Application Note

netX Design-In Guide

netX 100/500

Hilscher Gesellschaft für Systemautomation mbH

www.hilscher.com

DOC081106AN02EN | Revision 2 | English | 2012-10 | Released | Public

Table of Contents

1 Introduction 4
1.1 About this Document 4
1.2 Legal Notes 5
1.2.1 Copyright 5
1.2.2 Important Notes 5
1.2.3 Exclusion of Liability 5
1.2.4 Warranty 6
1.2.5 Export Regulation 6

2 netX100/500 Quick Start 7
2.1 Basic Circuit 7
2.2 Ethernet Interface Circuits 25
2.2.1 Twisted Pair Two Channel 25
2.2.2 Twisted Pair Single Channel 27
2.2.3 Fiber Optic with AFBR-5978Z 29
2.3 AS-Interface 33
2.4 CANopen 35
2.5 CC-Link 37
2.6 CompoNet 39
2.7 DeviceNet 41
2.8 PROFIBUS 43
2.9 MMC/SD-Card SPI-Circuit 45
2.10 UART ⇔ RS232 47
2.11 USB Device Mode 49
2.12 USB Host Mode 51
2.13 Embedded Trace Macrocell (ETM) 53
2.14 LCD Interface 55

3 Resource Overview 59
3.1 rcX Operating System 59
3.2 Third Party Operating Systems 60
3.3 Memory Requirements of Hilscher Stacks 61

4 Standard Circuits 62
4.1 RDY/RUN Pins, SYS LED 62
4.2 Secure EEPROM 65
4.3 Crystals, Clock generators 66
4.3.1 System Clock 66
4.3.2 Real Time Clock 67
4.4 Power On Reset and Reset In 69
4.5 Debug and Test Interfaces 70
4.5.1 JTAG Interface 70
4.5.2 ETM Interface 71
4.5.3 Boundary Scan 72
4.6 External Memory 74
4.6.1 FLASH Memory 74
4.6.2 SDRAM 81
4.7 Host Interface 84
4.7.1 DPM Mode 85
4.7.2 Extension Bus Mode 93
4.7.3 Multiplex Mode 94
4.7.4 External pull-ups/pull-downs, unused signals 96
4.8 UARTs 97
4.9 USB 98
4.9.1 Device Mode 98
4.9.2 Host Mode 100
4.10 Ethernet Interface 102
4.10.1 Twisted Pair 102
4.10.2 Fiber Optic 105
4.10.3 Ethernet PHYs unused 113
4.10.4 Ethernet Status LEDs 114
4.10.5 Real Time Ethernet 115

4.11 Fieldbus Interface 118
4.11.1 AS interface Master 120
4.11.2 CANopen Interface 121
4.11.3 CC-Link Interface 122
4.11.4 CompoNet Interface 123
4.11.5 DeviceNet Interface 124
4.11.6 PROFIBUS Interface 125
4.11.7 Fieldbus Status LEDs 126
4.12 A/D Converter 127
4.13 PWM Interface 130
4.14 Encoder Interface 132
4.15 LCD Interface 134
4.16 Touch Panel Interface 135
4.17 PIO 137
4.18 Power Supply 138
4.18.1 Core Voltage Regulator 138
4.18.2 Alternative Core Voltage Regulator 139
4.18.3 Common Supply Voltage Regulator 3.3 V 139
4.18.4 Voltage Regulator 5 V 140

5 General Design Considerations 141
5.1 Thermal Behavior 141
5.1.1 Basics 141
5.1.2 Estimates 141
5.1.3 Rules of thumb 143
5.2 EMC behavior 144
5.2.1 Layer Stack 144
5.2.2 Decoupling capacitors 145
5.2.3 Power Supply Input Filter 146
5.2.4 Reset Lines 146
5.2.5 Clock Circuits 147
5.2.6 Ethernet Interface 147
5.2.7 Memory Bus 149
5.2.8 Planes 149
5.3 Vias and Traces under the netX100/500 151

6 Reference Section 152
6.1 Crystals 152
6.2 Memory Components 152
6.3 List of Tables 155
6.4 List of Figures 156

7 Contacts 158

1 Introduction

1.1 About this Document

This document is directed to hardware developers creating a hardware design with a communication controller of the Hilscher netX family. It does not explain netX technology and features, which is covered by the corresponding Technical Data Reference Guides.

It describes the standard circuitry around all netX interfaces like memory interface (SDRAM, FLASH) USB, UARTs, XMACs (Ethernet and field bus), LCD, as well as power supply, reset and clock circuits along with the standard netX I/O resources (PIOs, GPIOs) that have been assigned a default functionality at Hilscher. This includes Status LEDs, control signals and sync signals for RTE applications

Although the system designer is basically completely free to select any available I/Os for I/O purposes, it is important that he complies with the standard port definitions, whenever loadable Firmware from Hilscher is to be used, as this kind of firmware necessarily assumes compliance with Hilscher standard assignments.

Note: Designers should be aware, that not all components supported by netX hardware (e.g. parallel FLASH) are necessarily also supported by existing software / firmware or tools from Hilscher! Hence it is strongly recommended, to consult the feature table in the following chapter, to make sure that all desired hardware features of the planned design are eventually supported by the firmware that will run on the design. This applies not only, but particularly to customers planning to use loadable firmware from Hilscher instead of doing own firmware development.

Resources that are currently not supported by loadable Firmware or where no drivers / code are yet available may still already be supported by existing Hilscher devices (e.g. Gateways). It is hence recommended to check with Hilscher Sales, if there is already an existing solution for your problem. Further, Hilscher offers several custom design services for netX hard- and software, as well as manufacturing services, providing an easy way to your custom product. For detailed information and quotes, please contact Hilscher Sales.

Hilscher also offers a schematic review service, allowing your hardware design to be checked by netX experienced hardware engineers. Hilscher Sales will be happy to provide an individual quote for this service, after receiving your schematics (PDF format).

Note: Before starting a design, it is strongly recommended, to consult the latest Errata Sheets (available on the Hilscher website www.hilscher.com) of the netX controllers!

1.2 Legal Notes

1.2.1 Copyright

1.2.2 Important Notes

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

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

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

1.2.3 Exclusion of Liability

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

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

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

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

1.2.4 Warranty

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

1.2.5 Export Regulation

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

2 netX100/500 Quick Start

2.1 Basic Circuit

This chapter describes the important basic parts of a netX100/500 circuit to use Hilscher loadable firmware. Some interfaces have a minimum circuit that have to be connected. Therefore, all interfaces circuit in this chapter shows the connection if they are unused. The following chapter described this interface circuits.

RUN/RDY

Let's start with the circuit on RDY/RUN pins. These pins operate as input after reset. The first stage boot loader of the netX100/500 checks the logic level and enters certain boot modes. One important boot mode is the serial boot mode which will allow to (re-) flash the firmware over USB or RS232. To get the netX100/500 in this boot mode, the RDY pin has to be pulled-down to GND over a 1.27 kΩ resistor. A push button or pin header between the resistor and the RDY pin make it possible to activate this boot mode when it is necessary.

For displaying system status, a yellow and green dual LED is recommended to connect antiparallel over to 220 Ω resistors to RDY and RUN pin.

Two 15 kΩ pull-up resistors connected to 3.3 V stabilize the RDY and RUN lines in resting level.

The following picture shows the basic RDY/RUN circuit for netX100/500.

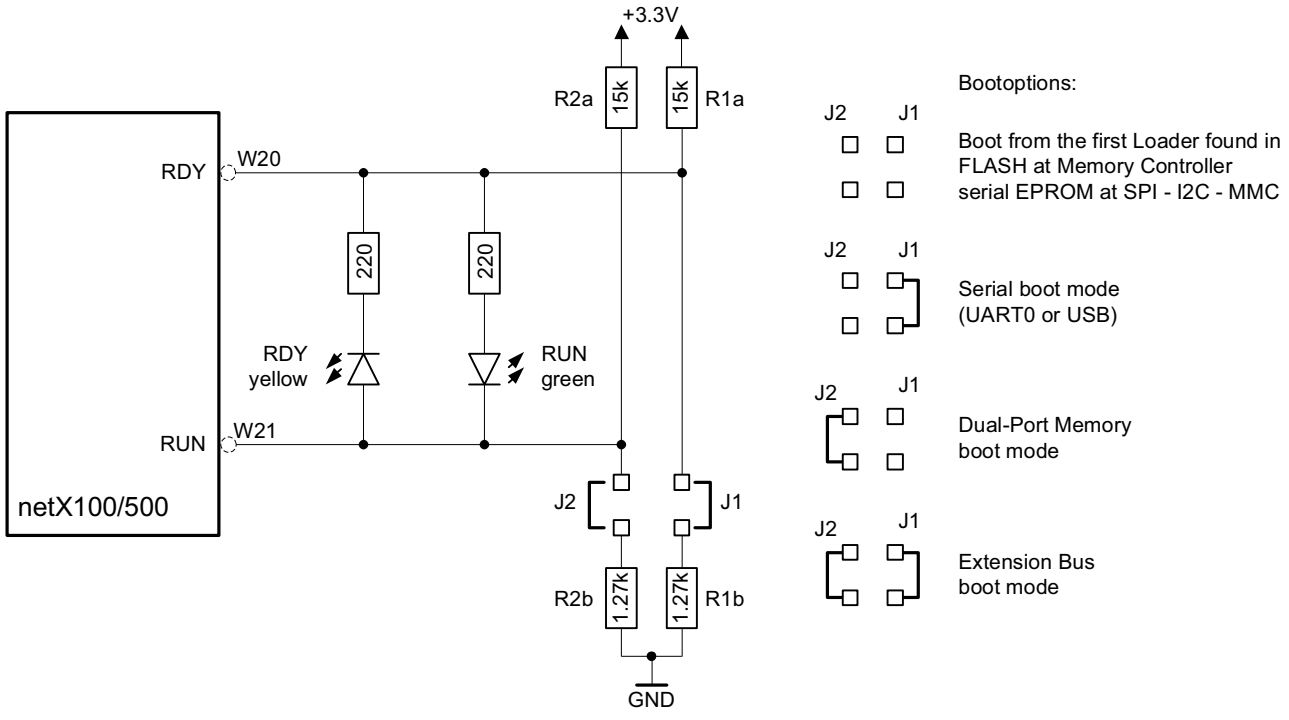


Figure 1: netX100/500 RDY/RUN Basic Circuit

More about RDY/RUN circuit can be found on:

→ Page 62 chapter 4.1 RDY/RUN Pins, SYS LED

SPI Flash Memory

The Hilscher loadable firmware is always stored in flash memory, which is connected via SPI with the netX100/500.

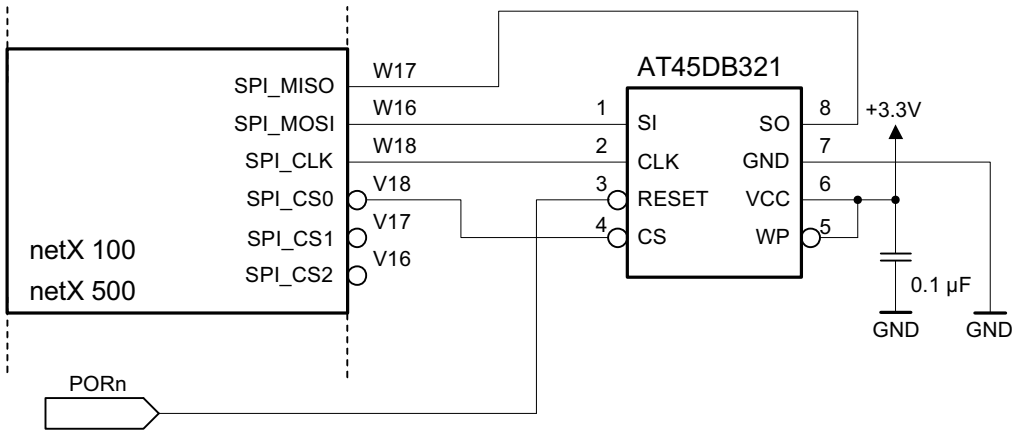


Figure 2: netX100/500 SPI Flash

More about SPI, Flash Memory, MMC/SD-Card circuit can be found on:

→ Page 45 chapter 2.9 MMC/SD-Card SPI-Circuit

→ Page 61 chapter 3.3 Memory Requirements of Hilscher Stacks

→ Page 74 chapter 4.6.1.1 SPI FLASH

→ Page 75 chapter 4.6.1.2 MMC/SD Card

→ Page 76 chapter 4.6.1.3 Parallel FLASH

Secure EEPROM

The Secure EEPROM connected via I²C-Bus with netX100/500 hold licensing information, MAC address and other information.

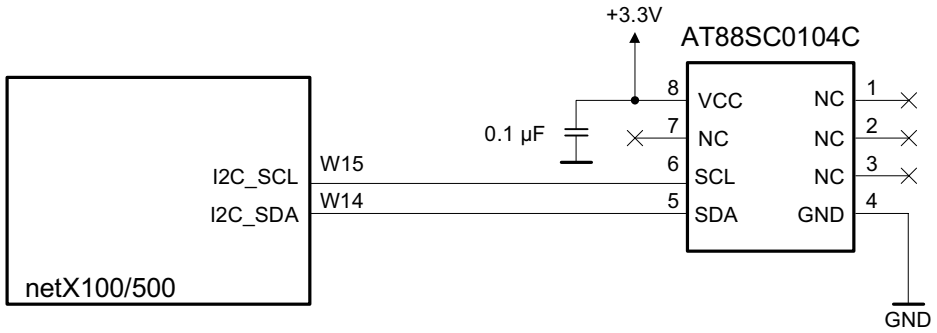


Figure 3: netX100/500 Secure Memory Basic Circuit

More about Secure EEPROM and I²C-Bus circuit can be found on:

→ Page 65 chapter 4.2 Secure EEPROM

Base Clock

The netX100/500 uses an internal oscillator along with an external crystal for generating 25 MHz base clock.

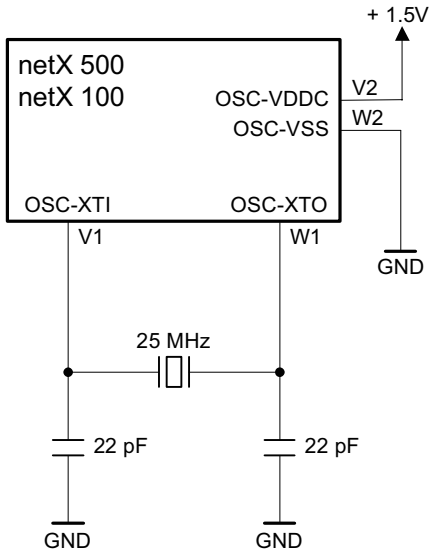


Figure 4: netX100/500 System Oscillator Circuit

More about base clock circuit can be found on:

→ Page 66 Chapter 4.3 Crystals, Clock generators

Real Time Clock

The netX500 real time clock is not necessary for the basic circuit and has to be connected like the netX100 Pins to 1.5 V and 3.3 V.

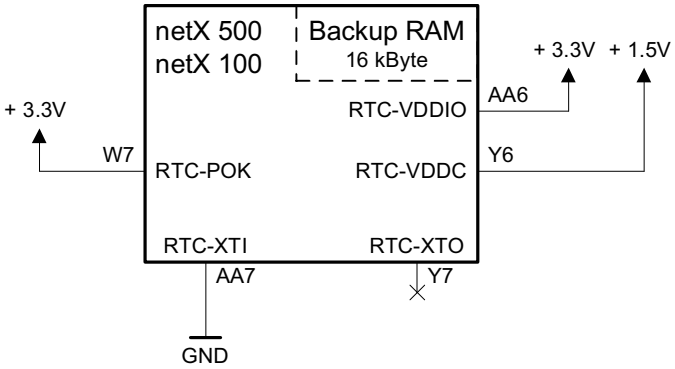


Figure 5: netX100/500 RTC Not Used in Basic Circuit

More about real time clock circuit can be found on:

→ Page 66 Chapter 4.3 Crystals, Clock generators

Reset

The netX100 and netX500 provide two inputs for reset signals, the Power On Reset (PORn) and the Reset In (RSTINn). While the use of the RSTINn is optional, the Power On Reset is mandatory.

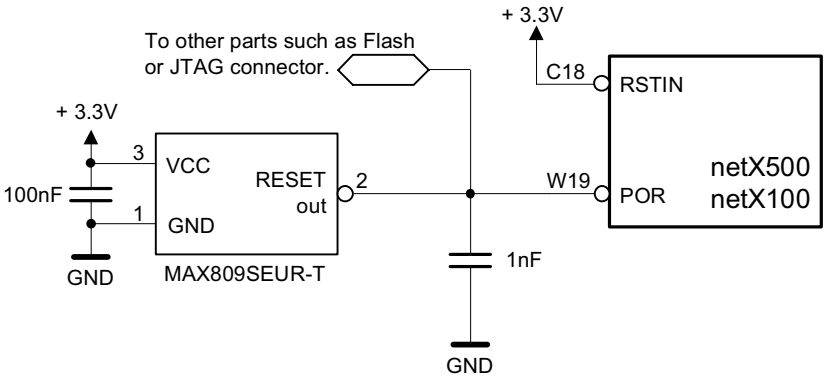


Figure 6: netX100/500 Reset Circuit

More about reset circuit can be found on:

→ Page 69 Chapter 4.4 Power On Reset and Reset In

USB

The device mode is the commonly used mode of the netX USB interface and allows to connect the netX to a PC (in serial boot mode), which can then download and flash firmware, read and modify register values and run hardware test applications by the help of freely available software tools from Hilscher.

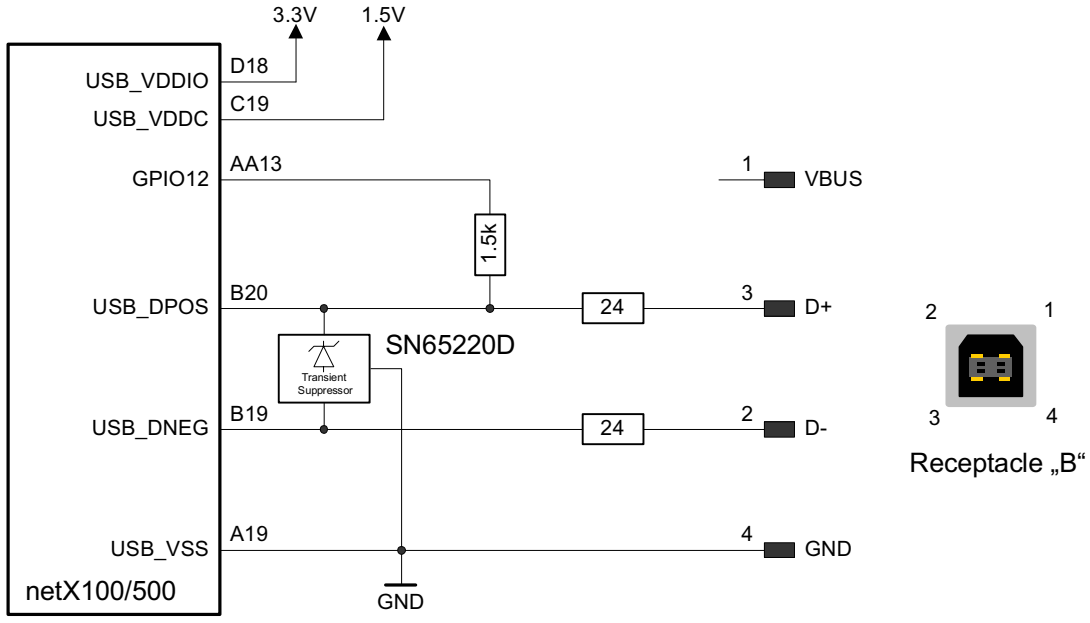


Figure 7: netX100/500 USB DOWN Stream Port (Device Mode)

More about USB circuit can be found on:

→ Page 69 Chapter 4.4 Power On Reset and Reset In

UART0

For using the serial boot mode on UBS port with unconnected UART0, the pin AA19 (UART0_RXD) and pin AA18 (UART0_CTS#) have to be connected with 10 kΩ pull-up resistors to +3.3 V.

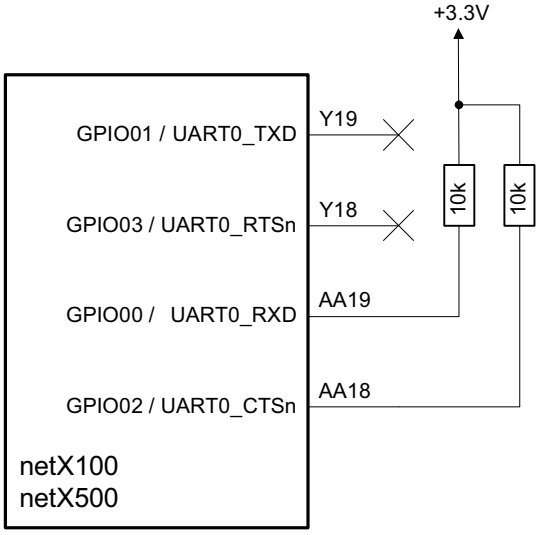


Figure 8: netX100/500 UART0 Unused

More about UART circuits can be found on:

→ Page 97 Chapter 4.8 UARTs

AD-Converter

If the ADC is not used, it must still be connected to the power supply

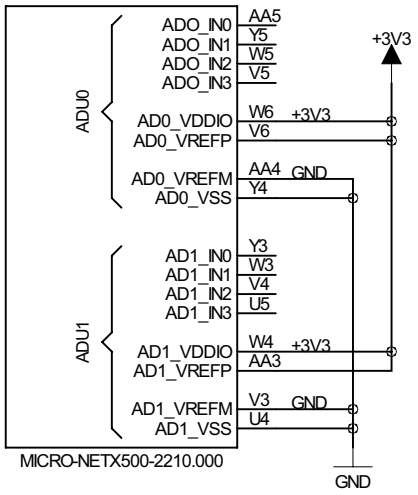


Figure 9: Unused ADC Basic Circuit

More about ADC-Converter circuits can be found on:

→ Page 127 chapter 4.12 A/D Converter

SDRAM

The standard Hilscher loadable firmware uses 8 MB SDRAM. The following figure shows the connection of ISSI SDRAM with 86 pin TSOP package (IS42S32200C1)

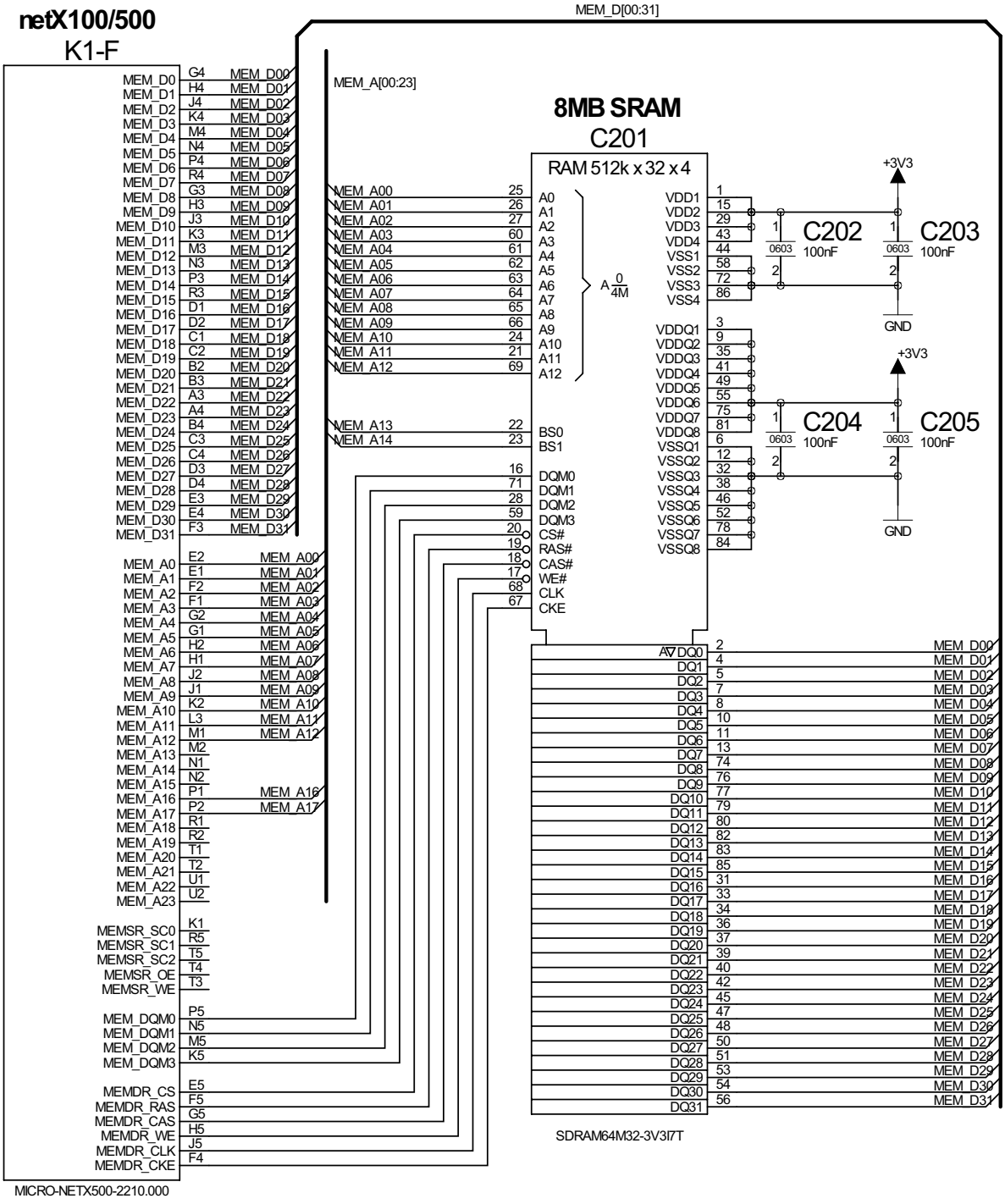


Figure 10: netX100/500 Connection of 8MB SDRAM

More about SDRAM circuits can be found on:

→ Page 81 Chapter 4.6.2 SDRAM

Ethernet

The basic circuit assumes that Ethernet is not used. Figure 11 shows the netX100/500 PHY power connection, if the Ethernet interface is not used.

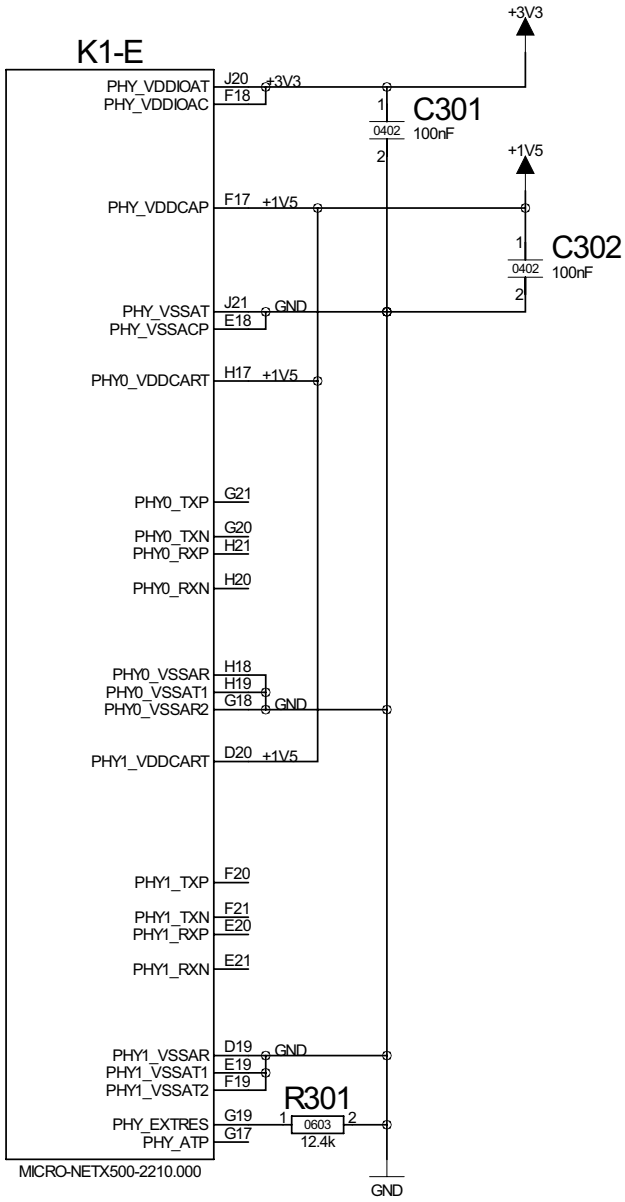


Figure 11: netX100/500 Ethernet Not Used

More about Ethernet circuits can be found on:

→ Page 25 Chapter 2.2 Ethernet Interface

→ Page 102 Chapter 4.10 Ethernet Interface

Host Interface

For the basic circuit, only the TCLK A11 pin has to be connected with a pull up resistor to ground. All other pins have to be connected to the companion chip.

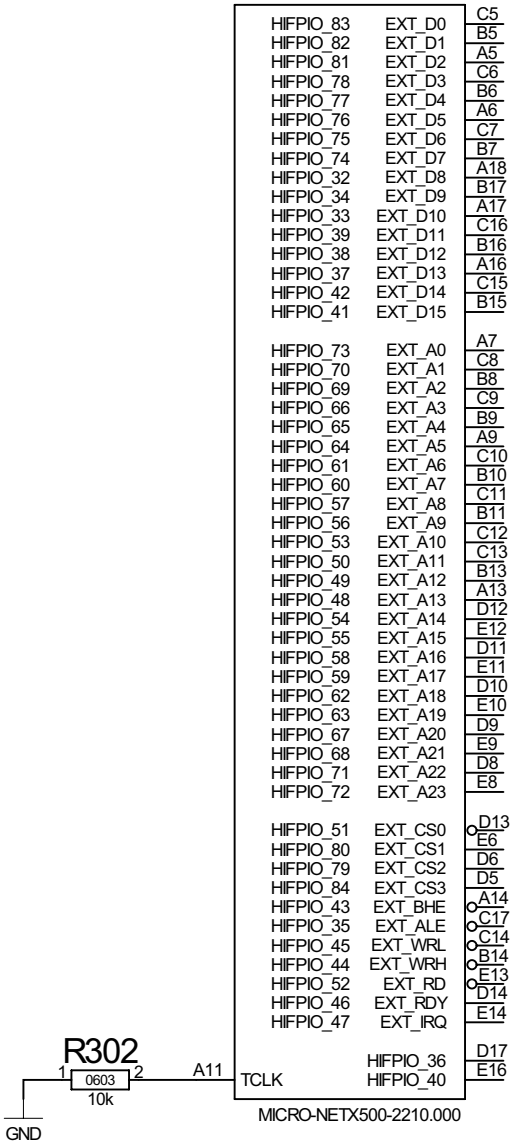


Figure 12: Unused Host Interface

Note: If the Host Interface is not connected, it is recommended to configure the pins in PIO mode (output) by software or connects pull-up/pull-down resistors. The reason is that the Host Interface pins are floating, because they do not have internal pull-up or pull-down resistor.

More about Host Interface circuits can be found on:

→ Page 84 Chapter 4.7

Host Interface

Power Supply 3.3 V

The following figure shows a power supply circuit for 9-24 V input voltage and 3.3 V/3 A output. The main part of the power supply is the MIC2198.

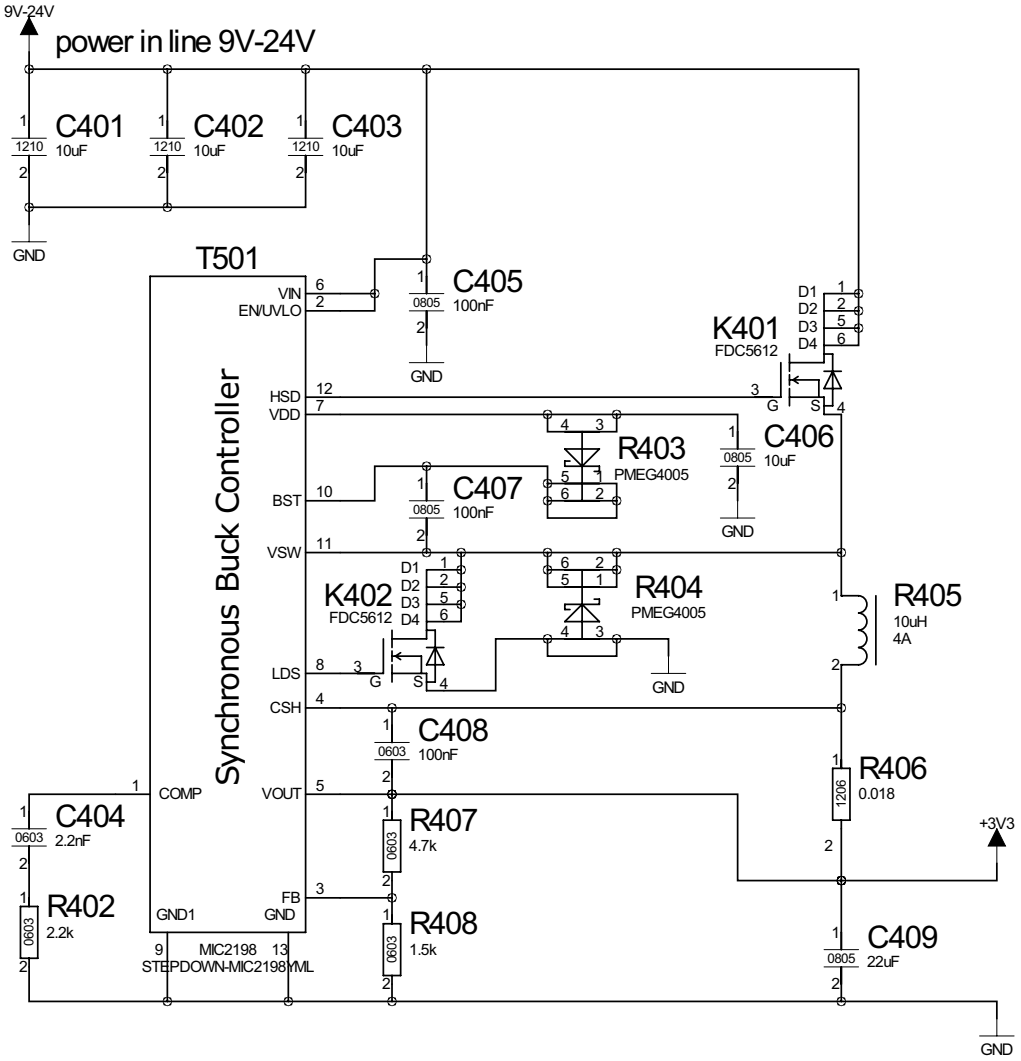


Figure 13: +3.3 V Power Supply Standard Circuit

Power Supply 1.5 V Core

The Hilscher standard circuit uses the FAN2001 to produce the 1.5 V core voltage.

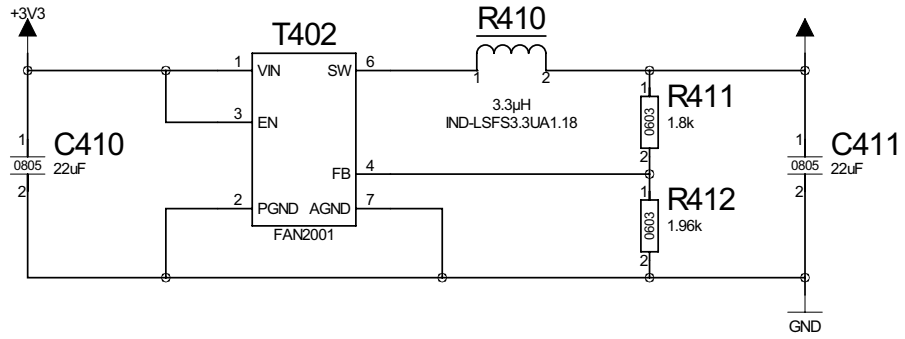


Figure 14: 1.5 V Core Voltage Regulator

More about power supply circuits can be found on:

→ Page 138 Chapter 4.18 Power Supply

netX100/500 Power Supply Pins

The following figure shows the connection of netX100/500 to the power supply.

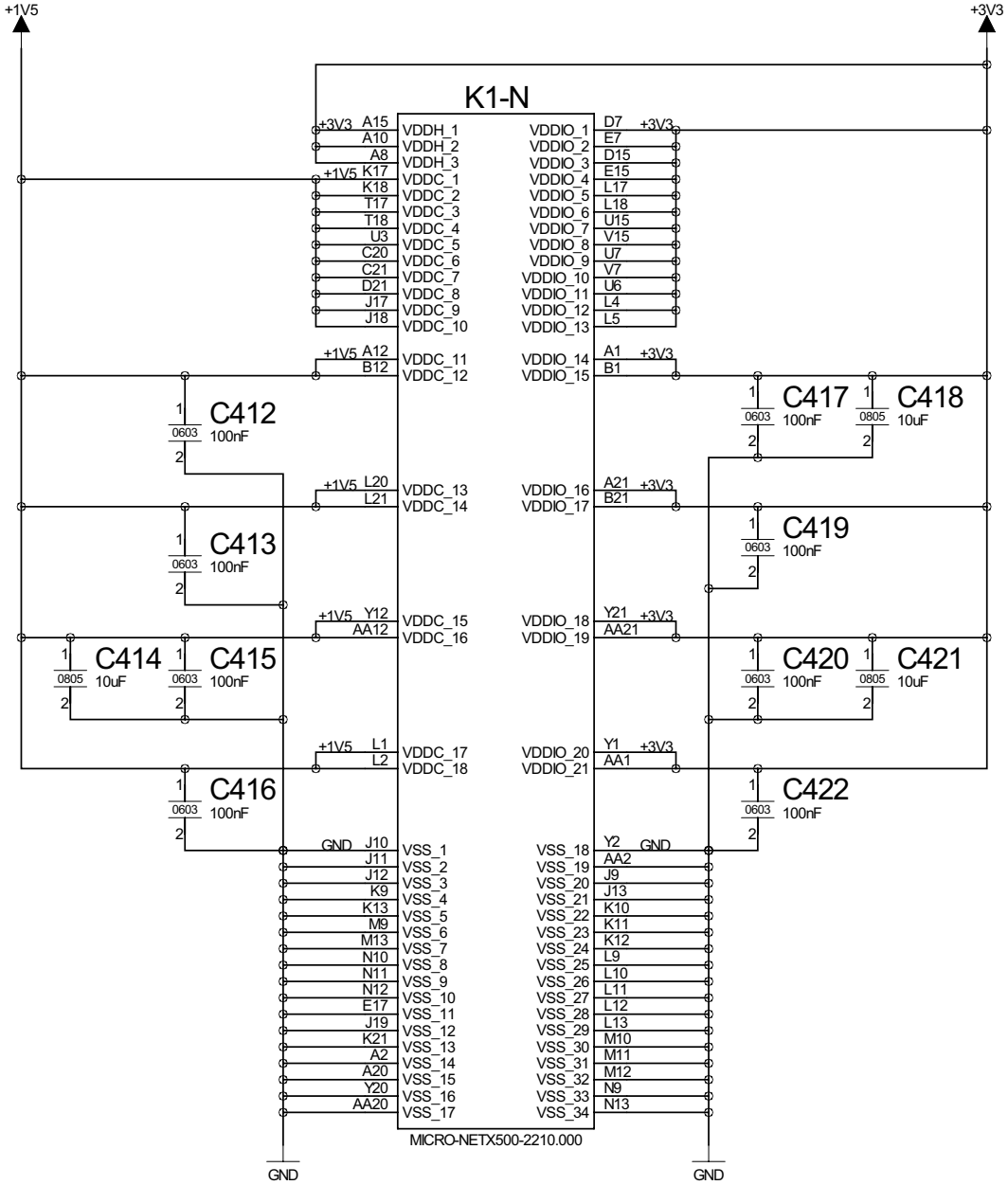


Figure 15: netX100/500 Power Connection

Pull-Down Resistors

The basic circuit does not use the pins D16 WDGACT (watchdog active) and B18 CLKOUT (clock out) and have always connected to GND over a 10 kΩ resistor in this case.

Not Connected Pins

All other pin of netX100/500 has not to be connected.

Basic Circuit Overview

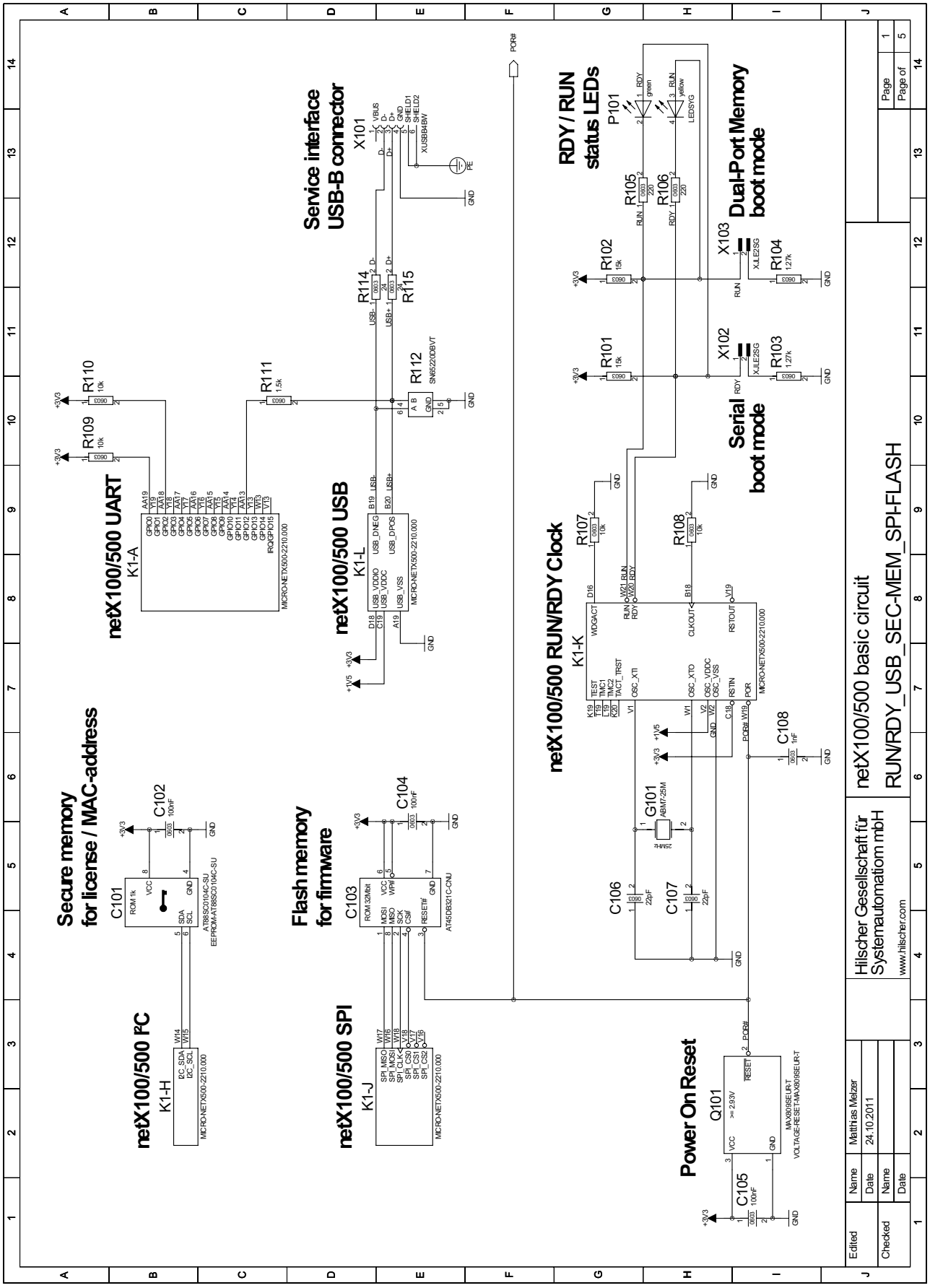


Figure 16: netX100/500 Basic Circuit RDY/RUN USB SEC-MEM SPI-FLASH

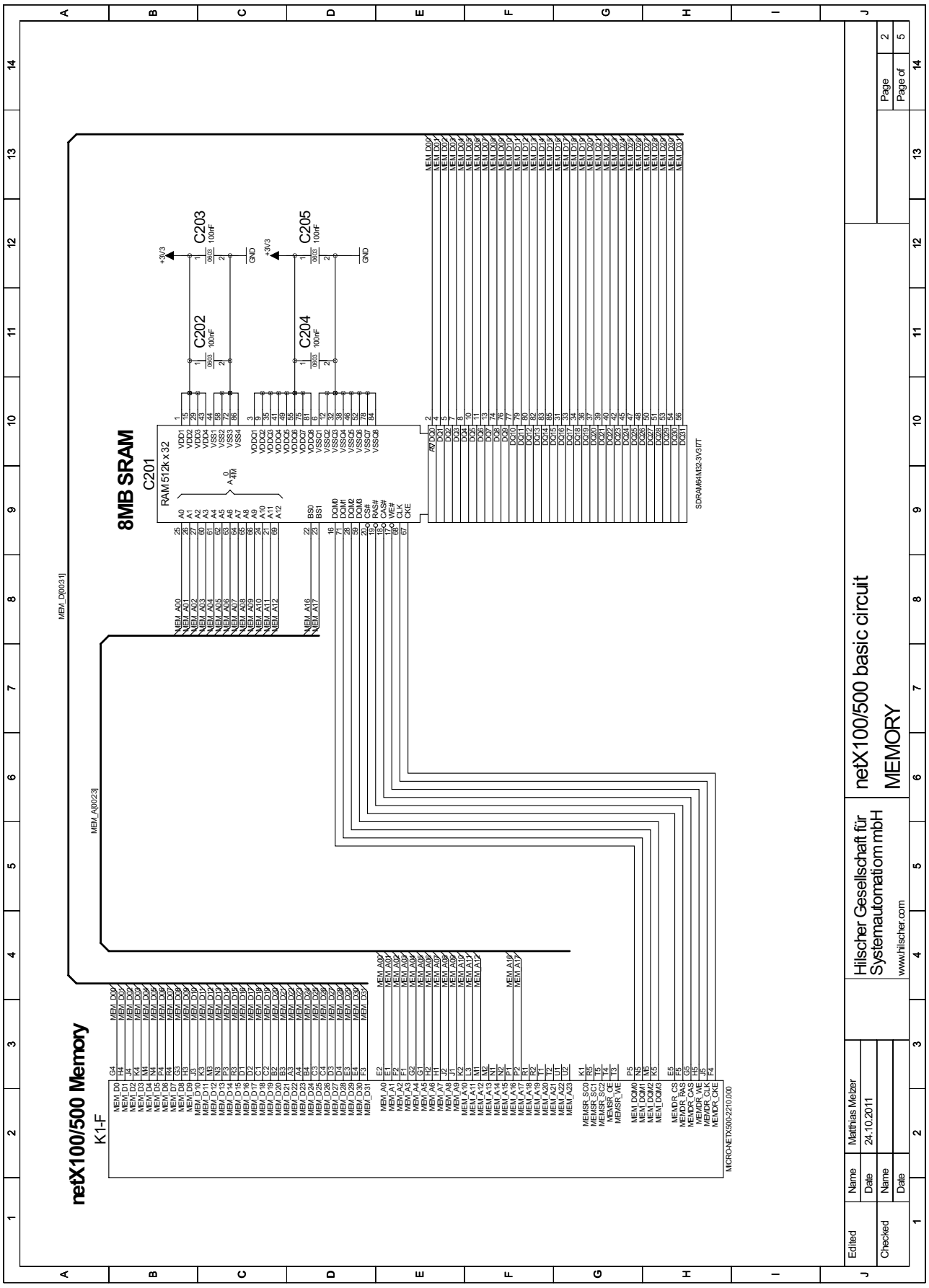


Figure 17: netX100/500 Basic Circuit SDRAM

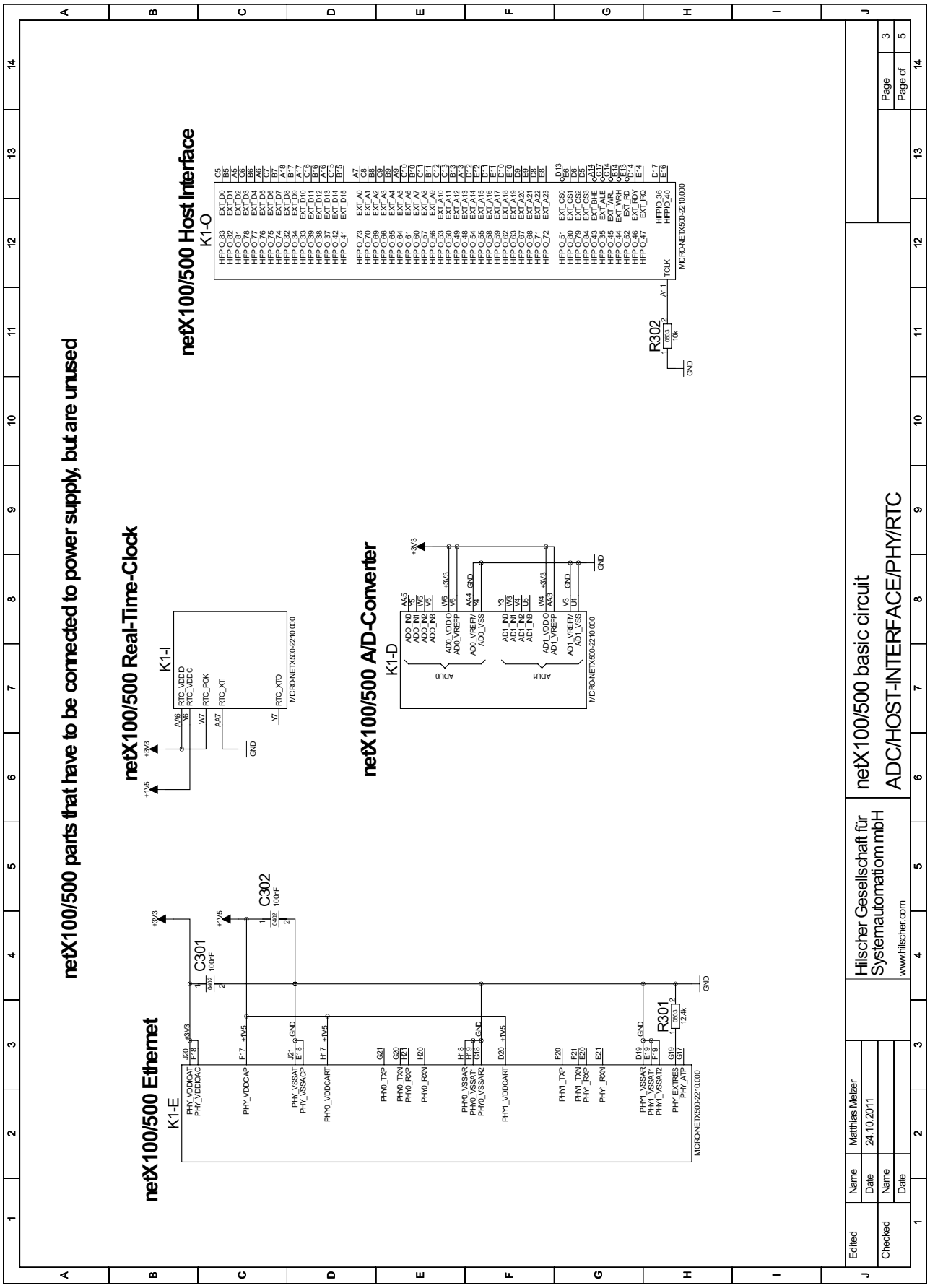


Figure 18: netX100/500 Basic Circuit ADC, Host Interface, PHY, RTC

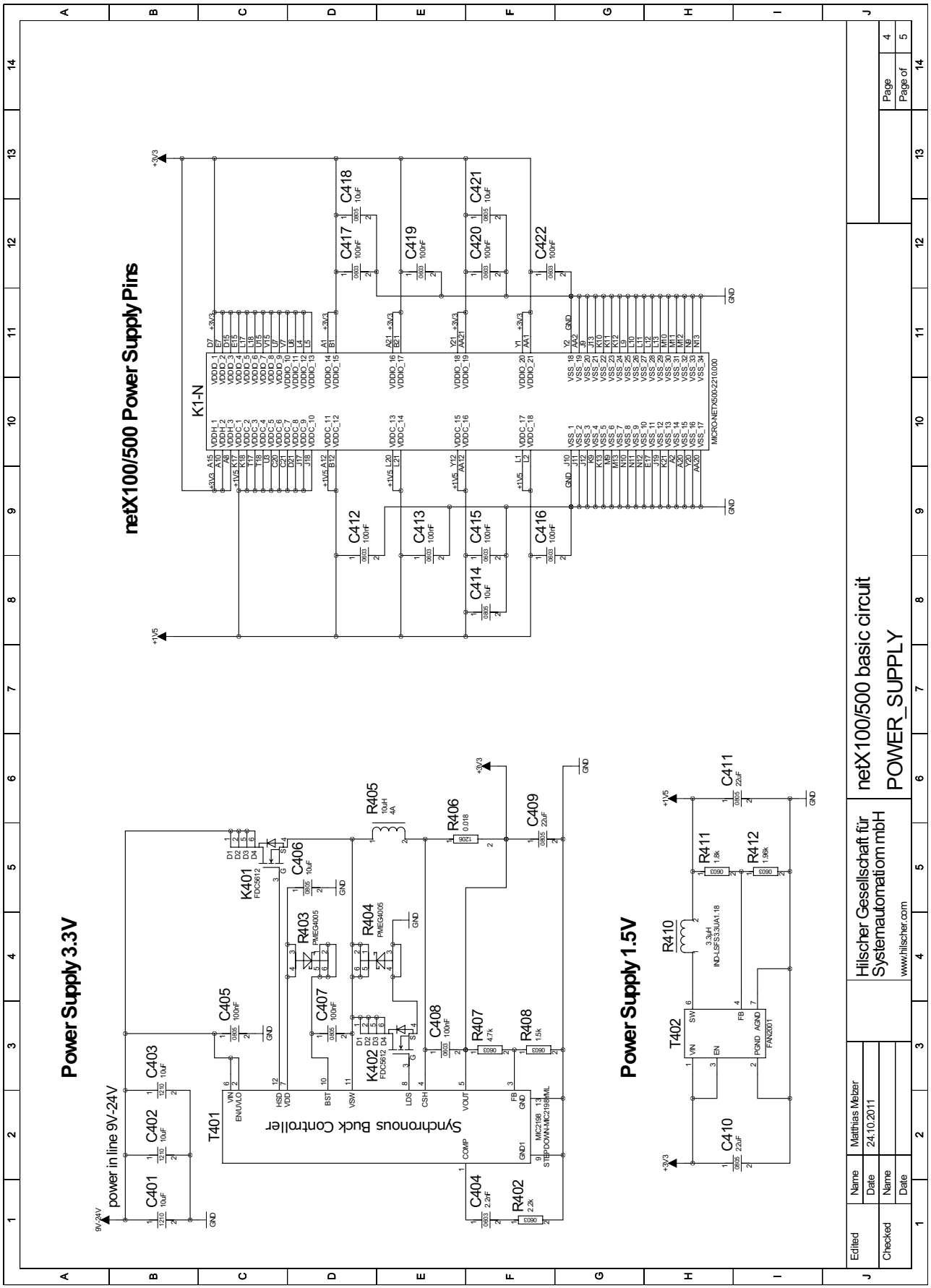


Figure 19: netX100/500 Basic Circuit Power Supply

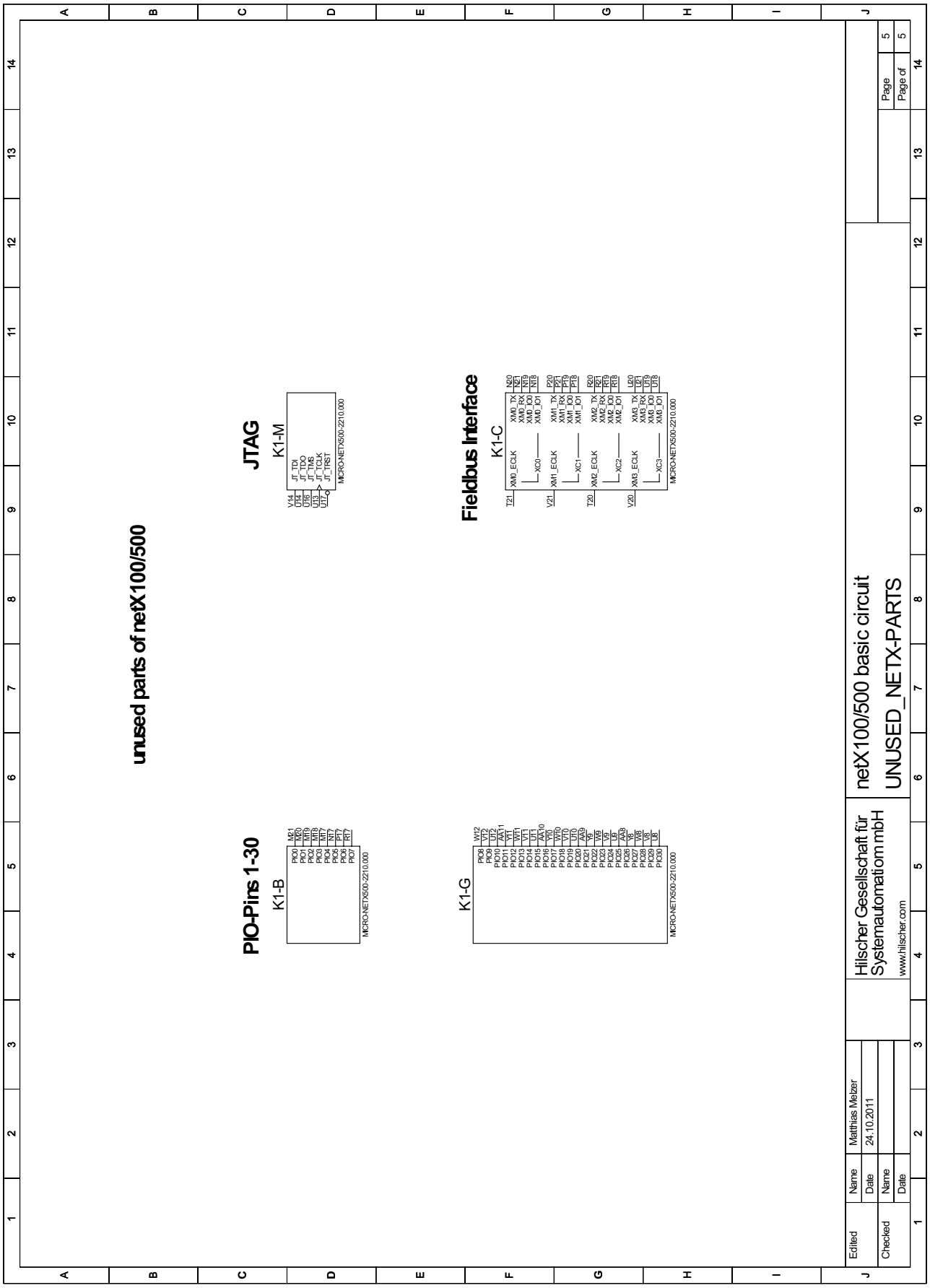


Figure 20: netX100/500 Basic Circuit Unused netX Parts

Bill of Materials

Page 1

REF DES	PART Type	PART NAME
C101	EEPROM	AT88SC0104CA-SU
C102	Ceramic Capacitor	100 nF 25 V 0603
C103	Flash Memory	AT45DB321D-SU
C104	Ceramic Capacitor	100 nF 25 V 0603
C105	Ceramic Capacitor	100 nF 25 V 0603
C106	Ceramic Capacitor	22 pF 50 V 0603
C107	Ceramic Capacitor	22 pF 50 V 0603
C108	Ceramic Capacitor	1 nF 50 V 0603
G101	Crystal	ABM7-25.000 MHZ-D2Y-T
K1	Microcontroller	netX100/500
P101	LED	HSMF-C156
Q101	Reset	MAX809SEUR-T
R101	Resistor	15 kΩ 63 mW 0603
R102	Resistor	15 kΩ 63 mW 0603
R103	Resistor	1.27 kΩ 63 mW 0603
R104	Resistor	1.27 kΩ 63 mW 0603
R105	Resistor	220 Ω 63 mW 0603
R106	Resistor	220 Ω 63 mW 0603
R107	Resistor	10 kΩ 63 mW 0603
R108	Resistor	10 kΩ 63 mW 0603
R109	Resistor	10 kΩ 63 mW 0603
R110	Resistor	10 kΩ 63 mW 0603
R111	Resistor	1.5 kΩ 63 mW 0603
R112	Diode	SN65220DBVT
R114	Resistor	24 Ω 63 mW 0603
R115	Resistor	24 Ω 63 mW 0603
X101	USB-B	KUSB-BS-1-N-BLK
X102	Pin Header	2 pin
X103	Pin Header	2 pin

Table 1: BOM Basic Circuit Page 1

Page 2

REF DES	PART Type	PART NAME
C201	SDRAM	IS42S32200C1-7TLI
C202	Ceramic Capacitor	100 nF 25 V 0603
C203	Ceramic Capacitor	100 nF 25 V 0603
C204	Ceramic Capacitor	100 nF 25 V 0603
C205	Ceramic Capacitor	100 nF 25 V 0603

Table 2: BOM Basic Circuit Page 2

Page 3

REF DES	PART Type	PART NAME
C301	Ceramic Capacitor	100 nF 25 V 0603
C302	Ceramic Capacitor	100 nF 25 V 0603
R301	Resistor	12.4 kΩ 63 mW 0603
R302	Resistor	10 kΩ 63 mW 0603

Table 3: BOM Basic Circuit Page 3

Page 4

REF DES	PART Type	PART NAME
C401	Ceramic Capacitor	10 µF 50 V 1210
C402	Ceramic Capacitor	10 µF 50 V 1210
C403	Ceramic Capacitor	10 µF 50 V 1210
C404	Ceramic Capacitor	2.2nF 50 V 0603
C405	Ceramic Capacitor	100 nF 50 V 0805
C406	Ceramic Capacitor	10 µF 10 V 0805
C407	Ceramic Capacitor	100 nF 50 V 0805
C408	Ceramic Capacitor	100 nF 25 V 0603
C409	Ceramic Capacitor	22 µF 6.3 V 0805
C410	Ceramic Capacitor	22 µF 6.3 V 0805
C411	Ceramic Capacitor	22 µF 6.3 V 0805
C412	Ceramic Capacitor	100 nF 25 V 0603
C413	Ceramic Capacitor	100 nF 25 V 0603
C414	Ceramic Capacitor	10 µF 10 V 0805
C415	Ceramic Capacitor	100 nF 25 V 0603
C416	Ceramic Capacitor	100 nF 25 V 0603
C417	Ceramic Capacitor	100 nF 25 V 0603
C418	Ceramic Capacitor	10 µF 10 V 0805
C419	Ceramic Capacitor	100 nF 25 V 0603
C420	Ceramic Capacitor	100 nF 25 V 0603
C421	Ceramic Capacitor	10 µF 10 V 0805
C422	Ceramic Capacitor	100 nF 25 V 0603
K401	Transistor	FDC5612
K402	Transistor	FDC5612
R402	Resistor	2.2 kΩ 63 mW 0603
R403	Diode	DIODE-SCHOTTKY-PMEG4005AEV
R404	Diode	DIODE-SCHOTTKY-PMEG4005AEV
R405	Inductor	CDRH8D43NP-100N
R406	Resistor	0.018 Ω 250mW 1206
R407	Resistor	4.7 kΩ 63 mW 0603
R408	Resistor	1.5 kΩ 63 mW 0603
R410	Inductor	CR32NP-3R3M
R411	Resistor	1.8 kΩ 63 mW 0603
R412	Resistor	1.96 kΩ 63 mW 0603
T401	Step-Down	MIC2198YML
T402	Step-Down	FAN2001MPX

Table 4: BOM Basic Circuit Page 4

2.2 Ethernet Interface Circuits

2.2.1 Twisted Pair Two Channel

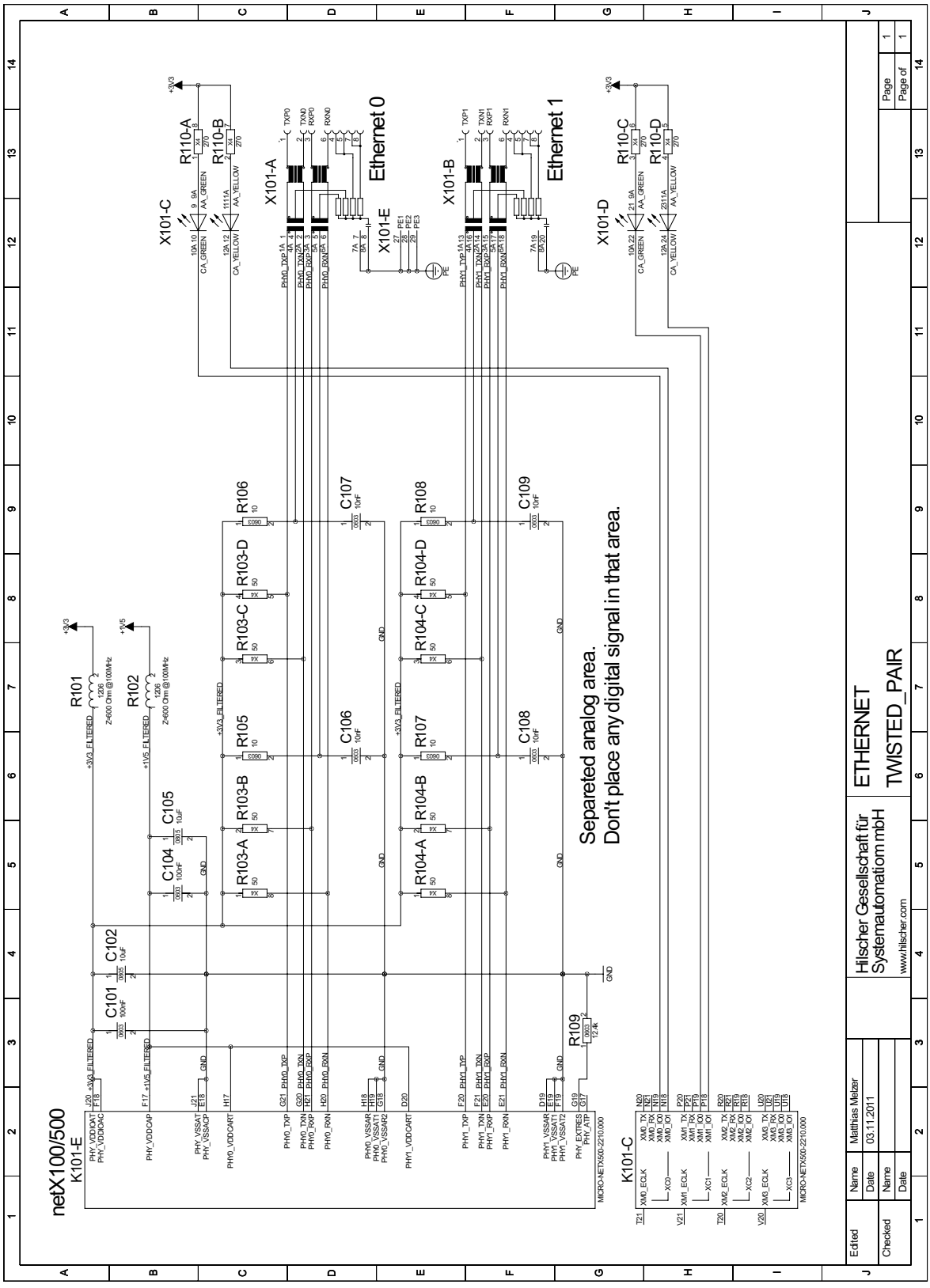


Figure 21: Ethernet TP Dual Channel Circuit

Bill of Materials

REF DES	PART NAME	PART NAME
C101	Ceramic Capacitor	100 nF 25 V 0603
C102	Ceramic Capacitor	10 µF 10 V 0805
C104	Ceramic Capacitor	100 nF 25 V 0603
C105	Ceramic Capacitor	10 µF 10 V 0805
C106	Ceramic Capacitor	10 nF 50 V 0603
C107	Ceramic Capacitor	10 nF 50 V 0603
C108	Ceramic Capacitor	10 nF 50 V 0603
C109	Ceramic Capacitor	10 nF 50 V 0603
K101	Microcontroller	netX100/500
R101	Ferrite	100 MHz 600 Ω 1 A 1206
R102	Ferrite	100 MHz 600 Ω 1 A 1206
R103	Resistor Array	4x 50 Ω 62 mW 1206
R104	Resistor Array	4x 50 Ω 62 mW 1206
R105	Resistor	10 Ω 63 mW 0603
R106	Resistor	10 Ω 63 mW 0603
R107	Resistor	10 Ω 63 mW 0603
R108	Resistor	10 Ω 63 mW 0603
R109	Resistor	12.4 kΩ 63 mW 0603
R110	Resistor Array	4x 270 Ω 62 mW 1206
X101	RJ45	ERNI-203313

Table 5: BOM Ethernet TP Two Channel

More about Ethernet circuits can be found on:

→ Page 102 chapter 4.10 Ethernet Interface

2.2.2 Twisted Pair Single Channel

The following figure shows the netX100/500 Ethernet circuit with only on channel.

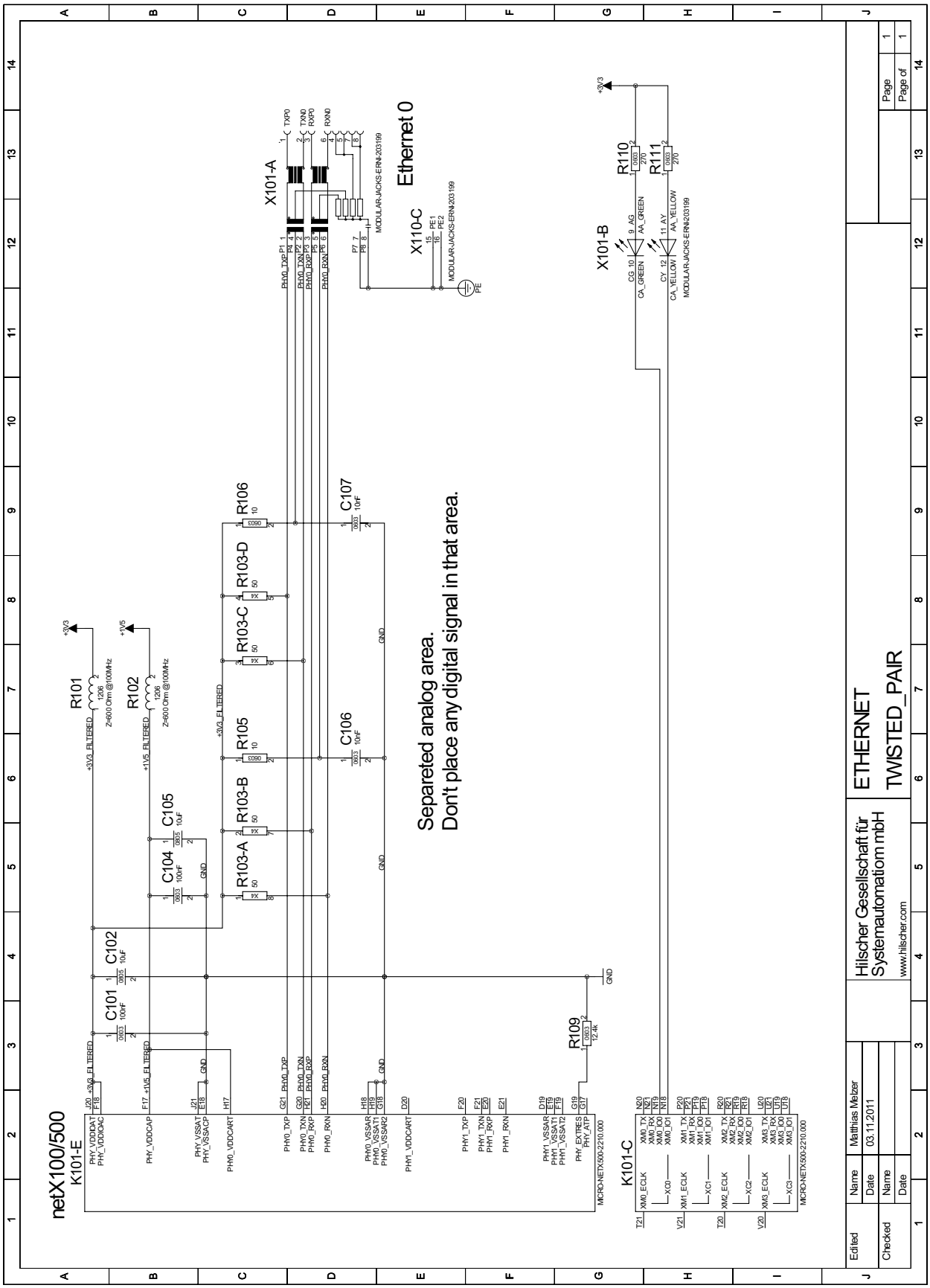


Figure 22: Ethernet TP Single Channel Circuit

Bill of Materials

REF DES	PART TYPE	PART NAME
C101	ceramic capacitor	100 nF 25 V 0603
C102	ceramic capacitor	10 µF 10 V 0805
C104	ceramic capacitor	100 nF 25 V 0603
C105	ceramic capacitor	10 µF 10 V 0805
C106	ceramic capacitor	10 nF 50 V 0603
C107	ceramic capacitor	10 nF 50 V 0603
K101	Microcontroller	netX100/500
R101	Ferrite	100 MHz 600 Ω 1 A 1206
R102	Ferrite	100 MHz 600 Ω 1 A 1206
R103	Resistor Array	4x 50 Ω 62 mW 1206
R105	Resistor	10 Ω 63 mW 0603
R106	Resistor	10 Ω 63 mW 0603
R109	Resistor	12.4 kΩ 63 mW 0603
R110	Resistor	270 Ω 63 mW 0603
R111	Resistor	270 Ω 63 mW 0603
X101	RJ45	ERNI-203199

Table 6: BOM Ethernet TP Single Channel

More about SDRAM circuits can be found on:

→ Page 102 chapter 4.10 Ethernet Interface

2.2.3 Fiber Optic with AFBR-5978Z

The first figure shows the connection on the netX100/500 side.

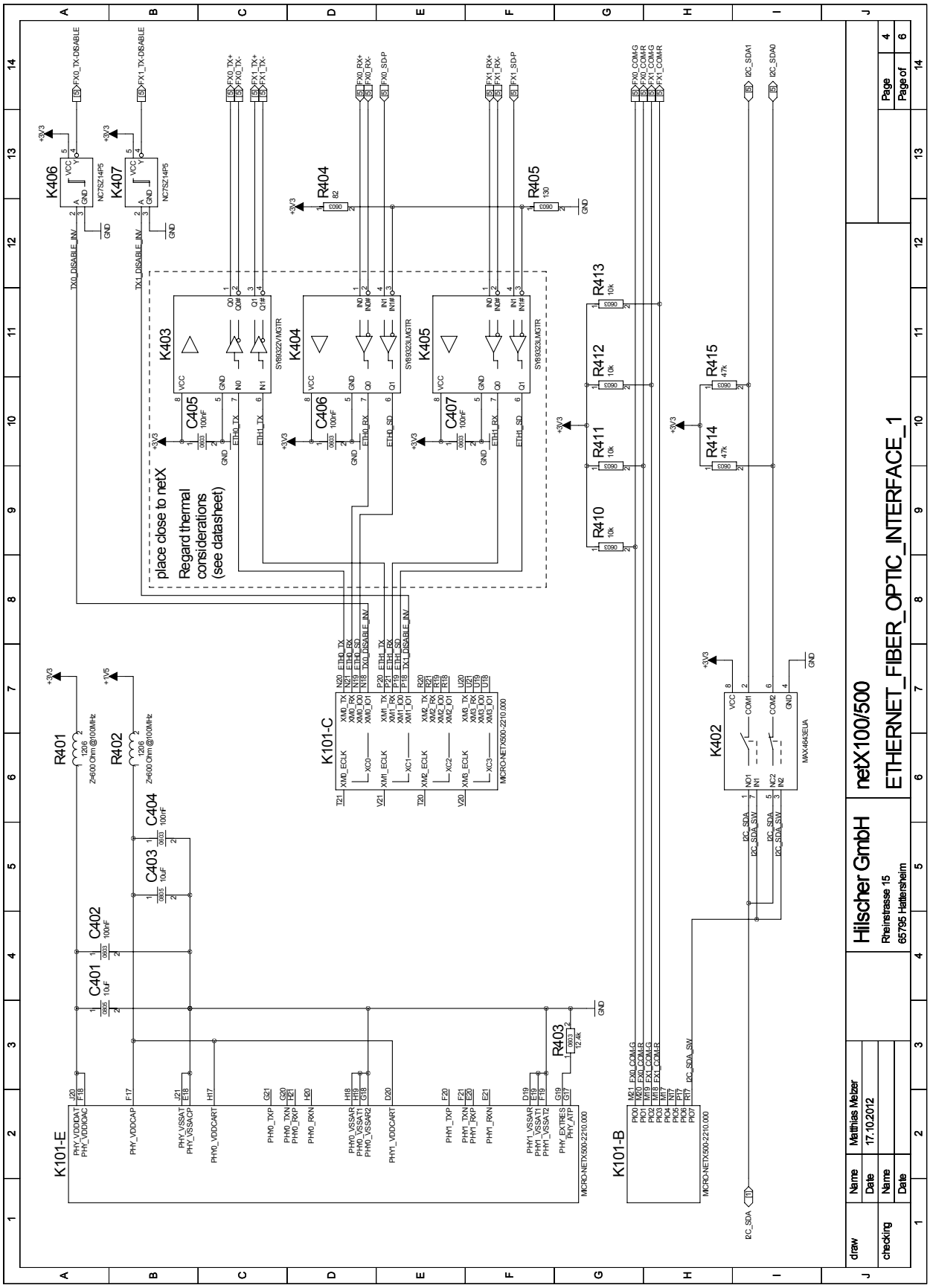


Figure 23: Fiber Optic Circuit on netX100/500 Side

The second figure shows the circuit on the AFBR-5978Z side. Important is to place the level transistors close to netX100/500 and AC-termination (R501 – R510 and R17 – R526) close to the level transistors.

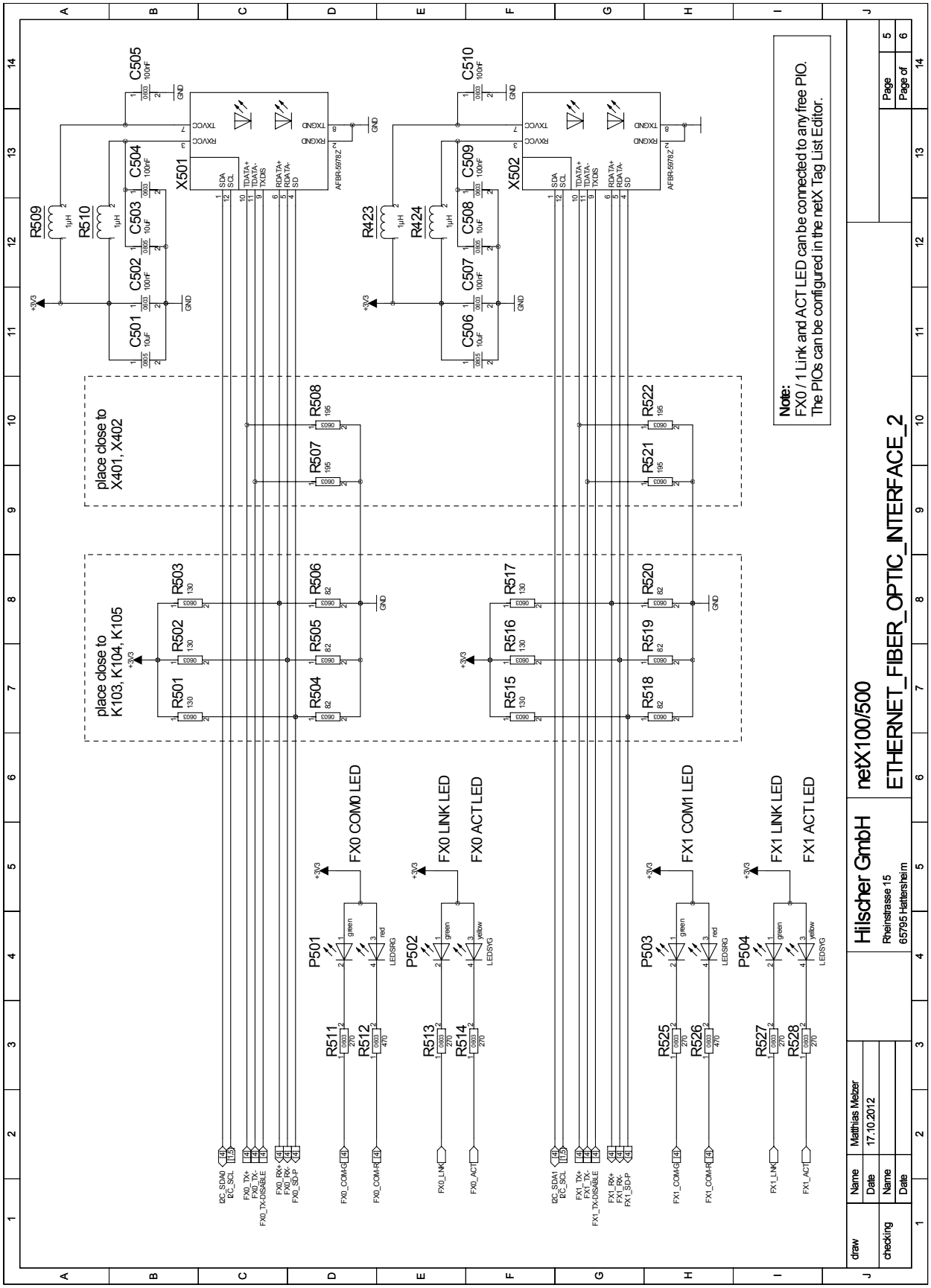


Figure 24: Fiber Optic Circuit on AFBR-5978Z Side

Bill of Materials

Page 1

REF DES	PART TYPE	PART NAME
C401	Ceramic Capacitor	10 µF 10 V 0805
C402	Ceramic Capacitor	100 nF 25 V 0603
C403	Ceramic Capacitor	10 µF 10 V 0805
C404	Ceramic Capacitor	100 nF 25 V 0603
C405	Ceramic Capacitor	100 nF 25 V 0603
C406	Ceramic Capacitor	100 nF 25 V 0603
C407	Ceramic Capacitor	100 nF 25 V 0603
K101	Microcontroller	netX100/500
K402	IC-Switch	MAX4643EUA
K403	Driver	SY89322VMGTR
K404	Driver	SY89323LMGTR
K405	Driver	SY89323LMGTR
K406	Schmitt Trigger and Inverter	NC7SZ14P5X
K407	Schmitt Trigger and Inverter	NC7SZ14P5X
R401	Ferrite	100 MHz 600 Ω 1 A 1206
R402	Ferrite	100 MHz 600 Ω 1 A 1206
R403	Resistor	12.4 kΩ 63 mW 0603
R404	Resistor	82 Ω 63 mW 0603
R405	Resistor	130 Ω 63 mW 0603
R410	Resistor	10 kΩ 63 mW 0603
R411	Resistor	10 kΩ 63 mW 0603
R412	Resistor	10 kΩ 63 mW 0603
R413	Resistor	10 kΩ 63 mW 0603
R414	Resistor	47 kΩ 63 mW 0603
R415	Resistor	47 kΩ 63 mW 0603

Table 7: BOM Fiber Optic Page 1

Page 2

REF DES	PART TYPE	PART NAME
C501	Ceramic Capacitor	10 µF 10 V 0805
C502	Ceramic Capacitor	100 nF 25 V 0603
C503	Ceramic Capacitor	10 µF 10 V 0805
C504	Ceramic Capacitor	100 nF 25 V 0603
C505	Ceramic Capacitor	100 nF 25 V 0603
C506	Ceramic Capacitor	10 µF 10 V 0805
C507	Ceramic Capacitor	100 nF 25 V 0603
C508	Ceramic Capacitor	10 µF 10 V 0805
C509	Ceramic Capacitor	100 nF 25 V 0603
C510	Ceramic Capacitor	100 nF 25 V 0603
P501	LED	HSMF-C155 1210
P503	LED	HSMF-C155 1210
R501	Resistor	130 Ω 63 mW 0603
R502	Resistor	130 Ω 63 mW 0603
R503	Resistor	130 Ω 63 mW 0603
R506	Resistor	82 Ω 63 mW 0603
R507	Resistor	82 Ω 63 mW 0603
R508	Resistor	82 Ω 63 mW 0603
R509	Resistor	82 Ω 63 mW 0603
R510	Resistor	82 Ω 63 mW 0603
R511	Ferrite	1 µH 0,28 Ω 600mA 1812
R512	Ferrite	1 µH 0,28 Ω 600mA 1812
R513	Resistor	270 Ω 63 mW 0603
R514	Resistor	470 Ω 63 mW 0603
R517	Resistor	130 Ω 63 mW 0603
R518	Resistor	130 Ω 63 mW 0603
R519	Resistor	130 Ω 63 mW 0603
R522	Resistor	82 Ω 63 mW 0603
R523	Resistor	82 Ω 63 mW 0603
R524	Resistor	82 Ω 63 mW 0603
R525	Resistor	82 Ω 63 mW 0603
R526	Resistor	82 Ω 63 mW 0603
R527	Ferrite	1 µH 0,28 Ω 600mA 1812
R528	Ferrite	1 µH 0,28 Ω 600mA 1812
R529	Resistor	270 Ω 63 mW 0603
R530	Resistor	470 Ω 63 mW 0603
X501	Fiber Optic Connector	AFBR-5978Z
X502	Fiber Optic Connector	AFBR-5978Z

Table 8: BOM Fiber Optic Page 2

More about SDRAM circuits can be found on:

→ Page 105 chapter 4.10.2 Fiber Optic

2.3 AS-Interface

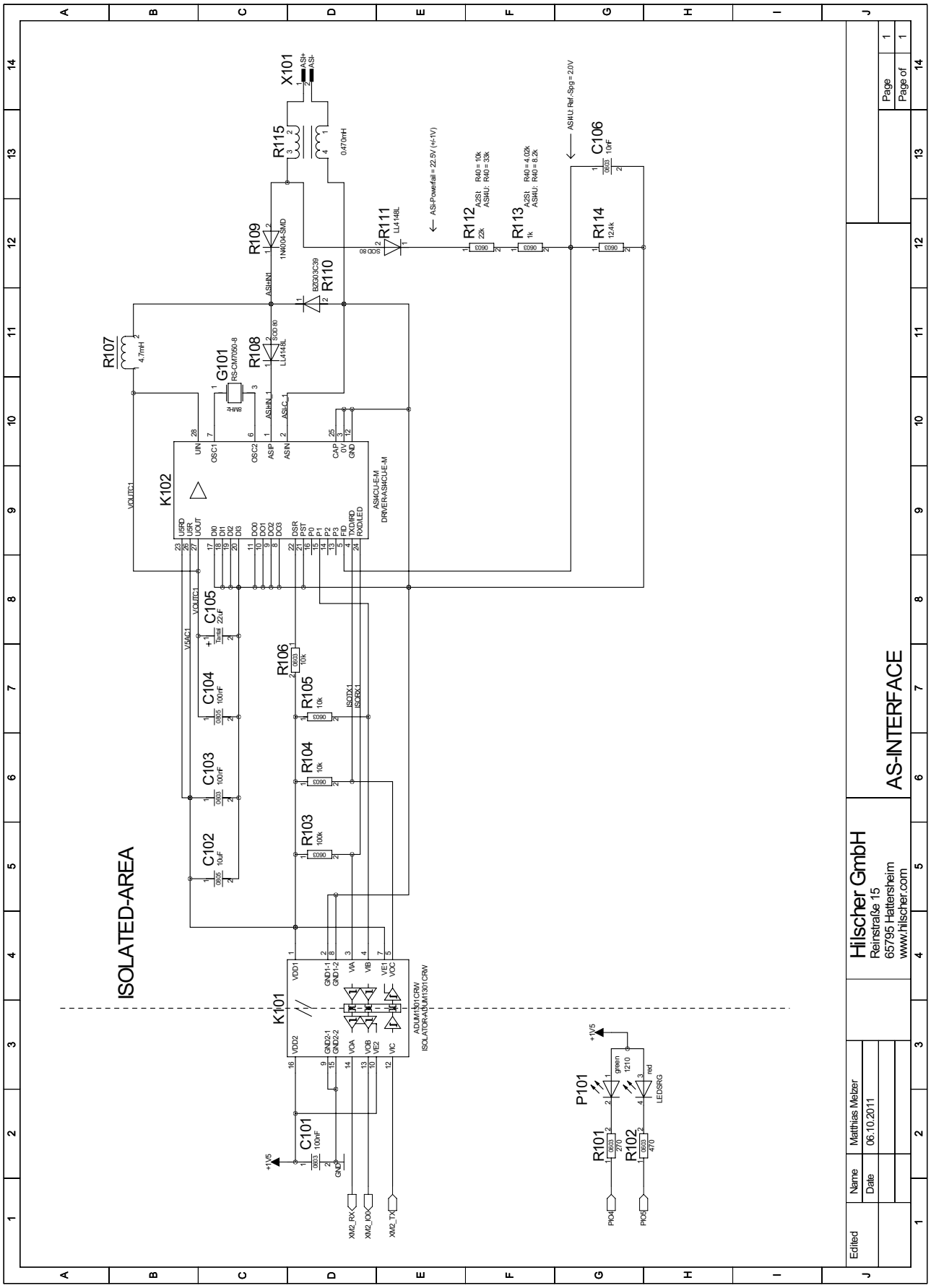


Figure 25: AS-Interface Circuit

Bill of Materials

REF DES	PART TYPE	PART NAME
C101	Ceramic Capacitor	100 nF 25 V 0603
C102	Ceramic Capacitor	10 µF 10 V 0805
C103	Ceramic Capacitor	100 nF 25 V 0603
C104	Ceramic Capacitor	100 nF 50 V 0805
C105	Tantalum Capacitor	22 µF 35 V
C106	Ceramic Capacitor	10 nF 50 V 0603
G101	Crystal	RS-CM7050 8 MHz
K101	Triple-Channel Digital Isolators	ADUM1301CRW
K102	Advanced AS-Interface IC	ASI4UC-E-M
P101	LED red/green	HSMF-C155
R101	Resistor	270 Ω 63 mW 0603
R102	Resistor	470 Ω 63 mW 0603
R103	Resistor	100 kΩ 63 mW 0603
R104	Resistor	10 kΩ 63 mW 0603
R105	Resistor	10 kΩ 63 mW 0603
R106	Resistor	10 kΩ 63 mW 0603
R107	Inductor	4.7 mH 0.1 A Würth Elektronik Part No. 744775347
R108	Diode	LL4148L
R109	Diode	SM4004
R110	Diode	BZG03C39
R111	Diode	LL4148L
R112	Resistor	22 kΩ 63 mW 0603
R113	Resistor	1 kΩ 63 mW 0603
R114	Resistor	12.4 kΩ 63 mW 0603
R115	Inductor	EPCOS B82790C0474N215
X101	Connector	MC1,5/2-G-3,81

Table 9: BOM AS-Interface Circuit

More about AS-Interface circuits can be found on:

→ Page 118 chapter 4.11 Fieldbus Interface

→ Page 120 chapter 4.11.1 AS interface Master

→ Page 126 chapter 4.11.7 Fieldbus Status LEDs

2.4 CANopen

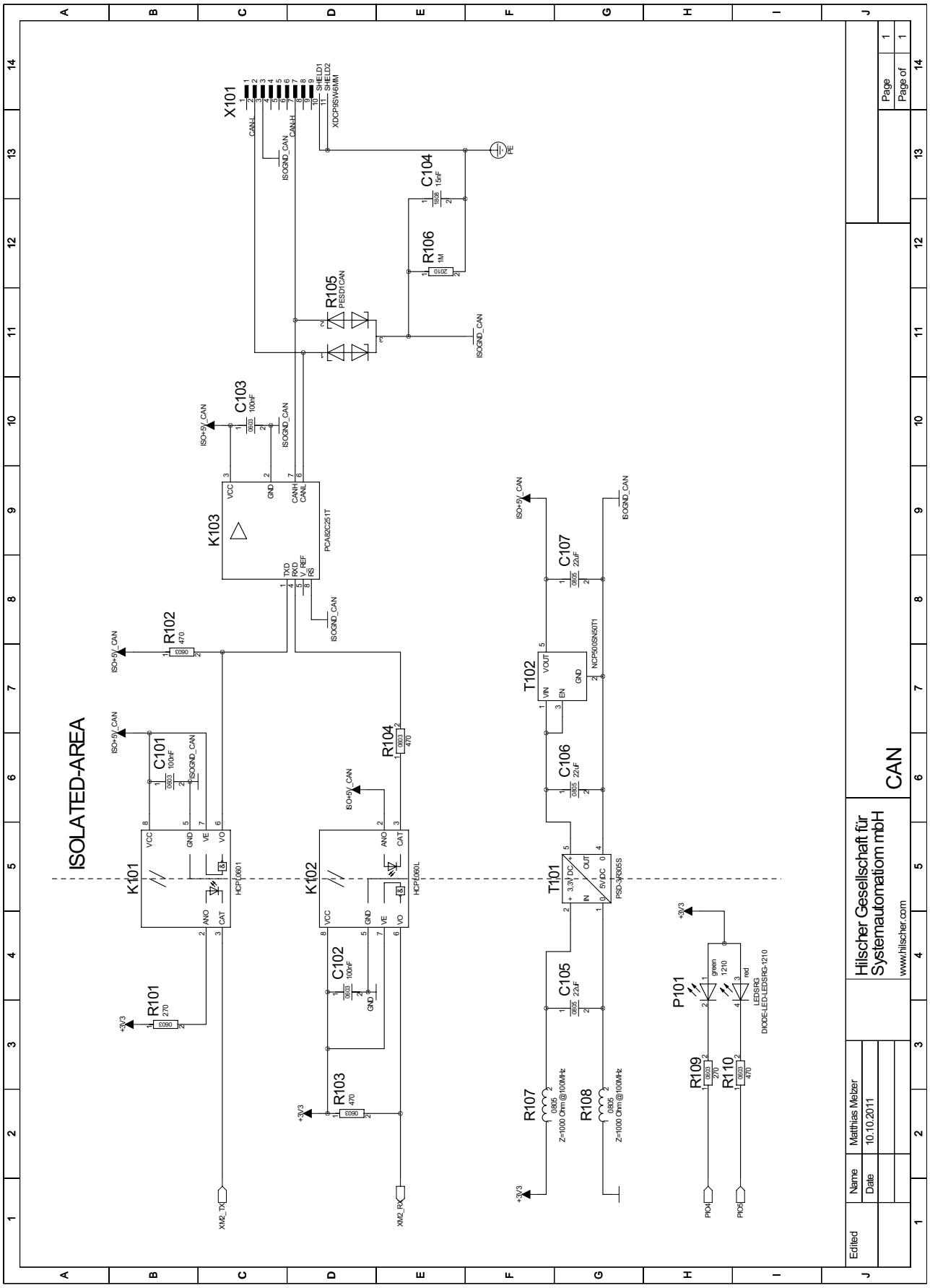


Figure 26: CANopen Circuit

Bill of Materials

REF DES	PART TYPE	PART NAME
C101	Ceramic Capacitor	100 nF 25 V 0603
C102	Ceramic Capacitor	100 nF 25 V 0603
C103	Ceramic Capacitor	100 nF 25 V 0603
C104	Ceramic Capacitor	15 nF 1000 V 1808
C105	Ceramic Capacitor	22 µF 6.3 V 0805
C106	Ceramic Capacitor	22 µF 6.3 V 0805
C107	Ceramic Capacitor	22 µF 6.3 V 0805
K101	Optocoupler	HCPL0601
K102	Optocoupler	HCPL060L
K103	Transceiver	PCA82C251T
P101	LED red/green	HSMF-C155
R101	Resistor	270 Ω 63 mW 0603
R102	Resistor	470 Ω 63 mW 0603
R103	Resistor	470 Ω 63 mW 0603
R104	Resistor	470 Ω 63 mW 0603
R105	ESD protection diode	PESD1CAN
R106	Resistor	1M Ω 500 mW 2010
R107	Ferrite	100 Ω 100 MHz 1 A Würth Elektronik Part No. 74279207
R108	Ferrite	100 Ω 100 MHz 1 A Würth Elektronik Part No. 74279207
R109	Resistor	270 Ω 63 mW 0603
R110	Resistor	470 Ω 63 mW 0603
T101	DC/DC Step-Up	PEAK part no. PSD-3R305S
T102	Voltage Regulator	NCP500SN50T1
X101	D-Sub9 Male	SUYIN USA part no. 070211MR009G200ZU

Table 10: BOM CANopen Circuit

More about CANopen circuits can be found on:

→ Page 118 chapter 4.11 Fieldbus Interface

→ Page 121 chapter 4.11.2 CANopen Interface

→ Page 126 chapter 4.11.7 Fieldbus Status LEDs

2.5 CC-Link

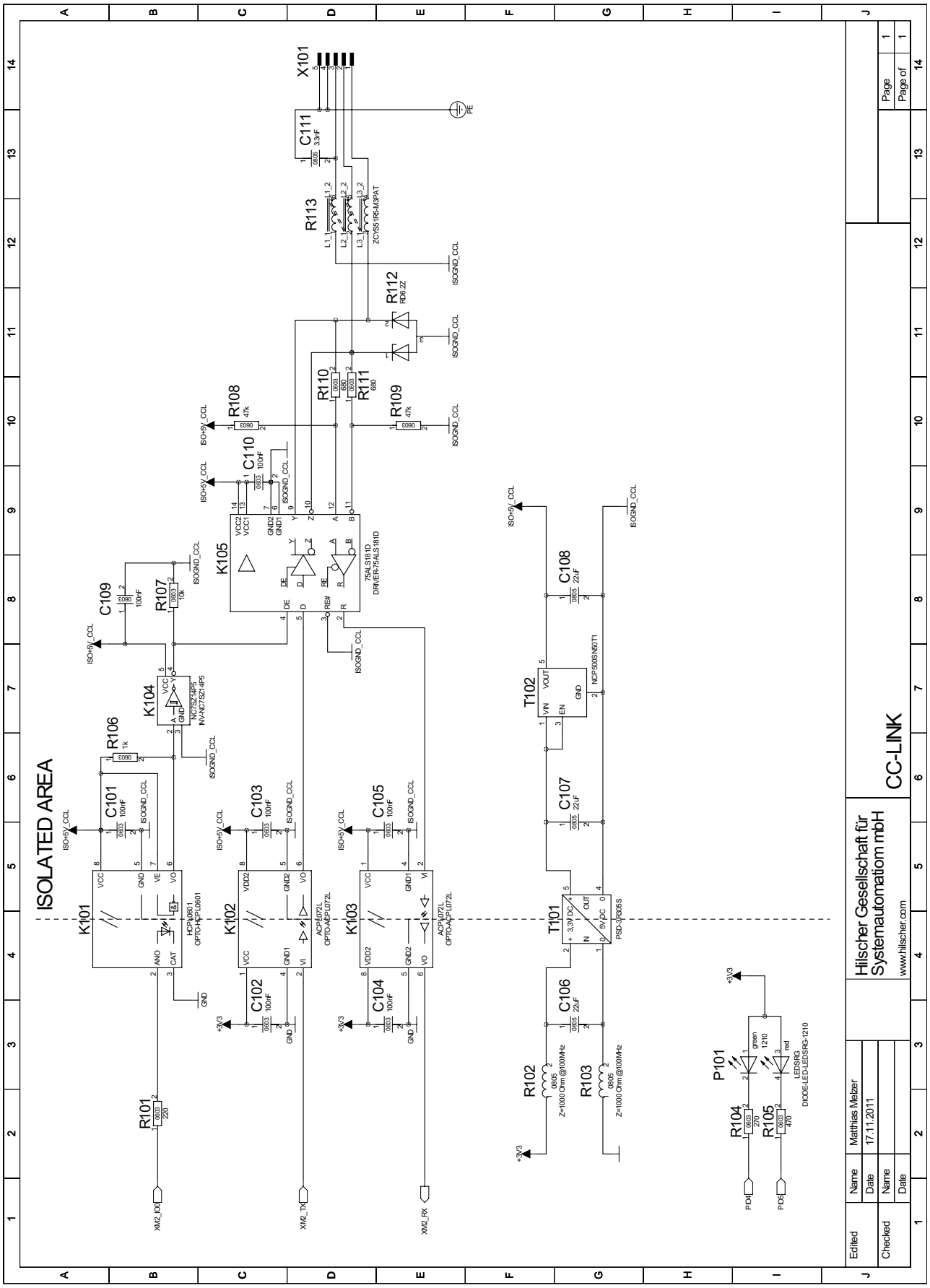


Figure 27: CC-Link Circuit

Bill of Material

REF DES	PART TYPE	PART NAME
C101	Ceramic Capacitor	100 nF 25 V 0603
C102	Ceramic Capacitor	100 nF 25 V 0603
C103	Ceramic Capacitor	100 nF 25 V 0603
C104	Ceramic Capacitor	100 nF 25 V 0603
C105	Ceramic Capacitor	100 nF 25 V 0603
C106	Ceramic Capacitor	22 µF 6.3 V 0805
C107	Ceramic Capacitor	22 µF 6.3 V 0805
C108	Ceramic Capacitor	22 µF 6.3 V 0805
C109	Ceramic Capacitor	100 nF 25 V 0603
C110	Ceramic Capacitor	100 nF 25 V 0603
C111	Ceramic Capacitor	3.3 nF 63 V 0805
K101	Optocoupler	HCPL0601
K102	Optocoupler	ACPL-072L
K103	Optocoupler	ACPL-072L
K104	Schmitt Trigger	FAIRCHILD Ord. No.:NC7SZ14P5X
K105	Transceiver	TEXAS INSTRUMENTS Ord. No. SN75ALS181NSR
P101	LED red/green	HSMF-C155
R101	Resistor	220 Ω 63 mW 0603
R102	Ferrite	100 Ω 100 MHz 1 A Würth Elektronik Part No. 74279207
R103	Ferrite	100 Ω 100 MHz 1 A Würth Elektronik Part No. 74279207
R104	Resistor	270 Ω 63 mW 0603
R105	Resistor	470 Ω 63 mW 0603
R106	Resistor	1 kΩ 63 mW 0603
R107	Resistor	10 kΩ 63 mW 0603
R108	Resistor	47 kΩ 63 mW 0603
R109	Resistor	47 kΩ 63 mW 0603
R110	Resistor	680 Ω 63 mW 0603
R111	Resistor	680 Ω 63 mW 0603
R112	ESD protection diode	RD6.2Z
R113	Inductor	EMC Components Part No. ZCYS51R5-M3PAT
T101	DC/DC Step-Up	PSD-3R305S
T102	Voltage Regulator	NCP500SN50T1
X101	Connector	MC1,5/5-G-3,81

Table 11: BOM CC-Link Circuit

More about CC-Link circuits can be found on:

→ Page 118 chapter 4.11 Fieldbus Interface

→ Page 122 chapter 4.11.3 CC-Link Interface

→ Page 126 chapter 4.11.7 Fieldbus Status LEDs

2.6 CompoNet

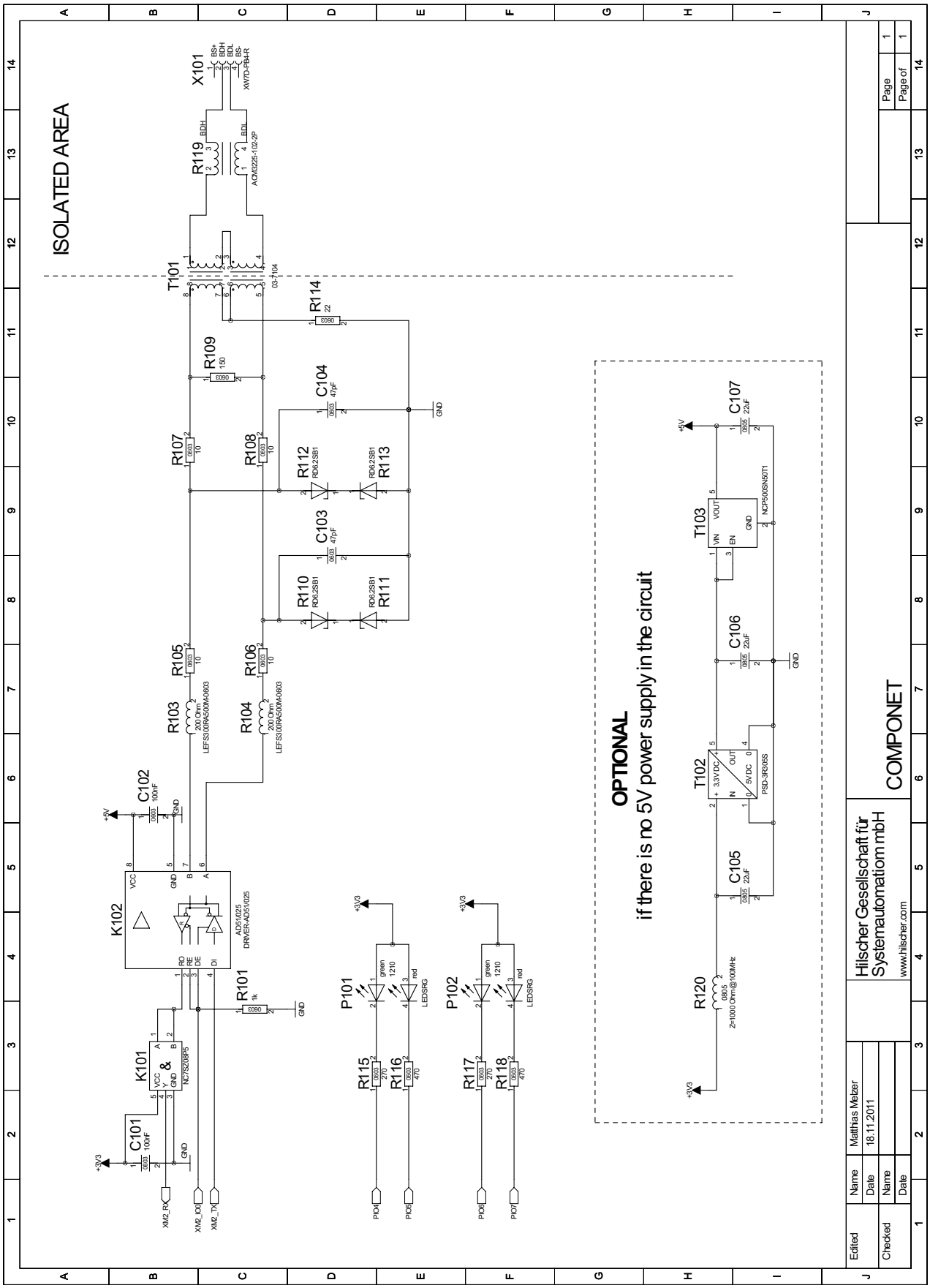


Figure 28: CompoNet Circuit

Bill of Material

REF DES	PART TYPE	PART NAME
C101	Ceramic Capacitor	100 nF 25 V 0603
C102	Ceramic Capacitor	100 nF 25 V 0603
C103	Ceramic Capacitor	47 pF 50 V 0603
C104	Ceramic Capacitor	47 pF 50 V 0603
K101	2-Input AND Gate	NC7SZ08P5
K102	Transceiver	AD51/025
P101	LED red/green	HSMF-C155
P102	LED red/green	HSMF-C155
R101	Resistor	1 kΩ 63 mW 0603
R103	Ferrite	TDK Part No. MMZ1608B301C
R104	Ferrite	TDK Part No. MMZ1608B301C
R105	Resistor	10 Ω 63 mW 0603
R106	Resistor	10 Ω 63 mW 0603
R107	Resistor	10 Ω 63 mW 0603
R108	Resistor	10 Ω 63 mW 0603
R109	Resistor	150 Ω 63 mW 0603
R110	Diode	NEC Type No. RD6.2S Class: B1
R111	Diode	NEC Type No. RD6.2S Class: B1
R112	Diode	NEC Type No. RD6.2S Class: B1
R113	Diode	NEC Type No. RD6.2S Class: B1
R114	Resistor	22 Ω 63 mW 0603
R115	Resistor	270 Ω 63 mW 0603
R116	Resistor	470 Ω 63 mW 0603
R117	Resistor	270 Ω 63 mW 0603
R118	Resistor	470 Ω 63 mW 0603
R119	Inductor	TDK Part No. ACM3225-102-2P
T101	Transformer	OMRON 03-7104
X101	Connector	OMRON XW7D-PB4-R

Table 12: BOM CompoNet Circuit

Optional Bill of Material

C105	Ceramic Capacitor	CKSS22UV6.3-0805
C106	Ceramic Capacitor	22 µF 6.3 V 0805
C107	Ceramic Capacitor	22 µF 6.3 V 0805
R120	Ferrite	100 Ω 100 MHz 1 A Würth Elektronik Part No. 74279207
T102	DC/DC Step-Up	PEAK part no. PSD-3R305S
T103	Voltage Regulator	NCP500SN50T1

Table 13: Optional BOM CompoNet Circuit

More about CompoNet circuits can be found on:

→ Page 118 chapter 4.11 Fieldbus Interface

→ Page 123 chapter 4.11.4 CompoNet Interface

2.7 DeviceNet

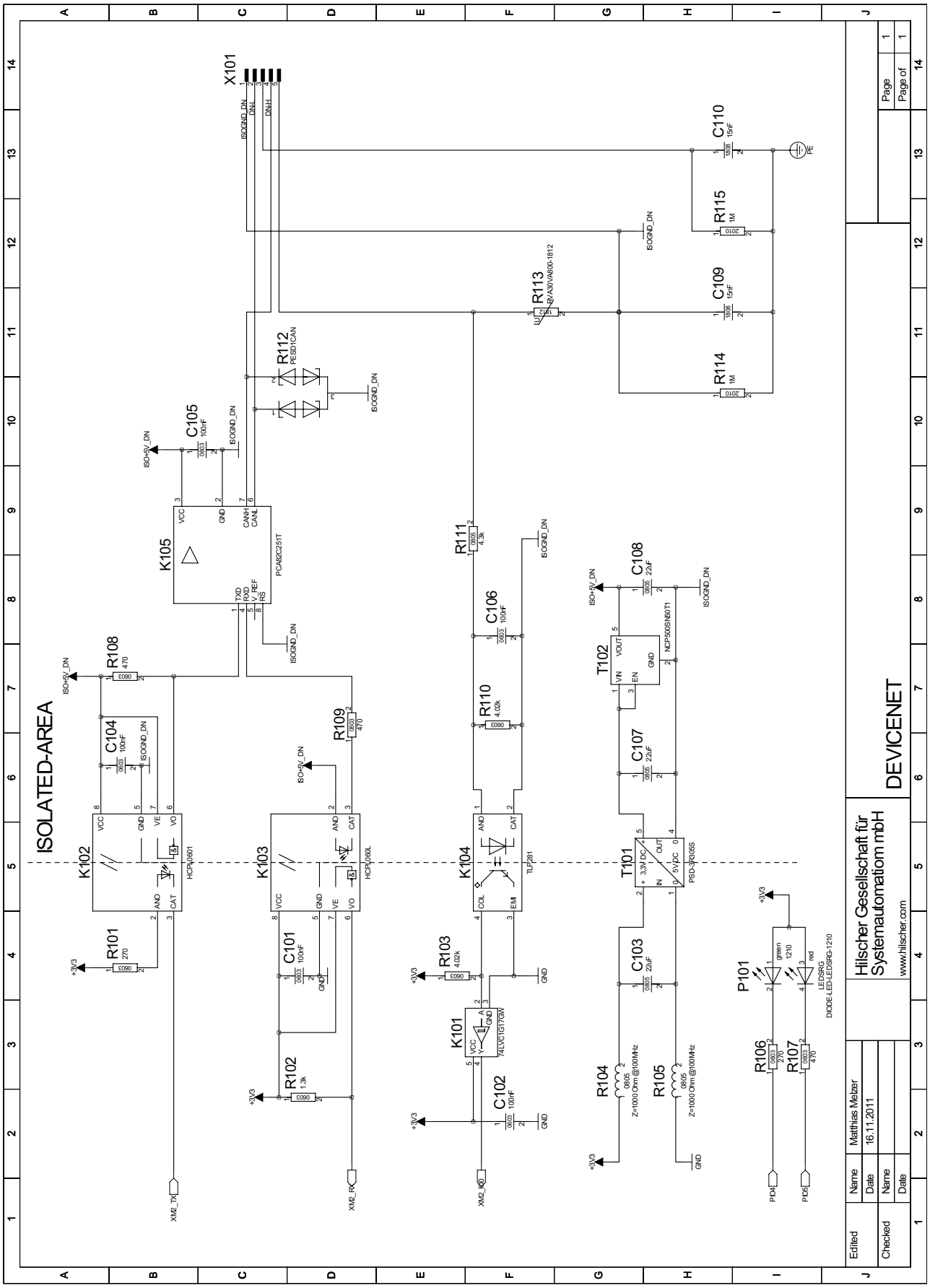


Figure 29: DeviceNet Circuit

Bill of Materials

REF DES	PART TYPE	PART NAME
C101	Ceramic Capacitor	100 nF 25 V 0603
C102	Ceramic Capacitor	100 nF 25 V 0603
C103	Ceramic Capacitor	22 µF 6.3 V 0805
C104	Ceramic Capacitor	100 nF 25 V 0603
C105	Ceramic Capacitor	100 nF 25 V 0603
C106	Ceramic Capacitor	100 nF 25 V 0603
C107	Ceramic Capacitor	22 µF 6.3 V 0805
C108	Ceramic Capacitor	22 µF 6.3 V 0805
C109	Ceramic Capacitor	15 nF 1000 V 1808
C110	Ceramic Capacitor	15 nF 1000 V 1808
K101	Schmitt Trigger	NXP Semiconductors part no. 74LVC1G17GW
K102	Optocoupler	HCPL0601
K103	Optocoupler	HCPL060L
K104	Photocoupler & Photo-Transistor	TLP281
K105	Transceiver	PCA82C251T
P101	LED red/green	HSMF-C155
R101	Resistor	270 Ω 63 mW 0603
R102	Resistor	1.3 kΩ 63 mW 0603
R103	Resistor	4.02 kΩ 63 mW 0603
R104	Ferrite	100 Ω 100 MHz 1 A Würth Elektronik Part No. 74279207
R105	Ferrite	100 Ω 100 MHz 1 A Würth Elektronik Part No. 74279207
R106	Resistor	270 Ω 63 mW 0603
R107	Resistor	470 Ω 63 mW 0603
R108	Resistor	470 Ω 63 mW 0603
R109	Resistor	470 Ω 63 mW 0603
R110	Resistor	4.02 kΩ 63 mW 0603
R111	Resistor	4.3 kΩ 125 mW 0805
R112	ESD protection diode	PEAK part no. PSD-3R305S
R113	Varistor	Epcos Type: SIOV-CN1812K30G Ord. Code: B72580 V0300K062
R114	Resistor	1M Ω 500 mW 2010
R115	Resistor	1M Ω 500 mW 2010
T101	DC/DC Step-Up	PEAK part no. PSD-3R305S
T102	Voltage Regulator	NCP500SN50T1
X101	Connector	MC1,5/5-G-3,81

Table 14: BOM DeviceNet Circuit

More about DeviceNet circuits can be found on:

→ Page 118 chapter 4.11 Fieldbus Interface

→ Page 124 chapter 4.11.5 DeviceNet Interface

→ Page 126 chapter 4.11.7 Fieldbus Status LEDs

2.8 PROFIBUS

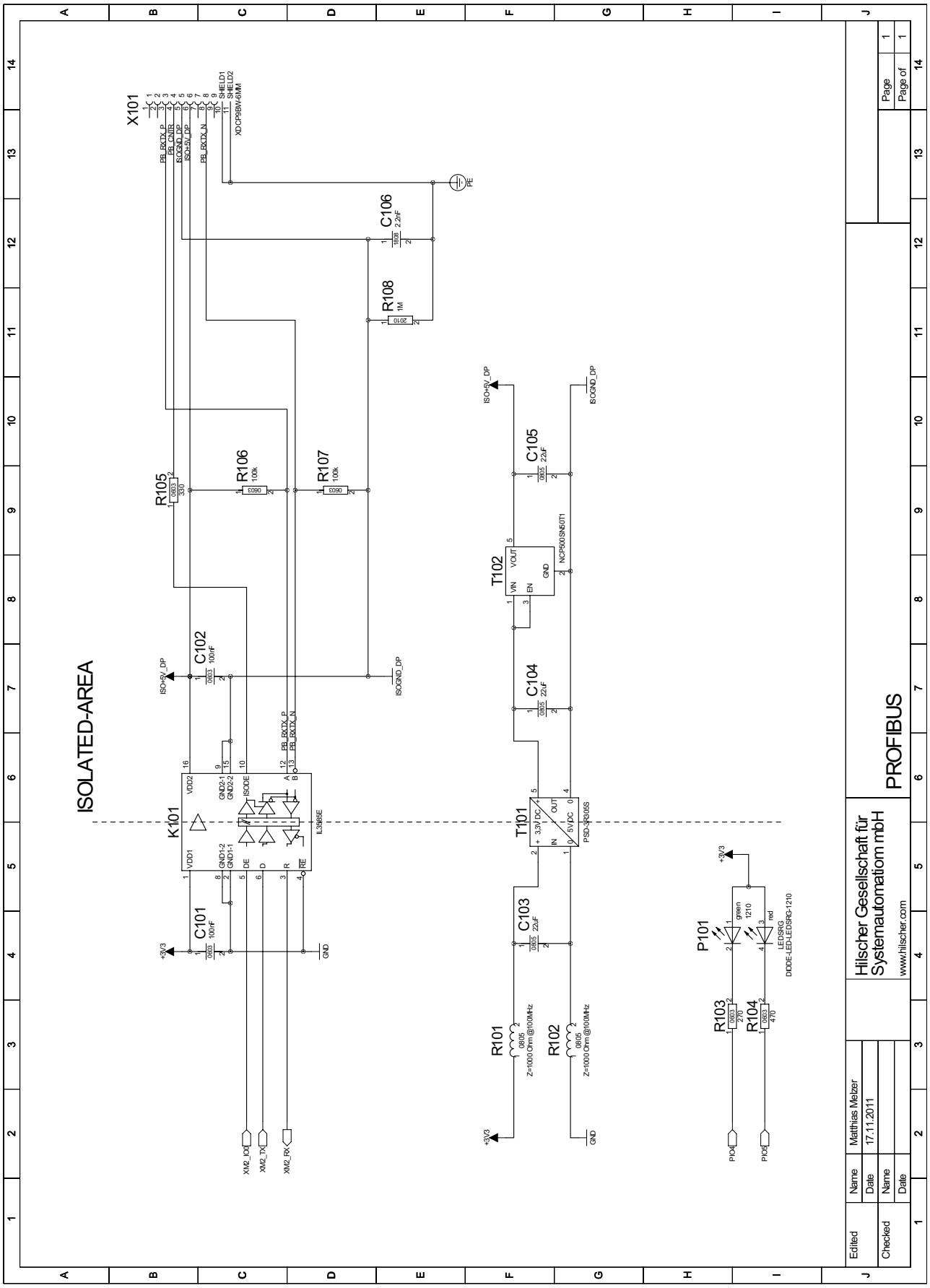


Figure 30: PROFIBUS Circuit

Bill of Material

REF DES	PART TYPE	PART NAME
C101	Ceramic Capacitor	100 nF 25 V 0603
C102	Ceramic Capacitor	100 nF 25 V 0603
C103	Ceramic Capacitor	22 µF 6.3 V 0805
C104	Ceramic Capacitor	22 µF 6.3 V 0805
C105	Ceramic Capacitor	22 µF 6.3 V 0805
C106	Ceramic Capacitor	2.2 µF 1000 V 1808
K101	Transceiver	IL3585E
P101	LED red/green	HSMF-C155
R101	Ferrite	100 Ω 100 MHz 1 A Würth Elektronik Part No. 74279207
R102	Ferrite	100 Ω 100 MHz 1 A Würth Elektronik Part No. 74279207
R103	Resistor	270 Ω 63 mW 0603
R104	Resistor	470 Ω 63 mW 0603
R105	Resistor	330 Ω 63 mW 0603
R106	Resistor	100 kΩ 63 mW 0603
R107	Resistor	100 kΩ 63 mW 0603
R108	Resistor	1M Ω 500 mW 2010
T101	DC/DC Step-Up	PEAK part no. PSD-3R305S
T102	Voltage Regulator	NCP500SN50T1
X101	D-Sub9 Female	SUYIN USA Part No. 070212FR009G200ZU

Table 15: BOM PROFIBUS Circuit

More about PROFIBUS circuits can be found on:

→ Page 118 chapter 4.11 Fieldbus Interface

→ Page 125 chapter 4.11.6 PROFIBUS Interface

→ Page 126 chapter 4.11.7 Fieldbus Status LEDs

2.9 MMC/SD-Card SPI-Circuit

The following figure shows the standard circuit to connect MMC/SD-Card with netX100/500. It is important GPIO15 Pin V13 of netX100/500 to connect to the MMC/SD-insert contact. Otherwise it is a normal SPI connection with chip select 1 on netX100/500 V17 pin.

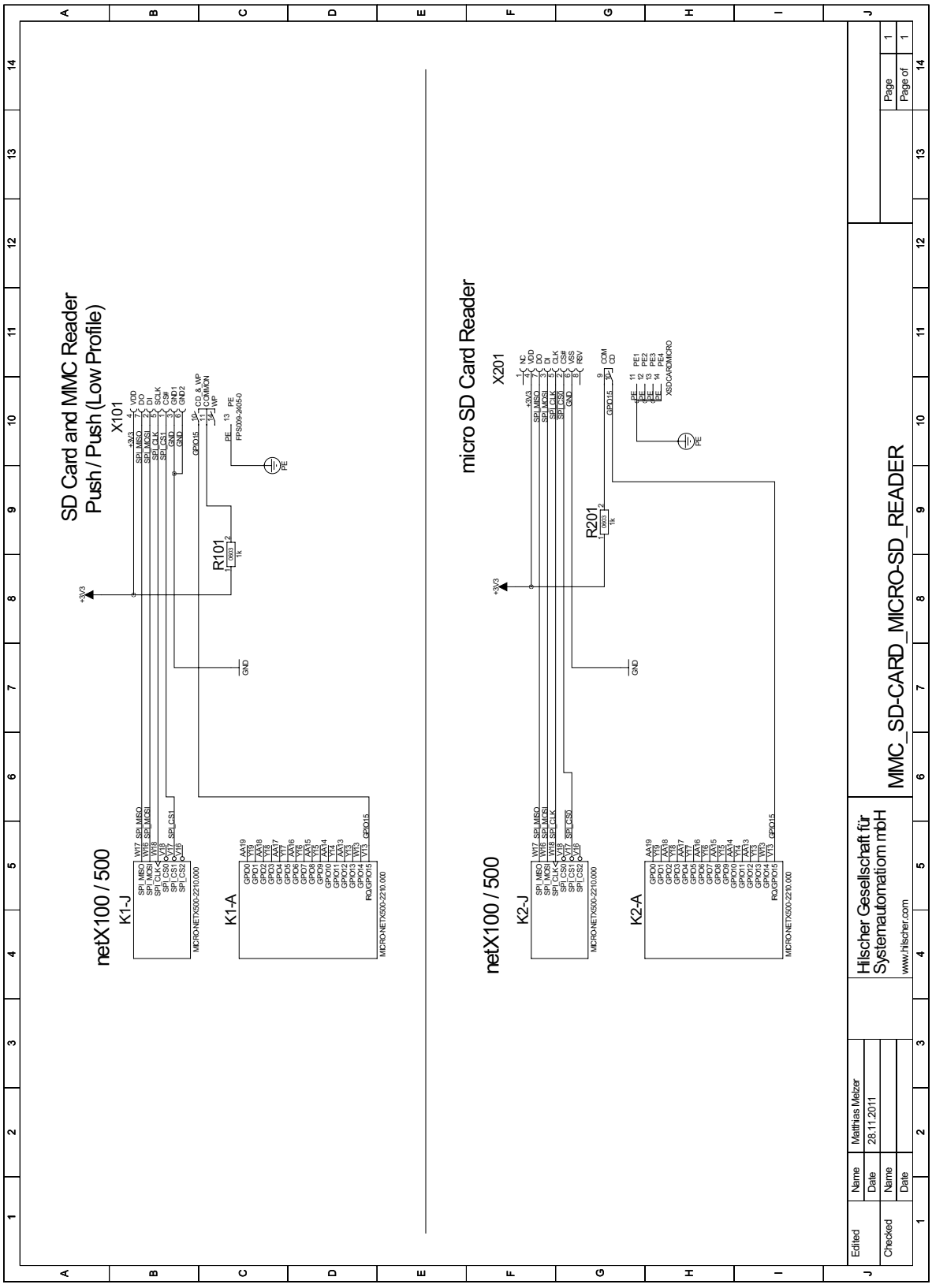


Figure 31: netX100/500 MMC/SD-Card SPI Circuit

Bill of Materials

MMC/SD Card

REF DES	PART TYPE	PART NAME
K1	Microcontroller	netX100/500
R101	Resistor	1 kΩ 63 mW 0603
X101	MMC/SD Card reader	Yamaichi Electronics Part No. FPS009-2405-0

Table 16: BOM MMC/SD Card Circuit

Bill of Materials

microSD

REF DES	PART TYPE	PART NAME
K2	Microcontroller	netX100/500
R201	Resistor	1 kΩ 63 mW 0603
X201	microSD Card reader	Amphenol Part No. GTFP08121HEU

Table 17: BOM microSD Circuit

More about MMC/SD-Card circuits can be found on:

→ Page 75 chapter 4.6.1.2 MMC/SD Card

2.10 UART ⇔ RS232

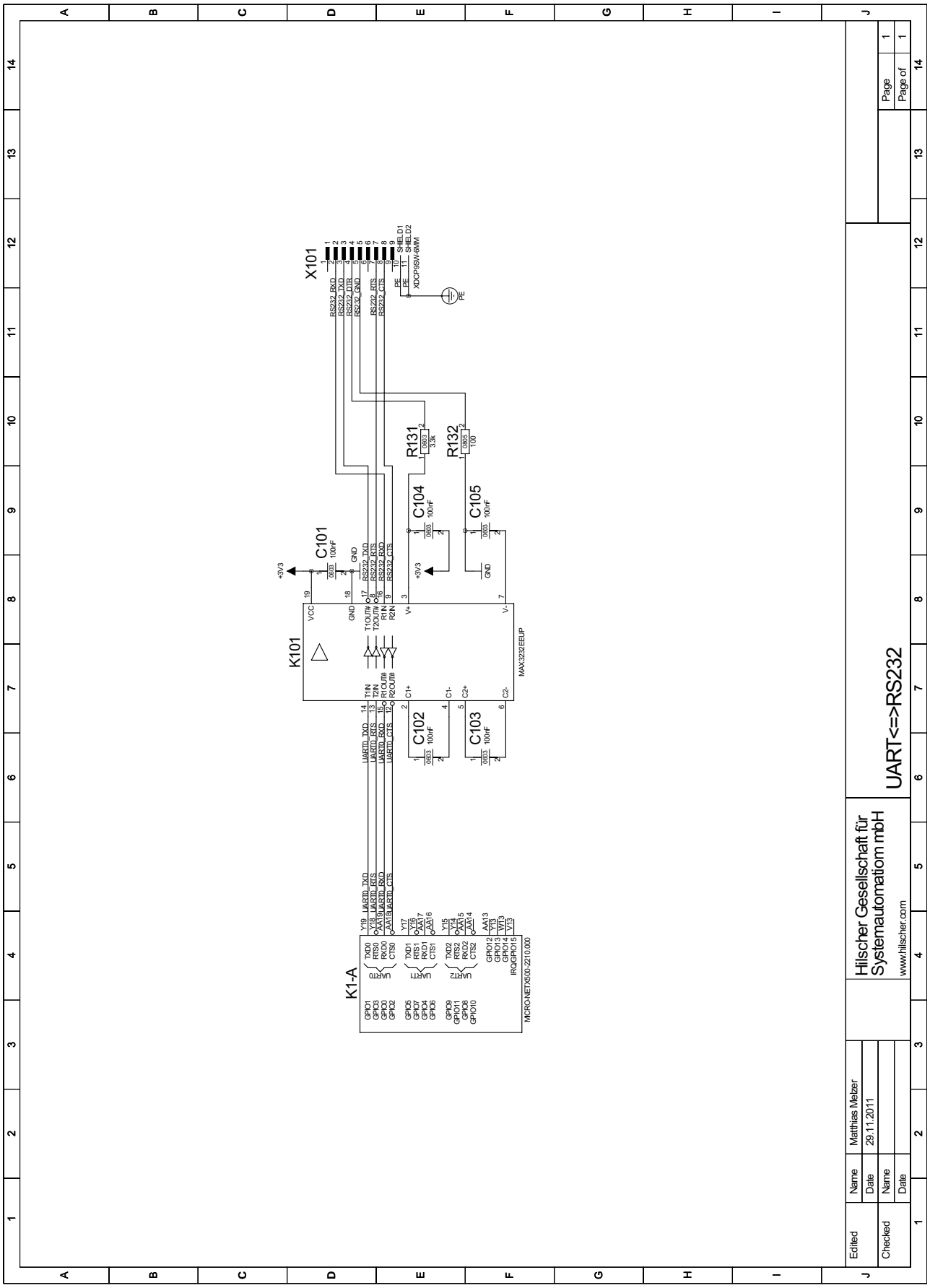


Figure 32: UART ⇔ RS232 Circuit

Bill of Material

REF DES	PART TYPE	PART NAME
C101	Ceramic Capacitor	100 nF 25 V 0603
C102	Ceramic Capacitor	100 nF 25 V 0603
C103	Ceramic Capacitor	100 nF 25 V 0603
C104	Ceramic Capacitor	100 nF 25 V 0603
C105	Ceramic Capacitor	100 nF 25 V 0603
K1	Microcontroller	netX100/500
K101	Transceiver	MAX3232EEUP
R131	Resistor	3.3 kΩ 63 mW 0603
R132	Resistor	100 Ω 125 mW 0805
X101	D-Sub9 Female	SUYIN USA Part No. 070212FR009G200ZU

Table 18: BOM UART ⇔ RS232

More about UART circuits can be found on:

→ Page 97 chapter 4.8 UARTs

2.11 USB Device Mode

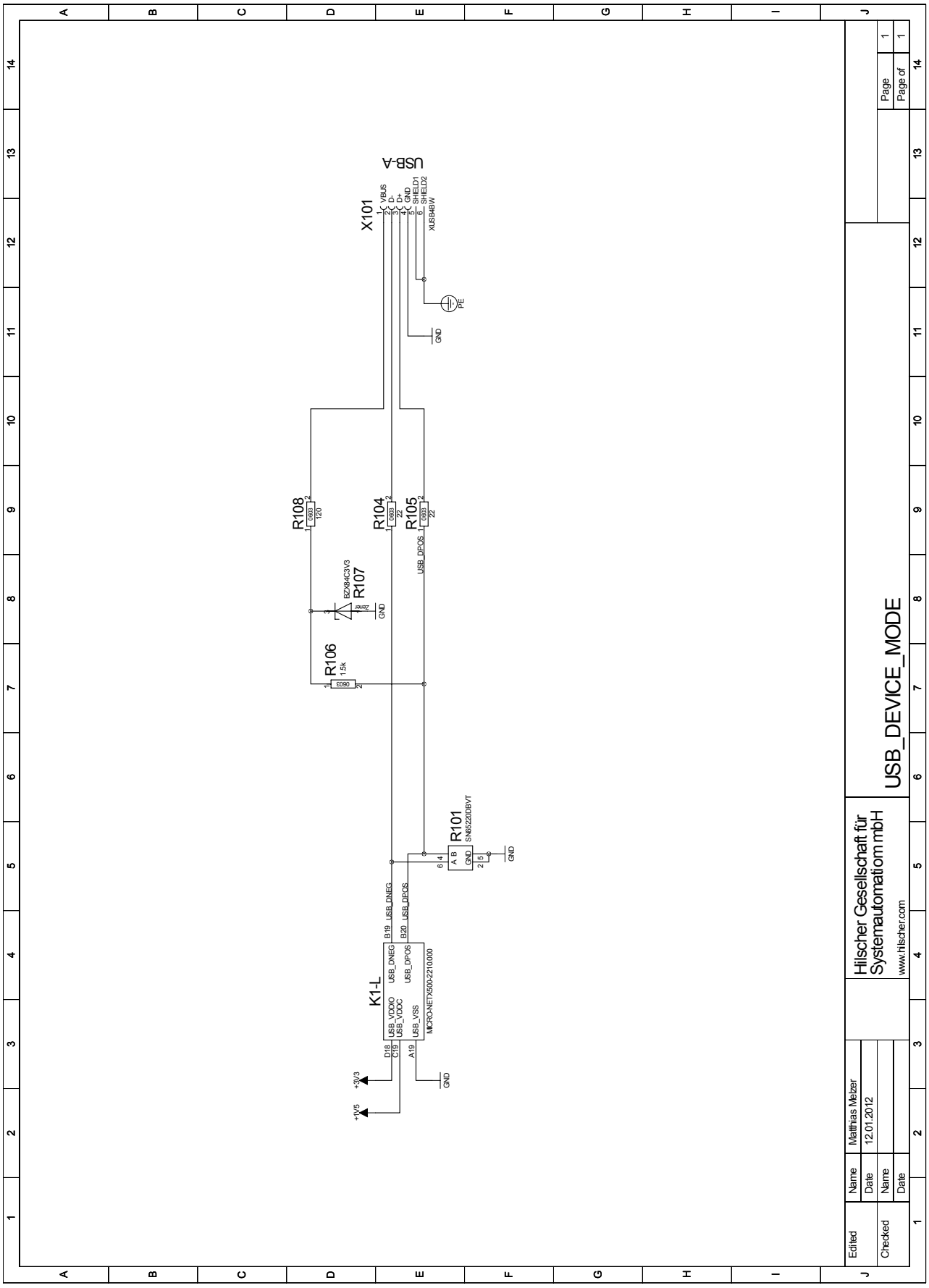


Figure 33: USB Device Mode Circuit

Bill of Material

REF DES	PART TYPE	PART NAME
K1	Microcontroller	netX100/500
R101	Diode	SN65220DBVT
R104	Resistor	22 Ω 63 mW 0603
R105	Resistor	22 Ω 63 mW 0603
R106	Resistor	1.5 kΩ 63 mW 0603
R107	Diode	BZX84C3V3
R108	Resistor	120 Ω 63 mW 0603
X101	Connector	USB B Connector

Table 19: BOM USB Device Circuit

More about USB circuits can be found on:

→ Page 98 chapter 4.9 USB

2.12 USB Host Mode

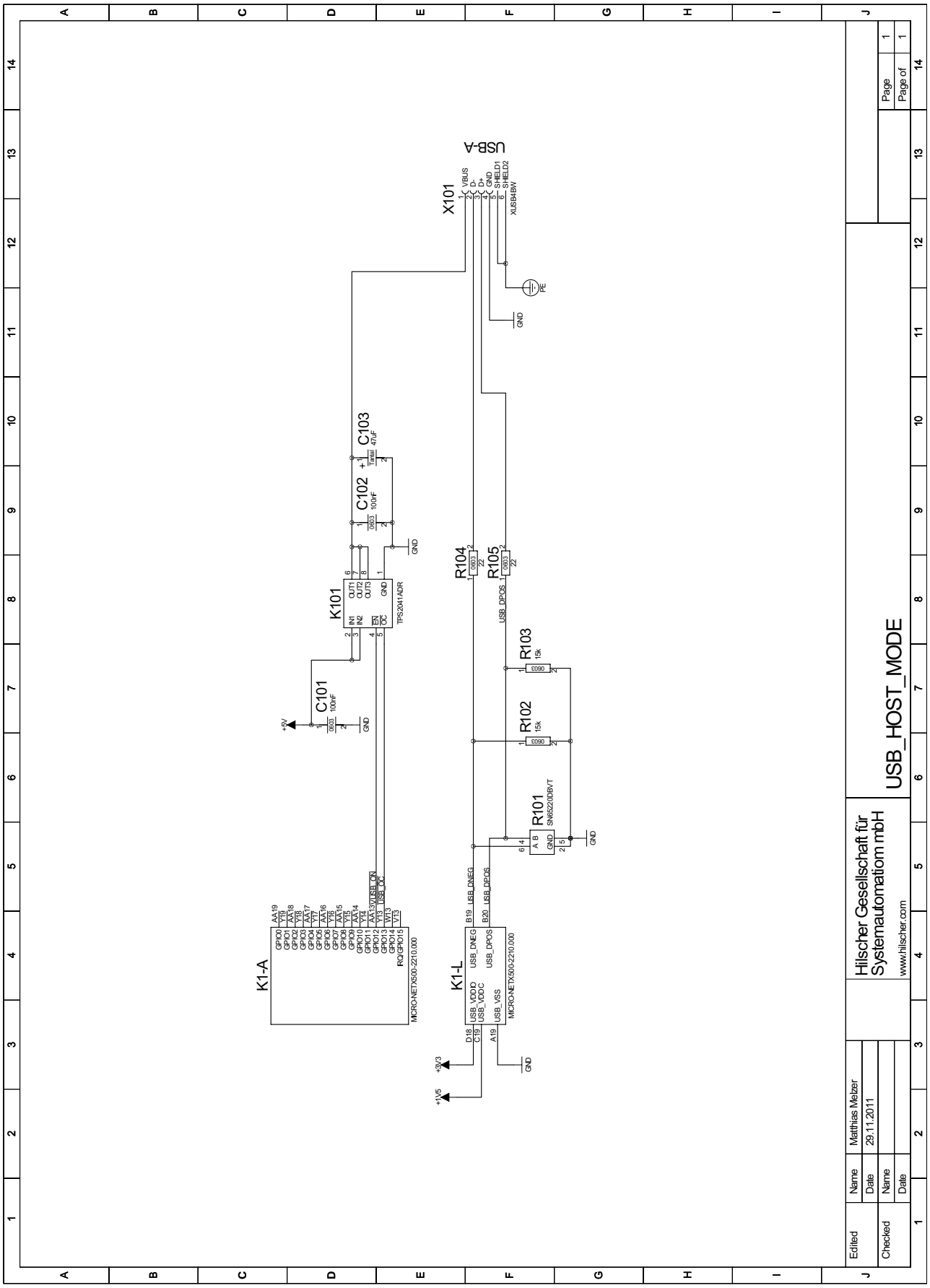


Figure 34: USB Host Mode Circuit

Bill of Material

REF DES	PART TYPE	PART NAME
C101	Ceramic Capacitor	100 nF 25 V 0603
C102	Ceramic Capacitor	100 nF 25 V 0603
C103	Tantalum Capacitor	47 µF 10 V
K1	Microcontroller	netX100/500
K101	Current-Limited Power-Distribution Switch	Texas Instruments Part No. TPS2041 ADR
R101	Diode	SN65220DBVT
R102	Resistor	15 kΩ 63 mW 0603
R103	Resistor	15 kΩ 63 mW 0603
R104	Resistor	22 Ω 63 mW 0603
R105	Resistor	22 Ω 63 mW 0603
X101	Connector	USB A Connector

Figure 35: BOM USB Host Mode Circuit

More about USB circuits can be found on:

→ Page 98 chapter 4.9 USB

2.13 Embedded Trace Macrocell (ETM)

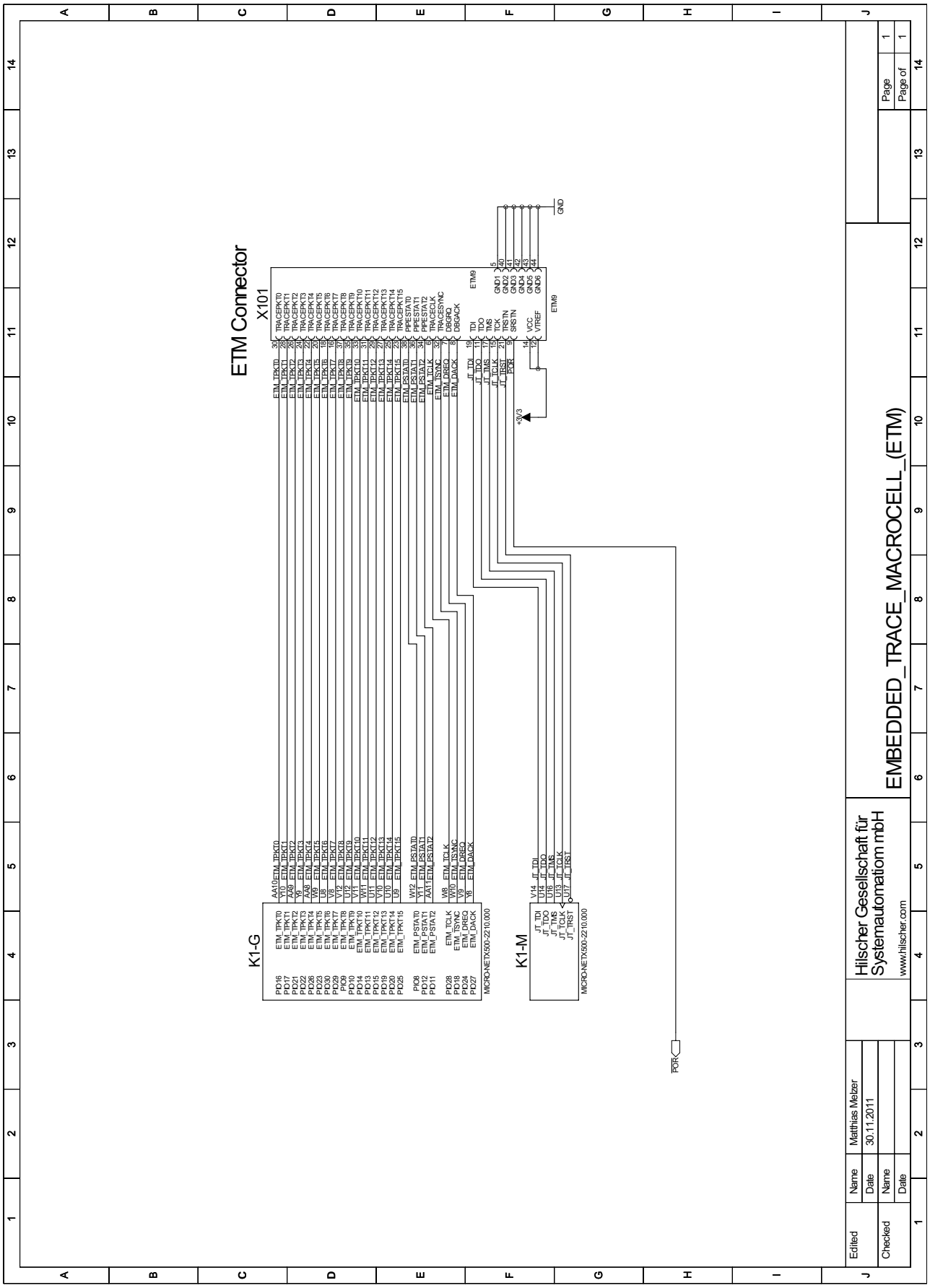


Figure 36: ETM Circuit

Bill of Materials

REF DES	PART TYPE	PART NAME
K1	Microcontroller	netX100/500
X101	Connector	TE connectivity Part No. 2-767004-2

Table 20: BOM ETM Circuit

More about debug and test interface circuits can be found on:

→ Page 70 chapter 4.5.1 JTAG Interface

→ Page 71 chapter 4.5.2 ETM Interface

→ Page 72 chapter 4.5.3 Boundary Scan

2.14 LCD Interface

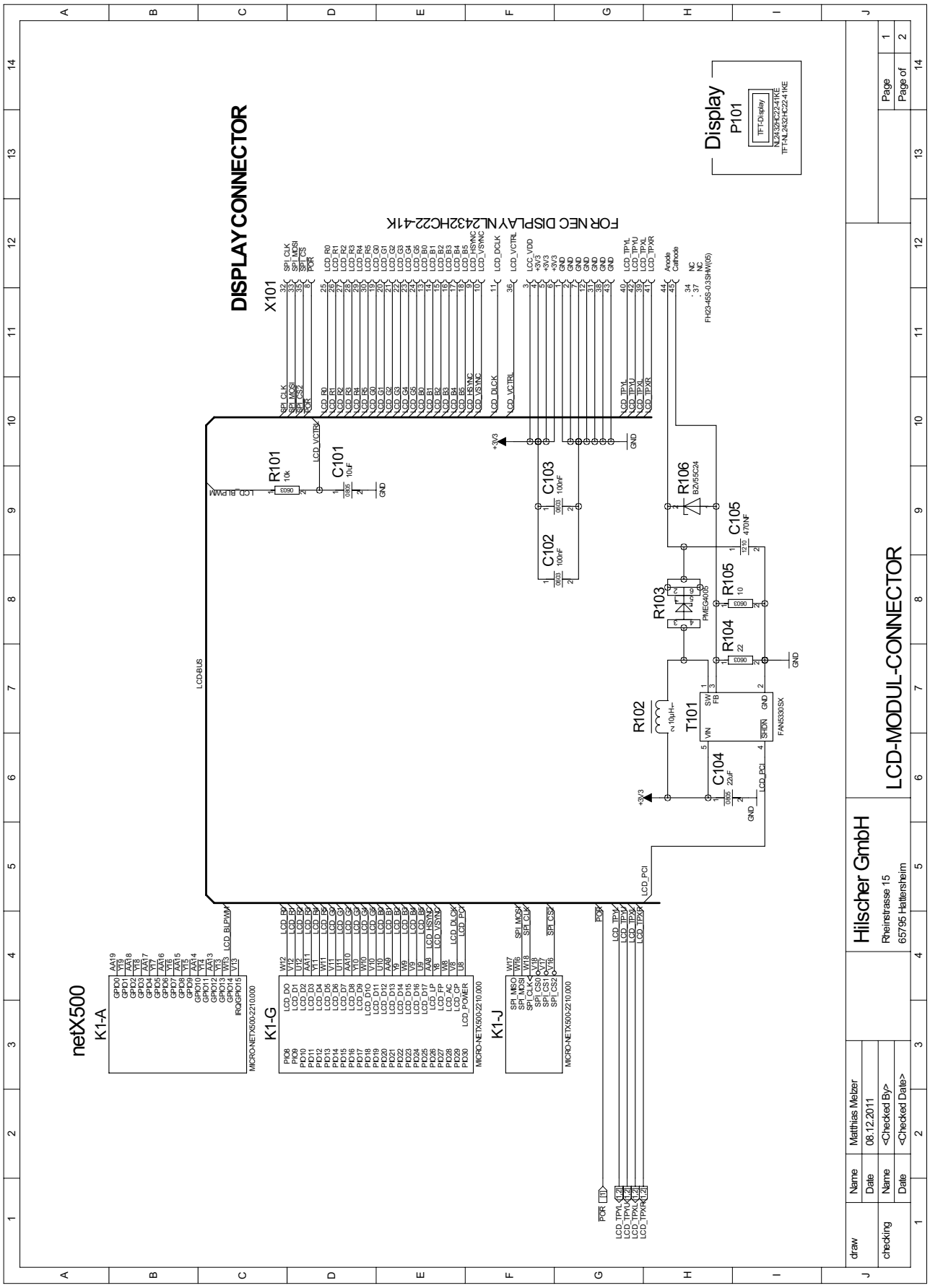


Figure 37: LCD Interface Circuit

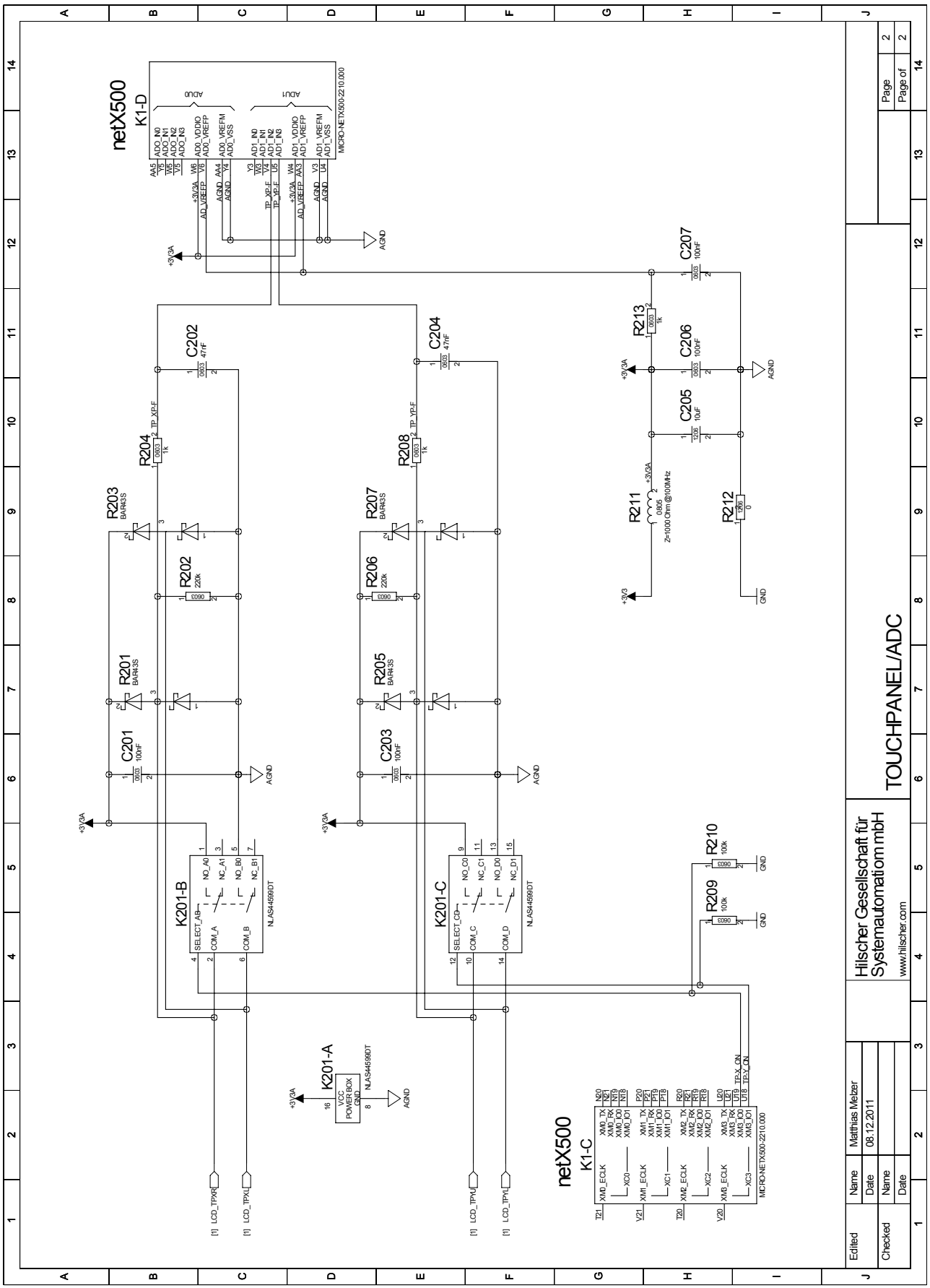


Figure 38: Touch Panel Circuit

Bill of Material

Page 1

REF DES	PART TYPE	PART NAME
C101	Ceramic Capacitor	10 µF 6.3 V 0805
C102	Ceramic Capacitor	100 nF 25 V 0603
C103	Ceramic Capacitor	100 nF 25 V 0603
C104	Ceramic Capacitor	22 µF 6.3 V 0805
C105	Ceramic Capacitor	470 nF 50 V 1210
K1	Microcontroller	netX500
P101	LC-Display	NEC NL2432HC22-41K
R101	Resistor	10 kΩ 63 mW 0603
R102	Inductor	Sumida CR32NP-100K
R103	Diode	Philips Semiconductors PMEG4005AEV
R104	Resistor	22 Ω 63 mW 0603
R105	Resistor	10 Ω 63 mW 0603
R106	Diode	Philips Semiconductors BZV55C24
T101	Step-up	Fairchild Semiconductor FAN5330
X101	Connector	Hirose Connectors Part No. FH23-45S-0.3SHW(05)

Table 21: BOM LCD Interface Circuit

Page 2

REF DES	PART TYPE	PART NAME
C201	Ceramic Capacitor	100 nF 25 V 0603
C202	Ceramic Capacitor	47nF 50 V 0603
C203	Ceramic Capacitor	100 nF 25 V 0603
C204	Ceramic Capacitor	47nF 50 V 0603
C205	Ceramic Capacitor	10 µF 10 V 1206
C206	Ceramic Capacitor	100 µF 25 V 0603
C207	Ceramic Capacitor	100 nF 25 V 0603
K201	IC-Switch	NLAS44599DT
R201	Diode	ST Microelectronics BAR43S
R202	Resistor	220 kΩ 63 mW 0603
R203	Diode	ST Microelectronics BAR43S
R204	Resistor	1 kΩ 63 mW 0603
R205	Diode	ST Microelectronics BAR43S
R206	Resistor	220 kΩ 63 mW 0603
R207	Diode	ST Microelectronics BAR43S
R208	Resistor	1 kΩ 63 mW 0603
R209	Resistor	100 kΩ 63 mW 0603
R210	Resistor	100 kΩ 63 mW 0603
R211	Inductor	Würth Elektronik Part No. 74279205
R212	Resistor	0 Ω 250mW 1206
R213	Resistor	1 kΩ 63 mW 0603

Table 22: BOM Touch Panel Circuit

More about LCD and touch panel circuits can be found on:

→ Page 134 chapter 4.15 LCD Interface

→ Page 135 chapter 4.16 Touch Panel Interface

3 Resource Overview

The following tables list netX Hardware Resources and functions and provide information on existing software support for these features. “No driver available” means, that Hilscher does currently not provide a driver or special functions for easy access to the corresponding resource, however this resource may of course still be used if the user develops the appropriate code by himself or integrates third party products (e.g. Flash File System for parallel FLASH).

3.1 rcX Operating System

Resource/Functionality	netX	Loadable Firmware	Linkable Object Modules
USB Device	100/500	for firmware update	for firmware update/debug
USB Host	100/500	not supported	no driver available
UART0	100/500	for firmware update	for firmware update/debug
UART1	100/500	Modbus RTU	supported
UART2	100/500	Modbus RTU	supported
SDRAM	100/500	required (min. 8MB)	required (min. 8MB) for standard application
Secure Memory	100/500	required (min. required see Table 25)	required (min. required see Table 25)
SPI Flash	100/500	required (minimum size see separate list)	required (minimum size see separate list)
MMC/SD Card	100/500	not supported	no driver available
Parallel FLASH	100/500	not supported	- no FLASH File System - only limited components
LC-Display	500	not supported	no driver available
Fieldbus Slave (1 Channel)	100/500	see Table 25	see Table 25
Fieldbus Master (1 Channel)	100/500	see Table 25	see Table 25
Ethernet Ports	100/500	RTE protocols (see separate list)	Standard Ethernet and RTE protocols (see separate list)
Host Interface	100/500	DPM interface	DPM, Extension Bus, PCI
Real Time Clock	500	not supported	no driver available
Gateway Functionality	100/500	not supported	user programmable
2 Channel Fieldbus	100/500	not supported	supported
AD Converter	100/500	not supported	no driver available
PWM Interface	100/500	not supported	no driver available
Encoder Interface	100/500	not supported	no driver available

Table 23: List of Resources – rcX OS

3.2 Third Party Operating Systems

Resource/Functionality	netX	Linux BSP	CE BSP	VxWorks BSP
USB Device	100/500	no driver available	supported	no driver available
USB Host	100/500	no driver available	supported (max. 7 pipes)	no driver available
UART0	100/500	- required by uboot - remote console or standard serial port	Debug or Standard port - required by eboot	supported
UART1	100/500	supported	supported	supported
UART2	100/500	supported	supported	supported
SDRAM	100/500	required (min. 32 MB with LCD)	required (min. 32 MB with LCD)	required (min. 8 MB, rec. 16 MB)
Secure Memory	100/500	Ethernet MAC Addresses	supported	Ethernet MAC Addresses and user zone
SPI Flash	100/500	no driver available	no driver available	- Flash File System (for AT45DB321C) - wear leveling provided
MMC/SD Card	100/500	supported	supported	no driver available
LC-Display	500	supported	supported	no driver available
Fieldbus Slave (1 Chanel)	100/500	no stacks available	no stacks available	no stacks available
Fieldbus Master (1 Chanel)	100/500	no stacks available	no stacks available	no stacks available
Ethernet Ports	100/500	Standard Ethernet only	Standard Ethernet only	Standard Ethernet only
Host Interface	100/500	PCMCIA interface (no support of bootable media)	Extension Bus or PCMCIA interface (no support of bootable media)	no driver available
Real Time Clock	500	supported	supported	no driver available
AD-Converter	100/500	no driver available	driver for Touch panel only	no driver available
PWM Interface	100/500	no driver available	no driver available	no driver available
Encoder Interface	100/500	no driver available	no driver available	no driver available

Table 24: List of Resources – 3rd Party OS

3.3 Memory Requirements of Hilscher Stacks

The following table lists all field bus and RTE protocols that are currently (status of February 2011) available as Loadable Firmware or Loadable Object Modules from Hilscher, along with the required size of the SPI FLASH, holding the appropriate firmware. The FLASH size information is based on the code sizes of the current Firmware Releases, include the additional memory required for the Hilscher second stage loader (currently 52 K) and the FLASH disk, leave some headroom for future extensions and are generally rounded up to the next available FLASH size step. If Linkable Object Modules are to be used, users must of course consider the additional memory required for their user application (where applicable).

All listed protocols also require different amounts of SDRAM, however the smallest available SDRAM components are meanwhile 8 MB anyway. This amount of memory meets the requirements of all current protocols and is hence indicated as minimum SDRAM size for netX designs.

Loadable Firmware/LOM	netX	SPI FLASH size
ASi Master	100/500	2MB
CANopen Slave	100/500	2MB
CANopen Master	100/500	2MB
DeviceNet Slave	100/500	2MB
DeviceNet Master	100/500	2MB
CC-Link Slave	100/500	2MB
PROFIBUS Slave	100/500	2MB
PROFIBUS Master	100/500	2MB
EtherCAT Slave	100/500	2MB
EtherCAT Master	100/500	2MB
Ethernet/IP Adapter	100/500	2MB
Ethernet/IP Scanner	100/500	2MB
Modbus RTU/TCP	100/500	2MB
POWERLINK Slave (with integrated. Hub)	100/500	2MB
PROFINET RT/IRT Slave (with integrated. Switch)	100/500	4MB
PROFINET RT Master (with integrated. Switch)	100/500	2MB
SERCOS III Slave	100/500	2MB

Table 25: FLASH Sizes for Hilscher Stacks

4 Standard Circuits

4.1 RDY/RUN Pins, SYS LED

The netX100 and netX500 provide two dedicated I/O pins that are used for up to two different purposes. These pins are named RDY and RUN and operate as inputs after reset. The first stage boot loader residing in the ROM of the netX chips checks the logic levels on these pins and enters certain boot modes, depending on these levels. After that, the first stage loader configures these pins as outputs, which are used to display status information. Once a firmware is started, it has complete control over these pins and their function may then be completely application specific.

Note: In netX schematics, the RDY and RUN pins are usually shown with a negating circle. However, the polarity of these pins (when used as outputs) depends on a register setting. Besides the two bits that enable the output driver (pin configured as output) and set the level of the pin, there is a third bit for each pin that determines the polarity (active high or active low). So it actually depends on these polarity bits, if the (output-) pins are active low or active high!

Further these polarity bits do not affect the pins when used as inputs (RDY and RUN inputs are never inverted).

For historical reasons (the active low signals RDY# and RUN# were already defined for the EC1 based devices) the RDY and RUN pins are however always shown as active low.

For displaying system status, a system LED (dual LED or two single LEDs) is defined:

LED	Color	Description
RDY	Yellow	netX with operating system is running
RUN	Green	User application is running without errors

Table 26: RDY/RUN LED Status

Basically, designers could use LEDs with other colors, however it is recommended to use the Hilscher definition (especially when interpreting blink codes for troubleshooting it is helpful if customer and support see the same colors).

The most flexible circuit for netX100/500 designs is shown in the following schematic. It allows setting all possible boot modes of the netX100/500 and can usually be found on evaluation hardware:

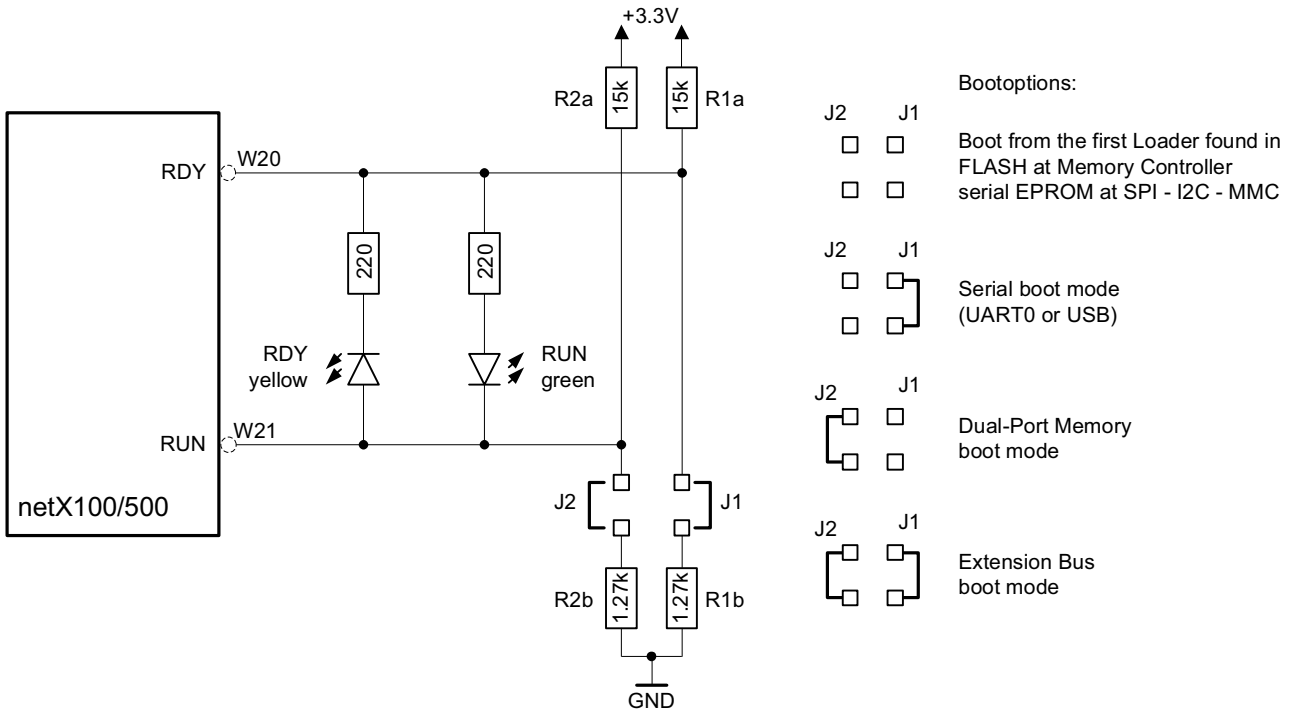


Figure 39: netX100/netX500 RDY/RUN Circuit

FAQ - Frequency Answered Questions

Q1: I'm not building an evaluation board. Do I really need all these jumpers?

A1: Well, that depends on the way your design intends to load its firmware. Standard design that use either a serial or parallel FLASH or an MMC card for storing the firmware, do not need J2 and R2b.

Designs that use a FLASH connected to the Extension Bus (a hardly used option) or Dual-Port Memory (DPM) boot mode will require both jumpers/pull-down resistors.

Q2: But I'm always using DPM boot mode, can't I just omit J1 and R1b and replace J2 by a wire?

A2: You could basically do that, but keep in mind, that you will then not be able to activate the serial boot mode for test or debug purposes.

Q3: I'm using a serial or parallel FLASH to boot from. Do I need any jumpers at all?

A3: Unless you want to program the FLASH before mounting and remove it for reprogramming every time you want to change the firmware, it is strongly recommended to have J1 (or a push button/switch) and R1b on your design. This allows activating the serial boot mode, which will then allow to (re-)flash the firmware.

Q4: Your reference schematics and some documentation do not show any pull-up resistors on RDY and RUN, further the netX100/500 has internal pull-ups on RDY and RUN, so do I really need these 15 kΩ pull-ups?

A4: Most designs actually work with the simplified circuit that relies on the internal pull-ups on the RDY and RUN pins, hence omit the pull-ups R1 A and R2a and use 10 kΩ pull-down resistor(s). However, due to the high tolerance of internal (on-chip) pull-up/pull-down resistors and due to the antiparallel LEDs that let the RDY and RUN signals influence each other, chances are, that such designs may not enter the desired boot mode, due to invalid logic levels on RDY and RUN.

Q5: I want to control the netX boot mode by applying the appropriate logic levels through external (active) components to the RDY and RUN pins. Is that possible?

A5: No. Shortly after detection of the desired boot mode by the boot loader, the loader or the firmware will use the RDY and RUN pins as outputs to drive the System LED(s) and will most likely drive against the logic levels applied by the external circuit. Besides that the System LED(s) will then not work, this will drive short circuit currents through the netX RDY/RUN pins that may damage the chip!

Q6: But I'm using push-buttons for the boot mode setting and would like to debounce the signals by Schmitt trigger buffers.

A6: This wouldn't make any sense at all. Since the button(s) must already be pressed when performing a reset or powering up the system, the signals will already be stable when the first stage loader checks them.

4.2 Secure EEPROM

The secure EEPROM available for netX controllers can hold licensing information, MAC addresses and other information. While it is generally recommended to design in this component, it is mandatory for all designs that are to run any Hilscher Master stacks (e.g. PROFIBUS Master, EtherCAT Master) or use the PCI interface of the netX500. A detailed Application Note explaining the purpose and use of the Secure EEPROM is available on the Hilscher website.

On the netX100/500, the secure EEPROM is connected to the I²C interface, as shown by the following schematic:

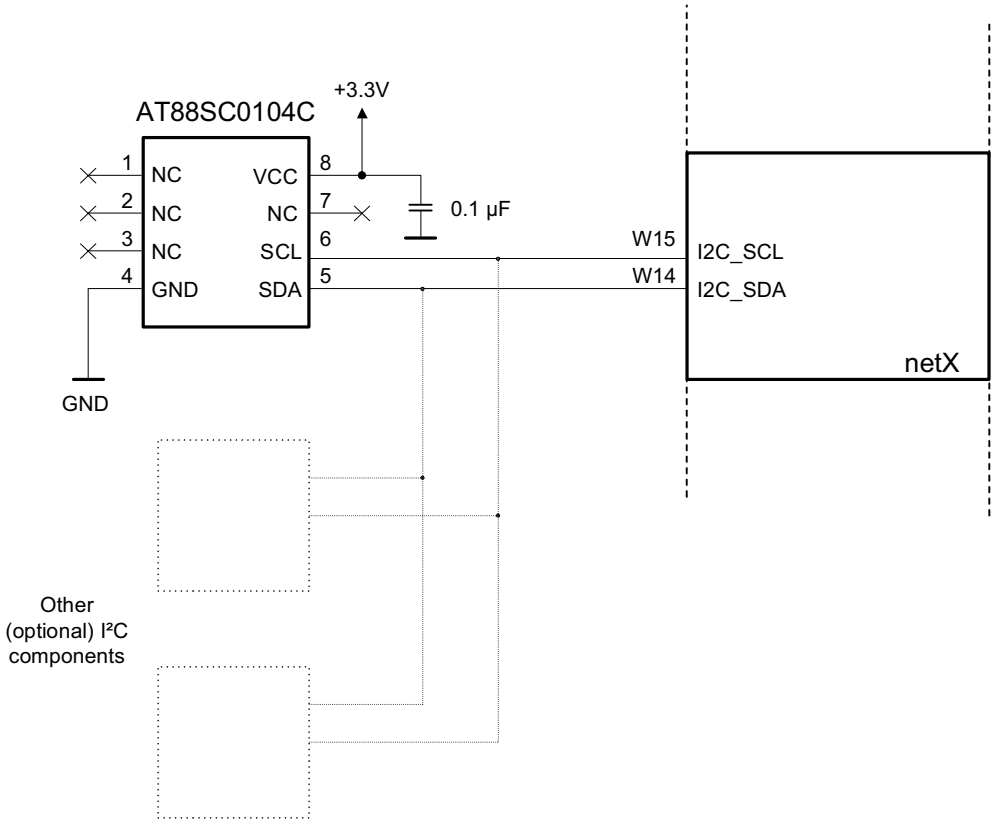


Figure 40: Sample Schematic, netX100/500 Secure Memory

The netX secure memory can be connected parallel to other I²C components as shown in the schematic above. It responds to device addresses starting with 0xB(1011), hence designers have to make sure, that no other connected I²C component uses this address space.

4.3 Crystals, Clock generators

4.3.1 System Clock

The netX100 and netX500 all use either an internal oscillator along with an external crystal or an external oscillator for generating the 25 MHz base clock, which is then stabilized by a PLL which generates all internal Clocks of the chip, except the clock for the internal Real Time Clock (netX500 only), which is covered in the following subchapter 4.3.2.

The following figure shows the clock circuits for the system clock generation:

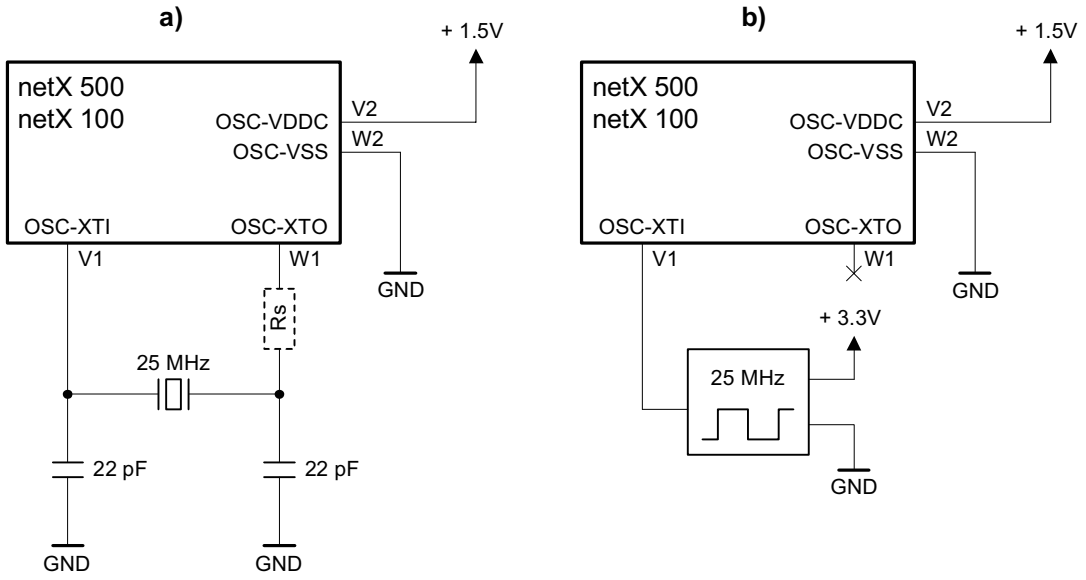


Figure 41: netX100/500 System Oscillator Circuit

The values of the capacitors and the serial Resistor (Rs) depend on the used crystal. When using the same crystal all Hilscher netX products are equipped with, the resistor Rs will not be used (and replaced by a wire), further the capacitors should have a value of 22pF.

If a different crystal is used, the data sheet of the crystal must be consulted to determine the appropriate values.

The Hilscher standard netX system crystal is a **CS10-25.000MAGJ-UT**, manufactured by Citizen.

Alternatively, an external oscillator can be used, which is then connected according to schematic b).

Note: When selecting an external oscillator or a different crystal, it must be provided, that these parts have a frequency of 25 MHz with a maximum tolerance of +/- 100 ppm throughout the complete temperature range, the design will be specified for!

Q1: We already stock crystals or oscillators with a different frequency or higher tolerance, can we also use these parts for the system oscillator?

A1: No. The 25 MHz clock is the base for all other netX clocks and has hence influence on any timing around the netX, like SDRAM timing, Baud Rates, Ethernet timing, etc. Deviating from the specified frequency will most likely result in a system that does not work properly.

4.3.2 Real Time Clock

4.3.2.1 Designs with RTC

The netX500 is equipped with a Real Time Clock that can be powered separately from the rest of the chip and will then continue to run when the system is powered down. The RTC module further provides a backup feature by powering a 16K portion of the netX500 internal RAM that can preserve data.

To avoid uncontrolled access to this part of the RAM as a result of a power fail, the Backup RAM will be isolated when the RTC-POK pin is pulled low. If this feature is to be used, a power supervision circuit must be connected to the main power supply. The supervision circuit must pull the RTC-POK signal low, when the main input voltage drops below a certain level. This of course does only make sense, if the power supplies are designed that way, that in case the input voltage fails, they will continue to deliver stable voltages to the 1,5 V and 3,3 V rails of the netX for a certain time, allowing the isolation process to complete safely before the power fails completely.

The following figure shows the complete circuit with some options:

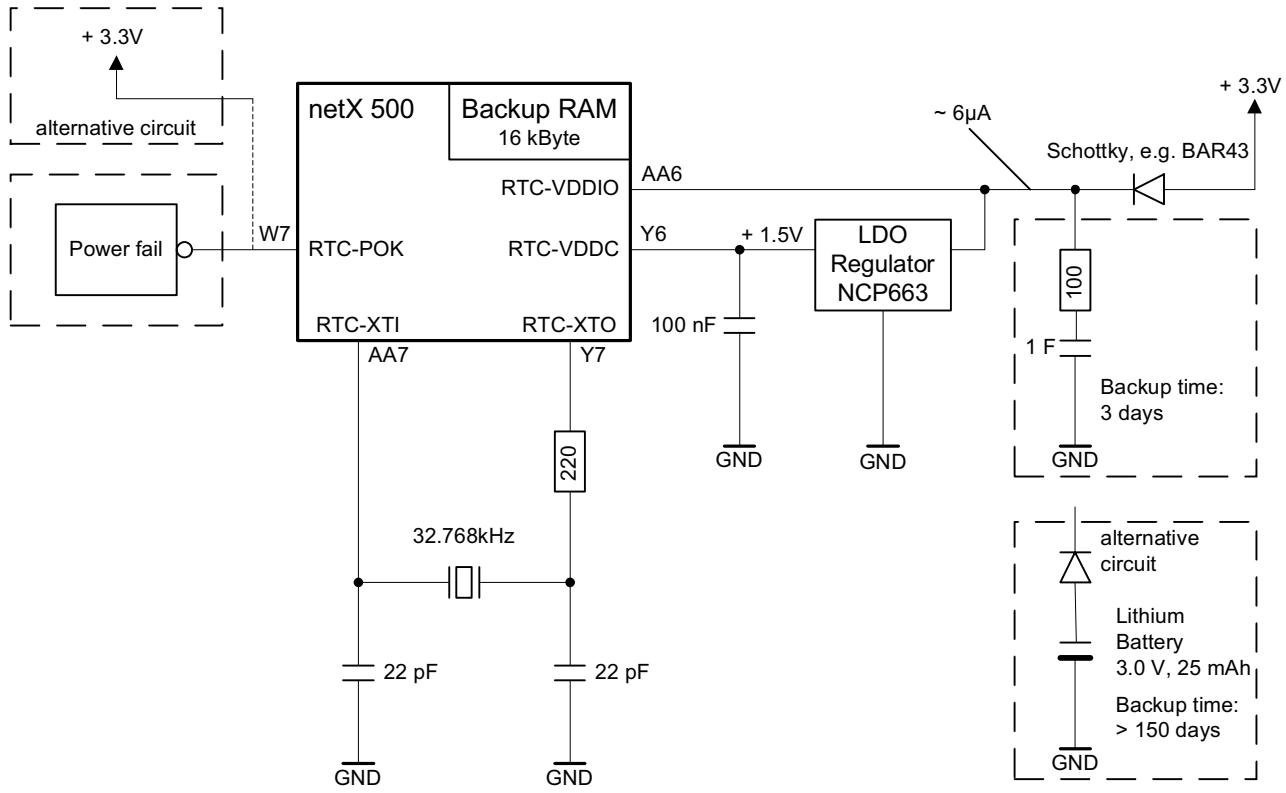


Figure 42: netX500 RTC Circuits

The NCP663 regulator is just a proposal. It has been selected, due to its low quiescent current.

If the power fail/Backup RAM feature is not used, the RTC_POK pin can simply be connected to VDDIO (+3,3 V).

The values for the capacitors and the serial resistor in the crystal circuit again depend on the selected crystal. The shown values are suitable for the Hilscher standard netX RTC crystal, which is a Q0.032768-JTX520-12.5-20T1-LF, manufactured by Jauch Quartz GmbH.

4.3.2.2 Designs without RTC or with netX100

When making designs with the netX500 that do not make use of the RTC feature or when making designs with netX100, the RTC pins must be connected as shown in the following figure:

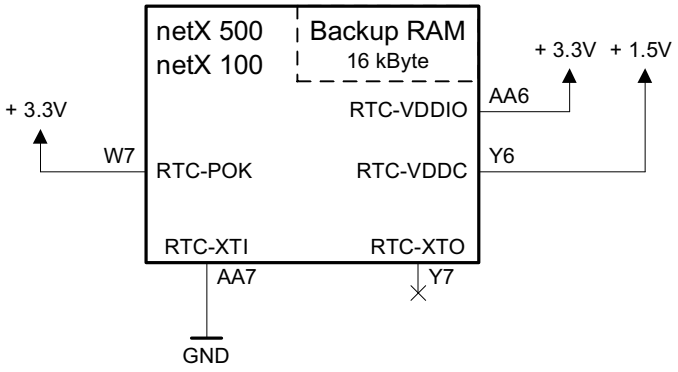


Figure 43: netX100/500 RTC Unused

4.4 Power On Reset and Reset In

The netX100 and netX500 provide two inputs for reset signals, the Power On Reset (PORn) and the Reset In (RSTINn). While the use of the RSTINn is optional, the Power On Reset is mandatory. Since the PORn input is equipped with a Schmitt-trigger gate, it could basically be connected to a capacitor (other pin of cap. connected to GND) and a pull-up resistor, however it is strongly recommended to connect this signal to the output of a reset generator with voltage supervision, to make sure, the netX will not be released from reset until the power supplies have reached sufficient and stable levels. Reset generator components are often available with either a push/pull or an open drain output. When the design will make use of the JTAG Interface of the netX, an open drain type should be selected, since this allows to simply connect the reset signal from the JTAG connector (which is also specified as open drain) to the output of the reset generator. Of course a pull-up resistor (e.g. 10 k) must be attached to the PORn signal when using open drain reset sources.

The optional RSTINn, which is commonly used by an external host processor to reset the netX, also provides a Schmitt-trigger gate. While the netX100/500 are not equipped with an internal pull-up resistor, it is recommended to tie it to VDDIO (+ 3,3 V), since this can improve EMC behavior.

When placing the components during PCB design, the reset source(s) should be placed near the reset inputs of the netX, to keep the traces off the reset signals short. Routing reset signals all over the PCB may result in bad EMC behavior of the design, since ESD may cause undesired resets of the chip.

Experience with several netX designs further has shown that a 1nF ceramic capacitor connected to GND and PORn, with the capacitor located close to the netX PORn pin, further improves resistivity against ESD.

The following figure shows the standard reset circuits:

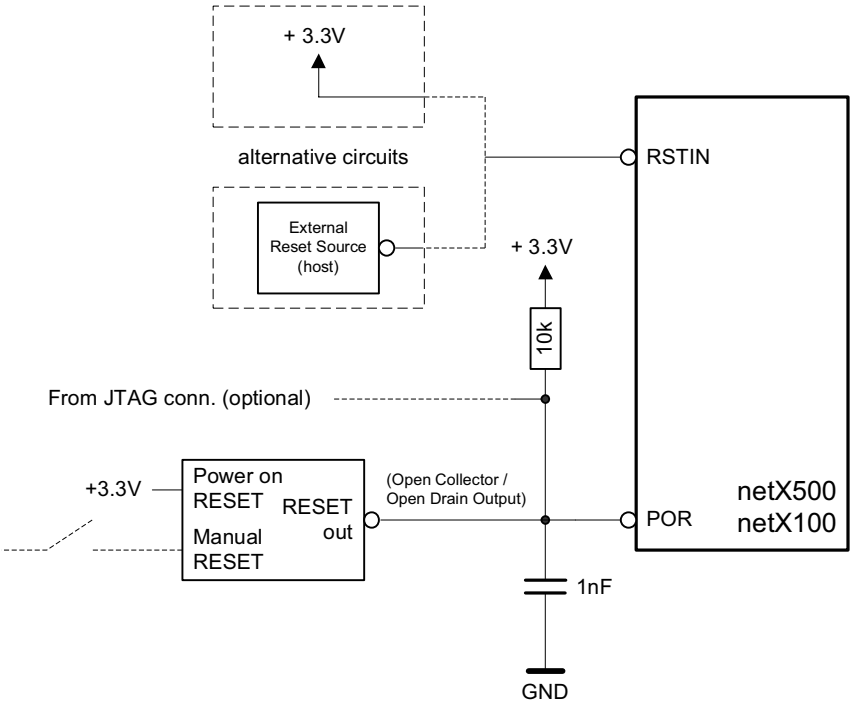


Figure 44:netX Reset Circuits

4.5 Debug and Test Interfaces

4.5.1 JTAG Interface

The netX100 and netX500 are equipped with a standardized JTAG Interface that allows loading, flashing and starting firmware, debugging of firmware and provides access to the Boundary Scan Test mode of the chips. Though this interface is rarely used during operation of final netX products, it is strongly recommended, to have at least retrofittable access to this interface on any netX design, especially on prototypes, even for designs where the customer does not intend to write his own software but use Hilscher loadable firmware instead. Debugging of a prototype can be somewhere between cumbersome and impossible if the JTAG interface is not accessible and for development of own netX software, a JTAG interface is essential anyway. Further, automatic testing systems used in production testing usually need access to this interface, so the JTAG interface pins should at least be connected to test points that can be contacted by prober systems.

Whenever there is enough space on the design's PCB, a standard 20 pin shrouded header with 0.1" or 2.54 mm pitch should be used for the JTAG interface, since this allows connecting standard JTAG debugging units to be plugged to the board instantly, without the need for any special cable adapters.

The following figure shows the standard netX JTAG circuit:

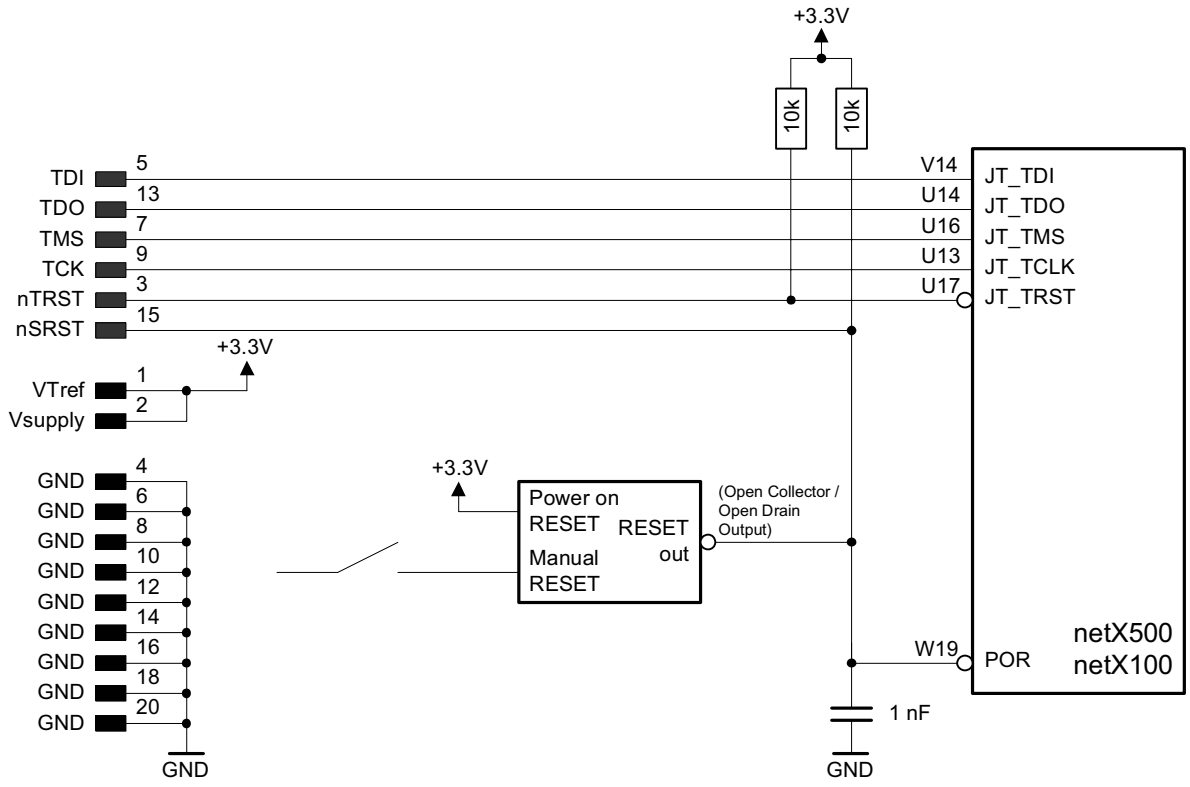


Figure 45: netX JTAG Circuits

In designs that will not require the use of the JTAG interface, the JTAG signals may be left unconnected. The internal pull-down on the JT_TRSTn will then constantly hold the JTAG interface in Reset state.

4.5.2 ETM Interface

The ETM Interface (Embedded Trace Macro cell) that is provided by the internal ARM CPU of the netX100/500 dramatically extends the debugging capabilities provided by the JTAG interface. It is generally recommended to implement when building netX Evaluation boards, but will most likely not be needed to debug prototype hardware. Even the 38 pin ETM board connectors (AMP Mictor 2-767004-2) are costly, let alone appropriate ETM debugging units, so implementing an ETM interface will usually only be interesting for customers that write own software for their netX design.

While the ETM interface on the netX500 is shared with the LCD Controller signals, which means, that driving a display is not possible when using the ETM port and vice versa.

netX500 designs driving an LC-Display, and providing an ETM connector at the same time, should provide a possibility to disconnect the display in order to avoid any negative effect on the ETM signals from the display. Also the ETM and the display connector should be located close to each other, to avoid long stubs (which may cause signal reflections) going from the connector in use to the one not in use.

The ETM connector is to be wired to the netX according to the following table:

Conn. Pin	ARM Signals	netX Signals		Conn. Pin	ARM Signals	netX Signals
1	N.C.			2	N.C.	
3	N.C.			4	N.C.	
5	GND	VSS		6	TRACECLK	ETM_TCLK
7	DBGRQ	ETM_DRQ		8	DBGACK	ETM_DACK
9	nSRST	Not used		10	EXTTRIG	
11	TDO	JT_TDO		12	VTRef	VCCIO
13	RTCK	Not used		14	VCC	VCCIO
15	TCK	JT_TCLK		16	TRACEPKT[7]	ETM_TPKT07
17	TMS	JT_TMS		18	TRACEPKT[6]	ETM_TPKT06
19	TDI	JT_TDI		20	TRACEPKT[5]	ETM_TPKT05
21	nTRST	JT_TRSTn		22	TRACEPKT[4]	ETM_TPKT04
23	TRACEPKT[15]	ETM_TPKT15		24	TRACEPKT[3]	ETM_TPKT03
25	TRACEPKT[14]	ETM_TPKT14		26	TRACEPKT[2]	ETM_TPKT02
27	TRACEPKT[13]	ETM_TPKT13		28	TRACEPKT[1]	ETM_TPKT01
29	TRACEPKT[12]	ETM_TPKT12		30	TRACEPKT[0]	ETM_TPKT00
31	TRACEPKT[11]	ETM_TPKT11		32	TRACESYNC	ETM_TSYNC
33	TRACEPKT[10]	ETM_TPKT10		34	PIPESTAT[2]	ETM_PSTAT2
35	TRACEPKT[9]	ETM_TPKT09		36	PIPESTAT[1]	ETM_PSTAT1
37	TRACEPKT[8]	ETM_TPKT08		38	PIPESTAT[0]	ETM_PSTAT0

Table 27: ETM Signals

Note: The AMP Mictor connector has four additional through-hole-pins in the center which have to be grounded for proper operation of the trace port!

For the PCB layout it is recommended to have the lines for the ETM signals as short as possible (the signal delay should be < 100ps). The length of the lines should be equal to avoid different signal delays. To improve signal quality, matching resistors can be

placed in the signal lines (located as close as possible to the chip pins (<10mm)) to match the output impedance of the chip signal driver with the PCB trace impedance

4.5.3 Boundary Scan

For automated production testing of a final netX product, it may be desirable to make use of the netX Boundary Scan feature. In that case, certain conditions must be met to allow entering the Boundary Scan test mode:

The first condition refers to the logic level on the TEST pin (netX100/500: K19). The design must provide a possibility to pull this signal to high level, which activates the test modes, which can be realized by a jumper or a contact pad for the test system which then controls this signal. Further, all JTAG signals must be available to the test system (e.g. by appropriate contact pads).

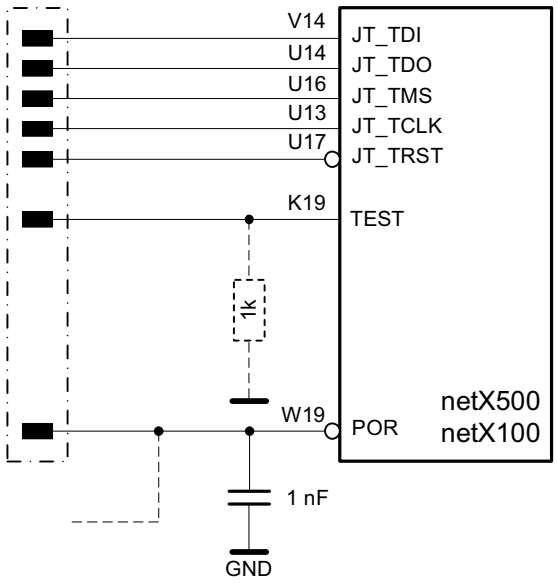


Figure 46: netX100/500 Boundary Scan JTAG/TEST Signals

The 1 kΩ pull-down resistor (placed as close as possible to the netX pin) can be omitted, when the signal trace between the contact pad and the netX pin is short, as then the internal pull-down resistor (nominal 50 kΩ) is sufficient to keep the TEST signal state inactive during normal operation.

The second condition refers to the state of some GPIO signals as shown in the following table. The design must provide the possibility to set the required logic levels on these signals and the levels must remain in that state throughout the complete Boundary Scan test procedure.

netX100/500		
Signal	Pin number	Signal State
GPIO14	W13	Pulled high
GPIO8	AA15	Pulled low or unconnected
GPIO9	Y15	Pulled low or unconnected
GPIO10	AA14	Pulled low or unconnected
GPIO11	Y14	Pulled low or unconnected

Table 28: netX100/500 Boundary Scan MMIO/GPIO Signals

4.6 External Memory

Basically, the netX100 and netX500 provide two different interfaces where firmware memory can be connected to: The (serial) SPI interface and the parallel FLASH/SRAM interface that shares most of its pins with the SDRAM controller.

Note: When the design is to be used with loadable firmware from Hilscher, an SPI FLASH is mandatory!

When connecting memory components to the parallel FLASH/SRAM/SDRAM interface, designers should always mind the capacitive load that is applied to the interface signals by the memory components. The memory interface of the netX100/500 is designed to handle a maximum load capacity of 50 pF on data- , address- and DQM3-0 lines and 25 pF on all other control signals.

The capacitive load directly influences the signal timing (the higher the load, the longer the signal delay) which has limited scope with SDRAMs. Since the allowed range of operating conditions (min./max. voltage, min./max. temperature) further influences signal timing, capacity limits needed to be defined, that ensure safe operation throughout the whole voltage and temperature range.

When exceeding these capacity limits, this may, to a certain amount, be compensated by two clock phase parameters of the SDRAM interface, hence such “out-of-spec-designs” are imaginable, but require careful evaluation!

4.6.1 FLASH Memory

4.6.1.1 SPI FLASH

SPI FLASH components consume considerably little space (SO-8 package) on the PCB, while being able to hold large firmware images of 4 MB or even greater, hence it is always recommended to add such a FLASH to any design if allowed by board size constraints. Even designs using a parallel FLASH as firmware memory or designs that receive their firmware through the DPM interface from an external host processor can benefit from an additional serial FLASH that can hold a second stage boot loader or non-volatile user data. Finally it may always simply be left unpopulated if really not used in the final product.

While an SPI Flash can be connected to three different chip select signals, it must be connected to chip select 0 (SPI_CS0), when the design is to be able to boot from this SPI Flash.

A standard component used by Hilscher is the AT45DB321 from ATMEL, providing a capacity of 4 MB. It is being connected as shown in the following schematics:

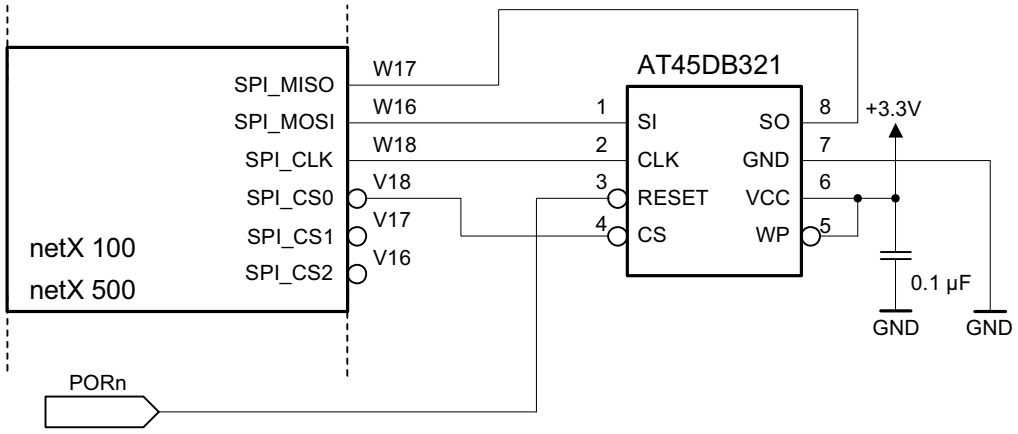


Figure 47: netX100/500 SPI Flash

4.6.1.2 MMC/SD Card

Instead of or in addition to an SPI Flash, MMC/SD cards can be connected to the netX, even allowing to boot a firmware image stored on such a card. To allow booting, the MMC/SD card must be connected to SPI chip select 1 (SPI_CS1).

Note: Please note, that the current ROM boot loader of the netX500 may have problems booting from certain MMC/SD cards due to timing issues (see also Errata sheet of netX500). In that case, an additional SPI Flash holding the second stage loader must be connected to SPI_CS0.

To detect insertion or removal of the MMC/SD card during operation, an insertion signal has been defined, that must be pulled high when an MMC/SD card is in the socket:

Function	Pin name and number	
MMC/SD insert	GPIO 15	V13

Table 29: MMC/SD Card Insertion Signal

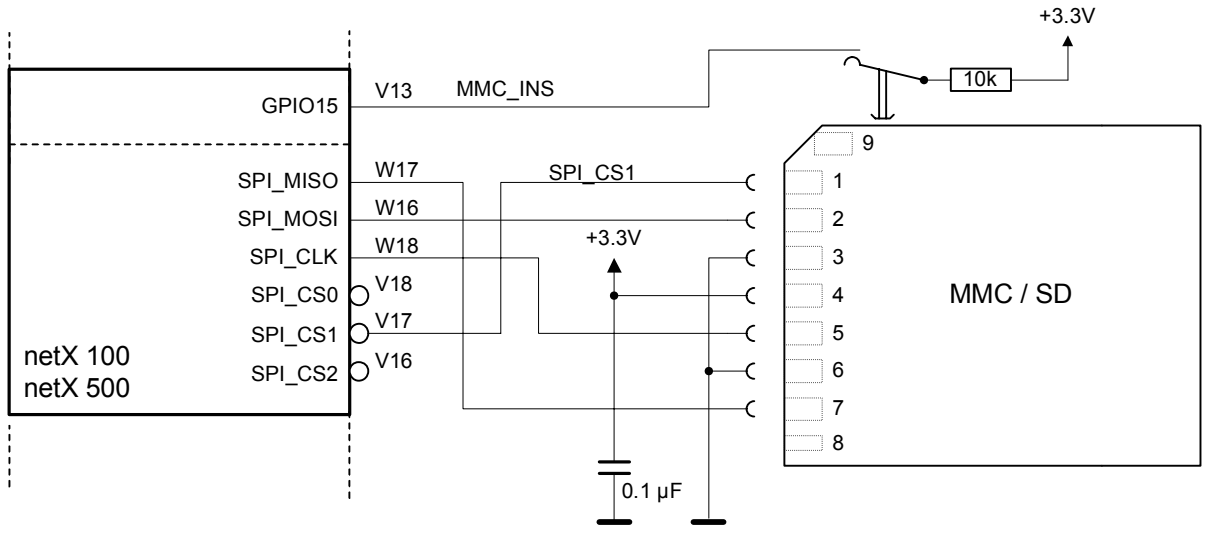


Figure 48: netx100/500 MMC/SD Card

Some SD-Cards disturbs the SPI data transmission from FLASH. A solution is to disconnect the SD-Card from MOSI and MISO lines with switchable bus driver when SD-Card chip select (SPI_CS1#) is high. The Figure 49 shows a solution.

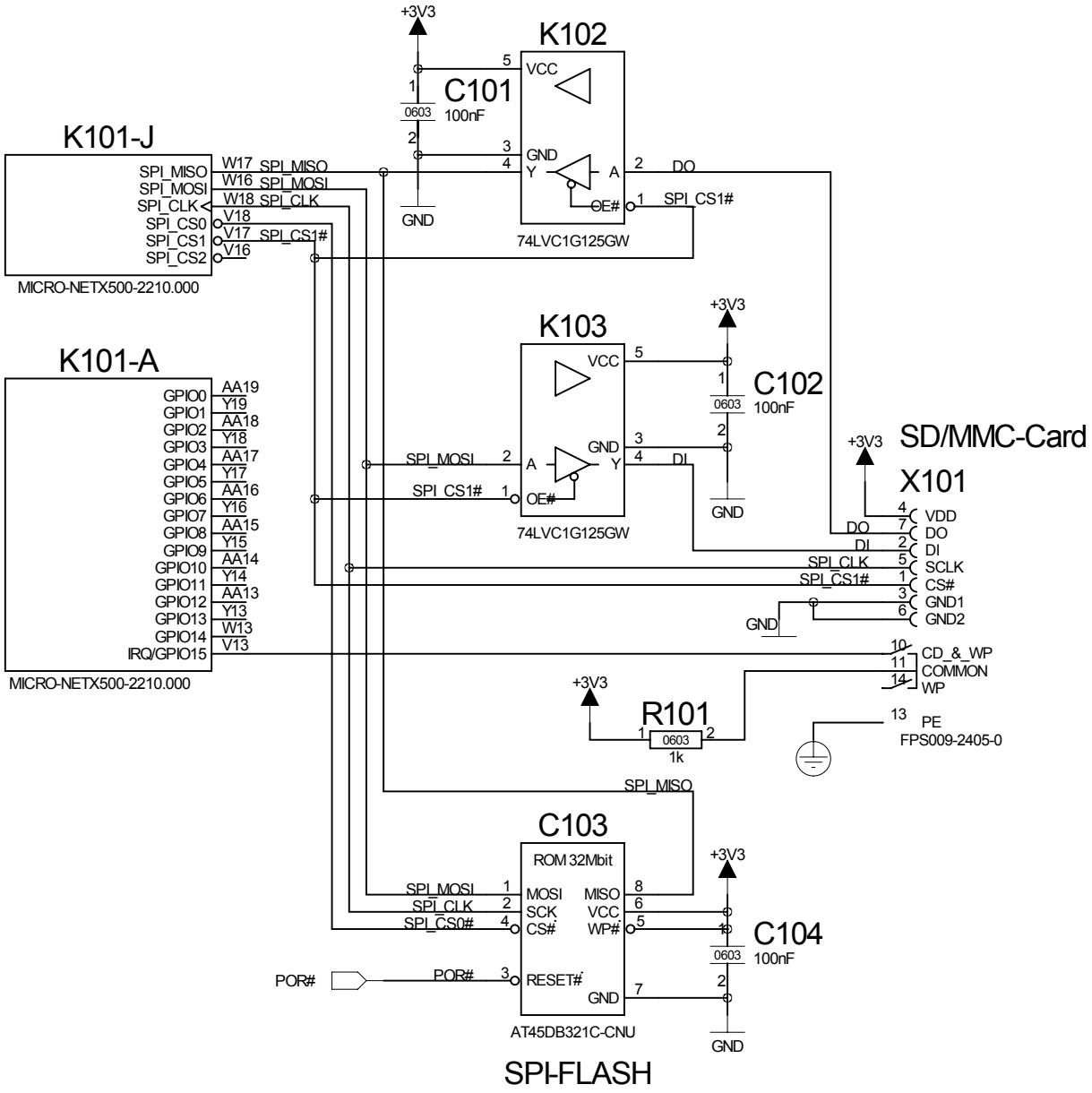


Figure 49: Solution, if SD Card Disturbs SPI Data Transmission from FLASH

4.6.1.3 Parallel FLASH

For large firmware images that come into play with graphical operating systems like Windows CE or for applications executing code directly out of FLASH, the use of parallel FLASH is inevitable. Parallel FLASH connected to the netX may be 8-, 16- or 32 Bit wide, while two 16 Bit components may be paired for 32 Bit wide access.

Note: rcX does not support parallel flash. It can no file operations are executed with rcX.

Though 16 Bit wide components are most common, for performance reasons 32 Bit components should be used when executing code directly out of FLASH.

The netX SRAM/FLASH memory controller provides three different chip select signals (MEM_CS[2:0]), allowing to select three different memory components or pairs of components (two

paired 16 Bit FLASHes use a common chip select signal), each with its own set of parameters (timing and bus width).

When the design is to boot from parallel FLASH, chip select 0 must be used for selecting this FLASH.

The memory controller is designed to never “waste” any address lines, regardless of the bus width setting. Hence in 8 Bit mode, address line A0 is used for low and high Byte selection, while in 16 Bit mode A0 selects low and high word and in 32 Bit mode, A0 is simply the LSB of a DWORD address.

For that reason, the data sheet of the desired FLASH component must be consulted, to determine the correct way of hooking up the address lines of the FLASH.

Many (16 Bit-) FLASH components (e.g. TE28F128J...) use address line A0 for low/high Byte selection when operating the component in 8 Bit mode and do not use A0 at all when in 16 Bit mode. Such components must hence have A0 of the FLASH grounded (to prevent floating), while A0 of the netX is connected to A1 of the FLASH, A1 to A2, A2 to A3, etc. The following schematics show an example:

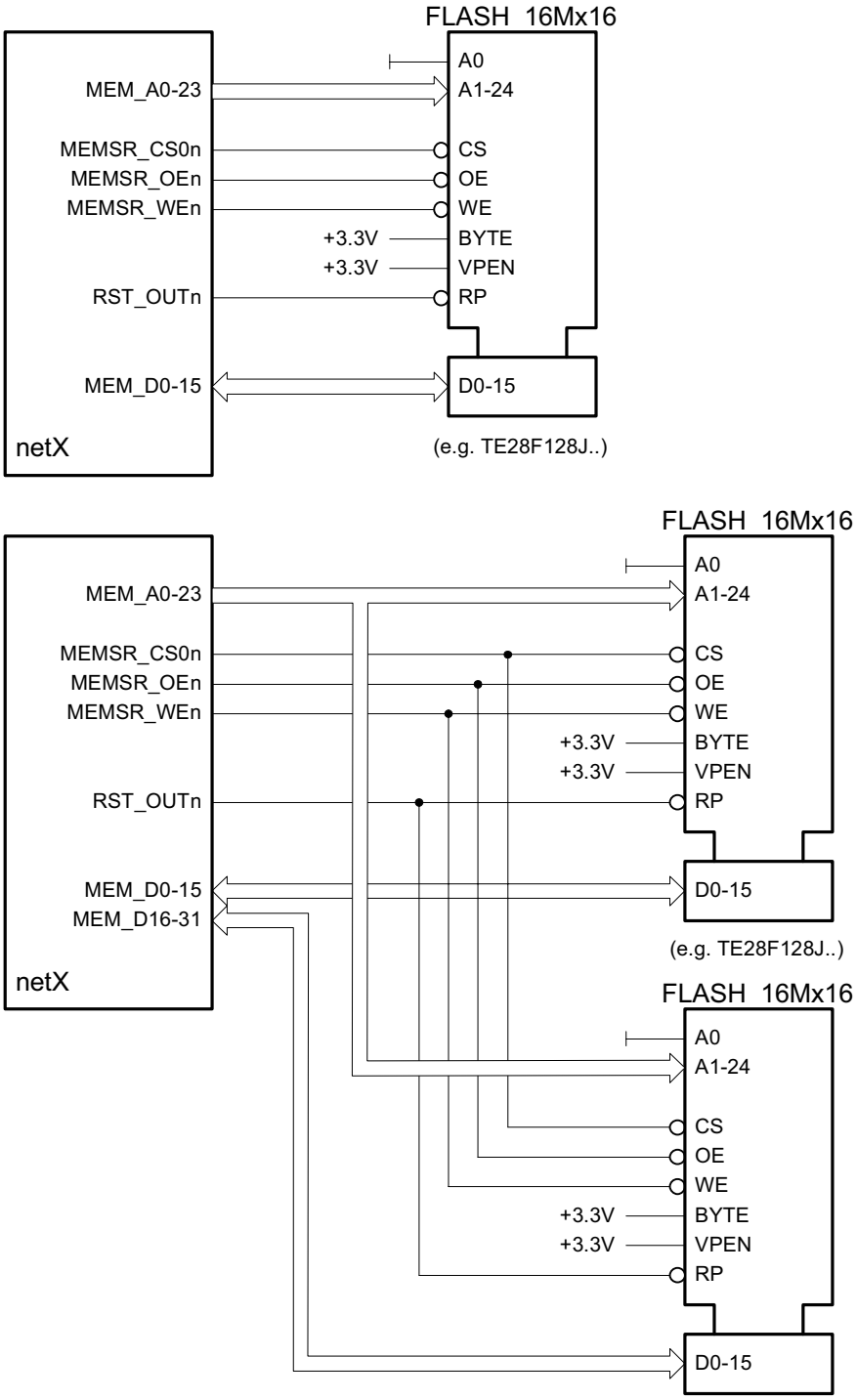


Figure 50: netX FLASH - Address Line A0 for Low/High Byte Selection

Other FLASH components (e.g. S29GL256P...) always use A0 as the LSB of a Word (16 Bit-) address, hence the address lines of such components must be connected straight forward as shown in the following example schematics:

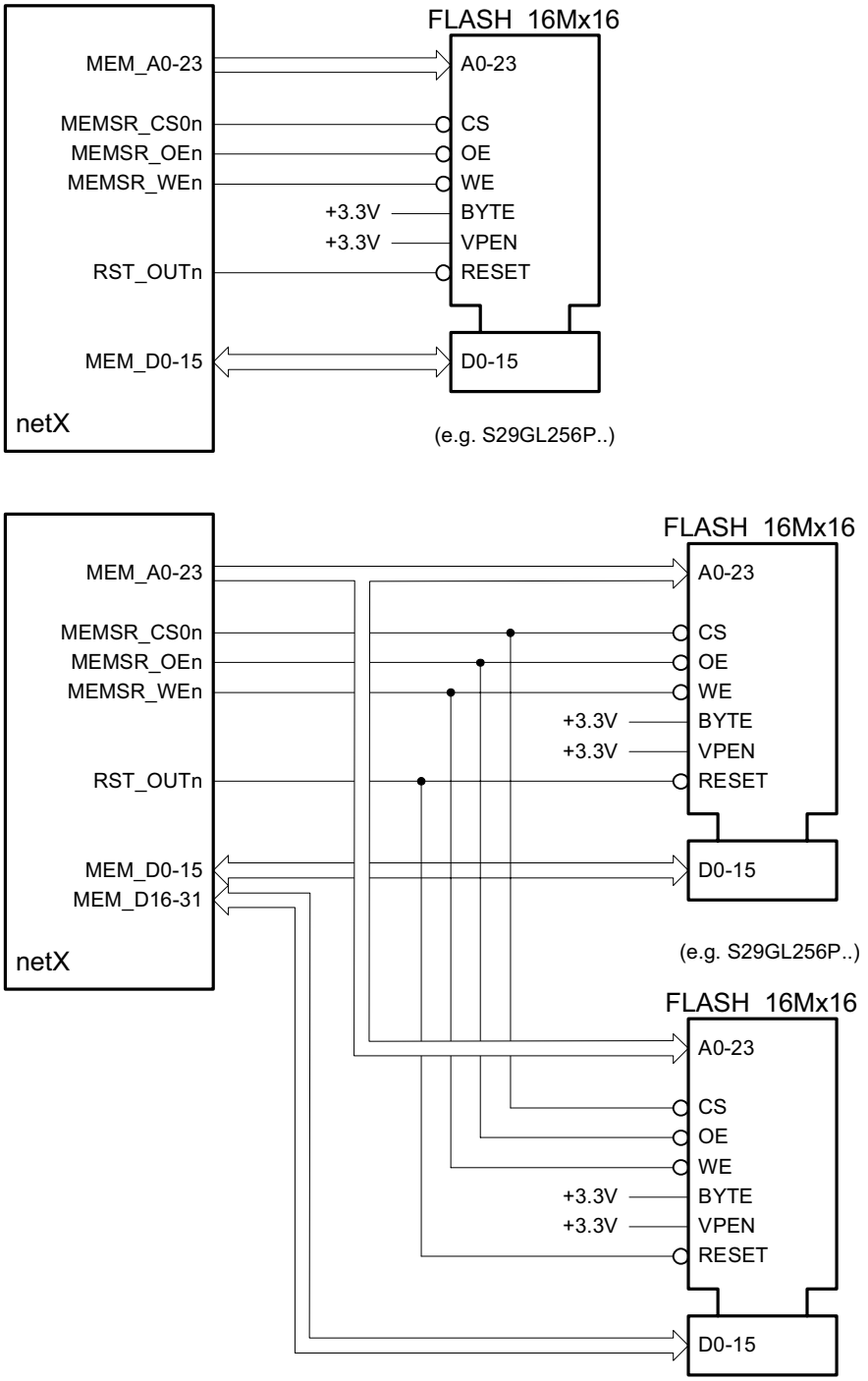


Figure 51: netX FLASH - A0 as the LSB of a Word Address

Q1: I would like to access parallel FLASH connected to the netX by another processor (e.g. host processor) while the netX does not use the FLASH or is being held in reset. Is that possible?

A1: This will only work with additional components. Since memory interfaces are usually not designed for multi-master access, the netX always drives all of the memory interface signals, except the data lines, even while being held in reset state. Hence accessing any memory components by another processor is only possible when the memory components can be isolated from the netX memory interface by appropriate bus switches.

Q2: Do I need any pull-up or pull-down resistors on the netX memory interface?

A2: Since the netX always drives all of the memory interface signals, except the data lines, such resistors will only make any sense on the data signals. If no memory components are connected to the FLASH/SRAM/SDRAM interface of a netX100/500 design, then pull-ups or pull-downs on the data lines will avoid possible cross currents and may hence reduce power consumption. If only 16 Bit components are used, this applies to the upper 16 data lines.

4.6.2 SDRAM

Most netX applications will require the use of SDRAM, since most of the internal RAM is usually occupied by the standard 64 kB DPM and buffers, leaving only little space for quite simple applications. For certain slave applications, an alternative to using SDRAM may be the use of parallel FLASH, while the firmware is directly executed out of this FLASH instead of copying the firmware from FLASH to RAM and executing from there (which is the standard situation).

SDRAM components connected to netX may be either 16 Bit or 32 Bit wide, while two 16 Bit components may be paired to allow 32 Bit wide access. Using two 8 Bit components (paired for 16 Bit) or four 8 Bit components (32 Bit) is also possible.

When using SDRAM, 32 Bit wide designs are generally recommended, to make use of the full performance of the memory controller. The use of one 32 Bit wide component instead of two 16 Bit (or four 8 Bit-) components is further recommended, due to easier PCB design and reduced load capacity (two 16 Bit components usually add twice the load to the address and control signals as a comparable 32 Bit component).

Q1: In the meantime DDR-3 RAM is state of the art. Why are the netX chips only equipped with an outdated SDRAM interface?

A1: Well, SDRAM isn't really outdated. DDR RAM technology was invented for the short-lived PC market, where it is commonly accepted, that components have extremely short life cycles, limited operating condition range and substantial power consumption. Since DDR RAMs work with internal PLLs and can hence not be used on older (slower) memory interfaces, DDR RAM technology is not suitable for the embedded/industrial market where customers usually look for availability of several years. Further even powerful embedded processor technology like ARM can necessarily not compete with common PC processors in terms of processing power, hence it would make little sense to connect such processors to DDR RAMs anyway.

Connecting SDRAM to the netX is pretty straight forward, besides address lines A16 and A17, which are used for the bank select signals BA0 and BA1.

The schematics on the following page show examples for connecting 8-, 16- and 32 Bit SDRAMs:

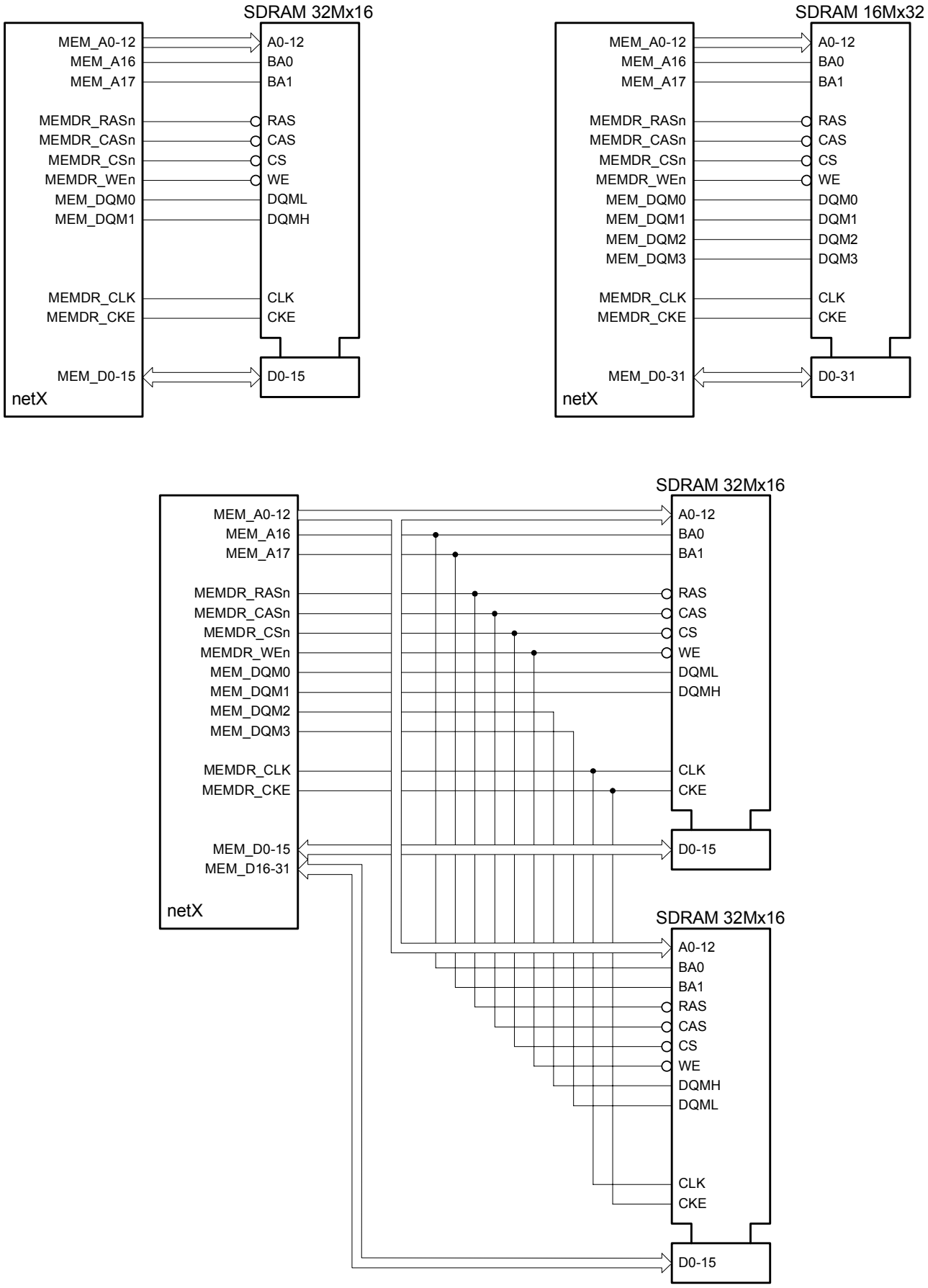


*Figure 52:netX SDRAM 1 *16 Bit, 1 * 32 Bit, 2 * 16 Bit*

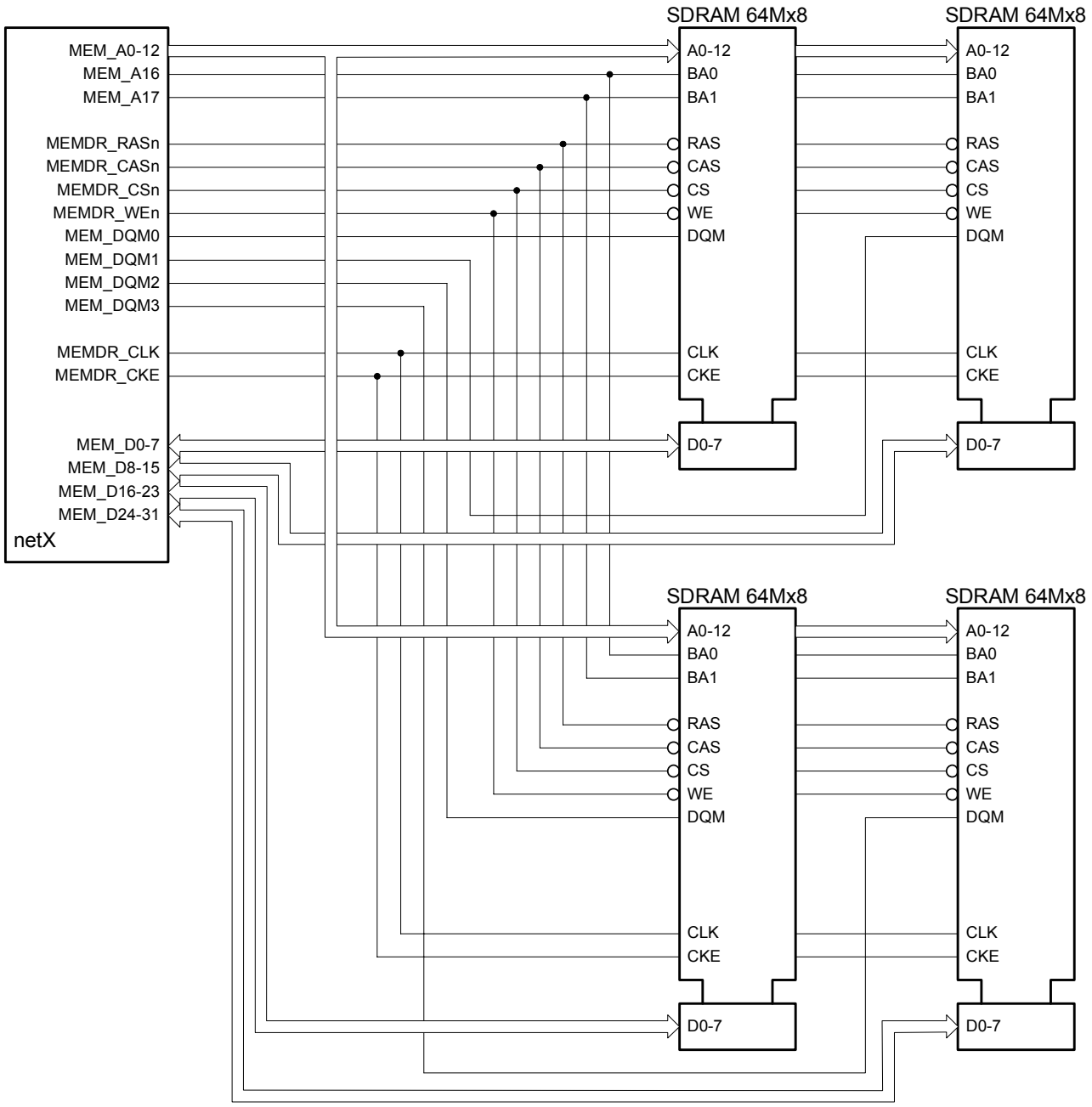


*Figure 53:netX SDRAM 4 * 8 Bit*

4.7 Host Interface

The netX100/500 provides a versatile parallel asynchronous interface, referred to as “host interface” or “HIF”, that can either be “active” (netX controls the interface and accesses other memory mapped components) or “passive” (netX behaves like a Dual Port Memory (DPM) and is accessed by an external host processor). In DPM mode, the host interface can either be 8 or 16 Bit wide. Signals that are not used for host interface operation of a design (e.g. upper address lines) can be separately configured as I/Os. The host interface always works in little endian mode.

As a third mode, the netX100/500 host interface can also operate as a PCI interface (device or PCI host). Since the use of the netX PCI interface is subject to certain restrictions and requires an appropriate contract, that mode is not covered in this design guide. Please contact the Hilscher sales department if you plan to use the netX PCI interface in your design.

Due to the fact that the behavior of the netX host interface is configurable to a large extent, most applications do not require any additional glue logic.

The host interface signal buffers of the netX100/500 are 5 V tolerant and PCI compliant and are equipped with internal clamping diodes, to raise signal undershoot/overshoot tolerance. The cathodes of the upper clamping diodes are internally wired to three power supply pins (VDDH) that must either be connected to +3.3 V (standard designs) or +5 V (host processor uses 5 V signaling voltage).

Note: The output level of the netX host interface will however always be 3.3 V and NO other signals of the netX100/500 are 5 V tolerant!

Since the netX100/500 host interface pins are not equipped with internal pull-up or pull-down resistors, all signals of the HIF are floating after reset. As floating signals should generally be avoided, designers should either apply external pull-up or pull-down resistors or ensure that the firmware configures all unused pins as I/O Outputs and drives them high or low.

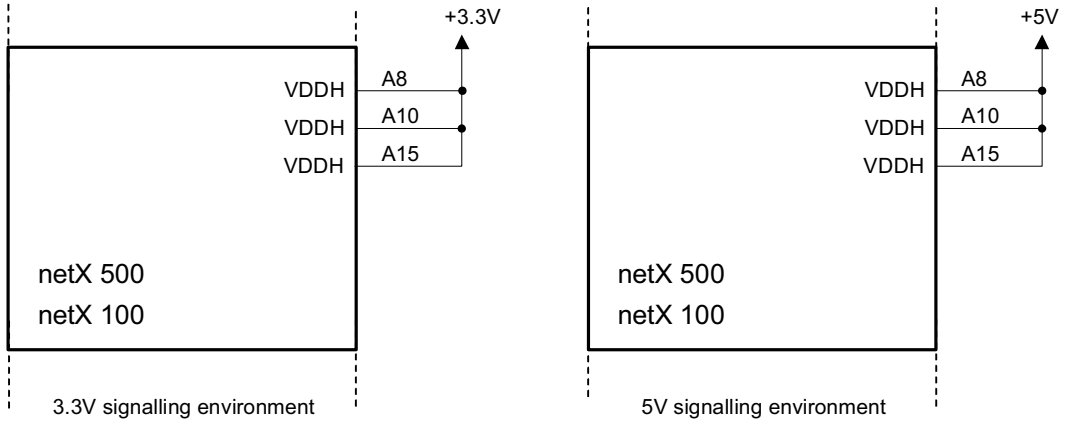


Figure 54: netX100/500 VDDH Pins

4.7.1 DPM Mode

In DPM mode, the host interface can be controlled by separate Read (RDn) and Write (WRn/WRLn, WRHn) signals (“Intel mode”) or by a combined R/WRn signal indicating the direction of the access and Byte strobe signals (“Motorola mode”). In “Intel mode”, either a single Write signal can be used (WRLn) or two Write signals (WRLn, WRHn) for writing to the low Byte (WRLn) and high Byte (WRHn) separately.

4.7.1.1 Address Bus

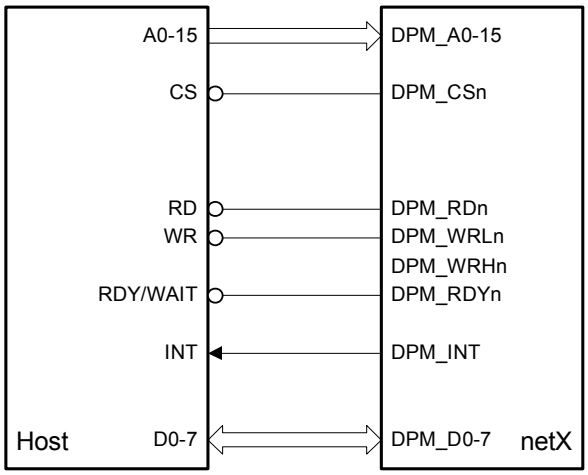
In case of a 16 Bit host systems with word access the address line A0 should be connected to GND. The following table shows the decoding logic for byte and word access.

BHE#	A0	Function
0	0	word access
0	1	access high byte
1	0	access low byte
1	1	no access

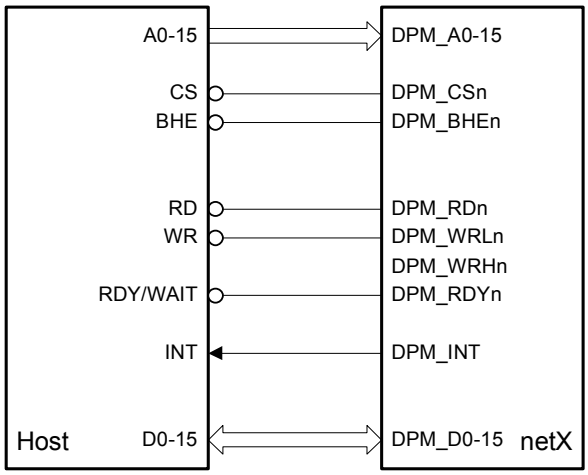
Table 30: Function Table of 16 Bit Decode Logic

4.7.1.2 Non-multiplexed mode

The following schematics show some examples for common setups in non-mulitplexed mode:

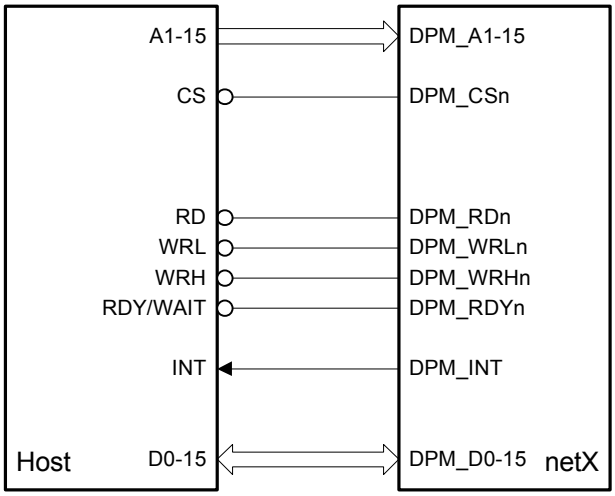


'Intel ™ interface', 8 Bit, non multiplexed



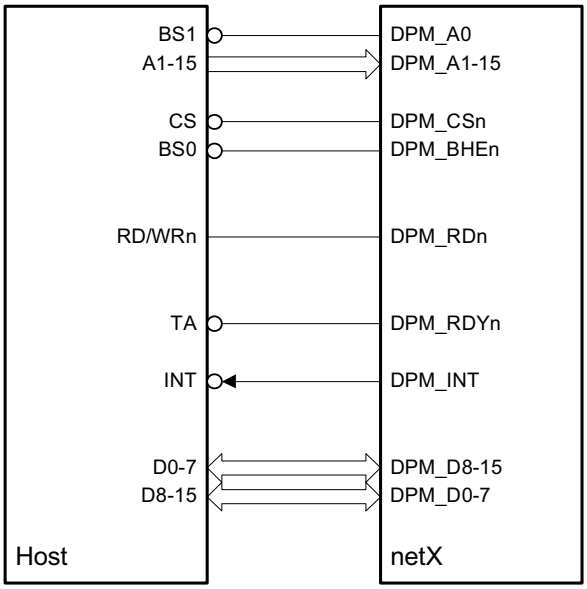
'Intel ™ interface', 16 Bit, non multiplexed

Figure 55: netX DPM Intel™ Type Interface Circuits (1)

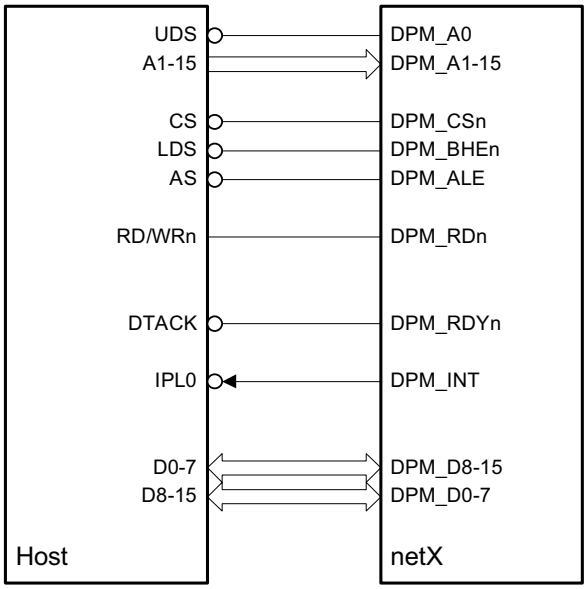


'Intel ™ interface', 16 Bit, non multiplexed, 2 write signals (Low Byte, High Byte)

Table 31: netX DPM Intel™ Type Interface Circuits (2)



Motorola™ ColdFire, 16 Bit, non multiplexed



Motorola™ M68000, 16 Bit, non multiplexed

Figure 56: netX DPM Motorola™ Type Interface Circuits

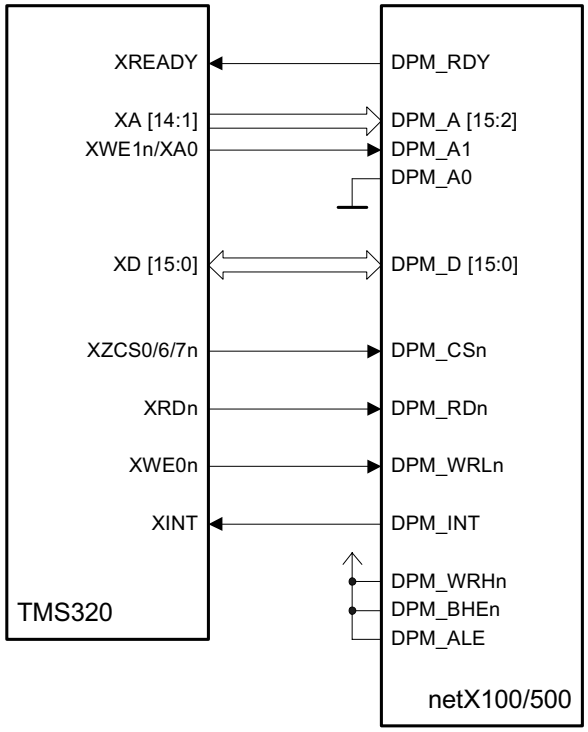
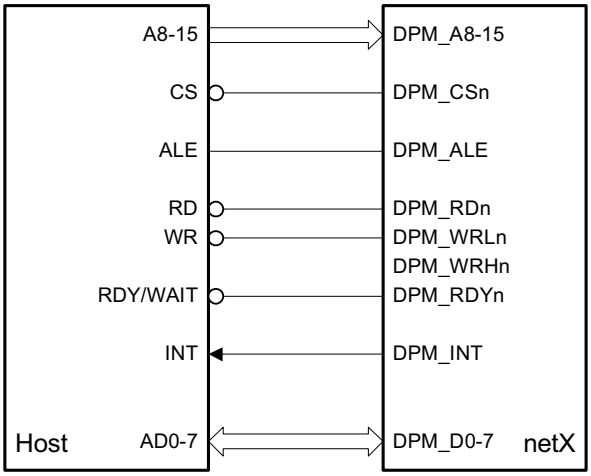


Figure 57: Texas Instruments TMS320x2833x™, 16 Bit, Non Multiplexed

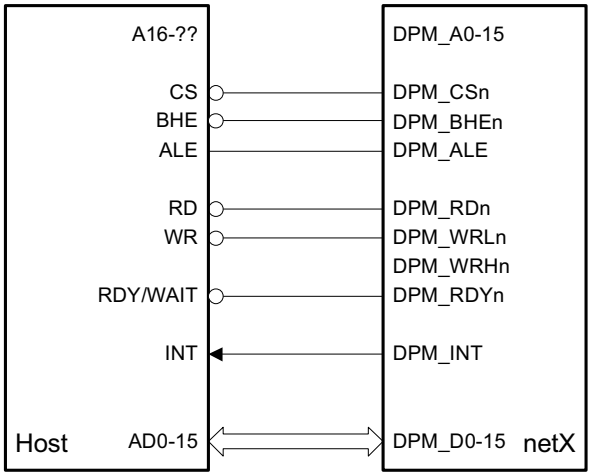
Note: The netx100/500 interprets the passed address as 8 Bit-Datum.

4.7.1.3 Multiplexed Mode

The netX host interface can also be operated in multiplexed mode, where the data lines are alternatingly used for data and the lower address signals. The following schematics show some examples for common setups in multiplexed mode:

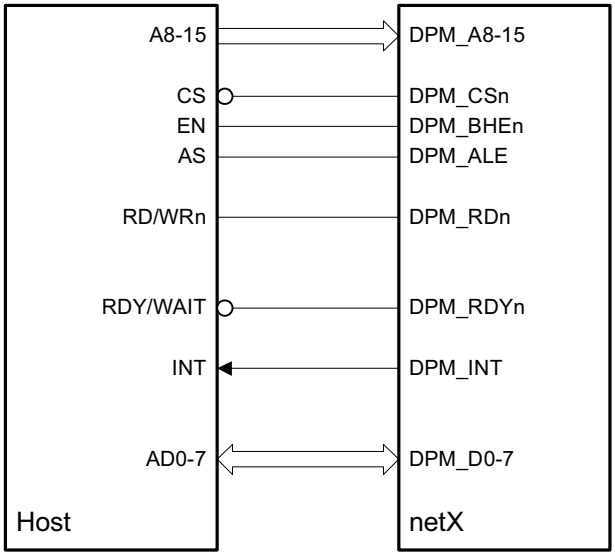


'Intel ™ interface', 8 Bit, multiplexed



'Intel ™ interface', 16 Bit, multiplexed

Figure 58: netX DPM Intel™ Type Circuits, Multiplexed



'Motorola™ interface', 8 Bit, multiplexed

Figure 59: netX DPM Motorola™ Type Interface Circuit, Multiplexed

4.7.1.4 ISA Mode

The netX host interface also provides a virtually glueless interface to ISA Bus or PC104 systems. The following schematic shows how to connect the host interface in that case:

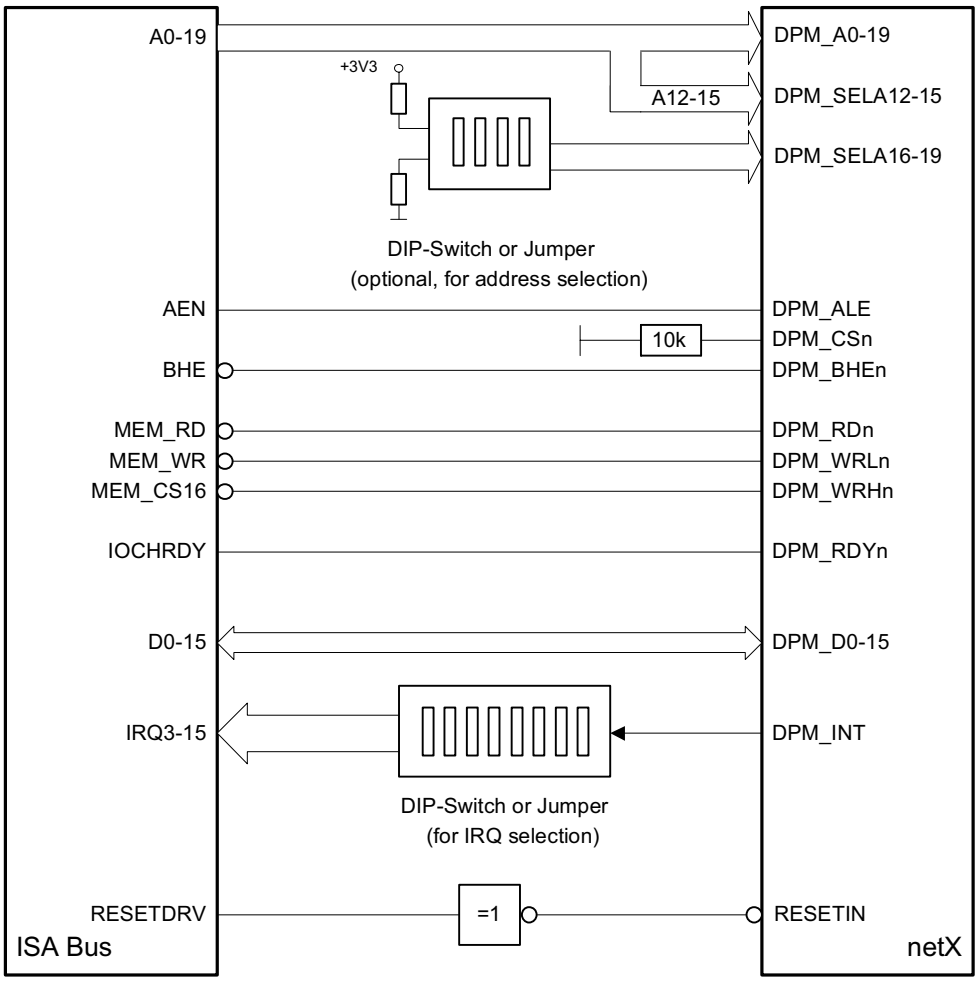


Figure 60: netX ISA Bus Interface Circuit

Since the ISA Bus requires its interface cards to perform address decoding, the netX host interface also provides an internal chip select generator that can be used instead of the DPM_CSn chip select signal.

The DIP Switch or jumpers connected to signals DPM_SELA16-19 allow to set the netX DPM to a certain base address on the ISA Bus. The switches or jumpers are optional, as the host interface can also be configured to use an internally stored compare value, allowing to set the base address by firmware.

The circuit shown above will result in a 64k address space occupied by the netX ISA Bus interface, which is the netX standard DPM size.

However, as memory space is often scarce in ISA Bus systems, the occupied memory space of the netX may be reduced by removing connections between DPM_SELA12-15 and the corresponding A12-15. If the connection between DPM_SELA15 and A15 is removed and (optionally) DPM_SELA15 is connected to the address switch / jumpers, the memory window is reduced to 32k. If this is also done for DPM_SELA14, the window is reduced to 16k, etc. Of course every reduction of the memory window size increases the granularity of the base address setting. Please also check the following subchapter for further details on the internal chip select generator.

4.7.1.5 Internal Chip Select Generator

As already mentioned in the previous subchapter 4.7.1.4, the netX DPM interface can use an internal chip select generator instead of the standard DPM_CSn external chip select signal. Although this was implemented mainly to meet the requirements of ISA Bus systems, the chip select generator can also be used for standard DPM applications.

The internal chip select generator, which is enabled by a certain register setting, compares the state of the host interface address lines A12 –A19 to either an internal compare value (stored in a register) or to the state of the host interface SEL_A12-19 signals. It can be configured for each address line separately, if the compare value is internal (register bit) or external (SEL_Axx).

Using the external source (SEL_Axx) for an address bit, allows to either connect the SEL_Axx signal to a jumper or switch (that either set this signal high or low) or to the corresponding address line (e.g. SEL_A12 to A12, SEL_A13 to A13, etc.), which makes this address bit “don’t care” for the chip select generator.

When using all 8 address Bits (A12-19) for decoding, which means that none of the SELA12-19 signals are connected to the corresponding address lines (none of the address bits are “don’t care”), this results in an addressable DPM size of only 4k, while the (external) base address of the netX DPM can be selected by applying the appropriate logic levels to the SEL_A12-19 signals or by setting the appr. Register value. Combinations are of course also possible (e.g. setting the most significant part of the base address by register value, while the rest of the address is set by jumpers/switches).

Starting from the 4k minimum, the accessible DPM size can continually be doubled to the maximum of 64k (standard size), by connecting the appr. number of SEL_Axx signals to the corresponding address lines and setting the corresponding compare value source to ‘external’.

Note: When reducing the addressable DPM size, please be aware of some drawbacks that come with smaller netX DPMs. Some system registers, which are “hard mapped” to the upper end of the 64k area can not be reached with a reduced DPM. Further, loadable firmware provided by Hilscher usually presumes that the full 64k area is available to the host and does hence not support designs with reduced DPM size. It is strongly recommended that hardware designers planning to reduce the addressable DPM size consult their software department, to make sure, this will not collide with the final application!

The following schematics show some examples of how to connect the address and SEL_Axx lines:

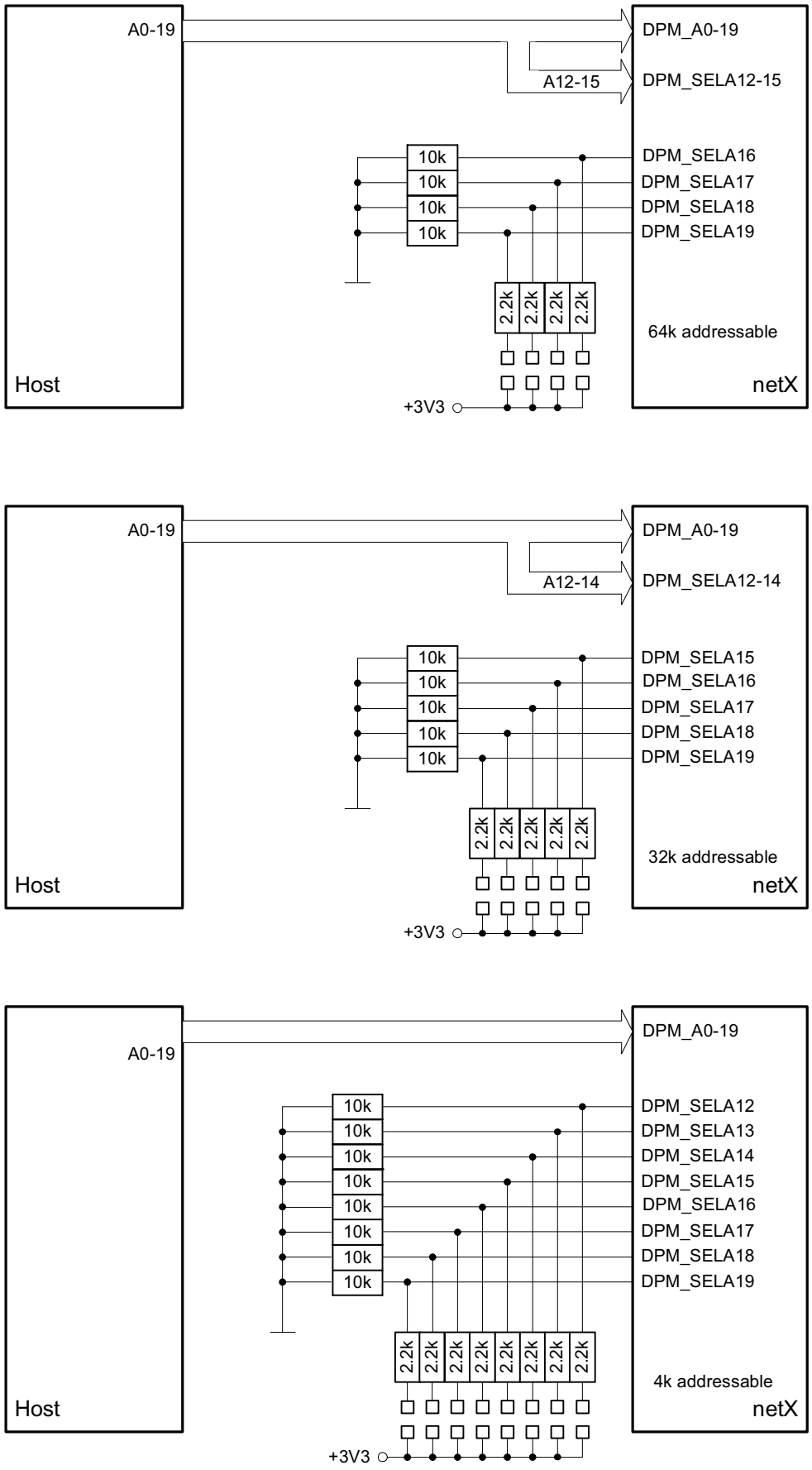


Figure 61: netX Internal Chip Select Generator Circuits

4.7.1.6 RDY/WAIT Signal

Since the DPM mode of the netX host interface provides a virtual Dual Port Memory instead of a real DPM (external accesses to the DPM are redirected to programmable memory locations inside the netX), there is no fixed access time for reading or writing the netX DPM, even when there is no access conflict.

Real DPM components however also use a Busy signal to compensate access conflicts (one side of the memory reads a certain memory location while the other side tries to write the same location or vice versa).

In order to make sure, that the host processor will read valid data, respectively the netX successfully accepted write data from the host, while keeping access cycles as short as possible, it is mandatory for the host processor to support a Ready or Wait signal from an external memory!

To allow glueless interfaces, the netX RDY/WAIT signal supports two different basic modes (Ready mode, where an active Ready signal indicates that the cycle may now be terminated and Wait mode, where an active Wait signal indicates, that the netX is still busy and the cycle may not yet be terminated) that can either work with active high or active low signals. Further, the output of the RDY/WAIT signal can be configured as push/pull or open drain – open source with sustained tri-state option (signal edge is actively driven). For more details and diagrams please consult the appr. netX Technical Reference Guide (separate documents for netX100/500).

4.7.2 Extension Bus Mode

In Extension Bus mode, the netX provides an active parallel, 8- or 16 Bit wide asynchronous bus interface with support for a Ready or Wait signal that allows external components to extend data cycles beyond the programmed cycle timing.

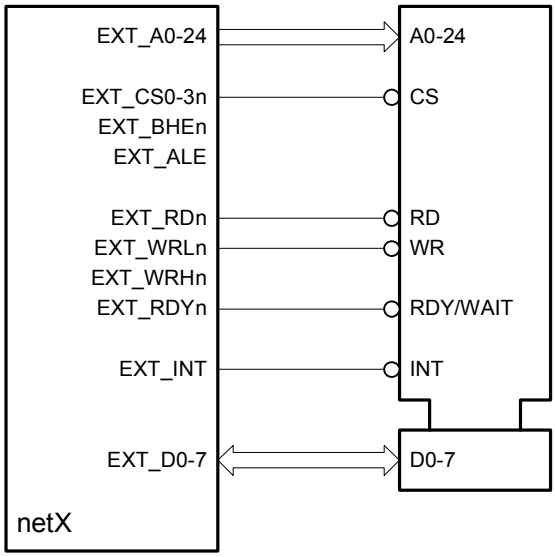
The Extension Bus can either use separate Read (EXT_RDn) and Write (EXT_WRn/WRLn, EXT_WRHn) signals (“Intel mode”) or a combined R/WRn signal indicating the direction of the access along with Byte strobe signals (“Motorola mode”). In “Intel mode”, either a single Write signal can be used (EXT_WRLn), combined with a Byte Enable Signals (EXT_BHEn) or two Write signals (EXT_WRLn, EXT_WRHn) for writing to the low Byte (EXT_WRLn) and high Byte (EXT_WRHn) separately.

The Extension Bus can be used to connect parallel peripherals like SRAMs, FLASHes, DPMs, etc. and can operate in non-multiplexed or multiplexed mode. The four different chip select signals allow to connect four completely different devices, as the configuration for each chip select area is done separately.

The netX also provides the possibility to boot from a memory device connected to the Extension Bus. In that case, the device must be connected to use the EXT_CS0n chip select signal and the Extension Bus boot mode must be selected.

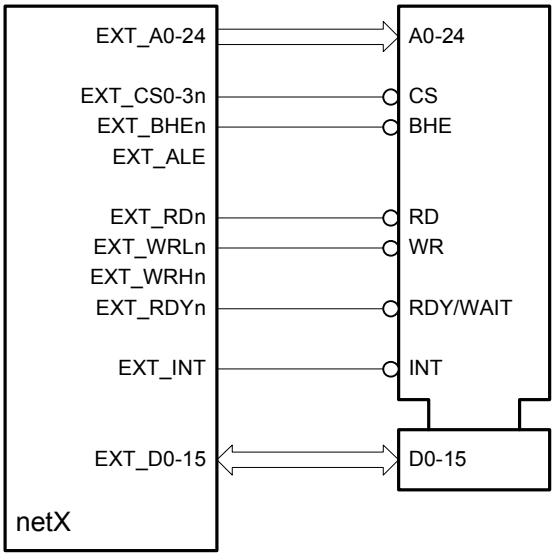
4.7.2.1 Non-multiplex Mode

The following schematics show some examples for common setups in non-multiplexed mode:

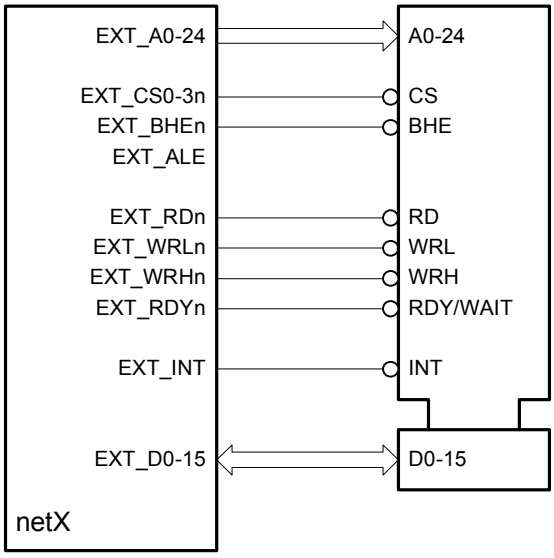


Intel™, 8 Bit, one write signal

Figure 62: netX Extension Bus Intel™ Type Interface Circuit 8Bit, non Multiplex

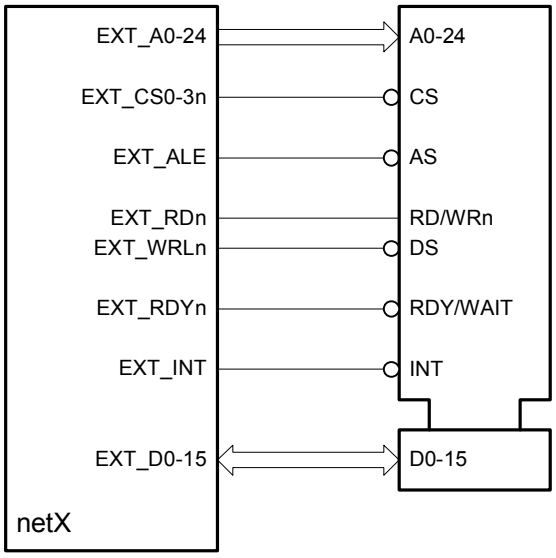


Intel™, 16 Bit, one write signal

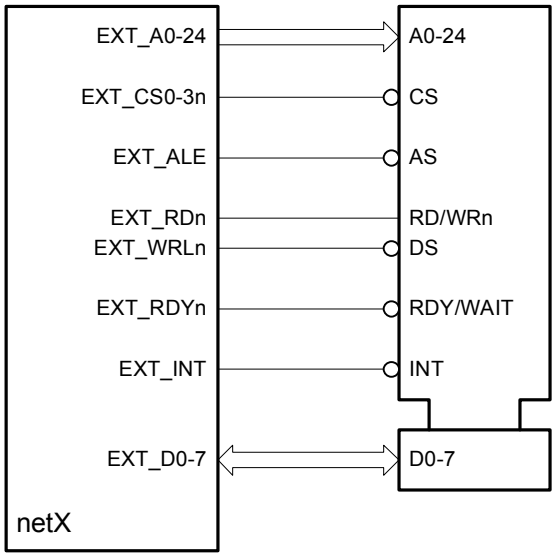


Intel™, 16 Bit, write low/high signals

Figure 63: netX Extension Bus Intel™ type Interface Circuit 16Bit, non Multiplex



Motorola™, 16 Bit, one data strobe

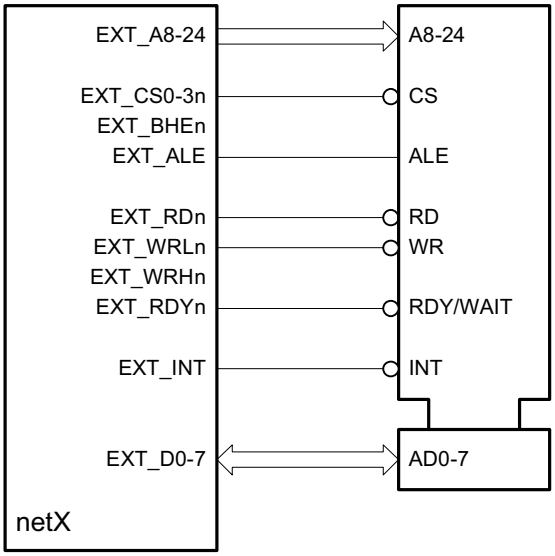


Motorola™, 8 Bit, one data strobe

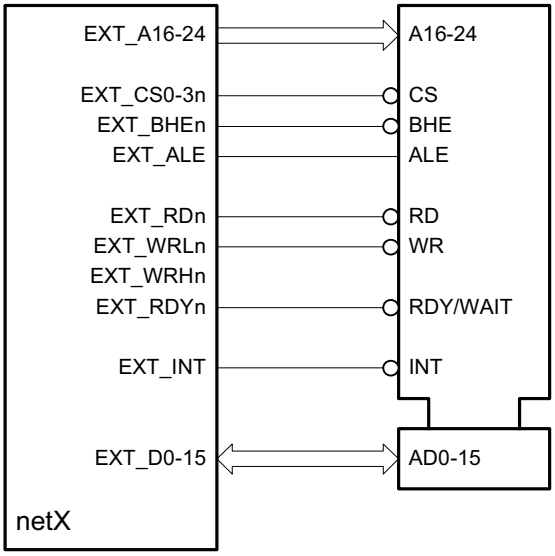
Figure 64: netX DPM Motorola™ Type Interface Circuits, non Multiplex

4.7.3 Multiplex Mode

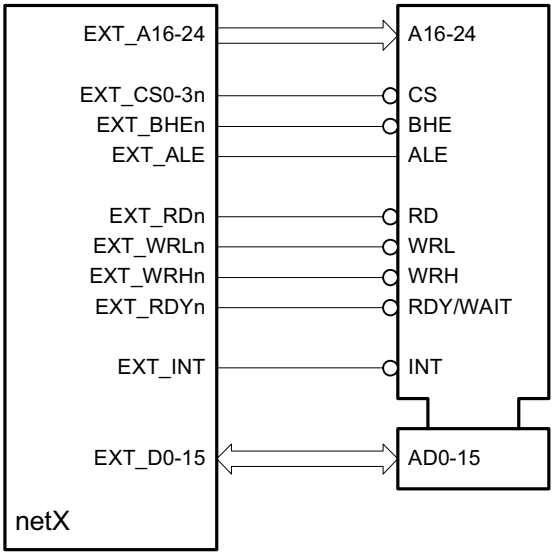
The netX host interface can also be operated in multiplexed mode, where the data lines are alternatingly used for data and the lower address signals. The following schematics show some examples for common setups in multiplexed mode:



Intel, 8 Bit, one write signal

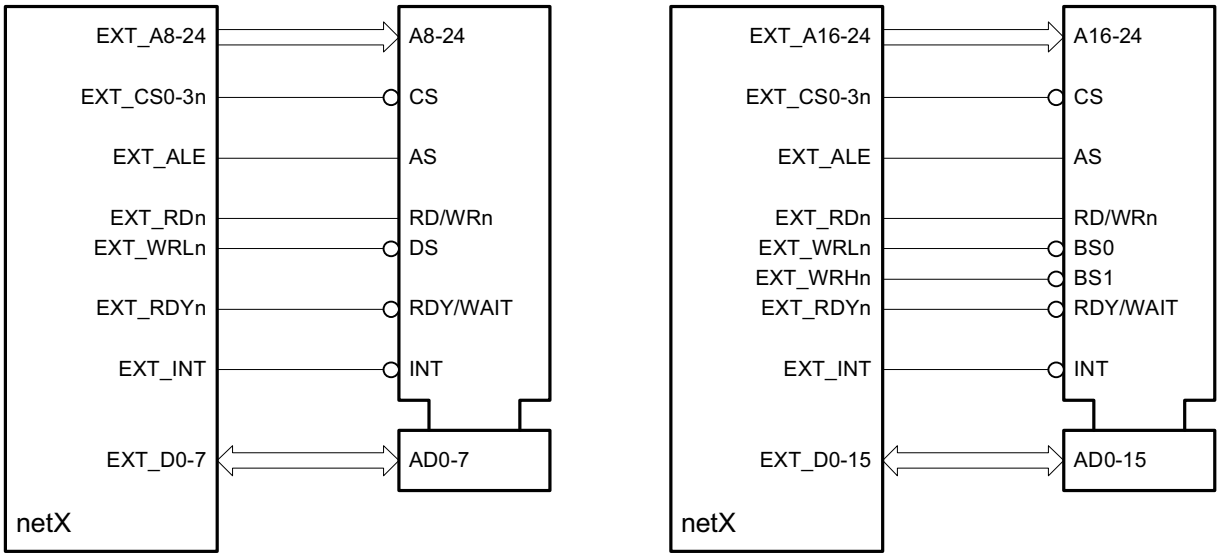


Intel, 16 Bit, one write signal



Intel, 16 Bit, write low/high signals

Figure 65: netX DPM Intel™ Type Interface Circuit, Multiplexed



Motorola™, 8 Bit, one data strobe *Motorola™, 16 Bit, two data strobes*

Figure 66: netX Extension Bus Motorola™ Interface Circuit, Multiplexed

4.7.4 External pull-ups/pull-downs, unused signals

As already mentioned at the beginning of the chapter 4.7, the netX100/500 does not provide any internal pull-ups or pull-downs on the host interface signals. For that reason, any unused signals of the host interface should either be externally pulled low or high, or should be configured as outputs and be driven low or high by the firmware of the design.

Since even the connected signals of the host interface may float upon reset (they are all configured as inputs by default), designers should make sure, that these initially floating signals will not cause any (start-up) problems with the host interface circuit of their design. If in doubt, add external pull-ups or pull-downs to ensure the appr. inactive state of the signals.

The CLOCKOUT signal and the WDGACT signals should always be pulled low or high when not used, since these signals can not simply be driven low or high. The TCLK signal should always be grounded (this is a dedicated input, so grounding is not a problem).

Q1: Instead of using external pull-down or pull-up resistors on unused host interface signals, can't just tie these signals to GND or 3.3 V?

A1: This is of course possible, but keep in mind, that almost all host interface pins can be configured to output mode, hence when tying such signals to GND or 3.3 V, there is always the risk of short circuit conditions, when misconfiguring the host interface (e.g. during software development).

4.8 UARTs

The netX provides a total of three UARTs (each with RX, TX, RTSn, CTSn) that interface directly to common RS-232 or RS-485 transceivers, as shown in the following example schematics. UART0 is the UART that must be used when the serial boot mode via UART is to be available. UART0 may further be used as diagnostic port by Hilscher firmware.

This example shows connection for UART0. UART1 and UART2 are connected equally.

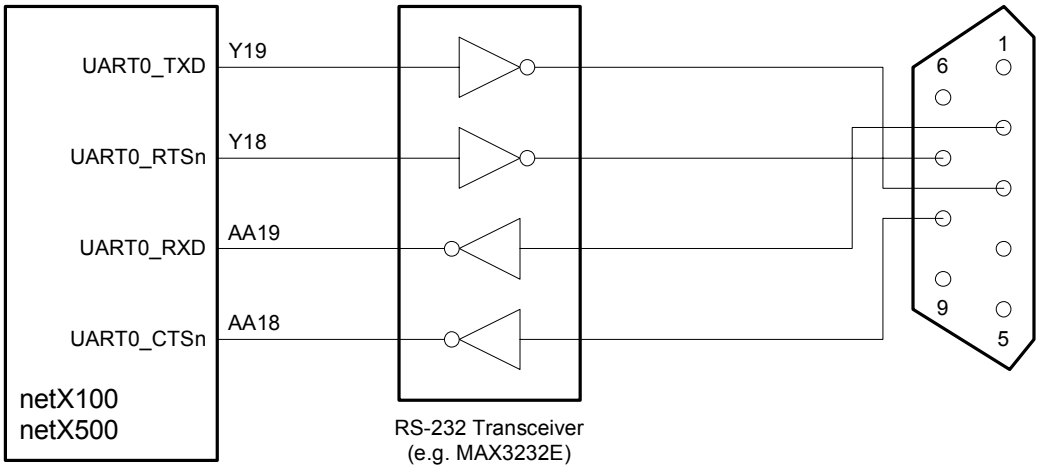


Figure 67: netX100/500 UART0

If UART0 is not be used and the USB port is to support the serial boot mode, then the following external pull-up resistors are required on netX100/500 UART0 pins:

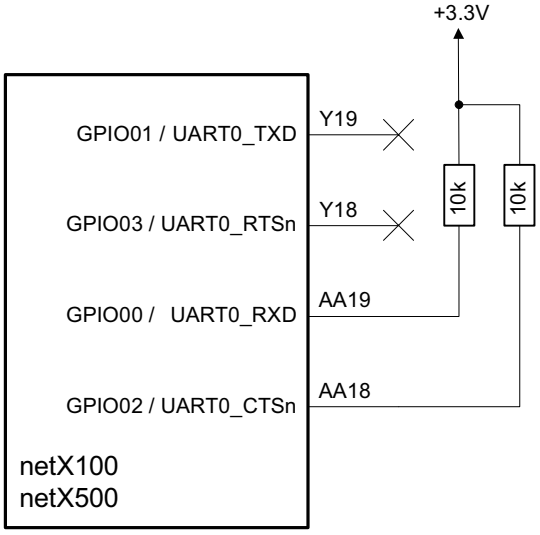


Figure 68: netX100/500 UART0 Unused

If UART0 is completely unconnected and the netX is configured for serial boot mode, then the ROM boot loader misinterprets the low level on RXD and CTSn (caused by the internal pull-downs on these unconnected pins) as a connection attempt through the UART0 port and switches to UART mode. The serial boot mode can then not be used via the USB port and can hence not be used at all (as UART0 is of course not accessible when not connected).

4.9 USB

The netX provides a USB 1.1 compliant USB interface that can either be used in device mode (Downstream Port) or in host mode (Upstream Port). The device mode is commonly used to connect the netX to a PC, while the host mode allows to access for example memory devices like USB-sticks, etc.

Both modes require different external circuits that are described in the following chapters. Though the netX is equipped with separate power supply pins for the USB interface (USB_VDDIO, USB_VDDC and USB_GND on the netX100/500), these pins can simply be connected to the corresponding voltage rail and do not require any special filtering etc, however please regard the following note related to the USB_VDDIO (+3.3 V) supply:

Note: The worst case short circuit current that may flow through the netX USB Buffers can reach 170mA when a short circuit on the USB cable occurs. When the design is to continue working under that condition, the netX VDDIO rail (+3.3 V) must either be able to deliver this additional current, or the USB_VDDIO supply must be connected to the 3.3 V rail through an appropriate fuse or current limiter (PTC) or a separate supply!

Like ALL power supply pins on the netX, the USB power supply pins must always be connected, even when the USB interface is not used!

4.9.1 Device Mode

4.9.1.1 Simple Circuit

The device mode is the commonly used mode of the netX USB interface and allows to connect the netX to a PC (in serial boot mode), which can then download and flash firmware, read and modify register values and run hardware test applications by the help of freely available software tools from Hilscher.

In order to be able to use this handy and yet simple debug and service connection, the implementation of a USB device port is always recommended whenever allowed by board size constraints

Note: With netX100/500 designs that do not use the UART0 port of the netX, please make sure, that the required pull-up resistors on UART0 are present, otherwise the serial boot mode via USB won't work (see chapter 4.8 for details).

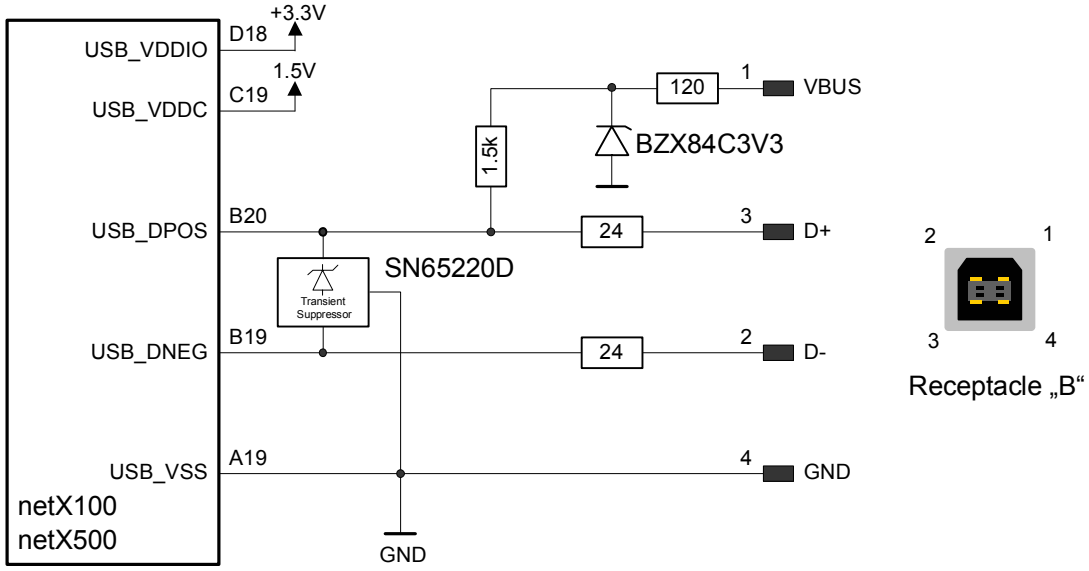


Figure 69: netX100/500 USB DOWNstream Port (Device Mode)

The shown example schematics show the standard circuit as it is most commonly used, also with many Hilscher netX designs. Whenever a reconnect to the USB host is necessary, this circuit requires unplugging and replugging the USB cable, since USB host and netX detect a connect through the 1.5k pull-up resistor that becomes active when the cable is plugged in on both sides. If your application requires reconnecting without unplugging/replugging, the pull-up resistor must be made “switchable”, by using the advanced circuit (which allows to simulate manual plugging/unplugging), described in the following chapter.

4.9.1.2 Advanced Circuit

The circuit shown in the previous subchapter is perfect when the USB port is only used along with the serial boot mode of the netX.

However, when using the USB port also during normal operation of the device (e.g. for diagnostic purposes), the following problem may occur:

Since the pull-up resistor on the D+ line, that lets the USB host (PC) detect the connection of a USB device (netX), is automatically activated when the netX device is plugged to the hosts USB port, chances are that the firmware of the netX device has not yet initialized the netX USB port. In that case, the host detects an unknown USB device and the USB cable needs to be disconnected and reconnected again after the netX USB port has been initialized by the firmware. If this is not acceptable, the following circuit(s) should be used:

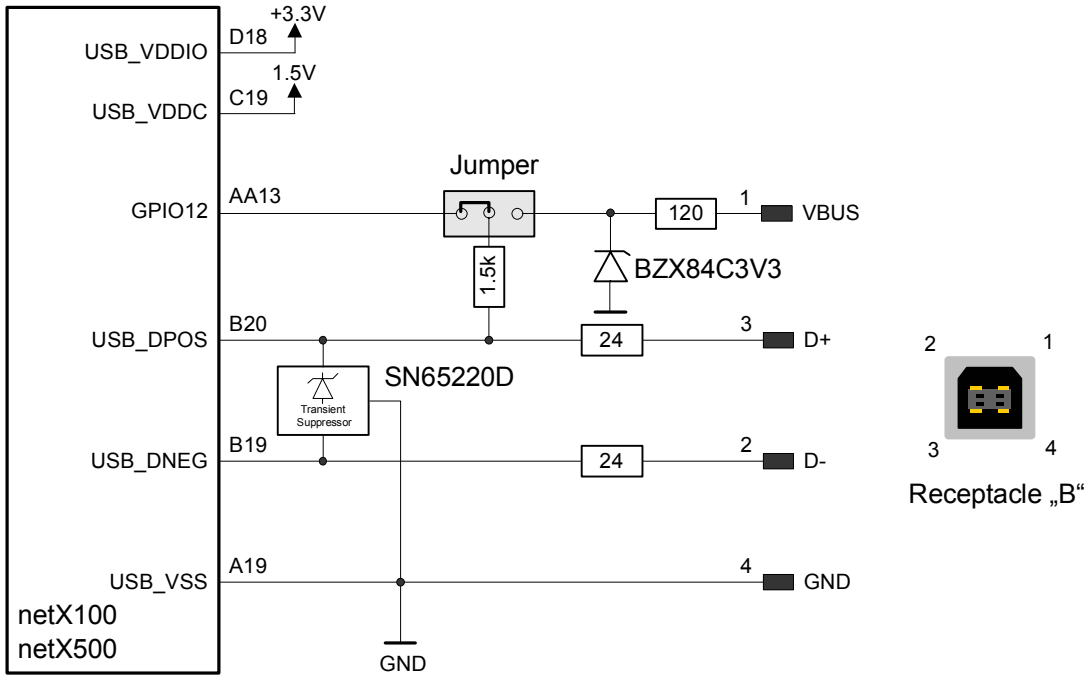


Figure 70: netX100/500 USB DOWNstream Port (Device Mode), Advanced Circuit

When using the circuits shown above, accessing the netX through the USB serial boot mode will not work, since the serial boot mode is handled by the netX ROM loader and the ROM loader will not activate GPIO12 hence the PC will not detect a connected USB device. For that reason, the circuit includes a jumper that allows the pull-up to be connected to the 3.3 V sourced through the USB Bus. For devices where using such a jumper or a switch is not applicable, a special non-standard diagnostic cable, including an external pull-up on D+ must be used, when the netX needs to be accessed in USB serial boot mode!

4.9.2 Host Mode

In host mode, the netX500 USB port can be used to access other USB devices, like USB-Sticks, Memory Card Reader, etc. It is most commonly used along with Windows CE images but may however basically be accessed by any operating system (see chapter 3.2 for information on current driver support!).

In host mode, the design must supply the +5 V bus voltage for the USB port. The most simple circuit would directly deliver the +5 V to the corresponding pin of the USB receptacle, however the use of a suitable power switch is recommended, as shown in the following example schematics. Such a power switch allows to detect short circuit or over current conditions and turn off the USB Bus voltage in such a case. It also provides the possibility to disconnect and reconnect devices, by turning off and on the bus voltage.

Two additional signals for connecting such a power switch to the netX have been defined:

Function	**Pin name and number**	
USB power switching	GPIO 12	AA3
USB over current detection	GPIO 13	Y13

Table 32: Additional USB Signals

GPIO12 has also been assigned another standard functionality, which however is not a conflict, since the other assignment refers to the USB device mode which can only be used alternatively to the Host mode.

The USB Bus power switching signal (VUSB_ONn) is used to turn on and off the USB Bus power. When used, this (active low) signal is to be connected to the enable input of a power switch.

The USB over current detection signal (USB_OCn) is used to signal a USB over current condition detected by the power switch to the netX. When used, this (active low) signal is to be connected to the over current output signal of a power switch.

Note: Please note, that there is no inherent hardware support for the above mentioned signals. All functions associated with these signals are only available, when supported by firmware!

The +5 V power supply must be dimensioned to meet the power requirements of the USB devices that are to be connected to the netX (some devices require up to 500mA when fully operational!). Please also consider the USB V1.1 specification for further details.

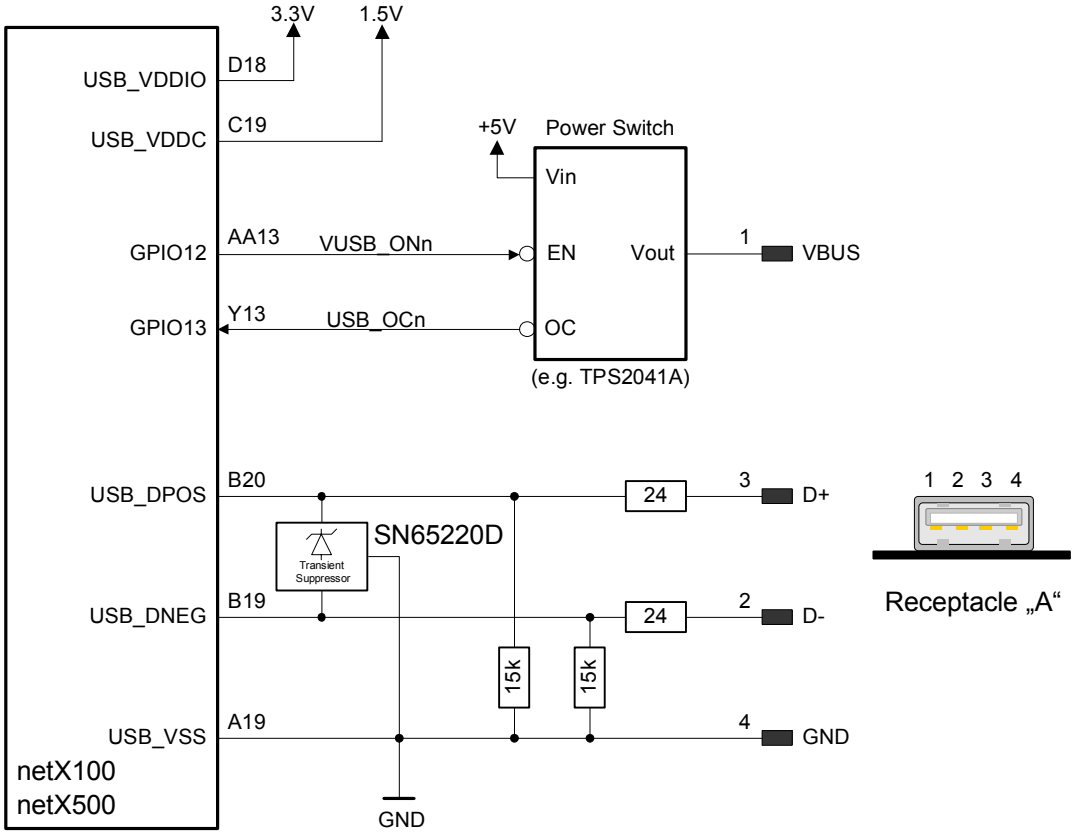


Figure 71: netX100/500 USB UPstream Port (Host Mode)

4.10 Ethernet Interface

netX100/500 have two integrated Physical Layer Units (PHYs) for Ethernet Communication, that allow to build systems with two Ethernet ports, while using only a few external components like pull-ups and transformer(s).

The PHYs can be operated in two basic modes, which are twisted pair (10BASE-T/100BASE-TX) and fiber optic mode (100BASE-FX).

4.10.1 Twisted Pair

For 10BASE-T or 100BASE-TX Ethernet communication, the PHY must be connected as shown in the following example schematic (Figure 72).

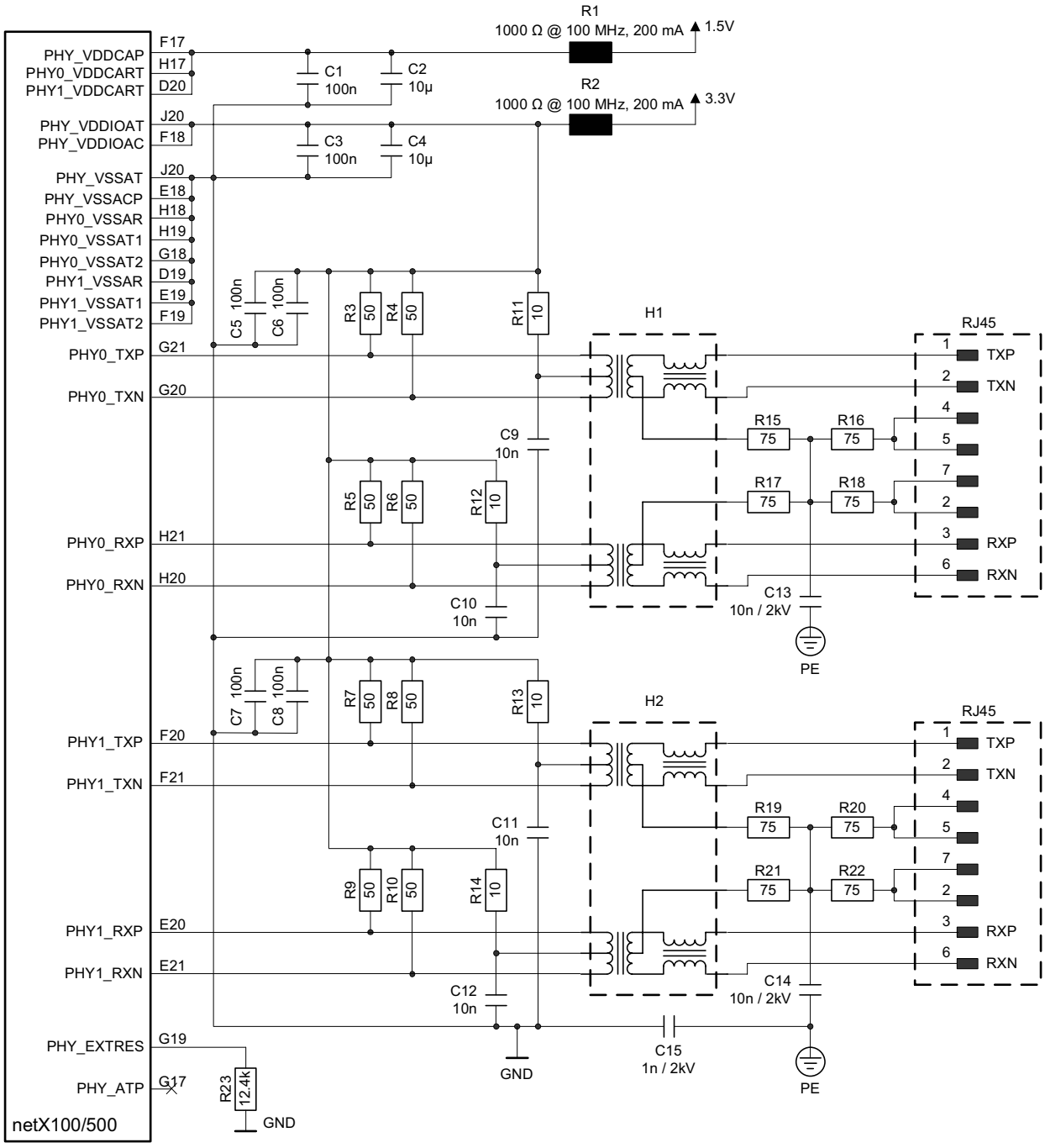


Figure 72: netX100/500 2 Channel Ethernet Circuit (Twisted Pair)

Component	Value	Tolerance	Rating
R1, R2	1000 Ω @ 100 MHz		200 mA
R3, R4, R5, R6, R7, R8, R9, R10	50 Ω	1%	125 mW
R11, R12, R13, R14	10 Ω	1 %	63 mW
R15, R16, R17, R18, R19, R20, R21, R22	75 Ω	1 %	63 mW
R23	12.4 kΩ	1%	63 mW
C1, C3, C5, C6, C7, C8	100 nF		6.3 V
C2, C4	10 µF		6.3 V
C9, C10, C11, C12	10 nF	20%	
C13, C14	10 nF		2 kV
C15	1 nF		2 kV
H1, H2	H1102 HX1188 (Pulse Eng.) TS6121C (Bothhand)		

Table 33: Ethernet Circuit Component Specification

The selected Ethernet transformer(s) (H1, H2, the Table 33 lists three examples) must be 1:1 ratio types with center tap and should be symmetric, which means, that transmit and receive path may be swapped. This is necessary to support the auto-crossover feature that is mandatory for most Real-time Ethernet protocols!

Instead of a separate transformer, secondary side resistors (75 Ω) and RJ45 jack, integrated jacks can be used, that combine all components (plus Status LEDs) in the housing of the jack. They are available as single-channel or 2-channel models.

Hilscher commonly uses a 2 channel integrated jack that also includes the PE capacitor (C3a/C3b) and is available from Pulse Engineering, ERNI and Trxcom:

Pulse Engineering: J8064D628AN

ERNI: 203313

Trxcom: TRJ26204B

If only one Ethernet port is required, this port should be connected according to the following schematic:

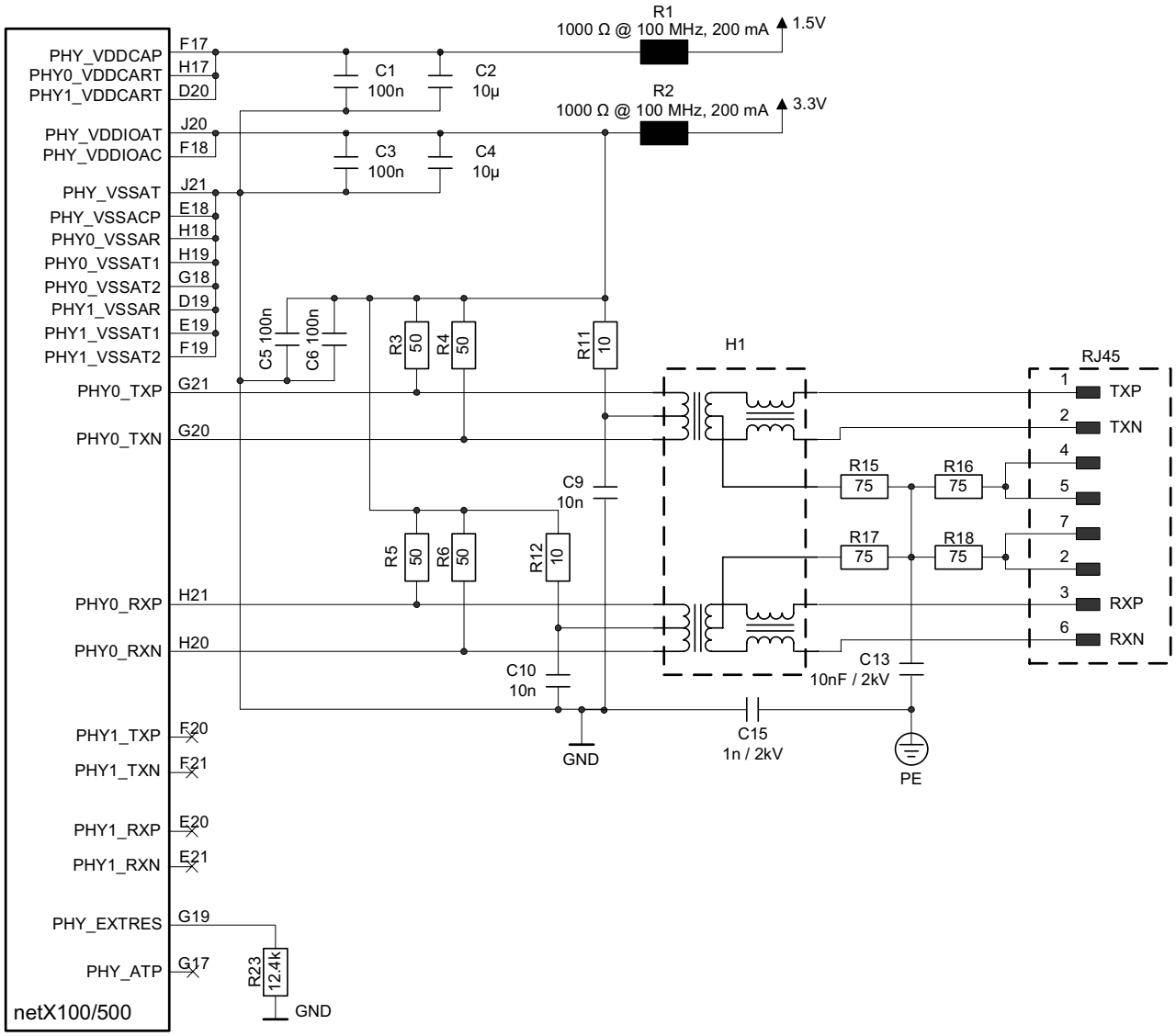


Figure 73: netX Single Channel Ethernet Circuit (Twisted Pair)

Q1: We do not stock 12.4 kΩ resistors and/or 50 Ω resistors. Can't we use 12 kΩ/49 Ω/51 Ω instead?

A1: The specified resistor values are directly taken from the specs of the internal PHY. Using out-of-spec resistor values will result in an out-of-spec Ethernet Interface that can not be guaranteed to work properly under all conditions.

4.10.2 Fiber Optic

For 100BASE-FX Ethernet communication, the netX requires external optical transceivers. Since these transceivers usually work with LVPECL (Low Voltage Positive Emitter Coupled Logic) levels, appropriate signal converters must further be connected between netX and transceivers. The signal converters should be placed as close as possible to the corresponding netX pins and the traces of the differential signal lines between buffers and transceivers should provide an impedance of 50 Ω (100 Ω differential impedance). The signal lines are to be terminated with a Thevenin termination as shown in the following schematics.

4.10.2.1 Transceivers without internal termination

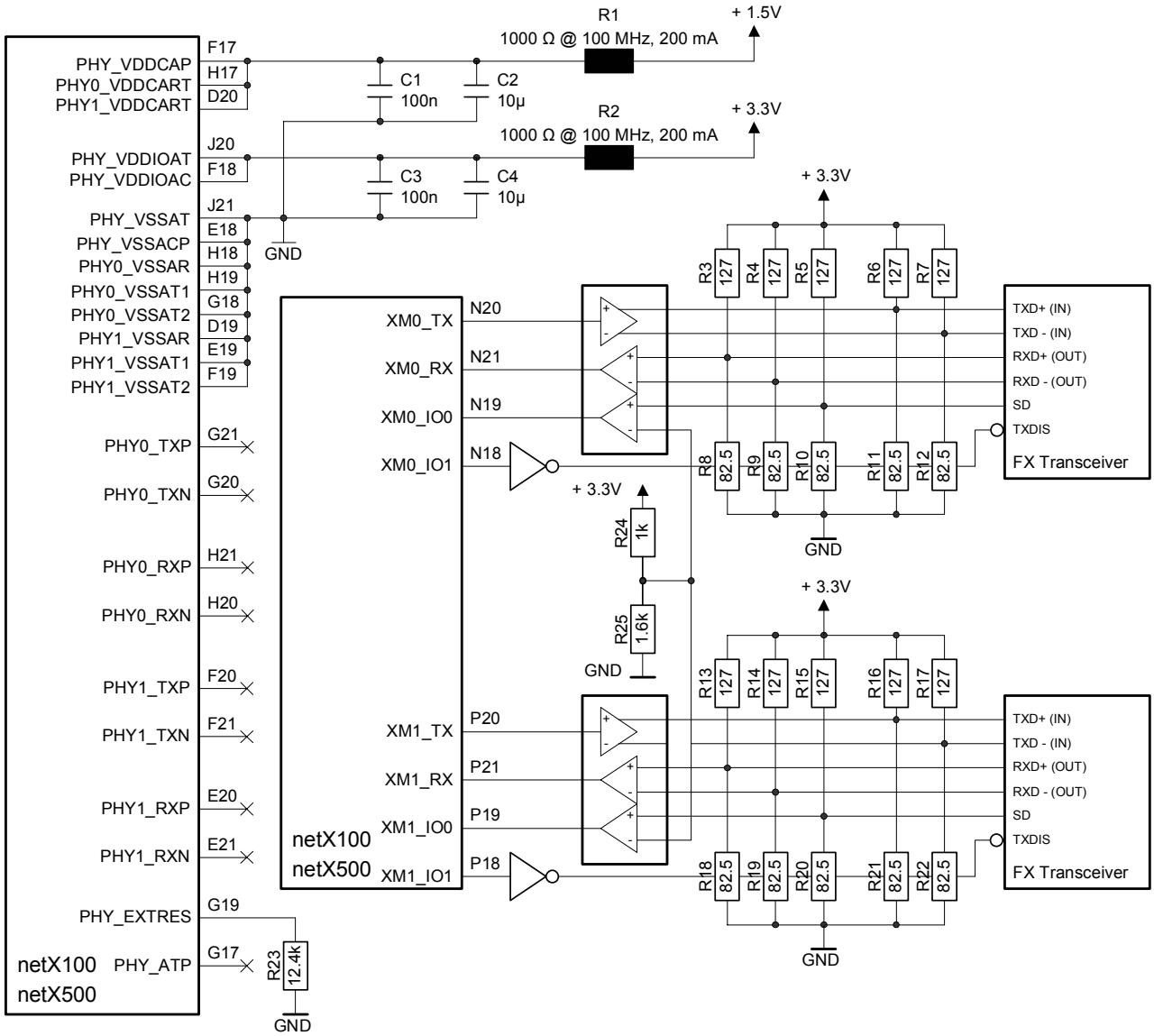


Figure 74: netX100/500 Ethernet Circuit (Fiber Optic)

Component	Value	Tolerance	Rating
R3, R4, R5, R6, R7, R13, R14, R15, R16, R17	127 Ω/130 Ω	1%	
R8, R9, R10, R11, R12, R18, R19, R20, R21, R22	82.5 Ω/82 Ω	1%	
R24	1 kΩ/82 Ω		
R24	1.6 kΩ/130 Ω		
R23	12.4 kΩ	1%	125 mW
C1, C3	100 nF		6.3 V
C2, C4	10 µF		6.3 V
R1, R2	1000 Ω @ 100 MHz		200 mA

Table 34: Fiber Optic Ethernet Circuit Component Specification

4.10.2.2 Transceivers with internal AC-termination

Some Fiber Optic Transceivers already provide an internal AC-termination on the TXDATA input lines, which results in a slightly different circuit:

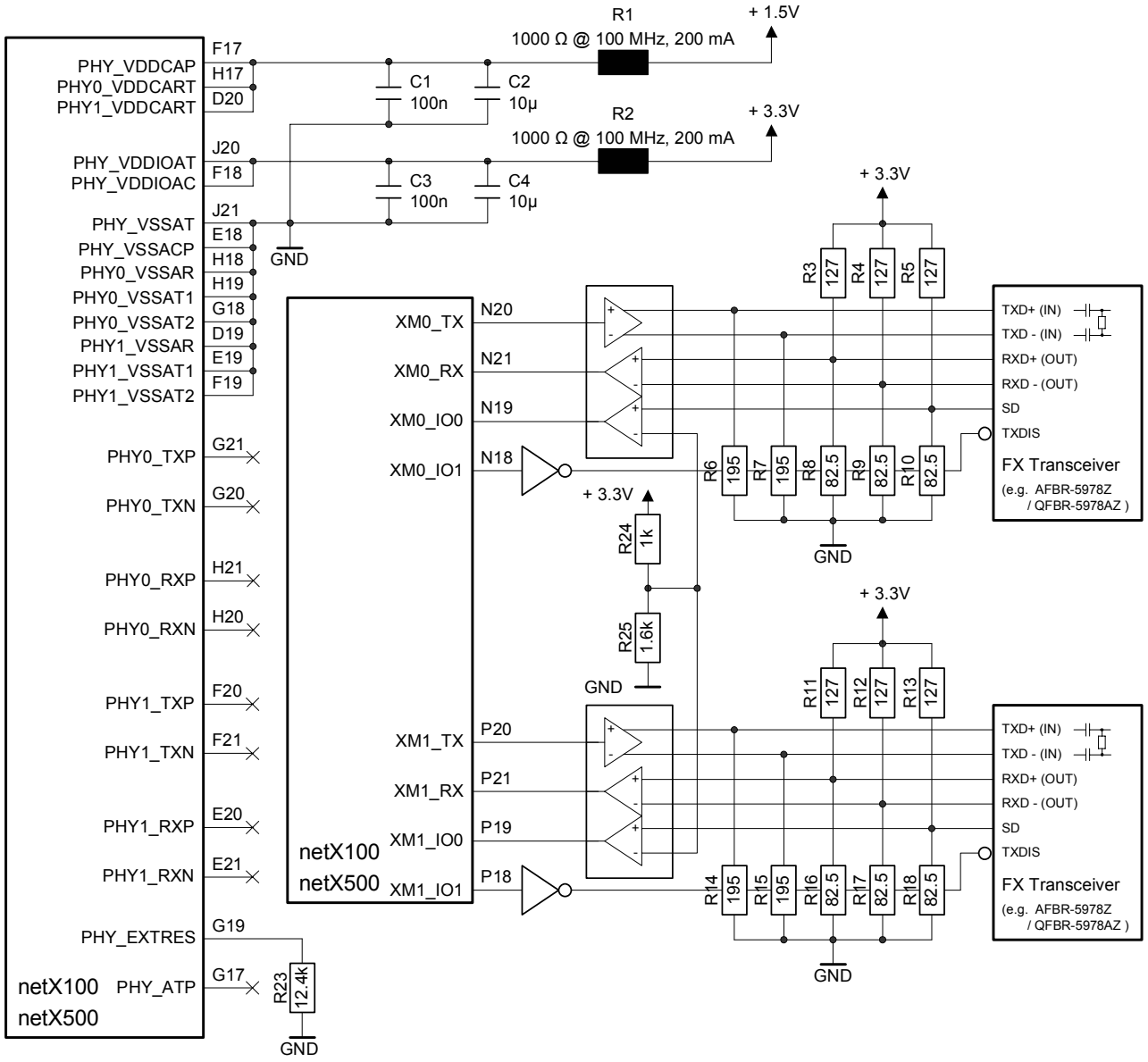


Figure 75: netX100/500 Ethernet Circuit (Fiber Optic)

Component	Value	Tolerance	Rating
R3, R4, R5, R11, R12, R13	127 Ω/130 Ω	1%	
R8, R9, R10, R16, R17, R18	82.5 Ω/82 Ω	1%	
R6, R7, R14, R15	195 Ω	1%	
R19	1 kΩ/82 Ω		
R20	1.6 kΩ/130 Ω		
R23	12.4 kΩ	1%	125 mW
C1, C3	100 nF		6.3 V
C2, C4	10 µF		6.3
R1, R2	1000 Ω @ 100 MHz		200mA

Table 35: Fiber Optic Ethernet Circuit Component Specification (AC-Termination)

Note: The termination resistors shall be placed as close as possible to the end of the signal lines, connected with short traces (no stubs). Due to the large number of resistors, this can hardly be accomplished with standard (0603) resistors. A special Resistor array, containing 8 resistor pairs (127 Ω pull-up, 82.5 Ω pull-down) with common GND and VCC connection in a small BGA package is available from CTS (type RT1250B7) that allows to fulfill this requirement.

The BIAS resistors on the TXD signals (circuit for internally AC-terminated Transceivers) are to be placed close to the beginning of the signal line, which means close to the LVTTl-to-LVPECL level translators.

Please also see the following chapter for hints on component placement.

The 127 Ω/82.5 Ω Thevenin termination results in a bias voltage of 2.0 V and a signal termination of appr. 50 Ω. Alternatively, 130 Ω/82 Ω can be used.

The recommended value for resistors R6, R7, R14, and R15 in the circuit for internally AC-terminated Transceivers depends on the LVTTL-to-LVPECL level translators that are used. 195 Ω is the recommended value when using devices from MICREL. When using other components please consult the device datasheet and appropriate Manufacturers application notes if available.

The values for R24 and R25 are flexible. Important is only the resistor ratio that results in a voltage of 2.0 V (VCC – 1.3 V).

The LVTTL-to-LVPECL signal converters usually have thermal PADs to achieve proper heat dissipation. Make sure to consider the thermal design notes in the datasheet of the appropriate devices.

The power supply (+3,3 V) for the Fiber optic Transceivers should be filtered according to the manufacturer’s recommendation (consult data sheet of transceiver).

Detailed schematics for netX100/500 Fiber Optic interfaces using the AFBR-5978Z/QFBR-5978AZ transceivers with internal AC-termination can be found on page 29 chapter 2.2.3 Fiber Optic with AFBR-5978Z

4.10.2.3 PCB Layout considerations

The following drawings provide hints on proper placement of the Fiber Optic interface components.

Transceivers without internal termination

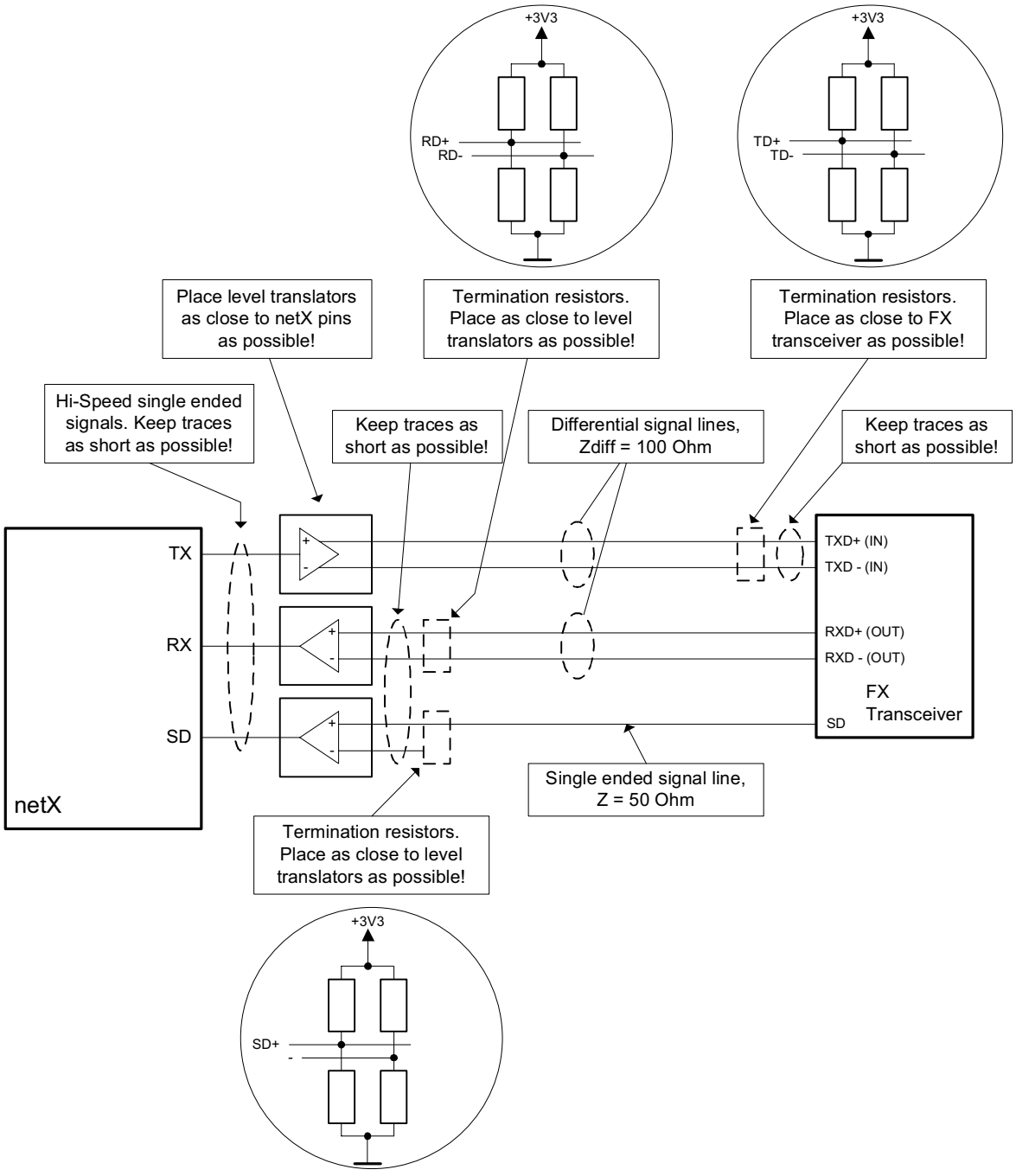


Figure 76: netX Ethernet Circuit (Fiber Optic), Component Placement

Transceivers with internal AC-termination

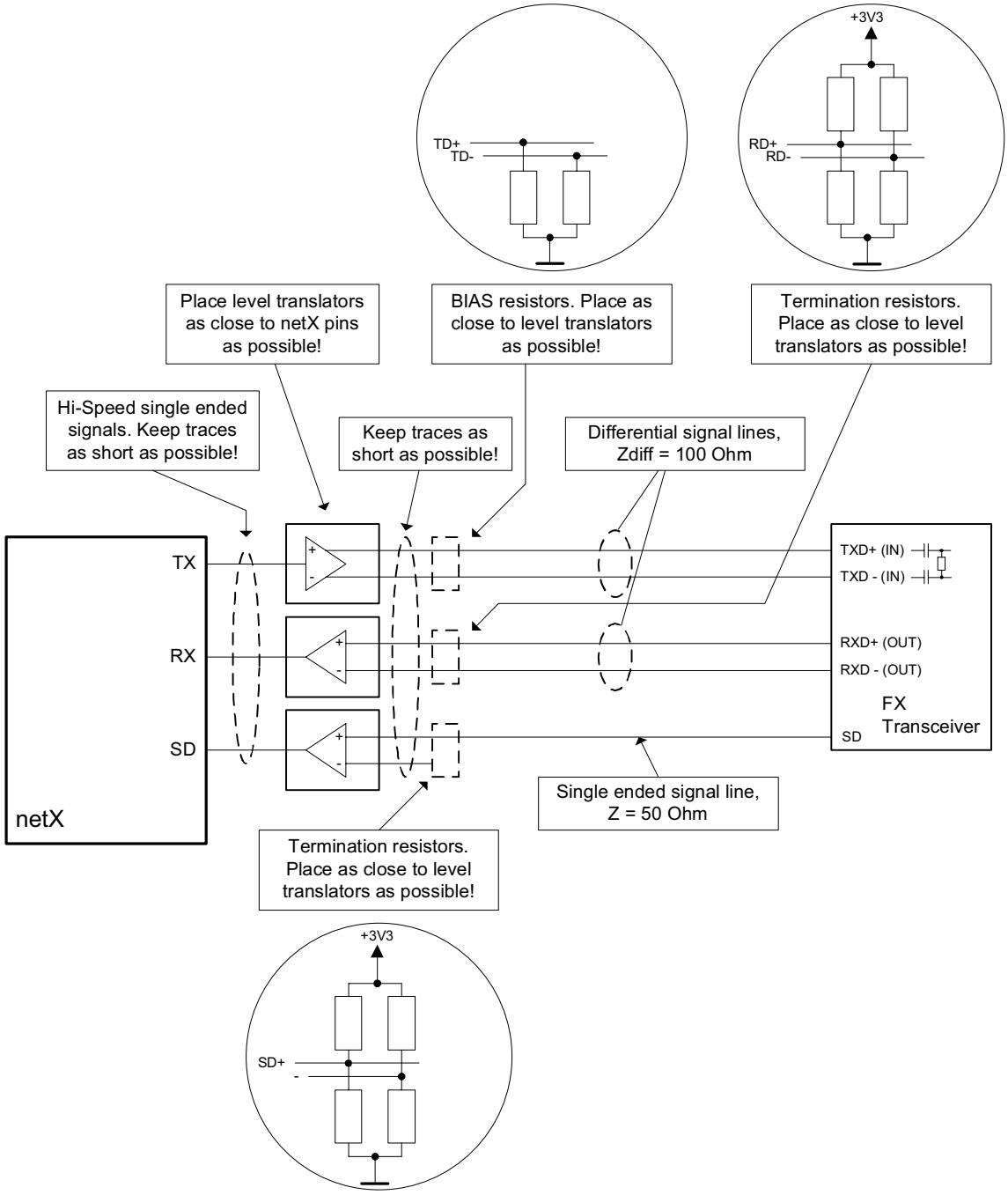


Figure 77: netX Ethernet Circuit (Fiber Optic), Component Placement with AC-Termination

Note:	Please also consider the routing hints for differential signal lines in chapter 4.2.6, which correspondingly also apply to the Fiber optic interface!

4.10.2.4 Diagnostic Monitoring Interface

For netX fiber optic designs that are to be used with Real-time Ethernet protocols, especially PROFINET, the fiber optic transceivers must be equipped with DMI (Digital Diagnostics Monitoring Interface), providing status information, like the AFBR-5978Z or QFBR-5978AZ from Avago Technologies.

While I²C components can usually be connected to a common signal bus as they can normally be individually addressed through their I²C device address, this is unfortunately not possible with the AFBR-5978Z/QFBR-5978AZ, since these devices do not provide a hardware configurable device address. Each AFBR-5978Z/QFBR-5978AZ uses the same device address, which allows coexistence with other I²C components (sensors, memories, etc.) but not with a second AFBR-5978Z/QFBR-5978AZ

Since Real-time Ethernet protocols usually require two Ethernet channels and hence also two Transceivers, solution for the addressing problem had to be found and is described below.

With netX100/500, an external component is required to switch the SDA signal line of the netX I²C controller between the transceiver interfaces, while they again share the same SCL signal line. Since the SDA is a bi-directional signal, the use of an integrated analog switch is the easiest solution.

As the I²C interface of the netX100/500 uses dedicated signal pins, they could be equipped with appropriate internal pull-up resistors (5 kΩ), which basically makes external signal pull-ups obsolete, however two week pull-ups (47 kΩ) are still required to avoid floating of the currently disconnected transceiver SDA signal. The schematic below also shows the secure EEPROM (holding license information and MAC addresses), connected to the same I²C bus, which is no problem, since I²C components of different type or model usually have different device addresses, which is also the case here.

This solution has been tested with a MAX325 analog switch, however any analog integrated switch with similar (or better) characteristics (Rdson, pin capacitance, bandwidth) may be used.

The following figure shows the schematic for netX100/500:

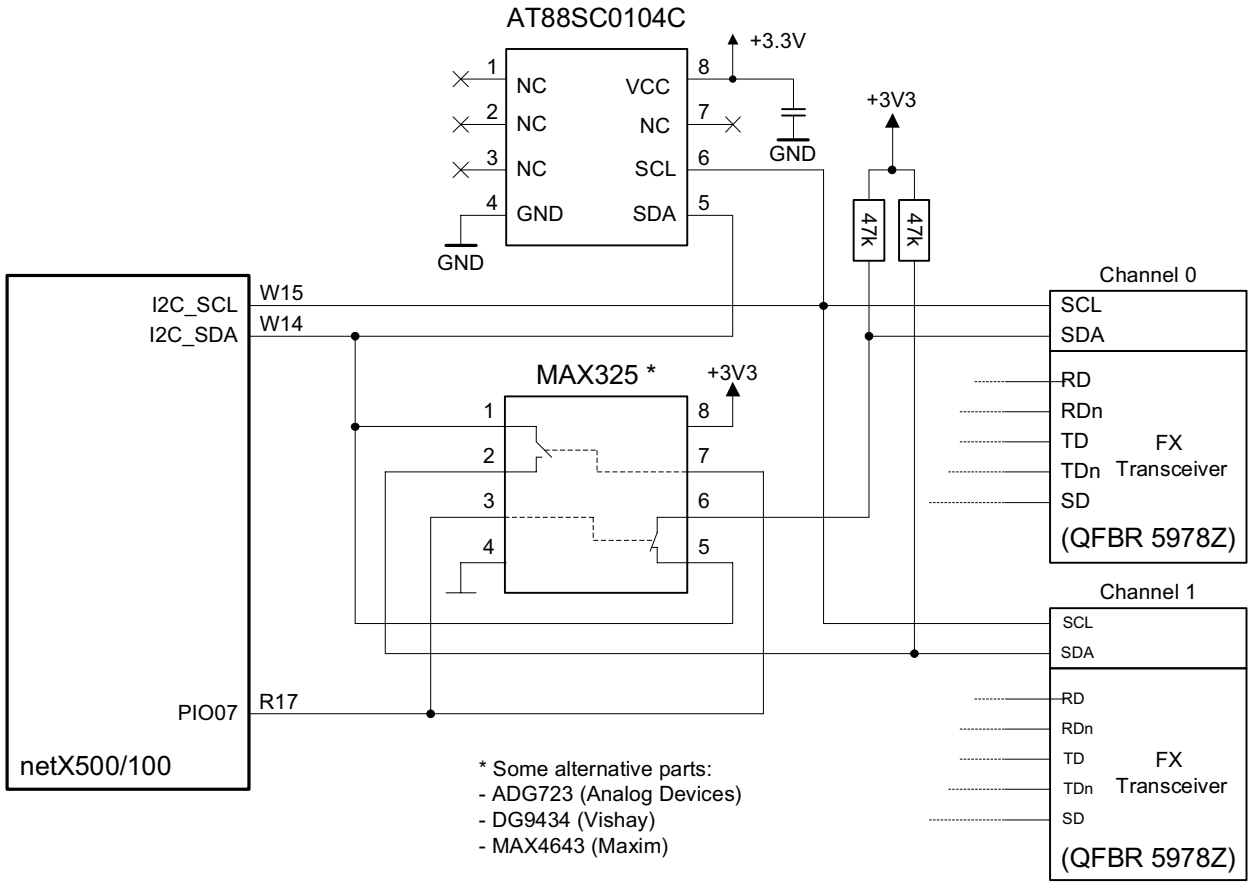


Figure 78: netX100/500 I²C Interface Fiber Optic Transceivers

Software considerations netX100/500

With netX100/500, the level of PIO07 must be set accordingly before accessing the desired transceiver. When PIO07 is set to Output mode and low level, the analog switch will connect the netX I²C SDA signal to the SDA signal of Transceiver 0, while setting PIO07 to Output mode and high level, will let the switch connect the SDA signal to Transceiver 1. The PIO signals PIO00 to PIO31 are controlled by the PIO_OUT and PIO_OUT_EN registers, please refer to the netX100/500 Program Reference Guide for details.

Note:	Please note, that the information above is only relevant when accessing the diagnostic interfaces of the Transceivers without using Hilscher firmware or stacks, since the appropriate functionality and the diagnostic interface communication routines will be integrated in any appropriate software from Hilscher.

4.10.3 Ethernet PHYs unused

If the internal Ethernet PHYs are not used in a netX design, the signal pins (RXP/RXN, TXP/TXN) should simply be left open. However, all power supply pins must still be connected, as well as the reference resistor, as shown on the following schematics:

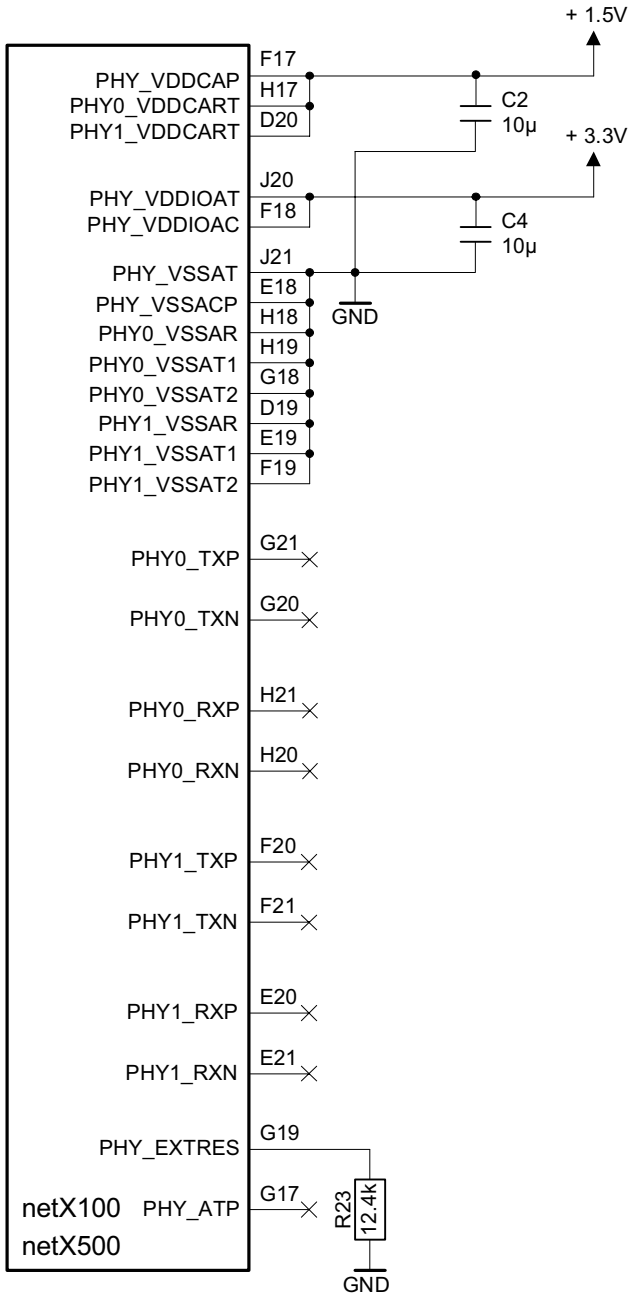


Figure 79: netX100/500 Ethernet Circuit (PHYs Not Used)

4.10.4 Ethernet Status LEDs

Each of the netX Ethernet ports provides two status LED signals: Link and Activity.

The link status LED is lit, when a link has been established on the corresponding Ethernet port, while the yellow activity LED flickers when data is received or transmitted on the corresponding port.

The following table shows the standard pin assignment for the status LEDs

Function	LED color	Pin name an number	
Ethernet Port 0, Link Sat.	green	XM0_IO0	N19
Ethernet Port 0, Activity	yellow	XM0_IO1	N18
Ethernet Port 1, Link Sat.	green	XM1_IO0	P19
Ethernet Port 1, Activity	yellow	XM1_IO1	P18

Table 36: Status LEDs for Ethernet Ports

netX100/500 use the IO signals of XMAC0 and XMAC1 for Ethernet status signaling. The schematics below must be used whenever the design is to be operated with loadable Firmware from Hilscher.

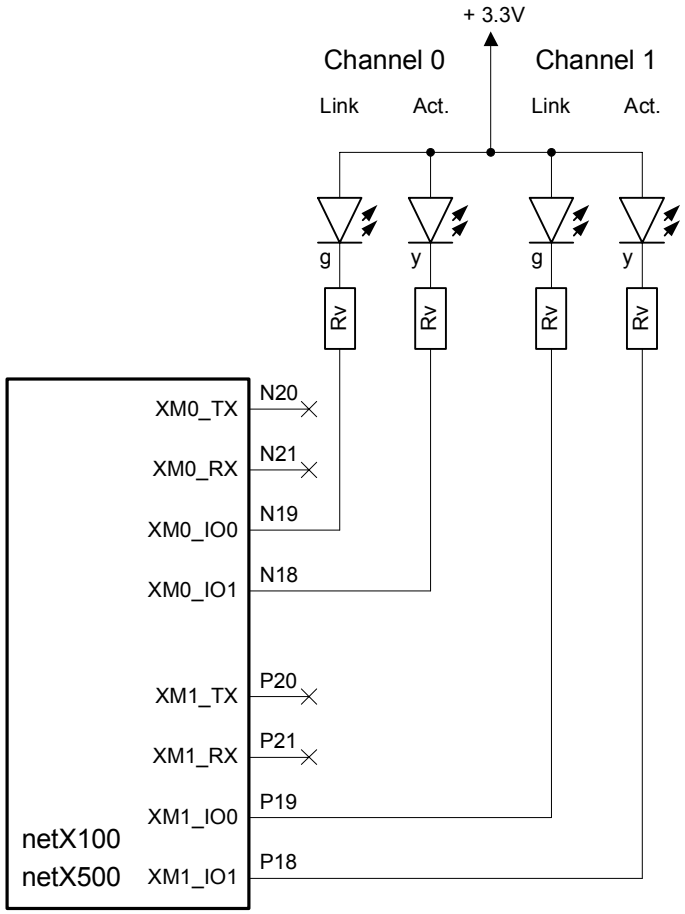


Figure 80: netX100/500 Ethernet Status LED Circuit

4.10.5 Real Time Ethernet

Besides running standard Ethernet protocols, the netX is able to run all current Real Time Ethernet (RTE) protocols. The physical interface for RTE protocols is identical to the standard Ethernet port as described in the preceding chapters, except some additional synchronization signals that may be required depending on protocol and application as well as some additional status LED signals. Both are described in the following chapters.

Note: Regardless if these signals are used in an application, any RTE protocol will always occupy the XPEC3/XMAC3 unit on the netX100/500, hence this unit can never be used for any other purpose like a field bus interface, whenever an RTE protocol is running.

4.10.5.1 Sync signals

The netX100/500 provides up to two additional I/O signals for these purposes, which are generally connected through the XM3_IO0 and XM3_IO1 pins (XMAC3 I/O pins). The following table lists the different protocols and the additional signals:

RTE protocol	Master/Slave	XMAC3 signal	Pin	Function	Type	Remarks
SERCOS III	Device	XM3_IO0	U19	CON_CLK	Out	Configurable
		XM3_IO1	U18	DIV_CLK	Out	Configurable
	Master	XM3_IO0	U19	CYC_CLK CON_CLK	In/Out	Configurable
		XM3_IO1	U18	DIV_CLK		
EtherCAT	Slave	XM3_IO0	U19	Sync 0	Out	
		XM3_IO1	U18	Sync 1	Out	
PROFINET IRT	Controller/Device	XM3_IO0	U19	IO_Output	Out	Trigger for Outputs valid, also used for certification (start of red phase)
		XM3_IO1	U18	IO_Input	Out	Trigger for sample inputs
Ethernet Powerlink	Controlled Node	XM3_IO0	U19	SoC	Out	configurable, “Start of Cycle” received event
		XM3_IO1	U18	-		

Table 37: Additional RTE Sync Signals netX100/500

4.10.5.2 RTE Status LEDs

In Addition to the standard Ethernet Status LEDs (Link Status and Activity), up to four further LEDs (two dual LEDs) have been defined by the different RTE Protocols. The signals used for driving these LEDs (PIO0-3) are identical with the signals used for Field bus status LEDs on XMAC0 and XMAC1 (see also chapter 2.1.3 for pinning details), which however is not really a conflict, since RTE applications usually require both Ethernet ports and a field bus stack can not run on XMAC0 or XMAC1 when these XMACs are already used by an Ethernet application.

Firmware	Label	Dual LED					Label	Meaning
		COM 0			COM 1			
		PIO0	PIO1		PIO2	PIO3		
PROFINET	SF		red			red	BF	SF: System Failure BF: Bus Failure
EtherCAT	RUN	green				red	ERR	RUN: Run ERR: Error
Powerlink	BS	green				red	BE	BS: Bus Status BE: Bus Error
SERCOS III Master	STA	green				red	ERR	STA: Status (Phase) ERR: Error
SERCOS III Slave	S3	green	red				-	S3: orange = red and green at the same time.
EtherNet/IP	MS	green	red		green	red	NS	MS: Module Status NS: Network Status
Open Modbus/TCP	RUN	green				red	ERR	RUN: Run ERR: Error
VARAN	RUN	green				red	ERR	RUN: Run ERR: Error

Table 38: Status LEDs for Real Time Ethernet Applications

All RTE Status LED I/Os are defined as active low.

The following schematic shows, how to connect the LEDs.

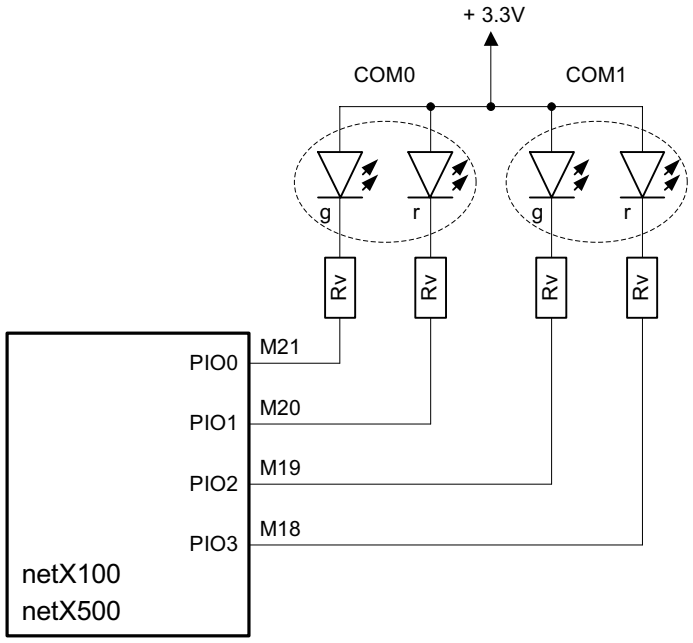


Figure 81: netX RTE Status LED Schematic

The appropriate value for Rv depends on type and color of the LED (standard value = 270 Ω).

Use either two red/green Dual-LEDs or two pairs of single LEDs with the LEDs of each pair placed close to each other for the COM0 and COM1 indicators.

4.11 Fieldbus Interface

The netX controllers are equipped with flexible communication processors (xPEC/xMAC units, also referred to as XC units) that allow to realize virtually any field bus interface on the market, by simply adding the appropriate physical layer circuit.

While the netX500 provides a total of four XC units, the netX100 has three XC units and an additional unit with limited functionality (used for Real-time Ethernet protocols).

The standard XC port for a single channel field bus application on the netX100/500 is xMAC2, as this still allows using the Ethernet interface on XC0 and XC1. xMAC3 is generally not available for field bus interfaces on the netX100 and can further not be used for field bus interfaces on the netX500 when running a Real-time Ethernet protocol!

All common field bus interfaces are serial interfaces with a Transmit and Receive signal, while some of them use an additional control signal or status signal, hence the xMAC units that directly connect to the field bus physical layer circuit provide an XMi_TX, an XMi_RX and two I/O signals (XMi_IO0 and XMi_IO1), whereas currently only one of them is used by current field bus interfaces.

Each XC unit also provides a clock input/output signal (XMi_ECLK) that either allows synchronizing external hardware to the XC clock or feeding an external clock to the XC unit. Both options are currently not used.

The pinning of the XMAC signals is fixed for netX100/500. For that reason it is strongly recommended to adopt the proposed MMIO assignment for the XMAC signals as shown in the following example schematic.

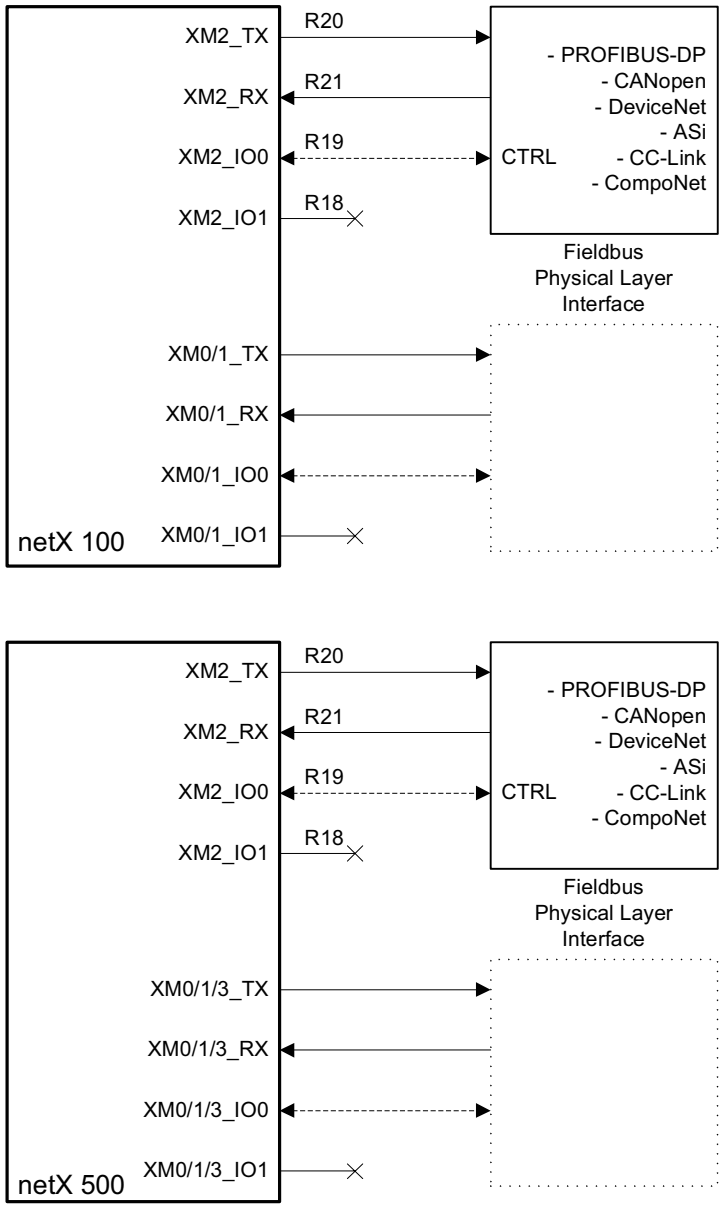


Figure 82: netX100/500 Fieldbus Interface

4.11.1 AS interface Master

As there are no pure ASi transceivers on the market and generating the sin² signals is not trivial, an ASi interface with the netX still requires the use of an appropriate ASi-ASIC (⟶ ZMD). The following simplified example schematic shows the physical layer circuit for a netX AS-interface, still using the discontinued A2SI chip which is replaced by the ASI4U. More detailed schematics can be found in the Reference Section in Chapter 5.

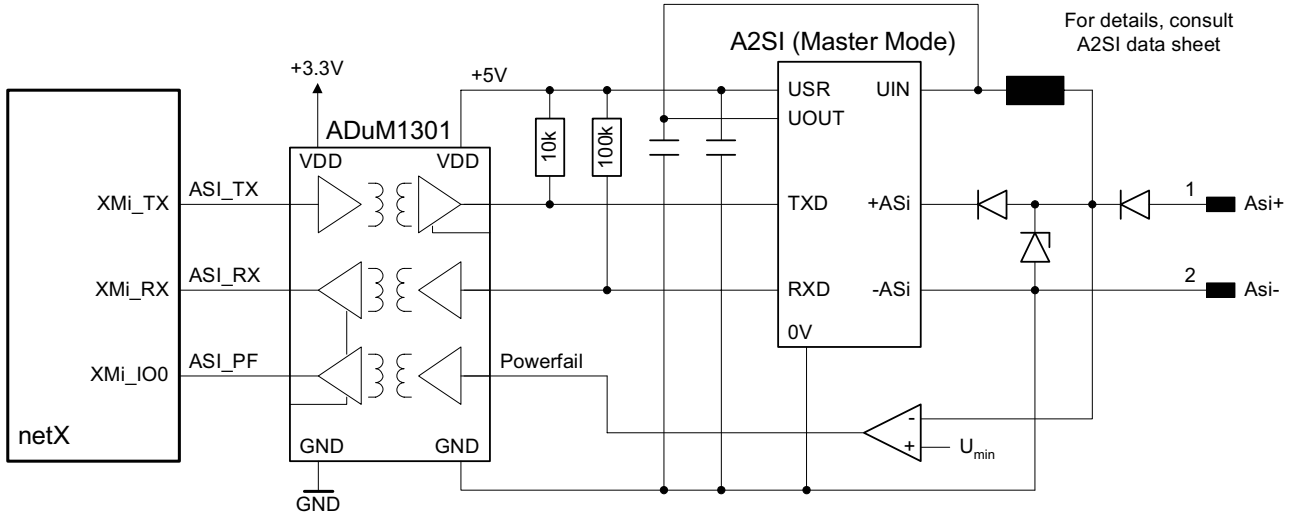


Figure 83: Basic Circuit for netX AS-Interface

4.11.2 CANopen Interface

A CANopen interface can be implemented as shown in the following schematics. More detailed schematics (pin numbers) can be found in the in the Reference Section in Chapter 5.

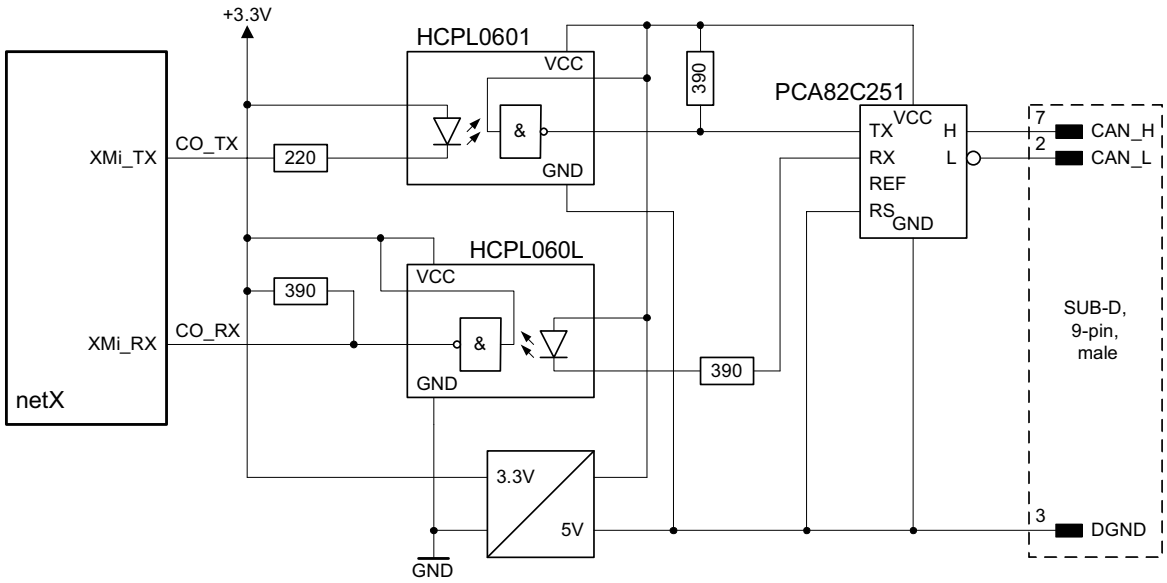


Figure 84: Basic Circuit of netX CANopen Interface

4.11.3 CC-Link Interface

A CC-Link interface can be implemented as shown in the following schematics, which are based on the reference schematics from the CC-Link specification. The purpose of the AND gate in the RX signal is simply to convert the 5V signaling voltage from the photo coupler to 3.3V level.

More detailed schematics (pin numbers) can be found in the in the Reference Section in Chapter 5.

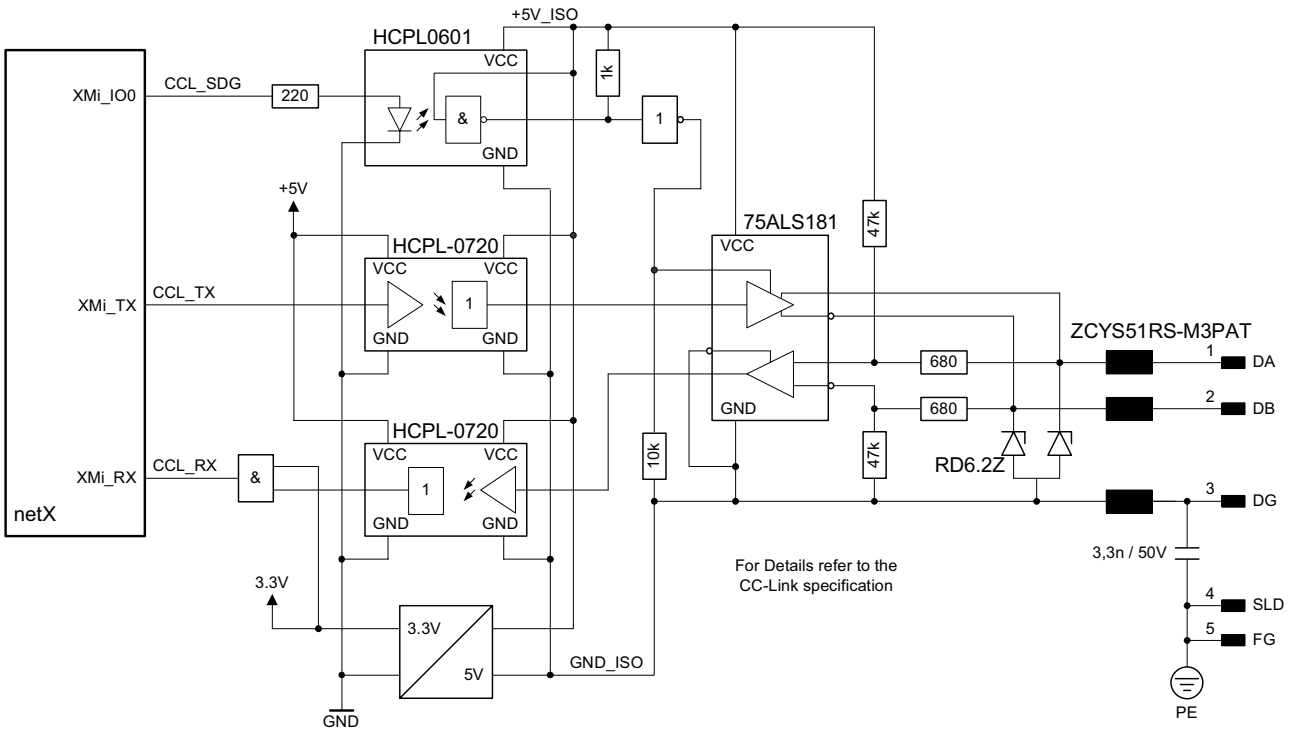


Figure 85: Basic Circuit for netX CC-Link Interface

4.11.4 CompoNet Interface

A CompoNet interface can be implemented as shown in the following schematics, which are based on the reference schematics from the CompoNet specification. The purpose of the AND gate in the RX signal is simply to convert the 5V signaling voltage from the AD51/025 to 3.3V level.

More detailed schematics (pin numbers) can be found in the in the Reference Section in Chapter 5.

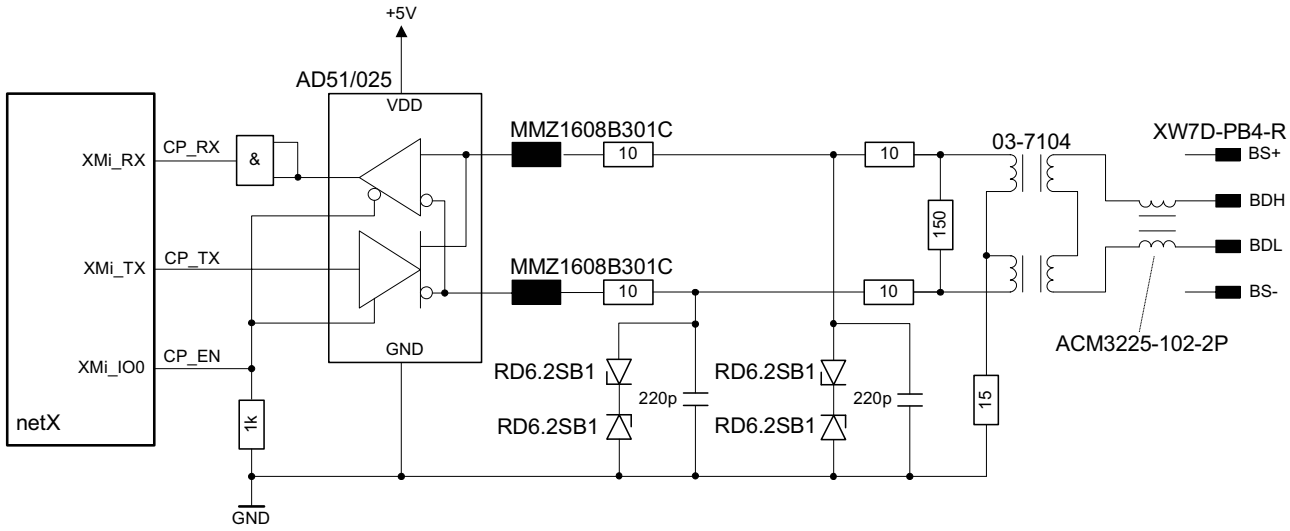


Figure 86: Basic Circuit for netX CompoNet Interface

4.11.5 DeviceNet Interface

A DeviceNet interface can be implemented according to the following simplified schematics. More detailed schematics can be found in the in the Reference Section in Chapter 5.

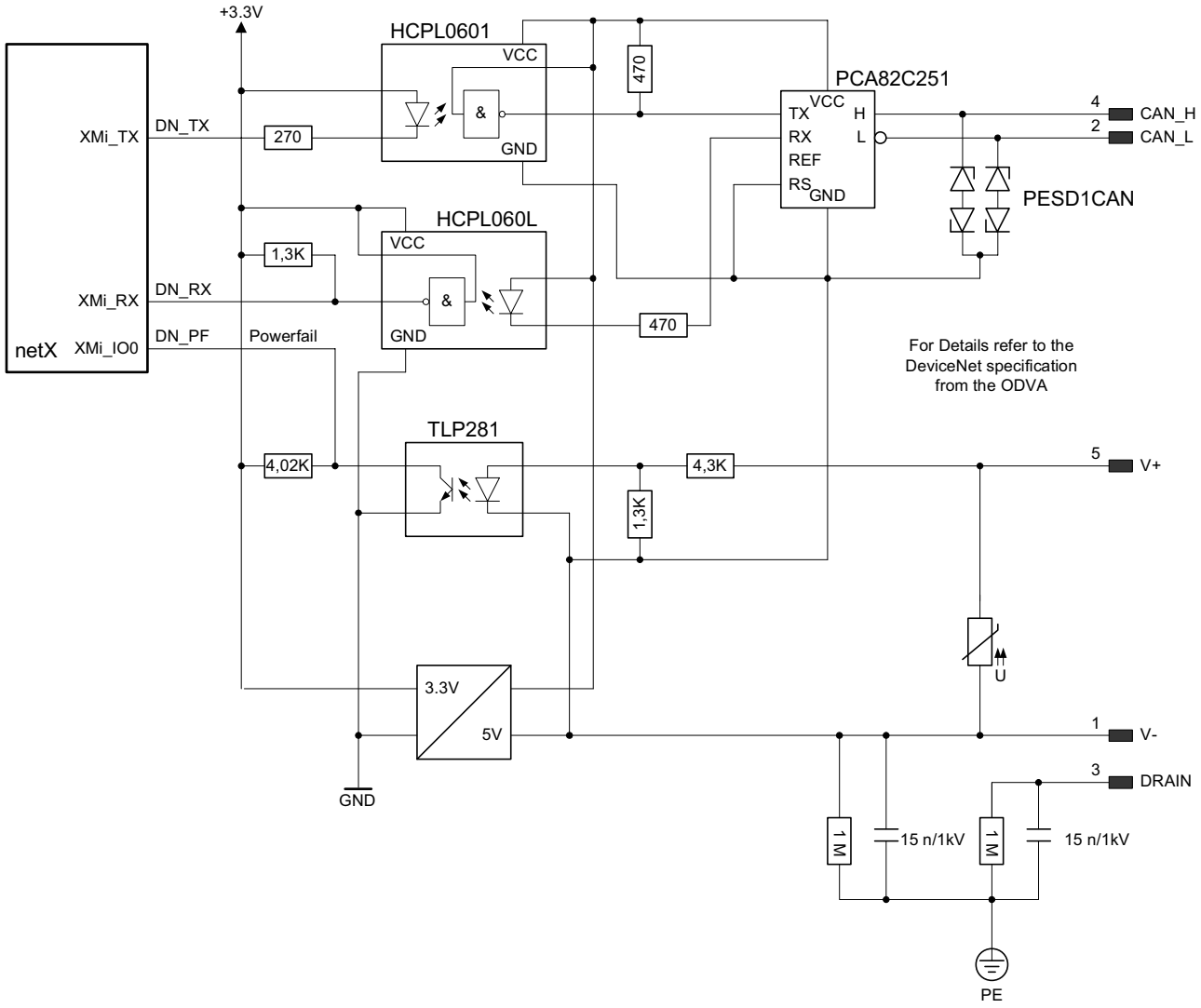


Figure 87: Basic Circuit for netX DeviceNet Interface

4.11.6 PROFIBUS Interface

A PROFIBUS interface can be implemented according to the following simplified schematics. More detailed schematics can be found in the in the Reference Section in Chapter 5.

Note: The (isolated) 3.3V to 5V DC-DC converter should either be a regulated model or should be equipped with an appropriate downstream LDO regulator. Otherwise, the secondary voltage may be too high, resulting in out-of-spec signal levels on the PROFIBUS line!

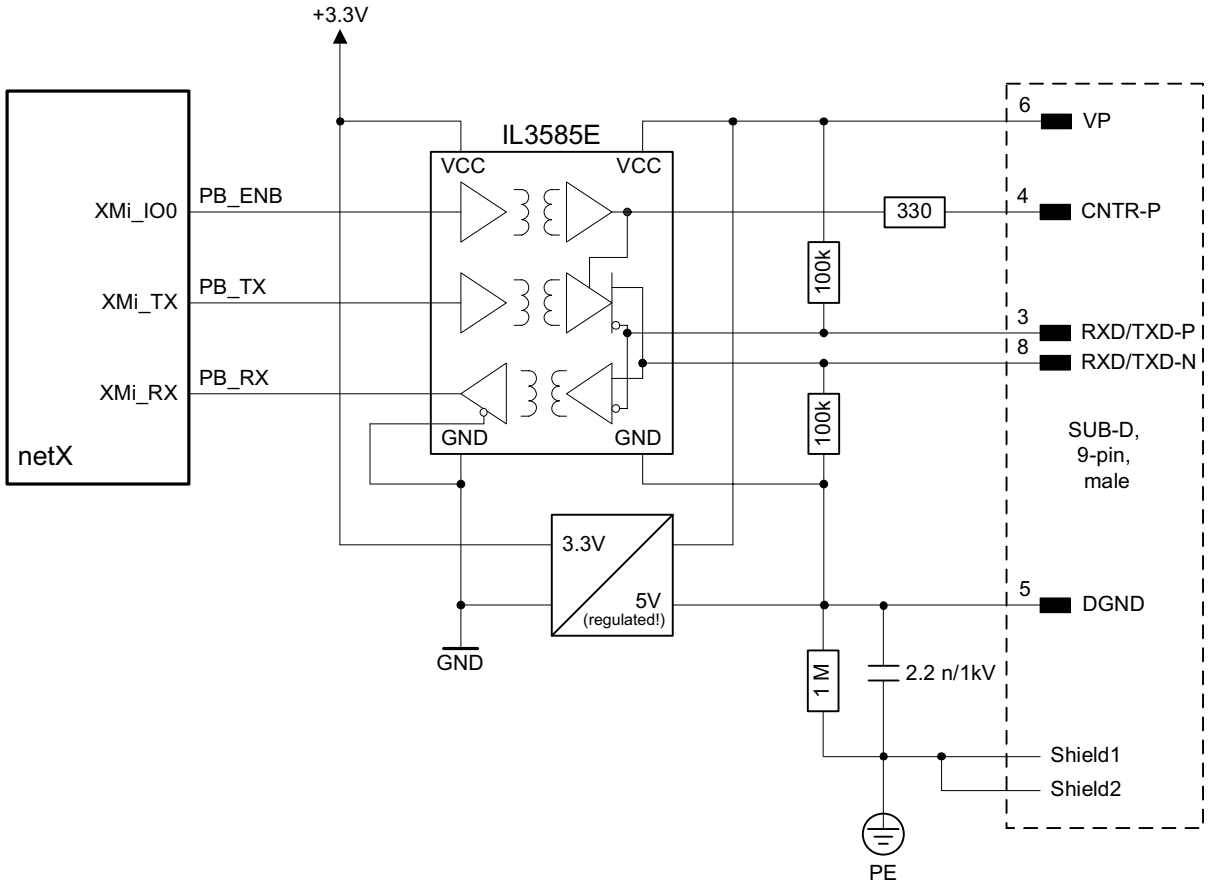


Figure 88: Basic Circuit for netX PROFIBUS Interface

4.11.7 Fieldbus Status LEDs

For each of the up to 4 possible Field bus ports, two[1)] status LED signals are defined, which function depends on the type of Field bus interface.

Function	Pin name and number netX100/500	
Field bus 0, COM0	PIO0	M21
Field bus 0, COM1	PIO1	M20
Field bus 1, COM0	PIO2	M19
Field bus 1, COM1	PIO3	M18
Field bus 2, COM0	PIO4	M17
Field bus 2, COM1	PIO5	N17
Field bus 3, COM0 Note 2)	PIO6	P17
Field bus 3, COM1 Note 2)	PIO7	R17

Table 39: Status LEDs for Fieldbus Ports

Fieldbus Protocol	LED Name	LED Color	
PROFIBUS-DP Master and Slave	COM	green	red
CANopen Master and Slave	CAN	green	red
DeviceNet Master and Slave	MNS	green	red
AS-Interface Master	COM	green	red
CC-Link Slave	L RUN/L ERR	green	red
CompoNet Master and Slave[1)]	Two LEDs required by spec: MS and NS	green	red

Table 40: Fieldbus Status LED Colors

Note: 1) CompoNet requires four status LED signals

2) Field bus 3 is generally not available on netX100 and also not on netX500 when running RTE applications, requiring hardware synchronization!

4.12 A/D Converter

When using the AD converters of the netX100/500, the supply voltages for the ADC circuit and its reference voltages should be filtered to avoid power supply noise to influence accuracy of the sampled values.

The following circuit shows a recommendation, however the filter components may have to be modified, to meet the special requirements of a particular system.

Note: All filter components are application specific and should be evaluated. Shown values are only an example!

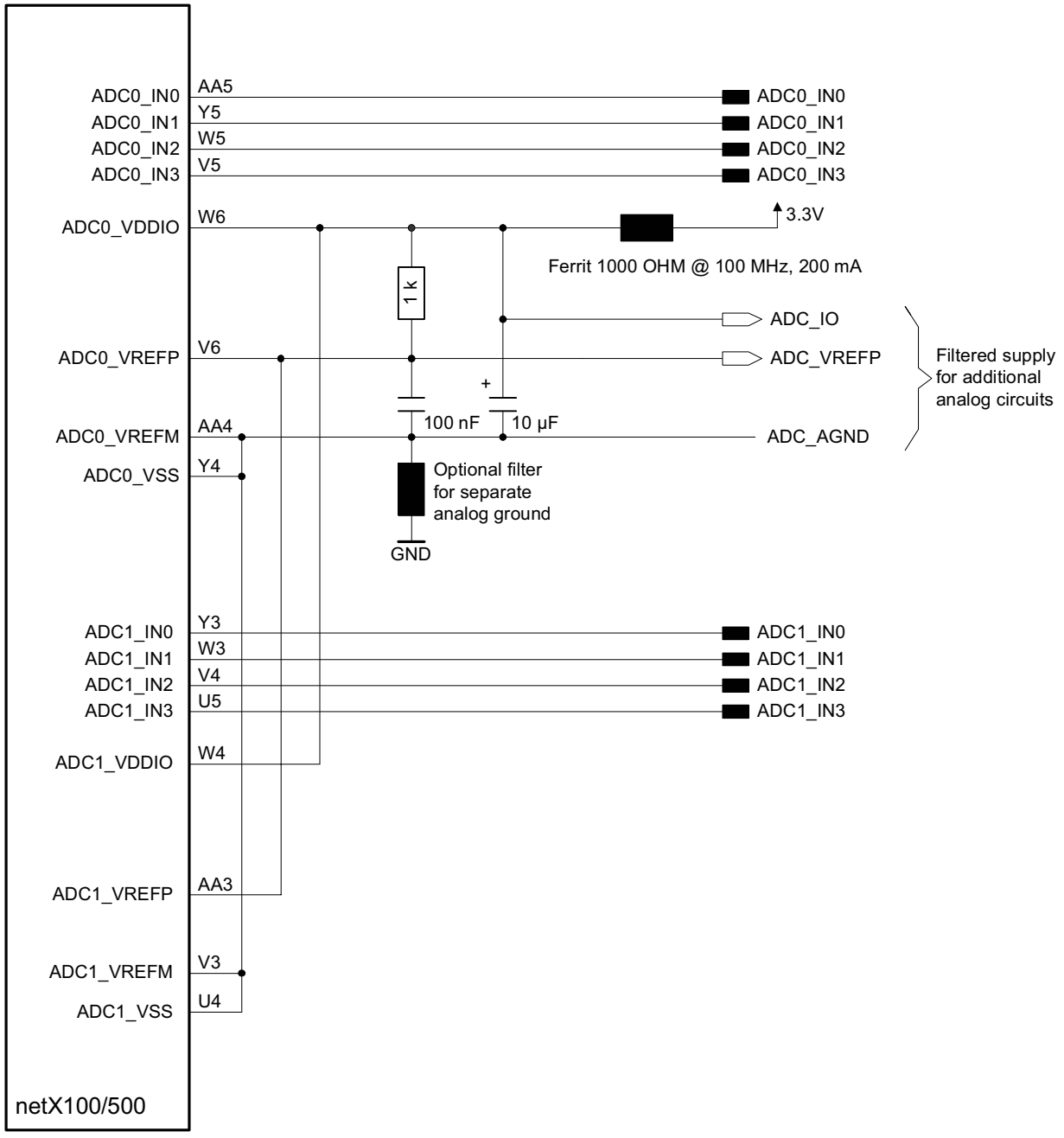


Figure 89: ADC Circuit

If the second ADC-channel is not used, it still needs to powered, as shown in the following schematic:

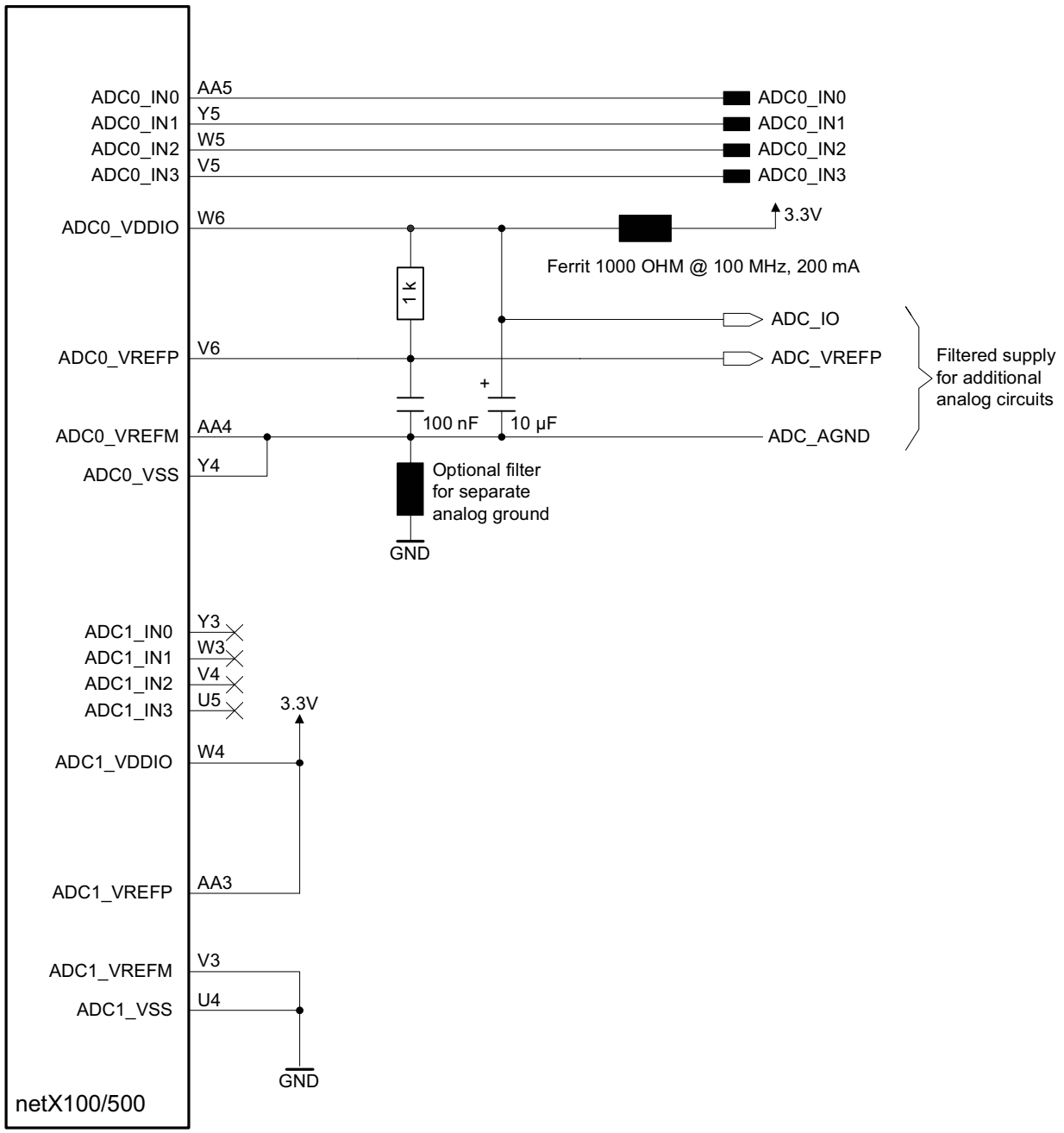


Figure 90: ADC Circuit, One Channel Used

If the ADCs are nit used, they still need to be powered, as shown in the following schematic:

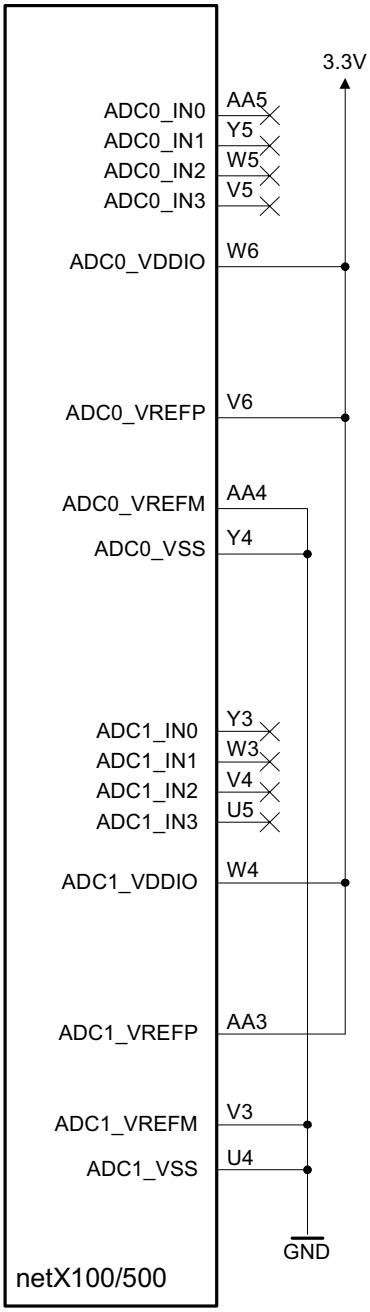


Figure 91: ADC Circuit, ADCs Unused

4.13 PWM Interface

The netX100/500 is equipped with two independent PWM units, each with a three phase complementary output (e.g. for controlling electric motors) and an additional single phase output for supplying resolver units.

Due to the necessarily limited pin resources of the netX controllers, the use of pin sharing was inevitable, which implies some restrictions regarding the parallel use of certain chip resources.

The following picture roughly shows the dependencies and restrictions that may come into play when using the PWM and/or Encoder interface and communication channels:

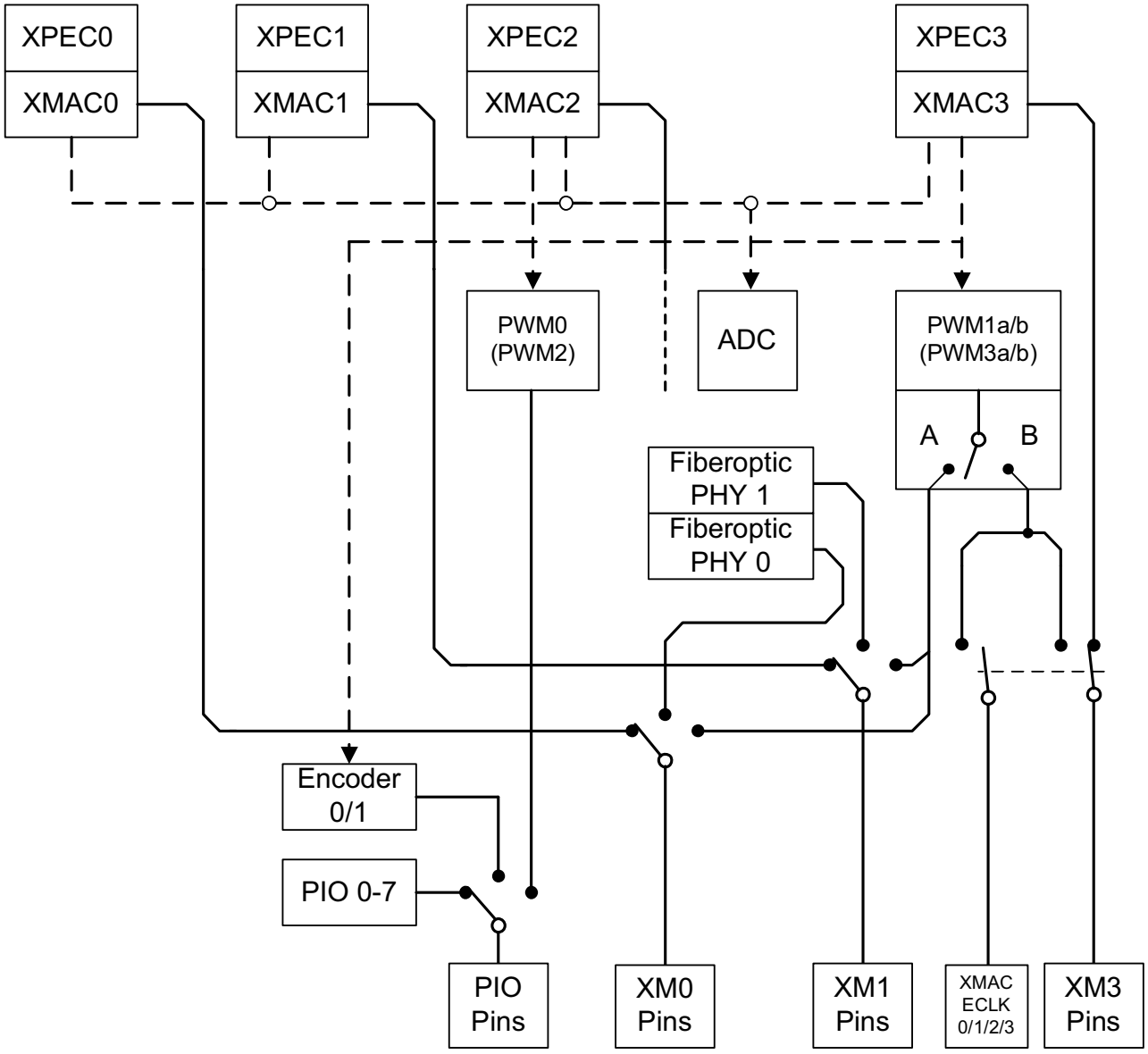


Figure 92: PWM/Encoder/xMAC/xPAC Resource Sharing

The picture shows the following major restrictions:

Designs that also make use of the encoder interface can only use unit PWM1. Since using PWM1 rules out the use of xMAC3 (which must be disabled when controlling these units from the ARM

CPU), Real Time Ethernet designs with Encoder and PWM interface are not possible (or would require customized xMAC code, that can only be created by Hilscher (custom development)).

When using unit PWM0, xMAC2 must be disabled (allowing this unit to be controlled by the ARM CPU), which excludes the use of a field bus protocol on xMAC2 or would again require customized xMAC code.

Further, using PWM0 or Encoder interface blocks PIO0-7 which are the standard resources for RTE and field bus status LEDs. In that case, other I/O resources must be used for the LEDs.

The external circuit for PWM applications strongly depends on the application, hence examples are not provided here. The appropriate pins for the signals of the PWM units which are PWM0E_U, Un, V, Vn, W, Wn, RSV (for Resolver) and FAILn for unit PWM0 and which are PWM1 A_U, Un, V, Vn, W, Wn, RSV and FAILn for unit PWM1, pinning option A and PWM1B_U, Un, V, Vn, W, Wn, RSV and FAILn for pinning option B, can be found in the corresponding pin table in Chapter 7.6 (Pin Table sorted by signals) of the netX100/500 Technical Data Reference Guide.

Note: The following popular pitfall with netX PWM designs must be avoided:

Both PWM units are equipped with an active low Failure Signal input that immediately stops the corresponding unit and sets all outputs low when active.

These signals are PWM0E_FAILn (pin P17, shared w. PIO06) for PWM unit 0 and PWM1 A_FAILn (pin N18, shared with XM0_IO1) for PWM unit 1, pinning option A or PWM1B_FAILn (pin T20, shared with XM2_ECLK) for pinning option B.

Since all these pins are equipped with internal pull-down resistors, the failure signals are always active when unconnected. Hence on any netX PWM design that does not make use of a failure signal, the appropriate signal must be pulled high by an external pull-up resistor, otherwise the corresponding PWM unit will not operate!

Please also note, that the FAILn signals are not 5 V tolerant, hence 3.3 V levels may only be used here!

4.14 Encoder Interface

The netX100/500 provides an Encoder unit allowing to connect two separate incremental rotary encoders (quadrature encoders), used for capturing position (angle), speed and direction of a rotating axle (mostly used with motion control applications).

As it can be seen on the picture in the previous chapter, using the Encoder unit rules out the parallel use of PWM unit 0 and also blocks PIO0-2 and/or PIO3-5 and PIO6-7 , which are the standard resources for RTE and field bus status LEDs. In that case, other I/O resources must be used for the LEDs.

However, since the pinning option for the Encoder unit allows enabling encoder 0 and encoder 1 and the two optional MP signals individually, the parallel use of a single encoder and up to 5 PIO signals is possible.

Using the encoder also requires XMAC2 to be disabled (allowing the encoder unit to be controlled by the ARM CPU), which excludes the use of a field bus protocol on XMAC2 (or which would require customized XMAC code, that can only be created by Hilscher (custom development)).

Each encoder interface provides three signal inputs named A, B and N. A and B are the two quadrature signals, while N is the index signal, indicating position 0 of the rotary encoder. There are two additional optional input signals named MP0 and MP1 that can be used for initiating the storage of the current position or system time to up to four different capture registers.

As all encoder signal inputs expect single ended digital signals with 3.3 V level, only appropriate encoders may be connected directly to the inputs. When using encoders with differential outputs and/or 5 V signaling levels, appropriate line receivers and/or level shifters must be connected between encoder and netX. The following schematics show an example (only one encoder and only one signal shown in detail) of a possible circuit:

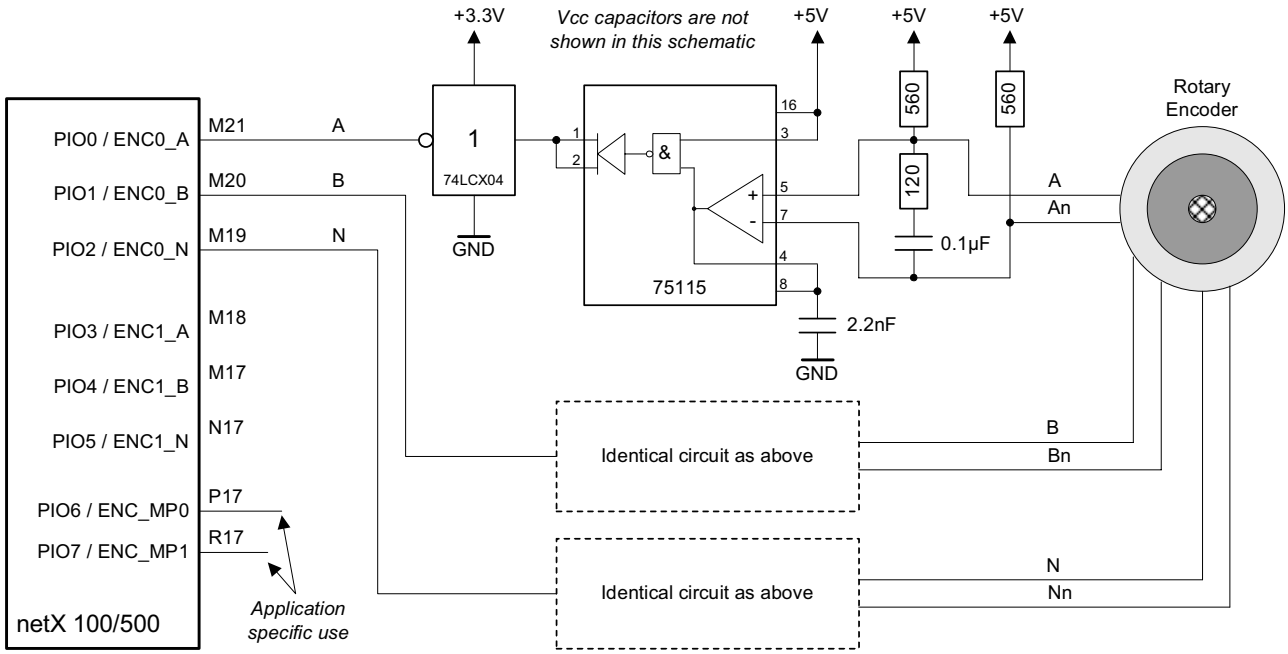


Figure 93: Encoder Circuit Example

Note: Designers should note, that the shown schematics are just an example and that the best circuit for their application may be different. Unless designers have appropriate experience with encoder circuitry, we strongly recommend to contact the manufacturer of their encoders to find the best solution for their design.

4.15 LCD Interface

The netX500 is equipped with an LCD controller, allowing to connect color/bw STN or DSTN as well as TFT displays. Connecting a display does usually not require any additional components and is accomplished by simply hooking up the display signals to the appropriate LCD signals of the netX500.

Depending on the display type (and the corresponding register settings of the LCD controller), the LCD data lines LCD_D[17:0] have different mappings. Please consult the netX100/500 Technical Data Reference Guide (chapter 2.24) for a table listing all possible display modes and the corresponding data signal mapping.

Chapter 7.4 (“Signal Definitions”) lists the 5 control signals of the LCD controller and their functionality in STN and TFT display mode.

Besides the designated LCD display signals another signal may be required for controlling the backlight of the connected display.

For this purpose, GPIO14 has been assigned an appropriate standard function.

It can be simply set or cleared in order to turn on and off the display backlight, or, if supported by the display, switched to PWM mode to control the intensity of the backlight. If the display can’t handle the PWM signal directly but requires a constant voltage for intensity control, an RC filter as shown below must be implemented.

Note: Please note, that this functionality has only been realized with Windows CE images for netX500 boards. Further, the GPIOs are in no way linked to or controlled by the LCD controller, hence the backlight control functionality must always be realized completely by software!

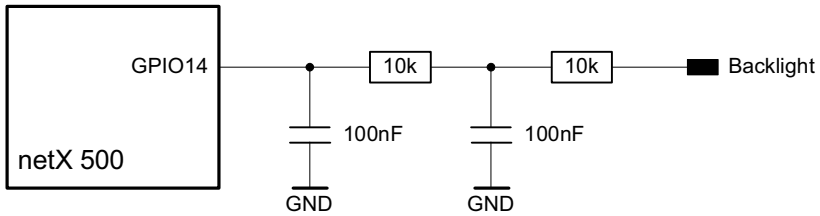


Figure 94: LCD-Backlight-PWM; RC-Filter; Windows CE

4.16 Touch Panel Interface

Though the netX500 is not equipped with a touch panel controller, such a functionality can easily be implemented by using two I/O signals and two channels of the internal AD converter, along with a few external components, as shown in the following schematic:

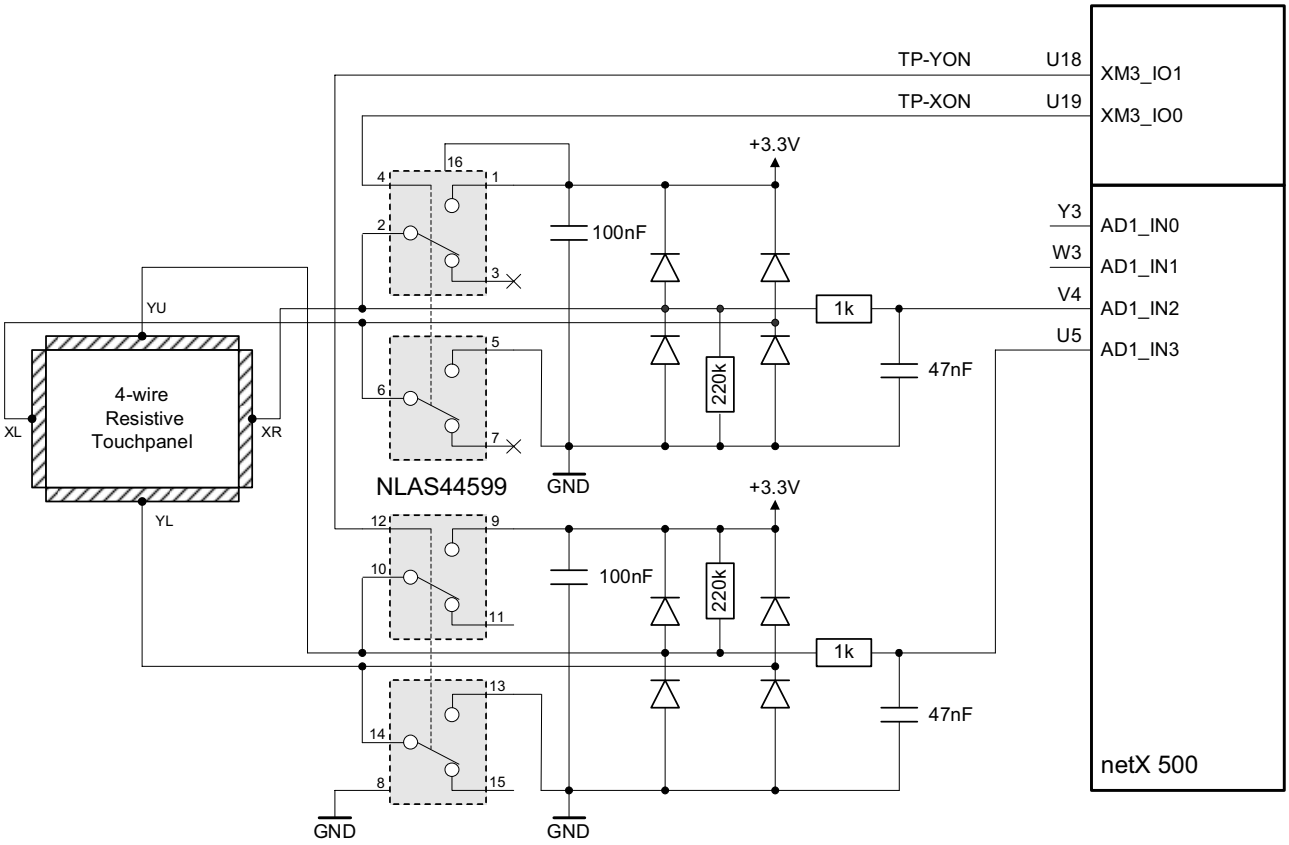


Figure 95: Touch Panel Circuit

The NLAS44599 (ON Semiconductor) is an integrated component containing 4 analog switches, perfectly meeting the requirements of this application. However, other analog switches or four single FETs may be used as well. The purpose of the diodes (e.g. BAR43S) is to protect the netX analog inputs from possible damage caused by ESD (as it lies in the nature of touch panels, that they are being touched by human hands, they are always subject to electrostatic discharge). The RC-filter comprising of the 1k resistor and the 47nF capacitor is just an example and may need to be adapted to the requirements of your design.

The resources (AD converter channels and I/Os) shown above, must also be chosen, when users want to run Windows CE images provided by Hilscher, already containing an appropriate touch panel driver. When other operating systems (e.g. rcX) are used, users need to program their own driver and may then want to select other I/O signals than the XM3 I/Os (as this would rule out the use of RTE protocols), while the ADC channels may then also be freely selected.

Please note, that the above schematics do not show the (always necessary) power supply of the AD converter. See chapter 4.12 (A/D Converter) for an example how to connect the ADC power supply pins. The filtered 3.3 V voltage from the ADC power supply circuit should also be used for the +3.3 V shown in the schematics above.

To make the circuit and the netX work as a touch panel controller, appropriate driver software must perform the following tasks:

Starting in idle mode, the two I/O signals (XM3_IO0 and IO1) must be held low and the analog value on the two ADC inputs must be captured periodically. In this state, all switches are turned off and due to the 200k pull-up and pull-down resistors, the value on the YU input will be around maximum, while the value on the XR input will be around 0, as long as the panel is not being touched. In this case the application is to remain in idle mode. If the XR input value is significantly higher than 0 and the YU input value significantly lower than max., it can be assumed, that the panel is currently being touched.

Now the TP-XON must be set to high level, which turns on the two upper switches, connecting XR to 3.3 V and XL to GND. At the position being touched, the conducting XL-XR and YL-YU layers of the panel contact each other, building a voltage divider with the voltage at the touch point being proportional to the X position of the touched point. After a (panel specific) settling time, this voltage (and hence the corresponding position) can be measured at the YU input.

When the X position has been captured, the TP-XON I/O (XM3_IO0) must be set to low level again, while the TP-YON I/O (XM3_IO1) must set to high level. This turns on the two lower switches, connecting YU to 3.3 V and YL to GND.

Now (again after a certain settling time) the voltage level at the XR input resembles the Y position of the current touch point. When the Y position has been captured, TP-YON I/O (XM3_IO1) must be set to low level again. Now the current position is known and the system can return to idle mode.

Due to non-linearity and wear/aging, it is recommended, to also implement a calibration function that shows a few calibration points at known positions on the display, while the user consecutively touches these points. By comparing the measured coordinates to the coordinates of the calibration points, appropriate correction factors can then be calculated, allowing to compensate the deviations.

The required settling time between changing the switch signals and performing the acquisition of the analog touch panel signals must be evaluated. In electrically noisy environments it may also be necessary to perform several measurements and use the average, trading response time for accuracy.

4.17 PIO

The netX chip contains several programmable input/output lines. Each of the 83 PIO can be used as simple input or output without any additional features. The first 31 PIO pins (PIO0-PIO30) are shared with Motion Control pins, LCD pins and ETM pins. The 53 other Pins (PIO32-PIO84) are shared with Host Interface Pins. (PIO31 does not exist). PIOs 0 – 7 usually drive Fieldbus and RT Ethernet status LEDs in standard applications.

PIO 0 – 30

The PIO 0 – 30 pins are equipped with 50 kΩ pull up resistors with a minimum resistor value of 20.6 kΩ and maximum resistor value of 116.4 kΩ. The maximum input/output current is 6mA.

PIO 32 – 84

The PIO 32 - 84 pins are not equipped with pull-up or pull-down resistors. The maximum input/output current is 18 mA.

4.18 Power Supply

netX100 and netX500 both require an I/O voltage supply of 3.3 V and a core voltage supply of 1.5 V.

For worst case scenarios, the 3.3 V supply should be able to source a current of 400mA for the netX part (netX, memory, etc.) of the circuit. When the core voltage regulator is also supplied by the 3.3 V rail, the max. current of course increases accordingly.

The core supply should be able to deliver 1 A.

For certain applications (e.g. applications that do not use the Ethernet ports) lower max. currents may be sufficient, however in that case, the corresponding Technical Data Reference Manual of the netX chip should be considered for details about power consumption.

It is recommended to implement a separate core voltage supply for the netX, even if a 1.5V supply is required by other parts of the system (e.g. FPGA). The output voltage of this separate supply should be “programmable” (either by appr. digital input pins or by external resistors. This allows to continue to use the same design when the used netX type has gone through a die shrink process, which always results in a reduced core voltage.

4.18.1 Core Voltage Regulator

A Hilscher standard circuit for the core voltage regulator uses a FAN2001 (Fairchild Semiconductor) step-down DC-DC converter and is shown below:

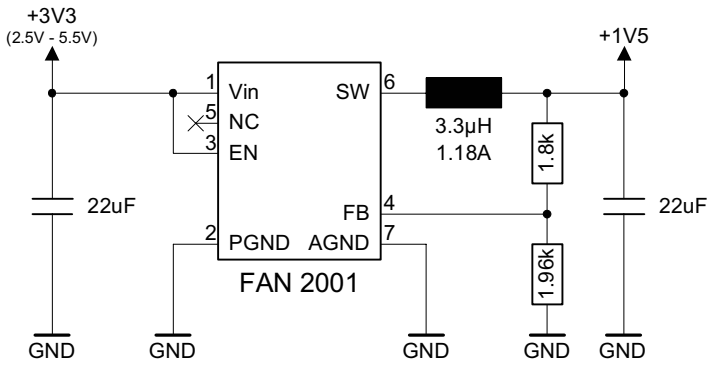


Figure 96: netX Core Voltage Regulator

The complete part name is FAN2001MPX, the inductor is a CR32NP-3R3 (Sumida), and the capacitors are ceramic types (X5R), resistors 1%.

4.18.2 Alternative Core Voltage Regulator

An alternative part, especially for designs with space constraints, is the EN5312Q (Enpirion), which delivers 1A, has an integrated inductor and provides 3 digital inputs for setting the output voltage, requiring only two 10 µF ceramic caps as additional external components (see www.enpirion.com for detailed information).

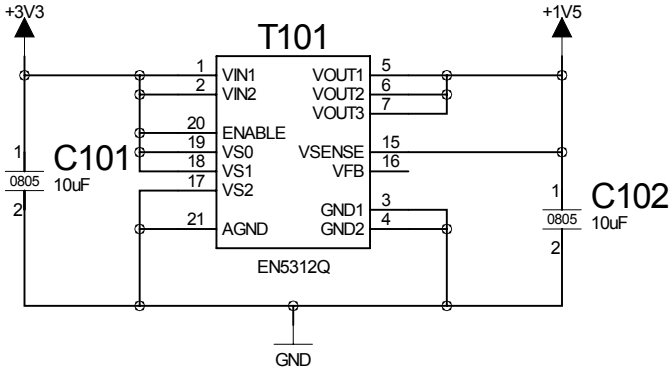


Figure 97: Alternative Core Voltage Regulator

4.18.3 Common Supply Voltage Regulator 3.3 V

A standard circuit for a common supply voltage of 24V, which operates from 9V to 30V and delivers 3A at 3.3V, is shown in the following schematic:

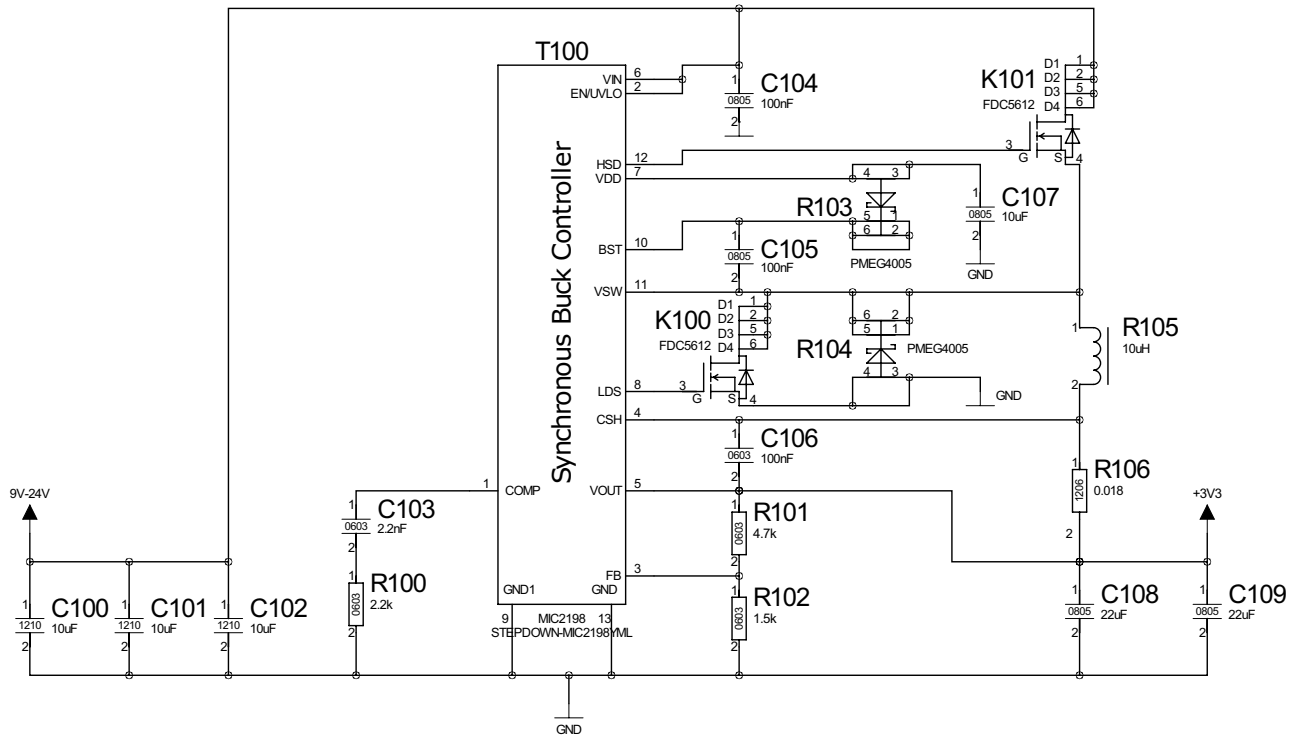


Figure 98: netX +3V3 Supply

This circuit provides high efficiency over the complete input voltage range, due to the synchronous switching, consumes appr. 320 mm2 board space and costs appr. 3.50 €.

4.18.4 Voltage Regulator 5 V

The same circuit with 6,8kΩ for R101 and 1,3kΩ for R102 delivers 5V.

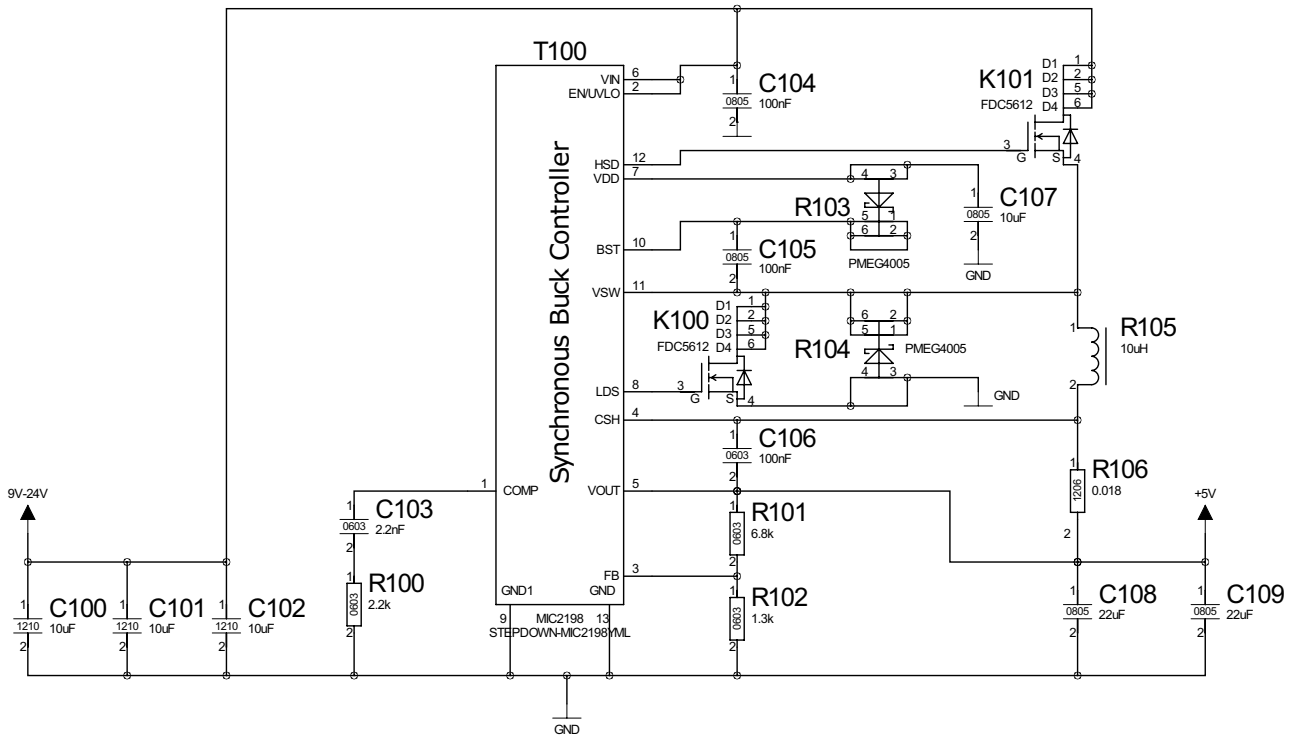


Figure 99: netX +5V Supply

5 General Design Considerations

5.1 Thermal Behavior

5.1.1 Basics

Since netX100/500 designs are often used in industrial environments, fitness for high temperature ranges is frequently an issue. Depending on the used interfaces and chip type, netX100/500 designs can have a power dissipation from typically 1.1 W (PHYs for Ethernet turned off) up to 1.6 W (integrated PHYs in use), which results in appropriate warming of the netX silicon. While there is a hard limit for the chip's junction temperature at 125 °C above which malfunction and permanent damage may occur, it is always desired to keep the junction temperature as low as possible, as a semiconductor's statistical life time generally decreases with rising temperature. Since additional power dissipation that is being avoided in the first place does not result in additional heat that needs to be dissipated, hardware designers should make sure, that all netX power supply circuits deliver nominal voltage levels (3.3 V I/O and 1.5 V core) and do not make use of the possible headroom (3.6 V I/O and 1.65 V core).

The BGA packages used with all current netX chips have mainly two paths of heat dissipation, which are the path through the package balls into the copper of the PCB (mainly power and ground planes) and the path from the chip surface (top) to the environment. The resulting thermal resistance of the first path strongly depends on the characteristics of the PCB, which also makes the thermal behavior of a design strongly dependant on the PCB. The only possibility to decrease the influence of the PCB is to use a heat sink, which can considerably reduce the thermal resistance of the second heat dissipation path and also improves the overall thermal behavior of the design.

5.1.2 Estimates

The Technical Data Reference Guides (chapter "Thermal package specification") of the netX chips provide the following formula that allows calculation of the chip junction temperature (T_j) at a given environment temperature (T_a), power (P_{netX}) and thermal resistance (R_{th}) of the heat sink:

$$T_j = T_a + (\theta_{jc} + R_{th}) \times P_{netX}$$

The chip specific value of θ_{ic} can be found in the above mentioned chapter of the appropriate Technical Data Reference Guide.

Please note, that the formula above only allows an estimate for the possible junction temperature of a particular design, as there is still an influence of the PCB characteristics. The parameters have been evaluated using certain test boards and may hence not be directly applied to a specific design. This applies even more to the second formula, for operation without heat sink!

Hilscher netX hardware is usually designed for a maximum junction temperature of appr. 100°C, which is also the recommended value for customer designs. The FIT-rate (FIT = Failure in Time, see appr. chapter of the Technical Data Reference Guide) for the silicon process which netX10/50/100/500 are based on, shows a significant rise in the temperature range between 100°C and the absolute maximum junction temperature, which was the reason for choosing the 100°C. However, since the absolute max. junction temperature is 125°C, it is at a device manufacturer's discretion if he wants to follow this recommendation or rather decides to accept a higher junction temperature and trade a decrease of the MTBF of his devices for a higher temperature range specification.

A major factor of the thermal behavior of a netX design is the size of the PCB. Design experience at Hilscher shows, that designs with a power density of more than 0.15W/cm2 are critical (assuming the 100°C limit for the junction temperature and a maximum ambient temperature of 70°C and considering the common maximum temperatures of peripheral components like SDRAM, FLASH, Reset Generator, etc.), hence the board size should be chosen accordingly. To calculate the power density of a design, simply divide the power dissipation by the area of the PCB.

Following there are two examples from the Hilscher netX product line, one that is well within this power density limit and one that is at this limit:

Mini PCI Card with Ethernet

Dimensions:	44.6 mm x 59.8 mm
Area:	26.67 cm²
Power consumption:	1.75 W, heat sink, 70°C
Power Density:	0.07 W / cm²

Figure 100: CIFX 90-RE

netIC with Ethernet

Dimensions:	21.0 mm x 42.0 mm
Area:	8.82 cm²
Power consumption:	1.3 W, heat sink, 70°C
Power Density	0.15 W / cm²

Figure 101: NIC 50-RE

5.1.3 Rules of thumb

Assuming the recommended 100 °C junction temperature limit, the following rules of thumb can be provided, which are based on Hilscher's design experience with netX chips:

- Designs with 45 x 60 mm area and heat sink usually work up to 70 °C.
- Designs with 45 x 60 mm area without heat sink usually work up to 55 °C.
- netX500 designs with 30 x 50 mm and heat sink usually work up to 70 °C.
- When using the internal PHYs, the netX temperature rises by appr. 15 °C.
- When using the heat sinks, recommended by Hilscher, the maximum temperature of the netX case decreases by approximately. 15 °C.
- The above rules assume unimpeded convection of the PCB. When a small closed cabinet is used, the maximum ambient temperature (inside the cabinet) must be decreased by approximately 15 °C.
- When the power density is higher than 0.15 W/cm², it will be critical to make a netX design which operates up to 70 °C.
- Avoid placing semiconductor components on the bottom side of the PCB within the netX chip area. Resistors or ceramic capacitors may be placed under the netX if they allow operating temperatures up to 100 °C (X7R ceramic).

Temperature tests with a netX500 Evaluation board have shown, that it is possible to build designs for extended temperature range (up to +85 °C) with heat sink, resulting in a netX case temperature of ~103 °C (see document "nxdkn-en_Rev3_Waerme.pdf", available on the Hilscher website).

5.2 EMC behavior

Designing for EMC is a quite complex issue, already filling countless pages of again countless books and papers and it is not the intention of this document, to further enrich this variety of publications, however a few basic guidelines / hints can be presented, that may be useful for the PCB designer routing a netX design.

5.2.1 Layer Stack

Though, depending on the complexity of the design, netX designs using 4-layer PCBs are possible, the use of a 6 Layer board (4 signal layers, one power and one ground plane layer) is recommended. 4 signal layers provide enough space to keep the power and ground planes free from any signal traces and allow shielding areas on top and bottom layers, which contributes to a satisfying EMC behavior of the design. An approved 6-layer stack for netX designs is shown in the following figure:

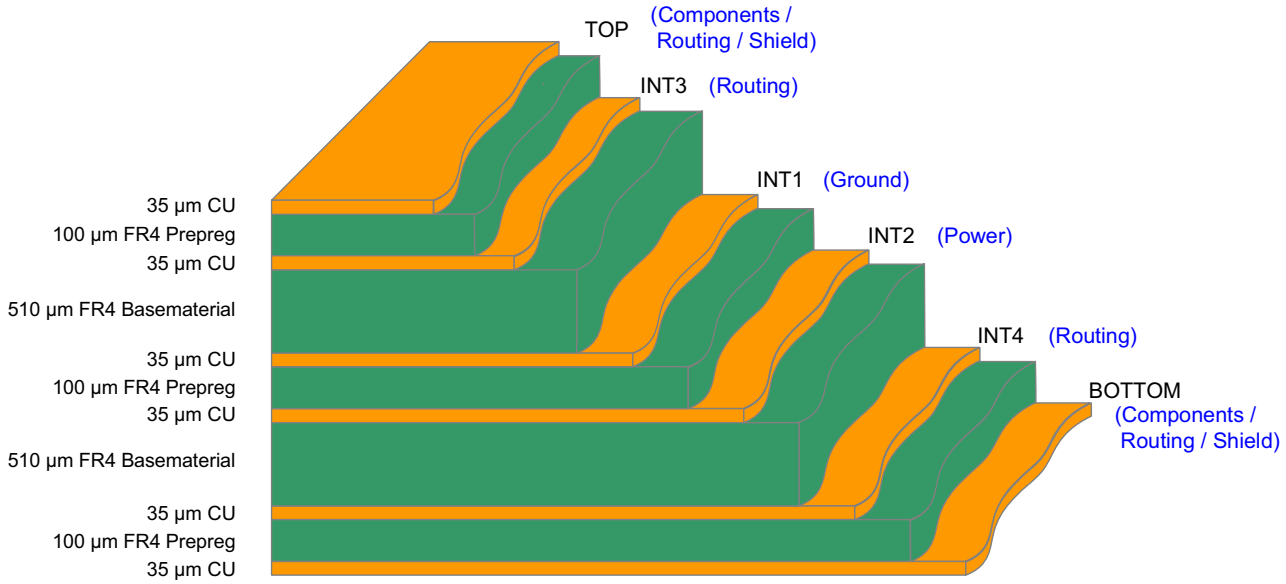


Figure 102: Approved netX PCB Layer Stack

5.2.2 Decoupling capacitors

As with any digital design, the use of a sufficient number of decoupling capacitors is important to provide a stable operation of the design and avoid unnecessary emission. The following picture shows an example how to arrange decoupling capacitors around a netX100/500.

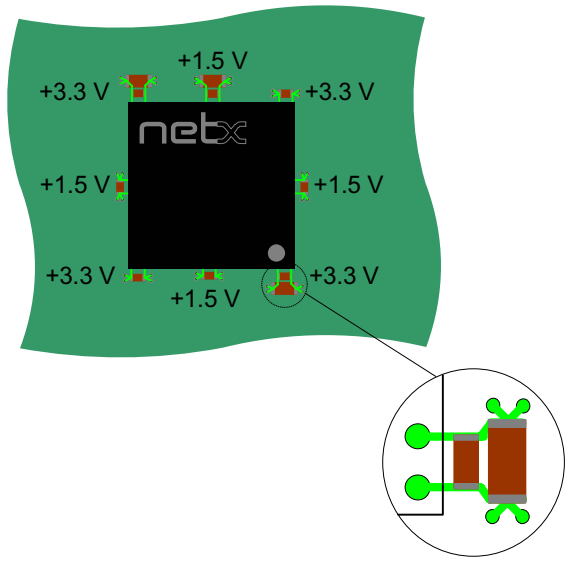


Figure 103: netX100/500 Decoupling Caps

As the picture already indicates, the power path goes from vias (located as close as possible to the caps) to the caps and from there directly to the netX power pins. However only power pins on the two outer BGA rings should be connected that way. All inner power pins should connect to the plane directly.

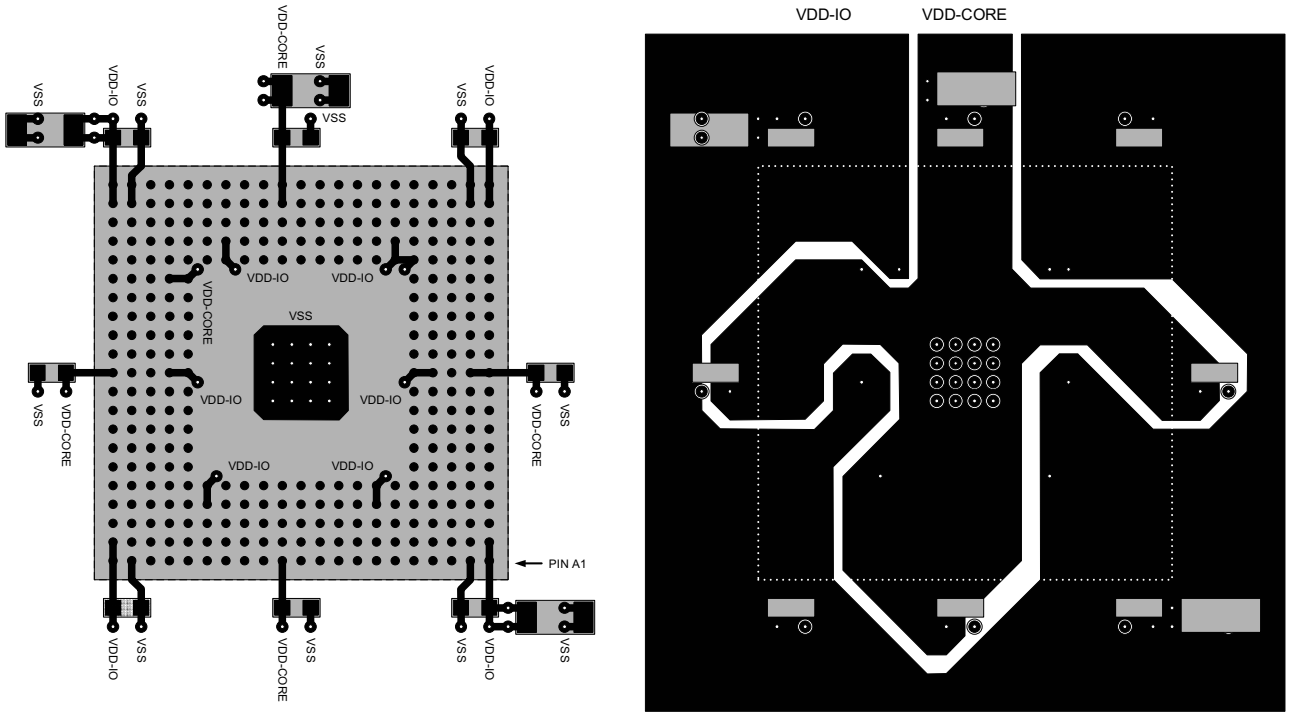
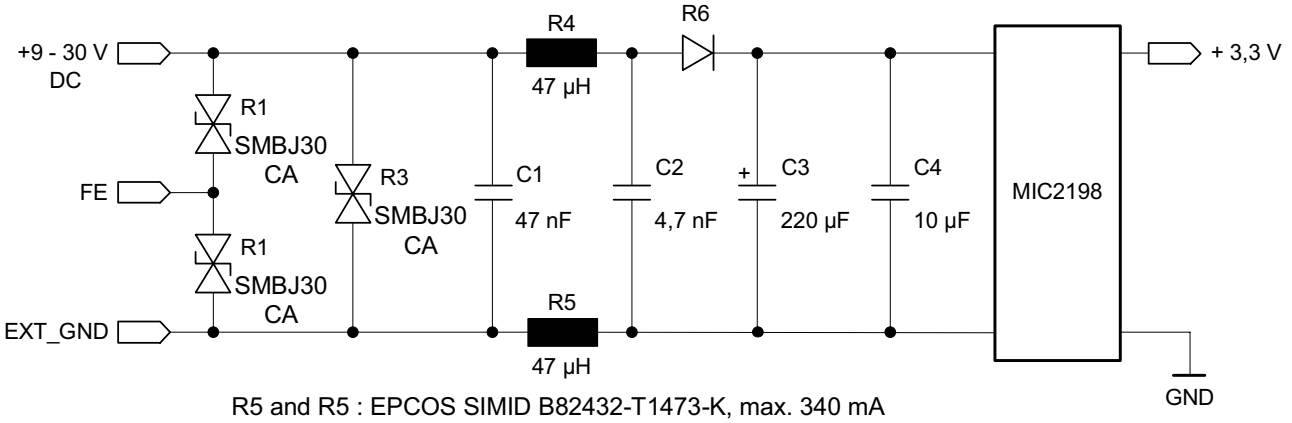


Figure 104: netX100/500 Decoupling Caps Vias and Inner Plane for VDD-IO and VDD-CORE

5.2.3 Power Supply Input Filter

When using the 9-30V power supply shown in chapter 3.17, the following filter circuit is recommended, to reduce line based emissions at 500 kHz and harmonics:



R5 and R5 : EPCOS SIMID B82432-T1473-K, max. 340 mA

Figure 105: Power Supply Filter

5.2.4 Reset Lines

As already mentioned in chapter 3.4, reset signal lines should be kept as short as possible and should be equipped with a 1nF ceramic capacitor (connected to the reset signal and ground), located close to the netX reset input pin, to reduce the risk of undesired resets due to noise or electrostatic discharge.

5.2.5 Clock Circuits

Any oscillator pins on the netX chips are located on outer BGA ball rings, allowing to keep the traces to a quartz crystal as short as possible.

The following picture shows a recommendation for placing and routing the oscillator components:

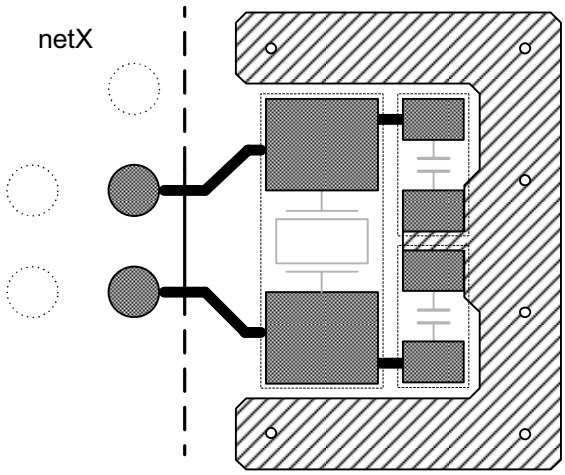


Figure 106: Oscillator Circuit with Ground Shield

5.2.6 Ethernet Interface

When routing the two signal pairs TXP/TXN and RXP/RXN of each netX Ethernet channel, some special requirements need to be considered:

- Each signal pair must be routed as a separate pair of traces which should be kept as short as possible (place magnetics and termination components as close to the netX as possible).
- Traces of a pair must be routed adjacent to each other, with constant spacing and equal length.
- The distance between signal pairs should be at least 5 times the spacing of the pair traces.
- Traces must be impedance controlled, maintaining a differential impedance of 100 Ohm.
- Minimize layer changes. If a layer change is inevitable, change layer with both traces at equal distance from start of trace and avoid changing to layers that use a different reference plane.
- Avoid connectors in the signal traces; the traces should begin and end on the same PCB. If a connector is inevitable, use impedance controlled connectors to minimize any discontinuities in trace impedance.
- Area where Ethernet signals are routed should be free from any other signals in adjacent layers (use only layers that are separated from the Ethernet signal layer(s) by a power or ground plane, when routing other signals in the Ethernet area).
- Use the schematic from chapter 3.10.1 along with the recommended components.

The following pictures show two examples of setups that keep the differential impedance of the signal pair around 100 Ohm:

a) Edge-Coupled Surface Micro strip

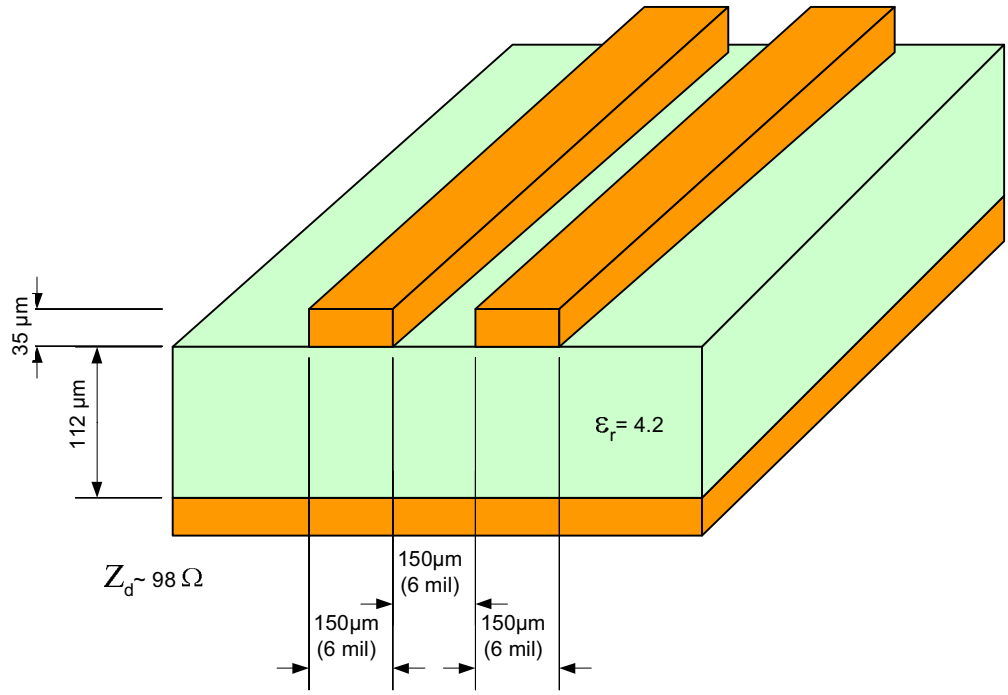


Figure 107: Edge-Coupled Sourface Micro Strip

To improve shielding, the Ethernet traces can be routed on an inner layer, using part of the top layer as shield:

b) Edge-Coupled Offset Strip line

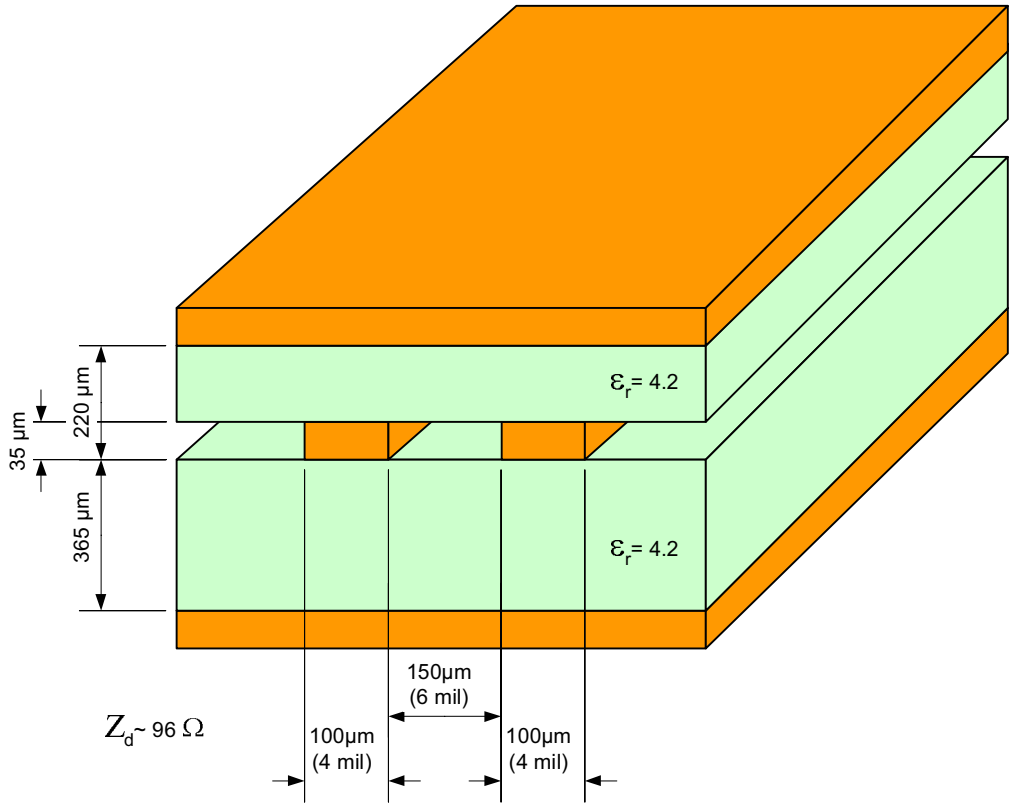


Figure 108:Edge-Coupled Offset Strip Line

5.2.7 Memory Bus

When connecting SDRAM and/or parallel FLASH/SRAM etc., route the connection in a bus structure (no tree) with the bus starting at the netX, as shown in the following picture. The bus should be as short as possible and the length of the SDRAM clock signal trace should match the length of the longest SDRAM signal trace.

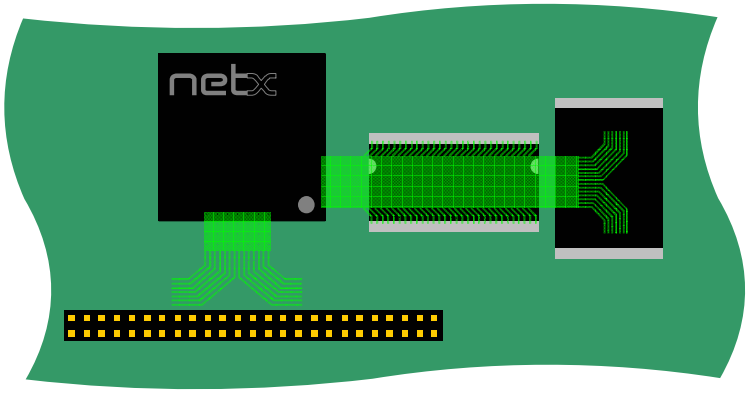


Figure 109: netX Memory Bus

5.2.8 Planes

When routing signal lines, make sure, they run over an appropriate return path (power or ground plane), that is contiguous, not more than 2 layers away and not interrupted by large gaps, as this always increases emission. When splitting planes can not be avoided, keep traces well within the plane area and do not route at the edge or even outside of the plane, as shown below:

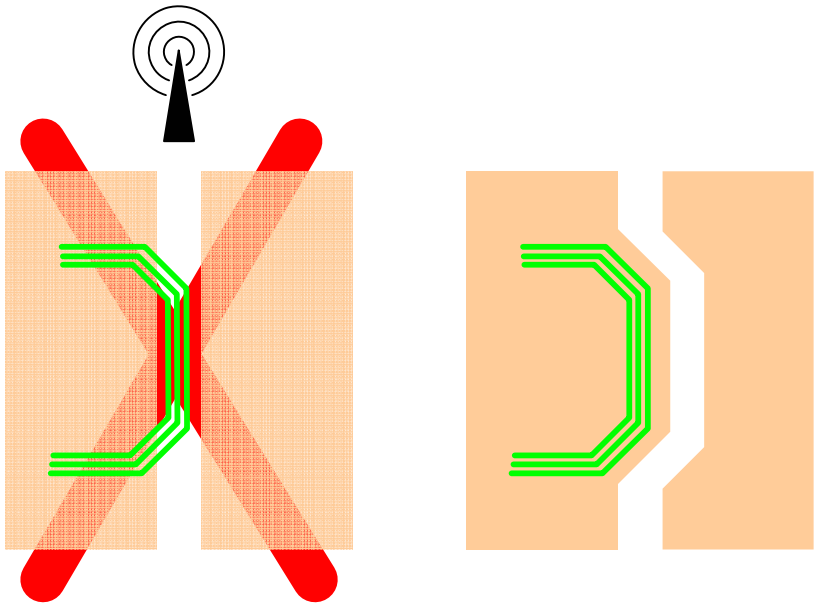


Figure 110: Routibg Example

5.3 Vias and Traces under the netX100/500

Using a PBGA package, the netX requires small traces and vias on the PCB. However, as a result of the proper chip pinout design, four layers and 0.15 mm traces respectively 0.2 mm through whole vias are sufficient for most applications which can hence be realized by inexpensive standard printed circuit board technology.

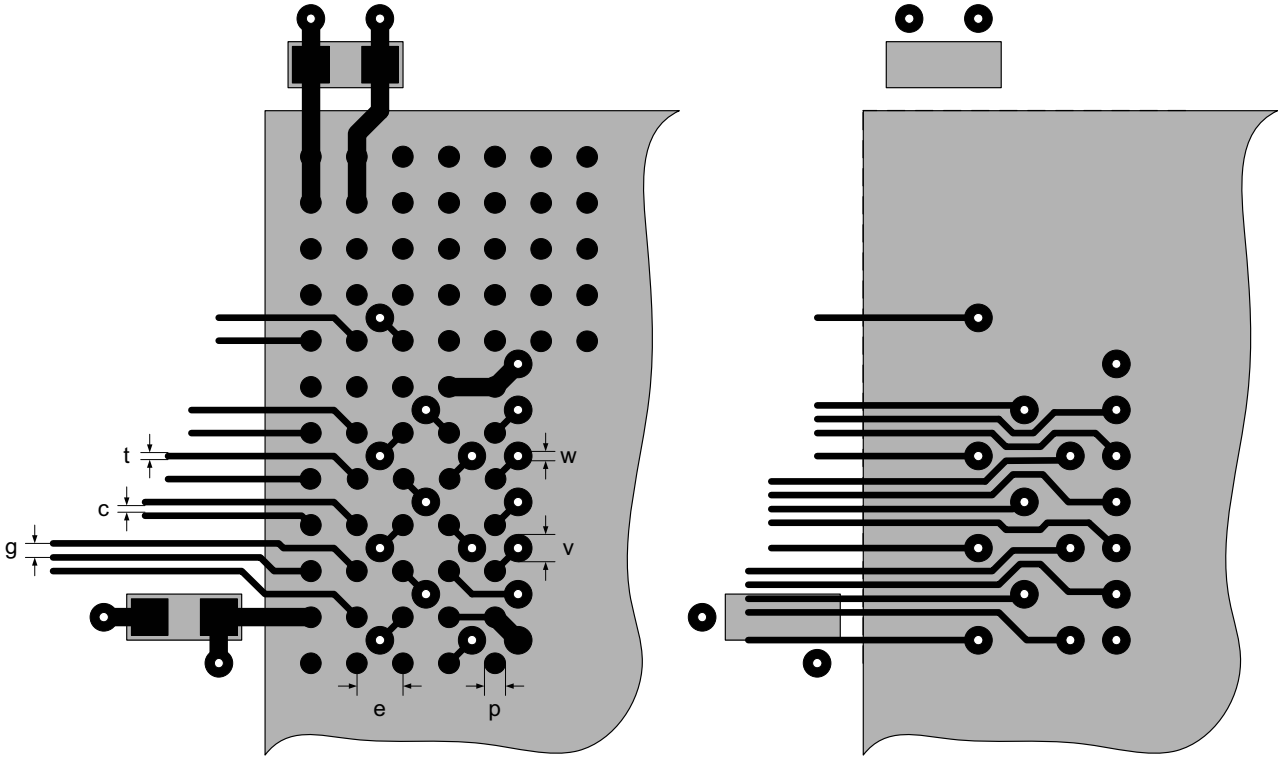


Figure 111: Vias and Traces under netX100/500 Top and Bottom Side

Dimension	Description	mm	mil
c	Clearance	0,15	6
e	Pitch	1,00	39.37
g	Grid	0.15	6
p	Pad	0,45	18
t	Trace Width	0.15	6
v	Via Diameter	0,60	24
w	Drill Hole	0,20	8
	PCB max. width	2,00	79

Table 41: Dimension of Printed Circuit Board Design

Note: Vias within the chip footprint area should be exactly centered between the pins to avoid possible soldering problems during manufacturing!

6 Reference Section

The following chapters list some key components for netX100/500 designs that have been successfully evaluated by Hilscher and, where applicable, are supported by Hilscher tools (status of January 2012).

6.1 Crystals

Use	Part Number	Manufacturer
System clock	CS10-25000MAGJ-UT	Citizen
Real Time Clock	Q0.032768-JTX520-12.5-20T1-LF	Jauch Quartz GmbH

Table 42: Crystal Reference

6.2 Memory Components

SPI FLASH

Manufacturer	Size	Type	Bootwizard	2nd Stage	rcX
ATMEL	2MB	AT25DF161	✓	-	-
	128kB	AT25F1024A	✓	-	-
	64kB	AT25F512	-	-	✓
	64kB	AT25F512A	-	-	✓
	2MB	AT26DF161	✓	-	-
	2MB	AT26DF161A	✓	-	-
	4MB	AT26DF321	✓	-	-
	128kB	AT45DB011B	✓	✓	✓
	256kB	AT45DB021B	✓	✓	✓
	512kB	AT45DB041B	✓	✓	✓
	1MB	AT45DB081B	✓	✓	✓
	1MB	AT45DB081D	✓	✓	✓
	2MB	AT45DB161B	✓	✓	✓
	2MB	AT45DB161D	✓	✓	✓
	4MB	AT45DB321B	✓	✓	✓
	4MB	AT45DB321C	✓	✓	✓
	4MB	AT45DB321D	✓	✓	✓
	8MB	AT45DB642D	✓	✓	✓[1)]
EON	4MB	EN25P32	✓	-	-
Macronix	2MB	MX25L1605D	✓	-	-
	4MB	MX25L3205D	✓	-	-
	8MB	MX25L6405D	✓	-	-
PMC	128kB	PM25LV010	✓	-	-
	64kB	PM25LV512	✓	-	-
Saifun	512kB	M25P40	✓	-	-
	64kB	SA25F005	✓	✓	-
	128kB	SA25F010	✓	✓	-
	512kB	SA25F020	✓	✓	-
ST	128kB	M25P10VP	✓	-	-

Manufacturer	Size	Type	Bootwizard	2nd Stage	rcX
	4MB	M25P32	✓	✓	-
	8MB	M25P64	✓	-	-
	1MB	M25PE80	✓	✓	✓
ST	2MB	M45PE16	✓	✓	✓
	256kB	M45PE20	✓	-	-
	512kB	M45PE40	✓	✓	✓
	1MB	M45PE80	✓	✓	✓
SST	256kB	SST25LF20A	✓	-	-
	512kB	SST25LF40A	✓	-	-
	1MB	SST25LF80A	✓	-	-
	128kB	SST25VF010	✓	-	-
	128kB	SST25VF010A	✓	-	-
	256kB	SST25VF020	✓	-	-
	512kB	SST25VF040	✓	-	-
	64kB	SST25VF512	✓	-	-
	64kB	SST25VF512A	✓	-	-
Spansion		CFI ParFlashes	✓	✓	-
	2MB	S25FL016A	✓	-	-
	4MB	S25FL032A	✓	-	-
Strata		CFI ParFlashes	✓	✓	-
Winbond	4MB	W25P32	-	-	✓
	2MB	W25Q16	✓	✓	-
	4MB	W25Q32	✓	✓	✓
	1MB	W25Q80	✓	✓	-
	4MB	W25X32	✓	-	-
Nymonix	128kB	NX25P10	✓	-	-
	256kB	NX25P20	✓	-	-
	512kb	NX25P40	✓	-	-

Table 43: Memory Component Reference SPI FLASH

1) Is not detected using the autodetect function. Parameters have to be entered in config file.

Parallel Flash

Manufacturer	Size	Part Number
Spansion	16MB	S29GL128P90TFIR1
	32MB	GL256N10FFI01

Table 44: Memory Component Reference Parallel FLASH

SDRAM

Manufacturer	Size	Part Number
ISSI	8MB	IS42S32200C1 (32Bit)
	32MB	IS42S32800B (32Bit)
	64MB	IS42S32160B (32Bit)
Micron	8MB	MT48LC2M32B2 (32Bit)
	16MB	MT48LC4M32B2 (32Bit)
	32MB	MT48LC8M32B2 (32Bit)
	64MB	MT48LC16M32 (16Bit)

Table 45: Memory Component Reference SDRAM

Appendix

6.3 List of Tables

Table 1: BOM Basic Circuit Page 1 23
Table 2: BOM Basic Circuit Page 2 23
Table 3: BOM Basic Circuit Page 3 24
Table 4: BOM Basic Circuit Page 4 24
Table 5: BOM Ethernet TP Two Channel 26
Table 6: BOM Ethernet TP Single Channel 28
Table 7: BOM Fiber Optic Page 1 31
Table 8: BOM Fiber Optic Page 2 32
Table 9: BOM AS-Interface Circuit 34
Table 10: BOM CANopen Circuit 36
Table 11: BOM CC-Link Circuit 38
Table 12: BOM CompoNet Circuit 40
Table 13: Optional BOM CompoNet Circuit 40
Table 14: BOM DeviceNet Circuit 42
Table 15: BOM PROFIBUS Circuit 44
Table 16: BOM MMC/SD Card Circuit 46
Table 17: BOM microSD Circuit 46
Table 18: BOM UART ⇔ RS232 48
Table 19: BOM USB Device Circuit 50
Table 20: BOM ETM Circuit 54
Table 21: BOM LCD Interface Circuit 57
Table 22: BOM Touch Panel Circuit 57
Table 23: List of Resources – rcX OS 59
Table 24: List of Resources – 3rd Party OS 60
Table 25: FLASH Sizes for Hilscher Stacks 61
Table 26: RDY/RUN LED Status 62
Table 27: ETM Signals 71
Table 28: netX100/500 Boundary Scan MMIO/GPIO Signals 73
Table 29: MMC/SD Card Insertion Signal 75
Table 30: Function Table of 16 Bit Decode Logic 85
Table 31: netX DPM Intel™ Type Interface Circuits (2) 86
Table 32: Additional USB Signals 100
Table 33: Ethernet Circuit Component Specification 103
Table 34: Fiber Optic Ethernet Circuit Component Specification 106
Table 35: Fiber Optic Ethernet Circuit Component Specification (AC-Termination) 107
Table 36: Status LEDs for Ethernet Ports 114
Table 37: Additional RTE Sync Signals netX100/500 115
Table 38: Status LEDs for Real Time Ethernet Applications 116
Table 39: Status LEDs for Fieldbus Ports 126
Table 40: Fieldbus Status LED Colors 126
Table 41: Dimension of Printed Circuit Board Design 151
Table 42: Crystal Reference 152
Table 43: Memory Component Reference SPI FLASH 153
Table 44: Memory Component Reference Parallel FLASH 153
Table 45: Memory Component Reference SDRAM 154

6.4 List of Figures

Figure 1: netX100/500 RDY/RUN Basic Circuit 7
Figure 2: netX100/500 SPI Flash 8
Figure 3: netX100/500 Secure Memory Basic Circuit 8
Figure 4: netX100/500 System Oscillator Circuit 9
Figure 5: netX100/500 RTC Not Used in Basic Circuit 9
Figure 6: netX100/500 Reset Circuit 10
Figure 7: netX100/500 USB DOWN Stream Port (Device Mode) 10
Figure 8: netX100/500 UART0 Unused 11
Figure 9: Unused ADC Basic Circuit 11
Figure 10: netX100/500 Connection of 8MB SDRAM 12
Figure 11: netX100/500 Ethernet Not Used 13
Figure 12: Unused Host Interface 15
Figure 13: +3.3 V Power Supply Standard Circuit 16
Figure 14: 1.5 V Core Voltage Regulator 16
Figure 15: netX100/500 Power Connection 17
Figure 16: netX100/500 Basic Circuit RDY/RUN USB SEC-MEM SPI-FLASH 18
Figure 17: netX100/500 Basic Circuit SDRAM 19
Figure 18: netX100/500 Basic Circuit ADC, Host Interface, PHY, RTC 20
Figure 19: netX100/500 Basic Circuit Power Supply 21
Figure 20: netX100/500 Basic Circuit Unused netX Parts 22
Figure 21: Ethernet TP Dual Channel Circuit 25
Figure 22: Ethernet TP Single Channel Circuit 27
Figure 23: Fiber Optic Circuit on netX100/500 Side 29
Figure 24: Fiber Optic Circuit on AFBR-5978Z Side 30
Figure 25: AS-Interface Circuit 33
Figure 26: CANopen Circuit 35
Figure 27: CC-Link Circuit 37
Figure 28: CompoNet Circuit 39
Figure 29: DeviceNet Circuit 41
Figure 30: PROFIBUS Circuit 43
Figure 31: netX100/500 MMC/SD-Card SPI Circuit 45
Figure 32: UART ⇔ RS232 Circuit 47
Figure 33: USB Device Mode Circuit 49
Figure 34: USB Host Mode Circuit 51
Figure 35: BOM USB Host Mode Circuit 52
Figure 36: ETM Circuit 53
Figure 37: LCD Interface Circuit 55
Figure 38: Touch Panel Circuit 56
Figure 39: netX100/netX500 RDY/RUN Circuit 63
Figure 40: Sample Schematic, netX100/500 Secure Memory 65
Figure 41: netX100/500 System Oscillator Circuit 66
Figure 42: netX500 RTC Circuits 67
Figure 43: netX100/500 RTC Unused 68
Figure 44:netX Reset Circuits 69
Figure 45: netX JTAG Circuits 70
Figure 46: netX100/500 Boundary Scan JTAG/TEST Signals 72
Figure 47: netX100/500 SPI Flash 75
Figure 48: netx100/500 MMC/SD Card 75
Figure 49: Solution, if SD Card Disturbs SPI Data Transmission from FLASH 76
Figure 50: netX FLASH - Address Line A0 for Low/High Byte Selection 78
Figure 51: netX FLASH - A0 as the LSB of a Word Address 79
Figure 52:netX SDRAM 1 *16 Bit, 1 * 32 Bit, 2 * 16 Bit 82
Figure 53:netX SDRAM 4 * 8 Bit 83
Figure 54: netX100/500 VDDH Pins 84
Figure 55: netX DPM Intel™ Type Interface Circuits (1) 85
Figure 56: netX DPM Motorola™ Type Interface Circuits 86
Figure 57: Texas Instruments TMS320x2833x™, 16 Bit, Non Multiplexed 87
Figure 58: netX DPM Intel™ Type Circuits, Multiplexed 88
Figure 59: netX DPM Motorola™ Type Interface Circuit, Multiplexed 88
Figure 60: netX ISA Bus Interface Circuit 89
Figure 61: netX Internal Chip Select Generator Circuits 91
Figure 62: netX Extension Bus Intel™ Type Interface Circuit 8Bit, non Multiplex 93
Figure 63: netX Extension Bus Intel™ type Interface Circuit 16Bit, non Multiplex 94
Figure 64: netX DPM Motorola™ Type Interface Circuits, non Multiplex 94
Figure 65: netX DPM Intel™ Type Interface Circuit, Multiplexed 95
Figure 66: netX Extension Bus Motorola™ Interface Circuit, Multiplexed 96

Figure 67: netX100/500 UART0 97
Figure 68: netX100/500 UART0 Unused 97
Figure 69: netX100/500 USB DOWNstream Port (Device Mode) 99
Figure 70: netX100/500 USB DOWNstream Port (Device Mode), Advanced Circuit 100
Figure 71: netX100/500 USB UPstream Port (Host Mode) 101
Figure 72: netX100/500 2 Channel Ethernet Circuit (Twisted Pair) 102
Figure 73: netX Single Channel Ethernet Circuit (Twisted Pair) 104
Figure 74: netX100/500 Ethernet Circuit (Fiber Optic) 105
Figure 75: netX100/500 Ethernet Circuit (Fiber Optic) 107
Figure 76: netX Ethernet Circuit (Fiber Optic), Component Placement 109
Figure 77: netX Ethernet Circuit (Fiber Optic), Component Placement with AC-Termination 110
Figure 78: netX100/500 I^2C Interface Fiber Optic Transceivers 111
Figure 79: netX100/500 Ethernet Circuit (PHYs Not Used) 113
Figure 80: netX100/500 Ethernet Status LED Circuit 114
Figure 81: netX RTE Status LED Schematic 116
Figure 82: netX100/500 Fieldbus Interface 119
Figure 83: Basic Circuit for netX AS-Interface 120
Figure 84: Basic Circuit of netX CANopen Interface 121
Figure 85: Basic Circuit for netX CC-Link Interface 122
Figure 86: Basic Circuit for netX CompoNet Interface 123
Figure 87: Basic Circuit for netX DeviceNet Interface 124
Figure 88: Basic Circuit for netX PROFIBUS Interface 125
Figure 89: ADC Circuit 127
Figure 90: ADC Circuit, One Channel Used 128
Figure 91: ADC Circuit, ADCs Unused 129
Figure 92: PWM/Encoder/xMAC/xPAC Resource Sharing 130
Figure 93: Encoder Circuit Example 132
Figure 94: LCD-Backlight-PWM; RC-Filter; Windows CE 134
Figure 95: Touch Panel Circuit 135
Figure 96: netX Core Voltage Regulator 138
Figure 97: Alternative Core Voltage Regulator 139
Figure 98: netX +3V3 Supply 139
Figure 99: netX +5V Supply 140
Figure 102: Approved netX PCB Layer Stack 144
Figure 103: netX100/500 Decoupling Caps 145
Figure 104: netX100/500 Decoupling Caps Vias and Inner Plane for VDD-IO and VDD-CORE 145
Figure 105: Power Supply Filter 146
Figure 106: Oscillator Circuit with Ground Shield 147
Figure 107: Edge-Coupled Sourface Micro Strip 148
Figure 108:Edge-Coupled Offset Strip Line 149
Figure 109: netX Memory Bus 149
Figure 110: Routibg Example 150
Figure 111: Vias and Traces under netX100/500 Top and Bottom Side 151

7 Contacts

Headquarters

Germany
Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support
Phone: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

Subsidiaries

China
Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support
Phone: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

France
Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support
Phone: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

India
Hilscher India Pvt. Ltd.
New Delhi - 110 065
Phone: +91 11 43055431
E-Mail: info@hilscher.in

Italy
Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-Mail: info@hilscher.it

Support
Phone: +39 02 25007068
E-Mail: it.support@hilscher.com

Japan
Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support
Phone: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Korea
Hilscher Korea Inc.
Suwon, Gyeonggi, 443-734
Phone: +82 (0) 31-695-5515
E-Mail: info@hilscher.kr

Switzerland
Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support
Phone: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA
Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-Mail: info@hilscher.us

Support
Phone: +1 630-505-5301
E-Mail: us.support@hilscher.com