Operating Instruction Manual

Configuration of Gateway and Proxy Devices

netTAP, netBRICK and netLINK

Hilscher Gesellschaft für Systemautomation mbH

www.hilscher.com

DOC081201OI08EN | Revision 8 | English | 2016-07 | Released | Public

Table of Contents

1 Introduction

1.1 About this Manual

This manual provides information on how to set and configure the device parameters of

- a netTAP device of the NT 50 device family
- a netTAP device of the NT 100 device family
- a netTAP device of the NT 151-RE-RE device family
- a netBRICK device of the NB 100 device family
- a netLINK proxy device

using the netGateway DTM, and what can be read from the diagnosis panes.

Dialog Panes

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

Section	Subsection	Page
Settings	*Overview Settings*	111
	Driver	112
	Device Assignment	121
Configuration	*Overview Configuration*	124
	Settings	*125*
	Licenses	*127*
	Signal Mapping	*140*
	Memory Card Management	*147*
Diagnosis	*Overview Diagnosis*	*256*
	General Diagnosis	*257*
	Firmware Diagnosis	*259*

Table 1: Descriptions Dialog Panes

1.1.1 Online Help

The netGateway DTM contains an integrated online help facility.

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

1.1.2 List of Revisions

Index	Date	Version	Component	Chapter	Revision
4	2011-03-10	1.400.x.x	netGatewayDTM.dll netGatewayGUI.ocx	4.3 4.x 4.15 4.25	Section *Licenses* updated. All sections containing the parameter 'Watchdog Time' changed to 'This function is not supported by gateway or proxy devices.' Section *Configuration sercos Slave* expanded, SDDML export added. Section *Configuration ASCII* Default Value for Response Timeout is 1000.
5	2013-02-13	1.501.x.x	netGatewayDTM.dll netGatewayGUI.ocx	1.4.6 4.2 4.10.1 4.15 4.23.3.1 7	Section *Status Bar* updated. Section *Settings*: address switch added. Section *Open Modbus/TCP Parameter* updated. Section *Configuration sercos Slave* updated. Function codes 7, 8 and 23 added. Chapter *Acyclic Communication of Proxy Devices* added.
6	2014-04-02	1.501.x.x	netGatewayDTM.dll netGatewayGUI.ocx	4.9.1 4.10.1 4.10.2 4.10.3 4.23.3 4.23.2	Section *EtherNet/IP Adapter Parameter* updated. Section *Open Modbus/TCP Parameter* updated. Section *Open* Modbus/TCP Client updated. Section *Open Modbus/TCP Server* and *Modbus* RTU Slave: Modicon-based addressing type as well as zero and one-based addressing types described. Section *Modbus RTU Master* updated.
7	2015-03-30	-	-	8	Chapter *Acyclic Communication of Gateway Devices* added.
8	2016-06-29	1.1000.x.x	netGatewayDTM.dll netGatewayGUI.ocx	all 2.5	New netTAP NT 151-RE-RE device added Section *Configuration of a NT 151-RE-RE (netTAP* Real-Time Ethernet Gateway added.

1.1.3 Conventions in this Manual

Notes, operation instructions and results of operation steps are marked as follows:

Notes

Important: <important note>

Note: <note>

<note, where to find further information>

Operation Instructions

1. <instruction>
2. <instruction>

or

➢ <instruction>

Results

➯ <result>

1.2 Legal Notes

1.2.1 Copyright

The images, photographs and texts in the accompanying material (user manual, accompanying texts, documentation, etc.) are protected by German and international copyright law as well as international trade and protection provisions. You are not authorized to duplicate these in whole or in part using technical or mechanical methods (printing, photocopying or other methods), to manipulate or transfer using electronic systems without prior written consent. You are not permitted to make changes to copyright notices, markings, trademarks or ownership declarations. The included diagrams do not take the patent situation into account. The company names and product descriptions included in this document may be trademarks or brands of the respective owners and may be trademarked or patented. Any form of further use requires the explicit consent of the respective rights owner.

1.2.2 Important Notes

The user manual, accompanying texts and the documentation were created for the use of the products by qualified experts, however, errors cannot be ruled out. For this reason, no guarantee can be made and neither juristic responsibility for erroneous information nor any liability can be assumed. Descriptions, accompanying texts and documentation included in the user manual do not present a guarantee nor any information about proper use as stipulated in the contract or a warranted feature. It cannot be ruled out that the user manual, the accompanying texts and the documentation do not correspond exactly to the described features, standards or other data of the delivered product. No warranty or guarantee regarding the correctness or accuracy of the information is assumed.

We reserve the right to change our products and their specification as well as related user manuals, accompanying texts and documentation at all times and without advance notice, without obligation to report the change. Changes will be included in future manuals and do not constitute any obligations. There is no entitlement to revisions of delivered documents. The manual delivered with the product applies.

Hilscher Gesellschaft für Systemautomation mbH is not liable under any circumstances for direct, indirect, incidental or follow-on damage or loss of earnings resulting from the use of the information contained in this publication.

1.2.3 Exclusion of Liability

The software was produced and tested with utmost care by Hilscher Gesellschaft für Systemautomation mbH and is made available as is. No warranty can be assumed for the performance and flawlessness of the software for all usage conditions and cases and for the results produced when utilized by the user. Liability for any damages that may result from the use of the hardware or software or related documents, is limited to cases of intent or grossly negligent violation of significant contractual obligations. Indemnity claims for the violation of significant contractual obligations are limited to damages that are foreseeable and typical for this type of contract.

It is strictly prohibited to use the software in the following areas:

- for military purposes or in weapon systems;
- for the design, construction, maintenance or operation of nuclear facilities;
- in air traffic control systems, air traffic or air traffic communication systems;
- in life support systems;
- in systems in which failures in the software could lead to personal injury or injuries leading to death.

We inform you that the software was not developed for use in dangerous environments requiring fail-proof control mechanisms. Use of the software in such an environment occurs at your own risk. No liability is assumed for damages or losses due to unauthorized use.

1.2.4 Warranty

Although the hardware and software was developed with utmost care and tested intensively, Hilscher Gesellschaft für Systemautomation mbH does not guarantee its suitability for any purpose not confirmed in writing. It cannot be guaranteed that the hardware and software will meet your requirements, that the use of the software operates without interruption and that the software is free of errors. No guarantee is made regarding infringements, violations of patents, rights of ownership or the freedom from interference by third parties. No additional guarantees or assurances are made regarding marketability, freedom of defect of title, integration or usability for certain purposes unless they are required in accordance with the law and cannot be limited. Warranty claims are limited to the right to claim rectification.

1.2.5 Export Regulations

The delivered product (including the technical data) is subject to export or import laws as well as the associated regulations of different counters, in particular those of Germany and the USA. The software may not be exported to countries where this is prohibited by the United States Export Administration Act and its additional provisions. You are obligated to comply with the regulations at your personal responsibility. We wish to inform you that you may require permission from state authorities to export, re-export or import the product.

1.2.6 Registered Trademarks

Windows® XP, Windows® Vista, Windows® 7 and Windows® 8 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 International, 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.

1.3 About netGateway DTM

You can use the netGateway DTM to configure

- a netTAP device of the NT 50 device family as a gateway,
- a netTAP device of the NT 100 device family as a gateway,
- a netTAP device of the NT 151-RE-RE device family as a gateway,
- a netBRICK device of the NB 100 device family as a gateway,
- a netTAP device of the NT 100 device family as a proxy respectively
- a netLINK proxy device of the type NL 51N-DPL as a proxy

within a FDT Framework.

1.3.1 Requirements

System Requirements

- PC with 1 GHz processor or higher
- Windows® XP SP3, Windows® Vista (32 bit) SP2, Windows® 7 (32 bit) or Windows® 7 (64 bit)
- Administrator rights
- Internet Explorer 5.5 or higher
- RAM: min. 512 MByte, recommended 1024 MByte
- Graphic resolution: min. 1024 x 768 pixel
- Keyboard and Mouse

Note: If the project file is saved and opened again or if it is used on another PC, the system requirements must match. Particularly the DTM must be installed on the used PC.

Requirements netGateway DTM

To configure a netTAP NT 50 device, a netTAP NT 100 device, a netTAP NT 151-RE-RE device, a netBRICK NB 100 device or a netLINK proxy NL 51N-DPL device with a DTM the following requirements have to be accomplished:

- Completed hardware installation of a netTAP NT 50 device, a netTAP NT 100 device, a netTAP NT 151-RE-RE device, a netBRICK NB 100 device respectively a netLINK proxy device, where at least a base firmware has to be loaded in the device
- Installed FDT/DTM V 1.2 compliant frame application
- Loaded DTM in the Device Catalog of the FTD Framework

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

1.4 Dialog Structure of the netGateway 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 Pane** (main area on the right side),
4. **OK**, **Cancel**, **Apply**, **Help**,
5. The **Status Line** containing information e. g. the online-state of the DTM.

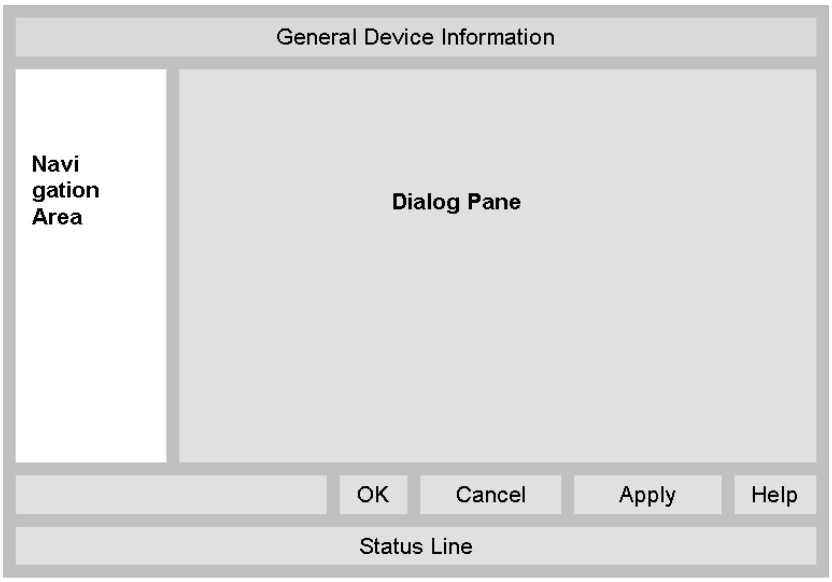


Figure 1: Dialog Structure of netGateway DTM

1.4.1 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 2: General Device Information

1.4.2 Navigation Area

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

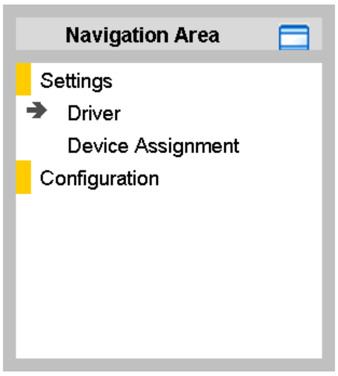


Figure 2: Navigation Area

- ➢ Select the required folder and subfolder.
- ➩ The corresponding Dialog pane is displayed.

<u>Hide / display Navigation</u>

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

1.4.3 Dialog Panes

At the dialog pane the **Settings**, **Configuration**, **Diagnosis/Extended Diagnosis** or the **Additional Tools** panes are opened via the corresponding folder in the navigation area.

Settings	
Driver	On the pane **Driver** you can select a driver from the driver list. For further information, refer to section *Driver* on page 112.
Device Assignment	On the pane **Device Assignment** you must select the device and assign the device to the driver. For further information, refer to section *Device Assignment* on page 121.
Configuration	
Setting	The **Settings** pane serves for selection of the protocols for the Gateway respectively the Proxyfunctionality. For further information, refer to section *Settings* on page *125.*
Licensing	On the pane **Licensing** you can verify which licenses are available in the netTAP NT 100 device, order licenses and transfer license files into the netTAP NT 100 device. For further information, refer to section *Licenses* on page 127*.*
Signal Mapping	On the pane **Signal Mapping** you can map the signals (I/O Data) of port X2 to the signals (I/O data) of port X3. For further information, refer to section *Signal Mapping* on page *140.*
Memory Card Management	On the pane **Memory Card Management** you can backup and restore the firmware and the configuration to respectively from an MMC card. For further information, refer to section *Memory Card* Management on page *147.*
Diagnosis	
Diagnosis	At the **Diagnosis** panes diagnosis information can be read. For further information, refer to section *Overview Diagnosis* on page 256.

Table 3: Overview Dialog Panes

Note: Accessing the **Diagnosis** panes of the netGateway DTM requires an online connection from the netGateway DTM to the device.

For further information, refer to section *Connecting/Disconnecting Device* on page 260.

1.4.4 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 4: OK, Cancel, Apply and Help

1.4.5 Table Lines

In the DTM dialog pane 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 5: Selecting, inserting, deleting Table Line

1.4.6 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.

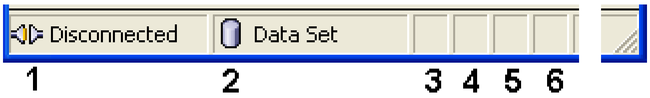


Figure 3: Status Bar – Status Fields 1 to 6

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 6: Status Bar Icons [1]

Offline State

Save operation succeeded

Firmware Download

Firmware Download successful

Online State and Diagnosis

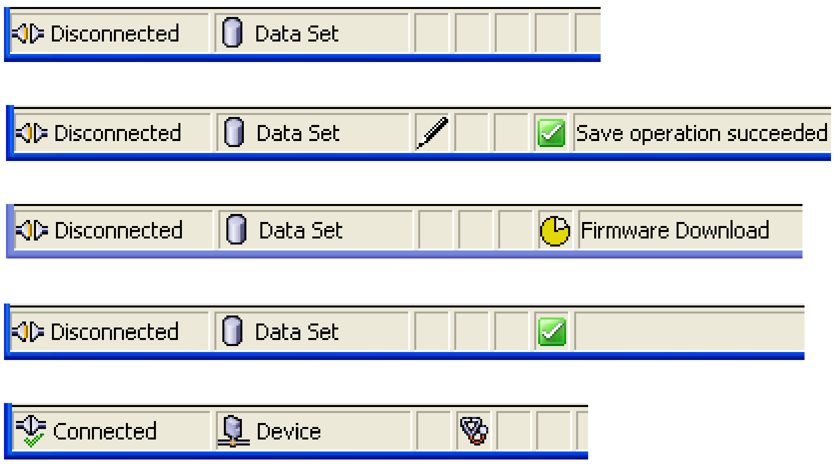


Figure 4: Status Bar Display Examples

2 Configuration Sequence for Gateway and Proxy

2.1 Gateway and Proxy Device

Two networks are connected via a gateway or a proxy device together. For the data transfer from one network into the other it is necessary that these data are mapped. This mapping is done within SYCON.net.

Gateway Devices

The mapping of the data can be done by the user and be configured within SYCON.net.

The following gateway devices are in the device catalog.

- netTAP NT 50 

- netTAP NT 100 

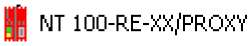
- netTAP NT 151-RE-RE 

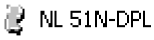
- netBRICK NB 100 

Proxy Devices

The mapping of data is already specified and done automatically by SYCON.net.

The following proxy devices are in the device catalog.

- netTAP NT 100 

- netLINK Proxy 

Note: The gateway or proxy device can only be used at the (gray) main line.

2.2 Configuration of a NT 50 as a Gateway

This section is valid for netTAP NT 50 devices as a gateway. These devices are called gateway devices in the following description.

The configuration of the NT 50 devices is explained exemplarily with the protocol conversion PROFIBUS-DP Slave to Modbus RTU Slave.

The device NT 50-DP-RS is necessary for the protocol conversion from PROFIBUS-DP Slave to Modbus RTU Slave.

The following steps have to be done to configure the device:

2.2.1 Requirement for the Configuration

The device is configured via the Ethernet port. Therefore it is necessary that the device gets an IP address assigned before.

Therefore do the followings steps:

1. Establish an Ethernet connection between the Ethernet network port of your PC and the Ethernet port of the netTAP NT 50 device
2. Start the “Ethernet-Device Setup” software. Therefore select **Start > Programs > SYCON.net System Configurator > Ethernet Device Setup.**
3. Search for connected devices. Therefore click on **Search Devices**. Devices are searched in the local network using broadcast telegrams.
4. Assign an IP address to the NT 50 device, which should be used for the device configuration.

   This address can be stored in a non volatile memory of the device.

2.2.2 Start SYCON.net and User Login

1. Start SYCON.net

➢ Select **Start > Programs > SYCON.net System Configurator > SYCON.net**

➩ SYCON.net is started

2. User Login

➢ In the window SYCON.net User Login click **OK** to login or enter your password and then click **OK** to login

➩ SYCON.net frame application appears

2.2.3 Insert the Gateway Device into the Configuration Window

➢ Go to the device catalog under vendor **Hilscher GmbH** to the category **Gateway / Stand-Alone Slave**. Use drag and drop with the NT 50-XX-XX gateway device to insert it at the (gray) main line.

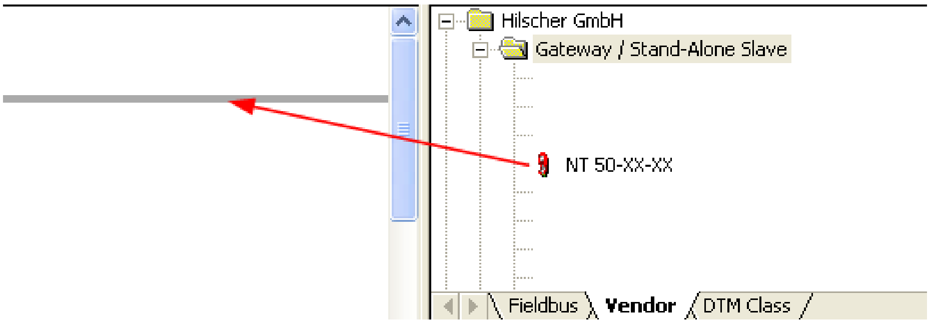


Figure 5: Insert Gateway Device into the Project

➩ The gateway device appears in the project

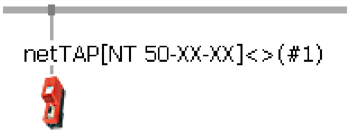


Figure 6: Gateway Device in the Project

2.2.4 Select the Protocol Conversion of the Gateway

1. Open the Gateway configuration window

➢ Double click the gateway device symbol
or
select from the context menu of the gateway device symbol the entry **Configuration > Gateway**

➩ The Gateway configuration window opens

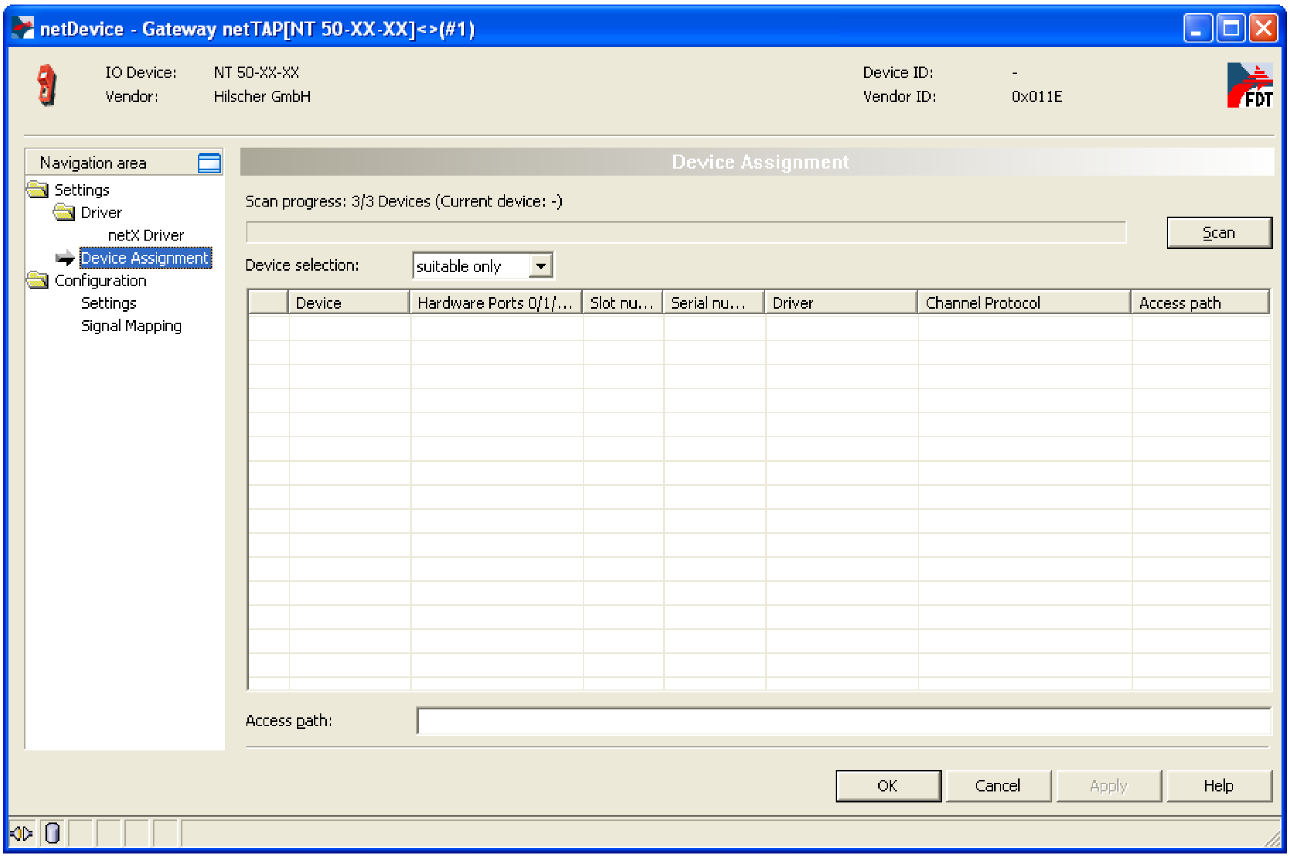


Figure 7: Device Assignment without Device

2. Open dialog for protocol selection

➢ Select in the navigation area **Configuration** > **Settings**

➩ The configuration window **Settings** opens

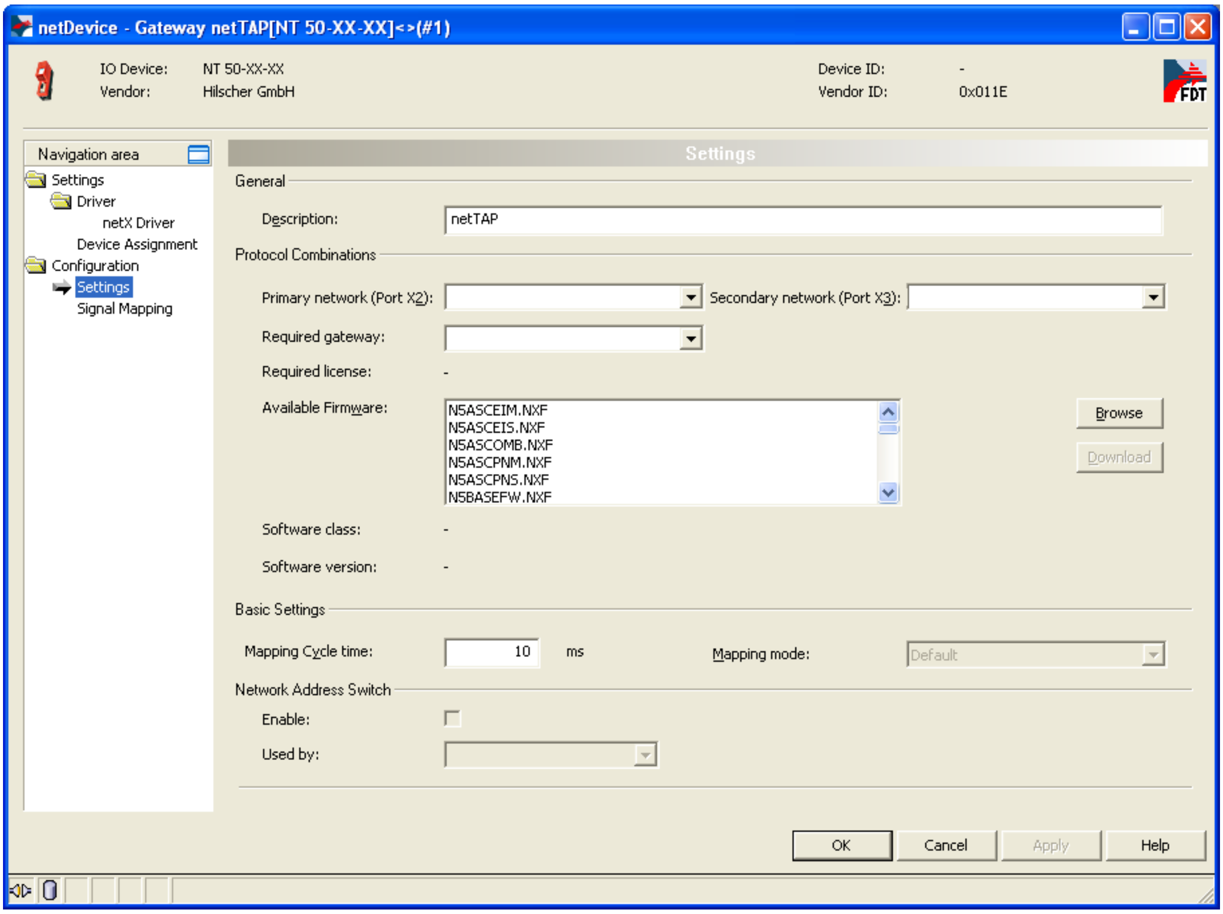


Figure 8: Gateway Protocol Selection (1)

3. Select the protocol for the primary network and for the secondary network and apply.

- Select in the window **Configuration > Settings** at **Protocol Combinations** for the **Primary Network (Port X2)** the protocol PROFIBUS-DP Slave
- Select then at **Protocol Combinations** for the **Secondary Network (Port X3)** the protocol Modbus RTU
- Click **Apply**

➾ The Gateway configuration window shows the following

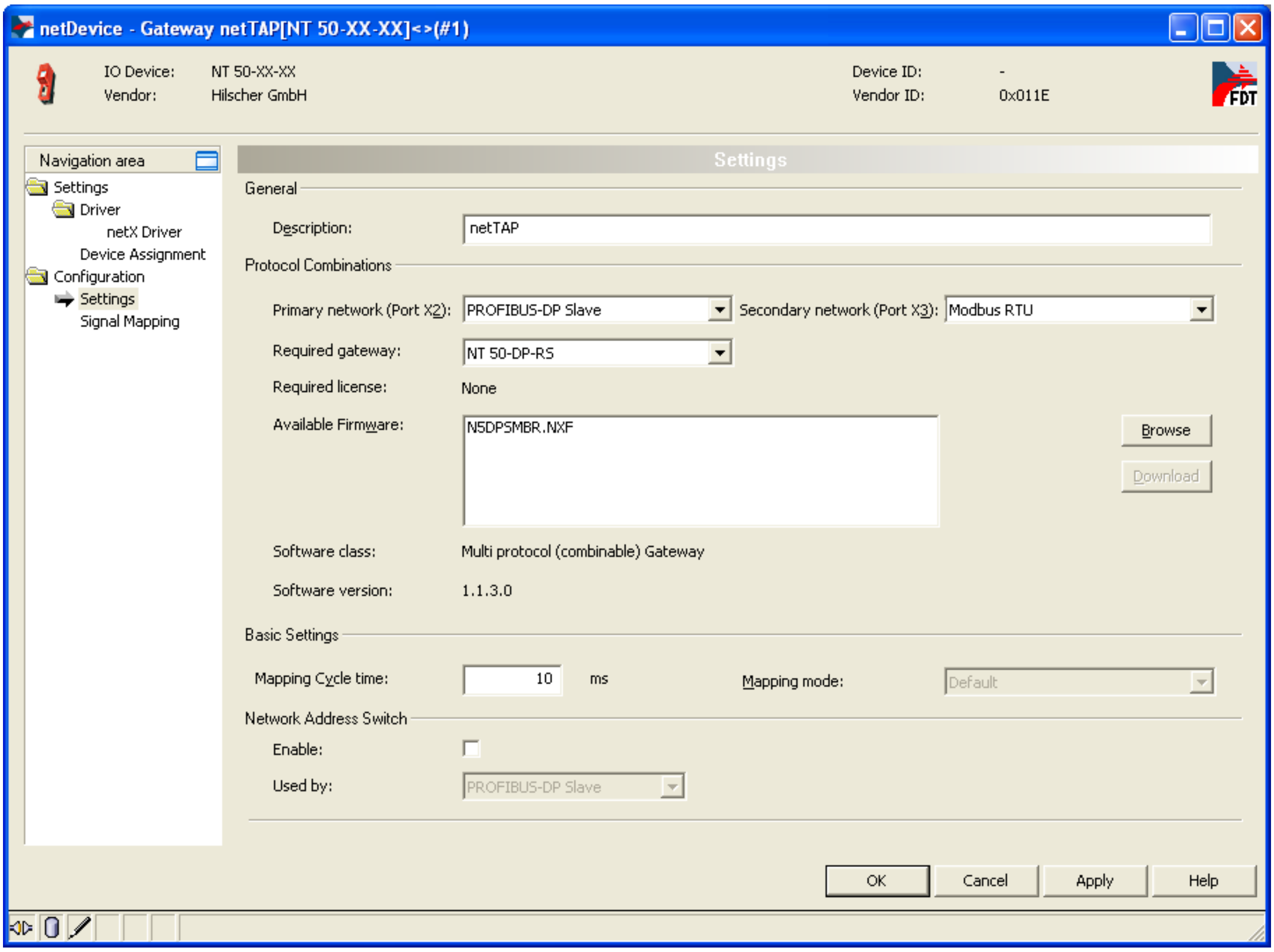


Figure 9: Gateway Protocol Selection (2)

The **Required gateway** as shown above depends on the selected protocols.

4. Enable address switch.

➢ Check **Enable** to enable that the PROFIBUS station address is set by the rotary switch at the device.
If **Enable** is uncheked, the the PROFIBUS station address is set by the configuration software.

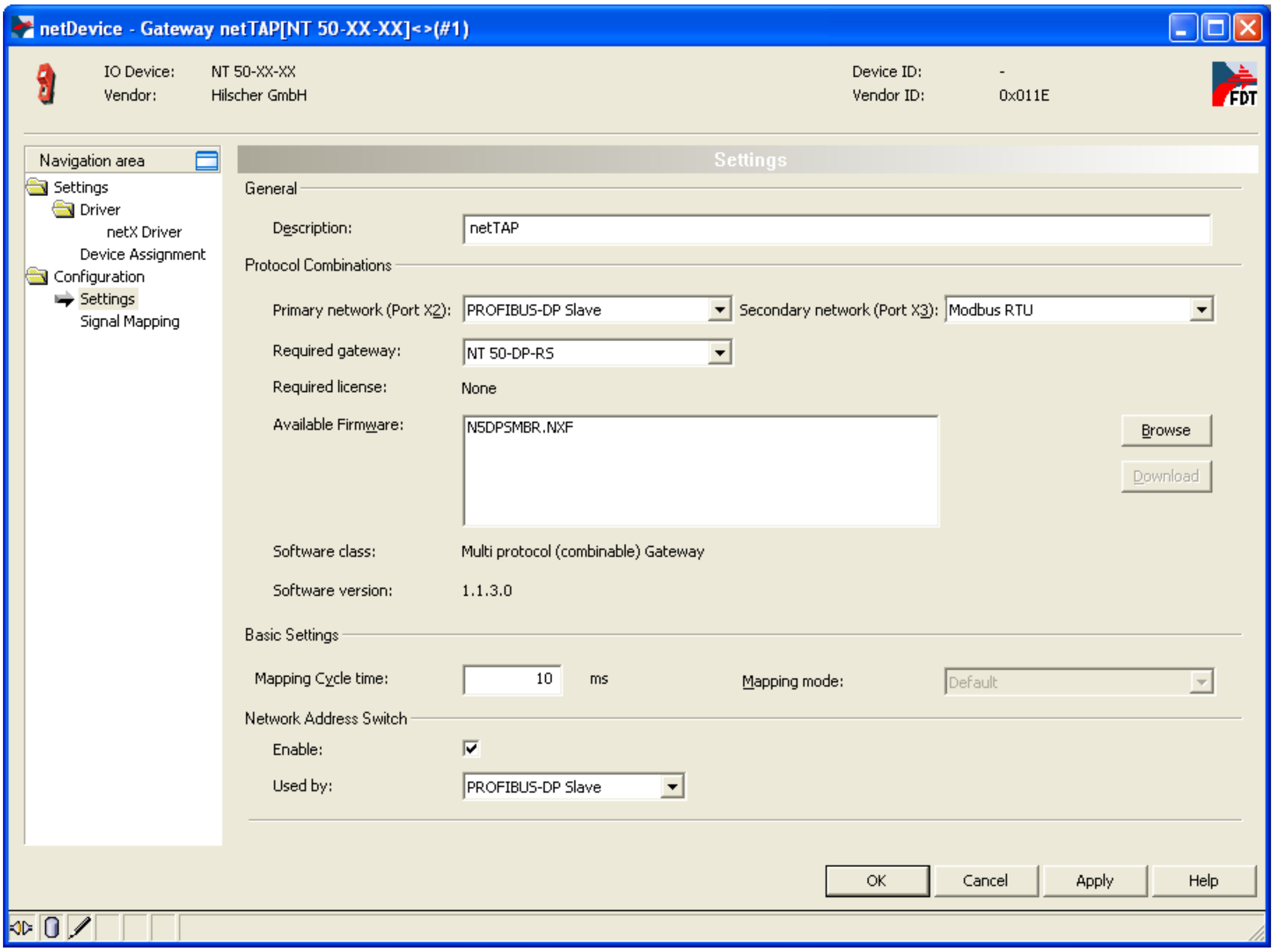


Figure 10: Gateway Protocol Selection (3)

5. Close the Gateway configuration window

➢ Click **OK**

➩ The Gateway configuration window closes

2.2.5 Configuration of the secondary Network

1. Configure the protocol at the secondary network (Port X3)

➢ Select from the context menu of the NT 50-XX-XX symbol the entry **Configuration > Modbus RTU**

➩ The Modbus RTU configuration window opens

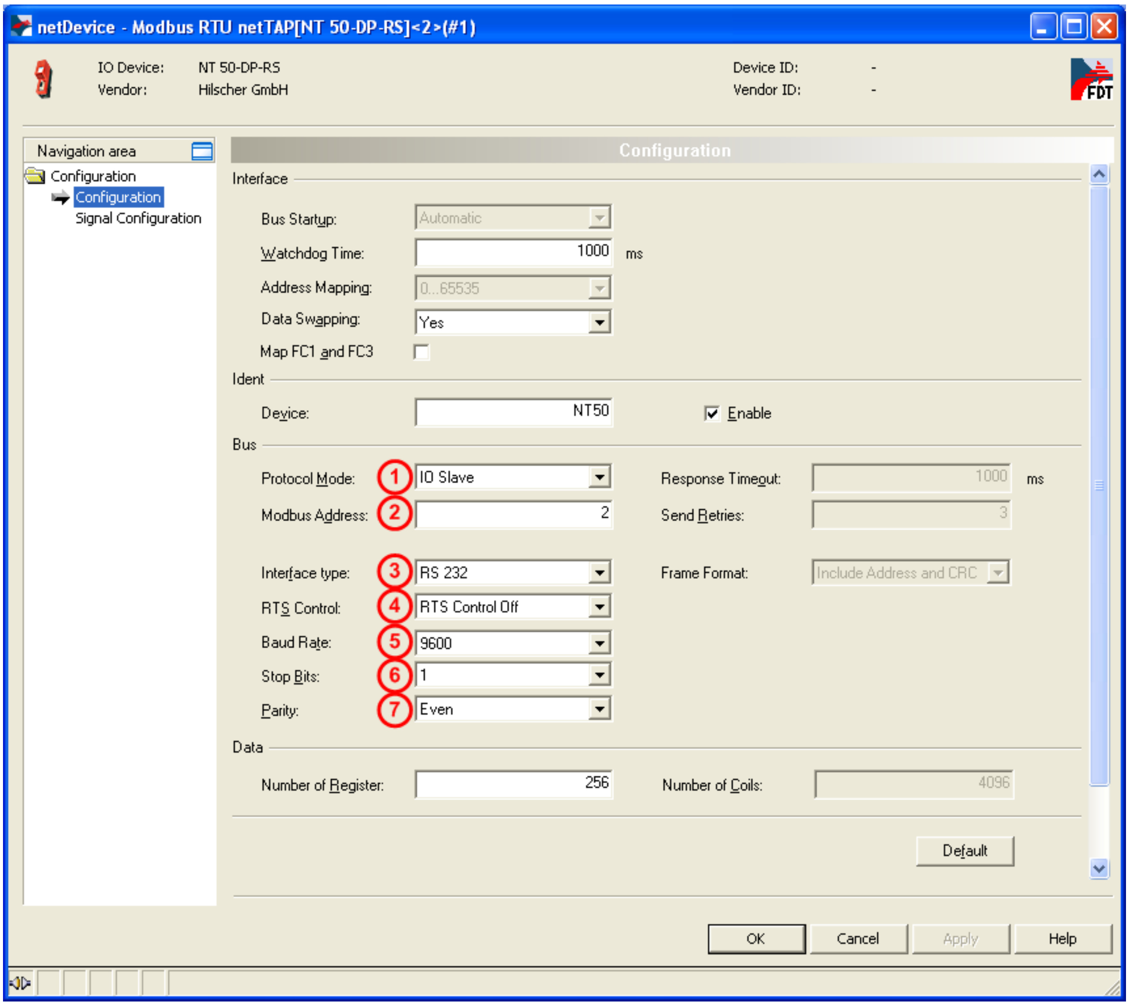


Figure 11: Protocol Configuration

2. Set protocol parameter

➢ Set the parameters. To configure the device as a Modbus RTU Slave set for ① **Protocol Mode** the value **IO Slave**. Set the following important parameter: ② **Modbus Address**, ③ **Interface type**, ④ **RTS Control**, ⑤ **Baud Rate**, ⑧ **Stop Bits** and ⑦ **Parity**. These parameter are important that the used Modbus RTU Master can communicate with the gateway device.

More information about Modbus RTU parameter is in section *Modbus RTU Parameter* on page 219.

3. Close the configuration window

➢ Click **OK**

➩ The configuration window closes

2.2.6 Configuration of the primary Network

1. Configure the protocol at the primary network (Port X2)

➢ Select from the context menu of the NT 50-XX-XX symbol the entry **Configuration > PROFIBUS-DP Slave**

➩ The PROFIBUS-DP Slave configuration window opens

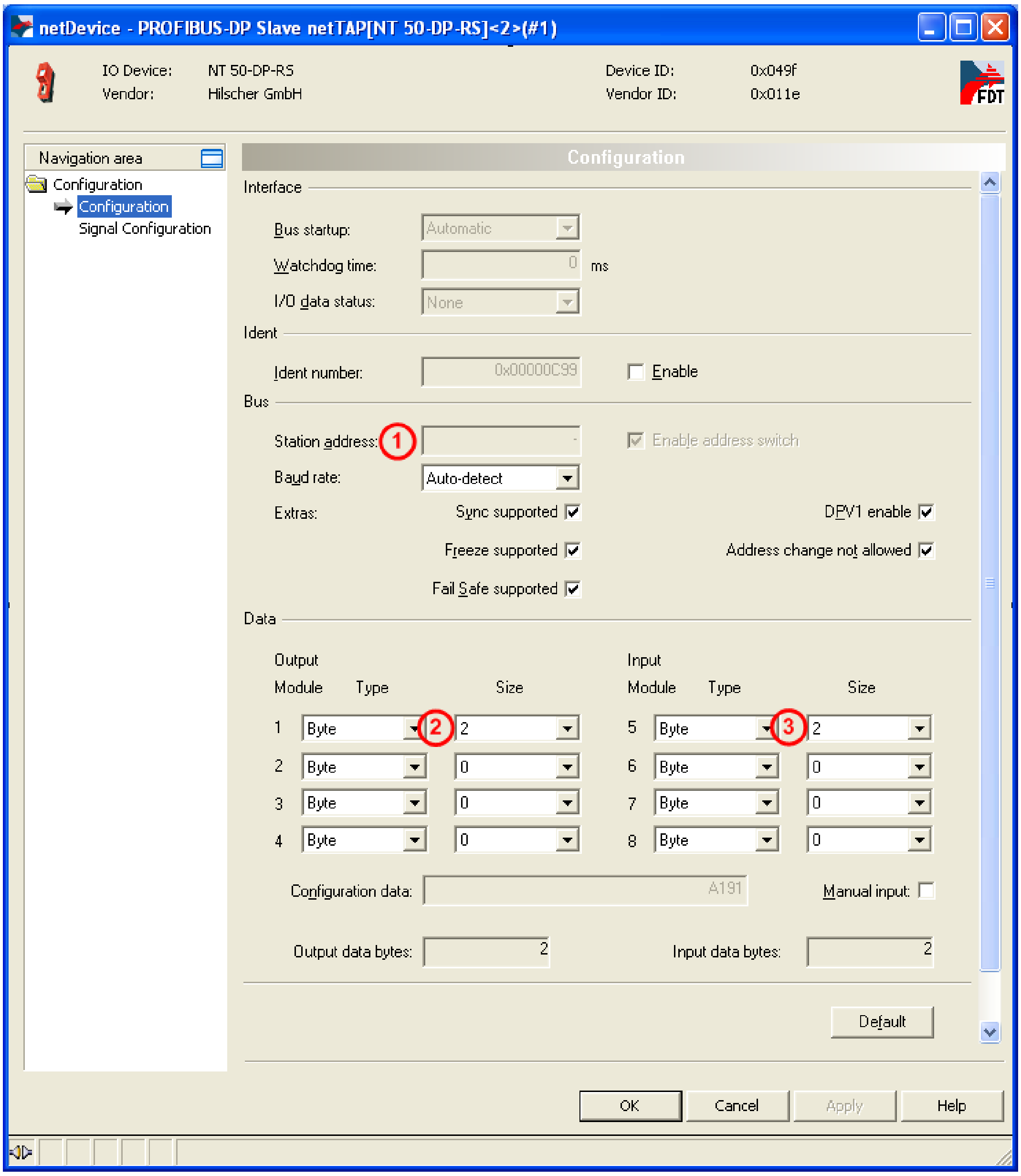


Figure 12: Protocol Configuration

2. Set parameter for protocol at the primary network (port X2).

➢ If the PROFIBUS station address is set by the rotary switch at the device, then the field **Station Address** 1 is grayed out.
If the PROFIBUS station address is set by configuration software, then enter in **Station Address** 1 the PROFIBUS station address for the gateway device.

➢ Set size and type of the modules for **Output** 2 and **Input** 3.

- Default values can be used normally for the other parameters. Depending on additionally requirements and the design of the PROFIBUS network as well as the used devices parameters needs to be adjusted if necessary.

  More information about PROFIBUS-DP Slave parameter is in section *PROFIBUS-DP Slave Parameter* on page 214.

3. Close the configuration window.

- Click **OK.**

- The configuration window closes.

2.2.7 Configure the Gateway – Signal Mapping

1. Open the Gateway configuration window

- Select from the context menu of the gateway device symbol the entry **Configuration > Gateway**

- The Gateway configuration window opens

2. Open the Signal Mapping window

- Select **Configuration > Signal Mapping**

- The window Signal Mapping opens

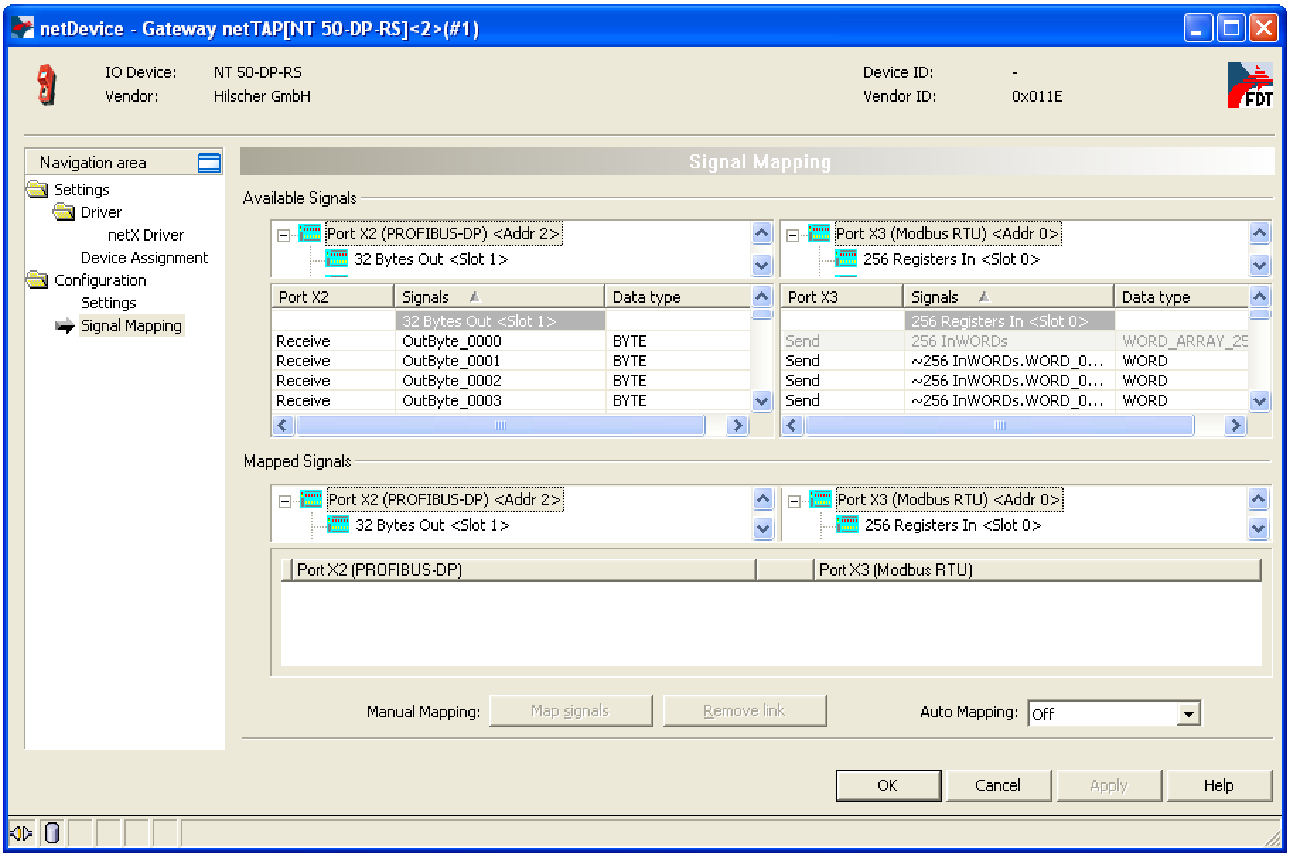


Figure 13: Gateway Signal Mapping

3. Signal Mapping: Data transfer from Port X2 to Port X3

- Map the signals, which are received on Port X2 (Port X2 receive), with signals, which should be sent on Port X3 (Port X3 send).
- For this, mark the signal received (Port X2) and the signal to be sent (Port X3) and the click the button **Map Signals**
  Or
  Map the signal using drag and drop. For this, drag the signal received (Port X2 receive) and drop it on the signal to be sent (Port X3 send)

4. Signal Mapping: Data transfer from Port X3 to Port X2

- Map the signals, which are received on Port X3 (Port X3 receive), with signals, which should be sent on Port X2 (Port X2 send).
- For this, mark the signal received (Port X3) and the signal to be sent (Port X2) and the click the button **Map Signals**
  Or
  Map the signal using drag and drop. For this, drag the signal received (Port X3 receive) and drop it on the signal to be sent (Port X2 send)

5. Signal Mapping: Transfer status info of Port X2 to Port X3

- If necessary, map the status information of Port X2 (Port X2 generated, which are generated device internal) to signals which should be sent on Port X3 (Port X3 send)
- For this mark the status signal (Port X2) and the signal which should be sent (Port X3) and click the button **Map Signals**
  Or
  Map the signal using drag and drop. For this, drag the status signal (Port X2 generated) and drop it on the signal to be sent (Port X3 send)

6. Signal Mapping: Transfer status info of Port X3 to Port X2

- ➢ If necessary, map the status information of Port X3 (Port X3 generated, which are generated device internal) to signals which should be sent on Port X2 (Port X2 send)
- ➢ For this mark the status signal (Port X3) and the signal which should be sent (Port X2) and click the button **Map Signals**
  Or
  Map the signal using drag and drop. For this, drag the status signal (Port X3 generated) and drop it on the signal to be sent (Port X2 send)

➾ An example of the Signal Mapping window shows the following figure

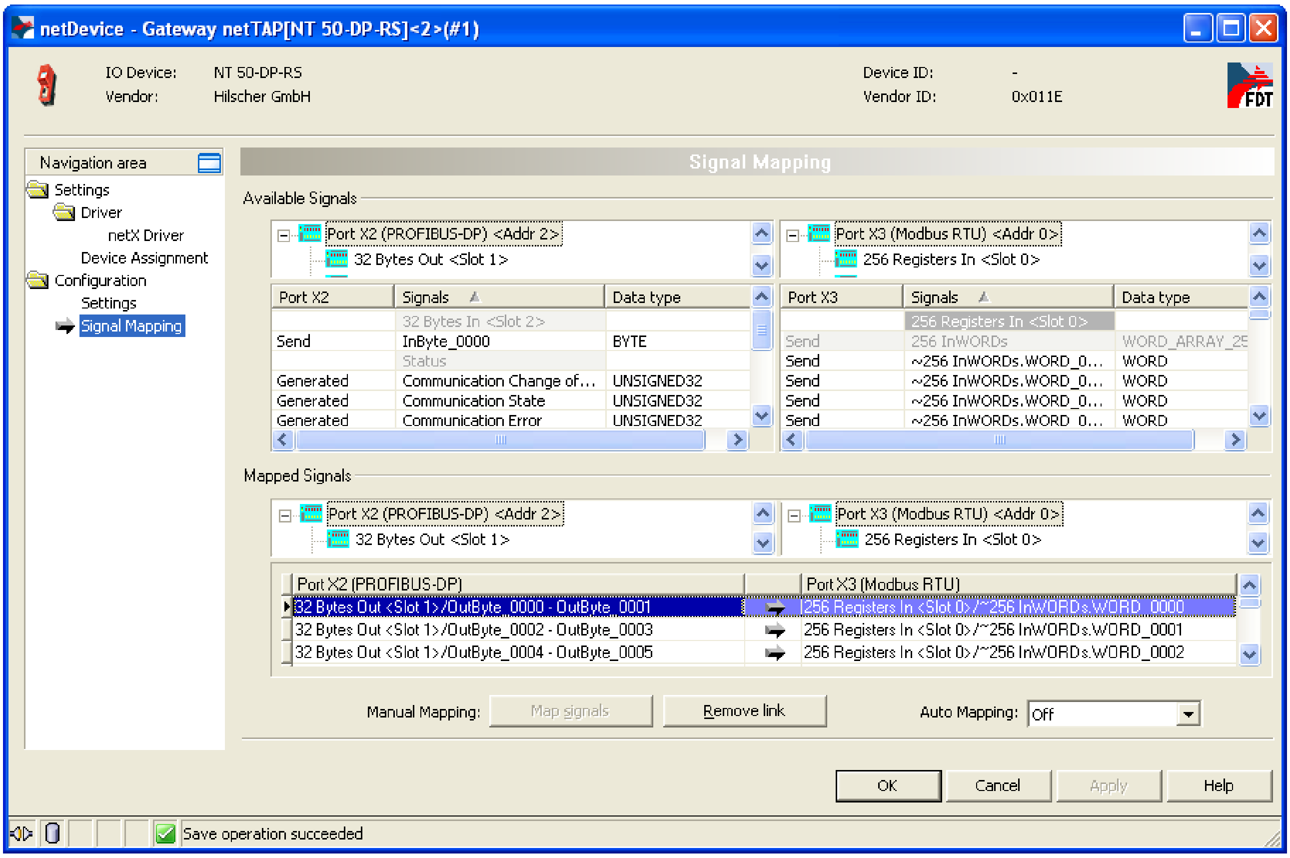


Figure 14: Gateway Signals mapped

2.2.8 Establish a Connection to the netTAP NT 50

1. Connect an Ethernet cable to the NT 50

➢ Build up an Ethernet connection between the Ethernet port of the NT 50 device and to the Ethernet port of the configuration PC.

2. Set the IP address of the NT 50 device

➢ Set with the Ethernet Device setup software, which is part of the SYCON.net installation, the IP address for the NT 50. With this software it is also possible to read an already assigned IP address from the NT 50 device.

The IP address is required for the communication with the SYCON.net software.

To communication from the SYCON.net software with the NT 50 device, an IP address has to be assigned before to the NT 50 device. How to assign an IP address is described in the document „Ethernet Device Configuration OI xx DE.pdf".

---

3. Open the gateway configuration window

➢ Select from the context menu of the gateway device symbol **Configuration > Gateway**

➮ The gateway configuration window opens

4. Select driver

➢ Select in the navigation area **Settings > Driver** and then check **netX Driver**.

➮ The following figure shows the selected driver.

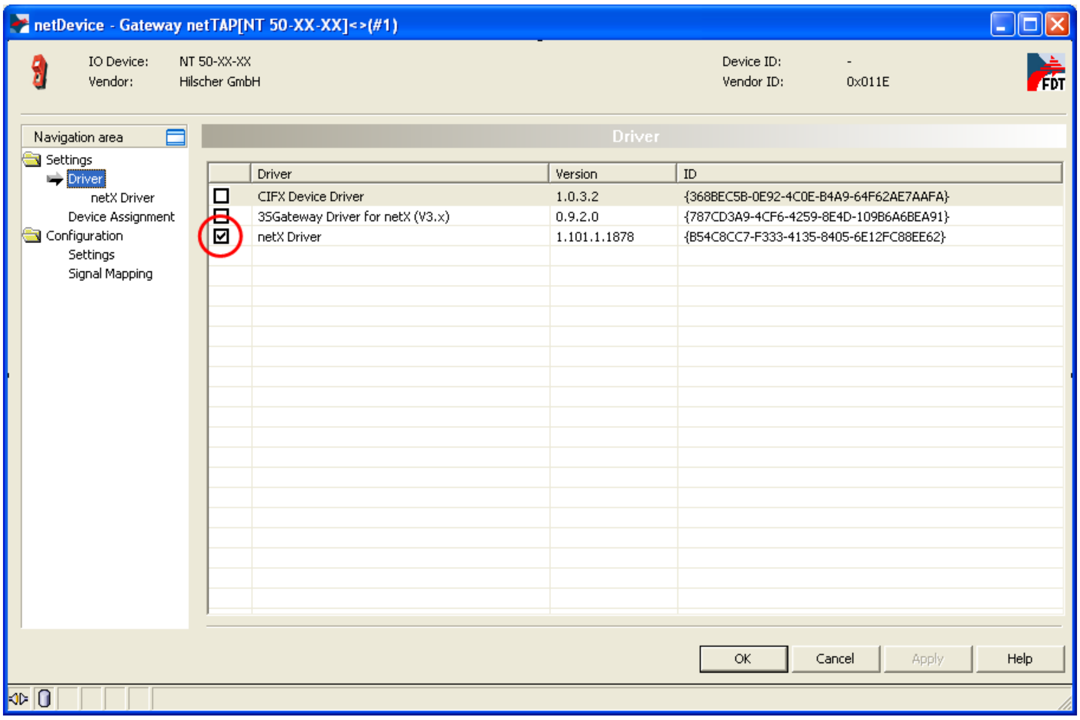


Figure 15: Select Driver

5. Set the IP search range

➢ Select in the navigation area **Settings > Driver > netX Driver**.

➯ The window for driver settings opens.

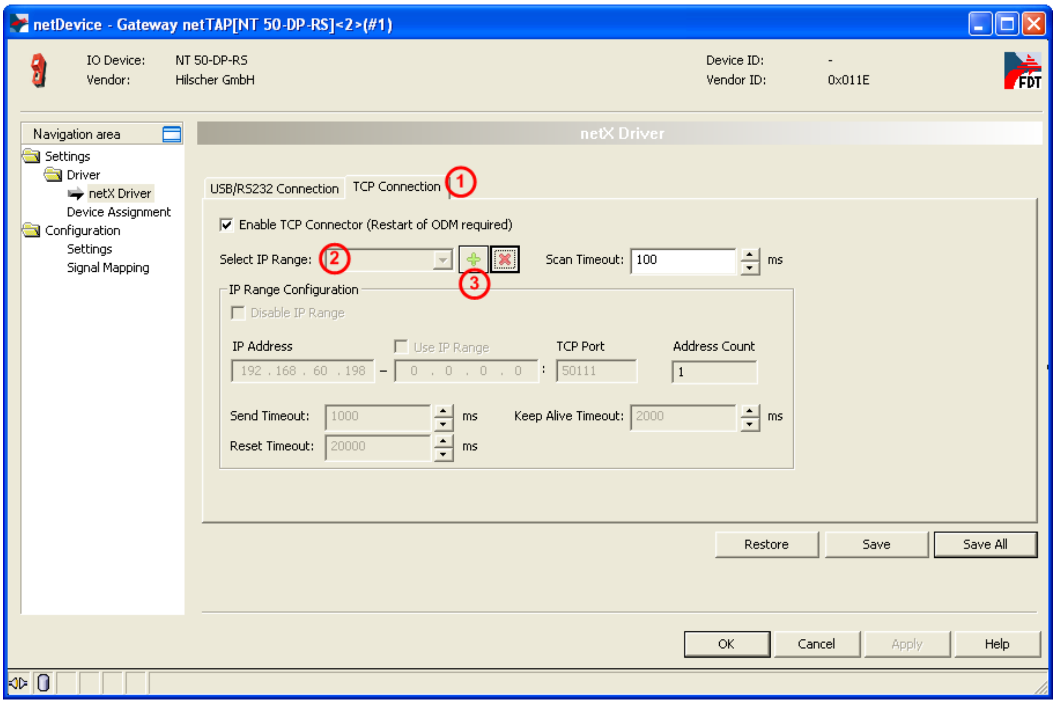


Figure 16: Set IP Address Search Range (1)

An IP address (one) or IP addresses (several, address range) are set in this window which SYCON.net uses to build up a connection to the proxy device.

➢ Select (as shown in the figure above):

① Select the tap **TCP Connection**.

② Is the **Select IP Range** grayed out (as shown above), then no IP range is defined.

Click on ③ **+** to add a new range, which enables the field below. Otherwise an address range configured earlier can be selected or by a mouse click on ③ **+** a new address range can be configured. Set the IP (start) address (and end address) the proxy devices are searched for. Make sure that the proxy device can be reached via Ethernet and is reachable via the address range.

➯ The fields to configure an IP address range are enabled.

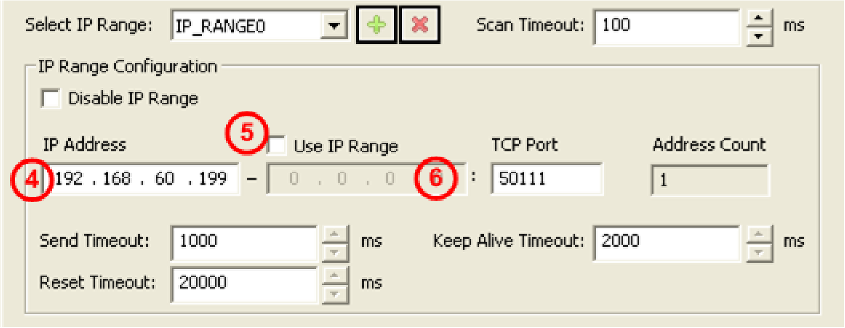


Figure 17: Set IP Address Search Range (2)

- You can enter one IP address of the proxy device at (4) as shown above or you can set an IP address range by a check at (5) **Use IP Range** and set the IP end address at (6). Make sure that the proxy device can be reached via Ethernet and is reachable via the address range.

Note: Set the IP address range not to large, which results in a long scan time.

- After the address range was set click on **Save**.

2.2.9 Device Assignment

- Select **Settings > Device Assignment**
- Click on ① **Scan**.
- The search process is started. Devices found are display in a list then.

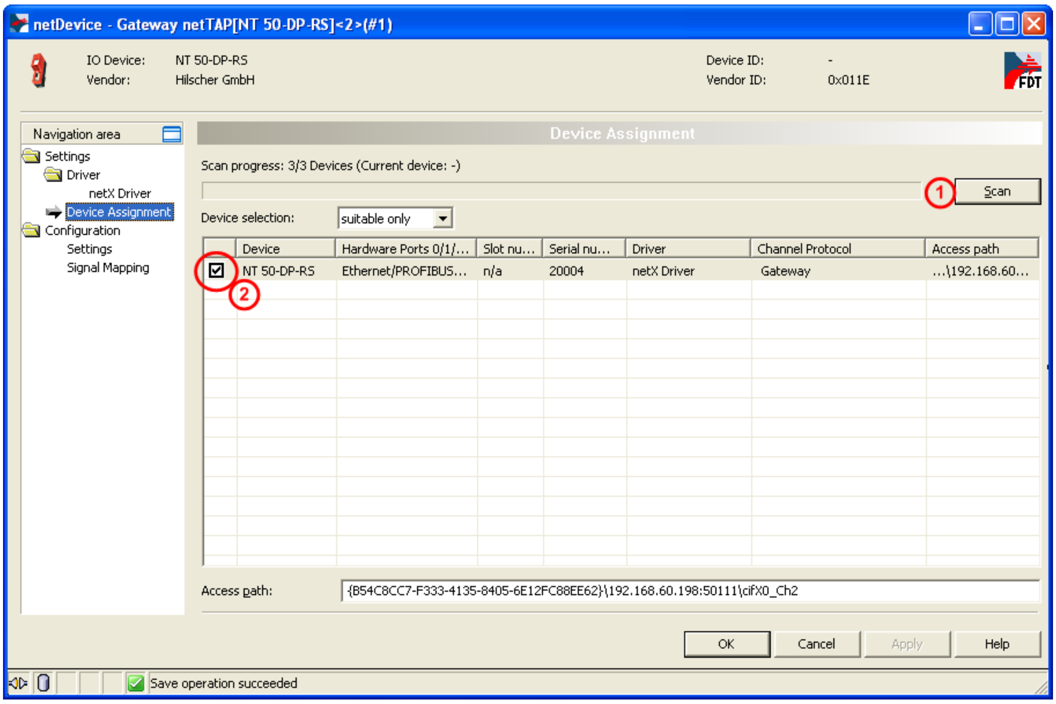


Figure 18: Select Device

- Select the device from the list ② by a check in the field of the device as shown above.
- Click on the button **OK**
- The configuration window closes.

2.2.10 Load Firmware

The firmware has to be loaded into the netTAP only for first commissioning only. Therefore do the following steps:

1. Open the Gateway configuration window

➢ Select from the context menu of the gateway device symbol the entry **Configuration > Gateway**

➩ The Gateway configuration window opens

2. Open the Settings window

➢ Select **Configuration > Settings**

➩ The window Settings opens

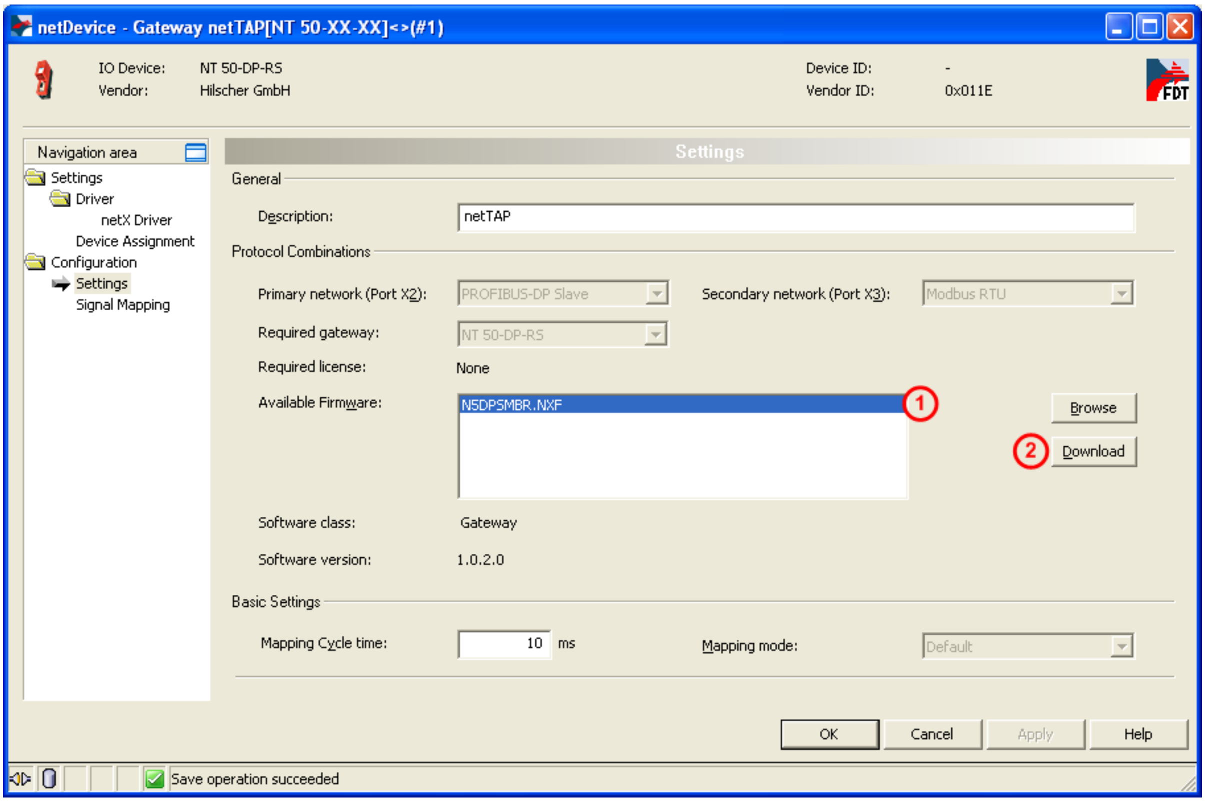


Figure 19: Firmware Download

3. Select Firmware

➢ Mark at **Available Firmware** the firmware file ①: N5DPSMBR.NXF (Firmware for PROFIBUS-DP Slave to Modbus RTU).

4. Load firmware into the device

➢ Click on **Download**

➩ The firmware is loaded into the netTAP device

Important: Do <u>not</u> remove the cable during the firmware download. Do <u>not</u> disconnect the power supply of the device during the firmware download.

5. Download of firmware

➢ Wait until the firmware was transferred completely into the device.

6. Close the Gateway configuration window after the download

➢ Click **OK**

➩ The Gateway configuration window closes

2.2.11 Load Configuration

1. Download configuration

➢ Select from the context menu of the gateway device symbol the entry **Download**

➢ Answer the security question with **Yes**, if the download should start

➩ SYCON.net builds up an online connection to the device. This the indicated by a green highlighted device name.

➩ The configuration is transferred info the gateway device

➩ The device performs a reset and then starts with the new configuration.

Note: The device performs a reset after the download. Because of that, the Ethernet connection gets lost and (has to be stopped from SYCON.net and then) has to be established again.

Stop the online connection to the device, which was established automatically to the device before.

2. Disconnect

➢ Select from the context menu of the proxy device symbol the entry **Disconnect**.

Note: To establish a connection again, do the steps described in section *Device Assignment* on page 34.

2.2.12 Save Project

Note: The configuration downloaded from SYCON.net into the device can't be uploaded from the device. Only the SYCON.net project can be downloaded into additional devices.

Save the SYCON.net project. In case of a device replacement the saved project can be opened with SYCON.net and loaded into the device.

- To save a project select menu **File > Save** respectively **File > Save As** or click on icon .

When you exit the program and the current configuration differs from the last saved configuration, then the following question appears:

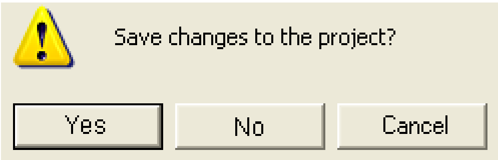


Figure 20: Security Question – Save Project

When you answer with **Yes**, then the project is saved. When you answer with **No**, then the project is not saved and the changes are lost. When you answer with **Cancel**, then the project is not saved.

2.3 Configuration of a NT 100 or NB 100 as a Gateway

This section is valid for netTAP NT 100 and netBRICK NB 100 devices as a gateway. These devices are called gateway devices in the following description.

The configuration of a netTAP NT 100 device is described as example. Differences and extras to other devices are pointed out respectively are described in own sections.

The configuration of the NT 100 devices is explained exemplarily with the protocol conversion PROFINET IO Device to PROFIBUS-DP Master.

The device NT 100-RE-DP with one master license is necessary for the protocol conversion from PROFINET IO Device to PROFIBUS-DP Master.

The following steps have to be done to configure the device:

2.3.1 Start SYCON.net and User Login

1. Start SYCON.net

➢ Select **Start > Programs > SYCON.net System Configurator > SYCON.net**

➯ SYCON.net is started

2. User Login

➢ In the window SYCON.net User Login click **OK** to login or enter your password and then click **OK** to login

➯ SYCON.net frame application appears

2.3.2 Insert the Gateway Device into the Configuration Window

- Go to the device catalog under vendor **Hilscher GmbH** to the category **Gateway / Stand-Alone Slave**. Use drag and drop with the NT 100-XX-XX respectively NB 100-XX-XX gateway device to insert it at the (gray) main line.

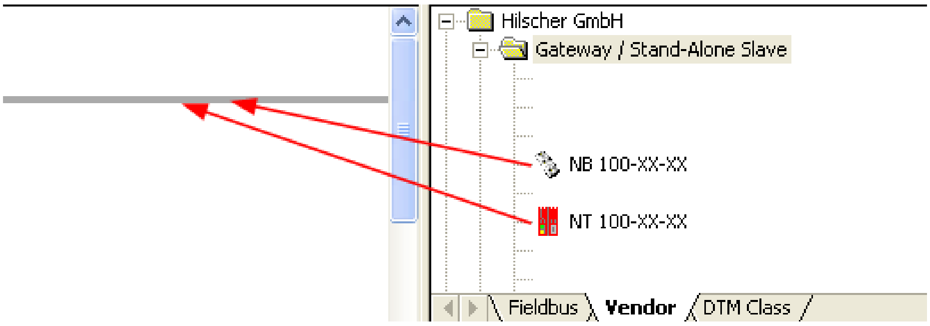


Figure 21: Insert Gateway Device into the Project

The gateway device appears in the project

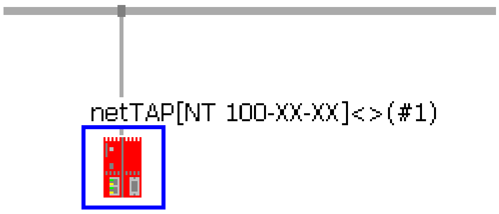


Figure 22: Gateway Device in the Project

NT 100 is shown as example.

2.3.3 Select the Protocol Conversion of the Gateway

1. Open the Gateway configuration window

➢ Select from the context menu of the gateway device symbol the entry **Configuration > Gateway**

➩ The Gateway configuration window opens

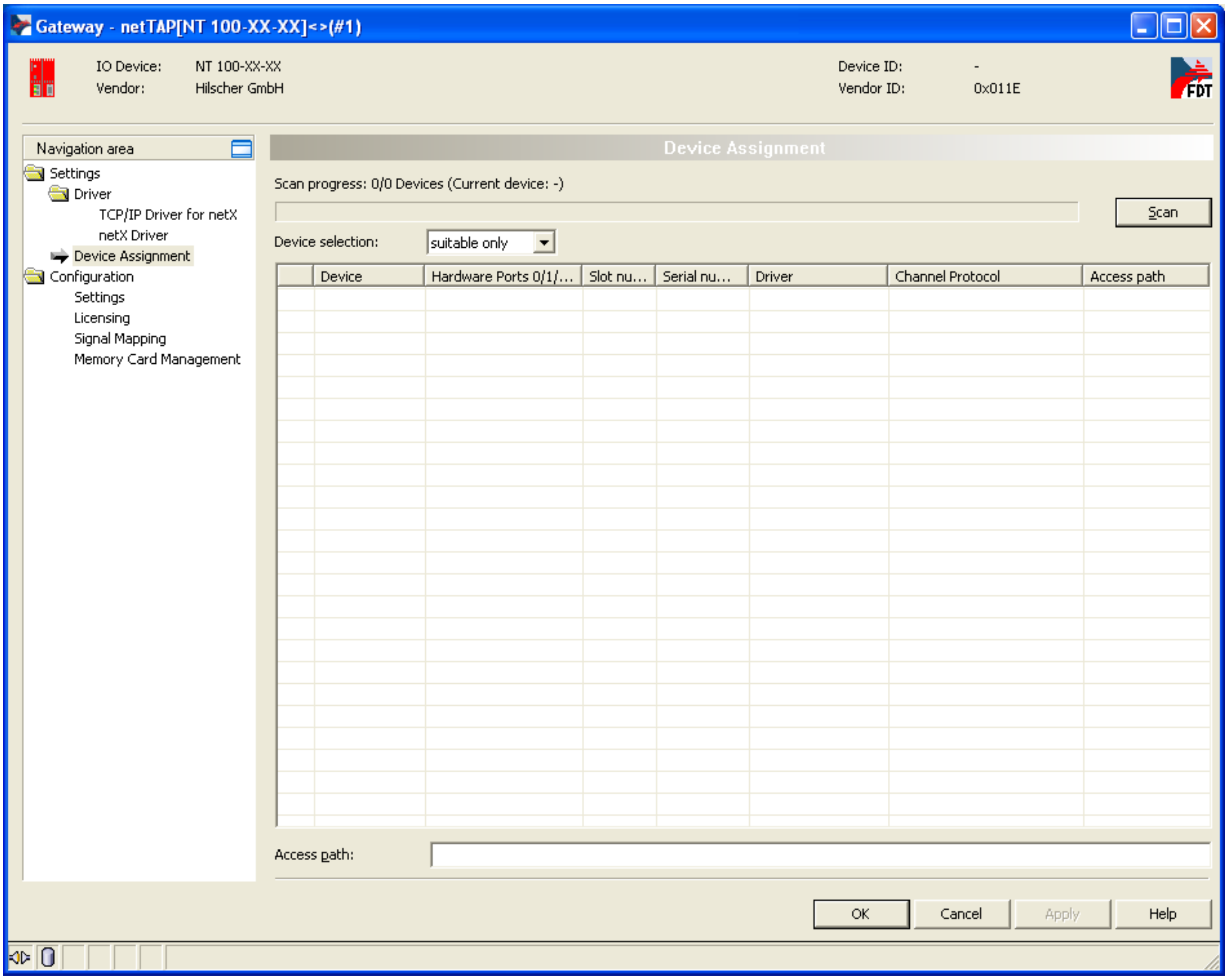


Figure 23: Device Assignment without Device

The entry **Memory Card Management** in the navigation area is only available for netTAP NT 100 devices.

2. Open dialog for protocol selection

➢ Select in the Navigation area **Configuration** > **Settings**

➲ The configuration window **Settings** opens

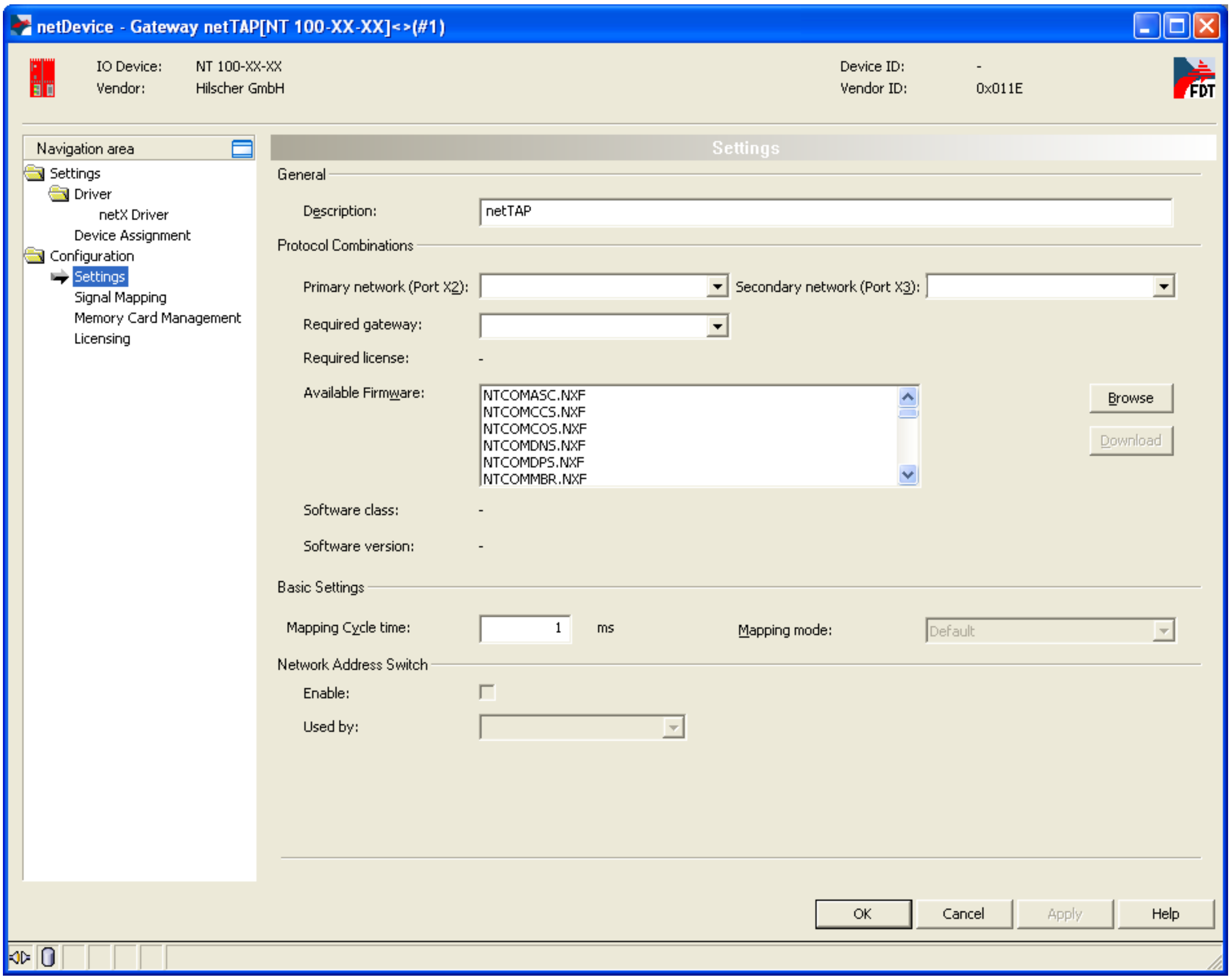


Figure 24: Gateway Protocol Selection (1)

The entry **Memory Card Management** in the navigation area is only available for netTAP NT 100 devices.

3. Select the protocol for the primary network and for the secondary network and apply

➢ Select in the window **Configuration > Settings** at **Protocol Combinations** for the **Primary Network (Port X2)** the protocol PROFINET IO Device

➢ Select then at **Protocol Combinations** for the **Secondary Network (Port X3)** the protocol PROFIBUS-DP Master

➢ Click the button **Apply**

➪ The Gateway configuration window shows the following

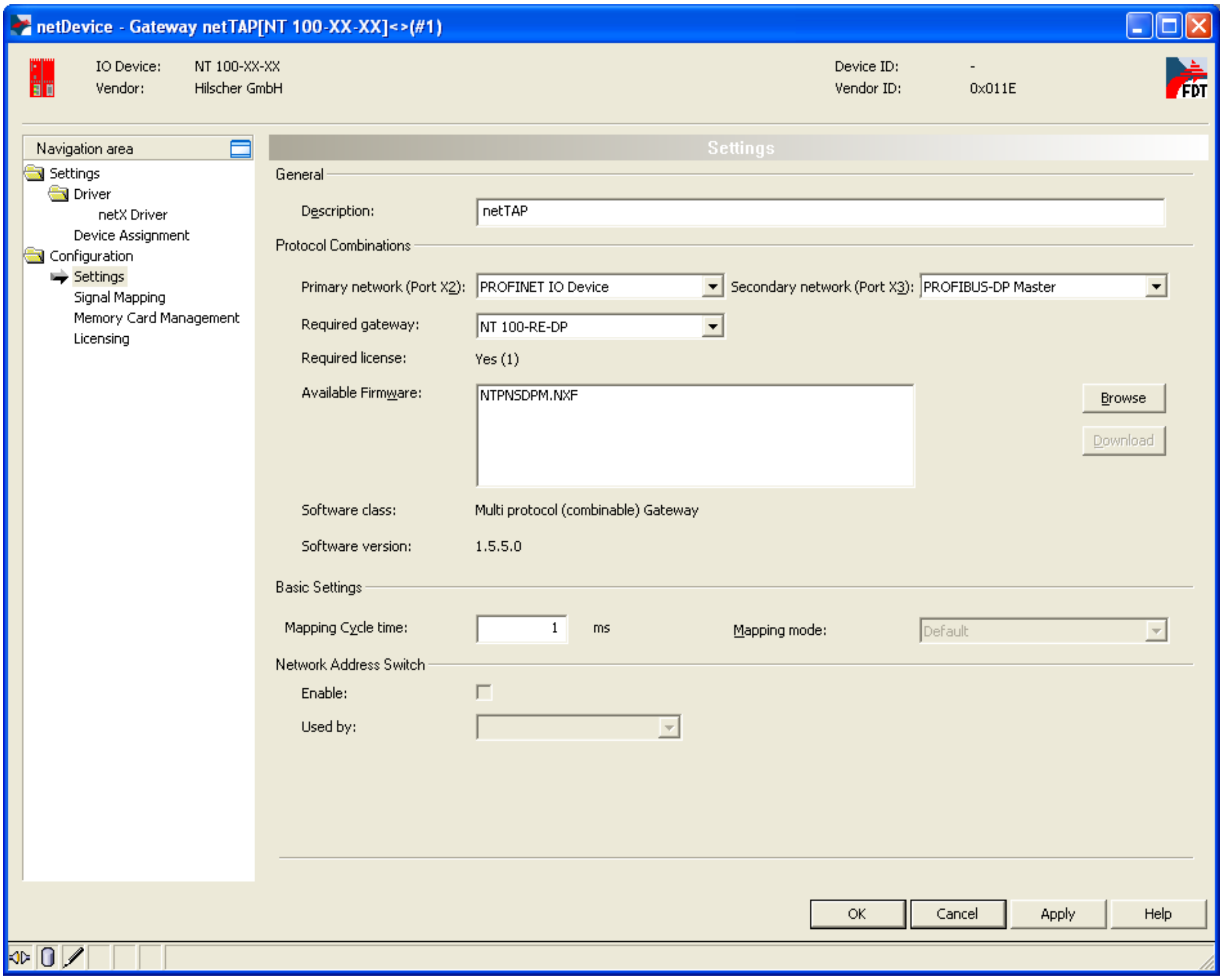


Figure 25: Gateway Protocol Selection (2)

The entry **Memory Card Management** in the navigation area is only available for netTAP NT 100 devices.

The **Required gateway** as shown above depends on the selected protocols.

4. Close the Gateway configuration window

➢ Click on the button **OK**

➪ The Gateway configuration window closes

2.3.4 Configuration of the primary Network

1. Configure the protocol at the primary network (Port X2)

➢ Select from the context menu of the gateway device symbol the entry **Configuration > PROFINET IO Device**

➯ The PROFINET IO Device configuration window opens

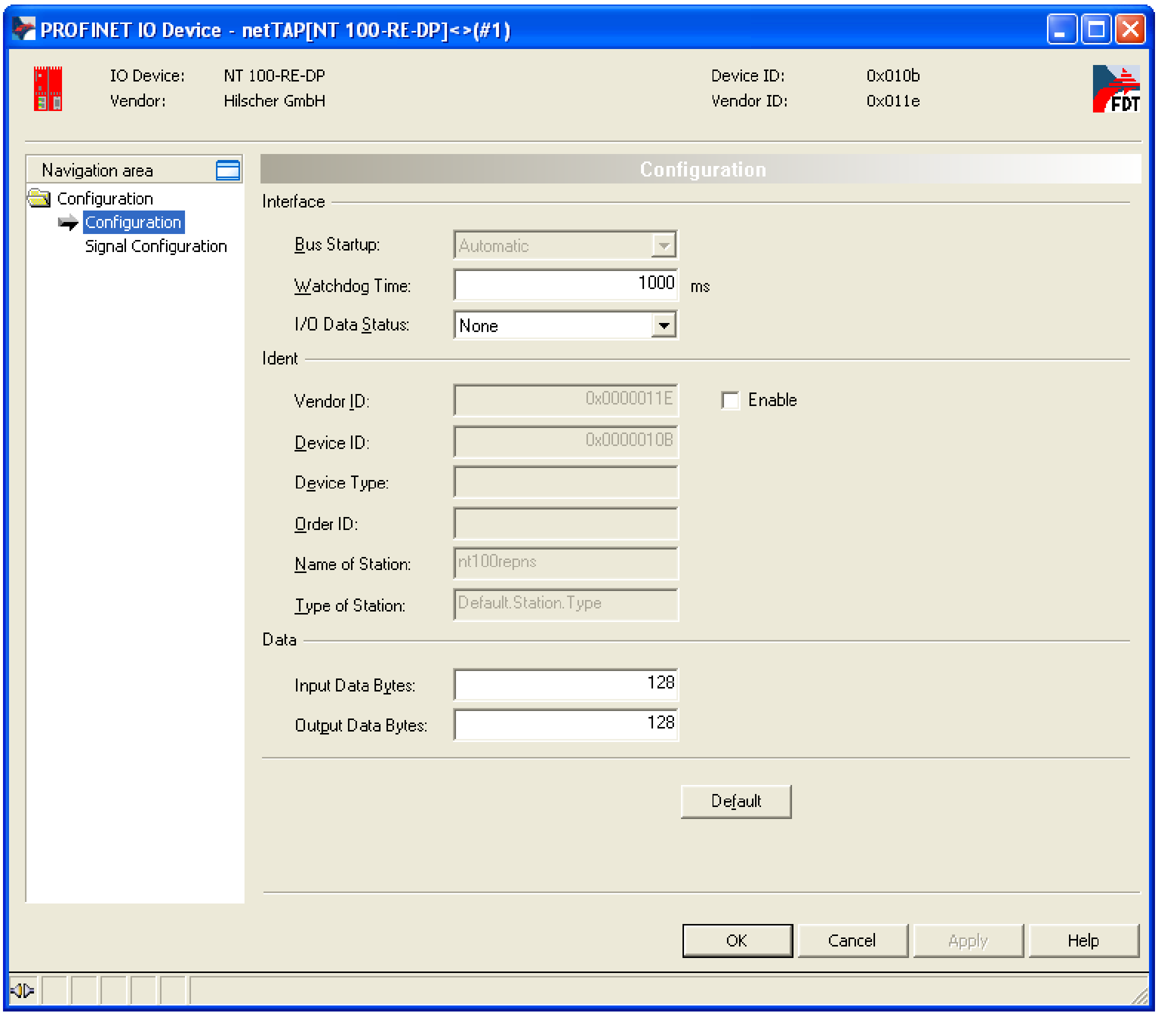


Figure 26: Protocol Configuration

➢ Set the parameters. Set especially the number of Input Data Bytes and Output Data Bytes.

More information about PROFINET IO Device parameter is in section *Configuration PROFINET IO Device (Gateway)* on page 179.

2. Close the configuration window

➢ Click on **OK**

➯ The configuration window closes

2.3.5 Configuration of the secondary Network

If the slave device is not listed in the device catalog, then it has to be imported into the device catalog first.

2.3.5.1 Expand PROFIBUS-DP Slave Device Catalog

If you want to use a PROFIBUS-DP slave device that is not listed in the device catalog, then you have to import the GSD file into the SYCON.net device catalog.

➢ Proceed as follows:

1. Missing PROFIBUS-DP Slave devices can be added to the device catalog using the menu **Network > Import Device Descriptions**.
2. Select the new GSD file.
3. Answer the question “Do you want to reload the device catalog?“ with **Yes** to reload the catalog.

More information about the device catalog is in the document „SYCONnet_netDevice_en.pdf“ in the directory „Documentation“ on the product DVD.

2.3.5.2 Insert PROFIBUS-DP Slave Device to the PROFIBUS Network

➢ Go to the device catalog. Use drag and drop with one or more PROFIBUS-DP Slave devices to insert it/them at the PROFIBUS bus line.

➭ The PROFIBUS-DP Slave device icons appear at the PROFIBUS network line (Secondary network)

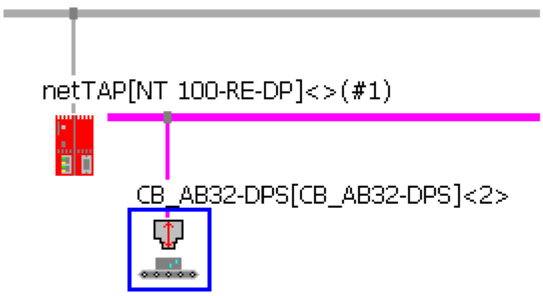


Figure 27: Gateway Device with Slave

2.3.5.3 Configure the PROFIBUS-DP Slave device

- ➢ Open the configuration window with a double click on the device icon of the PROFIBUS-DP Slave
- ➾ The configuration window of the PROFIBUS-DP Slave device opens
- ➢ Select in the navigation area **Configuration > Modules**.
- ➢ Select from **Available Modules** the module(s) and add it/them to the **Configured Modules** to configure the Slave. The **Configured Modules** has to match the configuration of the used PROFIBUS-DP slave device.

More information about the configuration of PROFIBUS-DP Slave devices are in the document PROFIBUS_Generic_Slave_DTM_en.pdf on the product DVD in section *Configuration*.

2.3.5.4 Configure PROFIBUS-DP Master

1. Open the PROFIBUS-DP Master (Port X3) configuration window

- ➢ Select from the context menu of the gateway device symbol the entry **Configuration > PROFIBUS-DP Master**
- ➾ The PROFIBUS-DP Master configuration window opens

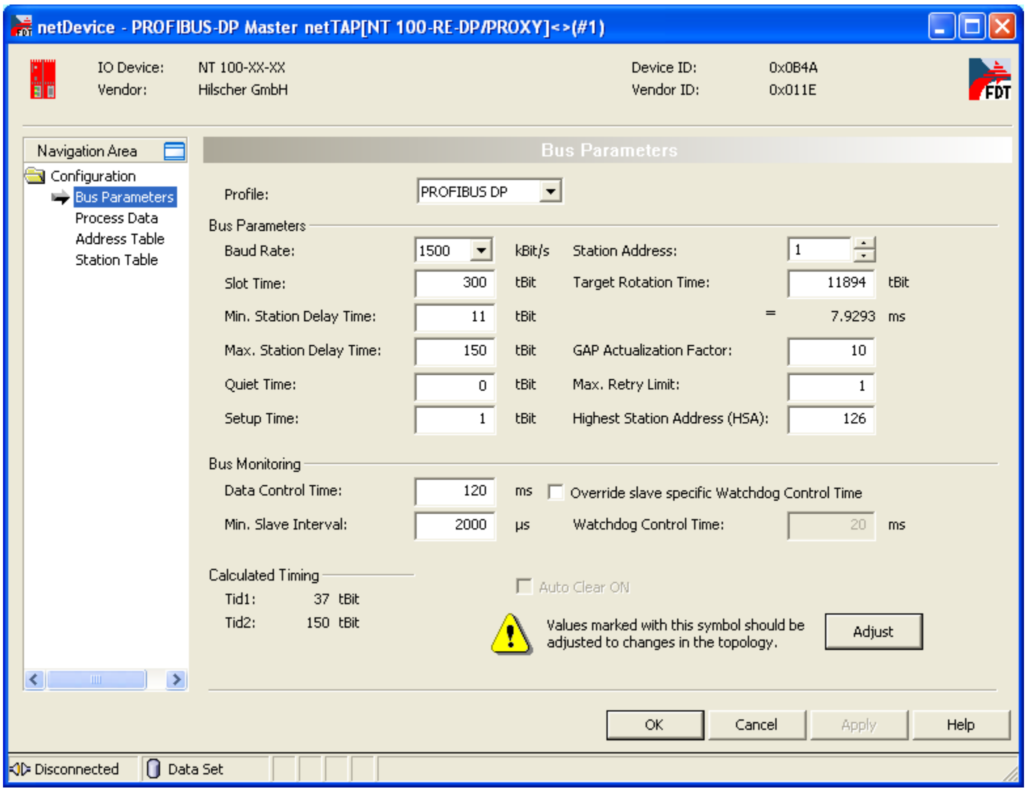


Figure 28: PROFIBUS-DP Bus Parameter

2. Configure the PROFIBUS-DP Master

➢ Set the parameter. Set especially under **Configuration > Bus Parameter** the bus parameter and under **Configuration > Stationtable** the station addresses (stations addresses of the PROFIBUS-DP Slave devices)

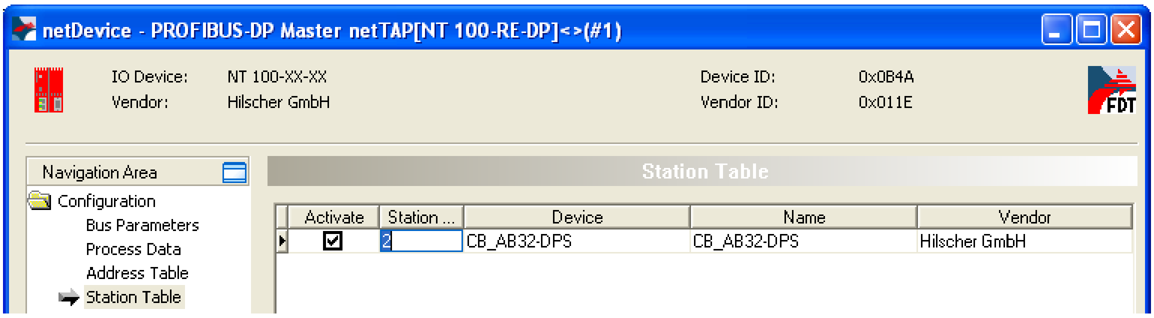


Figure 29: PROFIBUS-DP Slave Station Address

More information about the configuration of PROFIBUS-DP Master devices are in the document PROFIBUS_Master_netX_DTM_en.pdf in section *Configuration.*

3. Close the configuration window

➢ Click on the button **OK**

↳ The configuration window closes

2.3.6 Configure the Gateway – Signal Mapping

1. Open the Gateway configuration window

➢ Select from the context menu of the gateway device symbol the entry **Configuration > Gateway**

➮ The Gateway configuration window opens

2. Open the Signal Mapping window

➢ Select **Configuration > Signal Mapping**

➮ The window Signal Mapping opens

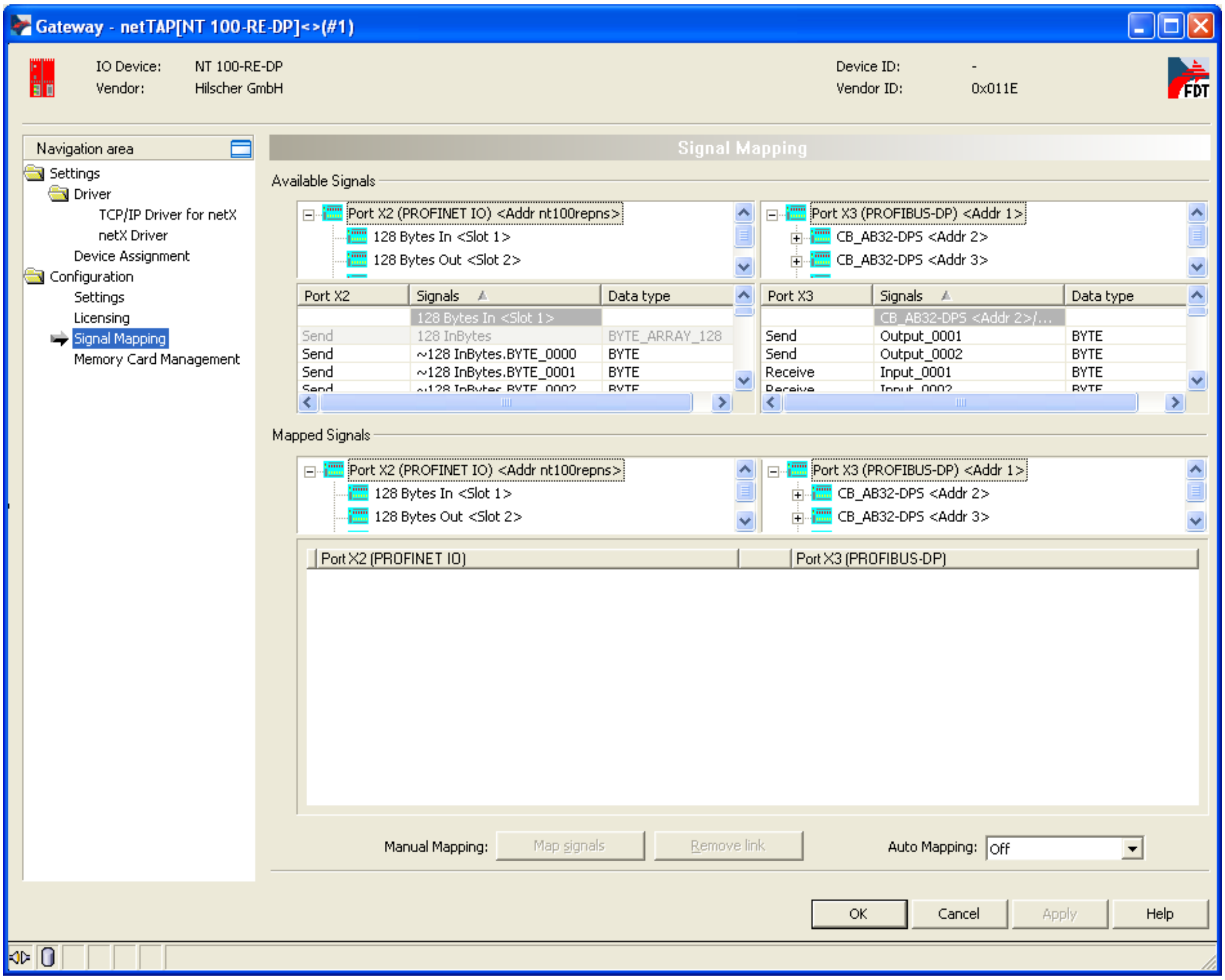


Figure 30: Gateway Signal Mapping

The entry **Memory Card Management** in the navigation area is only available for netTAP NT 100 devices.

3. Signal Mapping: Data transfer from Port X2 to Port X3

- Map the signals, which are received on Port X2 (Port X2 receive), with signals, which should be sent on Port X3 (Port X3 send).
- For this, mark the signal received (Port X2) and the signal to be sent (Port X3) and the click the button **Map Signals**
  Or
  Map the signal using drag and drop. For this, drag the signal received (Port X2 receive) and drop it on the signal to be sent (Port X3 send)

4. Signal Mapping: Data transfer from Port X3 to Port X2

- Map the signals, which are received on Port X3 (Port X3 receive), with signals, which should be sent on Port X2 (Port X2 send).
- For this, mark the signal received (Port X3) and the signal to be sent (Port X2) and the click the button **Map Signals**
  Or
  Map the signal using drag and drop. For this, drag the signal received (Port X3 receive) and drop it on the signal to be sent (Port X2 send)

5. Signal Mapping: Transfer status info of Port X2 to Port X3

- If necessary, map the status information of Port X2 (Port X2 generated, which are generated device internal) to signals which should be sent on Port X3 (Port X3 send)
- For this mark the status signal (Port X2) and the signal which should be sent (Port X3) and click the button **Map Signals**
  Or
  Map the signal using drag and drop. For this, drag the status signal (Port X2 generated) and drop it on the signal to be sent (Port X3 send)

6. Signal Mapping: Transfer status info of Port X3 to Port X2

➢ If necessary, map the status information of Port X3 (Port X3 generated, which are generated device internal) to signals which should be sent on Port X2 (Port X2 send)

➢ For this mark the status signal (Port X3) and the signal which should be sent (Port X2) and click the button **Map Signals**
Or
Map the signal using drag and drop. For this, drag the status signal (Port X3 generated) and drop it on the signal to be sent (Port X2 send)

➾ An example of the Signal Mapping window shows the following figure

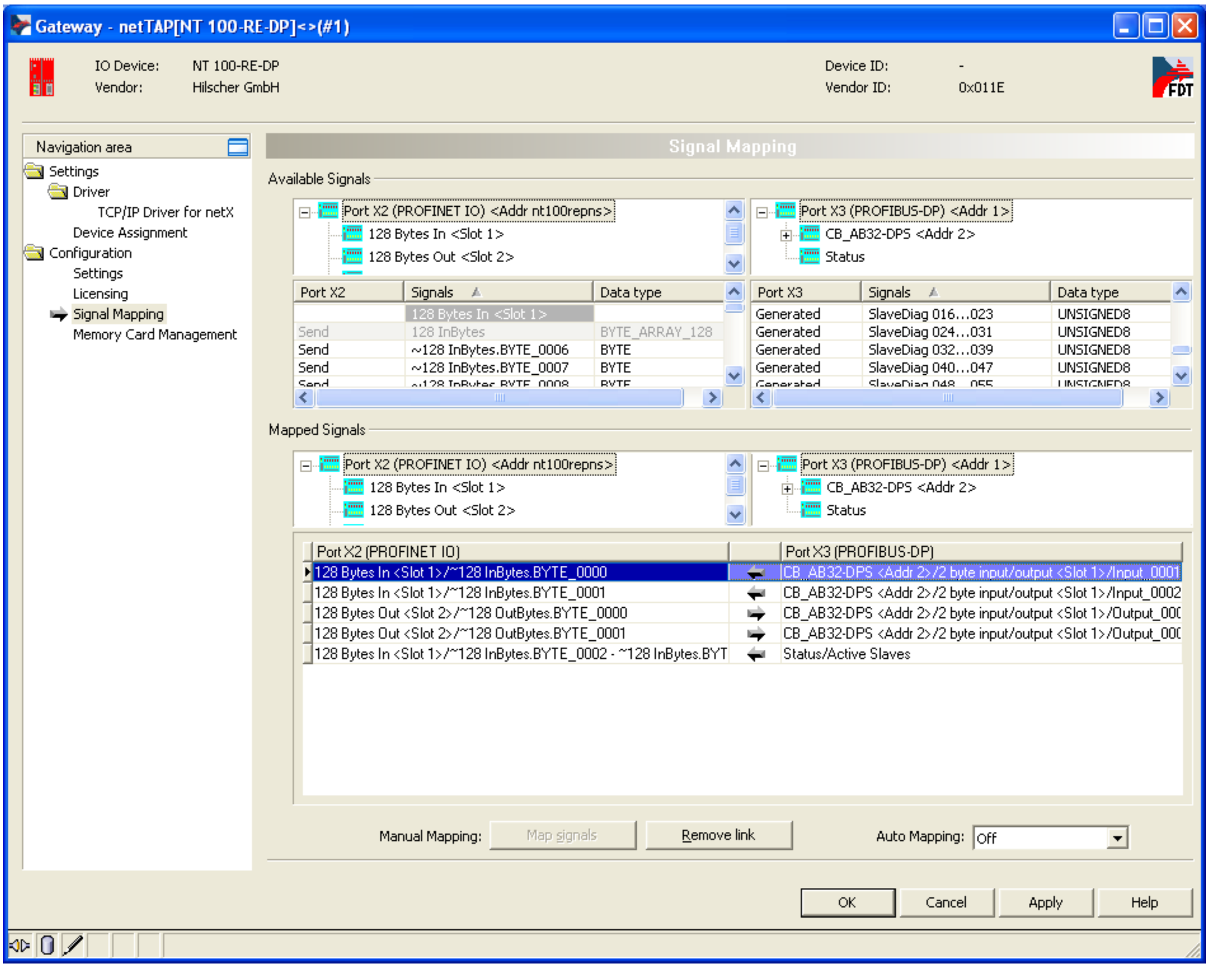


Figure 31: Gateway Signals mapped

The entry **Memory Card Management** in the navigation area is only available for netTAP NT 100 devices.

2.3.7 Establish a Connection to the Gateway Device

1. Connect the USB cable

➢ Connect with an USB cable the USB connector of the netTAP (respectively the netBRICK NB 100) device with the USB connector of the PC.

2. Select the driver

➢ Select **Settings > Driver** and then check **netXDriver**

➯ The following figure shows the selected Driver

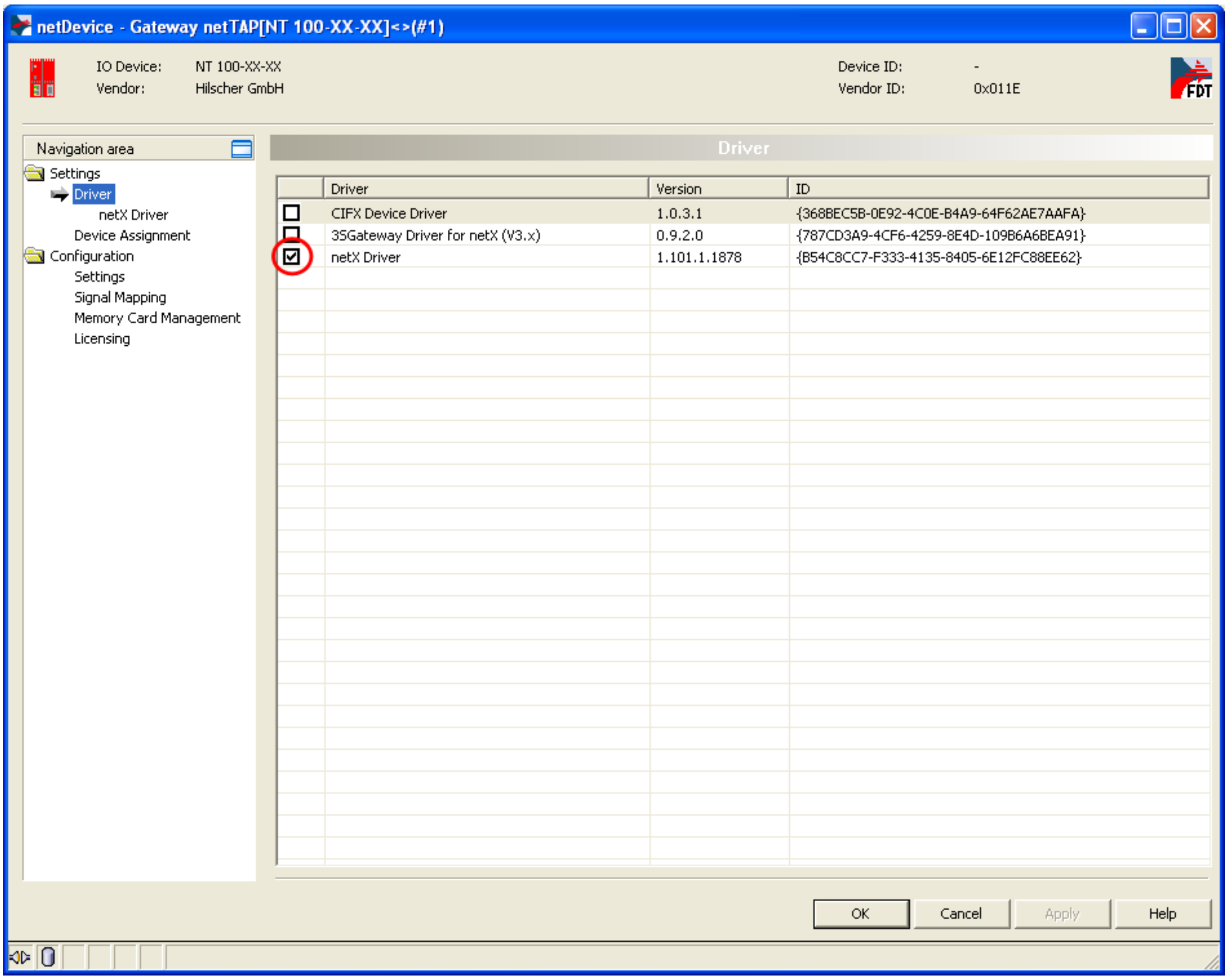


Figure 32: Select USB Driver (1)

The entry **Memory Card Management** in the navigation area is only available for netTAP NT 100 devices.

4. Select the netXDriver

➢ Select **Settings > Driver > netXDriver**

➯ The following figure shows the selected Driver

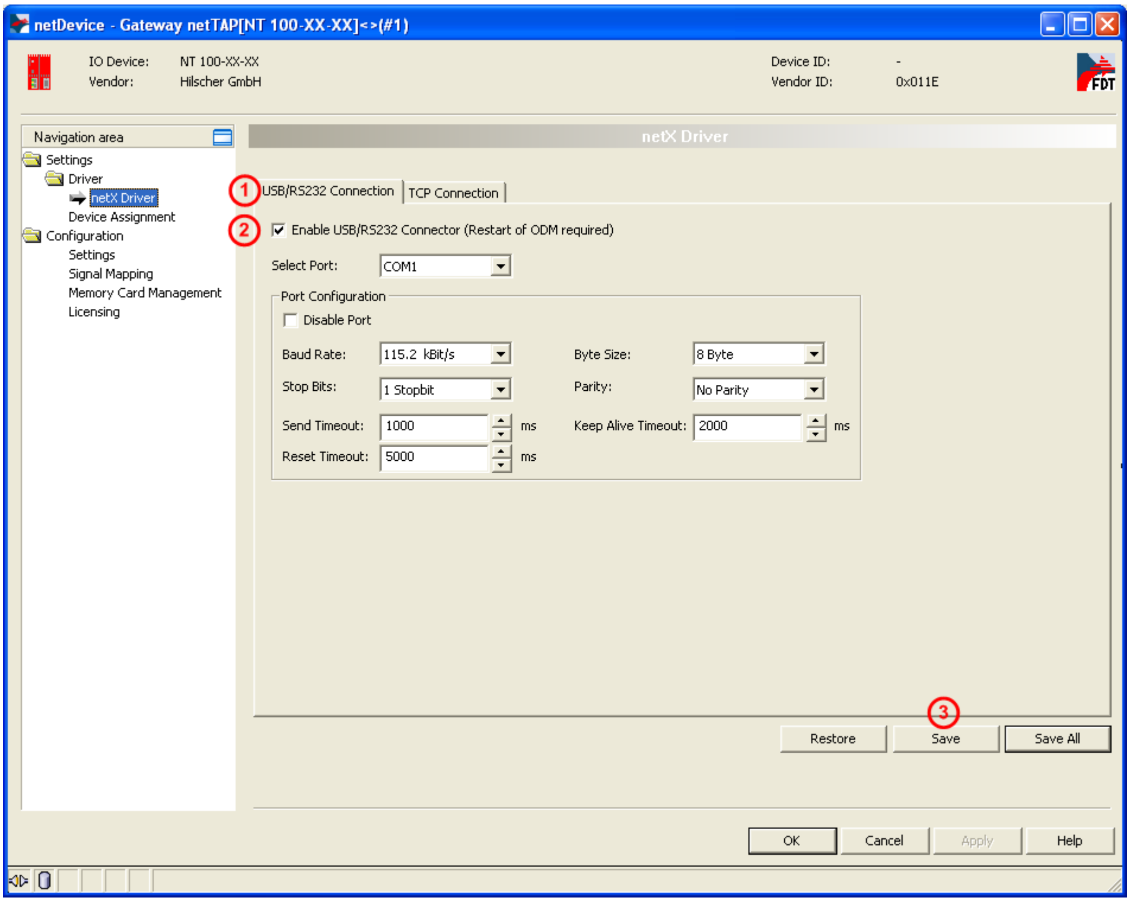


Figure 33: Select USB Driver (2)

5. Select the netXDriver

➢ ① Select the tab **USB/RS232 Connection**.

➢ ② Check **Enable USB/RS232 Connector**.

➢ ③ Click on **Save**.

Note: The standard setting of the netX Driver USB/RS232 can be used for netTAP respectively netBRICK devices without changes.

2.3.8 Device Assignment

➢ Select **Settings > Device Assignment**

➢ Click on (1) **Scan**.

➩ The search process is started. Devices found are display in a list then.

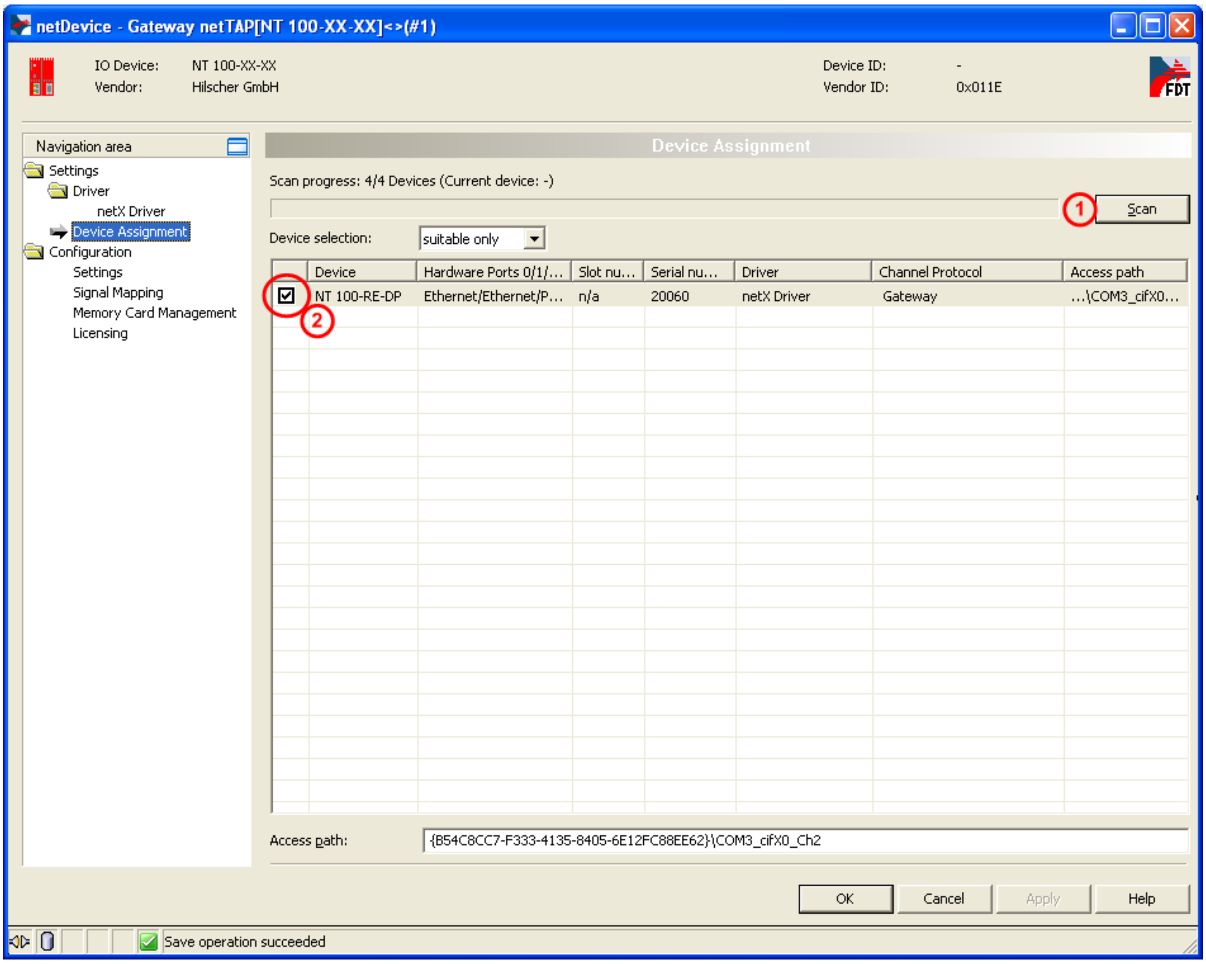


Figure 34: Select Device

➢ Select the device from the list (2) by a check in the field of the device as shown above.

➢ Click on **OK**

➩ The configuration window closes.

The entry **Memory Card Management** in the navigation area is only available for netTAP NT 100 devices.

2.3.9 Load Firmware

The firmware has to be loaded into the netTAP only for first commissioning only. Therefore do the following steps:

1. Open the Gateway configuration window

➢ Select from the context menu of the gateway device symbol the entry **Configuration > Gateway**

➩ The Gateway configuration window opens

2. Open the Settings window

➢ Select **Configuration > Settings**

➩ The window Settings opens

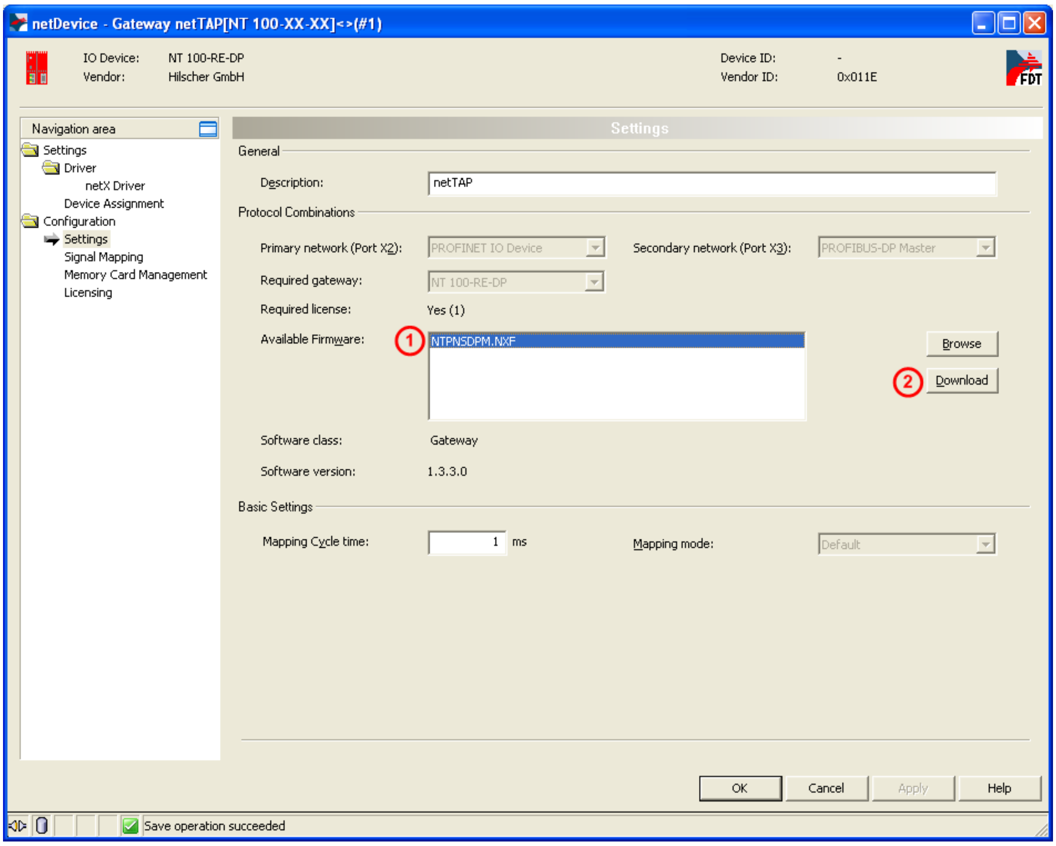


Figure 35: Firmware Download

3. Select Firmware

➢ Mark at **Available Firmware** the firmware file 1: NTPNSDPM.NXF (Firmware for PROFINET IO Device to PROFIBUS-DP Master).

4. Load firmware into the device

➢ Click on **Download**

➩ The firmware is loaded into the netTAP device

Important: Do not remove the cable during the firmware download. Do not disconnect the power supply of the device during the firmware download.

5. Download of firmware
- ➢ Wait until the firmware was transferred completely into the device.

6. Close the Gateway configuration window after the download
- ➢ Click **OK**
- ➩ The Gateway configuration window closes

2.3.10 Load Configuration

1. Download configuration
- ➢ Select from the context menu of the gateway device symbol the entry **Download**
- ➢ Answer the security question with **Yes**, if the download should start
- ➩ SYCON.net builds up an online connection to the device. This the indicated by a green highlighted device name.
- ➩ The configuration is transferred info the gateway device
- ➩ The device performs a reset and then starts with the new configuration.

2.3.11 Save Project

Note: The configuration downloaded from SYCON.net into the device can’t be uploaded from the device. Only the SYCON.net project can be downloaded into additional devices.

Save the SYCON.net project. In case of a device replacement the saved project can be opened with SYCON.net and loaded into the device.

- ➢ To save a project select menu **File > Save** respectively **File > Save As** or click on icon .

When you exit the program and the current configuration differs from the last saved configuration, then the following question appears:

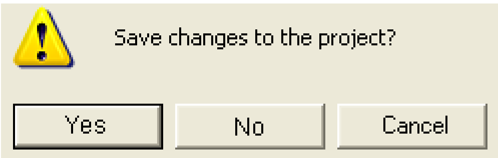


Figure 36: Security Question – Save Project

When you answer with **Yes**, then the project is saved. When you answer with **No**, then the project is not saved and the changes are lost. When you answer with **Cancel**, then the project is not saved.

2.3.12 Device Description File to configure the Master

A GSDML file is required to configure the PROFINET IO Controller.

- To configure the PROFINET IO Controller use the following GSDML file from the product DVD directory EDS/PROFINET.

  for NT 100: GSDML-V2.1-HILSCHER-NT 100-RE PNS-20090123.xml,

  for NB 100: GSDML-V2.1-HILSCHER-NB 100-RE PNS-20100226.xml,

  for NT 50: GSDML-V2.1-HILSCHER-NT 50-EN PNS-20100226.xml

Note: The product IDs are different for devices used as gateway and devices used as proxy. Make sure to use/import the right GSDML file at the PROFINET IO Controller.

- Load the GSDML file into the configuration software of the PROFINET IO Controller.

2.4 Configuration of a NT 100 as Proxy

For the conversion from PROFINET IO Device to PROFIBUS-DP Master with proxy functionality the device NT 100-RE-DP can be used. Therefore use the device NT 100-RE-XX/PROXY from the device catalog.

These devices are called proxy device in the following description.

The device NT 100-RE-DP with one master license is necessary for the protocol conversion from PROFINET IO Device to PROFIBUS-DP Master.

The following steps have to be done to configure the device:

2.4.1 Start SYCON.net and User Login

1. Start SYCON.net

➢ Select **Start > Programs > SYCON.net System Configurator > SYCON.net**

➩ SYCON.net is started

2. User Login

➢ In the window SYCON.net User Login click **OK** to login or enter your password and then click **OK** to login

➩ SYCON.net frame application appears

2.4.2 Insert the Proxy Device into the Configuration Window

➢ Go to the device catalog under vendor **Hilscher GmbH** to the category **Gateway / Stand-Alone Slave**. Use drag and drop with the NT 100-XX-XX proxy device to insert it at the (gray) main line.

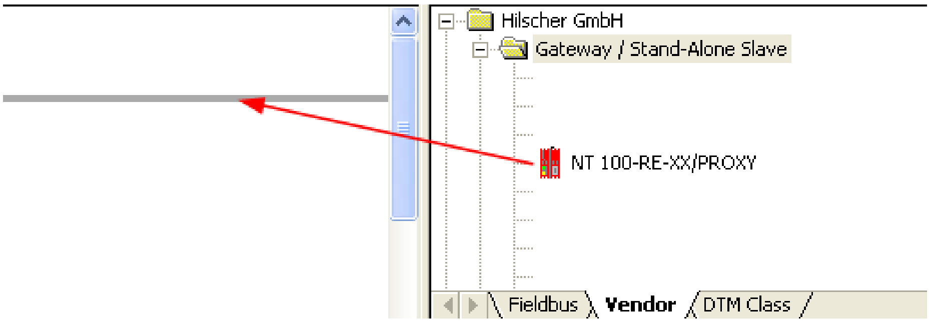


Figure 37: Insert Proxy Device into the Project

➮ The proxy device appears in the project

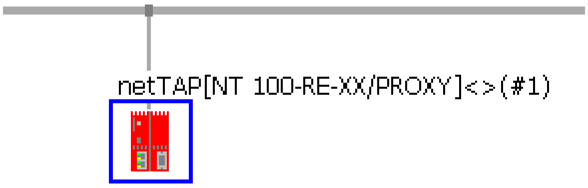


Figure 38: Proxy Device in the Project

2.4.3 Select the Protocol Conversion of the Proxy

1. Open the Proxy configuration window.

➢ Select from the context menu of the proxy device symbol the entry **Configuration > Proxy.**

➩ The Proxy configuration window opens.

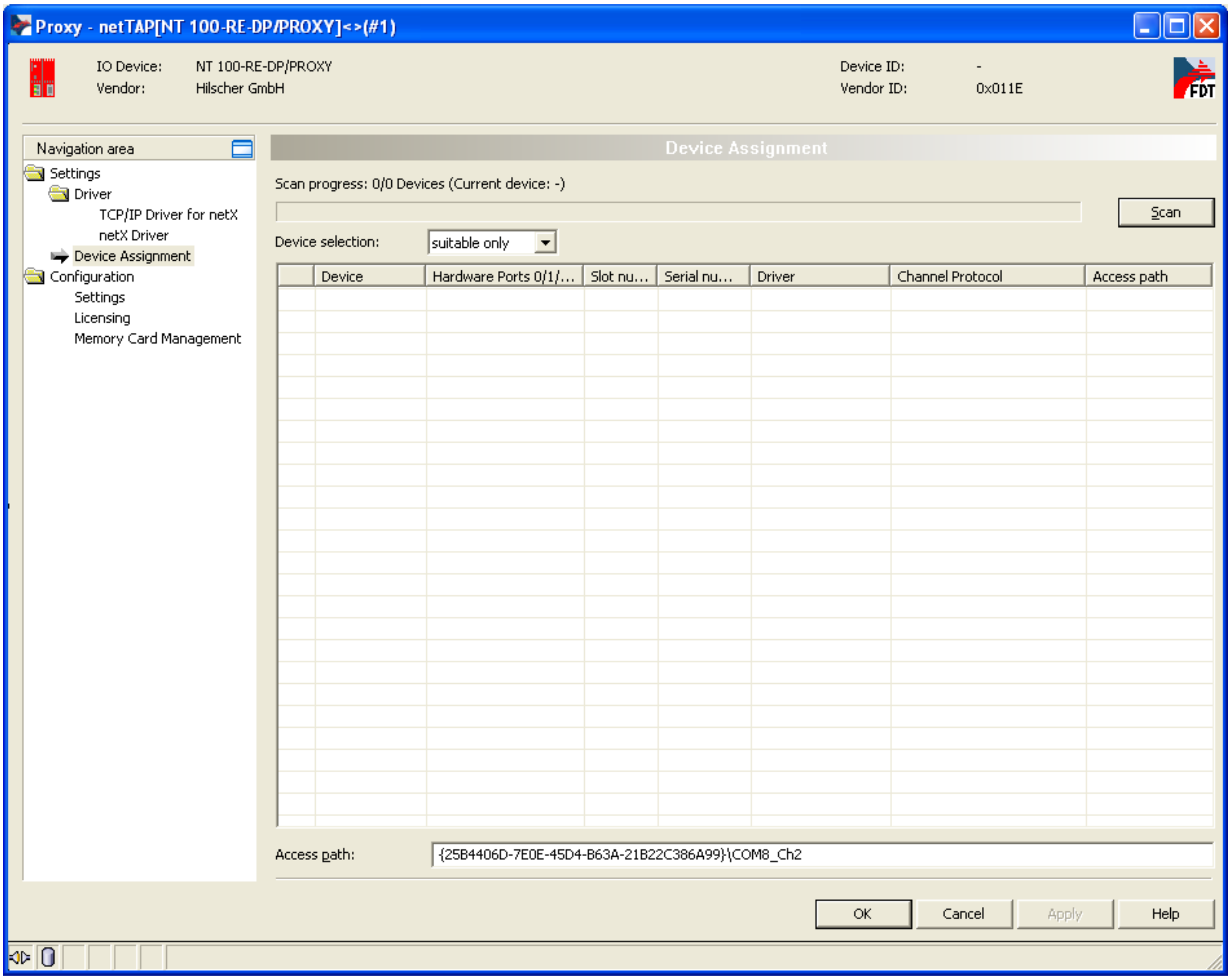


Figure 39: Device Assignment without Device

2. Open dialog for protocol selection

➢ Select in the Navigation area **Configuration** > **Settings**

➩ The configuration window **Settings** opens

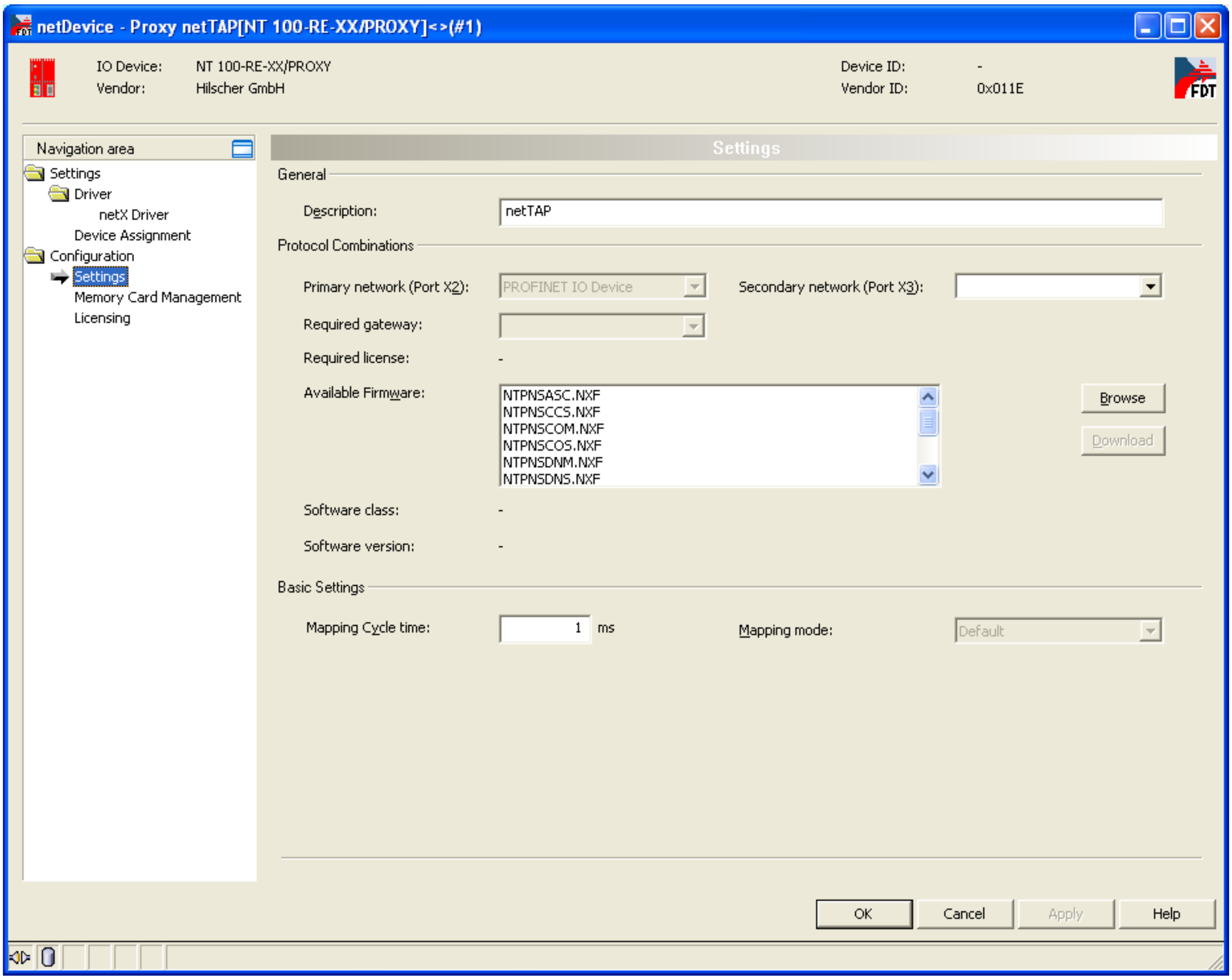


Figure 40: Proxy Protocol Selection (1)

3. Select the protocol for the primary network and for the secondary network and apply

➢ In the window **Configuration > Settings** at **Protocol Combinations** for the **Primary Network (Port X2)** the protocol PROFINET IO Device is already preselected.

➢ Select then at **Protocol Combinations** for the **Secondary Network (Port X3)** the protocol PROFIBUS-DP Master

➾ The proxy configuration window shows the following.

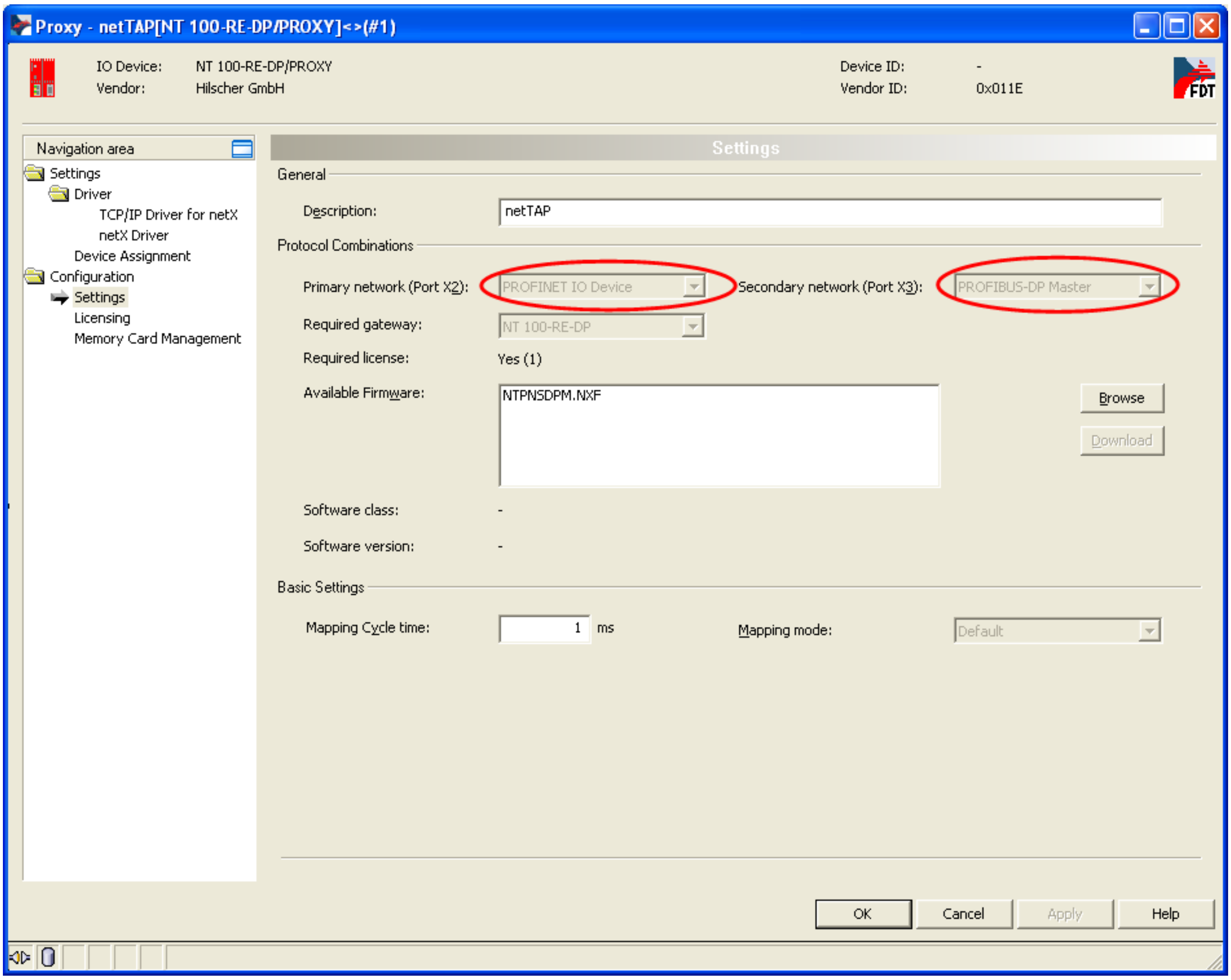


Figure 41: Proxy Protocol Selection (2)

The **Mapping Mode** is preselected with **Default** (only selection) and works as follows:

PROFIUS-DP Slave devices which are inserted at the bus line are represented by a slot number on PROFINET. The PROFINET slot number is equal to the PROFIBUS station address. The PROFIBUS modules are represented by a PROFINET sub slot number.

4. Close the proxy configuration window

➢ Click on **OK**

➾ The proxy configuration window closes

2.4.4 Configuration of the secondary Network

If the slave device is not listed in the device catalog, then it has to be imported into the device catalog first.

2.4.4.1 Expand PROFIBUS-DP Slave Device Catalog

If you want to use a PROFIBUS-DP slave device that is not listed in the device catalog, then you have to import the GSD file into the SYCON.net device catalog.

➢ Proceed as follows:

1. Missing PROFIBUS-DP Slave devices can be added to the device catalog using the menu **Network > Import Device Descriptions**.
2. Select the new GSD file.
3. Answer the question “Do you want to reload the device catalog?“ with **Yes** to reload the catalog.

More information about the device catalog is in the document „SYCONnet_netDevice_en.pdf“ in the directory „Documentation“ on the product DVD.

2.4.4.2 Insert PROFIBUS-DP Slave Device to the PROFIBUS Network

➢ Go to the device catalog. Use drag and drop with one or more PROFIBUS-DP Slave devices to insert it/them at the PROFIBUS bus line.

➩ The PROFIBUS-DP Slave device icons appear at the PROFIBUS network line (Secondary network)

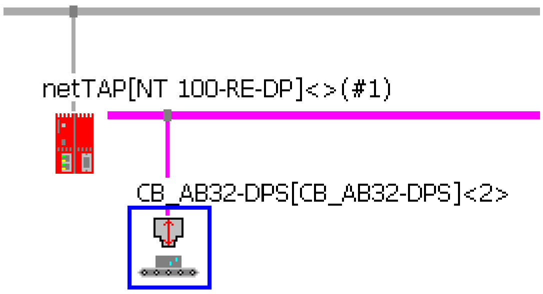


Figure 42: Proxy Device with Slave

2.4.4.3 Configure the PROFIBUS-DP Slave device

- Open the configuration window with a double click on the device icon of the PROFIBUS-DP Slave
- The configuration window of the PROFIBUS-DP Slave device opens
- Select in the navigation area **Configuration > Modules**.
- Select from **Available Modules** the module(s) and add it/them to the **Configured Modules** to configure the Slave. The **Configured Modules** has to match the configuration of the used PROFIBUS-DP slave device.

More information about the configuration of PROFIBUS-DP Slave devices are in the document PROFIBUS_Generic_Slave_DTM_en.pdf on the product DVD in section *Configuration*.

2.4.4.4 Configure PROFIBUS-DP Master

1. Open the PROFIBUS-DP Master (Port X3) configuration window

- Select from the context menu of the gateway device symbol the entry **Configuration > PROFIBUS-DP Master**
- The PROFIBUS-DP Master configuration window opens

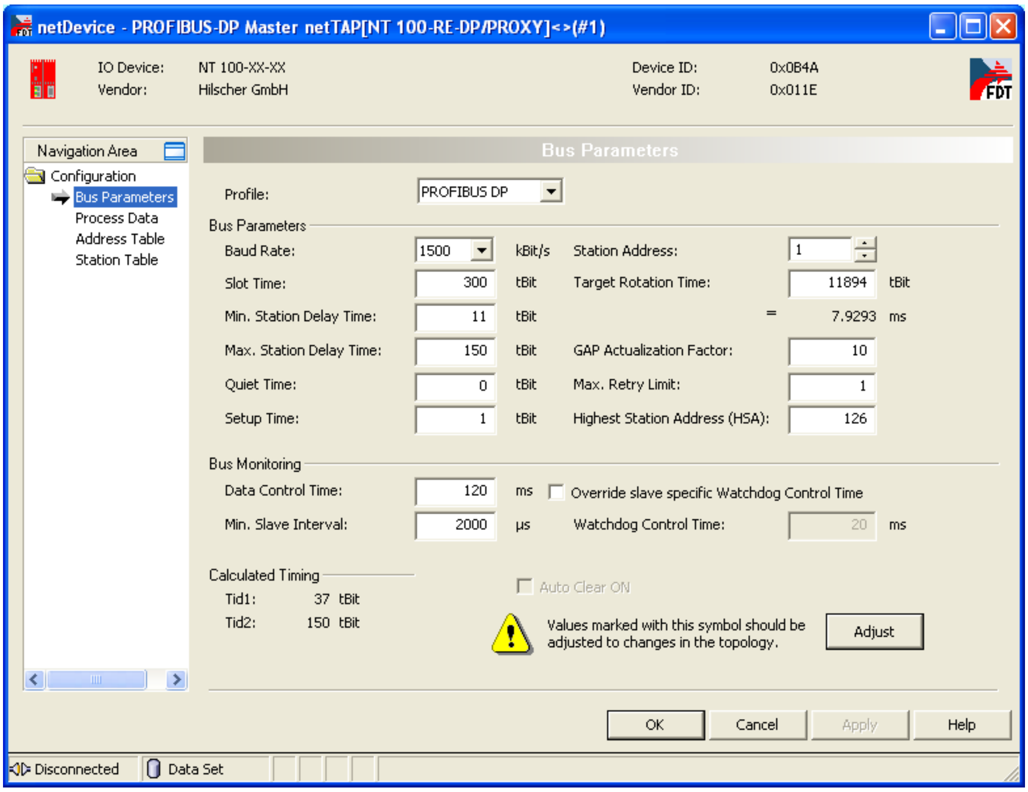


Figure 43: PROFIBUS-DP Bus Parameter

2. Configure the PROFIBUS-DP Master

➢ Set the parameter. Set especially under **Configuration > Bus Parameter** the bus parameter and under **Configuration > Station Table** the station addresses (stations addresses of the PROFIBUS-DP Slave devices)

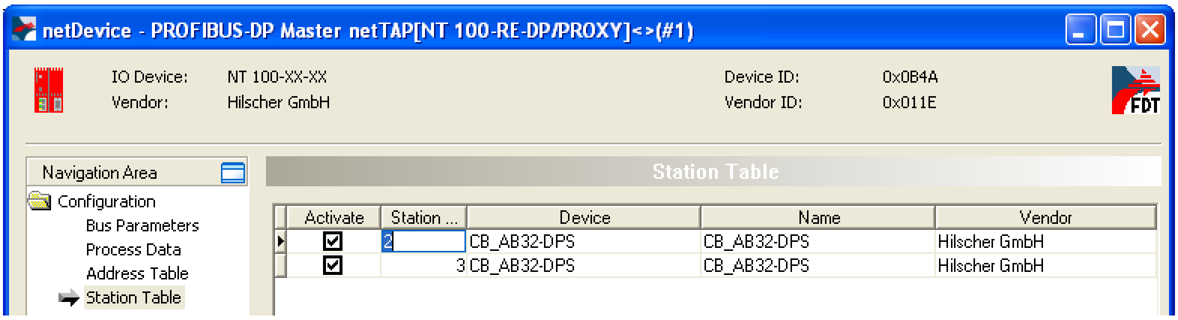


Figure 44: PROFIBUS-DP Slave Station Address

More information about the configuration of PROFIBUS-DP Master devices are in the document PROFIBUS_Master_netX_DTM_en.pdf in section *Configuration.*

3. Close the configuration window

➢ Click on the button **OK**

➩ The configuration window closes

2.4.5 Configuration of the primary Network

The PROFINET IO network is on port X2 of the proxy device.

1. Open PROFINET IO Device configuration window

➢ Select from the context menu of the proxy device symbol the entry **Configuration > PROFINET IO Device.**

➩ The PROFINET IO Device configuration window opens with the view of the modules

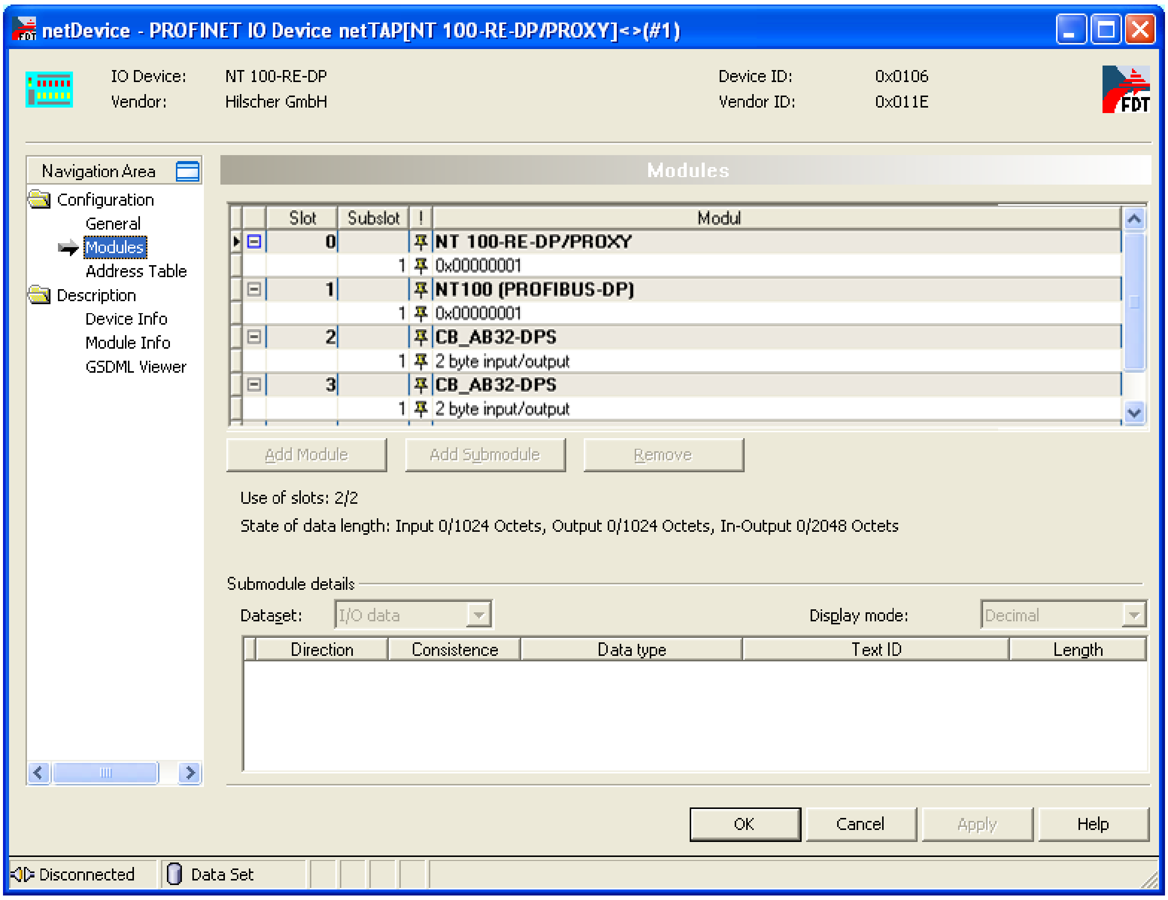


Figure 45: PROFINET IO Device Table

Here the modules with its slot number and sub slot number are displayed, as they will become visible at the PROFINET IO Controller. These slot numbers and sub slot numbers will be exported into the GSDML file, which is described later.

Slot 0 (NT 100-RE-DP/PROXY) and slot 1 (NT100 (PROFIBUS-DP)) are always present. Two additional slots (slot 2 in this case, which represent the PROFIBUS-DP Slave with station address 2; slot 3 in this case, which represent the PROFIBUS-DP Slave with station address 3) are automatically displayed when the PROFIBUS-DP Slaves were inserted to the PROFIBUS-DP network.

These Slot and sub slot numbers are used in the GSDML file. The GSDML file can be created (exported) which is described later.

2. Set the name of station

➢ Select in the navigation area **General**.

➪ The following configuration window opens

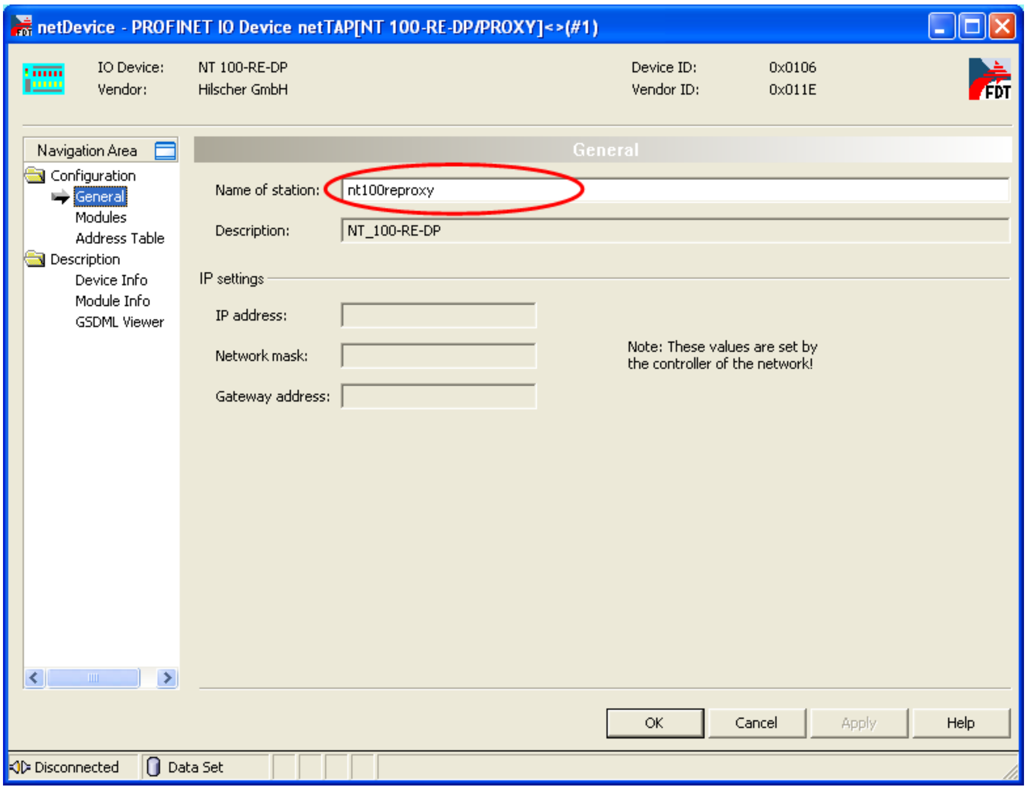


Figure 46: PROFINET IO Name of Station

➢ Enter the **Name of Station**

Note: The name of station is very important and is used by the PROFINET IO Controller to identify the proxy device.

3. Close the configuration window

➢ Click on the button **OK**

➪ The configuration window closes

2.4.6 Establish a Connection to the Proxy Device

1. Connect the USB cable

➢ Connect with an USB cable the USB connector of the netTAP device with the USB connector of the PC.

2. Select the driver

➢ Select **Settings > Driver** and then check **netXDriver**

➾ The following figure shows the selected Driver

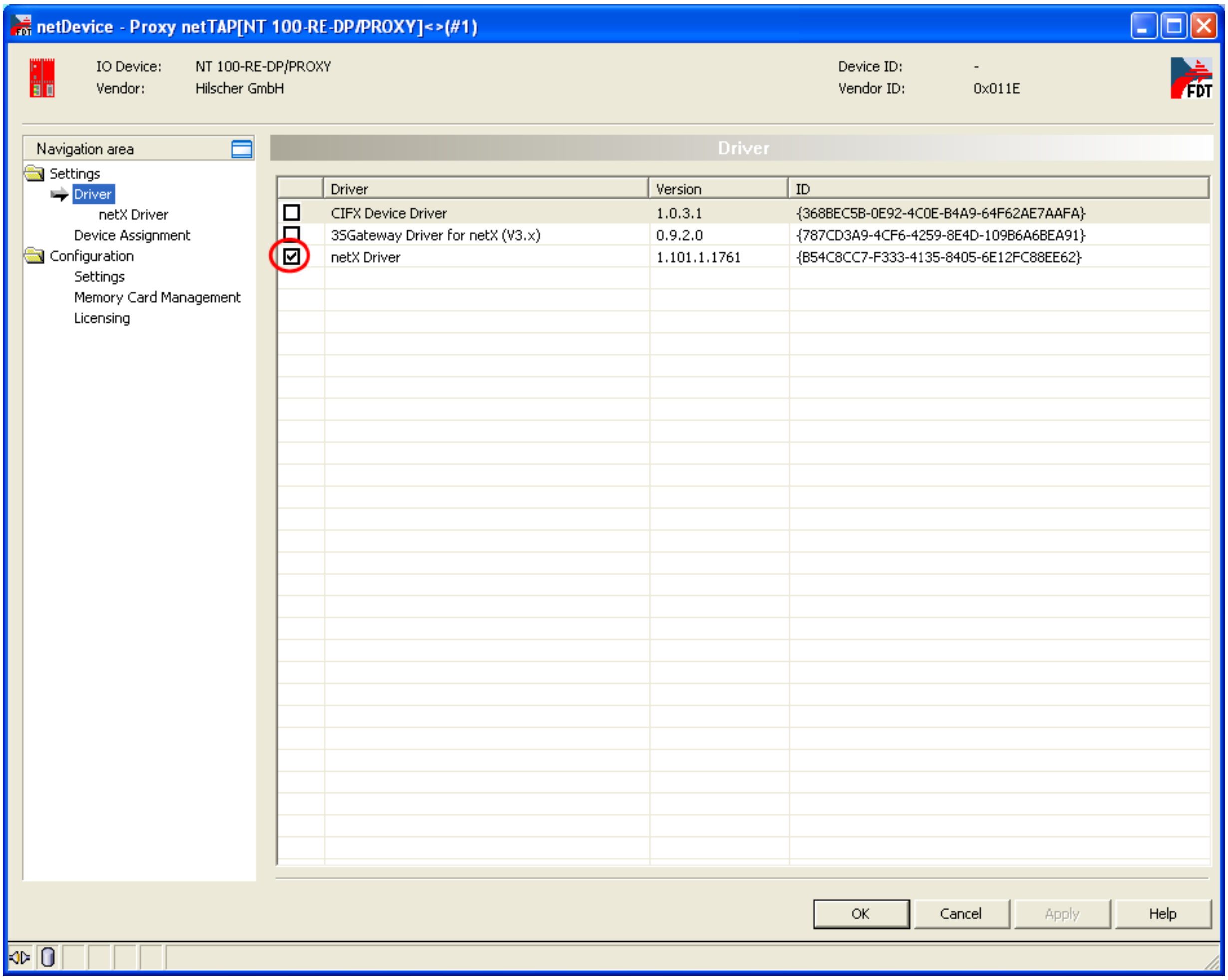


Figure 47: Select USB Driver (1)

4. Select the netXDriver

➢ Select **Settings > Driver > netXDriver**

➢ The following figure shows the selected Driver

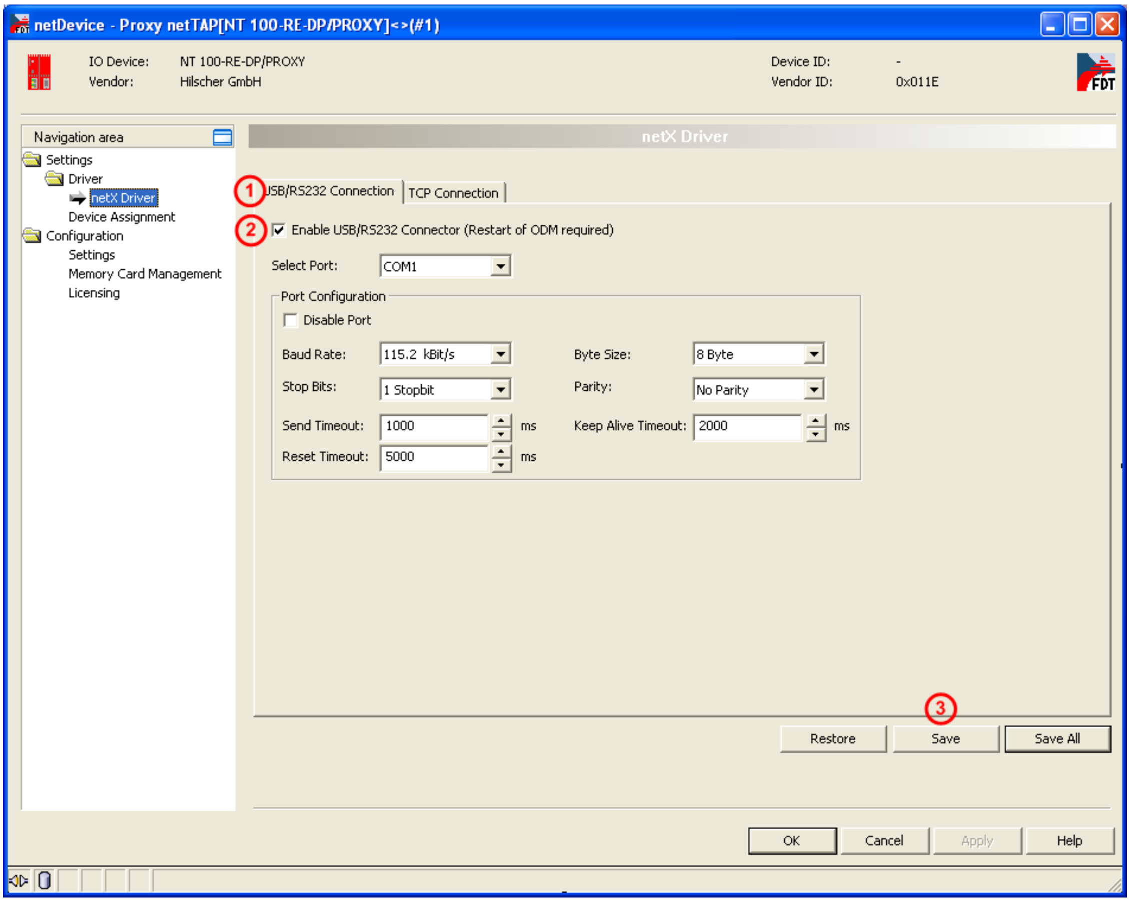


Figure 48: Select USB Driver (2)

5. Select the netXDriver

➢ (1) Select the tab **USB/RS232 Connection**.

➢ (2) Check **Enable USB/RS232 Connector**.

➢ (3) Click on **Save**.

Note: The standard setting of the netX Driver USB/RS232 can be used for netTAP respectively netBRICK devices without changes.

2.4.7 Device Assignment

➢ Select **Settings > Device Assignment**

➢ Click on ① **Scan**.

➩ The search process is started. Devices found are display in a list then.

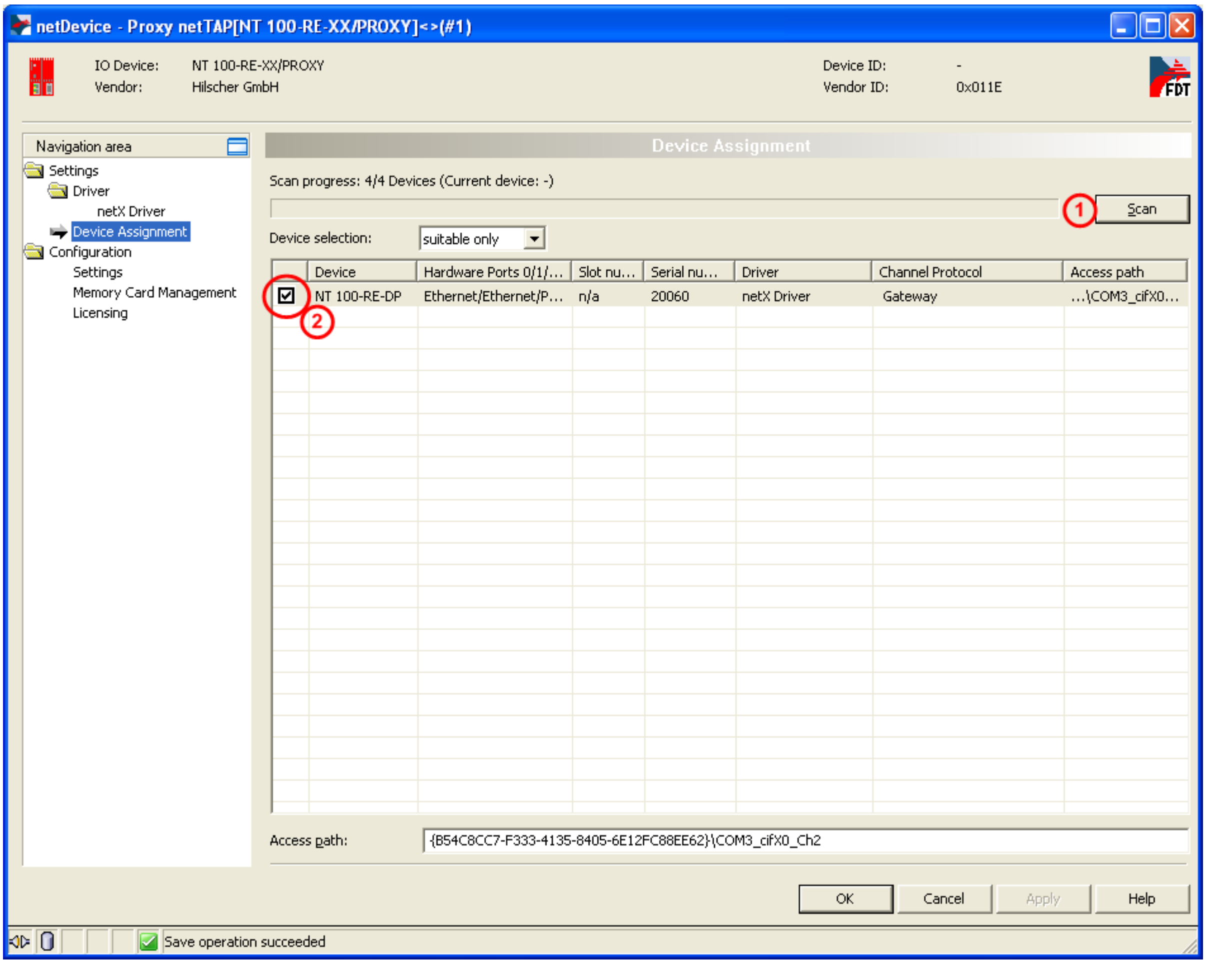


Figure 49: Select Device

➢ Select the device from the list ② by a check in the field of the device as shown above.

➢ Click on **OK**

➩ The configuration window closes.

2.4.8 Load Firmware

The firmware has to be loaded into the netTAP only for first commissioning only. Therefore do the following steps:

1. Open the proxy configuration window

➢ Select from the context menu of the gateway device symbol the entry **Configuration > Proxy**

➩ The proxy configuration window opens

2. Open the Settings window

➢ Select **Configuration > Settings**

➩ The window Settings opens

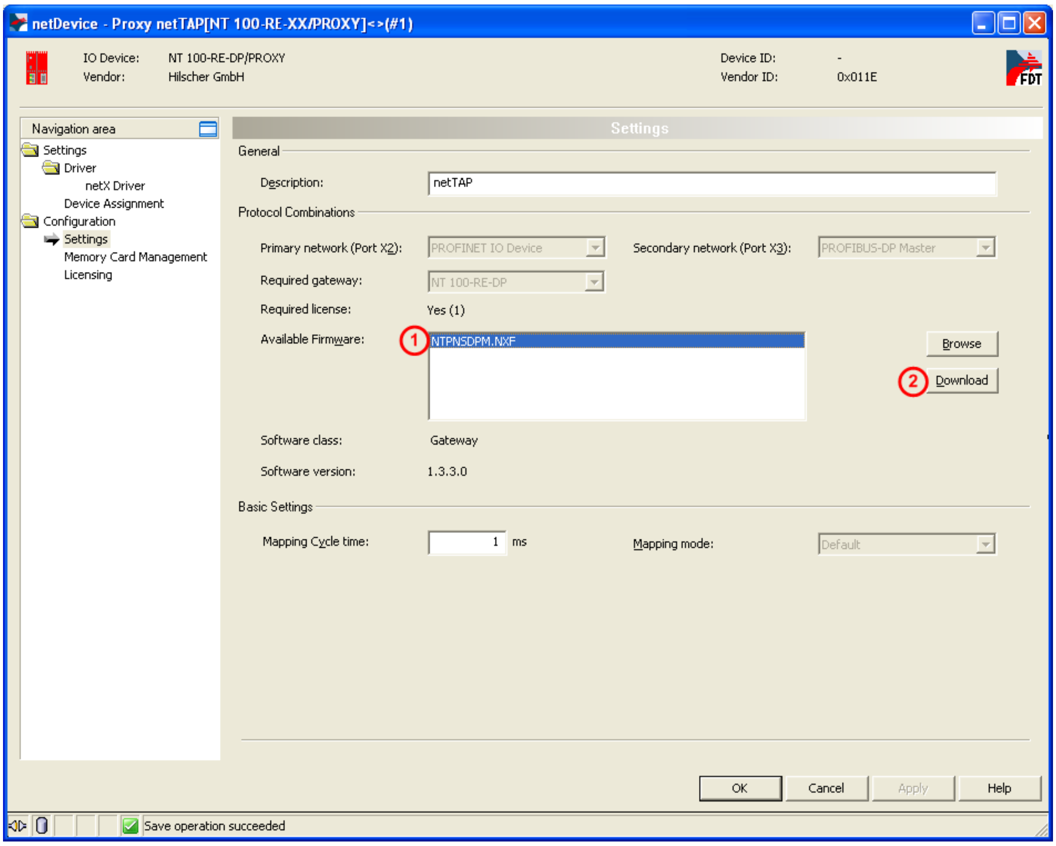


Figure 50: Firmware Download

3. Select Firmware

➢ Mark at **Available Firmware** the firmware file (1): NTPNSDPM.NXF (Firmware for PROFINET IO Device to PROFIBUS-DP Master).

4. Load firmware into the device

➢ Click on **Download**

➩ The firmware is loaded into the netTAP device

Important: Do <u>not</u> remove the cable during the firmware download. Do <u>not</u> disconnect the power supply of the device during the firmware download.

5. Download of firmware

➢ Wait until the firmware was transferred completely into the device.

6. Close the Gateway configuration window after the download

➢ Click **OK**

➮ The Gateway configuration window closes

2.4.9 Load Configuration

1. Download configuration

➢ Select from the context menu of the proxy device symbol the entry **Download**

➢ Answer the security question with **Yes**, if the download should start

➮ SYCON.net builds up an online connection to the device. This the indicated by a green highlighted device name.

➮ The configuration is transferred info the proxy device

➮ The device performs a reset and then starts with the new configuration.

2.4.10 Save Project

Note: The configuration downloaded from SYCON.net into the device can't be uploaded from the device. Only the SYCON.net project can be downloaded into additional devices.

Save the SYCON.net project. In case of a device replacement the saved project can be opened with SYCON.net and loaded into the device.

➢ To save a project select menu **File > Save** respectively **File > Save As** or click on icon .

When you exit the program and the current configuration differs from the last saved configuration, then the following question appears:

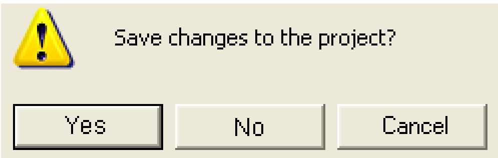


Figure 51: Security Question – Save Project

When you answer with **Yes**, then the project is saved. When you answer with **No**, then the project is not saved and the changes are lost. When you answer with **Cancel**, then the project is not saved.

2.4.11 Device Description File to configure the PROFINET IO Controller

A GSDML file is required to configure the PROFINET IO Controller. This file is creates as follows:

- ➢ Select from the context menu of the proxy device symbol the entry **Additional Functions > PROFINET IO Device > Export GSDML**.
- ➾ The dialog to save the XML file appears.
- ➢ Select the directory the GSDML file should be saved into.
- ➢ A name for the GSDML file is proposed. Accept it or change it useful.
- ➢ Click on **Save.**
- ➾ The GSDML file is saved.

The structure of the file name is:

1 Fixed prefix

2 Version number of the GSDML specification

3 Manufacturer name

4 Device type

5 Date: Format yyyymmdd

6 Time: Format hhmmss

7 File extension (always): xml

- ➢ Use the GSDML file to configure the PROFINET IO Controller

2.5 Configuration of a NT 151-RE-RE (netTAP Real-Time Ethernet Gateway)

This section provides step-by-step instructions for configuring the netTAP **NT 151-RE-RE** device. In the following description, this device is also called "gateway device".

The configuration is exemplified by using the protocol conversion PROFINET IO Device to PROFINET IO Device.

Information on how to use an SD memory card to copy configuration data from one netTAP into another netTAP device (e. g. in order to "clone" a spare device) can be found in the user manual *netTAP NT 151-RE-RE*, DOC150802UMxxEN.

2.5.1 Start SYCON.net and User Login

1. Start SYCON.net.

➢ Choose **Start > Programs > SYCON.net System Configurator > SYCON.net**

➩ SYCON.net is started.

2. User Login

➢ In the window **SYCON.net User Login** click **OK** to login or enter your password and then click **OK** to login.

➩ SYCON.net frame application appears.

2.5.2 Insert the Gateway Device into the Configuration Window

➢ In the device catalog, open **Vendor** tab, then open **Hilscher GmbH > Gateway / Stand-Alone Slave** folder. Select the **NT 151-RE-RE** gateway, then “drag and drop” the device onto the (gray) main line.

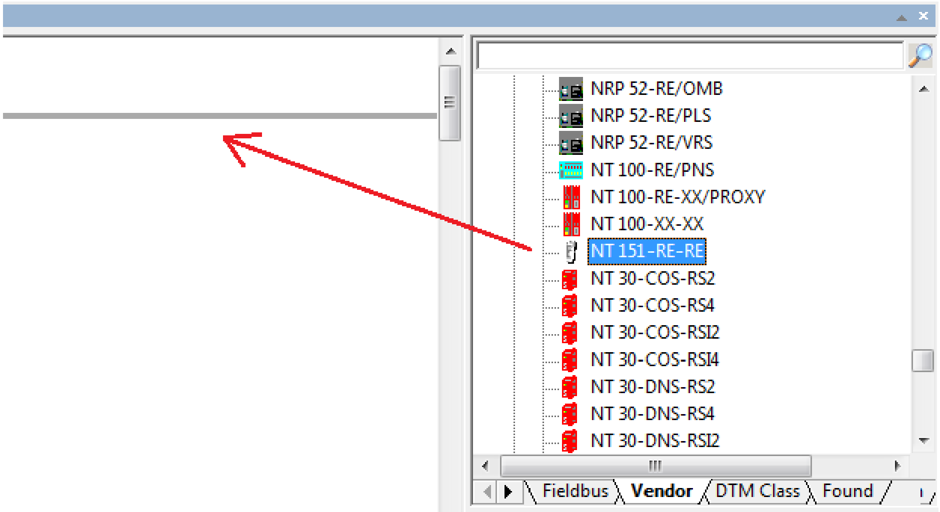


Figure 52: Insert Gateway Device into the Project

➩ The gateway device appears in the project:

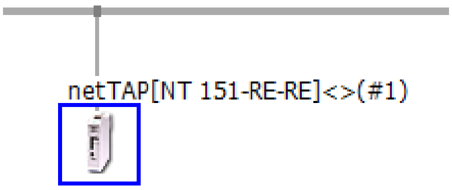


Figure 53: Gateway Device in the Project

2.5.3 Select the Protocol Conversion of the Gateway

1. Open the Gateway configuration window.

➢ In the context menu of the gateway device symbol, choose **Configuration > Gateway.**

➩ The Gateway configuration window opens:

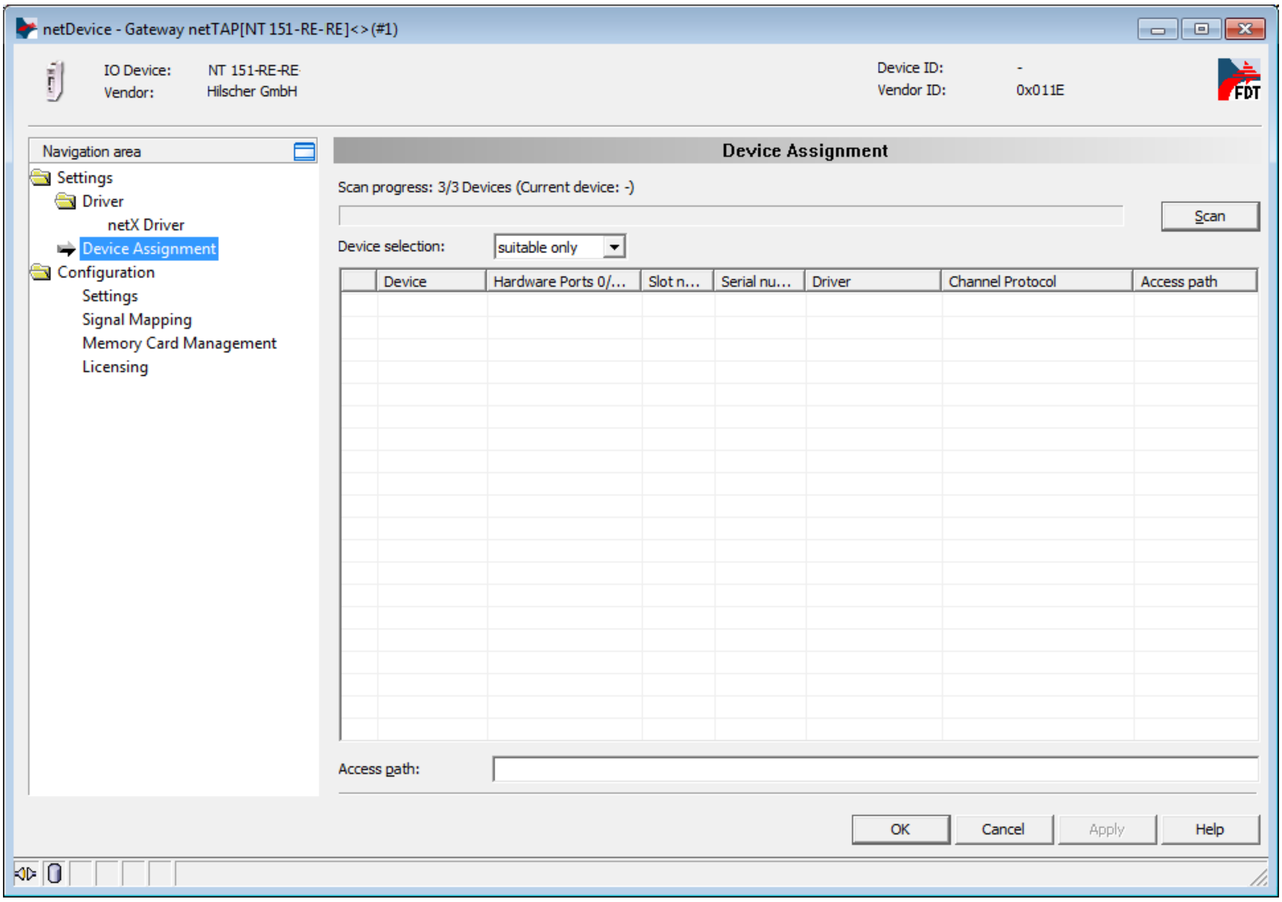


Figure 54: Device Assignment without Device

2. Open dialog for protocol selection.

➢ In the **Navigation area**, select **Configuration** > **Settings**.

➮ The configuration window **Settings** opens:

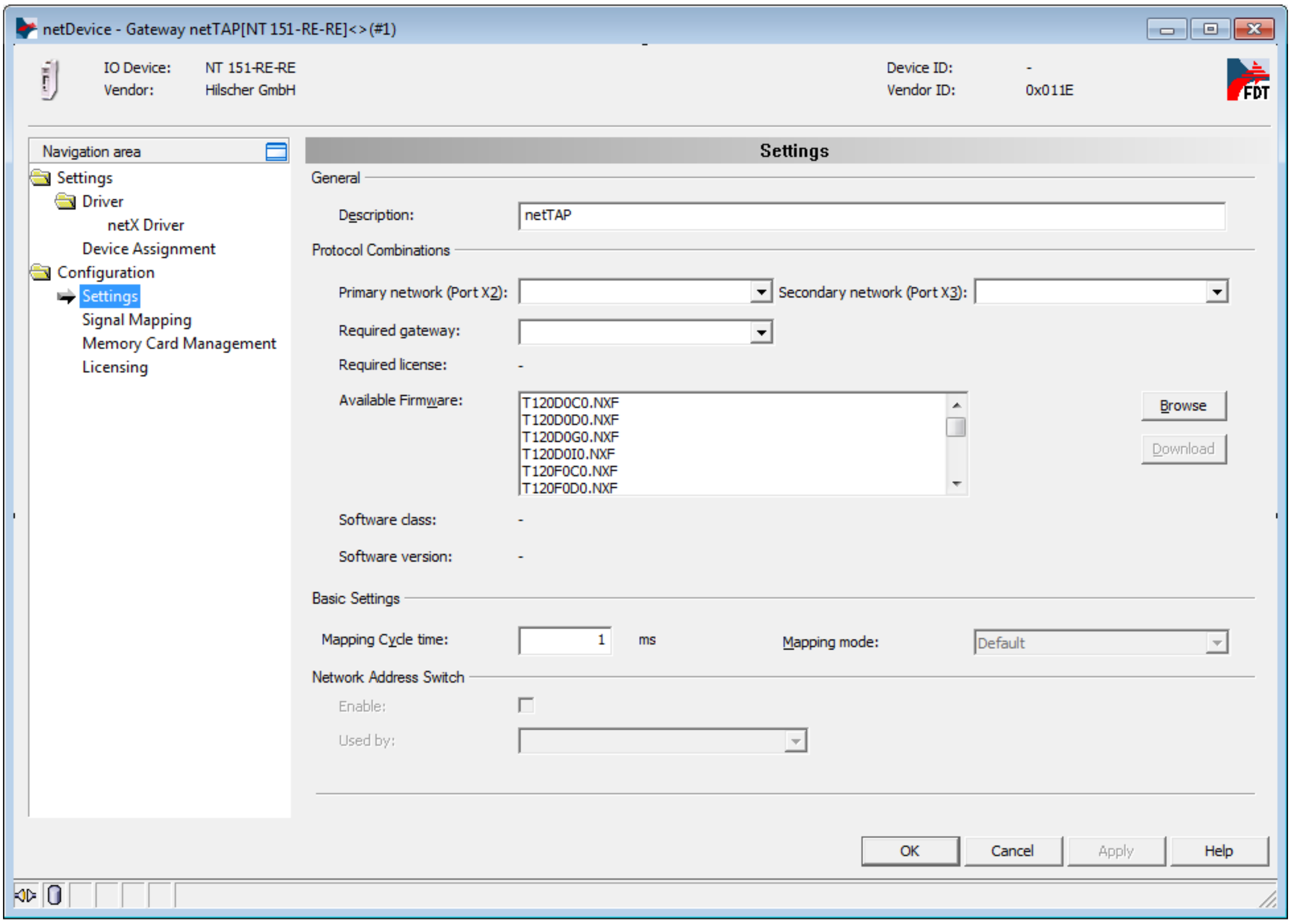


Figure 55: Gateway Protocol Selection (1)

3. Select the protocol for the primary network and for the secondary network.

➢ In the **Protocol Combinations** area, select `PROFINET IO Device` from the the **Primary Network (Port X2)** drop-down list.

➢ From the **Secondary Network (Port X3)** drop-down list, also select `PROFINET IO Device`.

➢ Click **Apply** button.

➩ The Gateway configuration window now looks as depicted below:

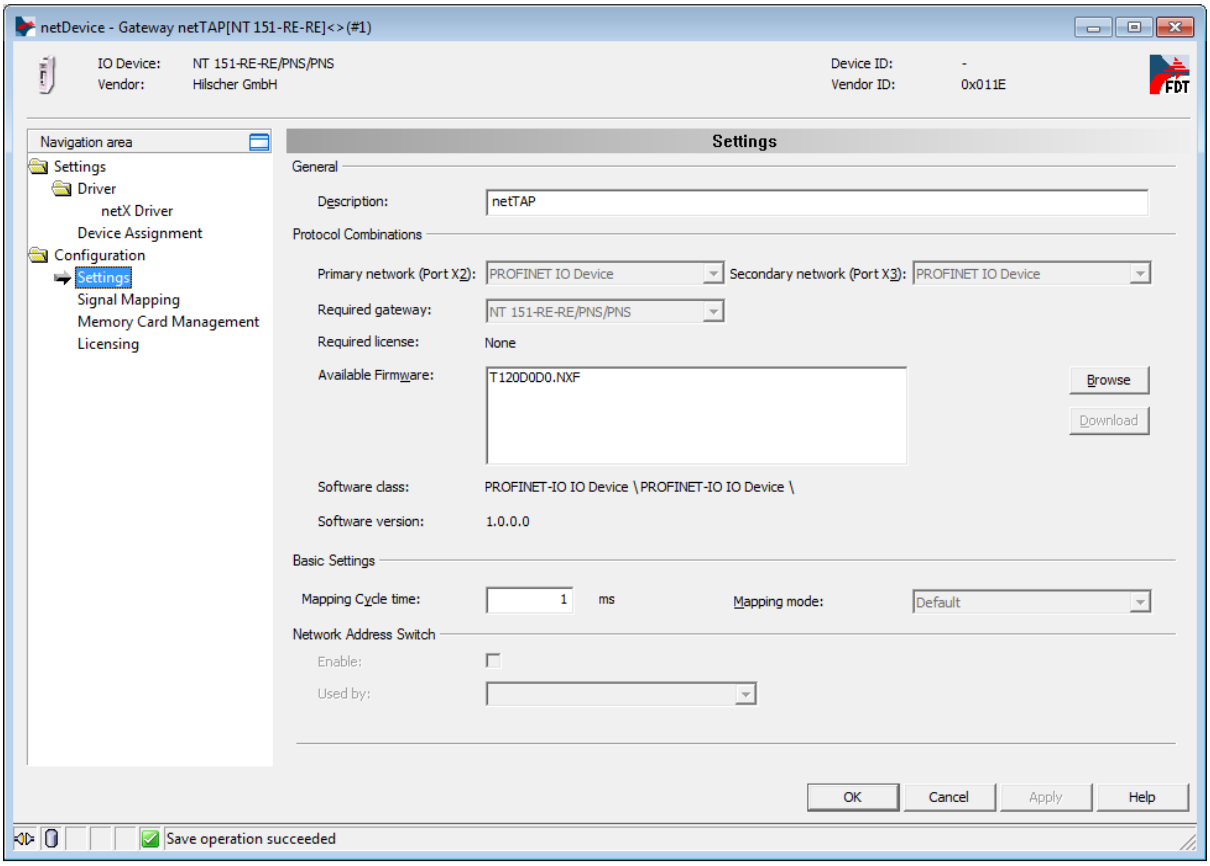


Figure 56: Gateway Protocol Selection (2)

4. Close the Gateway configuration window.

➢ Click **OK** button.

➩ The Gateway configuration window closes.

2.5.4 Configuration of the primary Network

1. Open the configuration window.

➢ In the context menu of the gateway device symbol, choose **Configuration > PROFINET IO Device** (choose the upper of the two **PROFINET IO Device** entries).

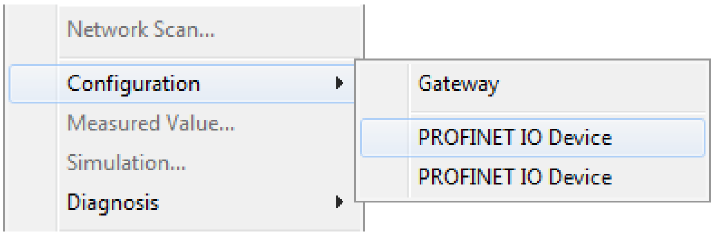


Figure 57: Open Port X2 via context menu

➩ The PROFINET IO Device configuration window opens for the primary network at Port X2:

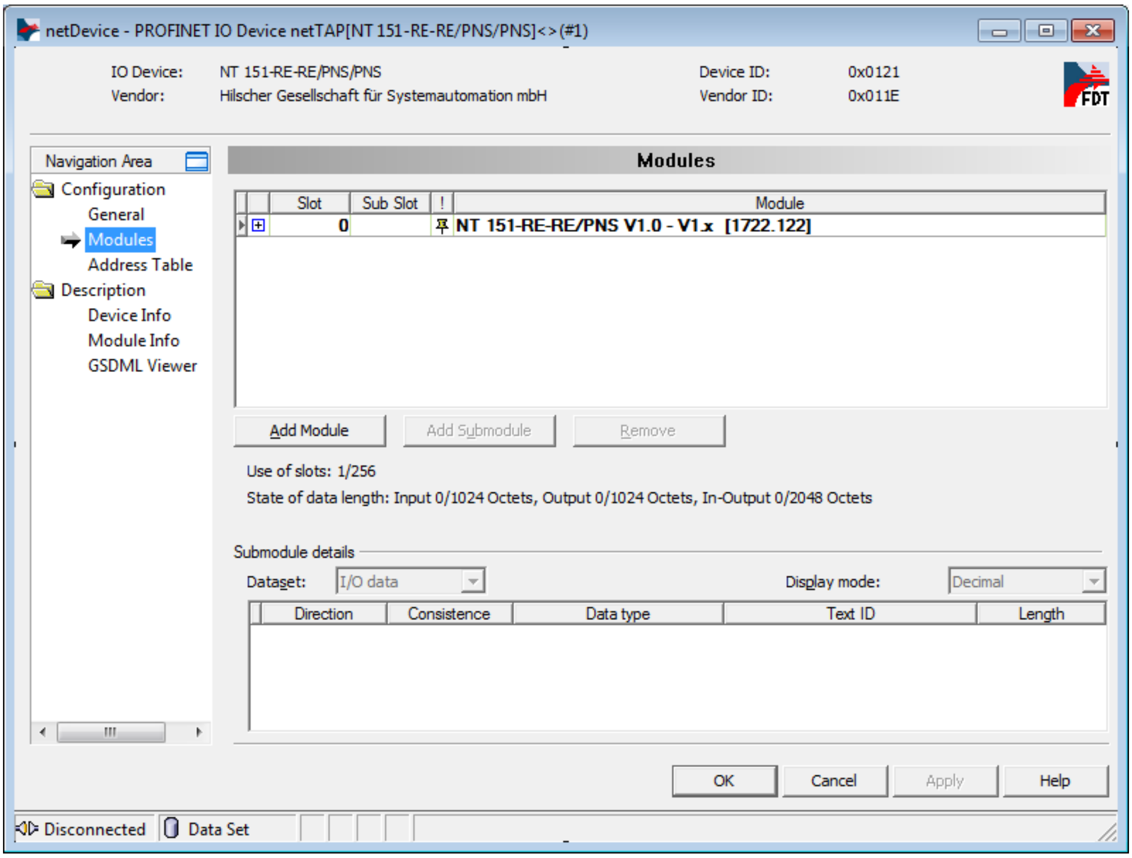


Figure 58: Configuring PROFINET IO Device at X2 (1)

2. Configure input data for primary network at Port X2.

➢ Click **Add Module** button to add a module for input data.

➢ In the **Modules** table, click on the newly added module entry, then select the number of input bytes for your gateway device at Port X2 from the drop-down list. For receiving e. g. 16 bytes of input data, select **16 Bytes Input** entry from the drop-down list.

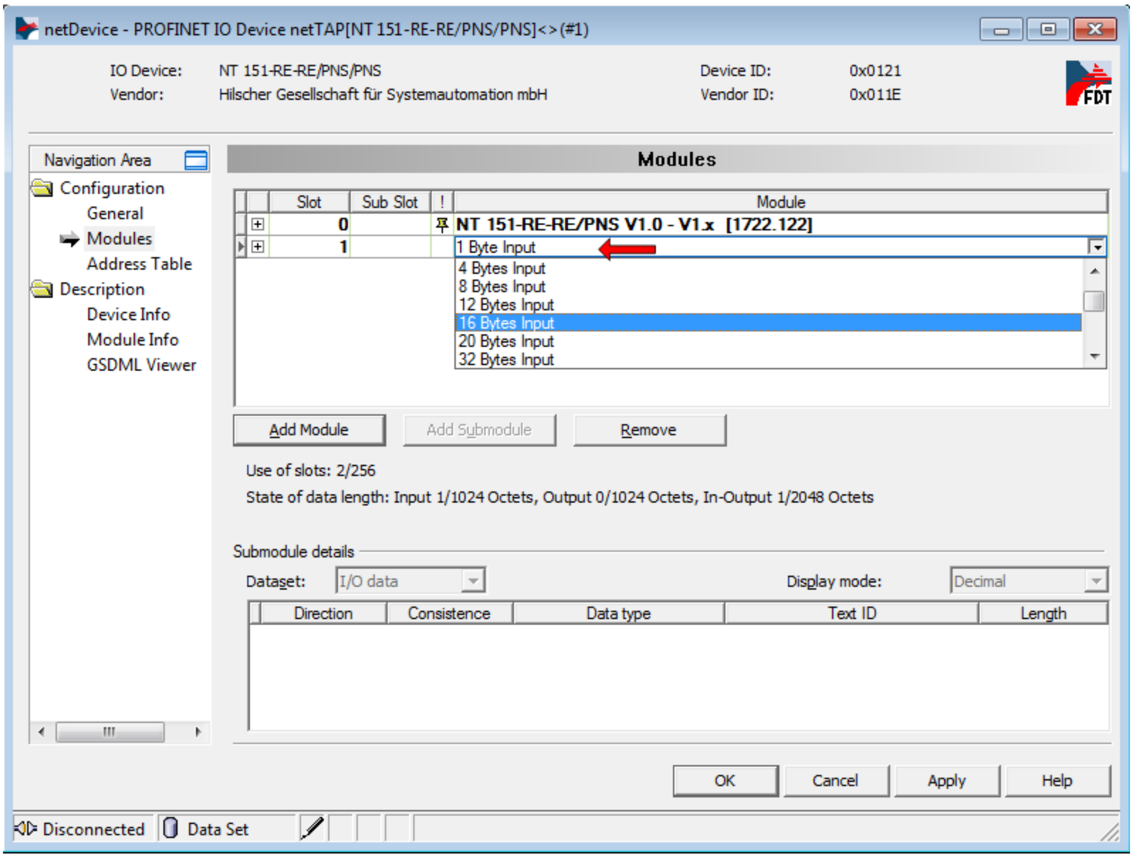


Figure 59: Configuring PROFINET IO Device at X2 (2)

3. Configure output data for primary network at Port X2.

➢ Click **Add Module** button again to add a module for output data.

➢ In the **Modules** table, click on the newly added second module entry, then select the number of output bytes for your gateway device at Port X2 from the drop-down list. For sending e. g. 32 bytes of output data, select **32 Bytes Output** entry from the drop-down list.

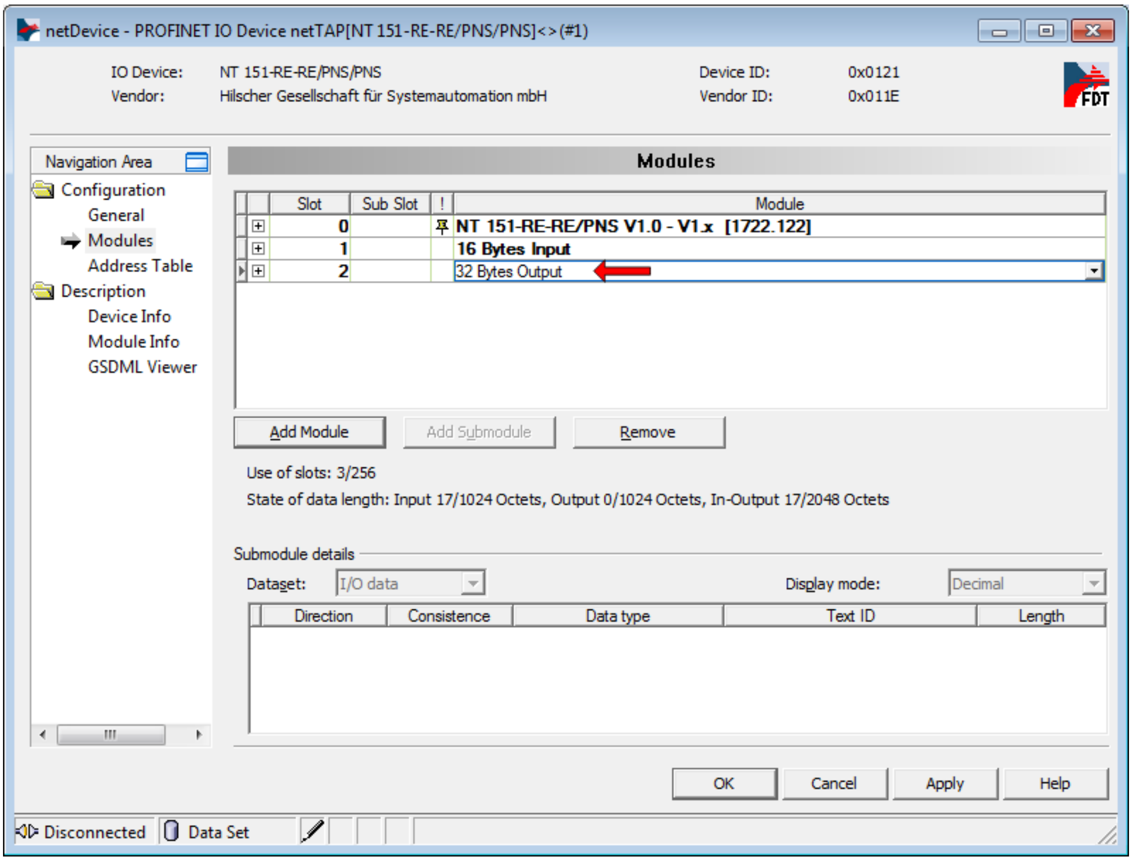


Figure 60: Configuring PROFINET IO Device at X2 (3)

Detailed information on how to configure a **PROFINET IO Device** at Port X2 can be found in the operating instruction manual *Generic DTM for PROFINET IO Devices*, DOC060305OIxxEN stored on the Gateway Solutions DVD in the directory `Documentation\english\1.Software\SYCON.net Configuration Software\Master Configuration\PROFINET IO Controller\IO Device Configuration`.

If you are using the **EtherCAT Slave** protocol at X2, see section *Configuration EtherCAT Slave* on page 149 for further information.

If you are using the **EtherNet/IP Adapter** protocol at X2, see section *Configuration EtherNet/IP Adapter* on page 153 for further information.

If you are using the **Sercos Slave** protocol at X2, see section *Configuration sercos Slave* on page 186 for further information.

As an alternative, you can open the corresponding online help by clicking the **Help** button in the opened configuration dialog window of the slave DTM, or by pressing the **F1** key on your keyboard.

4. Close the configuration window.

➢ Click **OK.**

➢ The configuration window closes.

2.5.5 Configuration of the secondary Network

1. Open the configuration window.

➢ In the context menu of the gateway device symbol, choose **Configuration > PROFINET IO Device** (choose the lower of the two **PROFINET IO Device** entries).

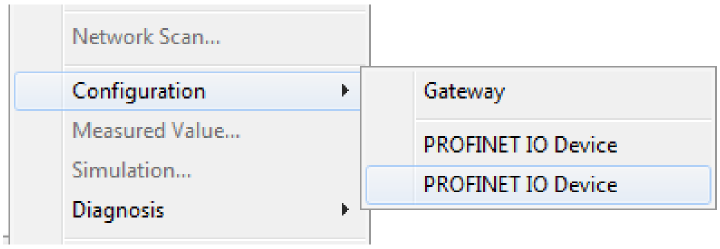


Figure 61: Open Port X3 via context menu

➢ The PROFINET IO Device configuration window opens for the secondary network at Port X3:

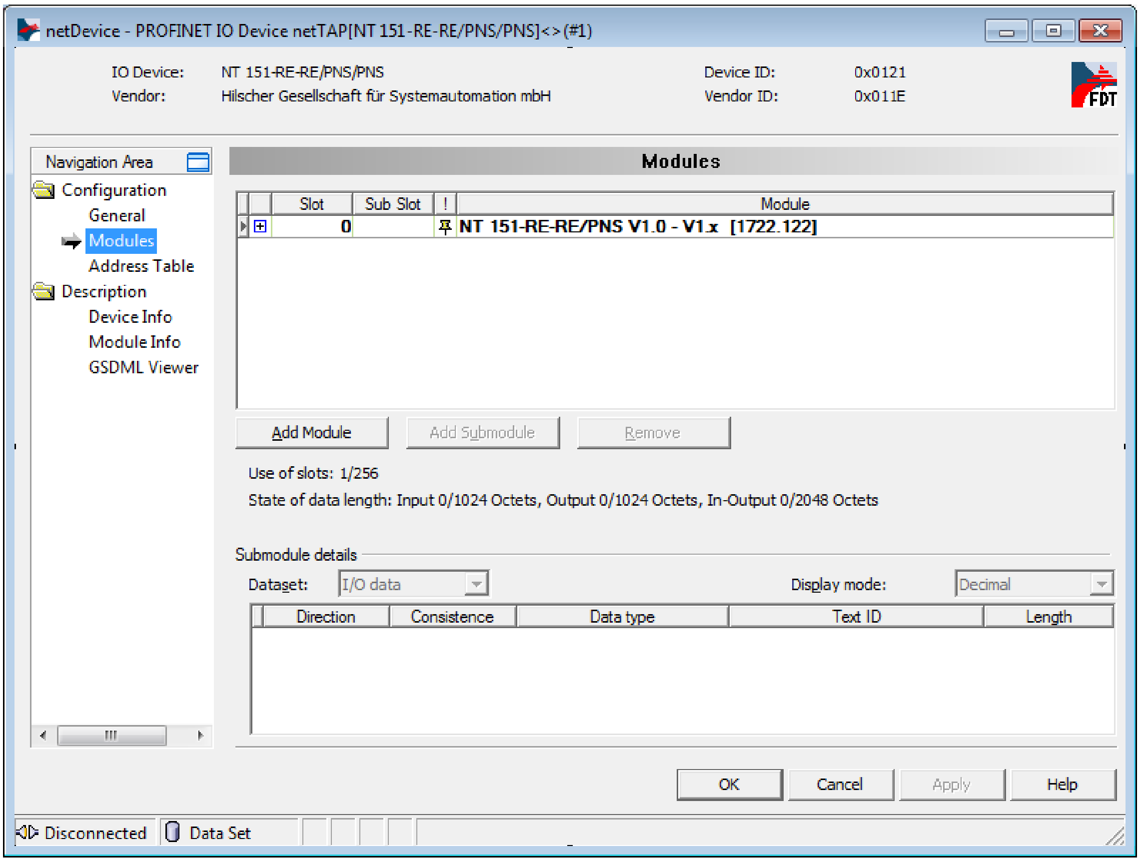


Figure 62: PROFINET IO Device Configuration Window

2. Configure input data for secondary network at Port X3.

➢ Click **Add Module** button to add a module for input data.

➢ In the **Modules** table, click on the newly added module entry, then select the number of input bytes for your gateway device at Port X3 from the drop-down list. For receiving e. g. 16 bytes of input data, select **16 Bytes Input** entry from the drop-down list.

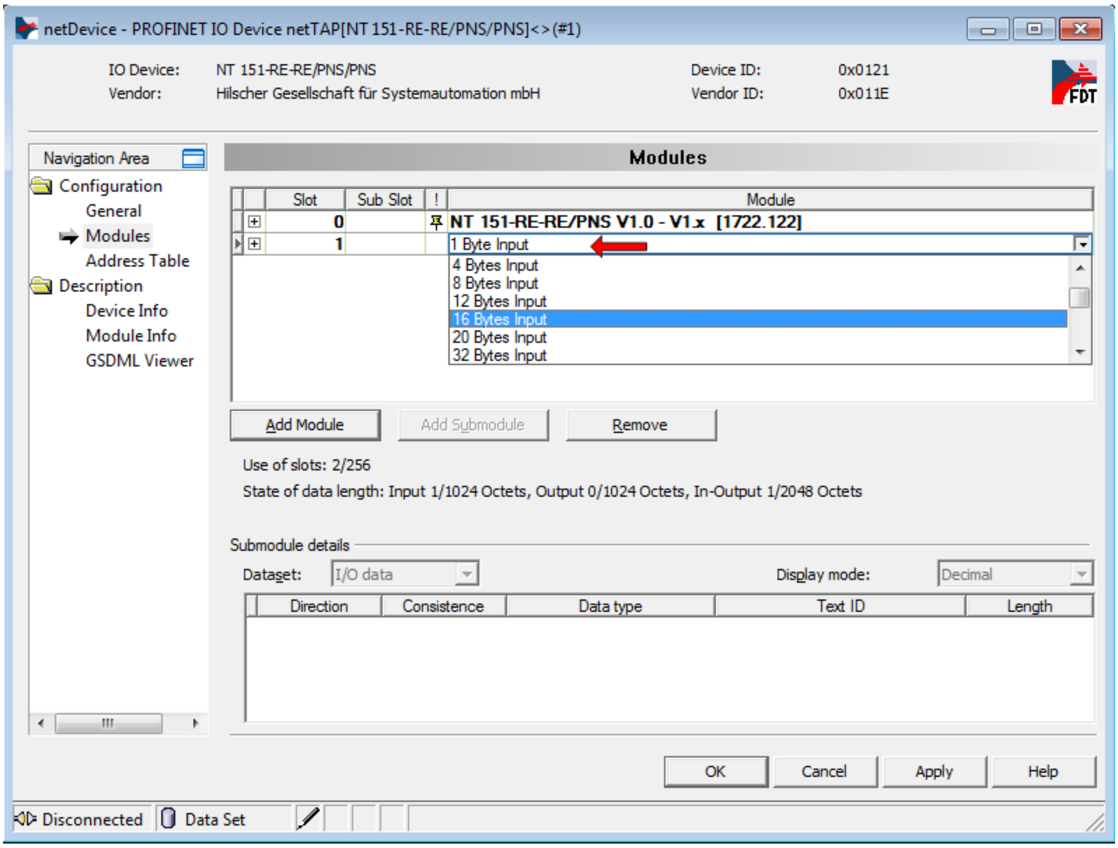


Figure 63: Configuring PROFINET IO Device at X3 (1)

3. Configure output data for secondary network at Port X3.

➢ Click **Add Module** button again to add a module for output data.

➢ In the **Modules** table, click on the newly added second module entry, then select the number of output bytes for your gateway device at Port X3 from the drop-down list. For sending e. g. 32 bytes of output data, select **32 Bytes Output** entry from the drop-down list.

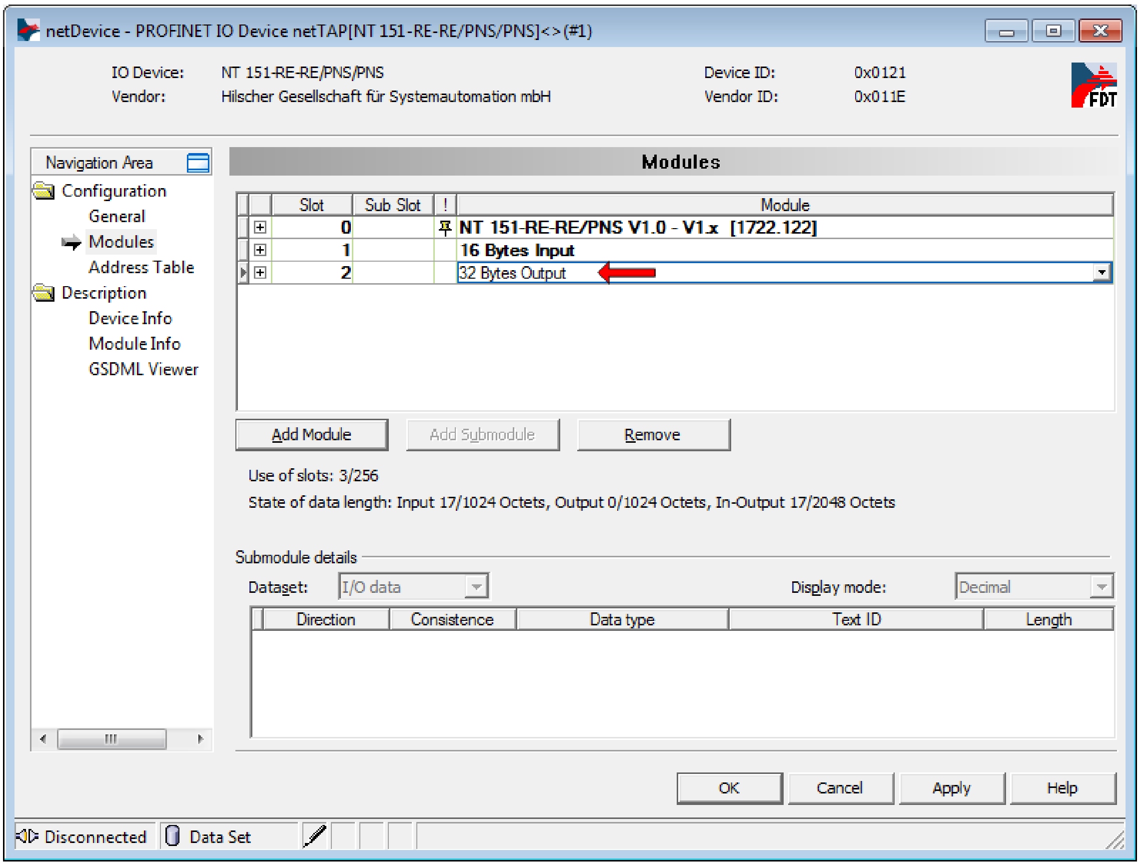


Figure 64: Configuring PROFINET IO Device at X3 (3)

Detailed information on how to configure a **PROFINET IO Device** at Port X3 can be found in the operating instruction manual *Generic DTM for PROFINET IO Devices*, DOC060305OIxxEN stored on the Gateway Solutions DVD in the directory `Documentation\english\1.Software\SYCON.net Configuration Software\Master Configuration\PROFINET IO Controller\IO Device Configuration`.

If you are using the **EtherCAT Slave** protocol at X3, see section *Configuration EtherCAT Slave* on page 149 for further information.

If you are using the **EtherNet/IP Adapter** protocol at X3, see section *Configuration EtherNet/IP Adapter* on page 153 for further information.

If you are using the **Sercos Slave** protocol at X3, see section *Configuration sercos Slave* on page 186 for further information.

As an alternative, you can open the corresponding online help by clicking the **Help** button in the opened configuration dialog window of the slave DTM, or by pressing the **F1** key on your keyboard.

Detailed information on how to configure a **master protocol** at Port 3 can be found in the operating instruction manual of the corresponding Master DTM. The Master DTM manuals are stored on the Gateway Solutions DVD in the directory `Documentation\english\1.Software\SYCON.net Configuration Software\Master Configuration\[protocol]`.
For EtherCAT Master, for example, you would need the operating instruction manual *DTM for Hilscher EtherCAT Master Device*, DOC080404OIxxEN.
As an alternative, you can open the corresponding online help by clicking the **Help** button in the opened configuration dialog window of the Master DTM, or by pressing the **F1** key on your keyboard.

4. Close the configuration window.

➢ Click on **OK.**

➩ The configuration window closes.

2.5.6 Signal Mapping

1. Open the Gateway configuration window.

➢ In the context menu of the gateway device symbol, choose **Configuration > Gateway**.

➩ The Gateway configuration window opens.

2. Open the **Signal Mapping** window.

➢ In the **Navigation area**, select **Configuration > Signal Mapping**.

➩ The **Signal Mapping** window opens:

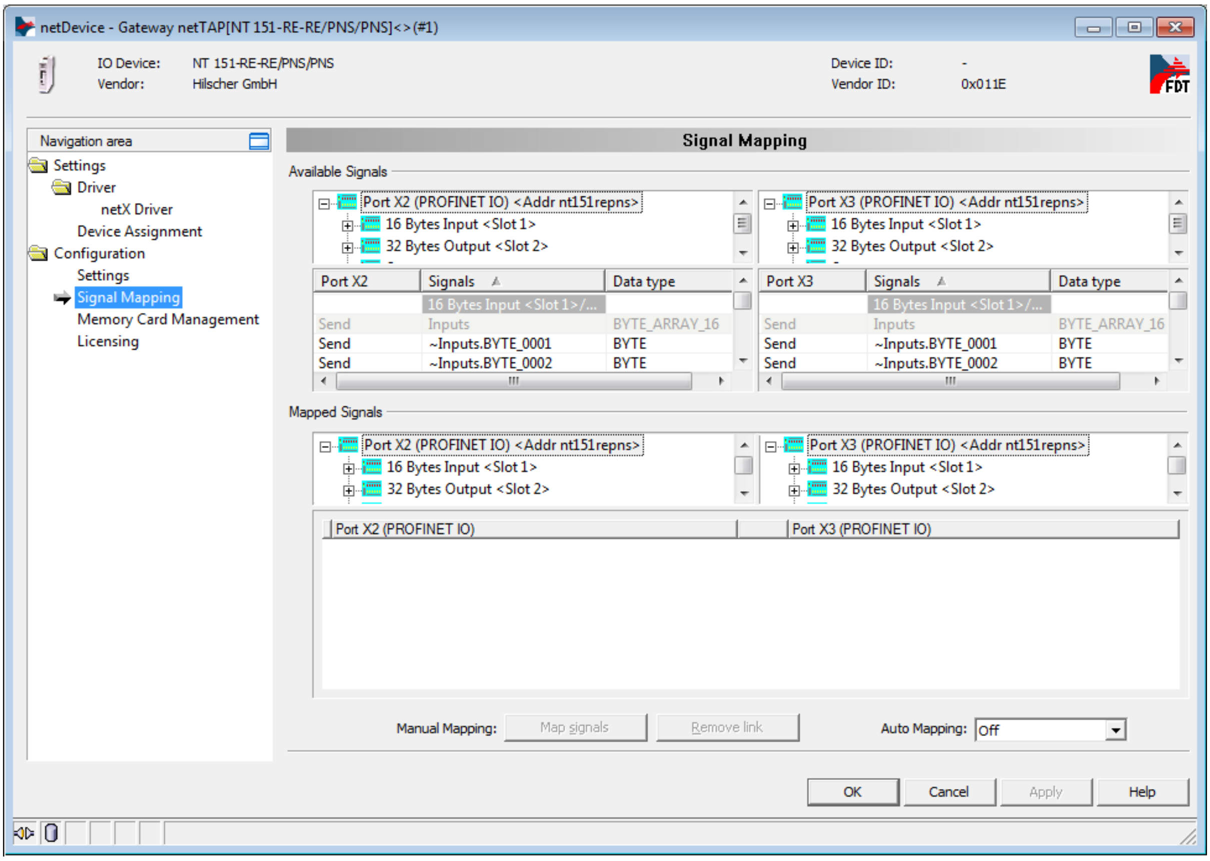


Figure 65: Gateway Signal Mapping

3. Configure data transfer from Port X2 to Port X3.

➢ Map the signals which are received on Port X2 (**Port X2 Receive**) with the signals which should be sent on Port X3 (**Port X3 Send**).

➢ For this, first select the received signal at Port X2 ①, then select the signal to be sent at Port X3 ②, then click **Map Signals** button ③.

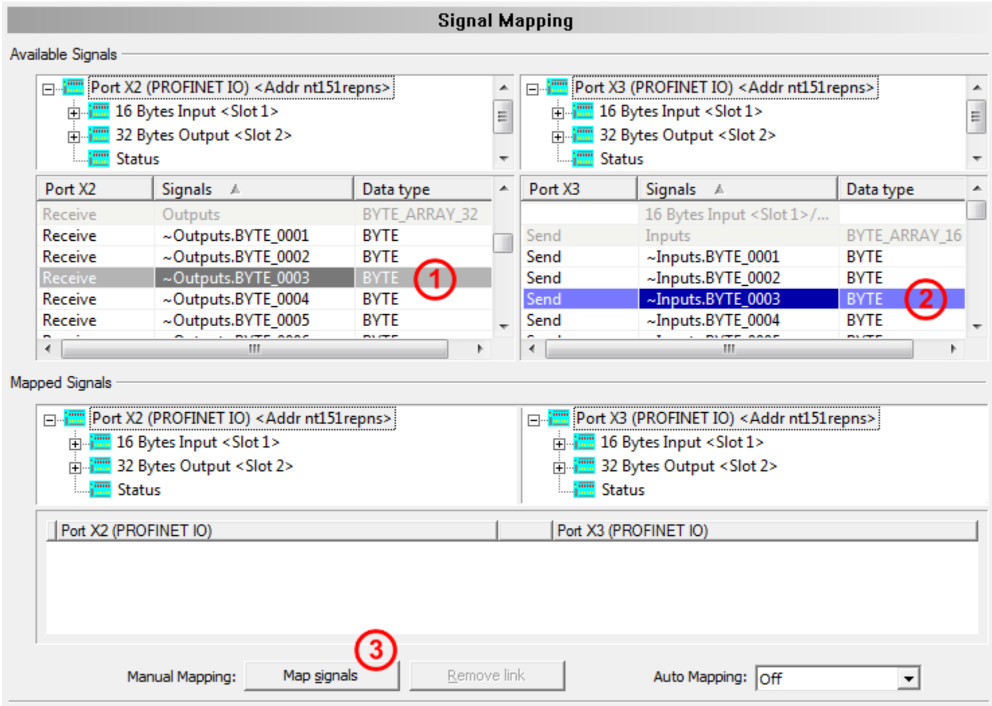


Figure 66: Map signals (1)

Note: You can also map the signals by using drag and drop. For this, drag the received signal (**Port X2 Receive**) and drop it on the signal which is to be sent (**Port X3 Send**).

➢ Repeat the procedure with the other required signals.

4. Configure data transfer from Port X3 to Port X2

➢ Map the signals which are received on Port X3 (**Port X3 Receive**) with the signals which should be sent on Port X2 (**Port X2 Send**).

➢ For this, first select the received signal at Port X3 ①, then select the signal to be sent at Port X2 ②, then click **Map Signals** button ③.

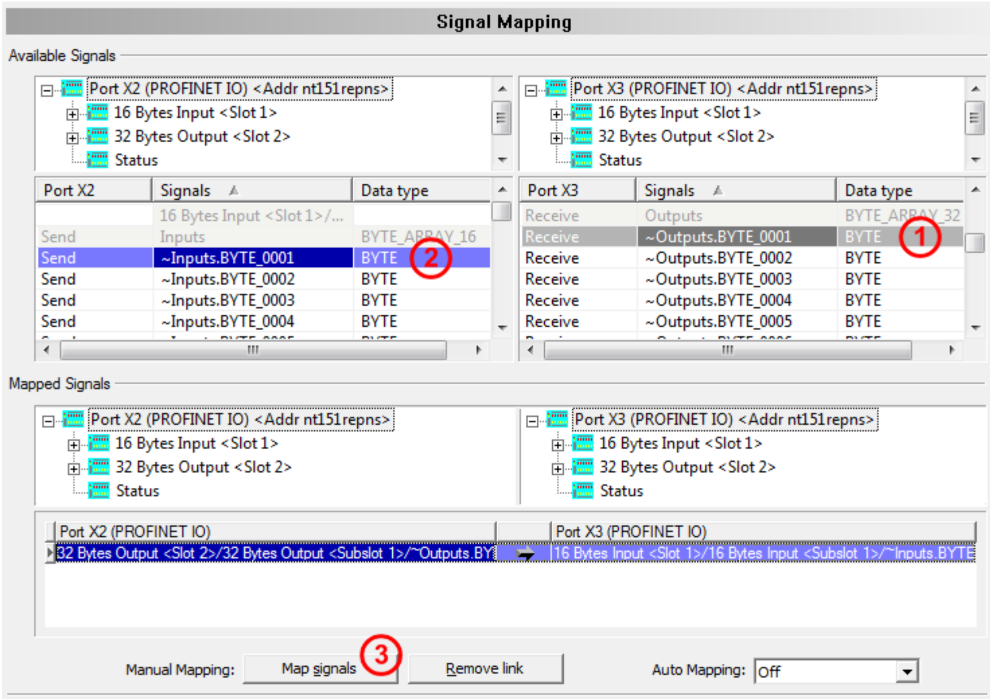


Figure 67: Map signals (2)

Note: You can also map the signals by using drag and drop. For this, drag the received signal (**Port X3 Receive**) and drop it onto the signal which is to be sent (**Port X2 Send**).

➢ Repeat the procedure with the other required signals.

5. Transfer status info of Port X2 to Port X3

➢ If necessary, map the internally generated status information of Port X2 (**Port X2 Generated**) to signals which should be sent on Port X3 (**Port X3 Send**)

➢ For this, select the status signal **Port X2 Generated** ①, then the signal which should be sent (**Port X3 Send**) ②, then click the **Map Signals** button.
(You can also map the signals by using “drag and drop”.)

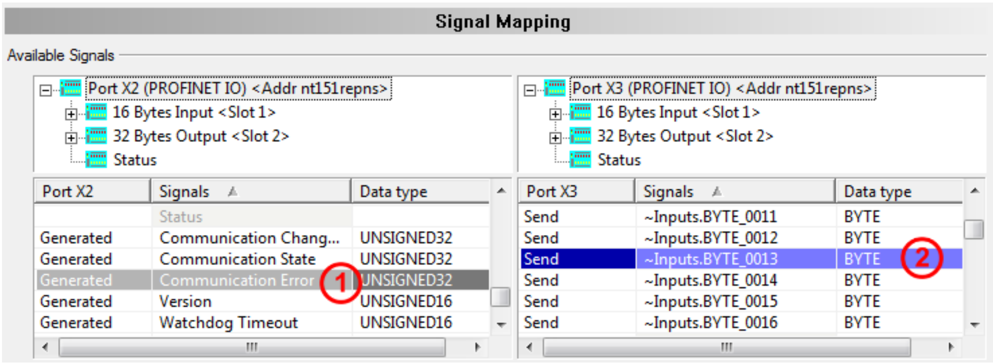


Figure 68: Map signals (3)

6. Transfer status info of Port X3 to Port X2

➢ If necessary, map the internally generated status information of Port X3 (**Port X3 Generated**) to signals which should be sent on Port X2 (**Port X2 Send**)

➢ For this, select the status signal **Port X3 Generated** ①, then the signal which should be sent (**Port X2 Send**) ②, then click the button **Map Signals**
(You can also map the signals by using “drag and drop”.)

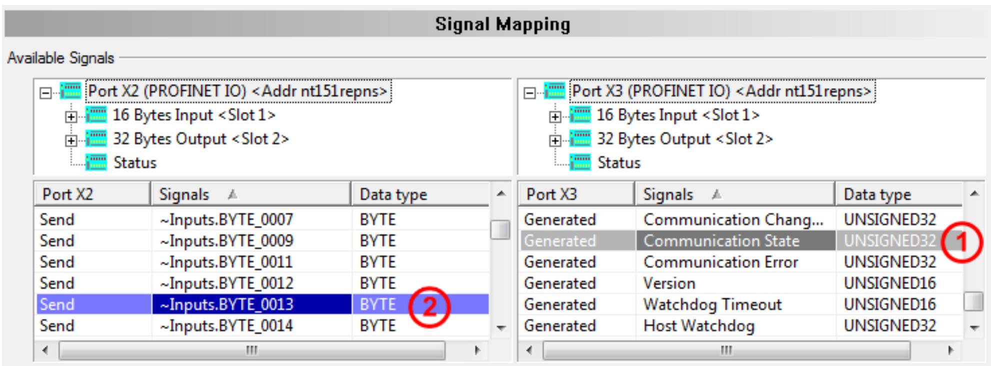


Figure 69: Map signals (4)

➢ An example of mapped signals shows the figure below:

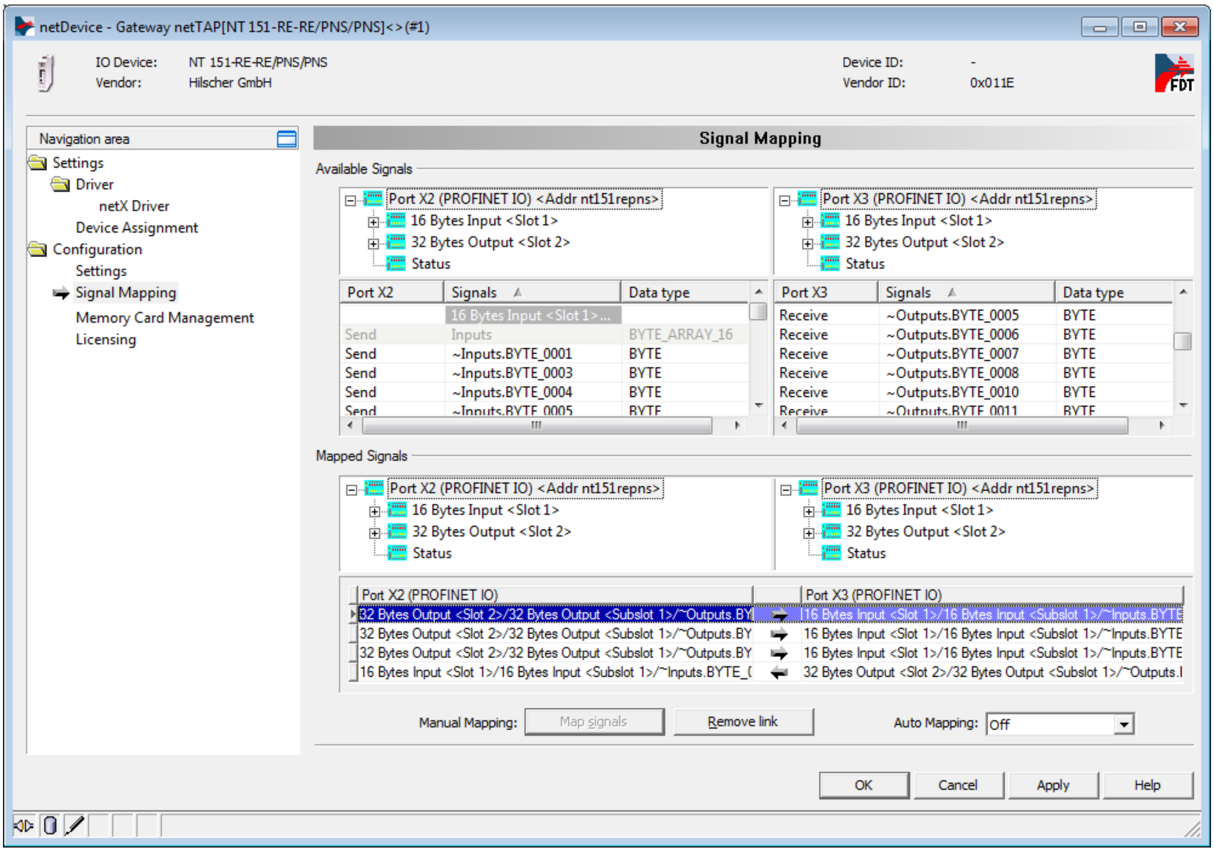


Figure 70: Gateway Signals mapped

2.5.7 Establish Connection to Gateway Device

1. Connect the netTAP device to the configuration PC via USB.

➢ Use a USB cable to connect the netTAP device to the configuration PC.

2. Open the Gateway configuration window.

➢ In the context menu of the gateway device symbol, choose **Configuration > Gateway**.

➩ The Gateway configuration window opens.

3. Select the driver.

➢ In the **Navigation area**, open **Settings > Driver** window, then check **netXDriver**.

➩ The following figure shows the selected driver:

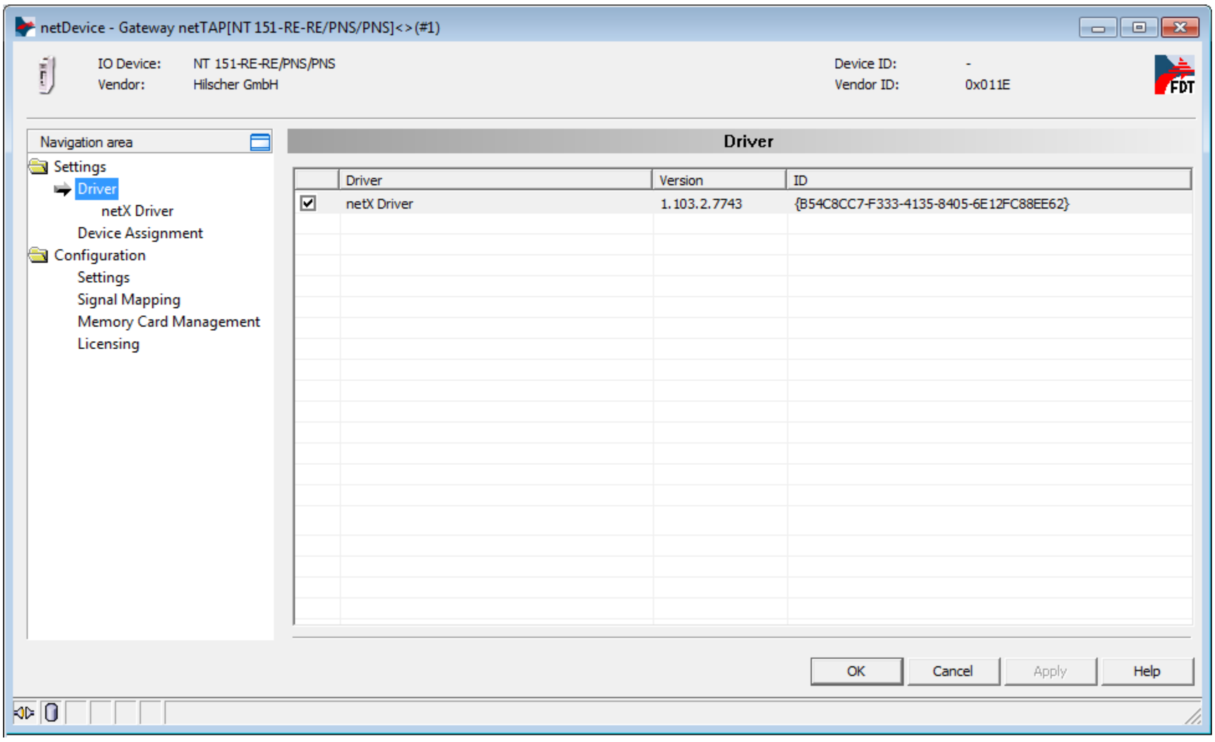


Figure 71: Select USB Driver (1)

4. Select the netX Driver.

➢ In the **Navigation area**, open **Settings > Driver > netXDriver** window.

➮ The following figure shows the selected driver:

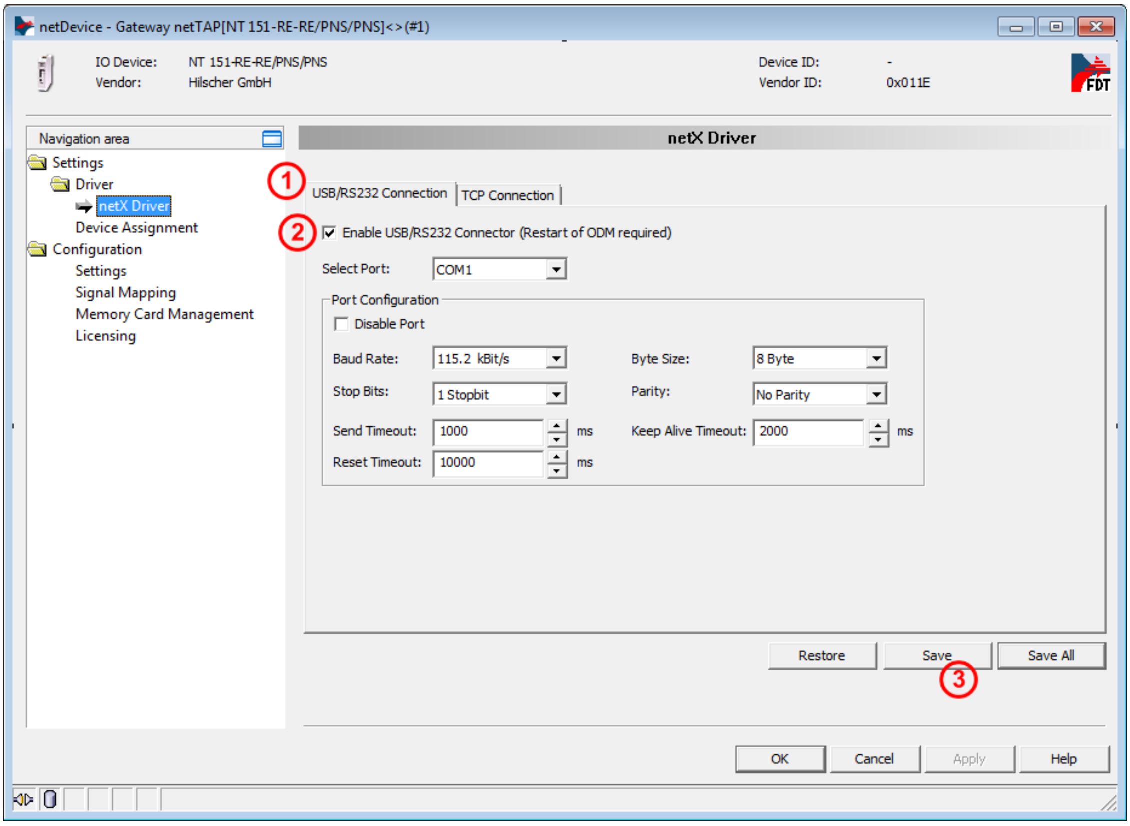


Figure 72: Select USB Driver (2)

5. Configure USB connection.

➢ ① Open the **USB/RS232 Connection** tab.

➢ ② Check **Enable USB/RS232 Connector**.

➢ ③ Click **Save** button.

Note: The standard settings of the netX Driver USB/RS232 can be used for netTAP devices without changes.

2.5.8 Device Assignment

- ➢ In the **Navigation area**, select **Settings > Device Assignment**.
- ➩ The netTAP DTM automatically starts searching for connected devices.
- ➢ Wait for a short while until the device is found and displayed in the list. If the device is not found, select **suitable only** entry from the **device selection** drop-down list, then click **Scan** button.

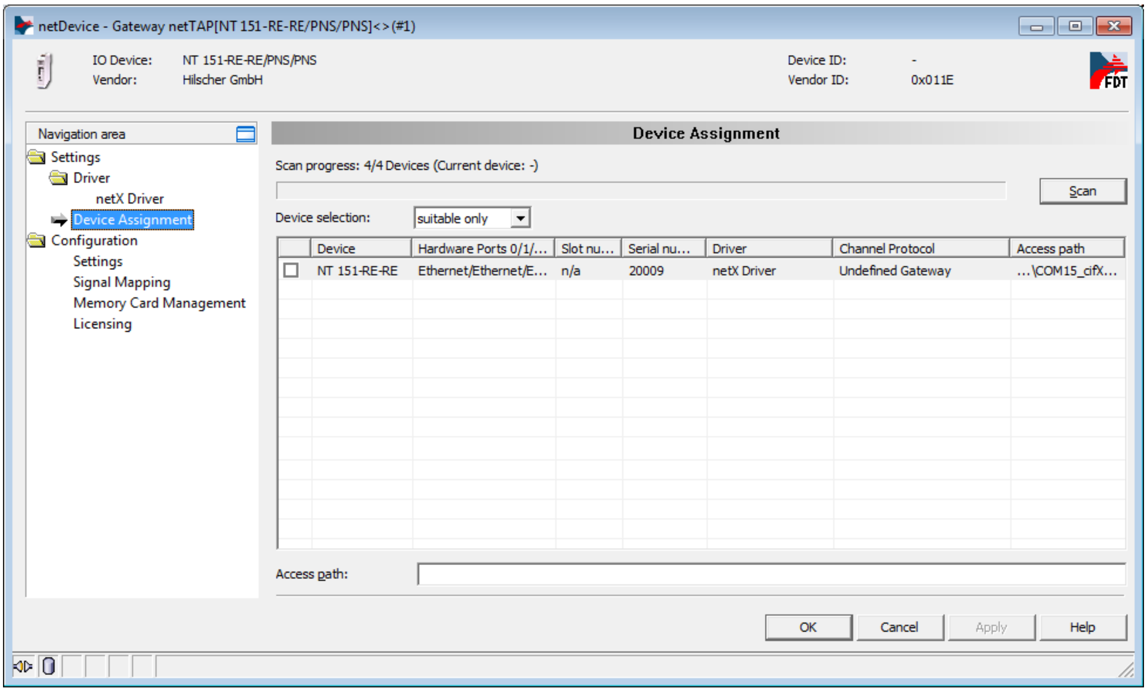


Figure 73: Found Device

➢ Select the NT 151-RE-RE device by checking the field in front of the device.

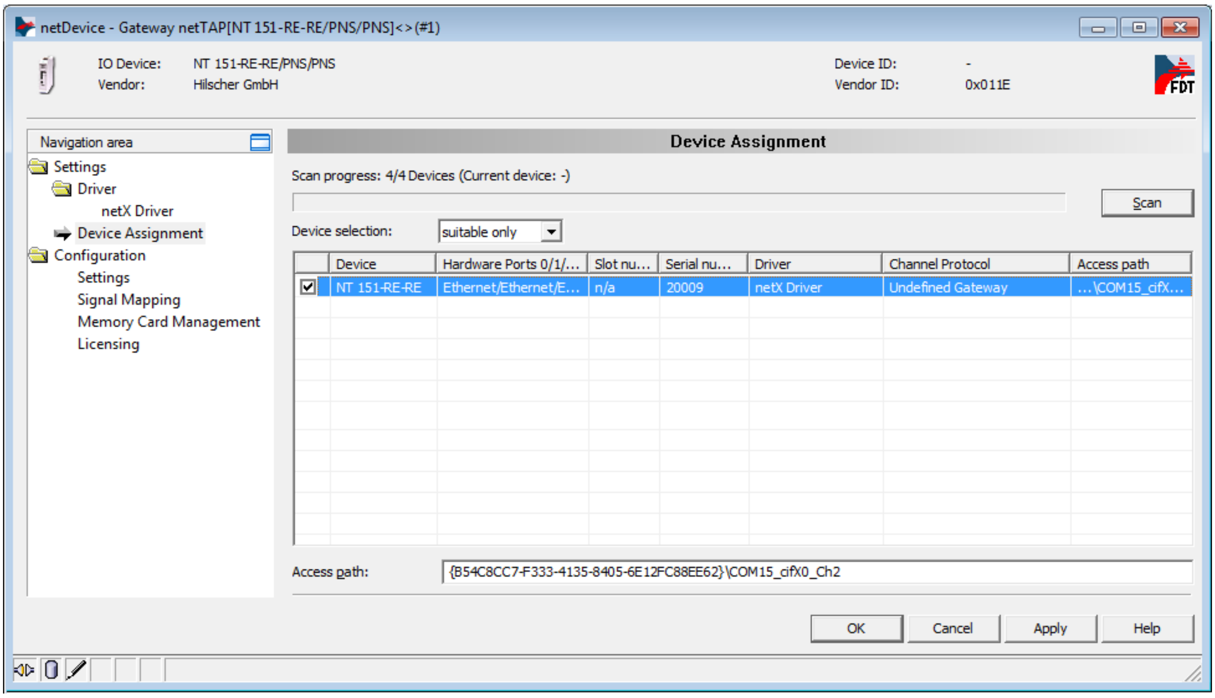


Figure 74: Found Device

➢ Click **OK**.

➩ The configuration window closes.

2.5.9 Loading or Updating Firmware (Firmware Download)

Note: The firmware download procedure decribed in this section is only necessary on first commissioning of a device delivered without pre-loaded firmware, or if updating of an old firmware file already existing in the device has become necessary.

In a firmware download, the firmware file is remanently stored in the flash memory of the netTAP device.

For downloading firmware, proceed as follows:

1. Open the Gateway configuration window.

➢ In the context menu of the gateway device symbol, choose **Configuration > Gateway**.

➩ The Gateway configuration window opens.

2. Open **Settings** window.

➢ In the **Navigation area**, select **Configuration > Settings**.

➩ The **Settings** window opens:

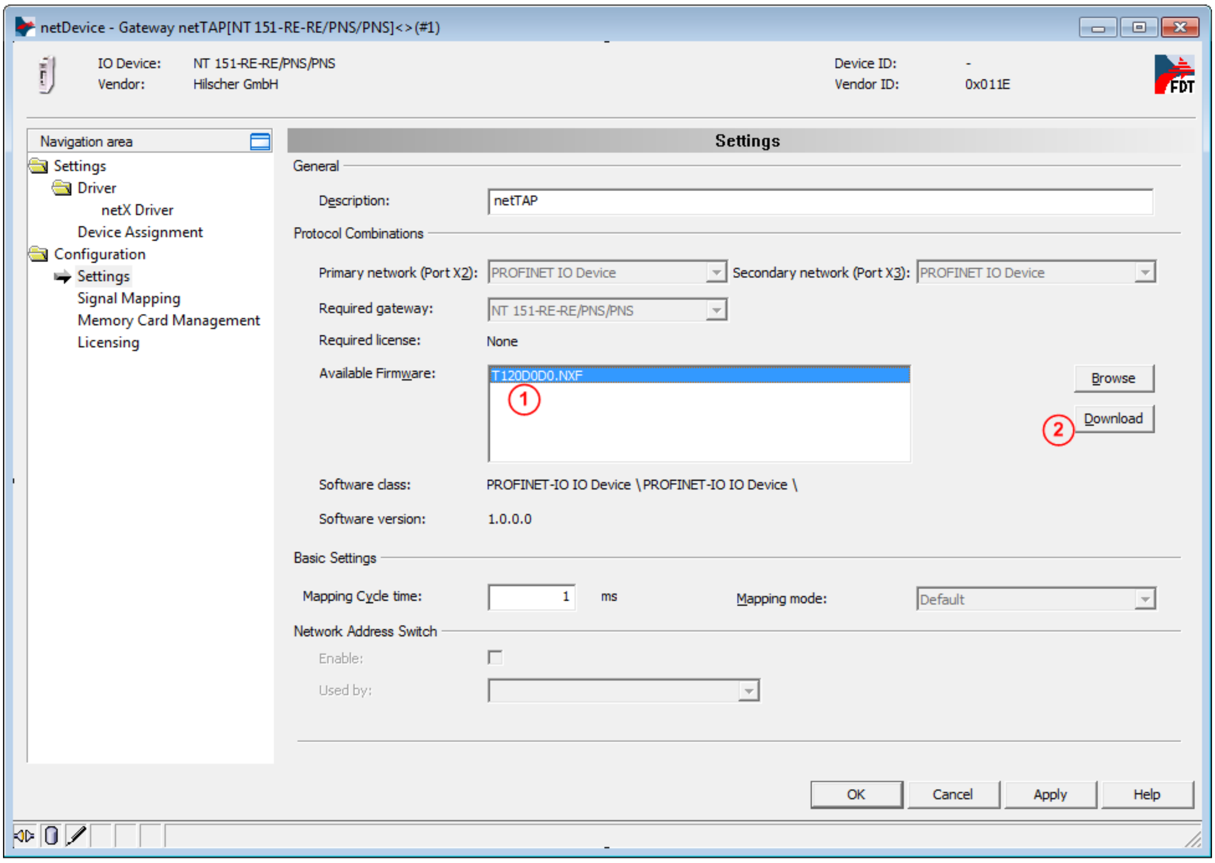


Figure 75: Firmware Download

3. Select Firmware.

➢ In the **Available Firmware** field, select the firmware file which you want to download ①. In this example, it is the T120D0D0.NXF file (firmware for protocol conversion PROFINET IO Device to PROFINET IO Device).

Note: The firmware files displayed in the **Available Firmware** field depend on the the chosen protocol combination. If you haven't selected any protocols in the **Primary network (Port X2)** and **Secondary network (Port X3)** drop-down lists yet, all loadable firmware files for the NT 151-RE-RE that are available on your PC will be displayed here.

4. Load firmware into the device.

➢ Click on **Download** ②.

➢ The firmware is loaded into the netTAP device

Important: Do not remove the cable during the firmware download. Do not disconnect the power supply of the device during the firmware download.

5. Download of firmware.

➢ Wait until the firmware was transferred completely into the device.

6. Close the Gateway configuration window after the download.

➢ Click **OK**.

➢ The Gateway configuration window closes.

2.5.10 Load Configuration

1. Create the netTAP configuration with SYCON.net as described in the previous sections.
2. Download configuration via USB.

➢ In the context menu of the gateway device symbol, select **Download** entry.

➢ Answer the security question with **Yes**, if the download should start.

➢ SYCON.net builds up an online connection to the device.

➢ The configuration is transferred info the gateway device.

➢ The device performs a reset and then starts with the new configuration.

2.5.11 Save Project

Save the SYCON.net project. In case of a device replacement, the saved project can be opened and re-edited (if necessary) with SYCON.net and then loaded into the spare device.

➢ To save a project, select menu **File > Save** respectively **File > Save As** or click on icon .

When you exit the program and the current configuration differs from the last saved configuration, then the following question appears:

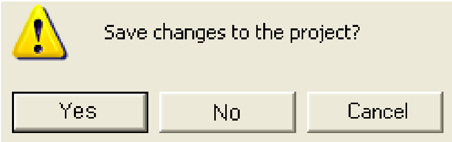


Figure 76: Security Question – Save Project

When you answer with **Yes**, then the project is saved. When you answer with **No**, then the project is not saved and the changes are lost. When you answer with **Cancel**, then the project is not saved.

2.5.12 Device Description File to configure the Master

The network master device, to which the **NT 151-RE-RE** device is to be connected, requires for its configuration a device description file containing the slave parameters of the **NT 151-RE-RE** for the corresponding protocol. These files are stored on the Gateway Solutions DVD in the folder: `Electronic Data Sheets (e.g. EDS,GSD,GSDML)\[protocol]`.

Thus the network master (in this example the PROFINET IO Controller), requires a GSDML file containing the parameters of the **NT 151-RE-RE** acting as PROFINET IO Device.

➢ To configure the PROFINET IO Controller use the following GSDML file:
`GSDML-V2.31-HILSCHER-NT 151-RE-RE PNS-20151021-1.xml`
This file is stored on the Gateway Solutions DVD in the folder:
`Electronic Data Sheets (e.g. EDS,GSD,GSDML)\PROFINET`.

➢ Load (import) the GSDML file into the configuration software of your PROFINET IO Controller.

2.6 Configuration of a NL 51N-DPL as Proxy

The netLINK Proxy NL 51N-DPL device can be used for the protocol conversion from PROFINET IO Device to PROFIBUS-DP Master with proxy functionality. The device can communicate to one PROFIBUS-DP Slave device.

The following sections describe how to configure the device.

2.6.1 Requirement for the Configuration

The device is configured via the Ethernet port. Therefore it is necessary that the device gets an IP address assigned before.

Therefore do the followings steps:

1. Establish an Ethernet connection between the Ethernet network port of your PC and the Ethernet port of the NL 51N-DPL.
2. Start the „Ethernet-Device Setup“ software. Therefore select **Start > Programs > SYCON.net System Configurator > Ethernet Device Setup.**
3. Search for connected devices. Therefore click on **Search Devices**. Devices are searched in the local network using broadcast telegrams.
4. Assign an IP address to the NL 51N-DPL device, which should be used for the device configuration.

   This address can be stored in a non volatile memory of the device.

Note: Use the IP address, which is later assigned by the PROFINET IO Controller, that you can use SYCON.net for diagnostic later.

2.6.2 Start SYCON.net and User Login

1. Start SYCON.net

➢ Select **Start > Programs > SYCON.net System Configurator > SYCON.net**

➩ SYCON.net is started

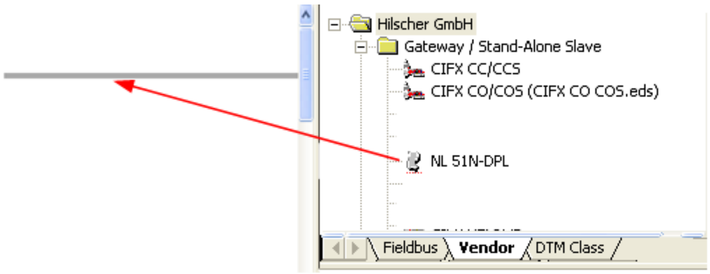
2. User Login

➢ In the window SYCON.net User Login click **OK** to login or enter your password and then click **OK** to login

➩ SYCON.net frame application appears

2.6.3 Insert the Proxy Device into the Configuration Window

➢ Go to the device catalog under vendor **Hilscher GmbH** to the category **Gateway / Stand-Alone Slave**. Use drag and drop with the NL 51N-DPL proxy device to insert it at the (gray) main line.



Or go to the device catalog under fieldbus **PROFINET IO** to the category **Gateway / Stand-Alone Slave**. Use drag and drop with the NL 51N-DPL proxy device to insert it at the (gray) main line.

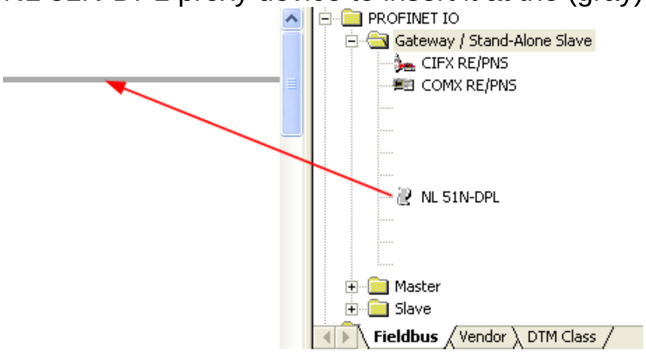


Figure 77: Insert Proxy Device into the Project

➩ The proxy device appears in the project

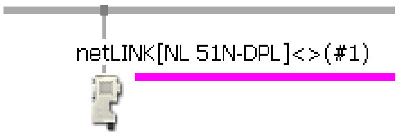


Figure 78: Proxy Device in the Project

2.6.4 Configuration of the secondary Network

If the slave device is not listed in the device catalog, then it has to be imported into the device catalog first.

2.6.4.1 Expand PROFIBUS-DP Slave Device Catalog

If you want to use a PROFIBUS-DP slave device that is not listed in the device catalog, then you have to import the GSD file into the SYCON.net device catalog.

➢ Proceed as follows:

1. Missing PROFIBUS-DP Slave devices can be added to the device catalog using the menu **Network > Import Device Descriptions**.
2. Select the new GSD file.
3. Answer the question “Do you want to reload the device catalog?“ with **Yes** to reload the catalog.

More information about the device catalog are in the document „SYCONnet_netDevice_en.pdf“ in the directory „Documentation“ on the product DVD.

2.6.4.2 Insert PROFIBUS-DP Slave Device to the PROFIBUS Network

➢ Go to the device catalog. Use drag and drop with one PROFIBUS-DP Slave device to insert it at the PROFIBUS bus line.

➯ The PROFIBUS-DP Slave device icon appears at the PROFIBUS network line (Secondary network)

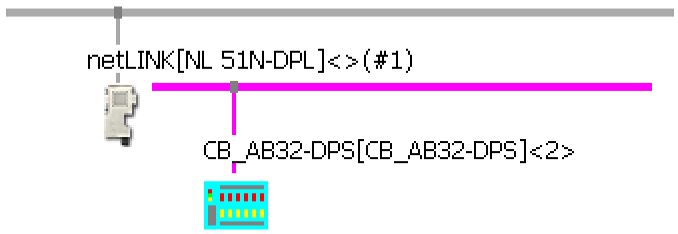


Figure 79: Proxy Device with Slave

Note: The NL 51N-DPL proxy device support communication only to one PROFIBUS-DP Slave device.

2.6.4.3 Configure the PROFIBUS-DP Slave device

- Open the configuration window with a double click on the device icon of the PROFIBUS-DP Slave
- ➯ The configuration window of the PROFIBUS-DP Slave device opens
- Select in the navigation area **Configuration > Modules**.
- Select from **Available Modules** the module(s) and add it/them to the **Configured Modules** to configure the Slave. The **Configured Modules** has to match the configuration of the used PROFIBUS-DP slave device.

More information about the configuration of PROFIBUS-DP Slave devices are in the document PROFIBUS_Generic_Slave_DTM_en.pdf on the product DVD in section *Configuration*.

2.6.4.4 Configure PROFIBUS-DP Master

1. Open the PROFIBUS-DP Master configuration window

- Select from the context menu of the proxy device symbol the entry **Configuration > PROFIBUS-DP Master**
- ➯ The PROFIBUS-DP Master configuration window opens

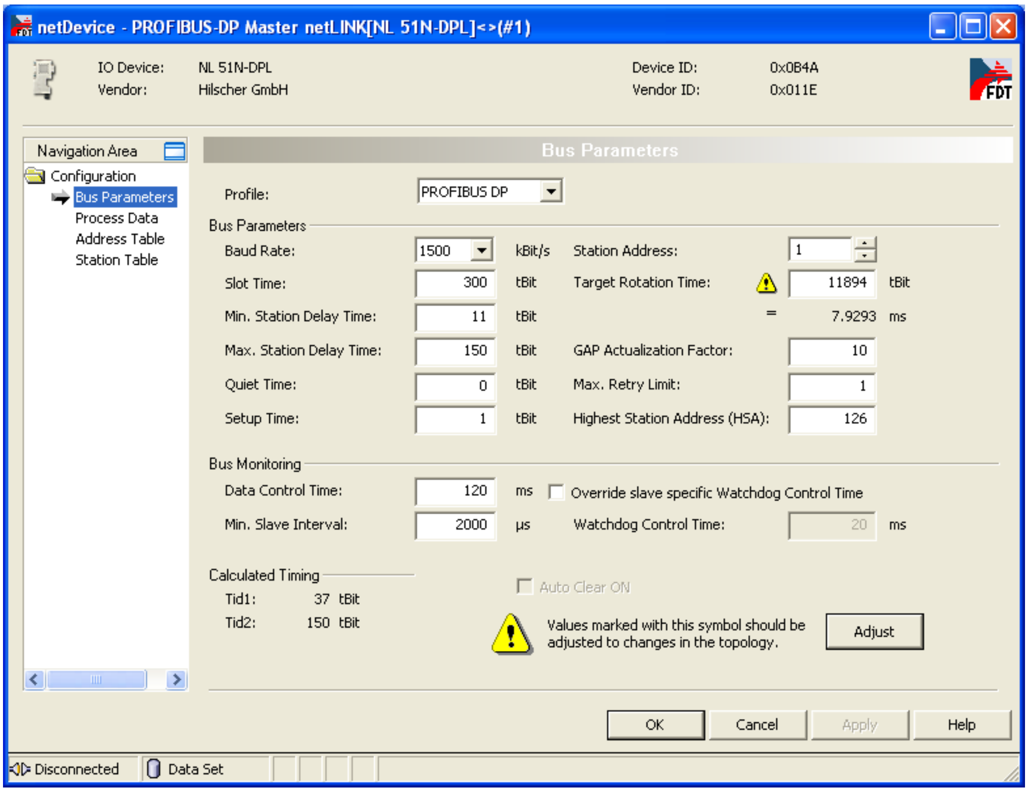
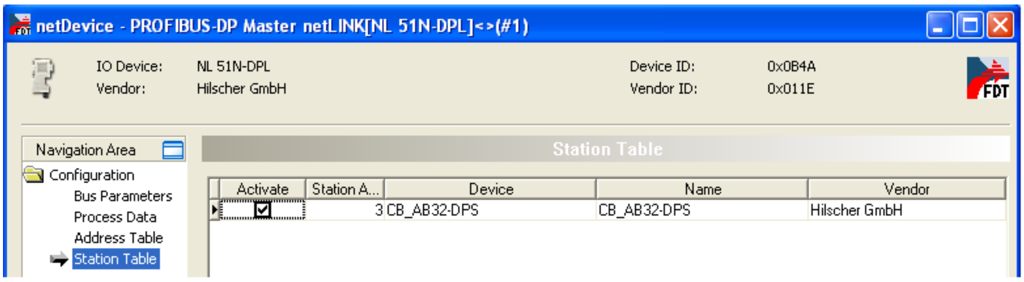


Figure 80: PROFIBUS-DP Bus Parameter

2. Configure PROFIBUS-DP Master

- Set the parameter. Set especially under **Configuration > Bus Parameter** the bus parameter and under **Configuration > Station Table** the station addresses (stations addresses of the PROFIBUS-DP Slave device).



More information about the configuration of PROFIBUS-DP Master devices are in the document PROFIBUS_Master_netX_DTM_en.pdf in section *Configuration.*

3. Close the configuration window

➢ Click on the button **OK**

➩ The configuration window closes

2.6.5 Configuration of the primary Network

The PROFINET IO network is at port X2 of the proxy device.

1. Open the PROFINET IO Device configuration window

➢ Select from the context menu of the proxy device symbol the entry **Configuration > PROFINET IO Device**

➪ The PROFINET IO Device configuration window opens and displays the modules

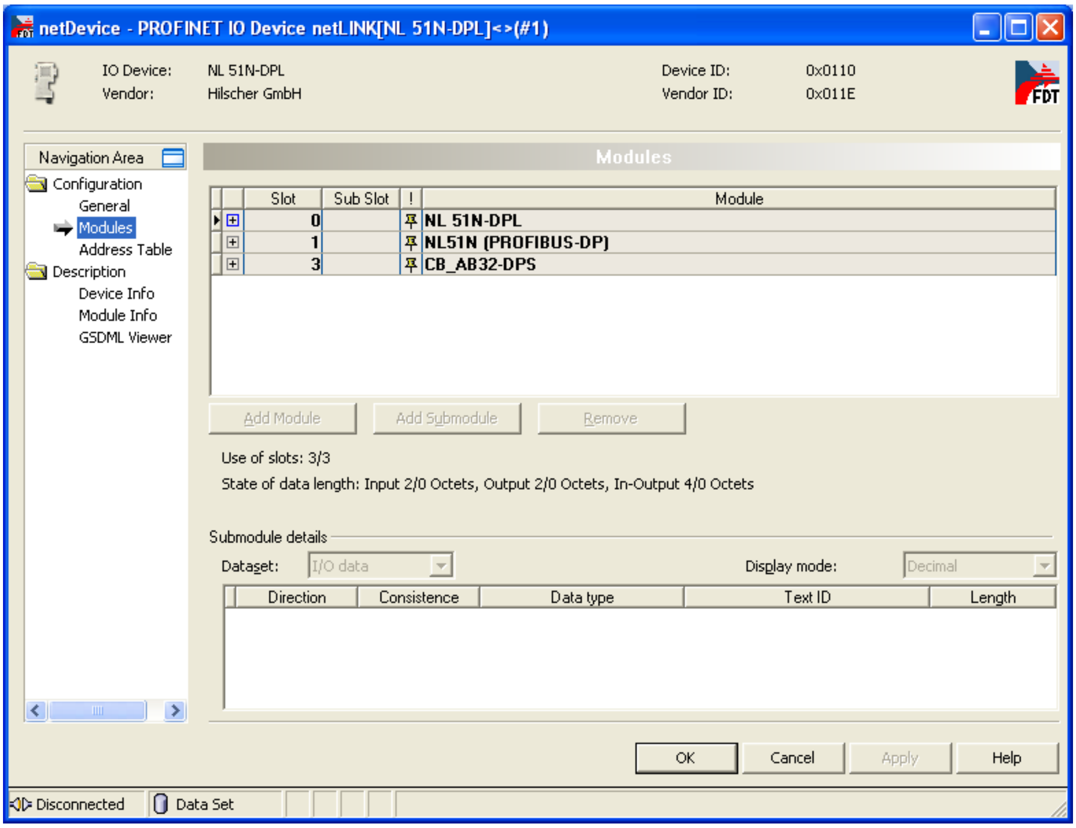


Figure 81: PROFINET IO Device Table

Here the modules with its slot number and sub slot number are displayed, as they will become visible at the PROFINET IO Controller. These slot numbers and sub slot numbers will be exported into the GSDML file, which is described later.

Slot 0 (NL 51N-DPL) and slot 1 (NL51N (PROFIBUS-DP)) are always present. An additional slot (slot 2 in this case, which represent the PROFIBUS-DP Slave with station address 2) is automatically displayed when a PROFIBUS-DP Slave was inserted to the PROFIBUS-DP network.

These Slot and sub slot numbers are used in the GSDML file. The GSDML file can be created (exported) which is described later.

2. Set the name of station

➢ Select in the navigation area **General**.

➩ The following configuration window opens

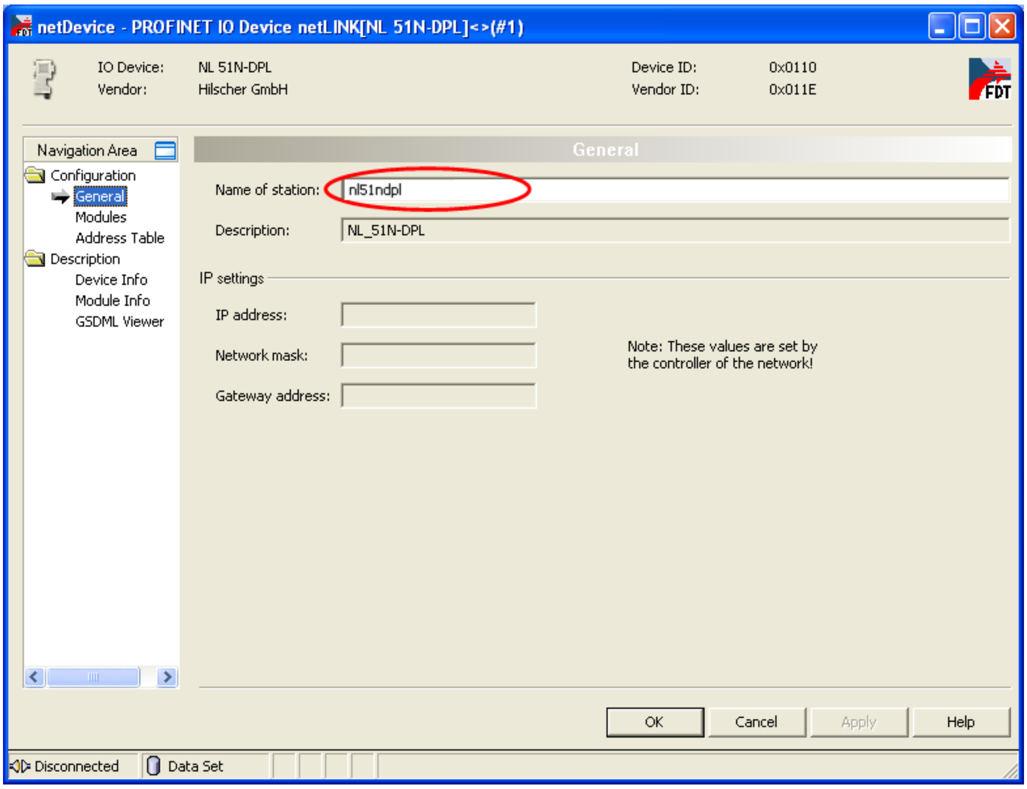


Figure 82: PROFINET IO Name of Station

➢ Enter the **Name of Station**

Note: The name of station is very important and is used by the PROFINET IO Controller to identify the proxy device.

3. Close the configuration window

➢ Click on the button **OK**

➩ The configuration window closes

2.6.6 Establish a Connection to the netLINK NL 51N-DPL

1. Connect an Ethernet cable to the NL 51N-DPL

➢ Connect an Ethernet cable to the Ethernet port of the NL 51N-DPL device and to the Ethernet port of the configuration PC. It is also possible to connect the NL 51N-DPL and the configuration PC to an Ethernet switch.

2. Set the IP address of the NL 51N-DPL device

➢ Set with the Ethernet Device setup software, which is part of the SYCON.net installation, the IP address for the NL 51N-DPL. With this software it is also possible to read an already assigned IP address from the NL 51N-DPL device.

The IP address is required for the communication with the SYCON.net software.

To communication from the SYCON.net software with the NL 51N-DPL device, an IP address has to be assigned before to the NL 51N-DPL device. How to assign an IP address is described in the document „Ethernet Device Configuration OI xx DE.pdf".

3. Open the Proxy configuration window

➢ Select from the context menu of the NL 51N-DPL symbol **Configuration > Proxy**

➪ The Proxy configuration window opens

4. Select driver

➢ Select in the navigation area **Settings > Driver** and then check **netX Driver**.

➪ The following figure shows the selected driver.

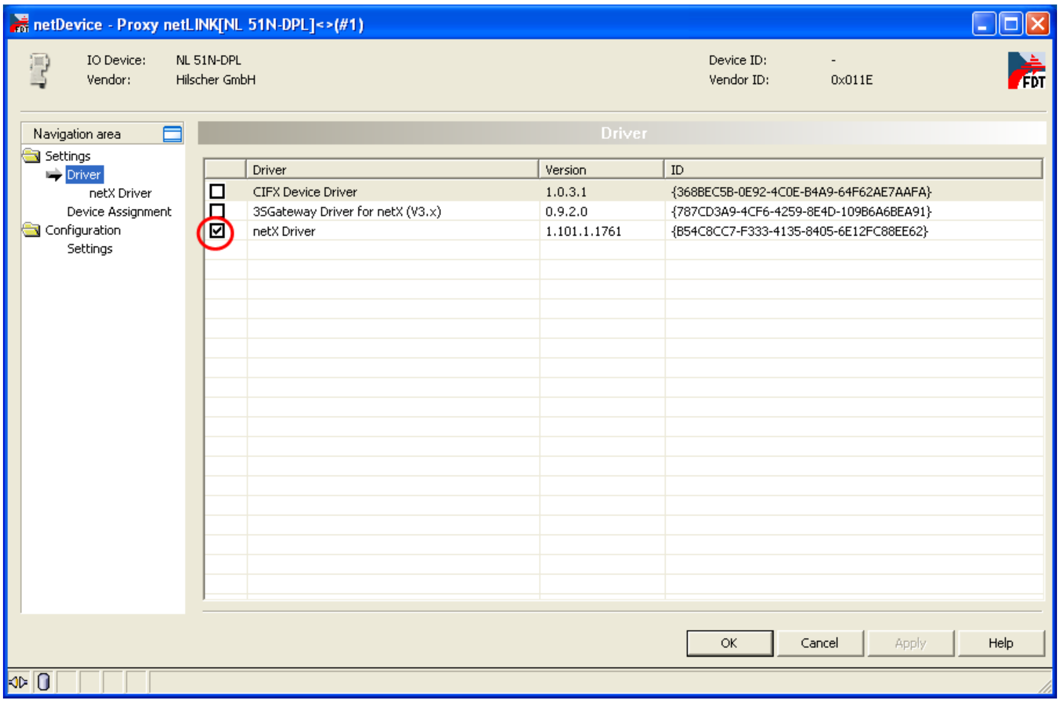


Figure 83: Select Driver

5. Set the IP search range

➢ Select in the navigation area **Settings > Driver > netX Driver**.

➩ The window for driver settings opens.

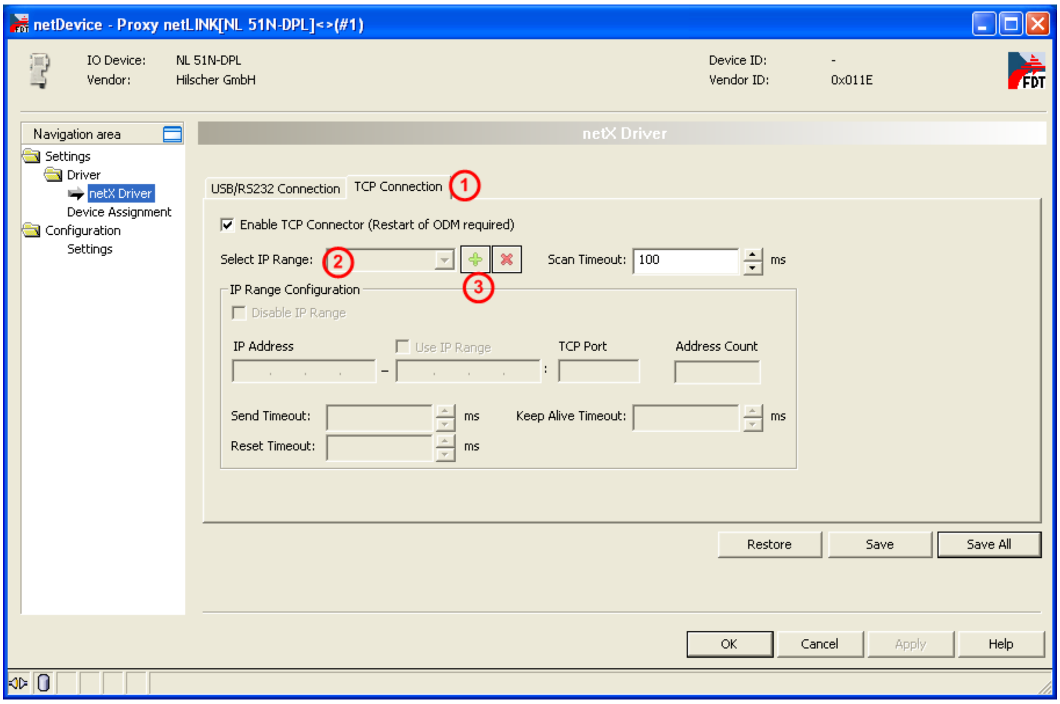


Figure 84: Set IP Address Search Range (1)

An IP address (one) or IP addresses (several, address range) are set in this window which SYCON.net uses to build up a connection to the proxy device.

➢ Select (as shown in the figure above):

① Select the tap **TCP Connection**.

② Is the **Select IP Range** grayed out (as shown above), then no IP range is defined.

Click on ③ **+** to add a new range, which enables the field below. Otherwise an address range configured earlier can be selected or by a mouse click on ③ **+** a new address range can be configured. Set the IP (start) address (and end address) the proxy devices is searched for. Make sure that the proxy device can be reached via Ethernet and is reachable via the address range.

➩ The fields to configure an IP address range are enabled.

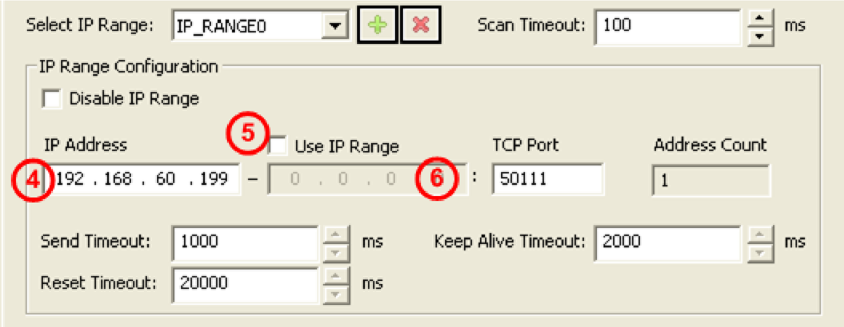


Figure 85: Set IP Address Search Range (2)

➢ You can enter one IP address of the proxy device at ④ as shown above or you can set an IP address range by a check at ⑤ **Use IP Range** and set the IP end address at ⑥. Make sure that the proxy device can be reached via Ethernet and is reachable via the address range.

Note: Set the IP address range not to large, which results in a long scan time.

➢ After the address range was set click on **Save**.

2.6.7 Device Assignment

- Select **Settings > Device Assignment**
- Click on ① **Scan**.
- The search process is started. Devices found are display in a list then.

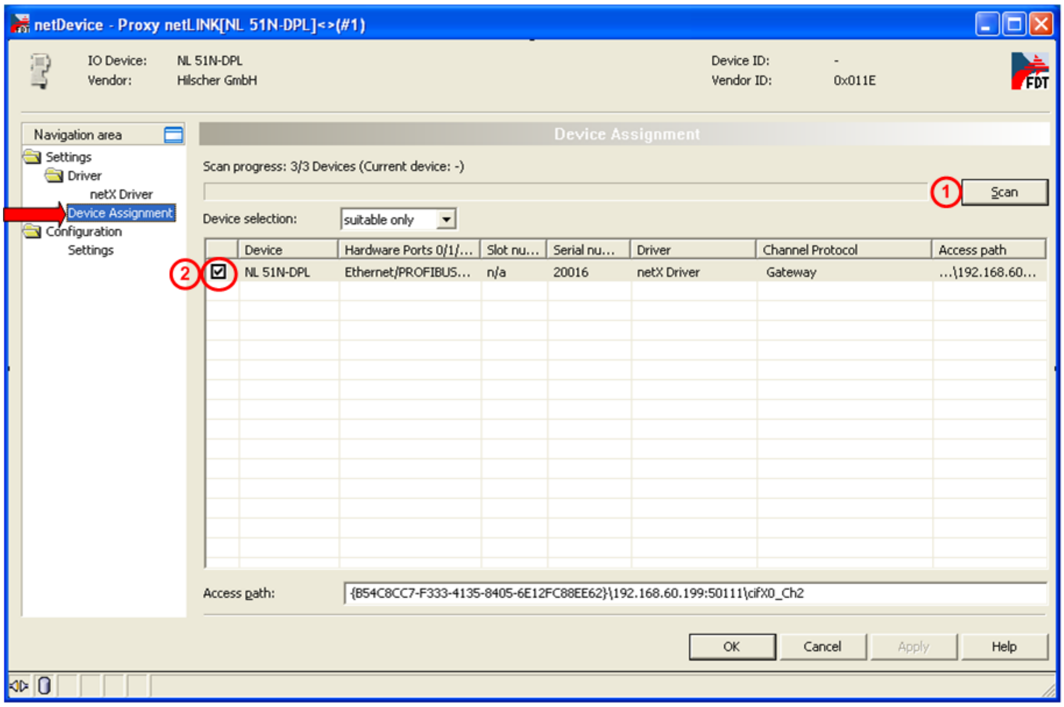


Figure 86: Select Device

- Select the device from the list ② by a check in the field of the device as shown above.
- Click on **OK**
- The configuration window closes.

2.6.8 Load Configuration

1. Download configuration

- ➢ Select from the context menu of the proxy device symbol the entry **Download**
- ➢ Answer the security question with **Yes**, if the download should start
- ➾ The configuration is transferred info the proxy device
- ➾ The device performs a reset and then starts with the new configuration.

Note: The device performs a reset after the download. Because of that, the Ethernet connection gets lost and (has to be stopped from SYCON.net and then) has to be established again.

Stop the online connection to the device, which was established automatically to the device before.

- ➢ Select from the context menu of the proxy device symbol the entry **Disconnect**.
- ➢ To establish a connection again, do the steps described in section *Device Assignment* on page 68.

2.6.9 Save Project

Note: The configuration downloaded from SYCON.net into the device can't be uploaded from the device. Only the SYCON.net project can be downloaded into additional devices.

Save the SYCON.net project. In case of a device replacement the saved project can be opened with SYCON.net and loaded into the device.

- To save a project select menu **File > Save** respectively **File > Save As** or click on icon .

When you exit the program and the current configuration differs from the last saved configuration, then the following question appears:

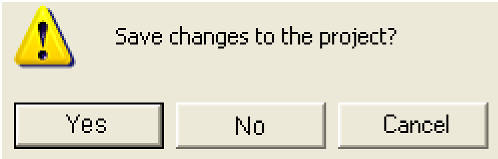


Figure 87: Security Question – Save Project

When you answer with **Yes**, then the project is saved. When you answer with **No**, then the project is not saved and the changes are lost. When you answer with **Cancel**, then the project is not saved.

2.6.10 Device Description File to configure the PROFINET IO Controller

A GSDML file is required to configure the PROFINET IO Controller. This file is creates as follows:

- ➢ Select from the context menu of the proxy device symbol the entry **Additional Functions > PROFINET IO Device > Export GSDML**.
- ➾ The dialog to save the XML file appears.
- ➢ Select the directory the GSDML file should be saved into.
- ➢ A name for the GSDML file is proposed. Accept it or change it useful.
- ➢ Click on **Save.**
- ➾ The GSDML file is saved.

The structure of the file name is:

1 Fixed prefix

2 Version number of the GSDML specification

3 Manufacturer name

4 Device type

5 Date: Format yyyymmdd

6 Time: Format hhmmss

7 File extension (always): xml

- ➢ Use the GSDML file to configure the PROFINET IO Controller

2.6.11 Update Firmware

To update the firmware an Ethernet connection to the device is necessary. To establish a connection, do the steps described in section *Device Assignment* on page 68.

To update the firmware proceed then as follows:

➢ Select from the context menu of the NL51N-DPL the entry **Configuration > Proxy**.

➩ The following window opens

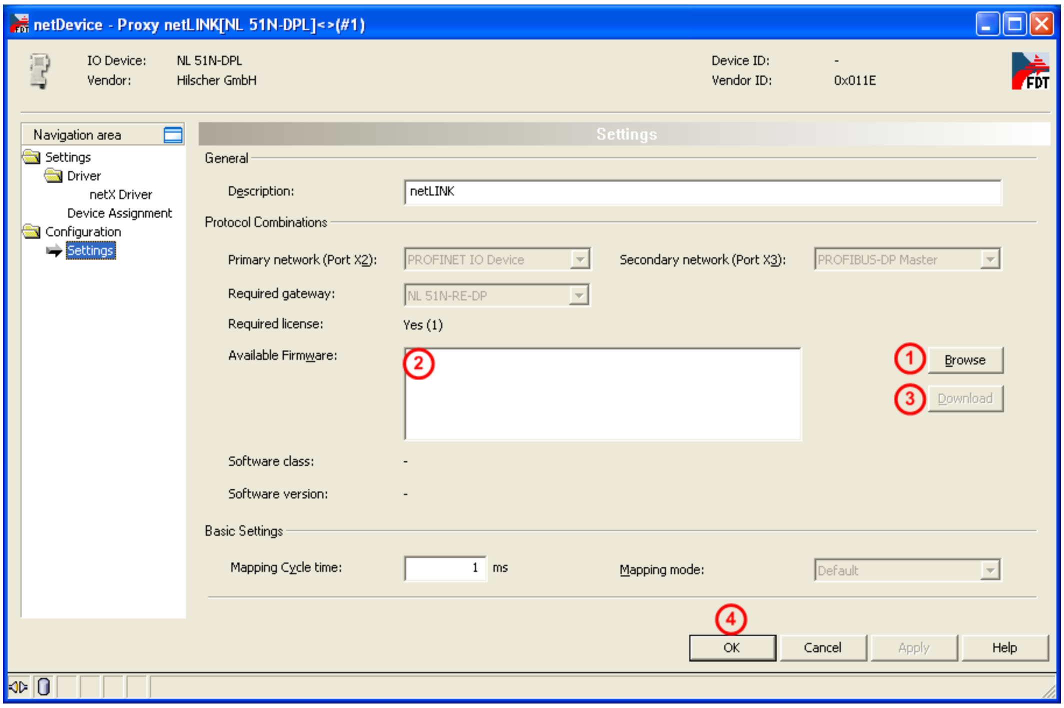


Figure 88: Firmware update

➢ Click with the left mouse on ① **Browse**

➩ The file dialog of the operating system opens.

➢ Select the new firmware file.

➩ The file name is displayed at ②.

➢ Select the new firmware file at ②. The file name is L51NPSDL.NXF.

➢ Click with the left mouse on ③ **Download**.

➩ The firmware file is transferred into the NL 51N-DPL.

➢ Close the dialog with ④ **OK**.

3 Settings

3.1 Overview Settings

Settings Dialog Panes

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

Section	Subsection	Page
Driver		112
	Selecting the Driver	112
	Configuring the Driver	113
	netX Driver	114
Device Assignment		121
	Scanning for Devices	121
	Selecting the Device	123

Table 7: Descriptions Settings

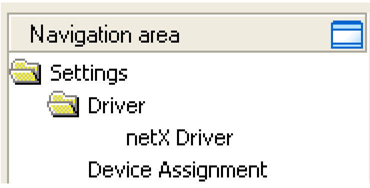


Figure 89: Navigation Area - Settings

Note: To edit the **Settings** dialog panes you need *User Rights* for "Maintenance".

3.2 Driver

The devices netTAP NT 100, netTAP NT 151-RE-RE and netBRICK NB 100 are configured via an USB interface of the device or diagnosis is performed via the USB interface. The devices netTAP NT 50 and netLINK NL 51N-DPL are configured via an Ethernet interface of the device or diagnosis is performed via the Ethernet interface. Therefore a driver is necessary.

The **Driver** dialog pane displays the driver/s to be used for a connection from the gateway DTM to the device.

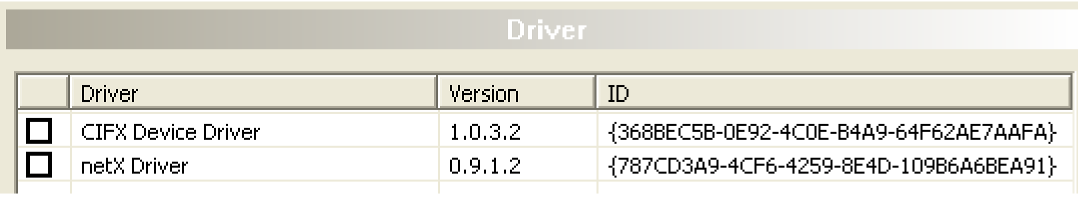


Figure 90: Driver Selection List

Parameter	Meaning
Driver	Name of the driver
Version	Version of the driver
ID	ID of the driver (driver identification)

Table 8: Driver Selection List Parameters

3.2.1 Selecting the Driver

To establish a connection you need to select a driver first.

Note: Not all the drivers displayed necessarily support the device. The used driver must be supported by the device or must be available for the device. Use the netX Driver for the devices NT 50, NT 100, NT 151-RE-RE, NB 100 and NL 51N-DPL.

To select the driver/s to be used:

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

➾ The available drivers are listed on the **Driver** dialog pane.

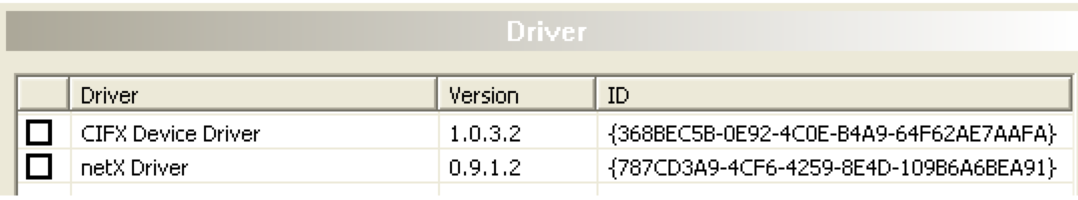


Figure 91: Driver Selection List - recognized or installed Drivers

2. Check the checkbox for the driver/s in the selection list.

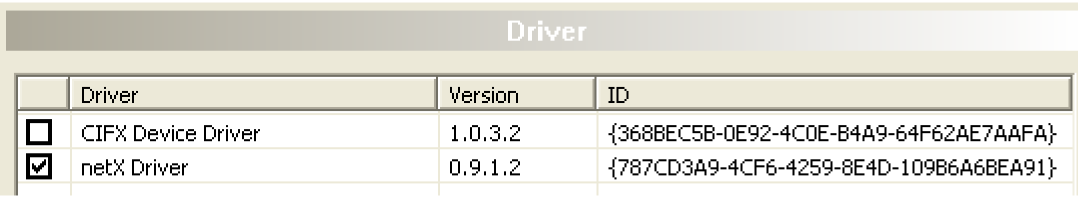


Figure 92: Driver Selection List - Driver/s selected

3.2.2 Configuring the Driver

All drivers for which you can access to an individual driver dialog pane are displayed in the navigation area under the **Driver** entry. In the driver dialog panes you can configure the driver settings.

Figure 93: Navigation Area – Settings

To configure a driver:

1. Select **Settings > Driver >** [*Name of the driver*].

➢ The corresponding driver dialog pane is displayed.

2. Configure the driver settings.

➢ To get detailed information how to configure the single drivers, please refer to the corresponding description of the driver.

You can access to the single descriptions of the drivers as online help via the DTM user interface (**F1** key).

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

➢ Select **Settings > Driver >** [*Name of the driver*].

➢ Press the **F1** key.

3.2.3 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.

3.2.3.1 netX Driver - USB/RS232 Connection

Function and use

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 to the device via the USB interface or via the RS232 interface of the device. 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 at the PC.

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

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:

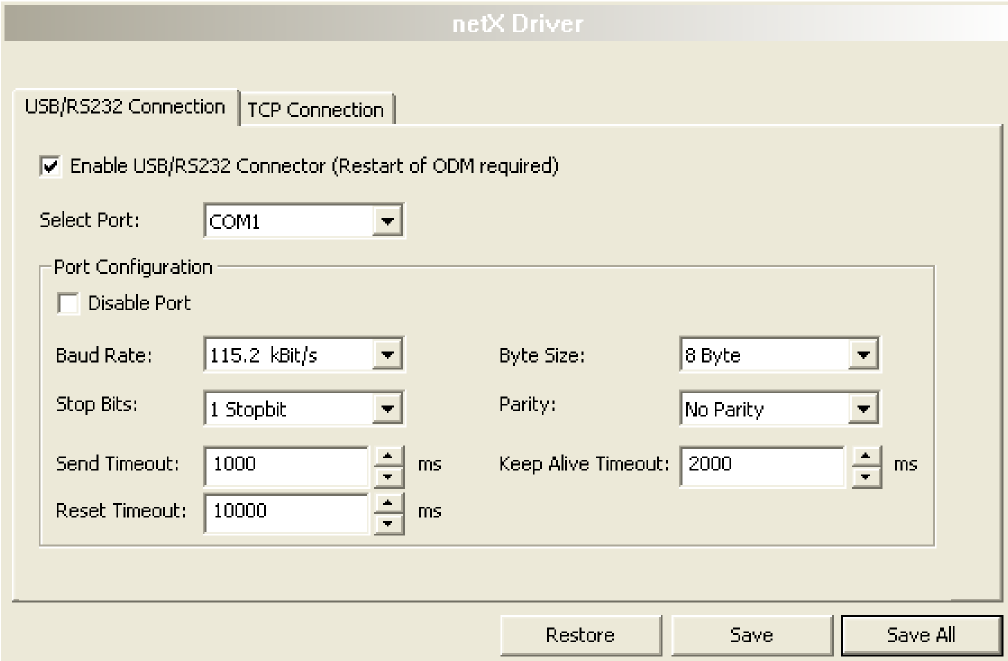


Figure 94: netX Driver > USB/RS232 Connection

Parameter	Meaning	Range of Value / 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 restarted1, to make the new setting valid. ____ [1] Restart the ODM server via the **ODMV3 Tray Application:** -In the foot line click to using the right mouse key. - In the context menu select **Service > Start.**	checked, unchecked; Default: unchecked
Select Port	Depending which COM ports (interfaces) are 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]
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 [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 (RS232 and USB): 1000 ms

Parameter	Meaning	Range of Value / Value
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 (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 is odd. Even parity: The parity is "even", if the number of bits occupied with 1 in the transmitted information word is even. Mark Parity: if the parity bit is always 1, this is named mark-parity (it contains no information). Space Parity: if the parity bit always 0, this is named space-parity (it is an empty space).	No Parity, Odd Parity, Even Parity, Mark Parity, Space Parity; Default (RS232): No Parity
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 is 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 9: Parameters netX Driver > USB/RS232 Connection

3.2.3.2 netX Driver - TCP/IP Connection

Function and use

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

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.

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

3.2.3.3 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**.

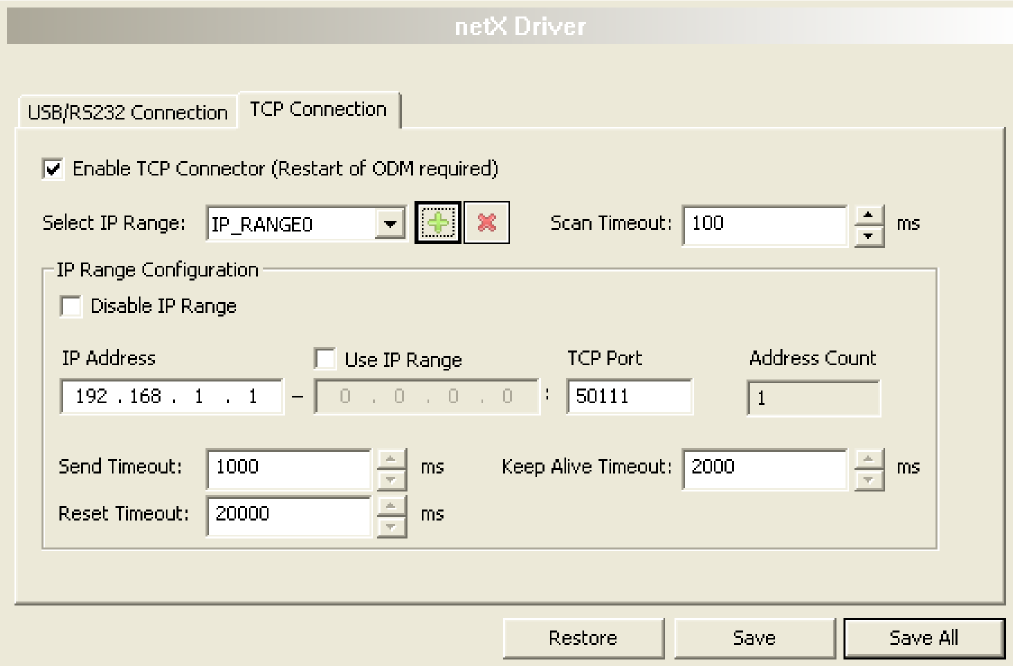


Figure 95: netX Driver > TCP Connection

Parameter	Meaning	Range of Value / 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 to  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 the scan timeout can be set, how long to wait for a response during a connection is established.	10 … 10000 [ms]; Default: 100 ms
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)

Parameter	Meaning	Range of Value / Value
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 is 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 10: 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.

3.2.4 Configure 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.

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.

3.3 Device Assignment

Note: In the **Device Assignment** you first must assign the device to the DTM by checking the check box. This is essential to establish an online connection from the DTM to the device later, as described in section *Connecting/Disconnecting Device* on page 260.

Therefore in the **Device Assignment** dialog pane you scan for the device(s) and select it.

3.3.1 Scanning for Devices

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

➯ The dialog pane **Device Assignment** is displayed.

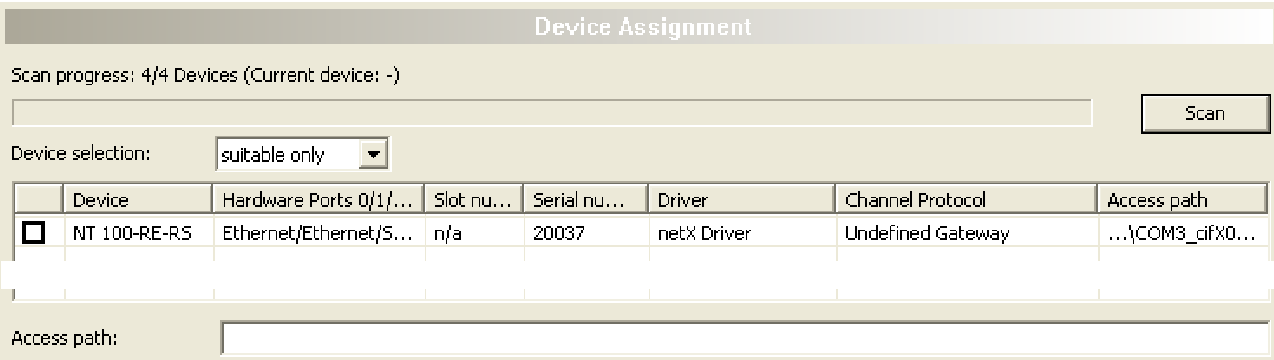


Figure 96: Device Assignment - detected Devices

2. Select the list box entry **suitable only**.
3. Select the **Scan** button, to start the scanning process.

➯ Under **Device Selection** all devices are displayed, which can be connected to the DTM via the preselected driver.

Parameter	Meaning	Range of Value / Value
Device selection	Selecting **suitable only** or **all** devices.	suitable only, all
Device	Device name.	
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 cifX card via the **Rotary Switch Slot Number (Card ID)**. The indication **n/a** means, that no **Slot-Nummer (Karten-ID)** exists. This occurs, if the cifX card is not equipped with a **Rotary Switch Slot Number (Card ID)** or for cifX cards equipped with a **Rotary Switch Slot Number (Card ID)**, if the rotary switch is set to the value 0 (zero) or all other device types.	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 (under Device selection 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 pane)	If under **Device selection** the check box for a device is checked, under **Access path** (at the lower side of the dialog pane) the driver identification or depending on the used driver additional data to the device. 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 11: Parameters of the Device Assignment

3.3.2 Selecting the Device

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

To select the physical EtherNet/IP Scanner device (with or without firmware):

1. In the **Device Assignment** pane in the selection list **Device selection** check the check box for the device.

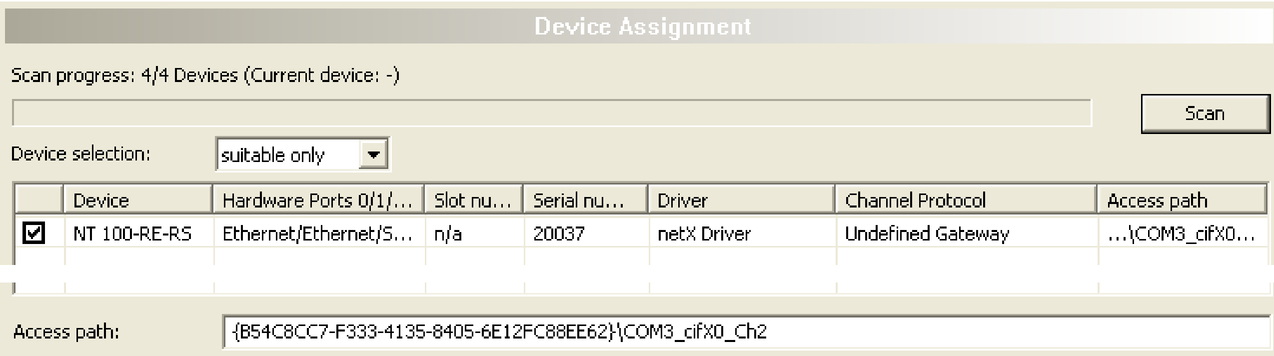


Figure 97: Device Assignment - Selecting the Device

➯ Under **Access path** (below in the dialog pane) 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 the **Apply** button, to apply the selection.

4 Configuration

4.1 Overview Configuration

<u>Configuration Dialog Panes</u>

The table below gives an overview for the Configuration dialog panes descriptions:

Section	Page
Settings	125
Licenses	127
Signal Mapping	140
Memory Card Management	147

Table 12: Descriptions of the Dialog Panes Configuration

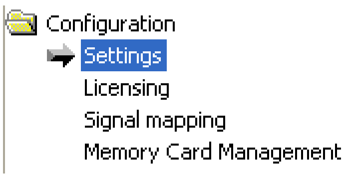


Figure 98: Navigation Area – Configuration

The following table shows the configuration panes for each device type.

Device Type	Configuration
NT 50 as gateway	Configuration Settings Signal Mapping
NT 100 as gateway	Configuration Settings Signal Mapping Memory Card Management Licensing
NT 100 as proxy	Configuration Settings Memory Card Management Licensing
NB 100 as gateway	Configuration Settings Signal Mapping Licensing
NT 151-RE-RE as gateway	Configuration Settings Signal Mapping Memory Card Management Licensing
NL 51N-DPL as proxy	Configuration Settings

Table 13: Navigation Area – Configuration – Device-dependent

4.2 Settings

In the **Settings** pane you can:

- select the protocol for Port X2 respectively for Port X3,
- transfer the firmware into the device,
- set the cycle time of the Gateway and
- enable the address switch for a slave protocol.

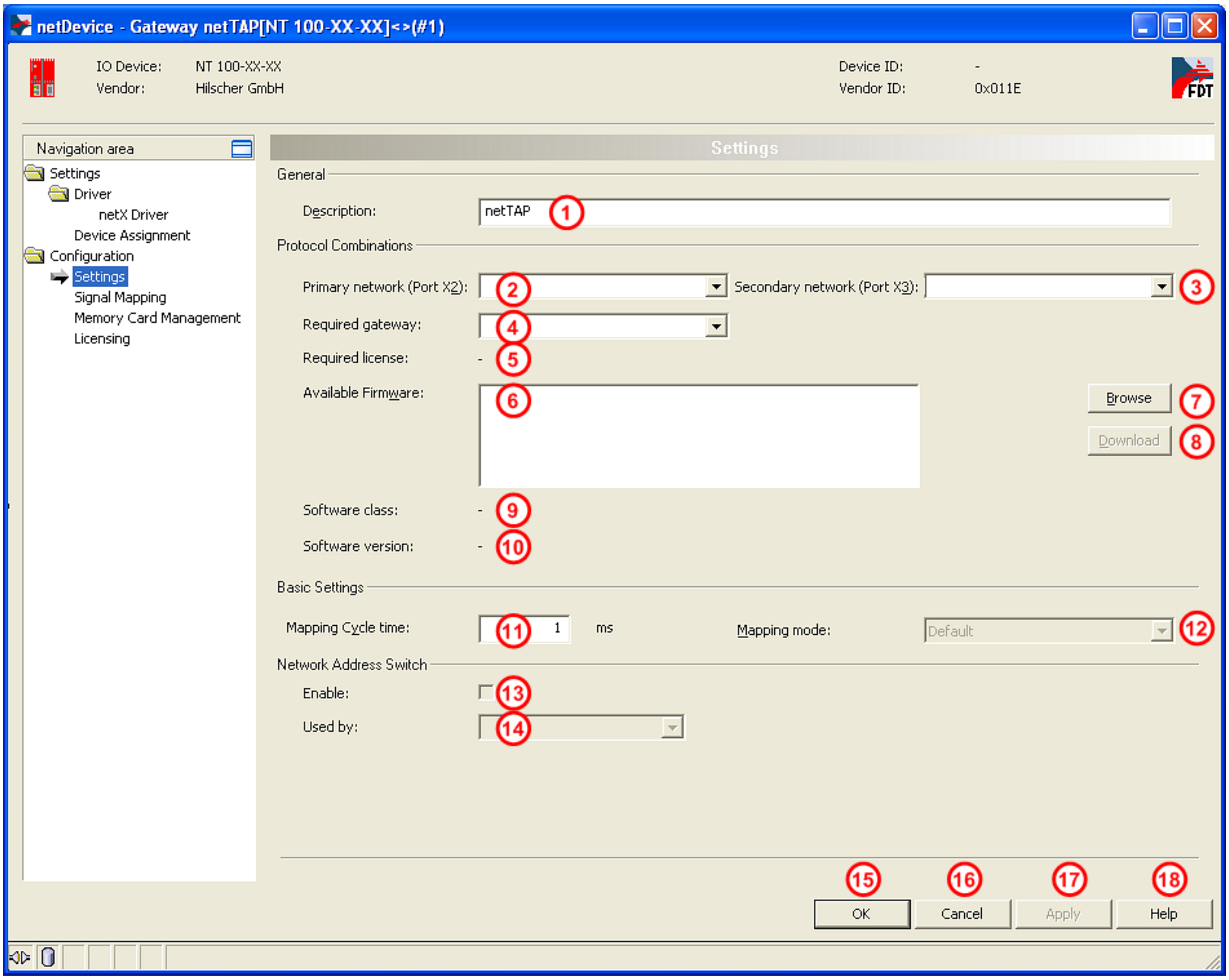


Figure 99: Settings

No.	Name	Description
General		
1	Description	Here you can enter a descriptive name also known as symbolic name for the device. This name ist displayed in front of the device. 
Protocol Combination		
2	Primary network (Port X2)	Here you can select the protocol for the primary network, which becomes active on port X2.
3	Secondary network (Port X3)	Here you can select the protocol for the secondary network, which becomes active on port X3.

No.	Name	Description
4	Required gateway	The required gateway device is shown here after you have select the protocols at 2 and 3. It is also possible to select the gateway device first. Then you can select at 2 and 3 protocols that are suitable for this gateway device.
5	Required license	If you have selected a protocol with master functionality at 2 and / or 3, then the number of master licenses required in the device is displayed here. How to order and transfer master licenses is described in section *Licenses* on page 127.
6	Available Firmware	Available firmware files for the selected protocols are listed in this area. The firmware files are stored in the SYCON.net installation directory of the PC in the sub folder "\netGatewayDTM\Firmware\".
7	Browse	**Browse** opens a dialog to select a firmware file. The selected firmware file is shown in area 6.
8	Download	**Download** transfers the selected firmware file into the device, which is selected in area 6. This is possible even when no protocols have been selected at 2 and 3.
9	Software class	The **Software class** of the selected firmware is displayed (for information only).
10	Software version	The firmware version of the selected firmware is displayed.
Basic Settings		
11	Mapping Cycle time	1 – 1000 ms (Default = 1 ms, is the cycle time for the device internal transfer of the input and output data from the buffer of port X2 to the buffer of X3 and visa versa.
12	Mapping mode	Always default (only displayed).
Netzwork Address Switch		
13	enable	☐ Not checked (default): The address is set by the configuration software and transferred by a download into the device.
		☑ Checked: The address is used from the slave protocol, which is set by the rotary switches at the device and can be used fort he following protocols: - CANopen Slave (to set the node address), - CC-Link Slave (to set the station address), - DeviceNet Slave (to set the MAC-ID) and - PROFIBUS DP Slave (to set the station address). Firmware version 1.1 (or higher) is required for the netTAP NT 50 device. Firmware version 1.5 (or higher) is required for netTAP NT 100 device. The address set and downloaded by the configuration software is always used for master protocols.
14	Used by	When two slave protocols are used, then you have to set if the address switches are valid for the primary network 2 or for the secundary network 3.
Common Buttons		
15	OK	The current settings are saved and then the dialog window is closed.
16	Cancel	The changed settings are discarded and then the dialog window is closed.
17	Apply	The current settings are saved and the dialog window stays open.
18	Help	Opens the online help in a separate window.

Table 14: Settings

4.3 Licenses

Using the license dialog, you can order licenses for **Master protocols** or **Utilities** and download them to your device.

4.3.1 Open License Dialog

Note: A connection to the device is necessary to open the license dialog.

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively netBRICK symbol **Configuration > Gateway** respectively **Configuration > Proxy**.

➮ The configuration dialog opens.

2. Open license dialog

➢ Select in the navigation area under the folder **Configuration** the entry **Licensing**.

➮ The license dialog opens.

4.3.2 License Dialog

In the **License**[1] pane you can:

- check, which licenses for Master protocols or Utilities are present in the device (Position 1 in the figure below),
- order licenses (Positions 2 to 11),
- transfer licenses to the device 12.

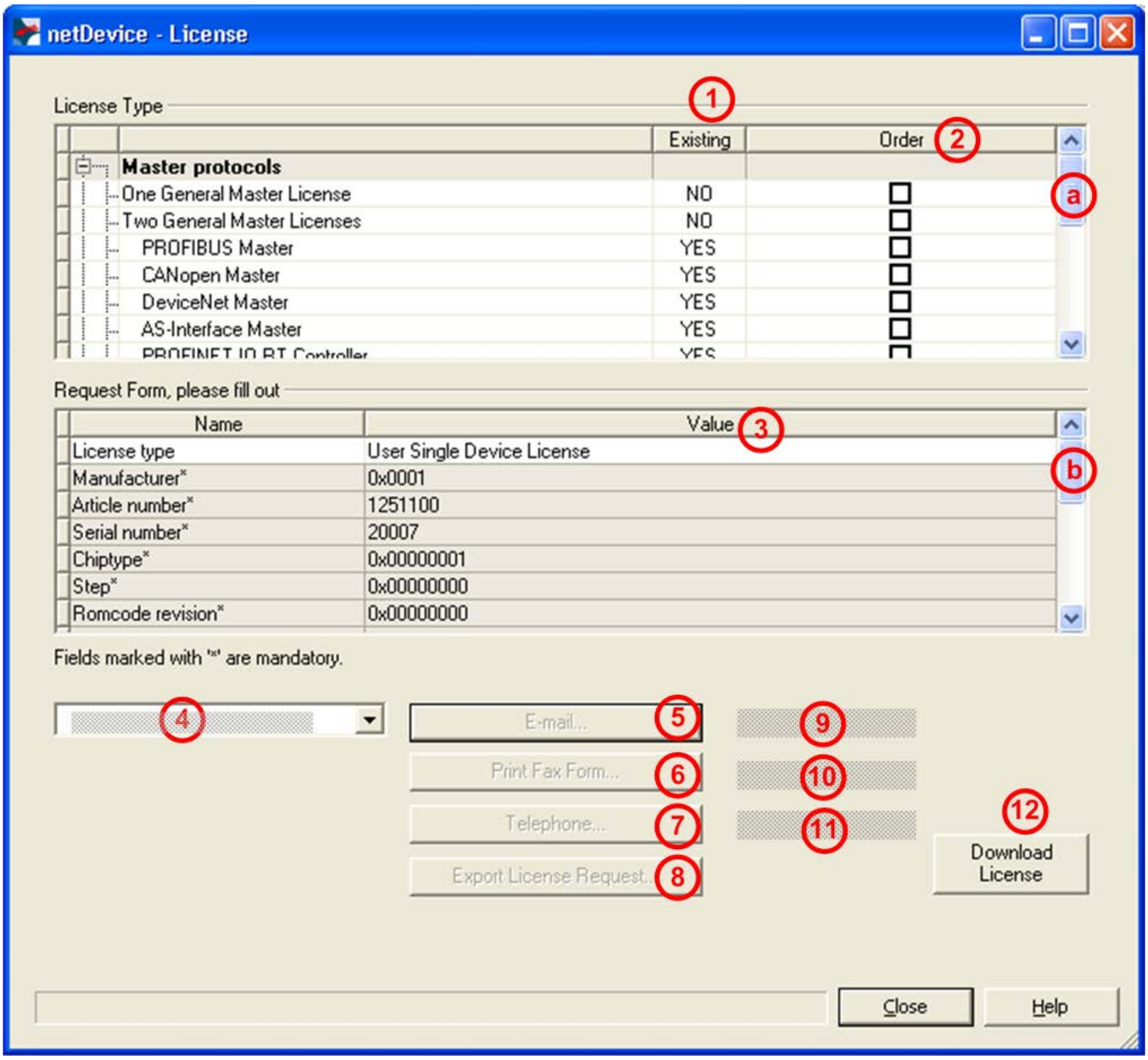


Figure 100: License Pane

Note: To display further entries under **License Type**, move the scroll box ⓐ downwards or upwards. To display further entries under **Request Form, please fill out**, move the scroll box ⓑ downwards or upwards.

1 The title bar contains the notation of the **device description**:
Symbolic Name [Device Description] <Station Address > (#Network ID).

4.3.3 Which Licenses are present in the Device?

Check, which licenses are present in the device.

How to proceed:

- ➢ Open the **License** pane as described under section *Open License Dialog* on page 127.

Figure 101: License Pane - License Type

- ➢ Under **License Type** click ⊞ at **Master protocols**.
- ➾ The **Master protocols** overview opens:

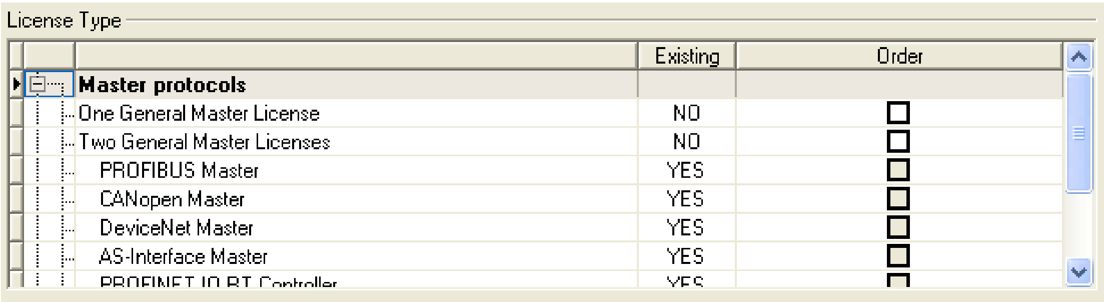


Figure 102: License Pane – License Type / Master protocols

- ➢ Or click ⊞ at **Utilities**.
- ➾ The **Utilities** overview opens:

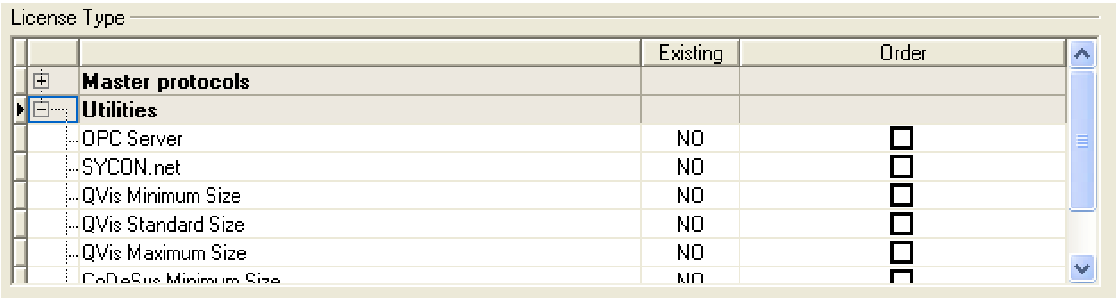


Figure 103: License Pane – License Type / Utilities

- ➾ The column **Existing** indicates which licenses are present in the device.

  Yes = License is present in the device.

  No = License is not present in the device.

Note: In newer versions of the present configuration software under **License Type** may be displayed additional licenses or other protocols that can be ordered later.

4.3.3.1 License for Master Protocols

One General Master License:

On the device maximally 1 communication protocol with master function can be implemented.

Two General Master Licenses:

On the device maximally 2 communication protocols with master function can be implemented.

The license includes the following Master protocols:

- AS-Interface Master
- CANopen Master
- DeviceNet Master
- EtherCat Master
- EtherNet/IP Scanner
- PROFIBUS Master
- PROFINET IO RT Controller
- SERCOS III Master

4.3.3.2 License for Utilities

- SYCON.net
- OPC Server
- QVis Minimum Size
- QVis Standard Size
- QVis Maximum Size
- CoDeSys Minimum Size
- CoDeSys Standard Size
- CoDeSys Maximum Size

For the utilities QVis and CoDeSys, only one license each may be chosen alternatively as:

- *Minimum Size*,
- *Standard Size* or
- *Maximum Size*.

4.3.4 How to order a License

To order a license, proceed as follows:

	Refer to Section:	Page
1. Open the license dialog.	*Open License Dialog*	127
2. Select the required licenses.	*Selecting License*	131
3. Enter the ordering data.	*Ordering Data*	132
4. Place your order.	*Ordering the License*	134

4.3.5 Selecting License(s)

You can select licenses for Master protocols and / or utilities.

1. Selecting license(s) for Master protocol(s):

➢ Under **License Type** click ⊞ at **Master protocols** in the **License** pane.

➢ Under **Order** check as many licenses must run simultaneously on your device:
One General Master License or
Two General Master Licenses.

2. And/or select license(s) for utility(utilities):

➢ In the **License** pane under **License Type** click ⊞ at **Utilities**.

➢ Under **Order** check the required utility(utilities) (*single or several*)[2]:

- SYCON.net
- OPC Server
- QVis Minimum Size*
- QVis Standard Size*
- QVis Maximum Size*
- CoDeSys Minimum Size**
- CoDeSys Standard Size**
- CoDeSys Maximum Size**

2 For *) and **) minimum size, standard size or maximum size can be selected only as an alternative.

4.3.6 Ordering Data

1. Device Information

➩ The *Device Information* required for the order are read from the device and automatically filled in the order.

2. Ordering Data

Enter the *Ordering Data* into the **License** pane.

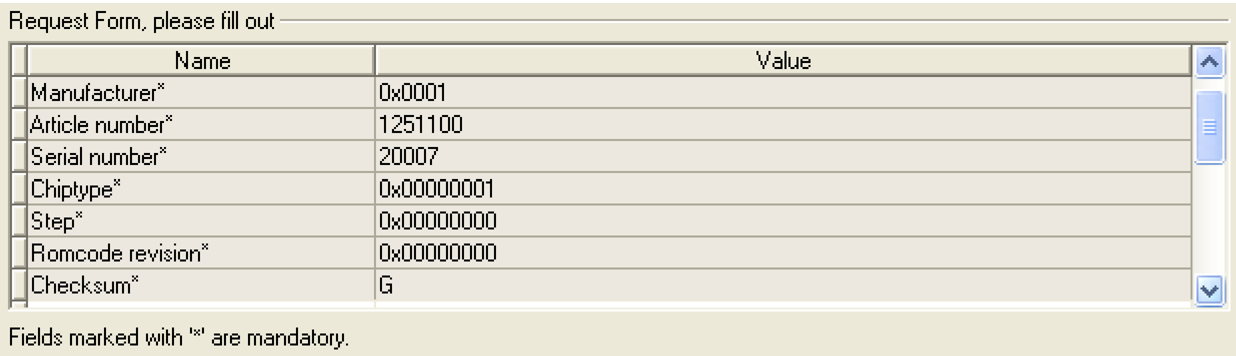
➢ Enter the **Data to manage the Order** (therefore refer to section *Data to manage the Order (License Information)*on page 133).

4.3.6.1 Device Information (Ordering data read from the Device)

The following ordering data are read from the device and displayed in the **License** pane:

- Manufacturer
- Device number
- Serial number
- Chiptype
- Step (chip revision)
- Romcode revision
- Checksum (checksum of the device data)

➩ The gray fields under **Request Form, please fill out** contain the ordering data read from the device:


Request Form, please fill out

Name	Value
Manufacturer*	0x0001
Article number*	1251100
Serial number*	20007
Chiptype*	0x00000001
Step*	0x00000000
Romcode revision*	0x00000000
Checksum*	G

Fields marked with '*' are mandatory.

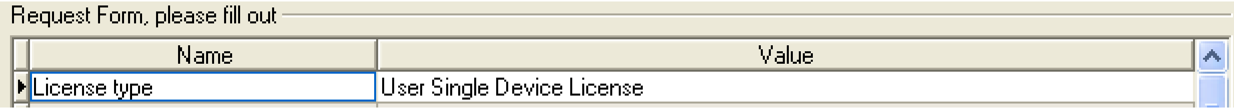
Figure 104: License Pane - Request Form, please fill out / Device Information

➩ These ordering data read out from the device are displayed automatically from the device.

4.3.6.2 Data to manage the Order (License Information)

For your order you must enter the following data to the **License** pane:

1. License Type (User Single Device License).


Request Form, please fill out

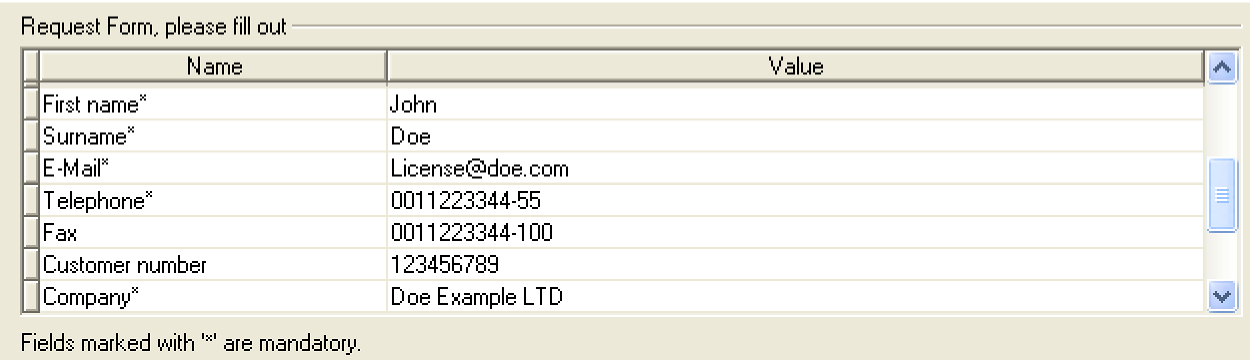
Name	Value
License type	User Single Device License

Figure 105: License Pane - Request Form, please fill out / License Type

➢ Select the license type under **Request Form, please fill out** > **Value**, (for future application, currently only *User Single Device License* can be selected).

2. Mandatory data to the order request (editable fields):

- First Name
- Surname
- E Mail (address, to which the license download link shall be send.)
- Telephone
- Company
- Address
- Country
- City, State, Zip


Request Form, please fill out

Name	Value
First name*	John
Surname*	Doe
E-Mail*	License@doe.com
Telephone*	0011223344-55
Fax	0011223344-100
Customer number	123456789
Company*	Doe Example LTD

Fields marked with '*' are mandatory.

Figure 106: License Pane - Request Form, please fill out / Mandatory data

➢ Enter all mandatory fields under **Request Form, please fill out** > **Value** (marked with*).

3. Additional order data, not mandatory (editable fields):

- Fax
- Customer Number
- Order Number
- Value added tax identification number

➢ Under **Request Form, please fill out** > **Value** enter all fields for the additional data, which are not mandatory.

4.3.7 Ordering the License

Place your order in the **License** pane. Therefore:

Figure 107: License Pane – Selecting the Subsidiary / Ordering / Contacts

1. Select the *Subsidiary* 4, to which the order shall be send.
2. Place the order:

	Refer to Section:	*Page*
• by **E-Mail** 5,	*Ordering the License by E Mail*	135
• or by **Fax** 6 or by **Telephone** 7,	*Ordering the License by Fax or by Telephone*	136
• or in a **File** 8.	*Exporting License Request to a File*	138

➩ The **Contact Data** of the selected subsidiary are displayed under Position 9, 10 and 11.

4.3.7.1 Ordering the License by E Mail

You can place your order by e-mail.

Figure 108: License Pane – placing the order by E-mail

- ➢ Click **E-mail…** ⑤.
- ✎ The order E-mail **License request** opens:

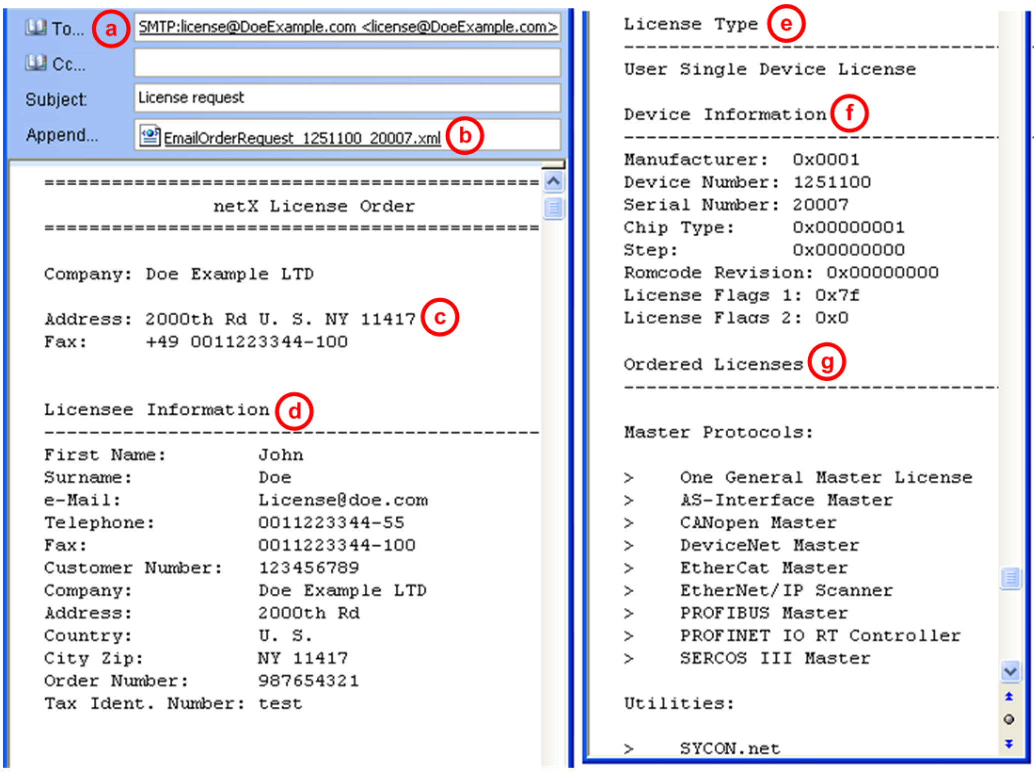


Figure 109: Example: Order E-Mail License request

✎ The order e-mail **License request** contains:

- the **E-mail…** of the selected subsidiary ⓐ,
- the automatically generated **XML file** ⓑ *EmailOrderRequest_-[Devicenumber]_[Serialnumber].xml* with a summary info of the **order information**,
- the **Order Address** ⓒ,
- the **License Information** ⓓ,
- the **License Type** ⓔ,
- the **Device Data** ⓕ,
- the **ordered Licenses** ⓖ.

- ➢ Send the order e-mail **License request**.
- ➢ The order process is complete.

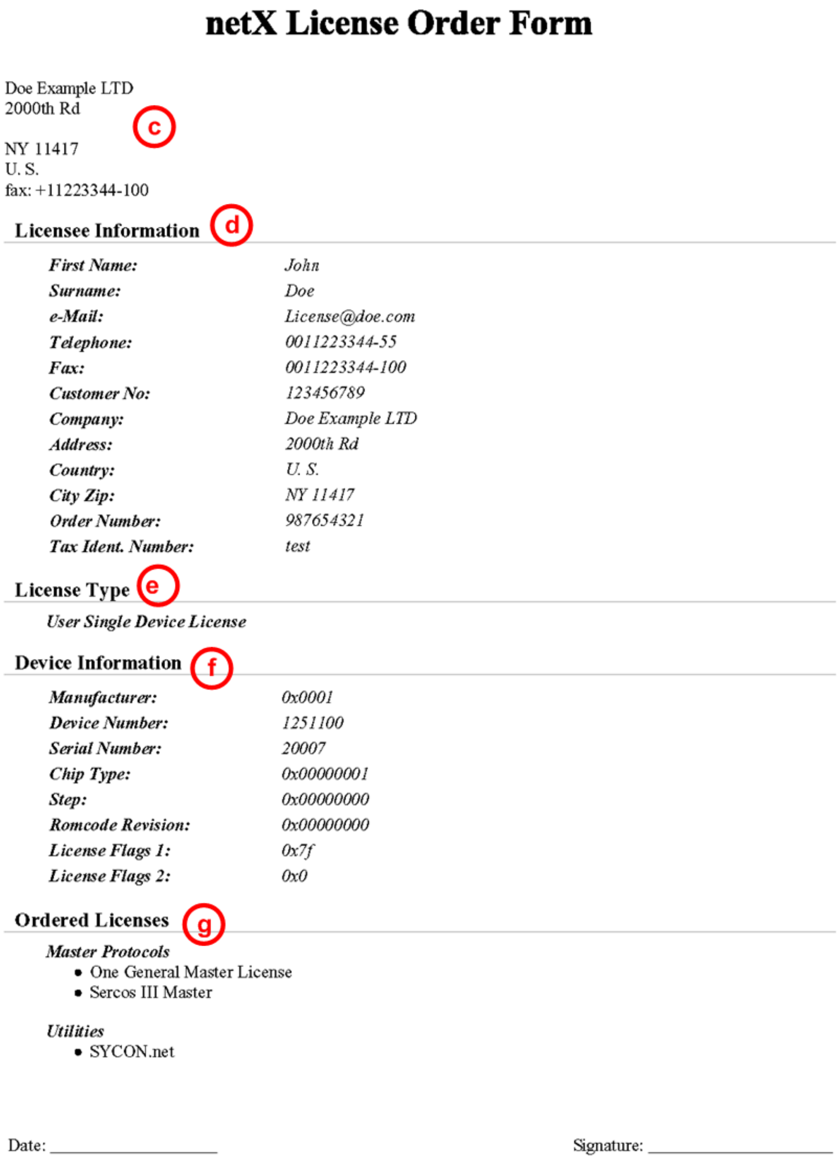
4.3.7.2 Ordering the License by Fax or by Telephone

You can place your order by Fax or by Telephone.

Figure 110: License Pane - placing the order by Fax or by Telephone

- Click **Print Fax Form** ⑥ or **Telephone…** ⑦.
- The summary of the ordering data *PrintOrderRequest_[Devicenumber]_[Serialnumber].html* is opened in a browser window.

Note: If your browser does not display the order data or the window **Move Element** or **Copy Element** are displayed, check the safety settings of your system.



netX License Order Form

Doe Example LTD
2000th Rd

NY 11417
U. S.
fax: +11223344-100

Licensee Information

First Name:	*John*
Surname:	*Doe*
e-Mail:	*License@doe.com*
Telephone:	*0011223344-55*
Fax:	*0011223344-100*
Customer No:	*123456789*
Company:	*Doe Example LTD*
Address:	*2000th Rd*
Country:	*U. S.*
City Zip:	*NY 11417*
Order Number:	*987654321*
Tax Ident. Number:	*test*

License Type

User Single Device License

Device Information

Manufacturer:	*0x0001*
Device Number:	*1251100*
Serial Number:	*20007*
Chip Type:	*0x00000001*
Step:	*0x00000000*
Romcode Revision:	*0x00000000*
License Flags 1:	*0x7f*
License Flags 2:	*0x0*

Ordered Licenses

Master Protocols
- One General Master License
- Sercos III Master

Utilities
- SYCON.net

Date: ________________ Signature: ________________

Figure 111: Example: Order Data Form PrintOrderRequest

- The order data form contains:
- the **Order Address** c,
- the **License Information** d,
- the **License Type** e,
- the **Device Data** f,
- the **ordered Licenses** g.

- Print the order data form, sign it and send it by Fax.

Figure 112: License Pane – Fax Number of the selected Subsidiary

- Use the Fax number 10, which is displayed after the subsidiary was selected in the **License** pane.

Or:

- Keep ready the order data form and communicate the order data via telephone.

Figure 113: License Pane – Telephone Number of the selected Subsidiary

- Use the telephone number 11, which is displayed after the subsidiary was selected in the **License** pane.
- The order process is complete.

4.3.7.3 Exporting License Request to a File

If you are working on a process computer without an e-mail client, you can export your order information to a file, save the file to a removable disk and place your order manually via e-mail from a different PC.

Figure 114: License Pane - Ordering by exported File and E-Mail

- ➢ Click **Export License Request…** (8).
- ➩ The window **Browse For Folder** is displayed.
- ➢ Choose for or create a new folder on a removable disk.
- ➢ Save the automatically generated **XML file** *EmailOrderRequest_-[Devicenumber]_[Serialnumber].xml* with a summary info of the **order information** to this folder.
- ➢ Send this file from a PC with an e-mail client manually via e-mail.
- ➢ Therefore use an e-mail address , which is displayed after the subsidiary was selected in the **License** pane (see Position (9) Figure *License Pane* on page 128).
- ➢ The order process is complete.

4.3.8 How to get the License and transfer it to the Device

Note: License files can only be delivered via e-mail. The e-mail contains a link to download the license file.

According to the license you ordered, you will receive an e-mail containing a **Link to download the License File**. This leads to a server PC on which the license file is provided. Using the received link you will have to save the license file on your PC and then transfer the license to your device. If your e-mail client is on another PC as your device, you must save your license file e. g. to an USB stick.

Steps how to proceed

1. Save the license file to a PC or a disk.

- ➢ Click to the **Link to download the License File** in the e-mail.
- ➢ Save the license file **.nxl* to a PC or a removable disk.

2. Download the license file to the device.

- ➢ Respectively connect the removable disk with the license file to the PC, which is connected to your device.
- ➢ Click **Download License** 12 in the **License** pane in the configuration software.

Figure 115: License Pane - Download License

- ➾ The File selection window **Open** is displayed.
- ➢ Therein select the license file *netX License Files (*.nxl)*.
- ➢ Click **Open**.
- ➾ The license file is transferred to the device.
- ➾ After this the license is present in the device and is activated with the next device reset.

3. Activate Device Reset

Hint: To activate the license in the first device, a device reset is required.

- ➢ To check whether the license has been activated, follow the steps in section *Which Licenses are present in the Device?* on page 129.

4.4 Signal Mapping

The signal mapping determines which user data of the primary network are assigned to which user data of the secondary network and vice versa.

The signal mapping is relevant for the gateway devices netTAP NT 50, NT 100, NT 151-RE-RE and netBRICK NB 100. The mapping of the data for gateway devices can be done by the user und be configured within SYCON.net. The mapping of data is already specified for proxy devices and done automatically by SYCON.net.

With the mapping for gateway devices

- the received data from the primary network are mapped to the send data of the secondary network,
- the received data from the secondary network are mapped to the send data of the primary network,
- status information of the primary network is mapped to the send data of the secondary network,
- status information of the secondary network is mapped to the send data of the primary network.

With this mapping a device internal list in generated. The device work each x ms through this list and copies the data.

On the pane I/O Data Mapping you can map the I/O Data of port X2 to the I/O data of port X3.

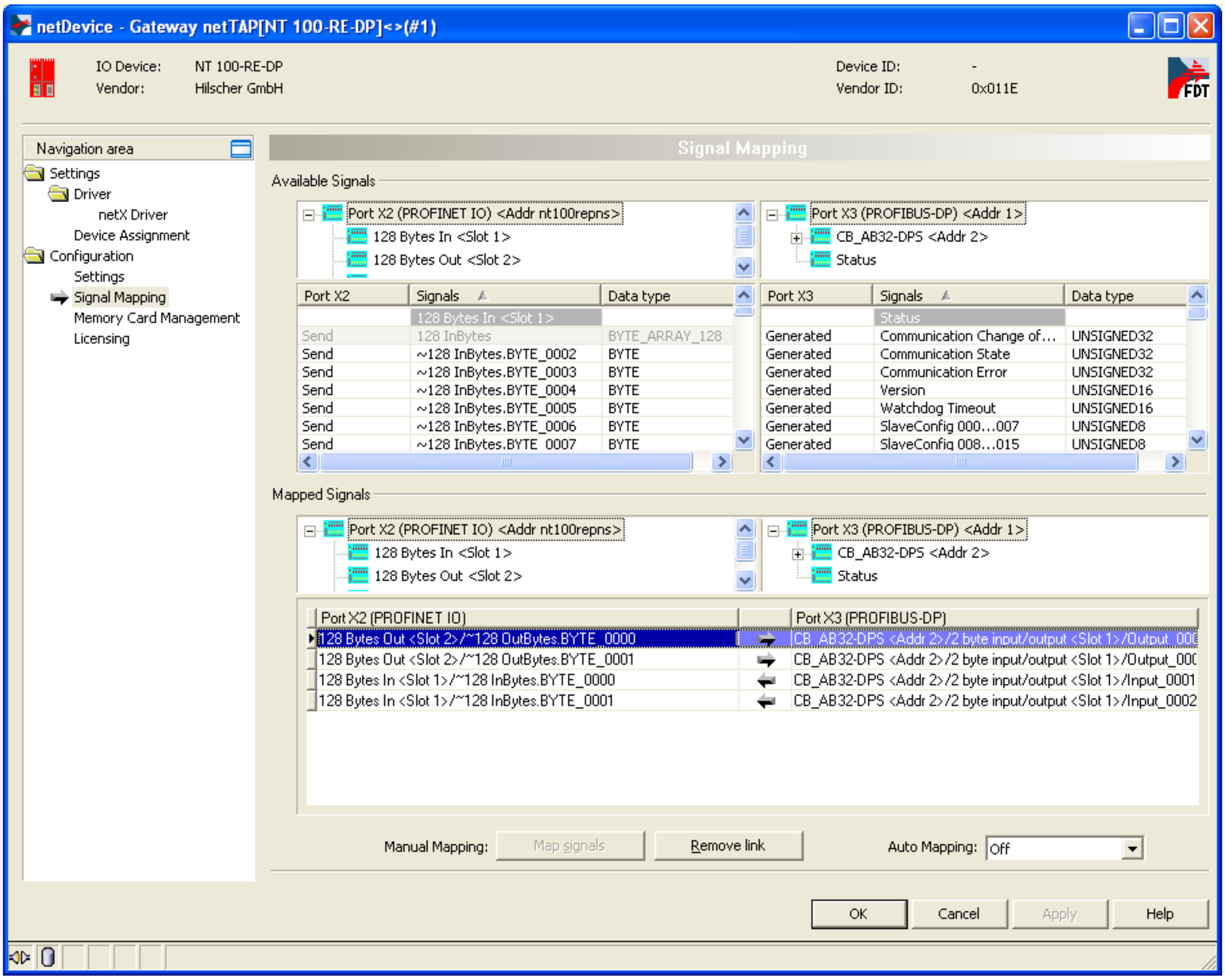


Figure 116: Signal Mapping

The I/O mapping is done the following way:

Data transfer from Port X2 to Port X3

- ➢ Map the signals, which are received on Port X2 (Port X2 receive), with signals, which should be send on Port X3 (Port X3 send).
- ➢ For this, mark the signal received (Port X2) and the signal to be send (Port X3) and the click the button **Map Signals**
  Or
  Map the signal using drag and drop. For this, drag the signal received (Port X2 receive) and drop it on the signal to be send (Port X3 send)

Data transfer from Port X3 to Port X2

- Map the signals, which are received on Port X3 (Port X3 receive), with signals, which should be send on Port X2 (Port X2 send).
- For this, mark the signal received (Port X3) and the signal to be send (Port X2) and the click the button **Map Signals**
  Or
  Map the signal using drag and drop. For this, drag the signal received (Port X3 receive) and drop it on the signal to be send (Port X2 send)

Transfer status info of Port X2 to Port X3

- If necessary, map the status information of Port X2 (Port X2 generated, which are generated device internal) to signals which should be send on Port X3 (Port X3 send)
- For this mark the status signal (Port X2) and the signal which should be send (Port X3) and click the button **Map Signals**
  Or
  Map the signal using drag and drop. For this, drag the status signal (Port X2 generated) and drop it on the signal to be send (Port X3 send)

Transfer status info of Port X3 to Port X2

- If necessary, map the status information of Port X3 (Port X3 generated, which are generated device internal) to signals which should be send on Port X2 (Port X2 send)
- For this mark the status signal (Port X3) and the signal which should be send (Port X2) and click the button **Map Signals**
  Or
  Map the signal using drag and drop. For this, drag the status signal (Port X3 generated) and drop it on the signal to be send (Port X2 send)

Auto mapping

The signal mapping can also be done automatically be the configuration software. Set **Auto Mapping** to **From Port X3 to Port X2** and then click on **Apply**. The automatic signal mapping is done.

4.4.1 Status Information

The status of the remote network coupled by the netTAP NT 100 can be mapped into the I/O data. It contains the items shown in the figures for master and slave and are described below.

Signals	Data type	Port X3
Status		
Active Slaves	UNSIGNED32	Generated
Communication Change of State	UNSIGNED32	Generated
Communication Error	UNSIGNED32	Generated
Communication State	UNSIGNED32	Generated
Configured Slaves	UNSIGNED32	Generated
Error Count	UNSIGNED32	Generated
Error Log Indicator	UNSIGNED32	Generated
Faulted Slaves	UNSIGNED32	Generated
Host Watchdog	UNSIGNED32	Generated
Slave Error Log Indicator	UNSIGNED32	Generated
Slave State	UNSIGNED32	Generated
Version	UNSIGNED16	Generated
Watchdog Timeout	UNSIGNED16	Generated

Figure 117: Status information of the Master

Signals	Data type	Port X2
Status		
Communication Change of State	UNSIGNED32	Generated
Communication Error	UNSIGNED32	Generated
Communication State	UNSIGNED32	Generated
Error Count	UNSIGNED32	Generated
Error Log Indicator	UNSIGNED32	Generated
Host Watchdog	UNSIGNED32	Generated
Version	UNSIGNED16	Generated
Watchdog Timeout	UNSIGNED16	Generated

Figure 118: Status information of the Slave

Some status information belong to the master firmware and to the slave firmware status and are marked with “Master and Slave”), others belong only to the master firmware status and are marked with “Master”.

Number of Active Slaves (Master)

The firmware maintains a list of slaves within the remote network to which the remote network’s master has successfully opened a connection. Ideally, the number of active slaves is equal to the number of configured slaves.

This field holds the number of active slaves.

Communication Change of State (Master and Slave)

The communication change of state register contains information about the current operating status of the communication channel and its firmware.

31	30	…	12	11	10	9	8	7	6	5	4	3	2	1	0

COMM_COS_READY
COMM_COS_RUN
COMM_COS_BUS_ON
COMM_COS_CONFIG_LOCKED
COMM_COS_CONFIG_NEW
COMM_COS_RESTART_REQUIRED
COMM_COS_RESTART_REQUIRED_ENABLE
unused, set to zero

Table 15: Communication Change of State

The Communication Change of State Flags have the following meaning:

Status	Meaning
Ready (Bit 0)	The Ready flag is set as soon as the protocol stack is started properly. Then the protocol stack is awaiting a configuration. As soon as the protocol stack is configured properly, the Running flag is set, too.
Running (Bit 1)	The Running flag is set when the protocol stack has been configured properly. Then the protocol stack is awaiting a network connection. Now both the Ready flag and the Running flag are set.
Bus On (Bit 2)	The Bus On flag is set to indicate to the host system whether or not the protocol stack has the permission to open network connections. If set, the protocol stack has the permission to communicate on the network; if cleared, the permission was denied and the protocol stack will not open network connections.
Configuration Locked (Bit 3)	The Configuration Locked flag is set, if the communication channel firmware has locked the configuration database against being overwritten. Re-initializing the channel is not allowed in this state.
Configuration New (Bit 4)	The Configuration New flag is set by the protocol stack to indicate that a new configuration became available, which has not been activated. This flag may be set together with the Restart Required flag.
Restart Required (Bit 5)	The Restart Required flag is set when the channel firmware requests to be restarted. This flag is used together with the Restart Required Enable flag below. Restarting the channel firmware may become necessary, if a new configuration was downloaded from the host application or if a configuration upload via the network took place.
Restart Required Enable (Bit 6)	The Restart Required Enable flag is used together with the Restart Required flag above. If set, this flag enables the execution of the Restart Required command in the netX firmware.

Communication State (Master and Slave)

The communication state field contains information regarding the current network status of the communication channel of the remote network. Depending on the implementation of the protocol stack of the remote network, all or a subset of the definitions below is supported.

Status	Value
Unknown	0
Offline	1
Stop	2
Idle	3
Operate	4

Communication Channel Error (Master and Slave)

This field holds the current error code of the communication channel of the remote network. If the cause of error is resolved, the communication error field is set to zero again.

All values different from 0 indicate that an error has occurred.

Errors may be signaled either from the operating system rcX or from the used protocol.

Number of Configured Slaves (Master)

The firmware maintains a list of slaves within the remote network to which the master has to open a connection. This list is derived from the configuration database created by SYCON.net. This field holds the number of configured slaves.

Error Count (Master and Slave)

This field holds the total number of errors detected since power-up within the remote network, respectively after reset. The protocol stack counts all sorts of errors in this field no matter if they were network related or caused internally. After power cycling, reset or channel initialization this counter is being cleared again.

Error Log Indicator (Master and Slave)

Note: This field is not yet supported.

Number of Faulted Slaves (Master)

If a slave of the remote network encounters a problem, it can provide an indication of the new situation to the master in certain Fieldbus systems. As long as those indications are pending and not serviced, this field holds a value unequal to zero. If no more diagnostic information is pending, the field is set to zero.

Host Watchdog (Master and Slave)

Note: This field is not yet supported.

Slave Error Log Indicator (Master)

Note: This field is not yet supported.

Slave State (Master)

The slave state field indicates whether the master of the remote network is in cyclic data exchange to all configured slaves. In case there is at least one slave missing or if the slave has a diagnostic request pending, the status is set to FAILED. For protocols that support non-cyclic communication only, the slave state is set to OK as soon as a valid configuration is found.

Status	Value
Undefined	0
Ok	1
Failed	2

Version (Master and Slave)

The version field holds the version number of this structure. The value is 1.

Watchdog Timeout (Master and Slave)

This field holds the configured watchdog timeout value of the protocol stack of the remote network. The value is specified in milliseconds.

4.5 Memory Card Management

Note: This function is only available for **netTAP NT 100** and **NT 151-RE-RE** devices, because only they have an MMC card slot.

In the **Memory Card management** pane you can:

- backup the firmware and the configuration from the netTAP device to an MMC memory card
- restore the firmware and the configuration from the MMC memory card into the netTAP device

On the pane MMC Management you can backup and restore the firmware and the configuration to respectively from an MMC card.

The **Folder** on the pane shows the directory structure of the file system of the netTAP device named **SYSVOLUME** and if an MMC card is inserted also the directory structure of the MMC card named **SDMMC**.

Backup: To backup the firmware and configuration files insert a formatted (Format FAT) MMC card into the MMC slot of the netTAP device. When the directory named **SDMMC** is shown, then click on the button **Backup** to copy the firmware and configuration files from the netTAP device to the MMC card.

Restore: To load the firmware and configuration files from an MMC card into the netTAP device, insert an MMC card into the MMC slot of the netTAP device which contains the firmware and configuration files. When the directory named SDMMC is shown, then click the button Restore to copy the firmware and configuration files from the MMC card into the netTAP device.

4.6 Configuration EtherCAT Master

The netTAP NT 100 device respectively the NT 151-RE-RE respectively the netBRICK NB 100 device as EtherCAT Master needs a configuration, e. g. information about how many EtherCAT Slave devices with how many input and output data are to be connected.

1. Insert EtherCAT Slave devices

➢ Drag from the device catalog EtherCAT Slave device(s) and drop it to the bus line of the EtherCAT Master.

2. Configure EtherCAT Slave devices

➢ Open the configuration dialog for each EtherCAT Slave device and configure the device.

Information about how to configure the EtherCAT Slave device is in the operating instruction manual „Generic DTM for EtherCAT Slave devices“ and has the file name EtherCAT_GenericSlave_DTM_en.pdf respectively open in the configuration dialog of the slave the online help by the F1 key.

3. Configure EtherCAT Master

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > EtherCAT Master**

Information about how to configure the EtherCAT Master device is in the operating instruction manual „DTM for EtherCAT Master devices“ and has the file name EtherCAT_Master_DTM_en.pdf respectively open in the configuration dialog of the master the online help by the F1 key.

4.7 Configuration EtherCAT Slave

Devices acting as EtherCAT Slaves need parameters.

This section describes how to set EtherCAT Slave parameters for the netTAP NT 100 device, NT 151-RE-RE device respectively the netBRICK NB 100 device.

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > EtherCAT Slave**

➯ The EtherCAT Slave configuration dialog opens.

➯ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the configuration dialog is displayed.

2. Set number of Input Data Bytes and Output Data Bytes

➢ Enter in the field Input Data Bytes the number of bytes the EtherCAT Master should send to the netTAP or netBRICK device, e. g. 32

➢ Enter in the field Output Data Bytes the number of bytes the EtherCAT Master should receive from the netTAP or netBRICK device, e. g. 64

3. Set more parameter.

➢ If necessary, set more parameter. The parameter are described in the following section *EtherCAT Slave Parameter* on page 150.

Note: Default values can be used normally for the other parameters. Depending on additionally requirements and the design of the EtherCAT network as well as the used devices parameters needs to be adjusted if necessary.

4. Open the signal configuration dialog.

➢ Select in the navigation area under the folder **Configuration** the entry **Signal Configuration**.

➯ The signal configuration dialog opens.

5. Signal configuration

➢ If you want to assign own signal names, then enter your signal names.. Further information is described in section *Signal Configuration* on page 255.

6. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➯ The configuration dialog closes

4.7.1 EtherCAT Slave Parameter

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 16: EtherCAT Slave Parameters

4.7.2 Settings at the used EtherCAT Master

Device Description File

To configure the EtherCAT Master an XML file (device description file) is required. The XML file is stored on the Gateway Solutions DVD in the directory `Electronic Data Sheets (e.g. EDS,GSD,GSDML)\EtherCAT`:

Device	XML File
NT 100-RE-XX	Hilscher NT 100-ECS-XX V2.2.xml
NB 100-RE-XX	Hilscher NB 100-ECS-XX V2.2.xml
NT 151-RE-RE	Hilscher NT 151-ECS-XX V4.2.X.xml

Table 17: XML File Names for netTAP and netBRICK

Configuration

Note: The settings in the used Master must comply with the settings in the Slave, to establish a communication. Important parameters are: Vendor ID, Product Code, Serial Number, Revision Number, Output and Input length.

In order that the EtherCAT Master can communicate with the netTAP or netBRICK device as EtherCAT Slave:

- Enter at the EtherCAT Master the number of data bytes (number of input data), that are configured in the netTAP or netBRICK device, e. g. 64.
- Enter at the EtherCAT Master the number of data bytes (number of output data), that are configured in the netTAP or netBRICK device, e. g. 32.
- You can set at the EtherCAT Master, if the Master verifies identification numbers of the EtherCAT Slave. This verification can be activated or deactivated at the EtherCAT Master. If the verification is used, then use respectively check the following values:

<table>
<tr><th>Device</th><th>Product code
(ProductCode)</th><th>Revision number
(RevisionNo)</th><th>Vendor
(Vendor ID)</th></tr>
<tr><td>NT 100-RE-XX</td><td>13 (0x000D)</td><td rowspan="2">0 (0x0000)</td><td rowspan="3">68 (0x0044)</td></tr>
<tr><td>NB 100-RE-XX</td><td>25 (0x0019)</td></tr>
<tr><td>NT 151-RE-RE</td><td>53 (0x0035)</td><td>131076 (0x020004)</td></tr>
</table>

Table 18: Identification parameters in EtherCAT XML Files

4.8 Configuration EtherNet/IP Scanner

The netTAP NT 50, NT 100 device respectively the NT 151-RE-RE device respectively the netBRICK NB 100 device as Ethernet/IP Scanner needs a configuration, e. g. information about how many Ethernet/IP Adapter devices with how many input and output data are to be connected.

1. Insert Ethernet/IP Adapter devices.

➢ Drag from the device catalog Ethernet/IP Adapter device(s) and drop it to the bus line of the Ethernet/IP Scanner.

2. Configure Ethernet/IP Adapter devices

➢ Open the configuration dialog for each Ethernet/IP Adapter device and configure the device.

Information about how to configure the Ethernet/IP Adapter device is in the operating instruction manual „Generic DTM for EtherNet/IP Adapter devices“ and has the file name EtherNetIP_GenericAdapter_DTM_en.pdf respectively open in the configuration dialog of the slave the online help by the F1 key.

3. Configure Ethernet/IP Scanner

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > Ethernet/IP Scanner**

Information about how to configure the Ethernet/IP Scanner device is in the operating instruction manual „DTM for EtherNet/IP Scanner devices“ and has the file name EtherNetIP_Scanner_DTM_en.pdf respectively open in the configuration dialog of the master the online help by the F1 key.

4.9 Configuration EtherNet/IP Adapter

Devices acting as EtherNet/IP Adapters need parameters.

This section describes how to set EtherNet/IP Adapter parameters for the netTAP NT 100 device, NT 151-RE-RE device respectively the netBRICK NB 100 device.

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > EtherNet/IP Adapter**

➯ The EtherNet/IP Adapter configuration dialog opens.

➯ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the settings dialog is displayed.

2. EtherNet/IP Adapter Parameter

   To set a fix IP address:

➢ Deselect the flag DHCP and BootP.

➢ Select IP address Enable and enter a valid IP address for the netTAP or netBRICK device.

➢ Select netmask enable and enter a valid netmask.

➢ If needed: Select the gateway enable and enter a valid IP address for the gateway.

3. Set length produced and consumed data length

➢ Enter in the field produced data length the number of data bytes the EtherNet/IP Scanner should receive from the netTAP or netBRICK device, e.g. 64 bytes

➢ Enter in the field consumed data length the number of data bytes the EtherNet/IP Scanner should send to the netTAP or netBRICK device, e.g. 32 bytes

4. Set more parameter

➢ If necessary, set more parameter. The parameter are described in the following section *EtherNet/IP Adapter Parameter* on page 155.

Note: Default values can be used normally for the other parameters. Depending on additionally requirements and the design of the EtherNet/IP network as well as the used devices parameters needs to be adjusted if necessary.

5. Open the signal configuration dialog

➢ Select in the navigation area under the folder **Configuration** the entry **Signal Configuration**.

➩ The signal configuration dialog opens

6. Signal configuration

➢ If you want to assign own signal names, then enter your signal names. Further information is described in section *Signal Configuration* on page 255.

7. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➩ The configuration dialog closes

4.9.1 EtherNet/IP Adapter Parameter

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 19: EtherNet/IP Adapter Parameters (Part 2)

4.9.2 Settings at the used EtherNet/IP Scanner

Device Description File

To configure the Scanner an EDS file (device description file) is required. The XML file is stored on the Gateway Solutions DVD in the directory `Electronic Data Sheets (e.g. EDS,GSD,GSDML)\EtherNetIP`:

Device	EDS File
NT 50-XX-EN or NT 50-EN-XX	HILSCHER NT 50-EN EIS V1.1.EDS
NT 100-RE-XX	HILSCHER NT 100-RE EIS V1.1.EDS
NB 100-RE-XX	HILSCHER NB 100-RE EIS V1.1.EDS
NT 151-RE-RE	HILSCHER NT 151-RE-RE EIS V1.1.EDS

Table 20: EDS File Names for netTAP and netBRICK

Configuration

Note: The settings in the used Scanner must comply with the settings in the Adapter, to establish a communication. Important parameters are: Input, Output length, Vendor ID, Product Type, Product Code, Major Rev, Minor Rev, IP Address and Netmask.

In order that the EtherNet/IP Scanner can communicate with the netTAP or netBRICK device as EtherNet/IP Adapter:

➢ Enter at the EtherNet/IP Scanner the IP address of the netTAP or netBRICK device.

➢ Use at the EtherNet/IP Scanner the instance ID 101, to receive data from the netTAP or netBRICK device.

➢ Enter at the EtherNet/IP Scanner for this instance ID (101) the number of data (number of receive data bytes), which were configured in the netTAP or netBRICK device, e. g. 64.

Note: Some EtherNet/IP Scanners need for setting this value (number of receive data) a value increased by 4 (this is the length of the Run/Idle header, that can be transferred in front of the user data). With the example of 64 bytes above therefore 68 (64 + 4) needs to be set for the number of data.

➢ If adjustable at the EtherNet/IP Scanner, set that the EtherNet/IP Scanner sends the 32-Bit Run/Idle header.

➢ Use at the EtherNet/IP Scanner the instance ID 100, to send data to the netTAP or netBRICK device.

➢ Enter at the EtherNet/IP Scanner for this instance ID (100) the number of data (number of send data bytes), which were configured in the netTAP or netBRICK device, e. g. 32.

➢ If adjustable at the EtherNet/IP Scanner, set that the EtherNet/IP Scanner receives the 32-Bit Run/Idle header.

- You can set at the EtherNet/IP Scanner, if the Scanner verifies identification numbers of the EtherNet/IP Adapter (named keying). This verification can be activated or deactivated at the EtherNet/IP Scanner. If the verification is used, then use or check for the following values:

Device	Product code (ProdCode)	Vendor (VendCode)	Product type (ProdType)	Major revision (MajRev)	Minor revision (MinRev)
NT 50-XX-EN; NT 50-EN-XX	275 (0x0113)	283 (0x011B)	12 (0x000C)	1	1
NT 100-RE-XX	271 (0x010F)				
NB 100-RE-XX	273 (0x0111)				
NT 151-RE-RE	288 (0x0120)				

Table 21: Identification parameters in EtherNet/IP EDS Files

4.10 Configuration Open Modbus/TCP

The netTAP NT 50 and NT 100 device respectively the netBRICK NB 100 device as Open Modbus/TCP needs parameter. The device can either work as a Client or as a Server.

These parameter can be edited as follows:

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > Open Modbus/TCP**

➮ The Open Modbus/TCP settings dialog opens.

➮ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the settings dialog is displayed.

2. Set protocol mode

➢ Set the **Protocol Mode** to "**I/O Server**", if the netTAP or netBRICK device should work as Open Modbus/TCP Server

➢ Set for the **Protocol Mode** to "**Client**", if the netTAP or netBRICK device should work as Open Modbus/TCP Client

➮ If you have set the **Protocol Mode** to "**Client**", then the entry **Command Table** appears in the navigation are

3. Open Modbus/TCP Parameter

   To set a fix IP address:

➢ Deselect the flag DHCP and BootP.

➢ Select IP address Enable and enter a valid IP address for the NT 100 device.

➢ Select netmask enable and enter a valid netmask.

➢ If needed: Select the gateway enable and enter a valid IP address for the gateway.

Note: Default values can be used normally for the other parameters. Depending on additionally requirements and the design of the Ethernet network as well as the used devices parameters needs to be adjusted if necessary. Further information is described in section *Open Modbus/TCP Parameter* on page 161.

4. Command Table

➢ If you have set the **Protocol Mode** to "**Client**", then select in the navigation area the entry **Command Table**. Specify in the Command Table which data the Open Modbus/TCP Client has to transfer to and from the Open Modbus/TCP Server by reading or writing. The Command Table is described in detail in section *Open Modbus/TCP* on page 163.

Note: 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.

5. Open the signal configuration dialog

➢ Select in the navigation area under the folder **Configuration** the entry **Signal Configuration**.

➩ The signal configuration dialog opens

6. Signal configuration

➢ If you want to assign own signal names, then enter your signal names. Further information is described in section *Signal Configuration* on page 255.

7. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➩ The configuration dialog closes

4.10.1 Open Modbus/TCP Parameter

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication starts automatically.	Default: Automatic
Watchdog time [ms]	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:: 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
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: checked

Parameter	Meaning	Range of Value/Value
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: unchecked.

Table 22: Open Modbus/TCP Parameters

4.10.2 Open Modbus/TCP Client

4.10.2.1 For what does the Command Table 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.

4.10.2.2 Parameter of the Command Table

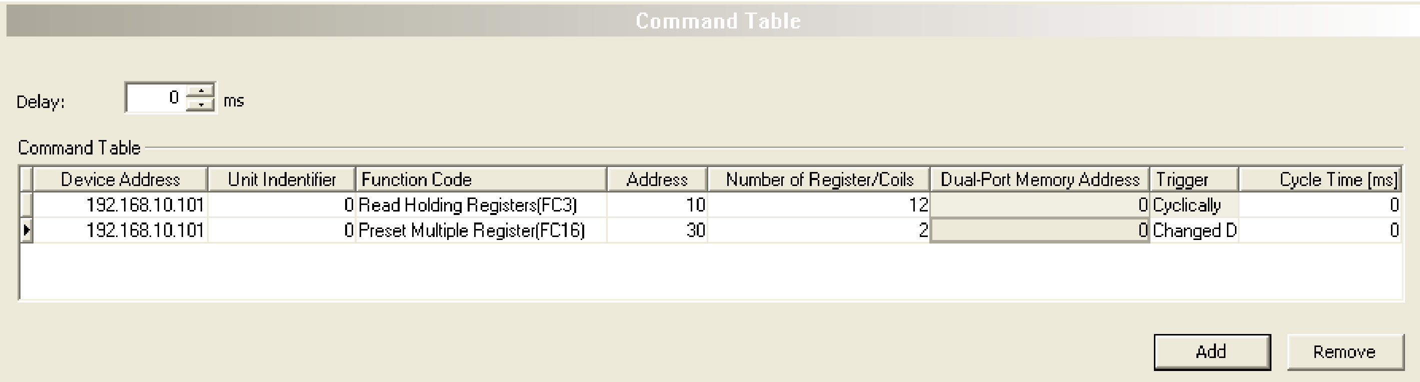


Figure 119: Pane Command Table

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

Table 23: 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>
<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 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.</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 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 24: Parameter of the Command Table

4.10.2.3 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.

4.10.2.4 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.

4.10.2.5 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 120: 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'.

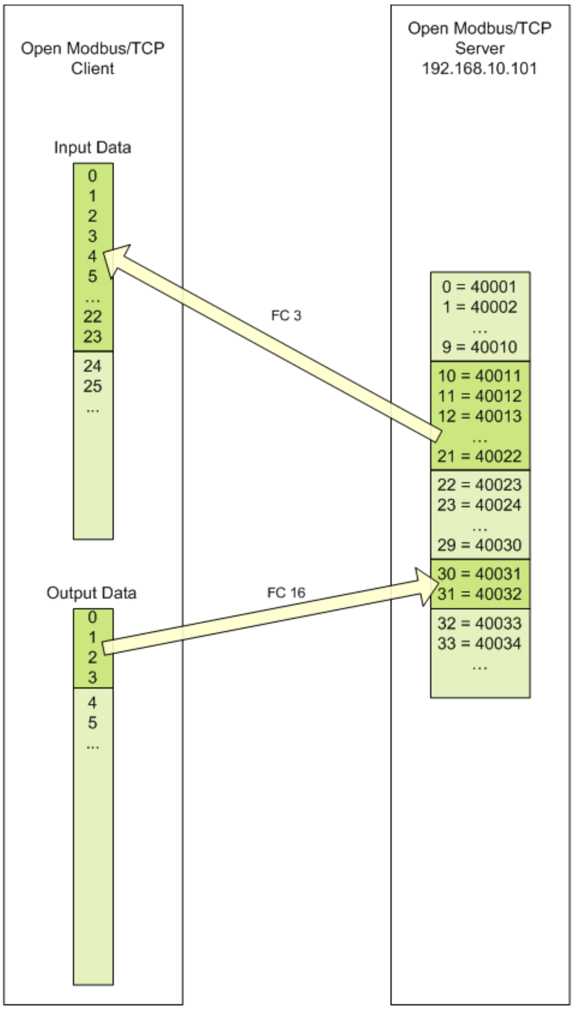


Figure 121: Examples - Reading Command with FC 3, Writing Command with FC16

4.10.3 Open Modbus/TCP Server

Settings at the used Open Modbus/TCP Client

In order that the Open Modbus/TCP Client can communicate with the netTAP or netBRICK device as an Open Modbus/TCP Server:

- Enter at the Open Modbus/TCP Client the IP address of the netTAP or netBRICK device.
- Use at the Open Modbus/TCP Client port number 502 an. Port 502 is used by the netTAP or netBRICK device to receive and send Open Modbus telegrams.
- Use at the Open Modbus/TCP Client function code 1, 2, 3, 4 or 23* to read data from the netTAP or netBRICK device. The following description shows the data assignment. (* function code 23 also writes data into the device).
- Use at the Open Modbus/TCP Client function code 5, 6, 15, 16 or 23* to write data into the netTAP or netBRICK device. The following description shows the data assignment. (* function code 23 also reads data from the device).

The Open Modbus/TCP Client can access to the netTAP or netBRICK device as an Open Modbus/TCP Server using function codes 1, 2, 3, 4, 5, 6, 15, 16 and 23.

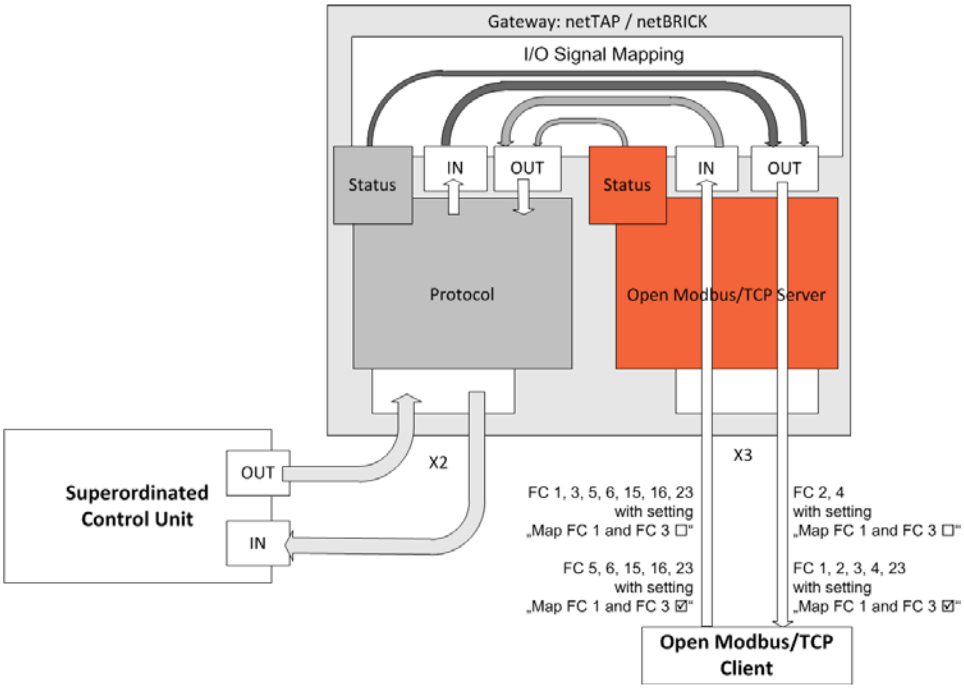


Figure 122: Access of the Open Modbus/TCP Client to the netTAP and netBRICK Device

Registers (16 bit value) and coils (1 bit value) share the same memory for input data (IN). Registers (16 bit value) and coils (1 bit value) share the same memory for output data (OUT). The following tables (Table 25 and Table 30) show the data assignment.

Adressing types of Open Modbus/TCP Clients

Open Modbus/TCP Clients of different manufacturers use different addressing types. The modicon-based addressing is very popular. Other Open Modbus/TCP Clients use a zero-based addressing and again other Open Modbus/TCP Clients use one-based addressing.

Note: Read the documentation of the used Open Modbus/TCP Client to find out the used addressing of this system.

Modicon-based addressing: If function code 3, 6, 16 and 23 is used, the first register (16 bit value) has address 40001, the second register has address 40002 etc. If function code 4 is used, the first register has address 30001, the second register has address 30002 etc. If function code 1, 5 and 15 is used, the first coil (1 bit value) has address 1, the second coil has address 2 etc. If function code 2 is used, the first coil has address 10001, the second coil has address 10002 etc.

The following table shows the address assignment for registers and coils for modicon-based Open Modbus/TCP Clients. The setting "**Map FC 1 and FC 3**" is **not set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP or netBRICK (Mapping)	Coil FC 1, FC 5, FC 15	Register FC 3, FC 6, FC 16, FC 23
IN 0	1 … 16	40001
IN 1	17 … 32	40002
IN 2	33 … 48	40003
…	…	…
IN 624	9985 … 9999	40625
…	-	…
IN 2879	-	42880

OUT netTAP or netBRICK (Mapping)	Coil FC 2	Register FC 4
OUT 0	10001 … 10016	30001
OUT 1	10017 … 10032	30002
OUT 2	10033 … 10048	30003
…	…	…
OUT 624	19985 … 19999	30625
…	-	
OUT 2879	-	32880

Table 25: Access of the Open Modbus/TCP Client using Modicon-based Addressing (1)

The following table shows the address assignment for registers and coils for modicon-based Open Modbus/TCP Clients. The setting "**Map FC 1 and FC 3**" is **set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP or netBRICK (Mapping)	Coil FC 5, FC 15	Register FC 6, FC 16, FC 23 (write)
IN 0	1 … 16	40001
IN 1	17 … 32	40002
IN 2	33 … 48	40003
…	…	…
IN 624	9985 … 9999	40625
…	-	…
IN 2879	-	42880

OUT netTAP or netBRICK (Mapping)	Coil FC 1	Coil FC 2	Register FC 4	Register FC 3, FC 23 (read)
OUT 0	1 … 16	10001 … 10016	30001	40001
OUT 1	17 … 32	10017 … 10032	30002	40002
OUT 2	33 … 48	10033 … 10048	30003	40003
…	…	…	…	…
OUT 624	9985 … 9999	19985 … 19999	30625	40625
…	-	-	…	…
OUT 2879	-	-	32880	42880

Table 26: Access of the Open Modbus/TCP Client using Modicon-based Addressing (2)

Zero-based addressing: The first register (16 bit value) as well as the first coil (1 bit value) has address 0, the second register as well as the first coil has address 1 etc. independed of the used function code.

The following table shows the address assignment for registers and coils for zero-based Open Modbus/TCP Clients. The setting “**Map FC 1 and FC 3**” is **not set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP or netBRICK (Mapping)	Coil FC 1, FC 5, FC 15	Register FC 3, FC 6, FC 16, FC 23
IN 0	0 … 15	0
IN 1	16 … 31	1
IN 2	32 … 47	2
…	…	…
IN 624	9985 … 9998	624
…	-	…
IN 2879	-	2879

OUT netTAP or netBRICK (Mapping)	Coil FC 2	Register FC 4
OUT 0	0 … 15	0
OUT 1	16 … 31	1
OUT 2	32 … 47	2
…	…	…
OUT 624	9985 … 9998	624
…	-	…
OUT 2879	-	2879

Table 27: Access of the Open Modbus/TCP Client using Zero-based Addressing (1)

The following table shows the address assignment for registers and coils for zero-based Open Modbus/TCP Clients. The setting “**Map FC 1 and FC 3**” is **set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP or netBRICK (Mapping)	Coil FC 5, FC 15	Register FC 3, FC 6, FC 16, FC 23 (schreiben)
IN 0	0 … 15	0
IN 1	16 … 31	1
IN 2	32 … 47	2
…	…	…
IN 624	9985 … 9998	624
…	-	…
IN 2879	-	2879

OUT netTAP or netBRICK (Mapping)	Coil FC 1, FC 2	Register FC3, FC 4, FC 23 (lesen)
OUT 0	0 … 15	0
OUT 1	16 … 31	1
OUT 2	32 … 47	2
…	…	…
OUT 624	9985 … 9998	624
…	-	…
OUT 2879	-	2879

Table 28: Access of the Open Modbus/TCP Client using Zero-based Addressing (2)

One-based addressing: The first register (16 bit value) as well as the first coil (1 bit value) has address 1, the second register as well as the first coil has address 2 etc. independed of the used function code.

The following table shows the address assignment for registers and coils for one-based Open Modbus/TCP Clients. The setting "**Map FC 1 and FC 3**" is **not set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP netBRICK (Mapping)	Coil FC 1, FC 5, FC 15	Register FC 3, FC 6, FC 16, FC 23
IN 0	1 … 16	1
IN 1	17 … 32	2
IN 2	33 … 48	3
…	…	…
IN 624	9986 … 9999	625
…	-	…
IN 2879	-	2880

OUT netTAP netBRICK (Mapping)	Coil FC 2	Register FC 4
OUT 0	1 … 16	1
OUT 1	17 … 32	2
OUT 2	33 … 48	3
…	…	…
OUT 624	9986 … 9999	625
…	-	…
OUT 2879	-	2880

Table 29: Access of the Open Modbus/TCP Client using One-based Addressing (1)

The following table shows the address assignment for registers and coils for one-based Open Modbus/TCP Clients. The setting "**Map FC 1 and FC 3**" is **set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP netBRICK (Mapping)	Coil FC 5, FC 15	Register FC 3, FC 6, FC 16, FC 23 (write)
IN 0	1 … 16	1
IN 1	17 … 32	2
IN 2	33 … 48	3
…	…	…
IN 624	9986 … 9999	625
…	-	…
IN 2879	-	2880

IN netTAP netBRICK (Mapping)	Coil FC 1, FC 2	Register FC3, FC 4, FC 23 (read)
OUT 0	1 … 16	1
OUT 1	17 … 32	2
OUT 2	33 … 48	3
…	…	…
OUT 624	9986 … 9999	625
…	-	…
OUT 2879	-	2880

Table 30: Access of the Open Modbus/TCP Client using One-based Addressing (2)

4.10.4 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

4.11 Configuration POWERLINK Controlled Node

Devices acting as POWERLINK Controlled Nodes need parameters.

This section describes how to set POWERLINK Controlled Node parameters for the netTAP NT 100 device, respectively the netBRICK NB 100 device.

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > POWERLINK Controlled Node**

➩ The POWERLINK Controlled Node configuration dialog opens.

➩ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the configuration dialog is displayed.

2. Set Node-ID

➢ Enter in the field **Node-ID** the address, which the netTAP or netBRICK device uses on the POWERLINK network to be addressed by the POWERLINK Managing Node.

3. Set input data length and output data length

➢ Set in the field **Input Data Length** the number of data bytes the netTAP or netBRICK device should receive from the POWERLINK Managing Node

➢ Set in the field **Output Data Length** the number of data bytes the netTAP or netBRICK device should send to the POWERLINK Managing Node

4. Enable PDO Mapping Version Check

➢ Uncheck the field **Disable PDO Mapping Version Check**

5. Set more parameter

➢ If necessary, set more parameter. The parameter are described in the following section *POWERLINK Controlled Node Parameter* on page 175.

Note: Default values can be used normally for the other parameters. Depending on additionally requirements and the design of the POWERLINK network as well as the used devices parameters needs to be adjusted if necessary.

6. Open the signal configuration dialog

➢ Select in the navigation area under the folder **Configuration** the entry **Signal Configuration**.

➾ The signal configuration dialog opens

7. Signal configuration

➢ If you want to assign own signal names, then enter your signal names. Further information is described in section *Signal Configuration* on page 255.

8. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➾ The configuration dialog closes

4.11.1 POWERLINK Controlled Node Parameter

Parameter	Meaning	Range of Value/Value
Interface		
Bus startup	Communication starts automatically	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
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: 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 NT 100-RE/PLS, NB 100-RE/PLS: 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

Parameter	Meaning	Range of Value/Value
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 31: POWERLINK Controlled Node/Slave Parameters

4.11.2 Settings at the used POWERLINK Managing Node

Device Description File

To configure the Managing Node a XDD file (device description file) is required. The XML file is stored on the Gateway Solutions DVD in the directory `Electronic Data Sheets (e.g. EDS,GSD,GSDML)\POWERLINK`:

Device	XDD File
NT 100-RE-XX	00000044_NT 100 RE PLS.XDD
NB 100-RE-XX	00000044_NB 100 RE PLS.XDD

Table 32: XDD File Names for netTAP and netBRICK

Configuration

Note: The settings in the used Managing Node must comply with the settings in the Controlled Node, to establish communication. Important parameters are: Vendor ID, Product Code, Serial Number, Revision Number, Node ID, Output and Input length.

In order that the POWERLINK Managing Node can communicate with the netTAP or netBRICK device as POWERLINK Controlled Node:

- ➢ Enter at the POWERLINK Managing Node the Node-ID, which is configured in the netTAP or netBRICK device.
- ➢ Enter for the length of output data in the POWERLINK Managing Node as set in the netTAP or netBRICK device for the Output Data Bytes.
- ➢ Enter for the length of input data in the POWERLINK Managing Node as set in the netTAP or netBRICK device for the Input Data Bytes.
- ➢ The POWERLINK Managing Node verifies identification numbers of the netTAP or netBRICK device as POWERLINK Controlled Node. Use or check for the following values for the netTAP or netBRICK device:

Device	Product code	Vendor	Serial number	Revision number
NT 100-RE-XX	24 (0x18)	68 (0x44)	0	0
NB 100-RE-XX	23 (0x17)			

Table 33: Identification parameters in POWERLINK XDD Files

4.12 Configuration PROFINET IO Controller

The netTAP NT 50, NT 100 device respectively the NT 151-RE-RE device respectively the netBRICK NB 100 device as PROFINET IO Controller needs a configuration, e. g. information about how many PROFINET IO Device devices with how many input and output data are to be connected.

1. Insert PROFINET IO Device devices

➢ Drag from the device catalog PROFINET IO Device device(s) and drop it to the bus line of the PROFINET IO Controller.

2. Configure PROFINET IO Device devices

➢ Open the configuration dialog for each PROFINET IO Device device and configure the device.

   Information about how to configure the PROFINET IO Device device is in the operating instruction manual „Generic DTM for PROFINET IO Device devices" and has the file name PROFINET_IO_GenericDevice_DTM_en.pdf respectively open in the configuration dialog of the slave the online help by the F1 key.

3. Configure PROFINET IO Controller

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > PROFINET IO Controller**

   Information about how to configure the PROFINET IO Controller device is in the operating instruction manual „DTM for PROFINET IO Controller devices" and has the file name PROFINET_IO_Controller_DTM_en.pdf respectively open in the configuration dialog of the master the online help by the F1 key.

4.13 Configuration PROFINET IO Device (Gateway)

Devices acting as PROFINET IO Devices need parameters.

This section describes how to set PROFINET IO Device parameters for the netTAP NT 100 device respectively the netBRICK NB 100 device.

Information on how to set PROFINET IO Device parameters for the netTAP **NT 151-RE-RE** device can be found in the Operating Instruction Manual *Generic DTM for PROFINET IO Devices*, DOC060305OIxxEN. This document is stored on the Gateway Solutions DVD in the directory: `Documentation\english\1.Software\SYCON.net Configuration Software\Master Configuration\PROFINET IO Controller\IO Device Configuration`.

1. Open the configuration dialog

- ➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > PROFINET IO Device**
- ➩ The PROFINET IO Device configuration dialog opens.
- ➩ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the settings dialog is displayed.

2. Set the number of Input Data Bytes and Output Data Bytes

- ➢ Enter in the field Input Data Bytes the number of data bytes, which should be transferred from the PROFINET IO Controller to the netTAP or netBRICK device, e. g. 32
- ➢ Enter in the field Output Data Bytes the number of data bytes, which should be transferred from the netTAP or netBRICK device to the PROFINET IO Controller, e. g. 64

3. Station name

The default name for the **Name of Station** for the NT 50 device is nt50enpns, for the NT 100 device nt100repns and for the NB 100 device nb100repns. If you need to set an other name of station or you use several netTAP or netBRICK devices in the PROFINET network, then set a unique name of station as follows:

- ➢ Select Enable in the Ident area.
- ➢ Enter in the field Name of Station the name of station, which the netTAP or netBRICK device should use in the PROFINET network.
- ➢ Enter in the field Vendor ID the value 0x011E.
- ➢ Enter in the field Device ID for the NT 50 value 0x010F, for the NT 100 value 0x010B respectively for the NB 100 value 0x010E.
- ➢ Values in the fields Device Type, Order ID and Type of Station are optional.

Note: Default values can be used normally for the other parameters. Depending on additionally requirements and the design of the PROFINET network as well as the used devices parameters needs to be adjusted if necessary. Further information is described in section *PROFINET IO Device Parameter* on page 181.

4. Open the signal configuration dialog

- ➢ Select in the navigation area under the folder **Configuration** the entry **Signal Configuration**.
- ➩ The signal configuration dialog opens

5. Signal configuration

- ➢ If you want to assign own signal names, then enter your signal names. Further information is described in section *Signal Configuration* on page 255.

6. Close configuration dialog

- ➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.
- ➩ The configuration dialog closes

4.13.1 PROFINET IO Device Parameter

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: 0x0000011E (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) 0x0000010B (hex), NT 100_RE/PNS (Proxy) 0x000000106 (hex), NB 100-RE/PNS (Gateway) 0x0000010E (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 34: PROFINET IO-Device Parameters

4.13.2 Settings at the used PROFINET IO Controller (Gateway)

Device Description File

To configure the Master a GSDML file (device description file) is required. The XML file is stored on the Gateway Solutions DVD in the directory `Electronic Data Sheets (e.g. EDS,GSD,GSDML)\PROFINET`:

Device	GSDML File
NT 50-XX-EN or NT 50-EN-XX	GSDML-V2.1-HILSCHER-NT 50-EN PNS-20100219.xml
NT 100-RE-XX	GSDML-V2.1-HILSCHER-NT 100-RE PNS-20090123.xml
NB 100-RE-XX	GSDML-V2.1-HILSCHER-NB 100-RE PNS-20091112.xml
NT 151-RE-RE	GSDML-V2.31-HILSCHER-NT 151-RE-RE PNS-20151021-1.xml

Table 35: GSDML File Names for netTAP and netBRICK

Configuration

Note: The settings in the used Controller must comply with the settings in the Device, to establish a communication. Important parameters are: Name of Station, Vendor ID, Device ID, Input and Output data length.

In order that the PROFINET IO Controller can communicate with the netTAP or netBRICK device as PROFINET IO Device:

Note: Under **Name of Station** in the PROFINET IO Controller the name must be used, which was configured in the netTAP or netBRICK device.

- Enter at the PROFINET IO Controller the Name of Station, which is configured in the NT 100 device.
- Use for the configuration of the PROFINET IO Controller modules e. g. ‚32 Bytes In', for receiving data from the netTAP or netBRICK device. Several modules can also be used, e. g. 2 times the module ‚16 Bytes In'. The number of data (counted in bytes), that results from the used modules for input data, has to match with the number of input data configured in the netTAP or netBRICK device.
- Use for the configuration of the PROFINET IO Controller modules e. g. ‚64 Bytes Out', for sending data to the netTAP or netBRICK device. Several modules can also be used, e. g. 4 times the module ‚16 Bytes Out'. The number of data (counted in bytes), that results from the used modules for output data, has to match with the number of output data configured in the netTAP or netBRICK device.

- The PROFINET IO Controller verifies identification numbers of the netTAP respectively netBRICK devices. The PROFINET IO Controller gets these numbers from the GSDML file:

Device	Device type (DeviceID)	Vendor ID (VendorID)
NT 50-XX-EN; NT 50-EN-XX	271 (0x010F)	286 (0x011E)
NT 100-RE-XX	267 (0x010B)	
NB 100-RE-XX	270 (0x010E)	
NT 151-RE-RE	289 (0x0121)	

Table 36: Identification parameters in GSDML Files

4.14 Configuration PROFINET IO Device (Proxy)

The NT 100 device respectively the netLINK NL 51N-DPL device as proxy with PROFINET IO Device needs parameter.

The number of input and output data for PROFINET IO result from the configuration of the PROFIBUS by the used PROFIBUS-DP slave devices and the used I/O modules per slave device.

The configuration of the device for PROFINET IO is determined by the configuration of the PROFIBUS network. In this process the station address of a PROFIBUS-DP slave device is mapped as slot module on PROFINET IO. The IO module of the respective slave device are mapped on PROFINET as subslot modules.

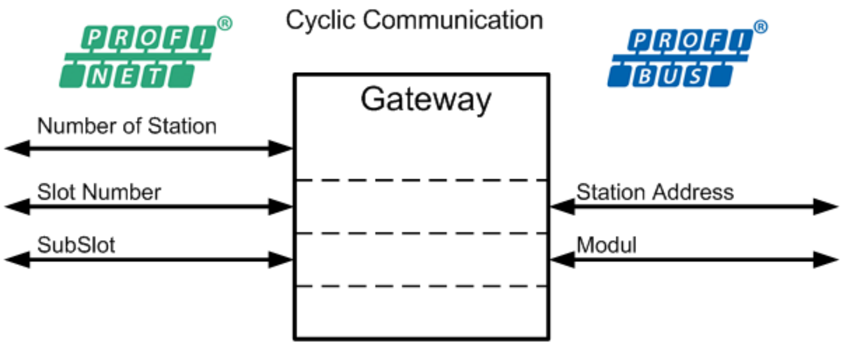


Figure 123: PROFINET – PROFIBUS Adress Mapping

Only the name of station needs to be set for the device.

This parameter can be edited as follows:

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netLINK symbol the entry **Configuration > PROFINET IO Device**

➩ The PROFINET IO Device configuration dialog opens.

➩ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the settings dialog is displayed.

2. Set name of station

The default name for the **Name of Station** for the NT 100 device as proxy is “nt100ereproxy” and for the NL 51N-DPL device “nl51ndpl”. If you need to set an other name of station or you use several netTAP or netLINK devices in the PROFINET network, then set a unique name of station as follows:

➢ Select in the navigation area under the folder **Configuration** the entry **General**.

➢ Enter in the field **Name of Station** the name of station, which the netTAP or netLINK device should use in the PROFINET network.

3. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➩ The configuration dialog closes

4.14.1 Settings at the used PROFINET IO Controller (Proxy)

Device Description File

To configure the Master a GSDML file (device description file) is required. The GSDML file depends on the PROFIBUS-DP configuration and is created by an export function. How to export the GSDML file and the structure of the GSDML file name is described for the netTAP NT 100 proxy device in section *Device Description File to configure the PROFINET IO Controller* on page 71 and for the netLINK NL 51N-DPL proxy device in section *Device Description File to configure the PROFINET IO Controller* on page 109.

Configuration

Note: The settings in the used Controller must comply with the settings in the Device, to establish a communication. Important parameters are: Name of Station, Vendor ID, Device ID, Input and Output data modules.

In order that the PROFINET IO Controller can communicate with the netTAP or netLINK device as PROFINET IO Device:

Note: Under **Name of Station** in the PROFINET IO Controller the name must be used, which was configured in the netTAP or netLINK device.

- ➢ Enter at the PROFINET IO Controller the Name of Station, which is configured in the netTAP or netLINK device.
- ➢ Use for the configuration of the PROFINET IO Controller modules based on the exported GSDML file.
- ➢ The PROFINET IO Controller verifies identification numbers of the netTAP respectively netLINK device: Vendor ID (value 0x011E (286)), device type (value for NT 100 as proxy is 0x0106 (262), value for NL 51N-DPL as proxy is 0x0110 (272)). The PROFINET IO Controller gets these numbers from the exported GSDML file.

4.15 Configuration sercos Slave

Devices acting as sercos Slave need parameters.

This section describes how to set sercos Slave parameters for the netTAP NT 100 device, NT 151-RE-RE device respectively the netBRICK NB 100 device.

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > SERCOS III Slave**

➮ The sercos Slave configuration dialog opens.

➮ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the configuration dialog is displayed.

2. Set Device ID

➢ Enter in the field **Device ID** the name, which the netTAP or netBRICK device uses on the sercos network to be identified by the sercos Master.

3. Accept Vendor Code

➢ Accept the default **Vendor Code** 0x000003E8 (1000 dec. = Hilscher).

4. Set Version SCP SYNC

➢ Set **Version SCP SYNC** to **0**, if the sercos Master uses the telegram sequence MDT-AT-NRT.

or

➢ Set **Version SCP SYNC** to **1**, if the sercos Master uses the telegram sequence MDT-NRT-AT.

5. Set Version SCP NRT

➢ Set **Version SCP NRT** to **0**. Use this setting only.

6. Set Address

➢ Enter in the field **SERCOS III Address** the address, which the netTAP or netBRICK device uses on the sercos network to be addressed by the sercos Master.

7. Set FSP Type

➢ Set for **FSP Type** value **IO V1**. Use this setting only.

8. Set Output Data Size for fixed Configuration

➢ Set in **Output Data Size for FixCfg** the number of output bytes, e. g. 2.

9. Set Input Data Size for fixed Configuration

➢ Set in **Input Data Size for FixCfg** the number of input bytes, e. g. 2.

10. Set no address change by the master

➢ Check **Master is not allowed to change sercos address** by that the sercos Master can't change the address. Use this setting only.

Note: If you use a sercos Master which is using SDDML files for the configuration of slave devices, then you have to export a new updated SDDML file from SYCON.net and import this SDDML file into the configuration software for the sercos Master. By that, the settings made here can be used at the sercos Master. Only then the sercos Master can communicate to the netTAP device respectively netBRICK device via sercos.

11. Export SDDML file

➢ Click **Export SDDML**

➮ The file dialog opens

➢ Enter the file name for the SDDML file. Use the following structure for the file name for a better identification: Manufacturer DeviceID output data size input data size. The file extension is always .xml. Example: Hilscher NT100-RE_S3S_FIXCFG 2out 2in.xml

➮ The settings are exported into a device description file.

12. Open the signal configuration dialog

➢ Select in the navigation area under the folder **Configuration** the entry **Signal Configuration**.

➮ The signal configuration dialog opens

13. Signal configuration

➢ If you want to assign own signal names, then enter your signal names. Further information is described in section *Signal Configuration* on page 255.

14. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➮ The configuration dialog closes.

4.15.1 sercos Slave Parameter

Parameter	Meaning	Range of Value/Value
Interface		
Bus Startup	Communication start automatic	Automatic, Default: Automatic
Watchdog Time [ms]	This function is not supported by gateway or proxy devices.	Default = 0 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 a new updated SDDML file from SYCON.net and import this SDDML file into the configuration software for the sercos Master.	Default: NT_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 a new updated SDDML file from SYCON.net 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/proxy device requires that the value is set to 1.	0 ... 1, Default: 0
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/proxy devices.	Default: 0

Further see next page

Parameter	Meaning	Range of Value/Value
Bus		
	Configuration parameter for IP communication	
Enabled	If 'Enabled' is unchecked, for the single Ident parameters each the respective default value is used.	checked, unchecked
IP Address	Valid IP address for the device If 'Enabled' is unchecked (Default setting), the device obtains its IP Address from a DHCP or BOOTP server. If 'Enabled' is checked, the device uses the manually entered value. This function is not supported by gateway/proxy devices.	Valid IP address, Default for 'Enabled': unchecked
Netmask	Valid Network mask for the device If 'Enabled' is unchecked (Default setting), the device obtains its Netmask from a DHCP or BOOTP server. If 'Enabled' is checked, the device uses the manually entered value. This function is not supported by gateway/proxy devices.	Valid network mask, Default for 'Enabled': unchecked
Gateway	Valid Gateway address for the device If 'Enabled' is unchecked (Default setting), the device obtains its Gateway Address from a DHCP or BOOTP server. If 'Enabled' is checked, the device uses the manually entered value. This function is not supported by gateway/proxy devices.	Valid gateway address, Default for 'Enabled': 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 by gateway/proxy devices.	checked, unchecked (Default)
	DHCP: If checked, the device obtains its IP Address, Netmask, Gateway Address from a DHCP server. This function is not supported by gateway/proxy devices.	checked (Default), unchecked

Further see next page

Parameter	Meaning	Range of Value/Value
	Configuration Parameters of the sercos Communication Profile Class SCP_Sync	
ConClk 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 „Control Clock 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 RE/S3S devices: 1000 ns
DivClk pulse distance *	This timing parameter defines the delay time from the start of the communication cycle to the first pulse of the communication synchronized hardware output signal DIV_CLK.	0 ... 16.777.210 ns, Default RE/S3S devices: 20000 ns
DivClk 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 „Divided Control Clock“ Mode1 this parameter is ignored.	0 ... 6.710.860 ns, Default RE/S3S devices: 20000 ns
DivClk 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 RE/S3S devices: 1000 ns
	*All timing values are handled as multiple of 10 ns (e. g. Control Clock Length = 1005 ns is handled as 1000 ns).	
DivClk multiplier	For „Divided Control Clock“ Mode0: Here the parameter indicates the number of pulses of the communication synchronized hardware output signal DIV_CLK within a communication cycle. For „Divided Control Clock“ Mode1: Here the parameter indicates the number of communication cycles.	0 ... 255, Default: 2
DivClk polarity	This communication synchronization flag defines the output state of the communication synchronized hardware output signal DIV_CLK.	Disabled, Active-high, Active-low, Default RE/S3S devices: Active-high
DivClk 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: „Divided Control Clock Length“ + 100 <= „Delay Time of Divided Clock“. Mode 1: The Div_Clk signal becomes active once after N communication cycles.	Mode 0, Mode 1, Default RE/S3S devices: Mode 1
ConClk polarity	This communication synchronization flag defines the output state of the communication synchronized hardware output signal CON_CLK.	Disabled, Active-high, Active-low, Default RE/S3S devices: Active-high

Further see next page

Parameter	Meaning	Range of Value/Value
Slave Configuration		
Number of Slaves	The number of used sercos addresses Fixed value for gateway/proxy devices.	1, Default: 1
Slave 1		
sercos Address	Address for the sercos Slave.	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, FSP Drive: A specification for mechanical drives. Set always FSP IO V1 for gateway/proxy devices.	Default: IO V1
SCP Configuration Type	SCP config type according to sercos third generation specification: 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 third generation specification 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. FixCFG Version 1.1.1 is fixed for gateway/proxy devices.	Default: Fix. Version 1.1.1
Output Data Size for Fixed Configuration	Output Data Size 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 SYCON.net and import this SDDML file into the configuration software for the sercos Master.	0 … 124 Byte, Default: 2 Byte
Input Data Size for Fixed Configuration	Intput Data Size 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 SYCON.net and import this SDDML file into the configuration software for the sercos Master.	0 … 124 Byte, Default: 2 Byte

Further see next page

Parameter	Meaning	Range of Value/Value
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 Gateway/proxy devices do not support User SCP Types. Therefore don't mark any User SCP 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/proxy devices this is always checked.	Default: checked
	Delete objects created by application on reset. For gateway/proxy devices this is always checked.	checked, Default: checked
	Master is not allowed to to change sercos address: enables (unchecked) or disables (checked), that the Master changes the sercos address. For gateway/proxy devices this must always be checked.	checked, unchecked, Default: checked
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	Real Time Data Maximum Length for the Slave connections 1, 2, 3 and 4 Only displayed.	Default: 126 Byte

Table 37: sercos Slave Parameter

4.15.2 Settings at the used sercos Master

Device Description File

If you use a sercos Master which is using SDDML files for the configuration of slave devices, then export with SYCON.net your settings into a SDDML file as described in section *Configuration sercos Slave* on page 186.

If you use default settings, then you can use the default SDDML file. The default SDDML files are stored on the Gateway Solutions DVD in the directory `Electronic Data Sheets (e.g. EDS,GSD,GSDML)\SERCOS_III`:

Device	XML File
NT 100-RE-XX	Hilscher NT100 RE S3S FixCFG FSPIO Default.xml
NB 100-RE-XX	Hilscher NB100 RE S3S FixCFG FSPIO Default.xml
NT 151-RE-RE	Hilscher NT151 RE S3S FixCFG FSPIO Default

Table 38: XML File Names for netTAP and netBRICK

These device description files includes the following settings:

<table>
<tr><th>Gerät</th><th>DeviceID</th><th>VendorCode</th><th>Output data size</th><th>Input data size</th></tr>
<tr><td>NT 100-RE-XX</td><td>NT_100-RE_S3S_FIXCFG</td><td rowspan="3">1000 (0x03E8)</td><td rowspan="3">2</td><td rowspan="3">2</td></tr>
<tr><td>NB 100-RE-XX</td><td>NB_100-RE_S3S_FIXCFG</td></tr>
<tr><td>NT 151-RE-RE</td><td>NT_151-RE_RE_S3S_FIXCFG</td></tr>
</table>

Table 39: Default settings in Sercos XML Files

If you use these default settings for the netTAP device respectively netBRICK device, then you can use the SDDML file listed above. If you have set at least one setting a different value than the default value, then you have to export the device description file from SYCON.net and import it into the configuration software of the sercos Master.

Configuration

Note: The settings in the used Master must comply with the settings in the Slave to establish communication. Important parameters are: sercos address, vendor code, Device ID, FSP type with value IO V1, SCP configuration type with value FixCFG, the output data size and the input data size.

4.16 Configuration CANopen Master

The netTAP NT 50-, NT 100 device respectively the netBRICK NB 100 device as CANopen Master needs a configuration, e. g. information about how many CANopen Slave devices with how many input and output data are to be connected.

1. Insert CANopen Slave devices

➢ Drag from the device catalog CANopen Slave device(s) and drop it to the bus line of the CANopen Master.

2. Configure CANopen Slave devices

➢ Open the configuration dialog for each CANopen Slave device and configure the device.

Information about how to configure the CANopen Slave device is in the operating instruction manual „Generic DTM for CANopen Slave devices" and has the file name CANopen_Generic_Slave_DTM_en.pdf respectively open in the configuration dialog of the slave the online help by the F1 key.

3. Configure CANopen Master

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > CANopen Master**

Information about how to configure the CANopen Master device is in the operating instruction manual „DTM for CANopen Master devices" and has the file name CANopen_Master_netX_DTM_en.pdf respectively open in the configuration dialog of the master the online help by the F1 key.

4.17 Configuration CANopen Slave

The netTAP NT 50 and NT 100 device respectively the netBRICK NB 100 device as CANopen Slave needs parameter.

These parameter can be edited as follows:

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > CANopen Slave**

➯ The DeviceNet Slave configuration dialog opens.

➯ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the configuration dialog is displayed.

2. Set Node-ID

➢ Enter in the field **Node-ID** the address, which the netTAP or netBRICK device uses on the CANopen network to be addressed by the CANopen Master.

3. Set baudrate

➢ Set the baudrate for the netTAP or netBRICK device, which is used in the CANopen network

4. Set more parameter

➢ If necessary, set more parameter. The parameter are described in the following section *CANopen Slave Parameter* on page 195.

Note: Default values can be used normally for the other parameters. Depending on additionally requirements and the design of the DeviceNet network as well as the used devices parameters needs to be adjusted if necessary.

Note: The maximum number of output data bytes and input data bytes is for the netTAP or netBRICK device 512 bytes each. The number of output data bytes and input data bytes is not adjustable in the Slave and therefore grayed 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 netTAP or netBRICK device during establishing the communication and set the number of output data bytes and input data bytes at this same time.

5. Open the signal configuration dialog

➢ Select in the navigation area under the folder **Configuration** the entry **Signal Configuration**.

➩ The signal configuration dialog opens

6. Signal configuration

➢ If you want to assign own signal names, then enter your signal names. Further information is described in section *Signal Configuration* on page 255.

7. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➩ The configuration dialog closes

4.17.1 CANopen Slave Parameter

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		
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 ... 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 Address	Node ID of CANopen slave	1 … 127, Default: 2
Baudrate	Baud rate of CANopen connection	1 Mbaud, 800 Kbaud, 500 Kbaud, 250 Kbaud, 125 Kbaud, 100 Kbaud, 50 Kbaud, 20 Kbaud, 10 Kbaud, Default: 1 MBaud

Table 40: CANopen Slave Parameters (Part 1)

Data		
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 41: CANopen Slave Parameters (Part 2)

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.

4.17.2 Settings at the used CANopen Master

Device Description File

To configure the Master an EDS file (device description file) may be required. The EDS file is on the delivered DVD in the directory EDS/CANopen:

Device	EDS File
NT 50-XX-CO or NT 50-CO-XX	NT50_CO_COS.EDS
NT 100-XX-CO or NT 100-CO-XX	NT100_CO_COS.EDS
NB 100-XX-CO or NB 100-CO-XX	NB100_CO_COS.EDS

Table 42: EDS File Names for netTAP and netBRICK

Configuration

Note: The settings in the used Master must comply with the settings in the Slave to establish communication. Important parameters are: node address and baudrate.

In order that the CANopen Master can communicate with the netTAP or netBRICK device as CANopen Slave:

- Enter at the CANopen Master the Node-ID, which is configured in the netTAP or netBRICK device.
- Configure at the CANopen Master all PDOs, which the Master should send to the netTAP or netBRICK device. For the netTAP or netBRICK device these are receive PDOs. E. g. the Master can be configured with 2 PDOs with 8 byte user data each (16 bytes in total) to be send to the netTAP or netBRICK device.
- Configure at the CANopen Master all PDOs, which the Master should receive from the netTAP or netBRICK device. For the netTAP or netBRICK device these are transmit PDOs. E. g. the Master can be configured with 4 PDOs with 8 byte user data each (32 bytes in total) to be received from the netTAP or netBRICK device.

4.18 Configuration CC-Link Slave

The netTAP NT 50 and NT 100 device as CC-Link Slave needs parameter.

These parameter can be edited as follows:

1. Open the configuration dialog

➢ Select from the context menu of the netTAP symbol the entry **Configuration > CC-Link Slave**

➩ The CC-Link Slave configuration dialog opens.

➩ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the configuration dialog is displayed.

2. Set Stationaddress

➢ Enter in the field **Stationaddress** the address, which the netTAP device uses on the CC-Link network to be addressed by the CC-Link Master.

3. Set baudrate

➢ Set the baudrate for the netTAP device, which is used in the CC-Link network

4. Set the CC-Link version

➢ Select with the field **CC-Link Version** the CC-Link version the netTAP device should use to communicate with the CC-Link Master.

5. Set the Station type

➢ If CC-Link Version 1 is to be used, then select with the field **Station Type** either **Remote Device Station** or **Remote IO Station** for the netTAP device.

6. Set the Number of Stations

➢ If Station type is **Remote Device Station**, then set the number of stations for the netTAP device: 1, 2, 3 or 4.

7. Set number of extension cycles

➢ If CC-Link Version 2 is to be used, then set the number of extension cycles for the netTAP device: 1, 2, 3 or 4.

Note: 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.

8. Set more parameter

➢ If necessary, set more parameter. The parameter are described in the following section *CC-Link Slave Parameter* on page 202.

Note: Default values can be used normally for the other parameters. Depending on additionally requirements and the design of the CC-Link network as well as the used devices parameters needs to be adjusted if necessary.

9. Open the signal configuration dialog

➢ Select in the navigation area under the folder **Configuration** the entry **Signal Configuration**.

➩ The signal configuration dialog opens

10. Signal configuration

➢ If you want to assign own signal names, then enter your signal names. Further information is described in section *Signal Configuration* on page 255.

11. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➩ The configuration dialog closes

4.18.1 CC-Link Slave Parameter

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		
Enabled	If ‘Enabled’ is unchecked, for the single Ident parameters each the respective default value is used.	
Vendor Code	Code for the Vendor	0 ... 65535 resp. 0x00000000 ... 0x0000FFFF (hex), Hilscher: 0x00000352 (hex)
Model Type	Model type	0 ... 255, Default: 1
Software Version	Software version	0 ... 63, Default: 0
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
Baud Rate	Network transmission rate	156 kBaud (Default) 625 kBaud 2500 kBaud 5 MBaud 10 MBaud
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

Table 43: CC-Link Slave Parameters (Part 1)

Data		
CC-Link Version	CC-Link Version 1 CC-Link Version 2	1 (Default) 2
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 Cycle	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
IO-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 44: CC-Link Slave Parameters (Part 2)

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 additionally important: number of cycles as well as number of extension cycles.

4.18.2 Settings at the used CC-Link Master

Device Description File

To configure the Master, a CSP file (device description file) may be required. The CSP file is on the delivered DVD in the directory EDS/CC-Link:

Device	CSP File
NT 50-XX-CC or NT 50-CC-XX	nt50-cc-ccs_1.csp, nt50-cc-ccs_2.csp, nt50-cc-ccs_3.csp, nt50-cc-ccs_4.csp, nt50-cc-ccs_io.csp
NT 100-XX-CC or NT 100-CC-XX	Nt100-cc-ccs_1.csp, nt100-cc-ccs_2.csp, nt100-cc-ccs_3.csp, nt100-cc-ccs_4.csp, nt100-cc-ccs_io.csp

Table 45: CSP File Names for netTAP

The files nt50-cc-ccs_1.csp and nt100-cc-ccs_1.csp describe the respective netTAP device as one Remote Device Station, the files nt50-cc-ccs_2.csp and nt100-cc-ccs_2.csp describe the respective netTAP device as two Remote Device Station, etc. The files nt50-cc-ccs_io.csp and nt100-cc-ccs_io.csp describe the netTAP device as one Remote IO Station.

Configuration

Note: The settings in the used Master must comply with the settings in the Slave to establish communication. Important parameters are: Slave Station Address, Baudrate and Station Type. For CC-Link Version 1 are additionally important: Station type. For CC-Link Version 2.00 are additionally important: number of cycles as well as number of extension cycles.

In order that the CC-Link Master can communicate with the netTAP device as CC-Link Slave:

- Enter at the CC-Link Master the Stationaddress, which is configured in the netTAP device.
- Set at the CC-Link Master the same CC-Link Version as configured in the netTAP device.
- Set at the CC-Link Master the same station type as configured in the netTAP device.
- If CC-Link Version 1 is to be used, then set at the CC-Link Master the same Station type as configured in the netTAP device.
- If Station type is **Remote Device Station** to be used, then set at the CC-Link Master the same the number of stations as configured in the netTAP device.
- If CC-Link Version 2 is to be used, then set at the CC-Link Master the same number of extension as configured in the netTAP device.

4.19 Configuration DeviceNet Master

The netTAP NT 50, NT 100 device respectively the netBRICK NB 100 device as DeviceNet Master needs a configuration, e. g. information about how many DeviceNet Slave devices with how many input and output data are to be connected.

1. Insert DeviceNet Slave devices

➢ Drag from the device catalog DeviceNet Slave device(s) and drop it to the bus line of the DeviceNet Master.

2. Configure DeviceNet Slave devices

➢ Open the configuration dialog for each DeviceNet Slave device and configure the device.

Information about how to configure the DeviceNet Slave device is in the operating instruction manual „Generic DTM for DeviceNet Slave devices“ and has the file name DeviceNet_Generic_Slave_DTM_en.pdf respectively open in the configuration dialog of the slave the online help by the F1 key.

3. Configure DeviceNet Master

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > DeviceNet Master**

Information about how to configure the DeviceNet Master device is in the operating instruction manual „DTM for DeviceNet Master devices“ and has the file name DeviceNet_Master_netX_DTM_en.pdf respectively open in the configuration dialog of the master the online help by the F1 key.

4.20 Configuration DeviceNet Slave

The netTAP NT 50 and NT 100 device respectively the netBRICK NB 100 device as DeviceNet Slave needs parameter.

These parameter can be edited as follows:

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > DeviceNet Slave**

➩ The DeviceNet Slave configuration dialog opens.

➩ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the configuration dialog is displayed.

2. Set MAC-ID

➢ Enter in the field **MAC-ID** the address, which the netTAP or netBRICK device uses on the DeviceNet network to be addressed by the DeviceNet Master.

3. Set baudrate

➢ Set the baudrate for the netTAP or netBRICK device, which is used in the DeviceNet network

4. Set produced data length and consumed data length

➢ Set in the field **Prod. Data Length** the number of data bytes the netTAP or netBRICK device should send to the DeviceNet Master

➢ Set in the field **Cons. Data Length** the number of data bytes the netTAP or netBRICK device should receive from to the DeviceNet Master

5. Set more parameter

➢ If necessary, set more parameter. The parameter are described in the following section *DeviceNet Slave Parameter* on page 208.

Note: Default values can be used normally for the other parameters. Depending on additionally requirements and the design of the DeviceNet network as well as the used devices parameters needs to be adjusted if necessary.

6. Open the signal configuration dialog

➢ Select in the navigation area under the folder **Configuration** the entry **Signal Configuration**.

➩ The signal configuration dialog opens

7. Signal configuration

➢ If you want to assign own signal names, then enter your signal names. Further information is described in section *Signal Configuration* on page 255.

8. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➩ The configuration dialog closes

4.20.1 DeviceNet Slave Parameter

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		
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 ... 0x0000FFFF (hex), Hilscher: 0x0000011B (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)
Minor Rev	Minor Revision	1 ... 255, Default: 1
Major Rev	Major 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

Table 46: DeviceNet Slave Parameters (Part 1)

Bus		
MAC ID	This parameter defines the DeviceNet address of the device within the network.	0 … 63, Default: 2
Baudrate	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
	Continue on CAN 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 Clear 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		
Prod. Data Length	Produced data length sets the number of send bytes.	0 … 255, Default: 8
Cons. Data Length	Consumed data length sets the number of receive bytes.	0 … 255, Default: 8

Table 47: DeviceNet Slave Parameters (Part 2)

4.20.2 Settings at the used DeviceNet Master

Device Description File

To configure the Master an EDS file (device description file) is required. The EDS file is on the delivered DVD in the directory EDS/DeviceNet:

Device	EDS File
NT 50-XX-DN or NT 50-DN-XX	NT50_DN_DNS.EDS
NT 100-XX-DN or NT 100-DN-XX	NT100_DN_DNS.EDS
NB 100-XX-DN or NB 100-DN-XX	NB100_DN_DNS.EDS

Table 48: EDS File Names for netTAP and netBRICK

Configuration

Note: 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.

In order that the DeviceNet Master can communicate with the netTAP or netBRICK device as DeviceNet Slave:

- Enter at the DeviceNet Master the MAC-ID, which is configured in the netTAP or netBRICK device.
- Enter for the length of output data in the DeviceNet Master the Cons. Data Length as set in the netTAP or netBRICK device.
- Enter for the length of input data in the DeviceNet Master the Prod. Data Length as set in the netTAP or netBRICK device.
- You can set at the DeviceNet Master, if the Master verifies identification numbers of the DeviceNet Slave (named keying). This verification can be activated or deactivated at the DeviceNet Master. If the verification is used, then use or check for the following values for the netTAP or netBRICK device: vendor 283 (0x011B), product code 45 (0x2D) for NT 50 respectively product code 43 (0x2B) for NT 100 respectively product code 47 (0x2F) for NB 100, product type 12 (0x000C), major revision 1, minor revision 1.

4.21 Configuration PROFIBUS-DP Master

The netTAP NT 50-, NT 100 device respectively the netBRICK NB 100 device as PROFIBUS-DP Master needs a configuration, e. g. information about how many PROFIBUS-DP Slave devices with how many input and output data are to be connected.

1. Insert PROFIBUS-DP Slave devices

➢ Drag from the device catalog PROFIBUS-DP Slave device(s) and drop it to the bus line of the PROFIBUS-DP Master.

2. Configure PROFIBUS-DP Slave devices

➢ Open the configuration dialog for each PROFIBUS-DP Slave device and configure the device.

Information about how to configure the PROFIBUS-DP Slave device is in the operating instruction manual „Generic DTM for PROFIBUS-DP Slave devices“ and has the file name PROFIBUS_Generic_Slave_DTM_en.pdf respectively open in the configuration dialog of the slave the online help by the F1 key.

3. Configure PROFIBUS-DP Master

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > PROFIBUS-DP Master**

Information about how to configure the PROFIBUS-DP Master device is in the operating instruction manual „DTM for PROFIBUS-DP Master devices“ and has the file name PROFIBUS_Master_netX_DTM_en.pdf respectively open in the configuration dialog of the master the online help by the F1 key.

4.22 Configuration PROFIBUS-DP Slave

The netTAP NT 50 and NT 100 device respectively the netBRICK NB 100 device as PROFIBUS-DP Slave needs parameter.

These parameter can be edited as follows:

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > PROFIBUS-DP Slave**

➾ The PROFIBUS-DP Slave configuration dialog opens.

➾ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the configuration dialog is displayed.

2. Set station address

➢ Enter in the field **Station address** the address, which the netTAP or netBRICK device uses on the PROFIBUS network to be addressed by the PROFIBUS-DP Master.

3. Set baudrate

➢ Use the setting Auto-Detect (default), if the netTAP or netBRICK device should detect the baudrate on the PROFIBUS network or set the baud rate for the device.

4. Set the number of input and output data

➢ Enter for the output module type and size of the output module. Up to 4 output modules can be set. With the output modules the number of data is configured which the netTAP or netBRICK device should receive from the PROFIBUS-DP Master.

➢ Enter for the input module type and size of the input module. Up to 4 input modules can be set. With the input modules the number of data is configured which the netTAP or netBRICK device should send to the PROFIBUS-DP Master.

Note: The input and output modules are ‚with consistence'.

5. Set more parameter

➢ If necessary, set more parameter. The parameter are described in the following section *PROFIBUS-DP Slave Parameter* on page 214.

Note: Default values can be used normally for the other parameters. Depending on additionally requirements and the design of the PROFIBUS network as well as the used devices parameters needs to be adjusted if necessary.

6. Open the signal configuration dialog

➢ Select in the navigation area under the folder **Configuration** the entry **Signal Configuration**.

➩ The signal configuration dialog opens

7. Signal configuration

➢ If you want to assign own signal names, then enter your signal names. Further information is described in section *Signal Configuration* on page 255.

8. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➩ The configuration dialog closes

4.22.1 PROFIBUS-DP Slave Parameter

Parameter	Meaning	Range of Value/Value
Interface		
Bus Startup	Automatic communication start	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 If ‘Enabled’ is unchecked, the default value is used.	0x00000000 ... 0x0000FFFF (hex), Default: NT 50: 0x000000C99 (hex), NT 100: 0x000000C0E (hex), NB 100: 0x00000C9C (hex)
Enable	If ‘Enabled’ is unchecked, the default value is used.	
Bus		
Station Address	PROFIBUS address of the device	0 … 126
Baudrate	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
	DPV1 Enable: If checked, DPV1 is supported or the DPV1 functions are activated.	
	Freeze supported: If checked, the Slave stack supports the FREEZE command or the FREEZE mode is activated.	
	Address change by the Master not allowed: If checked, the Slave rejects the Set Slave Address command to the Master. If not checked, the Slave supports the Set Slave Address command and accepts an address change by the Master.	
	Fail safe supported: If checked, the FAILSAFE operation is supported or the FAILSAFE mode is activated.	

Table 49: Parameters - PROFIBUS-DP Slave (Part 1)

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“ (Default), „Word“ 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, Words)
Output Data Bytes	Total of the output bytes of the modules 1 to 4	0 … 244, Default: 2
Input Data Bytes	Total of the input bytes of the modules 5 to 8	0 … 244, Default: 2
Custom data	Custom data: If 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 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
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. Depending on the setting of the field Custom data, the field Configuration data is for display only or an entry field.	Default: A1, 91 hex

Table 50: Parameters - PROFIBUS-DP Slave (Part 2)

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

4.22.2 Settings at the used PROFIBUS-DP Master

Device Description File

To configure the Master a GSD file (device description file) is required. The GSD file is on the delivered DVD in the directory EDS/PROFIBUS:

Device	GSD File
NT 50-XX-DP or NT 50-DP-XX	HIL_0C99.GSD
NT 100-XX-DP or NT 100-DP-XX	HIL_0C0E.GSD
NB 100-XX-DP or NB 100-DP-XX	HIL_0C9C.GSD

Table 51: GSD File Names for netTAP and netBRICK

Configuration

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

In order that the PROFIBUS-DP Master can communicate with the netTAP or netBRICK device as PROFIBUS-DP Slave:

- Enter at the PROFIBUS-DP Master the station address, which is configured in the netTAP or netBRICK device.
- Use for the configuration of the PROFIBUS-DP Master module(s) e. g. ‚64 Bytes Out', for sending data to the netTAP or netBRICK device. Use the module(s), which are configured in the netTAP or netBRICK device. (Modules with the size of zero in the netTAP or netBRICK configuration needs not to be considered for the configuration of the PROFIBUS-DP Master.)
- Use for the configuration of the PROFIBUS-DP Master module(s) e. g. ‚32 Bytes In', for receiving data from the netTAP or netBRICK device. Use the module(s), which are configured in the netTAP or netBRICK device. (Modules with the size of zero in the netTAP or netBRICK configuration needs not to be considered for the configuration of the PROFIBUS-DP Master.)

4.23 Configuration Modbus RTU Master/Slave

The netTAP NT 50 and NT 100 device respectively the netBRICK NB 100 device as Modbus RTU Master or Slave needs parameter.

These parameter can be edited as follows:

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > Modbus RTU**

➩ The Modbus RTU settings dialog opens.

➩ In the navigation area the entry **Configuration** under the folder **Configuration** is selected and the settings dialog is displayed.

2. Set protocol mode

➢ Set the **Protocol Mode** to "**I/O Slave**", if the netTAP or netBRICK device should work as Modbus RTU Slave

➢ Set for the **Protocol Mode** to "**Master**", if the netTAP or netBRICK device should work as Modbus RTU Master

➩ If you have set the **Protocol Mode** to "**Master**", then the entry **Command Table** appears in the navigation are

3. Set Modbus RTU Parameter

➢ Set or select in the configuration dialog the Modbus Address, the interface type, RTS control, baudrate, number of stop bits and parity for the netTAP or netBRICK device. The settings are described in the following section *Modbus RTU Parameter* on page 219.

➢ If you have set the **Protocol Mode** to "**Master**", then set the response timeout and the send retries.

4. Command Table

➢ If you have set the **Protocol Mode** to "**Master**", then select in the navigation area the entry **Command Table**. Specify in the Command Table which data the Modbus RTU Master has to transfer to and from the Modbus RTU Slave by reading or writing. The Command Table is described in detail in section *Modbus RTU Master* on page 221.

Note: 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.

5. Open the signal configuration dialog

- ➢ Select in the navigation area under the folder **Configuration** the entry **Signal Configuration**.
- ➩ The signal configuration dialog opens

6. Signal configuration

- ➢ If you want to assign own signal names, then enter your signal names. Further information is described in section *Signal Configuration* on page 255.

7. Close configuration dialog

- ➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.
- ➩ The configuration dialog closes

4.23.1 Modbus RTU Parameter

Parameter	Meaning	Range of Value/Value
Interface		
Bus Startup	Communication startsautomatically	Automatic (Default)
Watchdog Time [ms]	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 ‘Enabled’ 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 (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	Serial interface to be used	RS232, RS485, RS422, SPI Mode3, Default: RS232
RTS Control	Signal “Rquest to Send can be switched off or on. See hardware description	RTS Control Off (default), RTS Control On
Baudrate	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	Stop Bits: Defines the quantity of stop bits, 1 or 2, to be used in protocol for serial data communication	One stop bit (default) Two stop bits
Parity	Defines the parity bit for serial data communication.	None, Even, Odd, Default: Even
Frame Format	This parameter is not supported here.	

For more see next page

Parameter	Meaning	Range of Value/Value
Data		
Register	Number of register Not editable in protocol mode master	0 ... 2880, Default: 2880
Coils	Number of coils not editable	0… 65535, Default: Automatic

Table 52: Modbus RTU Parameters

4.23.2 Modbus RTU Master

4.23.2.1 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.

4.23.2.2 Parameter of the Command Table

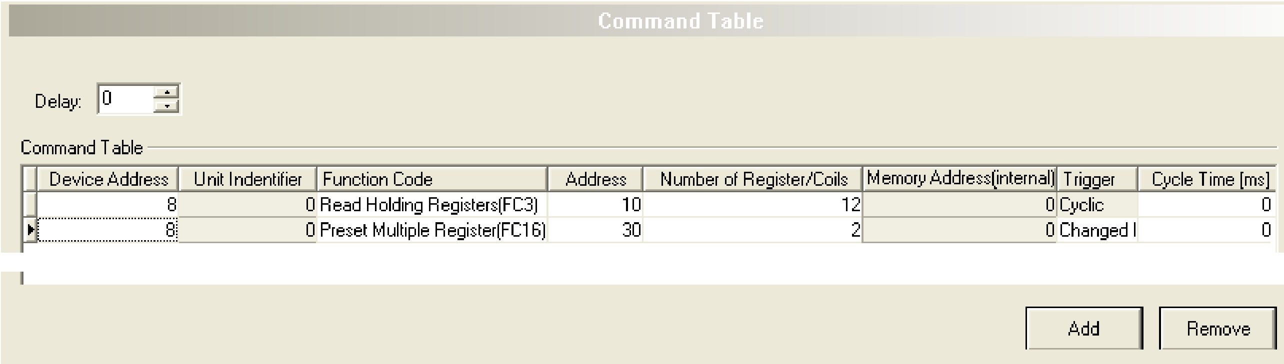


Figure 124: Pane Command Table

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

<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 Address</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

4.23.2.3 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.

4.23.2.4 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.

4.23.2.5 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 125: 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'.

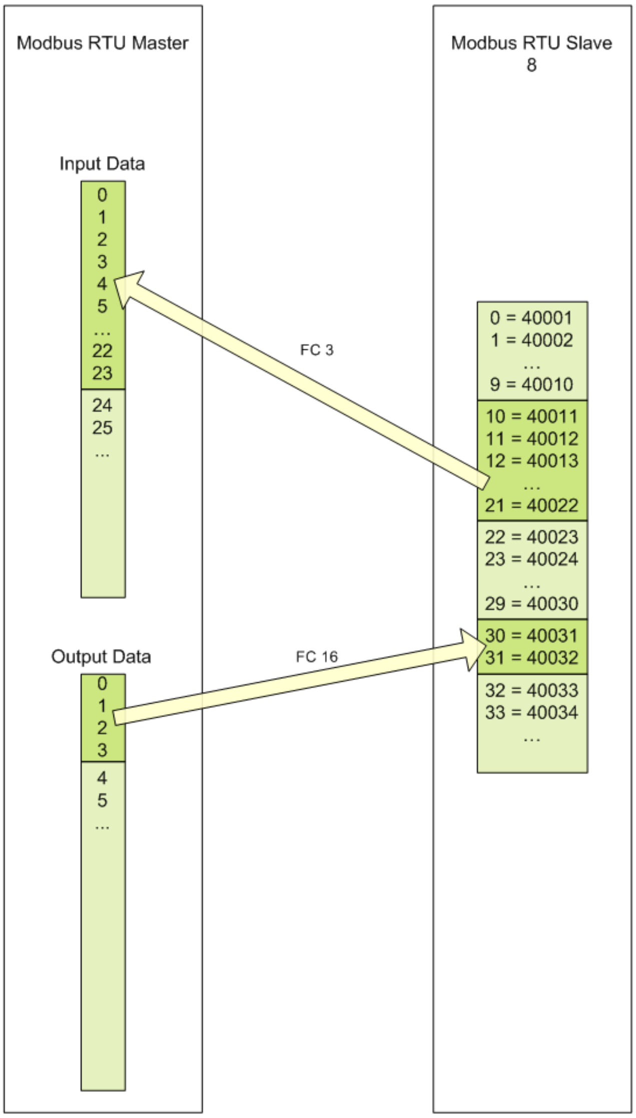


Figure 126: Examples - Reading Command with FC 3, Writing Command with FC16

4.23.2.6 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.

4.23.3 Modbus RTU Slave

4.23.3.1 Settings at the used Modbus RTU Master

Note: The settings at the used Modbus RTU Master must comply with the settings in the netTAP or netBRICK device (Modbus RTU Slave) to establish a communication. Important parameters are: Modbus RTU address, baud rate, number of stop bits and parity setting. However, the Modbus RTU address must be unique, e. g. the Modbus RTU address must be different from device to device.

The Modbus RTU Master can access to the netTAP or netBRICK device as a Modbus RTU Slave using function codes 1, 2, 3, 4, 5, 6, 7, 8, 15, 16 and 23.

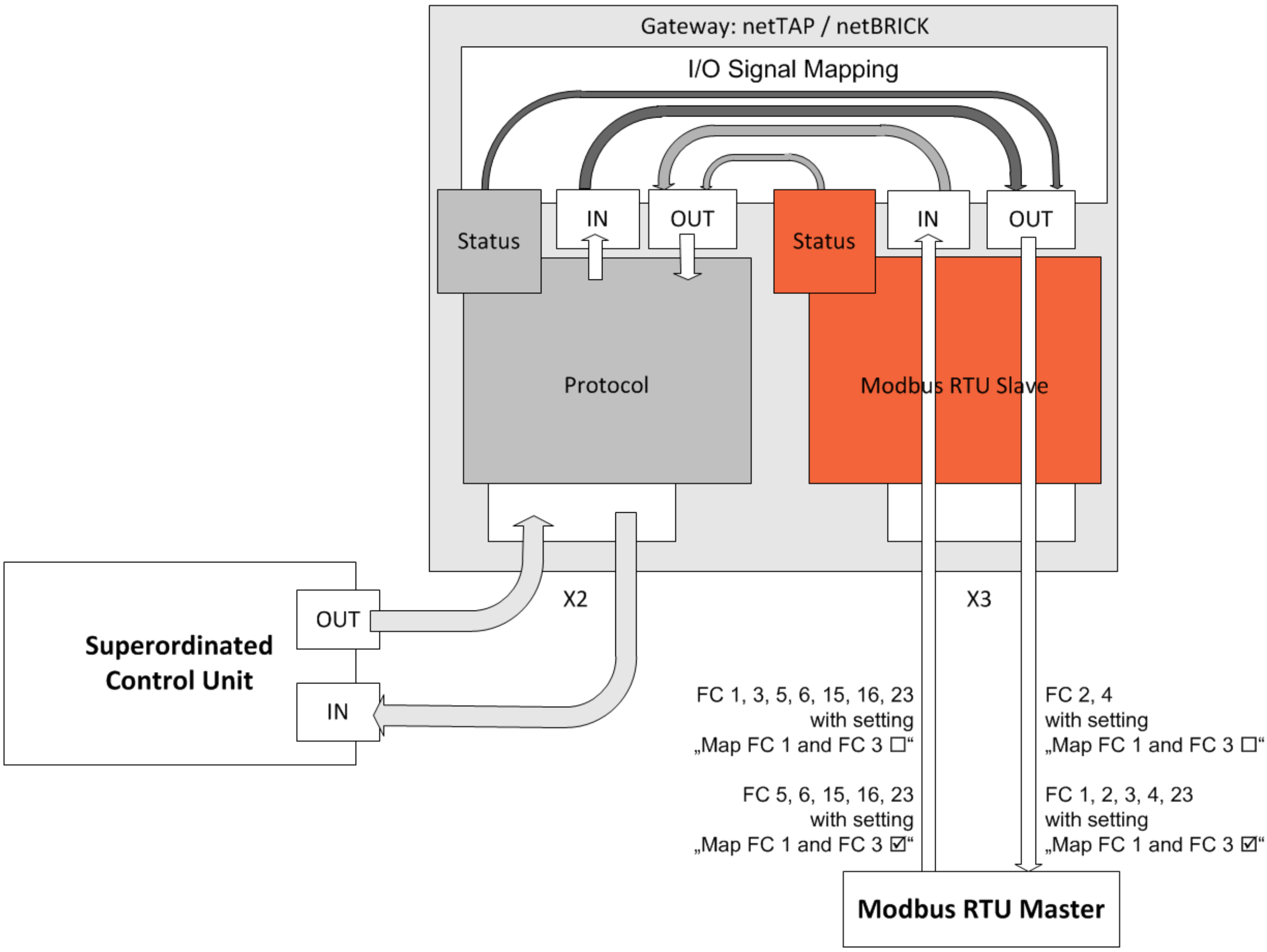


Figure 127: Access from the Modbus RTU Master to the netTAP and netBRICK Device

Registers (16 bit value) and coils (1 bit value) share the same memory for input data (IN). Registers (16 bit value) and coils (1 bit value) share the same memory for output data (OUT). The following tables (Table 55 to Table 60) show the data assignment.

Addressing types of Modbus RTU Master

Modbus RTU Master of different manufacturers use different addressing types. The modicon-based addressing is very popular. Other Modbus RTU Master use a zero-based addressing and again other Modbus RTU Master use one-based addressing.

Note: Read the documentation of the used Modbus RTU Master to find out the used addressing type.

Modicon-based addressing: If function code 3, 6, 16 and 23 is used, the first register (16 bit value) has address 40001, the second register has address 40002 etc. If function code 4 is used, the first register has address 30001, the second register has address 30002 etc. If function code 1, 5 and 15 is used, the first coil (1 bit value) has address 1, the second coil has address 2 etc. If function code 2 is used, the first coil has address 10001, the second coil has address 10002 etc.

The following table shows the address assignment for registers and coils for modicon-based Modbus RTU Master. The setting "**Map FC 1 and FC 3**" is **not set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP or netBRICK (Mapping)	Coil FC 1, FC 5, FC 15	Register FC 3, FC 6, FC 16, FC 23
IN 0	1 … 16	40001
IN 1	17 … 32	40002
IN 2	33 … 48	40003
…	…	…
IN 624	9985 … 9999	40625
…	-	…
IN 2879	-	42880

OUT netTAP or netBRICK (Mapping)	Coil FC 2	Register FC 4
OUT 0	10001 … 10016	30001
OUT 1	10017 … 10032	30002
OUT 2	10033 … 10048	30003
…	…	…
OUT 624	19985 … 19999	30625
…	-	
OUT 2879	-	32880

Table 55: Access from the Modbus RTU Master using Modicon-based Addressing (1)

The following table shows the address assignment for registers and coils for modicon-based Modbus RTU Master. The setting "**Map FC 1 and FC 3**" is **set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP or netBRICK (Mapping)	Coil FC 5, FC 15	Register FC 6, FC 16, FC 23 (write)
IN 0	1 … 16	40001
IN 1	17 … 32	40002
IN 2	33 … 48	40003
…	…	…
IN 624	9985 … 9999	40625
…	-	…
IN 2879	-	42880

OUT netTAP or netBRICK (Mapping)	Coil FC 1	Coil FC 2	Register FC 4	Register FC 3, FC 23 (read)
OUT 0	1 … 16	10001 … 10016	30001	40001
OUT 1	17 … 32	10017 … 10032	30002	40002
OUT 2	33 … 48	10033 … 10048	30003	40003
…	…	…	…	…
OUT 624	9985 … 9999	19985 … 19999	30625	40625
…	-	-	…	…
OUT 2879	-	-	32880	42880

Table 56: Access from the Modbus RTU Master using Modicon-based Addressing (2)

Zero-based addressing: The first register (16 bit value) as well as the first coil (1 bit value) has address 0, the second register as well as the first coil has address 1 etc. independed of the used function code.

The following table shows the address assignment for registers and coils for zero-based Modbus RTU Master. The setting "**Map FC 1 and FC 3**" is **not set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP or netBRICK (Mapping)	Coil FC 1, FC 5, FC 15	Register FC 3, FC 6, FC 16, FC 23
IN 0	0 … 15	0
IN 1	16 … 31	1
IN 2	32 … 47	2
…	…	…
IN 624	9985 … 9998	624
…	-	…
IN 2879	-	2879

OUT netTAP or netBRICK (Mapping)	Coil FC 2	Register FC 4
OUT 0	0 … 15	0
OUT 1	16 … 31	1
OUT 2	32 … 47	2
…	…	…
OUT 624	9985 … 9998	624
…	-	…
OUT 2879	-	2879

Table 57: Access from the Modbus RTU Master using Zero-based Addressing (1)

The following table shows the address assignment for registers and coils for zero-based Modbus RTU Master. The setting "**Map FC 1 and FC 3**" is **set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP or netBRICK (Mapping)	Coil FC 5, FC 15	Register FC 3, FC 6, FC 16, FC 23 (schreiben)
IN 0	0 … 15	0
IN 1	16 … 31	1
IN 2	32 … 47	2
…	…	…
IN 624	9985 … 9998	624
…	-	…
IN 2879	-	2879

OUT netTAP or netBRICK (Mapping)	Coil FC 1, FC 2	Register FC3, FC 4, FC 23 (lesen)
OUT 0	0 … 15	0
OUT 1	16 … 31	1
OUT 2	32 … 47	2
…	…	…
OUT 624	9985 … 9998	624
…	-	…
OUT 2879	-	2879

Table 58: Access from the Modbus RTU Master using Zero-based Addressing (2)

One-based addressing: The first register (16 bit value) as well as the first coil (1 bit value) has address 1, the second register as well as the first coil has address 2 etc. independed of the used function code.

The following table shows the address assignment for registers and coils for one-based Modbus RTU Master. The setting "**Map FC 1 and FC 3**" is **not set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP netBRICK (Mapping)	Coil FC 1, FC 5, FC 15	Register FC 3, FC 6, FC 16, FC 23
IN 0	1 … 16	1
IN 1	17 … 32	2
IN 2	33 … 48	3
…	…	…
IN 624	9986 … 9999	625
…	-	…
IN 2879	-	2880

OUT netTAP netBRICK (Mapping)	Coil FC 2	Register FC 4
OUT 0	1 … 16	1
OUT 1	17 … 32	2
OUT 2	33 … 48	3
…	…	…
OUT 624	9986 … 9999	625
…	-	…
OUT 2879	-	2880

Table 59: Access from the Modbus RTU Master using One-based Addressing (1)

The following table shows the address assignment for registers and coils for one-based Modbus RTU Master. The setting "**Map FC 1 and FC 3**" is **set** in the configuration of the netTAP or netBRICK device in this case.

IN netTAP netBRICK (Mapping)	Coil FC 5, FC 15	Register FC 3, FC 6, FC 16, FC 23 (write)
IN 0	1 … 16	1
IN 1	17 … 32	2
IN 2	33 … 48	3
…	…	…
IN 624	9986 … 9999	625
…	-	…
IN 2879	-	2880

IN netTAP netBRICK (Mapping)	Coil FC 1, FC 2	Register FC3, FC 4, FC 23 (read)
OUT 0	1 … 16	1
OUT 1	17 … 32	2
OUT 2	33 … 48	3
…	…	…
OUT 624	9986 … 9999	625
…	-	…
OUT 2879	-	2880

Table 60: Access from the Modbus RTU Master using One-based Addressing (2)

4.23.4 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

4.24 Configuration 3964R

The NT 100 device respectively the netBRICK NB 100 device as 3964R needs parameter.

These parameter can be edited as follows:

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > 3964R**

➯ The 3964R settings dialog opens.

➯ In the navigation area the entry **Settings** under the folder **Configuration** is selected and the settings dialog is displayed.

2. 3964R settings

➢ Select or set in the dialog window (at least) the interface type, RTS control, baudrate, number of data bits, number of stop bits, parity and conflict priority. The settings are described in the following section *3964R Settings* on page 233.

Note: Please note, that the conflict priority one device must have high priority the other device must have low priority.

3. Signal configuration

➯ The signal configuration is done with default value automatically by SYCON.net software.

4. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➯ The configuration dialog closes

4.24.1 3964R Settings

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)	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

Table 61: 3964R Parameter (Part 1)

Parameter	Meaning	Range of Value/Value
Bus		
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
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		
Max. Input	Maximum telegram length for receive data. Received data are stored in the input data area.	0 … 5736, 1024 (Default)
Max. Output	Maximum telegram length for send data. Send data are taken from the output data area.	0 … 5736, 1024 (Default)

Table 62: 3964R Parameter (Part 2)

4.24.2 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.

4.25 Configuration ASCII

ASCII (= American Standard Code for Information Interchange) is a coding for characters. ASCII is denoted a communication for serial data exchange here and is usually a human readable code and

This type of communication contains however less definitions for the detailed communication procedure. Many manufacturer specific implementations exist.

Basic functions are send and receive of exactly one specified telegram.

The functionality depicted here for send and receive of telegrams can be used when one of the following condition is fulfilled:

- The send telegram has a fixed structure
- The receive telegram has a fixed structure
- The polling telegram (send telegram without data) has a fixed structure and the receive telegram has a fixed structure
- The polling telegram (send telegram with data) has a fixed structure and the receive telegram has a fixed structure

The structure of the serial send and receive telegram is specified by configuration. The part of the telegram marked as user data is transferred unchanged with a superordinated memory. For this data exchange a handshake mechanism has to be used.

ASCII should not be used, when

- different telegram structures for send telegrams or
- different telegram structures for receive telegrams or
- communication procedures with more complexity are to be used or
- the user data needs to be evaluated, processed respectively converted

Then use 'Serial with netSCRIPT'. With 'Serial with netSCRIPT' it can be reached by programming (script)

- that user data is evaluated, processed respectively converted,
- more complex communication procedures and,
- different telegram structures

can be realized.

The netTAP NT 50 and NT 100 device respectively the netBRICK NB 100 device as ASCII needs parameter.

These parameter can be edited as follows:

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > ASCII**

➩ The ASCII settings dialog opens.

➩ In the navigation area the entry **Settings** under the folder **Configuration** is selected and the settings dialog is displayed.

2. ASCII settings

➢ Select in the dialog window the interface type, RTS control, baudrate, number of data bits, number of stop bits and parity. The settings are described in the following section *ASCII Settings* on page 238.

3. Open ASCII parameter dialog

➢ Select in the navigation area under **Configuration** the entry **ASCII Parameter**.

➩ The dialog **ASCII Parameter** is displayed.

4. Set parameter

➢ Select the operating mode, set the telegram structure, set the timing and the size of the send and receive buffer. The parameter are described in section *ASCII Parameters* on page 238.

5. Signal configuration

➩ The signal configuration is done with default value automatically by SYCON.net software.

6. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➩ The configuration dialog closes

4.25.1 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 63: ASCII Parameter

4.25.2 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

4.25.2.1 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]

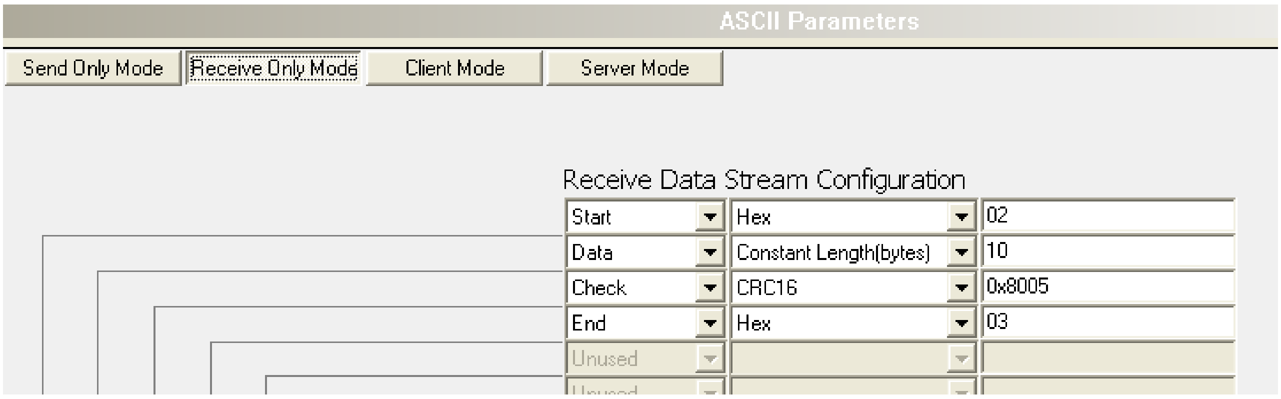


Table 64: ASCII Telegram Structure Example

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
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)

Parameter	Description	Range of Value / Parameter Type
	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.	

Table 65: 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 66: 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	Charac-ter	ASCII Hex	ASCII Dez	Charac-ter	ASCII Hex	ASCII Dez	Charac-ter	ASCII Hex	ASCII Dez	Cha-racter
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 67: ASCII Character Table

4.25.2.2 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 68: Size of the Send and Receive Buffer

4.25.2.3 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)

4.25.2.4 '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 69: Timing Parameters for the Operating 'Send Only' Mode

Configuration Example

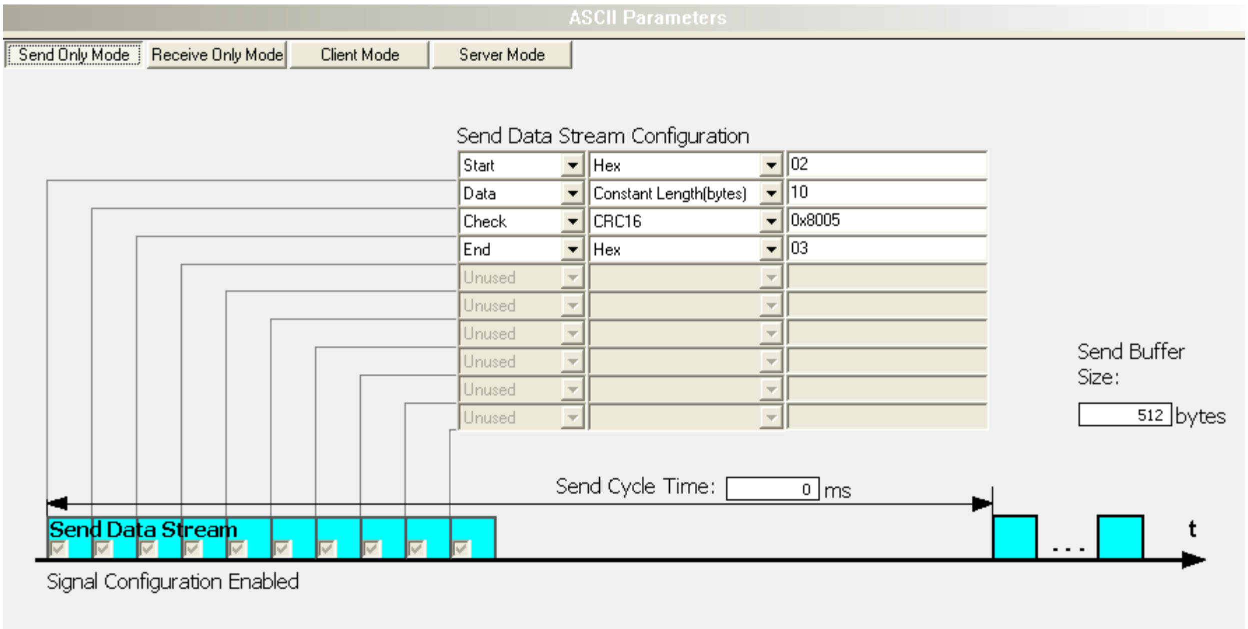
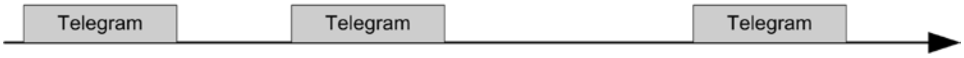


Table 70: Telegram Structure for the Operating 'Send Only' Mode

4.25.2.5 '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.

<u>Communication</u>

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.

<u>Timing Parameter</u>

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 … 231-1 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 … 231-1 Default: 0

Table 71: Timing Parameters for the Operating ‘Receive Only’ Mode

Configuration Example

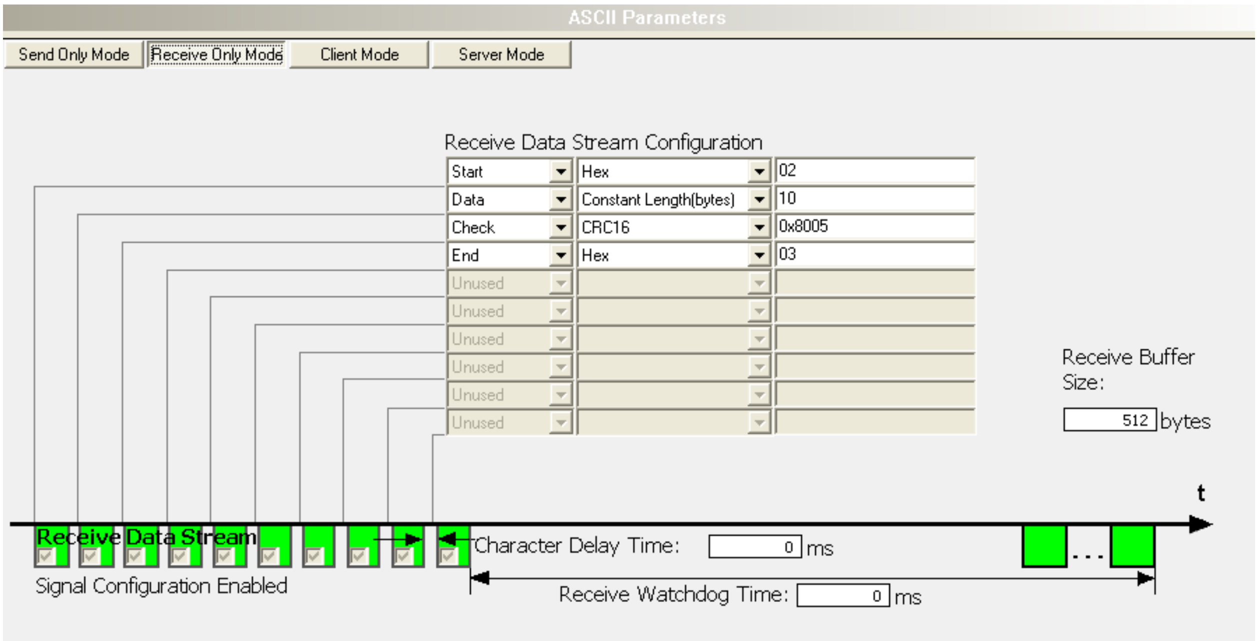


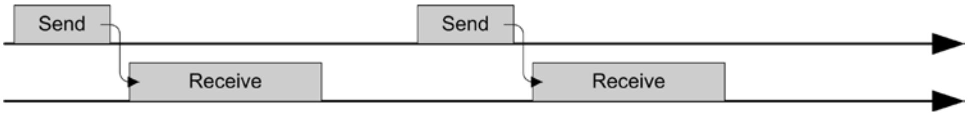
Table 72: Telegram Structure for the Operating 'Receive Only' Mode

4.25.2.6 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 73: Timing Parameters for the Operating Mode 'Client Mode'

Configuration Example

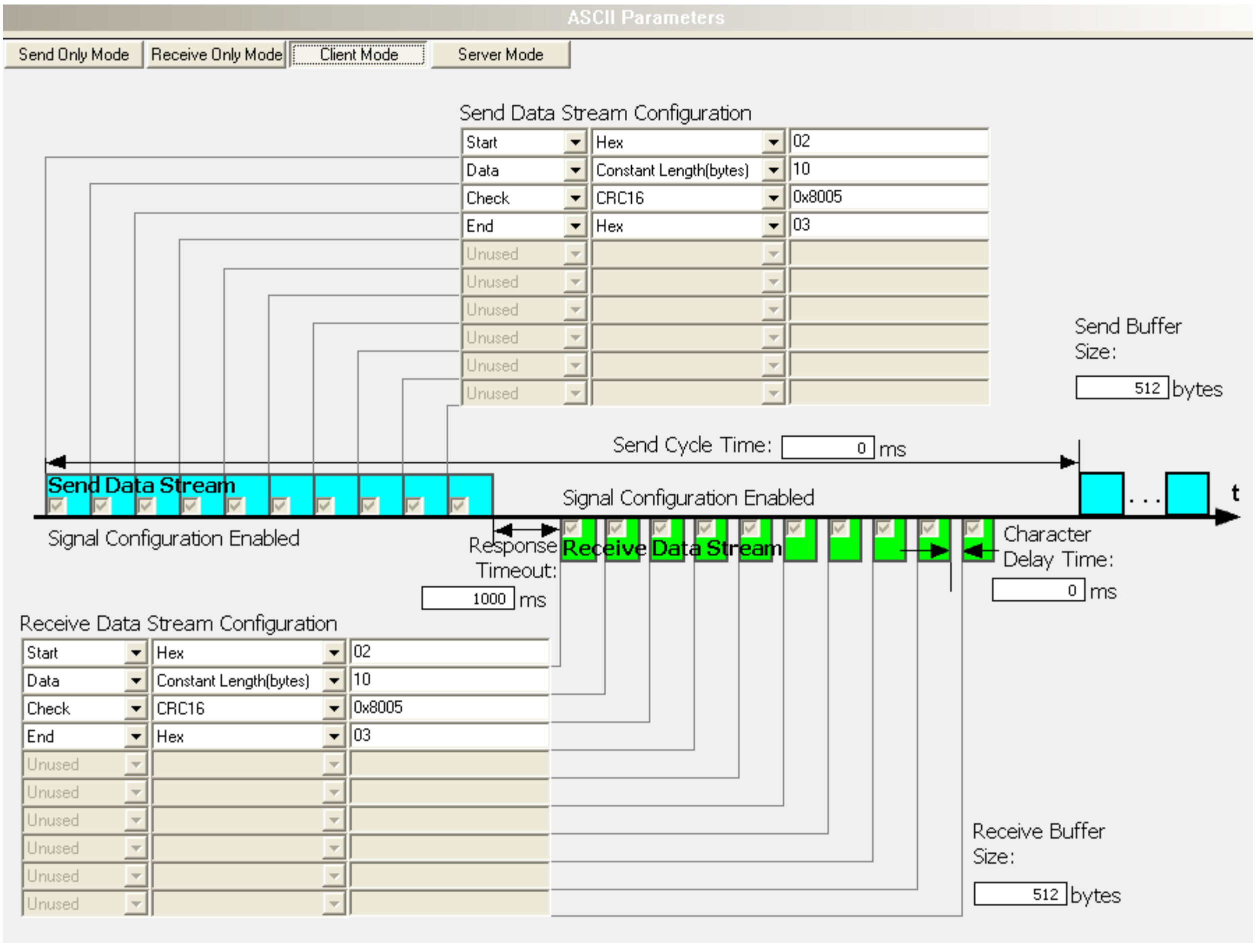
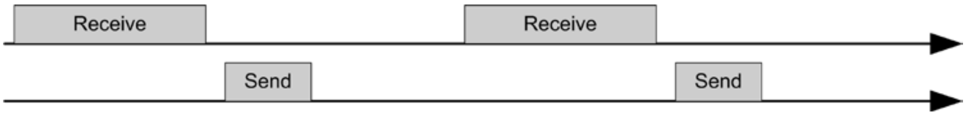


Table 74: Telegram Structure for the Operating Mode 'Client Mode'

4.25.2.7 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 75: Timing Parameters for the Operating Mode 'Server Mode'

Configuration Example

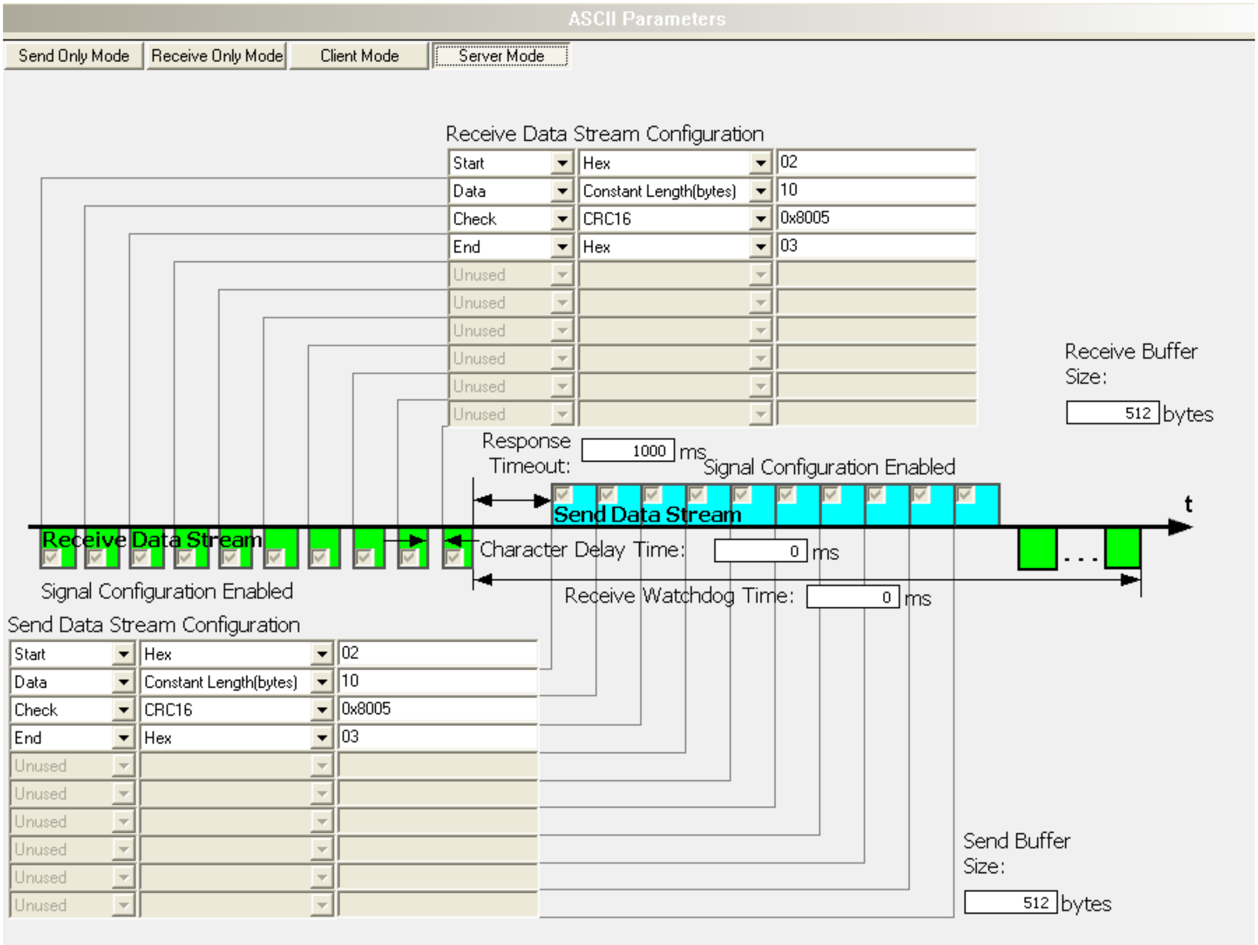


Table 76: Telegram Structure Parameters for the Operating Mode 'Server Mode'

4.25.3 Settings for the remote ASCII 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 check and respectively RTS control.

4.26 Configuration 'Serial with netSCRIPT'

The netTAP NT 100 device respectively the netBRICK NB 100 device as 'serial with netSCRIPT' needs parameter and beyond a script has to be created and loaded.

These parameter can be edited as follows:

1. Open the configuration dialog

➢ Select from the context menu of the netTAP respectively the netBRICK symbol the entry **Configuration > netSCRIPT**

➯ The netSCRIPT settings dialog opens.

➯ In the navigation area the entry **Settings** under the folder **Configuration** is selected and the settings dialog is displayed.

2. netSCRIPT settings

➢ Select in the dialog window the interface type, RTS control, baudrate, number of data bits, number of stop bits and parity. The settings are described in the following section *netSCRIPT Settings* on page 254

3. Signal configuration

➯ The signal configuration is done with default value automatically by SYCON.net software.

4. Close configuration dialog

➢ Click on **OK** to save the parameter or click on **Cancel**, if the parameter should not be saved.

➯ The configuration dialog closes

Note: The programming of a script, loading the script into the netTAP or netBRICK device, the debugging of the script is described in detail in an own manual: netSCRIPT Programming Language for Serial Communication UM xx EN.pdf

4.26.1 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 77: netSCRIPT Settings

4.26.2 Settings for the netSCRIPT Remote Device

Note: The settings in the used netSCRIPT 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 check and respectively RTS control.

4.27 Signal Configuration

At the pane **Signal Configuration** the data structure and the signal name of the in- and output signals is displayed. These signal are used for the signal mapping. Here for the single signals respectively the names of the signal names are set or the data types are configured.

SYCON.net generates for the signal names default names. These names can be overwritten by the user.

Parameter	Meaning	Range of Value/ Value
Type	Type of the input or output signals	String
Tag	Signal name for the single input or output signals. The name can be set by the user.	String
Data type	Data type of the single input or output signals. Depending by the used AS-Interface Slave profile the user can select the data type from a list.	BIT, WORD, SIGNED16, UNSIGNED16, etc., Default: depends on the used protocol
IO Type	Input or output signal	input, output

Table 78: Explanations to the Dialog Pane Signal Configuration

5 Diagnosis

5.1 Overview Diagnosis

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

The **Extended Diagnosis** helps to find communication and configuration errors, when default diagnosis fails.

Diagnosis Panes

The table below gives an overview for the individual **Diagnosis** dialog panes descriptions:

Section	Page
General Diagnosis	*257*
Firmware Diagnosis	*259*

Table 79: Descriptions of the Diagnosis Panes

Note: Accessing the **Diagnosis** panes of the netGateway DTM requires an online connection from the netGateway DTM to the netTAP NT 100 device.

For further information, refer to section *Connecting/Disconnecting Device* on page 260.

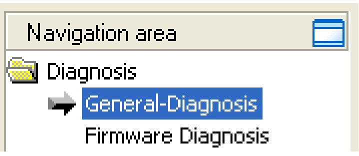


Figure 128: Navigation Area - Diagnosis

5.2 General Diagnosis

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

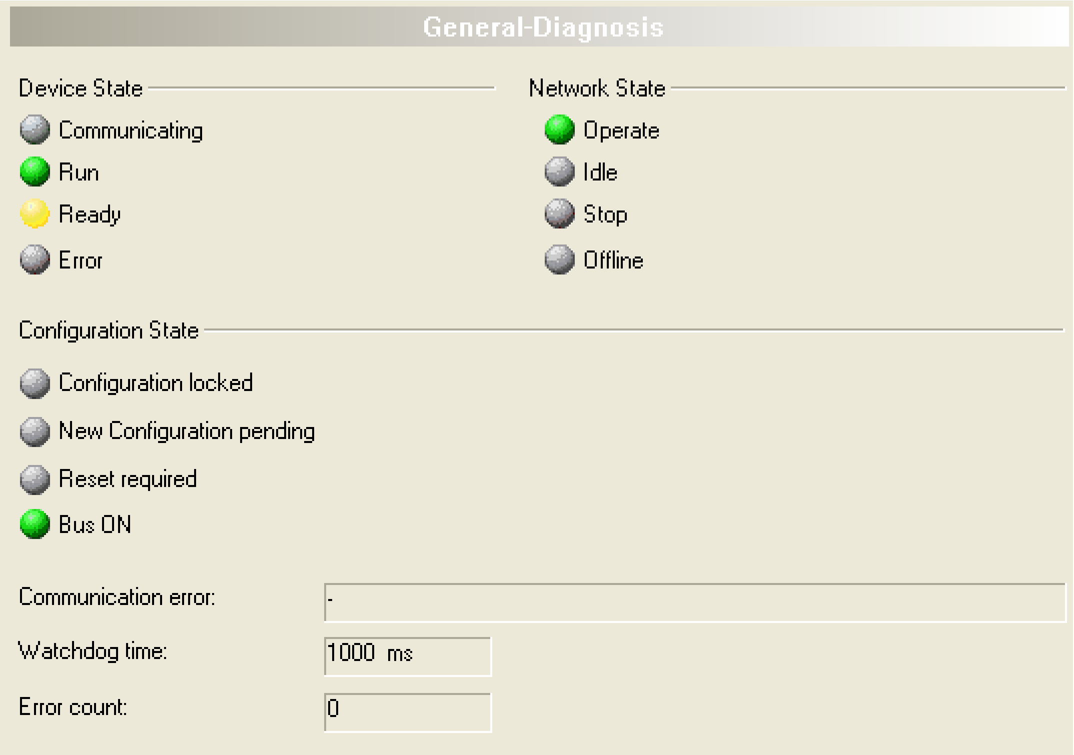


Figure 129: General Diagnosis

Indication	Meaning
Device State	
Communicating	Communicating: Shows that the netTAP firmware executes the network communication.
Run	Device is configured: Shows that the netTAP firmware has been configured correctly.
Ready	Ready: Shows that the netTAP firmware has been started correctly. The netTAP firmware waits for a configuration.
Error	Error: Shows that the netTAP firmware records a device status error. For further information to the error characteristics and the number of counted errors refer to the extended diagnosis.
Network State	
Operate	Operate: Shows that the netTAP firmware is in data exchange.
Idle	Idle: Shows that the netTAP firmware is in idle mode.
Stop	Stop: Shows that the netTAP firmware is in Stop state: There is no cyclic data exchange at the network. The netTAP firmware was stopped by the application program or it changed to the Stop state because of a bus error.
Offline	Offline: The netTAP firmware is offline pending it does not have a valid configuration.
Configuration State	
Configuration locked	Configuration locked: Shows that the netTAP firmware configuration is locked, to avoid the configuration data are typed over.
New Configura-tion pending	New Configuration pending: Shows that a new netTAP firmware configuration is available.
Reset required	Reset required: Shows that a firmware reset is required as a new netTAP firmware configuration has been loaded into the device.
Bus ON	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 telegrams are sent.

Table 80: Indication General Diagnosis

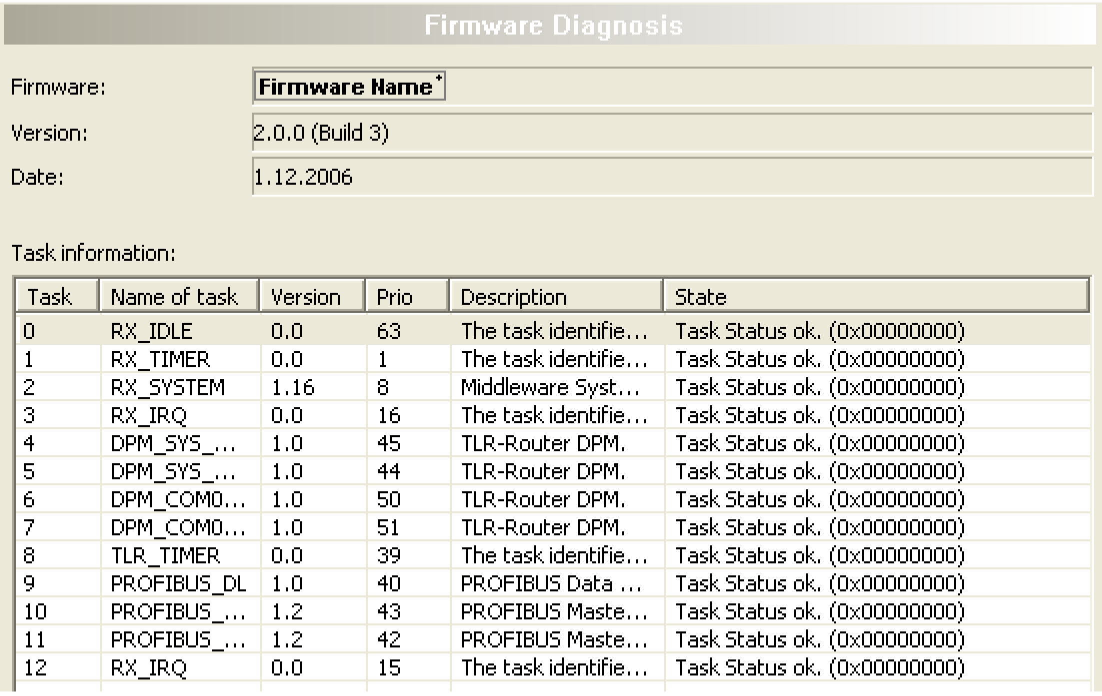
Parameter	Meaning
Communication Error	Communication Error: Shows the message text of the communication error. If the cause of the current error is resolved, „ – “ is displayed.
Watchdog time	Watchdog time: Shows the watchdog time in ms.
Error Count	Error Count: This field holds the total number of errors detected since power-up, respectively after reset. The protocol stack counts all sorts of errors in this field no matter if they were network related or caused internally.

Table 81: Parameter General Diagnosis

5.3 Firmware Diagnosis

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

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



Task	Name of task	Version	Prio	Description	State
0	RX_IDLE	0.0	63	The task identifie...	Task Status ok. (0x00000000)
1	RX_TIMER	0.0	1	The task identifie...	Task Status ok. (0x00000000)
2	RX_SYSTEM	1.16	8	Middleware Syst...	Task Status ok. (0x00000000)
3	RX_IRQ	0.0	16	The task identifie...	Task Status ok. (0x00000000)
4	DPM_SYS_...	1.0	45	TLR-Router DPM.	Task Status ok. (0x00000000)
5	DPM_SYS_...	1.0	44	TLR-Router DPM.	Task Status ok. (0x00000000)
6	DPM_COM0...	1.0	50	TLR-Router DPM.	Task Status ok. (0x00000000)
7	DPM_COM0...	1.0	51	TLR-Router DPM.	Task Status ok. (0x00000000)
8	TLR_TIMER	0.0	39	The task identifie...	Task Status ok. (0x00000000)
9	PROFIBUS_DL	1.0	40	PROFIBUS Data ...	Task Status ok. (0x00000000)
10	PROFIBUS_...	1.2	43	PROFIBUS Maste...	Task Status ok. (0x00000000)
11	PROFIBUS_...	1.2	42	PROFIBUS Maste...	Task Status ok. (0x00000000)
12	RX_IRQ	0.0	15	The task identifie...	Task Status ok. (0x00000000)

*Figure 130: Firmware Diagnosis (*The name of the Firmware is displayed.)*

Task Information:

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

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

Table 82: Description Table Task Information

6 Online Functions

6.1 Connecting/Disconnecting Device

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

Connecting Device

The following steps are needed to establish a connection from the netGateway DTM to a netTAP respectively netBRICK device:

Under **Settings** in the **Driver** pane:

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** pane:

1. Scan for the device.
2. Select the device and apply the selection.
3. In the DTM interface dialog select the **OK** button, to apply the selection and to close the DTM interface dialog.
4. Put a right-click on the netTAP respectively netBRICK symbol.
5. Select the **Connect** command from the right mouse button menu.

➾ The netTAP respectively netBRICK device now is connected to the netGateway DTM via an online connection. In the network view the device description at the device symbol is displayed with a green colored background.

Disconnecting Device

To disconnect an online connection from the netTAP respectively netBRICK device to a netGateway DTM take the following steps:

3. In the DTM interface dialog select the **OK** button, to close the DTM interface dialog.
1. Right-click on the netTAP respectively netBRICK symbol.
2. Select the **Disconnect** command from the context menu.

➾ Now the netTAP respectively netBRICK device is disconnected from the DTM.

6.2 Download Configuration

Note: To download configuration parameter data to the netTAP/netBRICK/netLINK device an online connection from the netGateway DTM to the device is required.

Further information can be found in the *Connecting/Disconnecting Device* section on page 260.

To transfer the configuration with the corresponding parameter data to the netTAP/netBRICK/netLINK device you must download the data to it using the frame application of the configuration software.

7 Acyclic Communication of Proxy Devices

7.1 Overview

This chapter relates to the following devices:

Device	Name in SYCON.net device catalog	Protocol conversion / firmware
NL 51N-DPL	NL 51N-DPL	PROFINET IO Device to PROFIBUS Master Link
NT 100-RE-DP	NT 100-RE-XX/PROXY	PROFINET IO Device to PROFIBUS Master

Table 83: Proxy Devices with Acyclic Communication

7.2 Adress Mapping

The read/write record service of the PROFINET IO controller are converted to DPV1 read/write services to the PROFIBUS DP slave for acyclic communication. Thereby the PROFINET IO Controller can access the PROFIUS DP slaves via the proxy device an address mapping exsists.

The following figure and table describes the mapping of the PROFINET IO slot, subslot and index to PROFIBUS station address, slot and index.

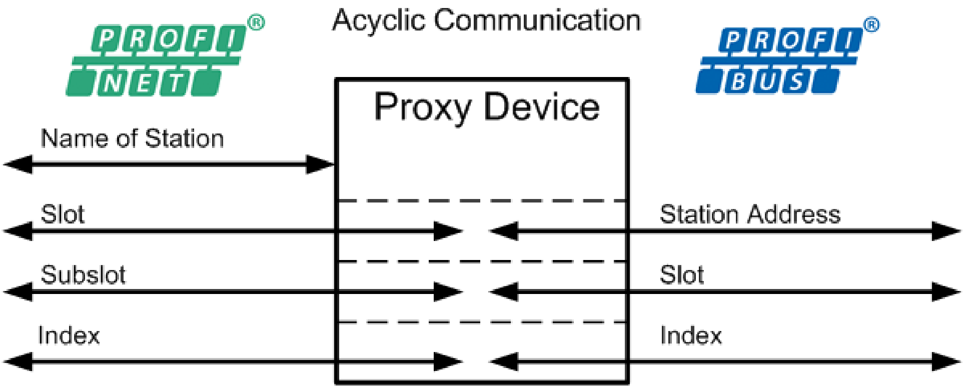


Figure 131: PROFINET – PROFIBUS Address Mapping

The proxy device is addressed by the PROFINET name of station. This makes is possible to address several proxy devices on the PROFINET network.

The following table shows details of the address mapping.

PROFINET			PROFIBUS	
Slot	**SubSlot**	**Index**	**Mapped PROFIBUS Detail**	**Remark**
0	X	X	DeviceSlot / Device Module	Not used
1	0	0x000-0x00FF	PB Slave Device 1 Slave Record Data Slot 0	Used
		0x0100	PB Slave Device 1 Parameter Data	Not used
		0x0101	PB Slave Device 1 Configuration Data	Not used
		0x0102	PB Slave Device 1 Read Input Record Data	Used
		0x0103	PB Slave Device 1 Read Output Record Data	Used
		0x0104-0xFFFF		Not used
	1-254	0x000-0x00FF	PB Slave Device 1 Slave Record Data Slot 1..254	Used
		0x0100-0xFFFF		Not used
	255-	X		Not used
2	0-254	See Slot 1	PB Slave Device 2	Used
And so on with the same mapping for all following slaves from 3 till 125				
126	0-254	See Slot 1	PB Slave Device 126	used
127	x	X		Not used
128	0-254	See Slot 1	PB Slave Device 0	Used

Table 84: PROFINET IO - PROFIBUS; Address Mapping of acyclic Data

7.3 Read/Write – Controlled by the PROFINET IO Controller

PROFINET IO read record is converted to PROFIBUS DPV1 read. The data of the PROFIBUS DP Slaves are forwarded unchanged to the PROFINET IO Controller.

PROFINET IO write record is converted to PROFIBUS DPV1 write. The data to be written are forwarded unchanged to the PROFIBUS DP slave.

A positive confirmation to the PROFINET IO Controller means, that the PROFIBUS DP Master in the proxy device has received a positive confirmation from the PROFIBUS DP slave.

Error Detection and Error Codes

The following figure shows where an error detection is possible.

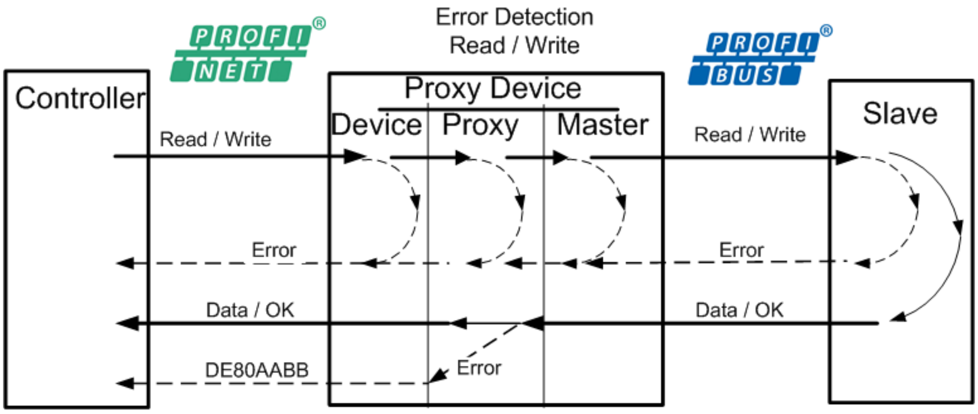


Figure 132: PROFINET IO – PROFIBUS DP Read/Write Error Detection

Errors detected in the proxy device (PROFINET IO device), are reported as normal PROFINET IO errors to the controller.
Errors detected in the proxy device, which are not addressing errors, are reported with error code 0xDE80AABB to the PROFINET IO controller.

All errors, except addressing errors, which are detected on PROFIBUS (master or slave) are reported with error code 0xDE80AABB to the PROFINET IO controller.

Error detection by the proxy device for read/write record:

The following error codes can occur:

Error Code	Definition, Descrption
0xDE80B000	DE = IODReadRes; negative response of a read request 80 = PNIORW; User Error B0 = invalid index invalid index for a read request.
0xDE80B200	DE = IODReadRes; negative response of a read request 80 = PNIORW; User Error B2 = invalid slot / sub slot (invalid slave address) invalid slot or sub slot
0xDF80B200	Invalid slave address for a write request.
0xDE80AABB	DE = IODReadRes; negative response of a read request 80 = PNIORW; User Error AA = user spezific BB = user spezific All errors (except addressing errors) detected in the proxy device (PROFIBUS Master or PROFIBUS Slave.
0xDF80AABB	DF = IODReadRes; negative response of a write request 80 = PNIORW; User Error AA = user spezific BB = user spezific All errors (except addressing errors) detected in the proxy device (PROFIBUS Master or PROFIBUS Slave.

Table 85: PROFINET IO – PROFIBUS DP; Error Codes

7.4 Alarms – Controlled by the PROFIBUS DP Slave

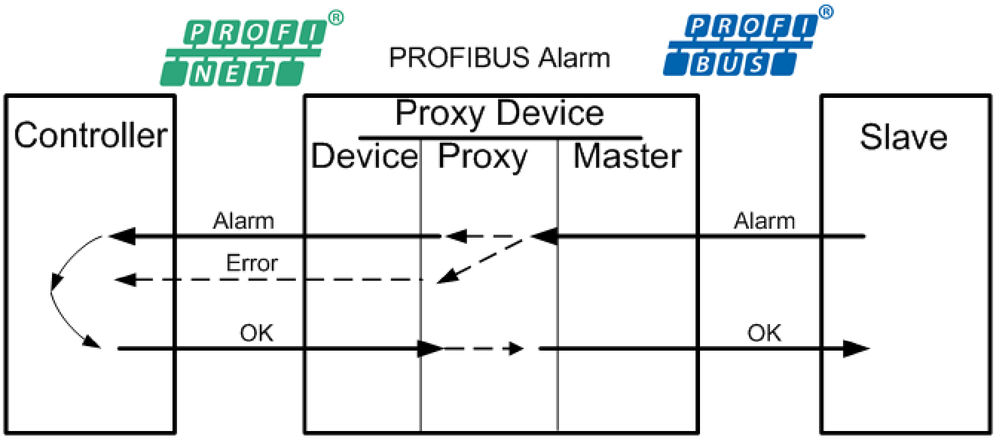


Figure 133: PROFINET IO- PROFIBUS-DP – Alarm Errors

PROFIBUS alarms are mapped to PROFINET alarms as follows:

PROFIBUS	PROFINET IO
Process alarm	Process alarm
Diagnosstic alarm	Diagnosstic alarm
Status alarm	Diagnosstic alarm with the structure of the PROFIBUS alarm
Update alarm	Diagnosstic alarm with the structure of the PROFIBUS alarm
Pull alarm	Pull alarm
Plug alarm	Plug alarm
Manufacturer specific alarm	Diagnosstic alarm with the structure of the PROFIBUS alarm

Table 86: PROFINET IO - PRPFIBUS; Mapping of Alarms

Alarm data are forwared unchanged.

Extended PROFIBUS diagnostic are mapped to PROFINET IO diagnostic with UserStructID 0x2000+x.

PROFIBUS diagnostic alarms are mapped to PROFINET IO diagnostic with UserStructID 0x4000+x.

PROFIBUS DPV1 parameter services are not transfered to PROFINET IO and via versa.

Note: Had a PROFIBUS DP slave device generated a pull alarm once and this device/module goes into communication again, then a manual reset to the proxy device is necessary (by a short disconnect of the power supply by example) to force the prxy device to reload the configuration data.

Was the PROFIBUS DP slave device removed only from the PROFIBUS, without a generation of a pull alarm, then the communication start automatically if the PROFIBUS DP slave device is reconnected to PROFIBUS.

Note: If a PROFIBUS DP slave device generates a plug alarm (a new device/module was added to PROFIBUS), then the proxy device has to be reconfigured with new configuration data by the user.

8 Acyclic Communication of Gateway Devices

8.1 Overview

This chapter relates to the following devices:

Device	Name in SYCON.net device catalog	Protocol conversion / firmware
NT 100-RE-EN	NT 100-XX-XX	PROFINET IO Device to EtherNet/IP Scanner

Table 87: Acyclic Communication of Gateway Devices

Besides cyclic data exchange, the **netTAP NT 100** gateway also supports acyclic communication for some protocol conversions.

In **cyclic communication**, small packages of I/O data ("process data") are exchanged at regular intervals with high-priority; in **acyclic communication** on the other hand, (in most cases) bigger data packages are being sent on special occasions, such as configuration download or slave diagnosis.

The figure below shows the handling of cyclic and acyclic data flow via gateway:

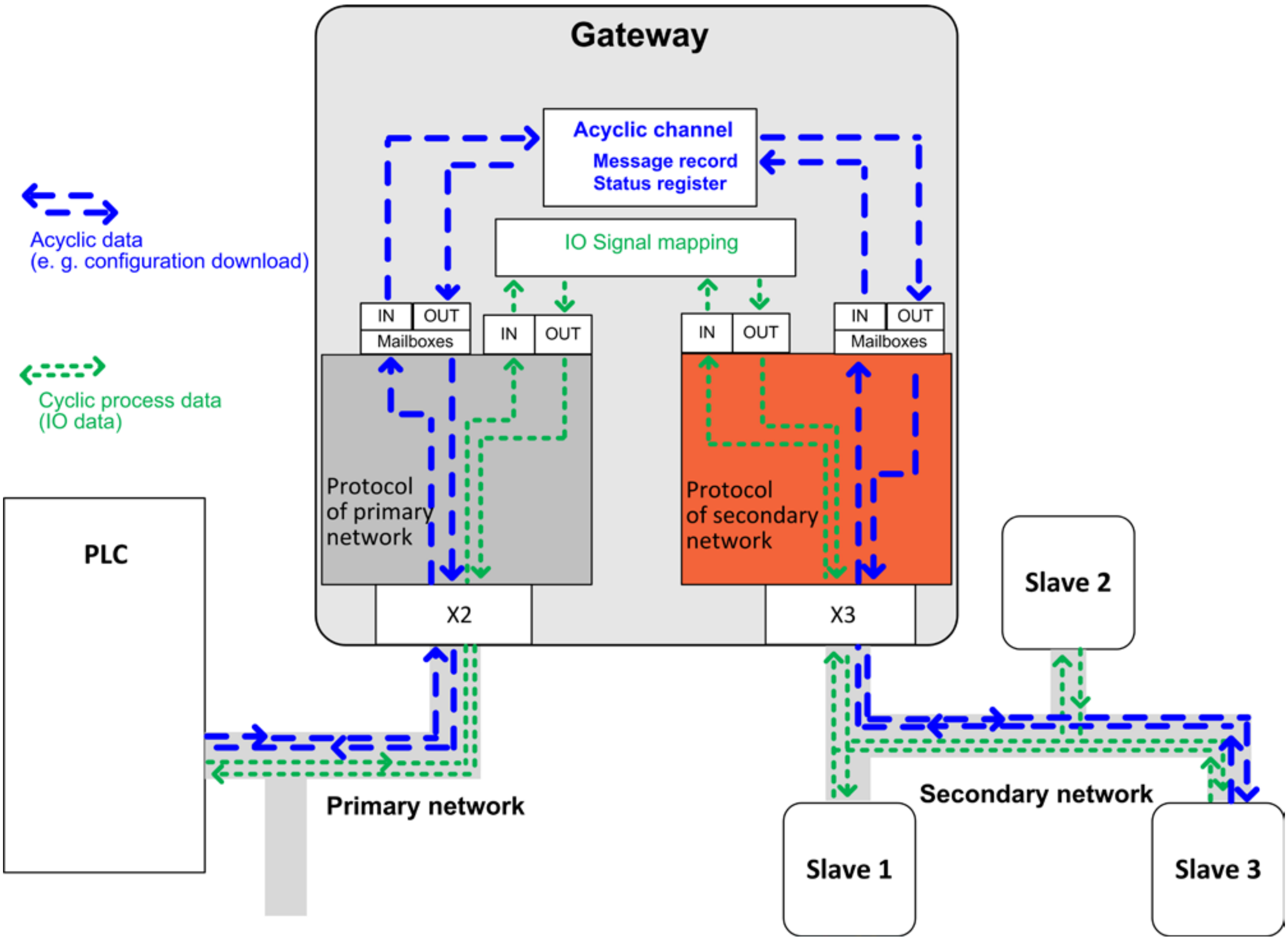


Figure 134: Acyclic Gateway Communication

In most Master/Slave applications, an acyclic data service is initiated by the PLC of the primary network. The acyclic message is being transposed and then forwarded to the secondary network by the gateway. After the service has been processed by the secondary network, the returned result is stored in the gateway, and the PLC can read the result from there.

Because the acyclic services of the different automation protocols are based on different record addressing models, and may also have different timing requirements, they cannot be converted directly from one protocol into another. The conversion therefore uses read- and writeable objects within the gateway controlling the acyclic data transfer of the secondary protocol (for information about the objects valid for your protocol, see corresponding section *Gateway with Protocol Conversion […]* below in this chapter).

The acyclic task, which the PLC writes into the acyclic object in the gateway, contains routing information, i. e. the address of the slave in the secondary network, the service type (read/write) and the target path within the slave device. In case of a writing task, the message also contains the payload data to be written into the slave.

The routing information unambiguously identifies the target device of the acyclic task for the secondary protocol, thus enabling the gateway to forward the task to the secondary protocol. The gateway then waits for the response and writes the result (which might also contain payload data read from the target device) to the object. From there it can be read by the initiator (i. e. the PLC) of the acyclic service.

The gateway stores each task per object as a so-called **Message Record**. For each Message Record, the processing state of the task (**Request State**) is also recorded in a **Status Register** linked to the Message Record. Message Record and Status Register together form an entity which is called **Acyclic Channel** in this document. An acyclic channel always occupies two objects in the gateway. Depending on the protocol conversion, several acyclic channels are supported and can be addressed and activated for services simultaneously.

8.2 Request States

The values of the **Request States** recorded in the **Status Register** are valid for all protocol conversions and are as follows:

Value	Request State	Description
0	MSG_STATE_INIT	The Acyclic Channel has just been initialized and no request has been made yet
1	MSG_STATE_IDLE	The Acyclic Channel has completed previous request and is ready to process next one
2	MSG_STATE_REQ_SENT	The PLC has written a request to be forwarded through this channel. Any following writes to the Message Record will be ignored, because the channel can handle only one request at a time
3	MSG_STATE_RSP_REQUESTED	The PLC has sent a read request of the Message Record and the channel is waiting for secondary network reply
4	MSG_STATE_RSP_RECEIVED	The secondary network has replied to the acyclic request and the channel is waiting for the PLC to read the Message Record

Table 88: Request States in Acyclic Communication

8.3 Sequence of a Reading Task

This section describes the typical sequence of events of an error-free **acyclic reading task** for a slave device in the secondary network (see also diagram on following page):

1. The PLC sends the task to the gateway, where it is stored in the **Message Record** of the acyclic channel and forwarded to the slave in the secondary network.
2. The gateway sets the **Request State** in the **Status Register** of the acyclic channel to value 2 and sends back an acknowledge message to the PLC.
   The slave executes the requested reading task.
3. The PLC queries the state of the request at regular intervals (“status polling”) by reading the **Status Register** of the acyclic channel in the gateway. As long as the gateway has not yet received the answer from the slave, the **Status Register** is returning **Request State** 2.

Note: Instead of polling the **Status Register**, the PLC can read the **Message Record** directly, expecting that the answer from the slave has already arrived at the gateway and can be read from there. If the answer from the slave has not yet arrived at the gateway at the time of the PLC reading the **Message Record**, the **Request State** in the **Status Register** is set to value 3 and the reply from the gateway to the PLC is stalled until the answer from the slave arrives.

4. The answer from the slave arrives at the gateway where it is stored in the **Message Record** of the acyclic channel. The gateway sets the **Request State** in the **Status Register** to value 4.
5. The PLC reads the **Status Register** again and the gateway returns **Request State** 4 to the PLC. The PLC now knows that the answer from the slave has arrived and that the requested data can be read from the **Message Record**.
6. The PLC now reads the requested data from the **Message Record** of the acyclic channel. In case the slave reported an error in step 4, the PLC can now react to the error message.
7. The gateway resets the **Request State** to value 1. The acyclic channel is now ready to process a new request.

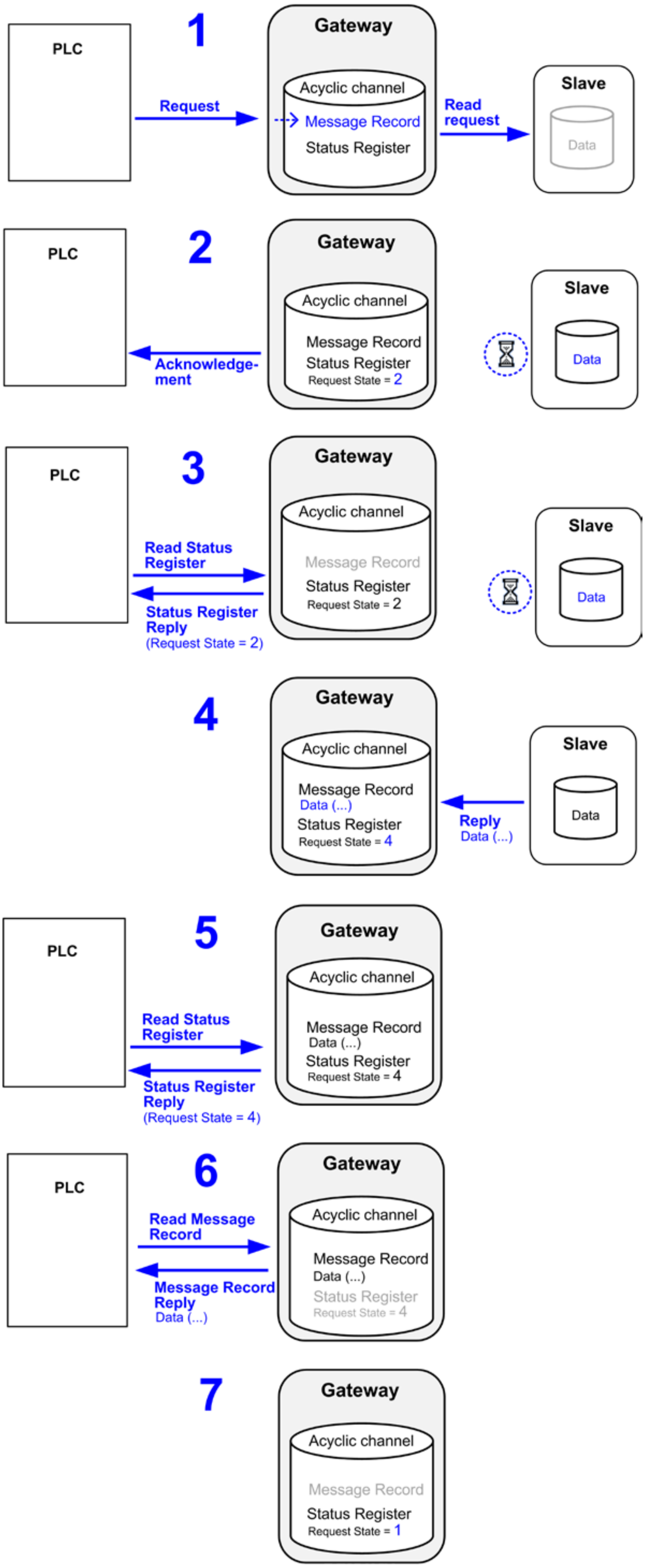


Figure 135: Sequence of Acyclic Reading Task

8.4 Sequence of a Writing Task

This section describes the typical sequence of events of an error-free **acyclic write request** for a slave device in the secondary network (see also diagram on following page):

1 The PLC sends the task to the gateway, where it is stored in the **Message Record** of the acyclic channel and forwarded to the slave in the secondary network.

2 The gateway sets the **Request State** in the **Status Register** of the acyclic channel to value 2 and sends back an acknowledge message to the PLC.
The slave executes the requested writing task.

3 The PLC queries the state of the request at regular intervals (“status polling”) by reading the **Status Register** of the acyclic channel in the gateway. As long as the gateway has not yet received a response from the slave, the **Status Register** is returning **Request State** 2.

Note: Instead of polling the **Status Register**, the PLC can read the **Message Record** directly, expecting that the response from the slave has already arrived at the gateway and can be read from there. If the response from the slave has not yet arrived at the gateway at the time of the PLC reading the **Message Record**, the **Request State** in the **Status Register** is set to value 3 and the reply from the gateway to the PLC is stalled until the response from the slave arrives.

4 The response from the slave arrives at the gateway where it is stored in the **Message Record** of the acyclic channel. The gateway sets the **Request State** in the **Status Register** to value 4.

5 The PLC reads the **Status Register** again and the gateway returns **Request State** 4 to the PLC. The PLC now knows that the response from the slave has arrived and can be read from the **Message Record**.

6 The PLC now reads the response from the **Message Record** of the acyclic channel. In case the slave reported an error in step 4, the PLC can now react to the error message.

7 The gateway resets the **Request State** to value 1. The acyclic channel is now ready to process a new request.

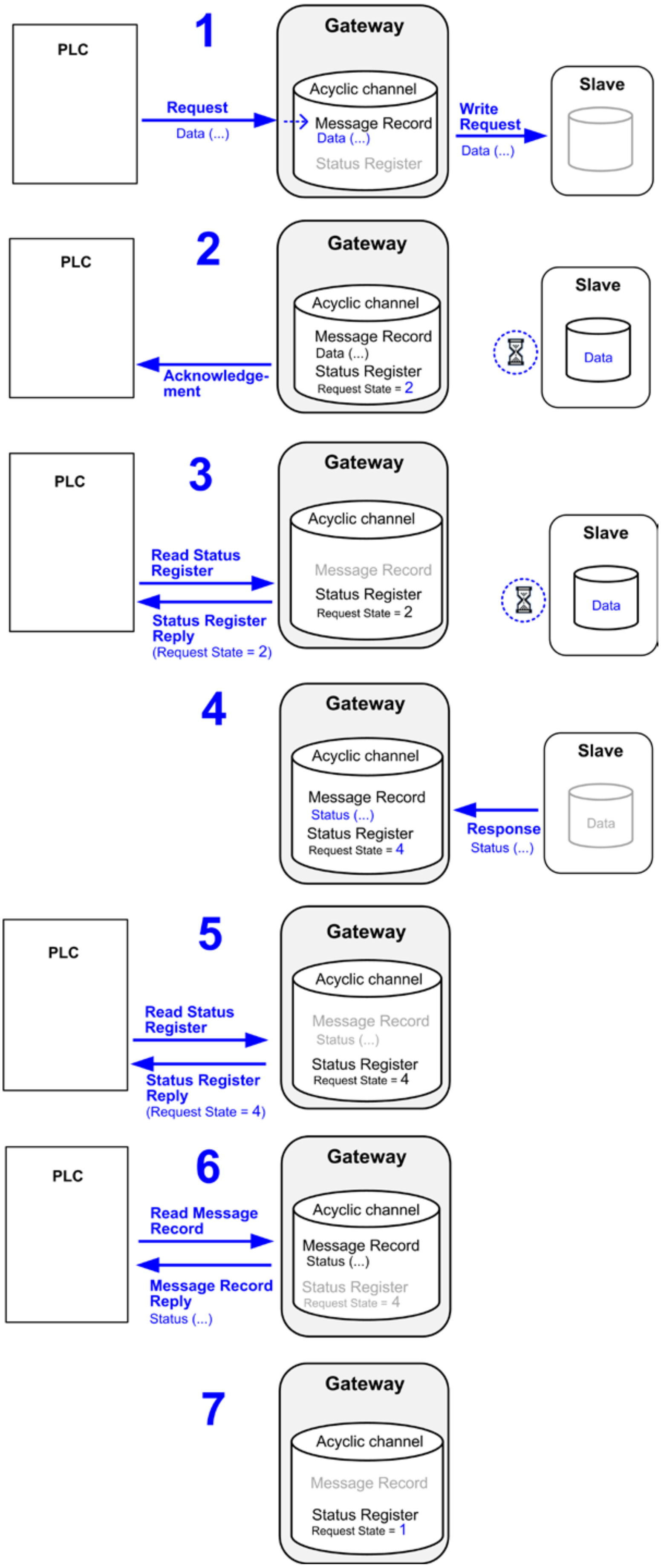


Figure 136: Sequence Acyclic Writing Task

8.5 Gateway with Protocol Conversion PROFINET IO-Device to EtherNet/IP Scanner

8.5.1 Overview

The PROFINET standard prescribes a record addressing model using **Slot**, **Subslot** and **Index** for read and write services. The data content of these so-called **Records** can be accessed by **Read/Write Data Record** services. The objects of an **acyclic channel** in the gateway must be addressed via these records.

Slot 0, **Subslot 1** in the gateway is reserved for the **Acyclic Channels**. The value of the **Index** specifies an individual acyclic channel. Two consecutive **Indices** together form the object pair of one acyclic channel (e. g. `1/2`, `3/4`, `1001/1002` etc.). The **Index** with the **uneven number** stores the **Message Record** (i. e. routing information, service type and payload data), the following **Index** with an **even number** denotes the **Status Register** recording the **Request State** of the corresponding acyclic task (see table *Request States in Acyclic Communication* on page 268).

Up to 255 acyclic channels (and thus parallel services) can be handled simultaneously by the gateway. The values of the index pairs intended for establishing an acyclic channel can be freely chosen within the range of `1` to `65534` (e. g. `1/2`, `3/4…65533/65534` are possible, index `0` and `65535` are not available).

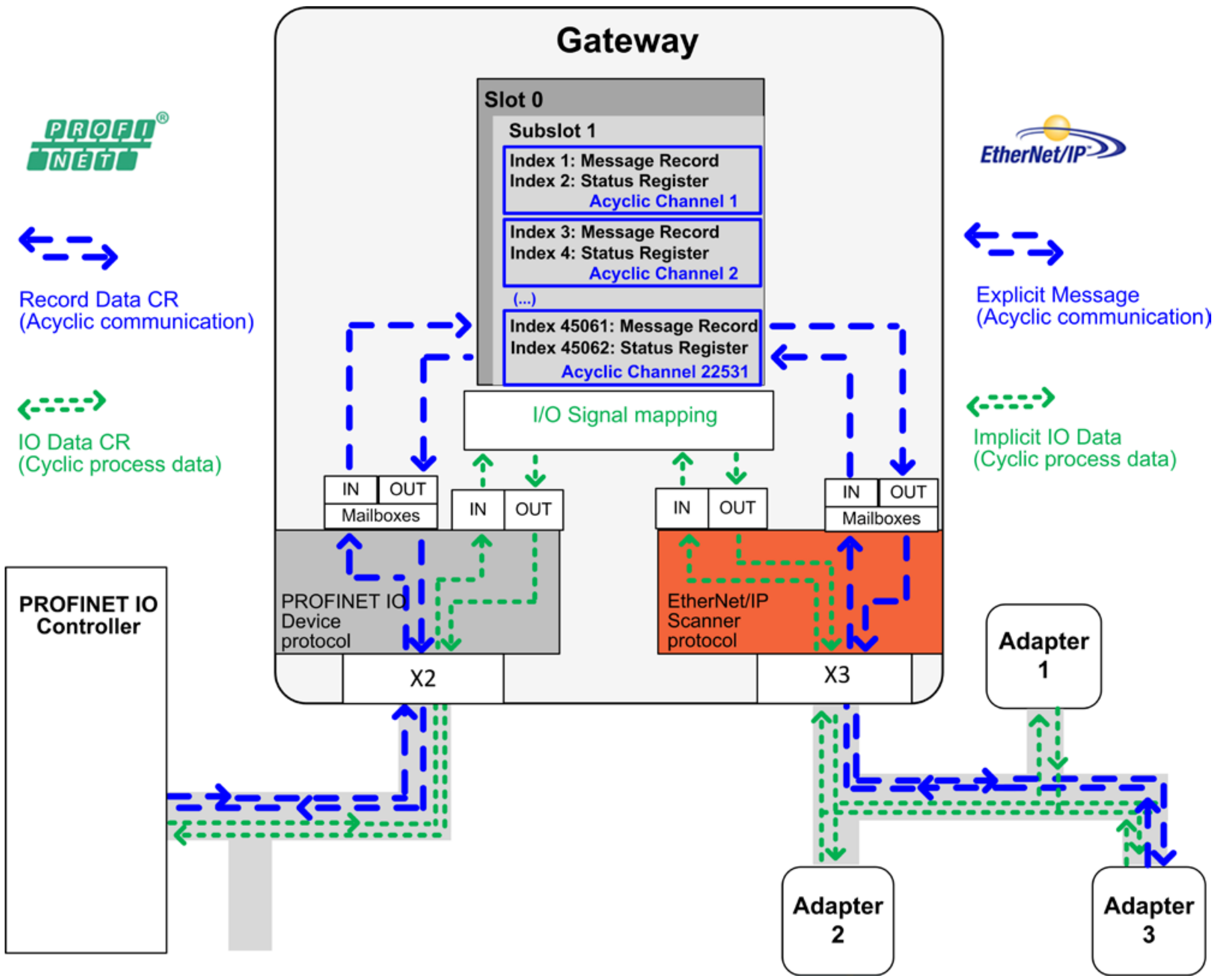


Figure 137: Acyclic Communication PROFINET IO-Device to Ethernet/IP Scanner

In order to initiate an acyclic service (write or read task) for an EtherNet/IP Adapter, the PROFINET PLC sends a **Write Data Record** (**WRREC**) request to **Slot 0**, **Subslot 1** and an **uneven Index** (e. g. `1`, `3`, `1201` etc.) in the gateway.

In order to then "read" the state of the acyclic task or the answer itself, the PROFINET PLC afterwards sends a **Read Data Record** (**RDREC**) request to the gateway, also addressed at **Slot 0**, **Subslot 1**. In case the PLC wants to read the **Request State**, it has to address its RDREC to the **even Index** denoting the **Status Register** (e. g. `2`, `4`, `1202` etc.); in case the PLC wants to read the answer received from the EtherNet/IP Adapter (given that it has already arrived), the PLC has to address its RDREC to the **uneven Index** denoting the **Message Record** (e. g. `1`, `3`, `1201` etc.).

8.5.2 Reading Data from an EtherNet/IP Adapter

8.5.2.1 Sequence of a Reading Task

This section describes the typical sequence of events of an error-free **acyclic read request** for an EtherNet/IP Adapter (see also diagram on following page):

1. The PROFINET PLC sends a **Write Data Record** (**WRREC**) to **Slot 0**, **Subslot 1** and an **uneven Index** number (1 or 3 or 5 etc) in the gateway. The gateway stores ("writes") the task in the **Message Record** of the acyclic channel and synchronously forwards the task as "unconnected message" to the EtherNet/IP Adapter.

   The gateway sends back a **WRREC ACK** (acknowledgement) to the PLC and sets the **Request State** in the **Status Register** of the acyclic channel to value 2.

2. The gateway sets the **Request State** in the **Status Register** of the acyclic channel to value 2 and sends back a **WRREC ACK** (acknowledgement) to the PLC.
   The EtherNet/IP Adapter executes the requested reading task.

3. The PROFINET PLC queries the state of the task at regular intervals by sending a **Read Data Record** (**RDREC**) to the **Status Register** of the acyclic channel in the gateway ("status polling"). As long as the gateway has not yet received the answer from the EtherNet/IP Adapter, the **Status Register** is returning **Request State** 2.

Note: Instead of polling the **Status Register**, the PLC can send a **RDREC** directly to the **Message Record**, expecting that the answer from the EtherNet/IP Adapter has already arrived at the gateway and can be read from there. If the answer from the Adapter has not yet arrived at the gateway at the time of the PLC reading the **Message Record**, the **Request State** in the **Status Register** is set to value 3 and the **RDREC REPLY** from the gateway to the PLC is stalled until the answer from the Adapter arrives.
Note that if the EtherNet/IP Adapter is slow, the RDREC of the Message Record might get timed-out. Therefore "status polling" should be done before reading the Message Record.

4. The answer from the EtherNet/IP Adapter arrives at the gateway where it is stored in the **Message Record** of the acyclic channel. The gateway sets the **Request State** in the **Status Register** to value 4.

5. The PROFINET PLC again sends a **Read Data Record** (**RDREC**) to the **Status Register**, and the gateway now returns **Request State** 4. The PLC now knows that the answer from the slave has arrived and that the requested data can be read from the **Message Record**.

6. The PLC now sends a **Read Data Record** (**RDREC**) to the **Message Record** of the acyclic channel in order to read the slave data.

7. The gateway resets the **Request State** to value 1. The acyclic channel is now ready to process a new request.

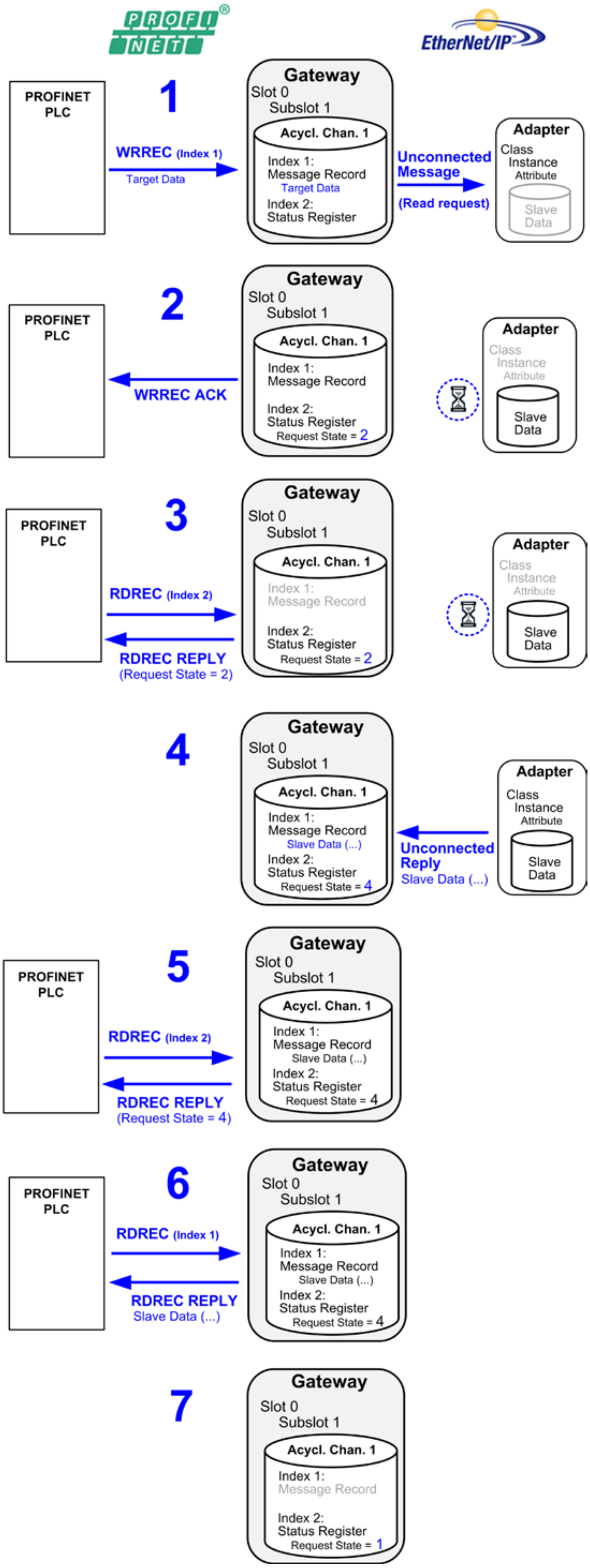


Figure 138: Sequence Acyclic Reading Task PROFINET to EtherNet/IP

8.5.2.2 Data Elements for a Reading Task

The PROFINET PLC sends the reading task for the EtherNet/IP Adapter to **Slot 0**, **Subslot 1** and an **uneven Index** in the gateway by using a **Write Data Record** containing the following elements:

Element		Description	Data type	Value
WRREC				
ID	PROFINET ID	Number, which unambiguously identifies the gateway in the Controller (Handle, Reference ID, Diagnostic address etc.)	-	(…)
	Slot	Slot (Object) for acyclic communication	uint16	0
	Subslot	Subslot (Object) for acyclic communication	uint16	1
Index		Object for Message Record in acyclic channel: uneven numbers 1…65533	uint16	n
LEN		Maximum length of the WRREC DATA in bytes	uint16	14
WRREC DATA				
IP Address		IP address of the EtherNet/IP Adapter	uint8[4]	(…)
Service		Service type (CIP Service Code): `Get Attribute Single`	uint16 Little Endian	14
Class		ID of the object class to be read from the EtherNet/IP Adapter	uint16 Little Endian	(…)
Instance		ID of the object instance to be read from the EtherNet/IP Adapter	uint16 Little Endian	(…)
Attribute		ID of the attribute to be read from the EtherNet/IP Adapter	uint16 Little Endian	(…)
DataLen		Length of the data field (payload data). Value 0 because no payload data is being sent to the Adapter in a reading task	uint16 Little Endian	0

Table 89: Elements of WRREC for Reading Task

8.5.2.3 Elements for Querying the Request State from the Status Register

The PROFINET PLC can read the request state from the **Status Register** of the acyclic channel by sending a **Read Data Record** (**RDREC**) to **Slot 0**, **Subslot 1** and the **even Index** following the uneven Index of the Message Record. The RDREC contains the following elements:

Element		Description	Data type	Value
RDREC				
ID	PROFINET ID	Number, which unambiguously identifies the gateway in the Controller (Handle, Reference ID, Diagnostic address etc.)	-	(…)
	Slot	Slot (Object) for acyclic communication	uint16	0
	Subslot	Subslot (Object) for acyclic communication	uint16	1
Index		Object for Status Register in acyclic channel: (even number following the index number of the Message Record)	uint16	n + 1
MLEN		Maximum length of the data in bytes	uint16	4

Table 90: Elements of RDREC for Status Register

Answer from the Gateway

The **Read Record Reply (RDREC Reply)** containing the Request State from the Status Register, which the gateway sends back to the PROFINET PLC, contains the following elements:

Element	Description	Data type	Value
RDREC Reply			
LEN	Length of the data in bytes	uint16	4
Data	Request State read from the Status Register: 0 = MSG_STATE_INIT 1 = MSG_STATE_IDLE 2 = MSG_STATE_REQ_SENT 3 = MSG_STATE_RSP_REQUESTED 4 = MSG_STATE_RSP_RECEIVED	uint32 Little Endian	0…4

Table 91: Elements RDREC Reply from Status Register

8.5.2.4 Elements for Reading the Answer from the Message Record

The PROFINET PLC can retrieve the answer from the EtherNet/IP Adapter by sending a **Read Data Record** (**RDREC**) to the **Message Record** of the acyclic channel (**Slot 0**, **Subslot 1** and the **uneven Index**). The RDREC contains the following elements:

Element		Description	Data type	Value
RDREC				
ID	PROFINET ID	Number, which unambiguously identifies the gateway in the Controller (Handle, Reference ID, Diagnostic address etc.)	-	(…)
	Slot	Slot (Object) for acyclic communication	uint16	0
	Subslot	Subslot (Object) for acyclic communication	uint16	1
Index		Object for Message Record in acyclic channel	uint16	n
MLEN		Maximum length of the data in bytes	uint16	(…)

Table 92: Elements of RDREC for Message Record

Answer from the Gateway

The **Read Record Reply (RDREC Reply)** containing the answer from the EtherNet/IP Adapter, which the gateway sends back to the PROFINET PLC, contains the following elements:

Element	Description	Data type	Value
RDREC Reply			
IP Address	IP address of the EtherNet/IP Adapter	uint8[4]	(…)
Service	Service type (CIP Service Code): `Get Attribute Single`	uint16 Little Endian	14
Status	Status value `0` means no error occurred If other value than `0` is returned, please refer to Status/Error Codes provided in chapter *EtherNet/IP Scanner Protocol API* in the Reference Manual *Hilscher Error Codes*, DOC100802PRxxEN (stored on the Gateway Solutions DVD in the directory: `Documentation\english\3.For Programmers\Error Codes Compilation`)	uint32 Little Endian	(…)
DataLen	Length of the payload data in bytes	uint16 Little Endian	0…1024
Data	Data read from the EtherNet/IP Adapter (payload data). Please note that payload data is being transported 1:1 (without swapping), therefore the PLC might needs to swap the data before processing it.	uint8[...]	(…)

Table 93: Elements of RDREC REPLY from Message Record

8.5.2.5 Example of a Reading Task

Activating the Reading Task

In case the PROFINET PLC wants to read the device name from an EtherNet/IP Adapter with the IP address `192.168.10.2`, the **Write Data Record** (WRREC) to the Gateway contains the following parameters:

Syntax									
WRREC				WRREC DATA					
PROFINET ID	Slot	Subslot	Index	IP Address	Service	Class	Instance	Attribute	DataLen
(…)	0x0000	0x0001	0x0001	0xC0 0xA8 0x0A 0x02	0x0E00	0x0100	0x0100	0x0700	0x0000

Table 94: Example of WRREC for a Reading Task

Querying the Request State from the Status Register

The **Read Data Record** (RDREC), which the PROFINET PLC sends to the gateway in order to read the **Request State** from the **Status Register** of the acyclic channel, contains the following parameters:

Syntax				
RDREC				
PROFINET ID	Slot	Subslot	Index	MLEN
(…)	0x0000	0x0001	0x0002	0x0004

Table 95: Example of RDREC to Status Register

Answer from the Gateway in Case of Unfinished Reading Task

If the gateway has not yet received the answer from the EtherNet/IP Adapter, the **RDREC REPLY**, which the Gateway sends back to the PROFINET PLC, contains the following parameters:

Syntax	
RDREC REPLY	
LEN	Data
0x0004	0x02000000

Table 96: Example of RDREC REPLY from Status Register in Case of an Unfinished Reading Task

Answer from the Gateway in Case of Finished Reading Task

If the answer from the EtherNet/IP Adapter has arrived at the gateway, the **RDREC REPLY**, which the Gateway sends back to the PROFINET PLC, contains the following parameters:

Syntax	
RDREC REPLY	
LEN	Data
0x0004	0x04000000

Table 97: Example of RDREC REPLY from Status Register in Case of a Finished Reading Task

Reading the Answer from the Message Record

The **Read Data Record** (RDREC), which the PROFINET PLC sends to the gateway in order to read the returned payload data of the EtherNet/IP Adapter from the **Message Record** of the acyclic channel, contains the following parameters (the expected length of the payload data is eight bytes):

Syntax				
RDREC				
PROFINET ID	Slot	Subslot	Index	MLEN
(…)	0x0000	0x0001	0x0001	0x0008

Table 98: Example of RDREC to Message Record

Answer from the Message Record with Payload Data from the EtherNet/IP Adapter

The **Read Record Reply (RDREC Reply)** containing the payload data of the EtherNet/IP Adapter, which the gateway sends to the PROFINET PLC, contains the following parameters (no error occurred, i. e. Status = `0` = `Success`):

Syntax				
RDREC REPLY				
IP Address	Service	Status	DataLen	Data
0xC0 0xA8 0x0A 0x02	0x0E00	0x00000000	0x0800	0x54 0x45 0x4d 0x45 0x4c 0x4b 0x4f 0x56

Table 99: Example of RDREC REPLY from Message Record

8.5.3 Writing Data into an EtherNet/IP Adapter

8.5.3.1 Sequence of a Writing Task

This section describes the typical sequence of events of an error-free **acyclic write request** for an EtherNet/IP Adapter (see also diagram on following page):

1. The PROFINET PLC sends a **Write Data Record** (**WRREC**) to **Slot 0**, **Subslot 1** and an **uneven Index** number (1 or 3 or 5 etc) within gateway. The gateway stores the task in the **Message Record** of the acyclic channel and synchronously forwards the writing task as “unconnected message” to the EtherNet/IP Adapter.
2. The gateway sets the **Request State** in the **Status Register** of the acyclic channel to value 2 and sends back a **WRREC ACK** (acknowledgement) to the PLC. The EtherNet/IP Adapter executes the requested writing task.
3. The PROFINET PLC queries the state of the task at regular intervals by sending a **Read Data Record** (**RDREC**) to the **Status Register** of the acyclic channel in the gateway (“status polling”). As long as the gateway has not yet received the response from the EtherNet/IP Adapter, the **Status Register** is returning **Request State** 2.

Note: Instead of polling the **Status Register**, the PLC can send a **RDREC** directly to the **Message Record**, expecting that the response from the EtherNet/IP Adapter has already arrived at the gateway and can be read from there. If the response from the Adapter has not yet arrived at the gateway at the time of the PLC reading the **Message Record**, the **Request State** in the **Status Register** is set to value 3 and the **RDREC REPLY** from the gateway to the PLC is stalled until the response from the Adapter arrives.
Note that if the EtherNet/IP Adapter is slow, the RDREC of the Message Record might get timed-out. Therefore “status polling” should be done before reading the Message Record.

4. The response from the EtherNet/IP Adapter containing a status message arrives at the gateway, where it is stored in the **Message Record** of the acyclic channel. The gateway sets the **Request State** in the **Status Register** to value 4.
5. The PROFINET PLC again sends a **Read Data Record** (**RDREC**) to the **Status Register**, and the gateway now returns **Request State** 4. The PLC now knows that the response from the slave has arrived and can be read from the **Message Record**.
6. The PLC now sends a **Read Data Record** (**RDREC**) to the **Message Record** of the acyclic channel in order to read the response. In case the EtherNet/IP Adapter reported an error in step 4, the PLC now receives an error code in the Status field of the RDREC REPLY and can react accordingly.
7. The gateway resets the **Request State** to value 1. The acyclic channel is now ready to process a new request.

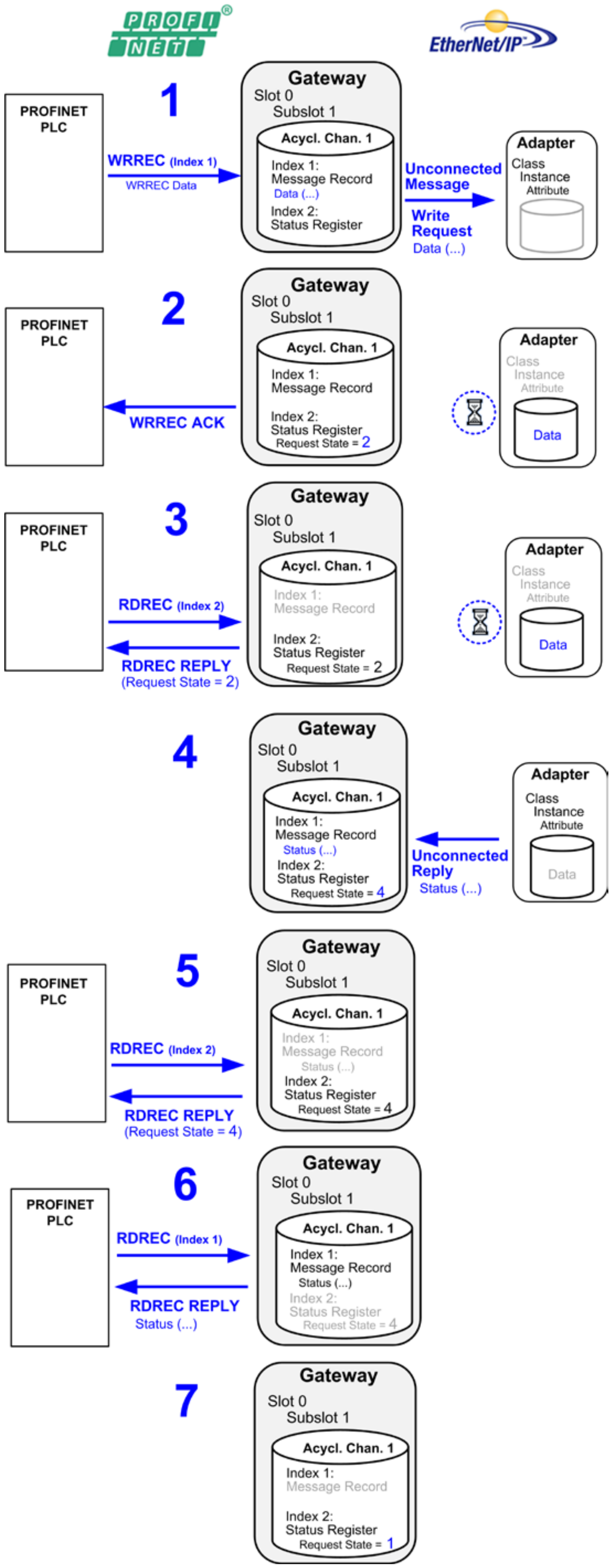


Figure 139: Sequence Acyclic Writing Task PROFINET to EtherNet/IP

8.5.3.2 Data Elements for a Writing Task

The PROFINET IO PLC sends the writing task for the EtherNet/IP Adapter to **Slot 0**, **Subslot 1** and an **uneven Index** in the gateway by using a **Write Data Record** containing the following elements:

Element		Description	Data type	Value
WRREC				
ID	PROFINET ID	Number, which unambiguously identifies the gateway in the Controller (Handle, Reference ID, Diagnostic address etc.)	-	(…)
	Slot	Slot (Object) for acyclic communication	uint16	0
	Subslot	Subslot (Object) for acyclic communication	uint16	1
Index		Object for Message Record in acyclic channel: uneven number 1…65533	uint16	n
LEN		Maximum length of the WRREC DATA in bytes	uint16	14 + DataLen
WRREC DATA				
IP Address		IP address of the EtherNet/IP Adapter	uint8[4]	(…)
Service		Service type (CIP Service Code): Set Attribute Single	uint16 Little Endian	16
Class		ID of the object class to be written in the EtherNet/IP Adapter	uint16 Little Endian	(…)
Instance		ID of the object instance to be written in the EtherNet/IP Adapter	uint16 Little Endian	(…)
Attribute		ID of the attribute to be written in the EtherNet/IP Adapter	uint16 Little Endian	(…)
DataLen		Length of the data field (payload data)	uint16 Little Endian	1…1024
Data		Data to be written in the EtherNet/IP Adapter (payload data). Please note that payload data is being transported 1:1 (without swapping), therefore the PLC might needs to swap the data before writing it.	uint8[1…1024]	(…)

Table 100: Elements of WRREC for Writing Task

8.5.3.3 Elements for Querying the Request State from the Status Register

The PROFINET IO PLC can read the request state from the **Status Register** of the acyclic channel by sending a **Read Data Record** (**RDREC**) to **Slot 0**, **Subslot 1** and the **even Index** following the uneven Index of the Message Record. The RDREC contains the following elements:

Element		Description	Data type	Value
RDREC				
ID	PROFINET ID	Number, which unambiguously identifies the gateway in the Controller (Handle, Reference ID, Diagnostic address etc.)	-	(…)
	Slot	Slot (Object) for acyclic communication	uint16	0
	Subslot	Subslot (Object) for acyclic communication	uint16	1
Index		Object for Status Register in acyclic channel: (even number following the index number of the Message Record)	uint16	n + 1
MLEN		Maximum length of the data in bytes	uint16	4

Table 101: Elements of RDREC for Status Register

Answer from the Gateway

The **Read Record Reply (RDREC Reply)** containing the Request State from the Status Register, which the gateway sends back to the PROFINET PLC, contains the following elements:

Element	Description	Data type	Value
RDREC Reply			
LEN	Length of the data in bytes	uint16	4
Data	Request State read from the Status Register: 0 = MSG_STATE_INIT 1 = MSG_STATE_IDLE 2 = MSG_STATE_REQ_SENT 3 = MSG_STATE_RSP_REQUESTED 4 = MSG_STATE_RSP_RECEIVED	uint32 Little Endian	0…4

Table 102: Elements RDREC Reply from Status Register

8.5.3.4 Elements for Reading the Response from the Message Record

The PROFINET PLC can retrieve the response from the EtherNet/IP Adapter by sending a **Read Data Record** (**RDREC**) to the **Message Record** of the acyclic channel (**Slot 0**, **Subslot 1** and the **uneven Index**). The RDREC contains the following elements:

Element		Description	Data type	Value
RDREC				
ID	PROFINET ID	Number, which unambiguously identifies the gateway in the Controller (Handle, Reference ID, Diagnostic address etc.)	-	(…)
	Slot	Slot (Object) for acyclic communication	uint16	0
	Subslot	Subslot (Object) for acyclic communication	uint16	1
Index		Object for Message Record in acyclic channel	uint16	n
MLEN		Maximum length of the data in bytes	uint16	10

Table 103: Elements of RDREC for Message Record

Answer from the Gateway

The **Read Record Reply (RDREC Reply)** containing the response and the status message from the EtherNet/IP Adapter, which the gateway sends back to the PROFINET PLC, contains the following elements:

Element	Description	Data type	Value
RDREC Reply			
IP Address	IP address of the EtherNet/IP Adapter	uint8[4]	(…)
Service	Service type (CIP Service Code): `Set Attribute Single`	uint16 Little Endian	16
Status	Status value `0` means no error occurred If other value than `0` is returned, please refer to Status/Error Codes provided in chapter *EtherNet/IP Scanner Protocol API* in the Reference Manual *Hilscher Error Codes*, DOC100802PRxxEN (stored on the Gateway Solutions DVD in the directory: `Documentation\english\3.For Programmers\Error Codes Compilation`)	uint32 Little Endian	(…)

Table 104: Elements of RDREC REPLY from Message Record

8.5.3.5 Example of a Writing Task

Activating the Writing Task

In case the PROFINET PLC wants to set the Input Range (Attribute = 7) of an Analog Input Point (Class = 10) of module 1 (Instance = 1) in an EtherNet/IP Adapter (IP address = 192.168.10.2) to the voltage range of 0V...10V (Value = 2), the **Write Data Record** (WRREC) to the gateway contains the following parameters:

Syntax										
WRREC				WRREC DATA						
PROFINET ID	Slot	Subslot	Index	IP Address	Service	Class	Instance	Attribute	DataLen	Data
(…)	0x0000	0x0001	0x0001	0xC0 0xA8 0x0A 0x02	0x1000	0x0A00	0x0100	0x0700	0x0100	0x02

Table 105: Example of WRREC for a Writing Task

Querying the Request State from the Status Register

The **Read Data Record** (RDREC), which the PROFINET PLC sends to the gateway in order to read the **Request State** from the **Status Register** of the acyclic channel, contains the following parameters:

Syntax				
RDREC				
PROFINET ID	Slot	Subslot	Index	MLEN
(…)	0x0000	0x0001	0x0002	0x0004

Table 106: Example of RDREC to Status Register

Answer from the Gateway in Case of Unfinished Writing Task

If the gateway has not yet received the response from the EtherNet/IP Adapter, the **RDREC REPLY**, which the Gateway sends back to the PROFINET PLC, contains the following parameters:

Syntax	
RDREC REPLY	
LEN	Data
0x0004	0x02000000

Table 107: Example of RDREC REPLY from Status Register in Case of an Unfinished Writing Task

Answer from the Gateway in Case of Finished Writing Task

If the response from the EtherNet/IP Adapter has arrived at the gateway, the **RDREC REPLY**, which the Gateway sends back to the PROFINET PLC, contains the following parameters:

Syntax	
RDREC REPLY	
LEN	Data
0x0004	0x04000000

Table 108: Example of RDREC REPLY from Status Register in Case of a Finished Writing Task

Reading the Response of the EtherNet/IP Adapter from the Message Record

The **Read Data Record** (RDREC), which the PROFINET PLC sends to the gateway in order to read the response of the EtherNet/IP Adapter from the **Message Record** of the acyclic channel, contains the following parameters:

Syntax				
RDREC				
PROFINET ID	Slot	Subslot	Index	MLEN
(…)	0x0000	0x0001	0x0001	0x000A

Table 109: Example of RDREC to Message Record

Answer from the Message Record Containing the Response from the EtherNet/IP Adapter

The **Read Record Reply (RDREC Reply)** containing the response from the EtherNet/IP Adapter, which the gateway sends to the PROFINET PLC, contains the following parameters (no error occurred, i. e. Status = `0` = `Success`):

Syntax		
RDREC REPLY		
IP Address	Service	Data
0xC0 0xA8 0x0A 0x02	0x1000	0x00000000

Table 110: Example of RDREC REPLY from Message Record

8.5.4 PNIO Error Codes for WRREC and RDREC

In case of a **Write**/**Read Data Record** error, the acyclic channel returns the following **PNIO Error Codes** to the PROFINET-Controller:

Write Data Record Error		Read Data Record Error	
Error Code	**Meaning**	**Error Code**	**Meaning**
0xDF80A200	General write fault	0xDE80A200	General read fault
0xDF80B300	Invalid secondary network message format	0xDE80A000	Secondary network interface replied with no data
0xDF80C200	Write to Message Record is invoked in invalid state	0xDE80B500	Read of Message Record is invoked in invalid state
0xDF80B800	Write record service is invoked on invalid Subslot	0xDE80B800	Read record service is invoked on invalid Subslot
0xDF80B80F	Maximum number of available acyclic channels exceeded	0xDE80B80F	Maximum number of available acyclic channels exceeded

Table 111: PNIO Error Codes WRREC and RDREC

9 Error Codes

9.1 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.

9.2 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 112: Overview Error Codes and Ranges

The fieldbus specific error codes are described in the manuals of the corresponding protocol tasks.

9.3 General Hardware Error Codes

9.3.1 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 113: RCX General Task Errors

9.3.2 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 114: RCX Common Status & Errors Codes

9.3.3 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 115: RCX Status & Error Codes

9.3.3.1 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 116: RCX Status & Error Codes Slave State

9.4 ODM Error Codes

9.4.1 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 117: ODM Error Codes - General ODM Error Codes

9.4.2 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 118: ODM Error Codes - General ODM Driver Error Codes

9.4.3 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 119: cifX Driver Specific ODM Error Codes

9.5 Error Codes cifX Device Driver and netX Driver

9.5.1 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 120: Generic Error Codes

9.5.2 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 121: Generic Driver Error Codes

9.5.3 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 122: Generic Device Error Codes

9.6 Error Codes netX Driver

9.6.1 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_ERROR	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 123: CIFX API Transport Error Codes

9.6.2 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 124: CIFX API Transport Header State Error Codes

9.7 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 125: ODM Error Codes DBM V4

10 Appendix

10.1 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** panes of the netGateway 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** panes, 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.

10.1.1 Settings

	Observer	Operator	Maintenance	Planning Engineer	Adminis-trator
Driver	D	D	X	X	X
Selecting the Driver	-	-	X	X	X
Device Assignment	D	D	X	X	X
Scanning for Devices	-	-	X	X	X
Selecting the Device	-	-	X	X	X

Table 126: Settings (D = Displaying, X = Editing, Configuring)

10.1.2 Configuration

	Observer	Operator	Maintenance	Planning Engineer	Adminis-trator
Settings	D	D	X	X	X
Signal Mapping	D	D	X	X	X
Memory Card Management	D	D	X	X	X

Table 127: Configuration (D = Displaying, X = Editing, Configuring)

10.2 List of Figures

10.3 List of Tables

10.4 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.

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 International. 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.

10.5 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