1. In the navigation area select **Settings > Firmware Download**.

➪ The dialog **Firmware-Download** window is displayed.

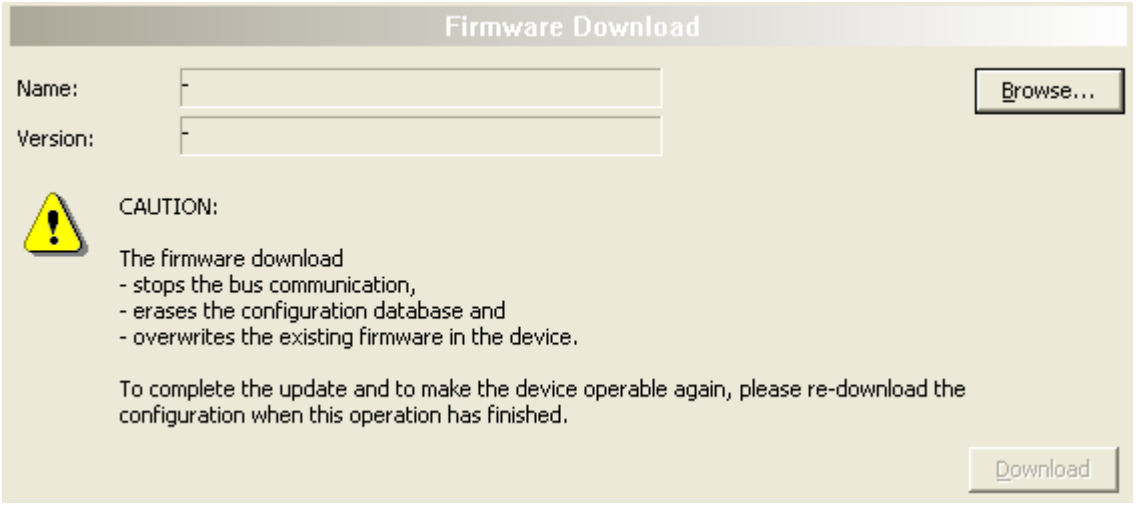


Figure 17: Firmware Download

Element	Meaning
Name	The path and name of the firmware file selected are displayed.
Version	The version and build version of the firmware file selected are displayed.
Browse…	Via ‘Browse…’ you can select the firmware file for the download.
Download	Via ‘Download’ you can download the firmware to the device.

Table 15: Parameter Firmware Download

➢ Select **Browse**.

Device is not assigned to the Hardware

If the device is not assigned to the Hardware, the error message ‘The device is not assigned to the hardware!’ is displayed:

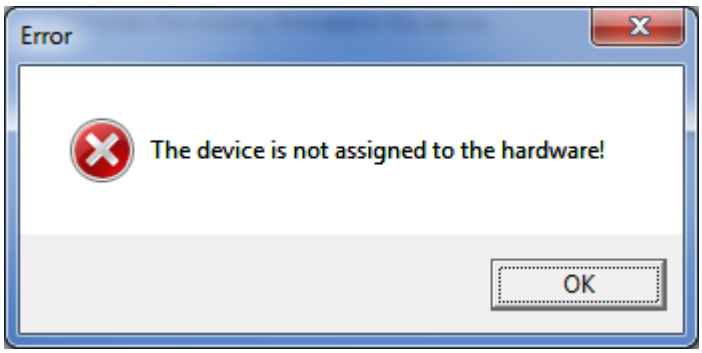


Figure 18: Error Message 'The device is not assigned to the hardware!'

- Click **OK** and select and assign the Master device as described in section *Device Assignment*.

Device is assigned to the Hardware

- The selection window **Select Firmware File** is displayed.
- Enlarge the selection window to view the columns **Hardware** and **Version**.

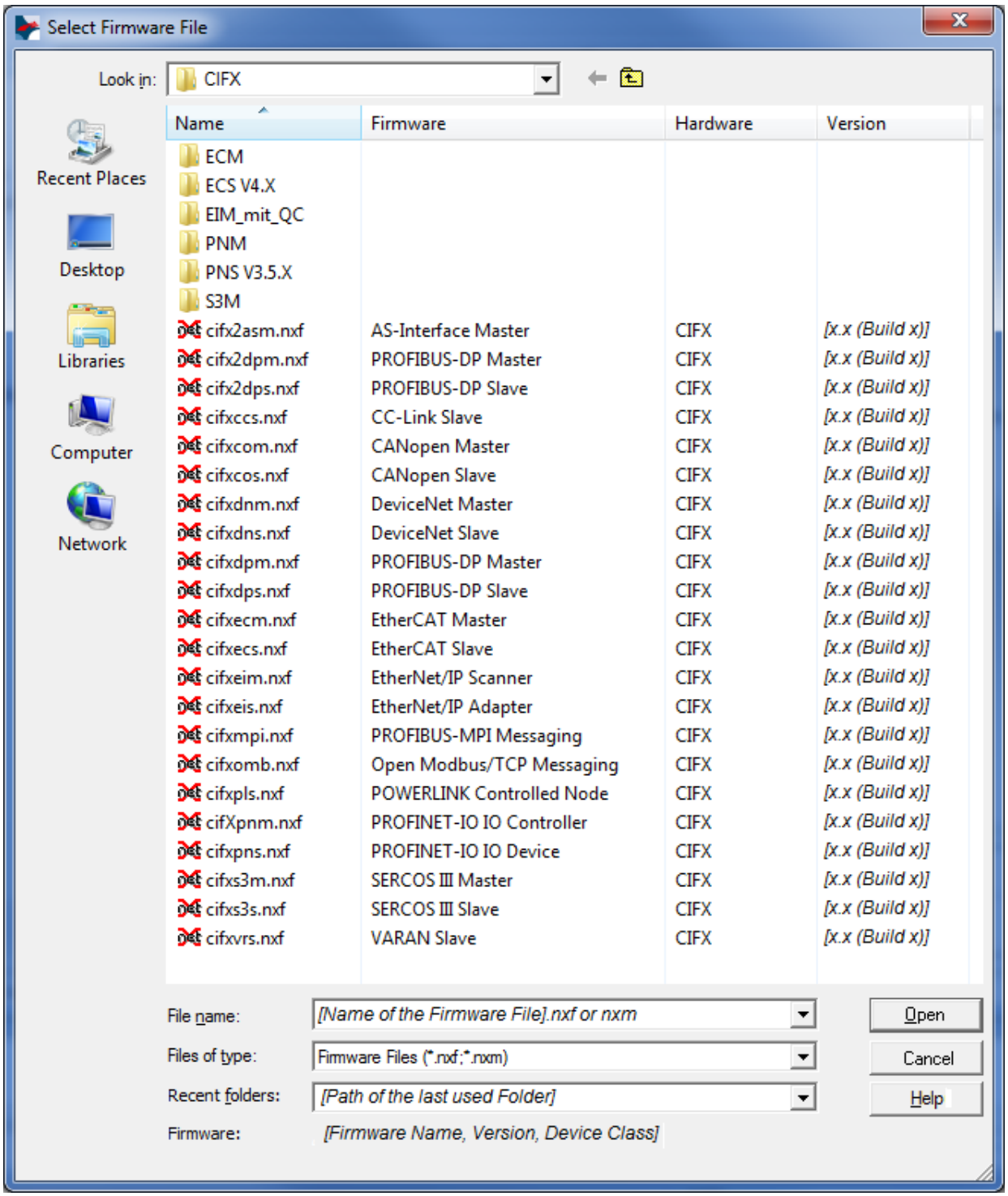


Figure 19: Window 'Select Firmware File' (Example CIFX)

Parameter	Meaning	Range of Value / Default Value
Column **Name**	File name of the firmware file To sort the entries of the window **Select Firmware File** by name click to the column head Name.	nxf, nxm
Column **Firmware**	Name of the firmware (consisting of the protocol name and protocol class)	
Column **Hardware**	Device class of the associated hardware	e. g. CIFX, COMX, COMX 51, NETJACK 10, NETJACK 50, NETJACK 51,

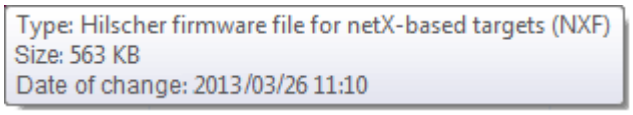
Parameter	Meaning	Range of Value / Default Value
		NETJACK 100, NETTAP 50 (Gateway), NETTAP 100 (Gateway), NETBRICK 100 (Gateway)
Column **Version**	Firmware version	x.x (build x)
Tooltip	To view the tooltip information move with the mouse pointer over the selected firmware line. 	
Files of Type	„*All Files (*.*)*“ if before in the **Device Assignment** window under **Device selection** *all* was selected. „ *Firmware Files (*.nxm)*“ or *Firmware Files (*.nxf)* if before in the **Device Assignment** window under **Device selection** *suitable only* was selected.	*All Files (*.*)*, *Firmware Files (*.nxm)*, *Firmware Files (*.nxf)*
Recent folders	Path of the recently opened folder	
Firmware	As soon as the firmware file has been selected, under **Firmware** the name, the version and the build version as well as the device class for the selected firmware is displayed.	Name, Version, Build Version, Device Class for the selected firmware
Help	Button, to open the online help of the DTM.	

Table 16: Parameters Select Firmware File

Further descriptions to the selection window **Select Firmware File** are included in the context sensitive help (**F1** key) of the Microsoft Corporation.

Note: After in the **Device Assignment** window under **Device selection** *all* or *suitable only* has been set, during a subsequent firmware download in the selection window **Select Firmware File** the following data are displayed or set:

(for list box entry →)	**all**	**suitable only**
In the selection window **Select Firmware File**:	all files from the selected folder	only firmware files from the selected folder
Under **Files of Type***:	„*All Files (*.*)*“	„*Firmware Files (*.nxm)*“, „*Firmware Files (*.nxf)*“
Validation:	A restricted validation will be performed if the selected firmware is applied for the download.	A validation is made whether the firmware file is suitable for the netSLAVE Master DTM.

*These settings in the selection window **Select Firmware File** can also be changed manually.

3. In the selection window mark the firmware file to be loaded using the mouse.

➪ In the selection window under **Firmware** the name and the version of the firmware are displayed.

4. In the selection window select the **Open** button.

Validation

➪ A validation is made, whether the selected firmware file is suitable for the netSLAVE Master device.

Invalid Firmware

NOTICE

Invalid Firmware

Loading invalid firmware files could render your device unusable.

➢ Only proceed with a firmware version valid for your device.

↬ If a firmware file is selected, which is not valid for the assigned device, the request **Select Firmware File** will be displayed.
'Invalid firmware for assigned device!
[*detailed explication*]
Shall firmware file nevertheless be applied for the download?'

Figure 20: Request Select Firmware File - Example Invalid Firmware

➢ Answer to the request with **No** and select a valid firmware.

↬ The selection window is closed.

Valid Firmware

↬ The selection window is directly closed (without dialog).

5. Start firmware upgrade.

⚠ WARNING

Communication Stop caused by Firmware Update, faulty System Operation possible, Overwriting of Firmware or Loss of Device Parameters

Before you initiate a firmware download process, while the bus is still in operation status:

➢ Stop the application program.

➢ Make sure that all network devices are placed in a fail-safe condition.

NOTICE

Firmware Corruption or Loss of Parameters caused by Power Disconnect during Firmware Download

➢ During firmware download process, do not interrupt the power supply to the PC, or to the device and do not perform a reset to the device!

➢ In the dialog window **Firmware Download** click to the **Download** button, to download the firmware.

↬ The request **Do you really want to download the firmware?** is displayed.

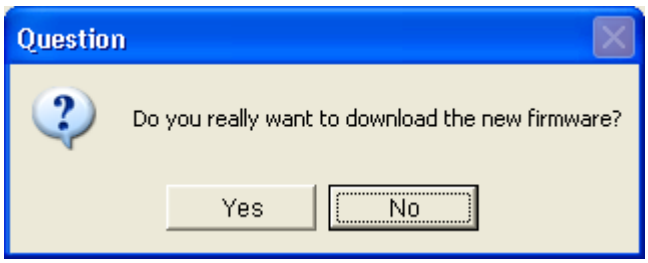


Figure 21: Request - Do you really want to download the firmware?

6. Click **Yes**.

➢ If you are sure, that you have selected the appropriate firmware file answer to the request with **Yes** otherwise with **No**.

↬ During the download a progress bar is displayed ('Download active, device performs initialization…'), in the status line a clock / green hook symbol is displayed and in the dialog window **Firmware Download** Download is grayed out.

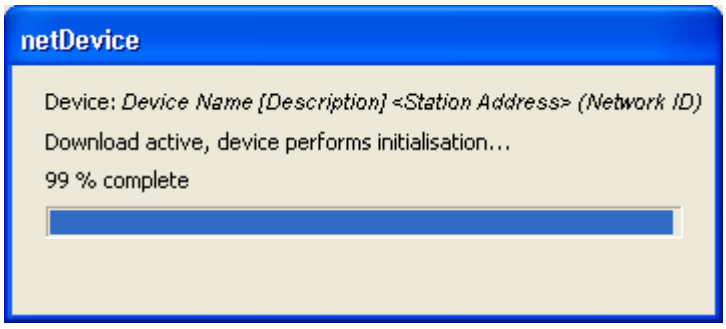


Figure 22: Firmware Download - Progress Bar

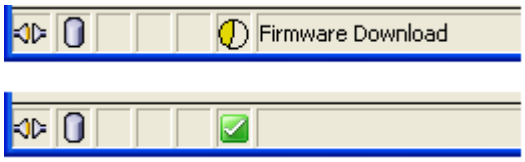


Figure 23: Clock Symbol and Hook Symbol green

- In the **Firmware-Download** dialog window the path and name as well as the version of the selected firmware file are displayed.

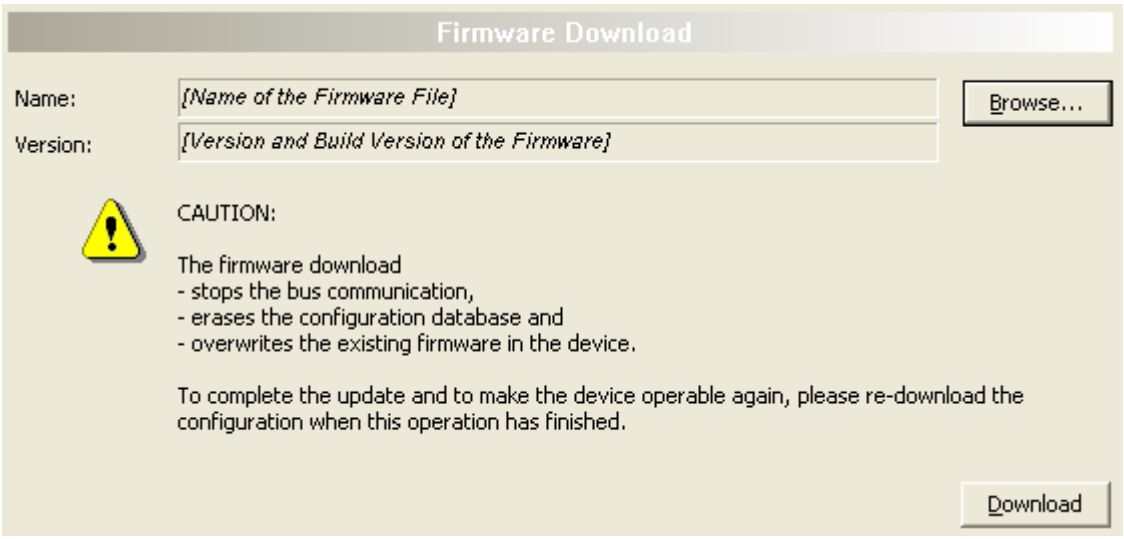


Figure 24: Firmware Download – Download

5 Configuration

5.1 Overview Configuration

The current parameters of the selected firmware protocol are displayed in the **Configuration** window. The configuration parameters can be edited.

➢ Click in the navigation area to the **Configuration** folder to open the **Configuration** window.

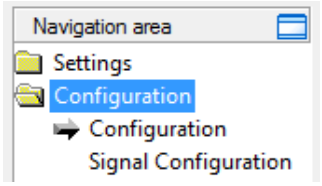


Figure 25: Navigation Area - Configuration

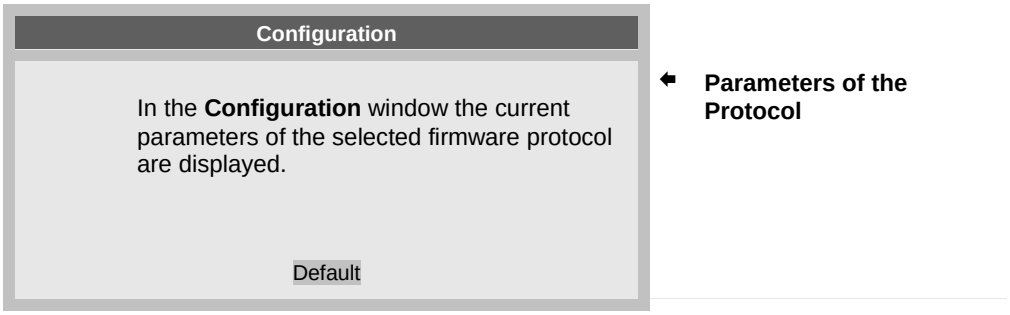


Figure 26: netX Configuration Tool - Configuration Window

The descriptions of the user interface of the single protocols are listed in the table in the next subsequent section.

Default

With the **Default** button the parameters can be reset to the default state.

Enable

If 'Enabled' is unchecked, the default value is used.

Error during Data Input

Error	Action
Parameter validation error! Input field cannot be empty.	Enter data.
Input validation error! Input field accepts only digits.	Enter only digits.
Input validation error! Input field accepts only digits and letters from A to F.	Enter only digits and letters from A to F.

Table 17: Error during Data Input

Gateway Devices

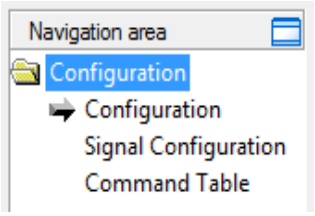


Figure 27: Navigation Area - Configuration for Protocol Conversions

For gateway applications for the configuration of the primary network and secondary network protocols of the different protocol conversions the following dialogs are provided:

- **Configuration**
- **Signal Configuration**
- **Command Table** (for devices with Master functionality)

The corresponding descriptions are listed in the table in the next subsequent section.

Note: In order to transfer the configuration to the Slave device, download the data of the configuration parameters in the Slave device. See section *Download Configuration* on page 120.

5.1.1 Parameters of the Protocol – Overview Configuration Parameters

Any **Parameters of the Protocol** are described in the subsections listed in the table hereafter.

Folder Name	Section	Subsection	Page
Configuration	*Real-Time Ethernet Systems*	*EtherCAT Slave Parameters****	53
		*EtherNet/IP Adapter Parameters****	54
		Open Modbus/TCP Parameters	56
		POWERLINK Controlled Node/Slave Parameters	58
		*PROFINET IO-Device Parameters****	60
		*Sercos Slave Parameters****	61
		VARAN-Client (Slave) Parameters	66
	Fieldbus Systems	*PROFIBUS DP Slave Parameters****	69
		*CANopen Slave Parameters****	71
		*DeviceNet Slave Parameters****	73
		CC-Link Slave Parameters	75
		CompoNet Slave Parameters	77
	Serial Protocols	*Modbus RTU Parameters****	79
		*3964R-Parameter****	81
		*ASCII Settings**	83
		*ASCII Parameters**	83
		*netSCRIPT Settings**	98

*Table 18: Descriptions Configuration Parameters (*only Gateways)*

5.1.2 Overview Command Table and Signal Configuration

The available descriptions about the **Command Table** and **Signal Configuration** for Gateways are listed in the table hereafter.

Folder Name	Section	Subsection	Page
Signal Configuration	*Signal Configuration*		99
Command Table	*Command Table*	*Command Table Modbus RTU*	105
		Command Table Open Modbus/TCP	111

Table 19: Descriptions Command Table and Signal Configuration for Gateways

5.2 Real-Time Ethernet Systems

5.2.1 EtherCAT Slave Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus Startup	Communication starts automatically	Automatic (Default)
Watchdog Time [ms]	This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
I/O Data Status	Status of the input or the output data. For each input and output data the following status information (in Byte) is memorized in the dual-port memory: Status 0 = None (default) Status 1 = 1 Byte (for future use) Status 2 = 4 Byte (for future use)	None, (1 Byte, 4 Byte) Default: None
Ident		
Enabled	If ‘Enabled’ is unchecked, for the single Ident parameters each the respective default value is used.	
Vendor ID	Identification number of the manufacturer	0x00000000 ... 0xFFFFFFFF (hex) Primary Hilscher Vendor ID: 0x00000044 (hex) Default: Secondary Hilscher Vendor ID: 0xE0000044 (hex)
Product Code	Product code of the device as specified by the manufacturer	0x00000000 ... 0xFFFFFFFF (hex), Default: NT 100-RE/ECS: 0x0000000D (hex), NB 100-RE/ECS: 0x00000019 (hex)
Revision Number	Revision number of the device as specified by the manufacturer	0x00000000 ... 0xFFFFFFFF (hex), Default: 0x00000000 (hex)
Serial Number	Serial number of the device	0x00000000 ... 0xFFFFFFFF (hex)
Data		
Input Data Bytes	Length of the input data in Byte	0 … 200 Byte Default: 200 Byte
Output Data Bytes	Length of the output data in Byte	0 … 200 Byte Default: 200 Byte

Table 20: EtherCAT Slave Parameters

5.2.2 EtherNet/IP Adapter Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication starts automatically	Automatic (Default)
Watchdog time [ms]	This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
I/O data status	Status of the input or the output data. For each input and output data the following status information (in Byte) is memorized in the dual-port memory: Status 0 = None (default) Status 1 = 1 Byte (for future use) Status 2 = 4 Byte (for future use)	Default: None
Ident		
Enable	If ‘Enable’ is unchecked, for the single Ident parameters each the respective default value is used.	
Vendor ID	Identification number of the manufacturer	0x00000000 ... 0x0000FFFF (hex), Hilscher: 0x00000011B (hex)
Product Code	Product code of the device as specified by the manufacturer	0x00000000 ... 0x0000FFFF (hex), Default NT 50-EN/EIS: 0x000000113 (hex), NT 100-RE/EIS: 0x00000010F (hex), NB 100-RE/EIS: 0x000000111 (hex)
Product Type	Communication Adapter	0x00000000 ... 0x0000FFFF (hex), Default: 0x00000000C (hex)
Major revision	Major revision of the EtherNet/IP Adapter device according to the EtherNet/IP specification.	0 ... 255, Default: 1
Minor revsion	Minor revsion of the EtherNet/IP Adapter device according to the EtherNet/IP specification.	0 ... 255, Default: 1
Device name	Device name of the device station as character string, e. g. EtherNet/IP Adapter (Slave).	0 - 31 ASCII characters, Examples: NT 50-EN/EIS, NT 100-RE/EIS, NB 100-RE/EIS
Bus		
IP Address	Valid IP Address for the device If ‘Enable’ is unchecked (Default setting), the device obtains its IP Address from a DHCP server or also from a BOOTP server, if this one is checked. If ‘Enable’ is checked, the device uses the manually entered value.	Valid IP address Default: unchecked
Netmask	Valid Network mask for the device If ‘Enable’ is unchecked (Default setting), the device obtains its Netmask from a DHCP server or also from a BOOTP server, if this one is checked. If ‘Enable’ is checked, the device uses the manually entered value.	Valid network mask Default: unchecked
Gateway	Valid Gateway Address for the device If ‘Enable’ is unchecked (Default setting), the device obtains its Gateway Address from a DHCP server or also from a BOOTP server, if this one is checked. If ‘Enable’ is checked, the device uses the manually entered value.	Valid gateway address Default: unchecked

Parameter	Meaning	Range of Value/Value
	There are three methods available, how the device can obtain its IP Address, Netmask and Gateway Address, one of which must be selected. These methods can also be combined. The device performs the following sequence in order to obtain the addresses: 1. from a DHCP server if DHCP is checked (if a DHCP server provides the requested addresses to the device, then the device uses these addresses) 2. from a BootP server if BootP is checked (if a BootP server provides the requested addresses to the device, then the device uses these addresses) 3. the addresses manually set are used. If the IP Address is set manually also the Network Mask must be set manually. The manually set Gateway Address is optional. If no DHCP server and no BootP server and no manually set addresses exist, then the protocol is not ready for initialization or for operation.	
Extras	BootP: If checked, the device obtains its IP Address, Netmask, Gateway Address from a BOOTP server.	Default: unchecked
	DHCP: If checked, the device obtains its IP Address, Netmask, Gateway Address from a DHCP server.	Default: checked
	100Mbit: Speed Selection, If checked, the device will operate at 100 MBit/s, else at 10 MBit/s. This parameter will not be in effect, when auto-negotiation is active.	Default: unchecked
	FullDuplex: Duplex Operation, If checked, full-duplex operation will be used. The device will operate in half-duplex mode, if this parameter is set to zero. This parameter will not be in effect, when auto-negotiation is active.	Default: unchecked
	Auto-neg.: Auto-Negotiation, If checked, the device will auto-negotiate link parameters with the remote hub or switch.	Default: checked
Data		
Produced data length	Maximum allowed length of the input data in Byte. This parameter should be equal to or higher than the complete projected input data length, otherwise the EtherNet/IP device will reject the cyclic communication requests.	0 … 504 Byte Default: 32 Byte
Consumed data length	Maximum allowed length of the output data in Byte. This parameter should be equal to or higher than the complete projected output data length, otherwise the EtherNet/IP device will reject the cyclic communication requests.	0 … 504 Byte Default: 32 Byte

Table 21: EtherNet/IP Adapter Parameters (Part 2)

5.2.3 Open Modbus/TCP Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication starts application controlled or automatic.	Default: Automatic
Watchdog time [ms]	Watchdog time within which the device watchdog must be retriggered from the application program while the application program monitoring is activated. When the watchdog time value is equal to 0 the application program monitoring is deactivated. This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
Protocol mode	Mode of data exchange: 'Client' (Message Mode) or 'IO Server' For the **'Client'** mode in the navigation area 'Command Table' is displayed. For the Open Modbus/TCP Client the Command Table is a list with commands for the reading or writing of data. For the **'IO Server'** mode, the communication partner has *read/write* access to the process data memory of the device from outside using of the function codes.	IO Server (default), Client
Data swap	Data-storage mode: No: Data will not be swapped Yes: Data will be swapped.	Yes, No, Default CIFX RE/OMB, COMX 51XX-RE/OMB, COMX 100XX-RE/OMB, NJ 50X-RE/OMB, NJ 51X-RE/OMB, NJ 100XX-RE/OMB: No, NT 50-EN/OMB, NT 100-RE/OMB, NB 100-RE/OMB: Yes
Map FC1 and FC3	If ‚unchecked', data are read from the intput area using FC1, FC3 and FC23. If checked, data are read from the output area using FC1, FC3 and FC23. FC1 then can be used instead of FC2 and FC3 instead of FC4.	checked, unchecked, Default: unchecked
Bus		
Provided server connections	Number of sockets to provide for server requests* *A value of 0 means that the Open Modbus/TCP task exclusively works as TCP Client. A value of 16 means that the Open Modbus/TCP task exclusively works as Server in the Message-Mode. With the default value 4 there are 4 Server connections provided. Then still up to 12 TCP client connections are available. The parameters 'Send Acknowledgement Timeout' (Send Timeout), 'Connect Acknowledgement Timeout' (Connect Timeout) and 'Close Acknowledgement Timeout' (Close Timeout) are for the Timeout between the Open Modbus/TCP Task and the TCP Task.	Sockets: 0 ... 16, Default: 4
Client connection watchdog time	Only for client jobs in Message Mode (packet mode). The connection to the destination-device stays open, until timeout is expired. **Note**: This timeout starts, after receiving the answer to a command For interface programming via Open Modbus/TCP, the value for Client Connection Watchdog Time (Omb Open Time) must be divided by the factor of 100. (Value range for the interface programming: 1 ... 60,000)	100 ... 6,000,000 ms, Default: 1000 ms
Response timeout	Only for client jobs in Message Mode (packet mode). After expiration of this time, the job will be canceled and an error is send to the application. Note: This timeout starts after command is send to the destination device via TCP. For interface programming via Open Modbus/TCP, the value for Response Timeout must be divided by the factor of 100. (Value range for the interface programming: 1 ... 60,000)	100 ... 6,000,000 ms, Default: 2,000 ms
Send acknowledge-ment timeout*	Parameter for TCP task (in milliseconds) . Used OMB task internal. It specifies the timeout for trying to send messages via TCP/IP If the value 0 is selected, the default value of 31,000 milliseconds is used.	0 ... 2,000,000,000 ms, Default: 31,000 ms

Parameter	Meaning	Range of Value/Value
Connect acknowledge-ment timeout*	Parameter for TCP task (in milliseconds). Used OMB task internal. It specifies the timeout for trying to establish a connection with the TCP task. If the value 0 is selected, the default value of 31,000 milliseconds is used.	0 ... 2,000,000,000 ms, Default: 31,000 ms
Close acknowledge-ment timeout*	Parameter for TCP task (in milliseconds). Used OMB task internal. It specifies the timeout for trying to close a connection with the TCP task. If the value 0 is selected, the default value of 13,000 milliseconds is used.	0 ... 2,000,000,000 ms, Default: 13,000 ms
IP address	Valid IP address for the device If 'Enable' is unchecked (Default setting), the device obtains its IP Address from a DHCP or BOOTP server. If 'Enable' is checked, the device uses the manually entered value.	Valid IP address Default: unchecked
Net mask	Valid Network mask for the device If 'Enable' is unchecked (Default setting), the device obtains its Netmask from a DHCP or BOOTP server. If 'Enable' is checked, the device uses the manually entered value.	Valid network mask Default: unchecked
Gateway	Valid Gateway address for the device If 'Enable' is unchecked (Default setting), the device obtains its Gateway Address from a DHCP or BOOTP server. If 'Enable' is checked, the device uses the manually entered value.	Valid gateway address Default: unchecked
	There are three methods available, how the device can obtain its IP Address, Netmask and Gateway Address, one of which must be selected. These methods can also be combined. The device performs the following sequence in order to obtain the addresses: 1. from a DHCP server if DHCP is checked (if a DHCP server provides the requested addresses to the device, then the device uses these addresses) 2. from a BootP server if BootP is checked (if a BootP server provides the requested addresses to the device, then the device uses these addresses) 3. the addresses manually set are used. If the IP Address is set manually also the Network Mask must be set manually. The manually set Gateway Address is optional. If no DHCP server and no BootP server and no manually set addresses exist, then the protocol is not ready for initialization or for operation.	
Extras	BootP: If checked, the device obtains its IP Address, Netmask, Gateway Address from a BOOTP server.	Default: unchecked.
	DHCP: If checked, the device obtains its IP Address, Netmask, Gateway Address from a DHCP server.	Default: checked.

Table 22: Open Modbus/TCP Parameters

Note: *This parameter is only applicable for client jobs in Message-Mode and does not work in in IO Server mode and Command table mode.

5.2.4 POWERLINK Controlled Node/Slave Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication start application controlled or automatic	Default: Automatic
Watchdog time [ms]	Watchdog time within which the device watchdog must be retriggered from the application program while the application program monitoring is activated. When the watchdog time value is equal to 0 the application program monitoring is deactivated. This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
I/O data status	Status of the input or the output data. For each input and output data the following status information (in Byte) is memorized in the dual-port memory: Status 0 = None (default) Status 1 = 1 Byte (for future use) Status 2 = 4 Byte (for future use)	Default: None
Disable host-triggered input data exchange	Decides, whether host-triggered update for input data exchange is enabled (hook checked) or disabled (hook unchecked).	Default: unchecked
Disable host-triggered output data exchange	Decides, whether host-triggered update for output data exchange is enabled (hook checked) or disabled (hook unchecked).	Default: unchecked
Ident		
Enable	If ‘Enable’ is unchecked, for the single Ident parameters each the respective default value is used.	
Vendor ID	Identification number of the manufacturer	0x00000000 ... 0xFFFFFFFF (hex), Hilscher: 0x00000044 (hex)
Serial number	Serial number of the device	0x00000000 ... 0xFFFFFFFF (hex), Default: 0x00000000 (hex)
Product code	Product code of the device as specified by the manufacturer	0x00000000 ... 0xFFFFFFFF (hex), Default: CIFX RE/PLS: 0x00000001 (hex), COMX 51XX-RE/PLS: 0x0000001C (hex), COMX 100XX-RE/PLS: 0x00000003 (hex), NJ 50X-RE/PLS: 0x00000023 (hex), NJ 51X-RE/PLS: 0x0000001B (hex), NJ 100XX-RE/PLS: 0x00000024 (hex), NXSTK 50-RE/PLS: 0x0000000B (hex), NIC 50-RE/PLS: 0x00000008 (hex), NETX 500 RE/PLS: 0x00000106 (hex), NETX 100 RE/PLS: 0x00000109 (hex) , NETX 50 RE/PLS: 0x00000107 (hex),

Parameter	Meaning	Range of Value/Value
Product code (*continued*)		NT 100-RE/PLS: 0x00000018 (hex), NB 100-RE/PLS: 0x00000017 (hex)
Revision number	Revision number of the device as specified by the manufacturer	0x00000000 ... 0xFFFFFFFF (hex), Default: 0x00000000 (hex)
Bus		
Node Id	EPL Node ID (EPL = Ethernet POWERLINK)	1...239, Default: 1
DNS node name	DNS-compatible name of the POWERLINK Controlled Node/Slave (optional)	
Gateway address	Gateway address for IP stack	Default: 192.168.100.254
Data		
Input data bytes	Length of the input data in byte	1… 1490 Byte, Default: 4 Byte
Output data bytes	Length of the output data in byte	1… 1490 Byte, Default: 4 Byte
Disable PDO mapping version check	Decides, whether PDO mapping version is checked (hook checked) or not checked (hook unchecked). **Select for netTAP and netBRICK devices always unchecked.**	Default: checked
Configure default objects	Decides, whether default objects are created (hook checked) or not (hook unchecked). If the objects will be created, the old set of previously existing objects will be cleared	Default: checked
Delete application objects	Decides, whether application objects are deleted (hook checked) or not (hook unchecked). When 'Configure Default Objects' is checked, then the firmware automatically does a 'Delete Application Objects'.	Default: checked

Table 23: POWERLINK Controlled Node/Slave Parameters

Note: To configure the Managing Node/Master, an XDD file (device description file) is required. The settings in the used Managing Node/Master must comply with the settings in the Controlled Node/Slave, to establish communication. Important parameters are: Vendor ID, Product Code, Serial Number, Revision Number, Node ID, Output and Input length.

5.2.5 PROFINET IO-Device Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication starts automatic	Default: Automatic
Watchdog time [ms]	This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
I/O data status	Status of the input or the output data. For each input and output data the following status information (in Byte) is memorized in the dual-port memory: Status 0 = None (default) Status 1 = 1 Byte (for future use) Status 2 = 4 Byte (for future use)	Default: None
Ident		
Enable	If 'Enable' is unchecked, for the single Ident parameters each the respective default value is used.	
Vendor ID	Identification number of the manufacturer, assigned by PROFIBUS Nutzerorganisation e. V.	0x00000000 ... 0xFFFFFFFF (hex), Hilscher: 0x00000011E (hex)
Device ID	Identification number of the device, fixed for every device.	0x00000000 ... 0x0000FFFF (hex), Default: NT 50-EN/PNS (Gateway): 0x0000010F (hex), NT 100-RE/PNS (Gateway) 0x00000010B (hex), NT 100_RE/PNS (Proxy) 0x000000106 (hex), NB 100-RE/PNS (Gateway) 0x00000010E (hex)
Device type	Description of the device type, freely eligible	Character string, 0 ... 25 characters
Order ID	Hilscher device number or order description of the customer for its device	Character string, 0 ... 20 characters
Name of station	Station name of the PROFINET IO-Device station. It has to match the station name configured in the PROFINET IO-Controller for this device. Must be DNS compatible name.	Character string, 1 ... 240 characters Default: NT 50-EN/PNS (Gateway): nt50enpns, NT 100-RE/PNS (Gateway) nt100repns, NT 100_RE/PNS (Proxy) nt100reproxy, NB 100-RE/PNS (Gateway) nb100repns
Type of station	Type name of the PROFINET station; name can be assigned freely.	Character string, 1 ... 240 characters Default: Default.Station.Type
Data		
Output data bytes	Maximum allowed length of the input data in Byte. This parameter should be equal to or higher than the complete projected input data length, otherwise the IO Device will reject the cyclic communication requests.	0 … 1024 Byte Default: 128 Byte
Input data bytes	Maximum allowed length of the output data in Byte. This parameter should be equal to or higher than the complete projected output data length, otherwise the IO Device will reject the cyclic communication requests.	0 … 1024 Byte Default: 128 Byte

Table 24: PROFINET IO-Device Parameters

5.2.6 Sercos Slave Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication start application controlled or automatic	Default: Automatic
Watchdog time [ms]	This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, Default = 1000 ms, 0 = Off
Ident		
Device ID	Identification number of the device. This is a unique device ID managed by the vendor and identifies the component number. The device ID is fixed for every device. Device ID according to the Sercos third generation specification as defined in IDN S-0-1300.x.05. The device ID can be changed by the user, if an other device ID than the default is useful for the used Sercos network. **Note:** The **Device ID** is part of the SDDML device description file. If you use a Sercos Master which is using SDDML files for configuration and the default device ID was changed, then you have to export via **Export SDDML** a new updated SDDML file from the Slave configuration software and import this SDDML file into the configuration software for the Sercos Master.	Default: NT_100-RE_S3S_FIXCFG, NB_100-RE_S3S_FIXCFG
Vendor code	Identification number of the manufacturer, assigned by Sercos International. Vendor code according to the Sercos third generation specification as defined in IDN S-0-1300.x.03. **Note:** The **Vendor code** is part of the SDDML device description file. If you use a Sercos Master which is using SDDML files for configuration and the default vendor code was changed, then you have to export via **Export SDDML** a new updated SDDML file from the Slave configuration software and import this SDDML file into the configuration software for the Sercos Master.	0x00000000 ... 0xFFFFFFFF (hex), Hilscher: 0x0000003E8 (hex)
Version of SCP Sync	Version of the Sercos Communication Profile SYNC 0 = SYNC deactivated. With this setting the configuration parameter for Sercos Communication Profile Class SCP_Sync are not relevant and are displayed grayed out. 1 = SYNC activates version 1. If the Sercos Master uses the telegram sequence MDT, then NRT and then AT, the gateway device requires that the value is set to 1.	0 ... 255, Default: 0 Possible values: 0, 1
Version of SCP NRT	Version of the Sercos Communication Profile NRT 0 = NRT deactivated. With this setting the configuration parameter for IP communication are not relevant and are displayed grayed out. 1 = NRT version 1 is activated. Use always value 0, because the NRT channel is not useable with gateway devices.	0 ... 255, Default: 0 Possible values: 0, 1
Bus		
	Configuration parameter for IP communication	
Enable	If ‘Enable’ is unchecked, for the single bus parameters each the respective default value is used.	checked, unchecked
IP address	Valid IP address for the device If ‘Enable’ is unchecked (Default setting), the device obtains its IP address from a DHCP or BOOTP server. If ‘Enable’ is checked, the device uses the manually entered value. This function is not supported.	Valid IP address, Default for ‘Enable’: unchecked

Parameter	Meaning	Range of Value/Value
Netmask	Valid Network mask for the device If ‘Enable’ is unchecked (Default setting), the device obtains its Netmask from a DHCP or BOOTP server. If ‘Enable’ is checked, the device uses the manually entered value. This function is not supported.	Valid network mask, Default for ‘Enable’: unchecked
Gateway	Valid Gateway address for the device If ‘Enable’ is unchecked (Default setting), the device obtains its Gateway Address from a DHCP or BOOTP server. If ‘Enable’ is checked, the device uses the manually entered value. This function is not supported.	Valid gateway address, Default for ‘Enable’: unchecked
	There are three methods available, how the device can obtain its IP address, Netmask and Gateway Address, one of which must be selected. These methods can also be combined. The device performs the following sequence in order to obtain the addresses: 1. from a DHCP server if DHCP is checked (if a DHCP server provides the requested addresses to the device, then the device uses these addresses) 2. from a BootP server if BootP is checked (if a BootP server provides the requested addresses to the device, then the device uses these addresses) 3. the addresses manually set are used. If the IP address is set manually also the Network Mask must be set manually. The manually set Gateway Address is optional. If no DHCP server and no BootP server and no manually set addresses exist, then the protocol is not ready for initialization or for operation.	
TCP flags	BootP: If checked, the device obtains its IP address, Netmask, Gateway Address from a BOOTP server. This function is not supported.	checked, unchecked, Default: unchecked
	DHCP: If checked, the device obtains its IP address, Netmask, Gateway Address from a DHCP server. This function is not supported.	checked, unchecked, Default: unchecked
	Configuration Data Parameters of the Sercos Communication Profile Class SCP_Sync	
ConClk pulse length*	Control Clock pulse length: This timing parameter defines the time how long the communication synchronized hardware output signal CON_CLK is set. The maximum valid value of the „ConClk pulse length“ depends on the configured cycle time. In general the signal CON_CLK must go to inactive again before the next cycle starts.	>= 1000 ns, Default: 1000 ns
DivClk pulse distance *	Divided Control Clock pulse distance: This timing parameter defines the delay time from the start of the communicatin cycle to the first pulse of the communication synchronized hardware output signal DIV_CLK.	0 ... 16.777.210 ns, Default: 1000 ns
DivClk Delay*	Divided Control Clock delay: This timing parameter defines the delay time respectively the distance between two pulses of the communication synchronized hardware output signal DIV_CLK. In the „DivClk mode“Mode 1 this parameter is ignored.	0 ... 6.710.860 ns, Default: 1000 ns
DivClk pulse length*	Divided Control Clock pulse length: This timing parameter defines the pulse length, i. e. how long the communication synchronized hardware output signal DIV_CLK is set.	1000 ... 20.000 ns, Default: 1000 ns
	*All timing values are handled as multiple of 10 ns (e. g. ConClk pulse length = 1005 ns is handled as 1000 ns).	

Parameter	Meaning	Range of Value/Value
DivClk multiplier	Divided Control Clock multiplier: For „DivClk mode“ Mode 0: Here the parameter indicates the number of pulses of the communication synchronized hardware output signal DIV_CLK within a communication cycle. For „DivClk mode“Mode 1: Here the parameter indicates the number of communication cycles.	1 ... 255, Default: 100
DivClk polarity	Divided Control Clock Polarity: This communication synchronization flag defines the output state of the communication synchronized hardware output signal DIV_CLK.	Disabled, High Activity, Low Activity, Default: High Activity
DivClk mode	Divided Control Clock Mode: This communication synchronization flag defines the mode of the communication synchronized hardware output signal DIV_CLK. Mode 0: The DIV_CLK signal becomes active several times within a communication cycle. The following condition must be fulfilled in mode 0: „DivClk pulse length“ + 100 <= „DivClk Delay“. Mode 1: The DIV_CLK signal becomes active once after N communication cycles.	Mode 0, Mode 1, Default: Mode 0
ConClk polarity	Control Clock Polarity: This communication synchronization flag defines the output state of the communication synchronized hardware output signal CON_CLK.	Disabled, High Activity, Low Activity, Default: Disabled
Slave Configuration		
Number of slaves	The number of used Sercos addresses Fix setting for Gateway devices.	For NT 100-RE/S3S, NB 100-RE/S3S: 1, Default: 1
Slave 1		
Sercos address	Address for the Sercos Slave. The address range is from 1 to 511.	1 … 511, Default: 1
FSP type	Function Specific Profile Type according to IDN S-0-1302.x.01 FSP type & Version: The FSP type indicates the function specific type of the resource. Supported FSP profiles: FSP_IO, FSP_Drive FSP IO: A specification for IO Devices to be controlled by Sercos, FSP Drive: A specification for mechanical drives to be controlled by Sercos	IO V1, Drive V1, Default: IO V1
SCP configuration type	SCP config type according to Sercos specification third generation: 0x0101 - SCP_FixCFG Version 1.1.1, 0x0201 - SCP_VarCFG Version 1.1.1, 0x0202 - SCP_VarCFG Version 1.1.3 According to the Sercos specification third generation the Sercos Communication Profiles Classes SCP-FixCFG and SCP-VarCFG define the basic communication and are mutually exclusive. SCP-FixCFG: Exact two connections are supported, one as consumer and one as producer. The content of the connections is defined by the Slave and cannot be changed by the Master. SCP-VarCFG: A certain number of connections is supported. The Slave defines this number and provides it to the Master. The content of all connections has to be configured (e.g. by the Master). The Slave provides lists of IDNs, that can be cyclically produced and consumed, so the Master can find it out.	Fix. Version 1.1.1, Var. Version 1.1.1, Var. Version 1.1.3 Default Fix. Version 1.1.1
Output data size for FixCfg	Output data size for Fixed Configuration in Byte for „SCP configuration type“ / „Fix. Version 1.1.1“ **Note:** The output data size is part of the SDDML device description file. If you use a Sercos Master which is using SDDML files for configuration and the default output data size was changed, then you have to export a new updated SDDML file from the Slave configuration software and import this SDDML file into the configuration software for the Sercos Master.	0 … 200 Byte, Default: 2 Byte

Parameter	Meaning	Range of Value/Value
Input data size for FixCfg	Input data size for Fixed Configuration in Byte for „SCP configuration type“ / „Fix. Version 1.1.1“ **Note:** The input data size is part of the SDDML device description file. If you use a Sercos Master which is using SDDML files for configuration and the default input data size was changed, then you have to export a new updated SDDML file from the Slave configuration software and import this SDDML file into the configuration software for the Sercos Master.	0 … 200 Byte, Default: 2 Byte
User SCP types	Up to 20 User SCP types are possible. The following User SCP types are already defined: 0x0401 - SCP_WD Version 1.1.1 for monitoring connections, 0x0501 - SCP_Diag Version 1.1.1 for bus-diagnosis, 0x0601 - SCP_RTB Version 1.1.1 for using Real time bits, 0x0901 - SCP_Mux Version 1.1.1 for multiplexed cyclic data, 0x0B01 - SCP_Sig Version 1.1.1 for using signal status/control words For gateway devices no user SCP types are supported. Therefore do not select any type.	SCP_WD Version 1.1.1, SCP_Diag Version 1.1.1, SCP_RTB Version 1.1.1, SCP_Mux Version 1.1.1, SCP_Sig Version 1.1.1
Slave flags	Use a default object dictionary For gateway devices check always.	For NT 100-RE/S3S, NB 100-RE/S3S: checked, unchecked, Default: checked
	Delete objects created by application on reset For gateway devices check always.	For NT 100-RE/S3S, NB 100-RE/S3S: checked, unchecked, Default: checked
	Master is not allowed to change Sercos address: enables (unchecked) or disables (checked) that the Master changes the Sercos address. For gateway devices check always.	For NT 100-RE/S3S, NB 100-RE/S3S: checked, unchecked, Default: checked
Slave Connections 1, 2, 3 and 4		
Connection Control offset	Connection Control Offset for the Slave connections 1, 2, 3 and 4 Connection Control Offset 1 (slave -> master): For gateway/proxy devices use 0, Connection Control Offset 2 (master -> slave): For gateway/proxy devices use 0, Connection Control Offset 3 (not supported): For gateway/proxy devices use 0, Connection Control Offset 4 (not supported): For gateway/proxy devices use 0.	0 ... 5.758, Default: 0
Real Time Data offset	Real Time Data Process Image Offset for the Slave connections 1, 2, 3 and 4 Real Time Data Offset 1 (slave -> master): For gateway/proxy devices use 2, Real Time Data Offset 2 (master -> slave): For gateway/proxy devices use 2, Real Time Data Offset 3 (not supported): For gateway/proxy devices use 0, Real Time Data Offset 4 (not supported): For gateway/proxy devices use 0.	0 ... 5.760, Default: 2
Maximum allowed Real Time Data length	Maximum allowed Real Time Data length for the Slave connections 1, 2, 3 and 4.	0 … 126 Byte, Default NT 100-RE/S3S, NB 100-RE/S3S: Connections 1 to 4: 126 Byte

Table 25: Sercos Slave Parameter

Note: To configure the Master, an XML file (device description file) is required. The settings in the used Master must comply with the settings in the Slave to establish communication.

5.2.7 VARAN-Client (Slave) Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication start application controlled or automatic	Application controlled, Automatic; Default: Automatic
Client watchdog time [ms]	Watchdog time within which the device watchdog must be retriggered from the application program while the application program monitoring is activated. When the watchdog time value is equal to0 the application program monitoring is deactivated.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
I/O data status*	Status of the input or the output data. *(not supported)	
Ident		
Check box	If the box is unchecked, for the single Ident parameters each the respective default value is used.	Checked, unchecked, Default: unchecked
Vendor ID	Vendor Identifier: Identification number of the manufacturer. This value is specific for every single vendor.	0x00000000 ... 0xFFFFFFFF (hex), Hilscher: 0x0000001D(hex)
Vendor name	Name of the manufacturer, specific for every single vendor.	Character string, 0 ... 64 characters, Default: Hilscher GmbH
Device ID	Device Identifier: Identification number of the device as indicated by the VARAN-BUS-NUTZERORGANISATION. This value is specific for every single device type and is fixed for every device. In case of an unknown device ID the manager rejects the client.	0x00000000 … 0xFFFFFFFF (hex), Default for CIFX RE/VRS: 0x0000048C (hex), COMX 51XX-RE/VRS: 0x00000000 (hex), COMX 100XX-RE/VRS: 0x0000048A (hex), NETX 50 RE/VRS: 0x000003FD (hex), NETX 100 RE/VRS: 0x000003FD (hex), NETX 500 RE/VRS: 0x000003FD (hex), NJ 50X-RE/VRS: 0x00000488 (hex); NJ 51X-RE/VRS: 0x00000000 (hex); NJ 100XX-RE/VRS: 0x00000489 (hex)
Device name	Name of the device as indicated by the VARAN-BUS-NUTZERORGANISATION.	Character string, 0 ... 64 characters, Default: CIFX RE/VRS, COMX RE/VRS, NETX 50 RE/VRS, NETX 100 RE/VRS, NETX 500 RE/VRS, NJ50x-RE/VRS; NJ100xx-RE/VRS
Product version	Version of the device as specified by the manufacturer.	0x00000000 ... 0xFFFFFFFF (hex), Default: 0x00000000 (hex)
Order number	Order number of the device as specified by the manufacturer.	0x00000000 ... 0xFFFFFFFF (hex) , Default: 0x00000000 (hex)

Parameter	Meaning	Range of Value/Value
Serial number	Serial number of the device as specified by the manufacturer.	0x00000000 ... 0xFFFFFFFF (hex), Default: 0x00000000 (hex)
License number	License number of the device as specified by the VARAN-BUS-NUTZERORGANISATION.	0x00000000 ... 0xFFFFFFFF (hex) , Default: 0x00000000 (hex)
Bus		
Client watchdog time	Client watchdog time in ms. The Client watchdog time is the maximum cyclic communication timeout allowed, before Client signals a bus error. Example: Client watchdog time set to 50ms and Client is unplugged from the network (or the Manager is set into stop mode), then the Client goes in an error state on the 50th ms after this event.	[0 … 130] ms, Default = 130 ms, 0 = Off
Memory area 2*	If ‘Memory area 2’ is checked, the parameters for Memory area 2 are enabled. (*currently not supported, set to default value)	Checked, unchecked, Default: unchecked
EMAC*	EMAC = Extended MAC-Address If ‘EMAC’ is checked, an integrated EMAC for IP data exchange with client application is used. (*currently not supported, set to default value)	Checked, unchecked, Default: unchecked
Memory area 1 / Memory area 2		
Read size / Memory area 1	Reading size of the memory area 1	0 … 128 Byte, Default: 128 Byte
Read size / Memory area 2*	Reading size of the memory area 2 *(not supported)	0 … 128 Byte, Default: 0 Byte
Write size / Memory area 1	Writing size of the memory area 1	0 … 128 Byte, Default: 128 Byte
Write size / Memory area 2*	Writing size of the memory area 2 *(not supported)	0 … 128 Byte, Default: 0 Byte
Read offset / Memory area 1	Reading offset of the memory area 1	0x00000000 ... 0x0000FFFF (hex), Default: 0x00002000 (hex)
Read offset / Memory area 2*	Reading offset of the memory area 2 *(not supported)	0x00000000 ... 0x0000FFFF (hex) , Default: 0x0000FFFF (hex)
Write offset / Memory area 1	Writing offset of the memory area 1	0x00000000 ... 0x0000FFFF (hex) , Default: 0x00002000 (hex)
Write offset / Memory area 2*	Writing offset of the memory area 2 *(not supported)	0x00000000 ... 0x0000FFFF (hex) , Default: 0x0000FFFF (hex)
Sync OUT 0 / Sync OUT 1		

Note!
Only the following 3 combinations between SyncOut0Mode + SyncOut1Mode are reasonable:
"Time for IN/OUT valid (TIO)" + "Disable",
"Time for IN valid (TI)" + "Time for OUT valid (T0)",
"Time for OUT valid (TO)" + "Time for IN valid (TI)"

Parameter	Meaning	Range of Value/Value
Mode / Sync OUT 0	The "SyncOut 0 Mode“ applies the "SYNC OUT 0 ID” to the output of the SYNC Out 0 signal. Using the setting - "Time for IN Valid (TI)“ for the output of the SYNC Out 0 signal the time for the data IN is valid, - "Time for OUT Valid (TO)“ for the output of the SYNC Out 0 signal the time for the data OUT is valid, - "Time for IN/OUT Valid (TIO)“ for the output of the SYNC Out 0 signal the time for the data IN/OUT is valid.	Time for IN/OUT Valid (TIO), Time for IN Valid (TI), Time for OUT Valid (TO), Default: Time for IN/OUT Valid (TIO)
Mode / Sync OUT 1	The "SyncOut 1 Mode “ applies the "SYNC OUT 1 ID” to the output of the SYNC Out 1 signal. Using the setting - "Disable“ means the output of the SYNC Out 1 signal is not available, - "Time for IN Valid (TI)“ for the output of the SYNC Out 1 signal the time for the data IN is valid, - "Time for OUT Valid (TO)“ for the output of the SYNC Out 1 signal the time for the data OUT is valid.	Disable, Time for IN Valid (TI), Time for OUT Valid (TO), Default: Disable
Output / Sync OUT 0	The “Sync out 0 flag/Output” enables / disables the output of the Memory area 1 SYNC Out 0 signal.	Enable, Disable, Default: Enable
Output / Sync OUT 1	The “Sync out 1 flag/Output” enables / disables the output of the Memory area 2 SYNC Out 1 signal.	Enable, Disable, Default: Disable
Polarity / Sync OUT 0	The “Sync out 0 flag/Polarity” defines the active high/low behavior of the SYNC Out 0 signal.	Active Low, Active High, Default: Active Low
Polarity / Sync OUT 1	The “Sync out 1 flag/Polarity” defines the active high/low behavior of the Memory area 2 SYNC Out 1 signal.	Active Low, Active High, Default: Active Low
Pulse length (x10ns)	Pulse length of the SYNC Out signal in 10ns steps. (e.g. the value 100 results as 10ns*100=1000ns=1µs pulse)	0 … 2.147.483.647, Default: 100
Data		
Recv data length	Length of the received data in Byte.	Default: 128 Byte
Send data length	Length of the send data in Byte.	Default: 128 Byte

Table 26: VARAN Client (Slave) Parameters

Note: The settings in the used VARAN Manager (Master) must comply with the settings in the Client to establish communication. Important parameters are: Station Name, Vendor ID, Device ID, Recv Data Length and Send Data Length and Memory area Read offset / Write offset.

5.3 Fieldbus Systems

5.3.1 PROFIBUS DP Slave Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication start application controlled or automatic	Automatic (Default)
Watchdog time [ms]	This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
I/O data status	Status of the input or the output data. For each input and output data the following status information (in Byte) is memorized in the dual-port memory: Status 0 = None (default) Status 1 = 1 Byte (for future use) Status 2 = 4 Byte (for future use)	Default: None
Ident		
Ident number	PROFIBUS Identification Number	0x00000000 ... 0x0000FFFF (hex), Default: NT 50 DP/DPS: 0x000000C99 (hex), NT 100 DP/DPS: 0x000000C0E (hex), NB 100 DP/DPS: 0x00000C9C (hex)
Enable	If ‘Enable’ is unchecked, the default value is used.	
Bus		
Station address	PROFIBUS address of the device	0 … 126
Enable address switch	Is only displayed here; is enabled in the **Settings** window. Defines if the station address is configured in the configuration software or at the address switch. If checked, the station address is configured at the address switch. The parameter ‘Enable Address Switch’ with the setting ‘checked’ can only be used for NT 100-DP/DPS and NT 50-DP/DPS.	Default NB 100 DP/DPS: unchecked, NT 100 DP/DPS, NT 50 DP/DPS: checked
Baud rate	Network Baud Rate	9,6 kBit/s 19,2 kBit/s 93,75 kBit/s 187,5 kBit/s 500 kBit/s 1,5 MBit/s 3 MBit/s 6 MBit/s 12 MBit/s 31,25 kBit/s 45,45 kBit/s Auto-detect Default: Auto-detect
Extras	Sync supported: If checked, the Slave stack supports the SYNC command or the SYNC mode is activated.	Default: checked
	Freeze supported: If checked, the Slave stack supports the FREEZE command or the FREEZE mode is activated.	
	Fail safe supported: If checked, the FAILSAFE operation is supported or the FAILSAFE mode is activated.	
	DPV1 enable: If checked, DPV1 is supported or the DPV1 functions are activated.	
	Address change not allowed:If checked, the Slave stack supports the Set Slave Address command. The bus address can be changed via the Master.	

Parameter	Meaning	Range of Value/Value
Data		
Output or Input	**Module:** for output modules for input modules	 1 … 4 5 … 8
	Type: Byte or Word	„Byte“, „Word“(Default) each with consistency
	Size: The number of Bytes or Words in the module.	0, 1, 2, 3, 4, 8, 12, 16, 20, 32, 64 (Byte, Word)
Configuration data	Configuration data for the output and input identifier bytes. The identifier bytes consists of the **Type** and the **Size**. The identifier bytes are the general identifier bytes according to the PROFIBUS standard.	Default: A1, 91 hex
Manual input	If “Manual input” is unchecked the field “Configuration data” shows the output and input identifier bytes, which results from the settings of the output modules and input modules. If “Manual input” is checked the field “Configuration data” is editable. Output and input identifier bytes can be entered into the Configuration Data field to configure the device. Then, the settings of the output modules and input modules have no meaning.	Default: unchecked
Output data bytes	Total of the output identifier bytes of the modules 1 to 4	0 … 244, Default: 2
Input data bytes	Total of the input identifier bytes of the modules 5 to 8	0 … 244, Default: 2

Table 27: Parameters - PROFIBUS-DP Slave (Part 2)

Note: The input and output modules each work with ’consistency’.

Note: To configure the Master, a GSD file (device description file) is required. The settings in the used Master must comply with the settings in the Slave to establish communication. Important parameters are: Station Address, Ident Number, Baudrate and Config Data (the configuration data for the output and input length).

5.3.2 CANopen Slave Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication start application controlled or automatic	Default: Automatic
Watchdog time [ms]	This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
I/O Data status	Status of the input or the output data. For each input and output data the following status information (in Byte) is memorized in the dual-port memory: Status 0 = None (default) Status 1 = 1 Byte (for future use) Status 2 = 4 Byte (for future use)	Default: None
Ident		
Enable	If ‘Enable’ is unchecked, for the single Ident parameters each the respective default value is used.	
Vendor ID	Identification number of the manufacturer	0x00000000 ... 0x0000FFFF (hex), Hilscher: 0x00000044 (hex)
Product code	Product code of the device as specified by the manufacturer	0x00000000 ... 0xFFFFFFFF (hex), Default: NT 50-CO/COS: 0x001ACB60 (hex), NT 100-CO/COS: 0x001A2020 (hex), NB 100-CO/COS: 0x001B3190 (hex)
Revision number	Revision number of the device as specified by the manufacturer	0x00000000 ... 0xFFFFFFFF (hex), Default: 0x00020000 (hex)
Serial number	Serial number of the device	0x00000000 ... 0xFFFFFFFF (hex)
Bus		
Node ID	Node ID of the CANopen Slave	1 … 127, Default: 2
Enable address switch	Is only displayed here; is enabled in the **Settings** window. Defines if the Node ID is configured in the configuration software or at the address switch. If checked, the Node ID is configured at the address switch. The parameter ‘Enable Address Switch’ with the setting ‘checked’ can only be used for NT 100-CO/COS and NT 50-CO/COS.	Default NB 100-CO/COS: unchecked, NT 100-CO/COS, NT 50-CO/COS: checked
Baud rate	Baud rate of CANopen connection Available Baud Rate:	Auto-Detect 1 Mbaud, 800 Kbaud, 500 Kbaud, 250 Kbaud, 125 Kbaud, 100 Kbaud, 50 Kbaud, 20 Kbaud, 10 Kbaud, Default: 1 MBaud
Data		

Parameter	Meaning	Range of Value/Value
Send object/ Receive object	**Send object**: Send object index **Receive object**: Receive object index	0x00002000 ... 0x00002003 (hex) 0x00002200 ... 0x00002203 (hex)
	Size: Number of data Bytes to send per send object or number of data Bytes to receive per send object.	128
Output data bytes	Total output data bytes of all send objects	512, Default: 512 Bytes*
Input data bytes	Total intput data bytes of all receive objects	512, Default: 512 Bytes*

Table 28: CANopen Slave Parameters

Note: To configure the Master, an EDS file (device description file) is required. The settings in the used Master must comply with the settings in the Slave to establish communication. Important parameters are: Node Address and Baudrate.

Note: *The maximum number of output data bytes and input data bytes is for the device 512 bytes each. The number of output data bytes and input data bytes is not adjustable in the Slave and therefore greyed out in the dialog.

The number of output data bytes and input data bytes, which are to be transferred effectively between the CANopen Master and Slave, are configured in the used CANopen Master. The CANopen Master configures the device during establishing the communication and set the number of output data bytes and input data bytes at this same time.

5.3.3 DeviceNet Slave Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication start application controlled or automatic	Default: Automatic
Watchdog time [ms]	This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
I/O data status	Status of the input or the output data. For each input and output data the following status information (in Byte) is memorized in the dual-port memory: Status 0 = None (default) Status 1 = 1 Byte (for future use) Status 2 = 4 Byte (for future use)	Default: None
Ident		
Enable	If ‘Enable’ is unchecked, for the single Ident parameters each the respective default value is used.	
Vendor ID	Identification number of the manufacturer	0x00000000 ... 0x0000FFFF (hex), Hilscher: 0x00000011B (hex)
Product code	Product code of the device	0x00000000 ... 0xFFFFFFFF (hex), Default NT 50 DN/DNS: 0x0000002D (hex), NT 100 DN/DNS: 0x00000002B (hex), NB 100 DN/DNS: 0x0000002F (hex)
Serial number	Serial number of the device	0x00000000 ... 0xFFFFFFFF (hex)
Product type	Communication Adapter	0x00000000 ... 0x0000FFFF (hex), Default: 0x00000000C (hex)
Major revision	Major revision	1 ... 255, Default: 1
Minor revision	Minor revision	1 ... 255, Default: 1
Product name	The variable Product Name is a text string that should represent a short description of the product/product family.	0 ... 31 ASCII Characters
Bus		
MAC ID	This parameter defines the DeviceNet address of the device within the network.	0 … 63, Default: 2
Enable address switch	Is only displayed here; is enabled in the **Settings** window. Defines if the MAC ID is configured in the configuration software or at the address switch. If checked, the MAC ID is configured at the address switch. The parameter ‘Enable Address Switch’ with the setting ‘checked’ can only be used for NT 100-DN/DNS and NT 50-DN/DNS.	Default NB 100 DN/DNS: unchecked, NT 100 DN/DNS, NT 50 DN/DNS: checked
Baud rate	Baud rate of DeviceNet connection	500 kBaud, 250 kBaud, 125 kBaud, Default: 500 kBaud
Extras	Ignore address switch: address switches are always ignored	Default: unchecked

Parameter	Meaning	Range of Value/Value
	Continue on bus off: Unchecked: A device reset by the user is necessary in case of a CAN bus off event (e. g. short circuit of the data lines) Checked: The device tries independently to continue operation in case of a CAN bus off event	Default: unchecked
	Continue on loss of network power (NP): Function not supported	Default: unchecked
	Receive-idle clears data: Unchecked: Received data keep their last state in case of idle state Checked: Received data were set to zero in case of idle state	Default: unchecked
	Receive Idle keeps data: Function not supported	Default: unchecked
Data		
Consumed data length	Consumed data length sets the number of receive bytes.	0 … 255, Default: 8
Produced data length	Produced data length sets the number of send bytes.	0 … 255, Default: 8

Table 29: DeviceNet Slave Parameters

Note: To configure the Master, an EDS file (device description file) is required. The settings in the used Master must comply with the settings in the Slave to establish communication. Important parameters are: MAC ID, Baudrate, Produced Size, Consumed Size, Vendor ID, Product Type, Product Code, Major Rev, Minor Rev.

5.3.4 CC-Link Slave Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication start application controlled or automatic	Application controlled, Automatic (Default)
Watchdog time [ms]	Watchdog time within which the device watchdog must be retriggered from the application program while the application program monitoring is activated. When the watchdog time value is equal to 0 the application program monitoring is deactivated. This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
I/O data status	Status of the input or the output data. For each input and output data the following status information (in Byte) is memorized in the dual-port memory: Status 0 = None (default) Status 1 = 1 Byte (for future use) Status 2 = 4 Byte (for future use)	None, (1 Byte, 4 Byte) Default: None
Ident		
Enable	If ‘Enable’ is unchecked, for the single Ident parameters each the respective default value is used.	
Vendor code	Code for the Vendor	0 ... 65535 bzw. 0x00000000 ... 0x0000FFFF (hex), Hilscher: 0x00000352 (hex)
Model type	Model type	0 ... 255, Default NT 50-CC/CCS, NT 100-CC/CCS: 1, CIFX CC/CCS, NETX 50 CC/CCS, NETX 100 CC/CCS, NETX 500 CC/CCS: 2 COMX 10XX-CCS/CCS: 3, NJ 10XX-CCS/CCS: 4
Software version	Software version	0 ... 63, Default: 4
Bus		
Station address	Station address of CC-Link Slave Note: The number of occupied stations plus station address must not exceed the parameter range	1 ... 64, Default: 1
Enable address switch	Is only displayed here; is enabled in the **Settings** window. Defines if the station address is configured in the configuration software or at the address switch. If checked, the station address is configured at the address switch. The parameter ‘Enable Address Switch’ with the setting ‘checked’ can only be used for COMX 10XX-CCS/CCS.	Default: CIFX CC/CCS, NT 50-CC/CCS, NT 100-CC/CCS, NETX 50-CC/CCS, NETX 100 CC/CCS, NETX 500 CC/CCS, NJ 10XX-CCS/CCS: unchecked COMX 10XX-CCS/CCS, checked
Baud rate	Network transmission rate	156 kBaud (Default) 625 kBaud 2500 kBaud 5 MBaud 10 MBaud

Parameter	Meaning	Range of Value/Value
Enable baud rate switch	Defines if the baudrate is configured in the configuration software or at the baudrate switch. If checked, the baudrate is configured at the baudrate switch. The parameter 'Enable Address Switch' with the setting 'checked' can only be used for COMX 10XX-CCS/CCS.	Default: CIFX CC/CCS, NT 50-CC/CCS, NT 100-CC/CCS, NETX 50-CC/CCS, NETX 100 CC/CCS, NETX 500 CC/CCS, NJ 10XX-CCS/CCS: unchecked COMX 10XX-CCS/CCS,: checked
Data		
CC-Link version	CC-Link Version 1 CC-Link Version 2	1 (Default) 2
Hold last received output data	Hold Clear Mode; Behavior in case of bus error Clear output data (unchecked) Hold last received output data (checked)	Default: unchecked
Station type	Type of CC-Link station Remote I/O Station: Remote Device Station	 0 (Default) 1
Number of stations	Number of occupied stations Remote I/O Station: Remote Device Station:	 1 (Default) 1 ... 4
Extension cycles	Number of extension cycles Allowed numbers for CC-Link version 1: Single/One cycle Allowed numbers for CC-Link version 2: Single/One cycle Double/Two cycles Quadruple/Four cycles Octuple/Eight cycles	 1 1 (Default) 2 4 8
I/O data bytes	The number of IO-Data bytes depends on the following settings: station type, number of stations and number extension cycles. The number of stations can only be configured with station type Remote Device Station version 1 and version 2 and the number of extension cycles can only be configured with version 2. Firmware/stack works according to CC-Link Version 2.0 input data output data Firmware/stack works according to CC-Link Version 1.11 input data output data	 12 … 368 Bytes 12 … 368 Bytes 4 … 48 Bytes 4 … 48 Bytes Default: 4

Table 30: CC-Link Slave Parameters

Note: To configure the Master, a CSP file (device description file) is required. The settings in the used Master must comply with the settings in the Slave to establish communication.

Important parameters are: Slave Station Address, Baudrate, Station Type and Vendor Code. For CC-Link Version 2.00 are important: number of cycles as well as number of extension cycles.

5.3.5 CompoNet Slave Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication start application controlled or automatic	Application controlled, Automatic (Default)
Watchdog time [ms]	Watchdog time within which the device watchdog must be retriggered from the application program while the application program monitoring is activated. When the watchdog time value is equal to 0 the application program monitoring is deactivated.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
I/O data status	Status of the input or the output data. For each input and output data the following status information (in Byte) is memorized in the dual-port memory: Status 0 = None (default) Status 1 = 1 Byte (for future use) Status 2 = 4 Byte (for future use)	None, (1 Byte, 4 Byte) Default: None
Ident		
Enable	If ‘Enable’ is unchecked, for the single Ident parameters each the respective default value is used.	
Vendor ID	Identification number of the manufacturer *If the value 0x00000000 (hex) is entered, the firmware uses the internal Hilscher ID 0x0000011B (hex).*	0x00000000 ... 0x0000FFFF (hex), Hilscher: 0x0000011B (hex)
Product code	Product code of the device *The firmware uses the entered value.*	0x00000000 ... 0xFFFFFFFF (hex), Default: CIFX CP/CPS: 0x00000201 (hex), NETX 50 CP/CPS: 0x00000203 (hex), NETX 100 CP/CPS: 0x00000204 (hex), NETX 500 CP/CPS: 0x00000205 (hex)
Serial number	Serial number of the device *If the value 0x00000000 (hex) is entered, the firmware uses the internal device serial number.*	0x00000000 ... 0xFFFFFFFF (hex)
Product type	Communication Adapter *If the value 0x00000000 (hex) is entered, the firmware uses the value 0x0000000C (hex).*	0x00000000 ... 0x0000FFFF (hex), Default: 0x0000000C (hex)
Major revsion	*If the value 0 is entered, the firmware uses the value 1.*	0 ... 255, Default: 1
Minor revision	*If the value 0 is entered, the firmware uses the value 1.*	0 ... 255, Default: 1
Product name	The variable Product Name is a text string that should represent a short description of the product/product family. *If no product name is entered, the device uses an internal default name.*	0 ... 31 ASCII Characters
Bus		
Node mode	Node mode of the CompoNet Slave	Word MIX Slave, Word IN Slave, Word OUT Slave, Bit MIX Slave, Bit IN Slave, Bit OUT Slave, Default: Word MIX Slave
Node address	Node address of the CompoNet Slave	Default: 2

Parameter	Meaning	Range of Value/Value
MAC ID	The MAC ID defines the CompoNet address of the device within the network. The MAC ID is calculated from the selected **Node type** and from the selected **Node Address** according to the following table. In this field the MAC ID is only displayed. **Node type** / **Node Address** / **MAC ID** Word IN / 0-0x3F / 0x0-0x3F (0-63) Word OUT / 0-0x3F / 0x40-0x7F (64-127) Word MIX / 0-0x3F / 0x0-0x3F (0-63) Bit IN / 0-0x7F / 0x80-0xFF (128-255) Bit OUT / 0-0x7F / 0x100-0x17F (256-383) Bit MIX / 0-0x7F / 0x80-0xFF (128-255)	0 … 383, Default: 2 (when using the default values for node type and node address)
Baud rate	Baud rate of the CompoNet connection	Auto-detect, 93,75 kbps, 1,5 Mbps, 3 Mbps, 4 Mbps Default: Auto-detect
Data		
ProduceddData	Produced data sets the number of send points (Bits). Produced data can be selected for the Node type „IN“ and „MIXED“, for „OUT“ it is grayed out.	Node type „Bit“: 2, 4 (Points), Default: 2 Node type „Word“: 8, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 170, 192, 208, 224, 240, 256 (Points), Default:16
Size (Produced data)	The number of bytes in the module for the produced data.	1 … 32 (Bytes)* [*= 8 … 256 Points] Default Node type „Bit“: 1 Node type „Word“: 2
Consumed data	Consumed data sets the number of receive points (Bits). Consumed data can be selected for the Node type „OUT“ and „MIXED“, for „IN“ it is grayed out.	Node type „Bit“: 2, 4 (Points), Default: 2 Node type „Word“: 8, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 170, 192, 208, 224, 240, 256 (Points), Default:16
Size (Consumed data)	The number of bytes in the module for the consumed data.	1 … 32 (Bytes)* [*= 8 … 256 Points] Default Node type „Bit“: 1 Node type „Word“: 2

Table 31: CompoNet Slave Parameters

Note: To configure the Master, an EDS file (device description file) is required. The settings in the used Master must comply with the settings in the Slave to establish communication. Important parameters are: Node Mode, MAC ID, Baudrate, Produced Data, Consumed Data, Vendor ID, Product Type, Product Code, Major Rev, Minor Rev.

5.4 Serial Protocols

5.4.1 Modbus RTU Parameters

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication start application controlled or automatic	Application controlled, Automatic (default)
Watchdog time [ms]	Watchdog time within which the device watchdog must be retriggered from the application program while the application program monitoring is activated. When the watchdog time value is equal to 0 the application program monitoring is deactivated. This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Watchdog timer off
Address mapping	Valid address range of the data. This parameter can not be edited here.	0 ... 65535
Data swapping	Data-storage mode: No: Data will not be swapped Yes: Data will be swapped.	Yes, No, Default:: Yes
Map FC1 and FC3	If unchecked, data are read from the intput area using FC1, FC3 and FC23. If checked, data are read from the output area using FC1, FC3 and FC23. FC1 then can be used instead of FC2 and FC3 instead of FC4.	checked, unchecked, Default: unchecked
Ident		
Device	Device code If ‘Enable’ is unchecked, the default value is used.	NT50, NT100 (Default), NB100, String
Bus		
Protocol mode	Determines the Mode of operation: Master or Slave. For „Master“ mode **Command Table** is displayed in the navigation area. For the Modbus RTU Master the Command Table is a list with commands for the reading or writing of data.	Master, IO Slave, Slave (default)
Response timeout	Timeout value (specified in milliseconds) , Only valid for Master mode: Defines the maximum waiting time of the master for an answer from the slave	10 … 10000 ms, Default: 1000
Modbus address	Defines the own Modbus address of the Gateway device.	1 … 247, Default: 2
Send retries	Defines number of telegram retries when connection fails. Relevant only for operation as „Master”.	0 … 10, Default: 3
Interface type	COM-Interface to be used	RS232, RS485, RS422, SPI Mode3, Default: RS232
RTS control	Signal “Return To Send line”* can be switched off or on. (*netTap100 - see hardware manual).	RTS Control Off (default), RTS Control On
Baud rate	Modbus RTU Network Baud Rate. Defines the transmission rate.	1,2 kBit/s, 2,4 kBit/s, 4,8 kBit/s, 9,6 kBit/s (Default), 19,2 kBit/s, 38,4 kBit/s, 57,6 kBit/s, 115,2 kBit/s
Stop bits	Defines the quantity of stop bits, to be used in protocol for serial data communication	1, 2 Default: 2
Parity	Defines the parity bit for serial data communication.	None, Even, Odd, Default: Even

Parameter	Meaning	Range of Value/Value
Frame format	This parameter is not supported here.	Include address and CRC, EXclude address and CRC, Default: Include address and CRC
Data		
Number of registers	Not editable in protocol mode master	0 ... 65535, Default: 2880
Number of coils	not editable	0… 65535, Default: Automatic

Table 32: Modbus RTU Parameters

Settings for Modbus RTU Slaves

Note: The settings in the used Modbus RTU Slaves must comply with the settings in the netTAP or netBRICK device as Modbus RTU Master to establish a communication. Important parameters are: Modbus RTU address, baud rate, number of stop bits and parity setting.

5.4.2 3964R-Parameters

Parameter	Meaning	Value Range/Value
Interface		
Bus startup	Communication start application controlled or automatic	Default: Automatic
Watchdog time [ms]	This function is not supported by gateway or proxy devices.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Watchdog timer off
I/O data status	Status of the input or the output data. For each input and output data the following status information (in Byte) is stored in the dual-port memory: Status 0 = None (default) Status 1 = 1 Byte (for future use) Status 2 = 4 Byte (for future use)	None, (1 Byte, 4 Byte) Default: None
Application mode	Mode how the superordinated control unit controls to send a 3964R telegram respectively detects the receipt of a 3964R telegram. Handshake Mode: The control is done with synchronization registers.	Default: Handshake Mode
Ident		
Device	Device name If ‘Enabled’ is unchecked, the default value is used.	
Bus		
Interface type	Type of the serial interface, which should be used	RS232 (Default), RS422, RS485
RTS control	Specifies, if RTS control is activated or deactivated. When using the interface type RS-485, then "RTS Control on" has to be activated (On).	RTS Control Off (Default), RTS Control On
Baud rate	Specifies the baud rate.	300 Bit/s 600 Bit/s 1200 Bit/s 2400 Bit/s 4800 Bit/s 9600 Bit/s (Default) 19200 Bit/s 38400 Bit/s 57600 Bit/s 115200 Bit/s
Data bits	Specifies the number of data bits, 7 or 8	7, 8 (Default)
Stop bits	Specifies the number of stop bits, 1 or 2	1 (Default) 2
Parity	Specifies the parity bit of the serial data communication. None means that no parity bit is used.	None (Default), Even, Odd
Conflict priority	A conflict occurs when both devices start a data transfer at the same time. The conflict priority specifies the behavior for this case: One device must have high priority the other device must have low priority. The device with high priority can continue with the data transfer while the device with low priority must wait with the data transfer.	High (Default), Low
Retry limit	When transmission errors occur, then the retry limit specifies the maximum number of retries to do a data transfer successfully. After the maximum number of retries has been reached the data transfer is aborted with error.	0 … 65535, Default = 6

Parameter	Meaning	Value Range/Value
Character delay time	The value specifies the time in ms within the remote device must have sent the next character of a telegram. If the time is exceeded, then an error is reported to the superordinated control unit.	[4 … 65535] ms, Default = 220 ms
Acknowledge timeout	The value specifies the time in ms the remote device must have sent an acknowledge. If the time is exceeded, then an error is reported to the superordinated control unit.	[256 … 65535] ms Default = 550 ms
Data		
Send data length max.	Maximum telegram length for receive data. Received data are stored in the input data area.	0 … 5736, 1024 (Default)
Receive data length max.	Maximum telegram length for send data. Send data are taken from the output data area.	0 … 5736, 1024 (Default)

Table 33: 3964R Parameters

Settings for the 3964R Remote Device

Note: The settings in the used remote device must comply with the settings in the netTAP or netBRICK device to establish a communication. Important parameters are: Interface type, baud rate, data bits, parity, RTS control, retry limit, character delay timeout and acknowledge timeout. The conflict priority has to be set different for the used devices.

5.4.3 ASCII Settings

Parameter	Meaning	Value Range/Value
Interface type	Type of the serial interface, which should be used	RS232 (Default), RS485, RS422
RTS Control	Specifies, if RTS control is activated or deactivated. When using the interface type RS-485, then "RTS Control on" has to be used.	RTS Control Off (Default), RTS Control On
Baud rate	Specifies the baud rate.	300 Bit/s 600 Bit/s 1200 Bit/s 2400 Bit/s 4800 Bit/s 9600 Bit/s (Default) 19200 Bit/s 38400 Bit/s 57600 Bit/s 115200 Bit/s
Data bits	Specifies the number of data bits, 7 or 8	7, 8 (Default)
Stop bits	Specifies the number of stop bits, 1 or 2	1 (Default) 2
Parity	Specifies the parity bit of the serial data communication. None means that no parity bit is used.	None (Default), Even, Odd

Table 34: ASCII Parameter

Settings for the remote ASCII Device

Note: The settings in the used remote ASCII device must comply with the settings in the netTAP or netBRICK device to establish a communication. Important parameters are: Interface type, baud rate, data bits, parity check and respectively RTS control.

5.4.4 ASCII Parameters

The operating modes are:

- ‘Receive Only’ Mode
- ‘Send Only’ Mode
- Client Mode (first send, then receive)
- Server Mode (first receive, then send)

The further configuration parameters specify the

- the telegram structure of the send telegram,
- the telegram structure of the receive telegram,
- the timing,
- the size of the send and receive buffer

5.4.4.1 Telegram Structure in Data Stream

A send or receive telegram in the simplest case only consists only by user data. Many implementations however add to the user data further characters which have a certain meaning, such as a start character, an end character, a checksum or a device address.

A typical telegram structure is e. g.

Start | D a t a | Checksum | End

with for example (in hexadecimal notation):

[0x02] [0x38][0x33][0x33][0x37][0x38][0x30][0x33][0x37][0x36][0x33] [0x69][0xA5] [0x03]

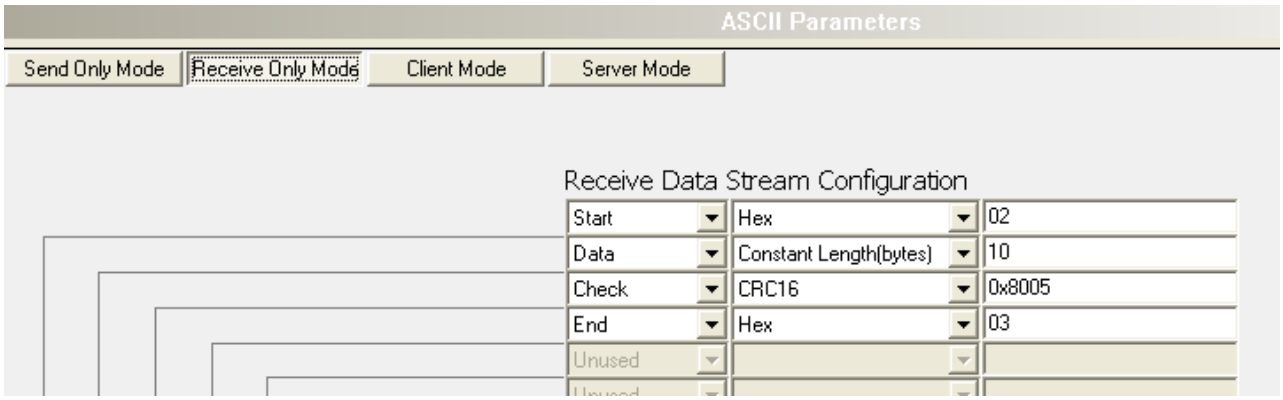


Table 35: ASCII Telegram Structure Example

The serial data stream of the send and receive telegram each can be defined with up to 10 structural elements.

Parameter	Description	Range of Value / Parameter Type
Unused	Structural element is not used	-
Start	Start character of the telegram consisting of one or more characters The information is given as an ASCII character (Char) or as hexadecimal (Hex). For the ASCII character 'STX' enter the hexadecimal value'02'. When sending, the start character is inserted in the telegram. When receiving, the start character is checked for equality.	ASCII, Hex, Decimal
Device	Device Address When sending, the device address is inserted in the telegram. Thereby the receiving device is specified. When receiving, the device address is checked for equality.	ASCII, Hex, Decimal
Object	Object Index or Start Address of the data in the device When sending, the object index / start address is inserted in the telegram. When receiving, the object index / start address is checked for equality.	ASCII, Hex, Decimal
Command	Command Identifier When sending, the Command Identifier is inserted in the telegram. When receiving, the Command Identifier is checked for equality.	ASCII, Hex, Decimal
Data	Length specification for Data Field or Telegram end (Character or Time) **Length Specification:** Constant Length (bytes): The data field has a fixed length. The information is given as number of bytes. Byte Number Data: The length specification is given in one byte. When sending, this byte is inserted in the telegram directly before the data field. When receiving, the value of this byte from the telegram is used as the length of the directly following data field. Word Number Data: The length specification is given in one word. When sending, this word is inserted in the telegram directly before the data area. When receiving, the value of this word from the telegram is used as the length of the directly following data field. **Termination Character:** Character Terminated: The data field is terminated by a fixed end identifier consisting of one byte. When sending, this termination character is inserted in the telegram after the data field. When receiving, this termination character is used to detect the end of the data field. The termination character is not part of the data field. **Telegram end by time:** Unspecific: The end of the data field is done with the character delay time.	Constant Length (Bytes) Byte Number Data Word Number Data Character Terminated Unspecific
End	End of the telegram character consisting of one or more characters The information is given as an ASCII character (Char) or as hexadecimal (Hex). For the ASCII character 'ETX' enter the hexadecimal value'03'. When sending, the end character is inserted in the telegram. When receiving, the end character is checked for equality.	ASCII, Hex, Decimal

Parameter	Description	Range of Value / Parameter Type
Check	Checksum CRC8: Any data of the telegram before the checksum are inserted in the initial value zero with the polynomial 1D (default) and the 1-byte result is used as checksum. CRC16: Any data of the telegram before the checksum are inserted in the initial value zero with the polynomial 8005 (default) and the 2-byte result is used as checksum. CRC32: Any data of the telegram before the checksum are inserted in the initial value zero with the polynomial 04C11DB7 (default) and the 4-byte result is used as checksum. Exor: Any data of the telegram before the checksum are set off against the initial value zero with exclusive-or and the 1-byte result is used as checksum. When sending, the computed checksum is inserted in the telegram. When receiving, the checksum is calculated based on of the received characters and is checked for equality with the received checksum.	CRC8, CRC16, CRC32, Exor
Don't care	Characters with no meaning Constant Length (bytes): Specifies the number of bytes, which have no meaning and which should be ignored. When sending, the number of characters with a value of zero is inserted in the telegram. When receiving, the number of characters is ignored and thus filtered out from the telegram.	Constant Length (Bytes)

Table 36: Telegram Structure

Type	Description
Hex	Hexadecimals The entry is made as a hexadecimal value. A hexadecimal value consists of 2 characters in the range 00 to FF and result in one byte. Each character has the range of value 0, 1, 2, ..., 9, A, B, ..., F.
ASCII	ASCII Character The entry is made as a character. A character is one of the following characters: A-Z, a-z, 0-9, !, ", §, $, %, &, /, (,), =, ?, , ; -, _, +, *
Decimal	Decimal value The entry is made as a decimal value.

Table 37: Parameter Types

Example: The specification of the character A is as hexadecimal value '41', as character 'A' or as a decimal value 65.

ASCII Character Table

ASCII Hex	ASCII Dez	Character	ASCII Hex	ASCII Dez	Character	ASCII Hex	ASCII Dez	Character	ASCII Hex	ASCII Dez	Character
00	0	NUL	20	32	SP	40	64	@	60	96	`
01	1	SOH ^A	21	33	!	41	65	A	61	97	a
02	2	STX ^B	22	34	"	42	66	B	62	98	b
03	3	ETX ^C	23	35	#	43	67	C	63	99	c
04	4	EOT ^D	24	36	$	44	68	D	64	100	d
05	5	ENQ ^E	25	37	%	45	69	E	65	101	e
06	6	ACK ^F	26	38	&	46	70	F	66	102	f
07	7	BEL ^G	27	39	'	47	71	G	67	103	g
08	8	BS ^H	28	40	(	48	72	H	68	104	h
09	9	TAB ^I	29	41	)	49	73	I	69	105	i
0A	10	LF ^J	2A	42	*	4A	74	J	6A	106	j
0B	11	VT ^K	2B	43	+	4B	75	K	6B	107	k
0C	12	FF ^L	2C	44	,	4C	76	L	6C	108	l
0D	13	CR ^M	2D	45	-	4D	77	M	6D	109	m
0E	14	SO ^N	2E	46	.	4E	78	N	6E	110	n
0F	15	SI ^O	2F	47	/	4F	79	O	6F	111	o
10	16	DLE ^P	30	48	0	50	80	P	70	112	p
11	17	DC1 ^Q	31	49	1	51	81	Q	71	113	q
12	18	DC2 ^R	32	50	2	52	82	R	72	114	r
13	19	DC3 ^S	33	51	3	53	83	S	73	115	s
14	20	DC4 ^T	34	52	4	54	84	T	74	116	t
15	21	NAK ^U	35	53	5	55	85	U	75	117	u
16	22	SYN ^V	36	54	6	56	86	V	76	118	v
17	23	ETB ^W	37	55	7	57	87	W	77	119	w
18	24	CAN ^X	38	56	8	58	88	X	78	120	x
19	25	EM ^Y	39	57	9	59	89	Y	79	121	y
1A	26	SUB ^Z	3A	58	:	5A	90	Z	7A	122	z
1B	27	Esc	3B	59	;	5B	91	[	7B	123	{
1C	28	FS	3C	60	<	5C	92	\	7C	124	|
1D	29	GS	3D	61	=	5D	93	]	7D	125	}
1E	30	RS	3E	62	>	5E	94	^	7E	126	~
1F	31	US	3F	63	?	5F	95	_	7F	127	DEL

Table 38: ASCII Character Table

5.4.4.2 Size of the Send and Receive Buffer

The size of the send or receive buffer is configured.

Note: The send telegram must fit into the send buffer. The receive telegram must fit into the receive buffer.

Parameter	Description	Range of Value
Sendbuffer Size	Size of the send buffer in bytes	0 … 1024 Default: 512 Bytes
Receivebuffer Size	Size of the receive buffer in bytes	0 … 1024 Default: 512 Bytes

Table 39: Size of the Send and Receive Buffer

5.4.4.3 Operating Modes and Timing

The modes are:

- ‘Receive Only’ Mode
- ‘Send Only’ Mode
- Client Mode (first send, then receive)
- Server Mode (first receive, then send)

5.4.4.4 'Send Only' Mode

Basic Principle

In the 'Send Only' mode the device only sends. The remote device only receives. The superordinated control unit has to use a handshake to transfer the data to the device.

Communication

The send operation can be triggered from the superordinated control unit or performed by the device cyclically.

- triggered

With each handshake of the superordinated control unit a telegram is send. For this, set the timing parameter **Send Cycle Time** to zero.

- Cyclic

The device sends in a constant cycle.

With each handshake of the superordinated control unit the send data initially are updated in the internal buffer of the device and sent during the next send cycle. For this, set the timing parameter **Send Cycle Time** to the cycle time (unequal to zero).

Timing Parameter

Parameter	Description	Range of Value
Send Cycle Time	Specifies whether the telegram will be sent cyclically or triggered. The value (unequal to zero) specifies the cycle time of the send telegram. The value 0 specifies that the send telegram is triggered.	0 … 2^{31}-1 Default: 0

Table 40: Timing Parameters for the Operating 'Send Only' Mode

Configuration Example

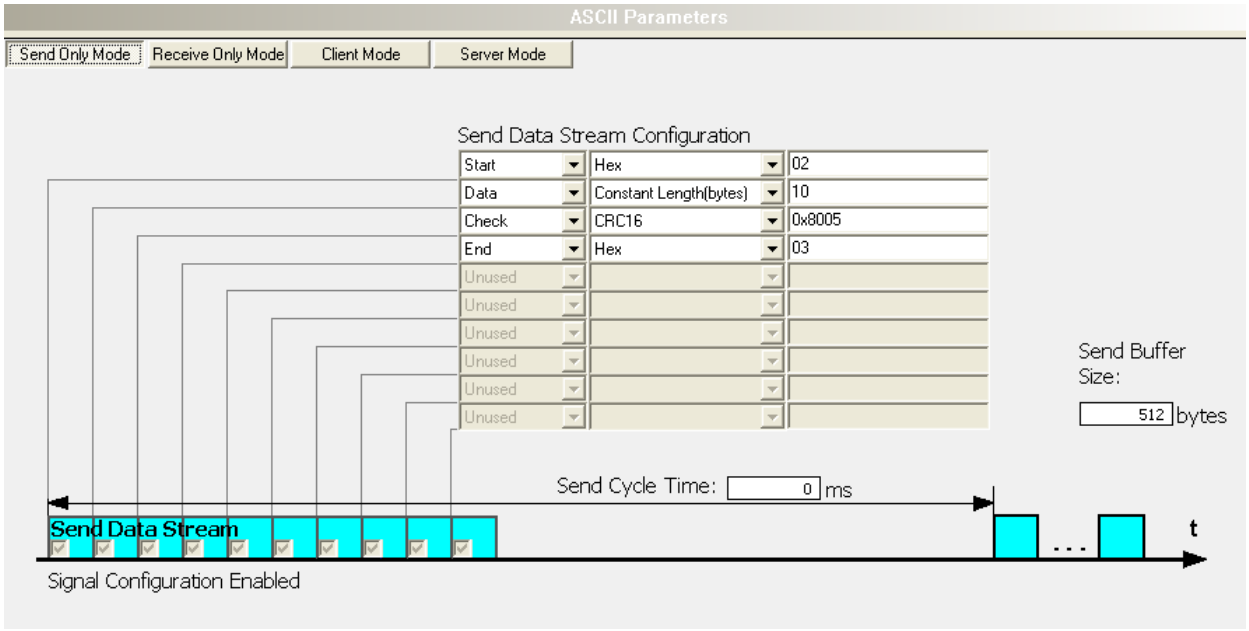


Table 41: Telegram Structure for the Operating 'Send Only' Mode

5.4.4.5 'Receive Only' Mode

Basic Principle

In the 'Receive Only' mode the device only receives. The remote device only sends.

The superordinated control unit must acknowledge the reception.

Communication

When configuring the telegram structure the method is defined how the end of the telegram is detected:

- Character
- Length
- Time

Each telegram received is indicated to the superordinated control unit by handshake. The superordinated control unit must acknowledge the reception. Only after the acknowledgement the reception of a further telegram can be indicated to the superordinated control unit.

Timing Parameter

Parameter	Description	Range of Value
Receive Watchdog Time	Specifies whether the reception of subsequent telegrams is monitored by time. Thereby the remote device can be monitored. The time is taken from telegram end to telegram end. The value (unequal to zero) specifies in what time the remote device must have sent the next telegram. If the time is exceeded, then an error is reported to the host. The value 0 specifies that the reception of subsequent telegrams is not monitored.	0 … 2^{31}-1 Default: 0
Character Delay Time	Specifies whether the time between two characters during reception is monitored by time. The value (unequal to zero) specifies in what time the remote device must have sent the next character. If the time is exceeded, then an error is reported to the host. The value 0 specifies that no monitoring is performed.	0 … 2^{31}-1 Default: 0

Table 42: Timing Parameters for the Operating 'Receive Only' Mode

Configuration Example

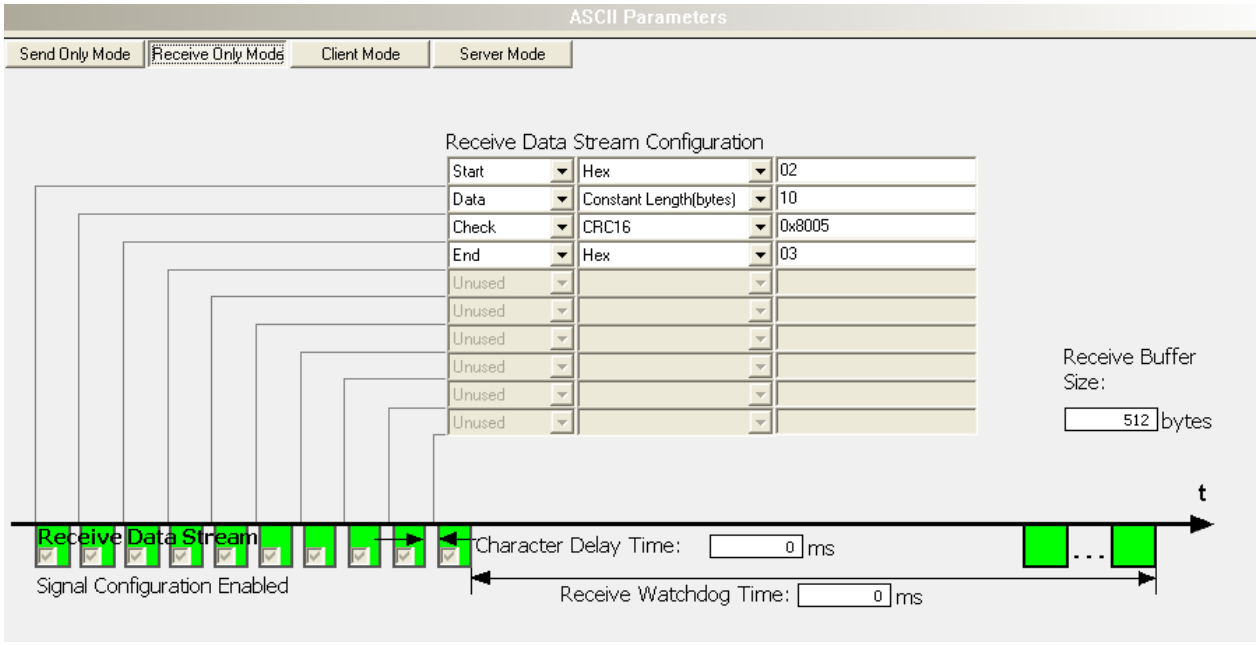


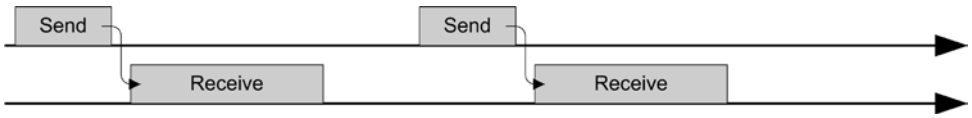
Table 43: Telegram Structure for the Operating 'Receive Only' Mode

5.4.4.6 Client Mode (First Send, Then Receive)

Basic Principle

In the 'Client Mode' the device sends a telegram to the remote device. The remote device then sends a telegram which is received by the device.

By this a polling (querying) of the remote devices can be performed. The device can send a polling telegram with or without user data.



The superordinated control unit must activate the send operation by handshake. The superordinated control unit must acknowledge each reception by handshake.

Communication

The send operation can be triggered or performed cyclically.

- triggered

With each handshake of the superordinated control unit a telegram is send. For this, set the timing parameter **Send Cycle Time** to zero.

The remote device then sends a telegram which is received by the device. The device can monitor the beginning of the receive telegram by time monitoring via the **Response Timeout** parameter.

The superordinated control unit must activate the send operation by handshake. The superordinated control unit must acknowledge each reception by handshake.

- Cyclic

The device sends in a constant cycle.

With each handshake of the host the send data initially are updated in the internal buffer and sent during the next sending cycle. For this, set the timing parameter **Send Cycle Time** to the cycle time (unequal to zero).

The remote device then sends a telegram which is received by the device. The device can monitor the beginning of the receive telegram by time monitoring via the **Response Timeout** parameter.

The superordinated control unit must activate the send data by handshake. The superordinated control unit must acknowledge each reception by handshake.

Timing Parameter

Parameter	Description	Range of Value
Send Cycle Time	Specifies whether the telegram will be sent cyclically or triggered. The value (unequal to zero) specifies the cycle time of the send telegram. The value 0 specifies that the send telegram is triggered.	0 … 2^{31}-1 Default: 0
Response Timeout	Specifies whether the reception of the response telegram is monitored by time. Thereby the remote device can be monitored. The time is taken from telegram end to telegram end. The value (unequal to zero) specifies in what time the remote device must have sent the response telegram. If the time is exceeded, then an error is reported to the host. The value 0 specifies that the reception of response telegrams is not monitored.	0 … 2^{31}-1 Default: 1000
Character Delay Time	Specifies whether the time between two characters during reception is monitored by time. The value (unequal to zero) specifies in what time the remote device must have sent the next character. If the time is exceeded, then an error is reported to the host. The value 0 specifies that no monitoring is performed.	0 … 2^{31}-1 Default: 0

Table 44: Timing Parameters for the Operating Mode 'Client Mode'

Configuration Example

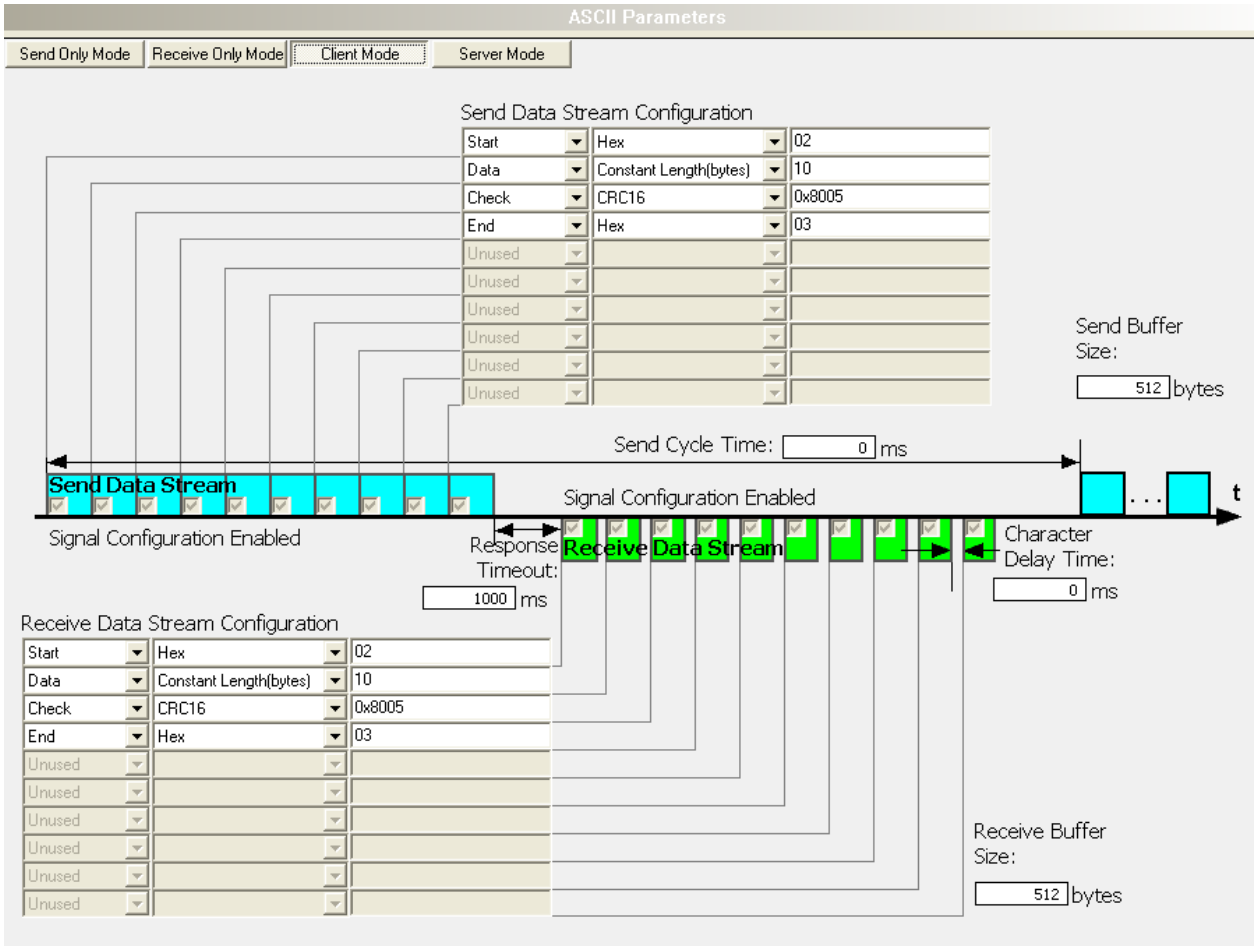


Table 45: Telegram Structure for the Operating Mode 'Client Mode'

5.4.4.7 Server Mode (First Receive, Then Send)

Basic Principle

In the 'Server Mode' the device receives a telegram from the remote device. The superordinated control unit must acknowledge each reception by handshake. The superordinated control unit must activate by handshake the send operation. The device then sends a telegram to the remote device.

Communication

When configuring the telegram structure the method is defined how the end of the telegram is detected:

- Character
- Length
- Time

Timing Parameter

Parameter	Description	Range of Value
Receive Watchdog Time	Specifies whether the reception of subsequent telegrams is monitored by time. Thereby the remote device can be monitored. The time is taken from telegram end to telegram end. The value (unequal to zero) specifies in what time the remote device must have sent the next telegram. If the time is exceeded, then an error is reported to the host. The value 0 specifies that the reception of subsequent telegrams is not monitored.	0 … 2^{31}-1 Default: 0
Response Timeout	Specifies whether the reception of telegrams is monitored by time. Thereby the remote device can be monitored. The time is taken from the end of the send telegram to the start of the receive telegram. The value (unequal to zero) specifies in what time the remote device must have sent the next telegram. If the time is exceeded, then an error is reported to the host. The value 0 specifies that the reception of telegrams is not monitored.	0 … 2^{31}-1 Default: 1000
Character Delay Time	Specifies whether the time between two characters during reception is monitored by time. The value (unequal to zero) specifies in what time the remote device must have sent the next character. If the time is exceeded, then an error is reported to the host. The value 0 specifies that no monitoring is performed.	0 … 2^{31}-1 Default: 0

Table 46: Timing Parameters for the Operating Mode 'Server Mode'

Configuration Example

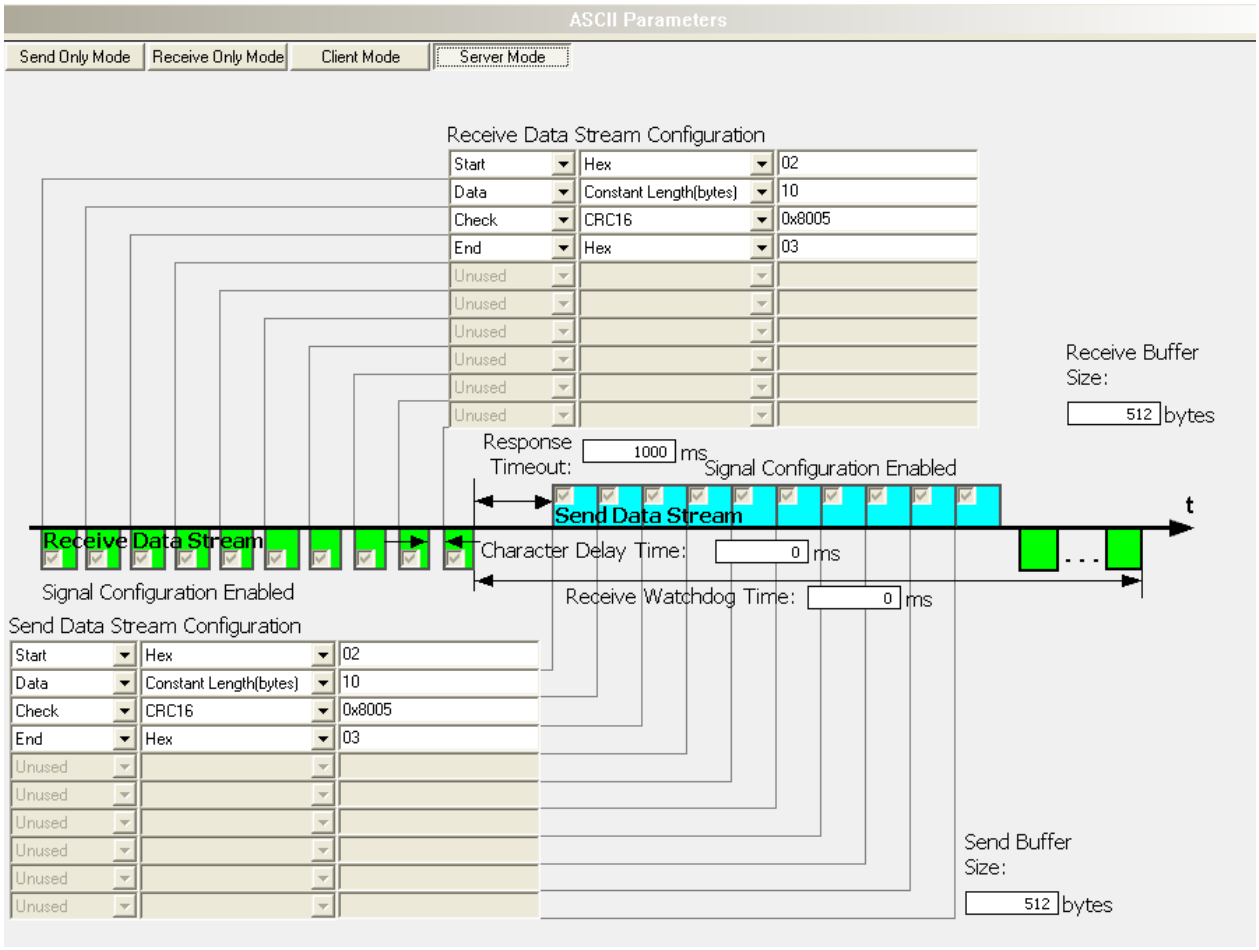


Table 47: Telegram Structure Parameters for the Operating Mode 'Server Mode'

5.4.5 netSCRIPT Settings

Parameter	Meaning	Value Range/Value
Interface type	Type of the serial interface, which should be used	RS232 (Default), RS485, RS422
RTS Control	Specifies, if RTS control is activated or deactivated. When using the interface type RS-485, then "RTS Control on" has to be used.	RTS Control Off (Default), RTS Control On
Baud rate	Specifies the baud rate.	300 Bit/s 600 Bit/s 1200 Bit/s 2400 Bit/s 4800 Bit/s 9600 Bit/s (Default) 19200 Bit/s 38400 Bit/s 57600 Bit/s 115200 Bit/s
Data bits	Specifies the number of data bits, 7 or 8	7, 8 (Default)
Stop bits	Specifies the number of stop bits, 1 or 2	1 (Default) 2
Parity	Specifies the parity bit of the serial data communication. None means that no parity bit is used.	None (Default), Even, Odd

Table 48: netSCRIPT Settings

Settings for the Remote netSCRIPT Device

Note: The settings in the used remote netSCRIPT device must comply with the settings in the netTAP or netBRICK device to establish a communication. Important parameters are: Interface type, baud rate, data bits, parity check and respectively RTS control.

5.5 Signal Configuration

In the **Signal Configuration** dialog you can define the data structure of the input or output data of your device transmitted at the bus.

The application requires the information on the meaning and data type of the input and output data specified via the **signals**.

Important: First configure the input and output data. Only carry out the steps for signal configuration afterwards.

In the **Signal Configuration** dialog you can define the data structure of the input or output data of your device and define the I/O data for your application

- assign data types,
- assign names or signal names, and
- define data structures.

The aim is to create a suitable signal configuration, which subsequently enables easy identification of the transmitted input and output data. This requires a structuring of the input and output data according to signals and the configuration of signal names or data types suitable for the individual application cases.

Signal names

The names assigned by default by the configuration software for the signals distinguish between input and output signals. You can replace these general names with suitable designations, such as "Setpoint" or "Status".

Merging or splitting signals

You can merge or split signals or data types by configuring the data type and the number of signals.

For example you can specify that 4 bytes of input data together match with 1 input signal of the data type ‚UNSIGNED32'.

4 Byte (input) = 1 ‚UNSIGNED32' (input)

To identify split data types, the configuration software assigns appropriate suffixes to the signal names, which depend on the selected new data type, for example _Byte_0, _Byte_4 ... or _Bit_1, _Bit_2 ...

5.5.1 Signal Configuration Window

➢ Select **Configuration** > **Signal Configuration** in the navigation area.

↬ The dialog window **Signal Configuration** is displayed.

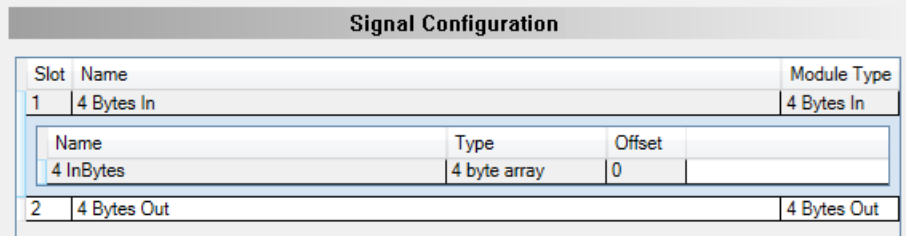


Figure 28: Signal Configuration Window (Example)

Parameter	Description	Range of Value/ Value
Slot	Number	1, 2, 3, …
Name	Number of Bytes In or Bytes Out	Number of Bytes In or Bytes Out
Module Type		
Signal level		
Name	Name of the input or output signal that can be set here. The configuration software assigns names by default: InBytes, OutBytes	String
Type	Data type of the input or output signal (depending on the configured size of the I/O data).	bit, byte, signed8, unsigned8, word, signed16/24, unsigned16/24, dword, signed32/40/48/56, unsigned32/40/48/56, lword, signed64, unsigned64, real32, real64
Offset	Offset of the input or output signal, related to the data in the input or output data memory of the device.	

Table 49: Explanations Signal Configuration Window

5.5.2 Create Signal Configuration

In the **Signal Configuration** window, you can edit the signal configuration in the lower table.

Edit Signal

➢ Right-click on the signal to be configured to open the context menu.

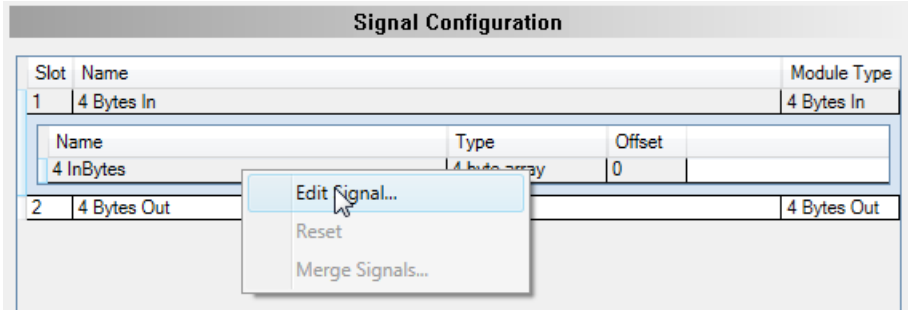


Figure 29: Edit Signal (Example)

➢ Click **Edit Signal**.

➢ The **Edit Signal** dialog window is opened.

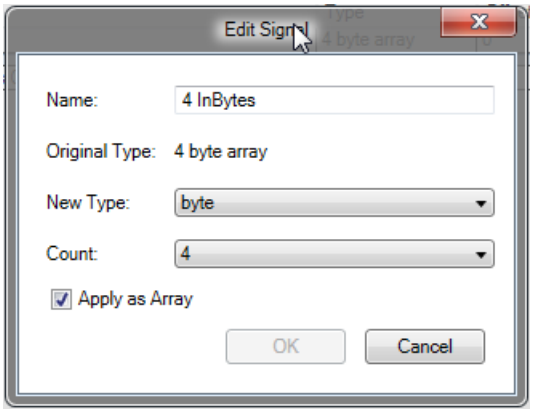


Figure 30: Edit Signal Dialog Window (Example)

Parameter	Description	Range of Value/ Value
Name	Here you can edit the signal name.	String
Original Type	Input or output signal data type preconfigured by the configuration software or by the user.	
New Type	Here you can edit the new data type for the input or output signal. Only permitted data types are displayed in the selection list.	bit, byte, signed8, unsigned8, word, signed16/24, unsigned16/24, dword, signed32/40/48/56, unsigned32/40/48/56, lword, signed64, unsigned64, real32, real64
Count	Here you can set the number of signals with the data type "New Type".	
Apply as Array	If checked, the signal is displayed as an array. If unchecked, the individual signals are displayed.	Checked, unchecked, Default: checked

Table 50: Explanations Edit Signal Dialog Window

- In the field **Name** edit the signal name.
- Use **New Type** to define the new data type or **Count** to define the number of signals.

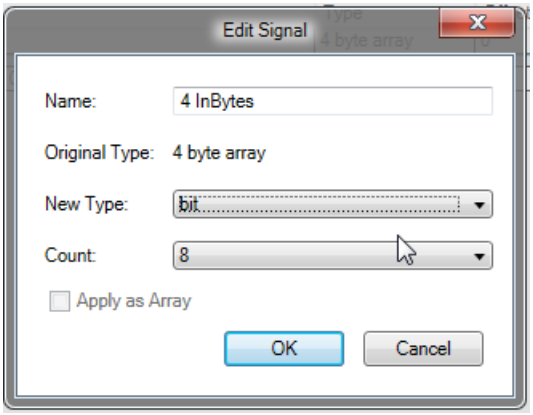


Figure 31: Edit Signal (Example)

- Click **OK**.

When splitting signals, the configuration software assigns suitable standard suffixes to the signal name to identify the subordinate signals.

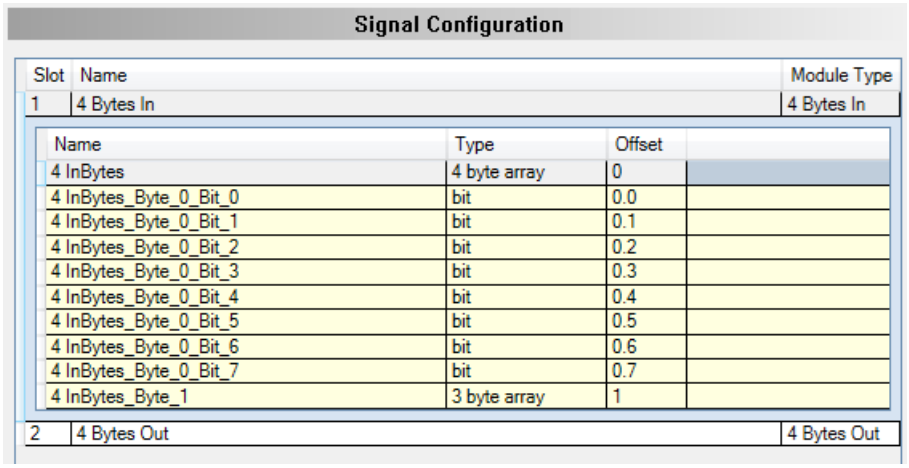


Figure 32: Signal split (Example)

- You can further split already split signals.
- Click **Apply** to save the created configuration.

Reset

Note: As long as you have not applied the created signal configuration, you can undo the steps you have taken by clicking **Reset**.

➢ Right-click on the configured signal to open the context menu.

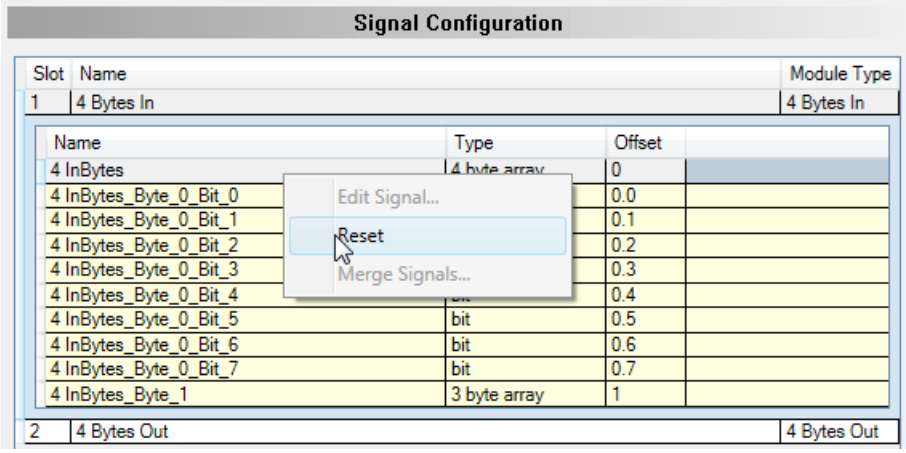


Figure 33: Reset (Example)

➢ Click **Reset**.

➢ The created signal configuration with a splitting of a signal is undone.

Merge Signals

➢ First, confirm a signal splitting with **Apply**.

➢ Then press **Shift** and mark the signals to be merged with the mouse pointer.

➢ Open the context menu by right-clicking on the marked signals.

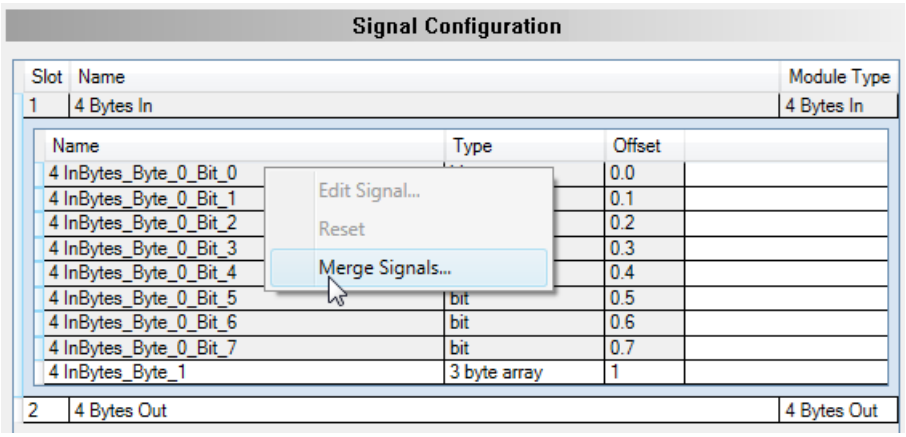


Figure 34: Merge Signals (Example)

➢ Click **Merge Signals**.

➢ The dialog window **Merge Signals** opens.

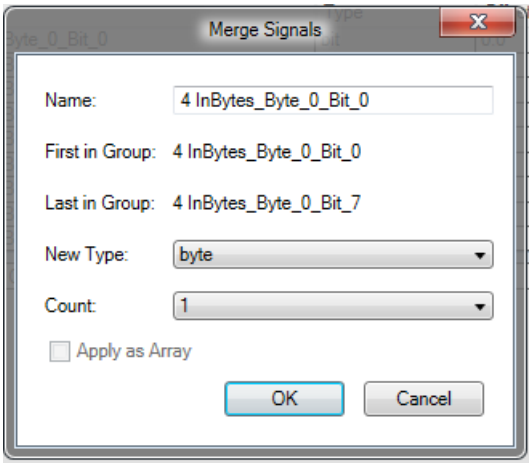


Figure 35: Dialog Window Merge Signals (Example)

Parameter	Description	Range of Value/ Value
Name	Here you can edit the signal name. The name displayed here contains the suffix assigned by the configuration software, example „_Byte_0“.	String
First in Group	Shows the name of the first signal from which the merge will start.	
Last in Group	Shows the name of the last signal up to which the merge is performed.	
New Type	Here you can edit the new data type for the input or output signal. Only permitted data types are displayed in the selection list.	bit, byte, signed8, unsigned8, word, signed16/24, unsigned16/24, dword, signed32/40/48/56, unsigned32/40/48/56, lword, signed64, unsigned64, real32, real64
Count	Shows the number of data types of the merged signal, which you can adjust here.	
Apply as Array	An array is formed when merging.	Allways checked

Table 51: Explanations Merge Signals Dialog Window

- For the signals that you want to merge, specify the name via **Name**, the data type via **New Type** or the number of data types of the merged signal via **Count**.
- Click **OK**.
- The signals are merged.
- Click **Apply** to save the created configuration.

5.6 Command Table

Overview

Section	Subsection	Page
Command Table	*Command Table Modbus RTU*	105
	Command Table Open Modbus/TCP	111

Table 52: Command Table for Gateways

5.6.1 Command Table Modbus RTU

5.6.1.1 For what the Command Table does serve?

For the Modbus RTU Master the **Command Table** is a list with commands for the reading or writing of data.

The **Command Table** is only relevant, if a device works as Master on the Modbus RTU.

From every command line the Modbus RTU Master produces a telegram for reading or for the writing of data to or from a Modbus RTU Slave device. For every command are indicated:

- the Modbus RTU Slave device address (Device Address),
- the Function code,
- the data address in the Modbus RTU Slave device (Address),
- the number of data (Number of Registers/Coils),
- and the data address in the Modbus RTU Master device (Memory Address (internal)).

For writing telegrams (FC 5, 6, 15 and 16) in the column

- Trigger

you can decide, whether the writing telegrams are executed every time (Cyclic) or only at data modification (Change data).

Furthermore

- a Cycle Time

can be set for every command.

The **Command Table** is processed from the first to the last entry (from above to below). After the execution of the last command the execution of the first command is started again.

For reading commands the Modbus RTU Master reads out data from the Modbus RTU Slave and saves them into its data memory.

For writing commands the Modbus RTU Master reads out data from its data memory and writes them into the Modbus RTU Slave.

The number of commands which can be defined depends from the firmware and from the Dual-Port Memory layout.

5.6.1.2 Opening Command Table

To view the **Table Command**, proceed as follows:

For Gateway devices:

1. In the netGateway DTM select **Configuration > Settings**.
2. In the **Settings** window under **Protocol Combination >**

➢ At **Primary network (Port X2)** select the primary network protocol.

➢ At **Secondary network (Port X3)** select the secondary network protocol Modbus RTU.

➢ Select **OK**.

3. In the **netDevice** window put a right click on the DTM icon.
4. In the context menu select **Configuration > Modbus RTU**.
5. In the **Configuration** window (Configuration Modbus RTU), under **Protocol Mode** set the entry to "Master".

➢ Answer to the query “This and following changes will force recreation of signal configuration! Do you want to proceed?” with **Yes**.

6. In the navigation area select **Configuration > Command Table**.

➢ The window **Command Table** is displayed.

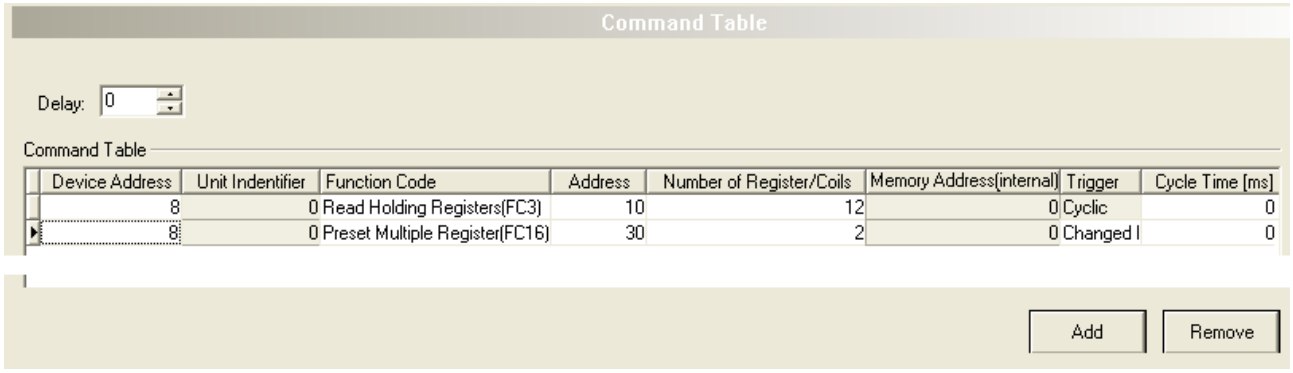


Figure 36: Window Command Table

Each command contains the following parameters:

Parameter	Description
Device Address	Modbus RTU Slave device address
Unit Identifier	(not relevant here)
Function Code	Function code
Address	data address in the Modbus RTU Slave device
Number of Registers/Coils	Number of data
Memory Address (internal)	Data address in the Modbus RTU Master device (= Byte address in the process image of the Modbus RTU Master device)
Trigger	Trigger for writing commands
Cycle Time	Cycle time

Table 53: Parameter of the Command Table

For a detailed description of the parameters refer to section *Parameter of the Command Table* beginning from page 107.

5.6.1.3 Parameter of the Command Table

<table>
<tr><th>Parameter</th><th>Meaning</th><th>Range of Value/Value</th></tr>
<tr><td>Device Address</td><td>Specifies the Modbus RTU Slave device address, from which the Modbus RTU master device reads out the data or into which it writes the data.</td><td>For Modbus RTU: 1 … 247
Default: 1</td></tr>
<tr><td>Unit Identifier</td><td>(not relevant here)</td><td>Default: 0</td></tr>
<tr><td>Function Code</td><td>Specifies the function code for read and write requests [3]:
Read Coils(FC1)
Read Inputs(FC2)
Read Holding Registers(FC3)
Read Input Registers(FC4)
Force Single Coil(FC5)
Preset Single Register(FC6)
Force Multiple Coils(FC15)
Preset Multiple Registers(FC16)</td><td>1 / 2 / 3 / 4 / 5 / 6 / 15 / 16
Default: „Read Coils (FC1)“</td></tr>
<tr><td>Address</td><td>Indicates the data address in the Modbus RTU Slave device. The parameter Address contains the register address in the Modbus RTU Slave device. The register address is counted for every function code beginning with 0.
The following table shows the assignment of the Address to the data address in the Modbus RTU Slave by Function code:
<table>
<tr><th rowspan="2">Parameter Address</th><th colspan="4">Data Address Modbus RTU Slave</th></tr>
<tr><th>FC 1
FC 5
FC 15</th><th>FC 2</th><th>FC 4</th><th>FC 3
FC 6
FC 16</th></tr>
<tr><td>0</td><td>1</td><td>10001</td><td>30001</td><td>40001</td></tr>
<tr><td>1</td><td>2</td><td>10002</td><td>30002</td><td>40002</td></tr>
<tr><td>2</td><td>3</td><td>10003</td><td>30003</td><td>40003</td></tr>
<tr><td>3</td><td>4</td><td>10004</td><td>30004</td><td>40004</td></tr>
<tr><td>4</td><td>5</td><td>10005</td><td>30005</td><td>40005</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
</table>

To know, which data address with which data of the Modbus RTU Slave devices is provided for reading or for writing, refer to the device description of the Modbus RTU Slave device manufacturer.</td><td>0 … 65.535
Default: 0</td></tr>
<tr><td>Number of Registers/Coils</td><td>Indicates the number of reading or writing data as registers or coils.
The maximum value depends from the function code.</td><td>1 … max. value, Default: 1
max. value for
FC1 = 2000 (Coils),
FC2 = 2000 (Coils),
FC3 = 125 (Registers),
FC4 = 125 (Registers),
FC5 = 1 (Coils),
FC6 = 1 (Registers),
FC15 = 1968 (Coils),
FC16 = 123 (Registers)</td></tr>
<tr><td>Memory Address (internal)</td><td>Byte address in the process image of the Modbus RTU Master device for input or output data of the Master
The configuration software calculates the byte address in the process image of the Master for the input or for the output data automatically. Because of this definition, for function codes for reading (FC 1, 2, 3 and 4) the data are assigned consecutively in the process image for input data and for function codes for writing (FC 5, 6, 15 and 16) the data are assigned consecutively in the process image for output data.</td><td>0 … 5759
Default: 0</td></tr>
</table>

Parameter	Meaning	Range of Value/Value
Trigger	Defines for all function codes for writing (FC 5, 6, 15 and 16); if the command is to be executed cyclically (Cyclic) or only if the data have changed (Change Data). For function codes for reading (FC 1, 2, 3 und 4) this parameter is not used.	0 = Cyclic 1 = Changed Data Default: „Cyclic“
Cycle Time	The cycle time defines, after how many milliseconds a writing or a reading command shall be reexecuted. The default value „0 ms“ means, that the command is executed as fast as possible. Otherwise the execution can be defined in steps of 10 ms. With the cycle time the temporal execution of the command can be influenced. If since the last execution of the command the set cycle time or more time has been passed, the command is reexecuted. If since the last execution of the command less time has been passed then the cycle time, the command is skipped. *If e. g. from a Modbus RTU Slave data are required only every 10 seconds, as these one change only slowly, then for the cycle time enter 10.000 ms.*	0, 10, 20, 30, … 60.000 ms Default: 0 ms
Delay	Between the single commands a delay time can be parameterized. This sometimes is necessary to avoid a too high load of the connected Slave devices by a continuous communication. **0**: The commands are processed without delay. **1 … 60.000 ms**: Delay time in ms, the Modbus RTU Master waits, before starting the next command.	0 … 60.000 ms Default: 0

Table 54: Parameter of the Command Table

5.6.1.4 Adding/removing Command

To add or to remove data sets for commands to the Command Table, proceed as follows:

Add command:

- In the table **Command Table** put the cursor in the line below which a new data set for a command shall be added.
- Click **Add**.

↬ Under the clicked line a new command is added.

Remove command:

- In the table **Command Table** put the cursor in the line with the data set to be deleted.
- Click **Remove**.

↬ The command of the clicked line is not shown any more.

5.6.1.5 Setting Parameters

Important: When entering the parameters, make sure that these address valid data registers. (*Refer to the device description of the Modbus RTU Slave device manufacturer.*)

To set the single parameters in the Command Table, proceed as follows:

1. **Device Address**, **Function Code**, **Address**, **Number of Registers/Coils**:

The fields for these parameters are editable.

- Enter the values for these parameters to the corresponding table cells.

2. **Trigger**:

- In **Function Code** field select an entry for a writing command (FC 5, 6, 15 or 16).
- In the column **Trigger** select the entry „Cyclic“ or „Change Data“.

3. **Cycle Time**:

- Put the cursor to the table cell and set the cycle time using the spin box in steps of 10 ms.

5.6.1.6 Examples Modbus RTU Reading or Writing Command

Device Address	Unit Indentifier	Function Code	Address	Number of Register/Coils	Memory Address(internal)	Trigger	Cycle Time [ms]
8	0	Read Holding Registers(FC3)	10	12	0	Cyclic	0
8	0	Preset Multiple Register(FC16)	30	2	0	Changed I	0

Figure 37: Examples - Reading Command with FC 3, Writing Command with FC16

Example Reading Command with FC 3: From the Modbus RTU Slave device with the Modbus address 8 from data address 40011 onwards 12 registers are read by function code 3. The data are assigned to the memory address(internal) 0.

Example Writing Command with FC 16: To the Modbus RTU Slave with the Modbus address 8 from data address 40031 onwards 12 registers are written by function code 16. The data are read from the process image of the Master from the memory address(internal) 0 onwards. The writing process is only performed, if the data in the process image of the Master on the memory addresses(internal) 0 to 3 have changed, as the parameter trigger is set to 'changed'.

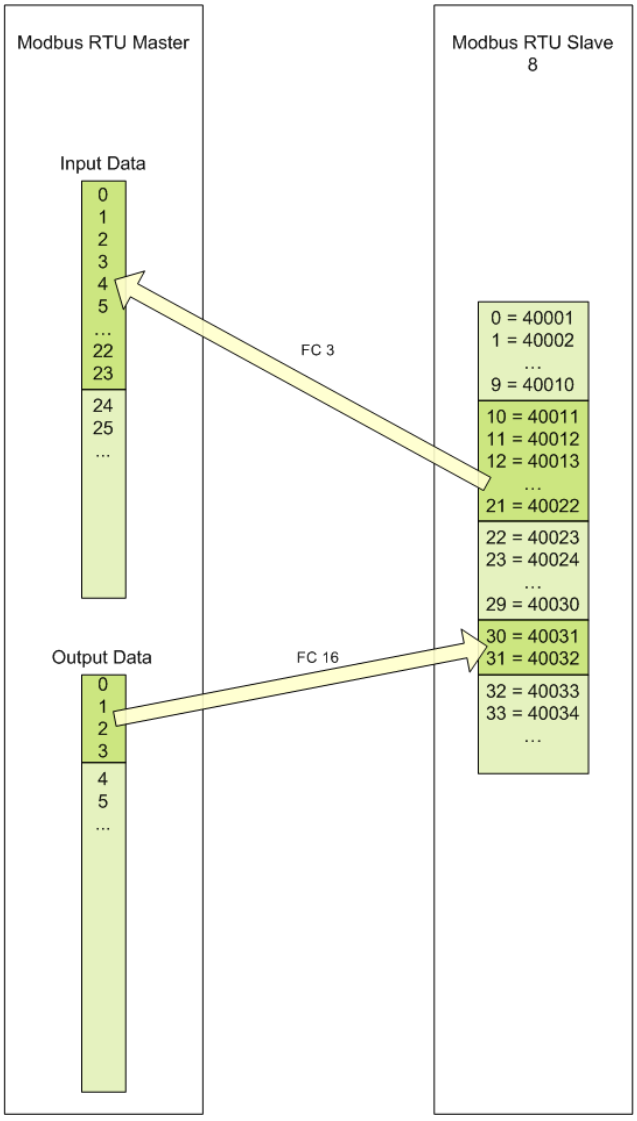


Figure 38: Examples - Reading Command with FC 3, Writing Command with FC16

5.6.2 Command Table Open Modbus/TCP

5.6.2.1 For what the Command Table does serve?

For the Open Modbus/TCP Client the **Command Table** is a list with commands for the reading or writing of data.

The **Command Table** is only relevant, if a device works as Client on the Open Modbus/TCP.

From every command line the Open Modbus/TCP Client produces a telegram for reading or for the writing of data to or from an Open Modbus/TCP Server device. For every command are indicated:

- the Open Modbus/TCP Server device address (Device Address),
- the Unit identifier (to identify a remote Server via gateway),
- the Function code,
- the data address in the Open Modbus/TCP Server device (Address),
- the number of data (Number of Registers/Coils),
- and the data address in the Open Modbus/TCP Client device (Memory Address (internal)).

For writing telegrams (FC 5, 6, 15 and 16) in the column

- Trigger

you can decide, whether the writing telegrams are executed every time (Cyclic) or only at data modification (Change data).

Furthermore

- a Cycle Time

can be set for every command.

The **Command Table** is processed from the first to the last entry (from above to below). After the execution of the last command the execution of the first command is started again.

For reading commands the Open Modbus/TCP Client reads out data from the Open Modbus/TCP Server and saves them into its data memory.

For writing commands the Open Modbus/TCP Client reads out data from its data memory and writes them into the Open Modbus/TCP Server.

The number of commands which can be defined depends from the firmware and from the Dual-Port Memory layout.

For Open Modbus one connection per IP address is established. In maximum 16 Client connections can be supported. In this case the parameter Open Server Sockets must be set from 4 (Default) to 0.

5.6.2.2 Opening Command Table

To view the **Table Command**, proceed as follows:

For Gateway devices:

1. In the netGateway DTM select **Configuration > Settings**.
2. In the **Settings** window under **Protocol Combination >**

➢ At **Primary network (Port X2)** select the primary network protocol.

➢ At **Secondary network (Port X3)** select the secondary network protocol Modbus/TCP.

➢ Select **OK**.

3. In the **netDevice** window put a right click on the DTM icon.
4. In the context menu select **Configuration > Open Modbus/TCP**.
5. In the **Configuration** window (Configuration Open Modbus/TCP), under **Protocol Mode** set the entry to "Client".

➢ Answer to the query “This and following changes will force recreation of signal configuration! Do you want to proceed?” with **Yes**.

6. In the navigation area select **Configuration > Command Table**.

↬ The window **Command Table** is displayed.

For devices without Gateway function:

1. In the **Configuration** window (Configuration Open Modbus/TCP), under **Protocol Mode** set the entry to "Client".

➢ Answer to the query “This and following changes will force recreation of signal configuration! Do you want to proceed?” with **Yes**.

2. In the navigation area select **Configuration > Command Table**.

↬ The window **Command Table** is displayed.

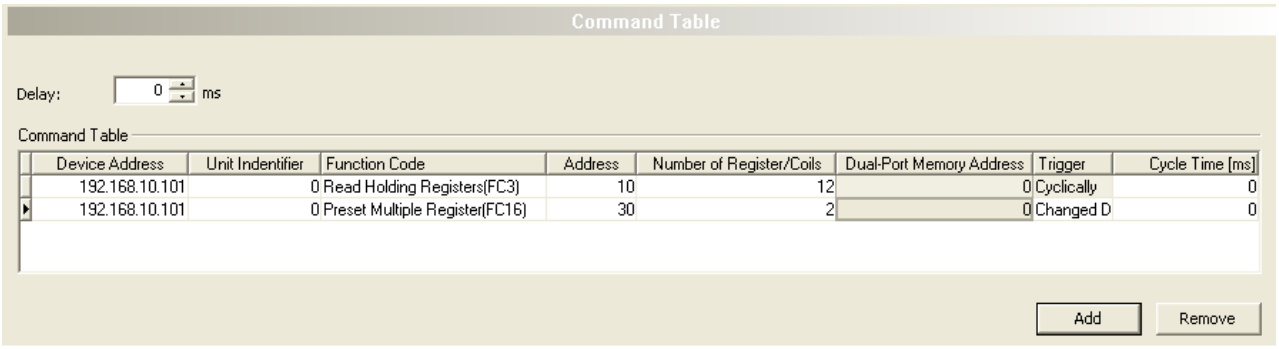


Figure 39: Window Command Table

Each command contains the following parameters:

Parameter	Description
Device Address	Open Modbus/TCP Server IP address
Unit Identifier	to identify a remote Server via gateway
Function Code	Function code
Address	data address in the Open Modbus/TCP Server device
Number of Registers/Coils	Number of data
Memory Address (internal)	Data address in the Open Modbus/TCP Client device

Parameter	Description
	(= Byte address in the process image of the Open Modbus/TCP Client device)
Trigger	Trigger for writing commands
Cycle Time	Cycle time

Table 55: Parameter of the Command Table

For a detailed description of the parameters refer to section *Parameter of the Command Table* beginning from page 113.

5.6.2.3 Parameter of the Command Table

<table>
<tr><th>Parameter</th><th>Meaning</th><th>Range of Value/Value</th></tr>
<tr><td>Device Address</td><td>Specifies the Open Modbus/TCP Server device address, from which the Open Modbus/TCP Client device reads out the data or into which it writes the data.</td><td>For Open Modbus/TCP:
000.000.000.000-
255.255.255.255,
Default: 000.000.000.000</td></tr>
<tr><td>Unit Identifier</td><td>To identify a remote Server connected on a serial line or on other buses.
The Unit Identifier is initialized by the Client. Do not change for response!</td><td>0 … 247,
Default: 0</td></tr>
<tr><td>Function Code</td><td>Indicates the function code for reading or writing commands [3]:
Read Coils(FC1)
Read Inputs(FC2)
Read Holding Registers(FC3)
Read Input Registers(FC4)
Force Single Coil(FC5)
Preset Single Register(FC6)
Force Multiple Coils(FC15)
Preset Multiple Registers(FC16)</td><td>1 / 2 / 3 / 4 / 5 / 6 / 15 / 16
Default: „Read Coils (FC1)“</td></tr>
<tr><td>Address</td><td>Indicates the data address in the Open Modbus/TCP Server device. The parameter Address contains the register address in the Open Modbus/TCP Server device. The register address is counted for every function code beginning with 0.
The following table shows the assignment of the Address to the data address in the Open Modbus/TCP Server by Function code:
<table>
<tr><th rowspan="2">Parameter
Address</th><th colspan="4">Data Address Open Modbus/TCP Server</th></tr>
<tr><th>FC 1
FC 5
FC 15</th><th>FC 2</th><th>FC 4</th><th>FC 3
FC 6
FC 16</th></tr>
<tr><td>0</td><td>1</td><td>10001</td><td>30001</td><td>40001</td></tr>
<tr><td>1</td><td>2</td><td>10002</td><td>30002</td><td>40002</td></tr>
<tr><td>2</td><td>3</td><td>10003</td><td>30003</td><td>40003</td></tr>
<tr><td>3</td><td>4</td><td>10004</td><td>30004</td><td>40004</td></tr>
<tr><td>4</td><td>5</td><td>10005</td><td>30005</td><td>40005</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
</table>

To know, which data address with which data of the Open Modbus/TCP Server devices is provided for reading or for writing, refer to the device description of the Open Modbus/TCP Server device manufacturer.</td><td>0 … 65.535
Default: 0</td></tr>
</table>

Parameter	Meaning	Range of Value/Value
Number of Registers/Coils	Indicates the number of reading or writing data as registers or coils. The maximum value depends from the function code.	1 … max. value, Default: 1 max. value for FC1 = 2000 (Coils), FC2 = 2000 (Coils), FC3 = 125 (Registers), FC4 = 125 (Registers), FC5 = 1 (Coils), FC6 = 1 (Registers), FC15 = 1968 (Coils), FC16 = 123 (Registers)
Memory Address (internal)	Byte address in the process image of the Open Modbus/TCP Client device for input or output data of the Client The configuration software calculates the byte address in the process image of the Client for the input or for the output data automatically. Because of this definition, for function codes for reading (FC 1, 2, 3 and 4) the data are assigned consecutively in the process image for input data and for function codes for writing (FC 5, 6, 15 and 16) the data are assigned consecutively in the process image for output data.	0 … 5759 Default: 0
Trigger	Defines for all function codes for writing (FC 5, 6, 15 and 16); if the command is to be executed cyclically (Cyclic) or only if the data have changed (Change Data). For function codes for reading (FC 1, 2, 3 und 4) this parameter is not used.	0 = Cyclic 1 = Changed Data Default: „Cyclic“
Cycle Time	The cycle time defines, after how many milliseconds a writing or a reading command shall be reexecuted. The default value „0 ms“ means, that the command is executed as fast as possible. Otherwise the execution can be defined in steps of 10 ms. With the cycle time the temporal execution of the command can be influenced. If since the last execution of the command the set cycle time or more time has been passed, the command is reexecuted. If since the last execution of the command less time has been passed then the cycle time, the command is skipped. *If e. g. from a Open Modbus/TCP Server data are required only every 10 seconds, as these one change only slowly, then for the cycle time enter 10.000 ms.*	0, 10, 20, 30, … 60.000 ms Default: 0 ms
Delay	Between the single commands a delay time can be parameterized. This sometimes is necessary to avoid a too high load of the connected Server devices by a continuous communication. **0**: The commands are processed without delay. **1 … 60.000 ms**: Delay time in ms, the Open Modbus/TCP Client waits, before starting the next command.	0 … 60.000 ms Default: 0

Table 56: Parameter of the Command Table

5.6.2.4 Adding/removing Command

To add or to remove data sets for commands to the Command Table, proceed as follows:

Add command:

- In the table **Command Table** put the cursor in the line below which a new data set for a command shall be added.
- Click **Add**.

↬ Under the clicked line a new command is added.

Remove command:

- In the table **Command Table** put the cursor in the line with the data set to be deleted.
- Click **Remove**.

↬ The command of the clicked line is not shown any more.

5.6.2.5 Setting Parameters

Important: When entering the parameters, make sure that these address valid data registers. (*Refer to the device description of the Open Modbus/TCP Server device manufacturer.*)

To set the single parameters in the Command Table, proceed as follows:

1. **Device Address**, **Function Code**, **Address**, **Number of Registers/Coils**:

The fields for these parameters are editable.

- Enter the values for these parameters to the corresponding table cells.

2. **Trigger**:

- In **Function Code** field select an entry for a writing command (FC 5, 6, 15 or 16).
- In the column **Trigger** select the entry „Cyclic“ or „Change Data“.

3. **Cycle Time**:

- Put the cursor to the table cell and set the cycle time using the spin box in steps of 10 ms.

5.6.2.6 Examples Open Modbus/TCP Reading or Writing Command

Device Address	Unit Indentifier	Function Code	Address	Number of Register/Coils	Dual-Port Memory Address	Trigger	Cycle Time [ms]
192.168.10.101	0	Read Holding Registers(FC3)	10	12	0	Cyclically	0
192.168.10.101	0	Preset Multiple Register(FC16)	30	2	0	Changed D	0

Figure 40: Examples - Reading Command with FC 3, Writing Command with FC16

Example Reading Command with FC 3: From the Open Modbus/TCP Server device with the Modbus address 8 from data address 40011 onwards 12 registers are read by function code 3. The data are assigned to the memory address (internal) 0.

Example Writing Command with FC 16: To the Open Modbus/TCP Server with the Modbus address 8 from data address 40031 onwards 12 registers are written by function code 16. The data are read from the process image of the Client from the memory address (internal) 0 onwards. The writing process is only performed, if the data in the process image of the Client on the memory addresses (internal) 0 to 3 have changed, as the parameter trigger is set to 'changed'.

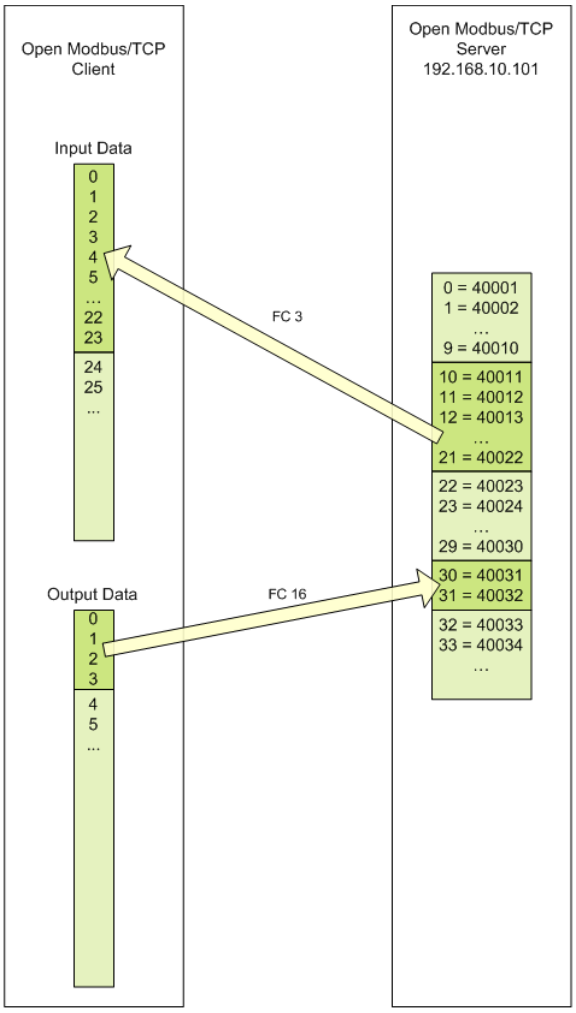


Figure 41: Examples - Reading Command with FC 3, Writing Command with FC16

5.6.3 Modbus References

[1] MODBUS Application Protocol Specification V1.1, http://www.modbus.org/, 12/06/02

[2] MODBUS Messaging on TCP/IP Implementation Guide V1.0b, October 24, 2006

[3] MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1a, June 4, 2004, http://www.Modbus-IDA.org

6 Online Functions

6.1 Connecting/Disconnecting Device

Note: Several netSLAVE DTM functions e. g. **Diagnosis** or the configuration download in the FDT Framework require an online connection from the netSLAVE DTM to the Slave device.

Connecting Device

The following steps are needed to establish a connection from the netSLAVE DTM to a Slave device:

Under **Settings** in the **Driver** window:

1. Verify that the default driver is checked and respectively check another or multiple drivers.
2. Configure the driver if necessary.

Under **Settings** in the **Device Assignment** window:

3. Scan for the devices (with or without firmware).
4. Select the device (with or without firmware) and apply the selection.

Before you download the firmware adhere to the necessary safety precautions to prevent personnel injury and property damage. For more refer to section *Safety Messages on Firmware or Configuration Download* on page 18.

Under **Settings** in the **Firmware Download** window if not yet a firmware was loaded to the device:

5. Select and download the firmware.

Under **Settings** in the **Device Assignment** window if not yet a firmware was loaded to the device:

6. Scan for the device (with firmware) once more.
7. Select the device (with firmware) once more.

An overview of the descriptions for these steps you find in the section *Overview Settings* on page 24.

8. In the DTM interface dialog select the **OK** button, to apply the selection and to close the DTM interface dialog.
9. Put a right-click on the Slave device icon.
10. Select the **Connect** command from the context menu.

➮ The Slave device now is connected to the netSLAVE DTM via an online connection. In the network view the device description at the device icon of the Slave is displayed with a green colored background.

Disconnecting Device

To disconnect an online connection from the Slave device to a netSLAVE DTM take the following steps:

1. In the DTM interface dialog select the **OK** button, to close the DTM interface dialog.
2. Right-click on the Slave device icon.
3. Select the **Disconnect** command from the context menu.

↛ In the network view the device description at the device icon of the Slave device is not any more displayed with a green colored background. Now the Slave device is disconnected from the DTM.

6.2 Download Configuration

The device configuration is created *offline* in the DTM (application program). A download to the device is required, to transfer the configuration with the parameter data to the device.

Note: To download configuration parameter data to the Slave device an online connection from the netSLAVE Slave DTM to the Slave device is required. Further information can be found in the *Connecting/Disconnecting Device* section on page 118.

Safety Precautions

If you plan to perform a configuration download via the netSLAVE Slave DTM be aware of the following:

WARNING

Communication Stop due to Configuration Download,
Faulty System Operation possible or Loss of Device Parameters

Before you initiate a configuration dowload process, while the bus is still in operation status:

- Stop the application program.
- Make sure that all network devices are placed in a fail-safe condition.

WARNING

Mismatching System Configuration,
faulty System or Device Operation possible

- In the device use only a configuration suitable for the system.

NOTICE

Loss of Parameters caused by Power Disconnect during Configuration Download

- During configuration download process, do not interrupt the power supply to the PC, or to the device and do not perform a reset to the device!

For more see next page.

Download Steps

To transfer the configuration with the corresponding parameter data to the Slave device you download the data using the frame application of the configuration software.

For netDevice the download is made via **Device** > **Download** or use the context menu with **Download**.

1. Select **Download** in the context menu of the device.

↬ If the download is started as long as the Slave devices are connected to the Master device, the following message is displayed: 'If you attempt to download during bus operation, communication between Master and Slaves is stopped. Do you really want to download?'

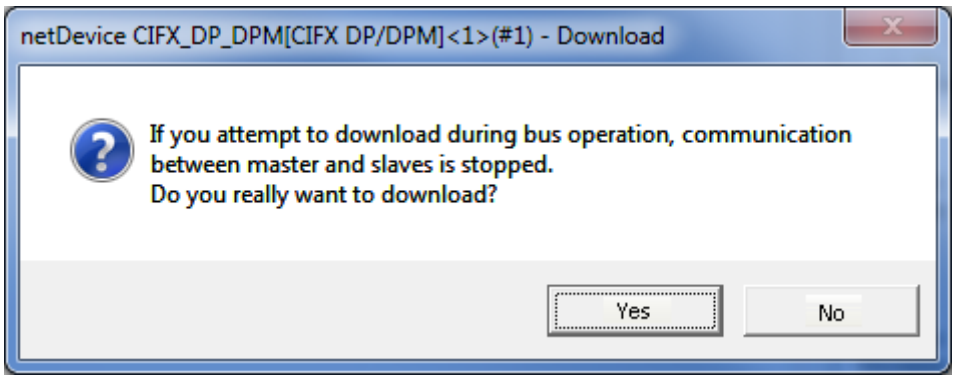


Figure 42: netDevice Message: Download

Important: If the communication between the Master and the Slave devices is stopped, the data exchange between the Master device and the Slave devices is stopped.

2. Click to **Yes** if you intend to download the configuration.

↬ Then the current configuration in the application program is down loaded to the device.

3. Otherwise click to **No**.

7 Diagnosis

7.1 Overview Diagnosis

The dialog **Diagnosis** serves to diagnose the device behavior and communication errors. For diagnosis the device must reside in online state.

Diagnosis Windows

The table below gives an overview for the individual **Diagnosis** dialog windows descriptions for the respective communication system.

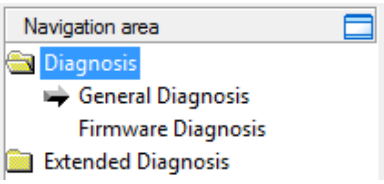
netSLAVe DTM	Section	Page
	*General Diagnosis*	123
	Firmware Diagnosis	125

Table 57: Descriptions of the Diagnosis Windows

Online Connection to the Device

Note: Accessing the **Diagnosis** windows of the netSLAVE DTM requires an online connection from the netSLAVE DTM to the Slave device . For further information, refer to section *Connecting/Disconnecting Device* on page 118.

Extended Diagnosis

The **Extended Diagnosis** helps to find communication and configuration errors, when default diagnosis fails. For further information refer to section *Overview Extended Diagnosis* on page 126.

7.2 General Diagnosis

Information regarding the Device State and other general diagnosis parameters are displayed in the **General Diagnosis** dialog.

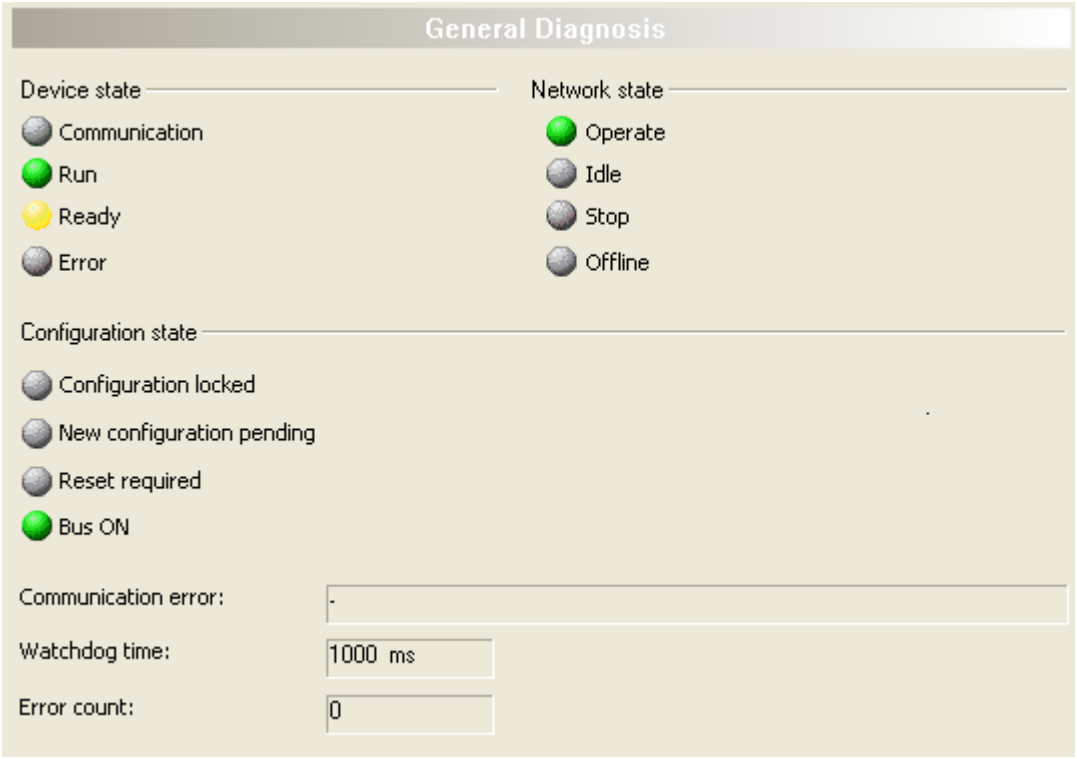


Figure 43: General Diagnosis

LED	Meaning	Color	State
Device State			
Communication	Shows whether the Slave device executes the network communication.	(green)	In COMMUNICATION state
		(gray)	Not in COMMUNICATION state
Run	Shows whether the Slave device has been configured correctly.	(green)	Configuration OK
		(gray)	Configuration not OK
Ready	Shows whether the Slave device has been started correctly. The Slave device waits for a configuration.	(yellow)	Device READY
		(gray)	Device not READY
Error	Shows whether the Slave device records a device status error. For further information about the error characteristics and the number of counted errors refer to the extended diagnosis.	(red)	ERROR
		(gray)	No ERROR
Network State			
Operate	Shows whether the Slave device is in data exchange. In a cyclic data exchange the input data or the output data of the Slave are transmitted to the Master.	(green)	In OPERATION state
		(gray)	Not in OPERATION state
Idle	Shows whether the Slave device is in idle state.	(yellow)	In IDLE state
		(gray)	Not in IDLE state

LED	Meaning	Color	State
Stop	Shows whether the Slave device is in Stop state: There is no cyclic data exchange at the network. The Slave device was stopped by the application program or it changed to the Stop state because of a bus error.	(red)	In STOP state
		(gray)	Not in STOP state
Offline	The Slave device is offline as long as it does not have a valid configuration.	(yellow)	In OFFLINE state
		(gray)	Not in OFFLINE state
Configuration State			
Configuration locked	Shows whether the Slave device configuration is locked, to avoid the configuration data are typed over.	(yellow)	Configuration LOCKED
		(gray)	Configuration not LOCKED
New Configuration pending	Shows whether a new Slave device configuration is available.	(yellow)	New Configuration pending
		(gray)	No new Configuration pending
Reset required	Shows whether a firmware reset is required as a new Slave device configuration has been loaded into the device.	(yellow)	RESET required
		(gray)	No RESET required
Bus ON	Shows whether the bus communication was started or stopped. I. e., whether the device is active on the bus or no bus communication to the device is possible and no response messages are sent.	(green)	Bus ON
		(gray)	Bus OFF

Table 58: Indication General Diagnosis

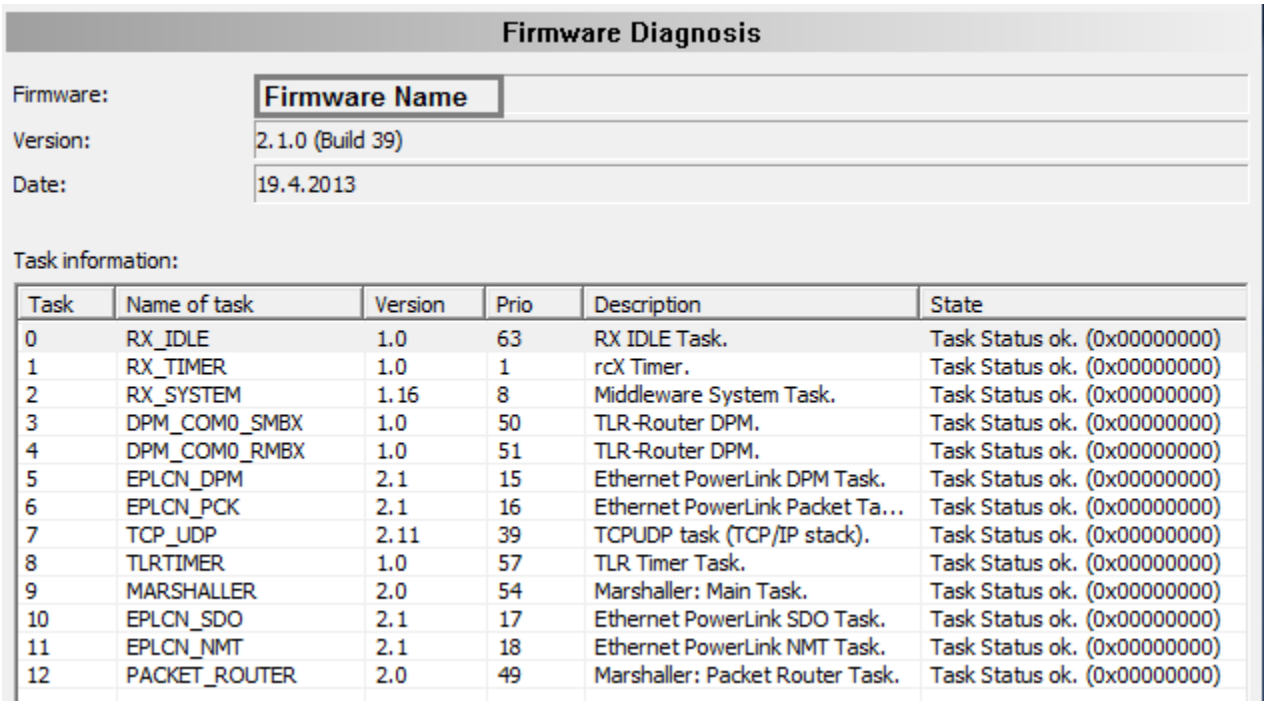
Parameter	Meaning
Communication Error	Shows the name of the communication error. If the cause of error is resolved, the value will be set to zero again.
Watchdog time	Shows the watchdog time in ms.
Error Count	This field holds the total number of errors detected since power-up, respectively after reset. The protocol stack counts all sorts of errors in this field no matter whether they were network related or caused internally.

Table 59: Parameter General Diagnosis

7.3 Firmware Diagnosis

In the dialog **Firmware Diagnosis** the current task information of the firmware is displayed.

Under **Firmware** or **Version** the name of the firmware and version (including the date) are indicated.



Firmware Diagnosis

Firmware: Firmware Name

Version: 2.1.0 (Build 39)

Date: 19.4.2013

Task information:

Task	Name of task	Version	Prio	Description	State
0	RX_IDLE	1.0	63	RX IDLE Task.	Task Status ok. (0x00000000)
1	RX_TIMER	1.0	1	rcX Timer.	Task Status ok. (0x00000000)
2	RX_SYSTEM	1.16	8	Middleware System Task.	Task Status ok. (0x00000000)
3	DPM_COM0_SMBX	1.0	50	TLR-Router DPM.	Task Status ok. (0x00000000)
4	DPM_COM0_RMBX	1.0	51	TLR-Router DPM.	Task Status ok. (0x00000000)
5	EPLCN_DPM	2.1	15	Ethernet PowerLink DPM Task.	Task Status ok. (0x00000000)
6	EPLCN_PCK	2.1	16	Ethernet PowerLink Packet Ta...	Task Status ok. (0x00000000)
7	TCP_UDP	2.11	39	TCPUDP task (TCP/IP stack).	Task Status ok. (0x00000000)
8	TLRTIMER	1.0	57	TLR Timer Task.	Task Status ok. (0x00000000)
9	MARSHALLER	2.0	54	Marshaller: Main Task.	Task Status ok. (0x00000000)
10	EPLCN_SDO	2.1	17	Ethernet PowerLink SDO Task.	Task Status ok. (0x00000000)
11	EPLCN_NMT	2.1	18	Ethernet PowerLink NMT Task.	Task Status ok. (0x00000000)
12	PACKET_ROUTER	2.0	49	Marshaller: Packet Router Task.	Task Status ok. (0x00000000)

Figure 44: Firmware Diagnosis, Example

Task Information:

The table **Task Information** is listing the task information of the single firmware tasks.

Column	Meaning
Task	Task number
Name of task	Name of the task
Version	Version number of the task
Prio	Priority of the task
Description	Description of the task
Status	Current status of the task

Table 60: Description Table Task Information

8 Extended Diagnosis

8.1 Overview Extended Diagnosis

The **Extended Diagnosis** of the netSLAVE DTM helps to find communication and configuration errors. Therefore it contains a list of diagnosis structures as online counter, states and parameters.

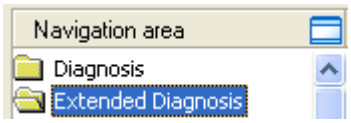


Figure 45: Navigation Area - Extended Diagnosis

Note: Accessing the **Extended Diagnosis** dialog windows of the netSLAVE DTM requires an online connection from the netSLAVE DTM to the Slave device . For further information, refer to section *Connecting/Disconnecting Device* on page 118.

Overview Dialog Windows "Extended Diagnosis"

For an overview of the descriptions of the dialogue windows under **Extended Diagnosis** for every communication system refer to the following subsections. All descriptions have exemplary character and may differ in detail for the single devices.

Type of System	Subsection/ Communication System	Manual Page
Real-Time Ethernet Systems	*EtherCAT Slave**	127
	*EtherNet/IP Adapter**	132
	Open Modbus/TCP	140
	*PROFINET IO Device**	145
	POWERLINK Controlled Node	149
	Sercos	150
	VARAN Client (Slave)	151
Fieldbus Systems	*PROFIBUS DP Slave**	158
	*CANopen Slave**	165
	*DeviceNet-Slave**	179
	CC-Link Slave	185
	CompoNet Slave	204
Serial Protocols	*3964R Slave**	208

*Table 61: Overview Extended Diagnosis (*only Gateways)*

For the descriptions of repeated used tasks refer to section *Descriptions for Tasks with similar Functions* on page 215.
For information about the parameters in the dialogue windows **Extended Diagnosis** refer also to the descriptions of the configuration parameters in section *Real-Time Ethernet Systems* beginning from page 53 and in section beginning from page *Fieldbus Systems* beginning from page 69, as well as to the corresponding **User Manual** of your device, in the section *Technical Data of the Communication Protocols*. This manual is located on the included DVD to your device.

8.2 Real-Time Ethernet Systems

8.2.1 EtherCAT Slave

8.2.1.1 Overview EtherCAT Slave

Dialog Windows "Extended Diagnosis" EtherCAT Slave

The table below gives an overview for the **Extended Diagnosis** dialog windows descriptions:

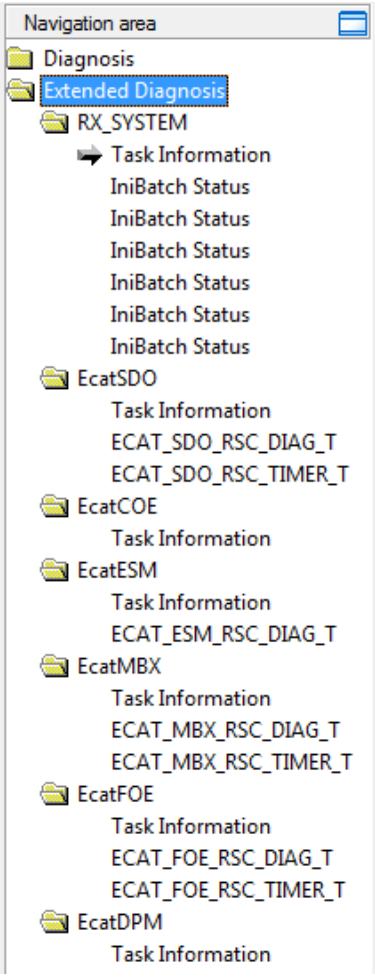
Navigation Area EtherCAT-Slave*	Folder Name in the Navigation Area	Dialog Window	Manual Page
	*EcatSDO*	*EcatSDO\ECAT_SDO_RSC_DIAG_T*	128
		EcatSDO\ECAT_SDO_RSC_TIMER_T	128
	EcatESM	*EcatESM\ECAT_ESM_RSC_DIAG_T*	129
	EcatMBX	*EcatMBX\ECAT_MBX_RSC_DIAG_T*	129
		EcatMBX\ECAT_MBX_RSC_TIMER_T	130
	EcatFOE	*EcatFOE\ECAT_FOE_RSC_DIAG_T*	130
		EcatFOE\ECAT_FOE_RSC_TIMER_T	131
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		** The displayed EtherCAT Slave exteded diagnosis corresponds to the EtherCAT Slave firmware version 2.5.x.x*	

Table 62: Descriptions of the Dialog Windows Extended Diagnosis EtherCAT Slave

8.2.1.2 EcatSDO\ECAT_SDO_RSC_DIAG_T

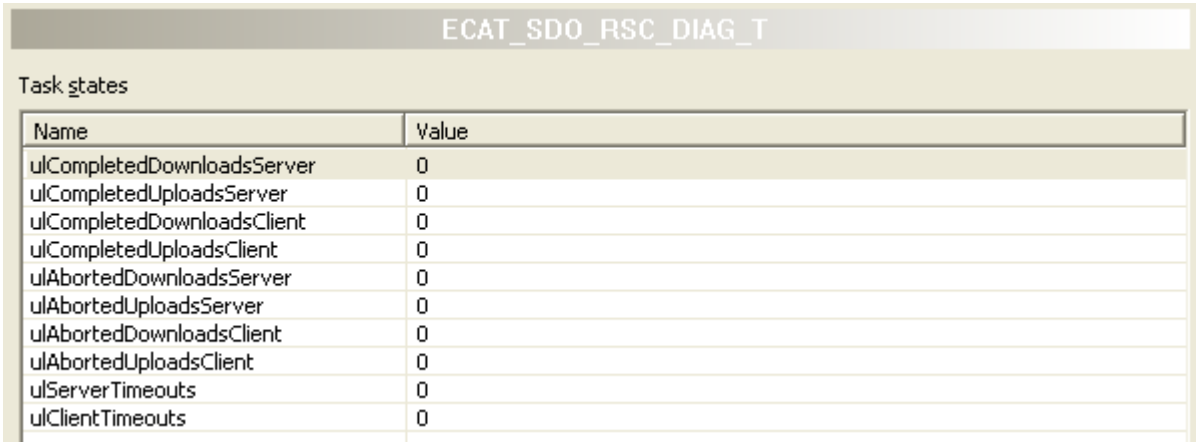


Figure 46: Extended Diagnosis EtherCAT Slave > EcatSDO > ECAT_SDO_RSC_DIAG_T

Name	Description
[*Service*]	Diagnosis counter of the ECAT_SDO_RSC_DIAG_T layer. Indicates the services processed. (The services of the single packets are described in the EtherCAT Slave Protocol API manual [9].)

Table 63: Extended Diagnosis EtherCAT Slave > EcatSDO > ECAT_SDO_RSC_DIAG_T

8.2.1.3 EcatSDO\ECAT_SDO_RSC_TIMER_T

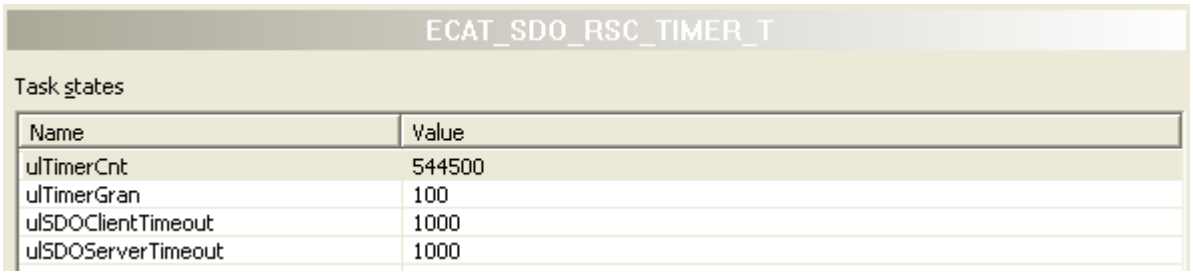


Figure 47: Extended Diagnosis EtherCAT Slave > EcatSDO > ECAT_SDO_RSC_TIMER_T

Name	Description
ulTimerCnt	Timer Count
ultimerGran	Granularity of the timer
ulSDOClientTimeout	Clients
ulSDOServerTimeout	Timeout value of the SDO Server

Table 64: Extended Diagnosis EtherCAT Slave > EcatSDO > ECAT_SDO_RSC_TIMER_T

8.2.1.4 EcatESM\ECAT_ESM_RSC_DIAG_T

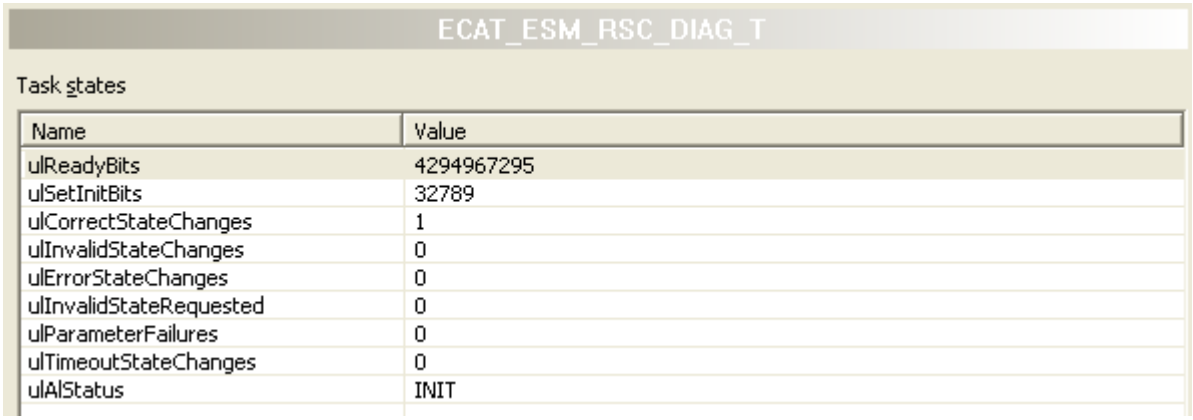


Figure 48: Extended Diagnosis EtherCAT Slave > EcatESM > ECAT_ESM_RSC_DIAG_T

Name	Description
ulReadyBits	Number of Ready Bits
ulSetInitBits	Number of SetInit Bits
[*Service*]	Diagnosis counter of the ECAT_ESM_RSC_DIAG_T layer. Indicates the services processed. (The services of the single packets are described in the EtherCAT Slave Protocol API manual [9].)
ulAlStatus	AL status, for more detailed information, see the EtherCAT Specification, Part 5 and 6

Table 65: Extended Diagnosis EtherCAT Slave > EcatESM > ECAT_ESM_RSC_DIAG_T

8.2.1.5 EcatMBX\ECAT_MBX_RSC_DIAG_T

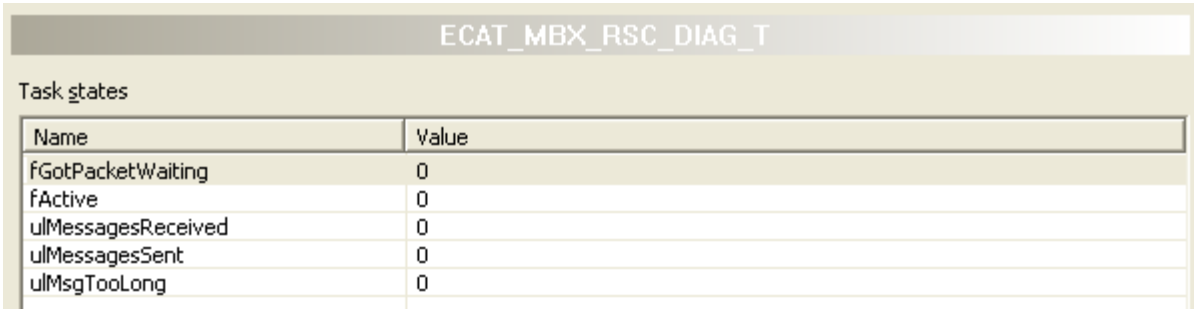


Figure 49: Extended Diagnosis EtherCAT Slave > EcatMBX > ECAT_MBX_RSC_DIAG_T

Name	Description
[*Service*]	Diagnosis counter of the ECAT_MBX_RSC_DIAG_T layer. Indicates the services processed. (The services of the single packets are described in the EtherCAT Slave Protocol API manual [9].)

Table 66: Extended Diagnosis EtherCAT Slave > EcatMBX > ECAT_MBX_RSC_DIAG_T

8.2.1.6 EcatMBX\ECAT_MBX_RSC_TIMER_T

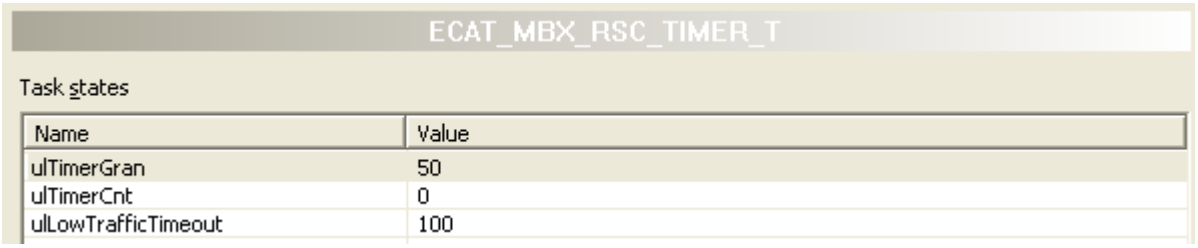


Figure 50: Extended Diagnosis EtherCAT Slave > EcatMBX > ECAT_MBX_RSC_TIMER_T

Name	Description
ulTimerCnt	Timer Count
ulTimerGran	Granularity of the timer
ulLowTrafficTimeout	Timeout value for low network data traffic

Table 67: Extended Diagnosis EtherCAT Slave > EcatMBX > ECAT_MBX_RSC_TIMER_T

8.2.1.7 EcatFOE\ECAT_FOE_RSC_DIAG_T

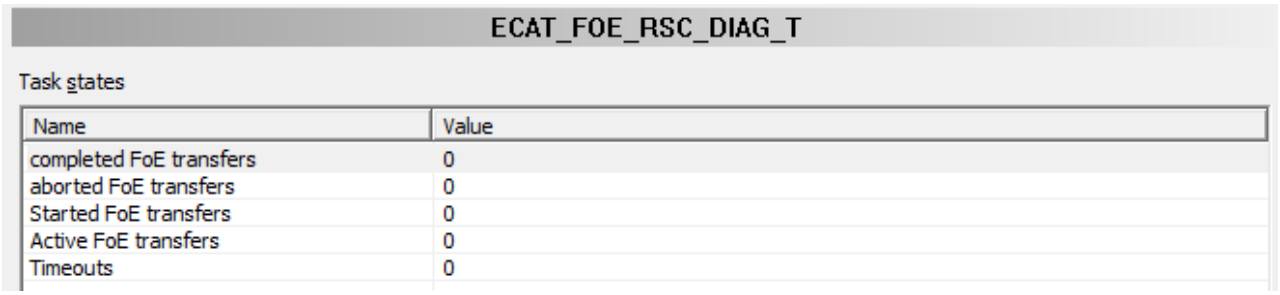


Figure 51: Extended Diagnosis EtherCAT Slave > EcatMBX > ECAT_FOE_RSC_DIAG_T

Name	Description
[*Service*]	Diagnosis counter of the ECAT_FOE_RSC_DIAG_T layer. Indicates the services processed. (The services of the single packets are described in the EtherCAT Slave Protocol API manual [9].)

Table 68: Extended Diagnosis EtherCAT Slave > EcatMBX > ECAT_FOE_RSC_DIAG_T

8.2.1.8 EcatFOE\ECAT_FOE_RSC_TIMER_T

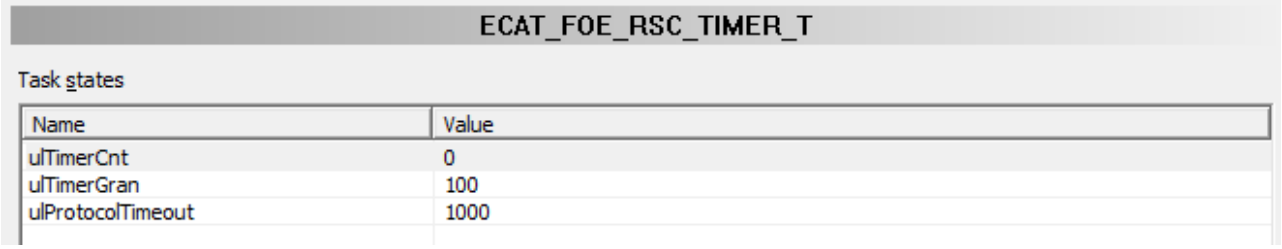


Figure 52: Extended Diagnosis EtherCAT Slave > EcatMBX > ECAT_FOE_RSC_TIMER_T

Name	Description
ulTimerCnt	Timer Count
ulTimerGran	Granularity of the timer
ulLowTrafficTimeout	Timeout value for low network data traffic

Table 69: Extended Diagnosis EtherCAT Slave > EcatMBX > ECAT_FOE_RSC_TIMER_T

8.2.2 EtherNet/IP Adapter

8.2.2.1 Overview EtherNet/IP Adapter

Dialog Windows "Extended Diagnosis" EtherNet/IP Adapter

The table below gives an overview for the **Extended Diagnosis** dialog windows descriptions:

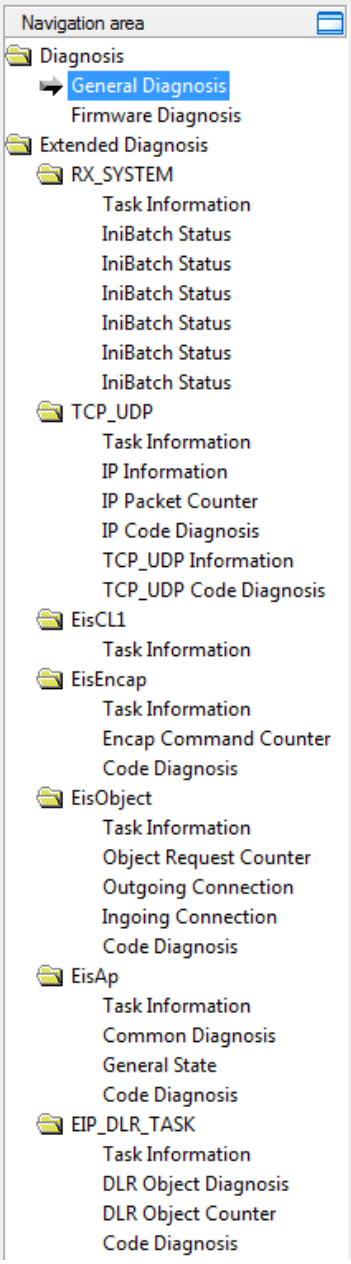
Navigation Area EtherNet/IP Adapter*	Folder Name in the Navigation Area	Dialog Window	Manual Page
	*EisEncap*	*EisEncap/Encap Command Counter*	133
	EisObject	*EisObject/Object Request Counter*	134
		EisObject/Outgoing Connection	135
		EisObject/Ingoing Connection	135
	EisAp	*EisAp/Common Diagnosis*	136
		EisAp/General Stat	136
	EIP_DLR_TASK	*EIP_DLR/DLR Object Diagnosis*	137
		EIS_DLR/DLR Object Counter	139
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		** The displayed EtherNet/IP Adapter exteded diagnosis corresponds to the EtherNet/IP Adapter firmware version 2.6.x.x.*	

Table 70: Descriptions of the Dialog Windows Extended Diagnosis EtherNet/IP Adapter

8.2.2.2 EisEncap/Encap Command Counter

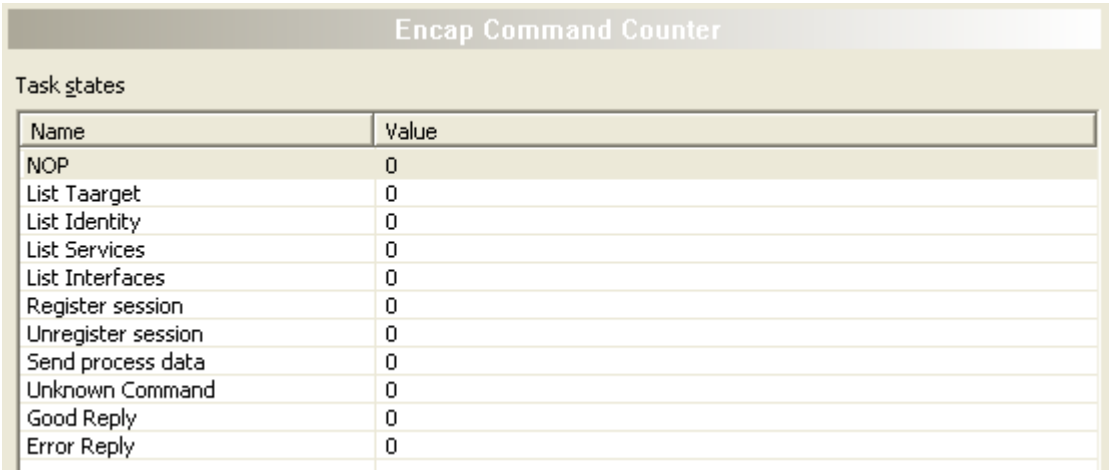


Figure 53: Extended Diagnosis EtherNet/IP Adapter > EisEncap > Encap Command Counter

Name	Description
NOP	Counter incoming encapsulation telegrams
List Target	Counter incoming encapsulation telegrams
List Identity	Counter incoming encapsulation telegrams
List Services	Counter incoming encapsulation telegrams
List Interfaces	Counter incoming encapsulation telegrams
Register Session	Counter incoming encapsulation telegrams
Unregister Session	Counter incoming encapsulation telegrams
Process data	Unit Date RR Data telegrams
Unknown Command	Counter incoming encapsulation telegrams
Good Reply	Counter incoming encapsulation telegrams
Error Reply	Counter incoming encapsulation telegrams

Table 71: Extended Diagnosis EtherNet/IP Adapter > EisEncap > Encap Command Counter

8.2.2.3 EisObject/Object Request Counter

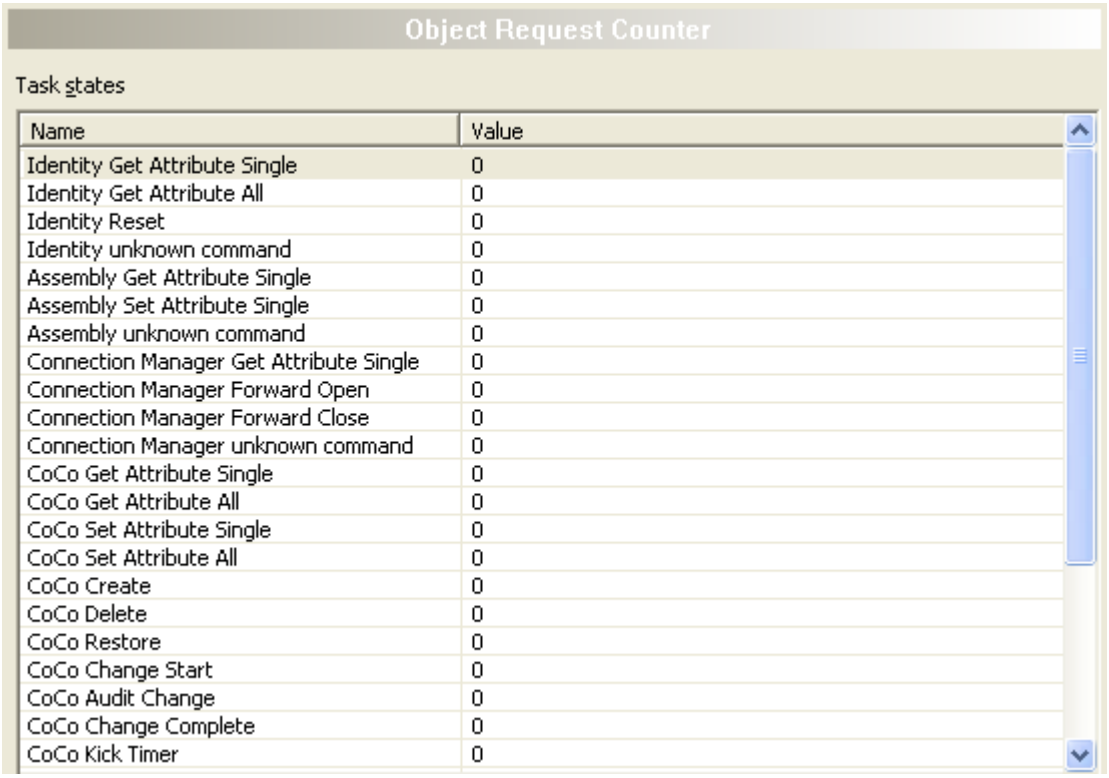


Figure 54: Extended Diagnosis EtherNet/IP Adapter > EisObject > Object Request Counter

The **Object Request Counter** table indicates the respective number of services sent to different objects.

Name	Description
Identity Get Attribute Single	Counter for the *Get Attribute Single* services to the *identity* object
Identity Get Attribute All	Counter for the *Get Attribute All* services to the *identity* object
Identity Reset	Counter for the *Reset* services to the *identity* object
Identity unknown Command	Counter for the *Unknown Command* services to the *identity* object
Assembly Get Attribute Single	Counter for the *Get Attribute Single* services to the *assembly* object
[*Object*] [Service]	Counter for the [*Object*] services to the [*Service*] object

Table 72: Extended Diagnosis EtherNet/IP Adapter > EisObject > Object Request Counter

8.2.2.4 EisObject/Outgoing Connection

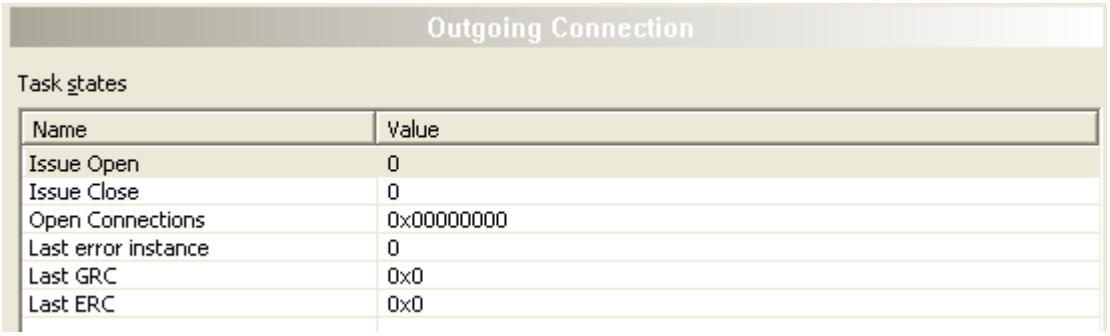


Figure 55: Extended Diagnosis EtherNet/IP Adapter > EisObject > Outgoing Connection

Name	Description
Issue Open	Counter for open issues
Issue Close	Counter for closed issues
Open Connections	Actual open connection
Last error instance	Last error instance
Last GRC	Last General Status Code
Last ERC	Last Extended Status Code

Table 73: Extended Diagnosis EtherNet/IP Adapter > EisObject > Outgoing Connection

8.2.2.5 EisObject/Ingoing Connection

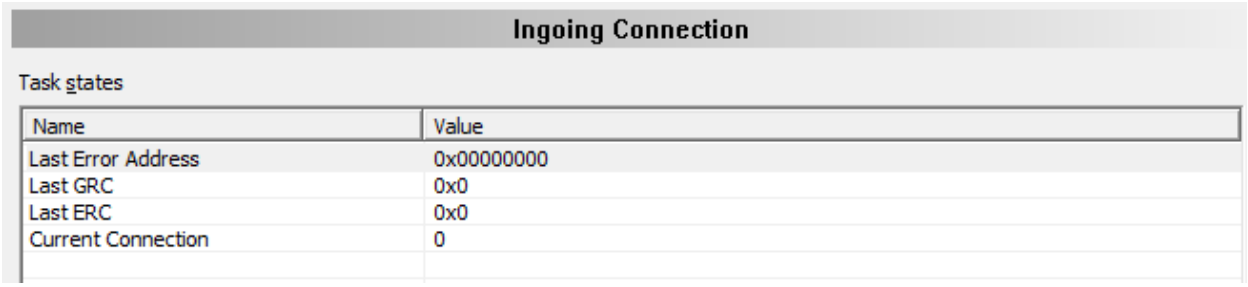


Figure 56: Extended Diagnosis EtherNet/IP Adapter > EisObject > Ingoing Connection

Name	Description
Last Errors Address	IP Address
Last GRC	Last General Status Code
Last ERC	Last Extended Status Code
Actual Connection	Actual open connection

Table 74: Extended Diagnosis EtherNet/IP Adapter > EisObject > Ingoing Connection

8.2.2.6 EisAp/Common Diagnosis

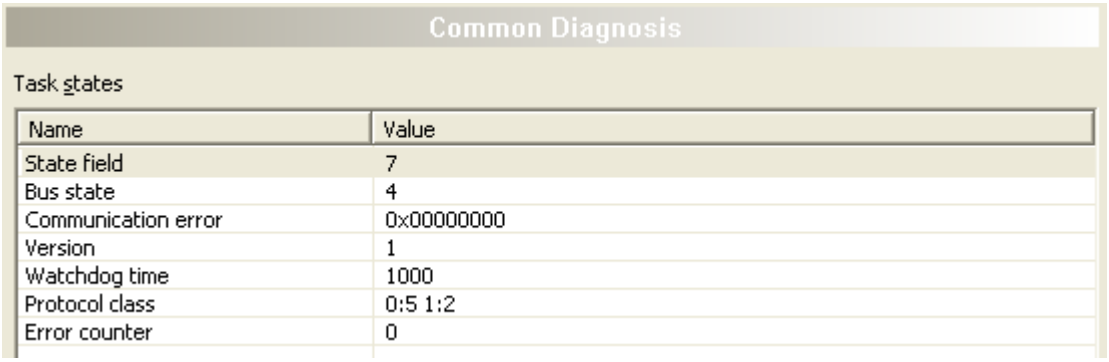


Figure 57: Extended Diagnosis EtherNet/IP Adapter > EisAp > Common Diagnostic

Name	Description
Status field	Communication change of state
Bus state	Communication state
Communication Error	Unique error number according to protocol stack (not supported yet)
Version	Version number of this diagnosis structure
Watchdog time	Configured watchdog time
Protocol Class	Protocol class: MASTER, SLAVE, CLIENT, SERVER, GATEWAY
Error Counter	Total number of detected error since power-up or reset

Table 75: Extended Diagnosis EtherNet/IP Adapter > EisAp > Common Diagnostic

8.2.2.7 EisAp/General Status

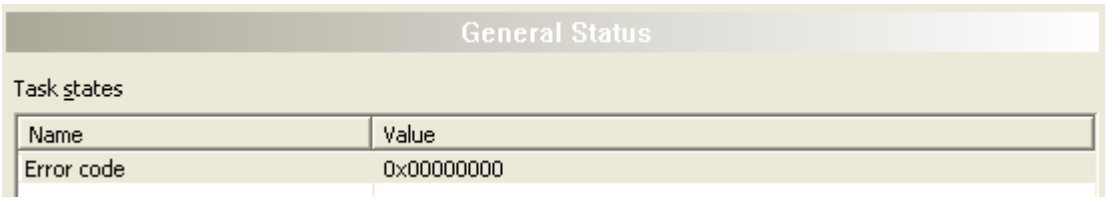
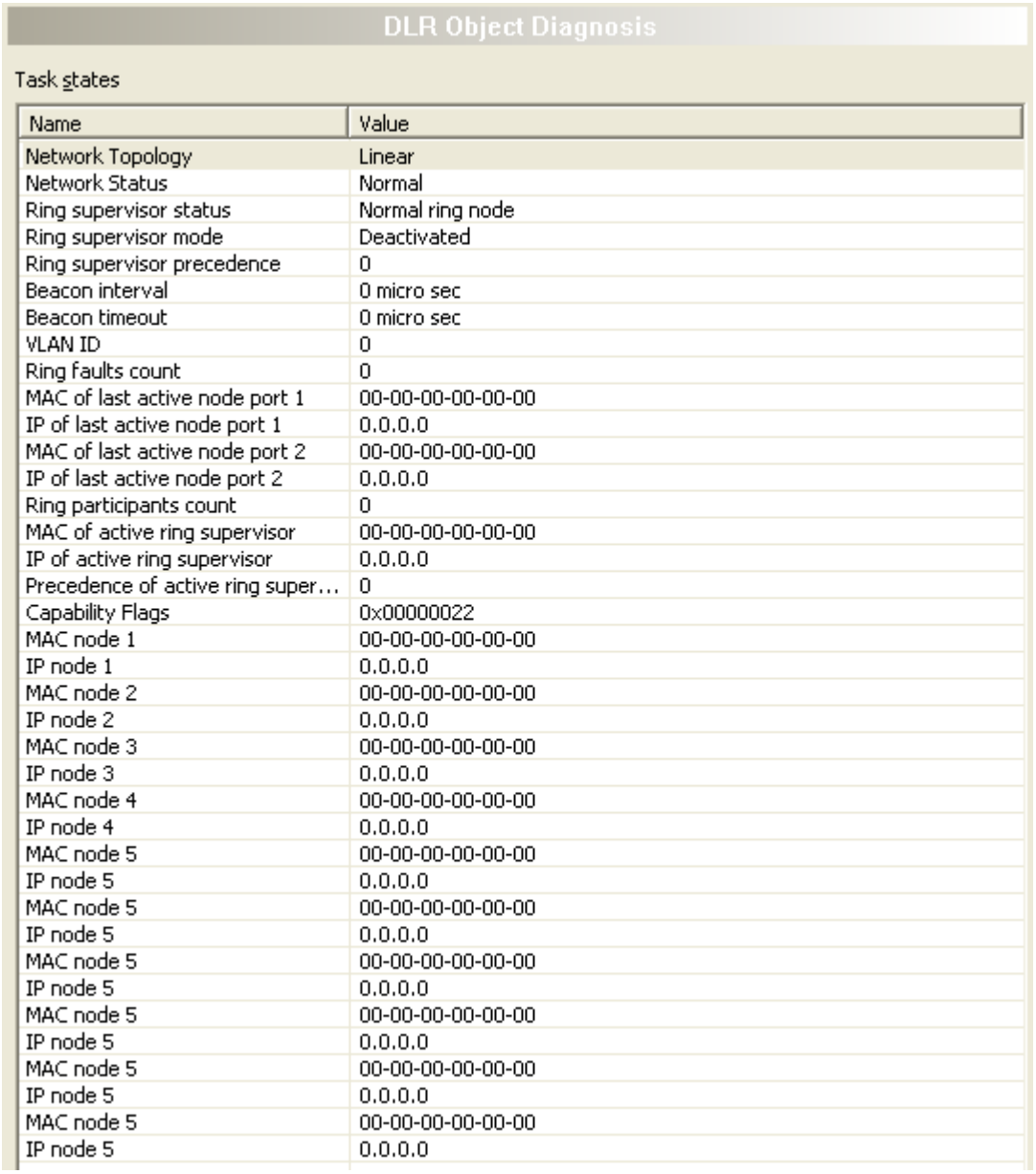


Figure 58: Extended Diagnosis EtherNet/IP Adapter > EisAp > General Status

Name	Description
Error Code	Code of the last occurred error

Table 76: Extended Diagnosis EtherNet/IP Adapter > EisAp > General Status

8.2.2.8 EIP_DLR/DLR Object Diagnosis



Name	Value
Network Topology	Linear
Network Status	Normal
Ring supervisor status	Normal ring node
Ring supervisor mode	Deactivated
Ring supervisor precedence	0
Beacon interval	0 micro sec
Beacon timeout	0 micro sec
VLAN ID	0
Ring faults count	0
MAC of last active node port 1	00-00-00-00-00-00
IP of last active node port 1	0.0.0.0
MAC of last active node port 2	00-00-00-00-00-00
IP of last active node port 2	0.0.0.0
Ring participants count	0
MAC of active ring supervisor	00-00-00-00-00-00
IP of active ring supervisor	0.0.0.0
Precedence of active ring super…	0
Capability Flags	0x00000022
MAC node 1	00-00-00-00-00-00
IP node 1	0.0.0.0
MAC node 2	00-00-00-00-00-00
IP node 2	0.0.0.0
MAC node 3	00-00-00-00-00-00
IP node 3	0.0.0.0
MAC node 4	00-00-00-00-00-00
IP node 4	0.0.0.0
MAC node 5	00-00-00-00-00-00
IP node 5	0.0.0.0
MAC node 5	00-00-00-00-00-00
IP node 5	0.0.0.0
MAC node 5	00-00-00-00-00-00
IP node 5	0.0.0.0
MAC node 5	00-00-00-00-00-00
IP node 5	0.0.0.0
MAC node 5	00-00-00-00-00-00
IP node 5	0.0.0.0
MAC node 5	00-00-00-00-00-00
IP node 5	0.0.0.0

Figure 59: Extended Diagnosis EtherNet/IP Adapter > EIP_DLR > DLR Object Diagnosis

Name	Description
Network Topology	Possible Values: Linear or Ring
Network Status	Possible Values: Normal, Ring fault, Unexpected loop detected, Partial network fault, Rapid fault/Restore cycle
Ring Supervisor Status	Possible Values: Backup, Active ring supervisor, Normal ring node, None DLR device, Current supervisor parameters not supported
Ring Supervisor Mode	Possible Values: Deactivated, Activated
Ring Supervisor Precedence	Precedence value assigned to a ring supervisor, and transmitted in Beacon frames (further see www.odva.org)
Beacon-Interval (Mikro-Sek)	Ring interval at which the supervisor sends beacon frames (further see www.odva.org)
Beacon-Timeout (Mikro-Sek)	Amount of time nodes will wait before timing out reception of Beacon frames and taking the appropriate action (depending on weather supervisor or normal ring node). (further see www.odva.org)
VLAN-ID	VLAN ID used when sending DLR protocol frames.

Name	Description
Ring faults count	Counts Ring faults
MAC of last active node port 1	MAC Address of last active node at port 1
IP of last active node port 1	IP Address of last active node at port 1
MAC of last active node port 2	MAC Address of last active node at port 2
IP of last active node port 2	IP Address of last active node at port 2
Ring participants count	Ring participants count
MAC of active ring supervisor	MAC Address of active ring supervisor
IP of active ring supervisor	IP Address of active ring supervisor
Precedence of active ring supervisor	The precedence the supervisor has in the ring.
Capability Flags	Indicates which capabilities the supervisor has.
MAC Node 1	MAC Address Node 1
IP Node 1	IP Address Node 1
to	
MAC Node 10	MAC Address Node 2
IP Node 10	IP Address Node 2

Table 77: Extended Diagnosis EtherNet/IP Adapter > EIP_DLR > DLR Object Diagnosis

8.2.2.9 EIS_DLR/DLR Object Counter

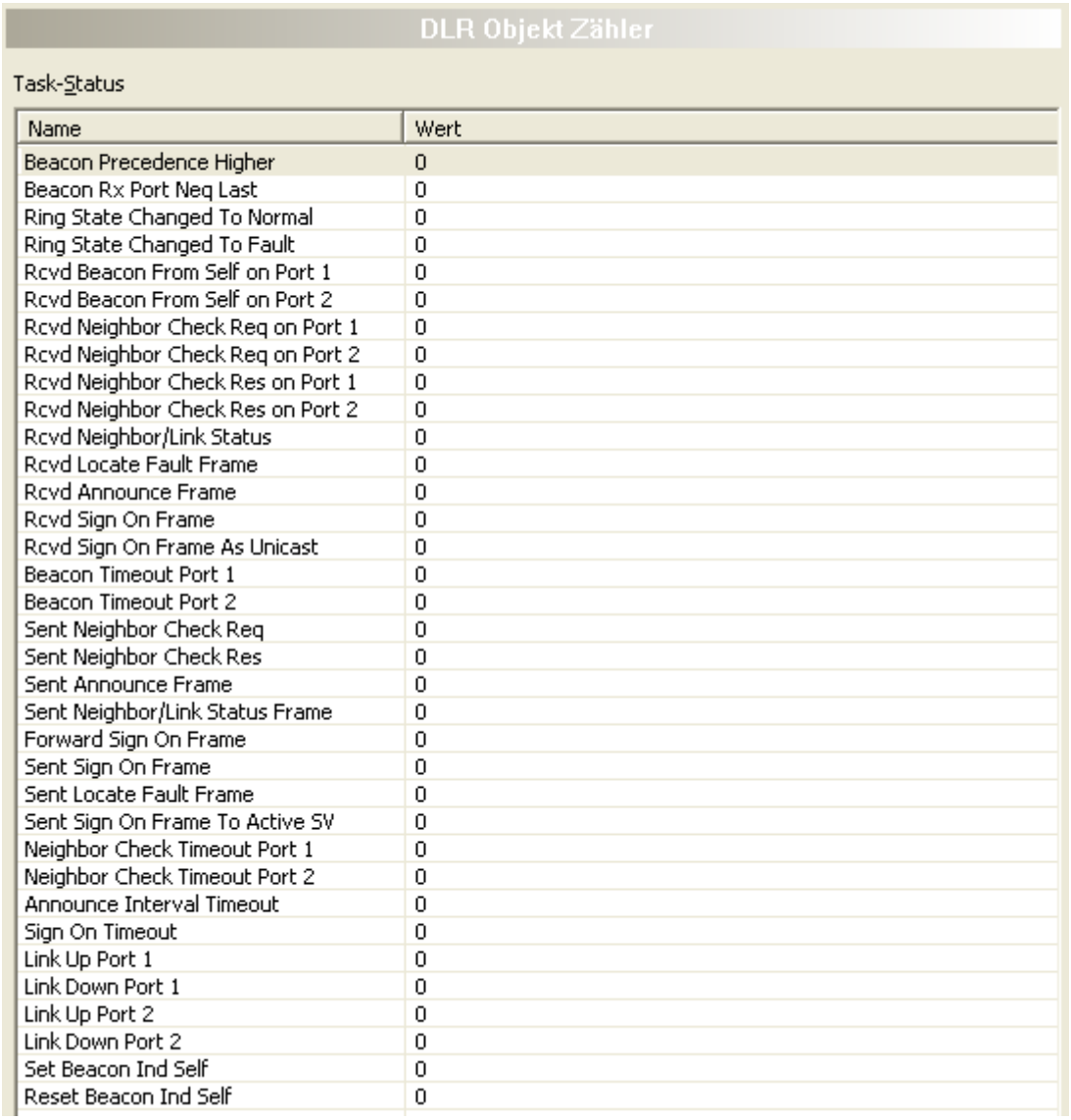


Figure 60: Extended Diagnosis EtherNet/IP Adapter > EIS_DLR > DLR Object Counter

Name	Description
[*Service*]	Diagnosis counter: Indicates the services processed. (The services of the single packets are described in the EtherNet/IP Slave Protocol API manual [2] [2].)

Table 78: Extended Diagnosis EtherNet/IP Adapter > EIS_DLR > DLR Object Counter

8.2.3 Open Modbus/TCP

8.2.3.1 Overview Open Modbus/TCP

Dialog Windows "Extended Diagnosis" Open Modbus/TCP

The table below gives an overview for the **Extended Diagnosis** dialog windows descriptions:

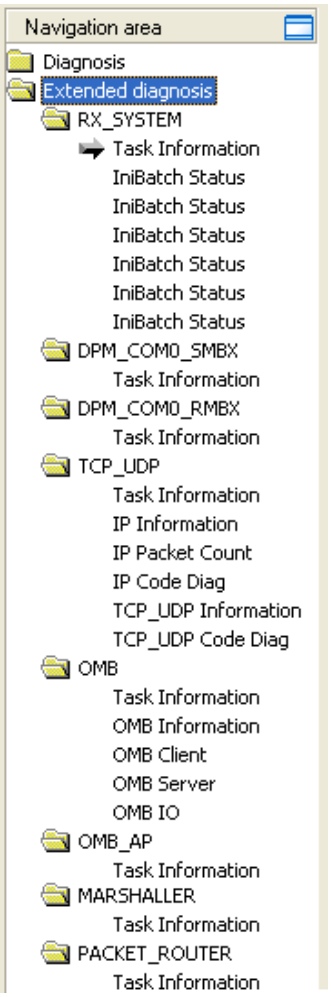
Navigation Area Open Modbus/TCP*	Folder Name in the Navigation Area	Dialog Window	Manual Page
	*OMB*	*OMB/OMB Information*	141
		OMB/OMB Client	142
		OMB/OMB Server	143
		OMB/OMB IO	144
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		**Note:** For netGateway devices the Extended Diagnosis for **DPM_COMO_SMBX**, **DPM_COMO_SMBX**, **Marshaller** and **Packet-Router** is not displayed.	
		** The displayed Open Modbus/TCP exteded diagnosis corresponds to the Open Modbus/TCP firmware version 2.5.x.x.*	

Table 79: Descriptions of the Dialog Windows Extended Diagnosis Open Modbus/TCP

8.2.3.2 OMB/OMB Information

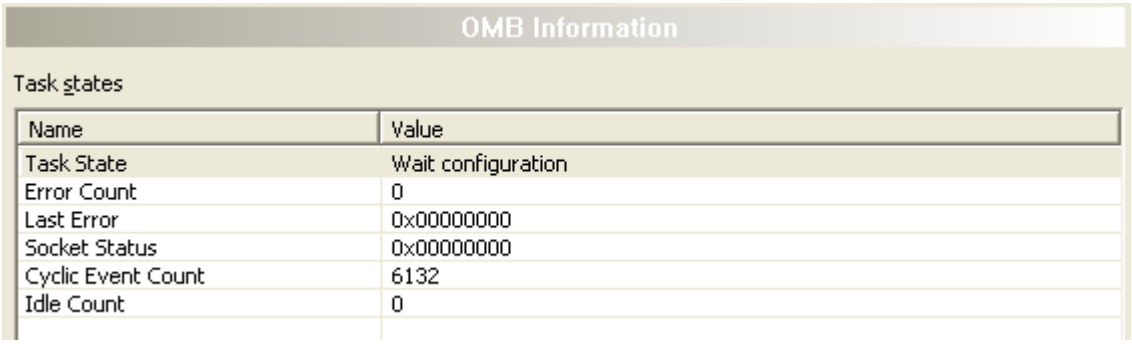


Figure 61: Extended Diagnosis Open Modbus/TCP > OMB > OMB Information

Name	Description
Task State	Actual state of the protocol process: - Task not initialized - Task is running - Task initialized - Initialization error - Waiting for configuration
Error Count	Counter for errors
Last Error	Last occurred error (Description see appropriate manual)
Socket Status	The socket status informs about the TCP sockets. More exactly, it contains the information whether sockets are open or closed in a bit-coded manner. The coding is as follows: The socket number corresponds to the position of the bit within the variable, i.e. bit 0 represents socket # 0 and bit 15 represents socket # 15. The coding is in that way that a bit value of 1 means the respective socket is open and a value of 0 indicates it is not open.
Cyclic Event Count	The cyclic event counter represents the number of cyclic events that have occurred.
Idle Count	The idle count is currently not used.

Table 80: Extended Diagnosis Open Modbus/TCP > OMB > OMB Information

8.2.3.3 OMB/OMB Client

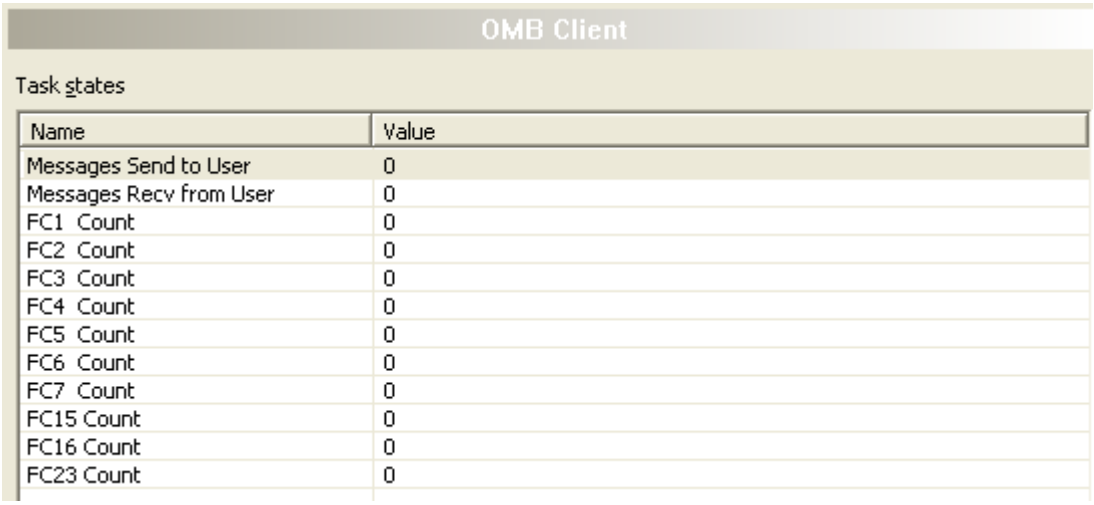


Figure 62: Extended Diagnosis Open Modbus/TCP > OMB > OMB Client

Name	Description
Messages Send to User	Number of messages send from the OMB client to the user
Messages Recv from User	Number of messages received from the user
FC[*N*] Count	Number of access events to the respective function code FC = Function code Possible values for *N* = 0 1, 2, 3, 4, 5, 6, 7, 15, 16, 23

Table 81: Extended Diagnosis Open Modbus/TCP > OMB > OMB Client

8.2.3.4 OMB/OMB Server

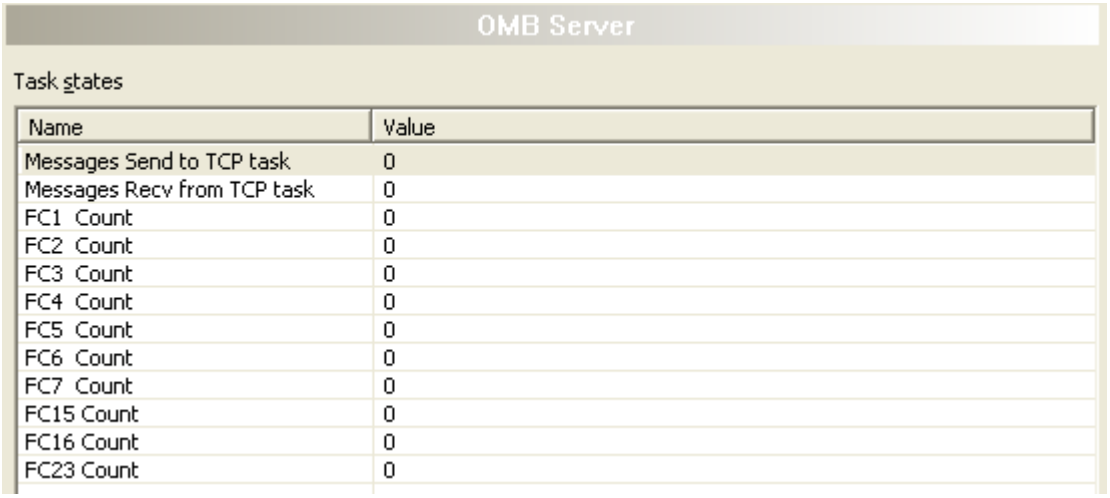


Figure 63: Extended Diagnosis Open Modbus/TCP > OMB > OMB Server

Name	Description
Messages Send to TCP task	Number of messages send from the OMB server to the TCP task
Messages Recv from TCP task	Number of messages received from the TCP task
FC[*N*] Count	Number of access events to the respective function code FC = Function code Possible values for *N* = 0 1, 2, 3, 4, 5, 6, 7, 15, 16, 23

Table 82: Extended Diagnosis Open Modbus/TCP > OMB > OMB Server

8.2.3.5 OMB/OMB IO

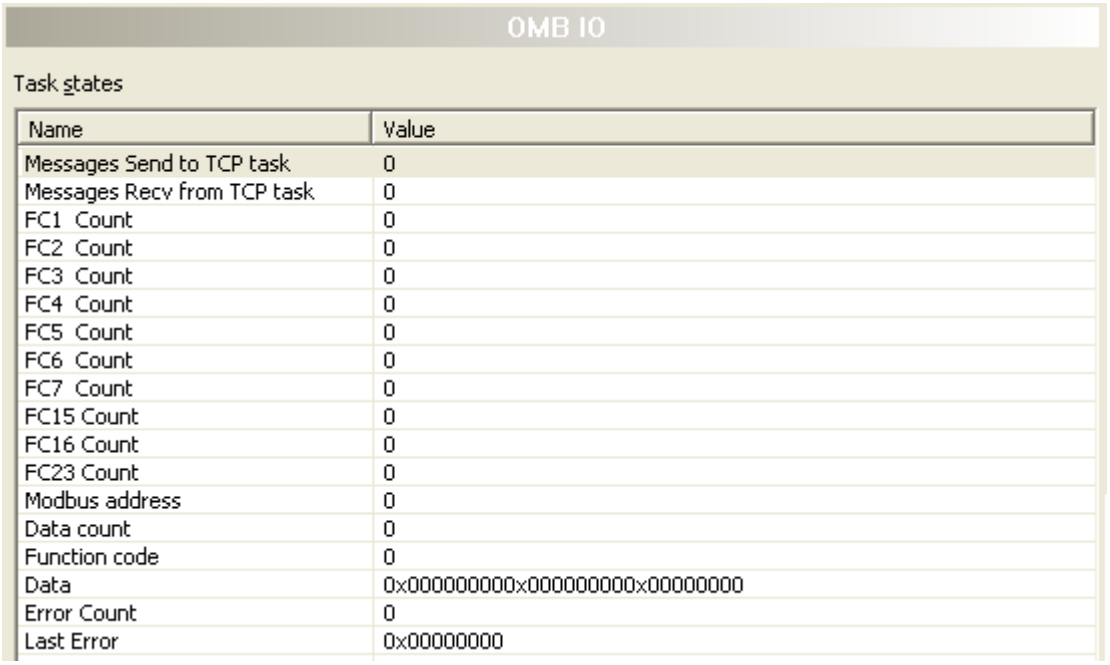


Figure 64: Extended Diagnosis Open Modbus/TCP > OMB > OMB IO

Name	Description
Messages Send to TCP task	Number of messages send from the OMB server to the TCP task
Messages Recv from TCP task	Number of messages received from the TCP task
FC[*N*] Count	Number of access events to the respective function code FC = Function code Possible values for *N* = 0 1, 2, 3, 4, 5, 6, 7, 15, 16, 23
Modbus address	Address within the Modbus Data model or memory map (Range 0 ... 65535).
Data count	Data counter
Function code	The Modbus standard function codes for send and receive operations: FC1: Read coils FC2: Read discrete inputs (input bits) FC3: Read multiple registers FC4: Read input registers FC5: Write coil FC6: Write single register FC7: Read Exception Status FC15 : Force multiple coils FC16: Write multiple registers FC23: Read/Write multiple registers
Data	The first data of the last sent function code (telegram) are displayed.
Error Count	Counter for errors
Last Error	Last occurred error (Description see appropriate manual)

Table 83: Extended Diagnosis Open Modbus/TCP > OMB > OMB IO

8.2.4 PROFINET IO Device

8.2.4.1 Overview PROFINET IO Device

Dialog Windows "Extended Diagnosis" PROFINET IO Device

The table below gives an overview for the **Extended Diagnosis** dialog windows descriptions:

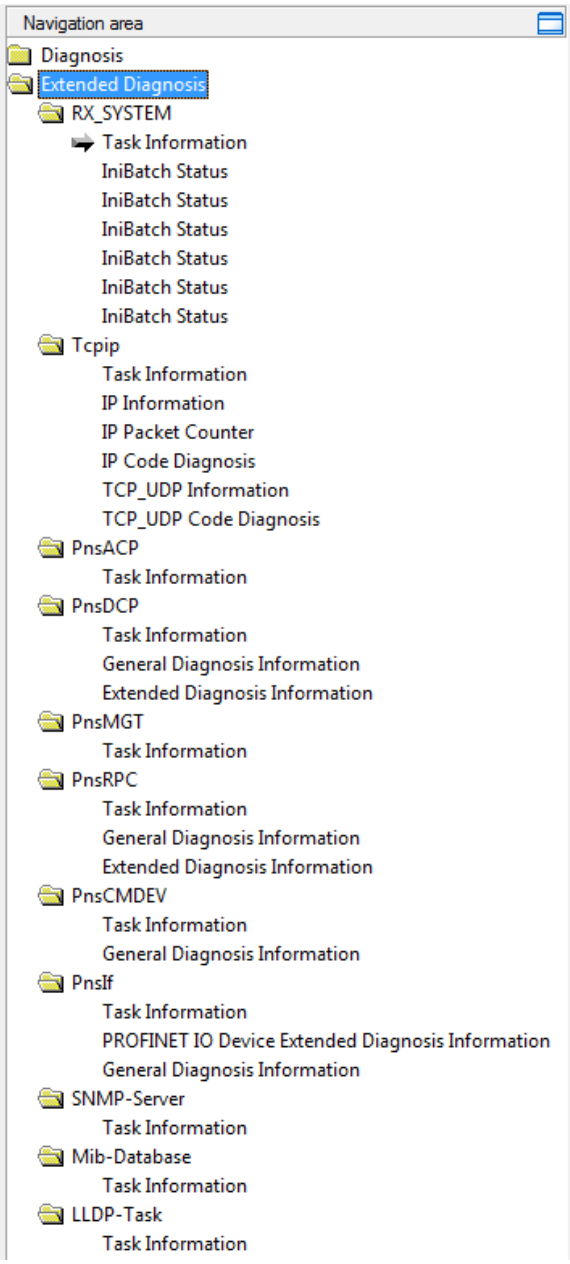
Navigation Area PROFINET IO Device*	-	Folder Name in the Navigation Area/Dialog Window	Manual Page
		*PnsDCP/Extended Diagnosis Information*	146
		PnsRCP/Extended Diagnosis Information	147
		PnsIf/PROFINET IO Device Extended Diagnosis Information	148
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		**The displayed PROFINET IO-Device exteded diagnosis corresponds to the PROFINET IO-Device firmware version 3.4.x.x*	

Table 84: Descriptions of the Dialog Windows Extended Diagnosis PROFINET IO Device

8.2.4.2 PnsDCP/Extended Diagnosis Information

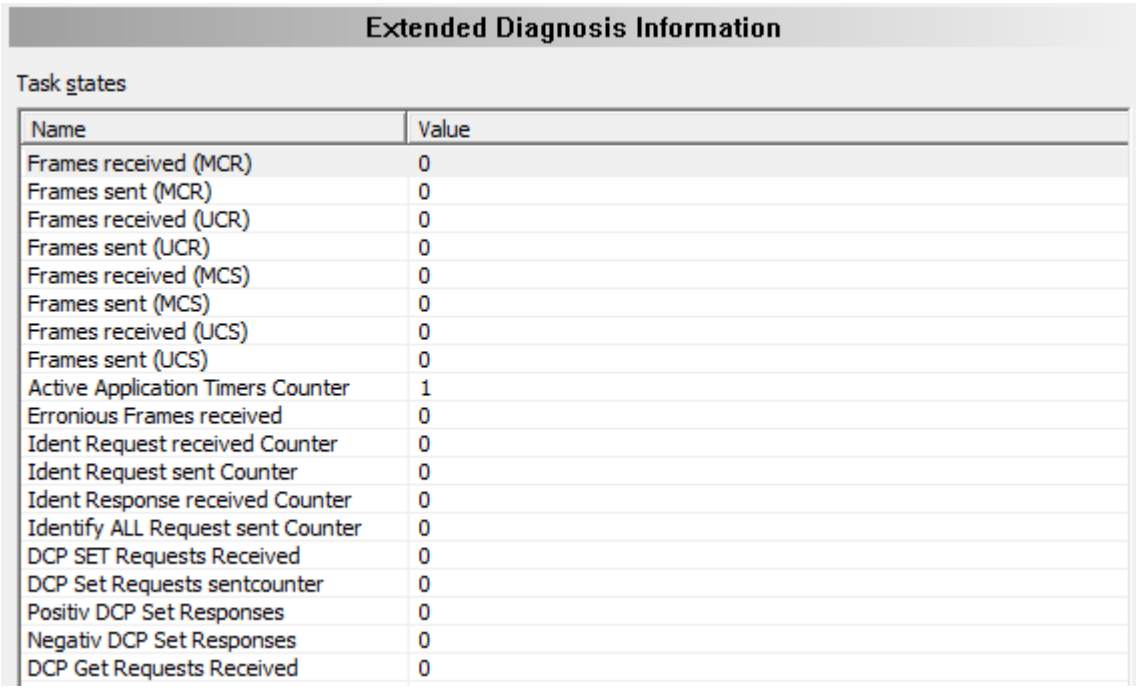


Figure 65: Extended Diagnosis PROFINET IO-Device > PnsDCP/Extended Diagnosis Information

The extended Diagnosis Information PNIO_DCP displays the counter reading of the four state machines from the PROFINET IO DCP protocol.

MCR: Multi Cast Receiver

UCR: Uni Cast Receiver

MCS: Multi Cast Sender

UCS: Uni Cast Sender

8.2.4.3 PnsRCP/Extended Diagnosis Information

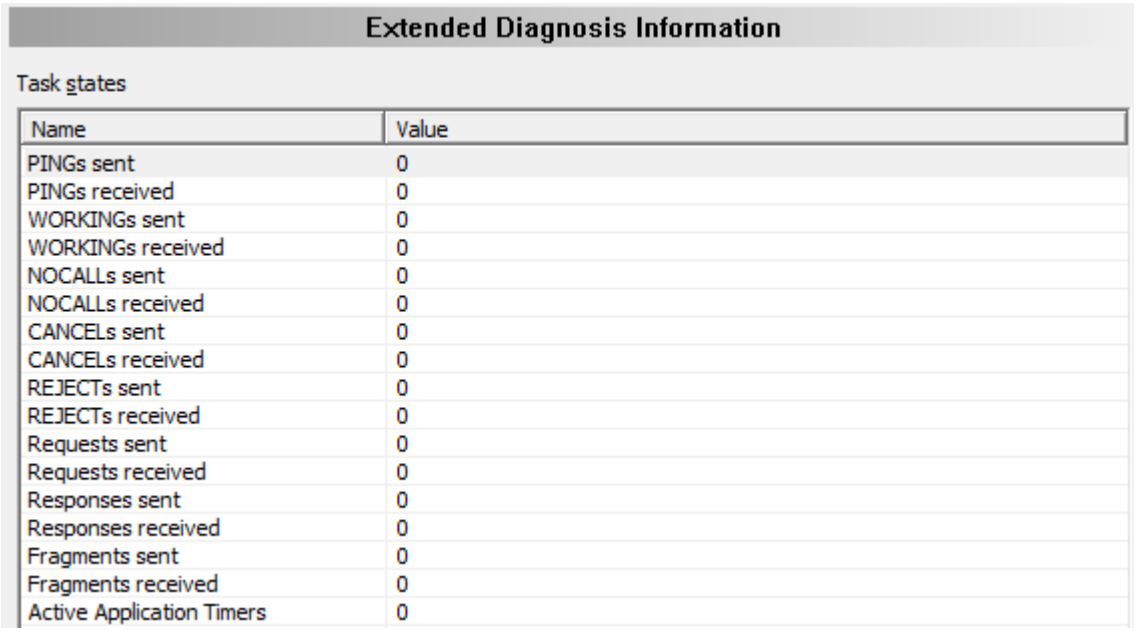


Figure 66: Extended Diagnosis PROFINET IO-Device > PnsRCP/Extended Diagnosis Information

The parameters under *RPC > Extended Diagnosis Information* represent PROFINET IO specific counters.

For further information refer to the PROFINET IO specification. [4], [17]

8.2.4.4 PnsIf/PROFINET IO Device Extended Diagnosis Information

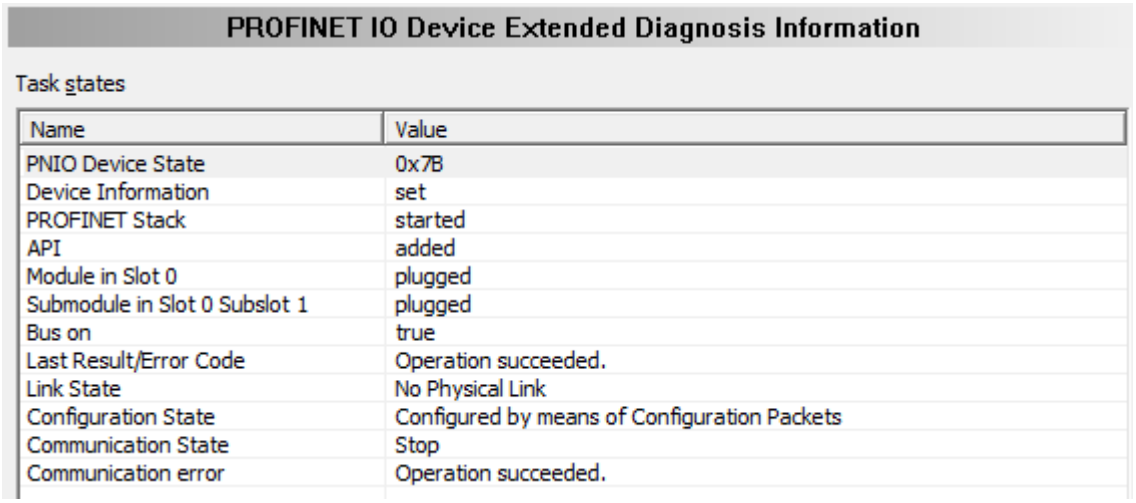


Figure 67: Extended Diagnosis PROFINET IO-Device > PnsIf/PROFINET IO Device Extended Diagnosis Information

Name	Description
PNIO Device State	Summary of the PROFINET IO stack status: set, not set
Device Information	Manufacturer information about the device, which is defined in the GSDML file.
PROFINET Stack	Status of the PROFINET IO stack: started, not started
API	API of the PROFINET IO stack: added, not added
Module in Slot 0	Module in Slot 0 of the PROFINET IO stack: plugged, not plugged
Submodule in Slot 0 Subslot 1	Submodule in Slot 0 Subslot 1 of the PROFINET IO stack: plugged, not plugged
Bus on	Network Communication: true, false
Last Result/Error Code	Last occurred result/ error of the PROFINET IO stack: e. g. “Operation succeeded”
Link State	State of the physical network connection of the PROFINET IO stack: Low Physical Link Speed, No Physical Link
Configuration State	Configuration State of the PROFINET IO stack • Not configured • Configured by means of Configuration Files • Error occurred while configuring with Configuration Files • Configured by means of Configuration Packets • Configuring by means of Configuration packets is running • Error occurred during configuring with Configuration Packets
Communication State	Communication State of the PROFINET IO stack • Unknown • Not configured • Stop • Idle • Operate • Communication error
Communication error	Communication error of the PROFINET IO stack: e. g. “Operation succeeded”

Table 85: Extended Diagnosis PROFINET IO-Device > PnsIf/PROFINET IO Device Extended Diagnosis Information

8.2.5 POWERLINK Controlled Node

8.2.5.1 Overview POWERLINK Controlled Node

“Extended Diagnosis” POWERLINK Controlled Node

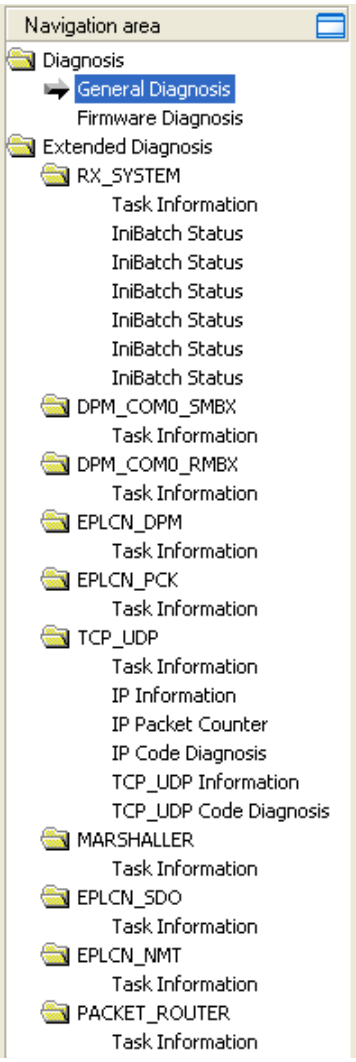
Navigation Area POWERLINK Controlled Node*	Folder Name in the Navigation Area	Dialog Window	Manual Page
		**Note:** Currently there is no specific extended diagnosis to the POWERLINK Controlled Node.	
		For the description of the tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		**Note:** For netGateway devices the Extended Diagnosis for **DPM_COMO_SMBX**, **DPM_COMO_SMBX**, **Marshaller** and **Packet-Router** is not displayed.	
		** The displayed POWERLINK Controlled Node exteded diagnosis corresponds to the POWERLINK Controlled Node firmware version 2.1.x.x.*	

Table 86: Extended Diagnosis POWERLINK Controlled Node

8.2.6 Sercos

8.2.6.1 Overview Sercos

“Extended Diagnosis” Sercos Slave

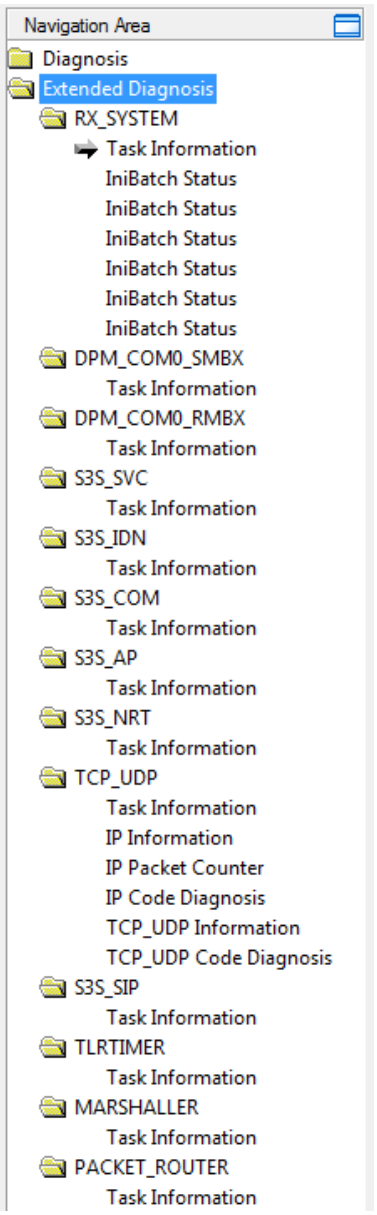
Navigation Area Sercos Slave	Folder Name in the Navigation Area	Dialog Window	Manual Page
		**Note:** Currently there is no specific extended diagnosis to the Sercos Slave.	
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		** The displayed POWERLINK Controlled Node exteded diagnosis corresponds to the Sercos Slave firmware version 3.1.x.x.*	

Table 87: Extended Diagnosis Sercos Slave

8.2.7 VARAN Client (Slave)

8.2.7.1 Overview VARAN Client (Slave)

Dialog Windows "Extended Diagnosis" VARAN Client (Slave)

Here you find an overview of the **Extended Diagnosis** for *VARAN Client (Slave)*.

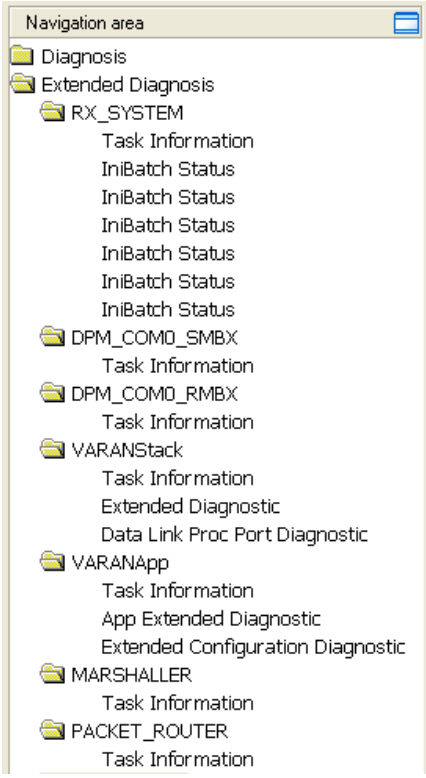
Navigation Area VARAN Client (Slave)*	Folder Name in the Navigation Area	Dialog Window	Manual Page
	VARANStack	*VARANStack/Extended Diagnostic*	152
		VARANStack/Data Link Proc Port Diagnostic	153
	VARANApp	*VARANApp/App Extended Diagnostic*	154
		VARANApp/Extended Configuration Diagnostic	155
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		** The displayed POWERLINK Controlled Node exteded diagnosis corresponds to the VARAN Client (Slave) firmware version 1.0.x.x.*	

Table 88: Descriptions of the Dialog Windows Extended Diagnosis VARAN Client (Slave)

8.2.7.2 VARANStack/Extended Diagnostic

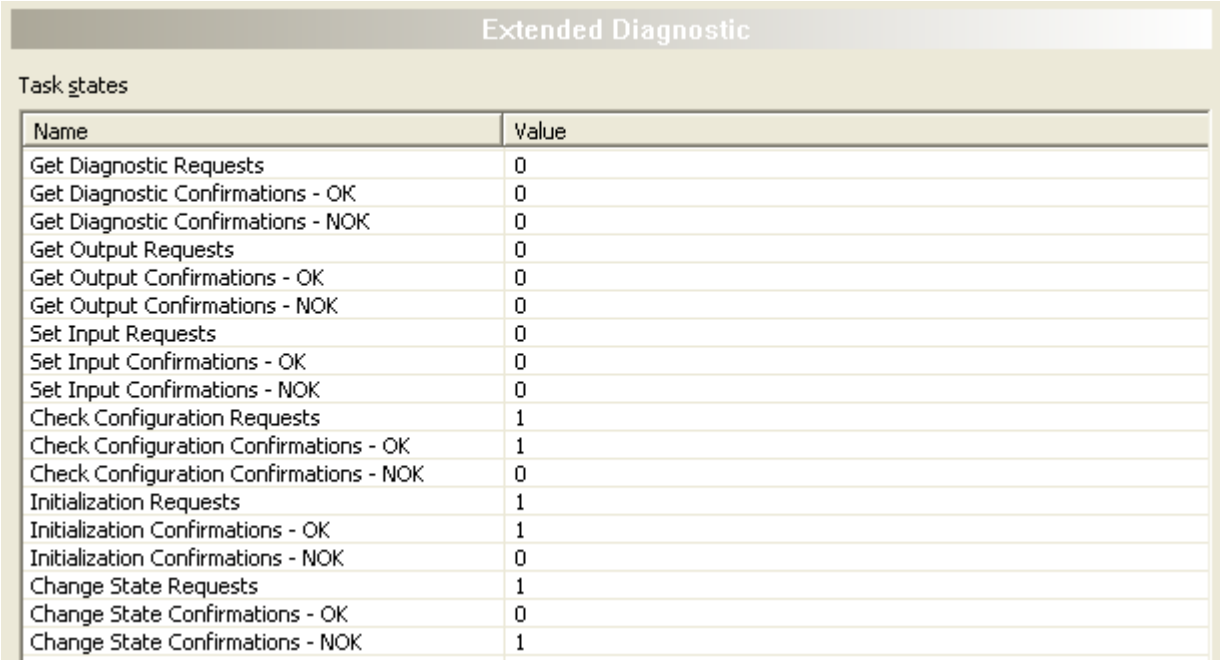
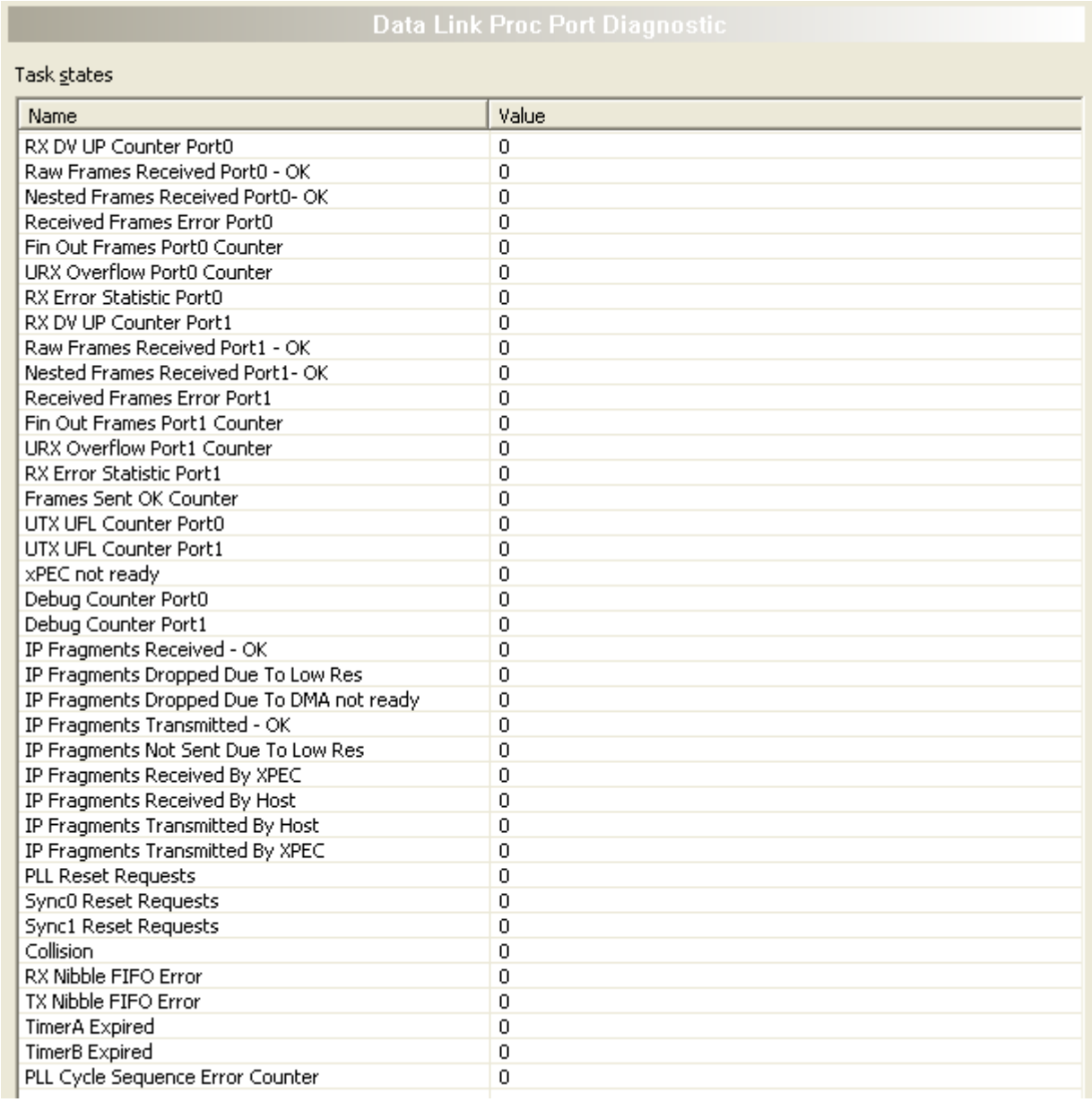


Figure 68: Extended Diagnosis VARAN Client (Slave) > VARANStack > Extended Diagnostic

Name	Description
[*Service*]	Diagnosis counter: Indicates the services processed. (The services of the single packets are described in the VARAN Client Protocol API manual [15], in section VARAN_CLIENT_CMD_GET_DIAG_REQ/CNF_T.)

Table 89: Extended Diagnosis VARAN Client (Slave) > VARANStack > Extended Diagnostic

8.2.7.3 VARANStack/Data Link Proc Port Diagnostic



Name	Value
RX DV UP Counter Port0	0
Raw Frames Received Port0 - OK	0
Nested Frames Received Port0- OK	0
Received Frames Error Port0	0
Fin Out Frames Port0 Counter	0
URX Overflow Port0 Counter	0
RX Error Statistic Port0	0
RX DV UP Counter Port1	0
Raw Frames Received Port1 - OK	0
Nested Frames Received Port1- OK	0
Received Frames Error Port1	0
Fin Out Frames Port1 Counter	0
URX Overflow Port1 Counter	0
RX Error Statistic Port1	0
Frames Sent OK Counter	0
UTX UFL Counter Port0	0
UTX UFL Counter Port1	0
xPEC not ready	0
Debug Counter Port0	0
Debug Counter Port1	0
IP Fragments Received - OK	0
IP Fragments Dropped Due To Low Res	0
IP Fragments Dropped Due To DMA not ready	0
IP Fragments Transmitted - OK	0
IP Fragments Not Sent Due To Low Res	0
IP Fragments Received By XPEC	0
IP Fragments Received By Host	0
IP Fragments Transmitted By Host	0
IP Fragments Transmitted By XPEC	0
PLL Reset Requests	0
Sync0 Reset Requests	0
Sync1 Reset Requests	0
Collision	0
RX Nibble FIFO Error	0
TX Nibble FIFO Error	0
TimerA Expired	0
TimerB Expired	0
PLL Cycle Sequence Error Counter	0

Figure 69: Extended Diagnosis VARAN Client (Slave) > VARANStack > Data Link Proc Port Diagnostic

Name	Description
[*Service*]	Diagnosis counter: Indicates the services processed. (The services of the single packets are described in the VARAN Client Protocol API manual [15], in section VARAN_CLIENT_CMD_GET_DIAG_REQ/CNF_T.)

Table 90: Extended Diagnosis VARAN Client (Slave) > VARANStack > Data Link Proc Port Diagnostic

8.2.7.4 VARANApp/App Extended Diagnostic

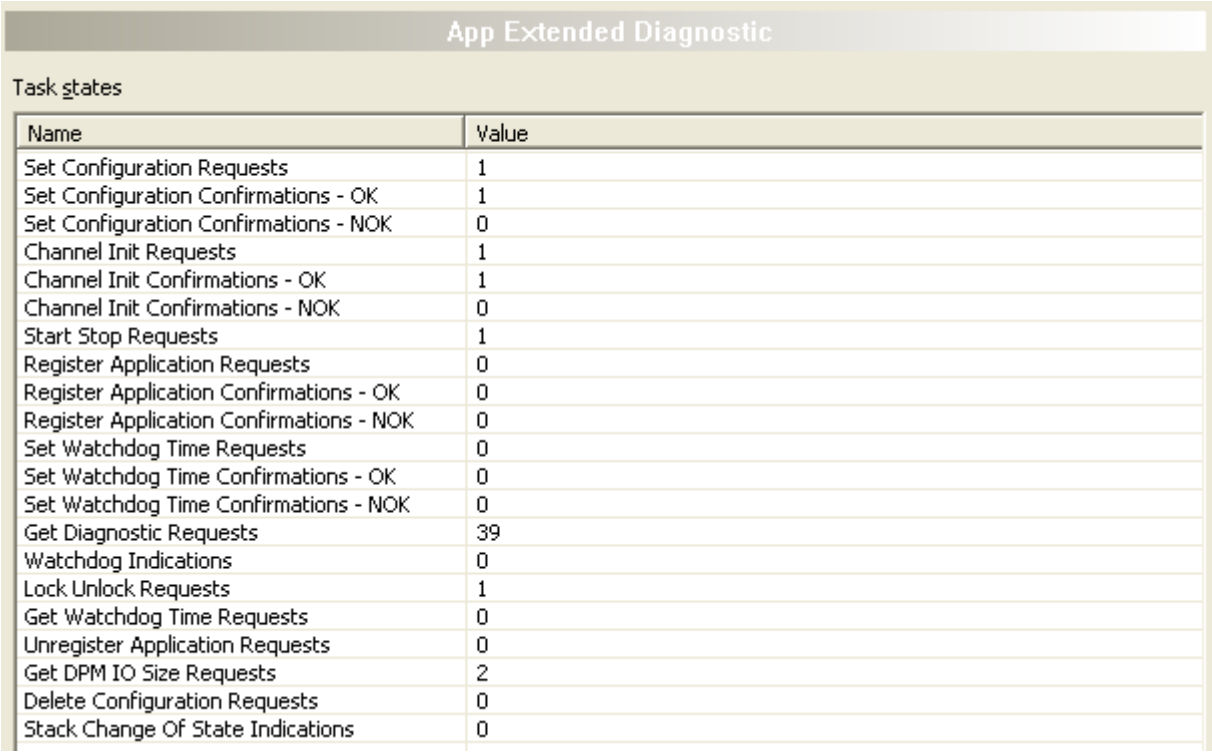


Figure 70: Extended Diagnosis VARAN Client (Slave) > VARANApp > App Extended Diagnostic

Name	Description
[*Service*]	Diagnosis counter: Indicates the services processed (refer also to the Dual-Port Memory Interface Manual [17]).

Table 91: Extended Diagnosis VARAN Client (Slave) > VARANApp > App Extended Diagnostic

8.2.7.5 VARANApp/Extended Configuration Diagnostic

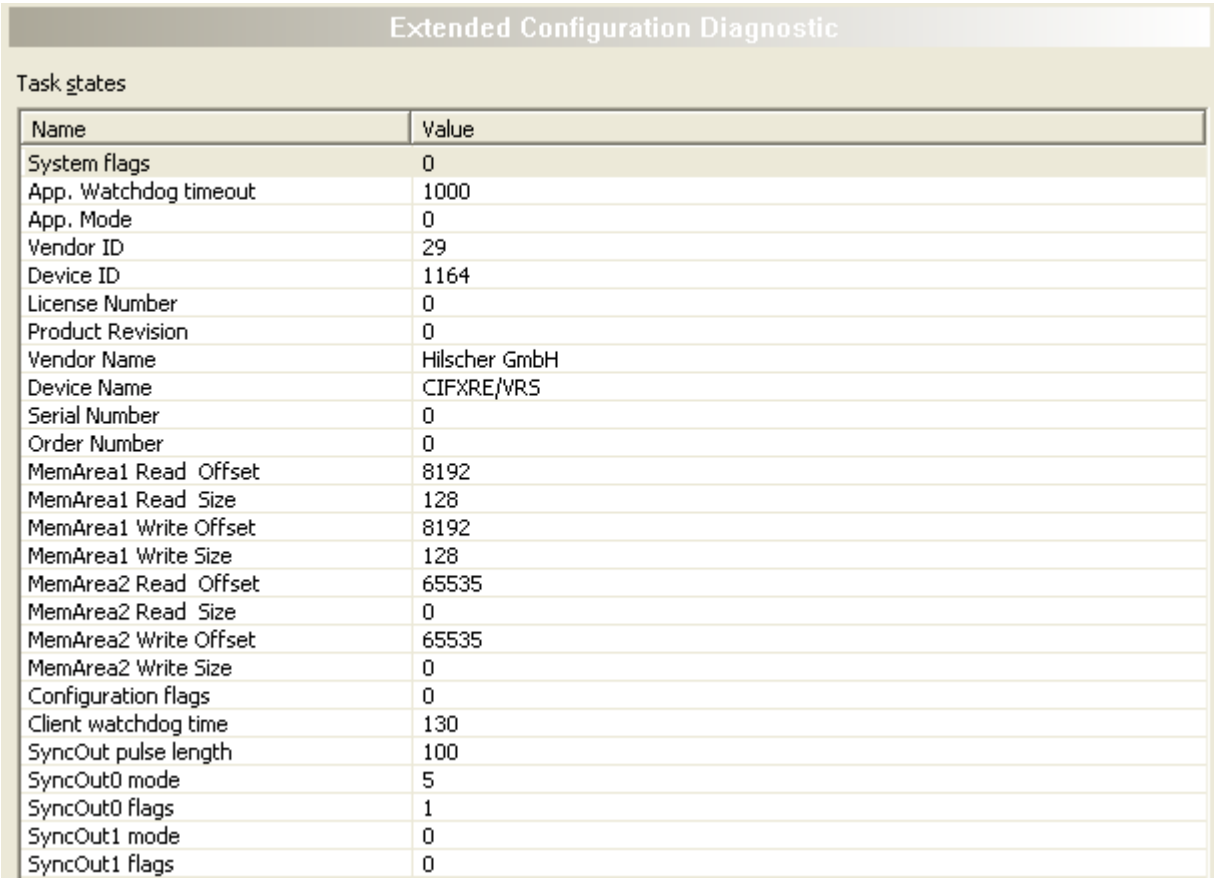


Figure 71: Extended Diagnosis VARAN Client (Slave) > VARANApp > Extended Configuration Diagnostic

Parameter	Meaning	Range of Value/Value
System flags [Bus Startup]	Communication start application controlled or automatic	1 (=Application controlled), 0 (=Automatic), Default: 0
App. Watchdog timeout [Client Watchdog Time]	Watchdog time within which the device watchdog must be retriggered from the application program while the application program monitoring is activated. When the watchdog time value is equal to0 the application program monitoring is deactivated.	[0, 20 … 65535] ms, default = 1000 ms, 0 = Off
App.Mode	Application mode 0 = VARAN_CLIENT_APP_MODE_IO	Default = 0
Vendor ID	Vendor Identifier: Identification number of the manufacturer. This value is specific for every single vendor.	0 … 2^{32} – 1, Hilscher: 29
Device ID	Device Identifier: Identification number of the device as indicated by the VARAN-BUS-NUTZERORGANISATION. This value is specific for every single device type and is fixed for every device. In case of an unknown device ID the manager rejects the client.	00 … 2^{32} – 1, CIFX RE/VRS: 1064; COMX100xx-RE/VRS: 1062; NIC50-RE/VRS: 1063; NJ50x-RE/VRS: 1060; NJ100xx-RE/VRS: 1061
License Number	License number of the device as specified by the VARAN-BUS-NUTZERORGANISATION.	0 … 2^{32} – 1, Default: 0
Product Revision [Product Version]	Version of the device as specified by the manufacturer.	0 … 2^{32} – 1, Default: 0

Parameter	Meaning	Range of Value/Value
Vendor Name	Name of the manufacturer, specific for every single vendor.	Zeichenkette, 0 ... 64 Zeichen, Default: Hilscher GmbH
Device Name	Name of the device as indicated by the VARAN-BUS-NUTZERORGANISATION.	Zeichenkette, 0 ... 64 Zeichen, Default: CIFX RE/VRS, COMX100xx-RE/VRS, NIC50-RE/VRS, NJ50x-RE/VRS, NJ100xx-RE/VRS
Serial Number	Serial number of the device as specified by the manufacturer.	0 … 2^{32} – 1, Default: 0
Order Number	Order number of the device as specified by the manufacturer.	0 … 2^{32} – 1, Default: 0
MemArea 1 Read Offset	Reading offset of the memory area 1in Byte.	0 … 65535 Byte, Default: 8192 Byte
Read Size / Memory Area 1	Reading size of the memory area 1in Byte.	0 … 128 Byte, Default: 128 Byte
Write Offset / Memory Area 1	Writing offset of the memory area 1in Byte.	0 … 65535 Byte, Default: 8192 Byte
Write Size / Memory Area 1	Writing size of the memory area 1 in Byte.	0 … 128 Byte, Default: 128 Byte
Read Offset / Memory Area 2*	Reading offset of the memory area 2 in Byte. (*currently not supported, set to default value)	0 … 65535 Byte, Default: 65535 Byte
Read Size / Memory Area 2*	Reading size of the memory area 2 in Byte. (*currently not supported, set to default value)	0 … 128 Byte, Default: 0 Byte
Write Offset / Memory Area 2*	Writing offset of the memory area 2 in Byte. (*currently not supported, set to default value)	0 … 65535 Byte, Default: 65535 Byte
Write Size / Memory Area 2*	Writing size of the memory area 2 in Byte. (*currently not supported, set to default value)	0 … 128 Byte, Default: 0 Byte
Confignuration Flags	Bit 0: Enable/Disable EMAC If set, the parameters for Memory Area 2 are enabled. Bit 1: Enable/Disable MemArea 2 If set, an integrated Extended MAC-Address (EMAC) for IP data exchange with client application is used. Bit 0 and 1 are currently not supported and they are set to default value.	Bit 0: 0 = EMAC disable, Bit 0: 1 = EMAC enable, Default value: 0 Bit 1: 0 = MemArea 2 disable, Bit 1: 1 = MemArea 2 enable, Default value: 0
Client Watchdog Time	Client watchdog time in ms. The Client watchdog time is the maximum cyclic communication timeout allowed, before Client signals a bus error.	[0 … 130] ms, Default = 130 ms, 0 = Off
SyncOut pulse length [Pulse Length (x10ns)]	Pulse length of the SYNC Out signal in 10ns steps. (e.g. the value 100 results as 10ns*100=1000ns=1µs pulse)	0 … 2.147.483.647, Default: 100

Sync OUT 0 / Sync OUT 1

Note!
Only the following 3 combinations between SyncOut0Mode + SyncOut1Mode are reasonable:

"Time for IN/OUT valid (TIO) – (**5**)"	+ "Disable – (**0**)",
"Time for IN valid (TI) – (**3**)"	+ "Time for OUT valid (T0) – (**4**)",
"Time for OUT valid (TO) – (**4**)"	+ "Time for IN valid (TI) – (**3**)"

Parameter	Meaning	Range of Value/Value
SyncOut0 mode	"SyncOut 0 Mode“ for the SYNC Out 0 signal, using the settings: "Disable – (**0**)": The output of the SYNC Out 0 signal is not available, "Time for IN valid (TI) – (**3**)": For the output of the SYNC Out 0 signal the time for the data IN is valid, "Time for OUT valid (TO) – (**4**)": For the output of the SYNC Out 0 signal the time for the data OUT is valid, "Time for IN/OUT valid (TIO) – (**5**)": For the output of the SYNC Out 0 signal the time for the data IN/OUT is valid.	0,3,4,5 Default: 5
SyncOut0 Flags	Bit 0: Enable/Disable Output The “Sync out 0 flag/Output” enables / disables the output of the Memory Area 1 SYNC Out 0 signal. Bit 1: Polarity active high/ low The “Sync out 0 flag/Polarity” defines the active high/low behavior of the SYNC Out 0 signal.	Bit 0: 0 = Output disable, Bit 0: 1 = Output enable, Default value: 1 Bit 1: 0 = active low, Bit 1: 1 = active high, Default value: 1
SyncOut1 mode	"SyncOut 1 Mode“ for the SYNC Out 1 signal, using the settings: "Disable – (**0**)": The output of the SYNC Out 1 signal is not available, "Time for IN valid (TI) – (**3**)": For the output of the SYNC Out 1 signal the time for the data IN is valid, "Time for OUT valid (TO) – (**4**)": For the output of the SYNC Out 1 signal the time for the data OUT is valid, "Time for IN/OUT valid (TIO) – (**5**)": For the output of the SYNC Out 1 signal the time for the data IN/OUT is valid.	0,3,4,5 Default: 0
SyncOut1 Flags	Bit 0: Enable/Disable Output The “Sync out 1 flag/Output” enables / disables the output of the Memory Area 2 SYNC Out 1 signal. Bit 1: Polarity active high/ low The “Sync out 1 flag/Polarity” defines the active high/low behavior of the SYNC Out 1 signal.	Bit 0: 0 = Output disable, Bit 0: 1 = Output enable, Default value: 0 Bit 1: 0 = active low, Bit 1: 1 = active high, Default value: 0

Table 92: Extended Diagnosis VARAN Client (Slave) > VARANApp > Extended Configuration Diagnostic

8.3 Fieldbus Systems

8.3.1 PROFIBUS DP Slave

8.3.1.1 Overview PROFIBUS DP Slave

Dialog Windows "Extended Diagnosis" PROFIBUS DP Slave

The table below gives an overview for the **Extended Diagnosis** dialog windows descriptions:

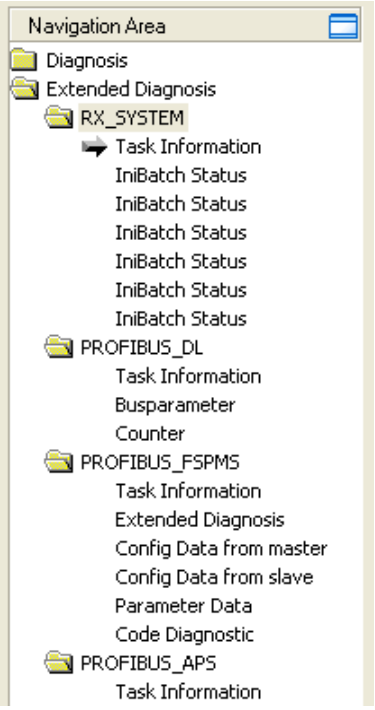
Navigation Area PROFIBUS DP Slave*	Folder Name in the Navigation Area	Dialog Window	Manual Page
	*PROFIBUS_DL*	*PROFIBUS_DL/Busparameter*	159
		PROFIBUS_DL/Counter	161
	PROFIBUS_ FSPMS	*PROFIBUS_FSPMS/Extended Diagnosis*	162
		PROFIBUS_FSPMS/ Config Data from Master	163
		PROFIBUS_FSPMS/ Config Data from Slave	163
		PROFIBUS_FSPMS/Parameter Data	164
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		** The displayed POWERLINK Controlled Node exteded diagnosis corresponds to the PROFIBUS DP Slave firmware version 2.4.x.x.*	

Table 93: Descriptions of the Dialog Windows Extended Diagnosis PROFIBUS DP Slave

8.3.1.2 PROFIBUS_DL/Busparameter

Under **Extended Diagnosis > PROFIBUS_DL > Busparameter** the values of the configured bus parameters are displayed which are active at the bus.

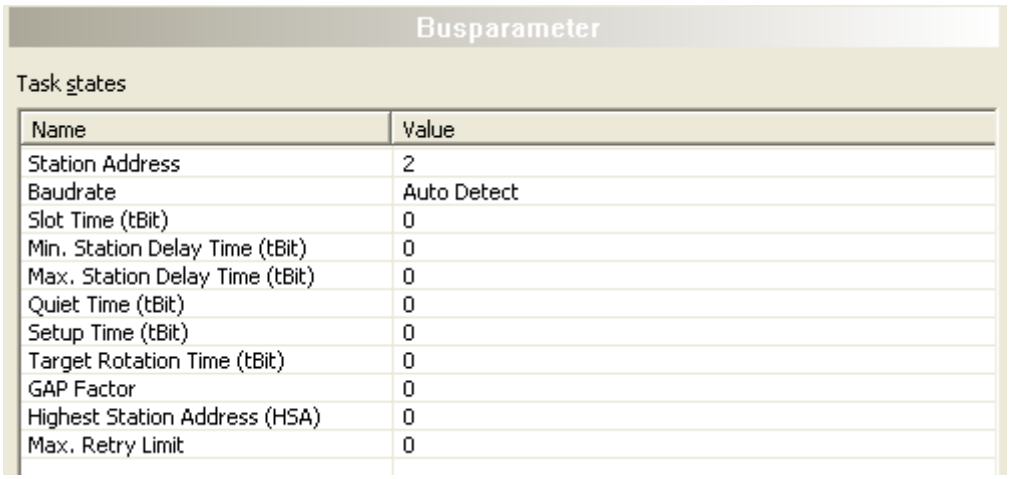


Figure 72: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_DL> Busparameter

Bus Parameters	**Meaning**		
Station Address	The Station Address is the individual device address of the Master device on the bus. Value range: 0 .. 125		
Baud Rate	The **Baud Rate** is the data transfer speed: number of Bits per second. The **Baud Rate** must be set to be the same for all devices on the bus. The result of changing the Baud rate is that all other parameters must be re-calculated.		
	Baud Rate	Bit time (t_{Bit})	Max cable length (type A)
	9,6 kBit/s 19,2 kBit/s 31,25 kBit/s 45,45 kBit/s 93,75 kBit/s 187,5 kBit/s 500 kBit/s **1500 kBit/s** 3000 kBit/s 6000 kBit/s 12000 kBit/s	104,2 us 52,1 us 32 us 22 us 10,7 us 5,3 us 2 us 666,7 ns 333,3 ns 166,7 ns 83,3 ns	1200 m 1200 m 1200 m 1200 m 1200 m 1000 m 400 m 200 m 100 m 100 m 100 m
Slot time (tBit)	'Wait for receipt' – Monitoring time of the sender (Requestor) of telegram for the acknowledgement of the recipient (Responder). After expiration, a retry occurs in accordance with the value of 'Max. telegram retries'. Value range: 37 .. 16383 (The default value depends from the baud rate.)		
Min. Station Delay Time (tBit)	This is the shortest time period that must elapse before a remote recipient (Responder) may send an acknowledgement of a received query telegram. The shortest time period between the reception of the last Bit of a telegram to the sending of the first Bit of a following telegram. Value range: 1 .. **11** . 65535		
Max. Station Delay Time (tBit)	This is the longest time period that must elapse before a Sender (Requestor) may send a further query telegram. Greatest time period between the reception of the last Bit of a telegram to the sending of the first Bit of a following telegram. The Sender (Requestor, Master) must wait at least for this time period after the sending of an unacknowledged telegram (e.g. Broadcast only) before a new telegram is sent. Value range: 1 .. 65535 (The default value depends from the baud rate.)		
Quiet Time (tBit)	This is the time delay that occurs for modulators (Modulator-trip time) and Repeaters (Repeater-switch time) for the change over from sending to receiving. Value range: 0 .. 127 (The default value depends from the baud rate.)		

Bus Parameters	Meaning
Setup Time (tBit)	Minimum period “reaction time” between the receipt of an acknowledgement to the sending of a new query telegram (Reaction) by the Sender (Requestor). Value range: 0 .. 255 (The default value depends from the baud rate.)
Target Rotation Time (tBit)	Pre-set nominal Token cycling time within the Sender authorization (Token) will cycle around the ring. How much time the Master still has available for sending data telegrams to the Slaves is dependent on the difference between the nominal and the actual token cycling time. The **Target rotation time (T_{TR})** is shown in Bit times (tBit) like the other Bus Parameters. Below the displayed Bit time, the **Target rotation time** is also displayed in milliseconds (ms). Value range: 1 .. 2^{24}-1 (=16.777.215) (The default value depends on the number of Slaves attached to the Master and their module configuration)
GAP Factor	Factor for determining after how many Token cycles an added participant is accepted into the Token ring. After expiry of the time period G*T_{TR}, the Station searches to see whether a further participant wishes to be accepted into the logical ring. Value range: 0 .. **10** .. 255
Highest Station Address (HSA)	The **Highest Station Address** is the highest bus address up to which a Master searches for another Master at the bus in order to pass on the Token. This station address must on no account be smaller than the Master station address. Value range: 1 .. **126**
Max. Retry Limit	Maximum number of repeats in order to reach a Station. Value range: 1 .. 15 (The default value depends from the baud rate.)

Table 94: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_DL> Busparameter

8.3.1.3 PROFIBUS_DL/Counter

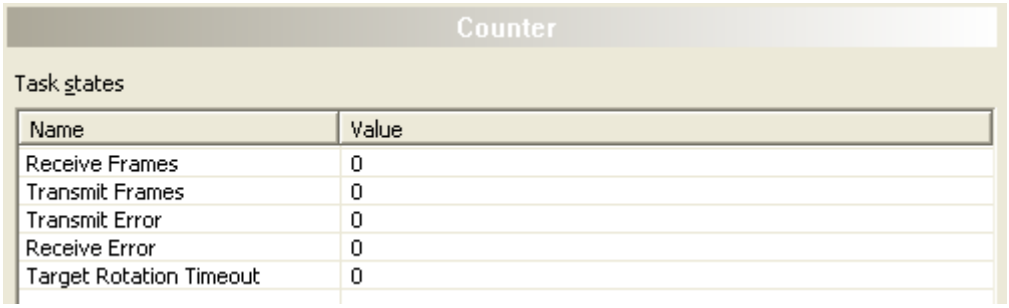


Figure 73: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_DL> Counter

The values of the counter **Receive Frames** and **Transmit Frames** show generally whether there is bus activity or not.

Name	Description
Receive Frames	Counter for number received frames
Transmit Frames	Counter for number transmitted frames
Transmit Error	Counter for number transmitted errors
Receive Error	Counter for number received errors
Target Rotation Timeout	Counter for number target rotation timeout

Table 95: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_DL> Counter

8.3.1.4 PROFIBUS_FSPMS/Extended Diagnosis

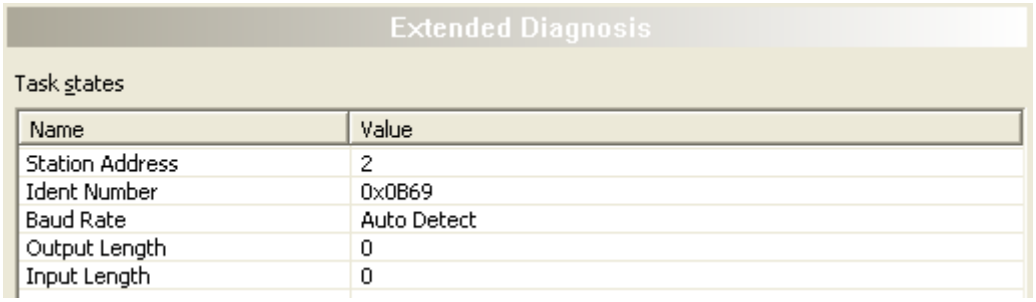


Figure 74: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_FSPMS > Extended Diagnosis

Name	Description	
Station Address	PROFIBUS address of the device Range of value: 0 … 125	
Ident Number	PROFIBUS Identification Number Range of value: 0 … 65535, Default: 0x0A12	
Baud Rate	9,6 kBit/s 19,2 kBit/s 93,75 kBit/s 187,5 kBit/s 500 kBit/s 1,5 MBit/s	3 MBit/s 6 MBit/s 12 MBit/s 31,25 kBit/s 45,45 kBit/s Auto detect
Output Length	Number of the output bytes Range of value: 0 … 244	
Input Length	Number of the input bytes Range of value: 0 … 244	

Table 96: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_FSPMS > Extended Diagnosis

8.3.1.5 PROFIBUS_FSPMS/ Config Data from Master

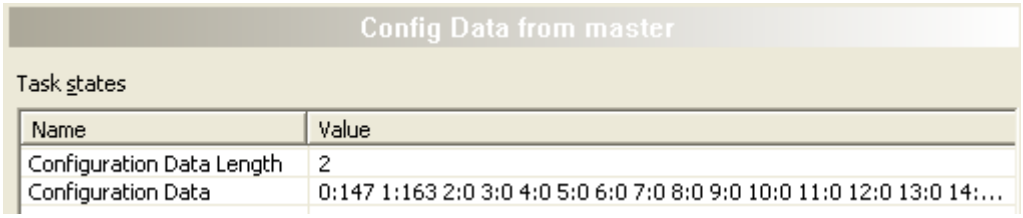


Figure 75: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_FSPMS > Config Data from Master

Name	Description
Configuration Data Length	Number of the configuration data from Master in byte Describes how many bytes of the configuration data are valid.
Configuration Data	Left value: numbering of every configuration data byte Right value: value of the respective configuration data byte

Table 97: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_FSPMS > Config Data from Master

Note: To view the **Value** column completely, double click to the right border of the column head. Move to the left or the right column side using the scroll bar.

8.3.1.6 PROFIBUS_FSPMS/ Config Data from Slave

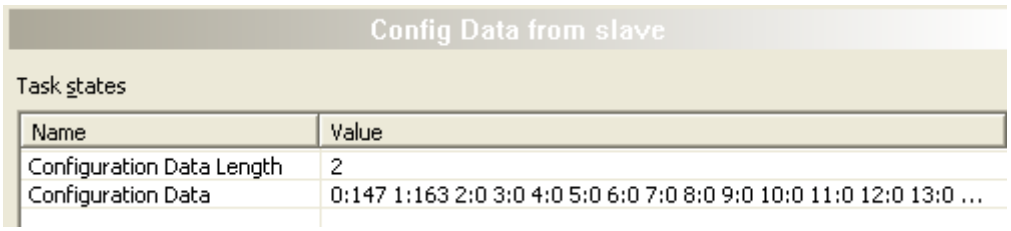


Figure 76: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_FSPMS > Parameter Data

Name	Description
Parameter Data Length	Number of the parameter data in byte Describes how many bytes of the parameter data are valid.
Parameter Data	Left value: numbering of every parameter data byte Right value: value of the respective parameter data byte

Table 98: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_FSPMS > Parameter Data

Note: To view the **Value** column completely, double click to the right border of the column head. Move to the left or the right column side using the scroll bar.

8.3.1.7 PROFIBUS_FSPMS/Parameter Data

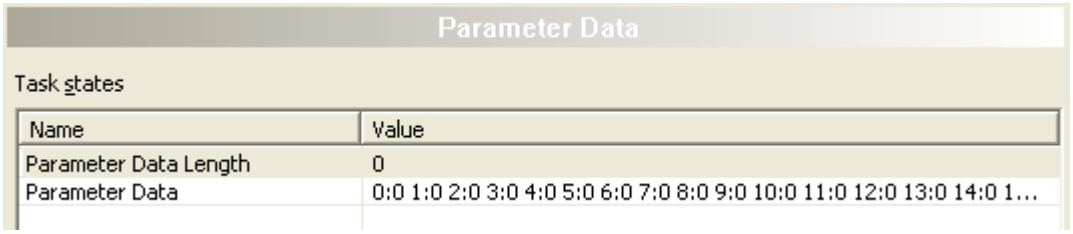


Figure 77: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_FSPMS > Parameter Data

Name	Description
Parameter Data Length	Number of the parameter data in byte Describes how many bytes of the parameter data are valid.
Parameter Data	Left value: numbering of every parameter data byte Right value: value of the respective parameter data byte

Table 99: Extended Diagnosis PROFIBUS DP Slave > PROFIBUS_FSPMS > Parameter Data

Note: To view the **Value** column completely, double click to the right border of the column head. Move to the left or the right column side using the scroll bar.

8.3.2 CANopen Slave

8.3.2.1 Overview CANopen Slave

Dialog Windows "Extended Diagnosis" CANopen Slave

The table below gives an overview for the **Extended Diagnosis** dialog windows descriptions:

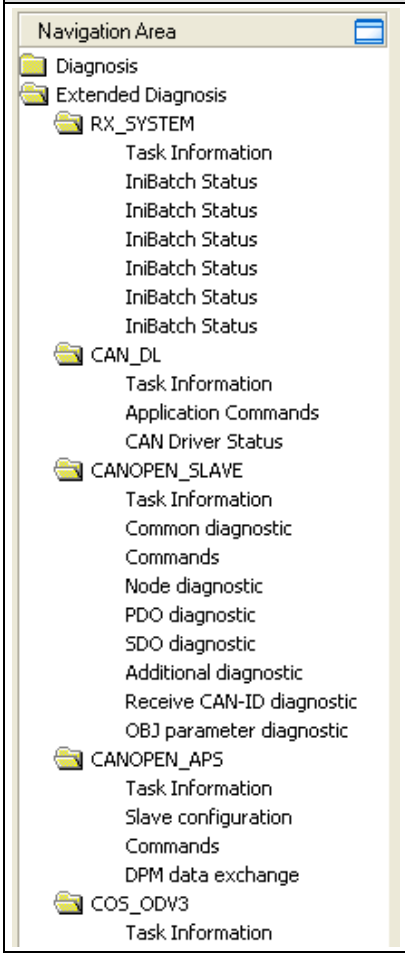
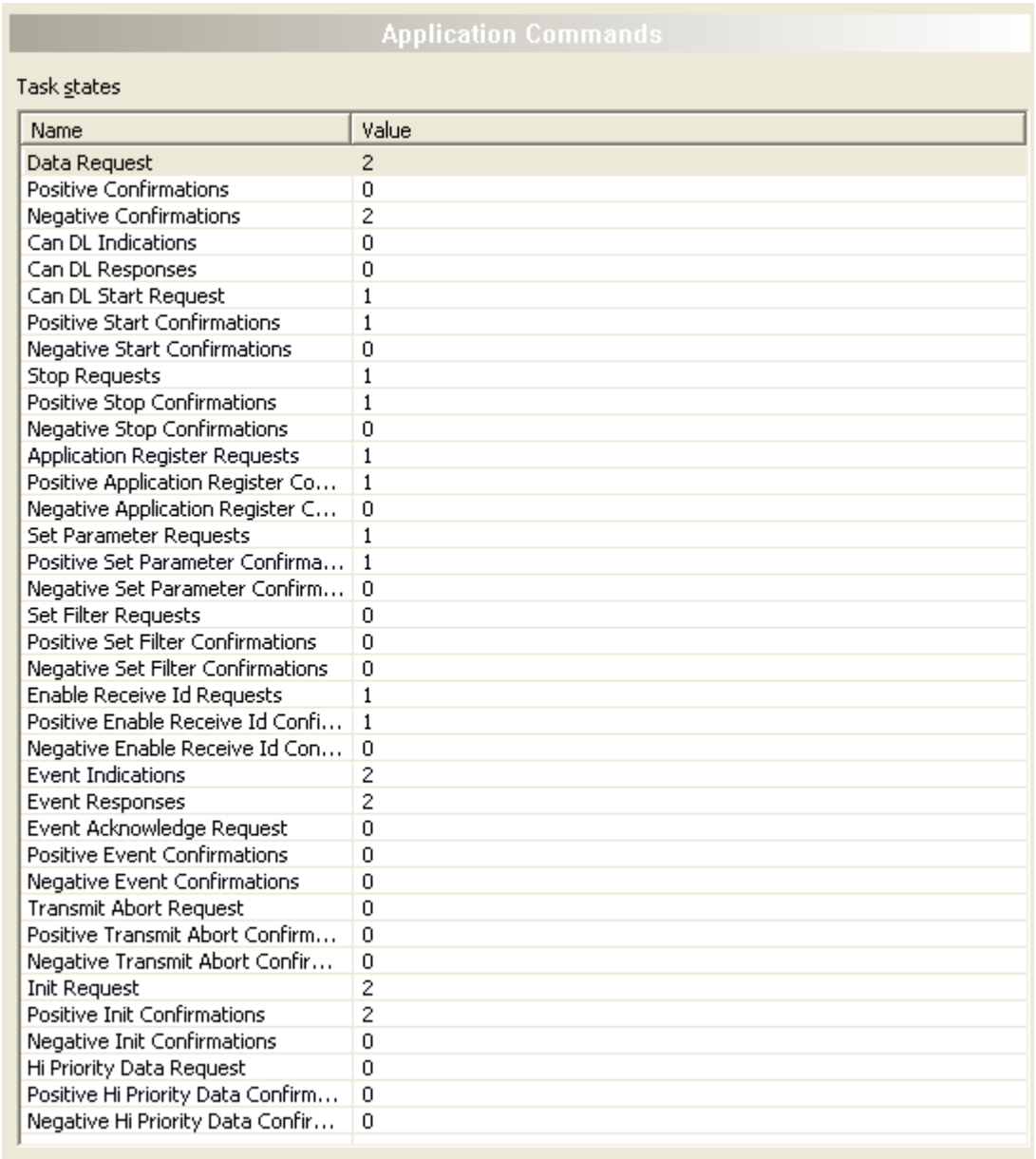
Navigation Area CANopen Slave*	Folder Name in the Navigation Area	Dialog Window	Manual Page
	*CAN_DL*	*CAN_DL/AP Commands Counter*	166
		CAN_DL/CAN Driver Status	167
	CANOPEN_SLAVE	*CANOPEN_SLAVE/Common Diagnostic*	168
		CANOPEN_SLAVE/Commands	169
	CANOPEN_APS	*CANOPEN_APS/Slave configuration*	176
		CANOPEN_APS/Commands	177
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		** The displayed CANopen Slave exteded diagnosis corresponds to the CANopen Slave firmware version 3.0.x.x.*	

Table 100: Descriptions of the Dialog Windows Extended Diagnosis CANopen Slave

8.3.2.2 CAN_DL/AP Commands Counter



Name	Value
Data Request	2
Positive Confirmations	0
Negative Confirmations	2
Can DL Indications	0
Can DL Responses	0
Can DL Start Request	1
Positive Start Confirmations	1
Negative Start Confirmations	0
Stop Requests	1
Positive Stop Confirmations	1
Negative Stop Confirmations	0
Application Register Requests	1
Positive Application Register Co…	1
Negative Application Register C…	0
Set Parameter Requests	1
Positive Set Parameter Confirma…	1
Negative Set Parameter Confirm…	0
Set Filter Requests	0
Positive Set Filter Confirmations	0
Negative Set Filter Confirmations	0
Enable Receive Id Requests	1
Positive Enable Receive Id Confi…	1
Negative Enable Receive Id Con…	0
Event Indications	2
Event Responses	2
Event Acknowledge Request	0
Positive Event Confirmations	0
Negative Event Confirmations	0
Transmit Abort Request	0
Positive Transmit Abort Confirm…	0
Negative Transmit Abort Confir…	0
Init Request	2
Positive Init Confirmations	2
Negative Init Confirmations	0
Hi Priority Data Request	0
Positive Hi Priority Data Confirm…	0
Negative Hi Priority Data Confir…	0

Figure 78: Extended Diagnosis CANopen Slave > CAN_DL > AP Commands Counter

Name	Description
[*Service*]	Diagnosis counter of the CAN layer. Indicates the services processed. (The services of the single packets are described in the CANopen Slave Protocol API manual [8].)

Table 101: Extended Diagnosis CANopen Slave > CAN_DL > AP Commands Counter

8.3.2.3 CAN_DL/CAN Driver Status

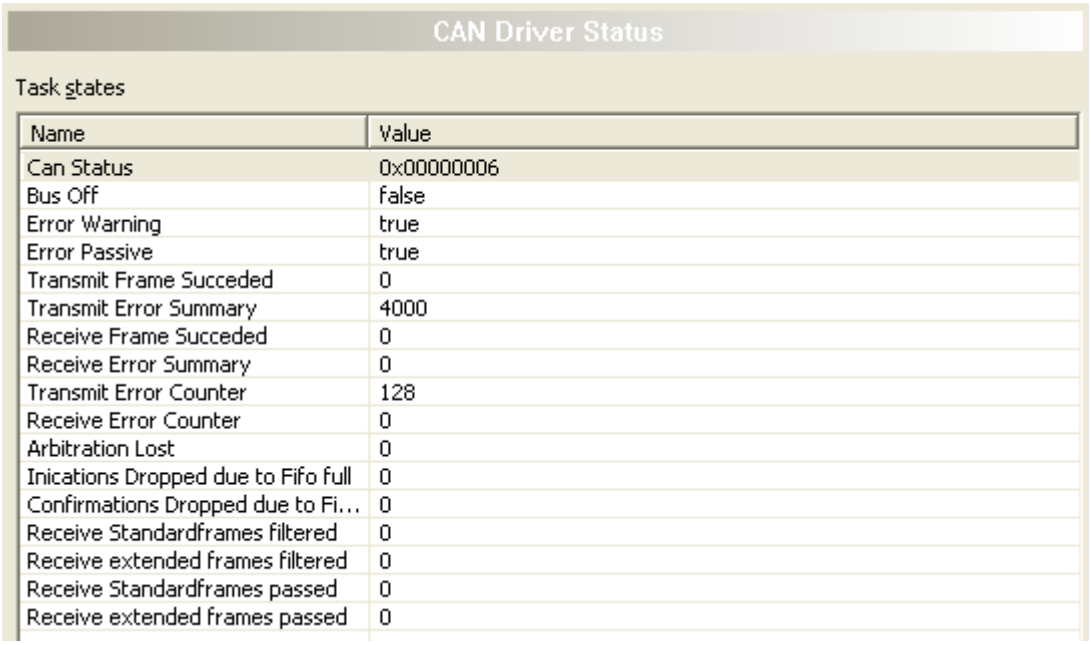


Figure 79: Extended Diagnosis CANopen Slave > CAN_DL > CAN Driver Status

Name	Description
CAN Status Bus Off Error Warning ErrorPassive	Diagnosis status of CAN specific error levels. Indicates the respective status of the CAN bus. (For further information refer to the CANopen Slave Protocol API manual [8].)
[*Service*]	Diagnosis counter of CAN errors. Indicates the services processed. (The services of the single packets are described in the CANopen Slave Protocol API manual [8].)

Table 102: Extended Diagnosis CANopen Slave > CAN_DL > CAN Driver Status

8.3.2.4 CANOPEN_SLAVE/Common Diagnostic

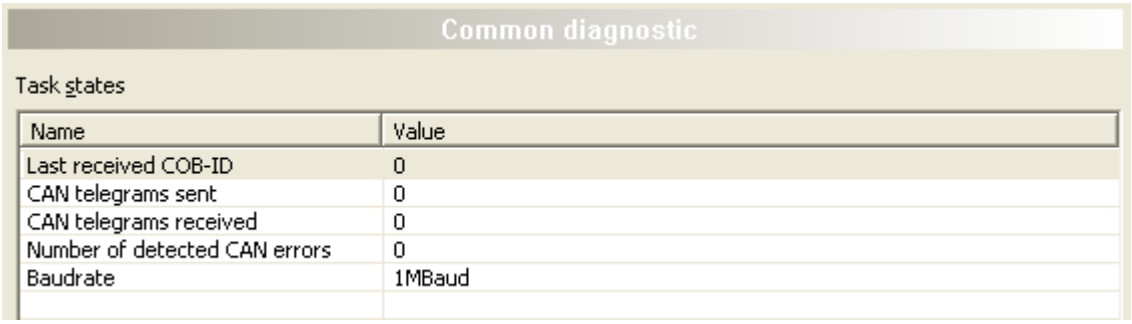
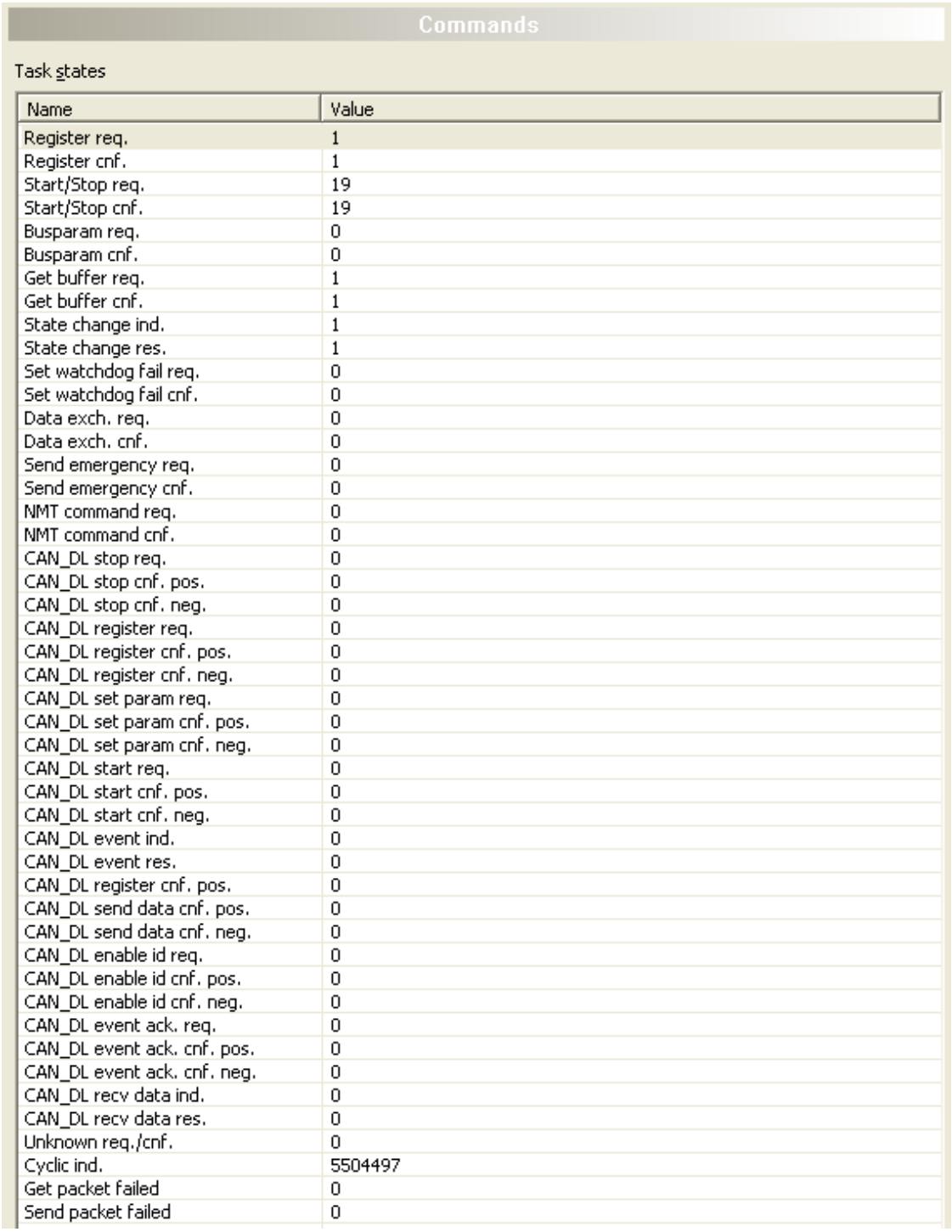


Figure 80: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > Common Diagnostic

Name	Description	
Last received COB-ID	Last received CAN-Message Header-ID	
CAN telegrams sent	Number of sent CAN-Messages	
CAN telegrams received	Number of received CAN-Messages	
Number of detected CAN errors	Number of detected CAN errors	
Bauderate	Baud rate of CANopen connection Available Baud Rate:	
	1 MBaud 800 KBaud 500 KBaud 250 KBaud 125 KBaud	100 KBaud 50 KBaud 20 KBaud 10 KBaud

Table 103: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > Common Diagnostic

8.3.2.5 CANOPEN_SLAVE/Commands



Name	Value
Register req.	1
Register cnf.	1
Start/Stop req.	19
Start/Stop cnf.	19
Busparam req.	0
Busparam cnf.	0
Get buffer req.	1
Get buffer cnf.	1
State change ind.	1
State change res.	1
Set watchdog fail req.	0
Set watchdog fail cnf.	0
Data exch. req.	0
Data exch. cnf.	0
Send emergency req.	0
Send emergency cnf.	0
NMT command req.	0
NMT command cnf.	0
CAN_DL stop req.	0
CAN_DL stop cnf. pos.	0
CAN_DL stop cnf. neg.	0
CAN_DL register req.	0
CAN_DL register cnf. pos.	0
CAN_DL register cnf. neg.	0
CAN_DL set param req.	0
CAN_DL set param cnf. pos.	0
CAN_DL set param cnf. neg.	0
CAN_DL start req.	0
CAN_DL start cnf. pos.	0
CAN_DL start cnf. neg.	0
CAN_DL event ind.	0
CAN_DL event res.	0
CAN_DL register cnf. pos.	0
CAN_DL send data cnf. pos.	0
CAN_DL send data cnf. neg.	0
CAN_DL enable id req.	0
CAN_DL enable id cnf. pos.	0
CAN_DL enable id cnf. neg.	0
CAN_DL event ack. req.	0
CAN_DL event ack. cnf. pos.	0
CAN_DL event ack. cnf. neg.	0
CAN_DL recv data ind.	0
CAN_DL recv data res.	0
Unknown req./cnf.	0
Cyclic ind.	5504497
Get packet failed	0
Send packet failed	0

Figure 81: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > Commands

Name	Description
[*Service*]	Diagnosis counter: Indicates the services processed. (The services of the single packets are described in the CANopen Slave Protocol API manual [8].)
Get packet failed	Number of errors during requesting a packet
Send packet failed	Number of errors during sending a packet

Table 104: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > Commands

8.3.2.6 CANOPEN_SLAVE/Node diagnostic

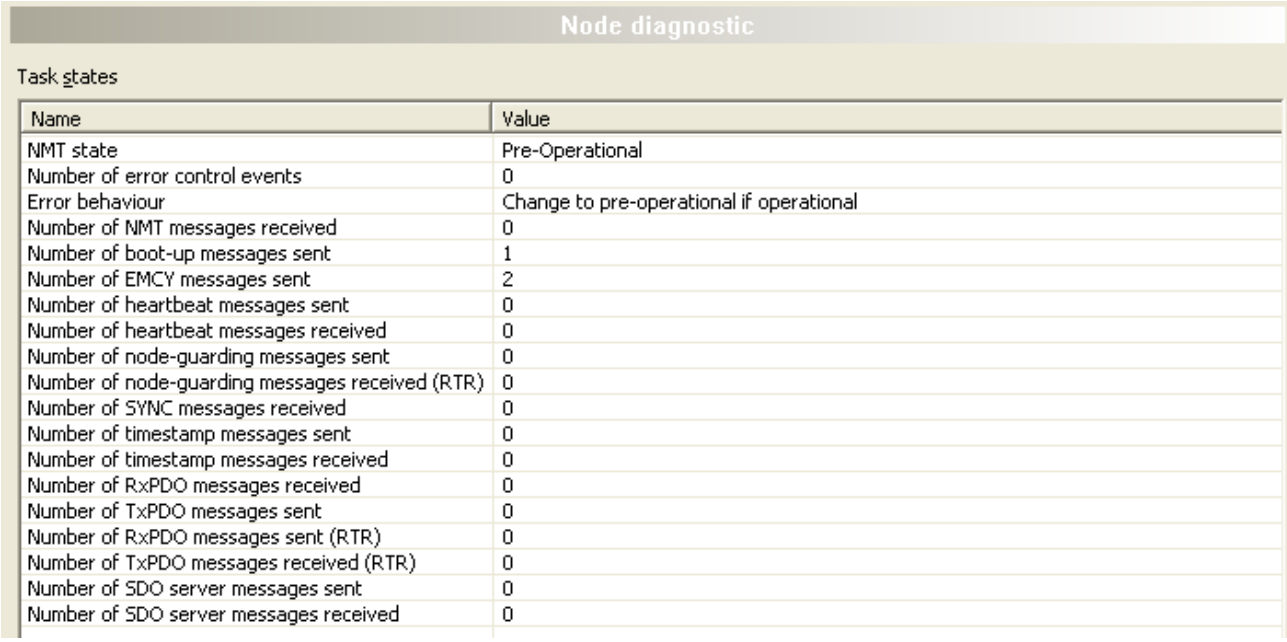


Figure 82: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > Node diagnostic

Name	Description	
NMT state	Possible NMT states: - Reset (at the beginning - reset node, later on - reset communication) - Stopped - Operational - Pre-operational NMT = Network Management (For further information refer to the CANopen Slave Protocol API manual [8].)	
Error behaviour	Possible Error behaviour - Change to pre-operational if operational - No change of NMT state - Change to stopped	
[*Services*]	Number of error control events	Diagnosis counter
	Number of NMT messages received Number of boot-up messages sent Number of EMCY messages sent Number of heartbeat messages sent Number of heartbeat messages received Number of node-guarding messages sent Number of node-guarding messages received (RTR) Number of SYNC messages received Number of timestamp messages sent Number of timestamp messages received Number of RxPDO messages received Number of TxPDO messages sent Number of RxPDO messages sent (RTR) Number of TxPDO messages received (RTR) Number of SDO server messages sent Number of SDO server messages received	Internal diagnosis counter: Indicate the send/received messages.

Table 105: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > Node diagnostic

8.3.2.7 CANOPEN_SLAVE/PDO diagnostic

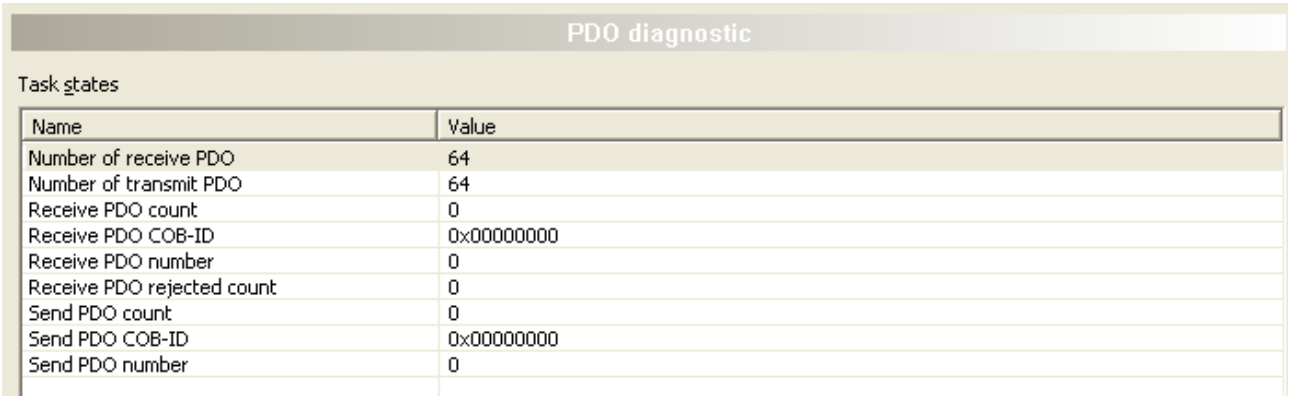


Figure 83: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > PDO diagnostic

Name	Description
[Services]	Diagnosis counter: Indicate the services processed. (The services of the single packets are described in the CANopen Slave Protocol API manual [8].)
Number of receive PDOs	Number of the used receive PDOs
Number of transmit PDOs	Number of the used send PDOs
Receive PDO number	Number of the last received PDO
Receive PDO COB-ID	CAN identifier related to the COB (Communication Object) of the last received PDO. Receive PDOs are output data of the Master and are received from the node. (For further information refer to CANopen Slave Protocol API manual [8].)
Send PDO COB-ID	CAN identifier related to the COB (Communication Object) of the last sent PDO. Send PDOs are input data of the Master and are send by the node. (For further information refer to CANopen Slave Protocol API manual [8].)

Table 106: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > PDO diagnostic

8.3.2.8 CANOPEN_SLAVE/SDO diagnostic

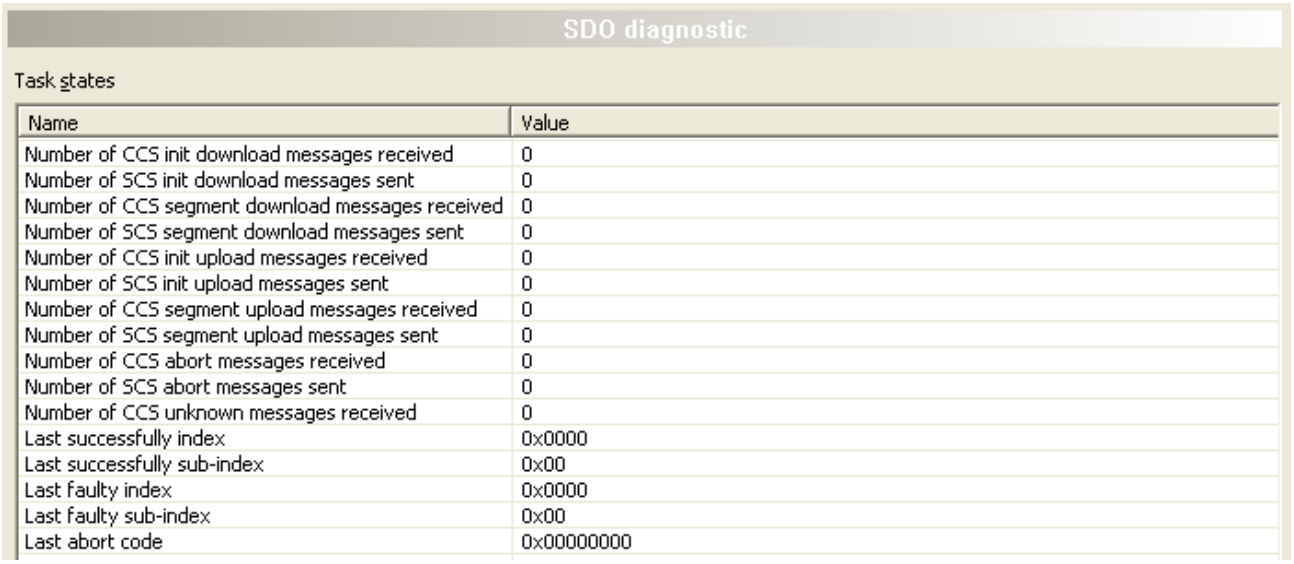


Figure 84: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > SDO diagnostic

Name	Description
[*Service*]	Diagnosis counter: Indicate the services processed. (The services of the single packets are described in the CANopen Slave Protocol API manual [8].)
Last successfully index Last successfully sub-index Last faulty index Last faulty sub-index Last abort code	Last index/sub-index of the successful or faulty send/received service data object (SDO) and code for abort.

Table 107: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > SDO diagnostic

8.3.2.9 CANOPEN_SLAVE/Additional diagnostic

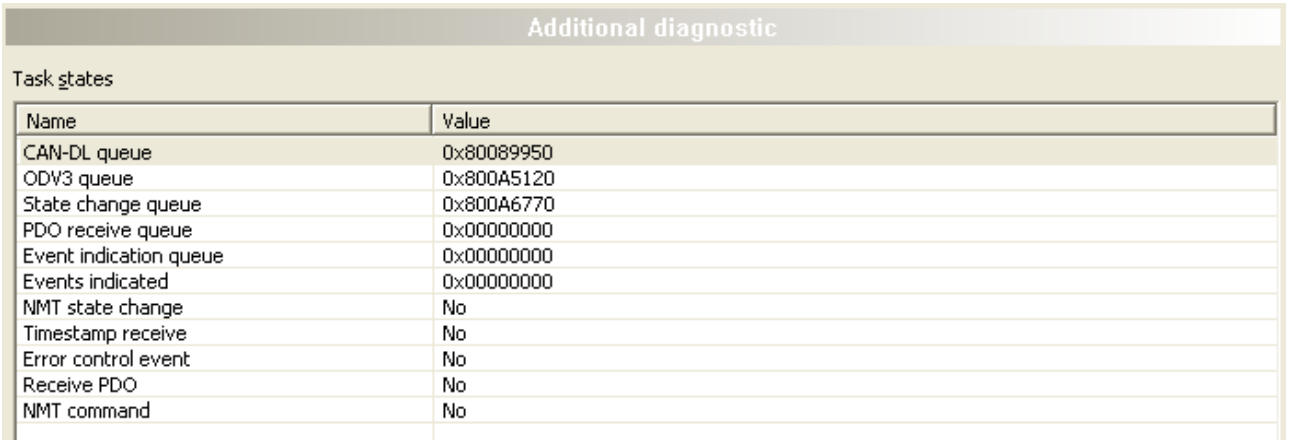
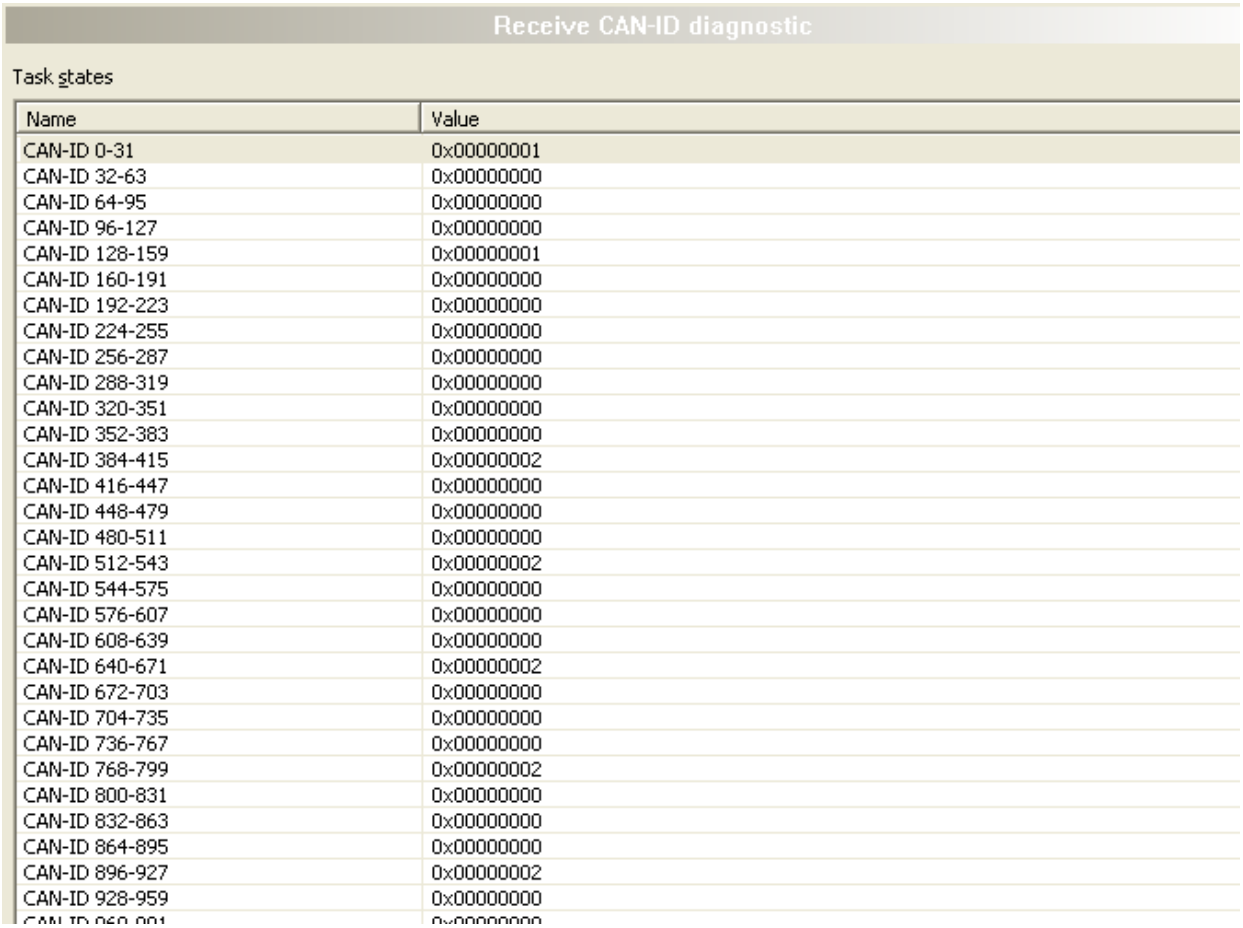


Figure 85: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > Additional diagnostic

Name	Description
CAN-DL queue ODV3 queue State change queue PDO receive queue Event indication queue	Handle to another queue for the internal packet communication (to be used for trouble shooting)
Events indicated	Event Bit-list (1 Bit per event) to indicate the events in hexadecimal form Example: NMT state change: *Yes* Timestamp receive: *Yes* Error control event: *Yes* Receive PDO: *Yes* NMT command: *Yes* Then “Events indicated” = 0x00000001F
NMT state change Timestamp receive Error control event Receive PDO NMT command	Events, indicated to the application No: This event is not indicated to the application. Yes: this event is indicated to the application.

Table 108: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > Additional diagnostic

8.3.2.10 CANOPEN_SLAVE/Receive CAN-ID diagnostic



Receive CAN-ID diagnostic

Task states

Name	Value
CAN-ID 0-31	0x00000001
CAN-ID 32-63	0x00000000
CAN-ID 64-95	0x00000000
CAN-ID 96-127	0x00000000
CAN-ID 128-159	0x00000001
CAN-ID 160-191	0x00000000
CAN-ID 192-223	0x00000000
CAN-ID 224-255	0x00000000
CAN-ID 256-287	0x00000000
CAN-ID 288-319	0x00000000
CAN-ID 320-351	0x00000000
CAN-ID 352-383	0x00000000
CAN-ID 384-415	0x00000002
CAN-ID 416-447	0x00000000
CAN-ID 448-479	0x00000000
CAN-ID 480-511	0x00000000
CAN-ID 512-543	0x00000002
CAN-ID 544-575	0x00000000
CAN-ID 576-607	0x00000000
CAN-ID 608-639	0x00000000
CAN-ID 640-671	0x00000002
CAN-ID 672-703	0x00000000
CAN-ID 704-735	0x00000000
CAN-ID 736-767	0x00000000
CAN-ID 768-799	0x00000002
CAN-ID 800-831	0x00000000
CAN-ID 832-863	0x00000000
CAN-ID 864-895	0x00000000
CAN-ID 896-927	0x00000002
CAN-ID 928-959	0x00000000

Figure 86: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > Receive CAN-ID diagnostic

Name	Description
CAN-ID 0-31 .. CAN-ID 2016-2047	One bit-field for each CAN-ID, which is currently processed by the stack. Range of values for the CAN-ID: CAN-ID 0 to CAN-ID 2047 Example: **Name** / **Value** / **Meaning** CAN-ID **0**-31 / 0x0000000**1** / CAN-ID **0** has been processed CAN-ID 32-63 / 0x00000000 / no CAN-ID has been processed CAN-ID 64-95 / 0x00000000 / no CAN-ID has been processed CAN-ID 96-127 / 0x00000000 / no CAN-ID has been processed CAN-ID **128**-159 / 0x0000000**1** / CAN-ID **128** has been processed .. CAN-ID 384-415 / 0x0000000**4** / CAN-ID **386** has been processed .. CAN-ID 2016-2047 / 0x00000000 / no CAN-ID has been processed

Table 109: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > Receive CAN-ID diagnostic

8.3.2.11 CANOPEN_SLAVE/OBJ parameter diagnostic

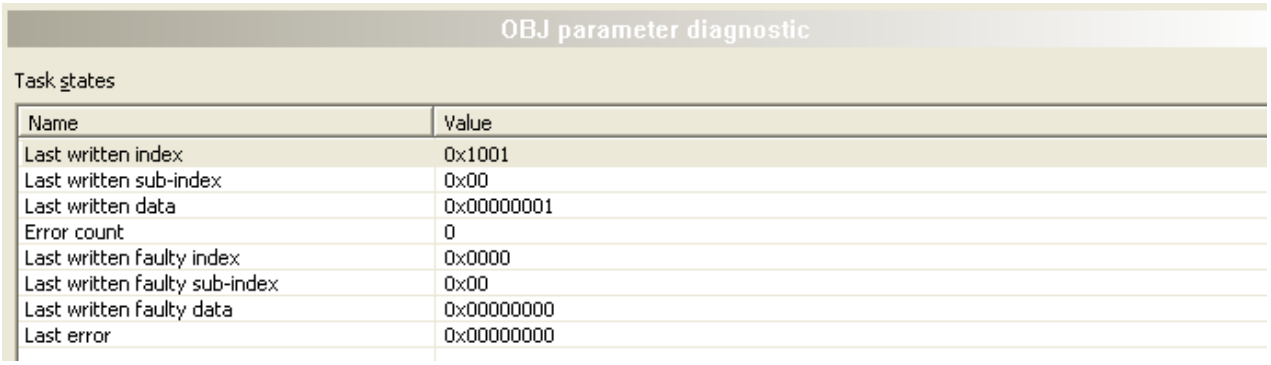


Figure 87: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > OBJ parameter diagnostic

Name	Description
Last written index Last written sub-index Last written data	Last written index/sub-index/data of the OBJ parameter diagnostic.
Error count	Diagnosis counter (described in the CANopen Slave Protocol API manual [8].)
Last written faulty index Last written faulty sub-index Last written faulty data Last Error	Last written faulty index/sub-index/data and last error of the OBJ parameter diagnostic.

Table 110: Extended Diagnosis CANopen Slave > CANOPEN_SLAVE > OBJ parameter diagnostic

8.3.2.12 CANOPEN_APS/Slave configuration

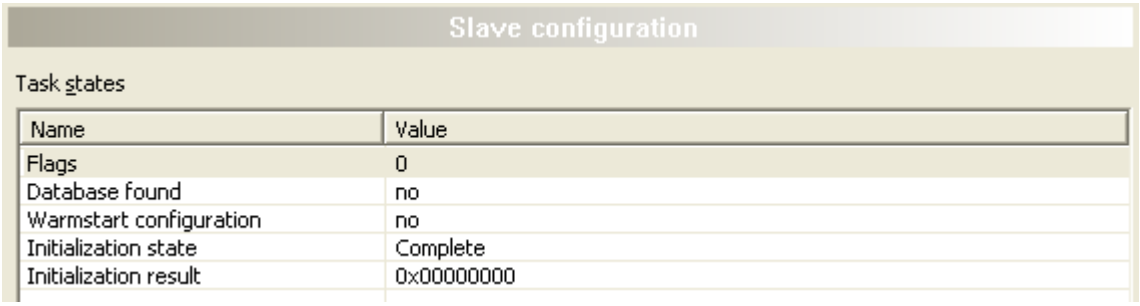


Figure 88: Extended Diagnosis CANopen Slave > CANOPEN_APS > Slave Configuration

Name	Description
Flags	Bit0 set: Configuration data base found not set: No configuration data base found Bit1 set: Configuration by packets not set: No packets for configuration
Database found	Yes: Configuration data base found No: No configuration data base found
Warmstart configuration	Yes: Configuration by packets No: No packets for configuration
Initialization state	Idle Send initialize request Wait for initialize confirmation Send register request Wait for register confirmation Send get buffer request Wait for get buffer confirmation Send bus parameter request Wait for bus parameter confirmation Complete Failed
Initialization result	Error code of the initialization, 0 = no error

Table 111: Extended Diagnosis CANopen Slave > CANOPEN_APS > Slave Configuration

8.3.2.13 CANOPEN_APS/Commands

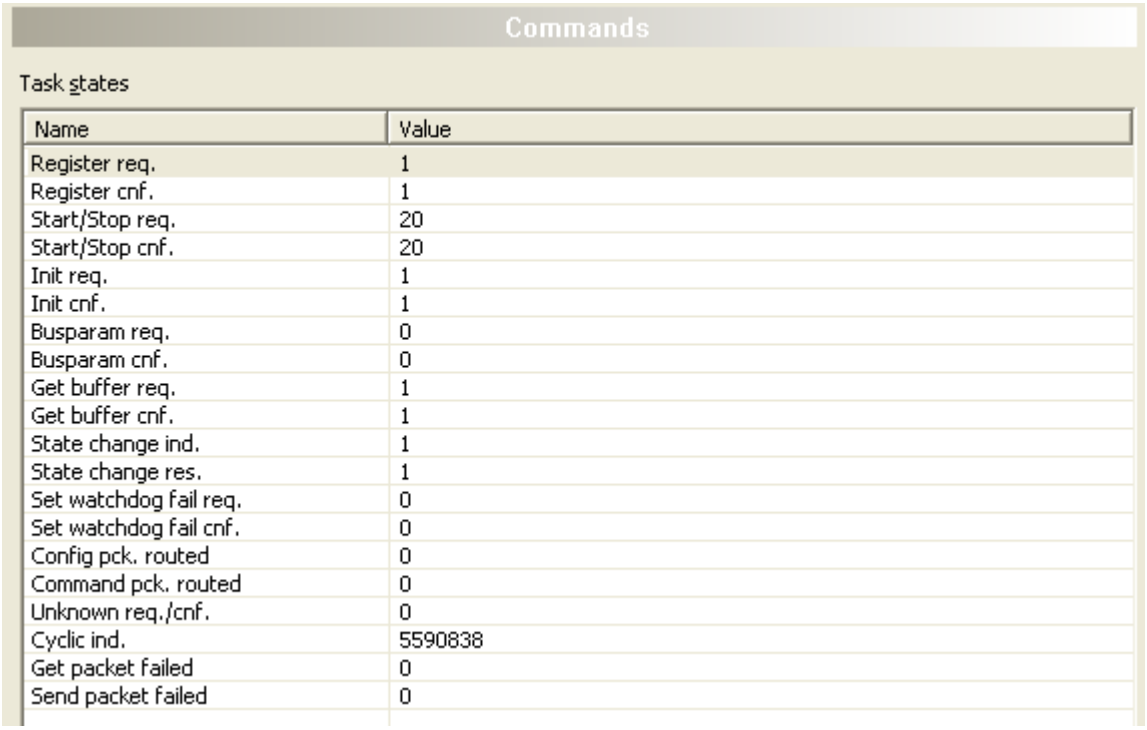
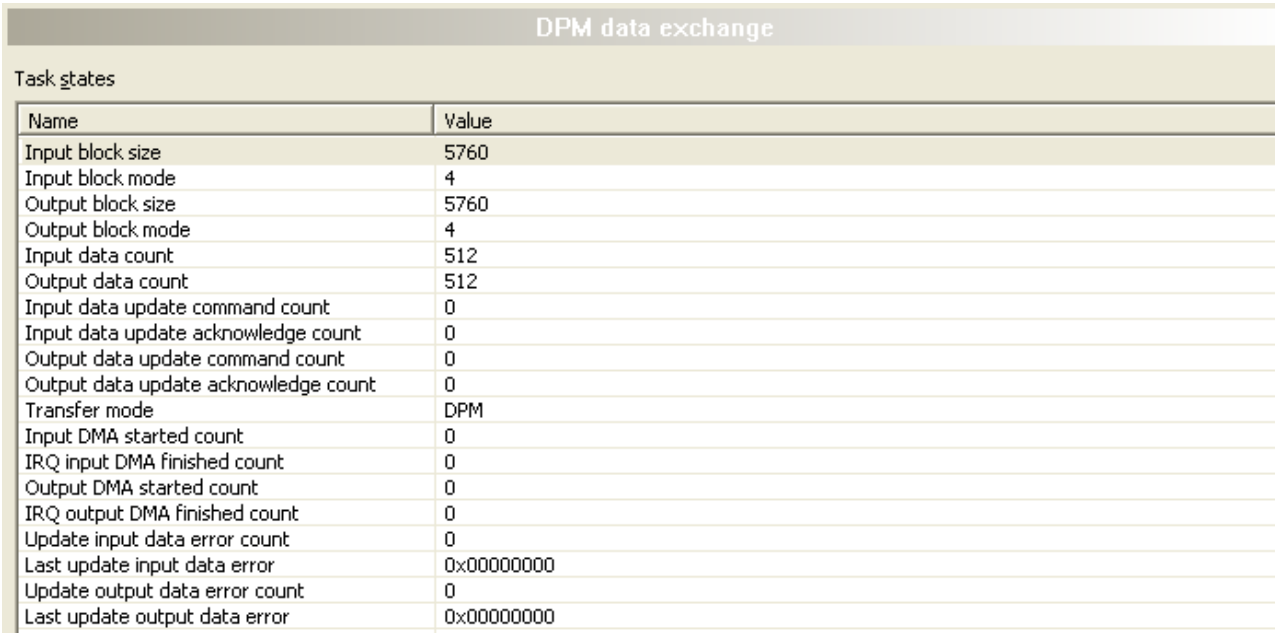


Figure 89: Extended Diagnosis CANopen Slave > CANOPEN_APS > Commands

Name	Description
[*Service*]	Diagnosis counter: Indicates the services processed. (The services of the single packets are described in the CANopen Slave Protocol API manual [8].)
Get packet failed	Number of errors during requesting a packet
Send packet failed	Number of errors during sending a packet

Table 112: Extended Diagnosis CANopen Slave > CANOPEN_APS > Commands

8.3.2.14 CANOPEN_APS/DPM data exchange



Name	Value
Input block size	5760
Input block mode	4
Output block size	5760
Output block mode	4
Input data count	512
Output data count	512
Input data update command count	0
Input data update acknowledge count	0
Output data update command count	0
Output data update acknowledge count	0
Transfer mode	DPM
Input DMA started count	0
IRQ input DMA finished count	0
Output DMA started count	0
IRQ output DMA finished count	0
Update input data error count	0
Last update input data error	0x00000000
Update output data error count	0
Last update output data error	0x00000000

Figure 90: Extended Diagnosis CANopen Slave > CANOPEN_APS > DPM data exchange

Name	Description
[*Services*]	Diagnosis counter: Indicate the services processed. (The services of the single packets are described in the CANopen Slave Protocol API manual [8].)
Input block size	Block size of the receive data in Byte
Input block mode	value (read out from the device) = 4 (host control)
Output block size	Block size of the send data in Byte
Output block mode	value (read out from the device) = 4 (host control)
Input data count	Diagnosis counter: Indicates the input data as an UINT 32 value*. * 1 UINT 32 = 4 Byte
Output data count	Diagnosis counter: Indicates the output data as an UINT 32 value*. * 1 UINT 32 = 4 Byte
Transfer mode	Possible values: “DPM”, “DMA” DPM = Dual-Port Memory DMA = Direct Memory Access
Last update input data error	Hexadecimal number identifying the last error occurred during the receive data are updated.
Last update output data error	Hexadecimal number identifying the last error occurred during the send data are updated.

Table 113: Extended Diagnosis CANopen Slave > CANOPEN_APS > Commands

8.3.3 DeviceNet-Slave

8.3.3.1 Overview DeviceNet Slave

Dialog Windows "Extended Diagnosis" DeviceNet Slave

The table below gives an overview for the **Extended Diagnosis** dialog windows descriptions:

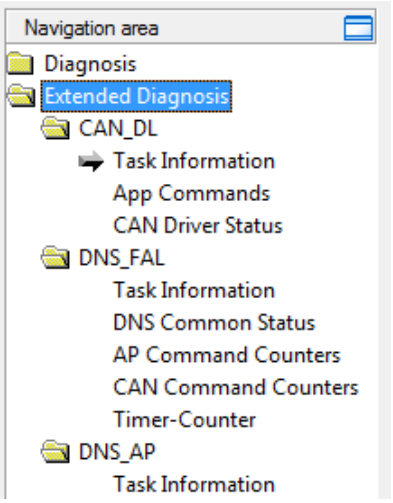
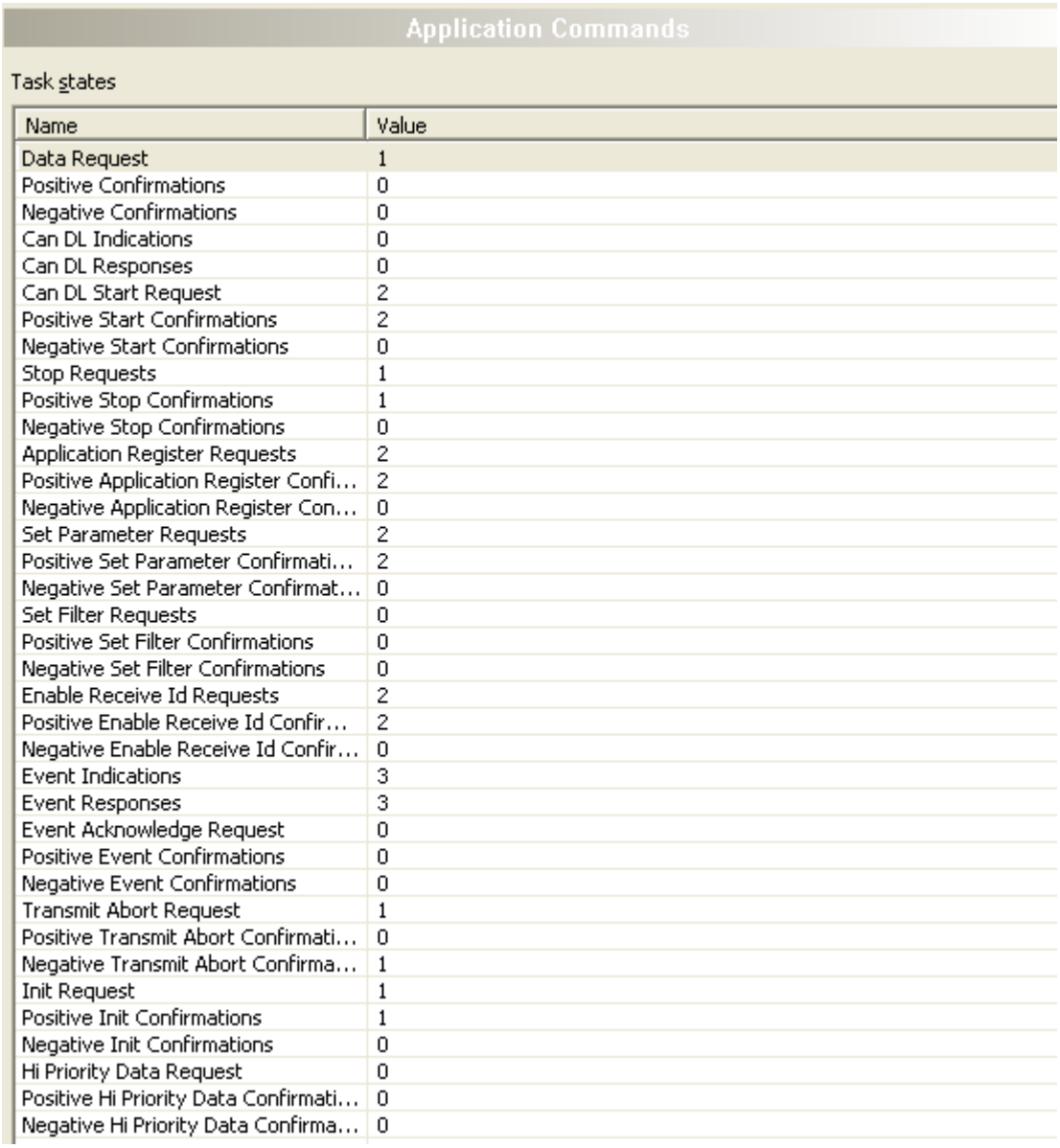
Navigationsbereich DeviceNet Slave*	Folder Name in the Navigation Area	Dialog Window	Manual Page
	*CAN_DL*	*CAN_DL/AP Commands Counter*	*180*
		CAN_DL/CAN Driver Status	*181*
	DNS_FAL	*DNS_FAL/DNS Common Status*	*182*
		DNS_FAL/AP Commands Counter	*183*
		DNS_FAL/CAN Command Counter	*183*
		DNS_FAL/Timer Counter	*184*
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		** The displayed DeviceNet Slave exteded diagnosis corresponds to the DeviceNet Slave firmware version 2.3.x.x.*	

Table 114: Descriptions of the Dialog Windows Extended Diagnosis DeviceNet Slave

8.3.3.2 CAN_DL/AP Commands Counter



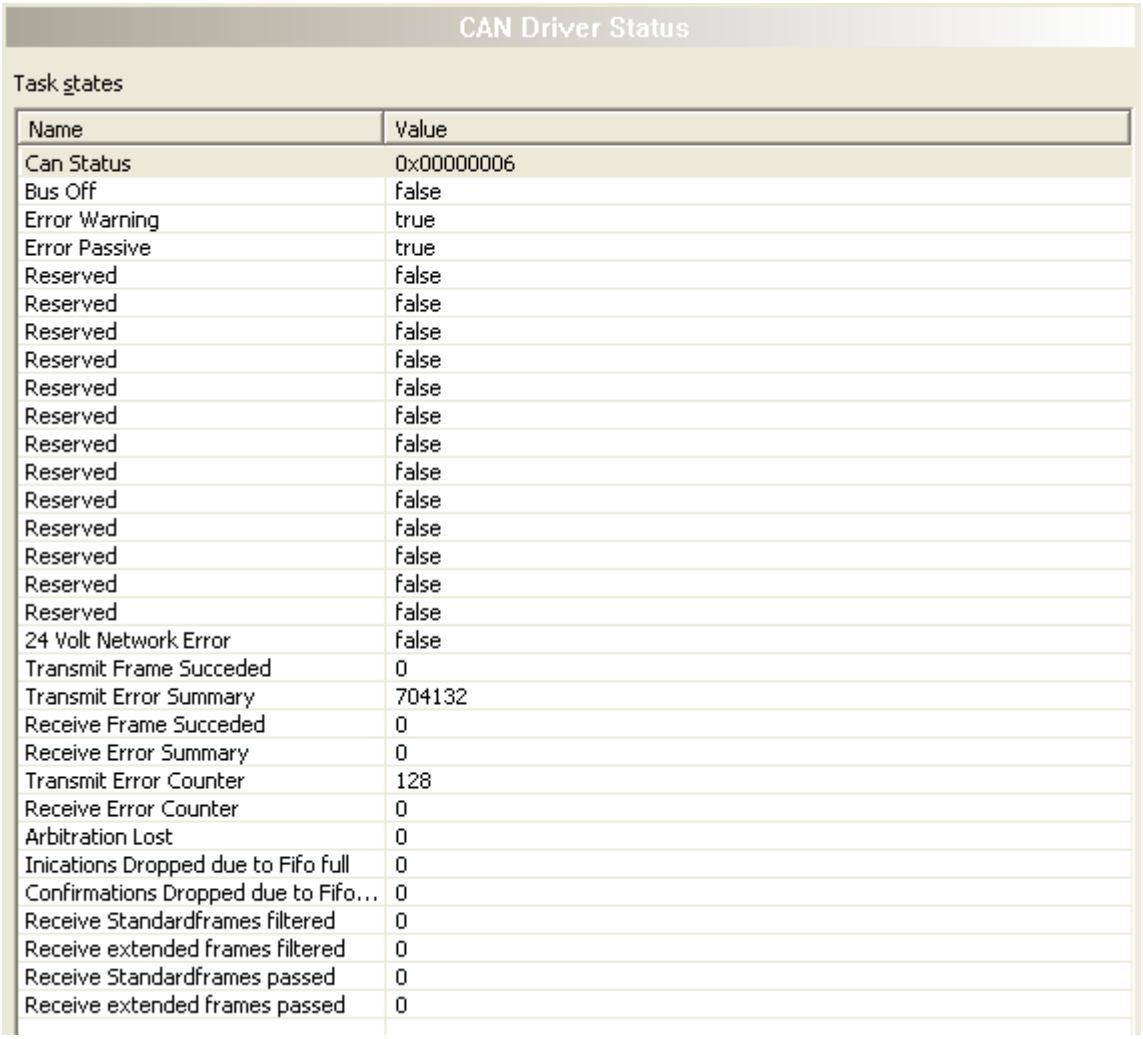
Name	Value
Data Request	1
Positive Confirmations	0
Negative Confirmations	0
Can DL Indications	0
Can DL Responses	0
Can DL Start Request	2
Positive Start Confirmations	2
Negative Start Confirmations	0
Stop Requests	1
Positive Stop Confirmations	1
Negative Stop Confirmations	0
Application Register Requests	2
Positive Application Register Confi...	2
Negative Application Register Con...	0
Set Parameter Requests	2
Positive Set Parameter Confirmati...	2
Negative Set Parameter Confirmat...	0
Set Filter Requests	0
Positive Set Filter Confirmations	0
Negative Set Filter Confirmations	0
Enable Receive Id Requests	2
Positive Enable Receive Id Confir...	2
Negative Enable Receive Id Confir...	0
Event Indications	3
Event Responses	3
Event Acknowledge Request	0
Positive Event Confirmations	0
Negative Event Confirmations	0
Transmit Abort Request	1
Positive Transmit Abort Confirmati...	0
Negative Transmit Abort Confirma...	1
Init Request	1
Positive Init Confirmations	1
Negative Init Confirmations	0
Hi Priority Data Request	0
Positive Hi Priority Data Confirmati...	0
Negative Hi Priority Data Confirma...	0

Figure 91: Extended Diagnosis DeviceNet Slave > CAN_DL > AP Commands Counter

Name	Description
[*Service*]	Diagnosis counter of the CAN layer. Indicates the services processed. (The services of the single packets are described in the DeviceNet Slave Protocol API manual [11].)

Table 115: Extended Diagnosis DeviceNet Slave > CAN_DL > AP Commands Counter

8.3.3.3 CAN_DL/CAN Driver Status



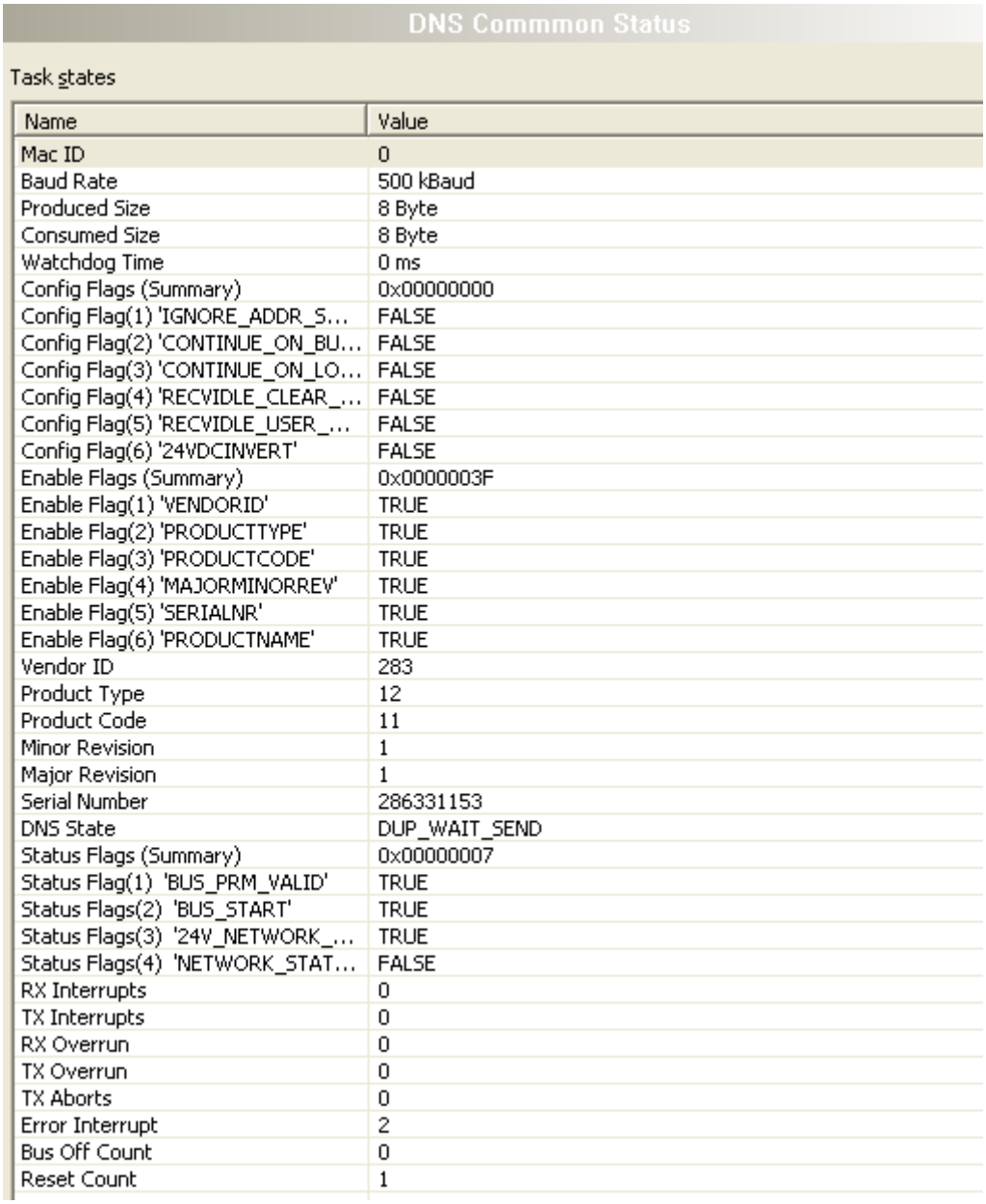
Name	Value
Can Status	0x00000006
Bus Off	false
Error Warning	true
Error Passive	true
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
Reserved	false
24 Volt Network Error	false
Transmit Frame Succeded	0
Transmit Error Summary	704132
Receive Frame Succeded	0
Receive Error Summary	0
Transmit Error Counter	128
Receive Error Counter	0
Arbitration Lost	0
Inications Dropped due to Fifo full	0
Confirmations Dropped due to Fifo...	0
Receive Standardframes filtered	0
Receive extended frames filtered	0
Receive Standardframes passed	0
Receive extended frames passed	0

Figure 92: Extended Diagnosis DeviceNet Slave > CAN_DL > CAN Driver Status

Name	Description
CAN Status Bus Off Error Warning ErrorPassive	Diagnosis status of CAN specific error levels. Indicates the respective status of the CAN bus. (For further information refer to the DeviceNet Slave Protocol API manual [11].)
Reserved	Diagnosis status
24 Volt Network Error	Diagnosis status of DeviceNet specific error levels. (For further information refer to the DeviceNet Slave Protocol API manual [11].)
[*Service*]	Diagnosis counter of CAN errors. Indicates the services processed. (The services of the single packets are described in the DeviceNet Slave Protocol API manual [11].)

Table 116: Extended Diagnosis DeviceNet Slave > CAN_DL > CAN Driver Status

8.3.3.4 DNS_FAL/DNS Common Status



Name	Value
Mac ID	0
Baud Rate	500 kBaud
Produced Size	8 Byte
Consumed Size	8 Byte
Watchdog Time	0 ms
Config Flags (Summary)	0x00000000
Config Flag(1) 'IGNORE_ADDR_S...	FALSE
Config Flag(2) 'CONTINUE_ON_BU...	FALSE
Config Flag(3) 'CONTINUE_ON_LO...	FALSE
Config Flag(4) 'RECVIDLE_CLEAR_...	FALSE
Config Flag(5) 'RECVIDLE_USER_...	FALSE
Config Flag(6) '24VDCINVERT'	FALSE
Enable Flags (Summary)	0x0000003F
Enable Flag(1) 'VENDORID'	TRUE
Enable Flag(2) 'PRODUCTTYPE'	TRUE
Enable Flag(3) 'PRODUCTCODE'	TRUE
Enable Flag(4) 'MAJORMINORREV'	TRUE
Enable Flag(5) 'SERIALNR'	TRUE
Enable Flag(6) 'PRODUCTNAME'	TRUE
Vendor ID	283
Product Type	12
Product Code	11
Minor Revision	1
Major Revision	1
Serial Number	286331153
DNS State	DUP_WAIT_SEND
Status Flags (Summary)	0x00000007
Status Flag(1) 'BUS_PRM_VALID'	TRUE
Status Flags(2) 'BUS_START'	TRUE
Status Flags(3) '24V_NETWORK_...	TRUE
Status Flags(4) 'NETWORK_STAT...	FALSE
RX Interrupts	0
TX Interrupts	0
RX Overrun	0
TX Overrun	0
TX Aborts	0
Error Interrupt	2
Bus Off Count	0
Reset Count	1

Figure 93: Extended Diagnosis DeviceNet Slave > DNS_FAL > DNS Common Status

Name	Description
[*Status*]	Common DNS Diagnosis status. Indicates the current status of the single tasks. (For further information refer to the DeviceNet Slave Protocol API manual [11].)

Table 117: Extended Diagnosis DeviceNet Slave > DNS_FAL > DNS Common Status

8.3.3.5 DNS_FAL/AP Commands Counter

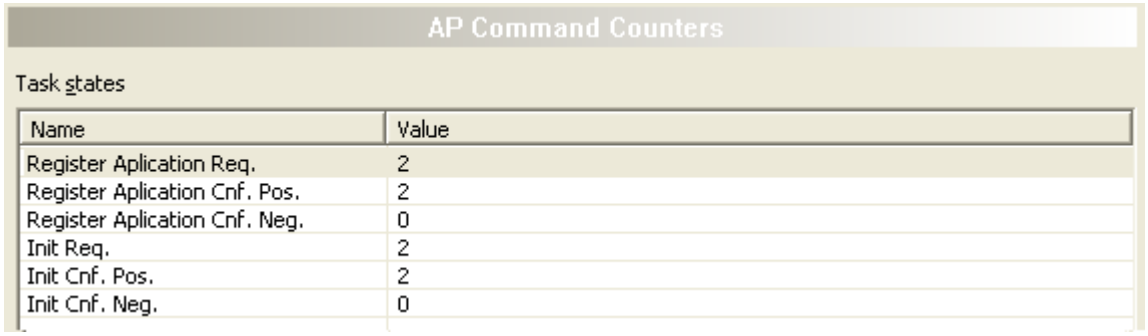


Figure 94: Extended Diagnosis DeviceNet Slave > DNS_FAL > AP Commands Counter

Name	Description
[*Service*]	Diagnosis counter of the FAL layer. Indicates the services processed. (The services of the single packets are described in the DeviceNet Slave Protocol API manual [11].)

Table 118: Extended Diagnosis DeviceNet Slave > DNS_FAL > AP Commands Counter

8.3.3.6 DNS_FAL/CAN Command Counter

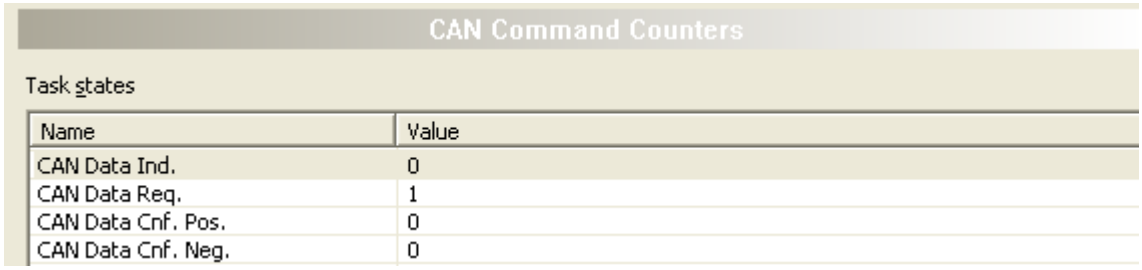


Figure 95: Extended Diagnosis DeviceNet Slave > DNS_FAL > CAN Command Counter

Name	Description
[*Service*]	Diagnosis counter of the FAL layer. Indicates the services processed. (The services of the single packets are described in the DeviceNet Slave Protocol API manual [11].)

Table 119: Extended Diagnosis DeviceNet Slave > DNS_FAL > CAN Command Counter

8.3.3.7 DNS_FAL/Timer Counter

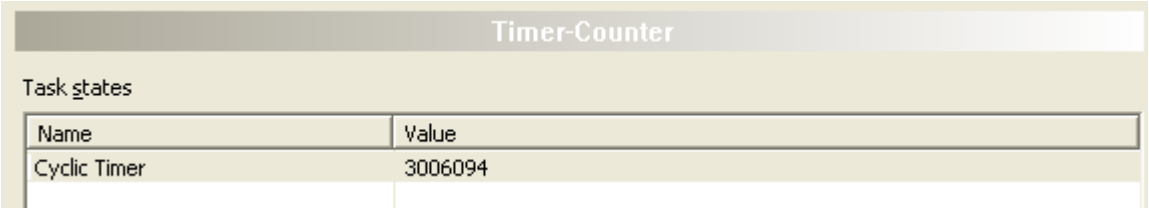


Figure 96: Extended Diagnosis DeviceNet Slave > DNS_FAL > Timer Counter

Name	Description
[*Service*]	Diagnosis counter of the FAL layer. Indicates the services processed. (The services of the single packets are described in the DeviceNet Slave Protocol API manual [11].)

Table 120: Extended Diagnosis DeviceNet Slave > DNS_FAL > Timer Counter

8.3.4 CC-Link Slave

8.3.4.1 Overview CC-Link Slave

Dialog Windows "Extended Diagnosis" CC-Link Slave

The table below gives an overview for the **Extended Diagnosis** dialog windows descriptions:

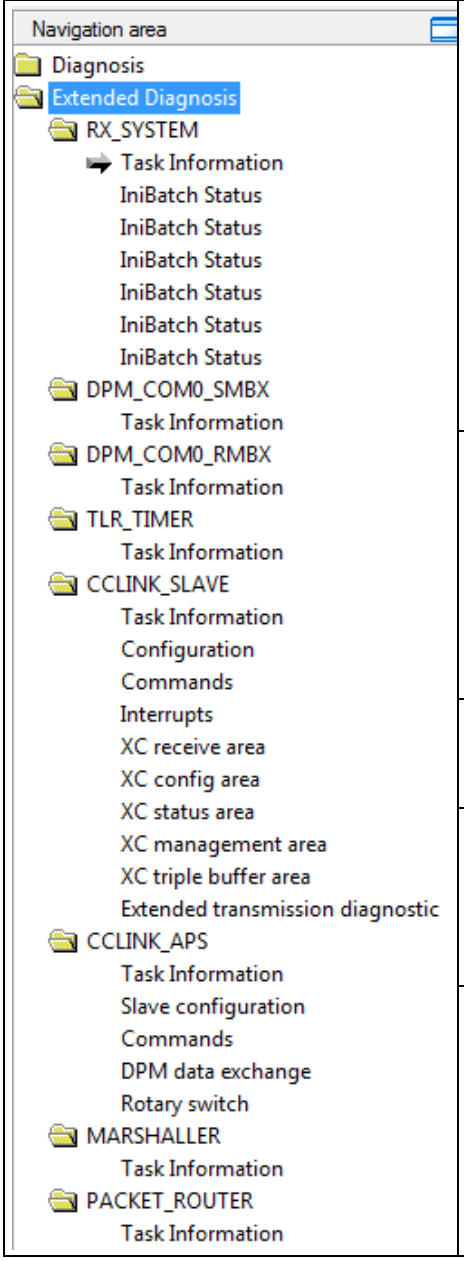
Navigation Area CC-Link Slave	Folder Name in the Navigation Area	Dialog Window	Manual Page
	*CCLINK_SLAVE*	*CCLINK_SLAVE/Configuration*	186
		CCLINK_SLAVE/Commands	188
		CCLINK_SLAVE/Interrupts	189
		CCLINK_SLAVE/XC receive area	189
		CCLINK_SLAVE/XC config area	190
		CCLINK_SLAVE/XC status area	193
		CCLINK_SLAVE/XC management area	196
		CCLINK_SLAVE/XC triple buffer area	198
		CCLINK_SLAVE/Extended transmission diagnostic	199
	CCLINK_APS	*CCLINK_APS/Slave configuration*	200
		CCLINK_APS/Commands	201
		CCLINK_APS/DPM data exchange	202
		CCLINK_APS/Rotary Switch	203
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		**Note:** For netGateway devices the Extended Diagnosis for **DPM_COMO_SMBX**, **DPM_COMO_SMBX**, **Marshaller** and **Packet-Router** is not displayed.	
		** The displayed CC-Link Slave exteded diagnosis corresponds to the CC-Link Slave firmware version 2.7.x.x.*	

Table 121: Descriptions of the Dialog Windows Extended Diagnosis CC-Link Slave

8.3.4.2 CCLINK_SLAVE/Configuration

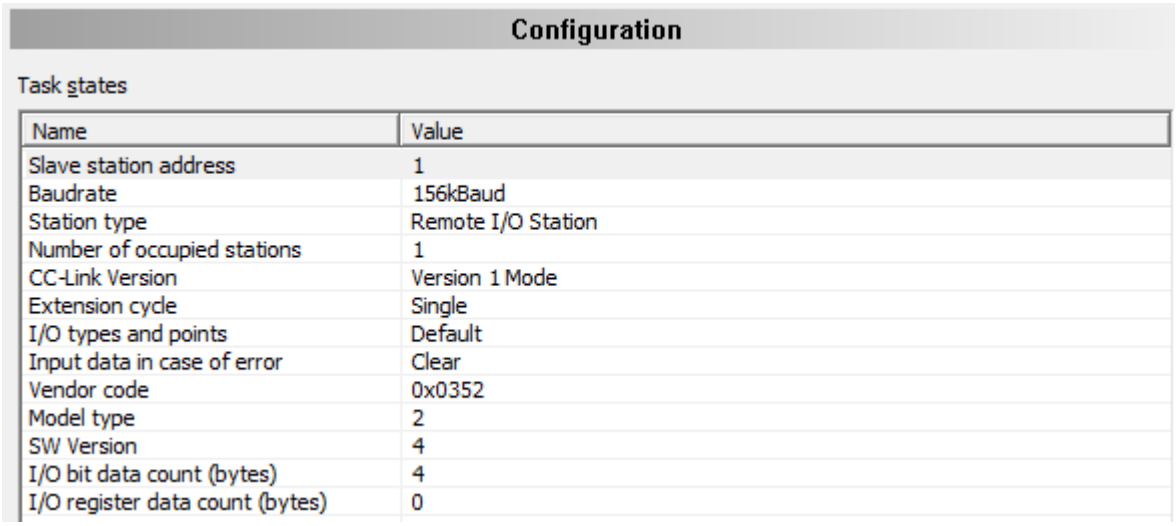


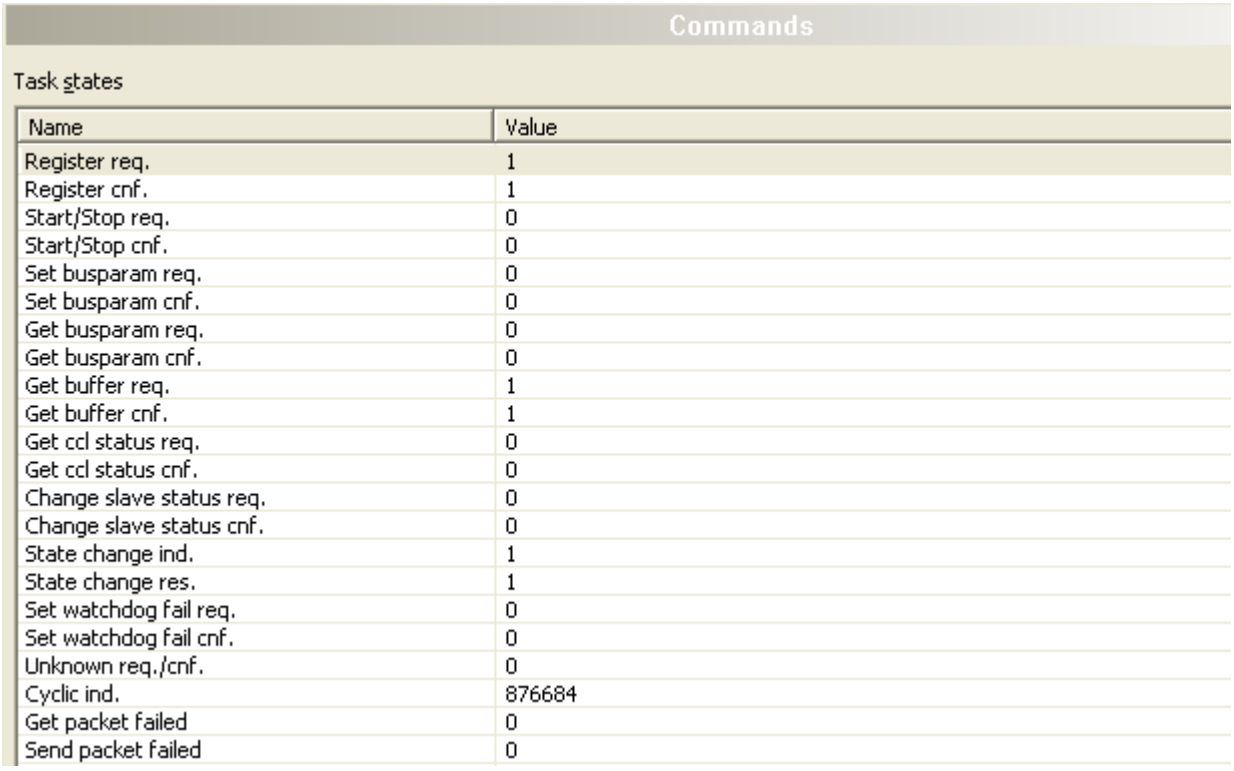
Figure 97: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > Configuration

Name	Description
Slave station address	Station address of CC-Link Slave Note: The number of occupied stations plus station address must not exceed the parameter range Range of value: 1 ... 64
Baudrate	- Not configured - 156kBaud - 625kBaud - 2500kBaud - 5MBaud - 10MBaud
Station type	- Not configured - Remote I/O Station - Remote Device Station - Intelligent Device Station
Number of occupied stations	Number of occupied stations Remote I/O Station: 1 (Default) Remote Device Station: 1 ... 4
CC Link Version	- Not configured - Version 1 Mode - Version 2 Mode
Extension cycle	Number of extension cycles: - Not configured - Single/One cycle - Double/Two cycles - Quadruple/Four cycles - Octuple/Eight cycles Allowed numbers for CC-Link version 1: Single Allowed numbers for CC-Link version 2: Single, Double, Quadruple, Octuple

Name	Description
I/O types and points	Total number of points; I/O type available: Mixed, Input, Output, Composite - Not configured - Default - Mixed, depending on number of occupied station - Mixed, 8 points (Bits) - Mixed, 16 points (Bits) - Mixed, 32 points (Bits) - Input, depending on number of occupied station - Input, 8 points (Bits) - Input, 16 points (Bits) - Input, 32 points (Bits) - Output, depending on number of occupied station - Output, 8 points (Bits) - Output, 16 points (Bits) - Output, 32 points (Bits) - Composite, depending on number of occupied station - Composite, 8 points (Bits) - Composite, 16 points (Bits) - Composite, 32 points (Bits) “Mixed” means that both input and output exist on the same module. “Composite” means a device that doesn’t use the same numbers for input and output.
Input data in case of error	- Not configured - Clear - Hold
Vendor code	Vendor code (If corresponding bit in the vendor code flags parameter is set) Range of value: 0 ... 65535, Default: 0x0352
Model type	Model type (If corresponding bit in the model type flags parameter is set) Range of value: 0 ... 255
SW Version	Software version (If corresponding bit in the software version flags parameter is set) Range of value: 0 ... 63, Default: 1
I/O bit data count (bytes)	Counter for I/O bits in bytes
I/O register data count (bytes)	Counter for I/O register in bytes

Table 122: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > Configuration

8.3.4.3 CCLINK_SLAVE/Commands



Name	Value
Register req.	1
Register cnf.	1
Start/Stop req.	0
Start/Stop cnf.	0
Set busparam req.	0
Set busparam cnf.	0
Get busparam req.	0
Get busparam cnf.	0
Get buffer req.	1
Get buffer cnf.	1
Get ccl status req.	0
Get ccl status cnf.	0
Change slave status req.	0
Change slave status cnf.	0
State change ind.	1
State change res.	1
Set watchdog fail req.	0
Set watchdog fail cnf.	0
Unknown req./cnf.	0
Cyclic ind.	876684
Get packet failed	0
Send packet failed	0

Figure 98: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > Commands

Name	Description
[*Service*]	Diagnosis counter: Indicates the services processed. (The services of the single packets are described in the CC-Link Slave Protocol API manual [7].)
Get packet failed	Number of errors during requesting a packet
Send packet failed	Number of errors during sending a packet

Table 123: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > Commands

8.3.4.4 CCLINK_SLAVE/Interrupts

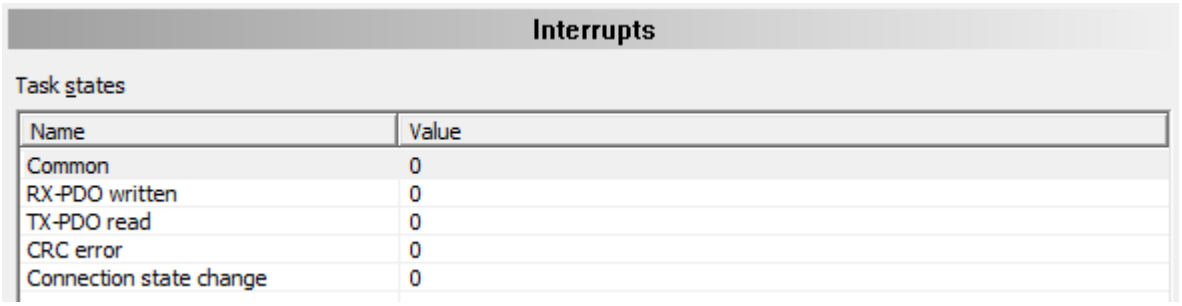


Figure 99: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > Interrupts

Name	Description
[*Service*]	Diagnosis counter: Indicates the services processed. (The services of the single packets are described in the CC-Link Slave Protocol API manual [7].)

Table 124: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > Interrupts

8.3.4.5 CCLINK_SLAVE/XC receive area

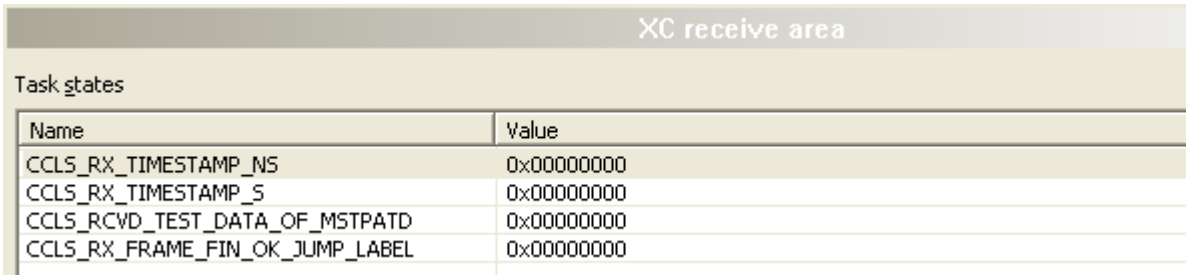


Figure 100: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > XC receive area

Note: The **Extended Diagnosis > CCLINK_SLAVE > XC receive area** serves only for internal diagnosis purposes.

8.3.4.6 CCLINK_SLAVE/XC config area

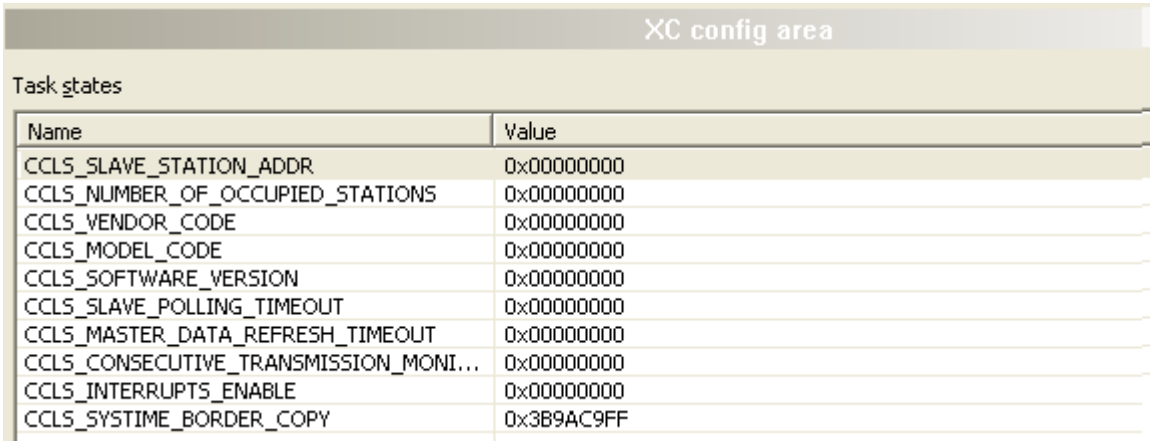


Figure 101: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > XC config area

ID	Value	Description
CCLS_SLAVE_STATION_ADDR	1 … 64	**Slave station address**
CCLS_NUMBER_OF_OCCUPIED_STATIONS	1 … 4	**Number of occupied stations**
CCLS_VENDOR_CODE		**Vendor code**
CCLS_MODEL_CODE	0 … 2^{32-1}	**Model code** The model codes are allocated in vendor modules. Bits 1 and 0: **Total number of IO-points** 00: dependent of the number of occupied stations, 01: 8 points, 10: 32 points, 11: 16 points Bits 3 and 2: **I/O types** 00: mixed: Where both inputs and outputs exist on the same module. The same I/O numbers (starting from RX0 and RY0, respectively) are used. 01: input, 10: output, 11: composite: A device that does not use the same numbers for input (RX) and output (RY) Bits 5 and 4: **Number of occupied stations** 00: 1 station occupied 01: 2 stations occupied 10: 3 stations occupied 11: 4 stations occupied Bit 7 and 6: *Reserved* Bit 8: **Switch Setting** 0: Normal, 1: Abnormal Bit 9: **Output status when error occurs** 0: Clear, 1: Hold Bits 13 to 10: *reserved*

ID	Value	Description
(continued) CCLS_MODEL_CODE		Bits 15 and 14: **Station Type** (Type of station) 00: remote I/O 01: remote device 10: intelligent device/local 11: Reserved Bits 23 and 16: **Model Type** (This byte defines the model of the slave station. This is allocated seperately for each model.) Bits 31 and 24: *reserved*
CCLS_SOFTWARE_VERSION	0 … 2^{32-1}	**Software version** (Machine information, set by ARM before CC-Link Slave is started) Bits 5 to 0: **Software Version Vendor** (This indicates the software version of each model; dependent on the vendor) Bits 7 and 6: **Protocol Version** (for Version 2) 00: Version 1 01: Version 2 10: Version 3 (Future function) 11: Version 4 (Future function) Bits 31 to 8: *reserved*
CCLS_SLAVE_POLLING_TIMEOUT	0 … 2^{32-1}	Slave polling timeout Bits 31 to 0: **Slave Station Polling**, Timout value in 10ns resolution, 0: timeout monitoring disabled
CCLS_MASTER_DATA_REFRESH_TIMEOUT	0 … 2^{32-1}	Master data refresh timeout Bits 31 to 0: **Master Data Refresh**, Timeout value in 10ns resolution, 0: timeout monitoring disabled
CCLS_CONSECUTIVE_TRANSMISSION_MONITORING_TIMEOUT	0 … 2^{32-1}	Consecutive transmission monitoring timeout Bits 31 to 0: **Monitoring of consecutive transmission time**, Timeout value in 10ns resolution, 0: timeout monitoring disabled
CCLS_INTERRUPTS_ENABLE	0 … 2^{32-1}	Bit 0: **RX Test Data written by Master** (enable event "Master has written new test-loopback data"; to inform ARM that Master Status was updated) Bit 1: **TX Test Data ready by Master** (enable event "Master has polled test-loopback data") Bit 2: **RX PDO written by Master** (enable event "Master has refreshed RXPDO") Bit 3: **TX PDO read by Master** (enable event "Master has polled TXPDO") Bit 4: **Refresh Cycle End** (enable event "Refresh cycle end")

ID	Value	Description
(continued) CCLS_INTERRUPTS_ENABLE		Bit 5: **Slave polling Timeout** (enable event "Slave polling timeout") Bit 6: **Master Data refresh Timeout** (enable event "Master data refresh timeout") Bit 7: **Consecutive Transmission Timeout** (enable event "Consecutive Transmission monitoring timeout") Bit 8: **CRC Error** (enable event "CRC error detected") Bit 9: **Slave connected State changed** (enable event "The connection state of the slave has changed") Bit 10: **RX Message Buffer written by Master** (enable event "Receive message buffer written by master") Bit 11: **TX Message Buffer read by Master** (enable event "Transmit message buffer read by master") Bit 15 to 12: *reserved* Bit 31 to 16: *reserved* (unusable)
CCLS_SYSTIME_BORDER_COPY	0 … 2^{32-1}	Bits 31 to 0: **Systime border copy**

Table 125: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > XC config area

8.3.4.7 CCLINK_SLAVE/XC status area

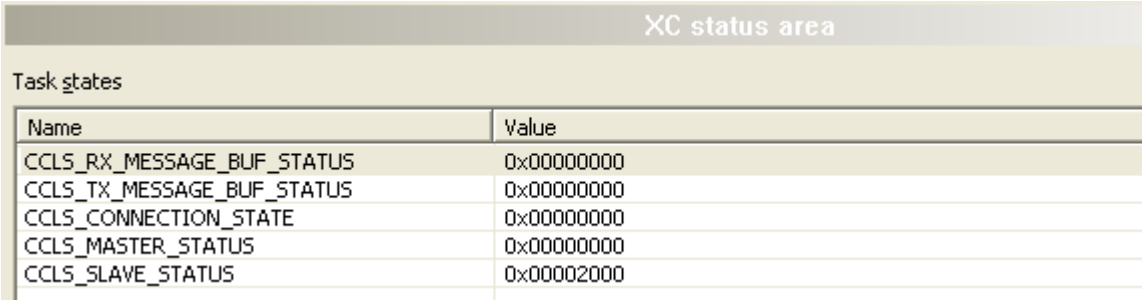


Figure 102: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > XC status area

ID	Value	Description
CCLS_RX_MESSAGE_BUF_STATUS	0 … 2^{32-1}	0: Receive Message Buffer Empty, otherwise: Receive Message Buffer Full, set by xPEC, reset by ARM
CCLS_TX_MESSAGE_BUF_STATUS	0 … 2^{32-1}	0: Transmit Message Buffer Empty, otherwise: Transmit Message Buffer Full, set by ARM, reset by xPEC
CCLS_CONNECTION_STATE	0 … 2^{32-1}	Bit 0: **State** (connection state of slave), 0: disconnected, 1: connected, written by xPEC Bit 1: **Network joined** 0: *reserved* 1: local slave was test polled by master Bits 15 and 2: *reserved* Bit 16: **Consecutive Transmission Timeout Status** 0: no error 1: error Bits 31 and 17: *reserved*
CCLS_MASTER_STATUS	0 … 2^{32-1}	Bit 0: **Status1 Master Status User Application Programm** (Operation status Master station user application program) 0: stop 1: run Bit 1: **Status1 Master Status User Application Programm Error Check** (An error occured in the master station user application program) 0: normal 1: error Bit 2: **Status1 refresh Startup** (Link refresh was started) 0: stop 1: start

ID	Value	Description
(continued) CCLS_MASTER_STATUS		Bit 3: **Status1 Transient data status** (Transient data is included) 0: no error 1: error Bit 4: **Status1 Transient data reception enabled** (Ready to receive transient data.) 0: disabled 1: enabled Bits 6 and 5: **Status1 Protocol version** 00: Version 1 01: Version 2 10: Version 3 (Future function) 11: Version 4 (Future function) Bit 7: **Status1 Master Station Type** (Station type of the master station) 0: Master station 1: Standby master station Bits 11 and 8: **Status 2 RY Info Transmission Points** (RY: Remote output length; multiplied with 32 bytes) Bits 15 and 12: **Stauts 2 RWW Info Transmission Points** (RWW: Remote register output length; multiplied with 64 bytes) Bit 31 to 16: *reserved*
CCLS_SLAVE_STATUS	0 … 2^{32-1}	Bit 0: **Station 1 fuse Status** (Salve station fuse has blown) 0: No error 1: Error Bit 1: **Station 1 unit Error invalid Number of Points** (Unit error (remote I/O stations), Invalid number of points flag; slave stations excluding remote I/O stations) 0: No error 1: Error Bit 2: **Station 1 no refresh receive** (No refresh transmission received from the master station; shall be set to zero because bit is inserted by xPEC) 0: Not received 1: Received Bit 3: **Station 1 no parameter receive** (No parameter information received from the master station) 0: Not received 1: Received Bit 4: **Station 1 switch change detection** (Switch setting changed after power is turned on or reset cancelled) 0: Not present 1: Present

ID	Value	Description
(continued) CCLS_SLAVE_STATUS	0 … 2^{32-1}	Bit 5: **Station 1 cyclic Communication** (Cyclic communication is enabled) 0: Enabled 1: Disabled Bit 6: *reserved* Bit 7: **Station 1 Watchdog Error** (Watchdog timer error has occurred) 0: Not present 1: Present Bit 8: **Station 2 Transient Data Status** (Transient data is included; shall be set to zero because bit is inserted by xPEC) 0: No transient data is included 1: Transient data is included Bit 9: **Status2 Transient data reception enabled** (Ready to receive transient data; shall be set to zero because bit is inserted by xPEC) 0: Disabled 1: Enabled Bit 10: **Status2 Transient Type** (Type of transient data) 0: 1:n communication (intelligent device station) 1: n:n communication (local station) Bit 11: *reserved* Bit 12: **Status2 Transmission Route Status** (Transmission route error) 0: No error 1: Error Bit 13: **Station 2 res fixed to one** *reserved* (set to 1) Bit 15 to 14: **State 2 extended Cycle Setting** (Extended cycle setting; Version 2) 00: Single setting (1*) 01: Double setting (2*) 10: Quadruple setting (4*) 11: Octuple setting (8*) Bit 31 to 16: *reserved*

Table 126: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > XC status area

8.3.4.8 CCLINK_SLAVE/XC management area

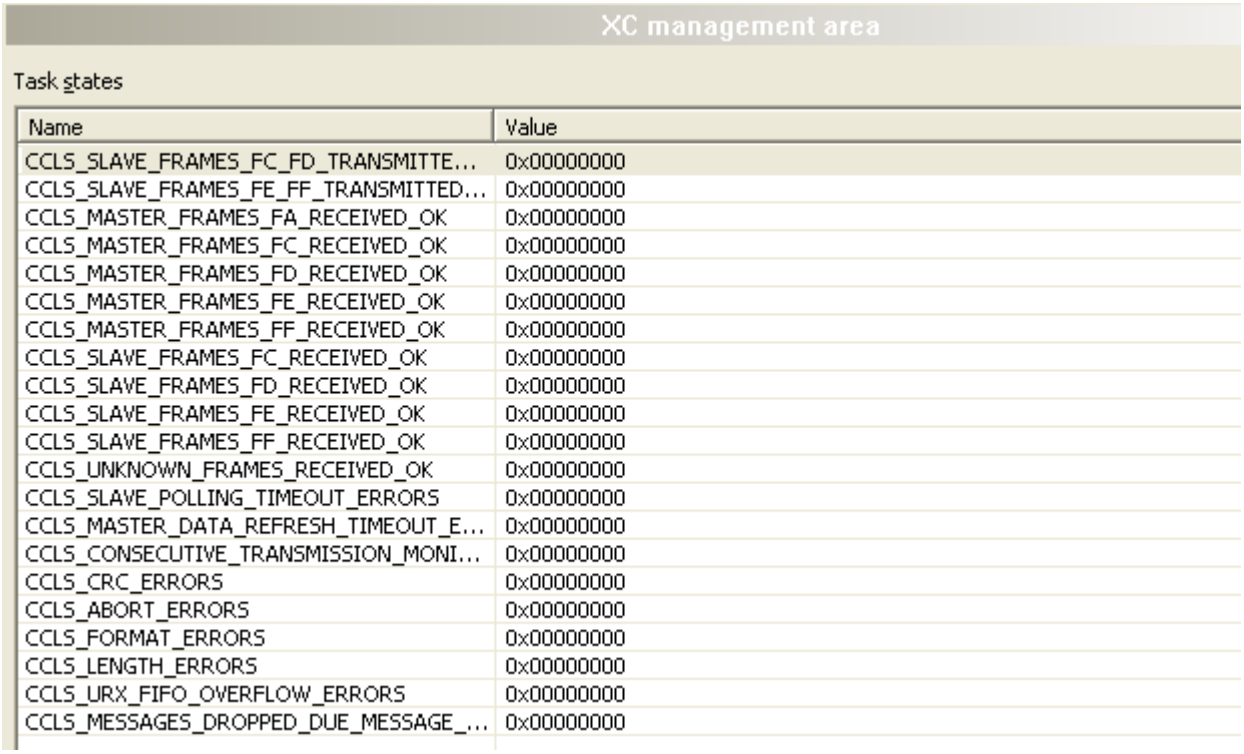


Figure 103: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > XC management area

ID	Value	Description
CCLS_SLAVE_FRAMES_FC_FD_TRANSMITTED_OK	0 … 2^{32-1}	Counter for frames of type "Slave station test loop-back data" that are successfully transmitted
CCLS_SLAVE_FRAMES_FE_FF_TRANSMITTED_OK	0 … 2^{32-1}	Counter for frames of type "Slave station response (refresh) data" that are successfully transmitted
CCLS_MASTER_FRAMES_FA_RECEIVED_OK	0 … 2^{32-1}	Counter for frames of type "Refresh cycle end data" that are successfully received
CCLS_MASTER_FRAMES_FC_RECEIVED_OK	0 … 2^{32-1}	Counter for frames of type "Master station test polling data" that are successfully received
CCLS_MASTER_FRAMES_FD_RECEIVED_OK	0 … 2^{32-1}	Counter for frames of type "Master station test polling and test data" that are successfully received
CCLS_MASTER_FRAMES_FE_RECEIVED_OK	0 … 2^{32-1}	Counter for frames of type "Master station polling data" that are successfully received
CCLS_MASTER_FRAMES_FF_RECEIVED_OK	0 … 2^{32-1}	Counter for frames of type "Master station polling and refresh data" that are successfully received
CCLS_SLAVE_FRAMES_FC_RECEIVED_OK	0 … 2^{32-1}	Counter for frames of type "Slave station test loopback data" that are successfully received
CCLS_SLAVE_FRAMES_FD_RECEIVED_OK	0 … 2^{32-1}	Counter for frames of type "Slave station response (refresh) data" that are successfully received
CCLS_SLAVE_FRAMES_FE_RECEIVED_OK	0 … 2^{32-1}	Counter for frames of type "Slave station response (refresh) data" that are successfully received

ID	Value	Description
CCLS_SLAVE_FRAMES_FF_RECEIVED_OK	0 … 2^{32-1}	Counter for frames of type "Slave station test loopback data" that are successfully received
CCLS_UNKNOWN_FRAMES_RECEIVED_OK	0 … 2^{32-1}	Counter for frames of unknown type that are successfully received
CCLS_SLAVE_POLLING_TIMEOUT_ERRORS	0 … 2^{32-1}	Counter for Slave Station Polling Timout errors
CCLS_MASTER_DATA_REFRESH_TIMEOUT_ERRORS	0 … 2^{32-1}	Counter for Master Data Refresh Timeout errors
CCLS_CONSECUTIVE_TRANSMISSION_MONITORING_TIMEOUT_ERRORS	0 … 2^{32-1}	Counter for consecutive transmission timeout errors
CCLS_CRC_ERRORS	0 … 2^{32-1}	Counter for frames that do not pass the CRC check
CCLS_ABORT_ERRORS	0 … 2^{32-1}	Counter for frames that were aborted during reception
CCLS_FORMAT_ERRORS	0 … 2^{32-1}	Counter for frames that have not the expected frame format
CCLS_LENGTH_ERRORS	0 … 2^{32-1}	Counter for frames that have not the expected frame length
CCLS_URX_FIFO_OVERFLOW_ERRORS	0 … 2^{32-1}	Counter for frames that have an urx fifo overflow error
CCLS_MESSAGES_DROPPED_DUE_MESSAGE_BUF_FULL	0 … 2^{32-1}	Counter for times a message was dropped due message buffer was full

Table 127: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > XC management area

8.3.4.9 CCLINK_SLAVE/XC triple buffer area

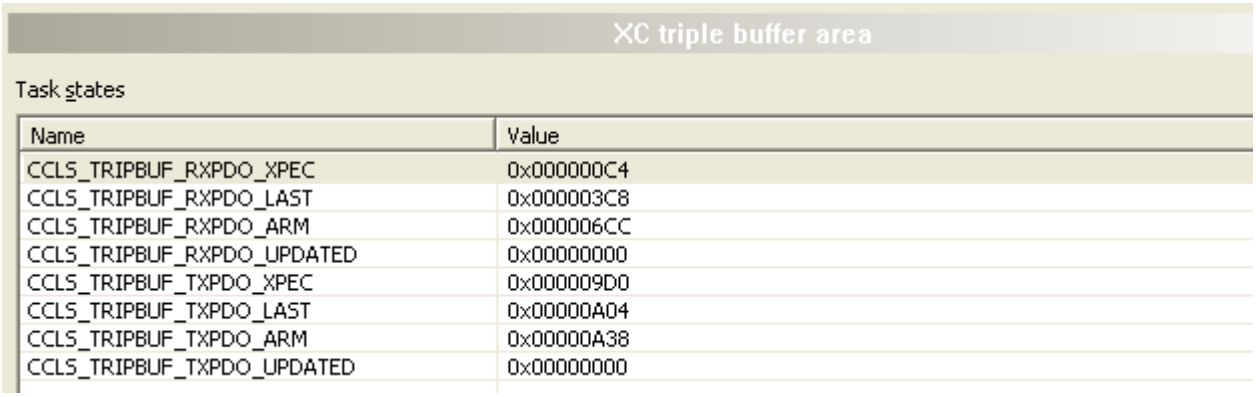


Figure 104: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > XC triple buffer area

ID	Value	Description
CCLS_TRIPBUF_RXPDO_XPEC	0 … 2^{32-1}	xPEC register of Tripple Buffer for RxPDO
CCLS_TRIPBUF_RXPDO_LAST	0 … 2^{32-1}	LAST register of Tripple Buffer for RxPDO
CCLS_TRIPBUF_RXPDO_ARM	0 … 2^{32-1}	ARM register of Tripple Buffer for RxPDO
CCLS_TRIPBUF_RXPDO_UPDATED	0 … 2^{32-1}	Updated register of Tripple Buffer for RxPDO
CCLS_TRIPBUF_TXPDO_XPEC	0 … 2^{32-1}	xPEC register of Tripple Buffer for TxPDO
CCLS_TRIPBUF_TXPDO_LAST	0 … 2^{32-1}	LAST register of Tripple Buffer for TxPDO
CCLS_TRIPBUF_TXPDO_ARM	0 … 2^{32-1}	ARM register of Tripple Buffer for TxPDO
CCLS_TRIPBUF_TXPDO_UPDATED	0 … 2^{32-1}	Updated register of Tripple Buffer for TxPDO

Table 128: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > XC triple buffer area

8.3.4.10 CCLINK_SLAVE/Extended transmission diagnostic

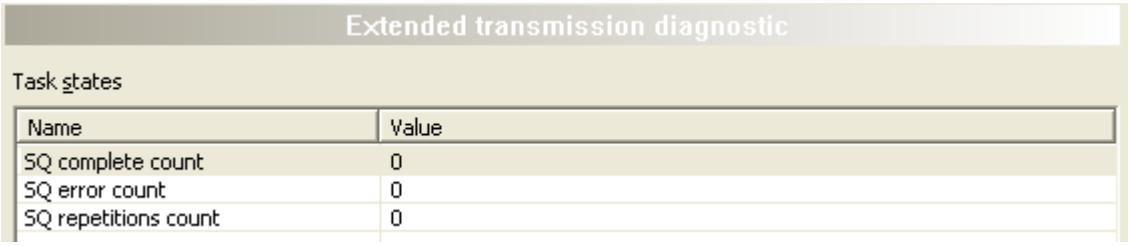


Figure 105: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > Extended transmission diagnostic

ID	Description
SQ complete count	Number of complete data transfers (SQ = sequence)
SQ error count	Number of detected sequence errors in the data transfers (SQ = sequence)
SQ repetitions count	Number of repeated data sequences (SQ = sequence)

Table 129: Extended Diagnosis CC-Link Slave > CCLINK_SLAVE > Extended transmission diagnostic

8.3.4.11 CCLINK_APS/Slave configuration

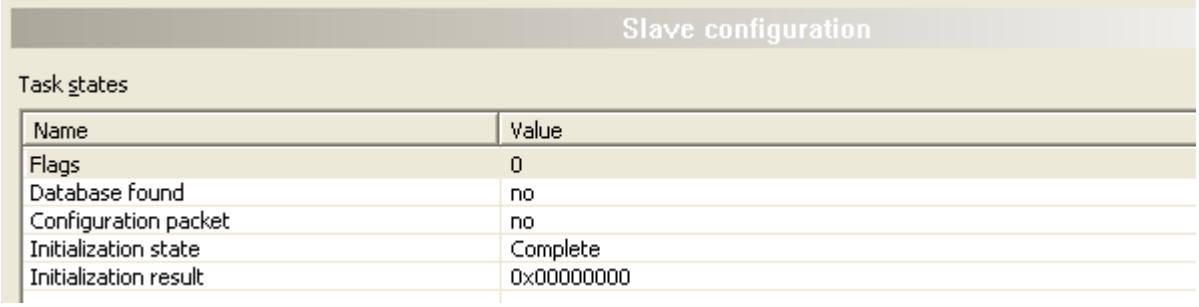


Figure 106: Extended Diagnosis CC-Link Slave > CCLINK_APS > Slave Configuration

Name	Description
Flags	Indication of flags
Database found	Indicates if the configuration database has been found (yes) or not (no)
Configuration packet	Indicates if the configuration shall be done with a configuration packet (yes) or not (no)
Initialization state	- Idle - Send initialize request - Wait for initialize confirmation - Send register request - Wait for register confirmation - Send get buffer request - Wait for get buffer confirmation - Send bus parameter request - Wait for bus parameter confirmation - Complete - Failed
Initialization result	Status code for error-free initializing = 0x00000000, otherwise error code

Table 130: Extended Diagnosis CC-Link Slave > CCLINK_APS > Slave Configuration

8.3.4.12 CCLINK_APS/Commands

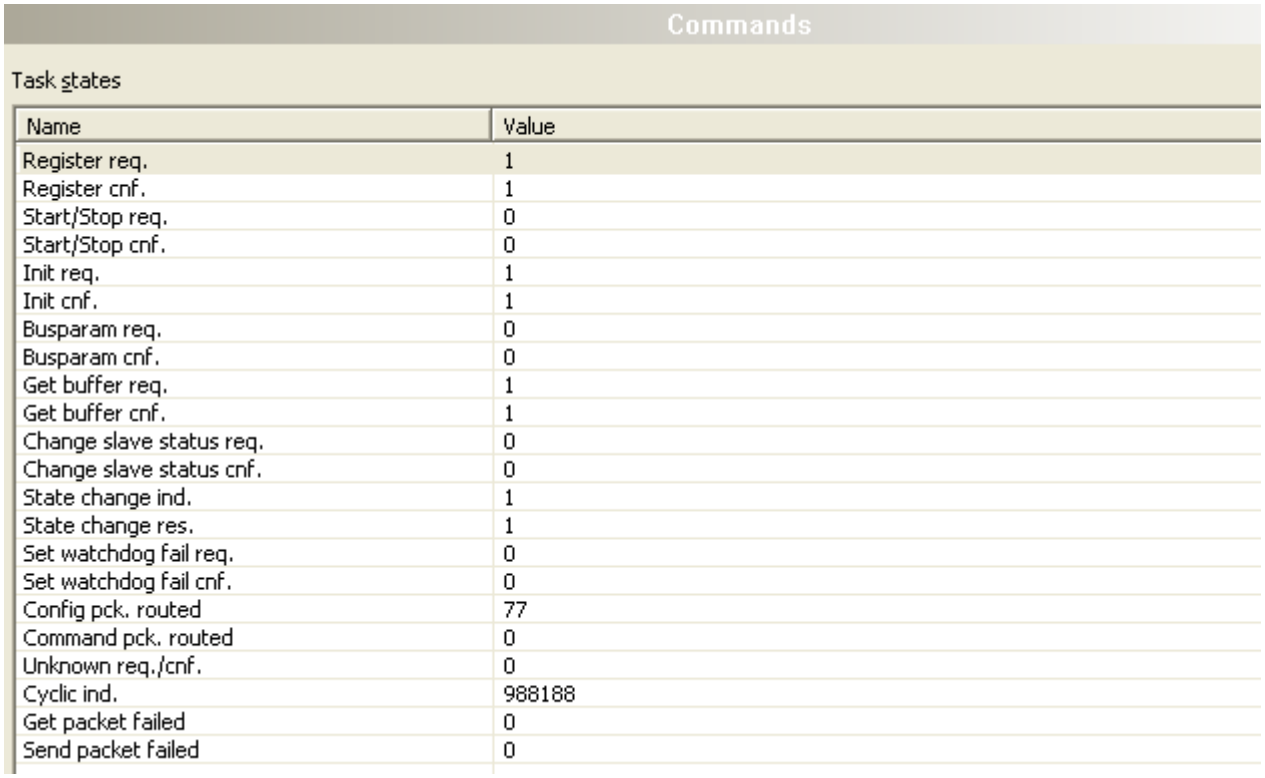


Figure 107: Extended Diagnosis CC-Link Slave > CCLINK_APS > Commands

Name	Description
[*Service*]	Diagnosis counter: Indicates the services processed. (The services of the single packets are described in the CC-Link Slave Protocol API manual [7].)
Get packet failed	Number of errors during requesting a packet
Send packet failed	Number of errors during sending a packet

Table 131: Extended Diagnosis CC-Link Slave > CCLINK_APS > Commands

8.3.4.13 CCLINK_APS/DPM data exchange

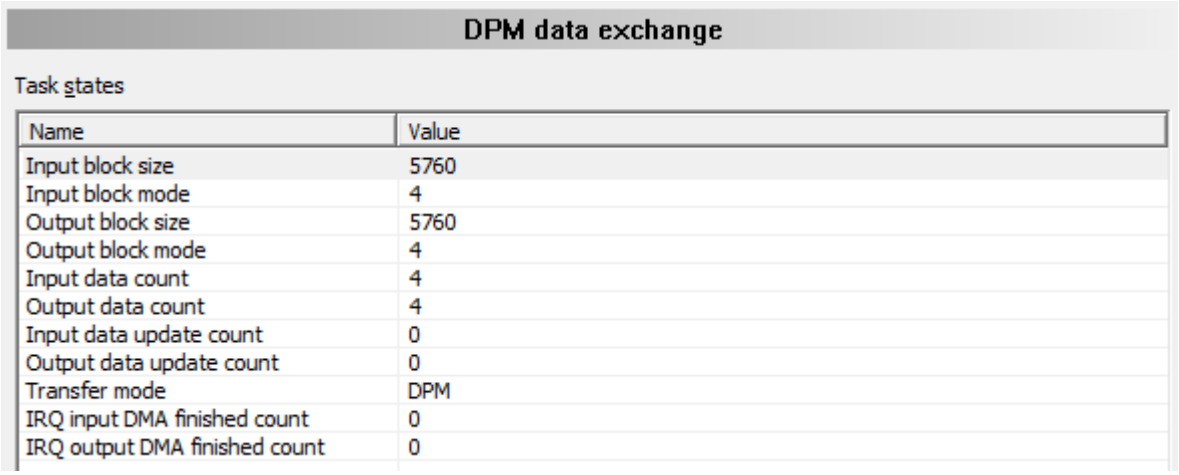


Figure 108: Extended Diagnosis CC-Link Slave > CCLINK_APS > DPM data exchange

Name	Description
Input block size	Size of the Input Data Image (cyclic data from the network) The default size of the input data image is 5760 byte.
Input block mode	Handshake mode of the input data block supported by the protocol stack, to synchronize process data exchange with the host application. 0x00 For compatibility reasons, This value is identical to 0x04 - Buffered Host Controlled IO Data Transfer 0x02 Buffered Device Controlled IO Data Transfer 0x03 Uncontrolled Mode 0x04 Buffered Host Controlled IO Data Transfer
Output block size	Size of the Output Data Image (cyclic data to the network) The default size of the output data image is 5760 byte.
Output block mode	Handshake mode of the output data blocks, see Input block mode
Input data count	Current Number of input data in bytes
Output data count	Current Number of output data in bytes
Input data update count	The total data count of input data in bytes (Counter for updating the input data)
Output data update count	The total data count of input data in bytes (Counter for updating the output data)
Transfer mode	DPM (Dual-Port Memory), DMA (Direct Memory Access)
IRQ input DMA finished count	The data count of IRQ input data in bytes when DMA is finished.
IRQ output DMA finished count	The data count of IRQ output data in bytes when DMA is finished.

Table 132: Extended Diagnosis CC-Link Slave > CCLINK_APS > DPM data exchange

8.3.4.14 CCLINK_APS/Rotary Switch

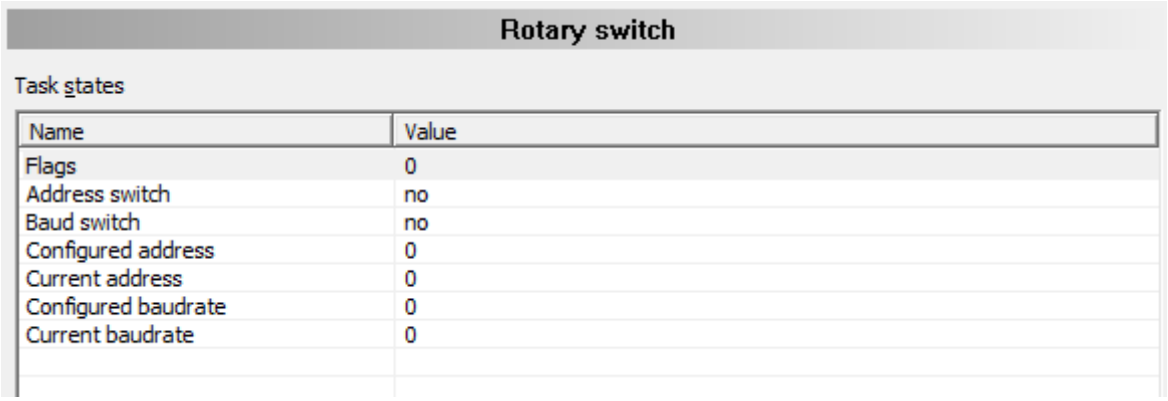


Figure 109: Extended Diagnosis CC-Link Slave > CCLINK_APS > Rotary Switch

Name	Description
[*Service*]	Diagnosis counter: Indicates the services processed. (The services of the single packets are described in the CC-Link Slave Protocol API manual [7].)
Address switch	“yes” means: The station address is configured at the address switch, (only for COMX 10XX-CCS/CCS). “no” means: The station address is configured in the configuration software.
Baud switch	“yes” means: The baudrate is configured at the baudrate switch, (only for COMX 10XX-CCS/CCS). “no” means: The baudrate is configured in the configuration software.

Table 133: Extended Diagnosis CC-Link Slave > CCLINK_APS > Rotary Switch

8.3.5 CompoNet Slave

8.3.5.1 Overview CompoNet Slave

Dialog Windows "Extended Diagnosis" CompoNet Slave

The table below gives an overview for the **Extended Diagnosis** dialog windows descriptions:

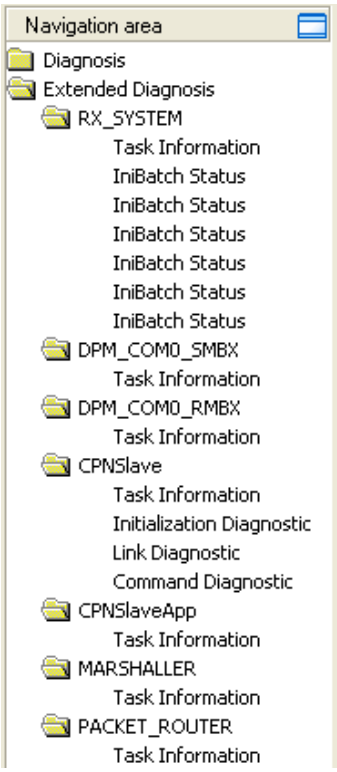
Navigation Area CompoNet Slave*	Folder Name in the Navigation Area	Dialog Window	Manual Page
	*CPNSlave*	*CPNSlave/Initialization Diagnostic*	*205*
		CPNSlave/Link Diagnostic	*206*
		CPNSlave/Command Diagnostic	*207*
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions*.	215
		** The displayed CompoNet Slave exteded diagnosis corresponds to the CompoNet Slave firmware version 1.0.x.x.*	

Table 134: Descriptions of the Dialog Windows Extended Diagnosis CompoNet Slave

8.3.5.2 CPNSlave/Initialization Diagnostic

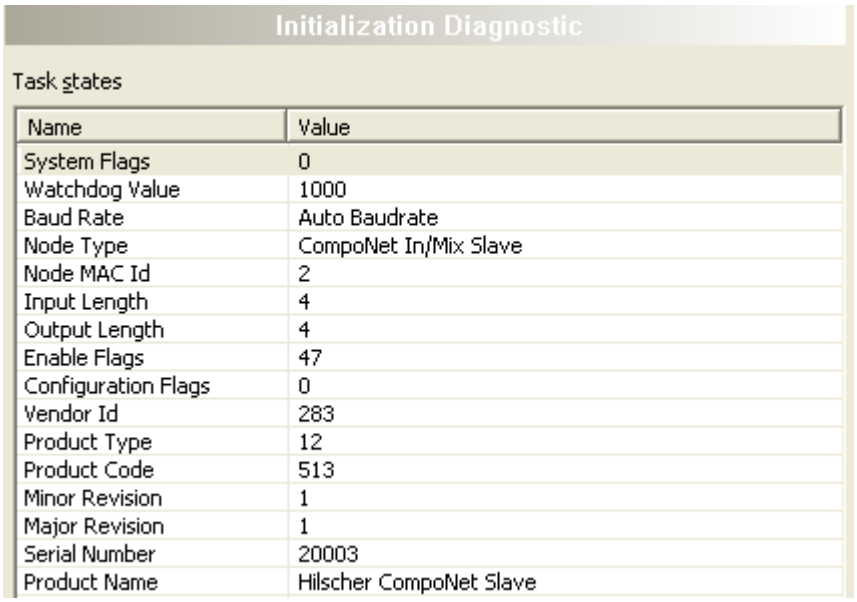
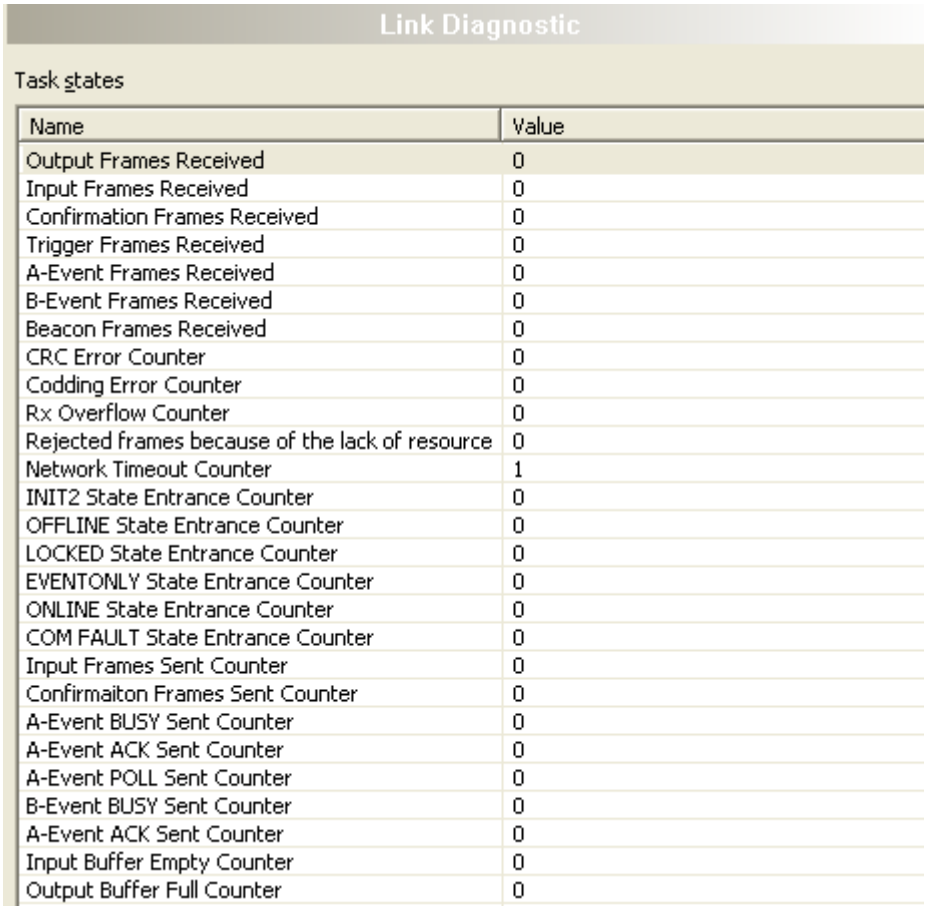


Figure 110: Extended Diagnosis CompoNet Slave > CPNSlave > Initialization Diagnostic

Name	Description
System Flags	Display of the current value of the system flags. Currently always zero (0)
Watchdog Value	The set value of the watchdog time. The value zero (0) means the watchdog must not be activated. A value greater than zero (0) indicates the watchdog time.
Baudrate	Baud rate of the CompoNet connection
Node Type	Node type of the CompoNet Slave
Node MAC Id	The MAC ID defines the CompoNet address of the device within the network
Input Length	Input length: number of input bits, value range: 2 ... 256 bits
Output Length	Initial length: number of output bits, value range: 2 ... 256 bits
Enable Flags	Number of enabled flags, which will use a default value (default)
Configuration Flags	Number of configuration flags, which must be configured
Vendor ID	Identification number of the manufacturer
Product Type	Communication Adapter
Product Code	Product code of the device, according to manufacturer's instructions
Minor Revision	Minor Revision of the task
Major Revision	Major Revision of the task
Serial Number	Serial number of the device
Product Name	The variable product name is a string (text string), which is a short description of the product / product family.

Table 135: Extended Diagnosis CompoNet Slave > CPNSlave > Initialization Diagnostic

8.3.5.3 CPNSlave/Link Diagnostic



Name	Value
Output Frames Received	0
Input Frames Received	0
Confirmation Frames Received	0
Trigger Frames Received	0
A-Event Frames Received	0
B-Event Frames Received	0
Beacon Frames Received	0
CRC Error Counter	0
Codding Error Counter	0
Rx Overflow Counter	0
Rejected frames because of the lack of resource	0
Network Timeout Counter	1
INIT2 State Entrance Counter	0
OFFLINE State Entrance Counter	0
LOCKED State Entrance Counter	0
EVENTONLY State Entrance Counter	0
ONLINE State Entrance Counter	0
COM FAULT State Entrance Counter	0
Input Frames Sent Counter	0
Confirmaiton Frames Sent Counter	0
A-Event BUSY Sent Counter	0
A-Event ACK Sent Counter	0
A-Event POLL Sent Counter	0
B-Event BUSY Sent Counter	0
A-Event ACK Sent Counter	0
Input Buffer Empty Counter	0
Output Buffer Full Counter	0

Figure 111: Extended Diagnosis CompoNet Slave > CPNSlave > Link Diagnostic

Name	Description
[*Service*]	Diagnosis counter of the Link Diagnostic layer. Indicates the services processed. (The services of the single packets are described in the CompoNet Slave Protocol API manual [12] [12].)

Table 136: Extended Diagnosis CompoNet Slave > CPNSlave > Link Diagnostic

8.3.5.4 CPNSlave/Command Diagnostic

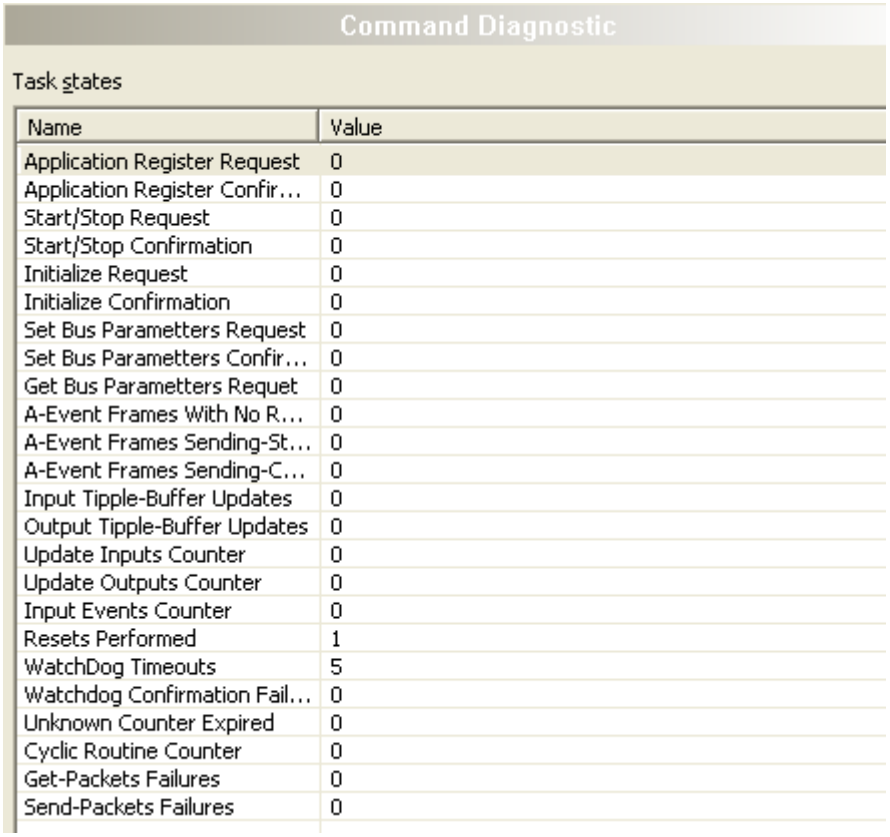


Figure 112: Extended Diagnosis CompoNet Slave > CPNSlave > Command Diagnostic

Name	Description
[*Service*]	Diagnosis counter of the Command Diagnostic layer. Indicates the services processed. (The services of the single packets are described in the CompoNet Slave Protocol API manual [12] [12].)
Get-Packet Failures	Number of errors during requesting a packet
Send Packet Failures	Number of errors during sending a packet

Table 137: Extended Diagnosis CompoNet Slave > CPNSlave > Command Diagnostic

8.4 Serial Protocols

8.4.1 3964R Slave

8.4.1.1 Overview 3964R Slave

Dialog Windows “Extended Diagnosis” R3964 Slave

The table below gives an overview for the **Extended Diagnosis** dialog windows descriptions:

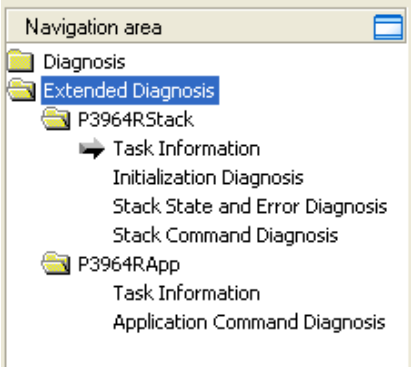
Navigation Area 3964R	Folder Name in the Navigation Area	Dialog Window	Manual Page
	*P3964RStack*	*P3964R Stack/Initialization Diagnosis*	209
		P3964R Stack/Stack State and Error Diagnosis	210
		P3964R Stack/Stack Command Diagnosis	212
	P3964RStack	*P3964R App/Application Command Diagnosis*	213
		For the description of further tasks refer to section *Descriptions for Tasks with similar Functions* on page 215.	215

Table 138: Descriptions of the Dialog Windows Extended Diagnosis R3964 Slave

8.4.1.2 P3964R Stack/Initialization Diagnosis

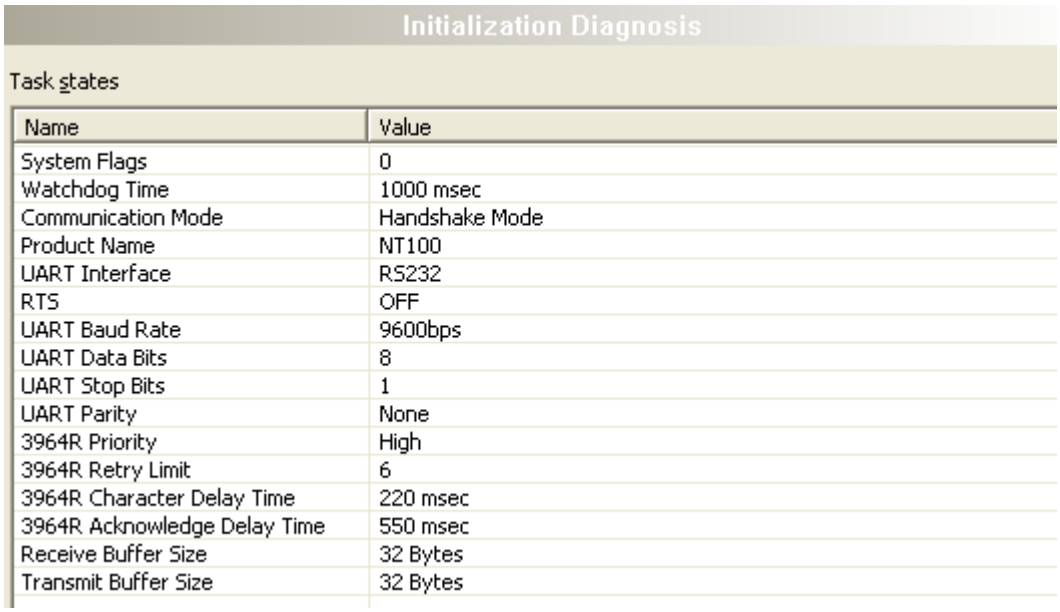


Figure 113: Extended Diagnosis R3964 Slave > P3964R Stack > Initialization Diagnosis

Name	Description
System Flags	Shows the current value of the System Flags. currently always zero (0).
Watchdogtime	Shows the configured watchdogtime. A value of zero (0) means, that no monitoring (no watchdog) was activated. A value greater than zero (0) shows the watchdogtime.
Communication Mode	Shows the configured Communication Mode. Possible value(s): Handshake Mode
Product Name	Shows the configured product name
UART Interface	Shows the configured interface type. Possible values: RS232, RS422, RS485.
RTS	Shows the configured setting for RTS control. Possible values: On, Off
UART Baud Rate	Shows the configured baud rate
UART Data Bits	Shows the configured value of data bits
UART Stop Bits	Shows the configured value of stop bits
UART Parity	Shows the configured parity. Possible values: None, even, odd
3964R Priority	Shows the configured priority. Possible values: High, low
3964R Retry Limit	Shows the configured value for retry limit
3964R Character Delay Time	Shows the configured value for character delay time
3964R Acknowledge Delay Time	Shows the configured value of acknowledge delay time
Receive Buffer Size	Shows the configured size of the receive buffer
Transmit Buffer Size	Shows the configured size of the send buffer

Table 139: Extended Diagnosis R3964 Slave > P3964R Stack > Initialization Diagnosis

8.4.1.3 P3964R Stack/Stack State and Error Diagnosis

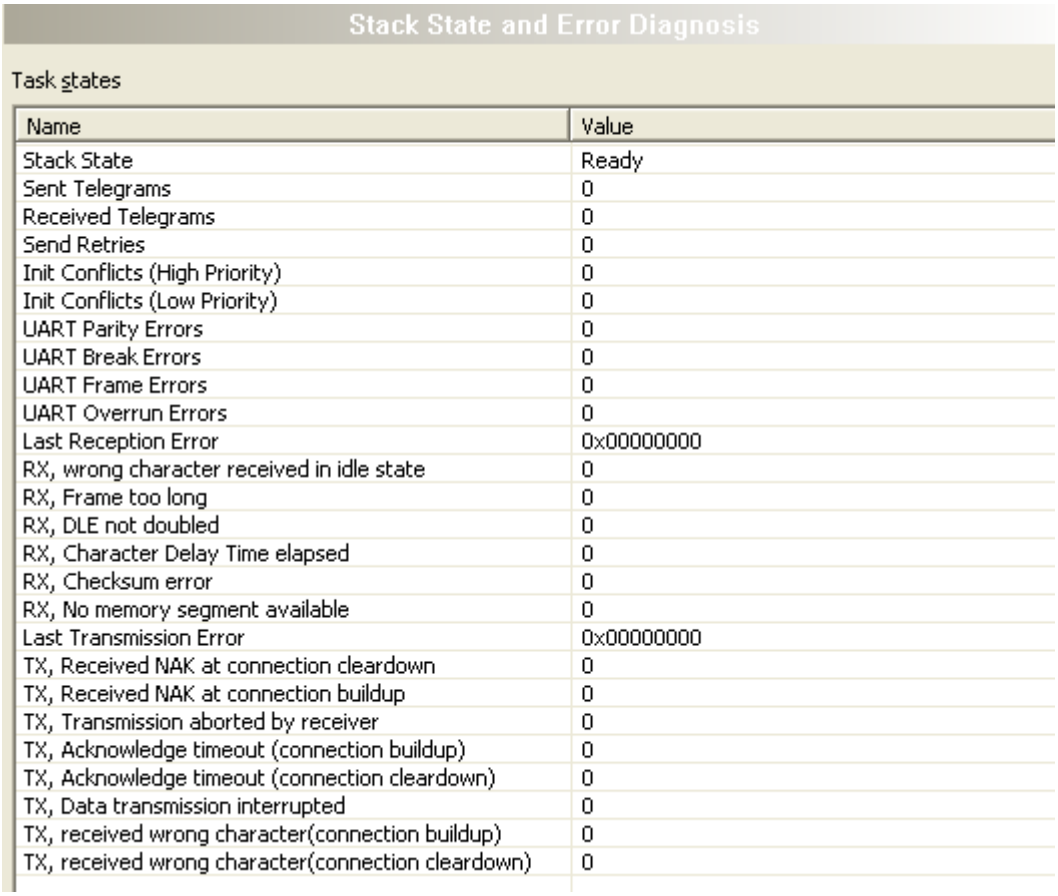


Figure 114: Extended Diagnosis R3964 Slave > P3964R Stack > Stack State and Error Diagnosis

Name	Description
Stack State	Shows the current state of the stack. Possible values: Not initialized, Ready, Stopped, TX, Sent STX, wait for DLE TX, Sent Data, wait for DLE TX, Send data (no DLE) TX, Send data (doubling DLE) TX, Send ETX TX, Send BCC RX, Sent DLE, wait for data RX, Wait for 2nd DLE or ETX RX, Wait for BCC Wait for memory segment Wait for character timeout and send NAK
Sent Telegrams	Number of telegrams sent
Received Telegrams	Number of telegrams received
Send Retries	Number of Send Retries
Init Conflicts (High Priority)	Number of 'How many Initialization Conflicts occured at high priority'
Init Conflicts (Low Priority)	Number of 'How many Initialization Conflicts occured at low priority
UART Parity Errors	Number of parity errors
UART Break Errors	Number of break errors

Name	Description
UART Frame Errors	Number of frame errors
UART Overrun Errors	Number of overrun errors
Last Reception Error	Last occurred reception error
RX, wrong character received in idle state	Number of wrong received character in idle state
RX, Frame too long	Number of frames, which were too long
RX, DLE not doubled	Number of DLE not doubled (second DLE was not received)
RX, Character Delay Time elapsed	Number of elapsed character delay time
RX, Checksum error	Number of checksum errors occured while receiving
RX, No memory segment available	Number of 'no receive buffer available' while receiving
Last Transmission Error	Last occurred send error
TX, Received NAK at connection cleardown	Number of received NAK at connection cleardown
TX, Received NAK at connection buildup	Number of received NAK at connection buildup
TX, Transmission aborted by receiver	Number of transmission aborted by receiver
TX, Acknowledge timeout (connection buildup)	Number of acknowledge timeout during connection buildup
TX, Acknowledge timeout (connection cleardown)	Number of acknowledge timeout during connection cleardown
TX, Data transmission interrupted	Number of interrupted data transmission
TX, received wrong character (connection buildup)	Number of wrong received character during connection buildup
TX, received wrong character (connection cleardown)	Number of wrong received character during connection cleardown

Table 140: Extended Diagnosis R3964 Slave > P3964R Stack > Stack State and Error Diagnosis

8.4.1.4 P3964R Stack/Stack Command Diagnosis

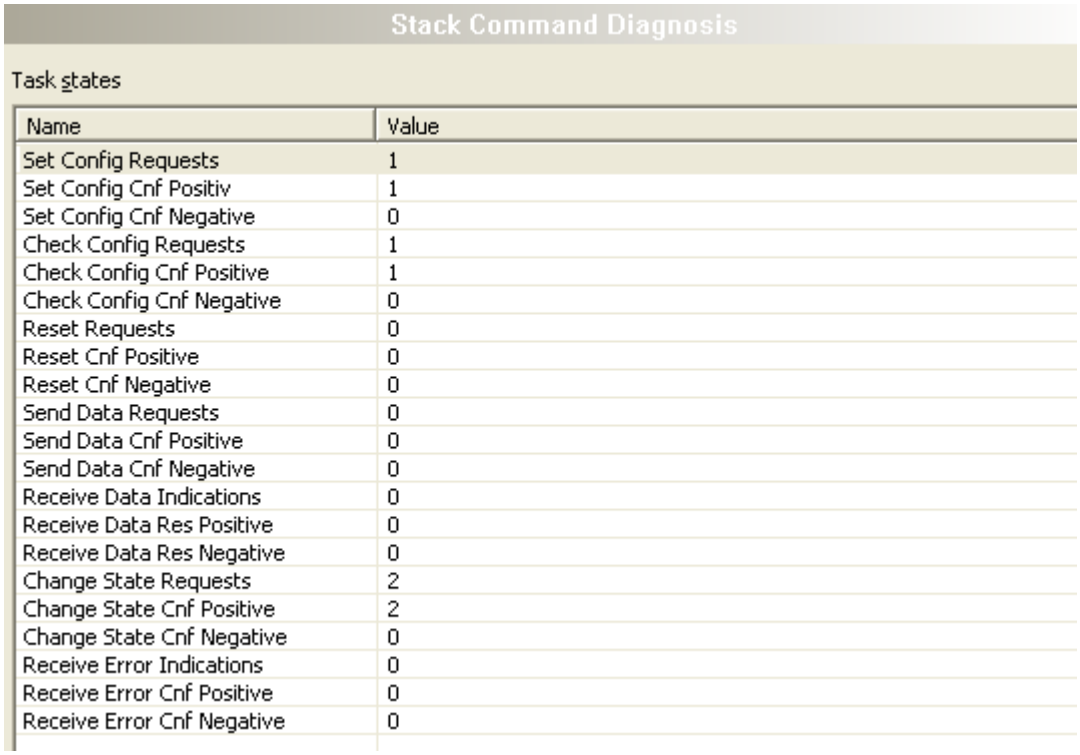


Figure 115: Extended Diagnosis R3964 Slave > P3964R Stack > Stack Command Diagnosis

Name	Description
Set Config Requests	Number of received 'Set Config Requests' packets
Set Config Cnf Positiv	Number of send 'Set Config Confirmation Positiv' packets
Set Config Cnf Negative	Number of send 'Set Config Confirmation Negative' packets
Check Config Requests	Number of received 'Check Config Requests' packets
Check Config Cnf Positiv	Number of send 'Check Config Confirmation Positiv' packets
Check Config Cnf Negative	Number of send 'Check Config Confirmation Negative' packets
Reset Requests	Number of received 'Reset Requests' packets
Reset Cnf Positiv	Number of send 'Reset Confirmation Positiv' packets
Reset Cnf Negative	Number of send 'Reset Confirmation Negative' packets
Send Data Requests	Number of received 'Send Data Requests' packets
Send Data Cnf Positiv	Number of send 'Send Data Confirmation Positiv' packets
Send Data Cnf Negative	Number of send 'Send Data Confirmation Negative' packets
Receive Data Indications	Number of send 'Receive Requests' packets
Receive Data Res Positiv	Number of received 'Receive Confirmation Positiv' packets
Receive Data Res Negative	Number of received 'Receive Confirmation Negative' packets
Change State Requests	Number of received 'Change Requests' packets
Change State Cnf Positiv	Number of send 'Change State Confirmation Positiv' packets
Change State Cnf Negative	Number of send 'Change State Confirmation Negative' packets
Receive Error Indications	Number of send 'Receive Indications' packets
Receive Error Res Positiv	Number of received ' Receive Error Response Positiv' packets
Receive Error Res Negative	Number of received 'Receive Error Response Negative' packets

Table 141: Extended Diagnosis R3964 Slave > P3964R Stack > Stack Command Diagnosis

8.4.1.5 P3964R App/Application Command Diagnosis

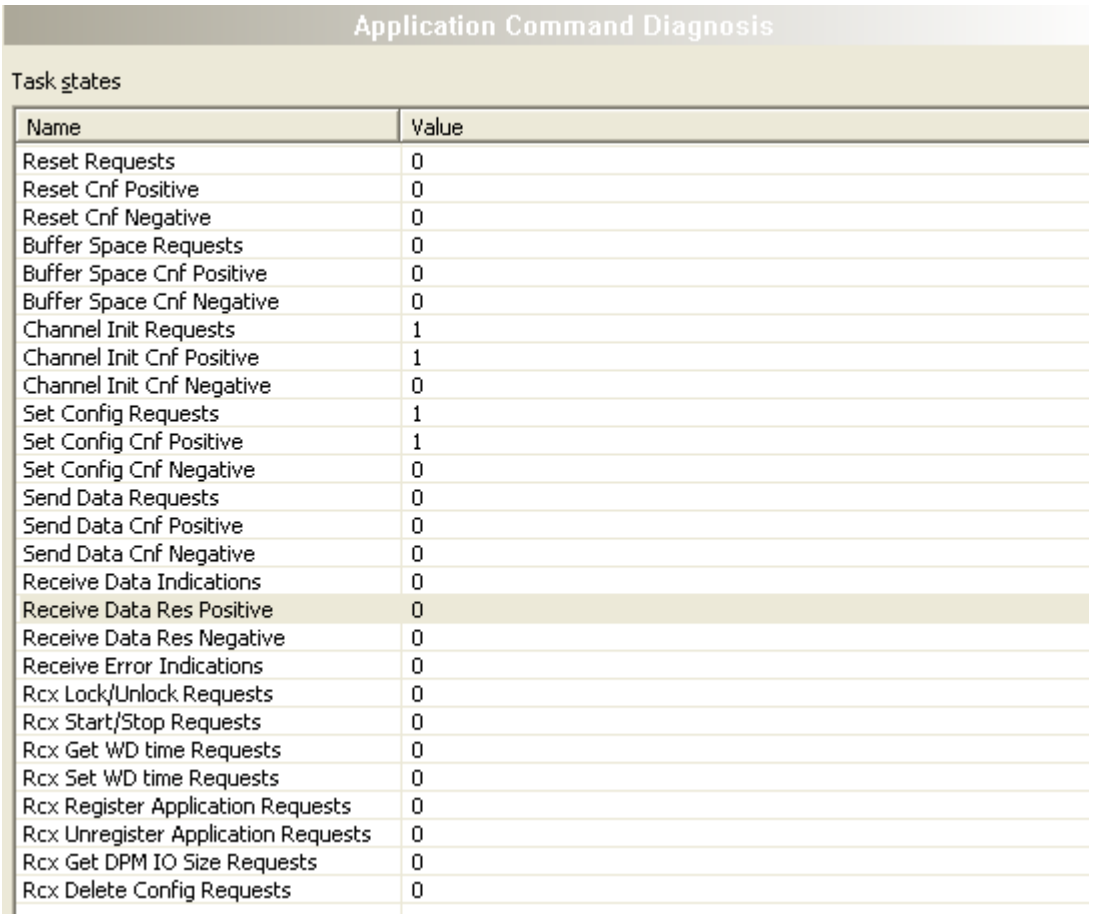


Figure 116: Extended Diagnosis R3964 Slave > P3964R App > Application Command Diagnosis

Name	Description
Reset Requests	Number of received 'Reset Requests' packets
Reset Cnf Positiv	Number of send 'Reset Confirmation Positiv' packets
Reset Cnf Negative	Number of send 'Reset Confirmation Negative' packets
Buffer Space Requests	Number of send 'Buffer Space Requests' packets
Buffer Space Cnf Positiv	Number of send 'Buffer Space Confirmation Positiv' packets
Buffer Space Cnf Negative	Number of received 'Buffer Space Confirmation Negative' packets
Channel Init Requests	Number of received 'Channel Init Requests' packets
Channel Init Cnf Positiv	Number of send 'Channel Init Confirmation Positiv' packets
Channel Init Cnf Negative	Number of send 'Channel Init Confirmation Negative' packets
Set Config Requests	Number of received 'Set Config Requests' packets
Set Config Cnf Positiv	Number of send 'Set Config Confirmation Positiv' packets
Set Config Cnf Negative	Number of send 'Set Config Confirmation Negative' packets
Send Data Requests	Number of send 'Send Data Requests' packets
Send Data Cnf Positiv	Number of received 'Send Data Confirmation Positiv' packets
Send Data Cnf Negative	Number of received 'Send Data Confirmation Negative' packets
Receive Data Indications	Number of received 'Receive Data Indications' packets
Receive Data Res Positiv	Number of send 'Receive Data Response Positiv' packets
Receive Data Res Negative	Number of send 'Receive Data Response Negative' packets
Receive Error Indications	Number of received 'Receive Error Indications' packets

Name	Description
Rcx Lock/Unlock Requests	Number of received 'Rcx Lock/Unlock Requests' packets
Rcx Start/Stop Requests	Number of received 'Rcx Start/Stop Requests' packets
Rcx Get WD time Requests	Number of received 'Rcx Get WD time Requests' packets
Rcx Set WD time Requests	Number of received 'Rcx Set WD time Requests' packets
Rcx Register Application Requests	Number of received 'Rcx Register Application Requests' packets
Rcx Unregister Application Requests	Number of received 'Rcx Unregister Application Requests' packets
Rcx Get DPM IO Size Requests	Number of received 'Rcx Get DPM IO Size Requests' packets
Rcx Delete Config Requests	Number of received 'Rcx Delete Config Requests' packets

Table 142: Extended Diagnosis R3964 Slave > P3964R App > Application Command Diagnosis

8.5 Descriptions for Tasks with similar Functions

8.5.1 Overview on the Descriptions

The following table provides an overview on the descriptions of the dialog windows **Extended Diagnosis** for tasks with similar functions

Task Group	Task or Description	Manual Page
Task Information	*Task Information*	215
IniBatch Status	*IniBatch Status*	216
General Diagnosis Information	*General Diagnosis Information*	217
Code Diagnosis	*Code Diagnosis*	218
TCPUDP	*IP Information*	219
	IP Packet Counter	220
	IP Code Diagnosis	221
	TCP_UDP Information	222
	TCP_UDP Code Diagnosis	223

Table 143: Descriptions of the Dialog Windows Extended Diagnosis for Tasks with similar Functions

8.5.2 Descriptions

8.5.2.1 Task Information

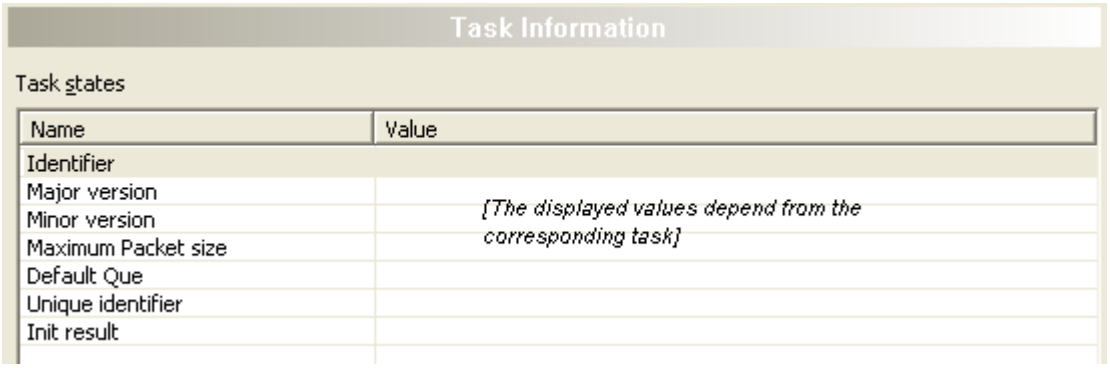


Figure 117: Extended Diagnosis > [Folder Name] > Task Information, Example

Name	Description
Identifier	Identification number of the task
Major version	Task version, contains incompatible changes
Minor version	Task version, contains compatible changes
Maximum package size	Maximum package size, which the task sends
Default Queue	Queue handle, which is accessible via DPM by mailbox.
UUID	Unique user ID, 16 Byte indicator used for task identification and its affiliation e. g. to a stack (therein different identification data are coded in).
Init result	Error Code, 0= no Error The description of the error codes can be found in this manual or in the corresponding software reference manuals.

Table 144: Extended Diagnosis > [Folder Name] > Task Information

8.5.2.2 IniBatch Status

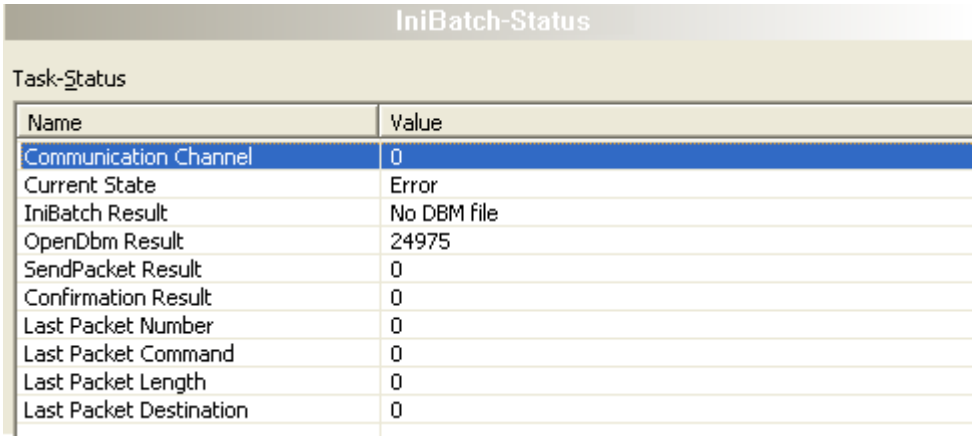


Figure 118: Extended Diagnosis > [Folder Name] > IniBatch Status, Example

Name	Description
Communication Channel	Number of the communication channel used by the device.
Current State	Idle; IniBatch packets in progress; Retrying to send last packet; Error
IniBatch Result	Ok; No DBM file; No Packet table; No data set available; Data set is shorter than packet length; Packet Buffer is shorter than Packet length; Invalid packet destination; Logical queue not defined Send packet failed; Too many retries; Error in confirmation packet status
OpenDbm Result	Error when opening the IniBatch data base Under "OpenDbm Result" the error code is typed in, when "IniBatch Result" == "No DBM file" (1) is.
SendPacket Result	Error when sending a packet Under "SendPacket Result" the error code is typed in, when "IniBatch Result" == “send packet failed" (8) is.
Confirmation Result	Confirmation error when sending packets Under "Confirmation Result" the package specific error code from the ulSta is typed in, when "IniBatch Result " == "Error in confirmation packet status" (10) is.
Last Packet Number	Value depends by the communication system
Last Packet Command	Value depends by the communication system
Last Packet Length	Value depends by the communication system
Last Packet Destination	Value depends by the communication system

Table 145: Extended Diagnosis > [Folder Name] > IniBatch Status, Example Display

The task status "Confirmation Result" is bus specific. The other task statuses are rcx-related error codes.

8.5.2.3 General Diagnosis Information

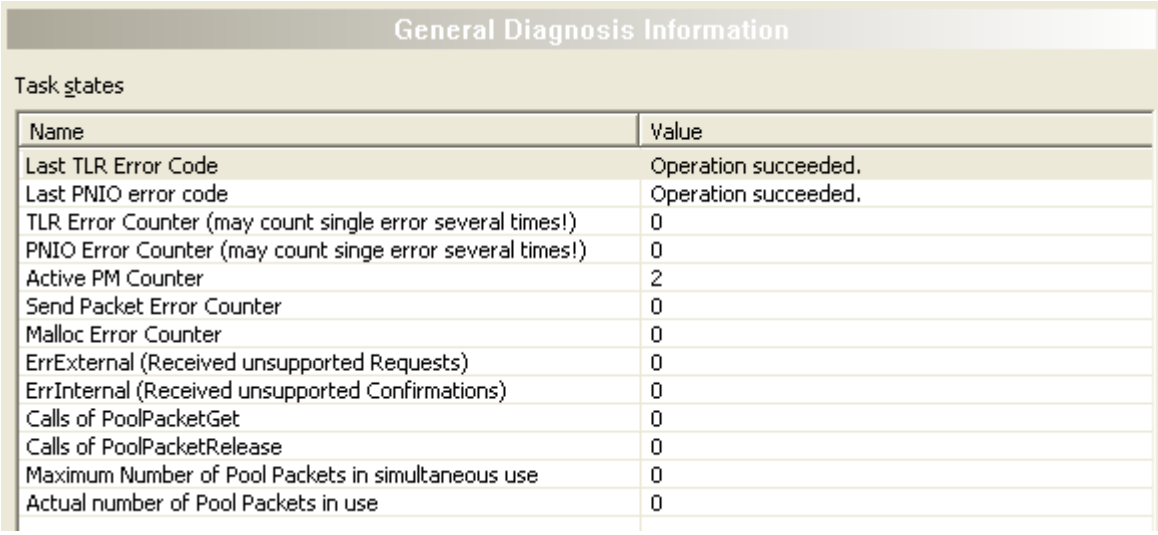


Figure 119: Extended Diagnosis > [Folder Name] > General Diagnosis Information

Name	Description
Last TLR error code	Error code of the last internally occurred error
Last PNIO error code	Error code of the last externally occurred error which has been reported by an IO Device.
TLR Error Counter (may count single error several times!)	Counts the number of occurred TLR error codes.
PNIO Error Counter (may count single error several times!)	Counts the number of occurred PNIO error codes.
Active PM Counter	Counter of the active protocol machines in the task.
Send Packet Error Counter	Counts how often a task sends a packet to another task and this fails.
Malloc Error Counter	Counts how often storage capacity is requested in the operating system and the operating system cannot follow this demand.
ErrExternal (Received unsupported Requests)	Counts how often an unknown request packet was received.
ErrInternal (Received unsupported Confirmations)	Counts how often an unknown confirmation packet was received.
Calls of PoolPacketGet*	Counts how many packets are retrieved from the pool.
Calls of PoolPacketRelease*	Counts how many packets are given back to the pool.
Maximum Number of Pool Packets in simultaneous use*	Counts how many pool elements were used simultaneously as a maximum.
Actual number of Pool Packets in use*	Counts how many pool elements are used currently.
*Not used for the diagnosis task „T_RPC“.	

Table 146: Extended Diagnosis > [Folder Name] > General Diagnosis Information

8.5.2.4 Code Diagnosis

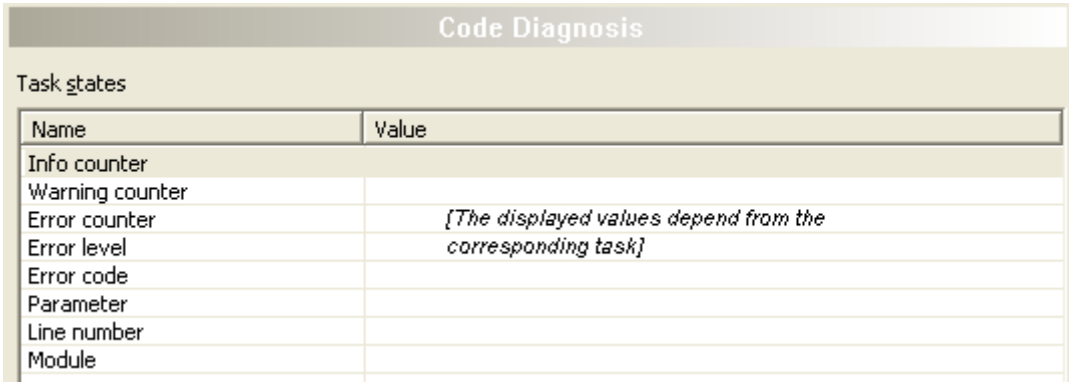


Figure 120: Extended Diagnosis > [Folder Name] > Code Diagnosis, Example

Name	Description
Info Counter	Counter for information reports
Warning Counter	Counter for warning reports
Error Counter	Counter for errors
Error Level	Level of the last occurred error
Error Code	Code of the last occurred error
Parameter	Additional information to the error
Line number	Line number inside the software module
Module	Software module

Table 147: Extended Diagnosis > [Folder Name] > Code Diagnosis

8.5.2.5 IP Information

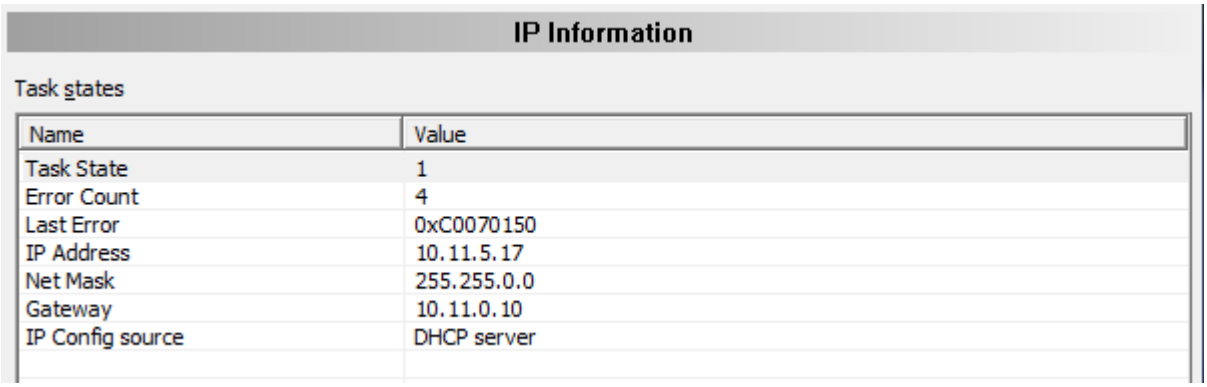


Figure 121: Extended Diagnosis > [Folder Name] > IP Information, Example

Name	Description
Task State	Actual state of the protocol process: 0 = Task not initialized 1 = Task is running 2 = Task initialized 3 = Initialization error
Error Counter	Counter for errors
Last Error	Last occurred error (Description see appropriate manual)
IP Address	IP address of the Slave station
Net Mask	Network mask of the Slave station
Gateway	Gateway address of the Slave station
IP Config source	Possible source: - None - DHCP server - BOOTP server - Database, Warmstart packe - ICMP (Ping) - Hilscher NetIdent Protocol

Table 148: Extended Diagnosis > [Folder Name] > IP Information

8.5.2.6 IP Packet Counter

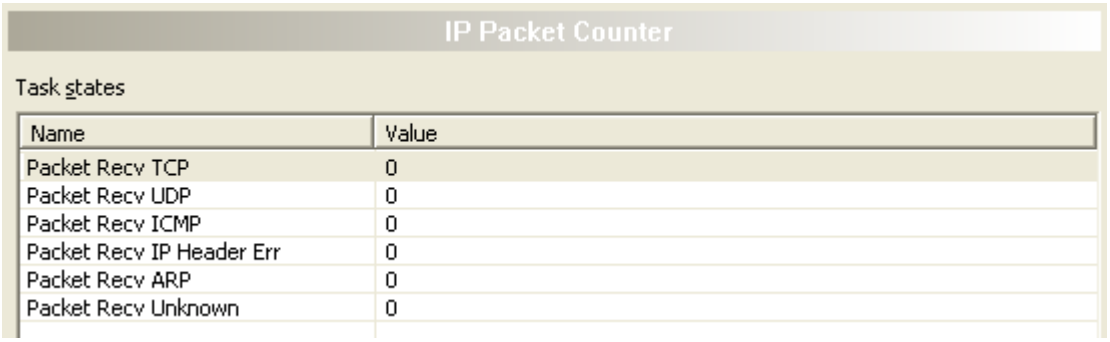


Figure 122: Extended Diagnosis > [Folder Name] > IP Packet Counter, Example

Name	Description
Packet Recv TCP	Counter for received TCP packets
Packet Recv UDP	Counter for received UDP packets
Packet Recv ICMP	Counter for received ICMP packets
Packet Recv IP Header Err	Counter for received IP packets with errors
Packet Recv ARP	Counter for received ARP packets
Packet Recv Unknown	Counter for received packets of an unknown type

Table 149: Extended Diagnosis > [Folder Name] > IP Packet Counter

8.5.2.7 IP Code Diagnosis

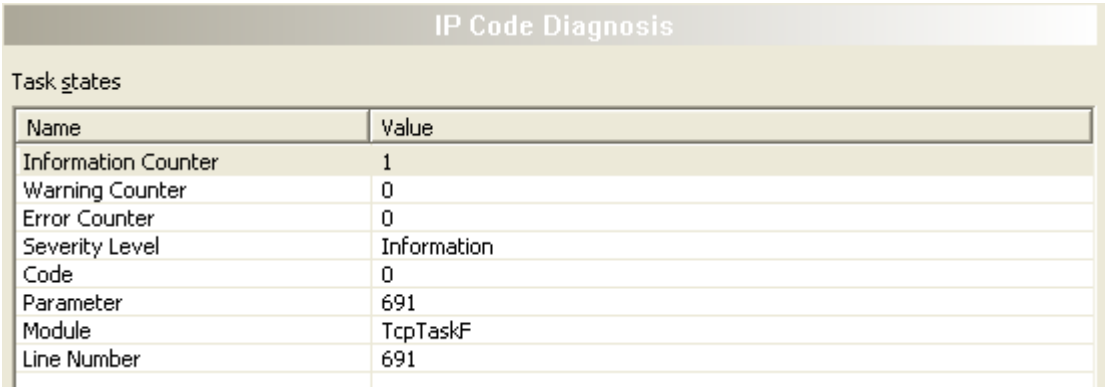


Figure 123: Extended Diagnosis > [Folder Name] > IP-Code Diagnosis, Example

Name	Description
Information Counter	Counter for information reports
Warning Counter	Counter for warning reports
Error Counter	Counter for errors
Severity Level	Level of the last occurred error
Code	Code of the last occurred error
Parameter	Additional information to the error
Module	Software module
Line number	Line number inside the software module

Table 150: Extended Diagnosis > [Folder Name] > IP-Code Diagnosis

8.5.2.8 TCP_UDP Information

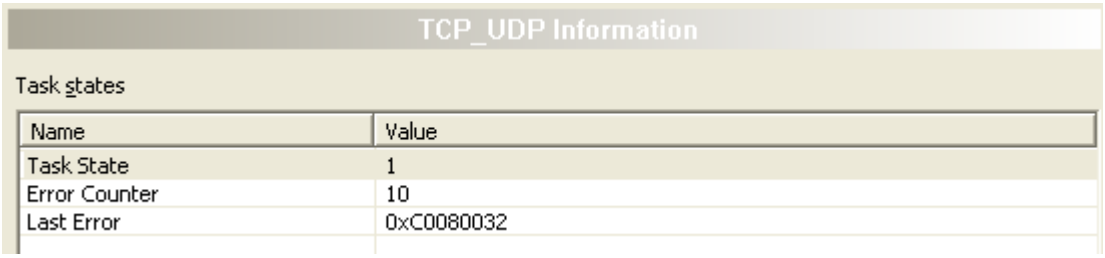


Figure 124: Extended Diagnosis > [Folder Name] > TCP_UDP-Information, Example

Name	Description
Task State	Actual state of the protocol process: 0 = Task not initialized 1 = Task is running 2 = Task initialized 3 = Initialization error
Error Count	Counter for errors
Last Error	Last occurred error (Description see appropriate manual)

Table 151: Extended Diagnosis > [Folder Name] > TCP_UDP-Information

8.5.2.9 TCP_UDP Code Diagnosis

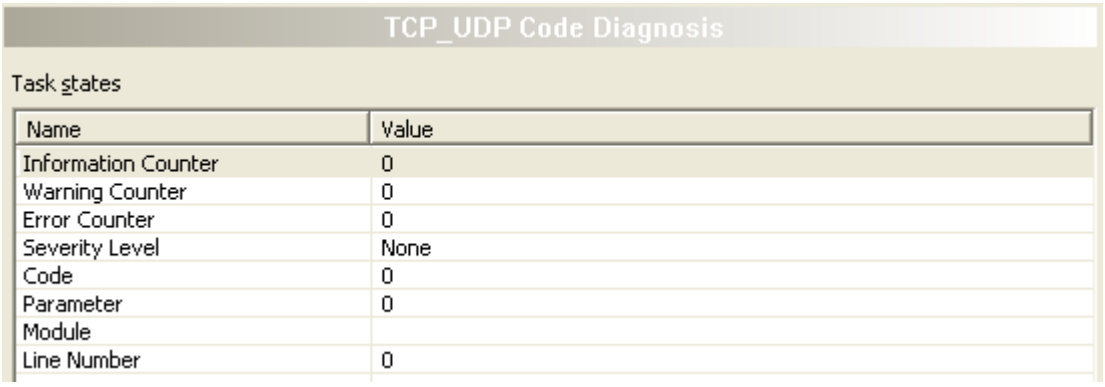


Figure 125: Extended Diagnosis > [Folder Name] > TCP_UDP Code Diagnosis Example, Display

Name	Description
Information Counter	Counter for information reports
Warning Counter	Counter for warning reports
Error Counter	Counter for errors
Severity Level	Level of the last occurred error
Code	Code of the last occurred error
Parameter	Additional information to the error
Module	Software module
Line number	Line number inside the software module

Table 152: Extended Diagnosis > [Folder Name] > TCP_UDP Code Diagnosis

9 Tools

9.1 Overview Tools

Under **Tools** the Packet Monitor and the IO Monitor are provided for test and diagnosis purposes.

Tools Windows

The table below gives an overview for the individual **Tools** dialog windows descriptions:

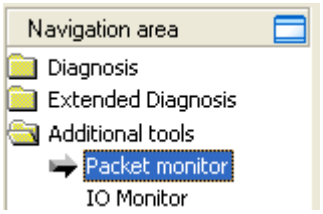
netSLAVE DTM	Folder Name / Section	Manual Page
 *Navigation Area - Tools*	*Packet Monitor*	*225*
	IO Monitor	*228*

Table 153: Descriptions of the Diagnosis Windows

Online Connection to the Device

Note: Accessing the **Tools** windows of the netSLAVE DTM requires an online connection from the netSLAVE DTM to the PROFIBUS DP Slave device. For further information, refer to section section *Connecting/Disconnecting Device* on page 118.

9.2 Packet Monitor

The **Packet Monitor** serves for test and diagnosis purposes.

Data packets, i. e. messages are self-contained blocks of defined data length. The packets are used to communicate with the firmware and they are exchanged between the application (configuration software) and the firmware in the device. Packets can be sent once or cyclically to the connected device controlled by the user and packets received can be displayed.

Data packets comprise from a **Packet Header** and the **Send Data** or from a **Packet Header** and the **Receive Data**. The packet header can be evaluated by the receiver of the packet and contain the sender and receiver address, the data length, an ID number, status and error messages and the command or response code. The minimum packet size amounts 40 Byte for the packet header. The sending and receiving data are added.

For further information to the packet description refer to the *Protocol API Manual*.

➢ Open the **Packet Monitor** via **Tools > Packet Monitor**.

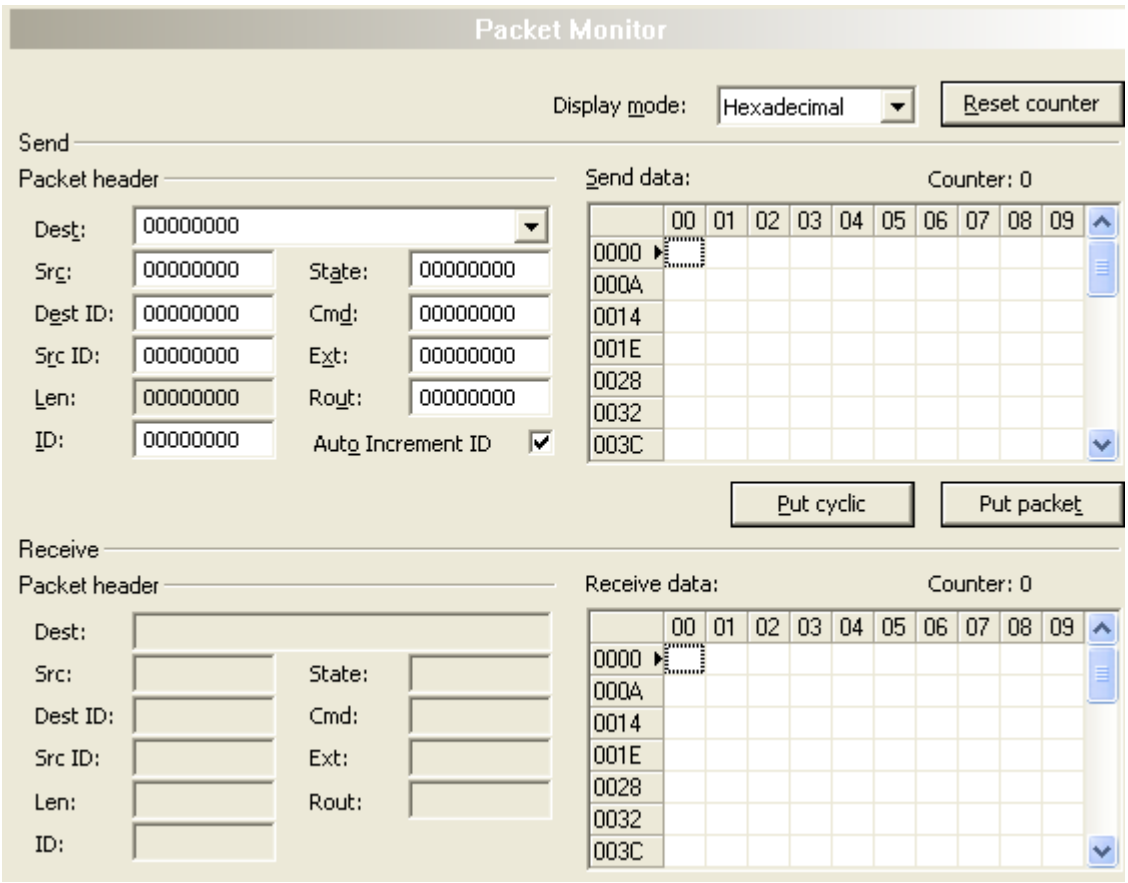


Figure 126: Packet Monitor

Display Mode switches the representation of the send and reception data between decimal and hexadecimal.

➢ Select **Reset Counter** to reset the packet counter.

9.2.1 Sending Packet

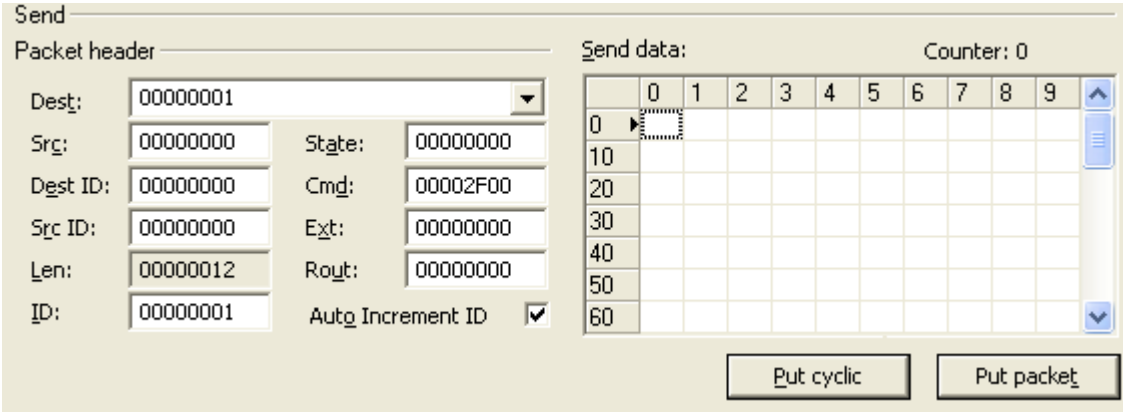


Figure 127: Send > Packet Header and Send Data

Packet Header

Under **Send > Packet Header** the elements of the packet header of the sending packet are displayed, which is transmitted from the application (configuration software) to the device. The packet header of the sending packets contain the elements described in the following table.

Element		Description
Dest	Destination Queue Handle	Contains the identifier of the receiver for the packet (destination task queue of the firmware).
Src	Source Queue Handle	Contains the identifier of the sender of the packet (sending task).
Dest ID	Destination Queue Reference	Contains an identifier for the receiver of unsolicited sent packets from the firmware to the application (configuration software).
Src ID	Source Queue Reference	Contains an identifier of the sender.
Len	Packet Data Length (in Bytes)	Length of the send respectively receive data.
ID	Packet Identification As Unique Number	Identifies identical data packets among each other.
State	Status / Error Code	Transmits status or error codes to the packet sender.
Cmd	Command / Response Code	Command or respond code.
Ext	Extension	Field for extensions (reserved).
Rout	Routing Information	Internal value of the firmware.

Table 154: Descriptions Packet Header

- Under **Dest** select the receiver (*destination task queue*).
- Under **Cmd** select the command identification *(Request)*.

Auto Increment ID is an increment for the identifier of the data packets and increments the ID by 1 for each newly sent packet.

Send Data

- Under **Send > Send data** enter the send data of the packet, which shall be transmitted from the application (configuration software) to the mailbox of the device. The meaning of the transmitted data depends on the command or response code.

Sending Packets once or cyclic

- To send packet once, select **Put packet**.
- To send packet cyclic, select **Put cyclic**.

9.2.2 Receiving Packet

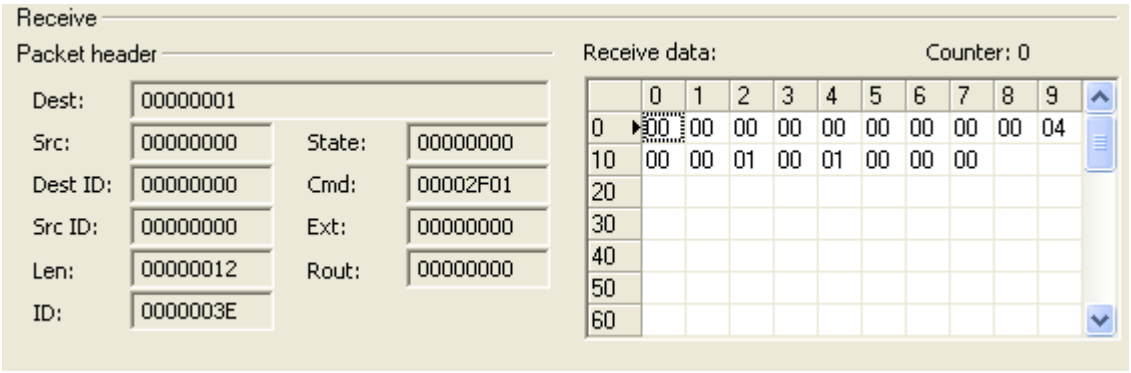


Figure 128: Packet Header and Receive Data

Packet Header

Under **Receive** > **Packet Header** the elements of the packet header of the receiving packet are displayed, which are transmitted back from the device to the application (configuration software). The packet header of the receiving packets contain the elements described in the following table.

Element		Description
Dest	Destination Queue Handle	Contains the identifier of the receiver for the packet (destination task queue of the firmware).
Src	Source Queue Handle	Contains the identifier of the sender of the packet (sending task).
Dest ID	Destination Queue Reference	Contains an identifier for the receiver of unsolicited sent packets from the firmware to the application (configuration software).
Src ID	Source Queue Reference	Contains an identifier of the sender.
Len	Packet Data Length (in Bytes)	Length of the send respectively receive data.
ID	Packet Identification As Unique Number	Identifies identical data packets among each other.
State	Status / Error Code	Transmits status or error codes to the packet sender.
Cmd	Command / Response Code	Command or respond code.
Ext	Extension	Field for extensions (reserved).
Rout	Routing Information	Internal value of the firmware.

Table 155: Descriptions Packet Header

Receive Data

Under **Receive** > **Receive Data** the receiving data of the packet, which is transmitted back from the device to the application (configuration software) are displayed.

9.3 IO Monitor

The **IO Monitor** serves for test and diagnosis purposes. It provides to view data of the process data image and to change output data easily. The display is always in a Byte manner.

Note: Only change and write output data if you know that no plant disturbances are caused by this. All output data written by the IO Monitor are transmitted at the bus and have effect on subordinate drives, IO etc.

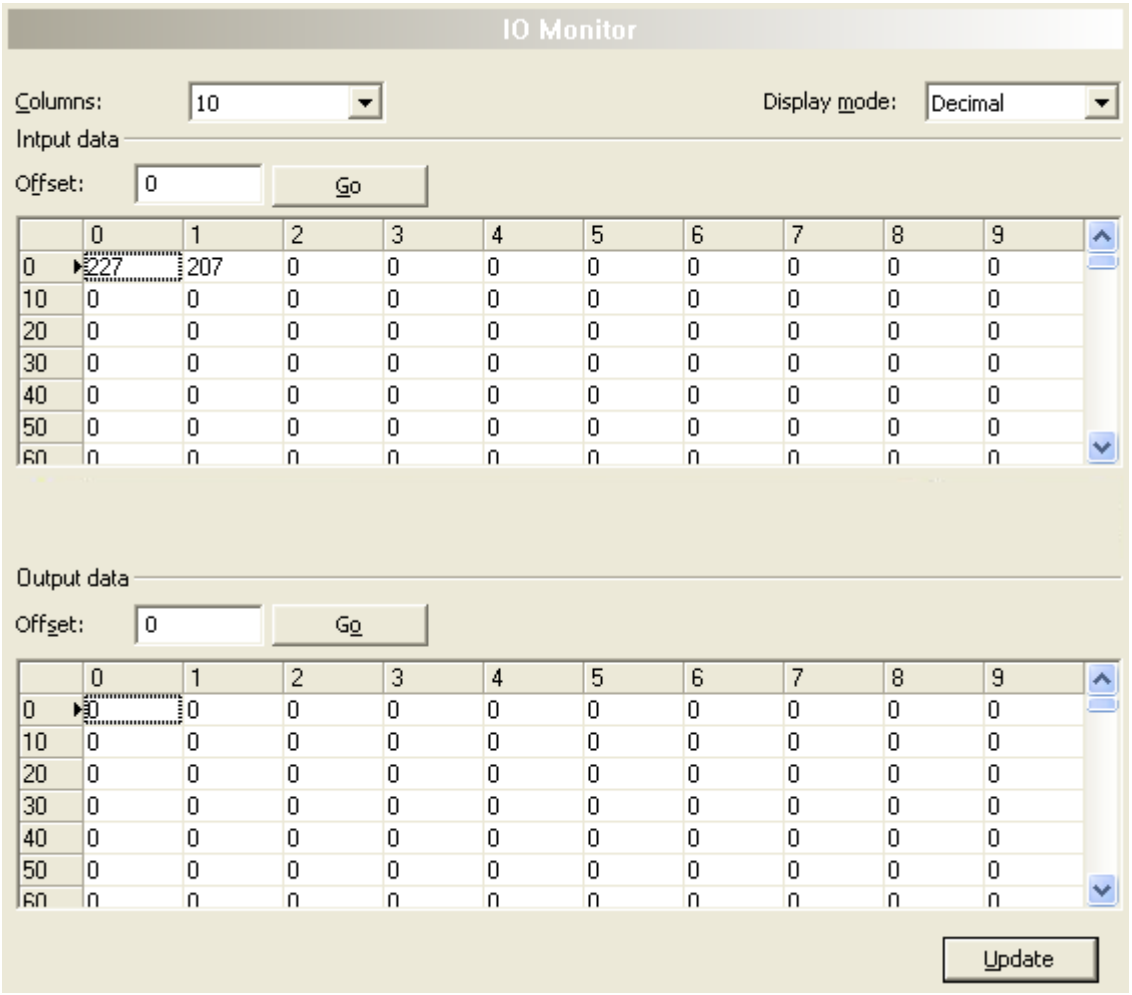


Figure 129: IO Monitor

Columns switches the number of columns.

Display mode switches the representation of the input and output data between decimal and hexadecimal.

Offset / **Go** moves the indication of the data to the entered offset value.

➢ Enter the output value and select **Update**.

↬ Always the data of the process image are displayed, also when these Bytes have not been reserved by the configuration.

10 Error Codes

10.1 Error Code Definition

For COM based application, like the ODM Server and ODM drivers, a common error definition is used, similar to the Microsoft Windows® HRESULT definition.

Error Code Structure:

COM Errors are HRESULTs, which are 32 bit values using the following layout:

```
 3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+-+-+-----------------------+-------------------------------+
|Sev|C|R|     Facility          |               Code            |
+---+-+-+-----------------------+-------------------------------+


where
 Sev       - is the severity code:
        00 - Success
        01 - Informational
        10 - Warning
        11 - Error
 C         - is the Customer code flag
 R         - is a reserved bit
 Facility  - is the facility code
 Code      - is the facility's status code
```

In this common error definition, several error code regions are already reserved by Windows® itself, the ODM and some other modules.

10.2 Overview Error Codes

Overview Error Codes	Range
General Hardware Errors RCX Operating System	*RCX General Task*: 0xC02B0001 to 0xC02B4D52
	RCX Common Status & Errors Codes: 0x00000000 to 0xC002000C
	RCX Status & Error Codes: 0x00000000 to 0xC0000008
ODM Server	*General ODM Error Codes*: 0x8004C700 to 0x8004C761
	General ODM Driver Error Codes: 0x8004C7A0 to 0x8004C7C2
ODM Drivers	*cifX Driver Specific ODM Error*: 0x8004C001 to 0x8004C0A4
cifX Device Driver and netX Driver	*Generic Error*: 0x800A0001 bis 0x800A0017
	Generic Driver Error: 0x800B0001 bis0x800B0042
	Generic Device Error: 0x800C0010 bis0x800C0041
netX Driver	*CIFX API Transport*: 0x800D0001 bis 0x800D0013
	CIFX API Transport Header State Error: 0x800E0001 bis 0x800E000B
DBM	*ODM Error Codes*: 0xC004C810 to 0xC004C878

Table 156: Overview Error Codes and Ranges

The fieldbus specific error codes are described in the manuals of the corresponding protocol tasks.

10.3 General Hardware Error Codes

10.3.1 RCX General Task Errors

Error Code (Definition)	Value	Description
RCX_E_QUE_UNKNOWN	0xC02B0001	Unknown Queue
RCX_E_QUE_INDEX_UNKNOWN	0xC02B0002	Unknown Queue Index
RCX_E_TASK_UNKNOWN	0xC02B0003	Unknown Task
RCX_E_TASK_INDEX_UNKNOWN	0xC02B0004	Unknown Task Index
RCX_E_TASK_HANDLE_INVALID	0xC02B0005	Invalid Task Handle
RCX_E_TASK_INFO_IDX_UNKNOWN	0xC02B0006	Unknown Index
RCX_E_FILE_XFR_TYPE_INVALID	0xC02B0007	Invalid Transfer Type
RCX_E_FILE_REQUEST_INCORRECT	0xC02B0008	Invalid File Request
RCX_E_TASK_INVALID	0xC02B000E	Invalid Task
RCX_E_SEC_FAILED	0xC02B001D	Security EEPROM Access Failed
RCX_E_EEPROM_DISABLED	0xC02B001E	EEPROM Disabled
RCX_E_INVALID_EXT	0xC02B001F	Invalid Extension
RCX_E_SIZE_OUT_OF_RANGE	0xC02B0020	Block Size Out Of Range
RCX_E_INVALID_CHANNEL	0xC02B0021	Invalid Channel
RCX_E_INVALID_FILE_LEN	0xC02B0022	Invalid File Length
RCX_E_INVALID_CHAR_FOUND	0xC02B0023	Invalid Character Found
RCX_E_PACKET_OUT_OF_SEQ	0xC02B0024	Packet Out Of Sequence
RCX_E_SEC_NOT_ALLOWED	0xC02B0025	Not Allowed In Current State
RCX_E_SEC_INVALID_ZONE	0xC02B0026	Security EEPROM Invalid Zone
RCX_E_SEC_EEPROM_NOT_AVAIL	0xC02B0028	Security EEPROM Eeprom Not Available
RCX_E_SEC_INVALID_CHECKSUM	0xC02B0029	Security EEPROM Invalid Checksum
RCX_E_SEC_ZONE_NOT_WRITEABLE	0xC02B002A	Security EEPROM Zone Not Writeable
RCX_E_SEC_READ_FAILED	0xC02B002B	Security EEPROM Read Failed
RCX_E_SEC_WRITE_FAILED	0xC02B002C	Security EEPROM Write Failed
RCX_E_SEC_ACCESS_DENIED	0xC02B002D	Security EEPROM Access Denied
RCX_E_SEC_EEPROM_EMULATED	0xC02B002E	Security EEPROM Emulated
RCX_E_INVALID_BLOCK	0xC02B0038	Invalid Block
RCX_E_INVALID_STRUCT_NUMBER	0xC02B0039	Invalid Structure Number
RCX_E_INVALID_CHECKSUM	0xC02B4352	Invalid Checksum
RCX_E_CONFIG_LOCKED	0xC02B4B54	Configuration Locked
RCX_E_SEC_ZONE_NOT_READABLE	0xC02B4D52	Security EEPROM Zone Not Readable

Table 157: RCX General Task Errors

10.3.2 RCX Common Status & Errors Codes

Error Code (Definition)	Value	Description
RCX_S_OK	0x00000000	Success, Status Okay
RCX_E_FAIL	0xC0000001	Fail
RCX_E_UNEXPECTED	0xC0000002	Unexpected
RCX_E_OUTOFMEMORY	0xC0000003	Out Of Memory
RCX_E_UNKNOWN_COMMAND	0xC0000004	Unknown Command
RCX_E_UNKNOWN_DESTINATION	0xC0000005	Unknown Destination
RCX_E_UNKNOWN_DESTINATION_ID	0xC0000006	Unknown Destination ID
RCX_E_INVALID_PACKET_LEN	0xC0000007	Invalid Packet Length
RCX_E_INVALID_EXTENSION	0xC0000008	Invalid Extension
RCX_E_INVALID_PARAMETER	0xC0000009	Invalid Parameter
RCX_E_WATCHDOG_TIMEOUT	0xC000000C	Watchdog Timeout
RCX_E_INVALID_LIST_TYPE	0xC000000D	Invalid List Type
RCX_E_UNKNOWN_HANDLE	0xC000000E	Unknown Handle
RCX_E_PACKET_OUT_OF_SEQ	0xC000000F	Out Of Sequence
RCX_E_PACKET_OUT_OF_MEMORY	0xC0000010	Out Of Memory
RCX_E_QUE_PACKETDONE	0xC0000011	Queue Packet Done
RCX_E_QUE_SENDPACKET	0xC0000012	Queue Send Packet
RCX_E_POOL_PACKET_GET	0xC0000013	Pool Packet Get
RCX_E_POOL_GET_LOAD	0xC0000015	Pool Get Load
RCX_E_REQUEST_RUNNING	0xC000001A	Request Already Running
RCX_E_INIT_FAULT	0xC0000100	Initialization Fault
RCX_E_DATABASE_ACCESS_FAILED	0xC0000101	Database Access Failed
RCX_E_NOT_CONFIGURED	0xC0000119	Not Configured
RCX_E_CONFIGURATION_FAULT	0xC0000120	Configuration Fault
RCX_E_INCONSISTENT_DATA_SET	0xC0000121	Inconsistent Data Set
RCX_E_DATA_SET_MISMATCH	0xC0000122	Data Set Mismatch
RCX_E_INSUFFICIENT_LICENSE	0xC0000123	Insufficient License
RCX_E_PARAMETER_ERROR	0xC0000124	Parameter Error
RCX_E_INVALID_NETWORK_ADDRESS	0xC0000125	Invalid Network Address
RCX_E_NO_SECURITY_MEMORY	0xC0000126	No Security Memory
RCX_E_NETWORK_FAULT	0xC0000140	Network Fault
RCX_E_CONNECTION_CLOSED	0xC0000141	Connection Closed
RCX_E_CONNECTION_TIMEOUT	0xC0000142	Connection Timeout
RCX_E_LONELY_NETWORK	0xC0000143	Lonely Network
RCX_E_DUPLICATE_NODE	0xC0000144	Duplicate Node
RCX_E_CABLE_DISCONNECT	0xC0000145	Cable Disconnected
RCX_E_BUS_OFF	0xC0000180	Network Node Bus Off
RCX_E_CONFIG_LOCKED	0xC0000181	Configuration Locked
RCX_E_APPLICATION_NOT_READY	0xC0000182	Application Not Ready
RCX_E_TIMER_APPL_PACKET_SENT	0xC002000C	Timer App Packet Sent

Table 158: RCX Common Status & Errors Codes

10.3.3 RCX Status & Error Codes

Error Code (Definition)	Value	Description
RCX_S_OK	0x00000000	SUCCESS, STATUS OKAY
RCX_S_QUE_UNKNOWN	0xC02B0001	UNKNOWN QUEUE
RCX_S_QUE_INDEX_UNKNOWN	0xC02B0002	UNKNOWN QUEUE INDEX
RCX_S_TASK_UNKNOWN	0xC02B0003	UNKNOWN TASK
RCX_S_TASK_INDEX_UNKNOWN	0xC02B0004	UNKNOWN TASK INDEX
RCX_S_TASK_HANDLE_INVALID	0xC02B0005	INVALID TASK HANDLE
RCX_S_TASK_INFO_IDX_UNKNOWN	0xC02B0006	UNKNOWN INDEX
RCX_S_FILE_XFR_TYPE_INVALID	0xC02B0007	INVALID TRANSFER TYPE
RCX_S_FILE_REQUEST_INCORRECT	0xC02B0008	INVALID FILE REQUEST
RCX_S_UNKNOWN_DESTINATION	0xC0000005	UNKNOWN DESTINATION
RCX_S_UNKNOWN_DESTINATION_ID	0xC0000006	UNKNOWN DESTINATION ID
RCX_S_INVALID_LENGTH	0xC0000007	INVALID LENGTH
RCX_S_UNKNOWN_COMMAND	0xC0000004	UNKNOWN COMMAND
RCX_S_INVALID_EXTENSION	0xC0000008	INVALID EXTENSION

Table 159: RCX Status & Error Codes

10.3.3.1 RCX Status & Error Codes Slave State

Error Code (Definition)	Value	Description
RCX_SLAVE_STATE_UNDEFINED	0x00000000	UNDEFINED
RCX_SLAVE_STATE_OK	0x00000001	OK
RCX_SLAVE_STATE_FAILED	0x00000002	FAILED (at least one slave)

Table 160: RCX Status & Error Codes Slave State

10.4 ODM Error Codes

10.4.1 General ODM Error Codes

Error Code (Definition)	Value	Description
CODM3_E_INTERNALERROR	0x8004C700	Internal ODM Error
ODM3_E_DESCRIPTION_NOTFOUND	0x8004C701	Description not found in ODM database
CODM3_E_WRITEREGISTRY	0x8004C710	Error writing to the registry
CODM3_E_BAD_REGULAR_EXPRESSION	0x8004C711	Invalid regular expression
CODM3_E_COMCATEGORIE_MANAGER_ FAILED	0x8004C712	Component Category Manager could not be instantiated
CODM3_E_COMCATEGORIE_ENUMERATION_ FAILED	0x8004C713	Driver could not be enumerated by the Category Manager
CODM3_E_CREATE_LOCAL_BUFFER	0x8004C714	Error creating local buffers
CODM3_E_UNKNOWNHANDLE	0x8004C715	Unknown handle
CODM3_E_QUEUE_LIMIT_REACHED	0x8004C717	Queue size limit for connection reached
CODM3_E_DATASIZE_ZERO	0x8004C718	Zero data length passed
CODM3_E_INVALID_DATA	0x8004C719	Invalid data content
CODM3_E_INVALID_MODE	0x8004C71A	Invalid mode
CODM3_E_DATABASE_READ	0x8004C71B	Error reading database
CODM3_E_CREATE_DEVICE_THREAD	0x8004C750	Error creating device thread
CODM3_E_CREATE_DEVICE_THREAD_STOP_ EVENT	0x8004C751	Error creating device thread stop event
CODM3_E_CLIENT_NOT_REGISTERED	0x8004C752	Client is not registered at the ODM
CODM3_E_NO_MORE_CLIENTS	0x8004C753	Maximum number of clients reached
CODM3_E_MAX_CLIENT_CONNECTIONS_ REACHED	0x8004C754	Maximum number of client connections reached
CODM3_E_ENTRY_NOT_FOUND	0x8004C755	Driver/device not found
CODM3_E_DRIVER_NOT_FOUND	0x8004C757	The requested driver is unknown to the ODM
CODM3_E_DEVICE_ALREADY_LOCKED	0x8004C758	Device is locked by another process
CODM3_E_DEVICE_UNLOCKED_FAILED	0x8004C759	Device could not be unlocked, lock was set by another process
CODM3_E_DEVICE_LOCK_NECCESSARY	0x8004C75A	Operation requires a device lock to be set
CODM3_E_DEVICE_SUBSCRIPTIONLIMIT	0x8004C75B	Maximum number of servers registered for this device reached
CODM3_E_DEVICE_NOTSUBSCRIBED	0x8004C75C	Process is not registered as a server on this device
CODM3_E_DEVICE_NO_MESSAGE	0x8004C75D	No message available
CODM3_E_TRANSFERTIMEOUT	0x8004C760	Message transfer timeout
CODM3_E_MESSAGE_INSERVICE	0x8004C761	Message in service

Table 161: ODM Error Codes - General ODM Error Codes

10.4.2 General ODM Driver Error Codes

Error Code (Definition)	Value	Description
CODM3_E_DRV_OPEN_DEVICE	0x8004C7A0	Packet type unsupported by driver
CODM3_E_DRV_INVALID_IDENTIFIER	0x8004C7A1	Invalid device identifier
CODM3_E_DRV_DEVICE_PARAMETERS_ MISMATCH	0x8004C7A3	Parameters differ from requested device
CODM3_E_DRV_BROWSE_NO_DEVICES	0x8004C7A4	No devices found
CODM3_E_DRV_CREATE_DEVICE_INST	0x8004C7A5	Device instance could not be created
CODM3_E_DRV_DEVICE_NOMORE_TX	0x8004C7A6	Device connection limit reached
CODM3_E_DRV_DEVICE_DUPLICATE_TX	0x8004C7A7	Duplicate transmitter ID
CODM3_E_DRV_DEVICE_NOT_CONFIGURED	0x8004C7A8	Device is not configured
CODM3_E_DRV_DEVICE_COMMUNICATION	0x8004C7A9	Device communication error
CODM3_E_DRV_DEVICE_NO_MESSAGE	0x8004C7AA	No message available
CODM3_E_DRV_DEVICE_NOT_READY	0x8004C7AB	Device not ready
CODM3_E_DRV_INVALIDCONFIGURATION	0x8004C7AC	Invalid driver configuration
CODM3_E_DRV_DLINVALIDMODE	0x8004C7C0	Invalid download mode
CODM3_E_DRV_DLINPROGRESS	0x8004C7C1	Download is active
CODM3_E_DRV_ULINPROGRESS	0x8004C7C2	Upload is active

Table 162: ODM Error Codes - General ODM Driver Error Codes

10.4.3 cifX Driver Specific ODM Error Codes

cifX Driver Specific ODM Error Codes		
Error Code (Definition)	**Value**	**Description**
DRV_E_BOARD_NOT_INITIALIZED	0x8004C001	DRIVER Board not initialized
DRV_E_INIT_STATE_ERROR	0x8004C002	DRIVER Error in internal init state
DRV_E_READ_STATE_ERROR	0x8004C003	DRIVER Error in internal read state
DRV_E_CMD_ACTIVE	0x8004C004	DRIVER Command on this channel is active
DRV_E_PARAMETER_UNKNOWN	0x8004C005	DRIVER Unknown parameter in function
DRV_E_WRONG_DRIVER_VERSION	0x8004C006	DRIVER Version is incompatible with DLL
DRV_E_PCI_SET_CONFIG_MODE	0x8004C007	DRIVER Error during PCI set configuration mode
DRV_E_PCI_READ_DPM_LENGTH	0x8004C008	DRIVER Could not read PCI dual port memory length
DRV_E_PCI_SET_RUN_MODE	0x8004C009	DRIVER Error during PCI set run mode
DRV_E_DEV_DPM_ACCESS_ERROR	0x8004C00A	DEVICE Dual port ram not accessable(board not found)
DRV_E_DEV_NOT_READY	0x8004C00B	DEVICE Not ready (ready flag failed)
DRV_E_DEV_NOT_RUNNING	0x8004C00C	DEVICE Not running (running flag failed)
DRV_E_DEV_WATCHDOG_FAILED	0x8004C00D	DEVICE Watchdog test failed
DRV_E_DEV_OS_VERSION_ERROR	0x8004C00E	DEVICE Signals wrong OS version
DRV_E_DEV_SYSERR	0x8004C00F	DEVICE Error in dual port flags
DRV_E_DEV_MAILBOX_FULL	0x8004C010	DEVICE Send mailbox is full
DRV_E_DEV_PUT_TIMEOUT	0x8004C011	DEVICE PutMessage timeout
DRV_E_DEV_GET_TIMEOUT	0x8004C012	DEVICE GetMessage timeout
DRV_E_DEV_GET_NO_MESSAGE	0x8004C013	DEVICE No message available
DRV_E_DEV_RESET_TIMEOUT	0x8004C014	DEVICE RESET command timeout
DRV_E_DEV_NO_COM_FLAG	0x8004C015	DEVICE COM-flag not set. Check if Bus is running
DRV_E_DEV_EXCHANGE_FAILED	0x8004C016	DEVICE I/O data exchange failed
DRV_E_DEV_EXCHANGE_TIMEOUT	0x8004C017	DEVICE I/O data exchange timeout
DRV_E_DEV_COM_MODE_UNKNOWN	0x8004C018	DEVICE I/O data mode unknown
DRV_E_DEV_FUNCTION_FAILED	0x8004C019	DEVICE Function call failed
DRV_E_DEV_DPMSIZE_MISMATCH	0x8004C01A	DEVICE DPM size differs from configuration
DRV_E_DEV_STATE_MODE_UNKNOWN	0x8004C01B	DEVICE State mode unknown
DRV_E_DEV_HW_PORT_IS_USED	0x8004C01C	DEVICE Output port already in use
DRV_E_USR_OPEN_ERROR	0x8004C01E	USER Driver not opened (device driver not loaded)
DRV_E_USR_INIT_DRV_ERROR	0x8004C01F	USER Can't connect to device
DRV_E_USR_NOT_INITIALIZED	0x8004C020	USER Board not initialized (DevInitBoard not called)
DRV_E_USR_COMM_ERR	0x8004C021	USER IOCTRL function failed
DRV_E_USR_DEV_NUMBER_INVALID	0x8004C022	USER Parameter DeviceNumber invalid
DRV_E_USR_INFO_AREA_INVALID	0x8004C023	USER Parameter InfoArea unknown
DRV_E_USR_NUMBER_INVALID	0x8004C024	USER Parameter Number invalid
DRV_E_USR_MODE_INVALID	0x8004C025	USER Parameter Mode invalid
DRV_E_USR_MSG_BUF_NULL_PTR	0x8004C026	USER NULL pointer assignment
DRV_E_USR_MSG_BUF_TOO_SHORT	0x8004C027	USER Message buffer too small

cifX Driver Specific ODM Error Codes		
Error Code (Definition)	**Value**	**Description**
DRV_E_USR_SIZE_INVALID	0x8004C028	USER Parameter Size invalid
DRV_E_USR_SIZE_ZERO	0x8004C02A	USER Parameter Size with zero length
DRV_E_USR_SIZE_TOO_LONG	0x8004C02B	USER Parameter Size too long
DRV_E_USR_DEV_PTR_NULL	0x8004C02C	USER Device address null pointer
DRV_E_USR_BUF_PTR_NULL	0x8004C02D	USER Pointer to buffer is a null pointer
DRV_E_USR_SENDSIZE_TOO_LONG	0x8004C02E	USER Parameter SendSize too large
DRV_E_USR_RECVSIZE_TOO_LONG	0x8004C02F	USER Parameter ReceiveSize too large
DRV_E_USR_SENDBUF_PTR_NULL	0x8004C030	USER Pointer to send buffer is a null pointer
DRV_E_USR_RECVBUF_PTR_NULL	0x8004C031	USER Pointer to receive buffer is a null pointer
DRV_E_DMA_INSUFF_MEM	0x8004C032	DMA Memory allocation error
DRV_E_DMA_TIMEOUT_CH4	0x8004C033	DMA Read I/O timeout
DRV_E_DMA_TIMEOUT_CH5	0x8004C034	DMA Write I/O timeout
DRV_E_DMA_TIMEOUT_CH6	0x8004C035	DMA PCI transfer timeout
DRV_E_DMA_TIMEOUT_CH7	0x8004C036	DMA Download timeout
DRV_E_DMA_DB_DOWN_FAIL	0x8004C037	DMA Database download failed
DRV_E_DMA_FW_DOWN_FAIL	0x8004C038	DMA Firmware download failed
DRV_E_CLEAR_DB_FAIL	0x8004C039	DMA Clear database on the device failed
DRV_E_DEV_NO_VIRTUAL_MEM	0x8004C03C	DMA USER Virtual memory not available
DRV_E_DEV_UNMAP_VIRTUAL_MEM	0x8004C03D	DMA USER Unmap virtual memory failed
DRV_E_GENERAL_ERROR	0x8004C046	DRIVER General error
DRV_E_DMA_ERROR	0x8004C047	DRIVER General DMA error
DRV_E_WDG_IO_ERROR	0x8004C048	DRIVER I/O WatchDog failed
DRV_E_WDG_DEV_ERROR	0x8004C049	DRIVER Device Watchdog failed
DRV_E_USR_DRIVER_UNKNOWN	0x8004C050	USER Driver unknown
DRV_E_USR_DEVICE_NAME_INVALID	0x8004C051	USER Device name invalid
DRV_E_USR_DEVICE_NAME_UKNOWN	0x8004C052	USER Device name unknown
DRV_E_USR_DEVICE_FUNC_NOTIMPL	0x8004C053	USER Device function not implemented
DRV_E_USR_FILE_OPEN_FAILED	0x8004C064	USER File could not be opened
DRV_E_USR_FILE_SIZE_ZERO	0x8004C065	USER File size zero
DRV_E_USR_FILE_NO_MEMORY	0x8004C066	USER Not enough memory to load file
DRV_E_USR_FILE_READ_FAILED	0x8004C067	USER File read failed
DRV_E_USR_INVALID_FILETYPE	0x8004C068	USER File type invalid
DRV_E_USR_FILENAME_INVALID	0x8004C069	USER Invalid filename
DRV_E_FW_FILE_OPEN_FAILED	0x8004C06E	USER Firmware file could not be opened
DRV_E_FW_FILE_SIZE_ZERO	0x8004C06F	USER Not enough memory to load firmware file
DRV_E_FW_FILE_NO_MEMORY	0x8004C070	USER Not enough memory to load firmware file
DRV_E_FW_FILE_READ_FAILED	0x8004C071	USER Firmware file read failed
DRV_E_FW_INVALID_FILETYPE	0x8004C072	USER Firmware file type invalid
DRV_E_FW_FILENAME_INVALID	0x8004C073	USER Firmware file name not valid
DRV_E_FW_DOWNLOAD_ERROR	0x8004C074	USER Firmware file download error
DRV_E_FW_FILENAME_NOT_FOUND	0x8004C075	USER Firmware file not found in the internal table
DRV_E_FW_BOOTLOADER_ACTIVE	0x8004C076	USER Firmware file BOOTLOADER active

cifX Driver Specific ODM Error Codes		
Error Code (Definition)	**Value**	**Description**
DRV_E_FW_NO_FILE_PATH	0x8004C077	USER Firmware file no file path
DRV_E_CF_FILE_OPEN_FAILED	0x8004C078	USER Configuration file could not be opend
DRV_E_CF_FILE_SIZE_ZERO	0x8004C079	USER Configuration file size zero
DRV_E_CF_FILE_NO_MEMORY	0x8004C07A	USER Not enough memory to load configuration file
DRV_E_CF_FILE_READ_FAILED	0x8004C07B	USER Configuration file read failed
DRV_E_CF_INVALID_FILETYPE	0x8004C07C	USER Configuration file type invalid
DRV_E_CF_FILENAME_INVALID	0x8004C07D	USER Configuration file name not valid
DRV_E_CF_DOWNLOAD_ERROR	0x8004C07E	USER Configuration file download error
DRV_E_CF_FILE_NO_SEGMENT	0x8004C07F	USER No flash segment in the configuration file
DRV_E_CF_DIFFERS_FROM_DBM	0x8004C080	USER Configuration file differs from database
DRV_E_DBM_SIZE_ZERO	0x8004C083	USER Database size zero
DRV_E_DBM_NO_MEMORY	0x8004C084	USER Not enough memory to upload database
DRV_E_DBM_READ_FAILED	0x8004C085	USER Database read failed
DRV_E_DBM_NO_FLASH_SEGMENT	0x8004C086	USER Database segment unknown
DEV_E_CF_INVALID_DESCRIPT_VERSION	0x8004C096	CONFIG Version of the descript table invalid
DEV_E_CF_INVALID_INPUT_OFFSET	0x8004C097	CONFIG Input offset is invalid
DEV_E_CF_NO_INPUT_SIZE	0x8004C098	CONFIG Input size is 0
DEV_E_CF_MISMATCH_INPUT_SIZE	0x8004C099	CONFIG Input size does not match configuration
DEV_E_CF_INVALID_OUTPUT_OFFSET	0x8004C09A	CONFIG Invalid output offset
DEV_E_CF_NO_OUTPUT_SIZE	0x8004C09B	CONFIG Output size is 0
DEV_E_CF_MISMATCH_OUTPUT_SIZE	0x8004C09C	CONFIG Output size does not match configuration
DEV_E_CF_STN_NOT_CONFIGURED	0x8004C09D	CONFIG Station not configured
DEV_E_CF_CANNOT_GET_STN_CONFIG	0x8004C09E	CONFIG Cannot get the Station configuration
DEV_E_CF_MODULE_DEF_MISSING	0x8004C09F	CONFIG Module definition is missing
DEV_E_CF_MISMATCH_EMPTY_SLOT	0x8004C0A0	CONFIG Empty slot mismatch
DEV_E_CF_MISMATCH_INPUT_OFFSET	0x8004C0A1	CONFIG Input offset mismatch
DEV_E_CF_MISMATCH_OUTPUT_OFFSET	0x8004C0A2	CONFIG Output offset mismatch
DEV_E_CF_MISMATCH_DATA_TYPE	0x8004C0A3	CONFIG Data type mismatch
DEV_E_CF_MODULE_DEF_MISSING_NO_SI	0x8004C0A4	CONFIG Module definition is missing,(no Slot/Idx)

Table 163: cifX Driver Specific ODM Error Codes

10.5 Error Codes cifX Device Driver and netX Driver

10.5.1 Generic Error Codes

Error Code (Definition)	Value	Description
CIFX_INVALID_POINTER	0x800A0001	Invalid pointer (NULL) passed to driver
CIFX_INVALID_BOARD	0x800A0002	No board with the given nameindex available
CIFX_INVALID_CHANNEL	0x800A0003	No channel with the given index available
CIFX_INVALID_HANDLE	0x800A0004	Invalid handle passed to driver
CIFX_INVALID_PARAMETER	0x800A0005	Invalid parameter
CIFX_INVALID_COMMAND	0x800A0006	Invalid command
CIFX_INVALID_BUFFERSIZE	0x800A0007	Invalid buffer size
CIFX_INVALID_ACCESS_SIZE	0x800A0008	Invalid access size
CIFX_FUNCTION_FAILED	0x800A0009	Function failed
CIFX_FILE_OPEN_FAILED	0x800A000A	File could not be opened
CIFX_FILE_SIZE_ZERO	0x800A000B	File size is zero
CIFX_FILE_LOAD_INSUFF_MEM	0x800A000C	Insufficient memory to load file
CIFX_FILE_CHECKSUM_ERROR	0x800A000D	File checksum compare failed
CIFX_FILE_READ_ERROR	0x800A000E	Error reading from file
CIFX_FILE_TYPE_INVALID	0x800A000F	Invalid file type
CIFX_FILE_NAME_INVALID	0x800A0010	Invalid file name
CIFX_FUNCTION_NOT_AVAILABLE	0x800A0011	Driver function not available
CIFX_BUFFER_TOO_SHORT	0x800A0012	Given buffer is too short
CIFX_MEMORY_MAPPING_FAILED	0x800A0013	Failed to map the memory
CIFX_NO_MORE_ENTRIES	0x800A0014	No more entries available
CIFX_CALLBACK_MODE_UNKNOWN	0x800A0015	Unkown callback handling mode
CIFX_CALLBACK_CREATE_EVENT_FAILED	0x800A0016	Failed to create callback events
CIFX_CALLBACK_CREATE_RECV_BUFFER	0x800A0017	Failed to create callback receive buffer

Table 164: Generic Error Codes

10.5.2 Generic Driver Error Codes

Error Code (Definition)	Value	Description
CIFX_DRV_NOT_INITIALIZED	0x800B0001	Driver not initialized
CIFX_DRV_INIT_STATE_ERROR	0x800B0002	Driver init state error
CIFX_DRV_READ_STATE_ERROR	0x800B0003	Driver read state error
CIFX_DRV_CMD_ACTIVE	0x800B0004	Command is active on device
CIFX_DRV_DOWNLOAD_FAILED	0x800B0005	General error during download
CIFX_DRV_WRONG_DRIVER_VERSION	0x800B0006	Wrong driver version
CIFX_DRV_DRIVER_NOT_LOADED	0x800B0030	CIFx driver is not running
CIFX_DRV_INIT_ERROR	0x800B0031	Failed to initialize the device
CIFX_DRV_CHANNEL_NOT_INITIALIZED	0x800B0032	Channel not initialized (xOpenChannel not called)
CIFX_DRV_IO_CONTROL_FAILED	0x800B0033	IOControl call failed
CIFX_DRV_NOT_OPENED(	0x800B0034	Driver was not opened
CIFX_DRV_DOWNLOAD_STORAGE_UNKNOWN	0x800B0040	Unknown download storage type (RAMFLASH based) found
CIFX_DRV_DOWNLOAD_FW_WRONG_CHANNEL	0x800B0041	Channel number for a firmware download not supported
CIFX_DRV_DOWNLOAD_MODULE_NO_BASEOS	0x800B0042	Modules are not allowed without a Base OS firmware

Table 165: Generic Driver Error Codes

10.5.3 Generic Device Error Codes

Error Code (Definition)	Value	Description
CIFX_DEV_DPM_ACCESS_ERROR	0x800C0010	Dual port memory not accessable (board not found)
CIFX_DEV_NOT_READY	0x800C0011	Device not ready (ready flag failed)
CIFX_DEV_NOT_RUNNING	0x800C0012	Device not running (running flag failed)
CIFX_DEV_WATCHDOG_FAILED	0x800C0013	Watchdog test failed
CIFX_DEV_SYSERR	0x800C0015	Error in handshake flags
CIFX_DEV_MAILBOX_FULL	0x800C0016	Send mailbox is full
CIFX_DEV_PUT_TIMEOUT	0x800C0017	Send packet timeout
CIFX_DEV_GET_TIMEOUT	0x800C0018	Receive packet timeout
CIFX_DEV_GET_NO_PACKET	0x800C0019	No packet available
CIFX_DEV_MAILBOX_TOO_SHORT	0x800C001A	Mailbox too short
CIFX_DEV_RESET_TIMEOUT	0x800C0020	Reset command timeout
CIFX_DEV_NO_COM_FLAG	0x800C0021	COM-flag not set
CIFX_DEV_EXCHANGE_FAILED	0x800C0022	IO data exchange failed
CIFX_DEV_EXCHANGE_TIMEOUT	0x800C0023	IO data exchange timeout
CIFX_DEV_COM_MODE_UNKNOWN	0x800C0024	Unknown IO exchange mode
CIFX_DEV_FUNCTION_FAILED	0x800C0025	Device function failed
CIFX_DEV_DPMSIZE_MISMATCH	0x800C0026	DPM size differs from configuration
CIFX_DEV_STATE_MODE_UNKNOWN	0x800C0027	Unknown state mode
CIFX_DEV_HW_PORT_IS_USED	0x800C0028	Device is still accessed
CIFX_DEV_CONFIG_LOCK_TIMEOUT	0x800C0029	Configuration locking timeout
CIFX_DEV_CONFIG_UNLOCK_TIMEOUT	0x800C002A	Configuration unlocking timeout
CIFX_DEV_HOST_STATE_SET_TIMEOUT	0x800C002B	Set HOST state timeout
CIFX_DEV_HOST_STATE_CLEAR_TIMEOUT	0x800C002C	Clear HOST state timeout
CIFX_DEV_INITIALIZATION_TIMEOUT	0x800C002D	Timeout during channel initialization
CIFX_DEV_BUS_STATE_ON_TIMEOUT	0x800C002E	Set Bus ON Timeout
CIFX_DEV_BUS_STATE_OFF_TIMEOUT	0x800C002F	Set Bus OFF Timeout
CIFX_DEV_MODULE_ALREADY_RUNNING	0x800C0040	Module already running
CIFX_DEV_MODULE_ALREADY_EXISTS	0x800C0041	Module already exists

Table 166: Generic Device Error Codes

10.6 Error Codes netX Driver

10.6.1 CIFX API Transport Error Codes

Error Code (Definition)	Value	Description
CIFX_TRANSPORT_SEND_TIMEOUT	0x800D0001	Time out while sending data
CIFX_TRANSPORT_RECV_TIMEOUT	0x800D0002	Time out waiting for incoming data
CIFX_TRANSPORT_CONNECT	0x800D0003	Unable to communicate to the deviceno answer
CIFX_TRANSPORT_ABORTED	0x800D0004	Transfer has been aborted due to keep alive timeout or interface detachment
CIFX_CONNECTOR_FUNCTIONS_READ_ERRO R	0x800D0010	Error reading the connector functions from the DLL
CIFX_CONNECTOR_IDENTIFIER_TOO_LONG	0x800D0011	Connector delivers an identifier longer than 6 characters
CIFX_CONNECTOR_IDENTIFIER_EMPTY	0x800D0012	Connector delivers an empty dentifier
CIFX_CONNECTOR_DUPLICATE_IDENTIFIER	0x800D0013	Connector identifier already used

Table 167: CIFX API Transport Error Codes

10.6.2 CIFX API Transport Header State Error Codes

Error Code (Definition)	Value	Description
CIFX_TRANSPORT_ERROR_UNKNOWN	0x800E0001	Unknown error code in transport header
CIFX_TRANSPORT_CHECKSUM_ERROR	0x800E0002	CRC16 checksum failed
CIFX_TRANSPORT_LENGTH_INCOMPLETE	0x800E0003	Transaction with inclomplete length detected
CIFX_TRANSPORT_DATA_TYPE_UNKOWN	0x800E0004	Device does not support requested data type
CIFX_TRANSPORT_DEVICE_UNKNOWN	0x800E0005	Device not availableunknown
CIFX_TRANSPORT_CHANNEL_UNKNOWN	0x800E0006	Channel not availableunknown
CIFX_TRANSPORT_SEQUENCE	0x800E0007	Sequence error detected
CIFX_TRANSPORT_BUFFEROVERFLOW	0x800E0008	Buffer overflow detected
CIFX_TRANSPORT_RESOURCE	0x800E0009	Device signals out of resources
CIFX_TRANSPORT_KEEPALIVE	0x800E000A	Device connection monitoring error (Keep alive)
CIFX_TRANSPORT_DATA_TOO_SHORT(	0x800E000B	Received transaction data too short

Table 168: CIFX API Transport Header State Error Codes

10.7 ODM Error Codes DBM V4

ODM Error Codes DBM V4		
Error Code (Definition)	**Value**	**Description**
CDBM_E_MD5_INVALID	0XC004C810	Checksum invalid
CDBM_E_INTERNALERROR	0XC004C811	Internal Error
CDBM_W_WRITEREGISTRY	0X8004C812	Error writing to the registry
CDBM_E_UNEXPECTED_VALUE_ IN_OLD_HEADER_FORMAT	0XC004C813	Error in a file containing the old DBM Header format.
CDBM_E_CHECKSUM_INVALID	0XC004C814	The Checksum of the old Header is invalid
CDBM_E_DB_ALREADY_LOADED_ FORMAT	0XC004C815	A database is already loaded
CDBM_E_NO_VALID_TRANSACTION	0XC004C816	No valid transaction handle given
CDBM_E_STD_STRUCT_ERROR	0XC004C817	An error occurred during validation of data
CDBM_E_UNSUPPORTED_ DATA_TYPE_FORMAT	0XC004C818	Unsupported DataType
CDBM_W_CLASS_DELETED_ FORMAT	0X8004C819 (Warning)	Using an Object which is marked as deleted
CDBM_W_CLIENT_DISCONNECTED	0X8004C81A (Warning)	A Client has already an outstanding connection to a Table. The connection is now destroyed.
CDBM_E_STRUCTURE_DEFINITION_ INVALID	0XC004C81B	A structure definition of an Element in a Table is invalid
CDBM_E_NO_DATA_AVAILABLE	0XC004C81C	No data available for this operation
CDBM_E_NO_VALID_STRUCTURE	0XC004C81D	No valid structure available for this operation
CDBM_E_NO_TOGGLE_STRING_ FOUND	0XC004C81E	No Toggle string found for this number
CDBM_E_ELEMENT_OUT_OF_RANGE	0XC004C81F	An element wasn’t found in the Record of a Table
CDBM_E_ELEMENT_NOT_IN_ TABLE	0XC004C820	The element is not part of the Table
CDBM_E_CANNOT_CONVERT_ INTO_CLIENT_TYPE	0XC004C821	The data can't be converted into the Client type
CDBM_E_TRANSACTION_ ALREADY_OPEN	0XC004C822	A transaction is already open. Please close this one first before opening a new one.
CDBM_I_OLD_WITHOUT_HEADER	0X4004C823 (Informational)	Use of an old DBM file Format without Header
CDBM_E_HR_FROM	0XC004C824	An HRESULT was received from a Subroutine
CDBM_E_PARAMETER	0XC004C825	A Parameter is invalid
CDBM_E_NOTIMPL	0XC004C826	Method is currently not implemented
CDBM_E_OUTOFMEMORY	0XC004C827	Out of memory
CDBM_E_NO_OPEN_TRANSACTION	0XC004C828	No transaction open
CDBM_E_NO_CONTENTS	0XC004C829	No contents available
CDBM_REC_NO_NOT_FOUND	0XC004C82A	Record not found
CDBM_STRUCTURE_ELEMENT_ NOT_FOUND	0XC004C82B	Element of the Structure not found
CDBM_E_NO_MORE_RECORDS_ IN_TABTYPE	0XC004C82C	Table type 3 can contain only one record
CDBM_E_WRITE	0XC004C82D	The data in the VARIANT must be given in a SafeArray
CDBM_E_WRITE_NO_PARRAY	0XC004C82E	The VARIANT contains no valid [parray] element

ODM Error Codes DBM V4		
Error Code (Definition)	**Value**	**Description**
CDBM_E_WRITE_CANT_ ACCESS_DATA	0XC004C82F	Unable to access SafeArray Data in the VARIANT
CDBM_E_WRITE_DATA	0XC004C830	To write the data of this Element it must be given as a BSTR, or as an Array of VT_UI1/VT_I1
CDBM_E_WRITE_BSTR_E1	0XC004C831	The BSTR string must have an even length.
CDBM_E_WRITE_BSTR_E2	0XC004C832	The BSTR string must contain only hex digits (0..9 and a/A..f/F).
CDBM_E_WRITE_CANT_ INTERPRET_ARRAY	0XC004C833	Unable to interpret data in the SafeArray.
CDBM_E_WRITE_VT_ERROR	0XC004C834	Data type in the SafeArray is not VT_UI1 or VT_I1.
CDBM_E_WRITE_LENGTH	0XC004C835	Data length is invalid for write operation of this type.
CDBM_WRITE_ELEMENT	0XC004C836	Element not found in the Record of the Table
CDBM_MIN_MAX_ERROR	0XC004C837	Can't write data because of min underflow or max overflow
CDBM_TABLE_EXIST	0XC004C838	Table already exist in the database
CDBM_MIN_MAX_INVALID	0XC004C839	The Min value is greater than the Max Value
CDBM_DEF_MIN_MAX_INVALID	0XC004C83A	The Default Value is not in the range between the Min value and the Max Value
CDBM_CANT_CHANGE_STRUCTURE_ WHILE_RECORDS_EXIST	0XC004C83B	It's not allowed to change the structure while Records exist in the Table
CDBM_NEW_STRUCT_NEEDS_TYPE	0XC004C83C	In a newly added structure the data type must be set also
CDBM_VALUE_ERROR	0XC004C83D	Range error while validating a value
CDBM_DATATYPE_ UNSUPPORTED_IN_RCS	0XC004C83E	The data type is unsupported in the RCS file format
CDBM_I_COUNT_OF_TABLES_ EXCEEDS_RCS_RANGE	0X4004C83F (Informational)	The count of Tables exceeds the RCS range of Tables. This can cause problems if the file is downloaded to RCS Systems
CDBM_I_COUNT_OF_TABLES_ EXCEEDS_OLDDBM_RANGE	0X4004C840 (Informational)	The count of Tables exceeds the DBM32.DLL range of Tables. This can cause problems if the file is used with older Tools using the DBM32.DLL
CDBM_UNSUPPORTED_DATATYPE_ IN_RCS_MODE	0XC004C841	The Data type is not compatible with the old database format
CDBM_WRITE_UNSTRUCTURED_1	0XC004C842	The data of an unstructured record can only be written with the 'Write' Method not with 'WriteElement'.
CDBM_READ_UNSTRUCTURED_1	0XC004C843	The data of an unstructured record can only be read with the 'Read' Method not with 'ReadElement'
CDBM_WRITE_DATA_LENGTH_ INVALID	0XC004C844	The given data length doesn’t correspond with the expected data length.
CDBM_UNKNOWN_VIEW_MODE	0XC004C845	The View Mode is unknown.
CDBM_E_DIAG_TABLE	0XC004C846	It doesn’t make much sense to add or delete records from a diagnostic table because those changes are never saved.

ODM Error Codes DBM V4		
Error Code (Definition)	**Value**	**Description**
CDBM_E_ADR_STRING_ERROR	0XC004C847	The given Address string doesn't fit the required format of this type where all address bytes must be in the range between 0 and FF
CDBM_ERROR_FROM_VAR_ CHANGE_TYPE	0XC004C848	Function VariantChangeType return an error when trying to convert the Parameter
CDBM_E_MINERROR	0XC004C849	Error while comparing the Value with the lower range
CDBM_E_MAXERROR	0XC004C84A	Error while comparing the Value with the upper range
CDBM_E_RANGE_ERROR	0XC004C84B	Value out of Range
CDBM_E_TABLE_TYPE1	0XC004C84C	Table type 1 doesn't have a unique record length over all records
CDBM_E_TABLE_TYPE3_ ADDREC	0XC004C84D	Table type 3 doesn't allow to insert more than one Record
CDBM_E_TABTYPE1	0XC004C84E	It's not allowed to insert more Records than structure definitions in Table Type 1
CDBM_E_TOGGLE_NOT_FOUND	0XC004C84F	Could not find the string for this value in the list of valid toggle strings
CDBM_E_TOGGLE_VALUE_IS_ EMPTY_STRING	0XC004C850	The toggle string for this value is empty.
CDBM_VARIANT2BYTEARRAY_ ERROR	0XC004C851	Error during conversion of Variant to byte array
CDBM_E_SET_ELEM_PROP_ DEPENDENCY	0XC004C852	The Toggle Type needs also the additional string and the additional number entries in the Method
CDBM_E_TABTYPE1_REC_ DOESNT_CORRESPOND_ WITH_ELEMENT	0XC004C853	When reading the records of Table type 1 elementwise the record number must correspond with the element number
CDBM_TABTYPE1_NO_DATA_ FOUND_FOR_RECORD	0XC004C854	When reading the records of Table type 1 and structure definitions are present it's assumed that for each structure element a corresponding record must exist
CDBM_E_TABTYPE1_WRITE_ ELEMENT_NE_RECORD	0XC004C855	When writing the records of Table type 1 elementwise and structure definitions are present it's only allowed to write the corresponding element number in each record
CDBM_E_TABTYPE1_WRITE_ ELEMENT_NOT_FOUND	0XC004C856	When writing the records of Table type 1 with an array and structure definitions are present it's assumed that a corresponding element number of this record exist
CDBM_I_TABLE_NAME_EXCEEDS_ RCS_RANGE	0X4004C857 (Informational)	The Table name exceeds the maximum length of RCS compatible Table names
CDBM_W_CUT_STRING	0X8004C858 (Warning)	The string exceeds the maximum length and will be limited to the maximum length
CDBM_I_STRING_TOO_SHORT	0X4004C859 (Informational)	The string is below the minimum length. The minimum length will be reduced.
CDBM_I_STRING_TOO_LONG	0X4004C85A (Informational)	The string is exceeding the maximum. The maximum length will be extended.
CDBM_E_STRING_TOO_SHORT	0XC004C85B (Error)	The string is below the minimum length.
CDBM_E_STRING_TOO_LONG	0XC004C85C (Error)	The string is exceeding the maximum length

ODM Error Codes DBM V4		
Error Code (Definition)	**Value**	**Description**
CDBM_E_WRONG_TYPE_ FOR_WRITE	0XC004C85D	Writing on the Element type with the given Data type is not implemented
CDBM_E_NO_APPEND_IN_ STRUCTURED_RECORDS	0XC004C85E	Method IDbmRecord::AppendData is not allowed for structured records
CDBM_E_DATA_UNAVAILABLE	0XC004C85F	No data available
CDBM_E_CANT_CONVERT_ INTO	0XC004C860	Unable to convert the value into the Element type
CDBM_E_DBM_FILE_OVERFLOW	0XC004C861	You try to write a RCS like database which needs too much bytes
CDBM_E_PW_ERROR	0XC004C862	Password not correct
CDBM_E_FILELENGTH_CORRUPT	0XC004C863	The file length doesn't correspond to the length given in the Header.
CDBM_E_STRUCT_TYPE	0XC004C864	Error in the file.
CDBM_E_MD5SUM_INVALID	0XC004C865	MD5 sum invalid
CDBM_E_STRUCT_LENGTH	0XC004C866	Error in the expected and given structure length at a specific offset in the file.
CDBM_E_APPEND	0XC004C867	Append of data is only allowed if the Record contains only one data field and the field type will support this
CDBM_APPEND_NOT_ SUPPORTED	0XC004C868	Append of Data not supported by this filed type
CDBM_DATA_TYPE_APPEND_ ERROR	0XC004C869	Can't append Data of this type.
CDBM_E_UNSTRUCTURED_TABLE_ DOESNT_SUPPORT_LENGTH	0XC004C86A	A Table without structure information doesn't support a record length
CDBM_E_DISABLED_WHILE_ TRANSACTION_IS_OPEN	0XC004C86B	The Method is disabled while a transaction is open. Please close this one first and call the Method again.
CDBM_E_UNABLE_TO_CALL_ READ_ON_LINKED_LIST	0XC004C86C	The Method is disabled on a LinkedList type. Please use the IRecordCollection on this type.
CDBM_E_ELEMENT_HAS_NO_ SUBSTRUCTURE	0XC004C86D	An Element from a Table has no substructure
CDBM_STRUCT_ERROR_FROM_ VAR_CHANGE_TYPE	0XC004C86E	Error from calling VariantChangeType
CDBM_E_FOREIGNKEY_DEF	0XC004C86F	The definition of a FOREIGNKEY must contain the name of the related Table in the description and this Table must exist at this time
CDBM_E_FOREIGNKEY_ REF_TAB	0XC004C870	The description of a FOREIGNKEY must refer to a Table of type 'eDbmTableTypeLinkedList'
CDBM_E_KEY	0XC004C871	To create a Record Collection with a KEY it's necessary to have the data type KEY at the first position in all Records of the searched Table
CDBM_E_KEY_TABLE_TYPE	0XC004C872	This Method needs a Table of type 'eDbmTableTypeLinkedList'
CDBM_DATATYPE_NOT_ IMPLEMENTED	0XC004C873	This data type is currently not implemented
CDBM_INSERT_POS_NOT_ FOUND	0XC004C874	The position of the Record where the new one should be inserted wasn't found
CDBM_E_INSERT_REC_QI	0XC004C875	Error during insertion of a Record
CDBM_E_TAB_PROP	0XC004C876	Invalid Property in Table
CDBM_E_KEY_NOT_FOUND	0XC004C877	The KEY wasn't found in the Table

ODM Error Codes DBM V4		
Error Code (Definition)	**Value**	**Description**
CDBM_E_KEY_INVALID	0XC004C878	The KEY is invalid for this operation

Table 169: ODM Error Codes DBM V4

11 Appendix

11.1 User Rights

User-rights are set within the FDT-container. Depending on the level the configuration is accessible by the user or read-only.

To access the **Settings**, **Configuration** and **Diagnosis** windows of the netSLAVE DTM you do not need special user rights. Also all users can select the decimal or hexadecimal Display mode or sort table entries.

Note: To edit, set or configure the parameters of the **Settings** and **Configuration** windows, you need user rights for *Maintenance,* for *Planning Engineer* or for *Administrator*.

The following tables give an overview of the user right groups and which user rights you need to configure the single parameters.

11.1.1 Settings

	Observer	Operator	Maintenance	Planning Engineer	Adminis-trator
Driver	D	D	X	X	X
Verify or adapt Driver Settings	-	-	X	X	X
Configuring netX Driver	-	-	X	X	X
Device Assignment	D	D	X	X	X
Scanning for Devices	-	-	X	X	X
Selecting the Device (with or without firmware)	-	-	X	X	X
Selecting the Device once more (with Firmware)	-	-	X	X	X
Firmware Download	D	D	X	X	X

Table 170: Settings (D = Displaying, X = Editing, Configuring)

11.1.2 Configuration

	Observer	Operator	Maintenance	Planning Engineer	Adminis-trator
Configuration	D	D	X	X	X

Table 171: Configuration (D = Displaying, X = Editing, Configuring)

11.2 References

[1] Device Type Manager (DTM) Style Guide, Version 1.0 ; FDT-JIG - Order No. <0001-0008-000>

[2] EtherNetIP Adapter Protocol API Manual, Revision 12, Hilscher GmbH 2013

[3] Open Modbus/TCP Protocol API Manual, Revision 7, Hilscher GmbH 2013

[4] PROFINET IO-Device Protocol API Manual (V3.4), Revision 13, Hilscher GmbH 2013

PROFINET IO-Device Protocol API Manual (V3.5), Revision 5, Hilscher GmbH 2013

[5] Sercos Slave Protocol API Manual (V3), Revision 12, Hilscher GmbH 2013

[6] PROFIBUS DP Slave Protocol API Manual, Revision 15, Hilscher GmbH 2013

[7] CC-Link Slave Protocol API Manual, Revision 8, Hilscher GmbH 2013

[8] CANopen Slave Protocol API Manual (V3), Revision 4, Hilscher GmbH 2013

[9] EtherCAT Slave Protocol API Manual, Revision 3 (V4), Hilscher GmbH 2013

EtherCAT Slave Protocol API Manual, Revision 21 (V2), Hilscher GmbH 2013

[10] POWERLINK Controlled Node/Slave Protocol API Manual, Revision 11, Hilscher GmbH 2013

[11] DeviceNet Master Protocol API Manual, Revision 9, Hilscher GmbH 2013

[12] CompoNet Slave Protocol API Manual, Revision 2, Hilscher GmbH 2010

[13] 3964R Slave Protocol API Manual, Revision 1, Hilscher GmbH 2010

[14] Modbus RTU Protocol API Manual, Revision 1, Hilscher GmbH 2010

[15] VARAN Client Protocol API Manual, Revision 3, Hilscher GmbH 2013

[16] Dual-Port Memory Interface Manual, Revision 12, Hilscher GmbH 2012

[17] Application Layer protocol for decentralized periphery and distributed automation, Technical Specification for PROFINET, Version 2.3, October 2010, Order No: 2.722, PROFIBUS Nutzerorganisation e.V., Karlsruhe

References Safety

[S1] ANSI Z535.6-2011 American National Standard for Product Safety Information in Product Manuals, Instructions, and Other Collateral Materials

[S4] 26514-2010 - IEEE Standard for Adoption of ISO/IEC 26514:2008 Systems and Software Engineering--Requirements for Designers and Developers of User Documentation

11.3 Conventions in this Manual

Instructions

1. Operational purpose
2. Operational purpose

➢ Instruction

Results

Result

Sign

Sign	Note
	General note
	Important note that must be followed to prevent malfunctions.
	Reference to further information

Table 172: Sign

Signal Words

Signal word	Description
DANGER	Indicates a hazardous situation which if not avoided, will result in death or serious injury.
WARNING	Indicates a hazardous situation which if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazardous situation which if not avoided, may result in minor or moderate Injury.
NOTICE	Indicates a property damage message.

Table 173: Signal Words

11.4 Legal Notes

Important notes

Utmost care was/is given in the preparation of the documentation at hand consisting of a user's manual, operating manual and any other document type and accompanying texts. However, errors cannot be ruled out. Therefore, we cannot assume any guarantee or legal responsibility for erroneous information or liability of any kind. You are hereby made aware that descriptions found in the user's manual, the accompanying texts and the documentation neither represent a guarantee nor any indication on proper use as stipulated in the agreement or a promised attribute. It cannot be ruled out that the user's manual, the accompanying texts and the documentation do not completely match the described attributes, standards or any other data for the delivered product. A warranty or guarantee with respect to the correctness or accuracy of the information is not assumed.

We reserve the right to modify our products and the specifications for such as well as the corresponding documentation in the form of a user's manual, operating manual and/or any other document types and accompanying texts at any time and without notice without being required to notify of said modification. Changes shall be taken into account in future manuals and do not represent an obligation of any kind, in particular there shall be no right to have delivered documents revised. The manual delivered with the product shall apply.

Under no circumstances shall Hilscher Gesellschaft für Systemautomation mbH be liable for direct, indirect, ancillary or subsequent damage, or for any loss of income, which may arise after use of the information contained herein.

Liability disclaimer

The hardware and/or software was created and tested by Hilscher Gesellschaft für Systemautomation mbH with utmost care and is made available as is. No warranty can be assumed for the performance or

flawlessness of the hardware and/or software under all application conditions and scenarios and the work results achieved by the user when using the hardware and/or software. Liability for any damage that may have occurred as a result of using the hardware and/or software or the corresponding documents shall be limited to an event involving willful intent or a grossly negligent violation of a fundamental contractual obligation. However, the right to assert damages due to a violation of a fundamental contractual obligation shall be limited to contract-typical foreseeable damage.

It is hereby expressly agreed upon in particular that any use or utilization of the hardware and/or software in connection with

- Flight control systems in aviation and aerospace;
- Nuclear fusion processes in nuclear power plants;
- Medical devices used for life support and
- Vehicle control systems used in passenger transport

shall be excluded. Use of the hardware and/or software in any of the following areas is strictly prohibited:

- For military purposes or in weaponry;
- For designing, engineering, maintaining or operating nuclear systems;
- In flight safety systems, aviation and flight telecommunications systems;
- In life-support systems;
- In systems in which any malfunction in the hardware and/or software may result in physical injuries or fatalities.

You are hereby made aware that the hardware and/or software was not created for use in hazardous environments, which require fail-safe control mechanisms. Use of the hardware and/or software in this kind of environment shall be at your own risk; any liability for damage or loss due to impermissible use shall be excluded.

Warranty

Hilscher Gesellschaft für Systemautomation mbH hereby guarantees that the software shall run without errors in accordance with the requirements listed in the specifications and that there were no defects on the date of acceptance. The warranty period shall be 12 months commencing as of the date of acceptance or purchase (with express declaration or implied, by customer's conclusive behavior, e.g. putting into operation permanently).

The warranty obligation for equipment (hardware) we produce is 36 months, calculated as of the date of delivery ex works. The aforementioned provisions shall not apply if longer warranty periods are mandatory by law pursuant to Section 438 (1.2) BGB, Section 479 (1) BGB and Section 634a (1) BGB [Bürgerliches Gesetzbuch; German Civil Code] If, despite of all due care taken, the delivered product should have a defect, which already existed at the time of the transfer of risk, it shall be at our discretion to either repair the product or to deliver a replacement product, subject to timely notification of defect.

The warranty obligation shall not apply if the notification of defect is not asserted promptly, if the purchaser or third party has tampered with the products, if the defect is the result of natural wear, was caused by unfavorable operating conditions or is due to violations against our

operating regulations or against rules of good electrical engineering practice, or if our request to return the defective object is not promptly complied with.

Costs of support, maintenance, customization and product care

Please be advised that any subsequent improvement shall only be free of charge if a defect is found. Any form of technical support, maintenance and customization is not a warranty service, but instead shall be charged extra.

Additional guarantees

Although the hardware and software was developed and tested in-depth with greatest care, Hilscher Gesellschaft für Systemautomation mbH shall not assume any guarantee for the suitability thereof for any purpose that was not confirmed in writing. No guarantee can be granted whereby the hardware and software satisfies your requirements, or the use of the hardware and/or software is uninterruptable or the hardware and/or software is fault-free.

It cannot be guaranteed that patents and/or ownership privileges have not been infringed upon or violated or that the products are free from third-party influence. No additional guarantees or promises shall be made as to whether the product is market current, free from deficiency in title, or can be integrated or is usable for specific purposes, unless such guarantees or promises are required under existing law and cannot be restricted.

Confidentiality

The customer hereby expressly acknowledges that this document contains trade secrets, information protected by copyright and other patent and ownership privileges as well as any related rights of Hilscher Gesellschaft für Systemautomation mbH. The customer agrees to treat as confidential all of the information made available to customer by Hilscher Gesellschaft für Systemautomation mbH and rights, which were disclosed by Hilscher Gesellschaft für Systemautomation mbH and that were made accessible as well as the terms and conditions of this agreement itself.

The parties hereby agree to one another that the information that each party receives from the other party respectively is and shall remain the intellectual property of said other party, unless provided for otherwise in a contractual agreement.

The customer must not allow any third party to become knowledgeable of this expertise and shall only provide knowledge thereof to authorized users as appropriate and necessary. Companies associated with the customer shall not be deemed third parties. The customer must obligate authorized users to confidentiality. The customer should only use the confidential information in connection with the performances specified in this agreement.

The customer must not use this confidential information to his own advantage or for his own purposes or rather to the advantage or for the purpose of a third party, nor must it be used for commercial purposes and this confidential information must only be used to the extent provided for in this agreement or otherwise to the extent as expressly authorized by the disclosing party in written form. The customer has the right, subject to the obligation to confidentiality, to disclose the terms and conditions of this agreement directly to his legal and financial consultants as would be required for the customer's normal business operation.

Export provisions

The delivered product (including technical data) is subject to the legal export and/or import laws as well as any associated regulations of various countries, especially such laws applicable in Germany and in the United States. The products / hardware / software must not be exported into such countries for which export is prohibited under US American export control laws and its supplementary provisions. You hereby agree to strictly follow the regulations and to yourself be responsible for observing them. You are hereby made aware that you may be required to obtain governmental approval to export, reexport or import the product.

11.5 Registered Trademarks

Windows® XP, Windows® Vista, Windows® 7, Windows® 8, Windows® 8.1 and Windows® 10 are registered trademarks of Microsoft Corporation.

CANopen® is a registered trademark of CAN in AUTOMATION - International Users and Manufacturers Group e.V (CiA), Nürnberg.

CC-Link is a registered trademark of Mitsubishi Electric Corporation, Tokyo, Japan.

CompoNet™, DeviceNet™ and EtherNet/IP™ are trademarks of ODVA (Open DeviceNet Vendor Association, Inc).

EtherCAT® is a registered trademark and a patented technology of Beckhoff Automation GmbH, Verl, Germany, formerly Elektro Beckhoff GmbH.

Modbus is a registered trademark of Schneider Electric.

POWERLINK is a registered trademark of B&R, Bernecker + Rainer Industrie-Elektronik Ges.m.b.H, Eggelsberg, Austria

PROFIBUS® und PROFINET® are registered trademarks of PROFIBUS & PROFINET International (PI), Karlsruhe.

Sercos and Sercos interface are registered trademarks of Sercos international e. V., Suessen, Germany.

All other mentioned trademarks are property of their respective legal owners.

11.6 List of Figures

11.7 List of Tables

11.8 Glossary

Auto-Negotiation

Auto-Negotiation is a feature of an interface: An interface with Auto-Negotiation will automatically determine a set of correct communication parameters.

Baud rate

Data transmission speed of a communication channel or interface.

Coil

A coil (in the meaning defined by Modbus terminology) is a single bit in memory that can be accessed (i.e. read or write) via Modbus.

cifX

Communication **I**nter**F**ace based on net**X**

CSP

electronic device data sheet, required for each CC-Link device

Device Description File

A file containing configuration information about a device being a part of a network that can be read out by masters for system configuration. Device Description Files use various formats which depend on the communication system. Often these formats are based on XML such as EDS files or GSDML files. Contains configuration information

DHCP

Dynamic Host Configuration Protocol

This is a protocol simplifying the configuration of IP networks by automatically assigning IP addresses.

DPM

Dual-Port Memory

DTM

Device Type Manager

The Device Type Manager (DTM) is a software module with graphical user interface for the configuration and/or for diagnosis of devices.

EDS

Electronic Data Sheet

XML based device description file.

EDS file

A special kind of Device Description File used by EtherNet/IP.

EtherCAT

A communication system for industrial Ethernet designed and developed by Beckhoff Automation GmbH.

EtherNet/IP

A communication system for industrial Ethernet designed and developed by Rockwell. It partly uses the CIP (Common Industrial Protocol).

EtherNet/IP Scanner

A Scanner exchanges real-time I/O data with Adapters and Scanners. This type of node can respond to connection requests and can also initiate connections on its own.

EtherNet/IP Adapter

An Adapter emulates functions provided by traditional rack-adapter products. This type of node exchanges real-time I/O data with a Scanner Class product. It does not initiate connections on its own.

Ethernet POWERLINK

A communication system for industrial Ethernet designed and developed by B&R. It partly uses CANopen technologies.

FDT

Field Device Tool

FDT specifies an interface, in order to be able to use DTM (Device Type Manager) in different applications of different manufacturers.

Full duplex

Full duplex denominates a telecommunication system between two communication partners which allows simultaneous communication in both directions is called a full-duplex telecommunication system. At such a system, it is possible to transmit data even if currently data are received. Full-duplex is the opposite of Half duplex.

Function code

A function code (in the meaning defined by Modbus terminology) is a standardized method to access (i.e. read or write) coils or registers via Modbus.

Gateway

A device interfacing between two different communication standards.

GSD

Generic Station Description, Device description file

GSD file

A special kind of Device Description File used by PROFIBUS (GSD = Generic Station Description).

GSDML

Generic Station Description Markup Language

XML based device description file.

GSDML file

A special kind of XML-based Device Description File used by PROFINET.

Half duplex

Half duplex denominates a telecommunication system between two communication partners which does not allow simultaneous, but alternating, communication in both directions is called a half-duplex telecommunication system. At such a system, receiving data inhibits the transmission of data. Half-duplex is the opposite of Full duplex.

Hub

A network component connecting multiple communication partners with each other. A hub does not provide own intelligence, thus it does not analyze the data traffic and sends received data to all connected communication partners. A hub can be used for setting up a star topology.

Industrial Ethernet

See Real-Time Ethernet

Master

Master devices initiate the data traffic on the bus. In the communication protocol Master devices are called active participants. A master may send messages without external request.

netSCRIPT

Is a script-based programming language for communication devices of Hilscher GmbH, witch allows users themselves to program flows to program / protocol conversions into sights.

netX

networX on chip, next generation of communication controllers

Object Dictionary

An object dictionary is a storage area for device parameter data structures. It is accessed in standardized manner.

ODMV3

The Online-Data-Manager Version 3 (ODMV3) is an application interface. The ODMV3 works as a server, which can be run as an out-proc server or system service. Its task is to provide different applications (e. g. SYCON.net), access to multiple devices and even share one device amongst several applications.

Open Modbus/TCP

A communication system for Industrial Ethernet designed and developed by Schneider Automation and maintained by the Modbus-IDA organization based on the Modbus protocols for serial communication.

PROFINET

A communication system for Industrial Ethernet designed and developed by PROFIBUS & PROFINET International (PI). It uses some mechanisms similar to those of the PROFIBUS field bus.

PROFINET IO Controller

A PROFINET control unit responsible for the defined run-up of an I/O subsystem and the cyclic or acyclic data exchange.

PROFINET IO Device

A PROFINET field device that cyclically receives output data from its IO-Controller and responds with its input data.

RE

RE stands for Real Time Ethernet

Real-Time Ethernet

Real-Time Ethernet (Industrial Ethernet) is an extension of the Ethernet networking technology for industrial purposes with very good Real-Time features and performance. There is a variety of different Real-Time Ethernet systems on the market which are incompatible with each other. The most important systems of these are

- EtherCAT
- EtherNet/IP
- Ethernet POWERLINK
- Open Modbus/TCP
- PROFINET
- Sercos
- VARAN

Register

A register (in the meaning defined by Modbus terminology) is a 16-bit wide storage area for data which can be accessed and addressed as a unit by some of the Modbus Function Codes.

Sercos

A communication system for industrial Ethernet designed and developed by Bosch-Rexroth and supported by Sercos International.

Slave

Slave devices are peripheral devices, like for example I/O devices or drives. Slave devices are also called passive participants. They do not receive the bus access authorization. That means, they may only accept received messages from the Master or send a message to the Master after enquiry of the Master.

Switch

A network component connecting multiple communication partners (or even entire branches of a network) with each other. A switch is an intelligent

network component which analyzes network traffic in order to decide on its own. For the connected communication partners a switch behaves transparently.

SYNC

Sychronisation cycle of the master

TCP/IP

Transport Control Protocol/Internet Protocol connection-orientated, secure transfer protocol as basis for the Internet-protocols

VARAN

Versatile **A**utomation **R**andom **A**ccess **N**etwork

A communication system for industrial Ethernet based on the DIAS-BUS developed by Sigmatek. The system is supported by the VARAN-BUS-NUTZERORGANISATION (VNO).

Watchdog Timer

A watchdog timer provides an internal supervision mechanism of a communication system. It supervises that an important event happens within a given timeframe (the watchdog time which can be adjusted accordingly, for instance by a parameter in the _Warmstart message) and causes an alarm otherwise (usually this is accomplished by changing the operational state of the communication system to a more safe state).

XDD file

A special kind of Device Description file used by Ethernet POWERLINK.

XML

XML means Extended Markup Language. It is a symbolic language for structuring data systematically. XML is standard maintained by the W3C (World-wide web consortium). Device Description Files often use XML-based formats for storing the device-related data appropriately.

11.9 Contacts

Headquarters

Germany
Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support
Phone: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

Subsidiaries

China
Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support
Phone: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

France
Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support
Phone: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

India
Hilscher India Pvt. Ltd.
Pune, Delhi, Mumbai
Phone: +91 8888 750 777
E-Mail: info@hilscher.in

Italy
Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-Mail: info@hilscher.it

Support
Phone: +39 02 25007068
E-Mail: it.support@hilscher.com

Japan
Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support
Phone: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Korea
Hilscher Korea Inc.
Seongnam, Gyeonggi, 463-400
Phone: +82 (0) 31-789-3715
E-Mail: info@hilscher.kr

Switzerland
Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support
Phone: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA
Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-Mail: info@hilscher.us

Support
Phone: +1 630-505-5301
E-Mail: us.support@hilscher.com