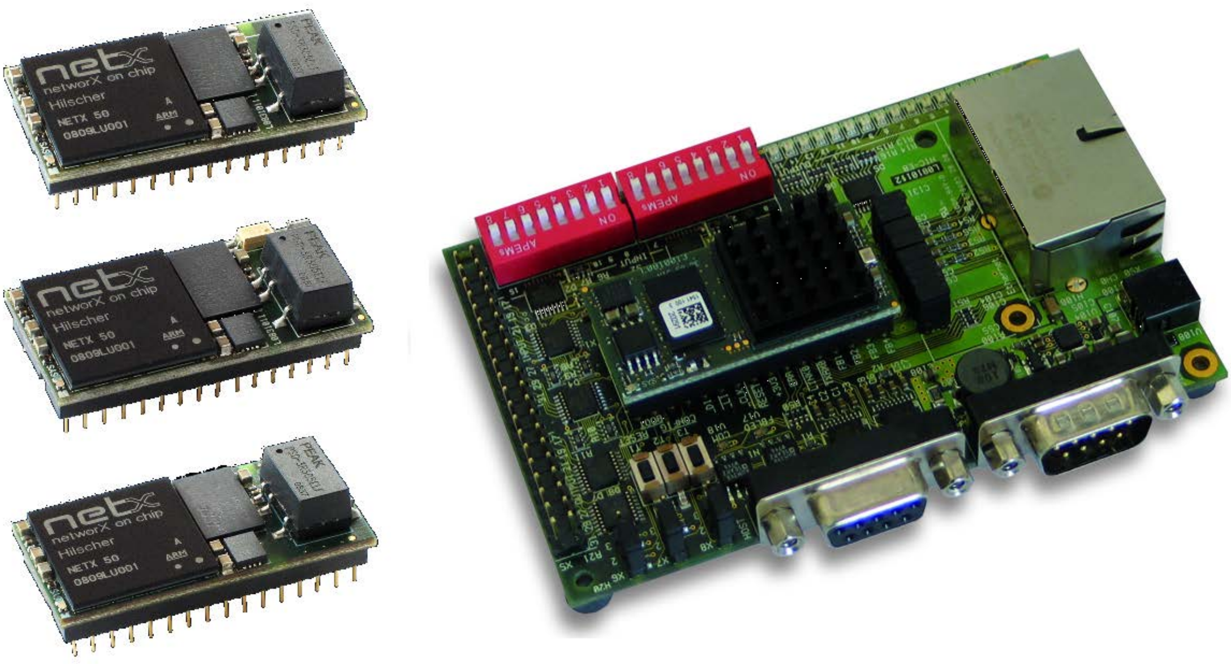
hilscher

empowering communication

User Manual and Design Guide

netIC

DIL-32 Communication IC for Real-Time Ethernet and Fieldbus



Hilscher Gesellschaft für Systemautomation mbH

www.hilscher.com

DOC080601UM38EN | Revision 38 | English | 2024-04 | Released | Public

Table of Contents

1 Introduction

1.1 Obligation to read and understand the Manual

Important!

- To avoid personal injury and to avoid property damage to your system or to your netIC, you must read and understand all instructions in the manual and all accompanying texts to your netIC, before installing and operating your netIC.
- First read the **Safety Chapter** on page 21!
- Keep the product DVD providing the product manuals available for later use.

1.2 About the User Manual and Design Guide

This user manual describes the hardware, installation, commissioning, and operation of the netIC product family from Hilscher based on the communication controllers of the netX family. The netIC is designed for use in simple field devices with some I/O data and time uncritical cycles.

The netIC product family consists of the netIC Real Time Ethernet Communication ICs

- NIC 50-RE
- NIC 50-RE\NHS and
- NIC 52-RE

and the netIC Fieldbus Communication ICs

- NIC 50-COS,
- NIC 50-DNS and
- NIC 50-DPS.

The user manual contains information required for installation, commissioning and use of these DIL-32 Communication ICs.

Additionally, it describes the corresponding evaluation boards NICEB and their use for loading and testing the firmware and the configuration of the netIC Communication ICs or for diagnostic purposes.

Finally, this document also describes the integration of the netIC Communication ICs into their target environment (host system).

1.2.1 List of Revisions

Index	Date	Chapter	Revisions
36	2019-03-28	1.2.2	Section *Reference to Hardware, Software and Firmware* updated and expanded.
		1.3.3	Section *Available Documentation* updated.
		7	Chapter *Configuration* revised.
		9.3.2	Section *LED CC-Link IE Field Basic* Slave updated.
		9.3.7	Section *LED PROFINET IO-RT-Device* updated.
		12.2.1	Section *System Information Block*: Feature Flags (register 19) added.
		12.2.5, 12.2.6	Sections *System Flags (Register 999)* and *Command Flags (Register 1999)*: Clear/Store configuration for netIC 52 added.
		16.3.6	Section *Busy Exception (05) for high load condition on Ethernet* updated.
		22	Section *Conformity Tests*: Table 152 updated.
37	2022-08-23	13.1.5	Section *Serial Shift IO Interface* corrected: 74HC165 and 74HC594 shift registers used.
		20.1	Section *Technical Data netIC DIL-32 Communication ICs*: UKCA sign added.
38	2024-04-09	1.3.2	Section *Device Description Files*: GSDML file name has been updated.
		20.1	Section *Technical Data netIC DIL-32 Communication ICs*: Values for environment updated.
		20.1.5	Section *NIC 52-RE*: Length of pins = 3.7 ± 0.2 mm
		All	NIC10-CCS removed. NIC50-REFO and NIC52-REFO removed.

Table 1: List of Revisions

1.2.2 Reference to Hardware, Software and Firmware

Hardware

Device (Catalog name)	Article number	Revision
NIC 50-RE	1541.100	Revision 4
NIC 50-RE/NHS	1541.101	Revision 4
NIC 50-COS	1541.540	Revision 1
NIC 50-DNS	1541.520	Revision 1
NIC 50-DPS	1541.420	Revision 2
NIC 52-RE	1544.100	Revision 3
NICEB	1540.000	Revision 3
NICEB-AIF-CO	Contained in adapter plug kit NICEB-CONKIT (Article number 1541.001)	Revision 1
NICEB-AIF-DN		Revision 1
NICEB-AIF-DP		Revision 1

Table 2: Reference to Hardware

Software

Software	Software Version
netX Configuration Tool-Setup: netX Configuration Tool.exe	V1.0900.x.x

Table 3: Reference to Software

Firmware

Firmware	Protocol	Firmware Version	For Hardware
NICMBECS.NXF	EtherCAT Slave	1.5	NIC 50-RE
NICMBEIS.NXF	EtherNet/IP Adapter	1.5	NIC 50-RE
NICMBOMB.NXF	Open Modbus/TCP	1.5	NIC 50-RE
NICMBPLS.NXF	POWERLINK Controlled Node	1.5	NIC 50-RE
NICMBPNS.NXF	PROFINET IO Device with FSU support	1.5	NIC 50-RE
NICMBS3S.NXF	Sercos Slave	1.5	NIC 50-RE
NICMBVRS.NXF	VARAN Client/Slave	1.5	NIC 50-RE
I07090R0.NXF	CC-Link IE Field Basic Slave	2.3	NIC 52-RE
I070F0R0.NXF	EtherCAT Slave	2.3	NIC 52-RE
I070H0R0.NXF	EtherNet/IP Adapter	2.3	NIC 52-RE
I070J0R0.NXF	Open Modbus/TCP	2.3	NIC 52-RE
I070K0R0.NXF	POWERLINK Controlled Node	2.3	NIC 52-RE
I070D0R0.NXF	PROFINET IO Device with FSU support	2.3	NIC 52-RE
I070L0R0.NXF	Sercos Slave	2.3	NIC 52-RE
NICMBCOS.NXF	CANopen Slave	1.5	NIC 50-COS
NICMBDNS.NXF	DeviceNet Slave	1.5	NIC 50-DNS
NICMBDPS.NXF	PROFIBUS DP Slave	1.5	NIC 50-DPS

Table 4: Reference to Firmware

Note: The functions "Clear/Store Configuration" is available for netIC 52 starting with firmware version 2.3 **and** requires the Second Stage Boot Loader V1.6 (or higher). netIC Communication Modules delivered with firmware version 2.3 (or higher), already have the Second Stage Bootloader V1.6 (or higher).

netIC 52 Communication Modules already delivered with firmware versionen V2.0, 2.1, 2.2 contains an older version oft he Second Stage Boot Loader and updating the firmware to V2.3 is insufficient (function "Clear/Store Configuration" cannot be used).

The Second Stage Boot Loader reserves 136 KB memory in the Flash memory for the function "Clear/Store Configuration". This reduces the previous available Flash memory by 136 KB.

Note: For NIC 52-RE, firmware version 2.0.1.0 and higher supports to deactivate Auto negotiation and to use fixed settings for the Ethernet speed (10 or 100 MBit/s) and duplex (half duplex or full duplex) for each Ethernet channel. Further, the MDI mode can be set for each Ethernet channel to MDI or MDI-X instead of the default Auto MDI-X. netX Configuration Tool from version 1.0900 has to be used to configure the NIC 52-RE device.

To deactivate Auto negotiation and Auto MDI-X may be necessary if the device that is direct connected via an Ethernet cable to NIC 52-RE requires fixed settings in order to use the Quick Connect function.

Note: netX Configuration Tool V1.0700.x.x requires firmware version 1.5.x.x for correct operation.

When updating to netX Configuration Tool V1.0700.x.x, you also have to update the firmware to V1.5.x.x, and vice versa. When updating the firmware to V1.5.x.x, a new configuration must be made and transferred. This can be accomplished using the netX configuration tool.

Note: Firmware version 1.1.x.x does not run on hardware revision 3 and 4 of the NIC 50-RE device. Use Firmware version 1.2.x.x or higher for hardware revision 3 and 4 of the NIC 50-RE device.

Note: The PROFINET IO Device firmware V1.2.x.x (and higher) for NIC 50-RE hardware revision 3 and 4, contains a new protocol stack implementation compared to the old firmware version 1.1.x.x. Firmware version 1.2.x.x (and higher) supports the FSU (Fast Start-Up) feature.

1.2.3 Conventions in this Manual

Operation instructions, a result of an operation step or notes are marked as follows:

Operation Instructions:

➢ <instruction>

or

1. <instruction>
2. <instruction>

Results:

➩ <result>

Notes:

Important: <important note>

Note: <note>

<note, where to find further information>

1.3 Contents of the Product DVD

The product DVD for the netIC Communication ICs contains:

- netX Configuration Tool setup including the serial driver
- Device Description Files (*GSD, GSDML, EDS, XML, XDD, CSP*)
- Documentation

1.3.1 Directory Structure of the DVD

All manuals on this DVD are delivered in the Adobe Acrobat® Reader format (PDF).

Directory Name	Description
Adobe Reader	Adobe Reader installation program
Documentation	Documentation in the Acrobat® Reader Format (PDF)
EDS	Device Description File
Example and API	Example and API
Firmware	Loadable Firmware
fscommand	Contains start-up menu of DVD
Software	netX Configuration Tool
Tools	Additional tools

Table 5: Directory Structure of the DVD

1.3.2 Device Description Files

The DVD ROM includes the device description files for the following slave devices:

Real-Time Ethernet / Fieldbus	File Name
EtherCAT Slave (NIC 50-RE)	Hilscher NIC 50-RE ECS V2.2.xml
EtherCAT Slave (NIC 52-RE)	Hilscher NIC 52 RE ECS V4.2.X.xml
EtherNet/IP Adapter (NIC 50-RE)	HILSCHER NIC 50-RE EIS V1.1.EDS
EtherNet/IP Adapter (NIC 52-RE)	HILSCHER NIC 52-RE EIS V1.1.EDS
Powerlink Controlled Node / Slave (NIC 50-RE)	00000044_NIC 50-RE PLS.xdd
Powerlink Controlled Node / Slave (NIC 52-RE)	00000044_NIC 52-RE PLS.xdd
PROFINET IO-RT-Device (NIC 50-RE) • NIC 50-RE/PNS V1.2.x - V1.4.15 • NIC 50-RE/PNS V1.4.16 - V1.5.x	GSDML-V2.3-HILSCHER-NIC 50-RE PNS-xxxxxxxx.xml GSDML-V2.35-HILSCHER-NIC 5X-RE PNS-xxxxxxxx.xml
PROFINET IO-RT-Device (NIC 52-RE) • NIC 52-RE/PNS V2.0 - V2.x	GSDML-V2.35-HILSCHER-NIC 5X-RE PNS-xxxxxxxx.xml
Sercos Slave (NIC 50-RE)	SDDML#v3.0#Hilscher#NIC_50_RE-FIXCFG_FSPIO#xxxx-xx-xx.xml
Sercos Slave (NIC 52-RE)	SDDML#v3.0#Hilscher#NIC_52_RE-FIXCFG_FSPIO#xxxx-xx-xx.xml
CC-Link IE Field Basic Slave	0x0352_NIC 52-RE CCIEBS_1_en.cspp
CANopen Slave	NIC 50-COS.eds
DeviceNet Slave	NIC50_DNS.EDS
PROFIBUS DP Slave	HIL_0C10.GSD

Table 6: Device Description Files

The device description file is required to configure the used Real Time Ethernet or Fieldbus Master device

The Real time Ethernet systems Open Modbus/TCP and VARAN do not use any device description files.

1.3.3 Available Documentation

All documents are available on the DVD delivered with the netIC evaluation boards NICEB underneath the directory *Documentation*.

The following overview on available documentation provides information, for which items you can find further information in which manual.

Manual	Contents	Document name
User Manual and Design Guide (this document)	Installation, commissioning, operation, hardware description and hardware design guide.	netIC - Real-Time Ethernet and Fieldbus Gateways UM DG xx EN.pdf
Operating Instruction Manual, netX Configuration Tool for netIC	Configuration of DIL-32 Communication ICs for Real Time Ethernet and Fieldbus	netIC Configuration by netX Configuration Tool OI xx EN.pdf
Functions of the Integrated WebServer in netIC DIL-32 Communication IC Devices	Description of integrated WebServer	Functions of the Integrated WebServer in netIC Devices AN xx EN.pdf
Functions of the Integrated FTP-Server in netIC Devices	Description of integrated FTP Server	Functions of the integrated FTP-Server in netIC Devices AN xx EN.pdf
netIC API Examples	Description of netIC API examples	netIC API and Example AN xx EN.pdf (Only English version)
Application Note: Protocol Parameter via Modbus	Configuration netIC	Protocol Parameter via Modbus AN xx EN.pdf
netX Dual-Port Memory Interface for netX based Products	Interface of netX Dual-Port Memory	netX Dual-Port Memory Interface DPM xx EN.pdf
Packet API: netX Dual-Port Memory, Packet-based services	General definition of packets. Description of System services, Communication Channel services, and Protocol Stack services.	netX Dual-Port Memory packet-based services API xx EN.pdf
CANopen Slave Protocol API Manual	Description of CANopen Slave Protocol API	CANopen Slave V3 Protocol API xx EN.pdf
CAN Data Link	Description of CAN Layer 2 Protocol API	CANopen Slave V3 Protocol API xx EN.pdf
Object Dictionary	Description of the Object Dictionary.	Object Dictionary V3 API xx EN.pdf
CC-Link IE Field Basic Slave Protocol API Manual	Description of CC-Link IE Field Basic Slave Protocol API	CC-Link IE Field Basic Slave Protocol API xx EN.pdf
DeviceNet Slave Protocol API Manual	Description of DeviceNet Slave Protocol API	DeviceNet Slave Protocol API xx EN.pdf
EtherCAT Slave V2 Protocol API Manual	Description of EtherCAT Slave V2 Protocol API (for NIC 50-RE)	EtherCAT Slave V2 Protocol API xx EN.pdf
EtherCAT Slave V4 Protocol API Manual	Description of EtherCAT Slave V4 Protocol API (for NIC 52-RE)	EtherCAT Slave V4 Protocol API xx EN.pdf
EtherNet/IP Adapter Protocol API Manual	Description of EtherNet/IP Adapter Protocol API (NIC 50-RE)	EtherNetIP Adapter Protocol API 12 EN.pdf
EtherNet/IP Adapter Protocol API Manual	Description of EtherNet/IP Adapter Protocol API (NIC 52-RE)	EtherNetIP Adapter Protocol API 20 EN.pdf
Open Modbus/TCP Protocol API Manual	Description of Open Modbus/TCP Protocol API (NIC 50-RE)	OpenModbusTCP Protocol API 09 EN.pdf
Open Modbus/TCP Protocol API Manual	Description of Open Modbus/TCP Protocol API (NIC 52-RE)	OpenModbusTCP Protocol API 10 EN.pdf

Manual	Contents	Document name
Powerlink Controlled node Protocol API Manual	Description of Powerlink Controlled Node Protocol API (NIC 50-RE)	Powerlink Controlled Node Protocol API 12 EN.pdf
Powerlink Controlled node Protocol API Manual	Description of Powerlink Controlled Node Protocol API (NIC 52-RE)	Powerlink Controlled Node Protocol V3 API 08 EN.pdf
PROFIBUS DP Slave Protocol API Manual	Description of PROFIBUS DP Slave Protocol API	PROFIBUS DP Slave Protocol API xx EN.pdf
PROFINET IO Device Protocol API Manual	Description of PROFINET IO RT Device Protocol API (NIC 50-RE)	PROFINET IO Device V3.4 Protocol API 14 EN.pdf
PROFINET IO Device Protocol API Manual	Description of PROFINET IO RT Device Protocol API (NIC 52-RE)	PROFINET IO-Device V3.13 Protocol API 18 EN.pdf
Sercos Slave Protocol API Manual	Description of Sercos Slave Protocol API (NIC 50-RE)	Sercos Slave Protocol API 12 EN.pdf
Sercos Slave Protocol API Manual	Description of Sercos Slave Protocol API (NIC 52-RE)	Sercos Slave Protocol API 17 EN.pdf
VARAN Client Protocol API Manual	Description of VARAN Client Protocol API	VARAN Client Protocol API xx EN.pdf

Table 7: Documents for netIC

1.4 Legal Notes

Important notes

Utmost care was/is given in the preparation of the documentation at hand consisting of a user's manual, operating manual and any other document type and accompanying texts. However, errors cannot be ruled out. Therefore, we cannot assume any guarantee or legal responsibility for erroneous information or liability of any kind. You are hereby made aware that descriptions found in the user's manual, the accompanying texts and the documentation neither represent a guarantee nor any indication on proper use as stipulated in the agreement or a promised attribute. It cannot be ruled out that the user's manual, the accompanying texts and the documentation do not completely match the described attributes, standards or any other data for the delivered product. A warranty or guarantee with respect to the correctness or accuracy of the information is not assumed.

We reserve the right to modify our products and the specifications for such as well as the corresponding documentation in the form of a user's manual, operating manual and/or any other document types and accompanying texts at any time and without notice without being required to notify of said modification. Changes shall be taken into account in future manuals and do not represent an obligation of any kind, in particular there shall be no right to have delivered documents revised. The manual delivered with the product shall apply.

Under no circumstances shall Hilscher Gesellschaft für Systemautomation mbH be liable for direct, indirect, ancillary or subsequent damage, or for any loss of income, which may arise after use of the information contained herein.

Liability disclaimer

The hardware and/or software was created and tested by Hilscher Gesellschaft für Systemautomation mbH with utmost care and is made available as is. No warranty can be assumed for the performance or flawlessness of the hardware and/or software under all application conditions and scenarios and the work results achieved by the user when using the hardware and/or software. Liability for any damage that may have occurred as a result of using the hardware and/or software or the corresponding documents shall be limited to an event involving willful intent or a grossly negligent violation of a fundamental contractual obligation. However, the right to assert damages due to a violation of a fundamental contractual obligation shall be limited to contract-typical foreseeable damage.

It is hereby expressly agreed upon in particular that any use or utilization of the hardware and/or software in connection with

- Flight control systems in aviation and aerospace;
- Nuclear fusion processes in nuclear power plants;
- Medical devices used for life support and
- Vehicle control systems used in passenger transport

shall be excluded. Use of the hardware and/or software in any of the following areas is strictly prohibited:

- For military purposes or in weaponry;
- For designing, engineering, maintaining or operating nuclear systems;
- In flight safety systems, aviation and flight telecommunications systems;
- In life-support systems;
- In systems in which any malfunction in the hardware and/or software may result in physical injuries or fatalities.

You are hereby made aware that the hardware and/or software was not created for use in hazardous environments, which require fail-safe control mechanisms. Use of the hardware and/or software in this kind of environment shall be at your own risk; any liability for damage or loss due to impermissible use shall be excluded.

Warranty

Hilscher Gesellschaft für Systemautomation mbH hereby guarantees that the software shall run without errors in accordance with the requirements listed in the specifications and that there were no defects on the date of acceptance. The warranty period shall be 12 months commencing as of the date of acceptance or purchase (with express declaration or implied, by customer's conclusive behavior, e.g. putting into operation permanently).

The warranty obligation for equipment (hardware) we produce is 36 months, calculated as of the date of delivery ex works. The aforementioned provisions shall not apply if longer warranty periods are mandatory by law pursuant to Section 438 (1.2) BGB, Section 479 (1) BGB and Section 634a (1) BGB [Bürgerliches Gesetzbuch; German Civil Code] If, despite of all due care taken, the delivered product should have a defect, which already existed at the time of the transfer of risk, it shall be at our discretion to either repair the product or to deliver a replacement product, subject to timely notification of defect.

The warranty obligation shall not apply if the notification of defect is not asserted promptly, if the purchaser or third party has tampered with the products, if the defect is the result of natural wear, was caused by unfavorable operating conditions or is due to violations against our operating regulations or against rules of good electrical engineering practice, or if our request to return the defective object is not promptly complied with.

Costs of support, maintenance, customization and product care

Please be advised that any subsequent improvement shall only be free of charge if a defect is found. Any form of technical support, maintenance and customization is not a warranty service, but instead shall be charged extra.

Additional guarantees

Although the hardware and software was developed and tested in-depth with greatest care, Hilscher Gesellschaft für Systemautomation mbH shall not assume any guarantee for the suitability thereof for any purpose that was not confirmed in writing. No guarantee can be granted whereby the hardware and software satisfies your requirements, or the use of the hardware and/or software is uninterruptable or the hardware and/or software is fault-free.

It cannot be guaranteed that patents and/or ownership privileges have not been infringed upon or violated or that the products are free from third-party influence. No additional guarantees or promises shall be made as to whether the product is market current, free from deficiency in title, or can be integrated or is usable for specific purposes, unless such guarantees or promises are required under existing law and cannot be restricted.

Confidentiality

The customer hereby expressly acknowledges that this document contains trade secrets, information protected by copyright and other patent and ownership privileges as well as any related rights of Hilscher Gesellschaft für Systemautomation mbH. The customer agrees to treat as confidential all of the information made available to customer by Hilscher Gesellschaft für Systemautomation mbH and rights, which were disclosed by Hilscher Gesellschaft für Systemautomation mbH and that were made accessible as well as the terms and conditions of this agreement itself.

The parties hereby agree to one another that the information that each party receives from the other party respectively is and shall remain the intellectual property of said other party, unless provided for otherwise in a contractual agreement.

The customer must not allow any third party to become knowledgeable of this expertise and shall only provide knowledge thereof to authorized users as appropriate and necessary. Companies associated with the customer shall not be deemed third parties. The customer must obligate authorized users to confidentiality. The customer should only use the confidential information in connection with the performances specified in this agreement.

The customer must not use this confidential information to his own advantage or for his own purposes or rather to the advantage or for the purpose of a third party, nor must it be used for commercial purposes and this confidential information must only be used to the extent provided for in this agreement or otherwise to the extent as expressly authorized by the disclosing party in written form. The customer has the right, subject to the obligation to confidentiality, to disclose the terms and conditions of this agreement directly to his legal and financial consultants as would be required for the customer's normal business operation.

Export provisions

The delivered product (including technical data) is subject to the legal export and/or import laws as well as any associated regulations of various countries, especially such laws applicable in Germany and in the United States. The products / hardware / software must not be exported into such countries for which export is prohibited under US American export control laws and its supplementary provisions. You hereby agree to strictly follow the regulations and to yourself be responsible for observing them. You are hereby made aware that you may be required to obtain governmental approval to export, reexport or import the product.

1.5 Registered Trademarks

Windows® XP, Windows® Vista and Windows® 7 are registered trademarks of Microsoft Corporation.

Adobe-Acrobat® is a registered trademark of the Adobe Systems Incorporated.

Pentium® is a registered trademark of Intel Corporation in the United States of America and further countries.

I^2C is a registered trademark of NXP Semiconductors, formerly Philips Semiconductors.

CANopen® is a registered trademark of CAN in AUTOMATION - International Users and Manufacturers Group e.V. (CiA), Nürnberg.

CC-Link® and CC-Link IE® are registered trademarks of Mitsubishi Electric Corporation, Tokyo, Japan.

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

Sercos interface® is a registered trademark of Sercos International e. V., Suessen, Germany.

All other mentioned trademarks are property of their respective legal owners.

1.6 EtherCAT Disclaimer

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

To get details and restrictions regarding using the EtherCAT technology refer to the following documents:

- “EtherCAT Marking rules”
- “EtherCAT Conformance Test Policy”
- “EtherCAT Vendor ID Policy”

These documents are available at the ETG homepage www.ethercat.org or directly over info@ethercat.org.

A summary over Vendor ID, Conformance test, Membership and Network Logo can be found within the appendix section of this document under section *EtherCAT Summary over Vendor ID, Conformance Test, Membership and Network Logo* on page 236.

2 Safety

2.1 General Note

The user manual, the accompanying texts and the documentation are written for the use of the products by educated personnel. When using the products, all safety instructions and all valid legal regulations have to be obeyed. Technical knowledge is presumed. The user has to assure that all legal regulations are obeyed.

2.2 Intended Use

2.2.1 Intended Use of the netIC Communication ICs

The netIC Communication ICs for Real Time Ethernet and Fieldbus described in this user manual are Modbus-RTU based Communication ICs to a communication network listed below for the DIL-32 socket.

Depending on the chosen model, the communication protocol listed hereafter can be realized using these netIC Fieldbus DIL-32 Communication ICs:

- EtherCAT Slave with NIC 50-RE and NIC 52-RE
- EtherNet/IP Adapter (Slave) with NIC 50-RE and NIC 52-RE
- Open Modbus/TCP (Server) with NIC 50-RE and NIC 52-RE
- Powerlink Controlled Node / Slave with NIC 50-RE
- PROFINET IO-RT-Device with NIC 50-RE, NIC 52-RE
- Sercos-Slave with NIC 50-RE and NIC 52-RE
- VARAN Client with NIC 50-RE
- CANopen Slave with NIC 50-COS
- DeviceNet Slave with NIC 50-DNS
- PROFIBUS DP Slave with NIC 50-DPS

The netIC Communication ICs may only be used as a part of a communication system as described in chapter "*Design-In - Integrating netIC into the Host System*" of this manual. They have exclusively been designed for use in connection with devices connected via serial interface supporting the Modbus RTU protocol on one hand and a communication network (listed above) on the other hand. Typically, the netIC Communication ICs are integrated into a host device.

2.2.2 Intended Use of the Evaluation Boards NICEB

The evaluation board NICEB described in this user manual is a board extending the netIC Communication IC, with all relevant interfaces needed in order to evaluate and test the functionality of the netIC Communication IC, to load the firmware and configuration data and to develop solutions for the integration of the netIC Communication IC into the intended target environment.

Also see sections *Design-In - Integrating netIC into the Host System* on page 109 and *netIC Evaluation Board* NICEB on page *147*.

The Evaluation Board NICEB should only be used in conjunction with the corresponding power supply delivered by Hilscher.

the Evaluation Board NICEB special adapters are required. The following table shows which adapter is required for which netIC Communication IC.

netIC Communication IC	Suitable Adapter
NIC 50-COS	NICEB-AIF-CO
NIC 50-DNS	NICEB-AIF-DN
NIC 50-DPS	NICEB-AIF-DP

Table 8: Suitable Adapters

Device Destruction!

- When using the NICEB Evaluation Board with the Fieldbus-Versions of the netIC Gateways NIC 50-COS, NIC 50-DNS respectively NIC 50-DPS: Remove the jumpers X4 on the NICEB. Setting the X4 jumpers would cause a **short circuit**!
- Therefore, **never** use a netIC **Fieldbus** Communication IC within the NICEB **with the Ethernet jumpers X4 set**!

Note: For using the NIC 50-RE or NIC 52-RE in connection with the Evaluation Board NICEB no adapter at all is required. The jumpers X4 must always be set when the NIC 50-RE or NIC 52-RE is mounted within the NICEB Evaluation Board!

Important: Hilscher is not liable for any damage caused by incorrect setting of jumpers, usage of an inadequate adapter or an inadequate power supply.

No CE Sign!

- The Evaluation Board NICEB has only been designed for test use. It has **no CE sign** and it has **not been tested regarding its emission and immunity behavior**. Therefore it is not suited for use in an industrial production environment!

2.2.3 Personnel Qualification

The netIC Real Time Ethernet and Fieldbus DIL-32 Communication ICs must only be installed, configured and removed by qualified personnel.

Professional qualification in the following specific areas of electrical engineering is required:

- Security and protection of health at work
- Mounting and attaching of electrical equipment
- Measurement and analysis of electrical functions and systems
- Evaluation of the security of electrical equipment

2.3 References Safety

[1] ANSI Z535.6-2011 American National Standard for Product Safety Information in Product Manuals, Instructions, and Other Collateral Materials

[2] IEC 60950-1, Information technology equipment - Safety -
Part 1: General requirements,
(IEC 60950-1:2005, modified); German Edition EN 60950-1:2006

[3] EN 61340-5-1 and EN 61340-5-2 as well as IEC 61340-5-1 and IEC 61340-5-2

2.4 Safety Instructions on Personal Injury

To ensure your own personal safety and to avoid personal injury, you necessarily must read, understand and follow the following and all other safety instructions in this guide.

2.4.1 Electrical Shock Hazard

A potentially lethal electrical shock may be caused by parts with more than 50V!

An electrical shock is the result of a current flowing through the human body. The resulting effect depends on the intensity and duration of the current and on its path through the body. Currents in the range of approximately ½ mA can cause effects in persons with good health, and indirectly cause injuries resulting from startle responses. Higher currents can cause more direct effects, such as burns, muscle spasms, or ventricular fibrillation.

In dry conditions permanent voltages up to approximately 42.4 V peak or 60 V DC are not considered as dangerous, if the contact area is equivalent to a human hand.

Reference: [2]

Therefore take care of the following rules when opening the device or working with the opened device:

- HAZARDOUS VOLTAGE may be inside of the device into which the netIC Communication IC is to be integrated. Therefore:
- First disconnect the power plug of the device into which the netIC Communication IC is to be integrated.
- Make sure, that the device is really free of electric power.
- Avoid touching open contacts or ends of wires!
- In any case, strictly adhere to the instructions given in the documentation of the device provided by its manufacturer.
- Open the device and install the netIC Communication IC only after having completed all of the preceding steps.

2.5 Warnings on Property Damage

To avoid property damage respectively device destruction to the DIL-32 Communication IC and to your system, you necessarily must read, understand and follow the following and all other property damage messages in this guide.

The following applies to all types of netIC Communication ICs described in this manual.

2.5.1 Device Destruction by exceeding the allowed Supply Voltage

2.5.1.1 netIC Communication ICs

The netIC Communication IC is designed for operation at a 3.3 V supply voltage. The use of a higher supply voltage than 3.3V may result in severe damage to the communication IC!

Therefore, the netIC Communication IC may not be powered by a 5V supply voltage! Operation with 5 V supply voltage will most probably cause device destruction.

Also the level of all I/O signals may not exceed a voltage of 3.3V.

For more information, see section 20.1 "*Technical Data netIC DIL-32 Communication ICs*" on page 196.

Evaluation Boards

The following is valid for the evaluation board NICEB:

NOTICE

Device Destruction!

- The voltage at the evaluation board must not exceed 30 V, otherwise the device and/or the evaluation board may be destroyed.

2.5.2 Electrostatic Discharge

Adhere to the necessary safety precautions for components that are vulnerable with electrostatic discharge.

The netIC Communication IC is sensitive to electrostatic discharge, which can cause intern al damage and affect normal operation.

Always follow these guidelines when you handle this netIC Communication IC:

- Touch a grounded object to discharge static potential.
- Wear a grounding ribbon.
- Do not touch connectors or pins on component boards.
- Do not touch circuit components inside the equipment.
- When not in use, store the equipment in appropriate static-safe packaging.

Reference: [3]

2.5.3 Device Damage by Erasing the Firmware or the Files `security.cfg` and `ftpuser.cfg` within the File System of the netIC Device

For those netIC devices which can be administered by the FTP Server or Web Server (NIC 50-RE and NIC 52-RE), the following additional warnings apply:

NOTICE

Device Destruction!

- The device will be left in an **unusable state** if the firmware is erased and subsequently the device is switched off or restarted before supplying the device with another suitable firmware.

NOTICE

Device Destruction!

- The device will be left in an **unusable state** if either of the files `security.cfg` and `ftpuser.cfg` is erased and subsequently the device is switched off or restarted.

2.5.4 Device Destruction by Constantly Writing Remanent Chips

---

NOTICE

Device Destruction!

- All remanent chips (e.g. serial Flash chips) have a limited number of write cycles. The maximum are 10000 or 100000 write cycles.
  Writing the remanent chip permanently (e.g. changing the configuration or changing the name of station) will cause a device desctruction.

2.6 Labeling of Safety Instructions

The safety instructions are pinpointed particularly. The instructions are highlighted with a specific safety symbol, a warning triangle and a signal word according to the degree of endangerment. Inside the note the danger is exactly named. Instructions to a property damage message do not contain a warning triangle.

Symbol	Symbol (USA)	Sort of Warning or Principle
		Warning of Personal Injury
		Warning of Lethal Electrical Shock
		Warning of danger by electrical current
		Warning of damages by electrostatic discharge
		Principle: Disconnect the power plug
		Principle: Mandatory read Manual

Table 9: Safety Symbols and Sort of Warning or Principle

Signal Word	Meaning
DANGER	Indicates a direct hazard with high risk, which will have a consequence of death or grievous bodily harm if it is not avoided.
WARNING	Indicates a possible hazard with medium risk, which will have a consequence of death or (grievous) bodily harm if it is not avoided.
CAUTION	Indicates a minor hazard with medium risk, which could have a consequence of minor or moderate bodily harm if it is not avoided.
NOTICE	Indicates a Property Damage Message.
Note	Indicates an important note in the manual.

Signal Word (USA)	Meaning (USA)
⚠ DANGER	Indicates a Hazardous Situation Which, if not Avoided, will Result in Death or Serious Injury.
⚠ WARNING	Indicates a Hazardous Situation Which, if not Avoided, could Result in Death or Serious Injury.
⚠ CAUTION	Indicates a Hazardous Situation Which, if not Avoided, may Result in Minor or Moderate Injury.
NOTICE	Indicates a Property Damage Message.
Note	Indicates an Important Note in the Manual.

Table 10: Signal Words

In this document the safety instructions and property damage messages are designed according both to the international used safety conventions as well as to the ANSI standard, refer to reference safety [1].

3 Description and Requirements

3.1 Description

Simple field devices such as barcode readers, identification systems, valve islands or digital and analogue inputs and outputs will require a connection to Fieldbus or Real-Time Ethernet systems. These devices do not have a high data throughput so it is very suitable to use a serial connection to the communication interface such as UART.

The netIC is a complete ‘Single Chip Module’ in the compact dimensions of a DIL-32 IC. It is based on the netX network controller and contains all components of a Fieldbus or Real-Time Ethernet interface with integrated 2-Port Switch and Hub. With the netX technology the whole spectrum of relevant Fieldbus and Real-Time Ethernet systems is covered by loadable Firmware with one netIC. These serial interfaces are available for application on which the user data are transferred with simple write-read commands. The widely known Modbus RTU protocol is implemented as a serial protocol on the Host interface.

Alternatively conventional shifting registers can be controlled via a synchronous serial interface so that no additional processor for a simple IO-Device is required.

Highlights

- Available for Fieldbus and all Real-Time Ethernet systems
- Integrated Switch and Hub
- Fits into a DIL-32 socket
- UART interface with Modbus RTU protocol
- Fiber Optic for Real-Time Ethernet (PROFINET IO Device) available
- PROFINET firmware contains support for Fast Start-Up (FSU)

The netIC only requires a 3.3 V power supply and two RJ45 Ports with integrated transmitter for operation on a Real-Time Ethernet system respectively all components of the Fieldbus Interface. Examples of schematic diagrams are included in chapter *Design-In - Integrating netIC into the Host System* beginning at page 109 of this document.

For each netIC type the NICEB evaluation board is available for testing, loading the Firmware and for the configuration. The configuration is transmitted from the Host system or can be saved with the **netX Configuration Tool** as a configuration file on the netIC.

3.1.1 Description of the netIC Real Time Ethernet DIL-32 Communication ICs NIC 50-RE and NIC 52-RE

The netIC Real Time Ethernet DIL-32 Communication ICs NIC 50-RE and NIC 52-RE represents a complete 'single chip module' with very compact dimensions as it is mounted into a DIL-32 package.

Depending of the loaded firmware, DIL-32 Communication ICs NIC 50-RE and NIC 52-RE process the communication of one of the following real time Ethernet systems:

- CC-Link IE Field Basic Slave
- EtherCAT Slave
- EtherNet/IP Adapter (Slave)
- Open Modbus/TCP
- Powerlink Controlled Node / Slave (only NIC 50-RE)
- PROFINET IO-RT-Device
- Sercos Slave
- VARAN Client /Slave (only NIC 50-RE)

Note: You must decide which one of these systems you intend to use as only one firmware can be loaded at the same time and the adaptation to the desired real time Ethernet system is done by exchange of the firmware.

Depending on the loaded firmware a switch or a hub are integrated within the netIC Real Time Ethernet DIL-32 Communication ICs NIC 50-RE and NIC 52-RE.

Loading and testing of the firmware or the configuration is possible by using the evaluation board NICEB described in section '*netIC Evaluation Board* NICEB*'* on page 147.

3.1.2 Description of the netIC Fieldbus DIL-32 Communication ICs

Each of the netIC Fieldbus DIL-32 Communication ICs represents a complete 'single chip module' with very compact dimensions as it is mounted into a DIL-32 package.

Depending on the device type of, the netIC Fieldbus-DIL-32 Communication IC communicates according to the standards of

- CANopen Slave (NIC 50-COS only)
- DeviceNet Slave (NIC 50-DNS only)
- PROFIBUS DP Slave (NIC 50-DPS only)

Note: For each type of Fieldbus DIL-32 Communication IC the suitable firmware must be loaded. The file name can be retrieved from *Table 4.* The firmware corresponding to the Real-Time Ethernet or Fieldbus system can be chosen by the icon in the **netX Configuration Tool**.

Loading and testing of the firmware or the configuration is possible by using the evaluation board NICEB described in section *netIC Evaluation Board* NICEB on page 147 together with a suitable adapter. *Table 8* explains which adapter fits for which device type.

3.2 System Requirements

The modules of the netIC family of Real-Time Ethernet and Fieldbus DIL-32 Communication ICs are designed as component of an electronic device or system (and not for stand-alone operation). This device or system is denominated as host system or target environment of the netIC Communication ICs in the scope of this document.

For a reasonable application of these modules of the netIC family of Real-Time Ethernet and Fieldbus DIL-32 Communication ICs, the following requirements must be fulfilled:

At the host system:

1. Mechanical connection: DIL-32 socket.
2. Electrical connection: Pin assignment according to description given within this document
3. Communication: has to be done using the Modbus-RTU protocol
4. Power supply: has to be done using pins 1 (3V3) and 32 (GND) of the netIC Communication IC. The voltage to be applied must not exceed the allowed range 3.3 V ± 5%.
5. At the connected communication system (Real-time Ethernet or Fieldbus):
6. A master of the communication system supported both by the netIC module and the loaded firmware.

For requirements #2 and #4 also refer to the description of the pin assignments in sections in sections 13.2 up to 13.2.4.

3.3 Preconditions for Operation of netIC Communication ICs

The following preconditions apply depending on the type of DIL-32 Communication IC (Real-Time Ethernet or Fieldbus-DIL-32 Communication IC):

1. The netIC Communication IC must correctly be mounted in the DIL-32 socket of its host system which needs to be designed according to the guide lines discussed in section *Design-In - Integrating netIC into the Host System* on page 109.
2. The netIC Communication IC must have been provided with the correctly suiting firmware for the Real-Time Ethernet or Fieldbus communication system (and protocol) of your choice intended to be executed on the device. The firmware must have been downloaded to the netIC Communication IC device using the evaluation board NICEB, also see next section.
3. The device must have been configured correctly using the **netX Configuration Tool** or via Modbus, see section "*Configuration*" on page 46.
4. A suitable power supply for the required voltage range is required.

For permissible ranges of supply voltage and environmental conditions required for operation, see section „*Technical Data netIC DIL-32 Communication ICs*“ on page 196.

Additionally, all safety precautions described in the preceding chapter must be adhered.

3.4 Preconditions for Usage of netIC Communication ICs together with Evaluation Boards NICEB

In the following situations it is necessary to use the netIC Communication IC device mounted in the evaluation board NICEB:

- You want to change or download the configuration data.
- You want to download the firmware to the DIL-32 Communication IC device.
- You want to access diagnosis functionality with the evaluation board.

For firmware download the following Real Time Ethernet or Fieldbus communication protocols can be chosen:

DIL-32 Communication IC Model	Applicable protocol/firmware
NIC 50-RE, NIC 50-RE/NHS	EtherCAT-Slave, EtherNet/IP-Adapter (Slave), Open Modbus/TCP, Powerlink Controlled Node, PROFINET IO-Device, Sercos Slave VARAN Client (Slave)
NIC 52-RE	CC-Link IE Field Basic Slave, EtherCAT-Slave, EtherNet/IP-Adapter (Slave), Open Modbus/TCP, Powerlink Controlled Node, PROFINET IO-Device, Sercos Slave
NIC 50-COS	CANopen Slave
NIC 50-DNS	DeviceNet Slave
NIC 50-DPS	PROFIBUS DP Slave

Table 11: Available Firmware/ Real-Time Ethernet and Fieldbus Communication Protocols

The following preconditions are necessary in order to operate any netIC Real Time Ethernet or Fieldbus DIL-32 Communication IC device within its evaluation board NICEB successfully.

1. The netIC Communication IC device must be mounted correctly in the DIL-32 socket of the NICEB evaluation board (For right orientation see “*Table 12: Position of the tag of the netIC devices”* on page 34).
2. The power supply delivered together with the evaluation board (Supply voltage 24 V) has to be connected to the power supply connector of the evaluation board. In any case, the supply voltage at the evaluation board must not exceed the maximum limit of 30 V, see subsection “*Device Destruction by exceeding the allowed Supply Voltage”* on page 25!
3. The diagnostic interface of the evaluation board has to be connected to a serial interface (COM port, RS232) of the PC using the cable CAB-SRV. The cable CAB-SRV is delivered together with the evaluation board.
4. The **netX Configuration Tool** has been successfully installed to that PC (unless configuration is done via Modbus). The requirements for this installation are listed below.
5. To use the diagnosis interface, a serial device driver is required. The driver is installed during the installation of the **netX Configuration Tool.**

For communication:

6. For communication using the chosen protocol a master device for the corresponding communication system needs to connected to
   - an Ethernet port of the evaluation board NICEB when using the NIC 50-RE or NIC52-RE DIL-32 Communication IC.
   - the correct Fieldbus adapter when using one the NIC 50-COS/ DNS/DPS DIL-32 Communication ICs.
7. For Modbus RTU communication the evaluation board must be set correctly for the chosen type of serial connection (RS232, RS422 or RS485) by setting jumpers accordingly. For detailed information on how to set the jumpers of the evaluation board correctly, see section *Host Interface Connector and Hardware Interface Configuration* on page 152.

3.4.1 System Requirements for netX Configuration Tool

The system requirements necessary for the application of the **netX Configuration Tool** are these:

- PC with 586-, Pentium® processor or higher
- The PC must provide a COM port RS232 interface.
- Operating system: Windows® XP SP3, Windows® Vista (32 bit) SP2, Windows® 7 (32 bit) or Windows® 7 (64 bit), Windows® 8 (32-Bit) or Windows® 8 (64-Bit)
- Administrator privilege required for installation
- Free space on hard disk: 50 MByte
- DVD ROM drive
- RAM: min. 256 MByte
- Graphics resolution: min 1024 x 768 pixels
- Keyboard and mouse

Limitations when using netX Configuration Tool

When using the netX Configuration Tool, the following restrictions apply:

Important Limitation for PROFINET-IO:
Only 4 input modules of each 64 bytes/words at maximum and 4 output modules of each 64 bytes/words at maximum can be configured with the netX Configuration Tool. Therefore the total amount of input data (and output data, as well) that can be used when configuring the netIC with netX Configuration Tool **is limited to 256 bytes/words**.

Important Limitation for PROFIBUS-DP:
Only 4 input modules of each 64 bytes/words at maximum and 4 output modules of each 64 bytes/words at maximum can be configured with the netX Configuration Tool.

For all other communication systems, no such restrictions apply.

4 Getting Started

4.1 Steps how to install and configure the netIC Communication IC Devices with the Evaluation Board

Before the *'Steps how to install and configure the netIC Communication IC Devices with the Evaluation Board'* are described, the following figures are shown which are referenced in the step by step description.

- The figure of the evaluation board NICEB with the names and position of connections, interfaces, pushbutton and LEDs
- The figure of the dialog structure of the software **netX Configuration Tool**

The following figure shows the evaluation board NICEB with specified names and positions of connectors, interfaces, push buttons and LEDs.

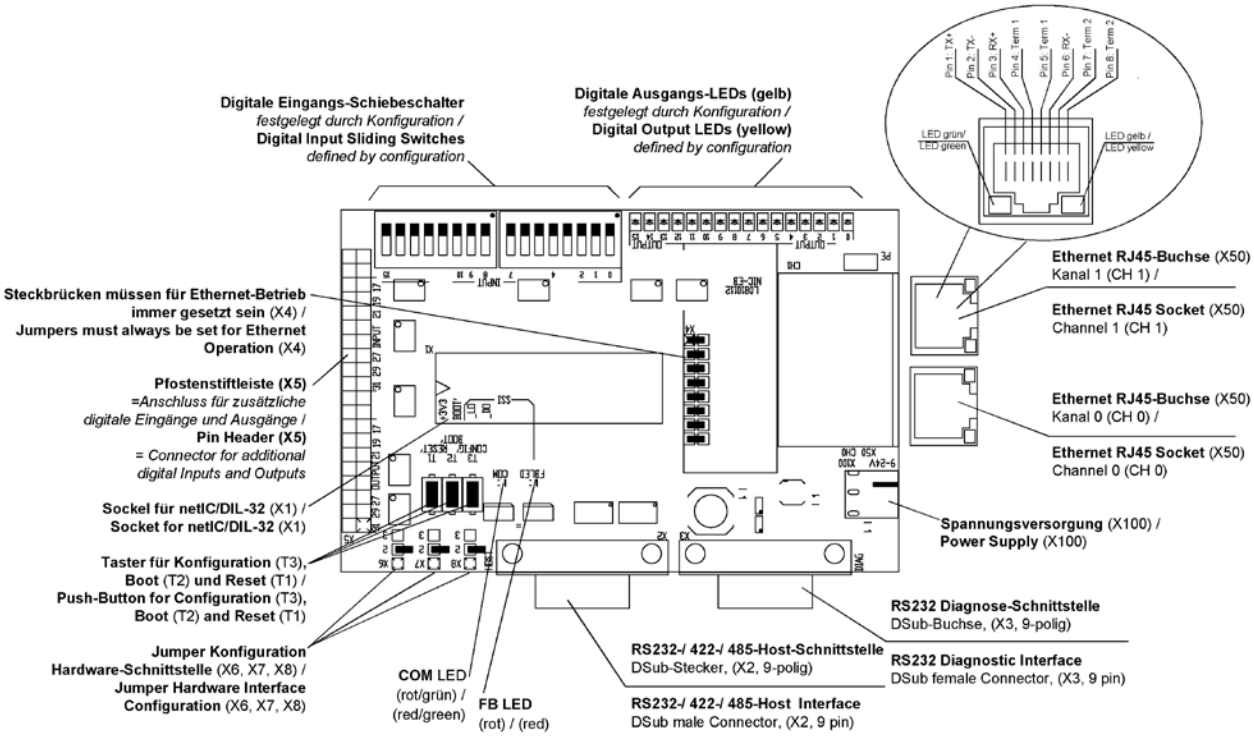


Figure 1: Device Drawing NICEB

The following two figures show the position of the tag. This tag is in the DIL-32 socket and visible in bottom view of the netIC Communication IC device. The tag is important when the netIC Communication IC device is mounted on the NICEB evaluation board. Both tags have to match.

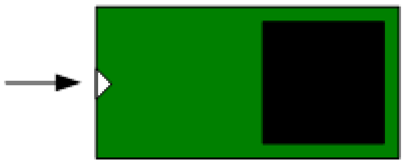
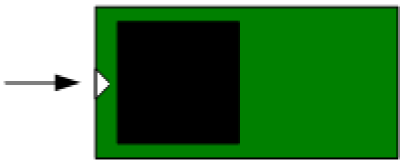
Position of the tag of the NIC 50-RE,NIC 50-RE/NHS and NIC 52-RE	Position of the tag of the NIC 50-COS, NIC 50-DNS and NIC 50-DPS
	

Table 12: Position of the tag of the netIC devices

With the **netX Configuration Tool** you can download firmware and configuration into the netIC Communication IC device and use diagnostic functions.

The graphical user interface of the **netX Configuration Tool** is composed of different areas and elements listed hereafter:

1. A header area containing the **Select Network and Language Bar** and the **Device Identification,**
2. The **Navigation Area** (area on the left side) including the menu buttons **Configuration** and **Diagnostic** and depending on the device additional menu buttons (at the lower side of the navigation area),
3. The **Dialog Pane** (main area on the right side),
4. The general buttons **OK, Cancel, Apply, Help,**
5. The **Status Bar** containing information e. g. the online-state of the **netX Configuration Tool.**

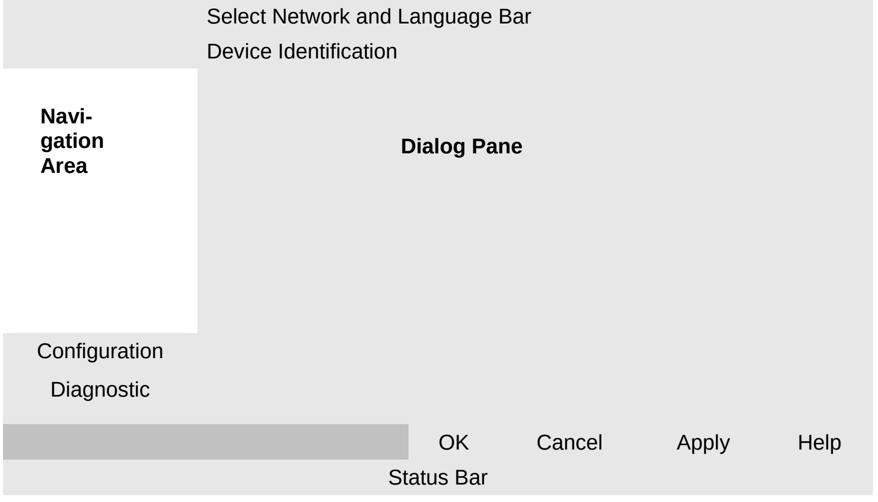


*Figure 2: Dialog Structure of **netX Configuration Tool***

4.1.1 Installation and Configuration Steps for DIL-32 Communication ICs of the NIC 50/52 Series

In order to configure a netIC Communication IC, an Evaluation Board is required. All current NIC50/52 types can be configured with the NICEB Evaluation Board.

The following table describes the typical steps to configure a netIC Communication IC using the NICEB Evaluation Board.

#	Step	Description	For detailed information see section	Page
1	Hardware Installation netIC on Evaluation Board NICEB	Installing the netIC DIL-32 Communication IC on the Evaluation Board: ➢ Disconnect the power line of the Evaluation Board NICEB. ➢ If you want to use the NIC 50-RE or the NIC 52-RE, then **set** all 8 Jumpers on X4. (See *Figure 46* on page *148*) ➢ If you want to use the NIC 50-COS, NIC 50-DNS or NIC 50-DPS, then **remove** all 8 jumpers on X4! Plug a suitable adapter on X4 (Connector: NICEB-AIF-DP for NIC 50-DPS, NICEB-AIF-CO for NIC 50-COS respectively NICEB-AIF-DN for NIC 50-DNS) and screw the adapter from the bottom of the Evaluation-Boards NICEB. ➢ Plug in carefully and mount the netIC Communication IC device into the DIL-32 socket X1 of the Evaluation Board NICEB. Make sure that the tag on the netIC Communication IC device matches the tag of the DIL-32 socket X1 of the NICEB evaluation board. See *Table 12*! ➢ Connect the diagnostic port of the evaluation board NICEB to the COM port (RS232) of the PC using the diagnostic cable CAB-SRV. ➢ Connect the power supply to the evaluation board. Then the SYS LED on the corner of the netIC will permanently show green light.	Device drawing: *Device Drawing Evaluation Board NICEB* Mounting the Adapter NICEB-AIF: *Mounting the Adapter NICEB-AIF*	*147* *39*
2	netX Configuration Tool Installation	➢ Run the installation program from the CD delivered with the netIC to install the **netX Configuration Tool**. Choose the first entry of the installation programs menu (*"netX Configuration and Diagnostic Utility"*). Follow the installation instruction on page 44 and the following.	Installation Instructions for netX Configuration Tool: *Short Description of netX Configuration Tool Installation*	*44*

For more see next page.

#	Step	Description	For detailed information see section	Page
3	Activate the configuration mode	➢ Check whether configuration mode is active. The FBLED on the NICEB indicates that the configuration mode is active by a regular red blinking with a frequency of 1 Hz. ➢ Firmware versions 1.3.0.0 and higher: Activation of the configuration mode is done automatically. Button T3 on the Evaluation Board does not have any influence. ➢ Firmware versions prior to 1.3.0.0: If configuration mode is not active, press the button T3 on the Evaluation Board in order to activate the configuration mode. **Note:** Activate the configuration mode first and then start the netX Configuration Tool and not vice versa. Be aware that Modbus RTU communication is inhibited in configuration mode.	*Status LEDs* *Switches/Push Buttons*	*150* *149*
4	Starting the netX Configuration Tool	➢ Select **Start > Programs > Hilscher GmbH > netX Configuration Tool**	*(See Operating Instruction Manual netX Configuration Tool for netIC 50)*	
5	Selecting the Language	➢ Select in the **Select Language Icon Bar** the language icon for the desired language of the graphical user interface.	*(See Operating Instruction Manual netX Configuration Tool for netIC 50)*	
6	Selecting the Firmware Protocol	➢ Select in the **Select Network Icon Bar** the firmware button for the firmware (Slave device) you intend to use with your device. If all firmware symbols are grayed out: ➢ Make sure once more, the device is operational. ➢ Check, if the diagnostic cable is connected correctly (see step 1) and the configuration mode (see step 3) is activated. ➢ Right click to the navigation area. ➢ Select the context menu **Reload**, to reestablish a connection to the device.	*(See Operating Instruction Manual netX Configuration Tool for netIC 50)*	
7	Setting the Parameters	➢ Click in the navigation area to the **Configuration** push button. ➢ Set the configuration parameters for the slave to be used. If you are not sure about the meaning of a single configuration parameter, we recommend to read the respective documentation or to choose the default value.	*(See Operating Instruction Manual netX Configuration Tool for netIC 50)*	
8	Configuring the Modbus, SSIO, Data Mapping Parameters (device-dependent)	➢ Click in the navigation area to the push button **Modbus RTU**, **Sync. Serial IO** or **Data Mapping**. ➢ Set the configuration parameters for Modbus RTU, Sync. Serial IO or Data Mapping.	*(See Operating Instruction Manual netX Configuration Tool for netIC 50)*	

For more see next page.

#	Step	Description	For detailed information see section	Page
9	Downloading and save the Firmware and the Configuration	➢ Click to **Apply**. ➯ The firmware and the configuration are downloaded to the device. ➯ The configuration is saved to the device. This may last some seconds. ➯ Signaling of the COM LED will change after successful configuration (details depend from the communication system).	*(See Operating Instruction Manual netX Configuration Tool for netIC 50)* *LED*	 *49*
10	**Only if you intend to use the integrated WebServer of the NIC 50-RE or the NIC 52-RE:** Copy files for the WebServer	➢ Copy files for the WebServer from the netIC installation DVD (Directory `Example and API\1. WebServer pages\Common\PORT_1\SX\PUB`) to the netIC, preferably to the directory `/sx/pub`. This can be accomplished either by the WebServer or by the FTP Server. Omitting this step will lead to inoperational WebServer due to missing files.	*(See documentation of WebServer/ FTP Server and list of files on page 66)*	*66*
11	**Only for Real-Time Ethernet NIC 50-RE or NIC 52-RE:** Configure the Real Time Ethernet and connect it to the NICEB	A suitable master for Real-Time Ethernet communication (with electrical interface) is necessary: ➢ Configure this master device ➢ Connect this master device to the Ethernet port of the NICEB using an Ethernet cable (RJ45)	-	-
12	**Only for Field bus NIC 50-COS, NIC 50-DNS, NIC 50-DPS:** Configure the Field Bus Master and connect it to the NICEB	A suitable master for Fieldbus communication is necessary: ➢ Configure this master device ➢ Connect with a suitable Fieldbus cable this master device with the Field bus port of the NICEB using the correct NICEB-AIF adapter for the Fieldbus and a suitable cable according to the cable specifications of the respective Fieldbus system.	-	-
13	Starting the Communication and checking the Diagnostic Data	➢ Click to **Diagnostic** in the navigation area. ➢ Click to **Start**. The communication to the Master is started. ➢ Check the device communication with help of the displayed diagnostic data. Open the extended **Diagnostic** pane: ➢ Click **Extended >>**. **Note!** In diagnostic mode the following restrictions have to be taken into account: • Diagnosis is only possible on the Fieldbus side of the netIC. A connection using the diagnostic interface interrupts the communication to the Modbus side. • In diagnostic mode the ***Output LEDs DO0-DO15*** are not serviced and the DIP switches are not read out.	*(See Operating Instruction Manual netX Configuration Tool for netIC 50)*	
14	How to quit the netX Configuration Tool	➢ **OK** or **Cancel** to quit the **netX Configuration Tool**.	*(See Operating Instruction Manual netX Configuration Tool for netIC 50)*	

15	Deactivate the configuration mode	➢ Only for firmware versions prior to 1.3.0.0: On the Evaluation Board, press the button T3 already mentioned before in step 3 to deactivate the configuration mode. The FBLED on the NICEB will turn off then indicating that the configuration mode is inactive. **Note:** Only when configuration mode is inactive, the Modbus RTU communication is possible.	*Status LEDs* *Switches/Push Buttons*	*150* *149*
16	Modbus RTU	For performing Modbus RTU communication a Modbus-RTU-Master is necessary: ➢ As Modbus RTU Master for example the program ModScan32 (with costs) can be used. ➢ Select the type of serial interface (RS232, RS422 or RS485) on the evaluation board by setting the corresponding jumpers. ➢ Connect the Modbus RTU Master to the RS232-/422-/485 host interface of the NICEB with a suitable cable.	*Host Interface Connector and Hardware Interface Configuration*	*152*

Table 13: Installation and Configuration Steps for DIL-32 Communication IC Devices of the NIC 50 Series and NIC 52 Series

4.2 Mounting the Adapter NICEB-AIF

A suitable adapter NICEB-AIF is required for the usage of NIC50-COS, NIC 50-DNS and NIC 50-DPS with the Evaluation Board NICEB. The following table shows which adapter is required for which DIL-32 Communication IC module.

netIC	Suitable Adapter
NIC 50-COS	NICEB-AIF-CO
NIC 50-DNS	NICEB-AIF-DN
NIC 50-DPS	NICEB-AIF-DP

Table 14: netIC Fieldbus DIL-32 Communication IC and suitable Adapter NICEB-AIF

These adapters are only available in the netIC Evaluation Board Connector Kit NICEB-CONKIT (Hilscher article number 1541.001) as a set of one of each type mentioned in *Table 14: netIC Fieldbus DIL-32 Communication IC and suitable Adapter NICEB-AIF*.

NOTICE

Device Destruction!

- When using the NICEB Evaluation Board with the Fieldbus-Versions of the netIC Gateways NIC 50-COS, NIC 50-DNS respectively NIC 50-DPS:
- Remove the jumpers X4 on the NICEB. Setting the X4 jumpers would cause a **short circuit**!
- Therefore, **never** use a netIC **Fieldbus** Communication IC within the NICEB **with the Ethernet jumpers X4 set**!

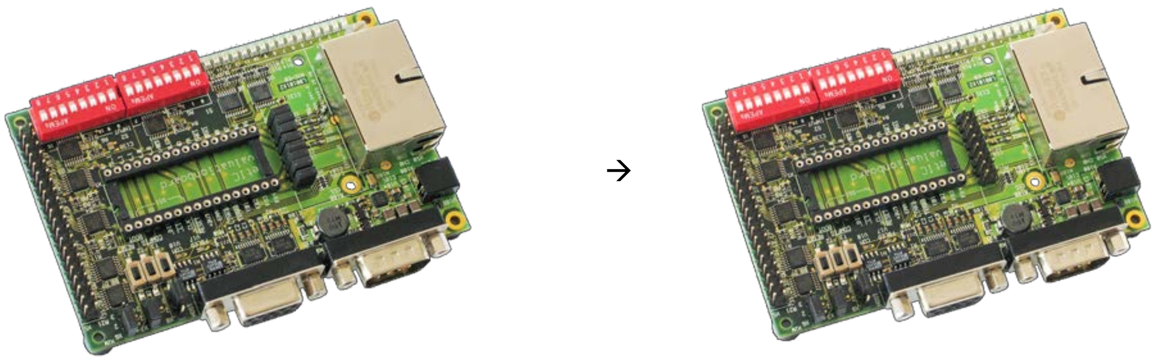


Figure 3: NICEB: Remove Jumpers X4

In order to mount the adapter onto the evaluation board NICEB, proceed as follows:

➢ First remove all the jumpers on X4!

➪ Your NICEB Evaluation Board should then look like this:

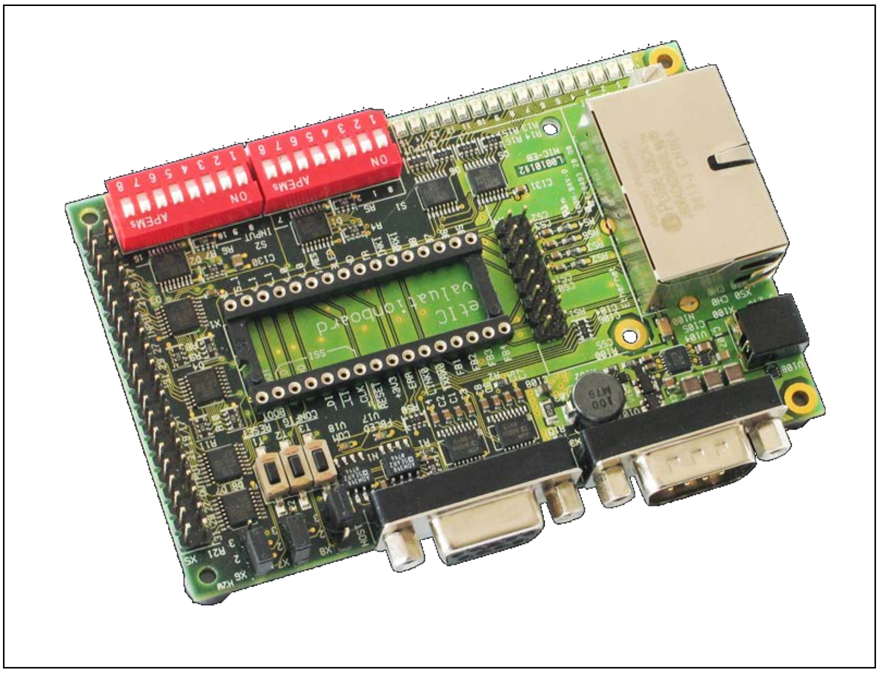


Figure 4: NICEB without netIC Communication IC Module and without Jumpers/Adapter

- ➢ Plug the adapter onto connector X4. Please note that all 16 pins of the adapter are connected correctly on the 16 pins of the connector X4!
- ➢ In order to fix the adapter, screw it from the rear side of the evaluation-board NICEB.
- ➢ You can then mount the netIC onto the DIL-32 socket of the evaluation board NICEB. Take care of the correct orientation of the netIC when inserting into the DIL-32 socket. The tag on the socket and the one on the netIC must match, see *Table 12: Position of the tag of the netIC devices* on page 34. If the evaluation board is oriented with the red switches on top as shown in the photos, you will find the tag on the left side of the DIL-32 socket.
- ➩ Your NICEB Evaluation Board should now look like the left picture in the respective picture row in *Figure 5: NICEB with Adapter* on page 41. (The order in is as follows: CANopen, DeviceNet and PROFIBUS-DP)
- ➩ The evaluation board NICEB is now mounted completely.

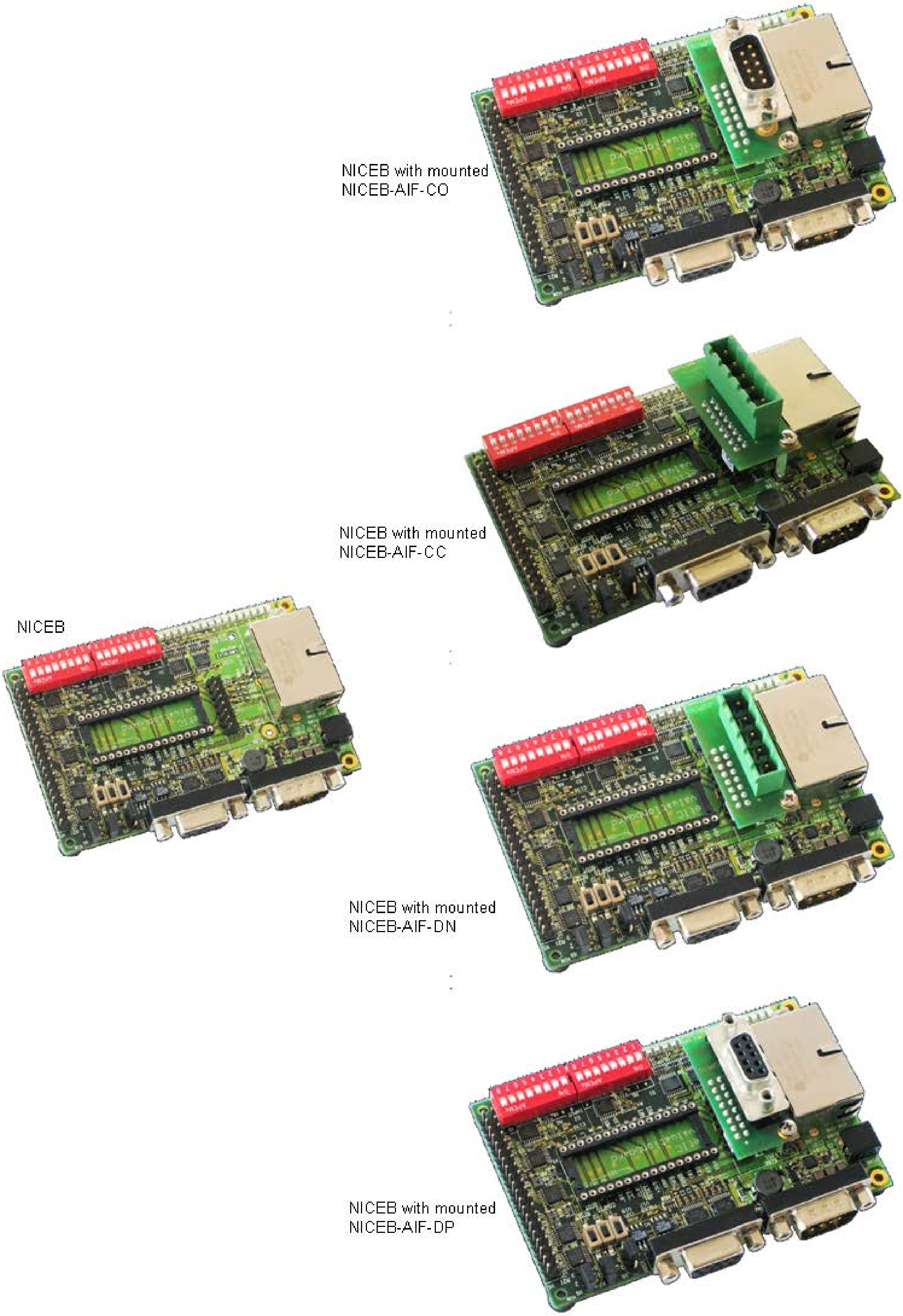


Figure 5: NICEB with Adapter

5 Installing the netIC DIL-32 Communication IC

5.1 Installation of netIC Communication IC into the Target Environment

For the installation of the netIC Communication IC into its target environment respectively the device into which the netIC Communication IC is to be integrated (denominated as host device in the following), proceed as follows:

1. **Step 1**: In order to avoid damage or destruction, adhere to the necessary safety precautions for components that are vulnerable by electrostatic discharge.

NOTICE

Electrostatically sensitive Devices

- To prevent damage to the host system and the netIC Communication IC, make sure, that the netIC Communication IC is grounded and the host system and make sure, that you are discharged when you mount/dismount the netIC Communication IC.

USA:

NOTICE

Electrostatically sensitive Devices

- To prevent damage to the host system and the netIC Communication IC, make sure, that the netIC Communication IC is grounded and the host system and make sure, that you are discharged when you mount/dismount the netIC Communication IC.

2. **Step 2**: If necessary, remove the housing of the host device according to the documentation supplied by the manufacturer of the host device. Obey strictly to the respective instruction manual for this device.

WARNING!

Lethal Electrical Shock caused by parts with more than 50V!

- HAZARDOUS VOLTAGE may be inside of the device into which the netIC Communication IC is to be integrated.
- First disconnect the power plug of the device into which the netIC Communication IC is to be integrated.
- Make sure, that the device is really free of electric power.
- Avoid touching open contacts or ends of wires!
- In any case, strictly adhere to the instructions given in the documentation of the device provided by its manufacturer.
- Open the device and install the netIC Communication IC only after having completed all of the preceding steps.

(Safety instruction for USA see next page)

USA:

⚠ WARNING

Lethal Electrical Shock caused by parts with more than 50V!

- HAZARDOUS VOLTAGE may be inside of the device into which the netIC Communication IC is to be integrated.
- First disconnect the power plug of the device into which the netIC Communication IC is to be integrated.
- Make sure, that the device is really free of electric power.
- In any case, strictly adhere to the instructions given in the documentation of the device provided by its manufacturer.
- Avoid touching open contacts or ends of wires!
- Open the device and install or remove the netIC Communication IC only after having completed all of the preceding steps.

3. **Step 3**::Plug in the netIC Communication IC device carefully but firmly to the DIL-32 socket intended to be used.
4. **Step 4**: Close the housing of host device carefully, if you opened it before. Again, adhere to the documentation supplied by the host device>'s manufacturer.
5. **Step 5**: Connect the host device to the power supply and then switch on the host device. Adhere of the commissioning rules of the supplier of the device. Check, whether the device behaves normally.
6. For deinstallation or replacement of the netIC see chapter 19 "*Decommissioning, Deinstallation, Replacement and Disposal*" on page 194.

6 Installing Software

6.1 Installing the netX Configuration Tool

6.1.1 Preconditions

Configuration and diagnosis of the netIC Real Time Ethernet and Fieldbus DIL-32 Communication IC devices are performed using the NICEB evaluation board and a Windows-based PC running the **netX Configuration Tool** communicating with the evaluation board via a serial connection.

The conditions which are necessary in order to work with the **netX Configuration Tool** are described in section *Preconditions for Usage of netIC Communication ICs together with Evaluation Boards NICEB* on page 31.

➮

6.1.2 Short Description of netX Configuration Tool Installation

The installation itself is then performed as follows:

Start the netX Configuration Tool setup program to install the **netX Configuration Tool**:

Therefore:

- ➢ Close all application programs on the PC!
- ➢ Insert the CD delivered with the NICEB device to the local CD ROM drive of the PC.
- ➮ The GUI of the CD starts.
- ➢ Start in the menu **netX Configuration and Diagnostic Utility** the netX Configuration Tool setup program and follow the installation steps according to the instructions on the screen.

Or:

- ➢ Select with the File Explorer **netX Configuration Tool** of the auto start menu and execute the installation steps according to the instructions on the screen.
- ➮ The **netX Configuration Tool** is installed.

6.1.3 Operating Instruction Manual and Online Help

A description of the user interface of the configuration program **netX Configuration Tool** and for configuration and diagnosis of netIC Communication ICs using this tool, see the **Operating Instruction netX Configuration Tool for netIC, Configuration of Real-Time Ethernet and fieldbus Communication ICs** (`netIC Configuration by netX Configuration Tool OI XX EN.pdf`)on the netIC CD to your device or on www.hilscher.com.

The **netX Configuration Tool** contains an integrated online help facility.

- ➢ To open the online help in **netX Configuration Tool**, click on the **Help** button or press the **F1** key.

6.2 Uninstalling the netX Configuration Tool

To uninstall the **netX Configuration Tool**:

- ➢ Select **Start > Control Panel > Software**
- ➢ Press the button Remove in the list beside the entry **netX Configuration Tool**.
- ➢ Answer the following question with yes.
- ➪ The **netX Configuration Tool** will be removed.

7 Configuration

7.1 Configuration methods

The netIC requires a configuration to operate in communication mode. The configuration contains protocol-independent and protocol-dependent settings.

Configuration method	Remark	Documentation
Configuring the netIC with netX Configuration Tool.	To configure netIC with netX Configuration Tool, ist he standard method and the easier of both methods.	Operation instructions manual: netX Configuration Tool for netIC File: netIC Configuration by netX Configuration Tool OI xx EN
Host application configurs the netIC.	netIC operates as Modbus Slave. The host application writes the Configuration parameter into the Modbus registers oft he netIC.	Application Note: Protocol Parameter via Modbus File: Protocol Parameter via Modbus AN xx EN

Table 15: Configuration methods

7.2 Configuration mode

As soon as the firmware can built-up a connection via the diagnostic interface to the netX Configuration Tool, the firmware activates the configuration mode. For this purpose, the netX Configuration Tool sends periodically requests tot he netIC firmware.

The FBLED on the Evaluation Board shows, whether the netIC is in configuration mode or not. Section FBLED on page 64 describes the states of this LED.

Notes

- The netX Configuration Tool and the firmware are designed for offline mode and therefore do not support online diagnosis.
- Only if the configuration mode is inactive, the Modbus RTU communication is possible.

8 Performance and Response Time

A basic information is how fast I/O data can be exchanged with netIC. The host interface of the netIC is a serial UART or SPI interface. The maximum speed of UART is 115 kBaud and of SPI 1 MHz. Naturally, the performance for I/O exchange of netIC via the serial interface is slower than that of a dual port memory interface. That is why the netIC is usable only for non-time-critical applications.

Several facts influence a statement of the performance of the netIC host interface, e.g.

- the amount of data that should be read or written to the netIC
- the used baud rate
- the used interface type (i.e. UART or SPI)
- including CRC in the frame or not (only valid for SPI)
- the used field bus protocol

The time for transmission and reception can be calculated deterministically by the used baud rate and the length of the transmission and reception frame in bytes.

A non deterministic time is the reaction time of the netIC. The reaction time is how long the netIC needs when it has received the last byte of request telegram until it starts to send the first byte of the response telegram. This reaction time depends from the used interface type UART or SPI. The reaction time covers the internal processing of a request and the preparation of a response frame.

The processing time is influenced by the number of received bytes and the number of bytes to be transmitted. In SPI mode, the internal processing time is also influenced by whether the CRC is included or not (for Modbus RTU the CRC is always included). Another aspect is also the used protocol stack. In general the protocol stack has always priority in processing time to fulfill the network communication. The host interface is of lower priority compared to the protocol stack. Some protocol stacks need more processing time than other ones. This may cause a jitter in the response time.

Table 16: Response Time Distribution of netIC Communication IC depending on applied Protocol below shows results of measurements of the netIC cycle time. It shows how many percent of all requests are processed in a period of time of 2 ms steps.

This test has been made under following conditions:

- Interface type: SPI running with 1 MHz
- Read 32 and Write 32 register with FC32
- Including CRC both in request and response telegram
- Each stack has been in network communication to a master with typical network load.
- The time includes the transmission and reception time of the frame and the processing time of the netIC.

Response time (in ms)	VRS	OMB	EIS	PNS	PNS FO	S3S	ECS	DNS	DPS	COS	PLS
1..2	-	-	-	-	-	-	-	-	-	-	-
3..4	90%	-	90%	90%	90%	90%	90%	-	65%		90%
5..6	10%	50%	10%	10%	10%	10%	10%	-	35%	70%	10%
7..8	-	40%	-	-	-	-	-	60%	-	20%	-
9..10	-	10%	-	-	-	-	-	40%	-	10%	-

Table 16: Response Time Distribution of netIC Communication IC depending on applied Protocol

Additionally tests have shown that including or excluding the CRC into the frame (only for SPI) influences the timing significantly.

9 LED

9.1 SYS LED

The following table describes the meaning of the system LED.

LED	Color	State	Meaning
SYS	**Duo LED yellow/green**		
	(green)	On	Operating System running
	(green/yellow)	Blinking green/yellow	Bootloader is waiting for firmware
	(yellow)	static	Bootloader is waiting for software
	-	Off	Power supply for the device is missing or hardware defect.

Table 17: System LED

The SYS LED is placed in one corner of the netIC Communication IC (see section *Device Drawing NIC 50-RE with Heat Sink* on page 239).

9.2 LED Fieldbus Systems

9.2.1 LED-Names of various Fieldbus Systems

LED	PROFIBUS DP-	CANopen	DeviceNet
System Status (yellow)/ (green)	SYS	SYS	SYS
Communication Status	COM (red / green)	CAN (red / green)	MNS (red / green)

Table 18: Correlation of Signal Names and LED-Names in various Fieldbus Systems

LED	Name	Meaning
System Status	SYS	System
Communication Status	COM	Communication Status
	CAN	CANopen Status
	L RUN/ L ERR	Run/Error
	MNS	Module Network Status

Table 19: Meaning of Signal Names for LEDs

9.2.2 LED PROFIBUS-DP Slave

The subsequent table describes the meaning of the LEDs for the netIC NIC 50-DPS device when the firmware of the PROFIBUS-DP Slave protocol is loaded to the device.

LED	Color	State	Meaning
COM	**Duo LED red/green**		
	(green)	On	RUN, cyclic communication
	(red)	On	Wrong configuration at PROFIBUS-DPside.
	(red)	Flashing cyclic	STOP, no communication, connection error
	(red)	Flashing acyclic	not configured

Table 20: LEDs PROFIBUS DP Slave

9.2.3 LED CANopen Slave

The subsequent table describes the meaning of the LEDs for the netIC NIC 50-COS device when the firmware of the CANopen Slave protocol is loaded to the device.

ED	Color	State	Meaning
CAN Name in the device drawing: **COM**	**Duo LED red/green**		
	(off)	Off	**RESET:** The device is executing a reset
	(green)	Single flash	**STOPPED:** The device is in STOPPED state
	(green)	Blinking	**PREOPERATIONAL:** The device is in the PREOPERATIONAL state
	(green)	On	**OPERATIONAL:** The device is in the OPERATIONAL state
	(red/green)	Flickering (alternatively red / green)	**Auto Baud Rate Detection active:** The Device is in the Auto Baud Rate Detection mode
	(red)	Single flash	**Warning Limit reached:** At least one of the error counters of the CAN controller has reached or exceeded the warning level (too many error frames).
	(red)	Double flash	**Error Control Event:** A guard event (NMT Slave or NMT-master) or a heartbeat event (Heartbeat consumer) has occurred.
	(red)	On	**Bus Off:** The CAN controller is bus off

Table 21: LEDs CANopen Slave

<u>LED State Definition for CANopen Slave for the CAN LED</u>

Indicator state	Definition
On	The indicator is constantly on.
Off	The indicator is constantly off.
Flickering	The indicator turns on and off with a frequency of 10 Hz: on for 50 ms, followed by off for 50 ms.
Blinking	The indicator turns on and off with a frequency of 2,5 Hz: on for 200 ms, followed by off for 200 ms.
Single Flash	The indicator shows one short flash (200 ms) followed by a long off phase (1,000 ms).
Double Flash	The indicator shows a sequence of two short flashes (each 200 ms), separated by a short off phase (200 ms). The sequence is finished by a long off phase (1,000 ms).

Table 22: LED State Definition for CANopen Slave for the CAN LED

9.2.4 LED DeviceNet Slave

The subsequent table describes the meaning of the LED for the netIC NIC 50-DNS device when the firmware of the DeviceNet Slave protocol is loaded to the device.

LED	Color	State	Meaning
MNS	**Duo LED red/green**		
	(green)	On	**Device Operational AND On-line, Connected** Device is online and has established all connections with all Slaves.
	(green)	Flashing (1 Hz)	**Device Operational AND On-line** Device is online and has established no connection in the established state. - Configuration missing, incomplete or incorrect.
	(green/red/off)	Flashing Green/Red/Off	**Selftest after power on:** Green on for 250 ms, then red on for 250 ms, then off.
	(red)	Flashing (1 Hz)	**Minor Fault and/or Connection Time-Out** Device is online and has established one or more connections in the established state. It has data exchange with at least one of the configured Slaves. Minor or recoverable fault: No data exchange with one of the configured Slaves. One or more Slaves are not connected. Connection timeout
	(red)	On	**Critical Fault orCritical Link Failure** Critical connection failure; device has detected a network error: duplicate MAC-ID or severe error in CAN network (CAN-bus off).
	(off)	Off	**Device is not powered** - The device may not be powered. **Device is not on-line and/or No Network Power** - The device has not completed the Dup_MAC_ID test yet. - The device is powered, but the network power is missing.

Table 23: LEDs DeviceNet Slave

<u>LED State Definition for DeviceNet Slave for the MNS LED</u>

Indicator state	Definition
On	The indicator is constantly on.
Off	The indicator is constantly off.
Flashing (1 Hz) green	The indicator turns on and off with a frequency of appr. 1 Hz: on for appr. 500 ms, followed by off for appr. 500 ms.
Flashing (1 Hz) red	The indicator turns on and off with a frequency of appr. 1 Hz: on for appr. 500 ms, followed by off for appr. 500 ms.

Table 24: LED State Definition for DeviceNet Slave for the MNS LED

9.3 LED Real Time Ethernet Systems

9.3.1 LED Names for each Real Time Ethernet System

Note: Depending from the loaded NIC 50-RE firmware the NIC 50-RE LED lines are configured to the corresponding Real-Time Ethernet system.

Pin #	Pin name (NIC)	Pin name (NICEB)	Color of LED	LED Names for NICEB Evaluation Boards							
				CC-Link IE Field Basic	**EtherCAT Slave**	**EtherNet/IP**	**Powerlink**	**Open Modbus/TCP**	**PROFINET IO**	**Sercos**	**V ARAN**
23	STA (green)	LED_COM (red/ green duo LED)	(green)	RUN	STATUS	NS	S/E	COM	SF	S3	COM
10	ERR (red)		(red)	ERR					BF		
11	LINK0n	Ethernet_ Connectors	(green)	L/A	LA_ IN	LINK	LA	LINK	LINK	LA	LINK
12	TXRX0n		(yellow)	-	-	ACT	-	ACT	RX/TX	-	ACT
22	LINK1n	Ethernet_ Connectors	(green)	L/A	LA_OUT	LINK	LA	LINK	LINK	LA	LINK
21	TXRX1n		(yellow)	-	-	ACT	-	ACT	RX/TX	-	ACT

Table 25: LED Names for each Real Time Ethernet System

LED	Name	Meaning
Communication Status	RUN	Run
	ERR	Error
	STA	Status
	SF	System Failure
	BF	Bus Failure
	MS	Module Status
	NS	Network Status
	BS	Bus Status
	BE	Bus Error
RJ45	LINK, L	Link
	ACT, A	Activity
	L/A	Link/Activity
	L/A IN	Link/Activity Input
	L/A OUT	Link/Activity Output

Table 26: Meaning of LED Names

9.3.2 LED CC-Link IE Field Basic Slave

For the CC-Link IE Field Basic Slave protocol, the communication LED **RUN/ERR** as well as the Ethernet LED **L/A** can assume the states described below.

LED	Color	State	Meaning
RUN/ERR (Run/Error) General name: **COM**	**Duo LED red/green**		
	(green)	On	Station in operation and cyclic transmission in progress.
	(green)	Blinking (2.5 Hz)	Station in operation and cyclic transmission stopped.
	(green)	Flickering (10 Hz)	Station not configured.
	(red)	On	Communication error.
	(red)	Triple Flash	DPM watchdog has expired.
	(off)	Off	Station is disconnected.
L/A Ch0 & Ch1	**LED green**		
	(green)	On	**Link:** The station is linked to the Ethernet, but does not send/receive Ethernet frames.
	(green)	Flickering (load dependent)	**Activity:** The station is linked to the Ethernet and sends/receives Ethernet frames.
	(off)	Off	The station has no link to the Ethernet.
Ch0 & Ch1	**LED yellow**		
	(off)	Off	This LED is not used.

Table 27: LED states for the CC-Link IE Field Basic Slave

LED State	Definition
Triple Flash	The indicator shows a sequence of three short flashes (each 200 ms), separated by a short off phase (200 ms). The sequence is finished by a long off phase (1,000 ms).
Blinking (2.5 Hz)	The indicator turns on and off with a frequency of 2.5 Hz: “on” for 200 ms, followed by “off” for 200 ms.
Flickering (10 Hz)	The indicator turns on and off with a frequency of 10 Hz: “on” for 50 ms, followed by “off” for 50 ms.
Flickering (load dependent)	The indicator turns on and off with a frequency of approximately 10 Hz to indicate high Ethernet activity: on for approximately 50 ms, followed by off for 50 ms. The indicator turns on and off in irregular intervals to indicate low Ethernet activity.

Table 28: LED state definitions for theCC-Link IE Field Basic Slave protocol

9.3.3 LED EtherCAT Slave

The subsequent table describes the meaning of the LEDs for the netIC NIC 50-RE device when the firmware of the EtherCAT Slave protocol is loaded to the device.

LED	Color	State	Meaning
STATUS Name in the device drawing: **COM**	**Duo LED red/green**		
	(off)	Off	**INIT**: The device is in state INIT.
	(green)	Blinking	**PRE-OPERATIONAL**: The device is in PRE-OPERATIONAL state.
	(green)	Single Flash	**SAFE-OPERATIONAL**: The device is in SAFE-OPERATIONAL state.
	(green)	On	**OPERATIONAL**: The device is in OPERATIONAL state.
	(red)	Blinking	**Invalid Configuration**: General Configuration Error. Possible reason: State change commanded by master is impossible due to register or object settings.
	(red)	Single Flash	**Local Error**: Slave device application has changed the EtherCAT state autonomously. Possible reason 1: A host watchdog timeout has occurred. Possible reason 2: Synchronization Error, device enters Safe-Operational automatically.
	(red)	Double Flash	**Process Data Watchdog Timeout**: A process data watchdog timeout has occurred. Possible reason: Sync Manager Watchdog timeout.
	(green) (red) (off)	Combinations of red and green: blinking, single and double flash	The status of the red and the green LED can be displayed combined. If for example the Ethernet cable is disconnected, then the following combination is displayed: Green single flash (SAFE-OPERATIONAL) and red double flash (Process Data Watchdog Timeout).
L/A IN/ RJ45 Ch0 **L/A OUT**/ RJ45 Ch1	**LED green**		
	(green)	On	A link is established
	(green)	Flashing	The device sends/receives Ethernet frames
	(off)	Off	No link established
RJ45 Ch0 RJ45 Ch1	**LED yellow**		
	-	-	This LED is not used.

Table 29: LEDs EtherCAT Slave

LED State Definition for EtherCAT Slave for the RUN and ERR LEDs

Indicator state	Definition
On	The indicator is constantly on.
Off	The indicator is constantly off.
Blinking	The indicator turns on and off with a frequency of 2,5 Hz: on for 200 ms, followed by off for 200 ms.
Single Flash	The indicator shows one short flash (200 ms) followed by a long off phase (1,000 ms).
Double Flash	The indicator shows a sequence of two short flashes (each 200 ms), separated by a short off phase (200 ms). The sequence is finished by a long off phase (1,000 ms).

Table 30: LED State Definition for EtherCAT Slave for the RUN and ERR LEDs

9.3.4 LED EtherNet/IP Adapter (Slave)

The subsequent table describes the meaning of the LEDs for the netIC NIC 50-RE device when the firmware of the EtherNet/IP Adapter (Slave) protocol is loaded to the device.

LED	Color	State	Meaning
NS Name in the device drawing: **COM**	**Duo LED red/green**		
	(green)	On	**Connected**: If the device has at least one established connection (even to the Message Router), the network status indicator shall be steady green.
	(green)	Flashing	**No connections**: If the device has no established connections, but has obtained an IP address, the network status indicator shall be flashing green.
	(red)	On	**Duplicate IP**: If the device has detected that its IP address is already in use, the network status indicator shall be steady red.
	(red)	Flashing	**Connection timeout**: If one or more of the connections in which this device is the target has timed out, the network status indicator shall be flashing red. This shall be left only if all timed out connections are reestablished or if the device is reset.
	(red/green)	Flashing	**Self-test**: While the device is performing its power up testing, the network status indicator shall be flashing green/red.
	(grey)	Off	**Not powered, no IP address**: If the device does not have an IP address (or is powered off), the network status indicator shall be steady off.
LINK/RJ45 Ch0 & Ch1	**LED green**		
	(green)	On	A connection to the Ethernet exists
	(off)	Off	The device has no connection to the Ethernet
ACT/RJ45 Ch0 & Ch1	**LED yellow**		
	(yellow)	Flashing	The device sends/receives Ethernet frames

Table 31: LEDs EtherNet/IP Adapter (Slave)

9.3.5 LED Open Modbus/TCP

The subsequent table describes the meaning of the LEDs for the netIC NIC 50-RE device when the firmware of the Open Modbus/TCP protocol is loaded to the device.

LED	Color	State	Meaning
RUN **ERR** Name in the device drawing: **COM**	**Duo LED red/green**		
	(off)	Off	**Not Ready** OMB task is not ready
	(green)	Flashing cyclic with 1Hz	**Ready, not configured yet**: OMB task is ready and not configured yet
	(green)	Flashing cyclic with 5Hz	**Waiting for Communication**: OMB task is configured
	(green)	On	**Connected**: OMB task has communication – at least one TCP connection is established
	(red)	Flashing cyclic with 2Hz (On/Off Ratio = 25 %)	System error
	(red)	On	Communication error active
LINK/RJ45 Ch0 & Ch1	**LED green**		
	(green)	On	A connection to the Ethernet exists
	(off)	Off	The device has no connection to the Ethernet
ACT/RJ45 Ch0 & Ch1	**LED yellow**		
	(yellow)	Flashing	The device sends/receives Ethernet frames

Table 32: LEDs Open Modbus/TCP

9.3.6 LED Powerlink Controlled Node / Slave

The subsequent table describes the meaning of the LEDs for the netIC NIC 50-RE device when the firmware of the Powerlink Controlled Node/Slave protocol is loaded to the device.

LED	Color	State	Meaning
BS/BE Name in the device drawing: **COM**	**Duo LED red/green**		
	-	Off	Slave initializing
	(green)	Flickering	Slave is in Basic Ethernet state
		Single Flash	Slave is in Pre-Operational 1
		Double Flash	Slave is in Pre-Operational 2
		Triple Flash	Slave is in ReadyToOperate
		On	Slave is Operational
		Blinking	Slave is Stopped
	(red)	On	Slave has detected an error
L/A/RJ45 Ch0 & Ch1	**LED green**		
	(green)	On	Link: A connection to the Ethernet exists
	(green)	Flashing	Activity: The device sends/receives Ethernet frames
	-	Off	The device has no connection to the Ethernet
RJ45 Ch0 & Ch1	**LED yellow**		
	(yellow)	-	-

Table 33: LEDs Powerlink Controlled Node/Slave

LED State Definition for Powerlink Controlled Node/Slave for the BS/BE LEDs

Indicator state	Definition
On	The indicator is constantly on.
Off	The indicator is constantly off.
Blinking	The indicator turns on and off with a frequency of approximately 2,5 Hz: on for approximately 200 ms, followed by off for 200 ms. Red and green LEDs shall be on alternately.
Flickering	The indicator turns on and off with a frequency of approximately 10 Hz: on for approximately 50 ms, followed by off for 50 ms. Red and green LEDs shall be on alternately.
Single Flash	The indicator shows one short flash (approximately 200 ms) followed by a long off phase (approximately 1,000 ms).
Double Flash	The indicator shows a sequence of two short flashes (each approximately 200 ms), separated by a short off phase (approximately 200 ms). The sequence is finished by a long off phase (approximately 1,000 ms).
Triple Flash	The indicator shows a sequence of three short flashes (each approximately 200 ms), separated by a short off phase (approximately 200 ms). The sequence is finished by a long off phase (approximately 1,000 ms).

Table 34: LED State Definition for Powerlink Controlled Node/Slave for the BS/BE LED

9.3.7 LED PROFINET IO-RT-Device

For the PROFINET IO-Device protocol, the communication LED **SF/BF** (System Failure, Bus Failure) as well as the Ethernet LEDs **LINK** and **RX/TX** can assume the states described below.

LED	Color	State	Meaning
SF/BF (System Failure, Bus Failure) General name: **COM**	**Duo LED red (BF) / green (SF)**		
	(green)	On	Watchdog timeout or “channel, generic or extended diagnosis present” or system error
	(green, off)	Flashing 1 (1 Hz, 3 s)	DCP signal service is initiated via the bus.
	(red)	On	No configuration or low speed physical link or no physical link
	(red, off)	Flashing 2 (2 Hz)	No data exchange
	(orange, green, red, off)	Flashing 3 (2 Hz, 3 s)	DCP signal service is initiated via the bus. No data exchange
	(orange, red)	Flashing 4 (2 Hz, 3 s)	DCP signal service is initiated via the bus. No configuration or low speed physical link
	(orange, green)	Flashing 5 (2 Hz)	Watchdog timeout or “channel, generic or extended diagnosis present” or system error No data exchange
	(orange)	On	Watchdog timeout or “channel, generic or extended diagnosis present” or system error No data exchange
	(off)	Off	No error
LINK Ch0 & Ch1	**LED green**		
	(green)	On	The device is linked to the Ethernet.
	(off)	Off	The device has no link to the Ethernet.
RX/TX Ch0 & Ch1	**LED yellow**		
	(yellow)	Flickering (load dependent)	The device sends/receives Ethernet frames.
	(off)	Off	The device does not send/receive Ethernet frames.

Table 35: LED states for the PROFINET IO-Device protocol

LED state	Definition
Flashing 1 (1 Hz, 3 s)	The indicator turns on and off with a frequency of 1 Hz for 3 seconds: “green” for 500 ms, followed by “off” for 500 ms.
Flashing 2 (2 Hz)	The indicator turns on and off with a frequency of 2 Hz: “red” for 250 ms, followed by “off” for 250 ms.
Flashing 3 (2 Hz, 3 s)	The indicator changes colors with a frequency of 2 Hz: “orange” for 250 ms, followed by “green” for 250 ms, “red” for 250 ms, followed by “off” for 250 ms.
Flashing 4 (2 Hz, 3 s)	The indicator changes colors for 3 seconds with a frequency of 2 Hz: “orange” for 500 ms, followed by “red” for 500 ms.
Flashing 5 (2 Hz)	The indicator changes colors with a frequency of 2 Hz: “orange” for 250 ms, followed by “green” for 250 ms.
Flickering (load dependent)	The indicator turns on and off with a frequency of approximately 10 Hz to indicate high Ethernet activity: on for approximately 50 ms, followed by off for 50 ms. The indicator turns on and off in irregular intervals to indicate low Ethernet activity.

Table 36: LED state definitions for the PROFINET IO-Device protocol

9.3.8 LED Sercos Slave

The subsequent table describes the meaning of the LEDs for the netIC NIC 50-RE device when the firmware of the sercos-Slave protocol is loaded to the device.

LED	Color	State	Meaning
S3 Name in the device drawing: **COM**	**Duo LED red/green/orange** (orange = red/green simultaneously)		
	(off)	Off	**NRT-Mode**: No sercos Communication
	(green)	On	**CP4**: Communication phase 4, Normal operation, no error
	(green)	Flashing (2 Hz)	**Loopback**: The network state has changed from „fast-forward“ to „loopback“.
	(green/ orange)	Flashing (1 x green / 3 s)	**CP1**: Communication phase 1: Flashing green for 250 ms, then orange on for 2 second and 750 ms
		Flashing (2 x green / 3 s)	**CP2**: Communication phase 2: Flashing green / orange / green, each for 250 ms, then orange on for 2 seconds and 250 ms
		Flashing (3 x green / 3 s)	**CP3**: Communication phase 3: Flashing green / orange / green / orange / green, each for 250 ms, then orange on for 1 second and 750 ms
	(orange/ green)	Flashing (2 Hz)	**HP0**: Hot-plug mode (not yet implemented): Flashing orange/green permanently at 2 Hz.
		Flashing (1 x orange / 3 s)	**HP2**: Hot-plug mode (not yet implemented): Flashing orange for 250 ms, then green on for 2 second an 750 ms
		Flashing (2 x orange / 3 s)	**HP3**: Hot-plug mode (not yet implemented): Flashing orange / green / orange, each for 250 ms, then green on for 2 seconds and 250 ms
	(orange)	On	**CP0**: Communication phase 0
	(orange)	Flashing (2 Hz)	**Identification**: Corresponds to C-DEV.Bit 15 in the Slave's Device Control indicating remote address allocation or configuration errors between Master and Slaves (for details refer to sercos Slave V3 Protocol API Manual).
	(green/ red)	Flashing (2 Hz), *The LED flashes at least for 2 seconds from green to red.*	**MST losses ≥ (S-0-1003/2)**: Depends on IDN S-0-1003 (for details refer to sercos Slave Protocol API manual). Corresponds to S-DEV.Bit 15 in the Device Status indicating a communication warning (Master SYNC telegrams have not been received)
	(red /orange)	Flashing (2 Hz)	**Application error (C1D)**: See GDP & FSP Status codes class error. See sercos Slave V3 Protocol API Manual.
	(red)	Flashing (2 Hz)	**Watchdog error**: Application is not running (not yet implemented)
	(red)	On	**Communication Error (C1D)**: Error detected according to sercos Class 1 Diagnosis, see SCP Status codes class error. See sercos Slave V3 Protocol API Manual.
L/A/RJ45 Ch0 & Ch1	**LED green**		
	(green)	On	Link: A connection to the Ethernet exists
	(green)	Flickering	Activity: The device sends/receives Ethernet frames
	(off)	Off	The device has no connection to the Ethernet
RJ45 Ch0 & Ch1	**LED yellow**		
	-	-	This LED is not used.

Table 37: LEDs sercos Slave

LED State Definition for sercos Slave for the S3 LED

Indicator state	Definition
On	The indicator is constantly on.
Off	The indicator is constantly off.
Flashing (2 Hz)	The indicator turns on and off with a frequency of 2 Hz: first color for appr. 250 ms, followed by the second color for appr. 250 ms.
Flickering	The indicator turns on and off with a frequency of approximately 10 Hz: on for approximately 50 ms, followed by off for 50 ms.

Table 38: LED State Definition for sercos Slave for the S3 LED

9.3.9 LED VARAN Client (Slave)

The subsequent table describes the meaning of the LEDs for the netIC NIC 50-RE device when the firmware of the VARAN Client protocol is loaded into the device.

LED	Color	State	Meaning
RUN/ERR Name in the device drawing: **COM**	**Duo LED red/green**		
	(off)	Off	Not configured.
	(green)	Blinking	Configured and communication is inactive.
	(green)	On	Configured and communication is active.
	(red)	Blinking	Not configured.
	(red)	On	Communication error occurred.
LINK RJ45 Ch0 & Ch1	**LED green**		
	(green)	On	A connection to the Ethernet exists
	(off)	Off	The device has no connection to the Ethernet
ACT RJ45 Ch0 & Ch1	**LED yellow**		
	(yellow)	Flashing	The device sends/receives Ethernet frames

Table 39: LEDs VARAN Client

LED State Definition for VARAN Client for the RUN/ERR LED

Indicator state	Definition
On	The indicator is constantly on.
Off	The indicator is constantly off.
Blinking	The indicator turns on and off with a frequency of 5 Hz: on for 100 ms, followed by off for 100 ms.

Table 40: LED State Definition for VARAN Client for the RUN/ERR LED

9.4 LEDs of the Evaluation Boards

9.4.1 FBLED

The FBLED is mounted at the NICEB evaluation board and triggered by the NIC signal line with the same name (also see section *Status LEDs* on page 150 of this document. It indicates that the NIC is currently in configuration mode or has discovered a module error:

LED	Color	State	Meaning
FBLED	(red)	Regular (cyclic) flash	Indicates that the netIC is in configuration mode and diagnosis can be done.
	(red)	Fast regular (cyclic) flash	Configuration error e.g. overlapping configuration parameter. The fast blinking of the FBLED does not indicate a general error of the firmware, it just indicates that the Memory Mapping or SSIO are configured improperly. In this case the protocol stack can work properly, but its state can be checked by the user only with the COM LED.
	(red)	Irregular (acyclic) flash	Indicates a module error

Table 41: Meaning of FBLED

At netIC modules with PROFINET firmware, the FBLED can be switched in order to display the SYNC signal. This feature is required for PROFINET IO certification. Switching is handled by the host application, see subsection 12.2.6 „*Command Flags (Register 1999)*".

9.4.2 Output LEDs DO0-DO15

Furthermore, the evaluation board NICEB is equipped with 16 LEDs connected to the output signal lines DO0-DO15 of the synchronous serial interface, see *Figure 52: Wiring Diagram of the Serial I/O Shift Register-Interface of the Evaluation Board* on page 156. These LEDs are used for test purposes and emit yellow light.

10 Troubleshooting

In case of any error, please follow the hints given here in order to solve the problem:

General

- ➢ Check, whether the requirements for netIC Real Time Ethernet or Fieldbus DIL-32 Communication IC operation are fulfilled:

  Further information on this topic you can find in section "*Preconditions for Operation of netIC Communication* ICs" on page 30.

SYS-LED

- ➢ Check the status of the SYS LED (in one corner of the NIC 50-RE device). A solid green SYS LED indicates that the firmware of the netIC Communication IC is operational.

LINK-LED (only NIC 50-RE/ NIC 52-RE)

- ➢ Check using the LINK LED status, whether a connection to the Ethernet has been established successfully. Depending on the environment of the netIC Real Time Ethernet DIL-32 Communication IC proceed as follows:
- If the netIC Real Time Ethernet DIL-32 Communication IC is mounted in its target environment: Check signals LINK0n at pin 11 for channel 0 and LINK1n at pin 22 for channel 1, respectively.
- If the netIC Real Time Ethernet DIL-32 Communication IC is mounted in the evaluation board NICEB: Check green LED at Ethernet connector of channel 0 or 1, respectively.

Mounting

- ➢ Check that the netIC Communication IC is mounted correctly in the DIL-32 socket.

Configuration

- ➢ Check the configuration in the master and the slave device. The configuration has to match.

Diagnostic using the netX Configuration Tool (Slave)

With the menu **netX Configuration Tool > Diagnostics** the diagnosis information of the DIL-32 Communication IC is shown. The shown diagnostic information depends on the used protocol.

Note: More information about the device diagnosis and its functions you find in the operating manual of the corresponding Real Time Ethernet system. Therefore refer to section *Available Documentation* page 15.

WebServer

If the WebServer does not operate (especially after firmware update or change), the files of the WebServer may be missing. These are:

- common.js
- device.jpg
- diag.css
- diag.sht
- favicon.ico
- fwupdate.sht
- home.sht
- index.htm
- jquery.js
- LEDColor.png
- logo.png
- menu.prt
- menubg.jpg
- reset.sht
- style.css
- upload.sht

These files are located in the directory *`Example and API\1. WebServer pages\Common\PORT_1\SX\PUB`* of the installation DVD. They have to be copied to the directory `/sx/pub` of the netIC, see *Table 13* on page 38.

11 Updating the Firmware of the netIC DIL-32 Communication IC

On the netIC, by default a boot loader is installed which starts a loaded firmware file from the file system. If the boot loader is active, the SYS LED on the netIC module is blinking alternating between green and yellow. If the firmware is running, the SYS LED is solid green.

11.1 Update by netX Configuration Tool

The netX Configuration Tool check before transferring the parameter, if a suitable firmware is present in the netIC Communication IC. If this is not the case, the netX Configuration Tool downloads a suitable firmware from its firmware pool into the netIC Communication IC device. The firmware files are located in the installation directory of the netX Configuration Tool (Standard: `C:\Program files\Hilscher GmbH\netX Configuration Tool V1.0800`) in the folder `Firmware`.

This method is the standard way.

11.2 Update by WebServer

Alternatively, you can update the firmware of the netIC Communication IC using the integrated WebServer. However, this method is only suitable if there is no change of the chosen Fieldbus or Real-Time Ethernet communication system.

In order to do so, proceed as described in the document "*Functions of the Integrated WebServer in netIC DIL-32 Communication IC Devices*" provided on the DVD delivered along with your netC Communication IC.

(`Functions of the Integrated WebServer in netIC DIL-32 Communication IC AN 01.doc`). See section "*Displaying and Updating Firmware*" in this document.

11.3 Update with ComproX Utility

If no firmware is present or a firmware download has been interrupted (for instance due to power fail during the firmware download), the update must be done manually via the boot loader. Therefore the Boot loader has to be activated on the netIC with the ComproX Utility:
`Tools\ComProX2\GUI\ComProX2_Vx.x.x.x_setup.exe`.

12 Data Model

12.1 Data Model - Overview

The Register Area of the serial Host Interface at the virtual Dual-Port-Memory is the central point connecting all interfaces. This Register Area has a fixed structure and is divided in different Data Areas for the

- Real-Time-Ethernet- or Fieldbus-System,
- the shift registers
- and internal Information-, Configuration- and Status structures.

The Host-System can read and, if write access is allowed (see *Table 43*), also write at all addresses with different amount of data by using Modbus RTU functions.

If the host wants to exchange data over Real-Time-Ethernet or Fieldbus, the host has to write the data at the corresponding place of the RTE Output Data Area respectively the host has to read it out of the RTE-Input Data Area.

The Data of the synchronous serial Interface are also placed in the Register Area to which the Host Interface has access. Should they send over RTE the Gateway-Task has to be configured to copy cyclically these data. The start address is configured with the netX Configuration Tool.

If internal Information and Status data should be available for a RTE-Controller/Fieldbus-Controller they must also copied from the corresponding Register Area into RTE-Output Area. This is also configurable and done by the cyclic Gateway-Task.

When the netIC Communication IC device works as a Modbus RTU Slave, the Modbus RTU Master can read with function code 3 from the register area and write with function code 16 into the register area of the netIC Communication IC device.

Using function code 3 during one single access at maximum 125 Modbus registers can be addressed, using function code 16 120 Modbus registers can be addressed during one single access at maximum.

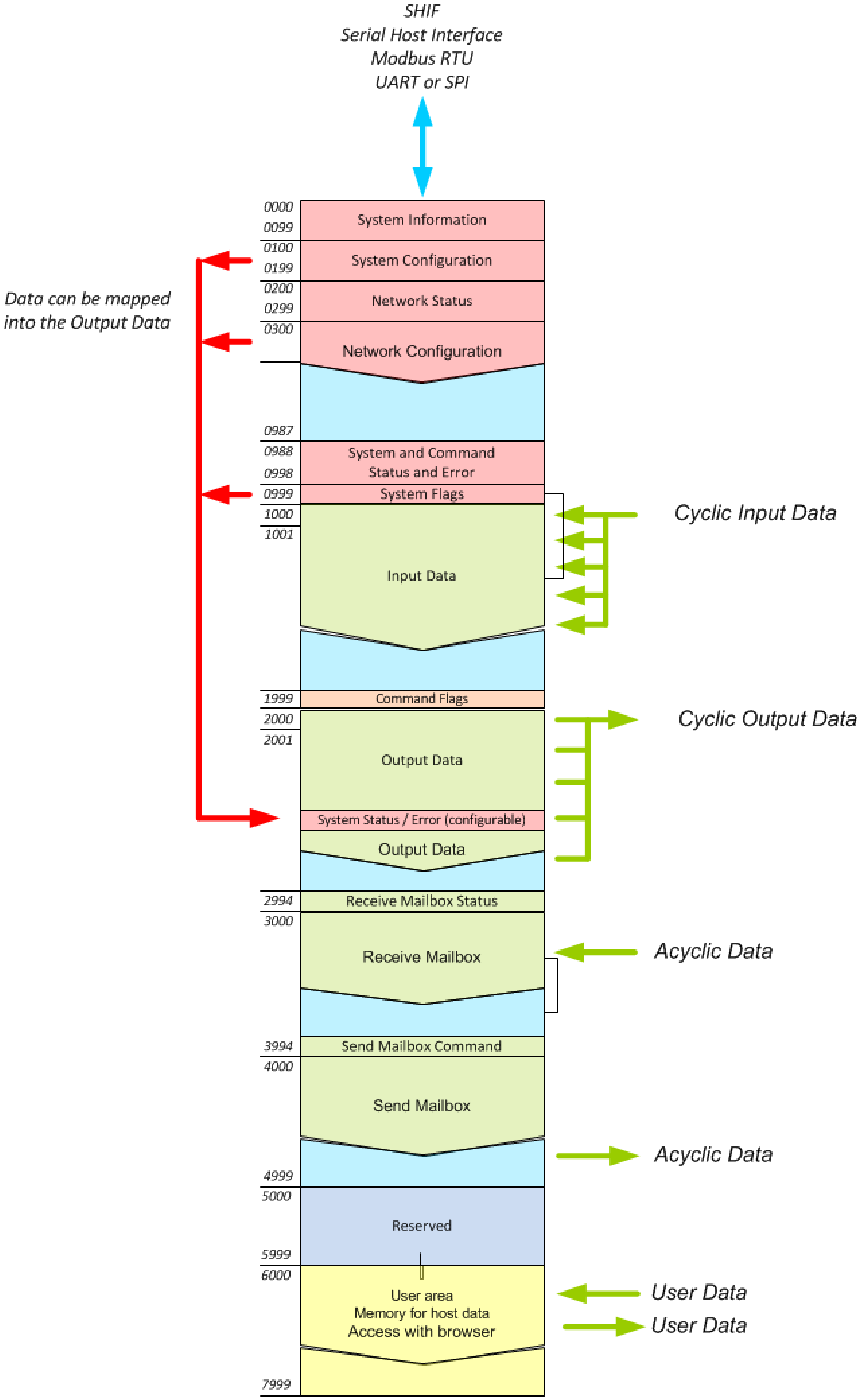


Figure 6: Register Area

shows the Register Area with its different Data Areas. The start addresses are fixed, while the size of input, output and configuration areas depends on the used protocol respectively its configuration.

The host can read the whole Register Area and write at the specific parts of the Register Area where write access is allowed (see most right column of *Table 43:* on page 72), while only the data at the Realtime Ethernet Input and Output Area are exchanged with the Realtime Ethernet Master (similarly for Fieldbus). This access is shown in the figure with arrows named Real-Time-Ethernet cyclic input/output data and Real-Time-Ethernet acyclic data.

The area for Synchronous Serial Data can be configured in the input respectively output data area.

System information, Network Status and System Status can be mapped into the output data area. This is shown in the figure named *'Data can be mapped into the RTE/Fieldbus- Output Data'*.

The red Data Path shows that the System- and Network Data (red area) can be copied in a configurable structure at any place at the RTE-/Fieldbus-Output Area.

With these mechanisms it is possible to generate an application specific Data Model of the Host System to the input and output data areas. Unused areas are initialized with the value 0.

Addressing the Registers in the Modbus Telegram and in SPI

Figure *Register Area* on page 69 shows the addresses of the registers **starting with 0**. These addresses have to be used both in the Modbus RTU telegramm and when using SPI.

Addressing the Registers on Application Level

In practice, at Modbus Master systems various addressing methods are applied **on application level**. In the user interface of the software, the first register accessible with function code 3 or 16 is addressed with 40001, the second one with 40002, and so on. This is the most frequently used kind of addressing. The software within the Modbus-RTU Master internally converts the address 40001 to the value 0 prior to sending the master telegram to the Modbus Slave.

Table 42: Mapping of register addresses (various Modbus RTU Master) shows up further commonly used kinds of addressing:

Register address (Function codes 3 and 16)				**Register address**
Modbus-Master with typical addressing	Modbus-Master with extended addressing	Modbus-Master Adressierung startet mit 0	Modbus-Master Adressierung startet mit 1	Register in the telegram and in the netIC
40001	400001	0	1	0
40002	400002	1	2	1
40003	400003	2	2	2
…	…	…	…	…

Table 42: Mapping of register addresses (various Modbus RTU Master)

Read the manual of the used Modbus RTU Master system to find out which addresses the master uses for function codes 3 and 16.

12.2 Register Area

The netIC Communication IC device provides different data areas within the register area as shown in an overview in the following table.

Register	Data type Max. size	Description of register	Details on page	Access
0 … 99	UINT8[200]	System Information See section *System Information Block.*	*74*	read
100 … 199	UINT8[200]	System Configuration See section *System Configuration Block*.	81	read/ write
200 … 299	UINT8[200]	Network Status The Network Status field holds the „Extended Status Block“ defined according to the selected Real-Time Ethernet communication system (protocol stack). See section 3.3.2 “*Extended Status*” of the according manual for more information.		read
300	UINT16	Network Configuration Data Length This register contains the length (i.e. the number of bytes) of the Network configuration stored in the Network Configuration Data field.		read/ write
301 … 987	UINT8[1374]	Network Configuration Data The Network Configuration field holds information defined according to the selected Real-Time Ethernet or Fieldbus communication system (protocol stack). The contents of this area is equal to the data area of a warmstart message(without header) of the chosen Real-Time Ethernet protocol. See the according manual for more information.		
988 … 989	UINT32	System Status The system status field holds information regarding netX operating system rcX. The value indicates the current state the rcX is in. Currently not supported and set to 0.		read
990 … 991	UINT32	System Error See “System Error”	85	
992	UINT16	Error Log Indicator Not supported.		
993	UINT16	Error Counter 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. After power cycling, reset or channel initialization this counter is being cleared again.		
994 … 995	UINT32	Communication Error This field holds the current error code of the communication channel. If the cause of error is resolved, the communication error field is set to zero (= `RCX_COMM_SUCCESS`) again. Error codes depend on the implementation of the protocol stack. They are listed in the Protocol API Manual of the respective protocol stack in chapter *“Status/Error Codes”*.		
996 … 997	UINT32	Communication Status See *Communication State*	85	
998	UINT16	Received Packet Size This register contains the size of the last message received.		
999	UINT16	System Flags See section *System Error*	*86*	
1000 … 1998	UINT8[1998]	Input Data Image See section *System Configuration Block*.	*81*	

Register	Data type Max. size	Description of register	Details on page	Access
1999	UINT16	Command Flags See section *Command* .	*88*	read/ write
2000 … 2993	UINT8[1988]	Output Data Image See section *System Configuration Block*.	*81*	
2994 … 2995	UINT32	Received Packet Command This register contains the command code of the last message received.		read
2996 … 2997	UINT32	Received Packet Error Code This register contains the error code of the last message which indicated an error.		
2998	UINT16	Received Packet Size This register contains the size of the last message received (i.e. the number of bytes). The firmware has to write this value.		
2999	UINT16	Received Packet Identifier This register contains the identifier of the last message received.		
3000 … 3993	UINT8[1988]	Received Packet This area contains the last message received.		
3994 … 3995	UINT32	Send Packet Command This register contains the command code of the last message sent.		read/ write
3996 … 3997	UINT32	Send Packet Error Code This register contains the last error code of the last message sent with an error.		
3998	UINT16	Send Packet Size This register contains the size of the last message sent (i.e. the number of bytes). The host has to write this value (via Modbus).		
3999	UINT16	Send Packet Identifier This register contains the identifier of the last message sent.		
4000 … 4999	UINT8[2000]	Send Packet This area contains the last message sent.		
5000 … 5999	UINT8[1000]	Reserved area This area is reserved for future use.		-
6000 … 7998	UINT8[1999]	User area (for host data, access ia WebServer) This area can be used to read and write own data to the virtual DPM. It is also accessible (read and write) via the Web Server integrated into the netIC firmware V1.5.x.x (and higher).		read/ write
7999	UINT16	Web Server shared memory with host (Sync. Register) This register can be used to synchronize the data access to the Web Server shared memory. It is coupled to the system flag SX_WRITE_IND in the following way: When this register is written, the flag SX_WRITE_IND in the system register will be set. When this register is read, the flag SX_WRITE_IND in the system register will be cleared. For more information, see Table 56: System Flags on page 87.		

Table 43: Register Area

The following rules apply:

- Unused areas are initialized with 0.
- You can access each of these registers externally using a Master via the serial Modbus RTU protocol or via SPI.
- The registers in the address ranges from 0 to 2993 and from 5000 and 7999 can also be accessed via the Web Server.
- The registers with addresses greater or equal to 5000 are only available when using firmware V1.5.x.x or higher.

12.2.1 System Information Block

The System Information Block consists of the following elements:

Start Register	Data Type Max. Size	Description
0	UINT32	Device Number
2	UINT32	Serial Number
4	UINT16	Device Class
5	UINT8	Hardware Revision
5	UINT8	Hardware Compatibility Index
6	UINT16	Hardware Options Channel 0
7	UINT16	Hardware Options Channel 1
8	UINT16	Hardware Options Channel 2
9	UINT16	Hardware Options Channel 3
10	UINT32	Virtual DPM Size
12	UINT16	Manufacturer Code / Manufacturer Location
13	UINT16	Production Date
14-16	UINT8[6]	Ethernet MAC Address (available with firmware 1.4.12.0 or higher)
17-18	UINT8[4]	Reserved
19	UINT16	Firmware V2.0.0.0 or higher: Feature Flags Register Firmware V1.x: Reserved
20	UINT8[8]	Firmware Version of the loaded firmware
24	UINT8[4]	Firmware Date of the loaded firmware
26	UINT8[64]	Firmware Name of the loaded firmware
58	UINT16	Communication Class of the loaded firmware
59	UINT16	Protocol Class of the loaded firmware
60	UINT16	Protocol Conformance Class of the loaded firmware
61-69	UINT8[18]	Reserved
70-74	UINT8[10]	Input Configuration Shift Registers
75-79	UINT8[10]	Output Status Shift Registers
80-99	UINT8[40]	Reserved

Table 44: System Information Block

The elements of the *System Information Block* have the following meaning:

Device Number (Device Identification, Register 0 and 1)

This field holds a device identification or item number.

Example:
A value of 1541420 translates into a device number of "1541.420" denoting a NIC 50-DPS.

If the value is equal to zero, the device number is not set.

Serial Number (Register 2 and 3)

This field holds the serial number of the netX DIL-32 Communication IC. It is a 32-bit value. If the value is equal to zero, the serial number is not set.

Device Class (Register 4)

This field identifies the hardware.

The following hardware device classes have been defined for the netIC Communication ICs depending on the used netX processor:

netIC type	Device Class
NIC 52	`0x0038`

Hardware Revision

This field indicates the current hardware revision of a module. It starts with 1 and is incremented by 1 with every significant hardware change.

Hardware Compatibility Index

The hardware compatibility index starts with zero and is incremented every time changes to the hardware require incompatible changes to the firmware. The hardware compatibility is used by the netX configuration tool before downloading a firmware file to match firmware and hardware. The application shall refuse downloading an incompatible firmware file.

Note: This *hardware compatibility* should not be confused with the *firmware version number*. The firmware version number increases for every addition or bug fix. The hardware compatibility is incremented only if a change makes firmware and hardware incompatible to each other compared to the previous version.

Hardware Options (Register 5)

The hardware options array allows determining the actual hardware configuration on the xC ports. It defines what type of (physical) interface is connected to the netX chip periphery. Each array element represents an xC port (port 0…3) of the netX 50 processor starting with port 0 in the first element.

Virtual DPM Size(Register 5)

This array element represents the size of entire virtual DPM specified in bytes.

Manufacturer Code / Manufacturer Location (Register 12)

The following value is used as manufacturer code / manufacturer location.

Hilscher Gesellschaft für Systemautomation mbH

```
#define RCX_MANUFACTURER_HILSCHER_GMBH    0x0001
```

Production Date (Register 13)

The production date entry is comprised of the calendar week and year (starting in 2000) when the module was produced. Both, year and week are shown in hexadecimal notation. If the value is equal to zero, the manufacturer date is not set.

High Byte	Low Byte
	Production (Calendar) Week (Range: 01 to 52)
Production Year (Range: 00 to 255)	

Example:
If *usProductionDate* is equal to 0x062B, it indicates a production year of 2006 and a production week of 43.

Feature Flags (Register 19)

This register is available starting with firmware version 2.0.0.7.

Bit (Mask)	Description
0 (0x0001)	0 = Function "Busy exception" for Modbus via SPI is not supported by firmware. 1 = Function "Busy exception" for Modbus via SPI is supported by firmware. Function "Busy exception" is available starting with firmware version 2.0.0.7. See section *Busy Exception (05) for high load condition on* Ethernet on page 187.
1 (0x0002)	0 = Clear/Store Configuration for netIC 52 is not supported by firmware. 1 = Clear/Store Configuration for netIC 52 is supported by firmware Function "Clear/Store Configuration for netIC 52" is available starting with firmware version 2.3.0.0.

Table 45: Feature Flags (Register 19)

Firmware Version (Register 20)

The firmware version of the currently loaded firmware.

Firmware Date (Register 24)

The firmware date of the currently loaded firmware.

Firmware Name (Register 26)

The firmware name of the currently loaded firmware.

Note: The first byte in the firmware name represents the length of the firmware name, then the characters follow.

The following values are defined for the various device-firmware combinations.

Device /Firmware	Returned Value of Firmware-Name	Returned Value of Length
NIC 50-COS/COS	CANopen Slave	13
NIC 50-DNS/DNS	DeviceNet Slave	15
NIC 50-DPS/DPS	PROFIBUS Slave	14
NIC 52-RE/CCIBS	CCLink IE Field Basic Slave	27
NIC 50-RE/ECS NIC 52-RE/ECS	EtherCAT Slave	14
NIC 50-RE/EIS NIC 52-RE/EIS	EthernetIP Slave	16
NIC 50-RE/PNS NIC 52-RE/PNS	PROFINET Slave	14
NIC 50-RE/OMB NIC 52-RE/OMB	ModbusTCP	9
NIC 50-RE/PLS	PowerLink Slave	15
NIC 50-RE/S3S NIC 52-RE/S3S	SERCOS III Slave	16
NIC 50-RE/VRS	Varan Slave	11

Table 46: Firmware names and their length

Communication Class (Register 58)

This array element holds further information regarding the protocol stack. It is intended to help identifying the 'communication class' of the protocol.

Code	Symbolic constant	Numeric value
UNDEFINED	RCX_COMM_CLASS_UNDEFINED	0x0000
UNCLASSIFIABLE	RCX_COMM_CLASS_UNCLASSIFIABLE	0x0001
MASTER	RCX_COMM_CLASS_MASTER	0x0002
SLAVE	RCX_COMM_CLASS_SLAVE	0x0003
SCANNER	RCX_COMM_CLASS_SCANNER	0x0004
ADAPTER	RCX_COMM_CLASS_ADAPTER	0x0005
MESSAGING	RCX_COMM_CLASS_MESSAGING	0x0006
CLIENT	RCX_COMM_CLASS_CLIENT	0x0007
SERVER	RCX_COMM_CLASS_SERVER	0x0008
IO-CONTROLLER	RCX_COMM_CLASS_IO_CONTROLLER	0x0009
IO-DEVICE	RCX_COMM_CLASS_IO_DEVICE	0x000A
IO-SUPERVISOR	RCX_COMM_CLASS_IO_SUPERVISOR	0x000B
GATEWAY	RCX_COMM_CLASS_GATEWAY	0x000C
MONITOR/ ANALYZER	RCX_COMM_CLASS_MONITOR	0x000D
PRODUCER	RCX_COMM_CLASS_PRODUCER	0x000E
CONSUMER	RCX_COMM_CLASS_CONSUMER	0x000F
SWITCH	RCX_COMM_CLASS_SWITCH	0x0010
HUB	RCX_COMM_CLASS_HUB	0x0011

Table 47: Possible Values of Communication Class

Other values are reserved.

Protocol Class (Register 59)

This field identifies the protocol stack.

Code	Symbolic constant	Numeric value
UNDEFINED	RCX_PROT_CLASS_UNDEFINED	0x0000
CANopen	RCX_PROT_CLASS_CANOPEN	0x0004
DeviceNet	RCX_PROT_CLASS_DEVICENET	0x0008
CCLink IE Field Basic	RCX_PROT_CLASS_CCLINK_IE_BASIC	0x002C
EtherCAT	RCX_PROT_CLASS_ETHERCAT	0x0009
EtherNet/IP	RCX_PROT_CLASS_ETHERNET_IP	0x000A
Open Modbus TCP	RCX_PROT_CLASS_OPEN_MODBUS_TCP	0x0012
Powerlink	RCX_PROT_CLASS_POWERLINK	0x001A
PROFIBUS DP	RCX_PROT_CLASS_PROFIBUS_DP	0x0013
PROFINET IO	RCX_PROT_CLASS_PROFINET_IO	0x0015
Sercos	RCX_PROT_CLASS_SERCOS_III	0x0018
VARAN	RCX_PROT_CLASS_VARAN	0x0027
OEM, Proprietary	RCX_PROT_CLASS_OEM	0xFFF0

Table 48: Possible Values of Protocol Class

Other values are reserved.

Conformance Class (Register 60)

This field identifies the supported functionality of the protocol stack (PROFIBUS supports DPV1 or DPV2, PROFINET complies with conformance class A/B/C, etc.). The entry depends on the protocol class of the communication channel (see above) and is defined in a protocol specific manual.

Input Configuration Shift Registers (Register 70 to 74)

It is possible to configure up to 10 bytes (i.e. 5 registers) of the SSIO Input data to be evaluated separately and written into the registers 70 to 74. For instance, these can be used by the host for handling rotary address switches connected to the shift registers.

Register 110 determines, how many bytes of the SSIO Input data are actually written into the registers 70 to 74.

If you want to switch off this feature, just set register 110 to 0 indicating no data are written to registers 70-74.

Also see *Figure 7* on page 80.

Output Status Shift Registers (Register 75 to 79)

It is possible to configure up to 10 bytes (i.e. 5 registers) of the SSIO Output data to be processed separately. These data are read out of the registers 75 to 79 and written to the SSIO Output data. For instance, the host can use this for cyclically transferring status information, e.g. when additional LEDs are connected to the shift registers..

Register 111 determines, how many bytes are actually read from registers 75 to 79 and then written to the SSIO Output data.

If you want to switch off this feature, just set register 111 to 0 indicating no data are read from registers 75-79.

Also see *Figure 7* on page 80.

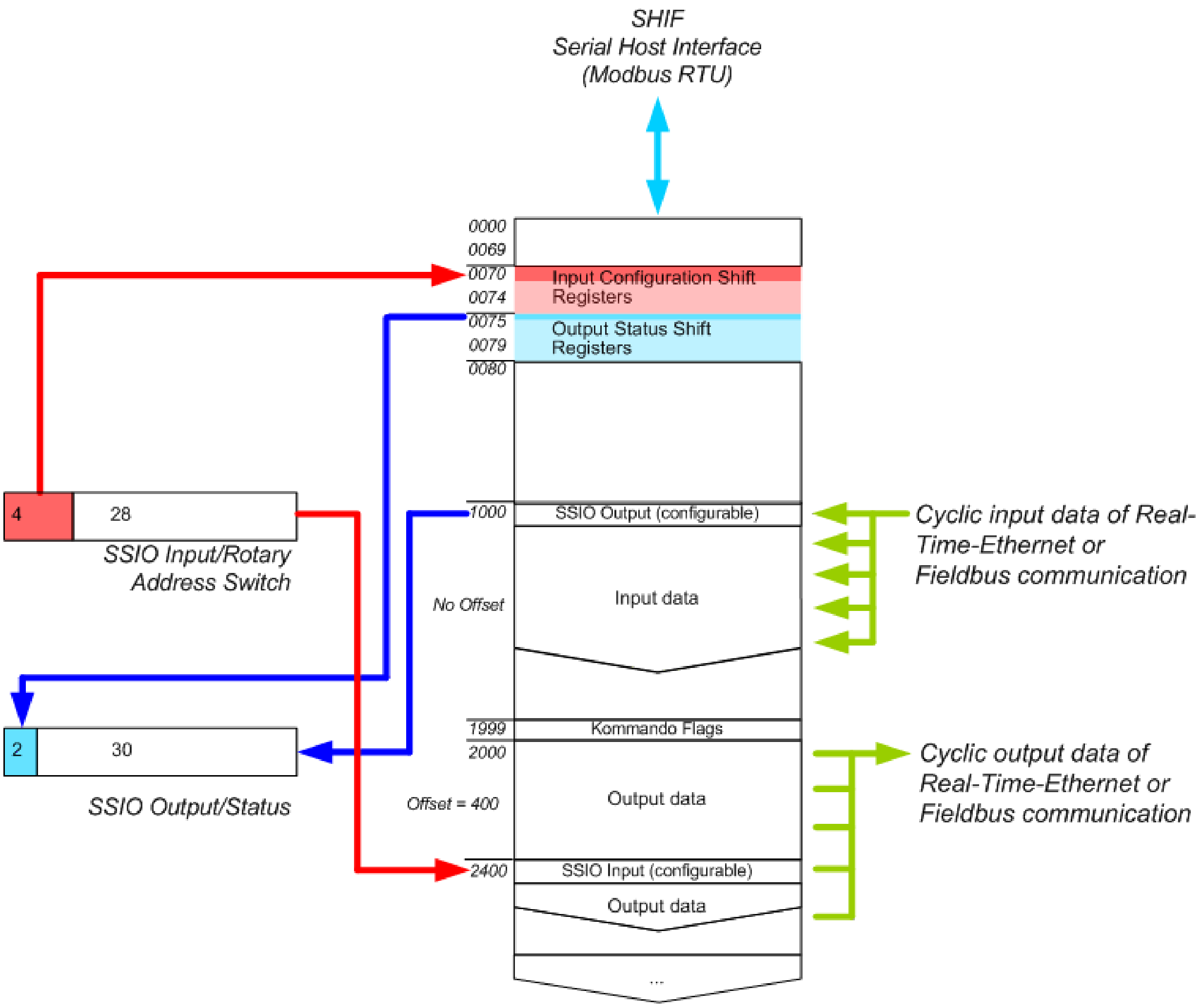


Figure 7: Example Configuration for SSIO Input and Output (SSIO Input: Offset 400, SSIO Output: Offset 0)

12.2.2 System Configuration Block

Name of Register Area	Register#	Data Type Max. Size	Description
SSIO Config	100	UINT16	Type (=SSIO) Set to zero.
SSIO Config	101	UINT16	SSIO Address Set to zero.
SSIO Config	102	UINT32	SSIO Baudrate
SSIO Config	104	UINT16	SSIO Number of Input Bytes: 0 … 256 bytes.
SSIO Config	105	UINT16	SSIO Number of Output Bytes: 0 … 256 bytes.
SHIF Config	106	UINT16	SHIF Type 0 = Modbus RTU / UART 1 = Modbus RTU / SPI All other values are reserved.
SHIF Config	107	UINT16	SHIF Baudrate With SHIF Type = Modbus RTU / UART: - Modbus RTU Baudrate With SHIF Type = Modbus RTU / SPI: - Reserved, set to 0.
SHIF Config	108	UINT16	SHIF Address With SHIF Type = Modbus RTU / UART: - Modbus RTU Address With SHIF Type = Modbus RTU / SPI: - Modbus RTU Address
SHIF Config	109	UINT16	SHIF Configuration flags (see below)
SSIO Mapping	110	UINT16	Number of Bytes from SSIO Input used as Config
SSIO Mapping	111	UINT16	Number of Bytes from SSIO Output used as Status
SSIO Mapping	112	UNIT16	Offset Address in FB Input Data Image
SSIO Mapping	113	UNIT16	Offset Address in FB Output Data Image
SSIO Config	114	UINT16	SSIO Watchdog Time
SSIO Config Swap	115	UINT16	Swap shift direction Bit 0 changes input shift direction. Bit 1 changes output shift direction. For each bit: 0 = default 1 = change byte order
Reserved	116 – 119	UINT16	Reserved
Diagnostic Mapping	120	UINT16	Offset Address in Output Data Image of Diagnostic Data
Diagnostic Mapping	121	UINT16	Number of Mapping Data
Diagnostic Mapping	122 – 199	UINT16 [78]	Mapping Data: ID1, Length1, ID2, Length 2, …

Table 49: System Configuration Block

Configuration of the Serial I/O Shift Registers

These IOs are used as input and output data. They can also be configured via the **netX Configuration Tool** instead of using Modbus RTU for this purpose.

- The data from the configuration shift registers are copied into the System Status Fields once during startup and then cyclically from there to the status shift registers. The network protocol interacts only with the values in the System Status Field.
- The baud rate of the Serial I/O Shift Register interface can be set via Register 102/103. There is also the possibility of an automatic baud rate detection. The following values are available:

Value	Meaning
0	Automatic baud rate detection
500	SPI is used, Serial I/O Shift Register interface thus restricted to 500 Baud
100000	100000 Baud
200000	200000 Baud
500000	500000 Baud
1000000	1000000 Baud
2000000	2000000 Baud
5000000	5000000 Baud

Table 50: Possible Values for the Baud Rate of the Serial I/O Shift Register Interface

- The number of Input Bytes of the Serial I/O Shift Register interface can be set in register 104. The allowed range of values contains all integer values between 0 and 256.
- The number of Output Bytes of the Serial I/O Shift Register interface can be set in register 105. The allowed range of values contains all integer values between 0 and 256.
- The input and output data are copied to the Output Data Image respectively to the Input Data Image. The Offset Address in each area can be configured individually (registers 112 and 113, respectively).
- For the supervision of the Serial I/O Shift Register interface, a watchdog timer is available. It can be activated by writing the watchdog time (a value in the range 20 to 65535, specified in milliseconds) into register 114. Writing the value 0 into register 114 will deactivate the watchdog timer.

For more information on setting these registers via Modbus RTU, see *Application Note: Protocol Parameter via Modbus*, section 3.1.

Also see *Figure 7* on page 80.

Configuration of the Serial Host Interface (SHIF)

The necessary configuration parameters can be set up by "*netX Configuration Tool*" or via Modbus RTU. Register 106 to 109 are used to configure the Serial Host Interface.

The serial host interface can operate in two different modes (SHIF Types):

- Modbus RTU/UART (SHIF Type 0, Register 106 = 0)
- or Modbus RTU/SPI (SHIF Type 1, Register 106 = 1)
- In mode Modbus RTU/ UART, the baudrate can be set via register 107 (using the lower 15 bits). The most significant bit is used as write protection flag. All other bits can be used for the selection of the desired baudrate, see Table 51: Contents of Baudrate Register

Bit	Description
0 .. 14	Baudrate value (x 100) 12 = 1200 Baud 24 = 2400 Baud 48 = 4800 Baud 96 = 9600 Baud 192 = 19200 Baud 384 = 38400 Baud 576 = 57600 Baud 1152 = 115200 Baud
15	Write protection flag In order to prevent switching the baudrate during running operation, there is a write protection flag. 0 - write protection off (deactivated) 1 - write protection on (active)

Table 51: Contents of Baudrate Register

- In mode Modbus RTU/ SPI, Register 107 must be set to 0. In this case, the baudrate is determined automatically by the netIC, the possible upper limit amounts to 1 MHz.
- The Modbus RTU Address (Slave ID) can be set via register 108. The allowed range of values extends from 1 to 247 (integer values).
- Register 109 allows setting the SHIF Configuration Flags, for more information see *Table 52: SHIF Configuration Flags below*:

Bit	Bit-mask	Description	Applicable for SHIF Type
0	0x00000001	PARITY_EVEN	Modbus RTU / UART
1	0x00000002	PARITY_ODD	Modbus RTU / UART
2	0x00000004	RTS_ON	Modbus RTU / UART
4	0x00000010	ENABLE_SWAP	Modbus RTU / UART
5	0x00000020	INCLUDE_CRC_AND_ADDR	Modbus RTU / SPI

Table 52: SHIF Configuration Flags

By default the Modbus RTU parameters are:

- Slave ID = 2;
- Baud rate = 9600;
- Parity = EVEN;
- Stop Bits = 1
- Data Bits = 8

For more information on setting these registers via Modbus RTU, see *Application Note: Protocol Parameter via Modbus*, section 3.2.

Diagnostic Mapping

Very often the master of the network needs some diagnostic information from the connected device. Therefore we can map this information in the Output Data image.

It works as follows:

- The start address in the Output Data Image of the Diagnostic Data and the number are configured.
- Each System Diagnostic Data has a unique ID-Number.
- For each Diagnostic Data is one Mapping Data configured. The order of the Mapping Data defines also the order of the Diagnostic Data in the Output Image.
- The Diagnostic Data are copied cyclically into the Output Image.

The following IDs are predefined in this context:

ID	Length (in byte)	Meaning
0	4	Device number
2	4	Serial number
20	8	Firmware version
24	4	Firmware date
26	64	Firmware name
200	200	Network Status
988	20	System Status, System Error, Error Log Indicator/Error Counter, Communication Error, Communication Status

Table 53: Predefined IDs

Register 122 contains the value of the ID (ID1), register 123 contains the length information of ID1, Register 124 contains the value of the ID (ID2), register 125 contains the length information of ID2, etc. The configuration is done using the **netX Configuration Tool** and is described in the **Operating Instruction Manual netX Configuration Tool for netIC 50**.

Alternatively, configuration via Modbus RTU is possible.

12.2.3 System Error

The system error field holds information about the general status of the netX firmware stacks.

An error code of zero indicates a faultless system. If the system error field holds a value other than *SUCCESS*, the *Error* flag in the *netX System flags* is set.

Code	Symbolic constant	Numeric value
SUCCESS	RCX_SYS_SUCCESS	0x00000000
RAM NOT FOUND	RCX_SYS_RAM_NOT_FOUND	0x00000001
INVALID RAM TYPE	RCX_SYS_RAM_TYPE	0x00000002
INVALID RAM SIZE	RCX_SYS_RAM_SIZE	0x00000003
RAM TEST FAILED	RCX_SYS_RAM_TEST	0x00000004
FLASH NOT FOUND	RCX_SYS_FLASH_NOT_FOUND	0x00000005
INVALID FLASH TYPE	RCX_SYS_FLASH_TYPE	0x00000006
INVALID FLASH SIZE	RCX_SYS_FLASH_SIZE	0x00000007
FLASH TEST FAILED	RCX_SYS_FLASH_TEST	0x00000008
EEPROM NOT FOUND	RCX_SYS_EEPROM_NOT_FOUND	0x00000009
INVALID EEPROM TYPE	RCX_SYS_EEPROM_TYPE	0x0000000A
INVALID EEPROM SIZE	RCX_SYS_EEPROM_SIZE	0x0000000B
EEPROM TEST FAILED	RCX_SYS_EEPROM_TEST	0x0000000C
SECURE EEPROM FAILURE	RCX_SYS_SECURE_EEPROM	0x0000000D
SECURE EEPROM NOT INITIALIZED	RCX_SYS_SECURE_EEPROM_NOT_INIT	0x0000000E
FILE SYSTEM FAULT	RCX_SYS_FILE_SYSTEM_FAULT	0x0000000F
VERSION CONFLICT	RCX_SYS_VERSION_CONFLICT	0x00000010
SYSTEM TASK NOT INITIALIZED	RCX_SYS_NOT_INITIALIZED	0x00000011
MEMORY ALLOCATION FAILED	RCX_SYS_MEM_ALLOC	0x00000012

Table 54: Possible Values of System Error

12.2.4 Communication State

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

Communication State	Symbolic constant	Numeric value
UNKNOWN	RCX_COMM_STATE_UNKNOWN	0x00000000
OFFLINE	RCX_COMM_STATE_OFFLINE	0x00000001
STOP	RCX_COMM_STATE_STOP	0x00000002
IDLE	RCX_COMM_STATE_IDLE	0x00000003
OPERATE	RCX_COMM_STATE_OPERATE	0x00000004

Table 55: Possible Values of Communication State

12.2.5 System Flags (Register 999)

These flags show the current state of the system and the communication of netIC Communication IC.

Bit (Mask)	Description
Bit 0 (0x0001)	READY The Ready flag is set as soon as the operating system has initialized itself properly and passed its self test. When the flag is set, the netX is ready to accept packets via the system mailbox. If cleared, the netX does not accept any packages.
Bit 1 (0x0002)	ERROR The Error flag is set when the netX has detected an internal error condition. This is considered a fatal error. The Ready flag is cleared and the operating system is stopped. An error code helping to identify the issue is placed in the ulSystemError variable in the system control block. The Error flag is not supported yet.
Bit 2 (0x0004)	COMMUNICATING The Communicating flag is set if the protocol stack has an open connection to the network master. If cleared, the input data should not be evaluated, because it may be invalid, old or both.
Bit 3 (0x0008)	NCF_ERROR The Error flag signals an error condition that is reported by the protocol stack. It could indicate a network communication issue or something to that effect. The corresponding error code is placed in the CommunicationError register.
Bit 4 (0x0010)	RX_MBX_FULL This flag shows that the Receive Mailbox contains a Packet. If the Packet is read out the flag will automatically cleared. This flag has to be checked cyclically by the host if a message was received.
Bit 5 (0x0020)	TX_MBX_FULL This flag shows that the Send Mailbox contains a Packet. If the Packet is taken over from the Fieldbus Protocol the flag is automatically cleared. Before sending a Packet this Flag has to be checked if it is zero otherwise it is not allowed to send a Packet.
Bit 6 (0x0040)	BUS_ON This flag reflects the current bus state, if the protocol stack accesses the bus
Bit 7 (0x0080)	FLS_CFG netIC 10 and netIC 50 only: This flag indicates that the netIC is configured by a configuration originating from the flash file system. It will be cleared, as soon as the command flag CLR_CFG is executed. As soon as the command flag STR_CFG is set, the netIC will set this flag. netIC 52 only: This flag indicates that a non-volatile configuration created and downloaded by the netX configuration tool is used to configure the NIC 52. It will be cleared, as soon as the command flag CLR_CFG is executed. As soon as you download a configuration using the netX configuration Tool into the netIC, the netIC will set this flag.
Bit 8 (0x0100)	LCK_CFG This flag indicates, whether the registers which contain any configuration data (network and system configuration data) are write protected or not. This flag will be set and cleared by using the command flags LCK_CFG and UNLOCK_CFG.
Bit 9 (0x0200)	WDG_ON This flag indicates, whether the watchdog function has been activated or not. This flag is set or cleared by the command flags WDG_ON and WDG_OFF

Bit (Mask)	Description
Bit 10 (0x0400)	RUNNING This flag indicates, whether the protocol stack has been configured and the initialization has been completed successfully. The configuration process may last for some seconds depending on the chosen protocol. If the configuration has been completed successfully, the RUNNING flag will be set. The host application can use this flag for synchronization purposes, for instance.
Bit 11 (0x0800)	SX_WRITE_IND This flag will be set when the web server writes to register 7999. It will be cleared when the host reads from register 7999. This flag can be used for synchronization between host and integrated web server.
Bit 12 (0x1000)	REM_CFG netIC 52 only: This flag indicates that a non-volatile configuration is used to configure the netIC 52. Modbus register 100–199 and 300–987 have been modified and stored by the host application before.
Bit 13 … 15	Reserved, set to zero.

Table 56: System Flags

12.2.6 Command Flags (Register 1999)

Writing a Command Flag sets up the appropriate command at netIC Communication IC. If the command is executed the netIC automatically clears this flag. If the Command Flags is set to 0, the netIC ignores it.

Bit (Mask)	Description
Bit 0 (0x0001)	RESET The application can set the Reset flag to execute a system-wide reset of the netIC and forces the system to restart. All network connections are interrupted immediately regardless of their current state. **Note:** The netIC will immediately execute the reset and the netIC firmware does not send a response to the host application. If a new configuration is desired to be applied from the flash memory, this flag must be set.
Bit 1 (0x0002)	BOOT_START Reserved, always 0
Bit 2 (0x0004)	APP_READY Reserved, always 0
Bit 3 (0x0008)	BUS_ON Using the BUS_ON flag, the host application allows the firmware to open network connections. If set, the netIC firmware tries to open network connections. **Note:** If the application sets BUS_ON and BUS_OFF at the same time, BUS_ON has the higher priority.
Bit 4 (0x0010)	INIT Setting the Initialization flag the application forces the protocol stack to restart and evaluate the configuration parameter again. All network connections are interrupted immediately regardless of their current state.
Bit 5 (0x0020)	BUS_OFF Using the BUS_OFF flag, the host application inhibits to open network connections. If set, no network connections are allowed and open connections are closed. **Note:** If the application sets BUS_ON and BUS_OFF at the same time, BUS_ON has the higher priority.
Bit 6 (0x0040)	CLR_CFG If this flag is set, the netIC will clear a non-volatile configuration created and downloaded by the netX Configuration Tool. This configuration data stored in the flash file system. As soon as this operation is finished, the flag FLS_CFG in the system register will be cleared. The netIC continues using the current configuration which is the configuration used before this clear operation was activated. Afterwards a reset of the netIC is required (RESET flag) that the netIC starts up without any configuration and the netIC is ready to be newly configured.
Bit 7 (0x0080)	netIC 52: STR_REM_CFG; netIC 10 and netIC 50: STR_CFG If this bit is set, the netIC will store Modbus register 100–199 and 300–987 into the Flash memory that has been modified by the host application before. Requirement: Before the host application can use the STR_REM_CFG flag, the host application must deleted the "old" configuration (created and downloaded by netX Configuration Tool) using the CLR_CFG flag. Otherwise it will cause an exception to set this flag. Whether a configuration is stored in the flash file system or not is indicated by the system flag FLS_CFG. To activate the stored configuration, the application has to set the INIT or RESET flag or alternatively a power cycle is required.
Bit 8 (0x0100)	LCK_CFG If this flag is set, the netIC causes an exception whenever the user wants to write any configuration register (network and system configuration). The lock status is reflected in the status flag LCK_CFG. **Note:** If the application sets LCK_CFG and UNLOCK_CFG at the same time, UNLOCK_CFG has the higher priority.
Bit 9 (0x0200)	UNLOCK_CFG If this flag is set, the netIC unlocks the write protection to any configuration register. The lock status is reflected in the status flag LCK_CFG. **Note:** If the application sets LCK_CFG and UNLOCK_CFG at the same time, UNLOCK_CFG has the higher priority.
Bit 10 (0x0400)	WDG_ON With this flag the watchdog function of the field bus and the shift register interface is activated. The status whether the watchdog is active or not is stored in the status flag WDG_ON. **Note:** If the application sets WDG_ON and WDG_OFF at the same time, WDG_OFF has the higher priority.

Bit (Mask)	Description
Bit 11 (0x0800)	WDG_OFF With this flag the watchdog function of the field bus and the shift register interface is deactivated. **Note:** If the application sets WDG_ON and WDG_OFF at the same time, WDG_OFF has the higher priority.
Bit 12 (0x1000)	CLR_REM_CFG netIC 52 only: If this flag is set, the netIC will clear a non-volatile configuration. This non-volatile configuration is Modbus register 100–199 and 300–987 that has been modified and stored by the host application before. As soon as this operation is finished, the flag REM_CFG in the system register will be cleared. The netIC continues using the current configuration which is the configuration used before this clear operation was activated. **Note:** The remanent data stored by the stack, e.g. Name of Station for PROFINET IO Device, is not deleted.
Bit 13 (0x2000)	Reserved, set to zero
Bit 14 … 15 (0xC000)	FIELDBUS_SPECIFIC_COMMANDS Only the PROFINET IO Device firmware supports fieldbus specific commands. See subsequent section.

Table 57: Command Flags

If fieldbus-specific commands are used, i.e. bit 14 or 15 of the command flags register 1999 is used, the lower bits 0 to 13 are not evaluated in the usual way. This means, *Table 57: Command Flags* is not applicable for bits 0 to 13 in such cases.

Fieldbus-specific commands

Only the PROFINET IO Device firmware supports the following fieldbus-specific commands.

Code to be entered in Command Flags (Register 1999)	Function
0xB015	`MEMORY_MAP_COMMAND_SET_BUSY_RSP_TIME_500` activates the.busy exception on high Ethernet data load, see section 16.3.6“*Busy Exception (05) for high* load condition on Ethernet
0xB010	`MEMORY_MAP_COMMAND_SET_BUSY_RSP_TIME_000` deactivates the busy exception on High Ethernet data load.
0xC000	`MEMORY_MAP_COMMAND_SET_` activates the packet filter function, see section *Packet Filter Function for PROFINET IO Device* on page *103*.
0xC002	`MEMORY_MAP_COMMAND_SET_` deactivates the packet filter function, see section *Packet Filter Function for PROFINET IO Device* on page *103*.
0xC006	`MEMORY_MAP_COMMAND_PNS_ENABLE_SYNC_TO_FBLED` Activates the SYNC pin. Required for PROFINET IRT certification.
0xC008	`MEMORY_MAP_COMMAND_PNS_DISABLE_SYNC_TO_FBLED` Deactivates the SYNC pin.

Table 58: Fieldbus specific commands

12.3 Cyclic Data

12.3.1 Data Mapping Cyclic Data

The following figure shows: The Modbus RTU Master can read data starting with address 41001 using function code 3. These data were received from the connected Real-Time Ethernet respectively Fieldbus.

Note: The SSIO data are located by default on the first two registers of the input data area.

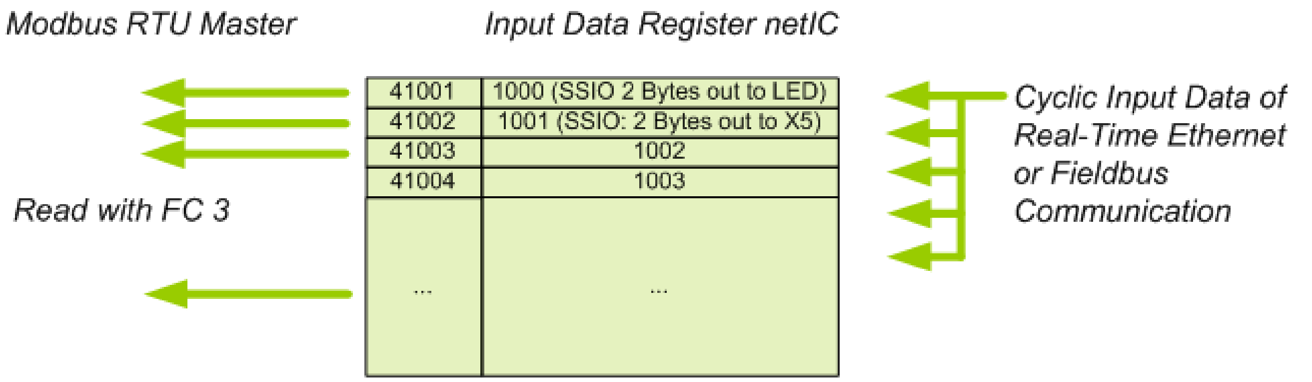


Figure 8: Register Area Input Data – Cyclic Data

The following figure shows: The Modbus RTU Master can write data starting with address 42003 using function code 16 (or 6). These data are sent to the connected Real-Time Ethernet respectively Fieldbus.

Note: The SSIO data are located by default on the first two registers of the output data area. Changing the default settings for the SSIO offset can be done to make it possible for the Modbus RTU Master to use also register 42001 respectively 42002 to send data to the connected Real-Time Ethernet respectively Fieldbus.

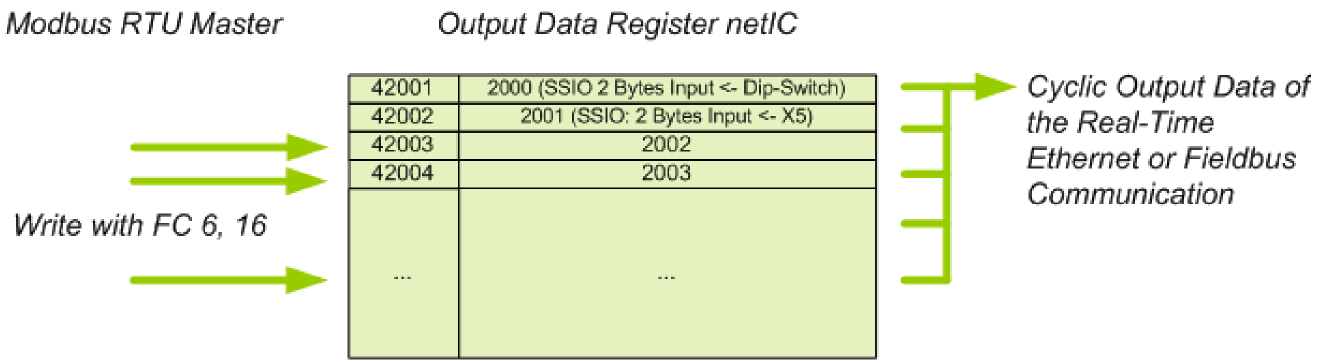


Figure 9: Register Area Output Data – Cyclic Data

12.3.2 Data Mapping Open Modbus/TCP

The following figure shows: The Modbus RTU Master can read data starting with address 41001 using function code 3. These data were written from the Open Modbus/TCP Client into the netIC.

Note: The SSIO data are located by default on the first two registers of the input data area.

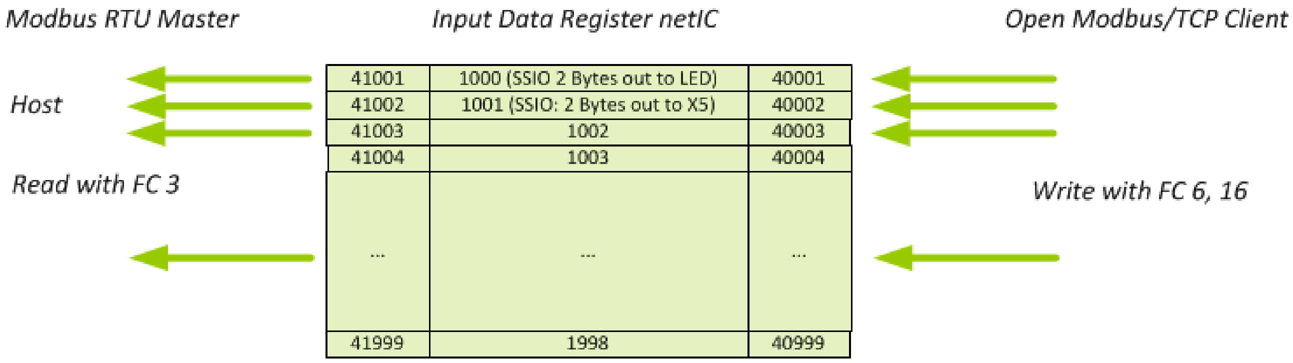


Figure 10: Register Area Input Data – Open Modbus/TCP

The following figure shows: The Modbus RTU Master can write data starting with address 42003 using function code 16 (or 6). These data can be read from the connected Open Modbus/TCP client.

Note: The SSIO data are located by default on the first two registers of the output data area. Changing the default settings for the SSIO offset can be done to make it possible for the Modbus RTU Master to use also register 42001 respectively. 42002 to send data to the connected Real-Time Ethernet respectively Fieldbus.

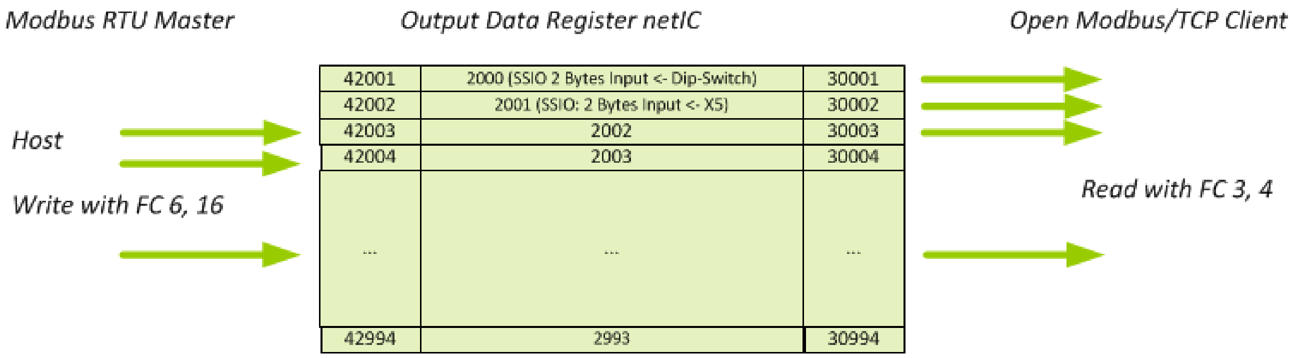


Figure 11: Register Area Output Data – Open Modbus/TCP

12.4 Acyclic Services

Many Real-Time Ethernet Systems have acyclic read and write services. In that case the interface is mailboxes instead of a data image. These Mailboxes are located also in the Register Image and are large enough to handle a whole Ethernet Frame.

The picture below illustrates the location of the used data input and output areas and registers in this context:

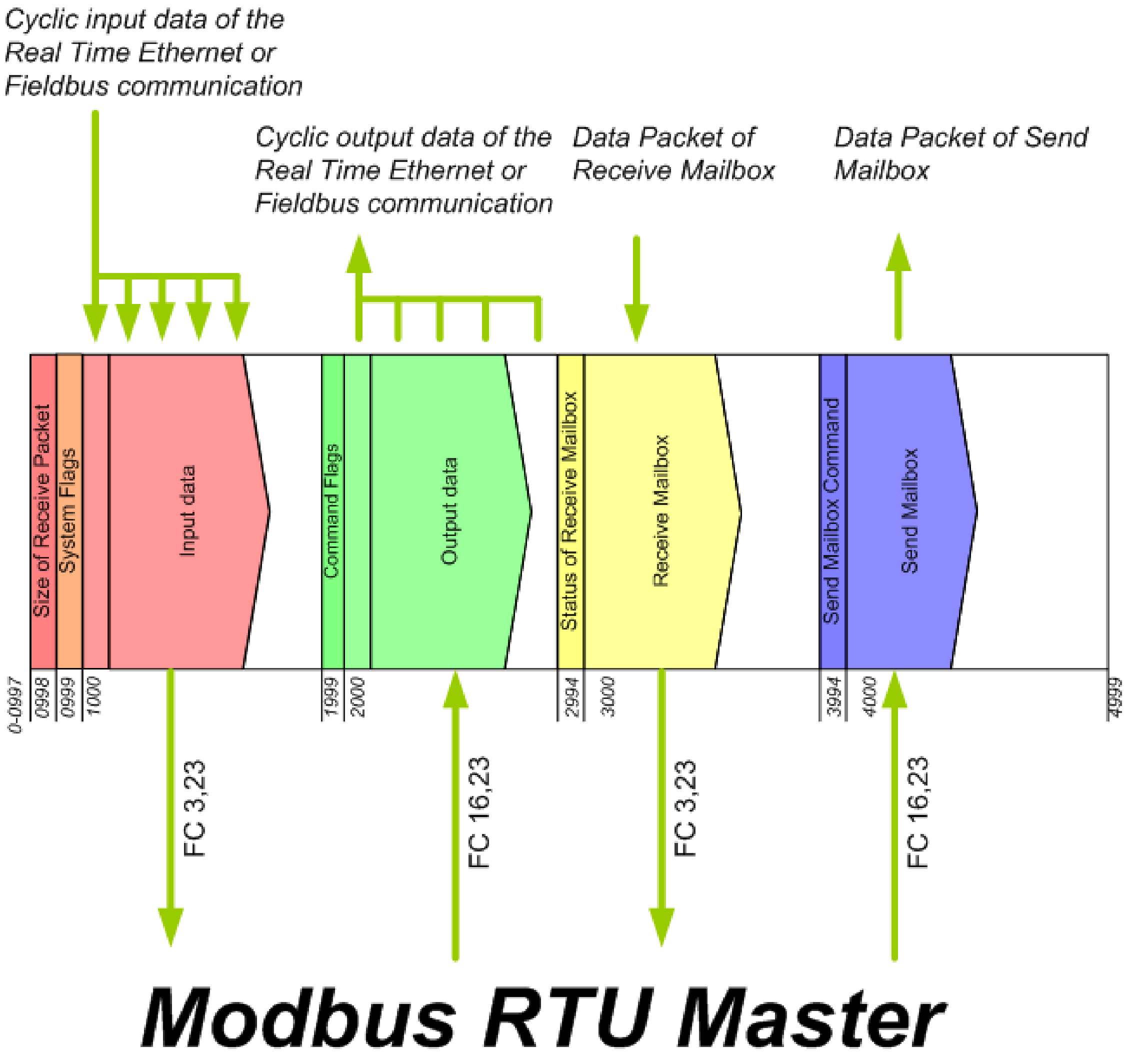


Figure 12: Location of Data Input and Output Areas and used Registers

The Modbus RTU Master can also handle these services over the serial Host Interface by reading the status information and writing the appropriate commands

To do so

- use function codes 3 or 23 for reading
- use function codes 16 or 23.for writing the commands into the mailbox

Note: However, be aware that doing so causes some amount of programming on the side of the Modbus RTU Master.

Note: Also take care of section *General Aspects* from page 188 which for example includes the description of the reduced packet header.

12.4.1 Sending Packets

With the following procedure, the Modbus RTU Master can cause the netIC to send (response and request) packets to its communication partner via Fieldbus or Real-Time Ethernet, where the netX acts as Modbus RTU Slave:

1. At first, check the flag TX_MBX_FULL within the System Flags (access Register 999, Bit 5 of the netIC via Modbus). As long as this flag has the value 1 the mailbox is occupied and it is not allowed to send a packet.
2. As soon as TX_MBX_FULL has the value 0, the packet head and data can be written in the defined Registers. If the packet is longer this can be done in a few steps as long as the *Send Packet Command* is not written. In this case we recommend writing the packet header in one step separately at the end of the transfer process.
3. By writing the register *Send Packet Command* (access Modbus-Register 3994 of the netIC) the packet will be transferred in the send mailbox and activated. The flag TX_MBX_FULL is set.
4. If the Network Protocol has taken over the packet the TX_MBX_FULL flag will be cleared.
5. If a request packet has been sent, a confirmation packet will always received. It is the duty of the application to read out the confirmation packet out of the mailbox and to evaluate it subsequently.
6. The typical case of use, however, is sending a response packet after reception of an indication packet. Evaluate the indication packet and adapt the contents of the response packet according to the results of the evaluation.

12.4.2 Receiving Packets

A received packet (either indication or confirmation) at the netIC is signaled to the Modbus RTU Master by the flag RX_MBX_FULL being set. In order to read this flag, access Register 999, Bit 4 of the netIC via Modbus

To read out the packet if the flag RX_MBX_FULL is set, the following procedure is appropriate:

- Read out the size in bytes of the received packet at the register *Received Packet Size*. Access Modbus Register 2998 to do so.

  This can alternatively be done with the cyclic read of the Input Data Image if it includes also the two registers directly before the Input Data Image. Therefore, there is a second register which also contains the size of the received Packet (Modbus Register 998).
- To identify the received packet and also to check if the confirmation of a send packets delivers an error the whole header has also to be read out.
- Read out the register *Received Packet Command* (Modbus-Register 2994) clears the RX_MBX_FULL flag. It is not necessary that this register is the last reading register of a Read Command. The RX_MBX_FULL flag will always cleared after a finishing reading data.

12.4.3 Application: Common Servicing of cyclic Input and Output Data and acyclic Input Data

The following subsection describes a special concept allowing you to process cyclic and acyclic data efficiently within one single common program loop.

The following processes are performed in parallel and synchronously within this loop.

- Cyclic reading of input data
- Cyclic writing of output data
- Check for possibly pending acyclic input data
- Reading these acyclic input data from the mailbox if present.

For more information concerning the Modbus Registers of the netIC, also see section *System Information Block* on page 74.

➢ In order to do so, proceed as follows:

1. Call the MODBUS function code FC 23 **<u>within a loop</u>** in order to perform servicing of the cyclic data. This MODBUS function code synchronously allows reading (of up to 119 16-bit-wide MODBUS registers) and writing (of up to 119 MODBUS registers) within separate areas of memory in parallel. It has the following parameters:

MODBUS FC 23

Variable	Description	Value (Example)
Device Address	Modbus device address	
Function Code	„Read/write multiple registers“	23
Data Address Read	Offset, from which reading is started	998
Data Count Read	Number of registers to be read	102
Data Address Write	Offset, from which writing is started	1999
Data Count Write	Number of registers to be written	101
Data	Data to be written follow here.	…

Table 59: MODBUS function code 23 for servicing cyclic data

The example values represent the case of reading of 102 registers beginning with register 998 and writing of 101 registers beginning with register 1999 in parallel.

If you want to read or write a number of registers exceeding the upper limit of the allowed number of registers, you cannot accomplish this with a single call of FC 23. Thus you have to divide the read or write access. To do so, we recommend to always read the highest addresses at first and the lower addresses always afterwards. As the last area of memory read the area beginning with address 998 to get the most current state of the system flags. Proceeding in reverse order as described above would cause the system flags to be possibly outdated.

➩ You have now written the data for the cyclic output area and read the cyclic input area. Additionally, the following is available:

- The system flags (i.e. the contents of register 999) are available for evaluation.
- If a new packet is available (Flag RX_MBX_FULL is set), the size of this received packet is already known (Contents of register 998).

2. Send Register Application Packet

Note: This step only needs to be performed once for initialization. This also applies for the next two steps. Set a flag within the programming loop after step 4, that indicates successful initialization and perform steps 2,3 and 4 only in case this flag has not been set in order to perform them only once during the first execution of the loop. This is necessary as step 2 requires a result of step 1 (i.e. availability of TX_MBX_FULL (Register 999, Bit 5) and therefore cannot be executed outside of the loop.

Before reception of indications is possible, first a Register Application Packet must have been sent. This functionality is available in all Hilscher protocol stacks.

In order to check whether sending of packets is currently possible, evaluate the current value of system flag TX_MBX_FULL (Register 999, Bit 5). This needs to be 0 if sending of packets is currently available.

If this is the case, the packet containing a reduced rcX header (Addresses 3994 up to 3999) can be sent using the MODBUS function code for writing (FC 16).

MODBUS FC 16

Variable	Description	Value (Example)
Device Address	Modbus Device Address	The device address with which the NIC50.RE has been configured via either the netX Configuration Tool or MODBUS/RTU.
Function Code	„Write multiple registers“	16
Data Address Write	Offset of first register to be written	3994
Data Count Write	Number of registers to be written	6
Data	Data to be written follow here	Data of Register Application Packet

Table 60: MODBUS function code 16 for writing the Register Application Packet

The registers 3994 (Send Packet Command), 3996 (Send Packet Error Code), 3998 (Send Packet Size) and 3999 (Send Packet Identifier) must have been set accordingly when sending the Register Application Packet.

3. Now evaluate the flag RX_MBX_FULL within the system flags (Bit 4 of register 999). If the flag has been set to TRUE, a data packet has arrived at the input mailbox and needs to be fetched from there (see *Table 56: System Flags* on page 87 of this document).

Note: This step needs to be performed only once for initialization. Take care of the detailed explanations at step 2!

4. In order to read out the acyclic data from the input mailbox call MODBUS function code FC 3, if the condition RX_MBX_FULL=TRUE is met. The parameters of FC 3 are these:

MODBUS FC 3

Variable	Description	Value (Example)
Device Address	Modbus Device Address	The device address with which the NIC50.RE has been configured via either the netX Configuration Tool or MODBUS/RTU.
Function Code	„Read/write multiple registers“	3
Data Address Read	Offset of first register to be read	2994
Data Count Read	Number of registers to be read	118

Table 61: Modbus function code 3 for reading out acyclic input data

The values of the example relate to reading of 118 registers beginning with register 2994.

Again, if you want to read a number of registers exceeding the upper limit of the allowed number of registers, you have to divide the read access into partial read accesses. In order to do so, again we recommend always to read the highest addresses at first and the lower addresses later on. As the last area of memory read the area beginning with address 2994 (Received Packet Command). The address 2994 should always be read within the last read access as earlier reading of address 2994 would cause premature deactivation of the protection of the input mailbox against overwriting from outside.

Note: This step needs to be performed only once for initialization. Take care of the detailed explanations at step 2!

5. Send Read Response Packet

   When an incoming Read Indication Packet is received, a Read Response Packet needs to be send as response. This can be accomplished as follows:

   In order to check whether sending of packets is currently possible, evaluate the current value of system flag TX_MBX_FULL (Register 999, Bit 5). This needs to be 0 if sending of packets is currently available.

   If this is the case, the packet can be sent using the MODBUS function code for writing (FC 16). Table 62 shows the necessary reduced packet header for this purpose:

MODBUS FC 16

Variable	Description	Value (Example)
Device Address	Modbus Device Address	The device address with which the NIC50.RE has been configured via either the netX Configuration Tool or MODBUS/RTU
Function Code	„Write multiple registers“	16
Data Address Write	Offset of first register to be written	3994
Data Count Write	Number of registers to be written	12+n
Data	Data to be written follow here	…

Table 62: Modbus function code 16 for writing the Read Response Packet

The registers 3994 (*Send Packet Command*), 3996 (*Send Packet Error Code*), 3998 (Send Packet Size) and 3999 (*Send Packet Identifier)* and the contents of the Read Response Packet must have been set accordingly when sending the Read Response Packet.

12.4.4 Example: Reception and Acknowledgement of an arriving PROFINET IO Read Request

In order to clarify the concept explained in the preceding subsection we discuss the example of an acyclic PROFINET IO Read Request arriving during running cyclic data communication here. We start with the following situation:

- A NIC50-RE with loaded PROFINET IO Device Firmware V3 (NIC50-RE/PNS) works as IO Device (Slave) on the PROFINET side and also as slave on the MODBUS/RTU side.
- A PROFINET IO Controller connected to this NIC50-RE/PNS sends a PROFINET IO read request to the NIC50-RE/PNS. This causes a PROFINET IO Device Read Indication to occur at the NIC50-RE/PNS.

Proceed as follows:

1. Read and write the cyclic data **within a loop** using MODBUS FC 23 as described in *Table 59* within the preceding section. (if your MODBUS interface does not support FC23, alternatively FC3 plus FC16 is also possible).
2. Send Register Application-Packet

   Before receiving of any indications is possible with the PROFINET IO-Device protocol stack, first a Register Application Packet needs to be send in order to register the application at the stack. This functionality is provided by the PROFINET IO-Device protocol stack.

   In order to check whether sending of packets is currently possible, evaluate the current value of system flag TX_MBX_FULL (Register 999, Bit 5). This needs to be 0 if sending of packets is currently available.

   If this is the case, the packet can be sent using the MODBUS function code for writing (FC 16). Here the reduced packet header:

   MODBUS FC 16

Variable	Description	Value (Example)
Function Code	„Write multiple registers“	16
Data Address Write	Offset of first register to be written	3994
Data Count Write	Number of registers to be written	12

Table 63: MODBUS function code 16 for writing the Register Application Packet

The registers 3994 (*Send Packet Command*), 3996 (*Send Packet Error Code*), 3998 (*Send Packet Size*) and 3999 (*Send Packet Identifier*) and all registers containing the data of the Register Application Packet must have been set accordingly when sending the Register Application Packet, see the subsequent table.

Register #	Description	Type	Value (Example)
3994	Send Packet Command	Unsigned integer (32 bit)	0x2F10
3995			0x0000
3996	Send Packet Error Code	Unsigned integer (32 bit)	0
3997			0
3998	Send Packet Size	Unsigned integer (16 bit)	0x0028
3999	Send Packet Identifier	Unsigned integer (16 bit)	Packet ID (any value)

Table 64: Register Application-Packet

3. Evaluate the system flag RX_MBX_FULL (this flag is Bit 4 of Register 999). Only in case this flag has been set to TRUE, an acyclic input data packet needs to be fetched from the input mailbox. Concerning the RX_MBX_FULL flag also see *Table 56: System Flags* on page 87 of this document.
4. If the confirmation packet of the Register Application request has been received, RX_MBX_FULL must be set to 1. In this case read the confirmation packet using the MODBUS function code 23 or 3 as described in the fourth step of the preceding section. You can identify this packet having the command code 0x00002F11 in register pair 2994/2995. Prior to reception of such a packet, no PROFINET IO Read Indication packets can be received.
5. Now, check for PROFINET IO Read Indications just in the same manner using the RX_MBX_FULL flag again. If the flag is set to 1, then read the PROFINET IO Read Indication packet using the MODBUS function code 23 or 3 as described in the fourth step of the preceding section. The data received beginning with register 2994 should comply with the subsequent table:

Register #	Description	Type	Value (Example)
2994 - 2995	Receive Packet Command	Unsigned integer (32 bit)	0x00001F36
2996 - 2997	Receive Packet Error Code	Unsigned integer (32 bit)	x
2998	Receive Packet Size	Unsigned integer (16 bit)	32
2999	Receive Packet Identifier	Unsigned integer (16 bit)	Packet ID
3000 - 3001	Record handle	Unsigned integer (32 bit)	
3002 - 3003	Device handle	Unsigned integer (32 bit)	
3004 - 3005	Sequence number	Unsigned integer (32 bit)	
3006 - 3007	API to be read	Unsigned integer (32 bit)	
3008 - 3009	Slot to be read	Unsigned integer (32 bit)	

Continued on next page

3010 - 3011	Subslot to be read	Unsigned integer (32 bit)	
3012 - 3013	Index to be read	Unsigned integer (32 bit)	
3014 - 3015	Read record data length	Unsigned integer (32 bit)	

Table 65: Register Set containing Data from Register Application Packet

6. Send Read Response Packet

   Sending the read response packet as necessary response to the read indication packet is accomplished as follows:

   In order to check whether the Modbus RTU Master may currently write a packet into the Send Mailbox, evaluate the current value of system flag TX_MBX_FULL (Register 999, Bit 5). This needs to be 0 if sending of packets is currently available.

   If this is the case, the packet can be sent using the MODBUS function code for writing (FC 16). Here the reduced packet header:

MODBUS FC 16

Register #	Description	Type
Function Code	„Write multiple registers“	16
Data Address Write	Offset of first register to be written	3994
Data Count Write	Number of registers to be written	52+n
Data	Data to be written follow here	Response data

Table 66: Modbus function code 16 for writing the Read Response Packet

The registers 3994 (Send Packet Command), 3996 (Send Packet Error Code), 3998 (Send Packet Size) and 3999 (Send Packet Identifier) and all registers of the Read Response Packet must have been set accordingly when sending the Read Response Packet. See subsequent table:

Register #	Description	Type	Value (Example)
3994	Send Packet Command	Unsigned integer (32 bit)	0x1F37
3996	Send Packet Error Code	Unsigned integer (32 bit)	x
3998	Send Packet Size	Unsigned integer (16 bit)	40+ n (n 0 Data length in Byte)
3999	Send Packet Identifier	Unsigned integer (16 bit)	Packet ID
4000	Record handle	Unsigned integer (32 bit)	Take over from received Read Indication-Packet
4002	Device handle	Unsigned integer (32 bit)	Take over from received Read Indication-Packet
4004	Sequence number	Unsigned integer (32 bit)	Take over from received Read Indication-Packet
4006	API to be read	Unsigned integer (32 bit)	Take over from received Read Indication-Packet
4008	Slot number to be read	Unsigned integer (32 bit)	Take over from received Read Indication-Packet
4010	Subslot number to be read	Unsigned integer (32 bit)	Take over from received Read Indication-Packet
4012	Index to be read	Unsigned integer (32 bit)	Take over from received Read Indication-Packet
4014	Read record data length	Unsigned integer (32 bit)	Take over from received Read Indication-Packet
4016	PROFINET error codes	Unsigned integer (32 bit)	On error: Suitable error code, otherwise 0
4018	Additional value 1	Unsigned integer (16 bit)	0
4020	Additional value 2	Unsigned integer (16 bit)	0
4022-4533	Data area	Field consisting of 1024 unsigned integer 8-bit values	Data to be transmitted

Table 67: Modbus function code 16 for writing the Read Response Packet

Additional value 1 and 2 are only relevant for working with PROFINET IO Profiles and can always be set to 0 if not working with PROFINET IO Profiles.

For more information refer to the PROFINET IO RT IRT Device Protocol API Manual, Revision 7. Concerning PROFINET IO error codes see chapter 11 there.

12.4.5 Packet Filter Function for PROFINET IO Device

The netIC mailboxes are the interface between host and netIC for acyclic communication and serve to transmit packets. The application program of the host system has to receive indication packets from the netIC, process them, and send the response packet to the netIC.

To reduce the quantity of packets to be processed, the host application program can activate a packet filter function in the netIC firmware. The netIC firmware filters out and processes certain PROFINET packets, i.e. answers requests received from PROFINET. Only the PROFINET IO Device firmware supports the packet filter function.

Table 68 lists all PROFINET packets to be processed by the netIC firmware while the packet filter function is activated.

PROFINET Packets	Command
PNS_IF_PARAM_END_IND	0x00001F0E
PNS_IF_AR_CHECK_IND	0x00001F14
PNS_IF_CHECK_IND	0x00001F16
PNS_IF_RESET_FACTORY_SETTINGS_IND	0x00001F18
PNS_IF_SAVE_STATION_NAME_IND	0x00001F1A
PNS_IF_START_LED_BLINKING_IND	0x00001F1E
PNS_IF_STOP_LED_BLINKING_IND	0x00001F20
PNS_IF_AR_INDATA_IND	0x00001F28
PNS_IF_AR_ABORT_IND	0x00001F2A
PNS_IF_APDU_STATUS_IND	0x00001F2E
PNS_IF_ALARM_IND	0x00001F30
PNS_IF_LINK_STATE_CHANGE_IND	0x00001F70
PNS_IF_SAVE_IP_ADDR_IND	0x00001FB8
PNS_IF_CONNECT_REQ_DONE_IND	0x00001FD4
PNS_IF_RELEASE_RECV_IND	0x00001FD6
PNS_IF_USER_ERROR_IND	0x00001FDC
PNS_IF_STORE_REMANENT_DATA_IND	0x00001FEA
PNS_IF_PARAMET_SPEEDUP_SUPPORTED_IND	0x00001FF8
PNS_IF_EVENT_IND	0x00001FFE

Table 68: PROFINET Packets supported by the Packet Filter Function

Prerequisite for using the Packet Filter Function

- netIC PROFINET IO Device firmware version: V2.0 or higher.
- The host application program has to activate the function as described below.

Activating the Packet Filter Function

To activate the packet filter function, the host application program sets bits 14 and 15, i.e. writes value 0xC000 in the command flags (register 1999).

The netIC firmware processes the PROFINET packets listed in Table 68 while the packet filter function is activated. If it is deactivated, the netIC firmware will forward all PROFINET packets via mailbox to the host application program for processing.

Deactivating the Packet Filter Function

To deactivate the packet filter function, the host application program has to write value 0xC002 to the command flags (register 1999).

Activating/deactivating the Packet Filter Function for individual Packets

The packet filter function also allows activating (netIC firmware processes the PROFINET packet) or deactivating (host application program processes the PROFINET packet) the filter function for individual PROFINET packets.

The packet filter function has to be activated first. For this purpose the host application program will set bits 14 and 15 i.e. writes value 0xC000 in the command flags (register 1999) and activate the packet filter function. The netIC firmware will then process the PROFINET packets listed in Table 68.

The host application program can now configure the packet filter function. Therefore the host application program can send filter configuration packets listet in Table 69 to the netIC to delete individual PROFINET packets from or add them again to the list. Alternatively, PROFINET packets can be added after clearing the list.

Packets to configure the Packet Filter Function	Command
The netIC firmware processes the PROFINET packet MEMORY_MAP_CMD_ADD_TO_FILTER_REQ	0x00004E10
The host application program processes the PROFINET packet MEMORY_MAP_CMD_DEL_FROM_FILTER_REQ	0x00004E12
The host application program processes all PROFINET packets MEMORY_MAP_CMD_CLEAR_FILTER_CNF	0x00004E14

Table 69: Packets for changing the Packet Filter Function

Programming

The following page shows the packet structure reference for programming.

Packet Structure Reference

```
#define MAX_PF_FILTERED_IND 256

/** Packet Commands ***************************************************************/
#define MEMORY_MAP_CMD_ADD_TO_FILTER_REQ (0x00004E10)
#define MEMORY_MAP_CMD_ADD_TO_FILTER_CNF (0x00004E11)

#define MEMORY_MAP_CMD_DEL_FROM_FILTER_REQ (0x00004E12)
#define MEMORY_MAP_CMD_DEL_FROM_FILTER_CNF (0x00004E13)

#define MEMORY_MAP_CMD_CLEAR_FILTER_REQ (0x00004E14)
#define MEMORY_MAP_CMD_CLEAR_FILTER_CNF (0x00004E15)

/****************************************************************
 * Add command into the Filter List
 ****************************************************************/
typedef struct MEMORY_MAP_ADD_TO_PACKET_FILTER_DATA_Ttag
{
  TLR_UINT32 aulCmdList[MAX_PF_FILTERED_IND];

}MEMORY_MAP_ADD_TO_PACKET_FILTER_DATA_T;

typedef struct MEMORY_MAP_ADD_TO_PACKET_FILTER_PCK_Ttag
{
  TLR_PACKET_HEADER_T tHead;
  MEMORY_MAP_ADD_TO_PACKET_FILTER_DATA_T tData;

}MEMORY_MAP_ADD_TO_PACKET_FILTER_PCK_T;

/****************************************************************
 * Delete commands from the Filter List
 ****************************************************************/
typedef struct MEMORY_MAP_DEL_FROM_PACKET_FILTER_DATA_Ttag
{
  TLR_UINT32 aulCmdList[MAX_PF_FILTERED_IND];

}MEMORY_MAP_DEL_FROM_PACKET_FILTER_DATA_T;

typedef struct MEMORY_MAP_DEL_FROM_PACKET_FILTER_PCK_Ttag
{
  TLR_PACKET_HEADER_T tHead;
  MEMORY_MAP_DEL_FROM_PACKET_FILTER_DATA_T tData;

}MEMORY_MAP_DEL_FROM_PACKET_FILTER_PCK_T;

/****************************************************************
 * Clear the Filter List
 ****************************************************************/
typedef struct MEMORY_MAP_CLEAR_PACKET_FILTER_PCK_Ttag
{
  TLR_PACKET_HEADER_T tHead;

}MEMORY_MAP_CLEAR_PACKET_FILTER_PCK_T;


typedef TLR_PACKET_HEADER_T MEMORY_MAP_PACKET_FILTER_CNF_T;
```


12.5 Commonly-used Register Area

Beginning with firmware version 1.5.x.x, there is an additional area of registers (#6000 … #7998) which are accessible by the host and also by the WebServer integrated in the netIC's firmware.

Both read and write access is possible.

You can use this area in order to store own data within the virtual DPM and to read those back.

Also this area can be used for storage of data of the WebServer.

Data access is synchronized by register #7999 in conjunction with the system flag `SX_WRITE_IND`.

12.6 Watchdog Function

The netIC firmware provides an own watchdog functionality. However, this function is only available if the netIC is configured as Modbus RTU Slave at the serial host interface.

- In order to activate the watchdog functionality the flag ‚WDG_ON' must be set within the Command Register (Register 1999).
- In order to deactivate the watchdog functionality the flag ‚WDG_OFF' must be set there

Whether the watchdog functionality is activated is indicated with the ‚WDG_ON' flag in System Register (Register 999). The watchdog functionality can be activated or deactivated by the host at any time during runtime by setting or clearing the respective bit.

The watchdog time may be adjusted by the „netX Configuration Tool".

Principle of Function:

If the watchdog functionality has been activated by setting the corresponding bit, the Modbus RTU Master must send valid Modbus requests within the configured watchdog time. As soon as the netIC has received a valid request, it will trigger the watchdog again.

If the netIC is not addressed in the defined watchdog time, the fieldbus interface and the synchronous serial interface will automatically caused to be brought into a secure state. In this context secured state means cleared outputs for the synchronous serial interface.

In general, triggering the watchdog function at the Field bus will also cause clearing of input and output data. It can also cause reactions which are defined as secure state for the specific Field bus in question. For instance, diagnostic informations can be sent to assigned master. Details concerning this topic are discussed in the manuals describing the according protocols.

If the communication channels are in secure state by triggering of the watchdog, Modbus RTU communication remains in operation.

Leaving the watchdog state is only possible by a reset. This may happen either by a hardware reset, or a software reset in which the master sets the corresponding reset flag in command register (Register Address 1999).

Watchdog times for the Field bus and the synchronous serial interface are separately configurable within the ‚netX Configuration Tool'. Therefore, the response time of the host application must be less than the lower of the watchdog time of both interface.

12.7 Structure of the Firmware

The netX processor integrated within the netIC Communication IC runs the multitasking Real-Time kernel rcX as operating system.

- The entire software is structured in different tasks running under rcX (see Figure 13 below).
- All tasks are connected with each other by the virtual dual-port-memory which can be seen as the central component for intercommunication and data exchange between the tasks (for instance, see 'Data Image' area of illustration below and the next section of this document).
- On the communication side these are the
- The Protocol Stack (Real-Time-Ethernet or Fieldbus),
- the Modbus RTU-Task for the host communication
- and the task managing the synchronous serial input/output interface.

On the other side the gateway task transfers all received and transmitted data cyclically between the different data areas.

The diagnostic task performs the access at the virtual Dual-Port-Memory. Also the download of a new firmware or configuration files to the netIC Communication IC can be managed by the *netX Configuration Tool*.

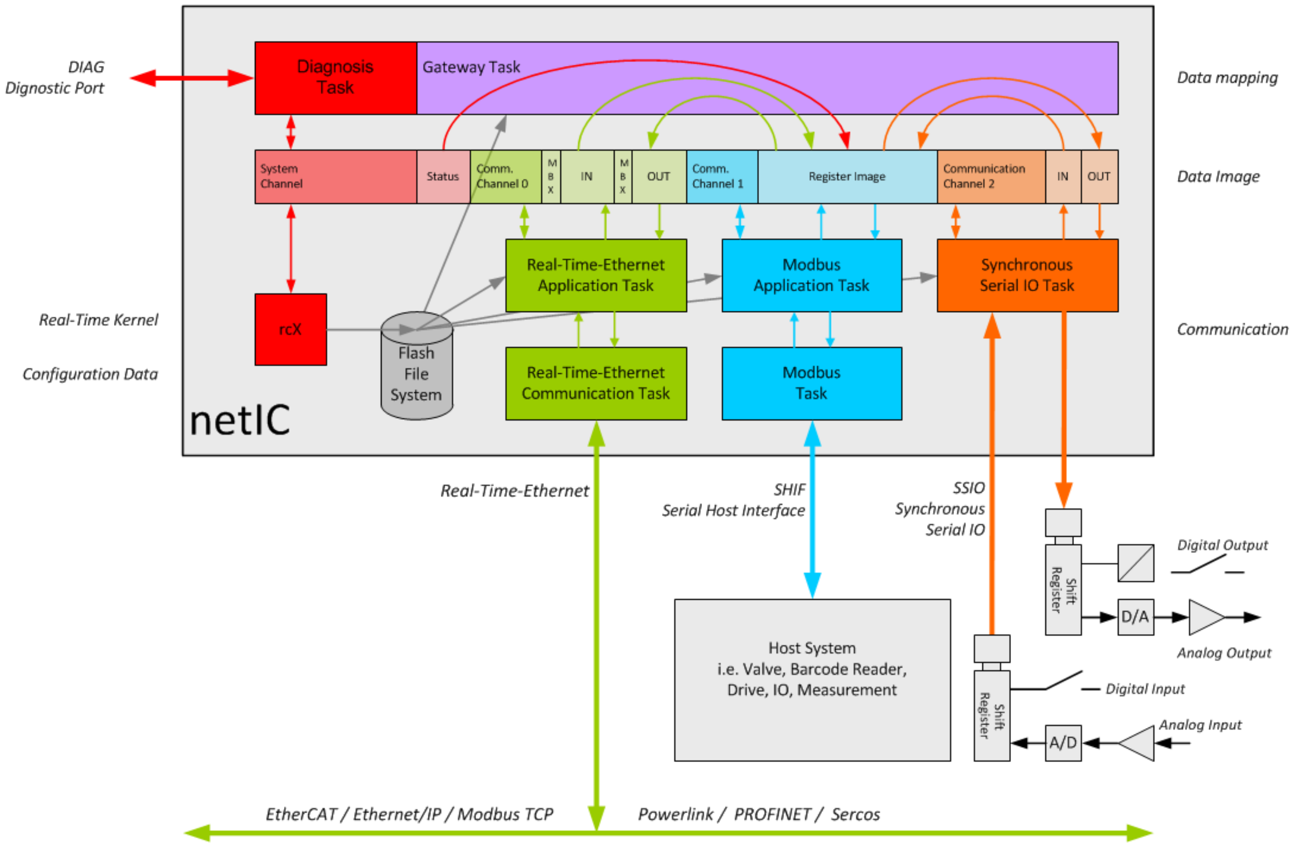


Figure 13: Structure of the Firmware of the Realtime Ethernet DIL-32 Communication IC NIC 50-RE

13 Design-In - Integrating netIC into the Host System

This chapter describes how to integrate the netIC Communication ICs into a host system. The design-in process can be divided into two steps, one concerning the signals and interfaces which are identical at all netIC Communication ICs and one concerning those signals and interfaces depending on the chosen netIC hardware and varying from module to module.

Reflecting this, the documentation of the design-in process is also divided into the two basic sections

- General Information about netIC
- Module-specific Information on the netIC.

13.1 General Information about netIC

13.1.1 General Design Rule

Do not place any electronic components in your design below the DIL32 socket! This may cause problems, especially if the NIC52-RE is used.

13.1.2 Block Diagram and Pin Assignment

netIC General Block Diagram

The following block diagram illustrates both how to apply the netIC and how it is structured internally:

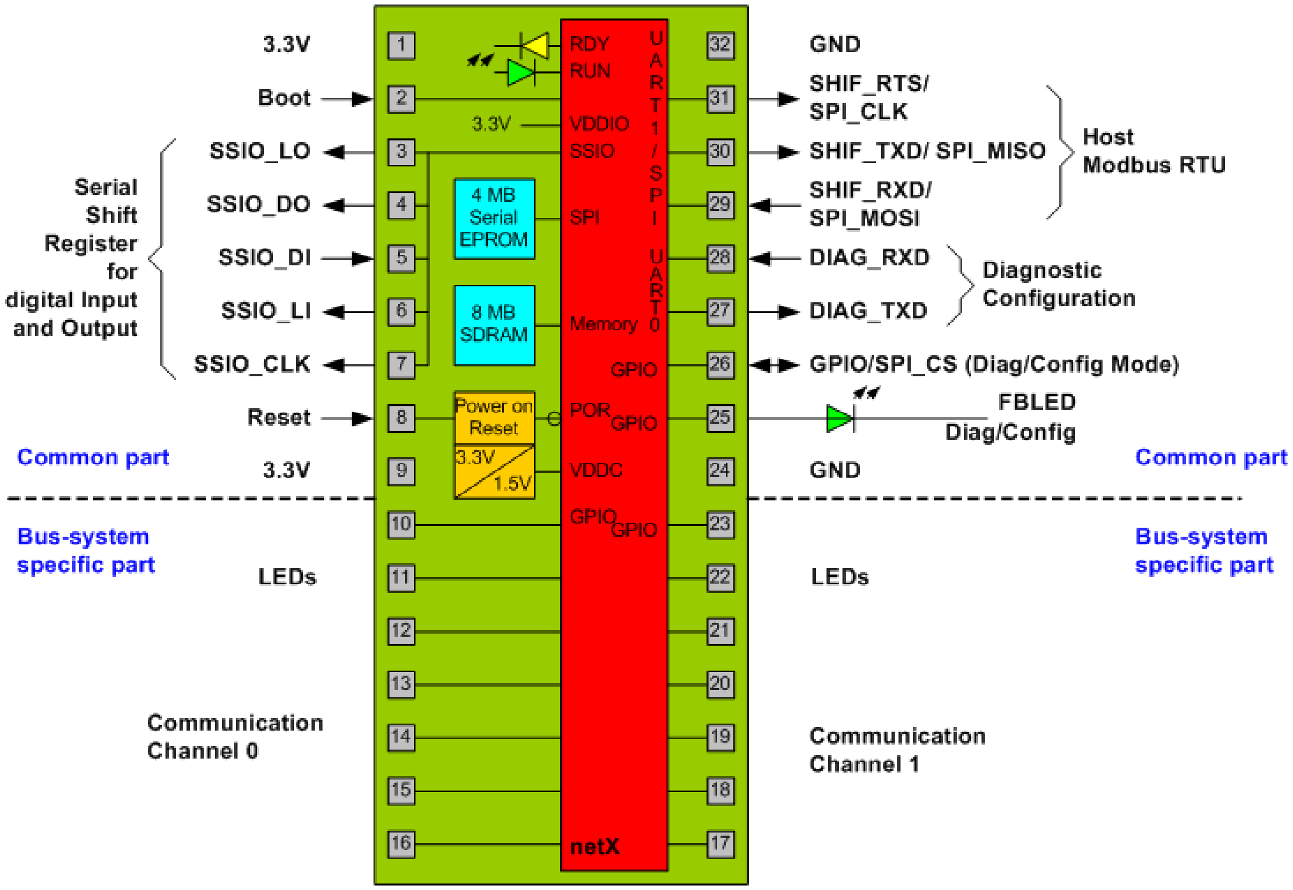


Figure 14: netIC General Block Diagram - External Connections and internal Structure

Pins 1 to 9 and 24 to 32 are generally used in the same way at all netIC Communication ICs while the pins 10 to 23 are used for communication system specific individual signals.

This is done as follows:

- At pins 13 to 20 signals of one (Fieldbus) or two (Real-time Ethernet) communication channels are located.
- At pins 10 to 12 and pins 21 to 23 always LED signals are located.

Pin Assignment netIC – Common Pins at all Types

The schematic illustration shows the common pin assignment of the netIC Communication ICs.

- The pins marked in blue color are the standard pins which do not depend from the Real-Time Ethernet or Fieldbus system.
- The red pins are used for LED signals depending on the communication system.
- The white pins represent communication lines of the communication system.
- Only the blue pins are relevant within the scope of this subsection.

Figure 15: Pinning of netIC

The following table explains the assignment of pins and signals and also provides the direction and the meaning of these signals:

Pins of the left side			
Pin	**Signal**	**Direction**	**Explanation**
1	+3V3		+3.3 V Power Supply
2	BOOTn	Input	Start Boot Mode
3	SSIO_LOn	Output	Synchronous Serial Interface - Latch Output Data
4	SSIO_DO	Output	Synchronous Serial Interface - Output Data
5	SSIO_DI	Input	Synchronous Serial Interface - Input Data
6	SSIO_LIn	Output	Synchronous Serial Interface - Latch Input Data
7	SSIO_CLK	Output	Synchronous Serial Interface - Shift Clock
8	RESETn	Input	Reset netIC **(not compatible with 5V!)**
9	+3V3		+3.3 V Power Supply
Pins of the right side			
Pin	**Signal**	**Direction**	**Explanation**
24	GND		Ground
25	FBLED	Output	FBLED Configuration/Diag-LED
26	GPIO/SPI_CS	In/Output	Config & Diag Mode: General Purpose IO (Config./Diag Mode) SPI Mode: SPI Chip-Select Signal (SPI Mode)
27	DIAG_TXD	Output	Diagnostic Interface - Transmit Data
28	DIAG_RXD	Input	Diagnostic Interface - Receive Data
29	SHIF_RXD/ SPI_MOSI	Input	Config & Diag Mode: Serial Host Interface - Receive Data SPI Mode: SPI Master out Slave in
30	SHIF_TXD/ SPI_MISO	Output	Config & Diag Mode: Serial Host Interface - Transmit Data SPI Mode: SPI Master in Slave out (SPI Mode)
31	SHIF_RTS/ SPI_CLK	Output/ Input	Config & Diag Mode: Serial Host Interface - Ready To Send SPI Mode: SPI Serial Clock (SPI Mode)
32	GND		Ground

Table 70: Pinning of netIC

The signals can be grouped as follows:

- Signals 1, 9, 24 and 32 are the pins for providing the operation voltage (Ground and 3.3 V power supply)
- Signals 3 to 7 represent the Synchronous_Serial_Interface .
- Signals 2 and 8 are needed for booting and reset purposes.
- Signals 27 to 28 belong to the Diagnostic_Interface
- Signals 29 to 31 belong to the Serial_Host_Interface_(SHIF) or to the SPI interface if the netIC has been configured to SPI Mode.
- Signal 25 belongs to the FBLED.
- Signals 26 switches between SHIF Mode and SPI Mode.

13.1.3 Power Supply

The netIC Communication IC is a complete system which needs only a 3.3V power supply to operate. The small SYS-LED at the lower left corner of the module indicates the current system status of the module.

The core supply voltage, the clock and a defined reset signal at power up for the netX 10/netX 50/netX 52 is generated internally.

Power supply and ground should be connected at the shortest distance to the power and ground Plane of the host system. We suggest one ceramic capacitor with 10 µF (X5R / X7R) between the pins for decoupling the power supply.

13.1.4 Host Interface

13.1.4.1 Reset Signal

The reset signal **RESETn** can be used to reset the netIC from the host system. A reset can be activated either

- with a push button manually,
- at power up from a power supervisor chip or
- using a reset signal by the host controller.

When the reset signal has a low level at the pin for at least 10 µs, the netIC device excecutes a reset.

NOTICE

Device Destruction!

- The reset signal **RESETn** is not compatible to a voltage of 5 V. A higher voltage than 3.3 V + 5% may cause damage at the netIC device.

A push button can be connected the reset pin directly to ground without any external debounce circuit.

If the reset pin will not be used, this pin can be left open (unconnected).

13.1.4.2 Boot Signal

The boot signal **BOOTn** is used to stay in the boot mode after reset and wait with polling the diagnostic line for serial commands. This is done by short circuit connection to Ground. This can be done by an open collector, open drain digital output or by push button or equivalent. Normally this is not necessary and mainly used on programming boards like the evaluation board not to start the firmware. This means in case the firmware or the configuration file is corrupted or has an internal error which hang up the netX 50 processor these files can be deleted in boot mode. The configuration and the firmware can be downloaded over the diagnostic port if there are no failure conditions at any time.

13.1.4.3 Configuration of Host Interface (GPIO Signal)

This signal is located at pin 26. This pin has a driving capability of 6 mA.

- It is a general signal which can be used as input or output with different functions.

Behavior at netIC start-up

At netIC Start-up, the GPIO signal (*General Peripheral Input/Output Signal*) is configured as input only.

The pin is reconfigured as SPI Chip Select signal (SPI_CS, input) if the following conditions are met:

- the configuration has been loaded already
- the usage of the SPI interface has been configured
- the firmware has already configured the SPI mode

The SPI Mode is supported by firmware version 1.3.12.x and higher:

Behavior of firmware beginning from version 1.3.12.x

These firmware versions support the SPI mode and use pin 26 as SPI Chip-Select Signal SPI_CS, if the netIC is configured to SPI Mode. Pin 26 cannot be used any more for switching on and off configuration mode via push button T3.

At the NICEB Evaluation Boards with firmware version up to 1.3.11.x the GPIO/SPI_CS signal has been used for switching on and off configuration mode via push button T3. These firmware versions do not support the SPI Mode.

Note:

This means, the reset button T3 is not serviced anymore in firmware version 1.3.12.x and therefore cannot be used for switching between configuration mode and standard mode. The configuration mode is automatically activated and deactivated after 10 seconds.

Rest level consideration for firmware with support for push button T3 as configuration mode switch (prior to version 1.3.12.x)

Note:

If pin 26 (GPIO/SPI_CS) is not used, it needs to be combined with a pull-up resistor dimensioned with 4.7 kΩ in your design. It may not be left open as this might cause problems with firmware versions older than version 1.3.12.x during start-up.

13.1.4.4 Serial Host Interface (SHIF)

Pins 29-31 represent the serial host interface of the netIC Communication IC. It consists of:

Signal	Pin	Description
SHIF_TXD	30	This is the transmit data signal of the serial host interface. This Interface is freely programmable.
SHIF_RXD	29	This is the receive data signal of the serial host interface. This Interface is freely programmable.
SHIF_RTS	31	The Return To Send Signal **SHIF_RTS** can be used to control RS422- or RS485 drivers.

Table 71: Pin Assignment serial Host Interface

The serial host interface of the netIC consists of normal UART signals for transmit and receive data. Normally they are connected with a RS232 driver as the physical interface to a host or a PC.

The host interface has also the signal SHIF_RTS to control the data direction or the enable signal of a RS422 or RS485 driver.

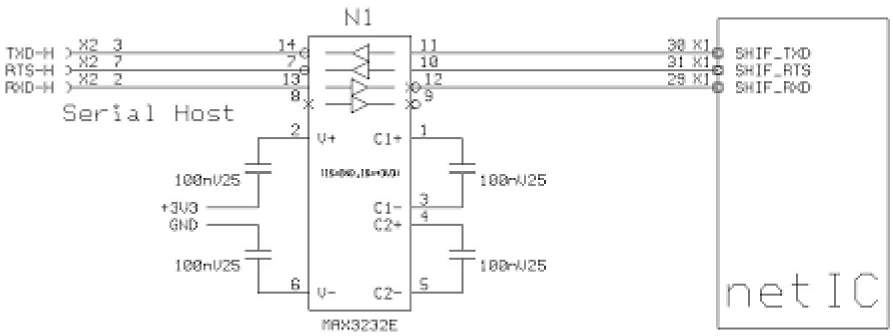


Figure 16: Proposal for the Design of the Serial Host Interface

13.1.4.5 SPI-Interface

By special configuration of the netIC (SPI Mode) also an SPI interface (Serial Peripheral Interface) can be implemented using the pins of the SHIF (29 to 31) and additionally pin 26, which is usually applied for GPIO.

In this case the following pin descriptions apply:

Signal	Pin	Description
SPI_CS	26	In SPI Mode, this pin represents the Chip-Select-Signal of the SPI interface of the NIC 50 (Logically 0 active). This line is often denominated as SS, CS or STE at SPI meaning *Slave Select*, *Chip Select* and *Slave Transmit Enable*, respectively.
SPI_MOSI	29	In SPI Mode, this pin represents the MOSI-Signal (*Master out Slave in*) of the SPI interface of the NIC 50, i.e. the input data line of the SPI interface of the netIC. This line is often denominated as SDI (*Serial Data In*).
SPI_MISO	30	In SPI Mode, this pin represents the MISO Signal (*Master in Slave out*) of the SPI interface of the NIC 50, i.e. the output data line of the SPI interface of the netIC. This line is often denominated as SDO (*Serial Data Out*).
SPI_CLK	31	In SPI Mode, this pin represents the serial clock signal of the SPI interface of the NIC 50. This line is often denominated as SCK (*Serial Clock*).

Table 72: Pin Assignment SPI Interface

The following is a proposal for the Design of an SPI (Serial Peripheral Interface) using the pins of the netIC's serial host interface.

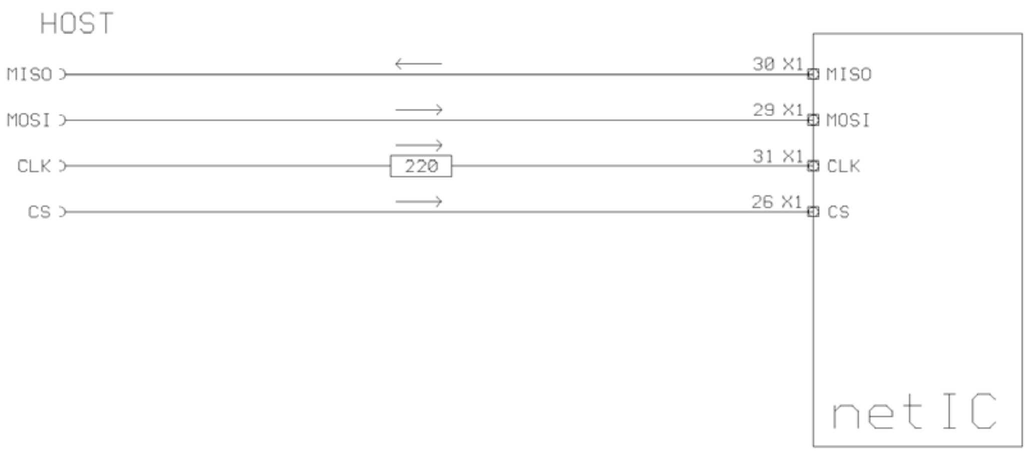


Figure 17: Proposal for the Design of an SPI Interface for the netIC

NOTICE

Device Destruction!

- The 220 Ω resistor in the CLK line of the SPI interface is required for protection against short circuit. Therefore never omit this resistor! This is due to the fact that the default setting at delivery is: RTS is driven.

For more information about the SPI interface itself and how to use it on the netIC, see chapter 16"*Serial Peripheral Interface (SPI) for netIC*" on page 167.

Signal level of SPM/SPI interface

At the NIC52-RE, the signal levels at the pins of the SPM/SPI interface to serial dual-port memory (i.e. pins 29, 30 und 31) can amount up to 1.3 V at rest (for instance at start-up). The exact value depends from the design of the external circuitry.

Note:
Design your host system in such a way, that voltages up to 1.3 V at pins 29, 30 and 31 are logically interpreted as LOW. This must be assured by the design of your host system in order to avoid malfunction.

The cause for this behavior lies in the difference of the internal design of netX 50 and netX 52 (netX 50 has internal pull-down resistors, netX 52 has internal pull-up resistors) and in the necessity of additional pull-down resistors (10 kΩ) in NIC52-RE due to compatibility reasons.

13.1.5 Serial Shift IO Interface

Pins 3-7 represent the serial shift IO interface which is present at all netIC Communication ICs.

The serial shift IO interface consists of

Signal	Pin	Description
SSIO_LOn	3	This signal represents the Latch Output Data, i.e. the data taken over from the shift register into the output register with the rising edge of that signal. The signal is also denominated as **LoadOut**.
SSIO_DO	4	This signal represents the Serial Output Data to be transferred into the serial shift IO interface flip-flops, The MSB is transmitted at first.
SSIO_DI	5	This signal represents the Serial Input Data to be received from the serial shift IO interface flip-flops, The MSB is transmitted at first.
SSIO_LIn	6	This signal represents the Latch Input Data of the serial shift IO interface. The signal is also denominated as **nLoadIn**. It works as follows: Signal at low level sets the flip-flops of the shift registers at the level of their parallel input data. Signal at high level saves the input data which then can be read out serially.
SSIO_CLK	7	This signal represents the clock signal for the serial shift IO interface for input and output data. Shifting or latching the data takes place with the leading edge of the SSIO_CLK signal.

Table 73: Pins Serial Shift IO Interface

Usually, these pins are applied to connect to shift registers, see below. This is easily to be accomplished and allows you to enlarge the amount of available IO signals with external low cost shift registers.

As an example, a complete schematic showing how to connect to shift registers is provided below. It is based on the 74HC165 and 74HC594 shift registers.

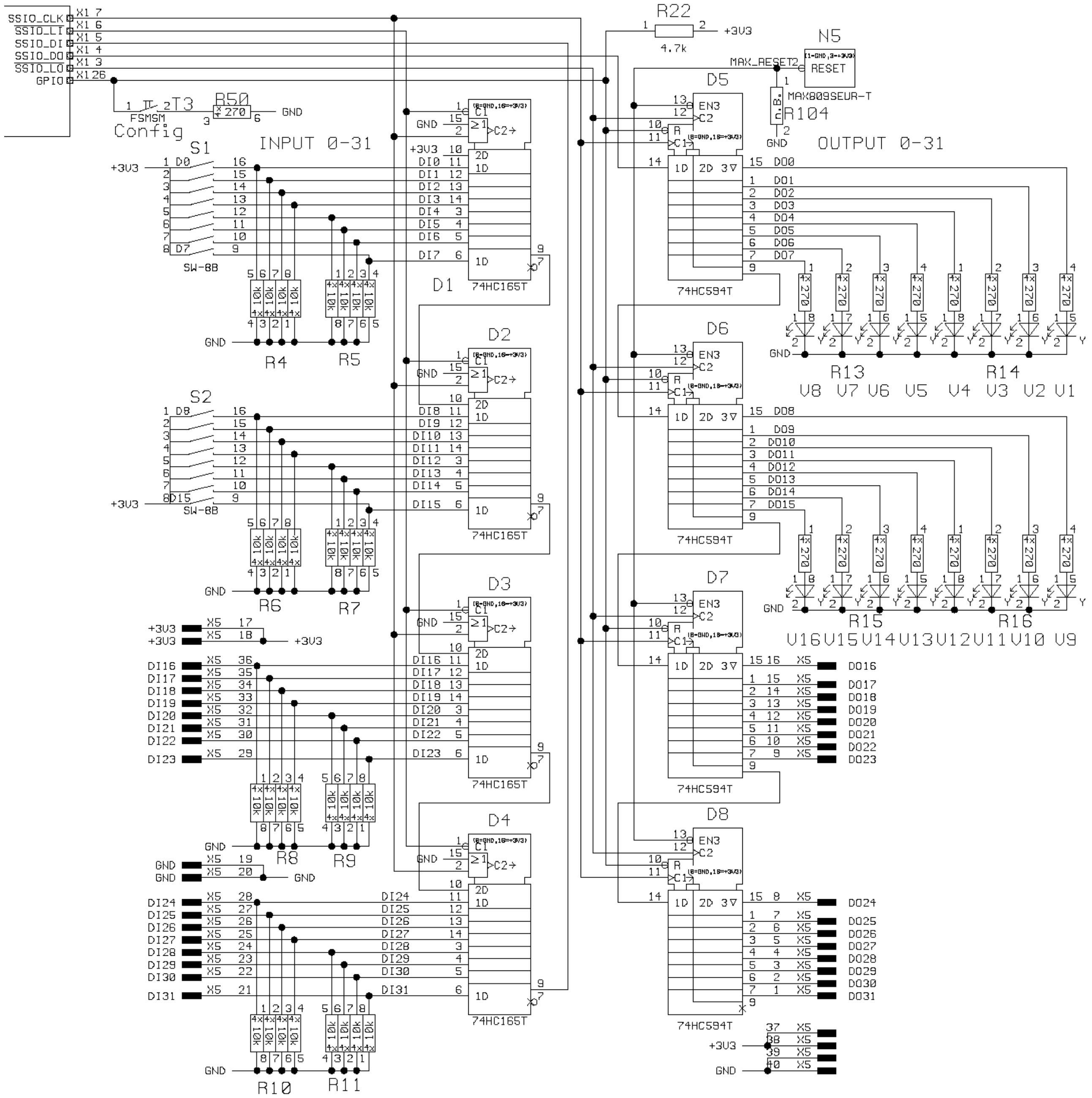


Figure 18: Proposal for the Design of the Serial Shift IO Interface

Figure 19 on page 120 and *Figure 20* on page 120 contain precise timing diagrams of the SSIO signals for input and output.

Table 74 and *Table 75* below contains minimum, typical and maximum values for the relevant SSIO timing parameters related to these diagrams for the NIC50-RE and the NIC52-RE, respectively:

Parameter	Minimum	Typical	Maximum
t_{clkh}		100 ns	
t_{clkl}		100 ns	
t_d	95 ns	100 ns	105 ns
t_1	100 ns	250 ns	
t_2	100 ns	38.55 µs	
t_3 (4 Byte)	100 ns	48.85 µs	
t_4	100 ns	6.8 µs	
t_5 (4 Byte)	100 ns	16.7 µs	
t_6	4 ms	Depends on setting of cycle time	13 ms
t_{LO}	100 ns	3.1 µs	
t_{LI}	100 ns	3.45 µs	-

Table 74: Minimum, typical and maximum Values in SSIO Interface Timing Diagram for NIC50-RE

Parameter	Minimum	Typical	Maximum
t_{clkh}	94 ns	96 ns	101 ns
t_{clkl}	99 ns	105 ns	116 ns
t_d	102 ns	104 ns	107 ns
t_1	94 ns	96 ns	101 ns
t_2	100 ns	10.83 µs	
t_3 (4 Byte)	100 ns	21.18 µs	
t_4	100 ns	5.11 µs	
t_5 (5 Byte)	100 ns	17.38 µs	
t_6	4 ms	Depends on setting of cycle time	13 ms
t_{LO}	100 ns	3.96 µs	
t_{LI}	100 ns	3.88 µs	-

Table 75: Minimum, typical and maximum Values in SSIO Interface Timing Diagram for NIC52-RE

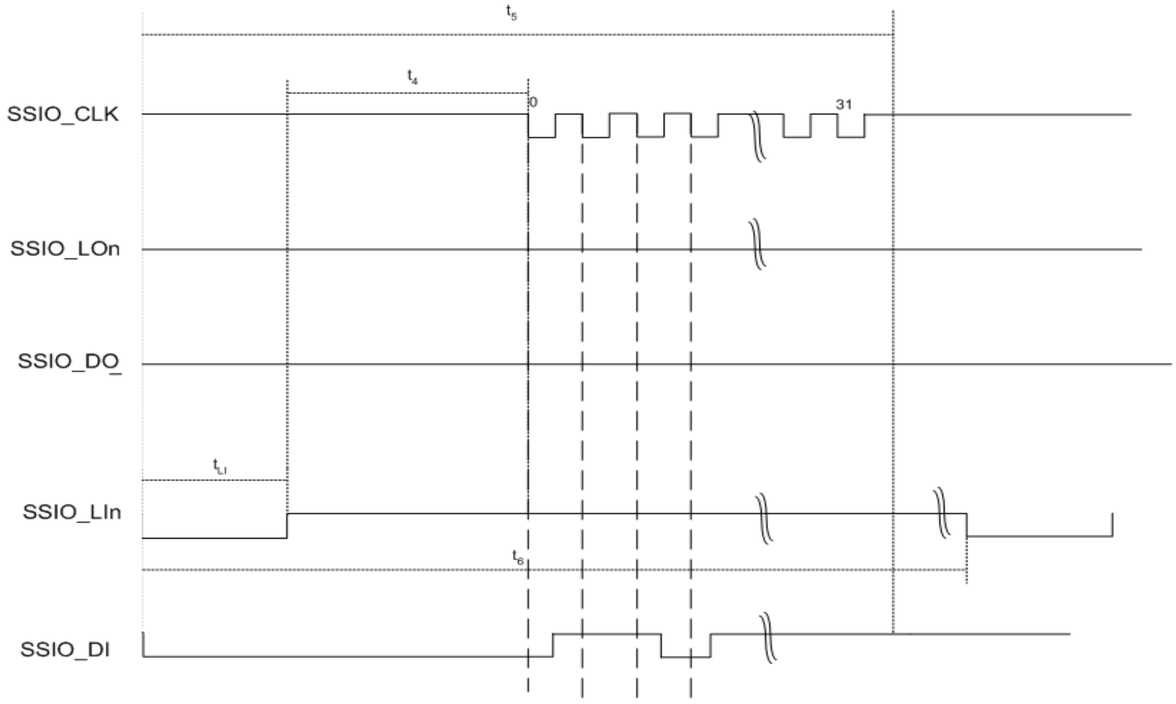


Figure 19: Timing Diagram of SSIO Interface for Input

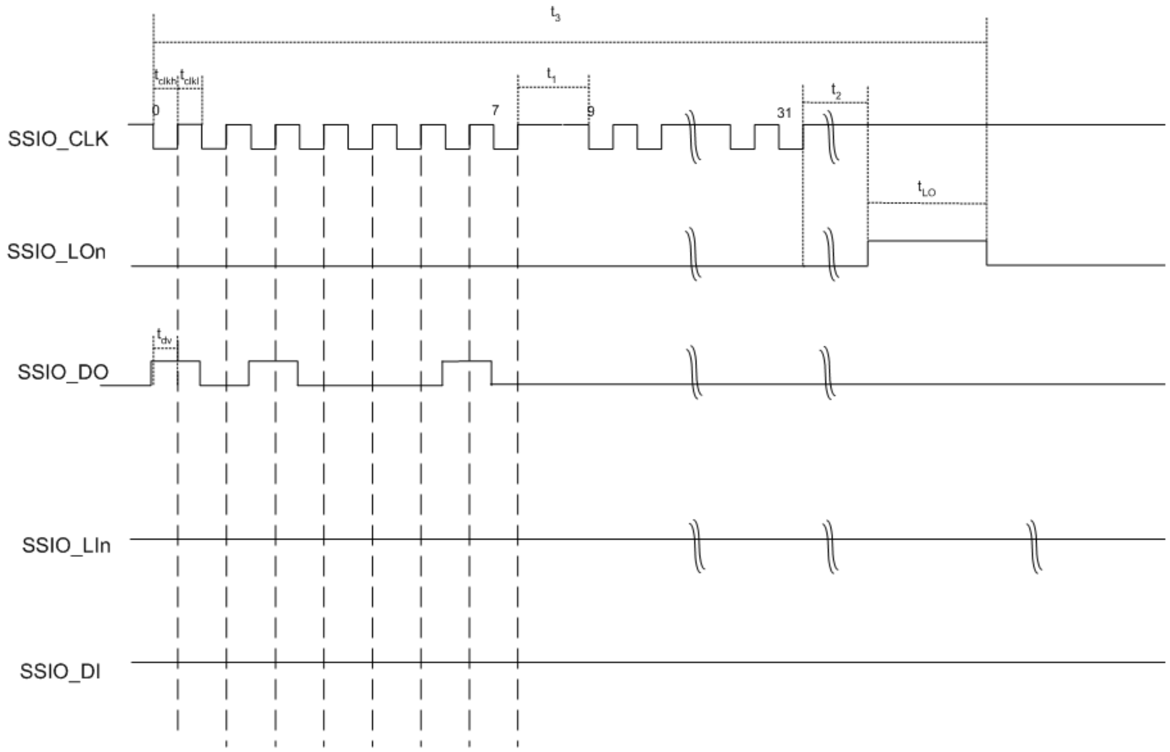


Figure 20: Timing Diagram of SSIO Interface for Output

13.1.6 Diagnostic Interface

Pins 27-28 represent the diagnostic interface of the netIC Communication ICs. It consists of:

Signal	Pin	Description
DIAG_TXD	27	This is the transmit data signal of the Diagnostic-Interface.
DIAG_RXD	28	This is the receive data signal of the Diagnostic-Interface.

Table 76: Pin Assignment Diagnostic Interface

The diagnostic interface of the netIC consists of normal UART signals for transmit and receive data. Normally they are connected with a RS232 driver as the physical interface to a host or a PC.

The default interface settings of the diagnostic serial interface at start-up are:

- 9600 Baud
- even parity
- 1 stop bit
- 8 data bits

In diagnostic mode the following restrictions have to be taken into account:
A connection using the diagnostic interface interrupts the communication of the serial host interface (Modbus RTU).
In diagnostic mode the *Output LEDs DO0-DO15* are not serviced and the DIP switches are not read out.

NOTICE

Malfunction!

If the serial diagnosis interface of the **NIC 50-RE** (Connectors DIAG_TXD/DIAG_RXD) is not connected, then an external pull-up-resistor of 10 kΩ is required in your hardware design at DIAG_RXD. Omitting this pull-up resistor will cause to severe boot-up problems!

This does not apply for the netIC Fieldbus DIL-32 Communication ICs of the NIC 50 family and to NIC 50-RE Revision 4 or higher (those devices include an internal pull-up-resistor)!

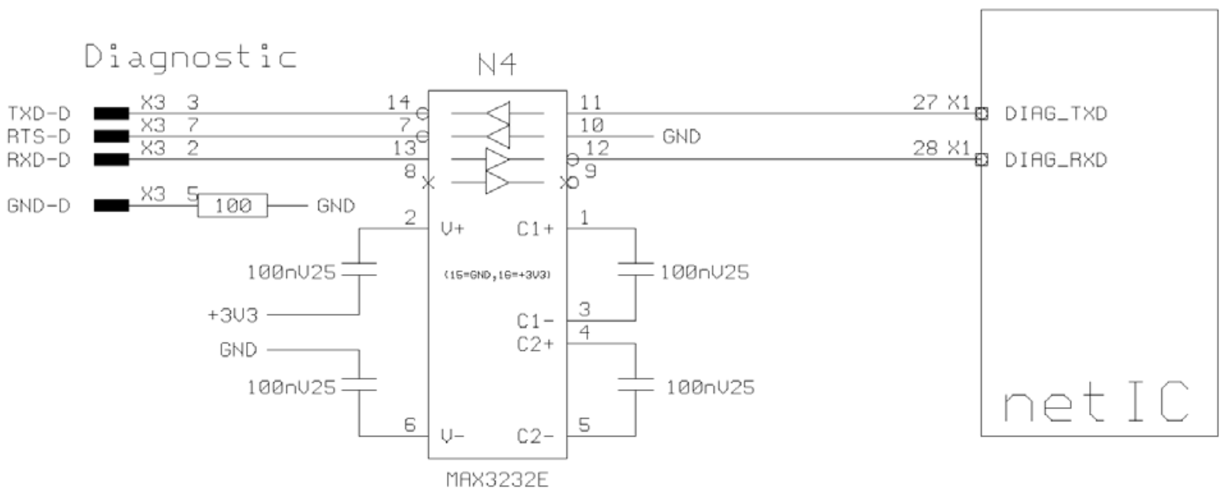


Figure 21: Proposal for the Design of the Diagnostic Interface LED Signals

13.1.7 LED Signals

Most of the standardized Protocol Stacks have defined LEDs to display status and error information of the Communication Interface. Related to the firmware the function can be different. The following list gives an overview:

LED Name	Description
FBLED	Is a general LED signal for the Diag/Config status. This LED signal is active high.
COM green (STA)	Is an active high LED signal available at the NIC 50-RE, drives a green LED and shows the operating status of the communication interface.
COM red (ERR)	Is an active high LED signal available at the NIC 50-RE, drives a red LED and shows the failure status of the communication interface.
LINK0 / 1	Are defined for Ethernet systems only to display the Link status. Because the isolation is done by a magnetic on the host board it is not necessary to save these pins. These signals are active low and the color of the LEDs are normally green.
TX/RX 0/1	Are defined for Ethernet Systems only to display the Activity status. Because the isolation is done by a magnetic on the host board it is not necessary to save these pins. These signals are active low and the color of the LEDs are normally yellow.

Table 77: Explanation of LED Signals

Note: All LED signals can drive a current of up to 6 mA.

Note: It is recommended to assign LEDs to these signal in your design of the host system, at least for signals STA and ERR. For instance, this can be accomplished according to the solution on the evaluation board, where one common duo LED (COM, red/green) has been used for signals STA and ERR.

13.2 Module-specific Information on the netIC

13.2.1 Real-Time-Ethernet DIL-32 Communication IC NIC 50-RE

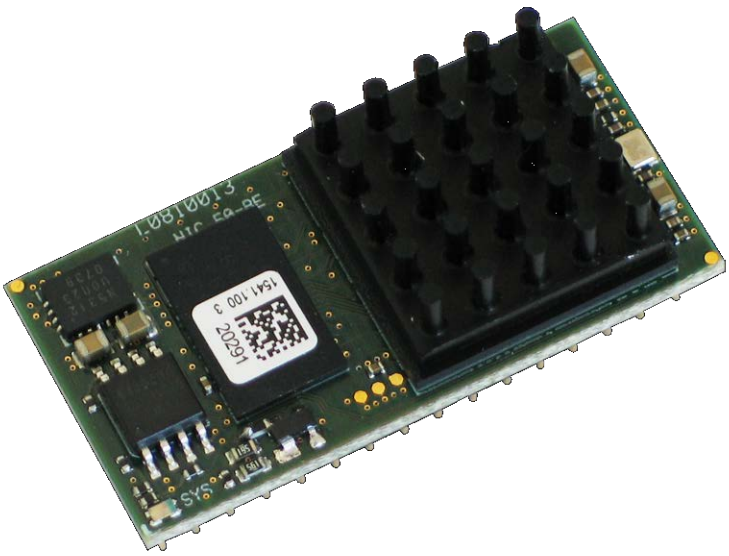


Figure 22: Photo NIC 50-RE with original Hilscher Heat Sink

13.2.1.1 NIC 50-RE Block Diagram

The following block diagram illustrates both how to apply the NIC 50-RE and how it is structured internally:

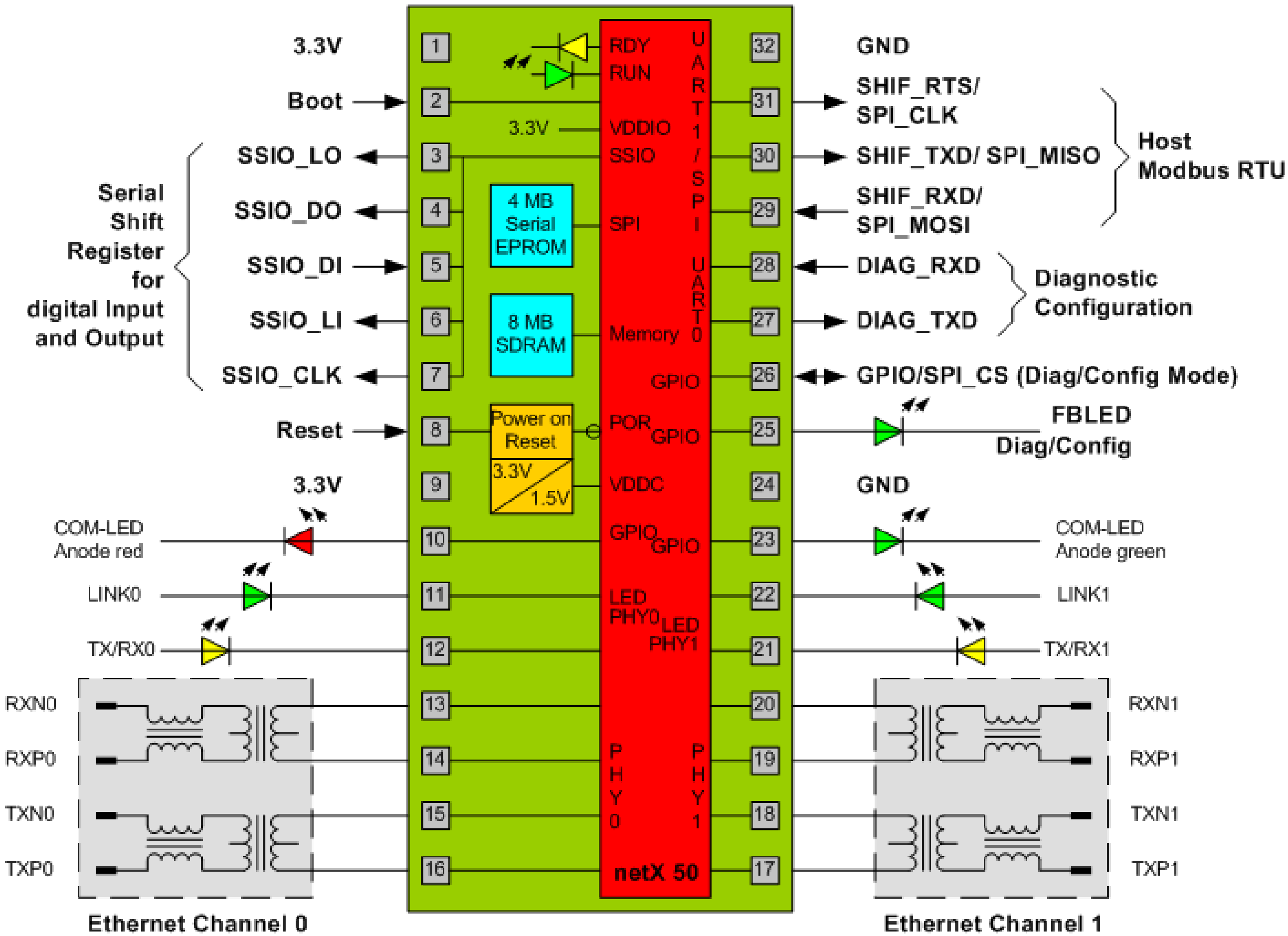


Figure 23: NIC 50-RE Block Diagram - External Connections and internal Structure

13.2.1.2 Pin Assignment NIC 50-RE

The schematic illustration shows the pin assignment of the NIC 50-RE:

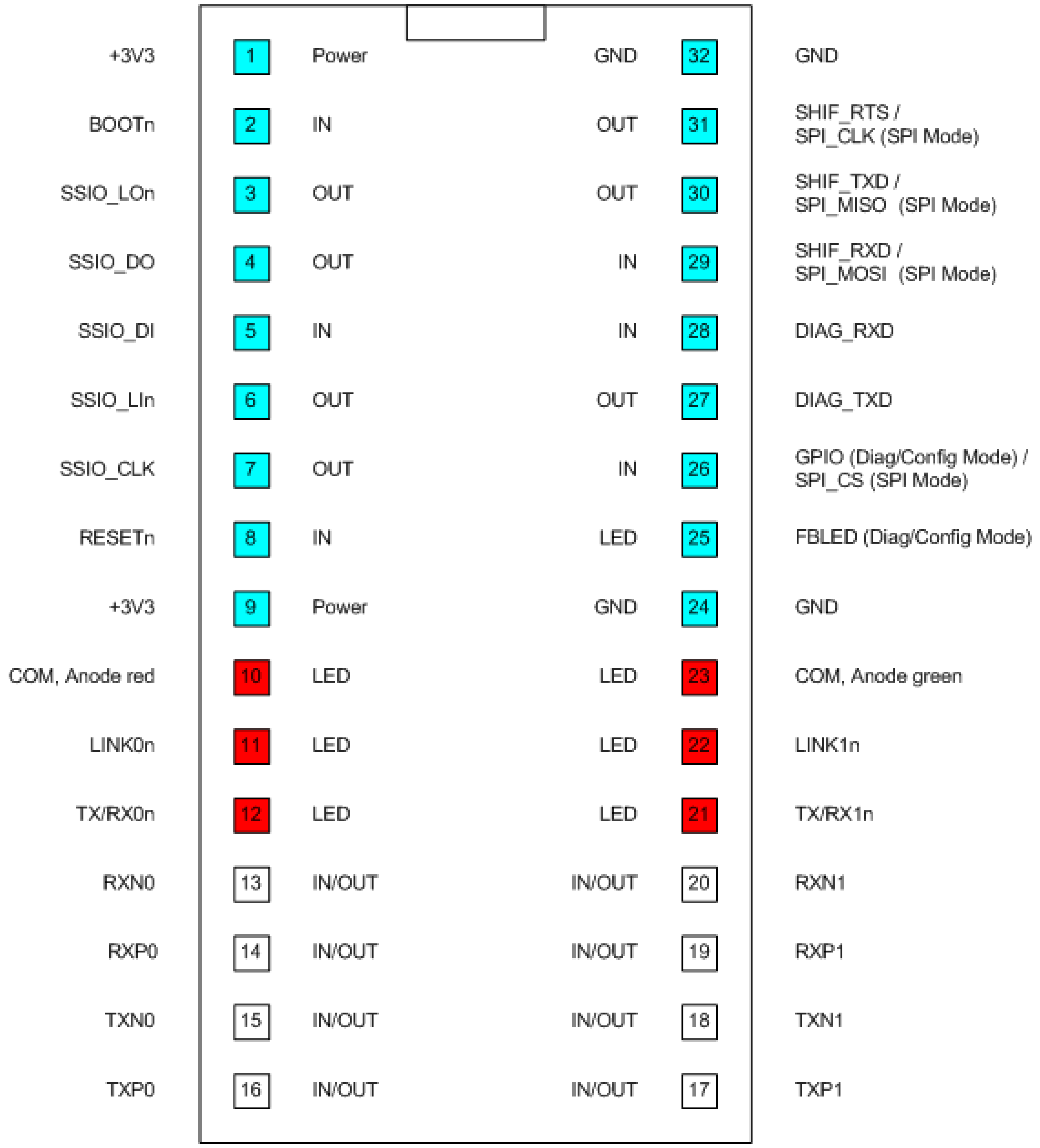


Figure 24: Pinning of NIC 50-RE

The pins marked blue are the standard pins which do not depend from the Real-Time Ethernet or Fieldbus system. The red pins are used for LED signals. The white pins depend on the communication system.

The following table explains the assignment of pins and signals and also provides the direction and the meaning of these signals:

Pin	Signal	Direction	Explanation
10	COM, red	Output	COM-LED - Anode red - Error
11	LINK0n	Output	Ethernet Channel 0 - Link-LED
12	TX/RX0n	Output	Ethernet Channel 0 - Activity-LED
13	RXN0	In/Output	Ethernet Channel 0 - Receive Data minus
14	RXP0	In/Output	Ethernet Channel 0 - Receive Data plus
15	TXN0	In/Output	Ethernet Channel 0 - Transmit Data minus
16	TXP0	In/Output	Ethernet Channel 0 - Transmit Data plus
17	TXP1	In/Output	Ethernet Channel 1 - Transmit Data plus
18	TXN1	In/Output	Ethernet Channel 1 - Transmit Data minus
19	RXP1	In/Output	Ethernet Channel 1 - Receive Data plus
20	RXN1	In/Output	Ethernet Channel 1 - Receive Data minus
21	TX/RX1n	Output	Ethernet Channel 1 - Activity-LED
22	LINK1n	Output	Ethernet Channel 1 - Link-LED
23	COM, green	Output	COM-LED - Anode green - Status

Table 78: Pinning of NIC 50-RE

The signals can be grouped as follows:

- Signals 11 to 16 belong to Ethernet_channel_0.
- Signals 17 to 22 belong to Ethernet_channel_1.
- Signals 10 and 23 belong to various LEDs.

Note: Due to the auto-crossover feature of the netX PHYs, receive and transmit of each Ethernet channel may be swapped.

13.2.1.3 Real-Time-Ethernet Interface of NIC 50-RE

Interface Description

The netIC Real-Time Ethernet DIL-32 Communication IC NIC 50-RE drives two Ethernet ports and has an internal switch and hub functions, respectively the different circuits which are related to the special features of some Real-Time-Ethernet systems to build up a line structure.

The external interface to the Ethernet lines is very simple because the PHYs are already integrated on the NIC 50-RE. Using RJ45 ports with integrated magnetics only a few resistors and capacitors are necessary to match the line impedance.

Pins 11-22 represent the Ethernet Interface of the NIC 50-RE. It consists of 2 channels, namely

- Pins 11-16 represent the interface of Ethernet channel 0 of the NIC 50-RE.
- Pins 17-22 represent the interface of Ethernet channel 1 of the NIC 50-RE.

Note: The device supports the Auto-Crossover function. Due to this fact the signals RX and TX may be switched.

The following assignment of pins and signals has been made:

Ethernet Channel	Signal	Pin	Description
Ethernet Channel 0	LINK0n	11	This signal controls the Link LED of Ethernet Port 0. The signal is active low and has to be connected at the cathode of the LED over an appropriate current resistor.
	TX/RX0n	12	This signal controls the Transmit/Receive or Activity LED of Ethernet Port 0. The signal is active low and has to be connected at the cathode of the LED over an appropriate current resistor.
	RXN0	13	Differential Ethernet receive line of Port 0.
	RXP0	14	
	TXN0	15	Differential Ethernet transmit line of Port 0.
	TXP0	16	
Ethernet Channel 1	LINK1n	22	This signal controls the Link LED of Ethernet Port 1. The signal is active low and has to be connected at Cathode of the LED over an appropriate current resistor.
	TX/RX1n	21	This signal controls the Transmit/Receive or Activity LED of Ethernet Port 1. The signal is active low and has to be connected at Cathode of the LED over an appropriate current resistor.
	RXN1	20	Differential Ethernet receive line of Port 1.
	RXP1	19	
	TXN1	18	Differential Ethernet transmit line of Port 1.
	TXP1	17	

Table 79: Pin Assignment Ethernet Interface

Design Recommendations

For termination of the center tap of the transformer and the unused cable lines a resistor and capacitor combination (1 nF/ 2000 V; 75 Ω) has to be connected like in the following schematic.

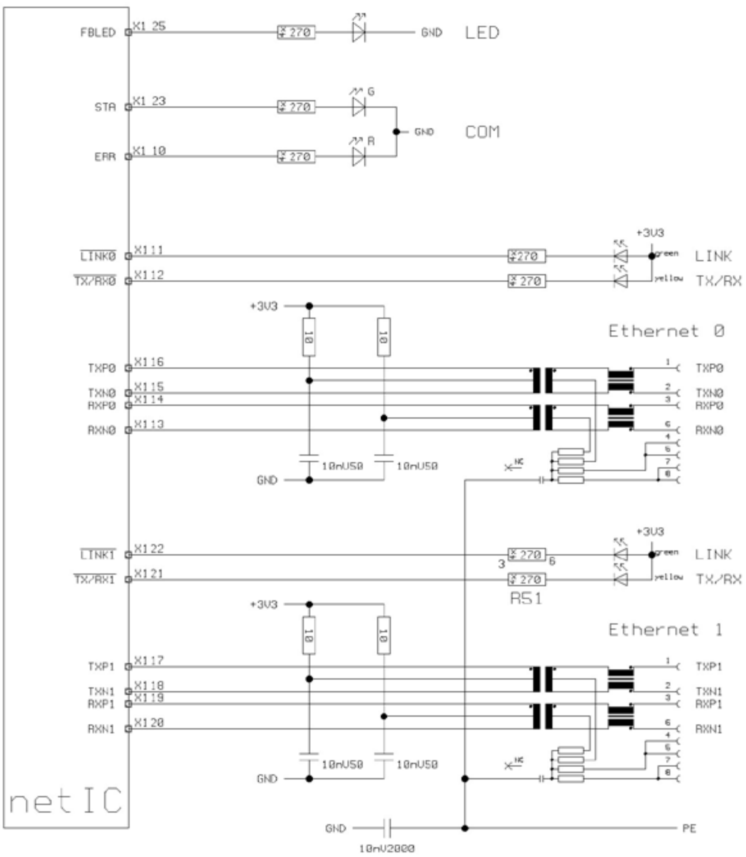


Figure 25: Proposal for the Design of the Real-Time-Ethernet Interface of the NIC 50-RE

The transformers 2x H1102 or 1x H1270 or similar can be used (Pulse).

The following requirements apply for suitable magnetics:

- Symmetric type
- Ratio 1.1
- Center tap

For noise immunity we recommend to connect the housing of the RJ45 connector to earth ground directly.

13.2.2 netIC CANopen DIL-32 Communication IC NIC 50-COS

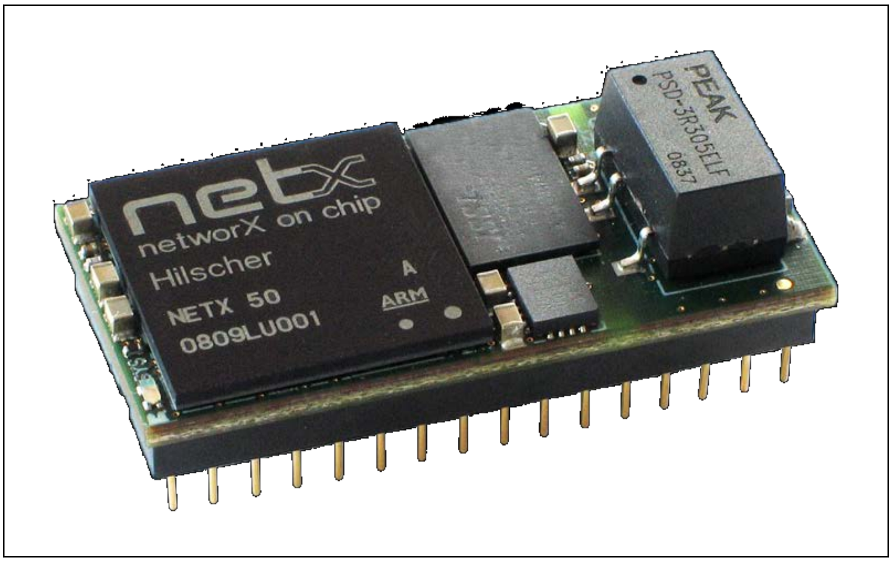


Figure 26: Photo NIC 50-COS

13.2.2.1 NIC 50-COS Block Diagram

The following block diagram illustrates both how to apply the NIC 50-COS and how it is structured internally:

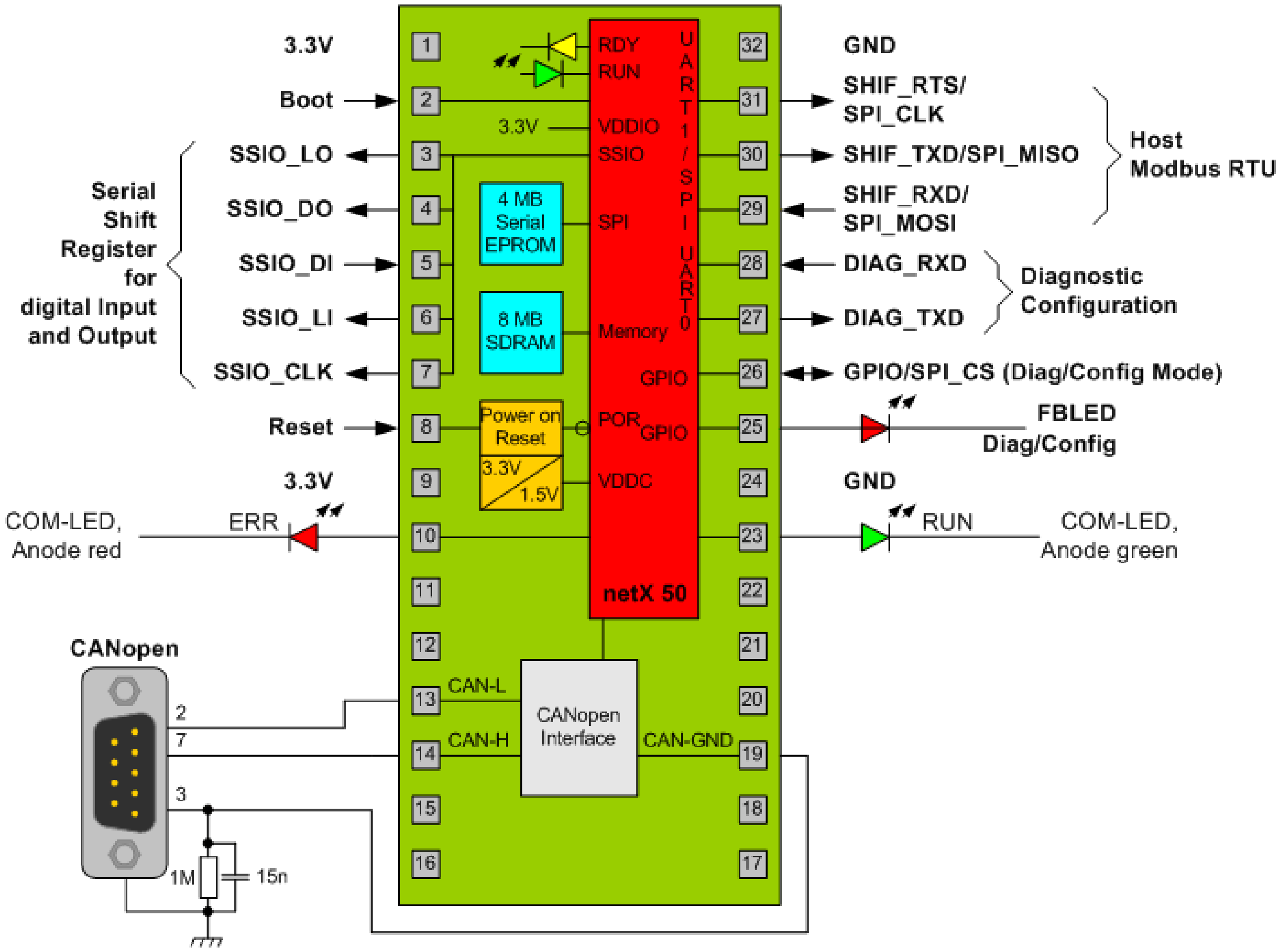


Figure 27: NIC 50-COS Block Diagram - External Connections and internal Structure

13.2.2.2 Pin Assignment NIC 50-COS

The schematic illustration shows the pin assignment of the NIC 50-COS:

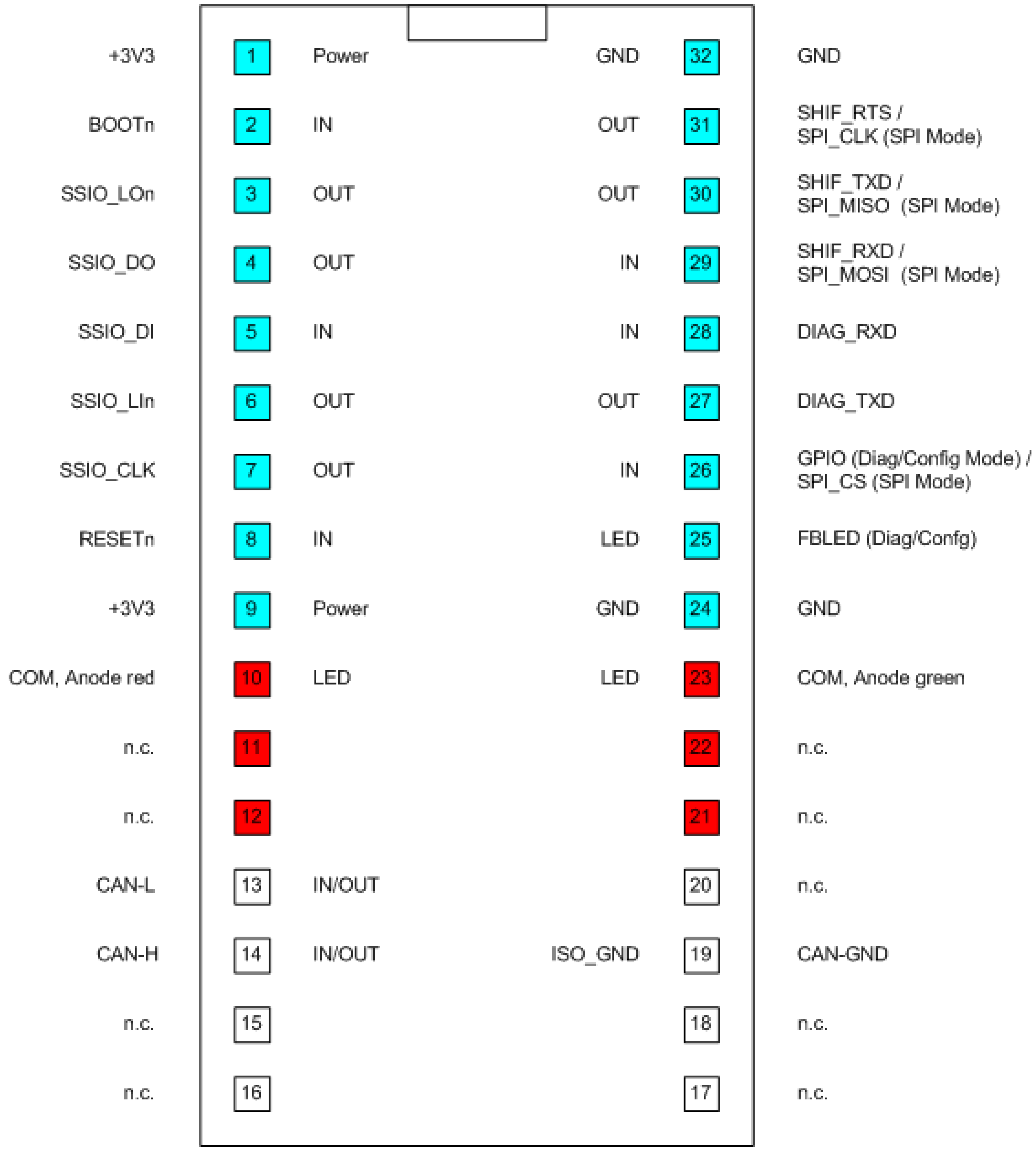


Figure 28: Pinning of NIC 50-COS

- Pins marked in white are specific to CANopen.
- Pins marked in light blue are common to all netIC Communication ICs.
- Pins marked in red are used for LED signals or not used.

The following table explains the assignment of pins and signals and also provides the direction and the meaning of these signals:

Pin	Signal	Direction	Explanation
10	COM red	Output	COM-LED - Anode red
11	-		Not connected
12	-		Not connected
13	CANL	In-/Output	CAN Signal L (Pin 2 of connector)
14	CANH	In-/Output	CAN Signal H (Pin 7 of connector)
15	-		Not connected
16	-		Not connected
17	-		Not connected
18	-		Not connected
19	CAN-GND		CAN Ground (Pin 3 of connector)
20	-		Not connected
21	-		Not connected
22	-		Not connected
23	COM, green	Output	COM-LED - Anode green

Table 80: Pinning of NIC 50-COS

The signals can be grouped as follows:

- Signals 13, 14 and 19 belong to the CANopen Interface.
- Signals 10 and 23 belong to various LEDs.

13.2.2.3 The CANopen Interface of the NIC 50-COS

Interface Description

The NIC 50-COS provides a single CANopen Slave interface for the connection with a CANopen Master. The CANopen interface of the NIC 50-COS is designed as potential free ISO 11898 interface.

All electric signals are conforming to the CANopen specification EN 50325/4.

Pins 13 to 14 and 19 to 20 belong to the CANopen interface of the NIC 50-COS.

The assignment of pins to signals is as follows:

Signal	Pin CANopen	Pin NIC 50-COS	Description
CANL	2	13	CAN Signal L
CANH	7	14	CAN Signal H.
CAN-GND	3	19	Ground for CANopen.

Table 81: Pin Assignment CANopen Interface

All signals not mentioned here are not connected.

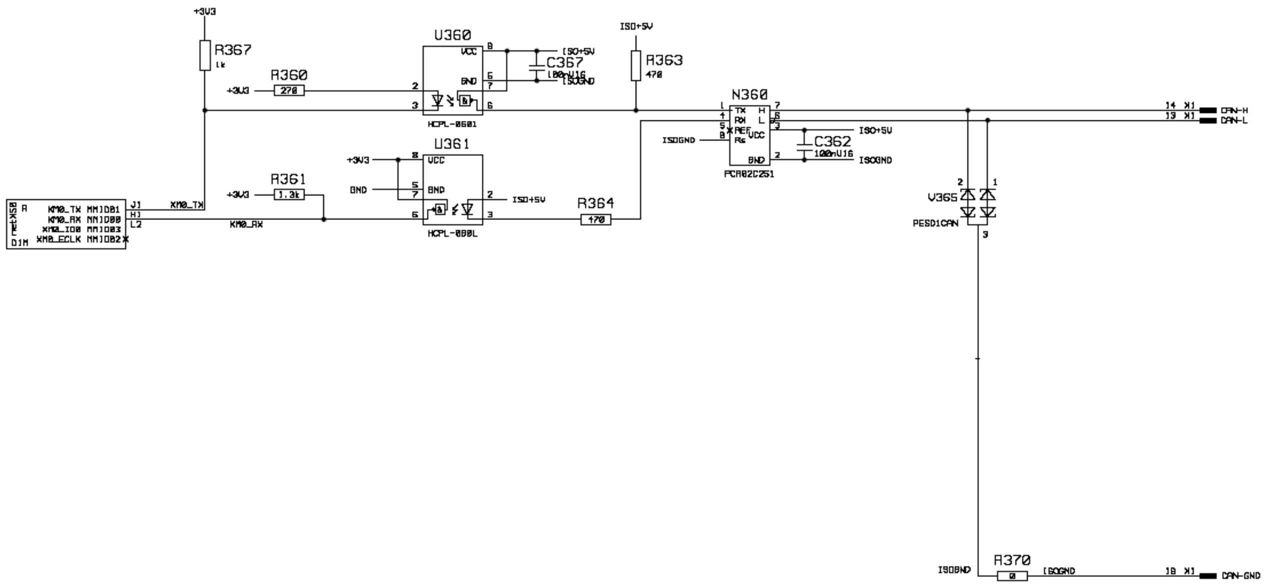


Figure 29: Plan of CANopen Interface of NIC 50-COS

Design Recommendations

The NIC 50-COS can be connected to a 9 pole D-Sub-connector similarly to the design of the NICEB-AIF-CO adapter, see section *CANopen-Adapter NICEB-AIF-CO* on page 159.

The signals and their corresponding pins on the connector are

Signal	Pin at NIC 50-COS	Pin CANopen	Description of signal
CAN-L	13	2	CANopen -Data line L (negative).
CAN-H	14	7	CANopen -Data line H (positive).
CAN-GND	19	3	Ground for CANopen.

Table 82: CANopen Interface of the NIC 50-COS - Signals and Pins

You should integrate an RC combination (1 MΩ/15 nF) between **CAN-GND** and the protective earth into your design according to the following schematic illustration:

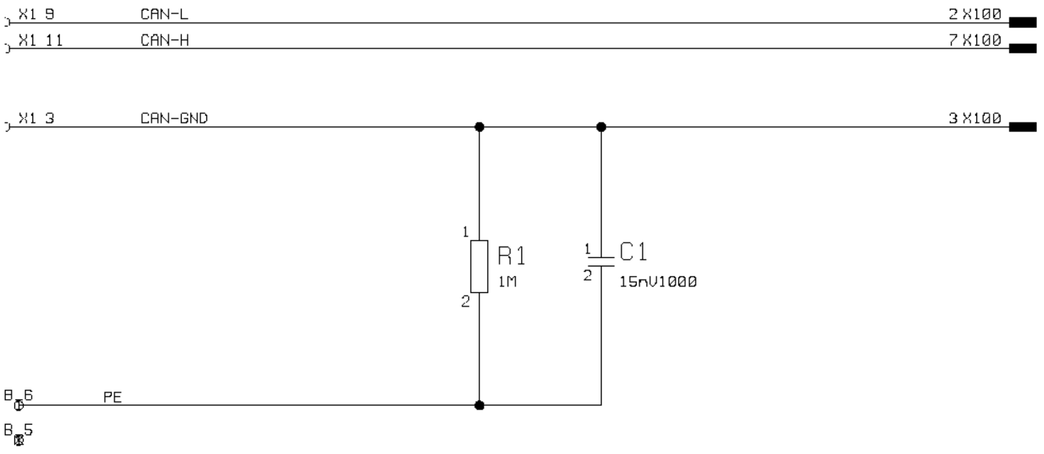


Figure 30: Proposal for the Design of the CANopen Interface of the NIC 50-COS

13.2.3 netIC DeviceNet DIL-32 Communication IC NIC 50-DNS

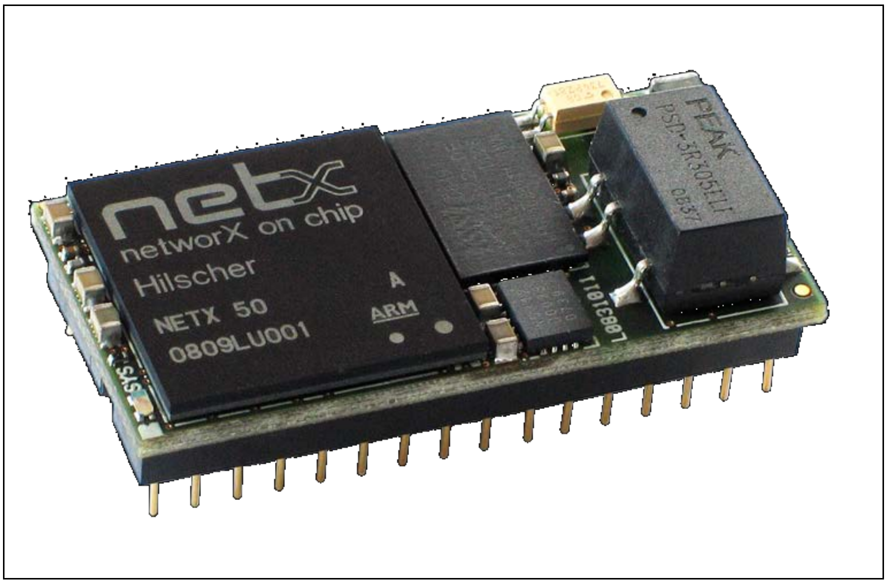


Figure 31: Photo NIC 50-DNS

13.2.3.1 NIC 50-DNS Block Diagram

The following block diagram illustrates both how to apply the NIC 50-DNS and how it is structured internally:

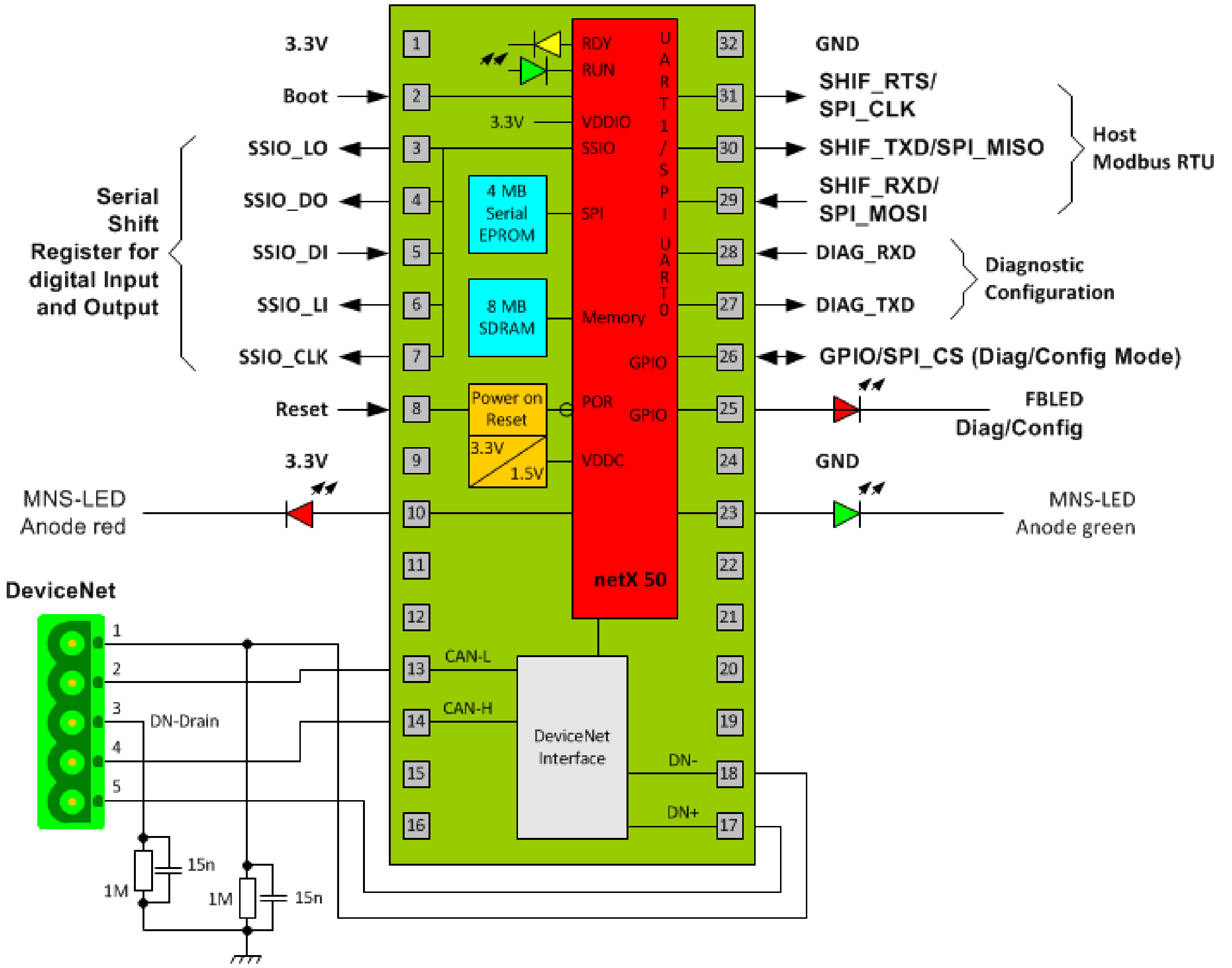


Figure 32: NIC 50-DNS Block Diagram - External Connections and internal Structure

13.2.3.2 Pin Assignment NIC 50-DNS

The schematic illustration shows the pin assignment of the NIC 50-DNS:

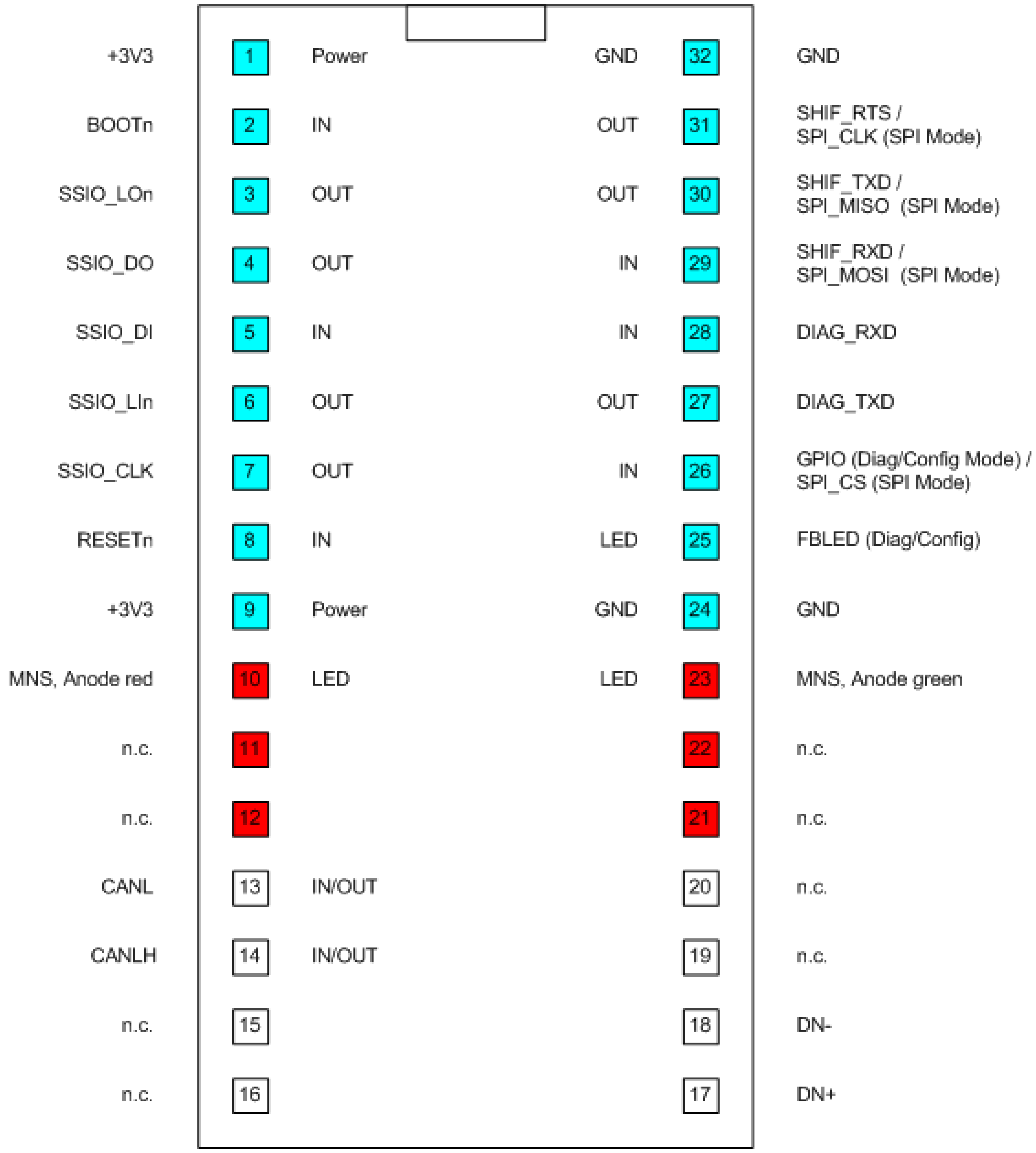


Figure 33: Pinning of NIC 50-DNS

- Pins marked in white are specific to DeviceNet.
- Pins marked in light blue are common to all netIC Communication ICs.
- Pins marked in red are used for LED signals or not used.

The following table explains the assignment of pins and signals and also provides the direction and the meaning of these signals:

Pin	Signal	Direction	Explanation
10	MNS	Output	MNS-LED, Anode red - Module Network Status
11	-		Not connected
12	-		Not connected
13	CANL	In-/Output	CAN Signal L (Pin 2 DeviceNet Connector)
14	CANH	In-/Output	CAN Signal H (Pin 4 DeviceNet Connector)
15	-		Not connected
16	-		Not connected
17	DN+	Input	24 V (Pin 5 DeviceNet Connector)
18	DN-		Ground DeviceNet (Pin 1 DeviceNet Connector)
19	-		Not connected
20	-		Not connected
21	-		Not connected
22	-		Not connected
23	MNS	Output	MNS–LED, Anode green – Module Network Status

Table 83: Pinning of NIC 50-DNS

The signals can be grouped as follows:

- Signals 13, 14, 17 and 18 belong to the DeviceNet Interface.
- Signals 10 and 23 belong to various LEDs.

13.2.3.3 The DeviceNet Interface of the NIC 50-DNS

Interface Description

The NIC 50-DNS provides a single DeviceNet interface for the connection with a DeviceNet Master.

All electric signals are conforming to the DeviceNet standard.

Pins 13, 14, 17 and 18 belong to the DeviceNet interface of the NIC 50-DNS.

The assignment of pins to signals is as follows:

Signal	Pin DeviceNet	Pin NIC 50-DNS	Description
CANL	2	13	CAN Signal L.
CANH	4	14	CAN Signal H.
DN+	5	17	24V for DeviceNet.
DN-	1	18	Ground for DeviceNet.

Table 84: Pin Assignment DeviceNet Interface

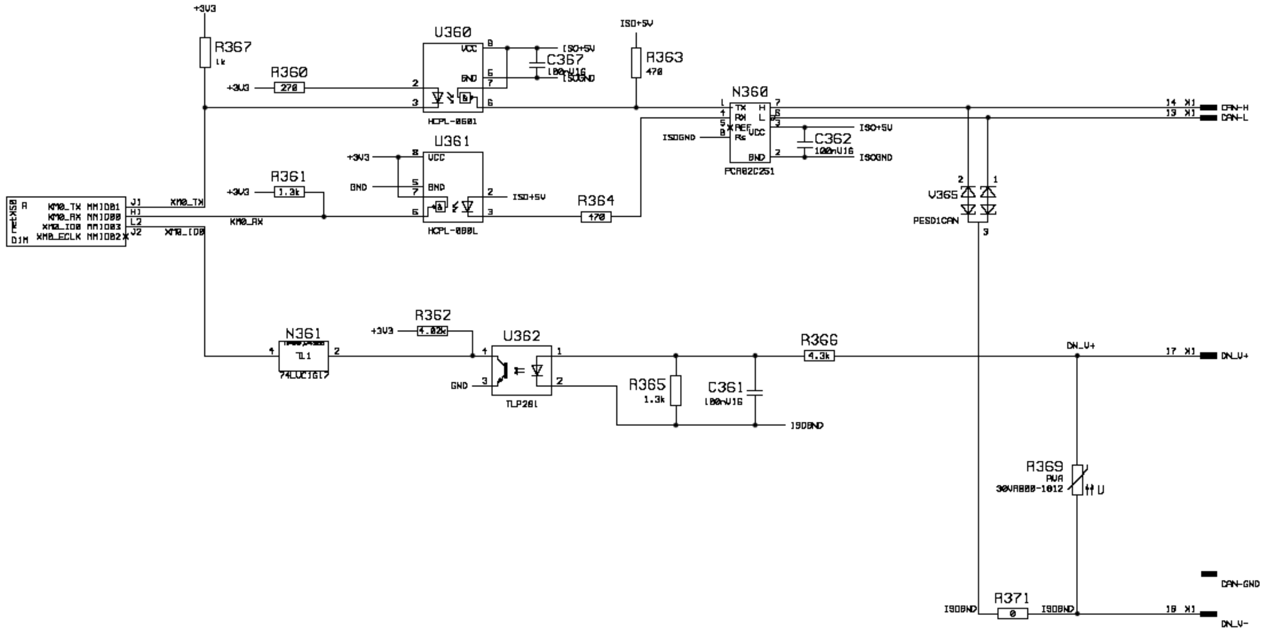


Figure 34: Plan of DeviceNet Interface of NIC 50-DNS

Design Recommendations

The NIC 50-DNS can be connected to a 5 pole Combicon connector similarly to the design of the NICEB-AIF-DN adapter, see section *DeviceNet-Adapter NICEB-AIF-DN* of this document.

The signals and their corresponding pins on the connector are

Signal	Pin at NIC 50-DNS	Pin DeviceNet	Description of signal
CAN-L	13	2	CAN-Data line L.
CAN-H	14	4	CAN-Data line H.
DN+	17	5	24 V power line for DeviceNet.
DN-	18	1	Ground for DeviceNet.
DN-Drain	-	3	Shield

Table 85: DeviceNet Interface of the NIC 50-DNS - Signals and Pins

You should integrate an RC combination (1MΩ / 15 nF) between **DN-** and protective earth and also between **DN-Drain** and protective earth into your design according to the following schematic illustration:

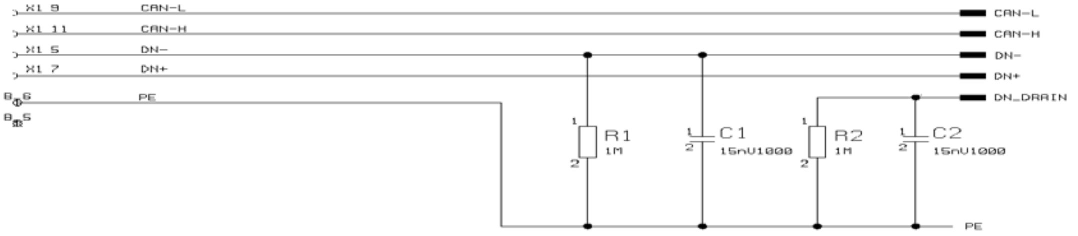


Figure 35: Proposal for the Design of the DeviceNet Interface of the NIC 50-DNS

13.2.4 netIC PROFIBUS-DP DIL-32 Communication IC NIC 50-DPS

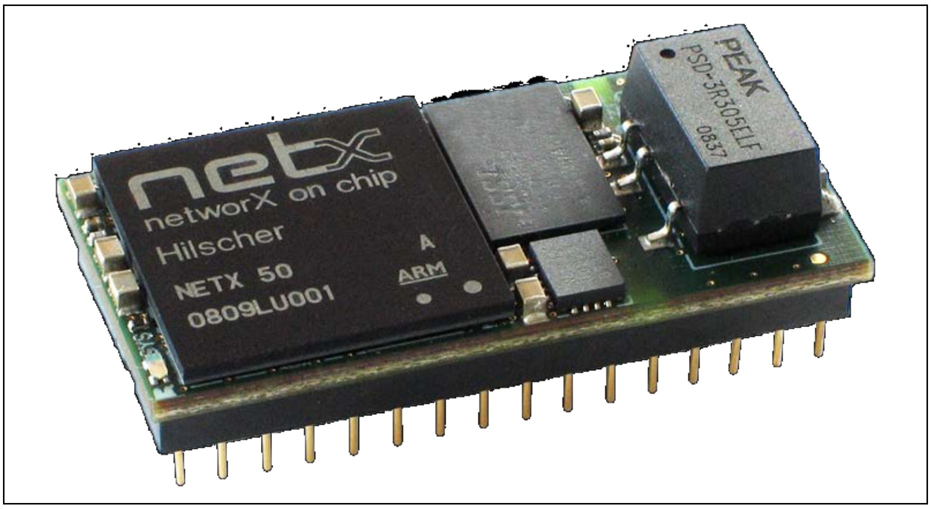


Figure 36: Photo NIC 50-DPS

13.2.4.1 Block Diagram NIC 50-DPS

The following block diagram illustrates both how to apply the NIC 50-DPS and how it is structured internally:

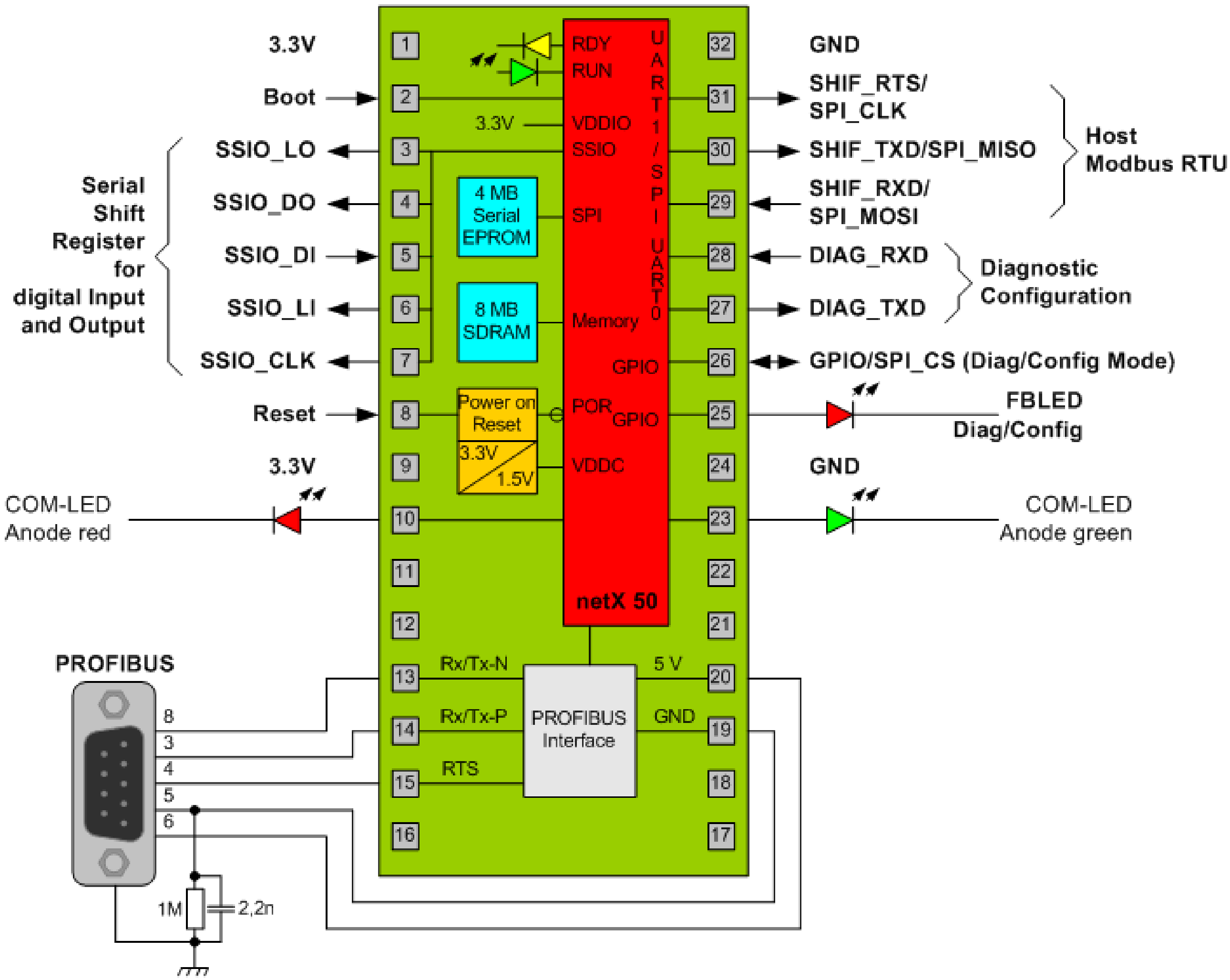


Figure 37: NIC 50-DPS Block Diagram - External Connections and internal Structure

13.2.4.2 Pin Assignment NIC 50-DPS

The schematic illustration shows the pin assignment of the NIC 50-DPS:

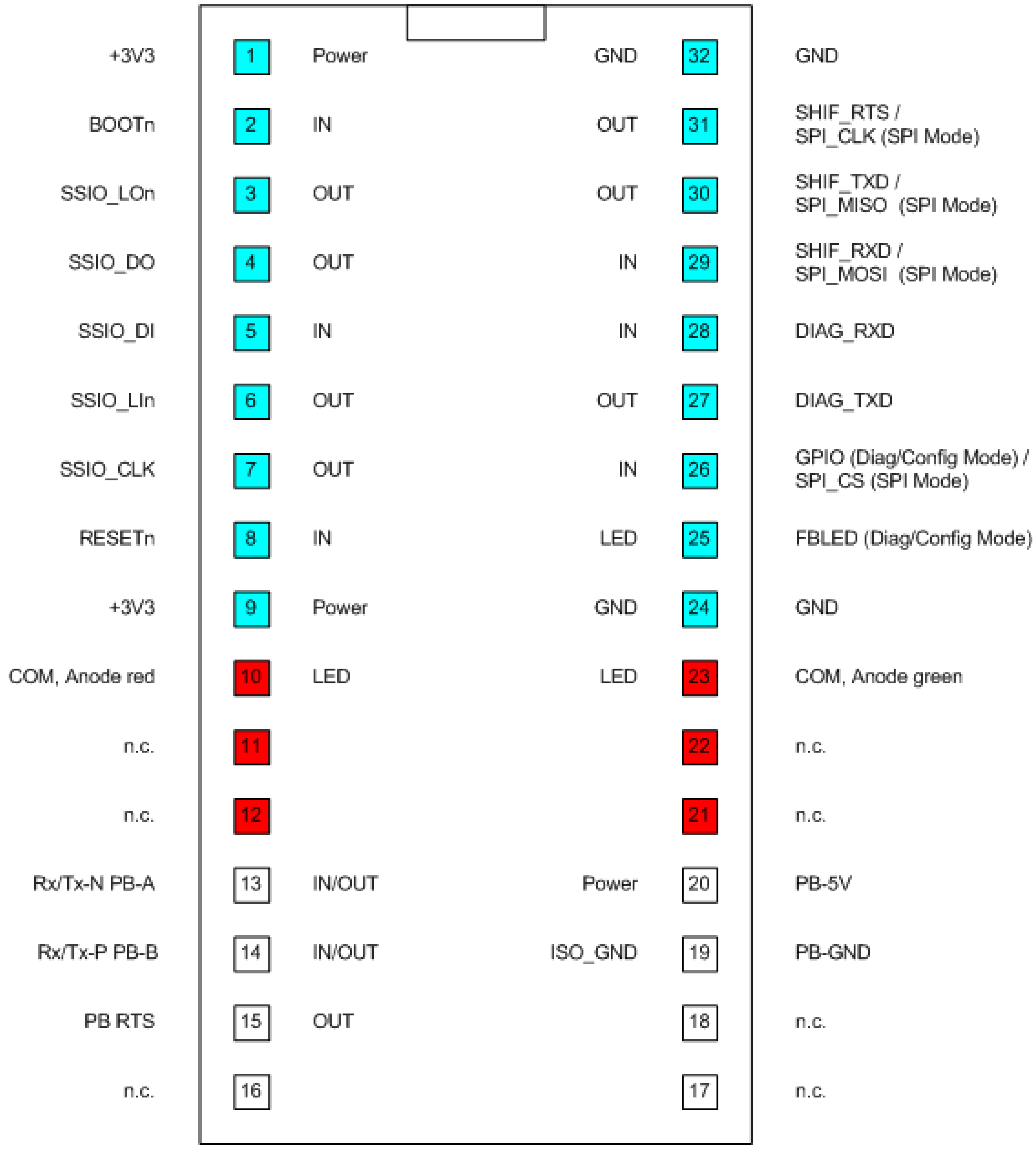


Figure 38: Pinning NIC 50-DPS

- Pins marked in white are specific to PROFIBUS-DP.
- Pins marked in light blue are common to all netIC Communication ICs.
- Pins marked in red are used for LED signals or not used.

The following table explains the assignment of pins and signals and also provides the direction and the meaning of these signals:

Pin	Signal	Direction	Explanation
10	COM, red	Output	COM-LED, Anode red
11	-		Not connected
12	-		Not connected
13	Rx/Tx-N, PB-A	In-/Output	Profibus Signal A (Pin 8 of connector)
14	Rx/Tx-P, PB-B	In-/Output	Profibus Signal B (Pin 3 of connector)
15	PB-RTS	Output	Profibus Signal RTS (Pin 4 of connector)
16	-		Not connected
17	-		Not connected
18	-		Not connected
19	PB-GND		Profibus Ground (Pin 5 of connector)
20	PB-5V	Output	Profibus 5V (Pin 6 of connector)
21	-		Not connected
22	-		Not connected
23	COM, green	Output	COM-LED, Anode green

Table 86: Pinning of NIC 50-DPS

The signals can be grouped as follows:

- Signals 13, 14, 19 and 20 belong to the PROFIBUS-DP Interface.
- Signals 10 and 23 belong to various LEDs.

13.2.4.3 The PROFIBUS DP Interface of the NIC 50-DPS

Interface Description

The NIC 50-DPS provides a single PROFIBUS-DP interface for the connection with a PROFIBUS-DP Master. The PROFIBUS-DP interface is designed as potential-free RS-485 interface.

All electric signals are conforming to the PROFIBUS-DP standard.

Pins 13 to 15 and 19 to 20 belong to the PROFIBUS-DP interface of the NIC 50-DPS.

The assignment of pins to signals is as follows:

Signal	Pin PROFIBUS	Pin NIC 50-DPS	Description
RX/TX– (PB-A)	8	13	PROFIBUS-DP-Data line A.
RX/TX+ (PB-B)	3	14	PROFIBUS-DP-Data line B.
RTS	4	15	Return To Send Line for line control.
PB-GND, ISO_GND	5	19	Ground for PROFIBUS-DP.
PB-5V, VP	6	20	5 V power line for PROFIBUS-DP.

Table 87: Pin Assignment PROFIBUS Interface

All signals not mentioned here are not connected.

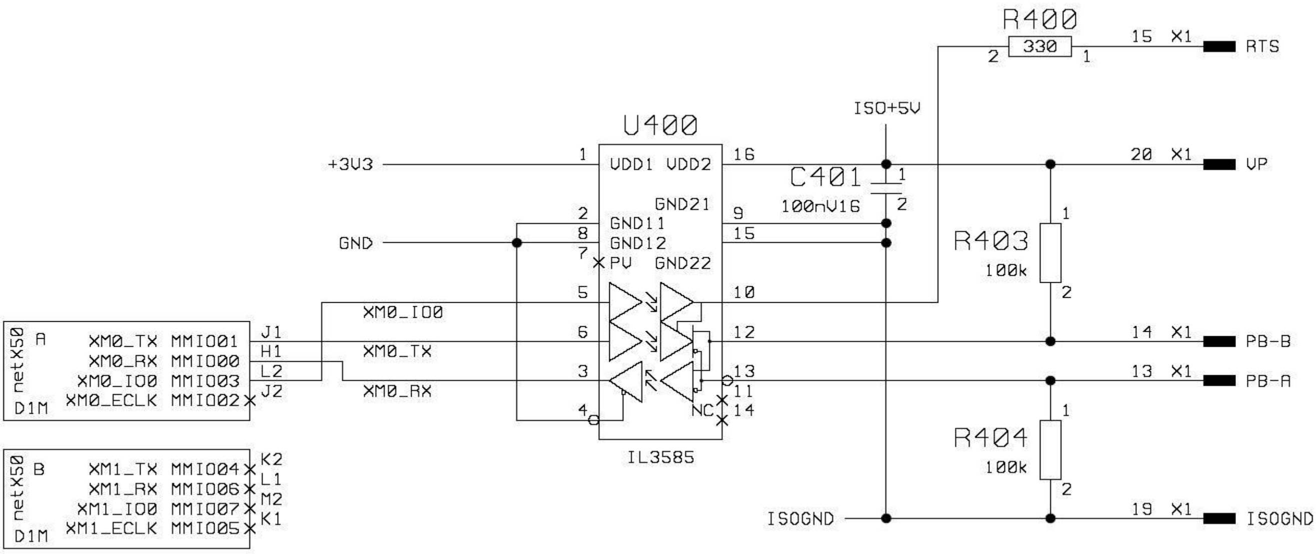


Figure 39: Plan of PROFIBUS DP Interface of NIC 50-DPS

Design Recommendations

The NIC 50-DPS can be connected to a 9 pole connector similarly to the design of the NICEB-AIF-DP adapter, see section *PROFIBUS DP-Adapter NICEB-AIF-DP* of this document.

The signals and their corresponding pins on the connector are

Signal	Pin at NIC 50-DPS	Pin PROFIBUS-DP	Description of signal
Rx/Tx– (PB-A)	13	8	PROFIBUS-DP-Data line A (negative).
Rx/Tx+ (PB-B)	14	3	PROFIBUS-DP-Data line B (positive).
RTS	15	4	Return To Send Line for line control.
PB-GND	19	5	Ground for PROFIBUS-DP.
PB-5V	20	6	5 V power line for PROFIBUS-DP.

Table 88: PROFIBUS DP Interface of the NIC 50-DPS - Signals and Pins

You should integrate an RC combination (1MΩ / 2.2 nF) between **PB-GND** and the protective earth into your design according to the following schematic illustration:

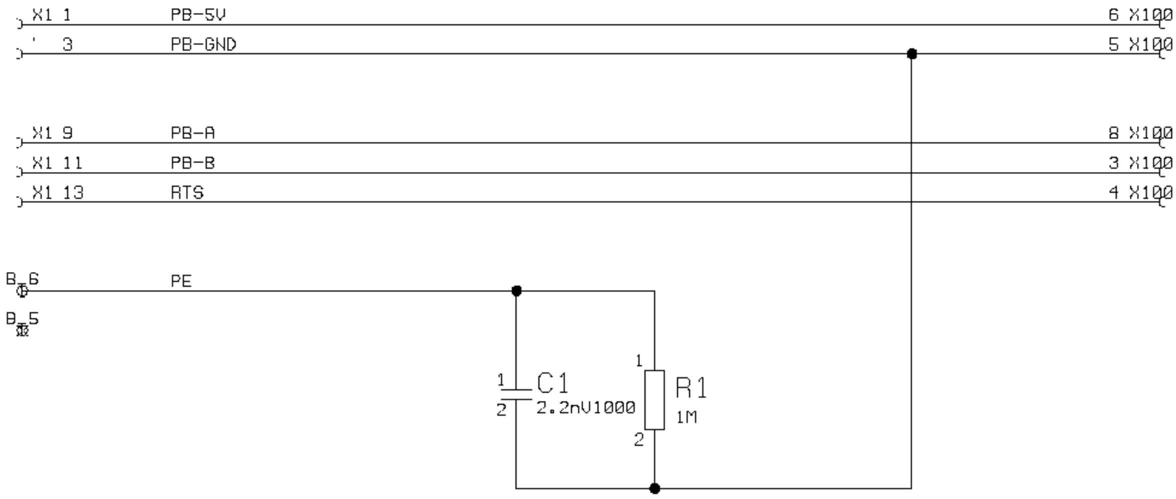


Figure 40: Proposal for the Design of the PROFIBUS DP Interface of the NIC 50-DPS

13.2.5 Real-Time-Ethernet DIL-32 Communication IC NIC 52-RE

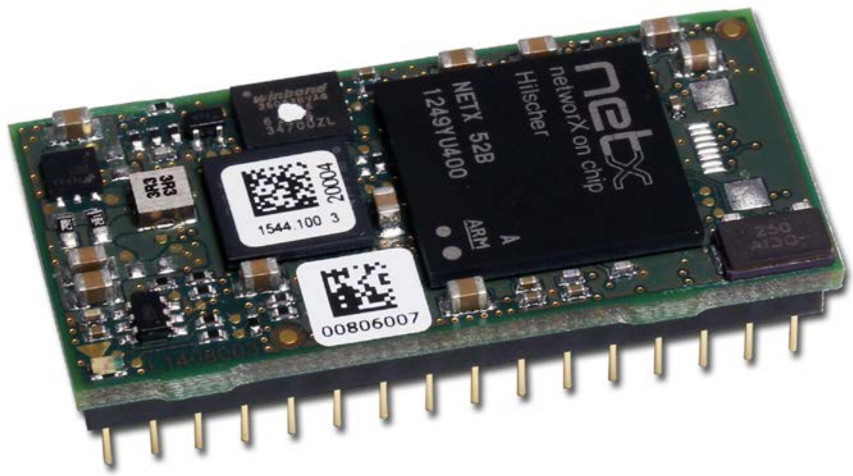


Figure 41: Photo NIC 52-RE (Top view)

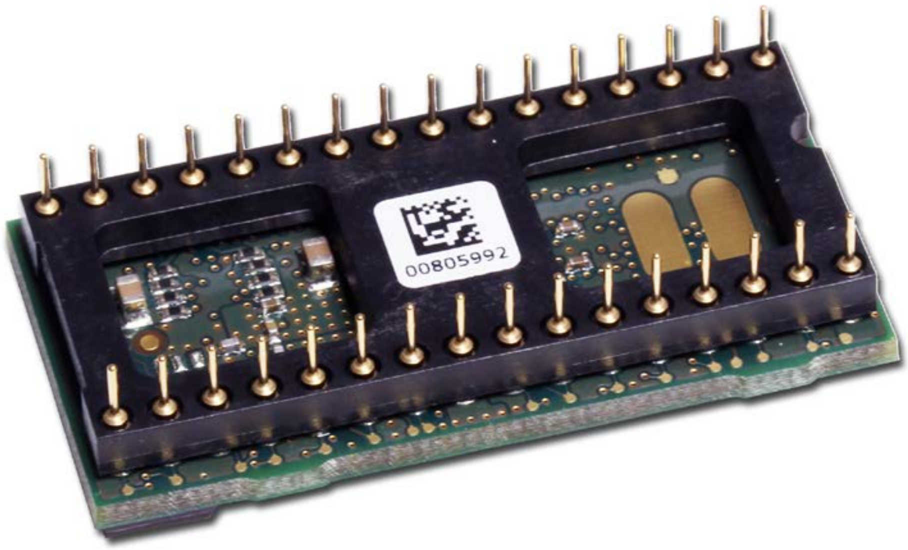


Figure 42: Photo NIC 52-RE (Bottom view)

13.2.5.1 NIC 52-RE Block Diagram

The following block diagram illustrates both how to apply the NIC 52-RE and how it is structured internally:

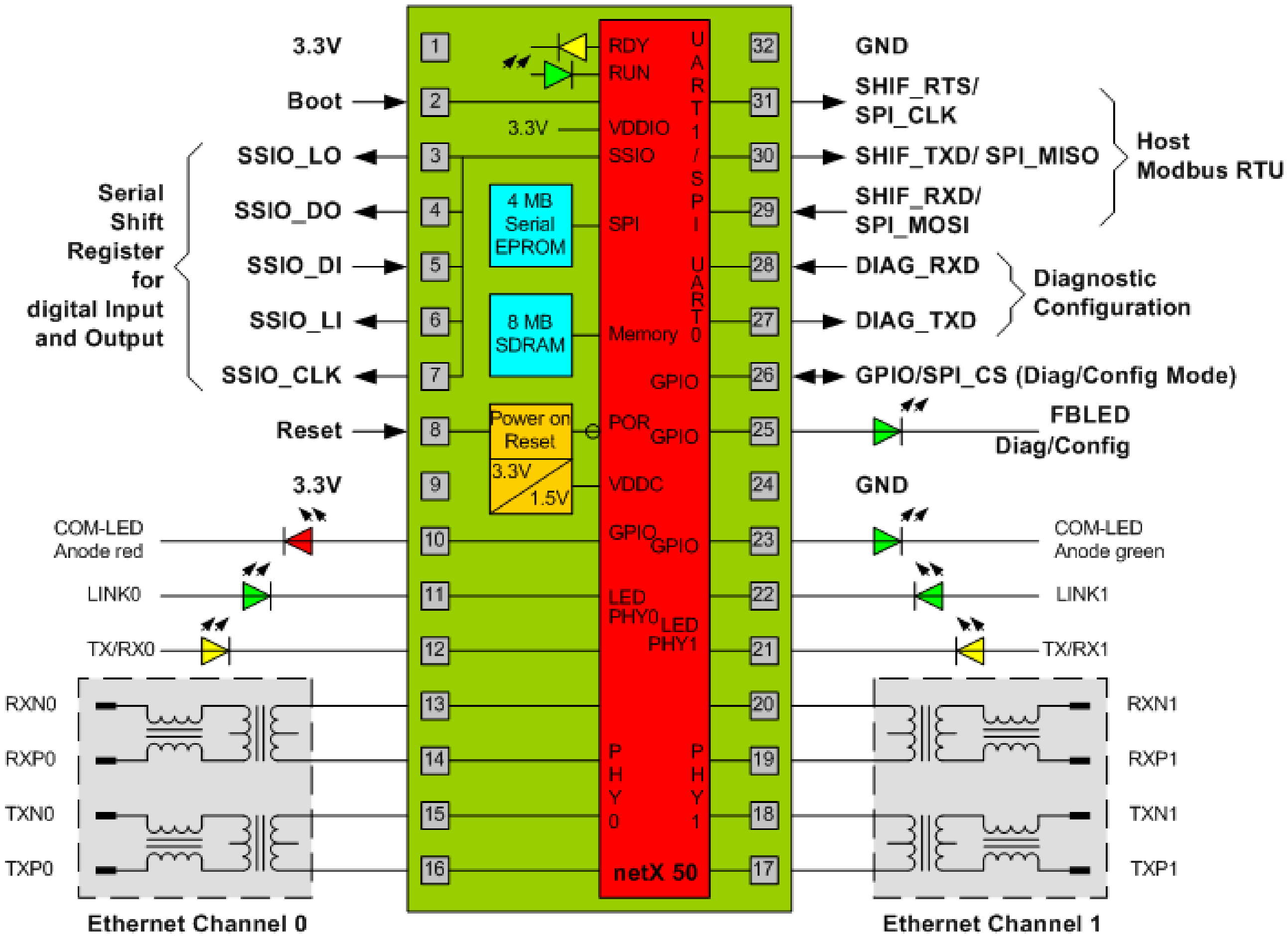


Figure 43: NIC 52-RE Block Diagram - External Connections and internal Structure

13.2.5.2 Pin Assignment NIC 52-RE

The schematic illustration shows the pin assignment of the NIC 52-RE:

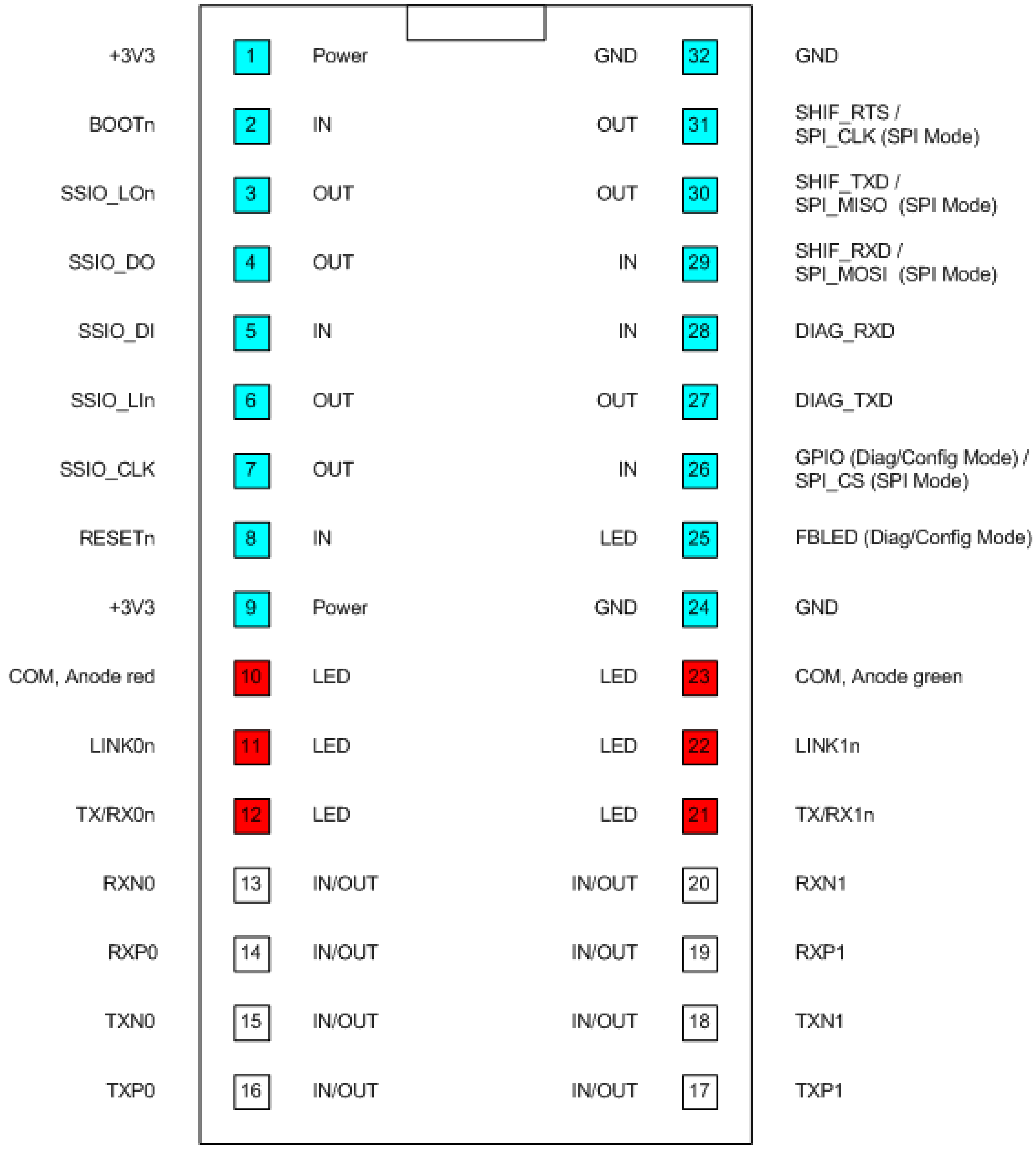


Figure 44: Pinning of NIC 52-RE

The pins marked blue are the standard pins which do not depend from the Real-Time Ethernet or Fieldbus system. The red pins are used for LED signals. The white pins depend on the communication system.

The following table explains the assignment of pins and signals and also provides the direction and the meaning of these signals:

Pin	Signal	Direction	Explanation
10	COM, red	Output	COM-LED - Anode red - Error
11	LINK0n	Output	Ethernet Channel 0 - Link-LED
12	TX/RX0n	Output	Ethernet Channel 0 - Activity-LED
13	RXN0	In/Output	Ethernet Channel 0 - Receive Data minus
14	RXP0	In/Output	Ethernet Channel 0 - Receive Data plus
15	TXN0	In/Output	Ethernet Channel 0 - Transmit Data minus
16	TXP0	In/Output	Ethernet Channel 0 - Transmit Data plus
17	TXP1	In/Output	Ethernet Channel 1 - Transmit Data plus
18	TXN1	In/Output	Ethernet Channel 1 - Transmit Data minus
19	RXP1	In/Output	Ethernet Channel 1 - Receive Data plus
20	RXN1	In/Output	Ethernet Channel 1 - Receive Data minus
21	TX/RX1n	Output	Ethernet Channel 1 - Activity-LED
22	LINK1n	Output	Ethernet Channel 1 - Link-LED
23	COM, green	Output	COM-LED - Anode green - Status

Table 89: Pinning of NIC 52-RE

The signals can be grouped as follows:

- Signals 11 to 16 belong to Ethernet_channel_0.
- Signals 17 to 22 belong to Ethernet_channel_1.
- Signals 10 and 23 belong to various LEDs.

Note: Due to the auto-crossover feature of the netX PHYs, receive and transmit of each Ethernet channel may be swapped.

13.2.5.3 Real-Time-Ethernet Interface of NIC 52-RE

See section *Real-Time-Ethernet Interface of NIC 50-RE* on page 125. The interface description and design recommendations given there also apply for the NIC 52-RE.

14 netIC Evaluation Board NICEB and Accessories

14.1 Evaluation Board NICEB

The evaluation board NICEB is equipped with a DIL-32 socket for one netIC Communication IC and with all interfaces necessary to evaluate its functions (Fieldbus interfaces are provided by a Fieldbus-specific adapter).

In detail, these are

- Two RJ45 ports with integrated magnetic and LINK- / ACTIVITY-LED
- RS232 diagnostic interface with DSUB-9 pin-connector for loading the firmware and configuration
- RS232-/ RS422- or RS485-Host Interface (configurable by jumpers) with DSUB-9 pin-connector
- 16 synchronous serial inputs with DIP switch
  16 synchronous serial inputs at post square connectors
- 16 synchronous serial outputs with LEDs
  16 synchronous serial outputs at post square connectors

Furthermore, the NICEB provides:

- LED for signal FBLED and Duo-Color-LED for signals STA and ERR
- Push buttons for Reset- / Boot- / Configuration
- 24 V power supply

14.1.1 Device Drawing Evaluation Board NICEB

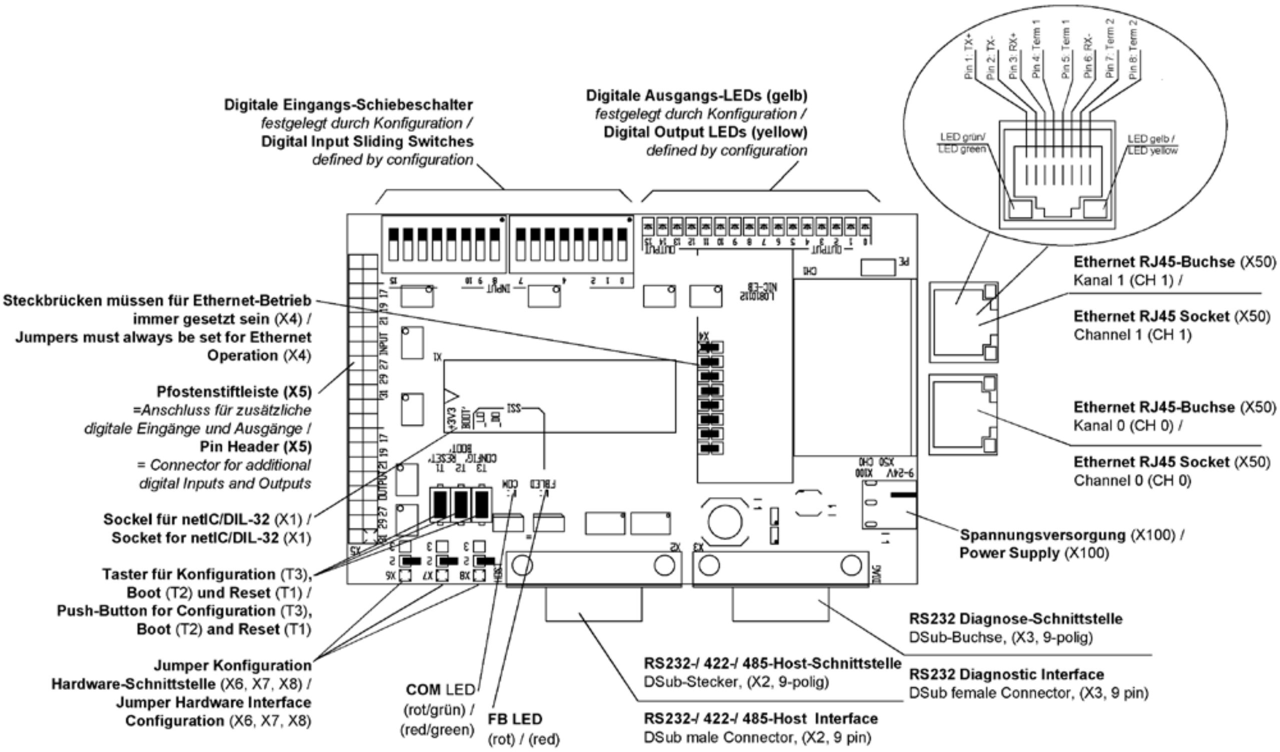


Figure 45: Device Drawing NICEB

14.1.2 Jumpers X4, X6-X8

The Evaluation-Board NICEB has 8 jumpers X4 (8 jumpers adjacently) und X6-X8 (3 single jumpers), see photo below.

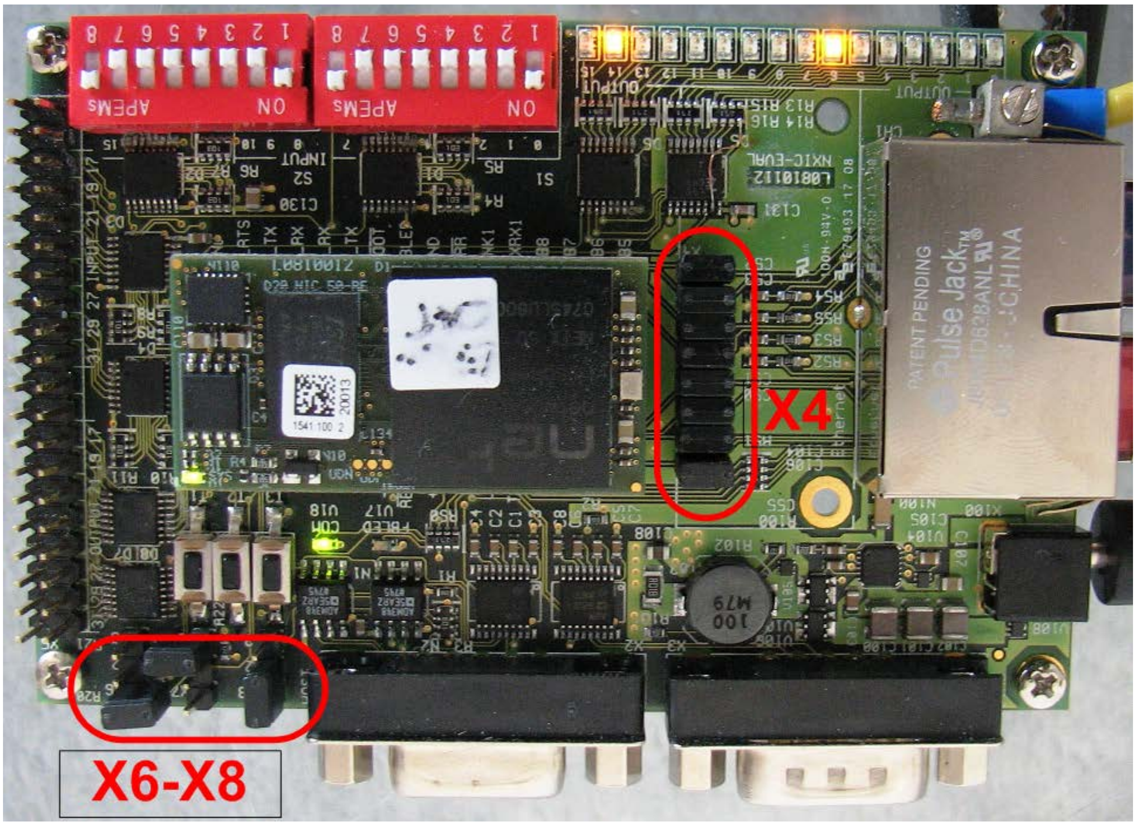


Figure 46: Photo of Evaluation-Board NICEB with Positions of Jumpers X4,X6-X8

Originally, the Evaluation-Board NICEB has been designed for testing only the NIC50-RE and for downloading firmware and configuration to it. This can be accomplished without any adapter. But also all Fieldbus versions of the Hilscher netIC Communication ICs (NIC 50-COS, NIC 50-DNS and NIC 50-DPS) can be mounted in to the DIL-32 socket of the NICEB, tested and supplied with firmware, if a suitable adapter is used and the jumpers **X4** for the Ethernet interface at the NICEB are removed all together.

NOTICE

Device Destruction by Short Circuit!

- When using the NICEB Evaluation Board with the Fieldbus-Versions of the netIC Gateways NIC 50-COS, NIC 50-DNS respectively NIC 50-DPS:
- Remove the jumpers X4 on the NICEB. Setting the X4 jumpers would cause a short circuit!
- Therefore, **never** use a netIC **Fieldbus** Communication IC within the NICEB **with the Ethernet jumpers X4 set**!

Hint:

For Ethernet operation with NIC 50-RE or NIC 52-RE together with the NICEB no adapters are needed. All jumpers X4 need to be set in order to operate correctly.

The suitable adapter for your Hilscher netIC Communication IC model can be easily determined with *Table 8: Suitable Adapters* on page 9 of this document.

The jumpers X6-X8 can clearly be recognized on *Figure 46: Photo of Evaluation-Board NICEB with Positions of Jumpers X4,X6-X8* on page 148. They are used for the configuration of the type of serial interface for Modbus communication. A precise description of the interface configuration is available at section „*Host Interface Connector and Hardware Interface Configuration*“ of this document, especially in *Table 93: Configuration of Hardware Interface to Host depending on Jumper Settings*.

14.1.3 Switches/Push Buttons

The following switches (push buttons) can be set, see photo below:

- RESET T1
- BOOT T2
- CONFIG T3 – GPIO

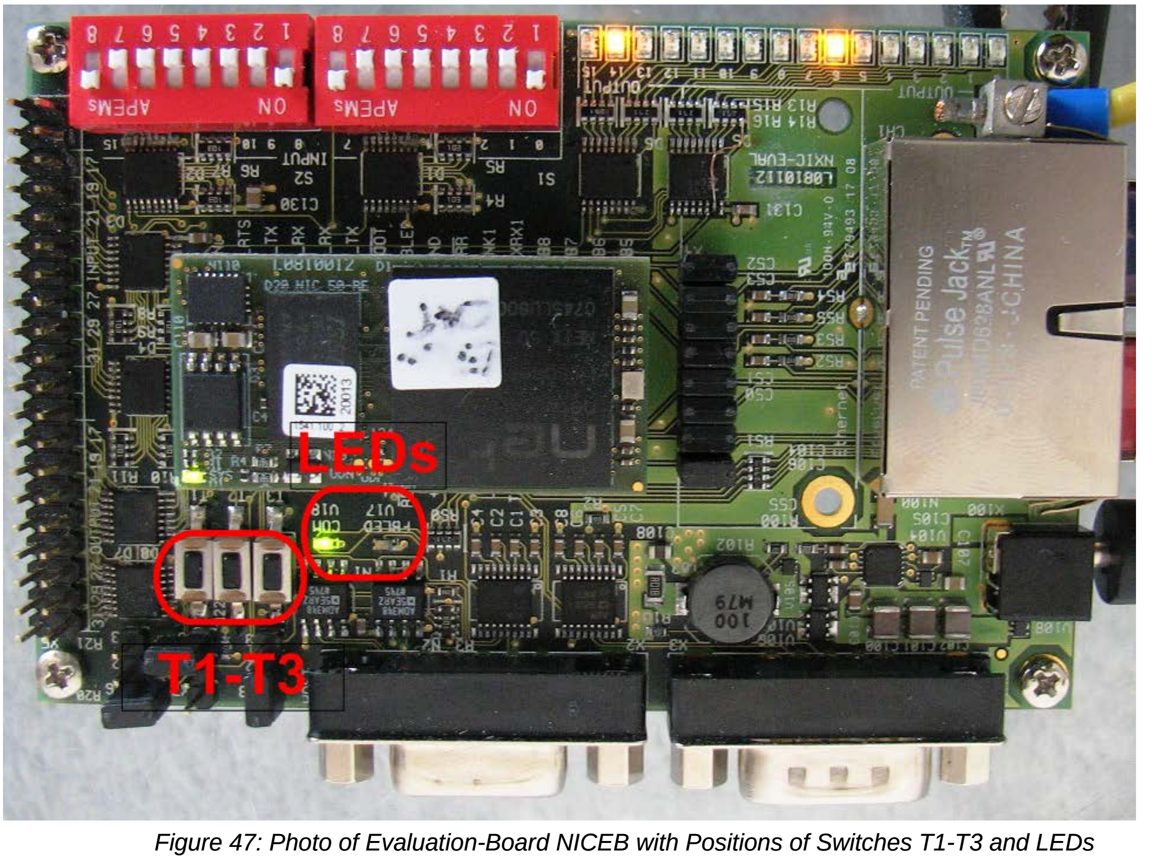


Figure 47: Photo of Evaluation-Board NICEB with Positions of Switches T1-T3 and LEDs

These push buttons are used for providing the following functions:

Push button	Position	Function
RESET T1	left	netIC reset. If this push button is pressed the netIC communication IC stops immediately and goes into reset state, i.e. a hardware reset is performed.
BOOT T2	center	Boot start. If this push button is pressed during power on the netIC communication IC goes into the boot-start mode. The netX ROM loader will be activated.
CONFIG T3 - GPIO	right	Firmware versions prior to 1.3.12.x This push button activates the configuration and diagnostic mode of the netIC Communication IC. Pressing the push button again will deactivate the configuration and diagnostic mode. Currently this push button is connected to signal GPIO/SPI_CS at pin 26. This push button is not serviced any longer by firmware versions 1.3.12.x and higher. There the configuration mode is detected and deactivated automatically.

Table 90: Push Buttons of Evaluation Board NICEB and their respective Functions

14.1.4 Status LEDs

The board has the following status LEDs controlled by the netIC Communication IC (see *Figure 47* at page 149):

LED Name	Color	Signal/ Description
COM	red/green duo-LED	This LED is controlled by the ERR (red) and STA (green) signal lines from the netIC Communication IC.
FBLED	red	This LED is controlled by the FBLED signal line from netIC Communication IC. Signals active configuration and diagnostic mode by blinking with 0.5 Hz.

Table 91: LEDs of Evaluation Board NICEB and their respective Signals

14.1.5 Connectors

The connectors available at the evaluation board NICEB are:

- Power Connector X100
- Diagnostic Interface Connector X3
- Host Interface Connector and Hardware X2
- Digital Input / Output Port X5
- Ethernet Connectors X50

These connectors described in detail subsequently.

14.1.5.1 Power Connector

The power supply of the NICEB evaluation board has to be connected to the power connector X100. The power supply voltage must be in the range between 9V and 30V DC. However, operation at 24 V is recommended. The input is protected against incorrect wiring by a diode.

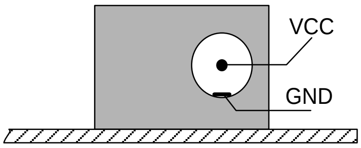


Figure 48: External Power Supply Connector

14.1.5.2 Diagnostic Interface Connector

The diagnostic interface of the netIC is connected via RS232 drivers to the 9 pin male D-Sub- connector X3. The following table shows the pinning of the connector.

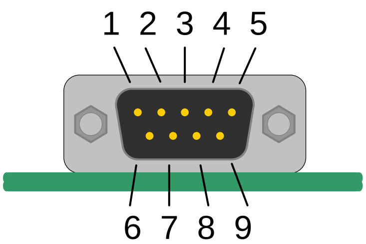


Figure 49: Diagnostic Interface Connector

Pin	Name	Description
5	GND	Ground over 100 Ohm resistor
3	TXD	Transmit Data from netIC
2	RXD	Receive Data to netIC
7	RTS	Return To Send signal from netIC

Table 92: Pinning of the Diagnostic Interface Connector

14.1.5.3 Host Interface Connector and Hardware Interface Configuration

The evaluation board contains a physical interface to the host. This interface X2 has been implemented as a 9 pin D-Sub connector, see illustration below.

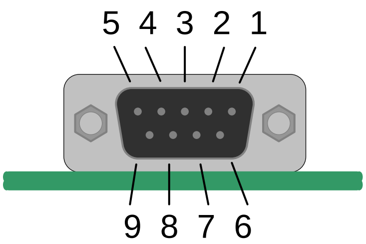


Figure 50: 9 Pin D-Sub Connector used as Host Interface Connector

It can be configured by 3 jumpers denominated as X6, X7 and X8 to match one of the following standards for serial interfaces:

- RS232
- RS422
- RS485

In case of configuration as RS422 or RS485 interface additionally there is possibility to choose between operation with an always active receiver and operation with receiver line (RS485) or transmit line (RS422) controlled by RTS signal. Depending on the chosen settings of jumpers X6 to X8, the following will occur:

- The function of the interface will behave according to the RS232, RS422 or RS485 standard.
- The function of the RS485 interface's receiver line or the RS422 interface's transmit line will be configured.
- The signals of the chosen interface type (RS232, RS422 or RS485) are available at the connector's pins according to the table below. These signals are derived from the netIC Real Time Ethernet DIL-32 Communication IC device's signal *SHIF_TXD*, *SHIF_RXD* and *SHIF_RTS*.

The subsequent table explains how to set the jumpers for the desired type of interface and which assignments of signals to pins of the connector will be established depending on that choice.

X6	X7	X8	Function	Name	Connector Pin	Description
1-2	1-2	1-2	RS485 [1]	TxD/RxD-P TxD/RxD-N GND	6 1 5	Transmit/Receive line positive Transmit/Receive line negative Ground over 100 Ω resistor
1-2	2-3	1-2	RS485 [2]	TxD/RxD-P TxD/RxD-N GND	6 1 5	Transmit/Receive line positive Transmit/Receive line negative Ground over 100 Ω resistor
open	1-2	1-2	RS422 [3]	TxD-P TxD-N RxD-P RxD-N GND	4 9 6 1 5	Transmit line positive Transmit line negative Receive line positive Receive line negative Ground over 100 Ω resistor
2-3	1-2	1-2	RS422 [4]	TxD-P TxD-N RxD-P RxD-N GND	4 9 6 1 5	Transmit line positive Transmit line negative Receive line positive Receive line negative Ground over 100 Ω resistor
open	2-3	2-3	RS232	TxD RxD RTS GND	3 2 7 5	Transmit line Receive line Return To Send line Ground over 100 Ω resistor

Table 93: Configuration of Hardware Interface to Host depending on Jumper Settings

[1] RS485 interface with always active receiver

[2] RS485 interface with RTS controlled receiver line

[3] RS422 interface with always active transmit line (no RTS control)

[4] RS422 interface with RTS controlled transmit line

The schematic illustration below shows the wiring diagram of the serial host interface of the evaluation board.

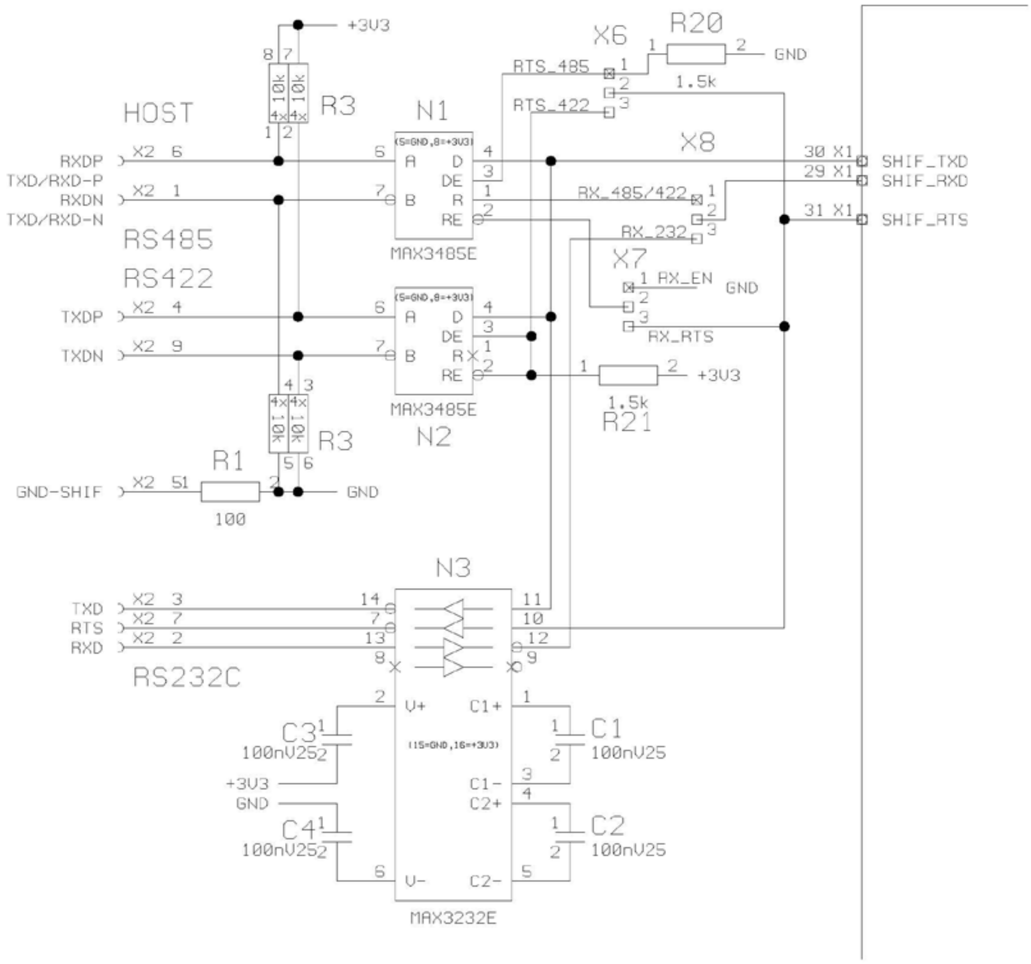


Figure 51: Wiring Diagram of the Serial Host-Interface of the Evaluation Board

14.1.5.4 Digital Input / Output Port

On the NICEB evaluation board, the synchronous serial interface of the netIC Real Time Ethernet DIL-32 Communication IC device is connected to shift registers to implement a digital input / output port. There are 16 yellow LEDs (Outputs) and 16 DIP switches (Inputs) on the evaluation board. Moreover connector X5 gives the user the possibility to connect external hardware to further 16 inputs and 16 outputs.

Note: The maximum current for each signal line has to be limited by a series resistor to a maximum of 6 mA! The maximum power that can be taken from the +3V3 line is max. 50 mA

The following table shows the pinning of connector X5.

Pin X5	Pin Name	Description
1	DO31	Digital Output
2	DO30	Digital Output
3	DO29	Digital Output
4	DO28	Digital Output
5	DO27	Digital Output
6	DO26	Digital Output
7	DO25	Digital Output
8	DO24	Digital Output
9	DO23	Digital Output
10	DO22	Digital Output
11	DO21	Digital Output
12	DO20	Digital Output
13	DO19	Digital Output
14	DO18	Digital Output
15	DO17	Digital Output
16	DO16	Digital Output
17	+3V3	Power +3.3V (< 40mA)
18	+3V3	Power +3.3V (< 40mA
19	GND	Ground
20	GND	Ground
21	DI31	Digital Input
22	DI30	Digital Input
23	DI29	Digital Input
24	DI28	Digital Input
25	DI27	Digital Input
26	DI26	Digital Input
27	DI25	Digital Input
28	DI24	Digital Input
29	DI23	Digital Input
30	DI22	Digital Input
31	DI21	Digital Input
32	DI20	Digital Input
33	DI19	Digital Input
34	DI18	Digital Input
35	DI17	Digital Input
36	DI16	Digital Input
37	+3V3	Power +3.3V (< 40mA)
38	+3V3	Power +3.3V (< 40mA
39	GND	Ground
40	GND	Ground

Table 94: Pin Assignment of Connector X5.

The next schematic shows the wiring diagram of the synchronous serial interface of the evaluation board NICEB (Revision 3):

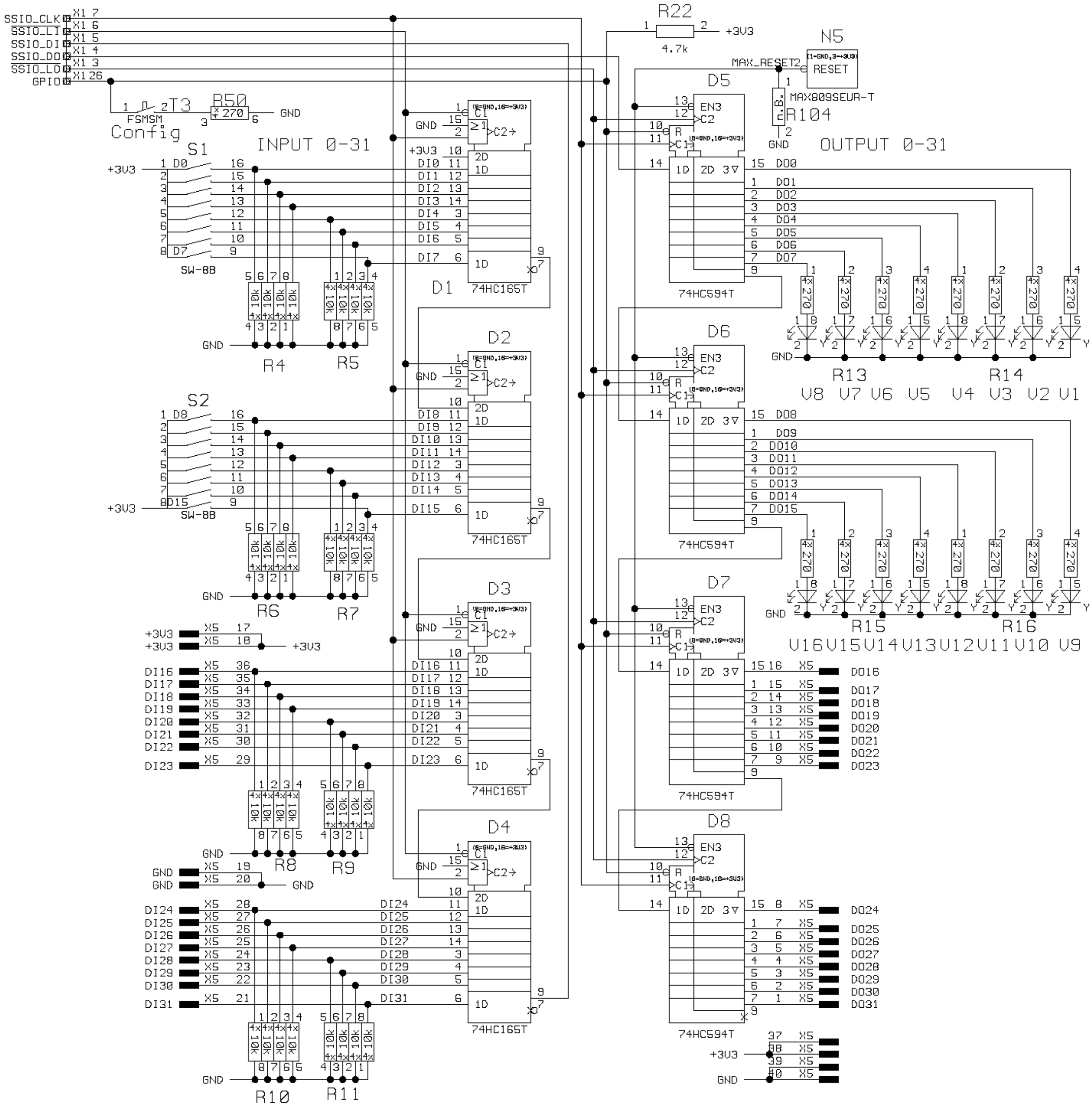


Figure 52: Wiring Diagram of the Serial I/O Shift Register-Interface of the Evaluation Board

14.1.5.5 Ethernet Connectors

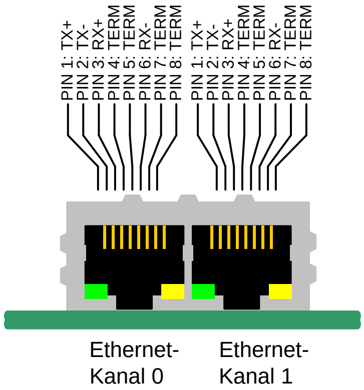


Figure 53: Schematic of Ethernet Connectors

Pin	Signal	Description
1	TX +	Transmit Data +
2	TX –	Transmit Data –
3	RX +	Receive Data +
4	TERM	Bob Smith Termination
5	TERM	
6	RX –	Receive Data –
7	TERM	Bob Smith Termination
8	TERM	

Table 95: Ethernet Interface Channel 0 and Channel 1 Pin Assignment

Auto-crossover function is supported in the netIC Communication ICs.

The following schematic picture shows the wiring diagram of the evaluation board. For Ethernet operation the eight jumpers at connector X4 always have to be set.

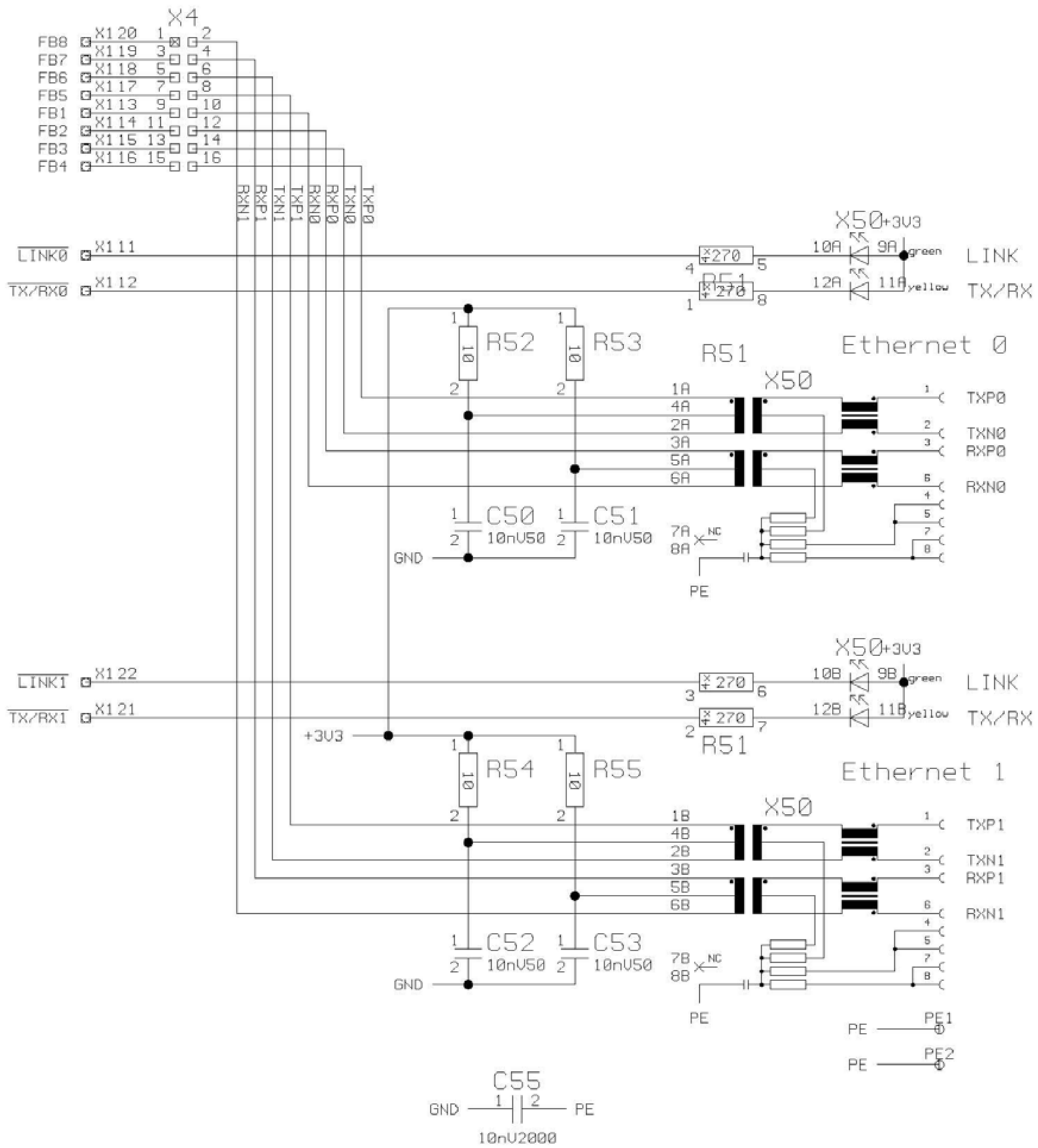


Figure 54: Wiring Diagram of the Ethernet-Interface of the Evaluation Board NICEB

14.2 Adapter NICEB-AIF for Fieldbus Connection

14.2.1 CANopen-Adapter NICEB-AIF-CO

14.2.1.1 Photo NICEB-AIF-CO

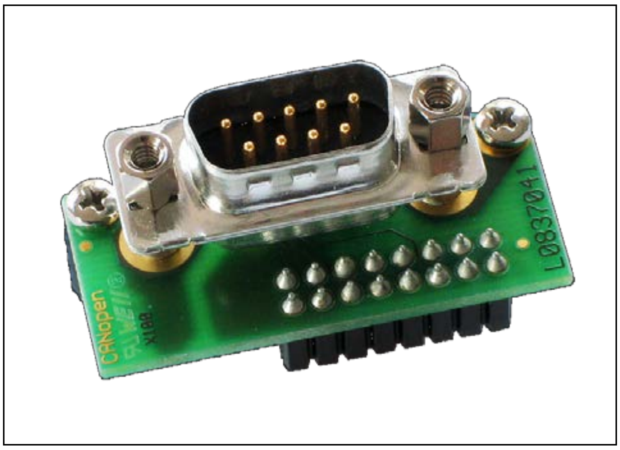


Figure 55: Photo CANopen Adapter NICEB-AIF-CO

14.2.1.2 Drawing of CANopen Interface NICEB-AIF-CO

The following drawing shows the CANopen interface (D-Sub-male connector, 9-pole) of the NICEB-AIF-CO:

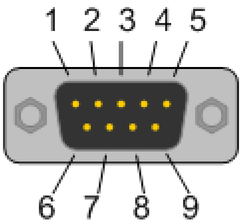


Figure 56: CANopen Interface (DSub male connector, 9 pin) of NICEB-AIF-CO

Connection with DSub male connector	Signal	Description
2	CAN_L	CAN Low Bus Line
3	CAN_GND	CAN Ground
7	CAN_H	CAN High Bus Line

Table 96: CANopen-Interface of NICEB-AIF-CO

14.2.1.3 CANopen-Interface of NICEB-AIF-CO

The CANopen-Interface of the NICEB-AIF-CO adapter has been designed as potential free interface according to ISO 11898.

Please use only CAN certified cable with the following characteristics:

Parameter	Value
Impedance	108…132 Ω
Capacity	< 50 pF/m

Table 97: Characteristics of CAN certified Cable

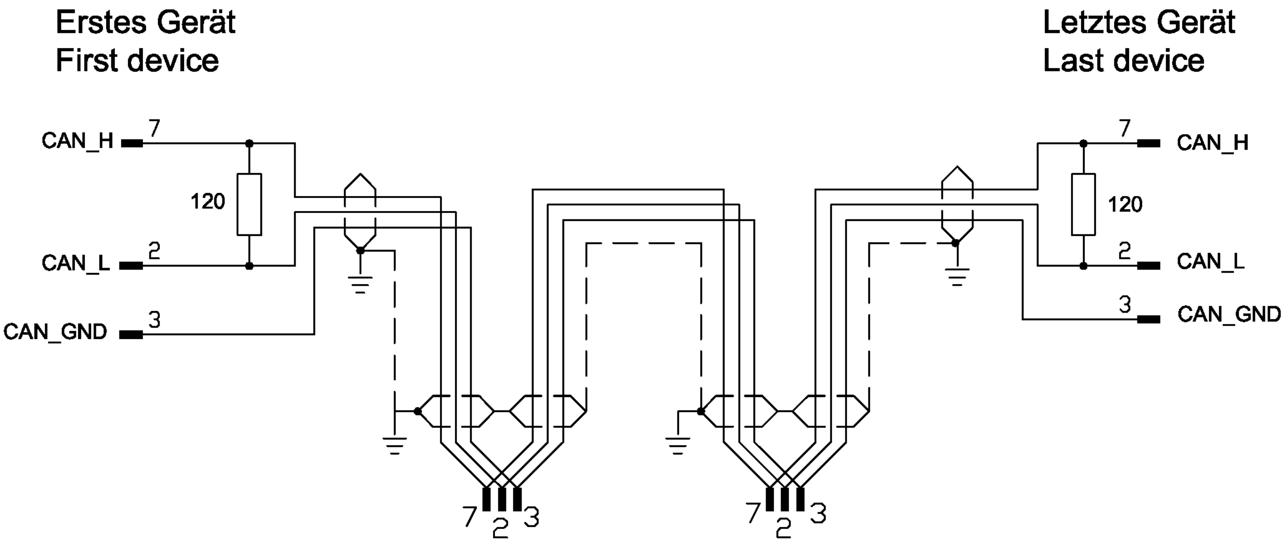


Figure 57: CAN- Network

At the ends of the network there must be two resistors of 120 Ω to terminate the cable.

It is allowed to use repeaters to increase the number of nodes, which may be connected, or to increase the maximum cable length.

Baud rate in kBit/s		Max. distance		Loop Resistance	Wire Gauge
10		1.000		26 Ω/km	0,75...0,80 mm²
20		1.000		26 Ω/km	0,75...0,80 mm²
50		1.000		26 Ω/km	0,75...0,80 mm²
125		500		40 Ω/km	0,50...0,60 mm²
250		250		40 Ω/km	0,50...0,60 mm²
500		100		60 Ω/km	0,34...0,60 mm²
800		50		60 Ω/km	0,34...0,60 mm²
1.000		40		70 Ω/km	0,25...0,34 mm²

Table 98: CAN Segment Length and Baud rate

14.2.2 DeviceNet-Adapter NICEB-AIF-DN

14.2.2.1 Photo NICEB-AIF-DN

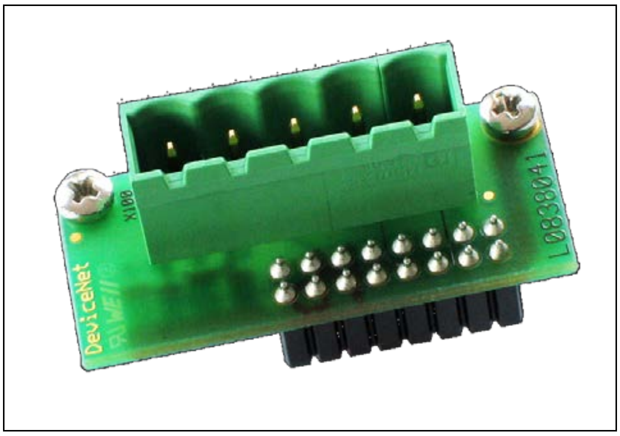


Figure 58: Photo DeviceNet-Adapter NICEB-AIF-DN

14.2.2.2 Drawing DeviceNet Interface NICEB-AIF-DN

The following drawing shows the DeviceNet Interface (CombiCon male Connector, 5 pin) of the NICEB-AIF-DN:

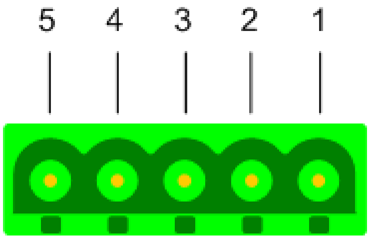


Figure 59: DeviceNet Interface (CombiCon male Connector, 5 pin) of NICEB-AIF-DN

Connection with CombiCon male connector	Signal	Color	Description
1	V-	Black	Reference potential DeviceNet power supply
2	CAN_L	Blue	CAN Low-Signal
3	Drain		Shield
4	CAN_H	White	CAN High-Signal
5	V+	Red	+24 V DeviceNet power supply

Table 99: DeviceNet-Interface of NICEB-AIF-DN

14.2.2.3 DeviceNet-Interface of NICEB-AIF-DN

The DeviceNet-Interface of the NICEB-AIF-DN has been designed as potential free ISO-11898 interface according to the DeviceNet specification.

Please ensure that termination resistors with 120 Ohm are available at both ends of the cable.

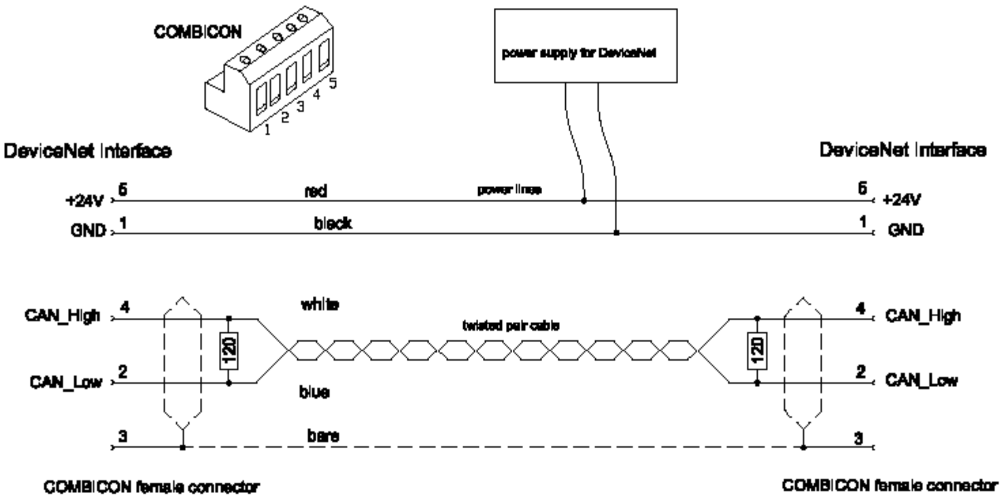


Figure 60: DeviceNet Network

Further devices can be connected via T-stubs to the bus cable. The maximum length of all T-stubs is 6 m. The whole length of the bus cable and all T-stubs does not exceed the maximum length listed in the following table. There are two different types of cables. If both cables types are used within the same network, the maximum length is:

Max. distance	**Baud rate in kBits/s**
L_{thick} + 5 x L_{thin} <= 500 m	at 125 kBaud
L_{thick} + 2,5 x L_{thin} <= 250 m	at 250 kBaud
L_{thick} + L_{thin} <= 100 m	at 500 kBaud

Table 100: DeviceNet Segment Length in dependence of the Baud rate

Up to 64 DeviceNet devices can be linked together over the bus. The maximum length of the bus cable depends on the used baud rate and the used cable type. Only special proved DeviceNet cable should be used.

The DeviceNet cable contains of the data line cables and the power supply cables.

The data line cables must match the following conditions:

Data line cable*		Impedance	Capacity	Loop Resistance	Wire Gauge (Diameter)
Thick		120 Ohm	<39,4 pF/m	<22,6 Ohm/km	2 * 1.1 mm
Thin		120 Ohm	<39,4 pF/m	<91,8 Ohm/km	2 * 0,6 mm

Table 101: Characteristics of DeviceNet Data Line Cable

The power supply cables must match the following conditions:

Power supply cable*		Loop Resistance	Wire Gauge (Diameter)
Thick		<11,8 Ohm/km	2 * 1.4 mm
Thin		<57,4 Ohm/km	2 * 0,7 mm

Table 102: Characteristics of DeviceNet Power Supply Cable

14.2.3 PROFIBUS DP-Adapter NICEB-AIF-DP

14.2.3.1 Photo NICEB-AIF-DP

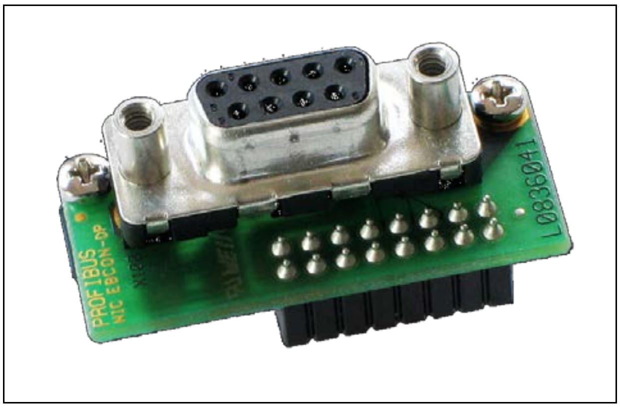


Figure 61: Photo PROFIBUS DP-Adapter NICEB-AIF-DP

14.2.3.2 Drawing PROFIBUS-DP-Interface NICEB-AIF-DP

The following drawing shows the PROFIBUS-DP-Interface (D-Sub-female connector, 9-pole) of the NICEB-AIF-DP:

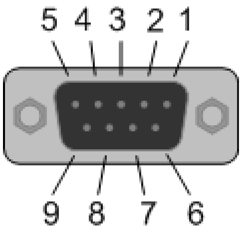


Figure 62: PROFIBUS-DP-Interface (D-Sub-female connector, 9-pole) of the NICEB-AIF-DP

Connection with D-Sub female connector	Signal	Meaning
3	RxD/TxD-P	Receive / Send Data-P respectively connection B plug
5	DGND	Reference potential
6	VP	Positive power supply
8	RxD/TxD-N	Receive / Send Data-N respectively connection A plug

Table 103: PROFIBUS-DP-Interface (D-Sub-female connector, 9-pole) of the NICEB-AIF-DP Adapter

14.2.3.3 PROFIBUS-DP-Interface of NICEB-AIF-DP

The PROFIBUS-DP-Interface of the NICEB-AIF-DP is designed as potential free RS-485 - interface.

Please ensure that termination resistors are available at both ends of the cable. If special PROFIBUS connectors are being used, these resistors are often found inside the connector and must be switched on. For baud rates above 1.5 MBaud use only special connectors, which also include additional inductance.

It is not permitted to have T-stubs on PROFIBUS high baud rates. Use only a special cable which is approved for PROFIBUS-DP. Make a solid connection from the cable shield to ground at every device and make sure that there is no potential difference between the grounds at the devices.

If the NIC50-DPS device is linked with only one other device on the bus, they must be at the ends of the bus line. The reason is that these devices must deliver the power supply for the termination resistors. Otherwise the Master can be connected at any desired position.

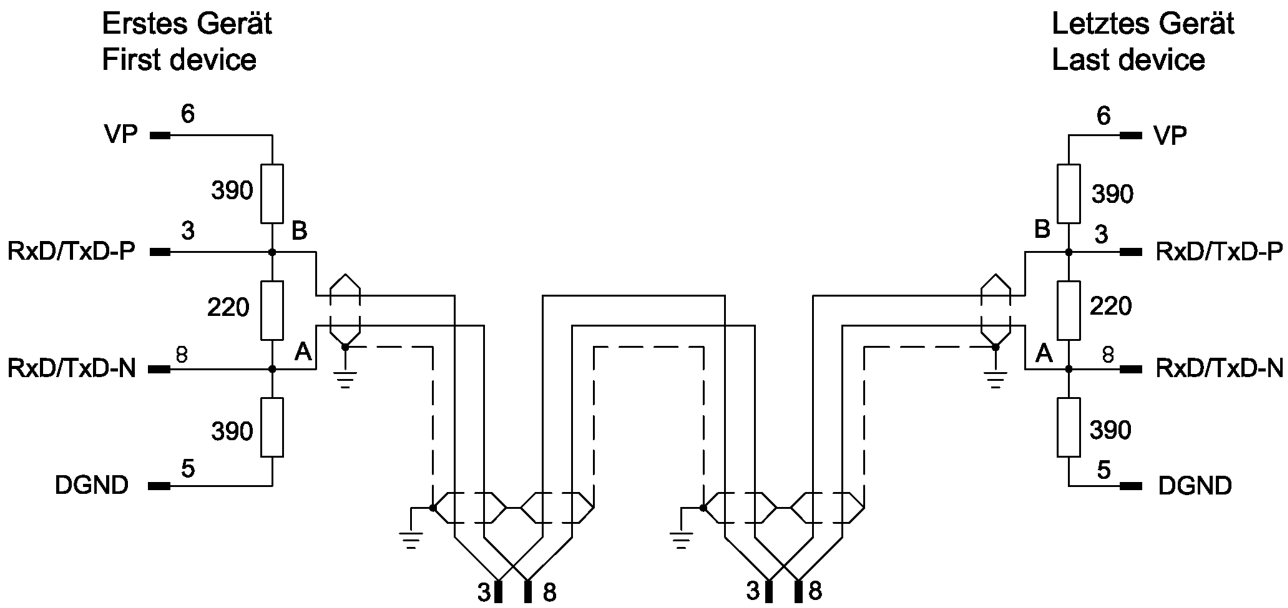


Figure 63: PROFIBUS-DP-Network

Up to 32 PROFIBUS devices can be connected to one bus segment. If several bus segments are linked to each other with repeaters, there can be up to 127 devices on the network.

The maximum length of a bus segment depends on the baud rate used, see *Table 104: PROFIBUS Segment Length in Dependence of the Baud Rate* on page 166.

Baud rate in kBit/s		**Max. distance**	
9,6		1.200 m	
19,2		1.200 m	
93,75		1.200 m	
187,5		1.000 m	
500		400 m	
1.500		200 m	
3.000		100 m	
6.000		100 m	
12.000		100 m	

Table 104: PROFIBUS Segment Length in Dependence of the Baud Rate

Only PROFIBUS certified cable, preferably the cable type A, should be used.

Parameter	**Value**
Impedance	135…165 Ω
Capacity	< 30 pF/m
Loop resistance	110 Ω/km
Wire gauge	0,64 mm

Table 105: Characteristics of PROFIBUS certified Cable

15 Communication

15.1 Sercos

This section describes, which registers the host application must read and write in order to accomplish exchange of I/O data via Sercos.

- At data transfer from the master to the slave(netIC), the Sercos Master sets control data within the MDT (Connection Control and IO Control), which must be evaluated and checked in the Sercos Slave by the Host.
- At data transfer from the slave to the master, the Host sets status data in the AT (Connection Control and IO Status), which must be evaluated and checked in the Sercos master.

In order to integrate control and status data into the data model of the netIC, we start from the following example configuration (Profile FSP IO):

Data	Data area	Register	Condition
Control data at data transfer from master to the slave			
Connection Control	Input data area	Register 1000	If **(1)** = 0
IO Control	Input data area	Register 1001	If **(2)** = 2
Data to be used	Input data area	From Register 1002	
Status data at data transfer from slave to the master			
Connection Control	Output data area	Register 2000	If **(3)** = 0
IO Status	Output data area	Register 2001	If **(4)** = 2
Data to be used	Output data area	From Register 2002	

Table 106: Example Configuration for Profile FSP IO, Connection Control prior to I/O Data

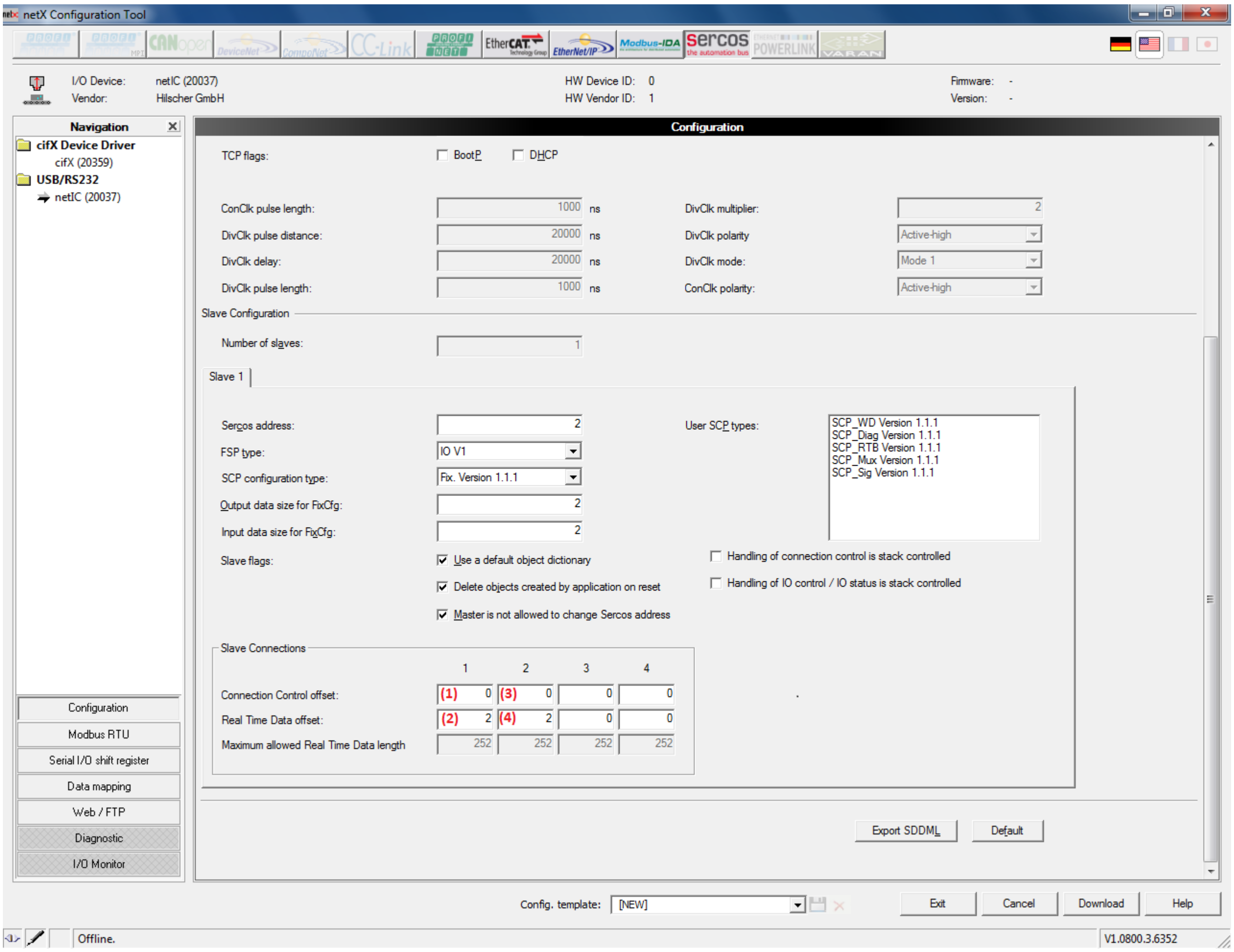


Figure 64: Screen "Configuration" of the netX Configuration Tool (only lower part shown)

In order to accomplish this configuration, adjust the following settings within the netX Configuration Tool:

- In pane "Data Mapping" set the offset addresses for I/O data for SSIO to the value **196** in order to achieve the register assignment according to Table 113.
- In pane "*Slave Connections*" set each value at *Connection Control Offset* to **0**.

- ➢ In pane “*Slave Connections”* set each value at *Real Time Data Process Image Offset* to **2**, see *Figure 64: Screen “Configuration“ of the netX Configuration Tool (only lower part shown)*
- ➾ The following register layout results within the netIC data model:

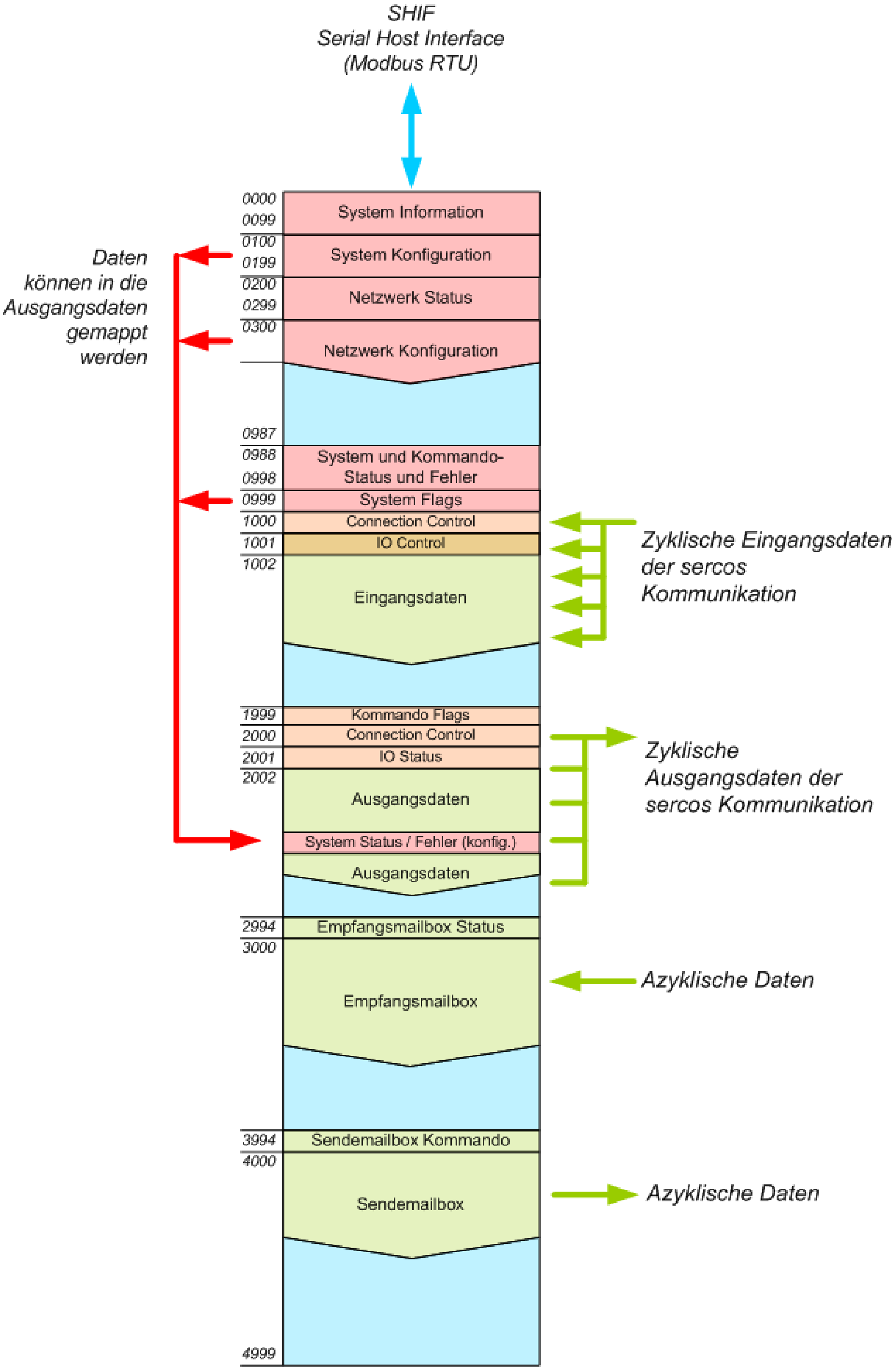


Figure 65: Data Model for Sercos (Example Configuration)

The following describes the control and status data of Sercos in a more detailed way.

15.1.1 Connection Control

This section concerns both Master-to-Slave communication (via MDT) and Slave-to-Master communication (via AT).

In Sercos, Connection Control (C-Con) is a two byte data area containing connection-related information. It is present within both the MDT and the AT. It can be accessed via MDT and AT. In the object dictionary, connection control is located at IDN S-0-1050.x.8.

Connection Control can be mirrored before (Register 1000 and register 2000) or after the area for operation data (input or output) in the register area of the netIC.

You can comfortably adjust these parameters using the netX Configuration Tool. Here the parameters are called *Connection Control Offset* (①,③), *Real Time Data Process Image Offset* (②,④) and *Real Time Data Maximum Length* (⑤,⑥). (The position numbers relate to *Figure 64* on page 168.)

Connection Control is managed by the firmware. The netIC supports bus synchronous operation. (Therefore bit 3 of Connection Control has the value 1.) However, the host only has asynchronous access onto the Sercos data.

For correct operation it is crucial to be aware about the function of the Producer Ready Bit (Bit 0 of C-Con):

The Producer Ready Bit specifies, whether currently data are produced in the connection to which the C-Con belongs and declares these data as valid. On every reception of a new MDT (in CP4), the Producer Ready Bit has to be set to 1. The New Data Bit is set to 0 on the transition of the Sercos slave's state machine to CP4. It must be toggled every time there is a change in the data.

15.1.2 IO Control

This section concerns only Master-to-Slave communication (via MDT).

In the FSP IO profile, which is used at netIC, IO Control is a two byte data area containing IO-related information. It is only present within the MDT. In the object dictionary, IO Control is located at IDN S-0-1500.x.01.

It is mirrored in register 1001 within the register area of the netIC, see *Figure 65: Data Model for Sercos (Example Configuration)*.

The host application must evaluate the IO Control.

Bit 15 contains the *Operation State Outputs Bit*. Here, 1 means „active" and 0 is equivalent to „inactive". All other bits of the IO Control are set to 0, so the IO Control is 0x8000 when the operation state of the outputs is active (i.e. ready for output) and 0x0000 if inactive.

With the Operation State Outputs Bit, the Sercos Master controls whether the output data at the Sercos Slave are transmitted to the Sercos Master (0x8000) or not (0x0000).

15.1.3 IO Status

This section concerns only Slave-to-Master communication (via AT).

In the FSP IO profile, IO Status is a two byte data area containing IO-related information. It is present only within the AT so it can be used to transmit status information to the Sercos master. In the object dictionary, IO Status is located at **IDN S-0-1500.x.02**.

It is mirrored in register 2001 within the register area of the netIC, see *Figure 65: Data Model for Sercos (Example Configuration)*.

The host application must manage the IO Status.

The following two bits are important:

The „*Outputs ready to operate bit*" (Bit 15) has the value 1, if the outputs are active, i.e. when the Operation State Outputs Bit in the IO Control of the previously received MDT has been set to 1 and the host application has successfully activated the outputs of the device. Setting bit 15 to 0 means the outputs are not ready and are set to substitute values. (The host uses this way for feedback to the Sercos Master).

Set the „*Inputs valid bit*" (Bit 14) to 1, if the device produces valid input data and to 0 if this is not the case.

By setting the IO Status to the hexadecimal value 0xC000, the host application indicates via the Sercos Slave to the Sercos Master, that the device is in full operation (Outputs active and valid input data) and output data have really been transmitted.

15.1.4 Reception of Real-Time Data

Note: Practically, the bus cycle time will have a significantly lower value (for instance, 1 ms) than the access time of the host onto the data within the netIC (for instance, 100 ms). This means:

- The host application is not aware of each change in Sercos data.
- The host application always accesses the Sercos data asynchronously.

The host application must perform the following steps in order to correctly evaluate the real-time data of the received MDT which are designated for the slave:

➢ The *Producer Ready Bit* (Bit 0) within Connection Control (Register 1000) must be checked whether it is equal to 1. Only if this is the case, further evaluation of the data may be done.

➢ The *Operation State Outputs* Bit (Bit 15) within the IO Control Word must be set to 1. Thus, check whether the IO Control Word (Register 1001) has the value 0x8000.

➢ Each time the *New Data Bit* (Bit 1) in the Connection Control toggles, i.e. changes its value, new input data are present for evaluation. At this time the Sercos slave must read out its input (Input data area beginning at register 1002) and can then evaluate its input data.

Note: If the host access time is exactly a multiple of the bus cycle time, it can happen, that the toggling of the New Data Bit within the Connection Control is not recognized.

15.1.5 Sending of Real-Time Data

The host application must proceed as follows to send the AT correctly:

- The host has to supply the data to be sent by the netIC within the AT,
- Write these output data into the output data area of the netIC (in the example: Write into the output data area beginning at register 2002).
- Set the mandatory bits within the IO Status (Register 2001). This means:
  1. If IO Control (Register 1001) had the value `0x8000` on the last MDT reception, and the outputs have been successfully activated, set the „*Outputs ready to operate bit*“ (Bit 15) within the IO Status to the value 1.
  2. If the device delivers valid inputs, set the „*Inputs valid bit*“ (Bit 14) of IO Status (Register 2001) to the value 1.
- Set the mandatory bits within the Connection Control (Register 2000). This means:
  1. Set the *New Data Bit* (Bit 1) within Connection Control to the value 1, if it is equal to 0 hat, and vice versa.

Note: The New Data Bit (Bit 1) within Connection Control (Register 2000) is automatically toggled between the values 0 and 1 by the protocol stack so you do not need to care about this.

  2. Set the *Producer ready bit* (Bit 0) within Connection Control (Register 2000) to the value 1 in order to indicate readiness to send data.

15.1.6 Example for Configuration and Application

This configuration and application example explains how to practically set up an operational Sercos data transmission conforming to the rules specified within the two preceding subsections of this document.

The following bits need to be taken into account:

Control-/ status word	Name of bits	Register Number	Bit-Number	Value	Meaning
Communication from Master to Slave (MDT)					
Connection Control	*Producer ready bit*	1000	Bit 0	 0 **1**	This bit should be set to the value 0 if the Master has declared its data as invalid. The producer does not yet generate any data in this connection. The producer generates data in this connection. If the producer did toggle the *New Data Bit*, the consumer may evaluate and process it.
Connection Control	*New Data Bit*	1000	Bit 1	**0<->1**	New Producer Data Initial value (in CP4) is 0. Each toggling (switching from 0 to 1 or vice versa) announces new data in the connection. Then the data are exchanged between the connection and the application. The value of bit 1 should always be identical to the value of bit 12 of Connection Control.
IO Control	*Operation State Outputs Bit*	1001	Bit 15	0 **1**	Outputs inactive (Substitute values have been activated) Outputs activated
Communication from Slave to Master (AT)					
Connection Control	*Producer ready bit*	2000	Bit 0	 0 **1**	This bit should be set to the value 0 if the Slave has declared its data as invalid. The producer does not yet generate any data in this connection. The producer generates data in this connection. If producer did toggle the *New Data Bit*, the consumer may evaluate and process it.
Connection Control	*New Data Bit*	2000	Bit 1	**0<->1**	New Producer Data Initial value (in CP4) is 0. Each toggling (i.e. switching from 0 to 1 or vice versa) by the firmware announces new data in the connection. Then the data are exchanged between the connection and the application. The value of bit 1 should always be identical to the value of bit 12 of Connection Control
IO Status	*Inputs valid bit*	2001	Bit 14	0 **1**	Inputs invalid, for instance local bus communication error. Inputs valid
IO Status	*Outputs ready to operate bit*	2001	Bit 15	0 **1**	Outputs set to substitute values or freeze. Outputs successfully activated and bit 15 in IO Control has been set.

Table 107: Relevant Bits of Control- and Status Word in the Configuration Example

The values for the bits used within the example are **marked** in column "*Value" of Table 107: Relevant Bits of Control- and Status Word in the Configuration Example* "

All other bits of control and status words not mentioned within Table 107 are treated as not set (0) within the scope of this example.

This leads to the following values:

For Master-to-Slave communication (MDT)

Connection Control = `0x0001` or `0x0003`

IO Control = `0x8000`

For Slave-to-Master communication (AT)

Connection Control = `0x0001` or `0x0003`

IO Status = `0xC000`

Within the example a fixed Sercos configuration is assumed (according to SCP_FixCFG).

The part of the associated data model displayed in *Figure 66* shows the location of Connection Control, IO Control, IO Status and input/output data including the proposed values for Connection Control, IO Control und IO Status:

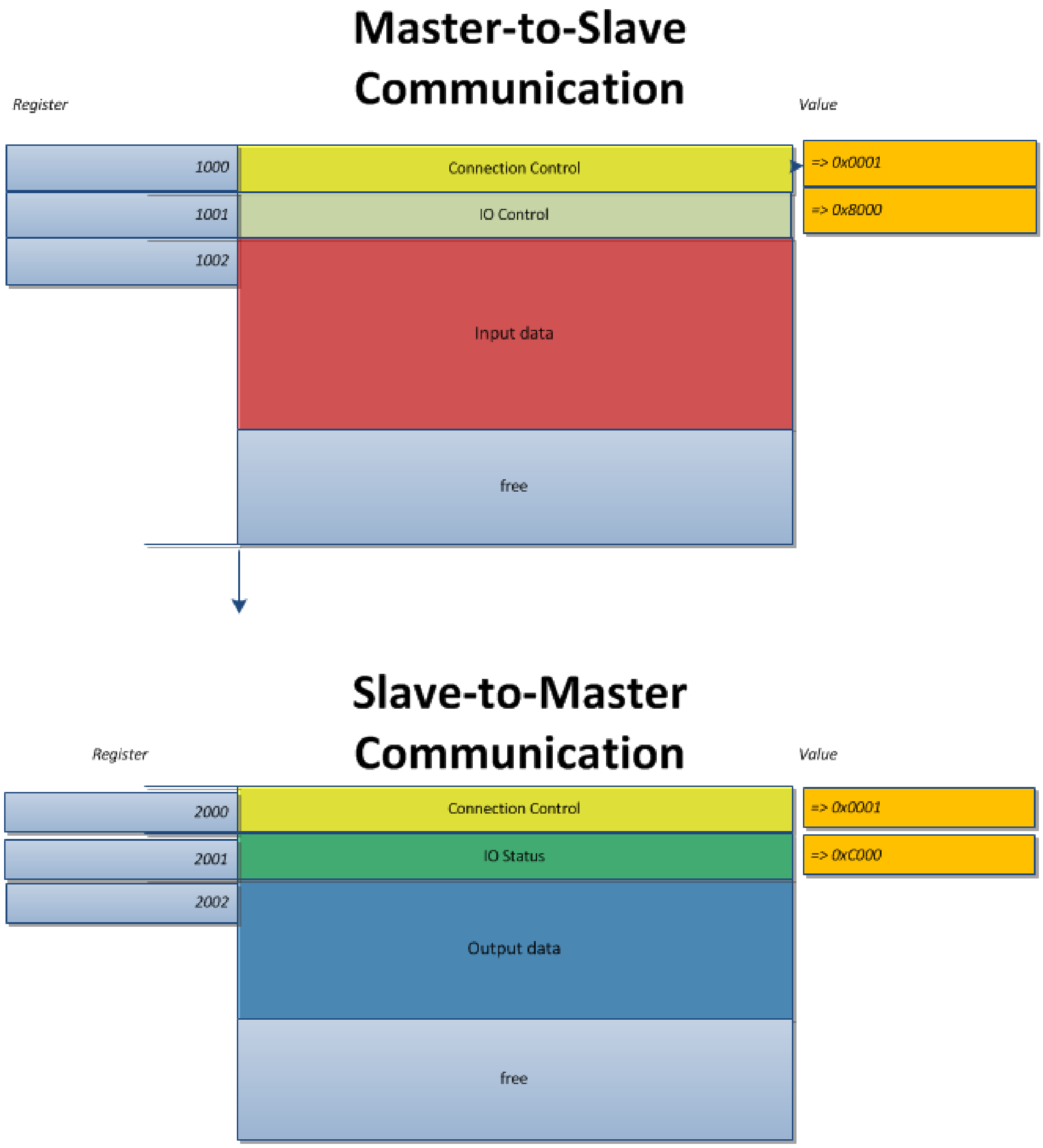


Figure 66: Data Model of the Configuration Example

As a simple tool for input of control data and controlling the test system, the IO-Monitor of the system configurator SYCON.net is used here. If also a

Sercos Master from Hilscher is used, the I/O-Monitor of the SYCON.net can also be used at the Sercos Master side.

Note: Alternatively, for this purpose the netX Configuration Tool or the cifX Test Program delivered along with the cifX Device Driver can be used.

The IO Monitor allows specifying values for Connection Control and IO Control and supervision of Connection Control and IO Status. This is done as follows:

The IO Monitor is opened in the DTM of the Sercos Master connected with the netIC below Tools/IO Monitor (in the navigation area). The perspective of the IO Monitor is that of the Sercos Master.

The Master–to-Slave communication is displayed in the area *Output Data*.

Register 1000 contains the Connection Control, register 1001 the IO Control. In this context, little endian representation is applied, therefore the bytes receive the values 0x01 (or in toggled state 0x03), 0x00, 0x00 and 0x80. Then the data to be transmitted follow (here in this order 0xAA, 0xBB, 0xCC and 0xDD).

The Slave–to-Master communication is displayed in the area *Output Data*.

Register 2000 contains the Connection Control, register 2001 the IO Status. Due to the applied little endian representation the bytes receive the values 0x01 (or in toggled state 0x03), 0x00, 0x00 and 0xC0 in this order. Then the data to be transmitted follow (here in this order 0xDD, 0xCC, 0xBB and 0xAA).

In the IO Monitor the test data of this example look like this:

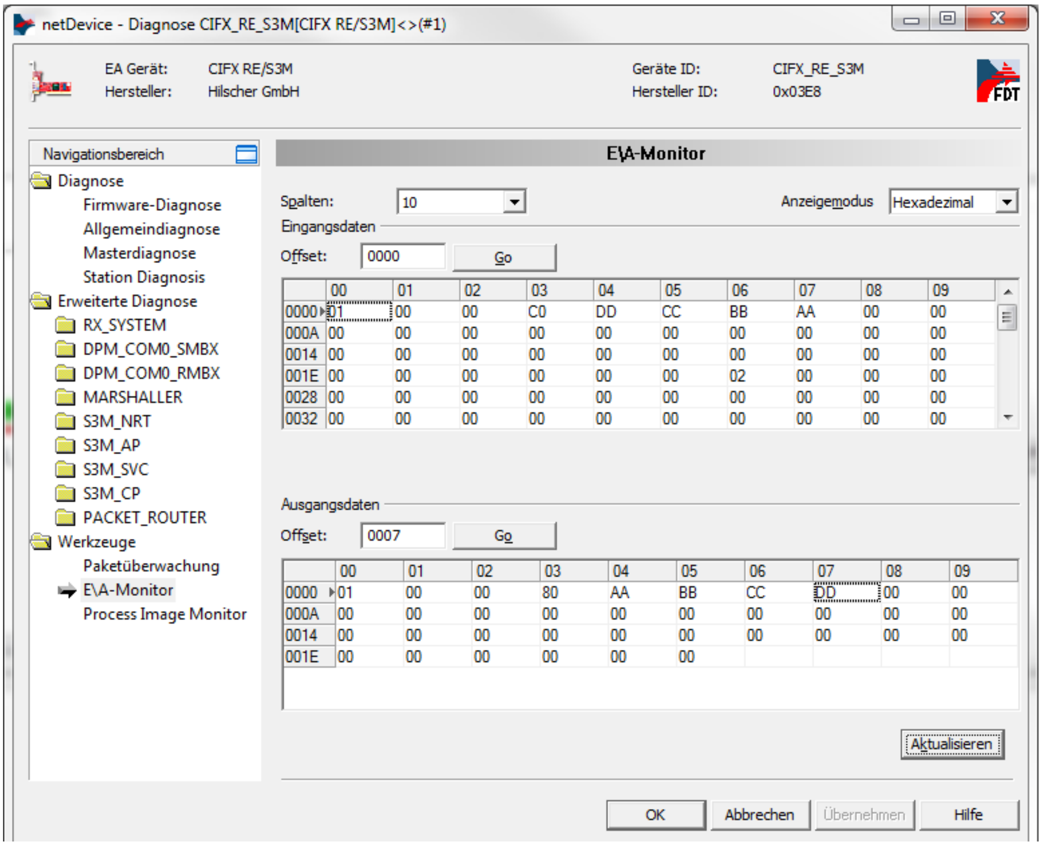


Figure 67:View of the Data of the Configuration Example in the IO Monitor of SYCON.net.

15.2 CC-Link IE Field Basic Slave

This section describes, which registers the host application must handle in which way in order to enable an exchange of I/O data via CC-Link IE Field Basic.

- When data is received by slave (netIC) from network: RWw, RY
- When data is transmitted by slave (netIC) to network: RWr, RX

The following example configuration shows how status data and user data are arranged in the data model of the netIC:

Variable	Data area	Register	Description
Status data and user data: data transfer from network to slave (RWw, RY)			
Cyclic state	Input data area	Register 1000 low byte	Bit 0: If set to 1, the entire data is actively transmitted
Station state	Input data area	Register 1000 high byte	Bit 0: If set to 1, the actual station is containing valid data
Remote station info	Input data area	Register 1001	Received value from master
Cyclic end code	Input data area	Register 1002	Unused
Reserved	Input data area	Register 1003	Reserved, always equals 0
Word data (user data)	Input data area	Register 1004 … 1035	RWw data (32 words per occupied station)
Bit data (user data)	Input data area	Register 1036 … 1039	RY data (8 byte per occupied station)
For every further occupied station, this fixed structure is repeated once (Registers 1040…1079 for occupied station 2, 1080 … 1119 for occupied station 3, and so on). At maximum, there are 16 occupied stations.			
Status data and user data: data transfer from slave to network (RWr, RX)			
Cyclic state	Output data area	Register 2000 low byte	Range of values: 0 … 1 Bit 0: If set to 1, the actual station is containing valid data
Cyclic stop	Output data area	Register 2000 high byte	Unused, always equals 0
Local station info	Output data area	Register 2001	Only settable in first array element, valid for all other occupied stations then.
Reserved	Output data area	Register 2002 … 2003	Reserved, always equals 0
Word data (user data)	Output data area	Register 2004 … 2035	RWr data (32 words per occupied station)
Bit data (user data)	Output data area	Register 2036 … 2039	RX data (8 bytes per occupied station)
For every further occupied station, this fixed structure is repeated once (Registers 2040…2079 for occupied station 2, 2080 … 2119 for occupied station 3, and so on). At maximum, there are 16 occupied stations.			

Table 108: Fixed structure of status data and user data at CC-Link IE Field Basic Slave

16 Serial Peripheral Interface (SPI) for netIC

16.1 Principle of Operation

The Serial Peripheral Interface (SPI) is a bus system for the synchronous serial communication of digital electronic circuits allowing versatile applications. It is based on the master-slave-principle. There are many possible applications for SPI.

The Serial Peripheral Interface was originally developed by Motorola, nevertheless, a really exact SPI specification is not available. The system *Microwire* from National Instruments is very similar to SPI.

For the **S**erial **P**eripheral **I**nterface, the following basic definitions apply

- There are two kinds of bus participants: Masters and Slaves
- In an SPI bus system, an arbitrary number of slaves is allowed.
- In an SPI bus system, there is only one single master. This master never communicates with more than one slave at a time. The master generates the time signal **SCK** (see below) and selects which bus participant to communicate with by means of the Chip-Select signal (#CS), see below.
- Each slave requires a clock signal and the Chip-Select signal. Unless the slave is selected for communication by the master, its data output line is in a state of high impedance in order to decouple it from the data bus. This avoids disturbances when sending data from multiple slaves to the master at the same time.
- Various clock frequencies up to the MHz area are supported (Upper limit at the netIC: 1 MHz).
- Three common lines connect all bus participants.
  3. MOSI (*Master Out Slave In*) respectively SDI (*Serial Data In*) (corresponds to netIC Pin 29, **SPI_MOSI**)
  4. MISO (*Master In Slave Out*) respectively SDO (*Serial Data Out*) (corresponds to netIC Pin 30, **SPI_MISO** and can be used in order to read back the data or for cascading.)
  5. SCK (*Serial Clock*) – a clock signal for synchronization of data communication (corresponds to netIC Pin 31, **SPI_CLK**)
- The signals MOSI and MISO can also be multiplexed or one of these may be absent at all.
- For every bus participant there is a Chip-Select signal (#CS) which is also sometimes denominated as #SS (*Slave Select*) or #STE (*Slave Transmit Enable*). This line is connected to netIC Pin 26, **SPI_CS**. It is logically 0-active(*active low*). If it is connected to ground (low level), the following happens:
  - The slave becomes active.
  - Supervision of the MOSI signal starts.
  - Data are sent to MISO according to the clock.
  - In each direction (Master->Slave/MOSI and Slave->Master/MISO) exactly 1 bit is transferred per clock cycle of the **SCK** signal.
- Due to the separate input and output data lines, SPI is capable of full-duplex data transmission.
- The internal logic of an SPI circuit usually contains at least one shift register for the transformation of serial data to parallel data in order to prepare the data for further processing. Such a shift register is

contained within the netIC, for instance. However, the implementation of the internal logic may also be a lot more complex. The length of this shift register is not defined. Often values of 8 bits or multiples of 8 bits are chosen.

16.1.1 SPI Modes

SPI allows you to identify 4 separate modes of operation (denominated as **SPI Mode 0** up to **SPI Mode 3**) as due to the lack of specification there is no precise definition whether the data are taken over at leading edge or falling edge.

These modes differ in the polarity and phase parameters CPOL and CPHA which are supported by all Motorola microcontrollers and many other SPI circuits.

The parameters have the following meanings:

- The polarity parameter CPOL (Clock Polarity) defines whether the edge is leading or falling:
- 0 (= Clock Idle Low): The clock signal is not inverted: The clock is normally LOW, a change to HIGH will be interpreted as leading edge.
- 1 (= Clock Idle High): The clock signal is inverted: The clock is normally HIGH, a change to LOW will be interpreted as leading edge.
- The phase parameter CPHA (Clock Phase) defines at which edge (i.e. leading or falling edge) data are read or put out.
- 0: Data are read at leading edge and written at falling edge.
- 1: Data are read at falling edge and written at leading edge.

The table below illustrates the relation between the SPI modes on the one hand and the parameters CPOL and CPHA on the other.

Mode	CPOL	CPHA	Edge for data take-over	Supported by netIC
0	0	0	First edge (High)	Not supported
1	0	1	Second edge (Low)	Not supported
2	1	0	First edge (Low)	Not supported
3	1	1	Second edge (High)	Supported

Table 109: Relation between SPI Modes and Parameters CPOL/CPHA

The netIC only supports the **SPI Mode 3** e. g. Polarity CPOL = 1 and Phase CPHA 1. This setting is fixed and cannot be changed therefore. The data are taken over at the second edge (High).

16.2 The netIC as SPI Device

16.2.1 Mode of Operation/Chip Select

The netIC can only be operated as a slave device within SPI systems. For SPI operation, the SPI mode must explicitly be activated in the netIC as described subsequently (Section 16.2.2 "*Activating the SPI-Mode*").

Please take the following into account:

Important: If the SPI Mode is activated, the support of the integrated Serial I/O Shift Register Interface (Pins 3 to 7 of the netIC) is **limited to** update frequencies of up to approximately **500 Hz**. For many fast digital I/O applications this update frequency is **not sufficient**.

Consequently, simultaneous operation of SPI and fast digital I/O applications is not possible at the netIC. However, after **deactivation** of the SPI Mode, the integrated Serial Shift IO Interface will **immediately** be available with **full performance**.

Important:

It is **not required** to hold down the #CS signal to LOW all the time during the entire request/poll/response cycle.

The Chip Select signal #CS can be released and selected later again in order to poll for the response. This allows the host to provide services to other SPI circuits while the netIC processes the request.

16.2.2 Activating the SPI-Mode

You can activate the SPI Mode via the netX Configuration Tool;

netX Configuration Tool

- In netX Configuration Tool, switching to SPI Mode is done by selecting the option **SPI Mode 3** instead of RS232, RS422 or RS485 within the combo box of the parameter **Interface Type** in the Modbus RTU Configuration Page of the netX Configuration Tool. This combo box is displayed in opened state in *Figure 68*.

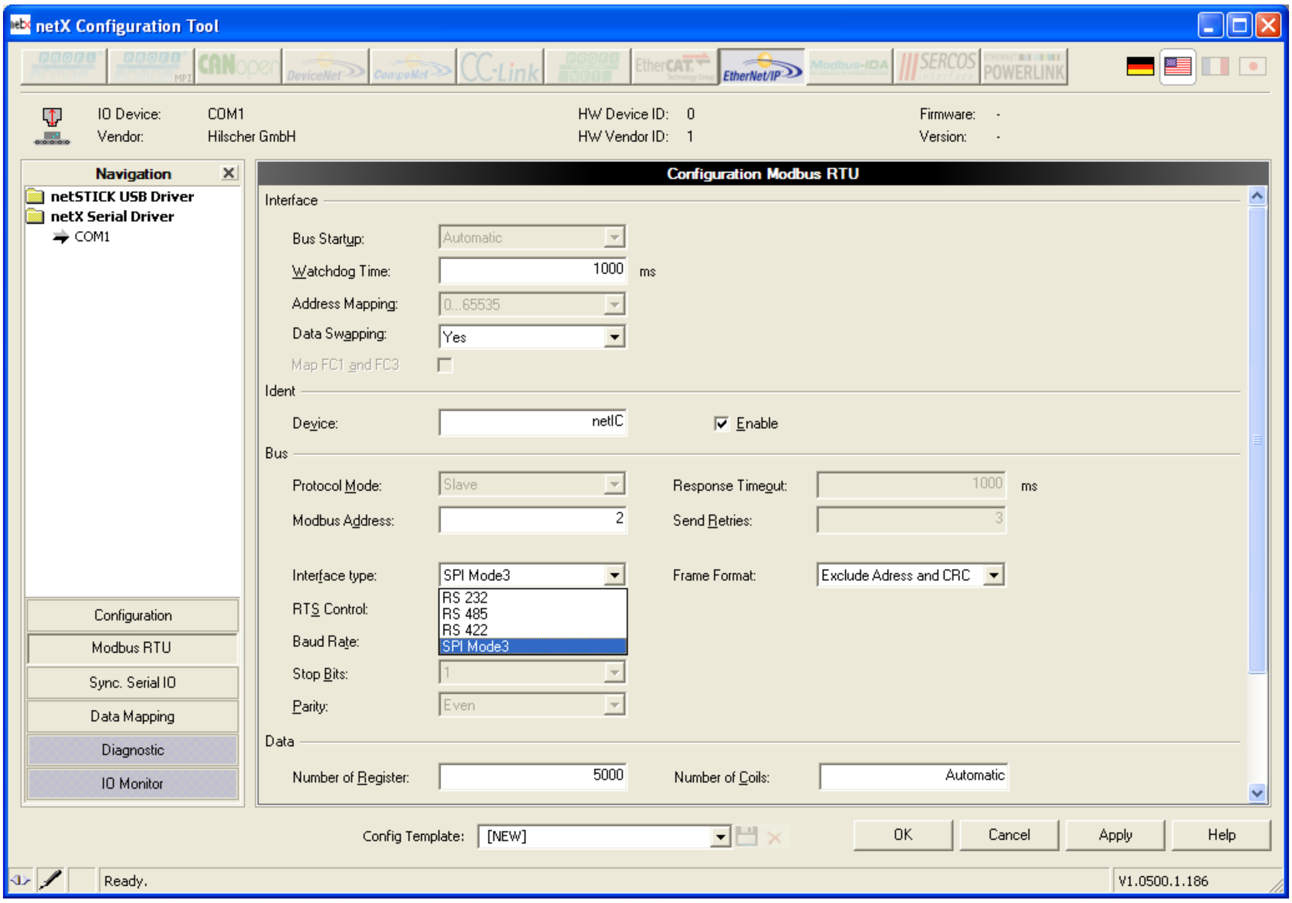


Figure 68: Modbus RTU Configuration Page in netX Configuration Tool - Parameter "Interface Type"

After selection of the SPI Mode the combo box *„Frame Format“* right next is enabled, see *Figure 69*. There you can adjust, whether the CRC checksum and the frame address are included in data transmission, or not. The default is not to include CRC checksum and address. This option offers improved performance compared to the other one.

➩ After clicking at the **Ok** button, the SPI Mode will be activated.

For more information, refer to the **Operating Instruction Manual netX Configuration Tool for netIC**.

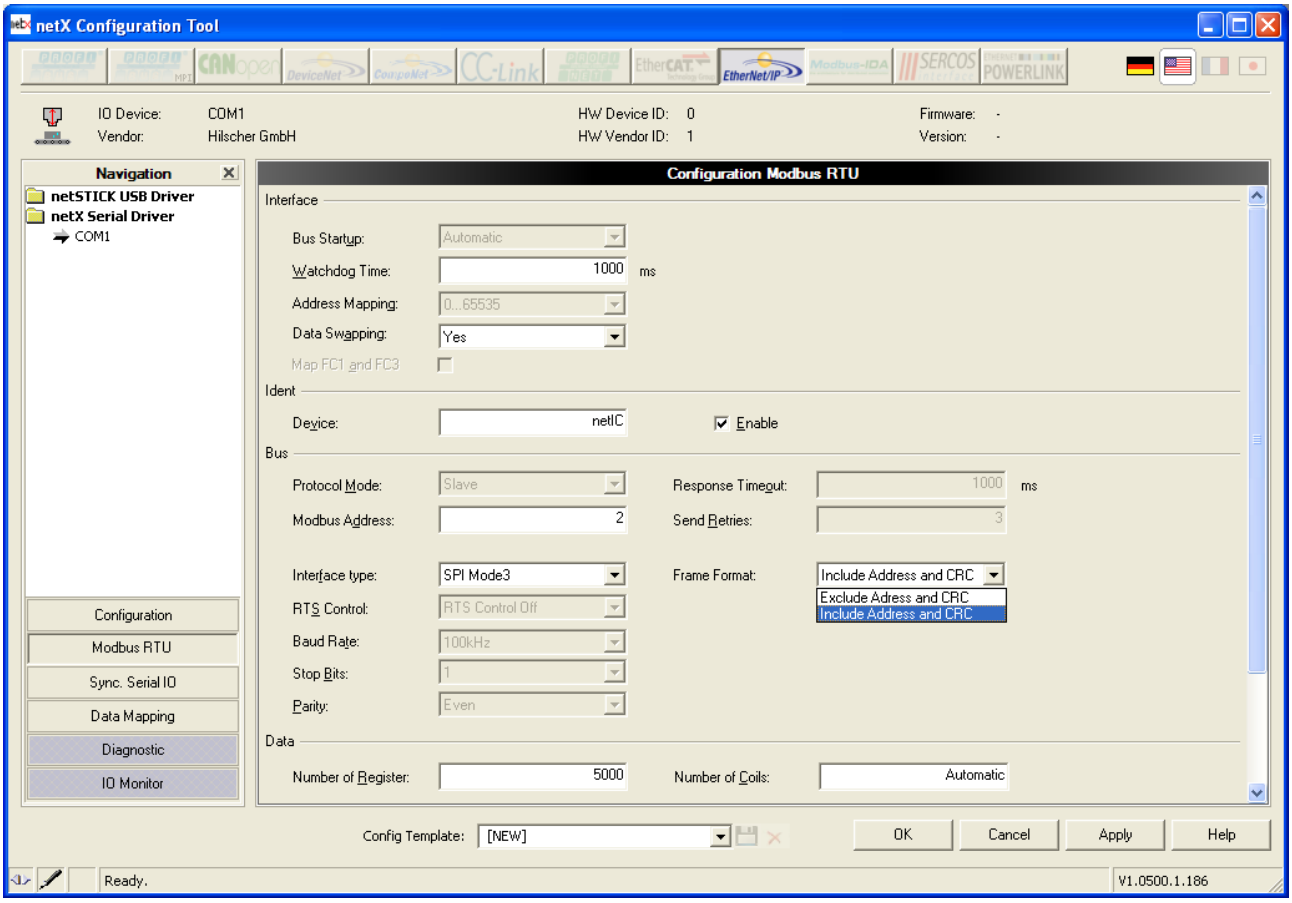


Figure 69: Modbus RTU Configuration Page in netX Configuration Tool - Parameter "Frame Format"

16.2.3 Deactivating the SPI-Mode

You can deactivate the SPI Mode either via Modbus RTU or via the netX Configuration Tool.

netX Configuration Tool

- ➢ Deactivating the SPI-Mode can be accomplished via the netX Configuration Tool by selecting one of the options RS232, RS422 or RS485 in the combo box *"Interface Type"*.
- ➩ Then the SPI Mode will be deactivated and the serial interface will continue to operate in the usual manner. Also there will no longer be a performance deterioration of the Serial Shift IO Interface .

16.3 MODBUS-Protocol via SPI

The netIC device offers a mode that allows the host to communicate with the netIC device via SPI. In this mode the Modbus RTU protocol is used within the SPI communication level.

The Modbus RTU is usually transmitted via a serial interface. The protocol is based on the request/confirmation concept: The Modbus RTU master sends a request to the Modbus RTU slave which will answer within a given time. This is an asynchronous process because the slave can respond at any time (provided that the response time limit is observed).

With SPI the master is the only active bus participant. If the SPI master is waiting for an answer from the SPI slave, it has to poll the SPI slave permanently: Only as long as the SPI master sends data to the SPI slave, can the SPI slave send data to the SPI master.

The netIC always processes requests internally in blocks of 8 bytes, i.e. as soon as the device has received 8 bytes, it will internally process them as a block. This procedure continues as long as the device can complete blocks of 8 bytes. Example: Should the request have a length of 9 bytes, the host will have to send at least 7 further bytes so that the netIC device can process the request. The SPI master then will have to keep polling the SPI slave, i.e. it has to keep sending further data to the SPI slave until the netIC has sent the answer to the host.

Only after the netIC device has answered the previous request, can the host send a new request to the netIC device.

The netIC firmware uses a timeout to monitor the SPI communication. The timeout is fixed to a value of 1000 ms. If the netIC does not receive any data for this period via the SPI, the netIC will abort an active request and switch to SPI base mode.

The SPI master can use this monitoring function

- to abort an active request or
- to reestablish the synchronization with the slave via SPI.

16.3.1 Definition of Protocol ‚Modbus via SPI'

At transmission via SPI the Modbus RTU protocol is used in a somewhat modified manner.

The format of the Modbus RTU telegrams is commonly known and very simple. The „pure" Modbus telegram without transport framing (serial or TCP) is defined as:

„<FC><DATA>" (Function code + Data)

By default, the serial telegram - *„Framing Address"* + *CRC* – is omitted due to performance reasons. This reduces effort at the SPI Master and the SPI Slave when processing the telegrams as no CRC calculation is performed. Also the protocol overhead is reduced.

Optionally, the transmission of the CRC checksum can be configured, see

Figure 69: Modbus RTU Configuration Page in netX Configuration Tool - Parameter "Frame Format" on page *181.*

Contrary to the Modbus specification, „Modbus via SPI" does not return the number of the read or written registers as byte count in a byte within the response telegram but as number of the read or written registers within two bytes. There the MSB format applies.

netIC supports the following Modbus function codes via SPI:

- 03 Read Multiple Holding Register
- 16 Write Multiple Holding Register
- 23 Read/Write Multiple Holding Register

Definition of Telegram Elements

Telegram Element	Meaning	Length of Element	Range of Values	Example (hex)
<FC>	Function Code	1 byte	3, 16, 23 (dec) <03>,<10>, <17> (hex)	<03>
<REG>	Register address (address starts with 0, see *Figure 6: Register Area*)	2 bytes	0..4999 (dec) or <00><00>..<13><87> (hex)	<00><0A>
<CNT>	Register count	2 bytes	**For NIC 50-XXX and NIC 52-XXX only:** FC 03, 16, 23: 1 … 504	<00><02>
<EXC>	Exception Code	1 byte		<02>
<DAT>	Data	N bytes CNT*2)	any	<AA><BB><CC><DD>
<CRC>	Checksum	2 bytes	calculated	<CRC><CRC>

Table 110: Definition of Telegram Elements

16.3.1.1 Modbus Exception Codes

The allowed values for telegram element *Exception Code* and their meanings are listed in the following table according to the MODBUS Application Protocol Specification V1.1b3, April 26, 2012, p.48-49, which is available at http://www.modbus.org/.

MODBUS Exception Codes		
Code	**Name**	**Meaning**
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the server. This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server is in the wrong state to process a request of this type, for example because it is unconfigured and is being asked to return register values.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the server. More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, the PDU addresses the first register as 0, and the last one as 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 4, then this request will successfully operate (address-wise at least) on registers 96, 97, 98, 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 5, then this request will fail with Exception Code 0x02 “Illegal Data Address” since it attempts to operate on registers 96, 97, 98, 99 and 100, and there is no register with address 100.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for server. This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register.
04	SERVER DEVICE FAILURE	An unrecoverable error occurred while the server was attempting to perform the requested action.
05	ACKNOWLEDGE	Specialized use in conjunction with programming commands. The server has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the client. The client can next issue a Poll Program Complete message to determine if processing is completed.
06	SERVER DEVICE BUSY	Specialized use in conjunction with programming commands. The server is engaged in processing a long–duration program command. The client should retransmit the message later when the server is free.
08	MEMORY PARITY ERROR	Specialized use in conjunction with function codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The server attempted to read record file, but detected a parity error in the memory. The client can retry the request, but service may be required
0A	GATEWAY PATH UNAVAILABLE	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

Table 111: MODBUS Exception Codes

16.3.2 Example FC3

Reading of multiple registers with FC3:

Read 2 registers beginning with address 10.

Seq.	Direction	Data stream	Master / Slave Status
The master sends a telegram, the slave confirms.			
1.1	M -> S	<03><00><0A><00><02>	Master: STA_TXD
1.2	M <- S	<00><00><00><00><00>	Slave: STA_RXD
The master polls for the answer, the slave returns BUSY.			
2.1	M -> S	<00><00><00>	Master: STA_POLL
2.2	M -> S	<00><00><00>	Slave: STA_BUSY
The master polls, the slave returns the answer.			
3.1	M -> S	<00><00><00><00><00><00><00>	Master: STA_POLLRSP
3.2	M <- S	<03><00><02><AA><BB><CC><DD>	Slave: STA_RSP

Table 112: Example FC3

16.3.3 Example FC16

Writing of multiple registers with FC16

Write 2 registers beginning with address 2000.

Seq.	Direction	Data stream	Master / Slave Status
The master sends a telegram, the slave confirms.			
1.1	M -> S	<10><07><D0><00>02><11><22><33><44>	Master: STA_TXD
1.2	M <- S	<00><00><00><00><00><00><00><00><00>	Slave: STA_RXD
The master polls for the answer, the slave returns BUSY.			
2.1	M -> S	<00><00><00>	Master: STA_POLL
2.2	M -> S	<00><00><00>	Slave: STA_BUSY
The master polls, the slave returns the answer.			
3.1	M -> S	<00><00><00><00><00>	Master: STA_POLLRSP
3.2	M <- S	<10><07><D0><00>02>	Slave: STA_RSP

Table 113: Example FC16

16.3.4 Example FC23

16.3.4.1 Example FC23 without CRC

Combined reading and writing of multiple registers with FC23:

Read 2 registers beginning with address 1000 and write 1 register beginning with address 2000.

Seq.	Direction	Data stream	Master / Slave Status
The master sends a telegram, the slave confirms.			
1.1	M -> S	<17><03><E8><00><02><07><D0><00>01><11><22>	Master: STA_TXD
1.2	M <- S	<00><00><00><00><00><00><00><00><00><00><00>	Slave: STA_RXD
The master polls for the answer, the slave returns BUSY.			
2.1	M -> S	<00><00><00>	Master: STA_POLL
2.2	M -> S	<00><00><00>	Slave: STA_BUSY
The master polls, the slave returns the answer.			
3.1	M -> S	<00><00><00><00><00><00><00>	Master: STA_POLLRSP
3.2	M <- S	<17><00><02><AA><BB><CC><DD>	Slave: STA_RSP

Table 114: Example FC23 without CRC

16.3.4.2 Example FC23 with Modbus Address and with CRC

Combined reading and writing of multiple registers with FC23:

Read 2 registers beginning with address 1000 and write 1 register beginning with address 2000.

Seq.	Direction	Data stream	Master / Slave Status
The master sends a telegram, the slave confirms.			
1.1	M -> S	<ADR><17><03><E8><00><02><07><D0><00><01><11><22> <CRC><CRC>	Master: STA_TXD
1.2	M <- S	<00><00><00><00><00><00><00><00><00><00><00>><00><00>	Slave: STA_RXD
The master polls for the answer, the slave returns BUSY.			
2.1	M -> S	<00><00><00>><00><00>	Master: STA_POLL
2.2	M -> S	<00><00><00>><00><00>	Slave: STA_BUSY
The master polls, the slave returns the answer.			
3.1	M -> S	<00><00><00><00><00><00><00>><00><00>	Master: STA_POLLRSP
3.2	M <- S	<ADR><17><00><02><AA><BB><CC><DD><CRC><CRC>	Slave: STA_RSP

Table 115: Example FC23 with Modbus Address and with CRC

16.3.5 Example FC16 with Exception

Writing of multiple registers with FC16:

Write 2 registers beginning with address 0 which is not allowed, because register 0 … 99 are only readable.

Seq.	Direction	Data stream	Master / Slave Status
The master sends a telegram, the slave confirms.			
1.1	M -> S	<10><00><00><00>02><11><22><33><44>	Master: STA_TXD
1.2	M <- S	<00><00><00><00><00><00><00><00><00>	Slave: STA_RXD
The master polls for the answer, the slave returns BUSY.			
2.1	M -> S	<00><00><00>	Master: STA_POLL
2.2	M -> S	<00><00><00>	Slave: STA_BUSY
The master polls, the slave returns the answer.			
3.1	M -> S	<00><00>	Master: STA_POLLRSP
3.2	M <- S	<90><02>	Slave: STA_RSP

Table 116: Example FC16 with Exception

16.3.6 Busy Exception (05) for high load condition on Ethernet

Note: This feature does apply for Modbus via SPI only.

This feature is supported starting with firmware version 2.0.0.7:

In case of high Ethernet network load, the netIC firmware can answer with exception code 05 on Modbus via SPI. The timeframe for this is 500 ms.

The host application can read register 19 `REG_SYSTEM_INFO_FEATURE_FLAGS` and check flag `REG_SYSTEM_INFO_FEATURE_FLAG_BUSY_RSP (0x0001)`. If this flag is set, the firmware supports this feature. Table 45 on page 76 describes register 19.

Note: Before using this function, the the host application has to activate it.

Activate Busy exception

In order to activate the Busy exception on High load Condition, the host application has to write the fieldbus-specific command `0xB015` using the Command Flags.

Deactivate Busy exception

In order to activate the Busy exception on High load Condition, the host application has to write the fieldbus-specific command `0xB010` using the Command Flags.

17 Requirements to the Host Application Program

17.1 General Aspects

17.1.1 Order of Data

Modbus RTU transfers 16-bit values (registers) in the Motorola format („Big Endian“): First the high byte, then the low byte is transferred. netIC, however, uses the Intel format („Little Endian“). Here first the low byte, then the high byte of a 16 bit word is stored. Therefore, the Modbus parameter „swap“ is by default set to „1“ causing an internal swap of low and high bytes.

At parameters that contain 2 registers, the low-order part (low word) of the parameter value is stored first. The high-order part (high word) of the parameter value is stored on the following register.

Example

A parameter is located on register addresses 311 and 312. Then the low-order part (low word) of the parameter value is located at register address 311 and the high-order part (high word) of the parameter value is located on register address 312.

17.1.2 Packet structure – reduced vs. standard header

Packets are divided in a header and the data part.

The packets are defined within the Protocol API, see the according Protocol API Manual for the used protocol stack. Contrary to all other protocol stacks from Hilscher, the header of any netIC protocol stack is a reduced part of the rcX packets to simplify the implementation for the user.

This section explains the reduced packet structure for netIC and the differences to Hilscher’s standard header.

The netIC API defines the following structure `NETIC_ACYCLIC_PCK_REQ_T` for the reduced header (see *netIC API Examples, Rev.1, Doc.No. DOC140203AN01EN, Sect 3.1*):

```
/** Acyclic packet request structure. */
typedef struct NETIC_ACYCLIC_PCK_REQ_Ttag {
uint32_t ulCmd; /**< Packet command. */
uint32_t ulError; /**< Packet error code. */
uint16_t usLen; /**< Packet data length in bytes. */
uint16_t usId; /**< Packet identifier. */
uint8_t abBuf[MB_MAX_DATA_REQ_LEN]; /**< Packet data. */
}
NETIC_ACYCLIC_PCK_REQ_T;
```

Hilscher’s standard rcX packet header `RCX_PACKET_HEADER` is defined as follows (see *netX Dual-Port Memory Interface for netX based Products*):

```
typedef struct RCX_PACKET_HEADERtag
{

UINT32   ulDest;      /* Destination Queue Handler */
UINT32   ulSrc;       /* Source Queue Handler      */
UINT32   ulDestId;    /* Destination Identifier    */
UINT32   ulSrcId;     /* Source Identifier         */
UINT32   ulLen;       /* Length of Data Field      */
UINT32   ulId;        /* Packet Identifier         */
UINT32   ulState;     /* Status / Error Code       */
UINT32   ulCmd;       /* Command / Response        */
UINT32   ulExt;       /* Extension Field           */
UINT32   ulRout;      /* Routing Information       */
} RCX_PACKET_HEADER;
```

Comparing the reduced header with the standard rcX packet header leads to the following results:

- The fields `ulDest`, `ulSrc`, `ulDestId`, `ulSrcId`, `ulExt` and `ulRout` are omitted in the reduced header.
- The field `ulCmd` is present in both headers and unchanged,
- The field `ulError` of the reduced header corresponds to field `ulState` of the standard header, both fields differ only in their name.
- The field `ulLen` of the standard header is truncated from 32bit to 16 bit within the reduced header (field `usLen` of the reduced header).
- The field `ulId` of the standard header (32bit) is replaced by a 16 bit identifier field within the reduced header (field `usId` of the reduced header). However, `ulId` and `usId` may have different contents.

Keep this different structure in mind when reading the Protocol API manuals as these are written under the assumption, that the standard header is used and do not take care for the special situation for netIC!

17.1.3 Using Acyclic Services

The host application program uses the 'Register Application' packet to notify the netIC, that the host application program wants to receive indication packets. Read and write services of acyclic communication that the netIC firmware has received via the fieldbus or Real-Time Ethernet are forwarded to the host application program then. The application program of the host system has to receive indication packets from the netIC, process them, and send the response packet to the netIC. Beside packets for read/write services, additional packets e.g. information about the link status (`RCX_LINK_STATUS_CHANGE_IND`) can be sent from the netIC firmware to the host application program. The host application program must process and send the response packet to the netIC too.

Important!

A host application program that uses the 'Register Application' packet in order to receive indication packets from the netIC **must answer each** indication packet received with a response packet to the netIC!

17.2 PROFINET IO Device Issues

17.2.1 Validity of PROFINET-Diagnosis and Alarms and their Handles after Disconnection

This section explains how the host application should handle PROFINET Diagnosis and Alarms, if the AR (Application Relation) of the PROFINET connection has been released and reestablished in the mean-time.

Imagine, that an alarm having been sent to the PROFINET IO Device protocol stack (via the `ADD_CHANNEL_DIAG` and `DIAG_ALARM_SEND` requests) is not confirmed by the PROFINET IO Controller and the AR is then released. Now the question arises, whether the alarm remains valid (and has to be confirmed after reconnecting) or the application explicitly needs to remove the diagnosis (by sending a `REMOVE_DIAG` packet) or the stack handles this on its own. In this context, it is furthermore relevant to know whether the delivered diagnosis handle is still valid then.

The following applies for diagnosis:

- A reported diagnosis and its handle are stored within the PROFINET IO Device protocol stack and remain valid even if the PROFINET IO-Controller releases and reestablishes the connection.

The following applies for process alarms:

- Process alarms and their handles are not stored within the PROFINET IO Device protocol stack. If these are still to be processed after disconnection, they have to be sent again.

17.2.2 Different Behavior of STA-LED of NIC52-RE and NIC50-RE when a PROFINET Diagnosis is reported

If a PROFINET diagnosis is reported to the PROFINET IO Device protocol stack running on the NIC52-RE via the packet `PNS_IF_ADD_CHANNEL_DIAG_REQ`, the STA-LED remains on until a hardware reset occurs. This behavior is contrary to that of the NIC50-RE where the STA-LED will not react to a `PNS_IF_ADD_CHANNEL_DIAG_REQ` packet.

The behavior of the NIC52-RE is intended and correct. Other Hilscher PROFINET Devices operate in the same way. Sending an appropriate `PNS_IF_REMOVE_DIAG_REQ` packet will switch off the STA-LED.

However, although being incorrect, the behavior of the NIC50-RE will not be changed in the near future.

17.2.3 Response of PROFINET IO Device application to `PNS_IF_READ_RECORD_IND` with special values

Some application-specific index values used in certification tests might cause errors within these tests if they are not handled correctly. Take the following into account in order to avoid such errors:

Index value `0x8030` ("*IsochronouseModeData*") is application-specific and thus cannot be handled by the PROFINET-IO Device protocol stack.

If your application does not support *IsochronouseModeData*, your application has to respond with errorcode "invalid index" as required by PROFINET-IO specification.

To achieve this, simply set the value "`PNIO_E_IOD_READ_ACCESS_INVALIDINDEX`" (`0xDE80B000`) into the Read Confirmations field ulPnio. Take care of using the correct byte order.

The same applies to the index value range `0x80A0` to `0x80AE`. This value range is reserved for "energy saving profiles". The PROFINET-IO Device protocol stack does not know whether this profile is supported by the application and thus forwards these Read Requests to application.

Hilscher therefore recommends the following handling of `PNS_IF_READ_RECORD_IND` indications:

```
PNIO_TestApp_Read_ind( PNIO_APPL_RSC_T FAR*       ptRsc,
                       PNS_IF_READ_RECORD_IND_T*  ptIndPck )
{
  PNS_IF_READ_RECORD_RSP_T* ptRspPck = (PNS_IF_READ_RECORD_RSP_T*) ptIndPck;

  if ( /* index handled by application */ )
  {
    ptRspPck->tData.ulReadLen = /* size of data delievered */;
    ptRspPck->tData.ulPnio    = PNIO_S_OK;
    /* copy data to abReadData */
  }
  else
  {
    /* unsupported index */
    ptRspPck->tData.ulReadLen = 0;
    ptRspPck->tData.ulPnio    = PNIO_E_IOD_READ_ACCESS_INVALIDINDEX;
  }

  ptRspPck->tHead.ulSta = TLR_S_OK;
  ptRspPck->tHead.ulLen = sizeof(ptRspPck->tData) - sizeof(ptRspPck->tData.abRecordData)
+ ptRspPck->tData.ulReadLen;
  ptRspPck->tData.usAddValue1 = 0x0000;
  ptRspPck->tData.usAddValue2 = 0x0000;

  TLR_QUE_RETURNPACKET(ptRspPck);

  return TLR_S_OK;
}
```

17.3 EtherNet/IP Adapter Issues

17.3.1 No Support for “Quick Connect” for EtherNet/IP Protocol Stack

If you use the EtherNet/IP protocol stack together with NIC50-RE or NIC52-RE, do not activate the functionality “Quick Connect” within your host application. Contrary to other Hilscher devices, this functionality is not available for netIC.

The activation of the “Quick Connect” functionality at the netIC can cause a very substantial increase of the start-up time.

17.3.2 Attributes TTL and MCast not remanent

The EtherNet/IP Adapter protocol stack for netIC does not store the attributes 8 (TTL Value) and 9 (Mcast Config) of the TCP/IP object (0xF5) on its own. If this feature is important for you or your device needs to be certified, you have to adapt your application accordingly in order to store these attributes on shut-down or change and to restore these at start-up.

To do so, the following steps need to be done in the application:

- Activate the attributes 8 (TTL) and 9 (Mcast) of the TCP/IP object (0xF5) for instance by using the packet `EIP_OBJECT_CIP_OBJECT_ATTRIBUTE_ACTIVATE_REQ`
- Write the stored values into these attributes using packet `EIP_OBJECT_CIP_SERVICE_REQ (0x1AF8)`.
- If the attributes are written by the EtherNet/IP network store the according value on reception of `EIP_OBJECT_CIP_OBJECT_CHANGE_IND (0x1AFA)`.

This feature is relevant for EtherNet/IP certification.

17.4 EtherCAT Slave Issues

The Ethernet ports are deactivated after a device reset and are enabled with the BusOn command only. The netIC Bus startup parameter sets if netIC starts the communication automatically or if the communication start is controlled by the application. In case, the Bus startup parameter is set to 'application controlled' the application must use the BusOn command to enable the Ethernet ports and to start the communication.

18 Application Examples

The netIC DVD contains application examples in C programming language in the folder `Example and API\0. Application Example`.

Path	Contents
Components\netIC_API	Operating system neutral sources of the netIC API. The document *Application note: netIC API Examples* describes the functions of the netIC API.
Components\Example	IOExchange: Application example for data exchange (without configuration of the netIC by the application). PROFIBUS DP Slave (DPS), EtherNet/IP Adapter (EIS), Open Modbus/TCP (OMB), PROFINET IO Device (PNS): Application examples for data exchange and configuration of the netIC by the application. The C file of the application example contains a description of the functions specific example.
ReleaseNotes	Revision list of the netIC ExampleKit.
Targets\Windows	Projects for Visual Studio 2013 (operating system Windows) File `Targets\Windows\netIC_Demo\netIC_Demo.sln` contains multiple projects. These projects use files of the folder `components` and its subfolders. Copy folder "`0. Application Example`" including all subfolders to your PC if you intent to use these Visual Studio project.

Table 117: Application examples on the netIC DVD

19 Decommissioning, Deinstallation, Replacement and Disposal

This section explains what you must take into account when putting the netIC Communication ICs out of operation.

Note: In order to avoid personal and material damage do not remove this device from a production line without having ensured a secure operation of the production line during and after the removal of the device.

19.1 Deinstallation and Replacement

This section explains what you must take into account when physically removing or replacing the netIC Communication ICs.

For the deinstallation of the netIC Communication IC from the device into which the netIC Communication IC had been integrated (also called “host system” or “target environment”), proceed as follows:

1. **Preparation**: In order to avoid damage or destruction, adhere to the necessary safety precautions for components that are vulnerable by electrostatic discharge.

NOTICE

Electrostatically sensitive Devices

- To prevent damage to the host system and the netIC Communication IC, make sure, that the netIC Communication IC is grounded and the host system and make sure, that you are discharged when you mount/dismount the netIC Communication IC.

USA:

NOTICE

Electrostatically sensitive Devices

- To prevent damage to the host system and the netIC Communication IC, make sure, that the netIC Communication IC is grounded and the host system and make sure, that you are discharged when you mount/dismount the netIC Communication IC.

➢ **Step 1**: If necessary, remove the housing of the host device according to the documentation supplied by the manufacturer of the host device. Obey strictly both to the following safety instruction and to the respective instruction manual for this device.

WARNING!

Lethal Electrical Shock caused by parts with more than 50V!

- HAZARDOUS VOLTAGE may be inside of the device into which the netIC Communication IC is to be integrated.
- First disconnect the power plug of the device into which the netIC Communication IC is to be integrated.
- Make sure, that the device is really free of electric power.
- In any case, strictly adhere to the instructions given in the documentation of the device provided by its manufacturer.
- Definitely avoid touching open contacts or ends of wires
- Open the device and install or remove the netIC Communication IC only after having completed all of the preceding steps.

USA:

⚠ WARNING

Lethal Electrical Shock caused by parts with more than 50V!

- HAZARDOUS VOLTAGE may be inside of the device into which the netIC Communication IC is to be integrated.
- First disconnect the power plug of the device into which the netIC Communication IC is to be integrated.
- Make sure, that the device is really free of electric power.
- In any case, strictly adhere to the instructions given in the documentation of the device provided by its manufacturer.
- Definitely avoid touching open contacts or ends of wires
- Open the device and install or remove the netIC Communication IC only after having completed all of the preceding steps.

➢ **Step 2**: Remove the netIC carefully from its DIL-32 socket.

➢ **Step 3**: If the netIC Communication IC needs be replaced by another one, then replace it by plugging in the netIC Communication IC device carefully but firmly into the designated DIL-32 socket.

➢ **Step 4**: If you had opened the housing of the device in step 2, then close it now. Close the housing of host device carefully, if you opened it before. Again, strictly adhere to the documentation supplied by the host device’s manufacturer.

➢ **Step 5**: Connect the device with its supply voltage and switch it on again. Adhere of the commissioning rules of the supplier of the device. Check, whether the device behaves normally.

➢ **Step 6**: Obey to the disposal rules explained below!

19.2 Disposal of Waste Electronic Equipment

According to the European Directive 2002/96/EG “Waste Electrical and Electronic Equipment (WEEE)”, waste electronic equipment may not be disposed of as household waste. As a consumer, you are legally obliged to dispose of all waste electronic equipment according to national and local regulations.

Waste Electronic Equipment

- This product must not be treated as household waste.
- This product must be disposed of at a designated waste electronic equipment collecting point.

20 Technical Data

20.1 Technical Data netIC DIL-32 Communication ICs

20.1.1 NIC 50-RE

NIC 50-RE	Parameter	Value
Communication controller	Type	netX 50 processor
Integrated memory	RAM	8 MB SDRAM
	FLASH	4 MB serial Flash EPROM
Modbus RTU communication	Type	Master/Slave
	Data transport	Modbus RTU protocol
Ethernet communication	Supported Real-Time Ethernet communication systems (determined by the loaded firmware)	EtherCAT Slave
		EtherNet/IP Adapter (Slave)
		Open Modbus/TCP
		Powerlink Controlled Node/Slave
		PROFINET IO Device
		Sercos Slave
		VARAN Client (Slave)
Ethernet interface	Transmission rate	100 MBit/s 10 MBit/s (depending on loaded firmware)
	Interface type	100 BASE-TX, isolated 10 BASE-T (depending on loaded firmware)
	Half duplex / Full duplex	supported (at 100 MBit/s)
	Auto-Negotiation	depending on firmware
	Auto-Crossover	depending on firmware
Serial interface to host (Modbus RTU)	UART	RXD, TXD, RTS
	UART Baudrate	1,2 kBit/s 2,4 kBit/s 4,8 kBit/s 9,6 kBit/s (default rate) 19,2 kBit/s 38,4 kBit/s 57,6 kBit/s 115,2 kBit/s
	UART Control	by RTS signal
	SPI	SPI_MOSI, SPI_MISO, SPI_CLK, SPI_CS (Chip Select))
	SPI Baudrate	Auto detection
	SPI Clockrate (Maximum)	1 MHz
	SPI Transmission rate (Typical value for 100 bit)	max. 102 KBit/s
	SPI Transmission mode	Full duplex
	I2C Master/Slave	not supported
Serial I/O Shift Register Interface	Input	max. 256* 8 bit shift registers
	Output	max. 256* 8 bit shift registers
	Baudrate (Maximum)	5000000 Baud
Diagnostic Interface	UART	RXD, TXD

Table 118: Technical Data NIC 50-RE (Part 1)

NIC 50-RE	Parameter	Value
Display	LED Display	SYS System Status
		Pins available with signals for external LEDs:
		COM Communication status
		TX/RX0n, TX/RX1n Ethernet activity status
		LINK0n, LINK1n Ethernet link status
		FBLED
Power supply	Voltage	+3,3 V ± 5 % DC
	Current at 3,3 V (typically)	400 mA
	Power Consumption	appr. 1.3 W
Environmental conditions	Ambient temperature range for operation	depending on used heat sink
	NIC 50-RE with original Hilscher heat sink	–20 °C … +70 °C (air flow 0.5 m/s)
	NIC 50-RE/NOHS (without any heat sink)	0 °C … +55 °C (air flow 0.5 m/s)
	NIC 50-RE with heat sink with R_{th} = 7 K/W	–20 °C … +70 °C (air flow 0.5 m/s)
	NIC 50-RE with Hilscher defined PCB heat sink	–20 °C … +60 °C (air flow 0.5 m/s)
	Ambient temperature range for storage	–40 °C … +85 °C
	Relative humidity	5 % … 85 % (non condensing)
	Altutude	0 … 2000 m
Device	Dimensions (L x W x H)	42 mm x 21 mm x 17.4 mm (including pins)
	Weight	appr. 10 g
	Length of pins	3.2 mm
	Diameter of pins	0.047 mm
	Distance of pins	2.54 mm
	Mounting	directly into DIL-32 socket
	Protection Class	
	RoHS	Yes
EMC	CE sign	Yes
	UKCA sign	Yes
	Emission	EN 55011 / BS EN 55011
	Immunity	EN 61000-4-2 / BS EN 61000-4-2
Configuration	by software tool (standard)	netX Configuration Tool
	via Modbus RTU	by writing into Modbus RTU registers

Table 119: Technical Data NIC 50-RE (Part 2)

Electrical Immunity to Interference and Radio Frequency

NIC 50-RE	Method	Criterion
Electrostatic discharge (ESD) according to IEC/EN 61000-4-2 / BS EN 61000-4-2	8 kV Air discharge method	Criterion B
	4 kV Contact discharge method	Criterion B
Fast transient interferences (Burst), according to IEC/EN 61000-4-4 / BS EN 61000-4-4	2 kV Communication and data lines	Criterion A
Surge voltage, according to IEC/EN 61000-4-5 / BS EN 61000-4-5	1 kV Communication and data lines,	Criterion A
Radiated RF, according to IEC/EN 61000-4-3 / BS EN 61000-4-3	80-2000MHz, 10V/m, 80% AM / 1kHz	Criterion A
	1.4-2.0GHz, 10V/m, 80% AM / 1kHz	Criterion A
Conducted RF, according to IEC/EN 61000-4-6 / BS EN 61000-4-6	0,15-80MHz, 10V, 80% AM / 1kHz for lines > 3m	Criterion A

Table 120: Electrical Immunity to Interference and Radio Frequency NIC 50-RE

20.1.2 NIC 50-COS

NIC 50-COS	Parameter	Value
Communication controller	Type	netX 50 processor
Integrated memory	RAM	8 MB SDRAM
	FLASH	4 MB serial Flash EPROM
Modbus RTU communication	Type	Master/Slave
	Data transport	Modbus RTU protocol
CANopen communication	Supported communication standard/ firmware	CANopen
CANopen interface	Transmission rate	10 kBits/s to 1 MBit/s
	Interface type	ISO 11898, potential free
Serial interface to host (Modbus RTU)	UART	RXD, TXD, RTS
	UART Baudrate	1,2 kBit/s 2,4 kBit/s 4,8 kBit/s 9,6 kBit/s (default rate) 19,2 kBit/s 38,4 kBit/s 57,6 kBit/s 115,2 kBit/s
	UART Control	by RTS signal
	SPI	SPI_MOSI, SPI_MISO, SPI_CLK, SPI_CS (Chip Select))
	SPI Baudrate	Auto detection
	SPI Clockrate (Maximum)	1 MHz
	SPI Transmission rate (Typical value for 100 bit)	max. 102 KBit/s
	SPI Transmission mode	Full duplex
	I2C Master/Slave	not supported
Serial I/O Shift Register Interface	Input	max. 256* 8 bit shift registers
	Output	max. 256* 8 bit shift registers
	SSIO Baudrate (Maximum)	5,000,000 Baud
Diagnostic Interface	UART	RXD, TXD
Display	LED Display	SYS System Status
		Pins available with signals for external LEDs:
		CAN CANopen status
		FBLED
Power supply	Voltage	+3,3 V ± 5 % DC
	Current at 3,3 V (typically)	330 mA
	Power Consumption	appr. 1.1 W
Environmental conditions	Ambient temperature range for operation NIC 50-COS without heat sink	–20 °C … +70 °C (air flow 0.5 m/s)
	Ambient temperature range for storage	–40 °C … +85 °C
	Relative humidity	5 % … 85 % (non condensing)
	Altutude	0 … 2000 m

Table 121: Technical Data NIC 50-COS (Part 1)

NIC 50-COS	Parameter	Value
Device	Dimensions (L x W x H)	42 mm x 21 mm x 17.4 mm (including pins)
	Weight	appr. 10 g
	Length of pins	3.2 mm
	Diameter of pins	0.047 mm
	Distance of pins	2.54 mm
	Mounting	directly into DIL-32 socket
	Protection Class	
	RoHS	Yes
EMC	CE sign	Yes
	UKCA sign	Yes
	Emission	EN 55011 / BS EN 55011
	Immunity	EN 61000-4-2 / BS EN 61000-4-2
Configuration	by software tool (standard)	netX Configuration Tool
	via Modbus RTU	by writing into Modbus RTU registers

Table 122: Technical Data NIC 50-COS (Part 2)

Electrical Immunity to Interference and Radio Frequency

NIC 50-COS	Method	Criterion
Electrostatic discharge (ESD) according to IEC/EN 61000-4-2 / BS EN 61000-4-2	8 kV Air discharge method	Criterion A
	4 kV Contact discharge method	Criterion A
Fast transient interferences (Burst), according to IEC/EN 61000-4-4 / BS EN 61000-4-4	2 kV Communication and data lines	Criterion A
Surge voltage, according to IEC/EN 61000-4-5 / BS EN 61000-4-5	1 kV Communication and data lines,	Criterion A
Radiated RF, according to IEC/EN 61000-4-3 / BS EN 61000-4-3	80-2000MHz, 10V/m, 80% AM / 1kHz	Criterion A
Conducted RF, according to IEC/EN 61000-4-6 / BS EN 61000-4-6	0,15-80MHz, 3V, 80% AM / 1kHz	Criterion A
	0,15-80MHz, 10V, 80% AM / 1kHz for lines >3m	Criterion A

Table 123: Electrical Immunity to Interference and Radio Frequency NIC 50-COS

20.1.3 NIC 50-DNS

NIC 50-DNS	Parameter	Value
Communication controller	Type	netX 50 processor
Integrated memory	RAM	8 MB SDRAM
	FLASH	4 MB serial Flash EPROM
Modbus RTU communication	Type	Master/Slave
	Data transport	Modbus RTU protocol
DeviceNet communication	Supported communication standard/ firmware	DeviceNet
DeviceNet interface	Transmission rate	125, 250, 500 kBits/s
	Interface type	ISO 11898, potential free
Serial interface to host (Modbus RTU)	UART	RXD, TXD, RTS
	UART Baudrate	1,2 kBit/s 2,4 kBit/s 4,8 kBit/s 9,6 kBit/s (default rate) 19,2 kBit/s 38,4 kBit/s 57,6 kBit/s 115,2 kBit/s
	UART Control	by RTS signal
	SPI	SPI_MOSI, SPI_MISO, SPI_CLK, SPI_CS (Chip Select))
	SPI Baudrate	Auto detection
	SPI Clockrate (Maximum)	1 MHz
	SPI Transmission rate (Typical value for 100 bit)	max. 102 KBit/s
	SPI Transmission mode	Full duplex
	I2C Master/Slave	not supported
Serial I/O Shift Register Interface	Input	max. 256* 8 bit shift registers
	Output	max. 256* 8 bit shift registers
	SSIO Baudrate (Maximum)	5000000 Baud
Diagnostic Interface	UART	RXD, TXD
Display	LED Display	SYS System Status
		Pins available with signals for external LEDs:
		MNS Module network status
		FBLED
Power supply	Voltage	+3,3 V ± 5 % DC
	Current at 3,3 V (typically)	370 mA
	Power Consumption	appr. 1.2 W

Table 124: Technical Data NIC 50- DNS (Part 1)

NIC 50-DNS	Parameter	Value
Environmental conditions	Ambient temperature range for operation NIC 50-DNS without heat sink	–20 °C … +70 °C (air flow 0.5 m/s)
	Ambient temperature range for storage	–40 °C … +85 °C
	Relative humidity	5 % … 85 % (non condensing)
	Altutude	0 … 2000 m
Device	Dimensions (L x W x H)	42 mm x 21 mm x 17.4 mm (including pins)
	Weight	appr. 10 g
	Length of pins	3.2 mm
	Diameter of pins	0.047 mm
	Distance of pins	2.54 mm
	Mounting	directly into DIL-32 socket
	Protection Class	
	RoHS	yes
EMC	CE sign	Yes
	UKCA sign	Yes
	Emission	EN 55011 / BS EN 55011
	Immunity	EN 61000-4-2 / BS EN 61000-4-2
Configuration	by software tool (standard)	netX Configuration Tool
	via Modbus RTU	by writing into Modbus RTU registers

Table 125: Technical Data NIC 50-DNS (Part 2)

Electrical Immunity to Interference and Radio Frequency

NIC 50-DNS	Method	Criterion
Electrostatic discharge (ESD) according to IEC/EN 61000-4-2 / BS EN 61000-4-2	8 kV Air discharge method	Criterion A
	4 kV Contact discharge method	Criterion A
Fast transient interferences (Burst), according to IEC/EN 61000-4-4 / BS EN 61000-4-4	2 kV Communication and data lines	Criterion B
Surge voltage, according to IEC/EN 61000-4-5 / BS EN 61000-4-5	1 kV Communication and data lines,	Criterion B
	0,5 kV DeviceNet Supply lines,	Criterion A
Radiated RF, according to IEC/EN 61000-4-3 / BS EN 61000-4-3	80-2000MHz, 10V/m, 80% AM / 1kHz	Criterion A
Conducted RF, according to IEC/EN 61000-4-6 / BS EN 61000-4-6	0,15-80MHz, 3V, 80% AM / 1kHz	Criterion A
	0,15-80MHz, 10V, 80% AM / 1kHz for lines >3m	Criterion A

Table 126: Electrical Immunity to Interference and Radio Frequency NIC 50-DNS

20.1.4 NIC 50-DPS

NIC 50-DPS	Parameter	Value
Communication controller	Type	netX 50 processor
Integrated memory	RAM	8 MB SDRAM
	FLASH	4 MB serial Flash EPROM
Modbus RTU communication	Type	Master/Slave
	Data transport	Modbus RTU protocol
PROFIBUS communication	Supported communication standard/ firmware	PROFIBUS DP
PROFIBUS interface	Transmission rate	Fixed values ranging from 9,6 kBits/s to 12 MBit/s
	Interface type	RS-485
	Auto-detection	yes
Serial interface to host (Modbus RTU)	UART	RXD, TXD, RTS
	UART Baudrate	1,2 kBit/s 2,4 kBit/s 4,8 kBit/s 9,6 kBit/s (default rate) 19,2 kBit/s 38,4 kBit/s 57,6 kBit/s 115,2 kBit/s
	UART Control	by RTS signal
	SPI	SPI_MOSI, SPI_MISO, SPI_CLK, SPI_CS (Chip Select))
	SPI Baudrate	Auto detection
	SPI Clockrate (Maximum)	1 MHz
	SPI Transmission rate (Typical value for 100 bit)	max. 102 KBit/s
	SPI Transmission mode	Full duplex
	I2C Master/Slave	not supported
Serial I/O Shift Register Interface	Input	max. 256* 8 bit shift registers
	Output	max. 256* 8 bit shift registers
	SSIO Baudrate (Maximum)	5000000 Baud
Diagnostic Interface	UART	RXD, TXD
Display	LED Display	SYS System Status
		Pins available with signals for external LEDs:
		COM Communication status
		FBLED
Power supply	Voltage	+3,3 V ± 5 % DC
	Current at 3,3 V (typically)	330 mA
	Power Consumption	appr. 1.1 W

Table 127: Technical Data NIC 50-DPS (Part 1)

NIC 50-DPS	Parameter	Value
Environmental conditions	Ambient temperature range for operation NIC 50-DPS without heat sink	–20 °C … +70 °C (air flow 0.5 m/s)
	Ambient temperature range for storage	–40 °C … +85 °C
	Relative humidity	5 % … 85 % (non condensing)
	Altutude	0 … 2000 m
Device	Dimensions (L x W x H)	42 mm x 21 mm x 17.4 mm (including pins)
	Weight	appr. 10 g
	Length of pins	3.2 mm
	Diameter of pins	0.047 mm
	Distance of pins	2.54 mm
	Mounting	directly into DIL-32 socket
	Protection Class	
	RoHS	yes
EMC	CE sign	Yes
	UKCA sign	Yes
	Emission	EN 55011 / BS EN 55011
	Immunity	EN 61000-4-2 / BS EN 61000-4-2
Configuration	by software tool (standard)	netX Configuration Tool
	via Modbus RTU	by writing into Modbus RTU registers

Table 128: Technical Data NIC 50-DPS (Part 2)

Electrical Immunity to Interference and Radio Frequency

NIC 50-DPS	Method	Criterion
Electrostatic discharge (ESD) according to IEC/EN 61000-4-2 / BS EN 61000-4-2	8 kV Air discharge method	Criterion A
	4 kV Contact discharge method	Criterion A
Fast transient interferences (Burst), according to IEC/EN 61000-4-4 / BS EN 61000-4-4	2 kV Communication and data lines	Criterion A
Surge voltage, according to IEC/EN 61000-4-5 / BS EN 61000-4-5	1 kV Communication and data lines,	Criterion A
Radiated RF, according to IEC/EN 61000-4-3 / BS EN 61000-4-3	80-2000MHz, 10V/m, 80% AM / 1kHz	Criterion A
	1.4-2.0GHz, 10V/m, 80% AM / 1kHz	Criterion A
Conducted RF, according to IEC/EN 61000-4-6 / BS EN 61000-4-6	0,15-80MHz, 10V, 80% AM / 1kHz	Criterion A
	0,15-80MHz, 10V, 80% AM / 1kHz for lines >3m	Criterion A

Table 129: Electrical Immunity to Interference and Radio Frequency NIC 50-DPS

20.1.5 NEC 52-RE

Important: All data given here apply for NIC 52-RE, revision 3.

NIC 52-RE	Parameter	Value
Article number	NIC 52-RE	1544.100
Communication controller	Type	netX 52 processor
Integrated memory	RAM	8 MB SDRAM
	FLASH	4 MB Quad SPI Flash
Modbus RTU communication	Type	Master/Slave
	Data transport	Modbus RTU protocol
Ethernet communication	Supported Real-Time Ethernet communication systems (determined by the loaded firmware)	CCLink IE Field Basic Slave
		EtherCAT Slave
		EtherNet/IP Adapter (Slave)
		Open Modbus/TCP
		PROFINET IO Device
		Sercos Slave
Ethernet interface	Transmission rate	100 MBit/s 10 MBit/s (depending on loaded firmware)
	Interface type	100 BASE-TX, isolated 10 BASE-T (depending on loaded firmware)
	Half duplex / Full duplex	supported (at 100 MBit/s)
	Auto-Negotiation	depending on firmware
	Auto-Crossover	depending on firmware
Serial interface to host (Modbus RTU)	UART Signals	RXD, TXD, RTS
	UART Baudrate	1,2 kBit/s 2,4 kBit/s 4,8 kBit/s 9,6 kBit/s (default rate) 19,2 kBit/s 38,4 kBit/s 57,6 kBit/s 115,2 kBit/s
	UART Control	by RTS signal
	SPI Signals	SPI_MOSI, SPI_MISO, SPI_CLK, SPI_CS (Chip Select))
	SPI Baudrate	Auto detection
	SPI Clockrate (Maximum)	1 MHz
	SPI Transmission rate (Typical value for 100 bit)	max. 102 KBit/s
	SPI Transmission mode	Full-duplex
Serial I/O Shift Register Interface	Input	max. 256* 8 bit shift registers
	Output	max. 256* 8 bit shift registers
	Baudrate (Maximum)	5000000 Baud
Diagnostic Interface	UART	RXD, TXD

Table 130: Technical Data NIC 52-RE (Part 1)

NIC 52-RE	Parameter	Value
Display	LED Display onboard	SYS System Status
	Pins available with signals for external LEDs on customer's design or on evaluation board	COM Communication status
		TX/RX0n, TX/RX1n Ethernet activity status
		LINK0n, LINK1n Ethernet link status
		FBLED
Power supply	Voltage	+3,3 V ± 5 % DC
	Current at 3,3 V (typically)	450 mA
	Power consumption	appr. 1.5 W
Environmental conditions	Ambient temperature range for operation	–20 °C … +70 °C (air flow 0.5 m/s)
	Ambient temperature range for storage	–40 °C … +85 °C
	Relative humidity	5 % … 85 % (non condensing)
	Altutude	0 … 2000 m
Device	Dimensions (L x W x H)	42 mm x 21 mm x 10.2 mm (including pins)
	Weight	appr. 8 g
	Length of pins	3.7 ± 0.2 mm
	Diameter of pins	0.047 mm
	Distance of pins	2.54 mm
	Mounting	directly into DIL-32 socket
	RoHS	Yes
EMC	CE sign	Yes
	UKCA sign	Yes
	Emission	EN 61000-6-4 / BS EN 61000-6-4
	Immunity	EN 61000-6-2 / BS EN 61000-6-2
Configuration	by software tool (standard)	netX Configuration Tool
	via Modbus RTU	by writing into Modbus RTU registers
Firmware update	Software	netX Configuration Tool
	über Web Server	Browser

Table 131: Technical Data NIC 52-RE (Part 2)

Electrical Immunity to Interference and Radio Frequency

NIC 52-RE	Method	Criterion
Electrostatic discharge (ESD) according to IEC/EN 61000-4-2 / BS EN 61000-4-2	8 kV Air discharge method	Criterion B
	6 kV Contact discharge method	Criterion B
Fast transient interferences (Burst), according to IEC/EN 61000-4-4 / BS EN 61000-4-4	2.2 kV Communication and data lines	Criterion B
Surge voltage, according to IEC/EN 61000-4-5 / BS EN 61000-4-5	1 kV Communication and data lines,	Criterion A
Radiated RF, according to IEC/EN 61000-4-3 / BS EN 61000-4-3	80-3000MHz, 10V/m, 80% AM / 1kHz	Criterion A
Conducted RF, according to IEC/EN 61000-4-6 / BS EN 61000-4-6	0,15-80MHz, 10V, 80% AM / 1kHz	Criterion A

Table 132: Electrical Immunity to Interference and Radio Frequency NIC 52-RE

20.2 Technical Data Evaluation Board

20.2.1 NICEB

NICEB	Parameter	Value
Power supply	Allowed voltage range	9 V -30 V DC, 24 V DC recommended
	Typical current at 24 V	Depends on netIC
	Power adapter plug	Contained in delivery
Switches/pushbuttons	Input data	16 DIP switches, connected to SSIO signal lines DI0-DI15
	Pushbuttons	For reset, boot, configuration/GPIO
LED display	Output data	16 LEDs yellow, connected to SSIO signal lines DO0-DO15
	COM communication status	1 Duo-LED green/red
	FBLED	1 LED red
Interface	DIL-32 socket	For all netIC types
	Ethernet interface	2 x RJ45
	Bus interface	Via Fieldbus Adapter (from connector kit NICEB-CONKIT)
	Host interface	9 pin D-Sub connector RS232/RS422/RS485, configurable by jumpers, female connector
	Diagnostic interface (For Firmware-Download and Configuration)	9 pin D-Sub connector RS232/RS422/RS485, configurable by jumpers, male connector
	Serial I/O Shift Register Interface	16 x Input and 16x Output on contact strip
Dimensions	Dimensions (L x W x H)	100 x 65 x 18 mm (without netIC)
Environment	RoHS	Yes

Table 133: Technical Data NICEB

No CE Sign!

- The Evaluation Board NICEB has only been designed for test use. It has **no CE sign** and it has **not been tested regarding its emission and immunity behavior**. Therefore it is not suited for use in an industrial production environment!

20.3 Technical Data of the Communication Protocols

20.3.1 CC-Link IE Field Basic Slave

Parameter	Description
Maximum number of cyclic input data	RY data: 128 bytes (1024 bits) RWw data: 512 words (16 bit)
Maximum number of cyclic output data	RX data: 128 bytes (1024 bits) RWr data: 512 words (16 bit)
Occupied stations	1 … 16 (1 station has 64 bits RY data, 32 words RWw data, 64 bits RX data, and 32 words RWr data.)
Acyclic communication	SLMP Server and Client
Data transport layer	Ethernet II, IEEE 802.3
Baud rate	100 MBit/s
Reference to firmware version (stack version)	V2.2 (V1.1)
Ports	
Cyclic data	61450 (UDP)
Discovery and SLMP Server	61451 (UDP)
SLMP Parameter	45237 (UDP)
SLMP Communication	20000 (UDP)

Table 134: Technical data CC-Link IE Field Basic Slave protocol

20.3.2 EtherCAT Slave

For NIC 50-RE only:

Parameter	Description
Maximum number of cyclic input data	1024 bytes (netX 50)
Maximum number of cyclic output data	1024 bytes (netX 50)
Type	Complex Slave
Functions	Emergency
FMMUs	8 (netX 50)
SYNC Manager	4 (netX 50)
Distributed Clocks (DC)	Supported, 32 Bit
Baud rate	100 MBit/s
Data transport layer	Ethernet II, IEEE 802.3
Limitations	The netIC gateway is designed for cyclic data exchange. Acyclic communication for user data transfer can only be used if the host application program supports this, which means programming effort for the host application program. Then 'SDO Master-Slave' can be used.
Reference to firmware/stack version	V2.5.x.x

Table 135: Technical Data EtherCAT Slave Protocol

For NIC 52-RE only:

Parameter	Description
Maximum number of cyclic input data	1024 bytes (netX 50/51/52)
Maximum number of cyclic output data	1024 bytes (netX 50/51/52)
Type	Complex Slave
Supported Protocols	SDO client and server side protocol (CoE component) CoE Emergency messages (CoE component) Ethernet over EtherCAT (EoE component) File Access over EtherCAT (FoE component) AoE (supported since stack version 4.3) Complete Access (supported since stack version 4.3)
Acyclic communication	SDO SDO Master-Slave SDO Slave-Slave (depending on Master capability)
Functions	Emergency
FMMUs	8 (netX 50/51/52)
SYNC Manager	4 (netX 50/51/52)
Distributed Clocks (DC)	Supported, 32 Bit
Baud rate	100 MBit/s
Data transport layer	Ethernet II, IEEE 802.3
Reference to firmware/stack version	V2.3 (Stack: V4.7)

Table 136: Technical Data EtherCAT Slave Protocol

20.3.3 EtherNet/IP Adapter (Slave)

NIC 50-RE only:

Parameter	Description
Maximum number of input data	504 bytes
Maximum number of output data	504 bytes
IO Connection (implicit)	1 exclusive owner, up to 2 listen only
IO Connection type	Cyclic, minimum 1 ms
UCMM	Supported
Maximum number of connections	8, explicit and implicit connections
Predefined standard objects	Identity Object Message Route Object Assembly Object Connection Manager Ethernet Link Object TCP/IP Object
Topology	Tree, Line, Ring
DLR (Device Level Ring)	Beacon based 'Ring Node'
ACD (Address Conflict Detection)	Supported
DHCP	Supported
BOOTP	Supported
Baud rates	10 and 100 MBit/s
Data transport layer	Ethernet II, IEEE 802.3
Integrated switch	Supported
Limitations	Quick Connect not suported The netIC gateway is designed for cyclic data exchange. Acyclic communication for user data transfer can only be used if the host application program supports this, which means programming effort for the host application program. Then sercives 'Get_Attribute, Set_Attribute' can be used. CIP Sync Services are not implemented TAGs are not supported
Reference to firmware/stack version	V1.5 (Stack V2.7)

Table 137: Technical Data EtherNet/IP Adapter (Slave) Protocol

NIC 52-RE only:

Parameter	Description
Maximum number of input data	504 bytes
Maximum number of output data	504 bytes
IO Connection (implicit)	1 exclusive owner, up to 2 listen only
IO Connection type	Cyclic, minimum 1 ms
UCMM	Supported
Maximum number of connections	8, explicit and implicit connections
Predefined standard objects	Identity Object Message Route Object Assembly Object Connection Manager Ethernet Link Object TCP/IP Object
Topology	Tree, Line, Ring
DLR (Device Level Ring)	Beacon based 'Ring Node'
ACD (Address Conflict Detection)	Supported
DHCP	Supported
BOOTP	Supported
Baud rates	10 and 100 MBit/s
Data transport layer	Ethernet II, IEEE 802.3
Integrated switch	Supported
Limitations	Quick Connect not suported The netIC gateway is designed for cyclic data exchange. Acyclic communication for user data transfer can only be used if the host application program supports this, which means programming effort for the host application program. Then sercives 'Get_Attribute, Set_Attribute' can be used. CIP Sync Services are not implemented TAGs are not supported
Reference to firmware/stack version	V2.3 (Stack V2.13)

Table 138: Technical Data EtherNet/IP Adapter (Slave) Protocol

20.3.4 Open Modbus/TCP

Parameter	Description
Maximum number of input data	999 Registers
Maximum number of output data	994 Registers
Acyclic communication	Read/Write Register: - Maximum 125 Registers per Read Telegram (FC 3, 4, 23), - Maximum 121 Registers per Write Telegram (FC 23), - Maximum 123 Registers per Write Telegram (FC 16) Read/Write Coil: - Maximum 2000 Coils per Read Telegram (FC 1, 2), - Maximum 1968 Coils per Write Telegram (FC 15)
Modbus Function Codes	1, 2, 3, 4, 5, 6, 7, 15, 16, 23 (FC 23 only if netIC in Open Modbus/TCP Server mode)
Protocol Mode	IO Server
Baud rates	10 and 100 MBit/s
Data transport layer	Ethernet II, IEEE 802.3
Reference to firmware/stack version	V1.5 (Stack V2.5) V2.3 (Stack V2.6)

Table 139: Technical Data Open Modbus/TCP Protocol

20.3.5 Powerlink Controlled Node / Slave

NIC 50-RE only:

Parameter	Description
Maximum number of cyclic input data	1490 bytes
Maximum number of cyclic output data	1490 bytes
Functions	SDO over ASND and UDP
Baud rate	100 MBit/s, half-duplex
Data transport layer	Ethernet II, IEEE 802.3
Ethernet POWERLINK version	V 2
Limitation	No slave to slave communication The netIC gateway is designed for cyclic data exchange. Acyclic communication for user data transfer can only be used if the host application program supports this, which means programming effort for the host application program. Then ‘SDO Upload/Download’ can be used.
Reference to firmware/stack version	V2.1.x.x

Table 140: Technical Data POWERLINK Controlled Node (Slave) Protocol

NIC 52-RE only:

Parameter	Description
Maximum number of cyclic input data	800 bytes
Maximum number of cyclic output data	800 bytes
Acyclic data transfer	SDO Upload/Download
Functions	SDO over ASND and UDP
Baud rate	100 MBit/s, half-duplex
Data transport layer	Ethernet II, IEEE 802.3
Ethernet POWERLINK version	V 2
Limitation	No slave to slave communication
Reference to firmware/stack version	V2.3 (Stack V3.4)

Table 141: Technical Data POWERLINK Controlled Node Protocol

20.3.6 PROFINET IO-RT-Device

NIC 50-RE only:

Parameter	Description
Maximum number of cyclic input data	256 bytes if netX Configuration Tool is used for configuration. 1024 bytes if the configuration is done via Modbus RTU (programming effort for the host application program)
Maximum number of cyclic output data	256 bytes if netX Configuration Tool is used for configuration. 1024 bytes if the configuration is done via Modbus RTU (programming effort for the host application program)
Maximum number of modules	Max. 4 input modules and max. 4 output modules can be configured with the netX Configuration Tool. Max. 19 modules if the configuration is done via Modbus RTU (programming effort for the host application program)
Supported protocols	RTC – Real Time Cyclic Protocol, Class 1 and 2 (unsynchronized), Class 3 (synchronized) RTA – Real Time Acyclic Protocol DCP – Discovery and configuration Protocol CL-RPC – Connectionless Remote Procedure Call LLDP – Link Layer Discovery Protocol SNMP – Simple Network Management Protocol MRP – MRP Client
Used Protocols (subset)	UDP, IP, ARP, ICMP (Ping)
Topology recognition	LLDP, SNMP V1, MIB2, physical device
VLAN- and priority tagging	yes
Context Management by CL-RPC	Supported
Identification & Maintenance	Read and write of I&M 1-4
Fast Startup	Supported. Hardware requirements: NIC 50-RE: Hardware revision 3 or higher
Minimum cycle time	1 ms for RTC1, RTC 2 and RTC3
Baud rate	100 MBit/s
Data transport layer	Ethernet II, IEEE 802.3

Parameter	Description
Limitations	The netIC gateway is designed for cyclic data exchange. Acyclic communication for user data transfer can only be used if the host application program supports this, which means programming effort for the host application program. Then services ‚Read/Write Record (max. 1024 bytes per telegram)’ or ‘Alarms (Process Alarm, Diagnostic Alarm)’can be used. RT over UDP not supported Multicast communication not supported Only one device instance is supported DHCP is not supported RT Class 2 synchronized ('flex') is not supported FastStartUp is implemented in the stack. However some additional hardware limitations apply to use it. Media Redundancy (except MRP client) is not supported Access to the submodule granular status bytes (IOPS & IOCS) is currently not supported if the application accesses the stack using the dual-port memory interface. The amount of configured IO-data influences the minimum cycle time that can be reached. Supervisor-AR is not supported, Supervisor-DA-AR is supported Only 1 Input-CR and 1 Output-CR are supported Multiple WriteRequests are not supported Using little endian (LSB-MSB) byte order for cyclic process data instead of default big endian (MSB-LSB) byte order may have an negative impact on minimum reachable cycle time
Reference to firmware/stack version	V3.4.x.x

Table 142: Technical Data PROFINET IO RT Device Protocol

NIC 52-RE only:

Parameter	Description
Maximum number of cyclic input data	256 bytes if netX Configuration Tool is used for configuration. 1024 bytes if the configuration is done via Modbus RTU (programming effort for the host application program)
Maximum number of cyclic output data	256 bytes if netX Configuration Tool is used for configuration. 1024 bytes if the configuration is done via Modbus RTU (programming effort for the host application program)
Maximum number of modules	Max. 4 input modules and max. 4 output modules can be configured with the netX Configuration Tool. Max. 19 modules if the configuration is done via Modbus RTU (programming effort for the host application program)
Supported protocols	RTC – Real Time Cyclic Protocol, Class 1 and 2 (unsynchronized), Class 3 (synchronized) RTA – Real Time Acyclic Protocol DCP – Discovery and configuration Protocol CL-RPC – Connectionless Remote Procedure Call LLDP – Link Layer Discovery Protocol SNMP – Simple Network Management Protocol MRP – MRP Client
Used Protocols (subset)	UDP, IP, ARP, ICMP (Ping)
Topology recognition	LLDP, SNMP V1, MIB2, physical device
VLAN- and priority tagging	yes
Context Management by CL-RPC	Supported
Identification & Maintenance	Read and write of I&M 1-4
Fast Startup	Supported.
Minimum cycle time	1 ms for RTC1, RTC 2 and RTC3
Baud rate	100 MBit/s
Data transport layer	Ethernet II, IEEE 802.3
Conformance class	Conformance class C
Netload class	Netload class I

Parameter	Description
Limitations	NIC 52-RE can be used in a PROFINET IRT network, however cannot be used with IRT communication due to the internal gateway structure and internal cycle times of the netIC. The netIC gateway is designed for cyclic data exchange. Acyclic communication for user data transfer can only be used if the host application program supports this, which means programming effort for the host application program. Then services ‚Read/Write Record (max. 1024 bytes per telegram)’ or ‘Alarms (Process Alarm, Diagnostic Alarm)’can be used. RT over UDP not supported Multicast communication not supported Only one device instance is supported DHCP is not supported RT Class 2 synchronized ('flex') is not supported Media Redundancy (except MRP client) is not supported Access to the submodule granular status bytes (IOPS & IOCS) is currently not supported if the application accesses the stack using the dual-port memory interface. The amount of configured IO-data influences the minimum cycle time that can be reached. Supervisor-AR is not supported, Supervisor-DA-AR is supported Only 1 Input-CR and 1 Output-CR are supported Multiple WriteRequests are not supported Using little endian (LSB-MSB) byte order for cyclic process data instead of default big endian (MSB-LSB) byte order may have an negative impact on minimum reachable cycle time
Reference to stack version	V2.3 (Stack V3.13)

Table 143: Technical Data PROFINET IO Device Protocol (NIC 52-RE)

20.3.7 Sercos Slave

NIC 50-RE only:

Parameter	Description
Maximum number of cyclic input data (Tx) of all slaves	200 bytes (including Connection Control)
Maximum number of cyclic output data (Rx) of all slaves	200 bytes (including Connection Control)
Maximum number of slave devices	1
Maximum number of applicable Sercos addresses	512 (1 … 511)
Minimum cycle time	250 µs
Topology	Line and ring
Communication phases	NRT, CP0, CP1, CP2, CP3, CP4
Baud rate	100 MBit/s
Data transport layer	Ethernet II, IEEE 802.3
Supported Sercos version	Sercos in the third generation Communication Specification Version 1.1.2
Supported Sercos Communication Profiles	SCP_FixCFG Version 1.1.1 SCP_VarCFG Version 1.1.1 SCP_VarCFG Version 1.1.3
Supported FSP profiles	FSP_IO
SCP_NRT support	No
Identification LED feature supported	yes
Limitations	The netIC gateway is only designed for cyclic data exchange. Max. 2 connections: 1 for consumer and 1 for producer Modifications of the Service-Channel Object Dictionary are volatile after reset (if resides on device) Hot plug is not supported Cross communication not supported NRT Channel is not supported, only forwarding
Reference to firmware version (stack version)	V1.5 (V3.1)

Table 144: Technical Data Sercos Slave Protocol (NIC 50-RE)

NIC 52-RE only:

Parameter	Description
Maximum number of cyclic input data (Tx) of all slaves	250 bytes (including Connection Control)
Maximum number of cyclic output data (Rx) of all slaves	250 bytes (including Connection Control)
Maximum number of slave devices	1
Maximum number of applicable Sercos addresses	512 (1 … 511)
Minimum cycle time	250 µs
Topology	Line and ring
Communication phases	NRT, CP0, CP1, CP2, CP3, CP4
Baud rate	100 MBit/s
Data transport layer	Ethernet II, IEEE 802.3
Supported Sercos version	Sercos in the third generation Communication Specification Version 1.1.2
Supported Sercos Communication Profiles	SCP_FixCFG Version 1.1.1 SCP_VarCFG Version 1.1.1 SCP_VarCFG Version 1.1.3
Supported FSP profiles	FSP_IO
SCP_NRT support	No
Identification LED feature supported	yes
Limitations	The netIC gateway is only designed for cyclic data exchange. Max. 2 connections: 1 for consumer and 1 for producer Modifications of the Service-Channel Object Dictionary are volatile after reset (if resides on device) Hot plug is not supported Cross communication not supported NRT Channel is not supported, only forwarding
Reference to firmware version (stack version)	V2.3 (Stack V3.5)

Table 145: Technical Data Sercos Slave Protocol (NIC 52-RE)

20.3.8 VARAN Client (Slave)

Parameter	Description
Maximum number of cyclic input data	128 bytes
Maximum number of cyclic output data	128 bytes
Memory Area	Read Memory Area 1, Write Memory Area 1, Read Memory Area 2, Write Memory Area 2
Functions	Memory Read Memory Write
Integrated 2 port splitter for daisy chain topology	Supported
Baud rate	100 MBit/s
Data transport layer	Ethernet II, IEEE 802.3
VARAN protocol version	1.1.1.0
Limitations	Integrated EMAC for IP data exchange with client application not supported SPI single commands (optional feature) not supported
Reference to firmware/stack version	V1.0.7.x

Table 146: Technical Data VARAN Client Protocol

20.3.9 CANopen Slave

Parameter	Description
Maximum number of cyclic input data	512 bytes
Maximum number of cyclic output data	512 bytes
Maximum number of receive PDOs	64
Maximum number of transmit PDOs	64
Exchange of process data	Via PDO transfer - synchronized, - remotely requested and - event driven (change of date)
	SDO upload/download Emergency message (producer)
Functions	Node guarding / life guarding, heartbeat PDO mapping NMT Slave SYNC protocol (consumer)
Baud rates	10 kBits/s, 20 kBits/s, 50 kBits/s, 100 kBits/s, 125 kBits/s, 250 kBits/s, 500 kBits/s, 800 kBits/s, 1 MBits/s
Data transport layer	CAN Frames
CAN Frame type	11 Bit
Limitation	The netIC gateway is designed for cyclic data exchange (PDOs). Acyclic communication for user data transfer can only be used if the host application program supports this, which means programming effort for the host application program. Then ‘SDO upload/download’ and ‘Emergency message (producer)’ can be used.
Reference to firmware/stack version	V2.4.x.x

Table 147: Technical Data CANopen Slave Protocol

20.3.10 DeviceNet Slave

Parameter	Description
Maximum number of cyclic input data	255 bytes
Maximum number of cyclic output data	255 bytes
Acyclic communication	Get_Attribute_Single/All Max. 240 bytes per request Set_Attribute_Single/All Max. 240 bytes per request
Connections	Poll Change-of-state Cyclic Bit-strobe
Explicit messaging	Supported
Fragmentation	Explicit and I/O
UCMM	Not supported
Baud rates	125 kBits/s, 250 kBit/s, 500 kBit/s Auto baudrate detection is not supported
Data transport layer	CAN frames
Reference to firmware/stack version	V2.3.x.x

Table 148: Technical Data DeviceNet Slave Protocol

20.3.11 PROFIBUS DP Slave

Parameter	Description
Maximum number of cyclic input data	244 bytes
Maximum number of cyclic output data	244 bytes
Maximum number of acyclic data (read/write)	240 bytes/telegram
Maximum number of modules	Max. 4 input modules and max. 4 output modules can be configured with the netX Configuration Tool.
Configuration data	Max. 244 bytes
Parameter data	237 bytes application specific parameters
Baud rate	9,6 kBits/s, 19,2 kBits/s, 31,25 kBits/s, 45,45 kBits/s 93,75 kBits/s, 187,5 kBits/s, 500 kBits/s, 1, 5 MBits/s, 3 MBits/s, 6 MBits/s, 12 MBit/s Auto baudrate detection is supported
Data transport layer	PROFIBUS FDL
Limitations	The netIC gateway is designed for cyclic data exchange. Acyclic communication for user data transfer can only be used if the host application program supports this, which means programming effort for the host application program. Then services ‘DP V1 Class 1 Read/Write’, ‘DP V1 Class 1 Alarm’ or ‘DP V1 Class 2 Read/Write/Data Transport’ can be used. SSCY1S – Slave to slave communication state machine not implemented Data exchange broadcast not implemented I&M API is not supported I&M0 with fixed settings only
Reference to firmware/stack version	V2.3.x.x

Table 149: Technical Data PROFIBUS DP Slave Protocol

20.3.12 Modbus RTU

Modbus RTU via UART

Parameter	Description and Value Range
Maximum number of input data	999 Registers
Maximum number of output data	994 Registers (951 Registers, if diagnostic is used)
Acyclic communication	Read/Write Register, Maximum 125 Registers per Read Telegram (FC 3, 4), Maximum 120 Registers per Write Telegram (FC 16) Read/Write Coil, Maximum 2000 Coils per Read Telegram (FC 1, 2), Maximum 1968 Coils per Write Telegram (FC 15)
Function Codes Modbus Master	1, 2, 3, 4, 5, 6, 15, 16
Function Codes Modbus Slave	3, 6, 16
Mode	Modbus Master or Modbus Slave
Baud rates	1200 bit/s, 2400 bit/s, 4800 bit/s, 9600 bit/s, 19200 bit/s, 38400 bit/s, 57600 bit/s, 115200 bit/s
Data bits	8 bits
Stop bits	1, 2 bit(s)
Parity	None, even, odd
Limitations	Broadcast not supported
Reference to stack version	V1.4

Table 150: Technical Data Modbus RTU Protocol (UART)

Modbus RTU via SPI

Parameter	Description and Value Range
Maximum number of input data	999 Registers
Maximum number of output data	994 Registers (951 Registers, if diagnostic is used)
Acyclic communication	For NIC 50-XXX and NIC 52-XXX only: Maximum 504 Registers per Read Telegram (FC 3), Maximum 504 Registers per Write Telegram (FC 16), Maximum 504 Registers per Read/Write Telegram (FC 23)
Function Codes Modbus Slave	3, 16, 23
Mode	SPI Slave, mode 3
Reference to stack version	V1.4

Table 151: Technical Data Modbus RTU Protocol (SPI)

21 FAQ

21.1 EtherNet/IP Adapter

21.1.1 State Conflict

When does a State Conflict error occur?

If an EtherNet/IP Scanner wants to change the IP address of the device using Set_attribute_single to class 0xF5, attribute 5, the device can answer with 'State Conflict' (Error code 0x0C). A State Conflict occurs, if the IP address of the device should be changed, but the device is configured to get the IP address via BootP or DHCP.

21.1.2 Warnings during Conformance Test

Warnings "SendUnitData Failed to Receive Data" are reported during the TCP/IP test and during identity test.

These warnings appear especially, if the Conformance Test Tool runs on a Windows 7 PC. It seems internal TCP problems happen on the PC which result that the tool does not receive certain answer telegrams from the DUT (device under test) because the the PC closes the TCP connection to the DUT to early.

21.2 PROFINET IO Device

Accessing index 0x8028 reports the IOCS status "bad" using the AR type "Device Access" (DA-AR) and the certification test reports the test result "Inconclusive".

Submodules are not in the data exchange state if the AR type "Device Access" (DA-AR) is used. The PROFINET specification allows the IOCS state 0x00 (bad) and 0x80 (good) and specifies 0x00 (bad) as preferred state. However, the certification test reports the test result "Inconclusive" although the IOCS state 0x00 (bad) is specified. The PROFINET IO Device reacts conform to the PROFINET specification.

21.3 COMPROX2

Do not connect COMPROX2 via Ethernet TCP/IP to any netIC DIL-32 Communication IC although COMPROX2 offers this connection type, because netIC does not support this. Connect COMPROX2 to netIC via UART only.

22 Conformity Tests

Conformity tests for Real-Time Ethernet systems are performed in order to confirm the conformity of a device to the standard oft he underlying communication system and to ensure the interoperability of all devices in a system even if those come from different vendors.

The netIC has been successfully examined with the following conformity tests:

System	Applied test
PROFINET IO	PROFINET Conformance Test: 2.35.2 (September 2018 test bundle)
EtherCAT	Conformance Test Tool: V2.1.33.0
EtherNet/IP	Conformance Test Tool: CT16
Sercos	Conformizer Version 2.2.10 Sercos revision: V1.1.2

Table 152: Applied conformity tests

22.1 PROFINET IO

The protocol stacks for NIC 52-RE for PROFINET-IO have successfully been tested with the Profinet Conformance Test on conformity of the standards of Conformance Class C and Netload Class I.

The PROFINET IO Conformance Test consists of five parts:

- Test of state machines: All state machines are requested by an automatic simulation software and their reaction is examined in a series of test scenarios.
- Hardware test: The requirements of applicable standards to the Ethernet-Controller oft he PROFINET IO device must be fulfilled concerning correct design of the bus interface, start-up behavior/ fault-free start-up, transmission rate, polarities and crossing of cables, address management, diagnosis and alarm processing and response to standard failures (Power off/on at IO-Device and IO-Controller)
- Interoperability test: An important requirement is that all devices in a system cooperate reliably without any problem, even if they originate from different manufacturers.
- Test of GSD file: The device description file as basis of an error-free configuration is tested along with the device.
- EMC tests: The device is tested with regard to EMC.

22.2 EtherCAT

The EtherCAT protocol stacks with NIC 52-RE have successfully been examined with the „Conformance Test Tool: "CTT-V1i20i80" regarding conformity tot he standards. The test file set "TF-SET_1i20i5" has been applied.

The Conformance Test Tool (CTT) only needs a standard PC with Microsoft Windows, there is no special hardware required. The Ethernet frames for communicating with the device under test are sent via the standard Ethernet port. The tool processes a series of predefined test cases one by one. The tool also tests the electronic description of the EtherCAT device (EtherCAT Slave Information, ESI) and contains an editor for EtherCAT Slave Information (ESI) files. EEPROM data (Slave Information Interface, SII) also can be read, edited and written.

Furthermore, the following topics are checked:

- Consistency of informations from CoE object dictionary, SII and ESI: Checks the validity of an ESI file inclkuding the length of SyncManagers, overlapping of der SyncManager areas etc.
- Plausibility of device description of SII and ESI
    - o Comparison of data from ESI file with the contents of the SII (EEPROM) and the object dictionary of the slave device
    - o Integrity and consistency check oft he SII and validation against the ESI and the object dictionary
- Conformity test of EtherCAT-State-Machine (ESM) against IEC 61158-6-12: The test cases cover all possibilities of state transitions and settings, for instance those of the SyncManager.
- Mailbox communication with SoE and CoE: All available CoE and SoE services have been tested including tests with repeated mailbox requests
- Interoperability tests

The tests have been performed according to the following specifications:

- ETG.7000.2 Conformance Test Record
- ETG.7000.3 Interoperability Test Directive
- ETG.7010 EtherCAT Conformance Guide

The real-time behavior concerning distributed-Clocks (DC) has been tested with a real-time capable master with DC support.

Also see section 24.1.2 “Conformance“.

22.3 EtherNet/IP

The EtherNet/IP protocol stacks with NIC 52-RE have successfully been tested with the EtherNet/IP™ Conformance Composite Test - CT15 regarding conformity to the standard.

The EtherNet/IP TM Conformance Composite Test - CT15 consists of the following parts:

- Protocol Conformance Test
- Test of physical layer
- EDS File Test
- TCP/IP Interface Object Test/ Object 0xF5 (245)
- Ethernet Link Object Tests/ Object 0xF6 (246)
- Port Scans (Direct connection from PC to device under test)
- QoS Object Tests/ Object 0x48 (72)
- DLR Object Tests / Object 0x47 (71)
- Timesync Object Tests / Object 0x43 (67)
- Test of Address Conflict Detection (ACD)
- Redundant Owner Tests - Target
- Redundant Owner Tests - Scanner
- Connection Configuration Object Tests / Object 0xF3 (243)
- Originator Connection List Object Tests/ Object 0x45 (69)

Additional tests performed with netIC

- TCP/IP Config Test
- ACD Behavior Test
- QoS Object Behavior Test
- DLR Test - Announce-based and Beacon-based Nodes
- Timesync Test

22.4 Sercos

The Sercos protocol stacks with NIC 52-RE have successfully been tested with Conformizer Version 2.3.1 for conformity to Sercos Revision V1.1.2. The Sercos Conformizer is a powerful development, test and certification system which is also used for the official conformity tests of Sercos devices.

The Conformizer consists of a specific Sercos interface card (as communication partner oft he sercos device under test) and a test software running under Windows XP without any real-time extensions. This test software has the following characteristics:

- The GUI is based on Eclipse
- Tests are processed in form of scripts written in the scripting language Ruby
- The following areas are affected by the tests
  - Communication
  - Generic Device Profile (GDP)
  - FSP IO
  - FSP Drive

23 Glossary

10-Base T

Standard for communication on Ethernet over twisted pair lines with RJ45 connectors and a Baud rate of 10 MBit/s (according to the IEEE 802.3 specification).

100-Base TX

Standard for communication on Ethernet over unshielded twisted pair lines with RJ45 connectors and a baud rate of 100 MBit/s according to the IEEE 802. specification

Auto-Crossover

Auto-Crossover is a feature of an interface: An interface with Auto-Crossover capability will automatically detect and correct if the data lines have been exchanged vice versa.

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.

Boot loader

Program loading the firmware into the memory of a device in order to be executed.

CC-Link IE Field Basic

Communication system for Industrial Ethernet designed and developed by Mitsubishi Electric Corporation, Tokyo, Japan, providing CC-Link IE Field with a speed of 100 Mbit/s based on TCP/IP

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.

ComproX

A tool used for loading the firmware into the netIC using the Boot loader. The program is delivered on the product CD of the netIC.

CRC

Cyclic Redundancy Check

A mathematic procedure for calculating checksums based on polynomial division in order to detect data transmission errors. For a more detailed description see the Wikipedia article

(http://en.wikipedia.org/wiki/Cyclic_redundancy_check).

DDF

Device_Description_File.

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

EDS file

A special kind of Device Description File used by Ethernet.

EtherCAT

A communication system for industrial Ethernet designed and developed by Beckhoff Automation GmbH.

Ethernet

A networking technology used both for office and industrial communication via electrical or optical connections. It has been developed and specified by the Intel, DEC and XEROX. It provides data transmission with collision control and allows various protocols. As Ethernet is not necessarily capable for real-time application, various real-time extensions have been developed, see Real-Time Ethernet.

EtherNet/IP

A communication system for industrial Ethernet designed and developed by Rockwell. It partly uses the CIP (Common Industrial Protocol).

Ethernet Powerlink

A communication system for industrial Ethernet designed and developed by B&R. It partly uses CANopen technologies.

FSU

FSU (Fast start-up) is a feature of PROFINET protocol stacks enabling them to start up within only one second. FSU is supported by the new PROFINET V3 protocol stack version 1.2 and higher for NIC50-RE.

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.

GPIO

General Periphery Input Out

Signal at pin 26 of the netIC. In Modbus RTU/SPI mode used as SPI Chip Select signal.

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.

I²C

I²C means Inter-Integrated Circuit. I²C is a serial data bus system developed by Philips Semiconductors. It makes use of the master-slave-principle. It can be used to connect devices with a low transmission rate to a system. In I²C systems two I/O pins are sufficient to control an entire network.

Industrial Ethernet

See Real-Time Ethernet

Modbus Data Model

The data model distinguishes four basic types of data areas:

- Discrete Inputs (inputs) = FC 2 (Read)
- coils (outputs) = FC 1, 5, 15 (Write and Read back)
- Input register (input data) = FC 4 (Read)
- Holding register (output data) = FC 3, 6, 16, 23 (Write and Read back).

It should be noted, however, that depending on the device manufacturer and device type:

- the data area in the device may be present or not,
- and two data areas can be combined into one data region. For example, discrete inputs and input registers can be a common data area, which can be accessed with read-FC 2 and FC 4.
- Further FC 1 and FC 3 are used instead of reading back the inputs to read the outputs.

Modbus RTU

A standard for serial communication developed by Schneider Automation that is used for communication of the host with the NIC 50-RE. It uses the Modbus Data Model.

netX

networX on chip, next generation of communication controllers.

netX Configuration Tool

The netX Configuration Tool allows users to operate cifX or netX-based devices in different networks. Its graphical user interface serves as a configuration tool for the installation, configuration and Diagnostic of the devices.

Object Dictionary

An object dictionary is a storage area for device parameter data structures. It is accessed in standardized manner.

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.

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.

RJ45

A connector type often used for Ethernet connection. It has been standardized by the Federal Communications Commission of the USA (FCC).

RoHS

Restriction of Hazardous Substances

This abbreviation denominates the directive of the European Union on the use of 6 hazardous substances in electronic products. It is titled *“Directive on the restriction of the use of certain hazardous substances in electrical and electronic equipment 2002/95/EC”,* adopted in 2003 and was getting effective on 1 July 2006.

RS232

An interfacing standard for serial communication on data lines defined by EIA (Electronic Industries Alliance) in *ANSI/EIA/TIA-232-F-1997.*

RS422

An interfacing standard for differential serial communication on data lines defined by EIA (Electronic Industries Alliance) in *ANSI/TIA/EIA-422-B-1994*.

RS485

An interfacing standard for differential serial communication on data lines defined by EIA (Electronic Industries Alliance) in *ANSI/TIA/EIA-485-A-1998*

SC-RJ

An industry standard for connectivity for optical data connections developed by Reichle & De Massari AG, Switzerland

Sercos

A communication system for industrial Ethernet designed and developed by Bosch-Rexroth GmbH and supported by SERCOS International.

Shift register

A digital electronic circuit for converting serial data to parallel data (vice versa also possible) based on the FIFO principle (first in first out). Each time a new bit of the serial data stream arrives at the shift register (this should happen within a fixed cycle time), it is stored in the first flip-flop of the shift register and the contents of each flip-flop is shifted to the next flip-flop.

SPI

SPI means Serial Peripheral Interface. SPI is a bus system for a synchronous serial data bus which has been developed by Motorola. SPI makes use of the master-slave-principle. It requires at least three lines used for data input, data output and clock and works in full duplex mode.

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.

Transceiver

A combined receiver and transmitter unit for communication over optical Ethernet

UART

UART means Universal Asynchronous Receiver Transmitter. It is a special kind of electronic circuit which is used for transmitting data serially with a fixed frame consisting of one start bit, five to nine data bits, an optional parity bit for the detection of transmission errors and one stop bit. Working asynchronously, it does not use an explicit clock signal.

VARAN

Versatile Automation Random Access Network

A communication system for industrial Ethernet designed and developed by SIGMATEK.

Warmstart

A part of the initialization process of netX-controlled communication systems. During warmstart the netX-controlled system is adjusted to the intended parameters of operation. These parameters are supplied by a special message, the warmstart message which is transferred to the _netX within the warmstart packet.

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.

24 Appendix

24.1 EtherCAT Summary over Vendor ID, Conformance Test, Membership and Network Logo

24.1.1 Vendor ID

The communication interface product is shipped with Hilscher's secondary vendor ID, which has to be replaced by the Vendor ID of the company shipping end products with the integrated communication interface. End Users or Integrators may use the communication interface product without further modification if they re-distribute the interface product (e.g. PCI Interface card products) only as part of a machine or machine line or as spare part for such a machine. In case of questions, contact Hilscher and/or your nearest ETG representative. The ETG Vendor-ID policies apply.

24.1.2 Conformance

EtherCAT Devices have to conform to the EtherCAT specifications. The EtherCAT Conformance Test Policies apply, which can be obtained from the EtherCAT Technology Group (ETG, www.ethercat.org).

Hilscher range of embedded network interface products are conformance tested for network compliance. This simplifies conformance testing of the end product and can be used as a reference for the end product as a statement of network conformance (when used with standard operational settings). It must however be clearly stated in the product documentation that this applies to the network interface and not to the complete product.

Conformance Certificates can be obtained by passing the conformance test in an official EtherCAT Conformance Test lab. Conformance Certificates are not mandatory, but may be required by the end user.

24.1.3 Certified Product vs. Certified Network Interface

The EtherCAT implementation may in certain cases allow one to modify the behavior of the EtherCAT network interface device in ways which are not in line with EtherCAT conformance requirements. For example, certain communication parameters are set by a software stack, in which case the actual software implementation in the device application determines whether or not the network interface can pass the EtherCAT conformance test. In such cases, conformance test of the end product must be passed to ensure that the implementation does not affect network compliance.

Generally, implementations of this kind require in-depth knowledge in the operating fundamentals of EtherCAT. To find out whether or not a certain type of implementation can pass conformance testing and requires such testing, contact EtherCAT Technology Group (“ETG”, www.ethercat.org) and/or your nearest EtherCAT conformance test centre. EtherCAT may allow the combination of an untested end product with a conformant network interface. Although this may in some cases make it possible to sell the end product without having to perform network conformance tests, this approach is generally not endorsed by Hilscher. In case of questions, contact Hilscher and/or your nearest ETG representative.

24.1.4 Membership and Network Logo

Generally, membership in the network organization and a valid Vendor-ID are prerequisites in order to be able to test the end product for conformance. This also applies to the use of the EtherCAT name and logo, which is covered by the ETG marking rules.

Vendor ID Policy accepted by ETG Board of Directors, November 5, 2008

24.2 Use of VARAN Client

In order to use the netIC Communication IC with VARAN, you need a license which you can acquire at the VNO (VARAN Bus-Nutzerorganisation, Bürmooser Straße 10, A-5112 Lamprechtshausen, info@varan-bus.net) after getting a member of VON.

The license as well as the Vendor ID and the Device ID can be adjusted with the SYCON.net configuration software or with the netX Configuration Tool.

24.3 Change in the Use of DHCP and Default IP Address in the EtherNet/IP Firmware

The following change applies to all Firmware versions for EtherNet/IP beginning with 1.4.16.x: Change: If DHCP is set to get an IP address, then DHCP is used constantly.

Behavior before change: If DHCP is set to get to get the IP address parameter, then after 3 tries without success a default IP address was used.

Now: DHCP is used constantly to get an IP address. There is no default IP address any more.

24.4 Device Drawings

24.4.1 Device Drawing NIC 50-RE with Heat Sink

The following drawing shows the dimension of the NIC 50-RE with the original Hilscher heat sink. Moreover there is a NIC 50-RE version without the original Hilscher heat sink (NIC 50-RE/NHS), which is described in the next subsection.

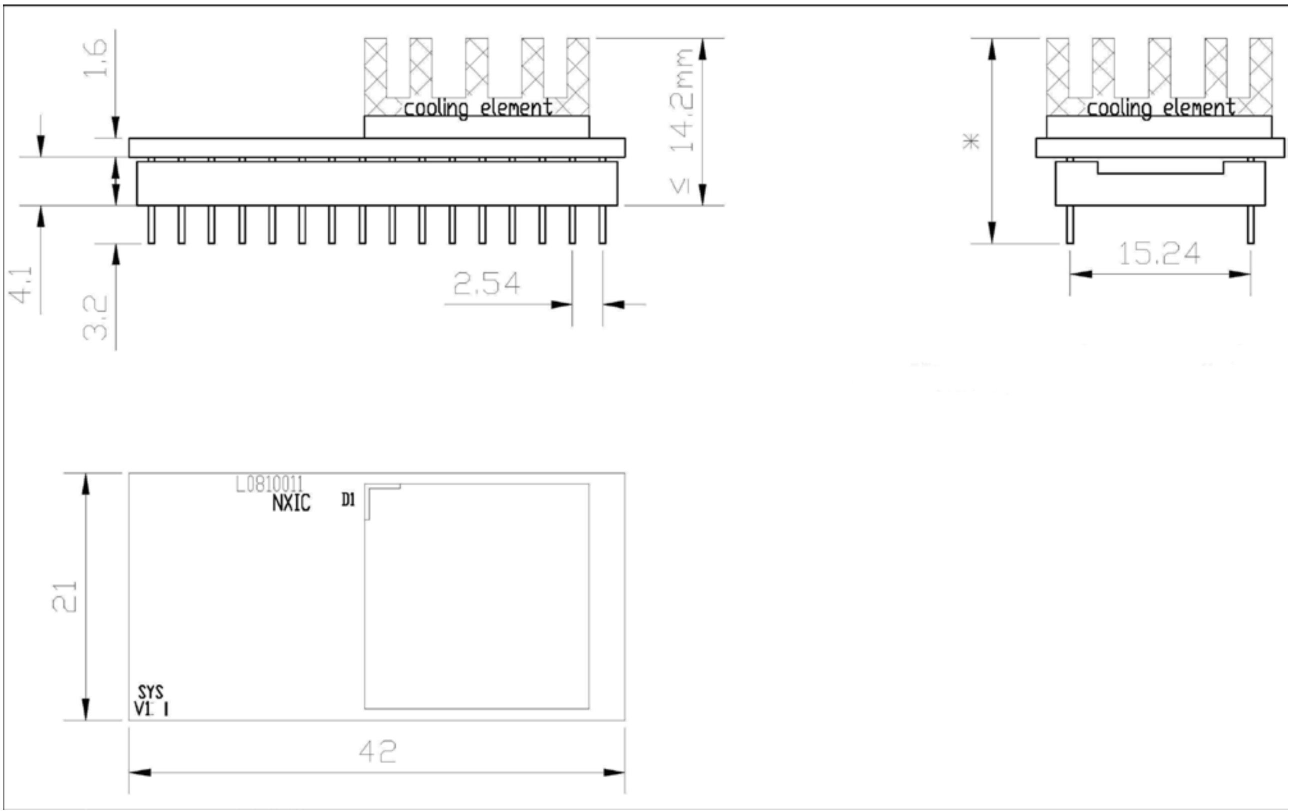


Figure 70: Device Drawing NIC 50-RE

The heat sink of the NIC 50-RE is on the opposite side of the module compared to the netIC Fieldbus DIL-32 Communication IC modules, see position of the netX chip in *Table 12: Position of the tag of the netIC devices* on page 34.

24.4.2 Device Drawing NIC 50-RE/NHS without Heat Sink and PCB Thermal Pad

Moreover there is a NIC 50-RE version without the original Hilscher heat sink (NIC 50-RE/NHS). When the NIC 50-RE/NHS is used it is necessary to apply a PCB heat sink instead of the original heat sink.

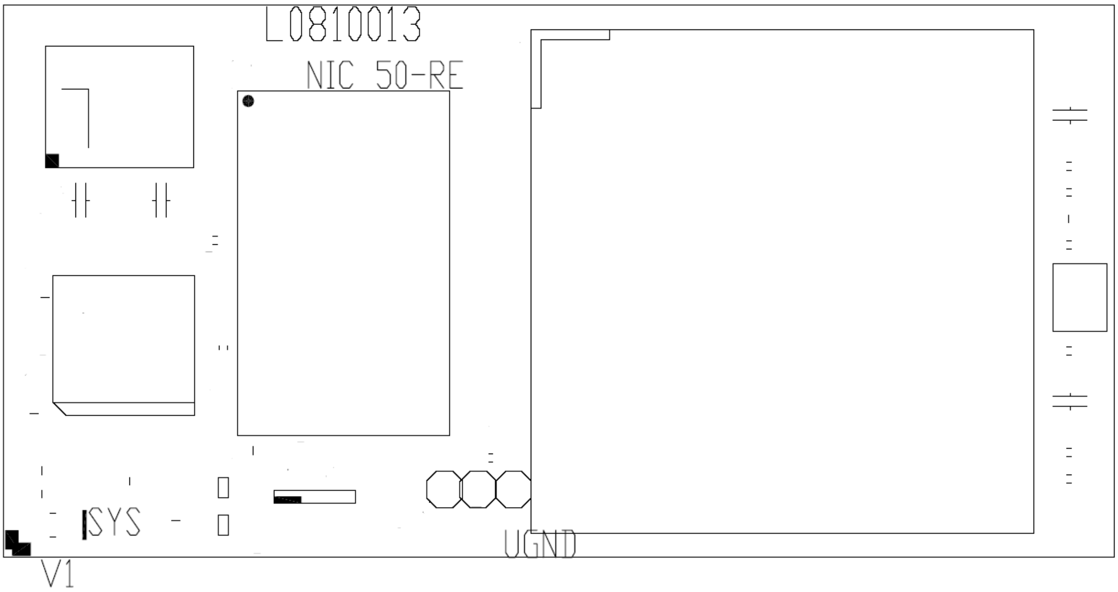


Figure 71: Device Drawing NIC 50-RE/NHS without Heat Sink

Special Design Rules for optimal Cooling Conditions when using the NIC 50-RE/NHS without original Hilscher Heat sink over a PCB Heat sink

- Usage of special thermal pad to contact directly to the heat sink of the PCB (Thickness 0,5 mm, compressible)
- On both sides of the PCB there must be a copper heat sink area of approx. 900 mm² (20 mm x 45 mm) with a thickness of at least 35µm.
- The heat sink area, where the netX is to be mounted, has to be metalized with Ni-Au .
- The other area of the heat sink with no contact to the netX has to be coated with standard PCB solder resist. A metalized area is not allowed there
- In the heat sink area of the netX chip there must be inserted thermal vias with 0.25 mm hole diameter (corresponding to 0.3 mm drill size) and 1 mm distance between each of the vias. So it is possible to insert 18 x 18 = 324 thermal vias.
- A standard copper thickness of the via sleeve is 25 µm.

The following picture shows the mounting of the NIC 50-RE without Hilscher's original heat sink and over a PCB heat sink.

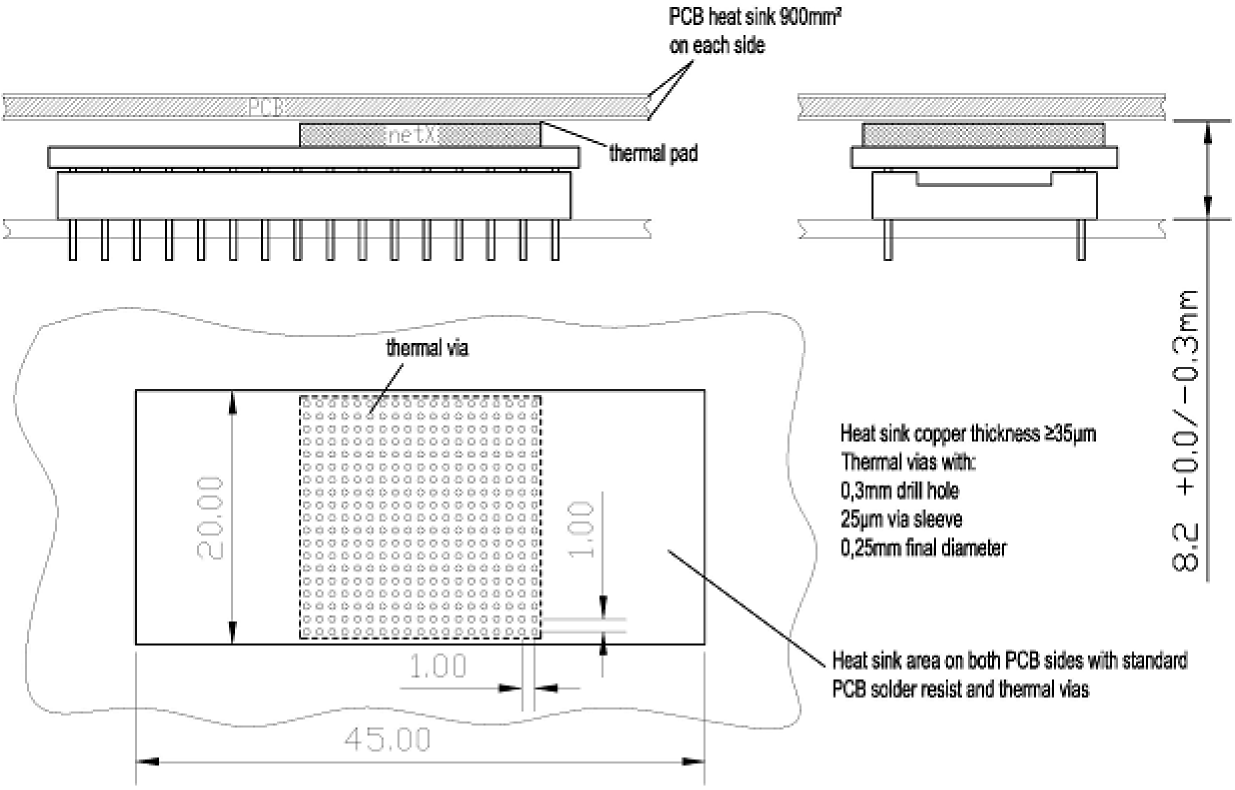


Figure 72: Device Drawing NIC 50-RE without Hilscher's original heat sink and over a PCB heat sink

It is also possible to use another heat sink. Then the thermal pad can be demounted and another heat sink can be glued on the netIC module. To reach the maximum operation temperature of +70°C it is necessary that the thermal resistance R_{th} of the used heat sink is less than 7 K/W.

24.4.3 Device Drawing NIC 50-COS

The drawing in *Figure 73: Device Drawing NIC 50-COS (top side)* below on this page displays the front side NIC 50-COS without any heat sink.

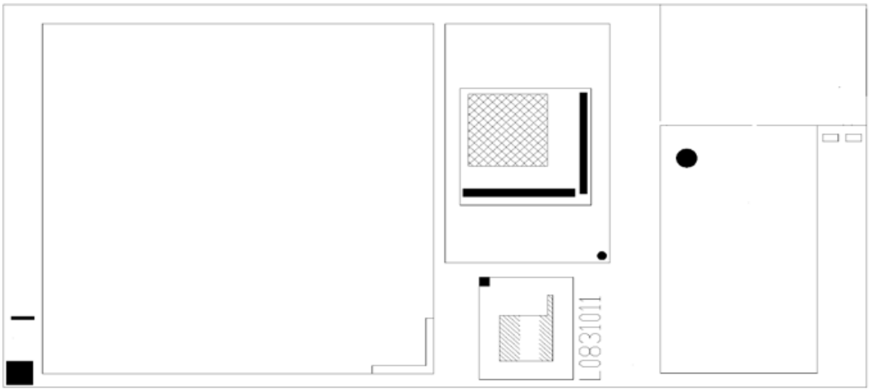


Figure 73: Device Drawing NIC 50-COS (top side)

24.4.4 Device Drawing NIC 50-DNS

The drawing in *Figure 74: Device Drawing NIC 50-DNS (top side)* below on this page displays the front side NIC 50-DNS.

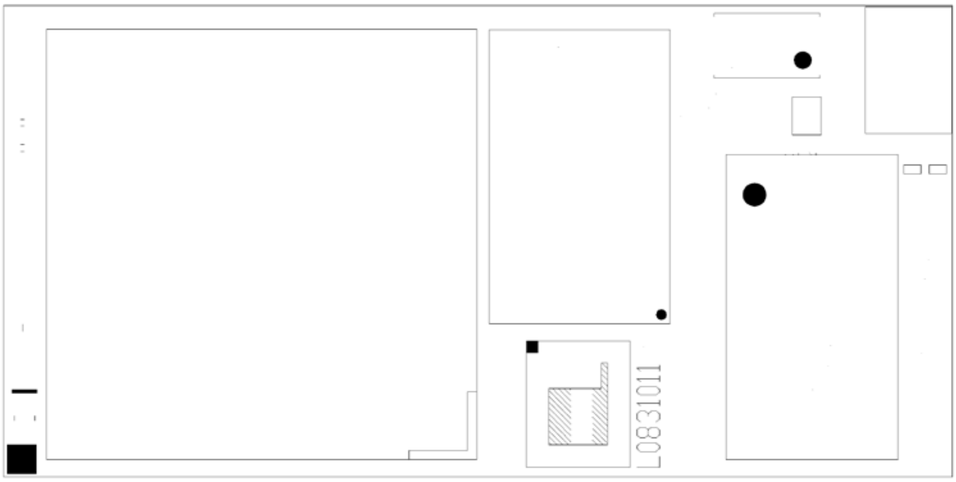


Figure 74: Device Drawing NIC 50-DNS (top side)

24.4.5 Device Drawing NIC 50-DPS

The drawing in *Figure 75: Device Drawing NIC 50-DPS (top side)* below on this page displays the front side NIC 50-DPS.

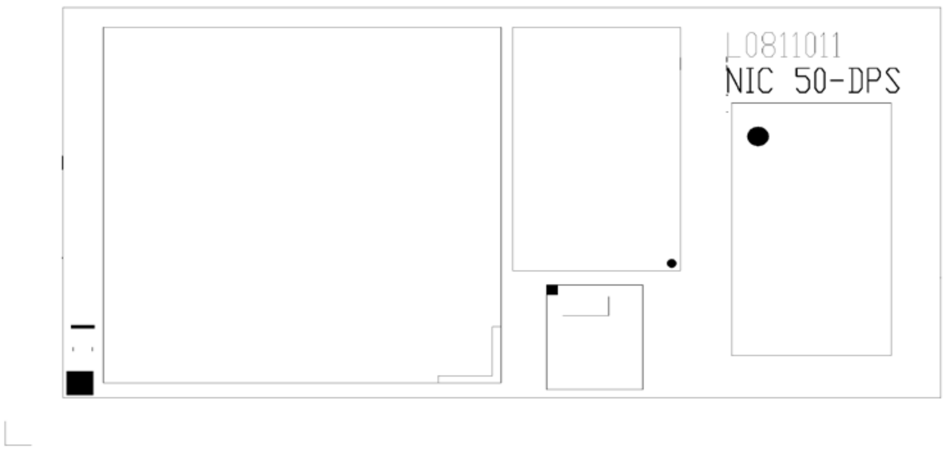


Figure 75: Device Drawing NIC 50-DPS (top side)

24.4.6 Device Drawing NIC 52-RE

The following drawing shows some of the dimensions of the NIC 50-RE:

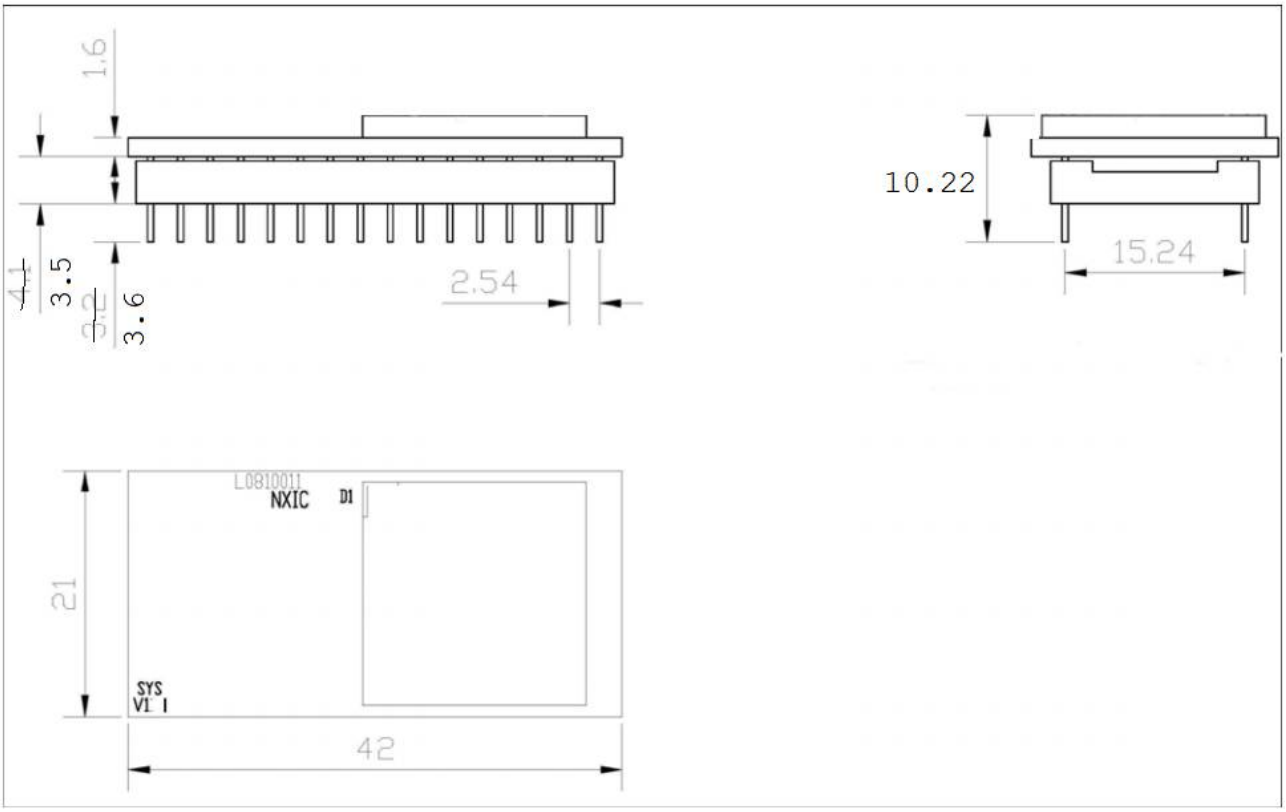


Figure 76: Dimensioned Device Drawing NIC 52-RE (top side)

The drawing in the figure below displays the front side NIC 52-RE without any heat sink.

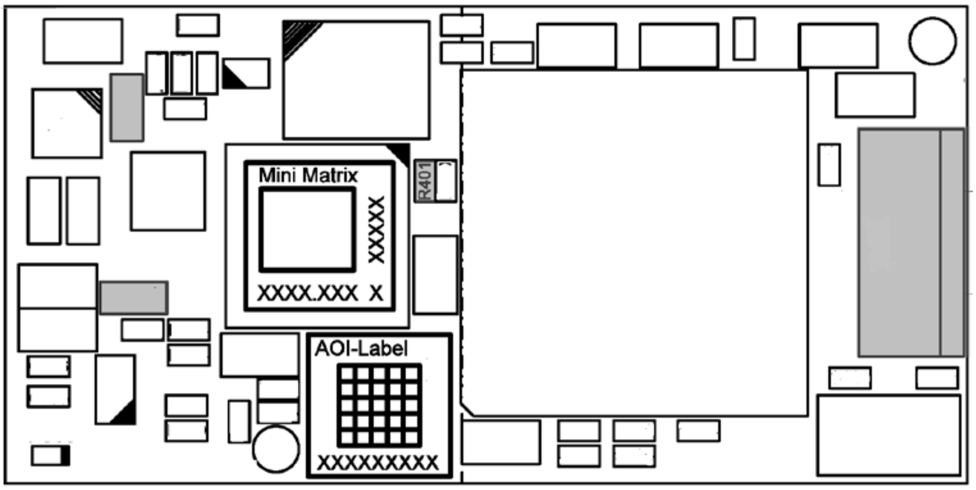


Figure 77: Dimensioned Device Drawing NIC 52-RE (top side)

24.5 Use of Hubs and Switches

For the corresponding communication systems, the use of hubs and/or switches is either forbidden or allowed. The following table shows the acceptable use of hubs and switches by each communication system:

Communication System	Hub	Switch
EtherCAT	**forbidden**	**only allowed between EtherCAT Master and first EtherCAT Slave (100 MBit/s, Full Duplex)**
EtherNet/IP	**allowed**	**allowed (10 MBit/s/100 MBit/s, Full or Half Duplex, Auto-Negotiation)**
Open Modbus/TCP	**allowed**	**allowed (10 MBit/s/100 MBit/s, Full or Half Duplex, Auto-Negotiation)**
POWERLINK	**allowed**	**forbidden**
PROFINET IO RT	**forbidden**	**Only allowed, if the switch supports ‚Priority Tagging' and LLDP (100 MBit/s, Full Duplex)**
Sercos	**forbidden**	**forbidden**
VARAN	**forbidden**	**forbidden**

Table 153: Use of Hubs and Switches

Important: Failure of Network Communication on older netX Processors at certain Conditions

- If you intend to operate the NIC 50-RE with 10 MBit/s in Half-Duplex Mode (only possible with Ethernet/IP or Open Modbus/TCP firmwares), see section "*Failure in 10 MBit/s Half Duplex Mode and Workaround*" on page 246.

24.6 Failure in 10 MBit/s Half Duplex Mode and Workaround

Important: The failure described here only affects older **NIC 50-RE** DIL-32 Communication ICs with serial numbers up to **22104.**

Affected Hardware

Hardware with the communication controller netX 50, netX 100 or netX 500; netX/Internal PHYs.

When can this Failure occur?

When using standard Ethernet communication with 10 MBit/s half duplex mode, the PHY gets stuck in case of network collisions. Then no further network communication is possible. Only device power cycling allows Ethernet communication again.

This problem can only occur with Ethernet TCP/UDP IP, EtherNet/IP or Modbus TCP protocols when using hubs at 10 MBit/s. The issue described above is not applicable for protocols which use 100 MBit/s or full duplex mode.

Solution / Workaround:

Do not use 10 MBit/s-only hubs. Use either switches or 10/100 MBit/s Dual Speed hubs, to make sure the netX Ethernet ports are connected with 100 MBit/s or in full duplex mode.

This erratum is fixed with all components of the ‘Y’ charge (9 digit charge number shows ‘Y’ at position 5 (nnnnYnnnn).

Reference

“Summary of 10BT problem on EthernetPHY”,
RenesasElectronics Europe, April 27, 2010

24.7 List of Figures

24.8 List of Tables

24.9 Contacts

Headquarters

Germany
Hilscher Gesellschaft für Systemautomation mbH
Rheinstraße 15
D-65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-mail: info@hilscher.com

Support
Phone: +49 (0) 6190 9907-990
E-mail: hotline@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.
69800 Saint Priest
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, Bangalore
Phone: +91 8888 750 777
E-mail: info@hilscher.in

Support
Phone: +91 8108884011
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

Republic of Korea
Hilscher Korea Inc.
13494, Seongnam, Gyeonggi
Phone: +82 (0) 31-739-8361
E-mail: info@hilscher.kr

Support
Phone: +82 (0) 31-739-8363
E-mail: kr.support@hilscher.com

Austria
Hilscher Austria GmbH
4020 Linz
Phone: +43 732 931 675-0
E-mail: sales.at@hilscher.com

Support
Phone: +43 732 931 675-0
E-mail: at.support@hilscher.com

Switzerland
Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-mail: info@hilscher.ch

Support
Phone: +41 (0) 32 623 6633
E-mail: support.swiss@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