



x750 EtherCAT/CAN Gateway

Inhalt

1	Safety precautions.....	3
2	Overview	7
3	Hardware Description	8
3.1	Connections	8
3.2	LEDs.....	9
3.2.1	LED Assignment.....	9
4	Hardware Installation.....	11
5	Configuration	12
5.1	Application Example	12
5.2	Configuration Sequence	12
5.2.1	Step 1: Add Gateway	13
5.2.2	Step 2: Add CANOpen Devices to the Project.....	13
5.2.3	Step 3: Configure the Gateway	14
6	EtherCAT Communication.....	16
6.1	CAN Interface.....	16
6.1.1	Object Dictionary Structure.....	16
6.1.2	Object Dictionary	17
6.1.3	Standard Objects (1000 _h ...1FFF _h)	18
6.1.4	Manufacturer Specific Objects (2000 _h -5FFF _h).....	22
6.1.5	Profile Specific Objects (6000 _h -FFFF _h).....	26
7	Technical Data	37
7.1	General Technical Data.....	37
7.2	Microprocessor and Memory	37
7.3	CAN Interface.....	37
7.4	EtherCAT Interface.....	38
7.5	Operating System and License Information.....	38
8	Interfaces and Connector Assignments.....	39
8.1	24V-Power Supply Voltage.....	39
8.2	CAN.....	40
8.2.1	CAN Connector	40
8.3	EtherCAT.....	42
8.4	Conductor Connection/ Conductor Cross Sections	43
9	Order Information	44
10	License Terms.....	45
10.1	GPL V1.....	45
10.2	GPL V2.....	46
10.3	OpenBSD License.....	47
10.4	MIT Lizenz.....	47

1

Safety precautions



This manual contains important information and instructions on safe and efficient handling of the product. Carefully read this manual before commencing any work and follow the instructions.

The manual is a product component, please retain it for future use.

Trademark Notices

- CANopen® and CiA® are registered EU trademarks of CAN in Automation e.V.
- Windows is a registered trademark of Microsoft Corporation in the United States and other countries.
- EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- QNX® is a registered trademark of QNX Software Systems GmbH & Co. KG
- All other trademarks, product names, company names or company logos used in this manual are reserved by their respective owners

Document History

The changes in the document listed below affect changes in the hardware as well as changes in the description of the facts, only.

Revision	Chapter	Changes versus previous version	Date
1.0	---	Creation	2022-11-18

Technical details are subject to change without further notice.

Classification of Warning Messages and Safety Instructions

This manual contains noticeable descriptions, warning messages and safety instructions, which you must follow to avoid personal injuries and property damage.



This is the safety alert symbol.

It is used to alert you to potential personal injury hazards. Obey all safety messages and instructions that follow this symbol to avoid possible injury or death.

DANGER, WARNING, CAUTION

Depending on the hazard level the signal words DANGER, WARNING or CAUTION are used to highlight safety instructions and warning messages. These messages may also include a warning relating to property damage.



DANGER

Danger statements indicate a hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

Warning statements indicate a hazardous situation that, if not avoided, could result in death or serious injury.

CAUTION

Caution statements indicate a hazardous situation that, if not avoided, could result in minor or moderate injury.

NOTICE

Notice statements are used to notify people on hazards that could result in things other than personal injury, like property damage.

NOTICE

This NOTICE statement contains the general mandatory sign and gives information that must be heeded and complied with for a safe use.

INFORMATION

INFORMATION

Notes to point out something important or useful.

DANGER

Safety Instructions

- When working with the product follow the instructions below and read the manual carefully to protect yourself from injury and the CAN-EtherCAT from damage.
 - Do not use damaged or defective cables to connect the product.
 - In case of damages to the product, which might affect safety, appropriate and immediate measures must be taken, that exclude an endangerment of persons and domestic animals and property.
 - The product may only be driven by power supply current circuits, that are contact protected.
 - A power supply, that provides a safety extra-low voltage (SELV) according to EN 60950-1, complies with these conditions.
 - Do not open the housing of the product.
 - The product must be securely installed before commissioning.
 - Never let liquids get inside the product. Otherwise, electric shocks or short circuits may result.
 - Protect the product from dust, moisture, and steam.
 - Protect the product from shocks and vibrations.
 - The product may become warm during normal use. Always allow adequate ventilation around the product and use care when handling.
 - Do not operate the product adjacent to heat sources and do not expose it to unnecessary thermal radiation. Ensure an ambient temperature as specified in the technical data.
-

DANGER

- Hazardous Voltage - Risk of electric shock due to unintentional contact with uninsulated live parts with high voltages inside of the system into which the EtherCAT/CAN Gateway is to be integrated.
 - All current circuits which are connected to the device must be sufficiently protected against hazardous voltage (SELV according to EN 60950-1) before you start with the installation
-

Qualified Personal

This documentation is directed exclusively towards personal qualified in control and automation engineering.

The installation and commissioning of the product may only be carried out by qualified personal, which is authorized to put devices, systems, and electric circuits into operation according to the applicable national standards of safety engineering.

Conformity

The x750 Gateway EtherCAT/CAN is an industrial product and meets the demands of the EU regulations and EMC standards printed in the conformity declaration.

WARNING

In a residential, commercial, or light industrial environment the product may cause radio interference in which case the user may be required to take adequate measures.

Data Safety

This product is equipped with an EtherCAT or other interface which is suitable to establish a connection to data networks. Depending on the software used on the device, these interfaces may allow attackers to compromise normal function, get illegal access or cause damage.

Lenze does not take responsibility for any damage caused by the device if operated at any networks. It is the responsibility of the device's user to take care that necessary safety precautions for the device's network interface are in place.

Intended Use

The intended use of the product is the operation as EtherCAT/CAN Gateway.

The guarantee given by Lenze does not cover damages which result from improper use, usage not in accordance with regulations or disregard of safety instructions and warnings.

- The product is intended for indoor use.
- The operation of the product in hazardous areas, or areas exposed to potentially explosive materials is not permitted.
- The operation of the product for medical purposes is prohibited.

Service Note

The x750 Gateway EtherCAT/CAN does not contain any parts that require maintenance by the user. The x750 Gateway EtherCAT/CAN does not require any manual configuration of the hardware.

Unauthorized intervention in the device voids warranty claims.

Disposal

Products which have become defective in the long run have to be disposed in an appropriate way or must be returned to the manufacturer for proper disposal. Please, contribute to environmental protection.

Typographical Conventions

Throughout this manual the following typographical conventions are used to distinguish technical terms.

Convention	Example
File and path names	<code>/dev/null</code> or <code><stdio.h></code>
Function names	<code><i>open()</i></code>
Programming constants	<code>NULL</code>
Programming data types	<code>uint32_t</code>
Variable names	<code><i>Count</i></code>

Number Representation

All numbers in this document are base 10 (partially followed by “d”) unless designated otherwise. Hexadecimal numbers are followed by “h”. For example, 42 is represented as 2Ah in hexadecimal.

2

Overview

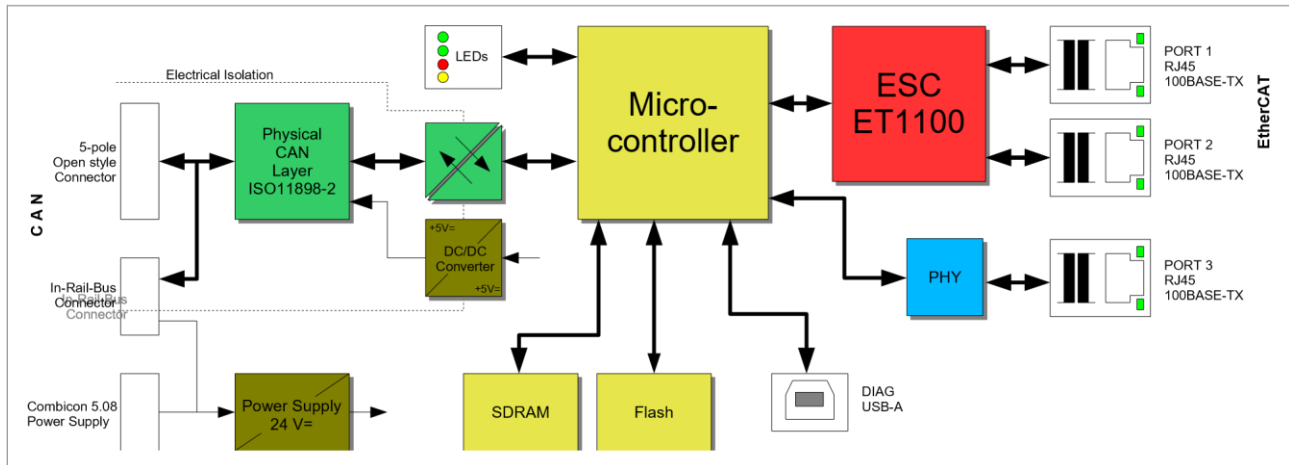


Figure 1: Block circuit diagram

The x750 Gateway EtherCAT/CAN device connects an EtherCAT® network with one CAN network. In this case the Gateway acts as an EtherCAT slave device according to „Module Profile Number 5000“ of the „Modular Device Profile Description“ (ETG.5001 documents).

The Gateway allows CAN modules with CANopen® (CiA® DS 301) or Layer 2 (ISO 11898-1) implementations to connect with a real-time EtherCAT network. The Gateway does not limit the number of CAN nodes.

The high-speed CAN interface is compliant with ISO 11898-2, and it supports transfer rates from 50 kBit/s up to 1 MBit/s. The 100BASE-TX EtherCAT interface is IEEE802.3 compatible and runs at 100 MBit/s. The CAN interface, as well as the EtherCAT interface, is electrically isolated.

The configuration of the Gateway is accomplished through the EtherCAT master.

3 Hardware Description

3.1 Connections

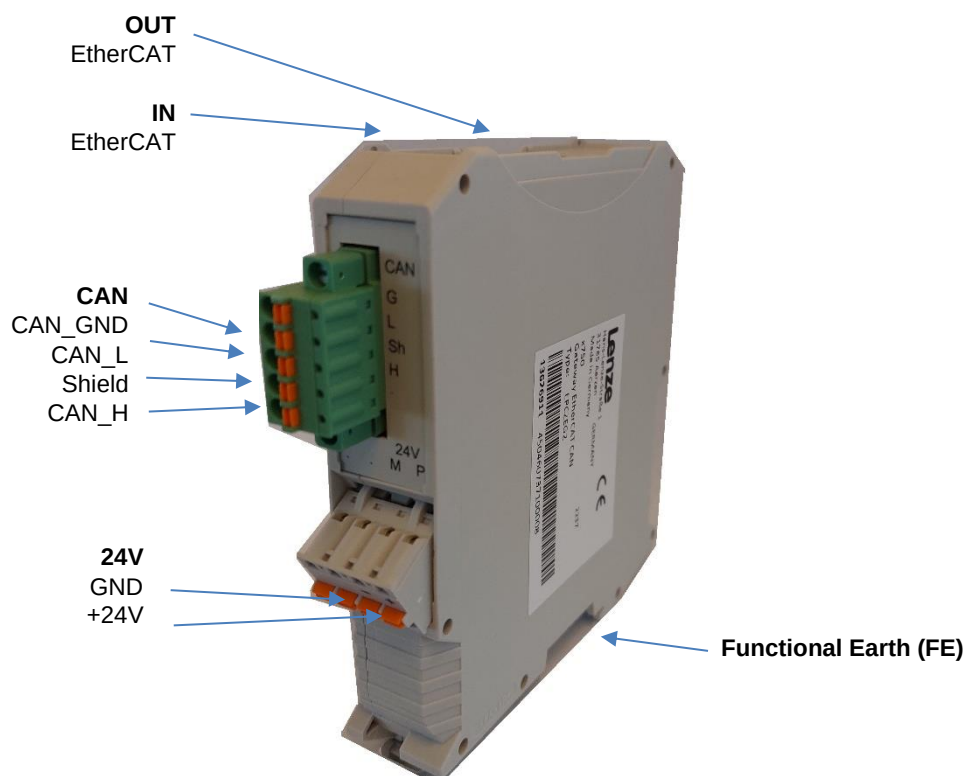


Figure 2; Connections for operating condition

NOTICE

Read chapter "Hardware Installation" on page 11, before you start with the installation of the hardware!

Please refer to page 39 ff. for signal assignment of the connectors.

3.2

LEDs

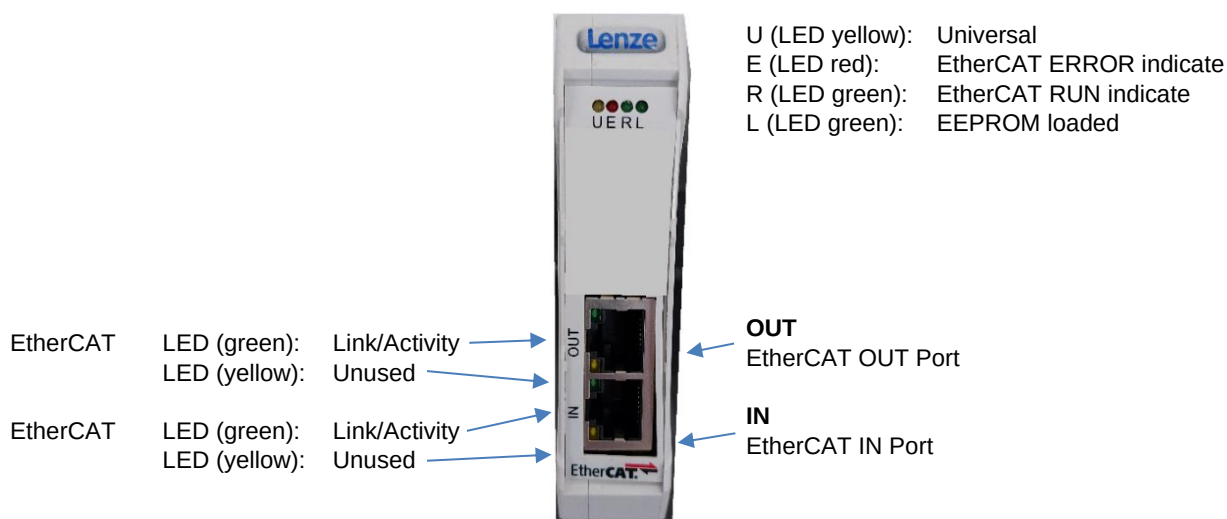


Figure 3: Connectors and LEDs

3.2.1

LED Assignment

LEDs at EtherCAT-RJ45 Connectors *IN* and *OUT*

LED	Color	LED Status	Description
Link/ Activity	green	Off	No EtherCAT link present
		Blinking	EtherCAT link present, EtherCAT activity (reception of EtherCAT data)
Spare	yellow	-	Unused

Table 1: EtherCAT LED functionality (integrated in RJ45)

Meaning of EtherCAT-LEDs U, E, R, L

LED Status	Description
Blinking	LED repeats: 200 ms on, 200 ms off.
Flicker	LED repeats: 50 ms on, 50 ms off.
single flash	LED repeats: 200 ms on, 1000 ms off.
double flash	LED repeats: 200 ms on, 200 ms off, 200 ms on, 1000 ms off.

Table 2: LED states (according to ETG.1300 documents)

Hardware Description

LED	Color	Function	LED Status	Description
U	yellow	Universal	off	No information available
			blinking	Reserved
			flicker	FoE firmware transfer is in progress
			on	FoE firmware transfer finished. Visible only for a few seconds – then actual firmware update is started
			any	Set by CoE object 0x2000.2, see 6.1.4.1.
E	red	EtherCAT ERROR Indicator	off	No error
			blinking	State change failed
			single flash	State changed due to configuration error
			double flash	SM watchdog time out
R	green	EtherCAT RUN Indicator*	off	Init
			blinking	Pre-Operational
			single flash	Safe-Operational
			on	Operational
L	green	EEPROM Loaded*	flicker	Bootstrap
			off	unable to retrieve ET1100 configuration from EEPROM
			on	successful retrieval of ET1100 configuration from EEPROM

*Directly connected to the ET1100

Table 3: EtherCAT LED functionality

4

Hardware Installation

For proper installation and setup please follow the recommended steps as shown here:

Step	Procedure	see page
	Read the safety instructions at the beginning of this document carefully before you start with the hardware installation!	3
	Danger Hazardous Voltage - Risk of electric shock due to unintentional contact with uninsulated live parts with high voltages inside of the system into which the Gateway is to be integrated. - All current circuits which are connected to the device must be sufficiently protected against hazardous voltage (SELV according to EN 60950-1) before you start with the installation. - Ensure the absence of voltage before starting any electrical work.	
1.	Mount and connect the Gateway and connect the interfaces (Power supply, CAN bus, EtherCAT).	11
2.	Please note that the CAN bus has to be terminated at both ends! Additionally, the CAN_GND signal has to be connected to earth at exactly one point in the CAN network. Any CAN node that does not support a galvanic isolation represents the equivalent of a Ground (GND) connection.	
3.	Turn on the 24 V-power supply voltage of the Gateway.	

5

Configuration

5.1

Application Example

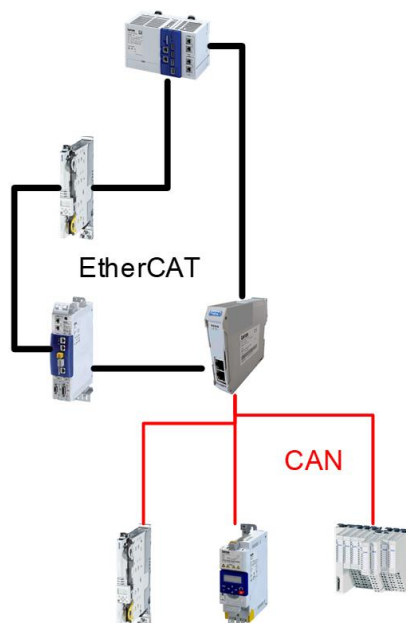


Figure 4: EtherCAT-CAN Gateway connection example

The Gateway can be placed anywhere in the EtherCAT network, except when using Distributed Clocks.

NOTE

The Gateway does not support Distributed Clocks. If Distributed Clocks are used, this must be taken into account during placement.

5.2

Configuration Sequence

The following chapter describes the CAN configuration of the Gateway by the Lenze PLC Designer.

INFORMATION

The x750 Gateway EtherCAT/CAN works only with Controllers of the c5xx Series and their successor.

For the Lenze Controller c3xx, c32xx, p3xx and p5xx please order the internal CAN Options.

5.2.1

Step 1: Add Gateway

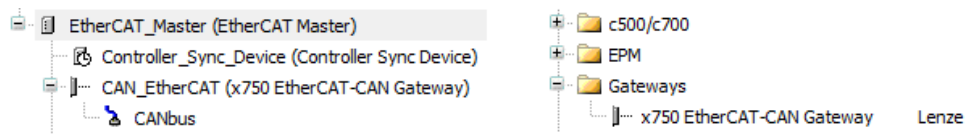


Figure 5: Add x750 Gateway EtherCAT/CAN to EtherCAT Master

Add the CANOpen Manager to the Project

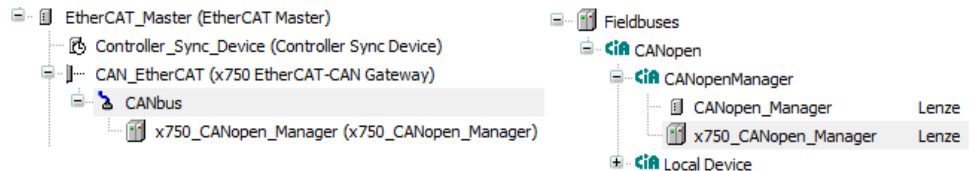


Figure 6: Add x750_CANOpen_Manager to the CANbus

5.2.2

Step 2: Add CANOpen Devices to the Project

INFORMATION

Supported devices:

- All devices imported directly from the CANopen device description file (*.eds) can be used with the x750 CANopen Manager.
- The EDS files for the Lenze devices can be found at the following link: <https://www.lenze.com/en-de/go/akb/200413930/1/>
- The CANopen devices 8400, 9400 (except CiA 402), i500 and Lenze Generic Drive available in PLC Designer are supported from PLC Designer version 3.26.

All other Lenze devices and the following Lenze device functions are not supported:

- FAST-Motion Axis Reference
- FreeEncoder reference (LM FreeEncoder)
- FastIO reference (FastIO_InRef, FastIO_OutRef)
- Parameter set download (Especially for EPM-S110 an integration must be done via the EDS file)
- Gateway function for the Lenze Engineering Tools

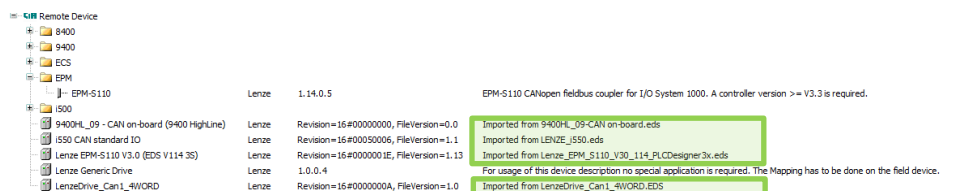


Figure 7: Connect only Imported or supported Products to x750_CANOpen_Manager

5.2.3 Step 3: Configure the Gateway

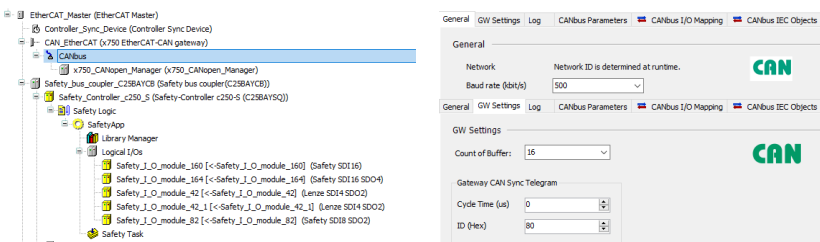
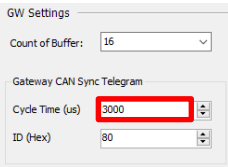
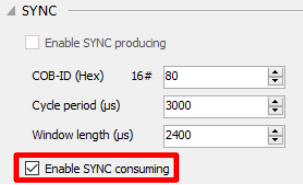
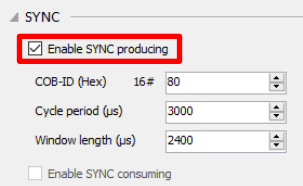


Figure 8: Configuration of the x750 Gateway EtherCAT/CAN

Details of the Parameter:

Parameter	Description																		
Network	The network number of the x750 Gateway EtherCAT/CAN can be found in the variable <i>m_usiNetId</i> (corresponds to the order of the gateways in the project tree, starting at 0). GW SettingsGeneralLogCANbus I/O MappingCANbus IEC ObjectsStatusInformation + Add... ✎ Edit... ✕ DeleteGo to Variable <table><tr><th>Expression</th><th>Type</th><th>Value</th></tr><tr><td>Device.Application.CAN_EtherCAT_CMD</td><td>Lenze_CANMiniDriver.Lenze_CANMiniDriver</td><td></td></tr><tr><td>m_usiNetId</td><td>USINT</td><td>0</td></tr><tr><td>sHardwareVersion</td><td>STRING</td><td>'1.3'</td></tr><tr><td>sFirmwareVersion</td><td>STRING</td><td>'2.7'</td></tr><tr><td>Device.Application.CAN_EtherCAT_CANbus</td><td>Lenze_CANMiniDriver.ESDCANbus</td><td></td></tr></table> Note: It should be noted that <i>m_usiNetId</i> must be incremented by 1 if <i>m_usiNetId</i> is used as parameter Network in the CAA CiA405 library.	Expression	Type	Value	Device.Application.CAN_EtherCAT_CMD	Lenze_CANMiniDriver.Lenze_CANMiniDriver		m_usiNetId	USINT	0	sHardwareVersion	STRING	'1.3'	sFirmwareVersion	STRING	'2.7'	Device.Application.CAN_EtherCAT_CANbus	Lenze_CANMiniDriver.ESDCANbus	
Expression	Type	Value																	
Device.Application.CAN_EtherCAT_CMD	Lenze_CANMiniDriver.Lenze_CANMiniDriver																		
m_usiNetId	USINT	0																	
sHardwareVersion	STRING	'1.3'																	
sFirmwareVersion	STRING	'2.7'																	
Device.Application.CAN_EtherCAT_CANbus	Lenze_CANMiniDriver.ESDCANbus																		
Baudrate	CAN-Baudrate (Possible Baudrates see Chapter 6.1.5.9)																		
Count of Buffer	Number of Rx/Tx messages that can be processed in one EtherCAT cycle. (Rx/Tx-queue-size; see chapter 6.1.5.5)																		

Parameter	Description
CAN Sync Telegram 	Generation in the gateway (Cycle Time >0) The x750 Gateway EtherCAT/CAN can independently generate a highly accurate equidistant sync telegram for the CAN bus (cyclic telegram). In this case the SYNC reception must be activated at the <i>x750_CANOpen_Manager</i>.  Parameter <i>x750_CANOpen_Manager</i> Generation in the x750_CANOpen_Manager If the SYNC telegram is to be generated in the <i>x750_CANOpen_Manager</i>, the setting must be activated in the <i>x750_CANOpen_Manager</i> and the Cycle Time in the x750 Gateway EtherCAT/CAN (see Chapter 6.1.4.2) must be set to 0. Note: The cycle time of the Sync telegram should be selected 3 times as high as the EtherCAT cycle time (e.g., 1ms EtherCAT cycle time, 3ms cycle time of the SYNC telegram) to achieve an equidistant distance between the SYNC telegrams on the CAN.  Parameter <i>x750_CANOpen_Manager</i>
Cycle Time	see Chapter 6.1.4.2
ID	see Chapter 6.1.4.2

INFORMATION

Details of Parameter Setup of CAN Devices can be found in the Online Help of the PLC Designer

(Engineering Tools ⇒ PLC Designer ⇒ Fieldbus configuration editors ⇒ CAN-Based Fieldbusses ⇒ Canopen Manager (Master))

The following Topics are not supported

- CANopen Safety
- Redundancy

INFORMATION

Project planning recommendation

- Up to 10 CAN stations per Gateway with one RxPDO and TxPDO each with 8 bytes of data per participant (CAN cycle time depending on the used CAN baud rate).
- Cycle time EtherCAT = 1/3 * Cycle time CAN

6 EtherCAT Communication

6.1 CAN Interface

The CAN Interface is based on a modular device profile (Fieldbus Gateway, Profile No. 5000), and it supports one CAN module. This module includes one CAN Tx message queue in the output area and one CAN Rx message queue in the input area.

6.1.1 Object Dictionary Structure

The object dictionary is composed of the following areas:

Index	Object Dictionary Areas
0000 _h ... 0FFF _h	Data Type Area
1000 _h ... 1FFF _h	Communication Area
2000 _h ... 5FFF _h	Vendor Specific Area
6000 _h ... 6FFF _h	Input Area (CAN Rx message queue)
7000 _h ... 7FFF _h	Output Area (CAN Tx message queue)
8000 _h ... 8FFF _h	Configuration Area (CAN interface configuration)
F000 _h ... FFFF _h	Device Area

Table 4: Object dictionary structure

The following explains the definition of a standard and an extended CAN message queue. For proper operation one of both CAN message queues must be chosen. This can be accomplished by writing the CAN interface settings object (8000_h). The RPDO and TPDO mapping objects (1600_h and 1A00_h) will change accordingly.

6.1.1.1 Output Data

The CAN interface output data include the Tx message queue plus the control data for the Rx and Tx message queues. The CAN interface output data is always required.

6.1.1.2 Input Data

The CAN interface input data include the Rx message queue plus the status information for the Rx and Tx message queues. The CAN interface input data is always required.

6.1.2

Object Dictionary

The Gateway layer 2 implementation supports the following objects:

Index	Name
1000 _h	Device type
1008 _h	Device name
1009 _h	Hardware version
100A _h	Software version
1018 _h	Identity
1600 _h	RPDO-Map CAN interface
1A00 _h	TPDO-Map CAN interface
1C00 _h	Sync manager type
1C12 _h	RPDO assign
1C13 _h	TPDO assign
2000 _h	Other Settings
2010 _h	Statistics
6000 _h	CAN interface input (11-bit identifier)
6001 _h	CAN interface input (29-bit identifier)
7000 _h	CAN interface output (11-bit identifier)
7001 _h	CAN interface output (29-bit identifier)
8000 _h	CAN interface configuration
8001 _h	CAN filter table
F000 _h	Modular Device Profile
F108 _h	CAN Status
F800 _h	CAN bus parameter

6.1.3 Standard Objects (1000_h...1FFF_h)

6.1.3.1 Object 1000_h Device Type

Index	Subindex	Description	Data Type	RW	Default
1000 _h	0	<i>Device Type</i>	UINT32	RO	13881389 _h

Variable Description

Low Word: Used CoE profile (5001_h)

High Word: Module Profile according to the modular device profile 5000_h

6.1.3.2 Object 1008_h Device Name

Index	Subindex	Description	Data Type	RW	Default
1008 _h	0	<i>Device Name</i>	STRING5	RO	MEESC

Variable Description

Gateway Device Name.

6.1.3.3 Object 1009_h Hardware Version

Index	SubIndex	Description	Data Type	RW	Default
1009 _h	0	<i>Hardware Version</i>	STRING3	RO	1.3

Variable Description

Gateway Hardware Version.

6.1.3.4 Object 100A_h Software Version

Index	SubIndex	Description	Data Type	RW	Default
100A _h	0	<i>Software Version</i>	STRING3	RO	2.6

Variable Description

Gateway Software Version.

6.1.3.5 Object 1018_h Identity

Index	SubIndex	Description	Data Type	RW	Default
1018 _h	0	Number of sub-indexes	UINT8	RO	4 _h
	1	Vendor ID	UINT32	RO	3B _h (59 _d)
	2	Product code	UINT32	RO	37500000 _h
	3	Revision	UINT32	RO	see below
	4	Serial number	UINT32	RO	see below

Variable Description

Gateway identification characteristics.

Vendor ID Lenze Vendor-ID = 3B_h (59_d)

Product code Gateway product code

Revision Revision number of the Gateway

Serial number Serial number

6.1.3.6 Object 1C00_h Sync Manager Type

Index	SubIndex	Description	Data Type	RW	Default
1C00 _h	0	Number of sub-indexes	UINT8	RO	4 _h
	1 _h	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	1 _h
	2 _h	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	2 _h
	3 _h	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	3 _h
	4 _h	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	4 _h

Variable Description

Sync-Manager Type:

Sync-Manager Type Channel 1: Mailbox Write

Sync-Manager Type Channel 2: Mailbox Read

Sync-Manager Type Channel 3: Process Data Write (Outputs)

Sync-Manager Type Channel 4: Process Data Read (Inputs)

6.1.3.7

Object 1600_h RPDO-Map CAN-Interface

This object defines the CAN interface mapping into the EtherCAT input data.

The first three sub-indexes contain the size of the Tx and Rx counters plus the number of Tx messages. The size of the CAN Rx message queue is configured through object 8000_h.

Object 8000_h is also used to define the CAN message ID mode, either 11-bit (Object 7000_h) or 29bit (7001_h). Depending on the settings the contents of objects 7000_h and 7001_h are mapped in object 1600_h.

Object 1600_h is always required and must be defined in the PDO Assign Object 1C12_h, subindex 1_h.

Index	SubIndex	Description	Data Type	RW
1600 _h	0	Number of CAN-Messages+3	UINT8	RO
	1 _h	1. PDO Mapping entry, (Object 700z _h (CAN interface output), entry 01 _h (Tx Counter))	UINT32	RO
	2 _h	2. PDO Mapping entry (Object 700z _h (CAN interface output), entry 02 _h (Rx Counter))	UINT32	RO
	3 _h	3. PDO Mapping entry (Object 700z _h (CAN interface output), entry 03 _h (Number of Tx Messages))	UINT32	RO
	4 _h	4. PDO Mapping entry (Object 700z _h (CAN interface output), entry 04 _h (Tx Message 1))	UINT32	RO
	...	..	...	...
	m _h	m. PDO Mapping entry (Object 700z _h (CAN interface output), entry m (Tx Message m-3))	UINT32	RO

6.1.3.8

Object 1A00_h TPDO-Map CAN-Interface

This object defines the CAN interface mapping into the EtherCAT output data.

The first three sub-indexes contain the size of the Tx and Rx counters plus the number of Tx messages. The size of the CAN Tx message queue is configured through object 8000_h.

Object 8000_h is also used to define the CAN message ID mode, either 11-bit (object 7000_h) or 29 Bit (object 7001_h). Depending on the settings the contents of objects 6000_h and 6001_h are mapped in object 1A00_h.

Object 1A00_h is always required and must be defined in the PDO Assign Object 1C13_h, subindex 1_h.

Index	SubIndex	Description	Data Type	RW
1A00 _h	0	Number of CAN-Messages+4	UINT8	RO
	1 _h	1. <i>PDO Mapping entry</i> (Object 6000 _h (CAN interface input), entry 01 _h (Tx Counter))	UINT32	RO
	2 _h	2. <i>PDO Mapping entry</i> (Object 6000 _h (CAN interface input), entry 02 _h (Rx Counter))	UINT32	RO
	3 _h	3. <i>PDO Mapping entry</i> (Object 6000 _h (CAN interface input), entry 03 _h (Number of Rx Messages))	UINT32	RO
	4 _h	4. <i>PDO Mapping entry</i> (Object 6000 _h (CAN interface input), entry 04 _h (Tx Transaction Number))	UINT32	RO
	5 _h	5. <i>PDO Mapping entry</i> (Object 6000 _h (CAN interface input), entry 05 _h (Rx Message 1))	UINT32	RO
	...	..		...
	m _h	m. <i>PDO Mapping entry</i> (Object 6000 _h (CAN interface input), entry m (Rx Message m-4))	UINT32	RO

6.1.3.9 Object 1C12_h RPDO-Assign

Object 1C12_h assigns the mapping of the CAN interface RPDOs.

Index	SubIndex	Description	Data Type	RW	Default
1C12 _h	0	Number of sub-indexes	UINT8	RO	1 _h
	1 _h	CAN-Interface RPDO	UINT16	RO	1600 _h

6.1.3.10 Object 1C13_h TPDO-Assign

Object 1C13_h assigns the mapping of the CAN interface TPDOs.

Index	SubIndex	Description	Data Type	RW	Default
1C13 _h	0	Number of sub-indexes	UINT8	RO	1 _h
	1 _h	CAN-Interface TPDO	UINT16	RO	1600 _h

6.1.4 Manufacturer Specific Objects (2000_h-5FFF_h)

6.1.4.1 Object 2000_h Other Settings

Index	Subindex	Name	Data Type	RW	Default
2000 _h	0	Number of sub-indexes	UINT8	RO	2 _h
	1 _h	Reserved	BOOL	RO	0
	2 _h	Custom LED state	UINT8	RW	0

Parameter Description

Custom LED state: Used to overwrite the state of the “Universal” LED (Chapter 3.2.1) Values / State:

- 0: Off
- 1..4: Flash x1...Flash x4
- 13: Blink
- 14: Flicker
- 15: On

Reading this value reflects only the value that was last written – not the actual LED state. The state set by writing this object is overwritten whenever the Gateway itself sets a state for the LED (i.e., the LED is usually turned off when the EtherCAT State changes)

6.1.4.2

Object 2002_h Cyclic CAN Telegram

Configuration of a cyclic CAN telegram with fixed contents.

Index	Subindex	Name	Data Type	RW	Default
2002 _h	0	<i>Number of sub-indexes</i>	UINT8	RO	4 _h
	1 _h	<i>Cycle Time</i>	UINT32	RW	0 _h
	2 _h	<i>Id</i>	UINT32	RW	80 _h (128 _d)
	3 _h	<i>DLC (Data Length Code)</i>	UINT8	RW	0 _h
	4 _h	<i>Data</i>	OCTET-STRING [8]	RW	00000000 _h

Parameter Description

Cycle Time 0 - Deactivated
 Cycle Time in μs (min. 10 μs)
 (mathematical) rounded to 10 μs resolution.
 e.g. 224 μs $\Rightarrow$ 220 μs
 225 μs $\Rightarrow$ 230 μs .

Id Identifier of the cyclic CAN Frame

DLC Data Length Code of the cyclic CAN Frame

Data Data of the cyclic CAN Frame

INFORMATION

Writing of the parameters is only possible in EtherCAT status Pre-Operation.
Cycle Time and *Id* are part of the Gateway Configuration, see Chapter 5.2.2.

NOTICE

For cyclic telegrams it must be noted that CAN bus and EtherCAT are not synchronized.
 This can lead to the fact that the times of the receipt of the cyclic telegrams fluctuate.

6.1.4.3

Object 2010_h Statistics

Index	Sub-Index	Name	Data Type	RW	Default
2010 _h	0	Number of sub-indexes	UINT8	RO	63 _d
	1 _h	Reset	UINT32	RW	
	2 _h	Cyclic handler time (min.)	UINT32	RO	
	3 _h	Cyclic handler time (max.)	UINT32	RO	
	4 _h	Cyclic handler time (avg.)	UINT32	RO	
	5 _h	CAN handler time (min.)	UINT32	RO	
	6 _h	CAN handler time (max.)	UINT32	RO	
	7 _h	CAN handler time (avg.)	UINT32	RO	
	8 _h	Watchdog triggered	UINT32	RO	
	30 _h	CAN Frames Tx Requested	UINT32	RO	
	31 _h	CAN Frames Tx	UINT32	RO	
	32 _h	CAN Frames Rx	UINT32	RO	
	34 _h	CANDriver Controller overrun	UINT32	RO	
	35 _h	CANDriver FIFO overrun	UINT32	RO	
	36 _h	CANDriver Error Frames	UINT32	RO	
	37 _h	CANDriver Aborted Frames	UINT32	RO	
	38 _h	CANDriver Rx Frames	UINT32	RO	
	39 _h	CANDriver Rx RTR Frames	UINT32	RO	
	4A _h	CANDriver Rx Frames Ext.	UINT32	RO	
	4B _h	CANDriver Rx RTR Frames Ext.	UINT32	RO	
	4C _h	CANDriver Tx Frames	UINT32	RO	
	4D _h	CANDriver Tx RTR Frames	UINT32	RO	
	4E _h	CANDriver Tx Frames Ext.	UINT32	RO	
	4F _h	CANDriver Tx RTR Frames Ext.	UINT32	RO	

Parameter Description

Parameter	Description
Reset	When this object is written the statistics are reset. (Reading this value shows the time stamp of the last reset – milliseconds since device start up)
Cyclic handler time (min.)	For debugging purposes only. (Minimum time in application's cyclic handler within its last 10000 calls, in μ s)
Cyclic handler time (max.)	For debugging purposes only. (Maximum time in application's cyclic handler within its last 10000 calls, in μ s)
Cyclic handler time (avg.)	For debugging purposes only. (Average time in application's cyclic handler within its last 10000 calls, in μ s)
CAN handler time (min.)	For debugging purposes only. (Minimum time in application's CAN handler within its last 1000 calls, in μ s)
CAN handler time (max.)	For debugging purposes only. (Maximum time in application's CAN handler within its last 1000 calls, in μ s)
CAN handler time (avg.)	For debugging purposes only. (Average time in application's CAN handler within its last 1000 calls, in μ s)
Watchdog triggered	Times application watchdog was triggered due to missing process data (Outputs). (Watchdog value is calculated by ESC registers 400 _h and 420 _h)
CAN Frames Tx Requested	Total No. of CAN frames that should have been sent – according to write accesses to “Tx Counter” objects. etc.
CAN Frames Tx	Number of CAN frames that were successfully forwarded to the CAN driver
CAN Frames Rx	Total No. of CAN frame successfully copied to the Rx objects
CANDriver Controller overrun	CAN Driver statistics: No. of CAN Controller overruns
CANDriver FIFO overrun	CAN Driver statistics: No. of FIFO overruns
CANDriver Error Frames	CAN Driver statistics: No. of error frames
CANDriver Aborted Frames	CAN Driver statistics: No. of aborted frames
CANDriver Rx Frames	CANDriver Rx RTR Frames
CANDriver Rx RTR Frames	CAN Driver statistics: No. of standard RTR frames received
CANDriver Rx Frames Ext.	CAN Driver statistics: No. of extended frames received
CANDriver Rx RTR Frames Ext.	CAN Driver statistics: No. of extended RTR frames received
CANDriver Tx Frames	CAN Driver statistics: No. of standard frames sent
CANDriver Tx RTR Frames	CAN Driver statistics: No. of standard RTR frames sent
CANDriver Tx Frames Ext.	CAN Driver statistics: No. of extended frames sent
CANDriver Tx RTR Frames Ext.	CAN Driver statistics: No. of extended RTR frames sent

6.1.5

Profile Specific Objects (6000_h-FFFF_h)

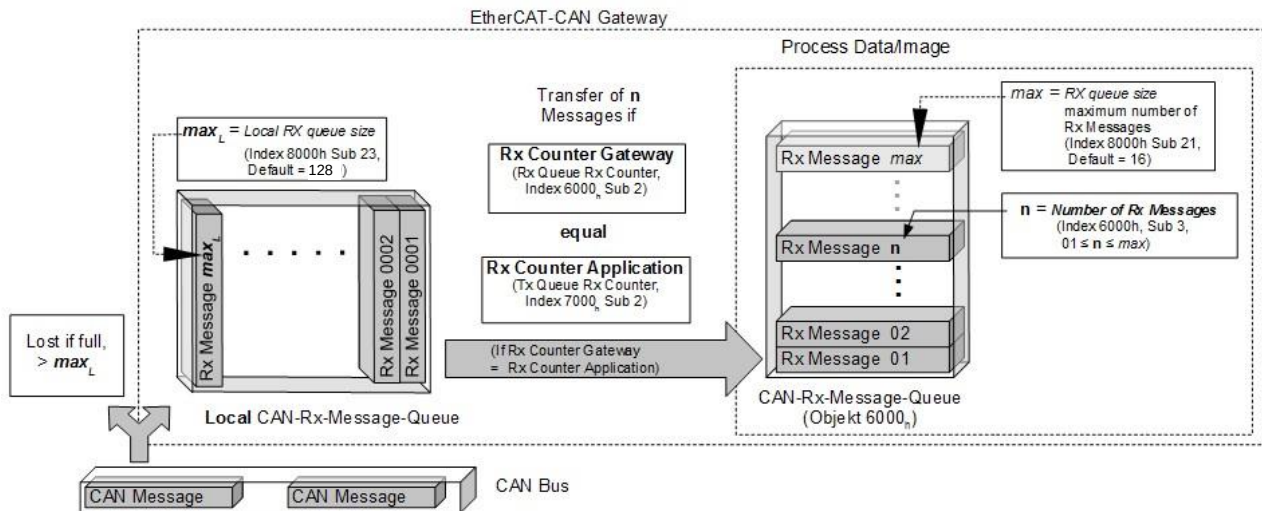
These objects are identical for all EtherCAT Slave devices supporting the profile number 5000 ("CAN Interface").

6.1.5.1

Object 6000_h CAN Rx Message Queue

Relationship of message queues

(here according to the default configuration of queue sizes)



$n...$ Number of Rx messages, which are accumulated now of test / comparison of the „Rx counter“(Number of Messages, Index 6000_h, Subindex 3_h), but maximum 16 (Default, Index 8000_h Subindex 21_h)

Figure 9: Relationship of the CAN Rx message queues

The number of transmitted Rx messages (n) is written in Number of Rx Messages (Sub-Index 3_h) and must not be changed until the "Rx Counters" are equal again. For the chronological sequence see the example in Figure 11 on page 28.

Index	Subindex	Name	Data Type	RW
6000 _h	0	Number of sub-indexes	UINT8	RO
	1 _h	Tx Counter Gateway	UINT16	RO
	2 _h	Rx Counter Gateway	UINT16	RO
	3 _h	Number of Rx Messages	UINT16	RO
	4 _h	Tx Transaction Number	UINT16	RO
	5 _h	Rx Message 1	OCTET-STRING [10]	RO
	...	...	...	...
	m _h	Rx Message m-4	OCTET-STRING [10]	RO

This object contains the CAN interface input messages with 11 Bit ID.

Parameter Description

Parameter	Description
<i>Tx Counter Gateway</i>	The Tx counter is increased by the Gateway to indicate that the CAN Tx messages were copied from the output data to the local CAN send queue (see Figure 12).
<i>Rx Counter Gateway</i>	The Rx counter is increased by the Gateway every time when new CAN Rx data arrived and the <i>Rx Counter Gateway</i> (6000 _h , sub-index 02) is identical with <i>Rx Counter Application</i> (7000 _h , sub-index 02). This indicates that new Rx data has been written into the process input data (see Figure 9).
<i>Number of Rx Messages</i>	Contains the number of CAN Rx messages in the following input data when the Rx Counter was increased (1...m-4).
<i>Tx Transaction Number</i>	Contains the transaction number of the last sent Tx (see Figure 10)
<i>Rx Message 1...(m-4)</i>	1. to (m-4). CAN Rx message The message is composed of the following components: - Bit 0-3: CAN message length (0...8 bytes) - Bit 4: RTR Bit - Bit 5-15: CAN Identifier (11-bit CAN ID) - Bit 16-79: CAN-Rx data

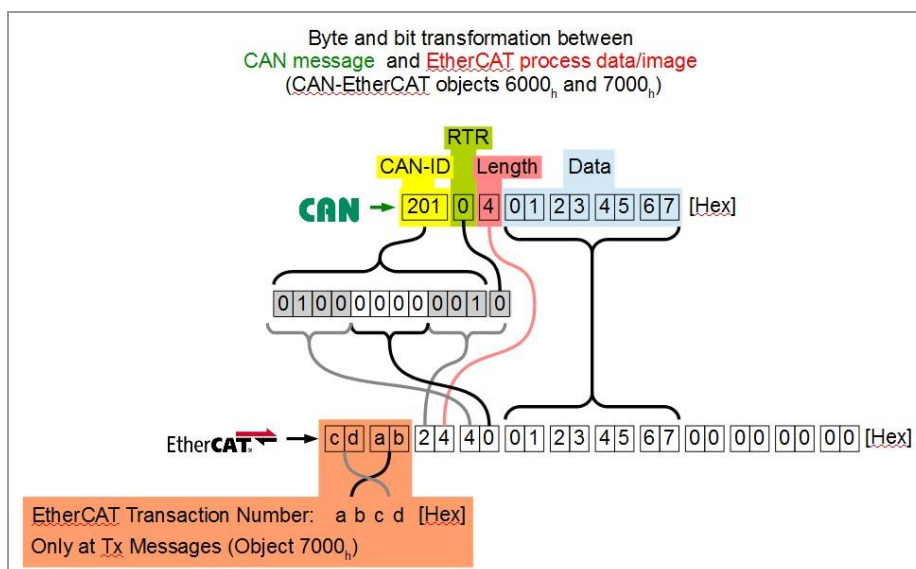


Figure 10: Formatting of the CAN data in the EtherCAT process image (11-bit Rx and Tx)

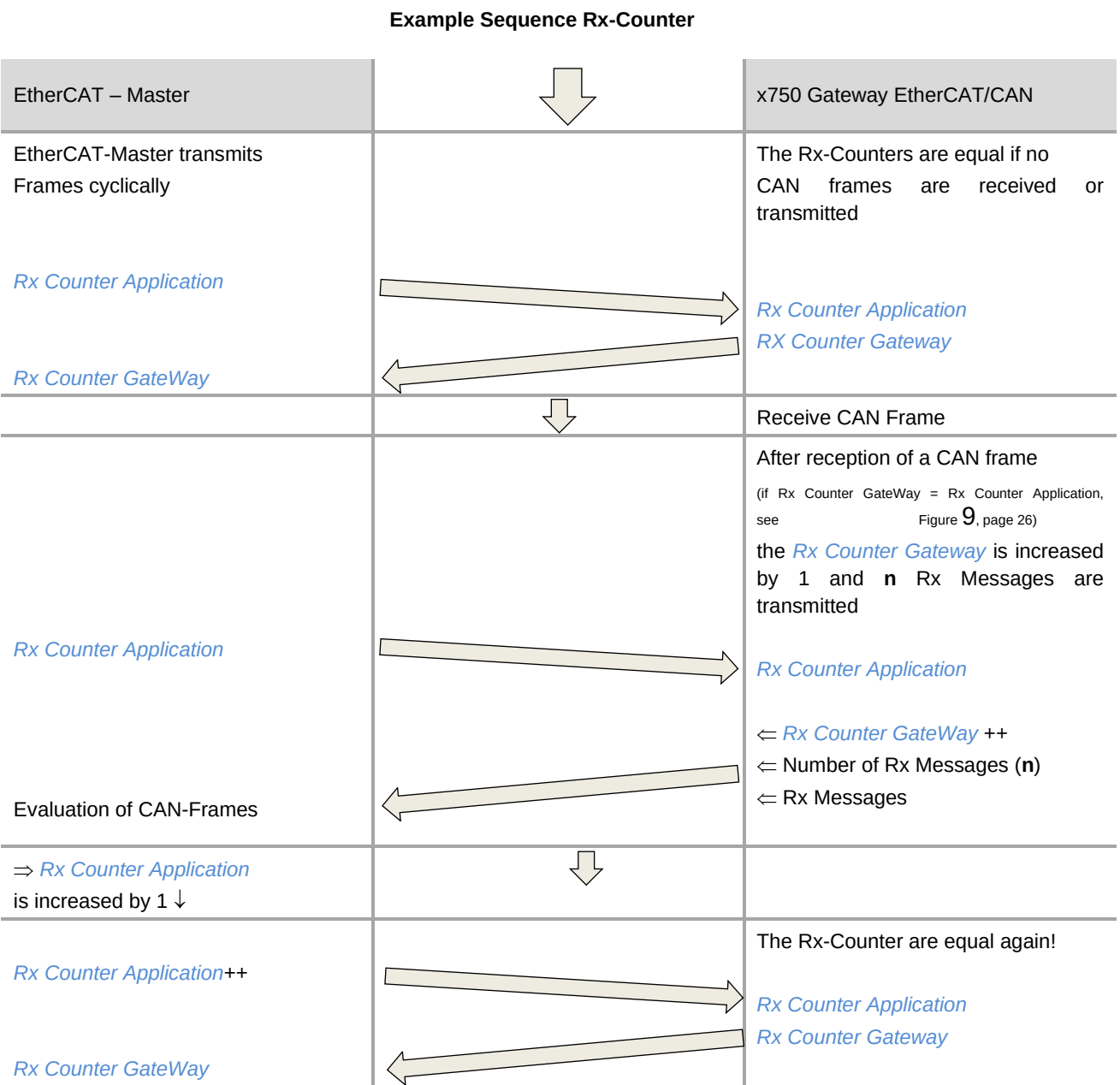


Figure 11: Chronological sequence of the Rx-Counters

6.1.5.2

Object 6001_h CAN Rx Extended Message Queue

Index	Subindex	Name	Data Type	RW
6001 _h	0	Number of sub-indexes	UINT8	RO
	1 _h	Tx Counter Gateway	UINT16	RO
	2 _h	Rx Counter Gateway	UINT16	RO
	3 _h	Number of Rx Messages	UINT16	RO
	4 _h	Tx Transaction Number	UINT16	RO
	5 _h	Rx Message 1	OCTET-STRING [14]	RO
	...			
	m _h	Rx Message m-4	OCTET-STRING [14]	RO

This object contains the CAN interface input messages with 29-bit ID.

See Figure 9 for the relationship of the CAN Rx Message Queues. For 29-Bit-Identifiers objects 6001_h and 7001_h are used instead of objects 6000_h and 7000_h accordingly. For the chronological sequence see the example in Figure 12 on page 30.

Parameter Description

Parameter	Description
<i>Tx Counter Gateway</i>	The Tx counter is increased by the Gateway to indicate that the CAN Tx messages were copied from the output data to the CAN send queue (see Figure 12).
<i>Rx Counter Gateway</i>	The Rx counter is increased by the Gateway every time when new CAN Rx data arrived and the <i>Rx Counter Gateway</i> (6001 _h , sub-index 2 _h) is identical with <i>Rx Counter Application</i> (7001 _h , sub-index 2 _h). This indicates that new Rx data has been written into the process input data (see Figure 9).
<i>Number of Rx Messages</i>	Contains the number of CAN Rx messages in the following input data when the Rx counter was increased (1...m-4).
<i>Tx Transaction Number</i>	Contains the transaction number of the last sent Tx message (see Figure 10).
<i>Rx Message 1...(m-4)</i>	1 to (m-4). CAN Rx message The message is composed of the following components: Bit 0-3: CAN-Rx message length (0...8 byte) Bit 5-15: reserved Bit 16-44: CAN Identifier (11- or 29-bit CAN identifier) Bit 46: RTR bit Bit 47: 0 = 11-bit CAN identifier 1 = 29-bit CAN identifier Bit 48-111: CAN Rx data

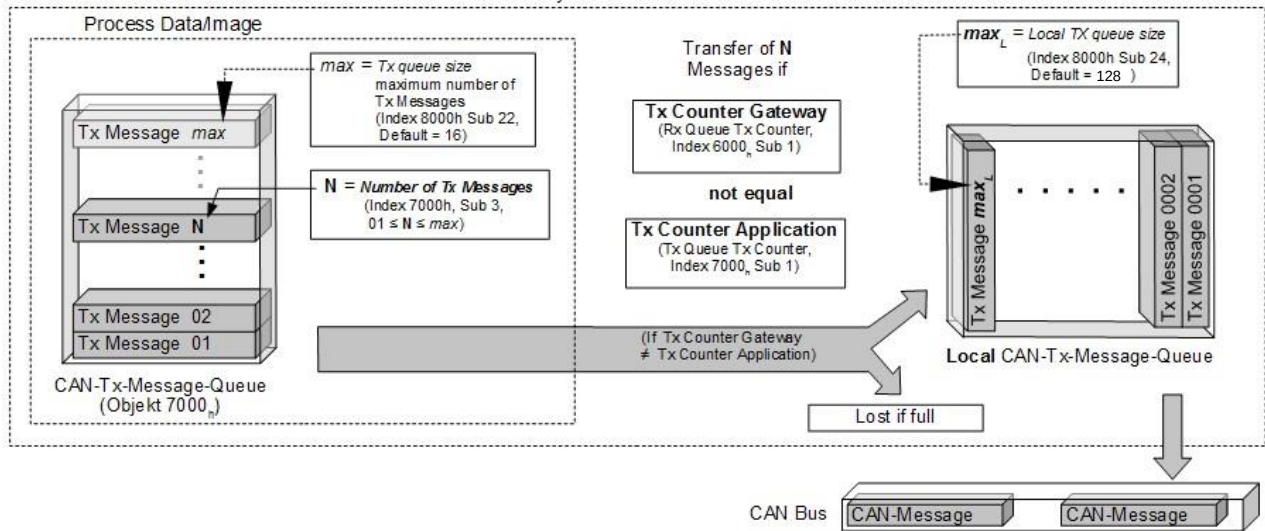
6.1.5.3

Object 7000_h CAN Tx Message Queue

Relationship of message queues

(here according to the default configuration of queue sizes)

EtherCAT-CAN Gateway



N ... Number of Tx messages, which are accumulated at the moment of test / comparison of the Tx counter“(Number of Tx Messages, Index 7000_h, Subindex 3_h), but maximum 16 (Default, Tx queue size Index 8000_h Subindex 22_h).

Figure 12: Relationship of the CAN Tx message queues

The number of effectively transmitted Tx messages (N) is written in Number of Tx Messages (SubIndex 3_h) and will not be changed until the "Tx Counters" are equal again.

For the chronological sequence see the example in Figure 13 on page 32.

Index	Subindex	Name	Data Type	RW	Default
7000 _h	0	Number of sub-indexes	UINT8	RO	
	1 _h	Tx Counter Application	UINT16	RO	
	2 _h	Rx Counter Application	UINT16	RO	
	3 _h	Number of Tx Messages	UINT16	RO	
	4 _h	Tx Message 1	OCTET-STRING [12]	RO	
	...	...	...	...	
	m _h	Tx Message m-3	OCTET-STRING [12]	RO	

This object contains the CAN interface output messages with 11-bit ID.

The maximum value of the sub-index, and thus the number of Tx messages, is defined in the RxPDO-Mapping-Object (object 1600_h).

Parameter Description

Parameter	Description
<i>Tx Counter Application</i>	This counter must be increased when or after writing the CAN Tx message to the output data (see Figure 12).
<i>Rx Counter Application</i>	This counter must be increased by the EtherCAT Master application for each CAN Rx message list it has received and read. This indicates that the received Rx messages have been read (see Figure 9).
<i>Number of Tx Messages</i>	Contains the number of CAN Tx messages, which are transmitted with every increase of the Tx counter ($N = 1 \dots m-3$).
<i>Tx Message 1...(m-3)</i>	CAN Tx messages which are transmitted with every increase of the Tx counter. The message is composed of the following components: - Bit 0-15: Transaction Number The transaction number of the last transmitted CAN Tx message; readable in the input data. - Bit 16-19: CAN message length (0...8 bytes) - Bit 20: RTR bit - Bit 21-31: CAN identifier (11-bit CAN ID) - Bit 32-95: CAN Tx data See Figure 10 on page 27.

Example Chronological Sequence Tx-Counter

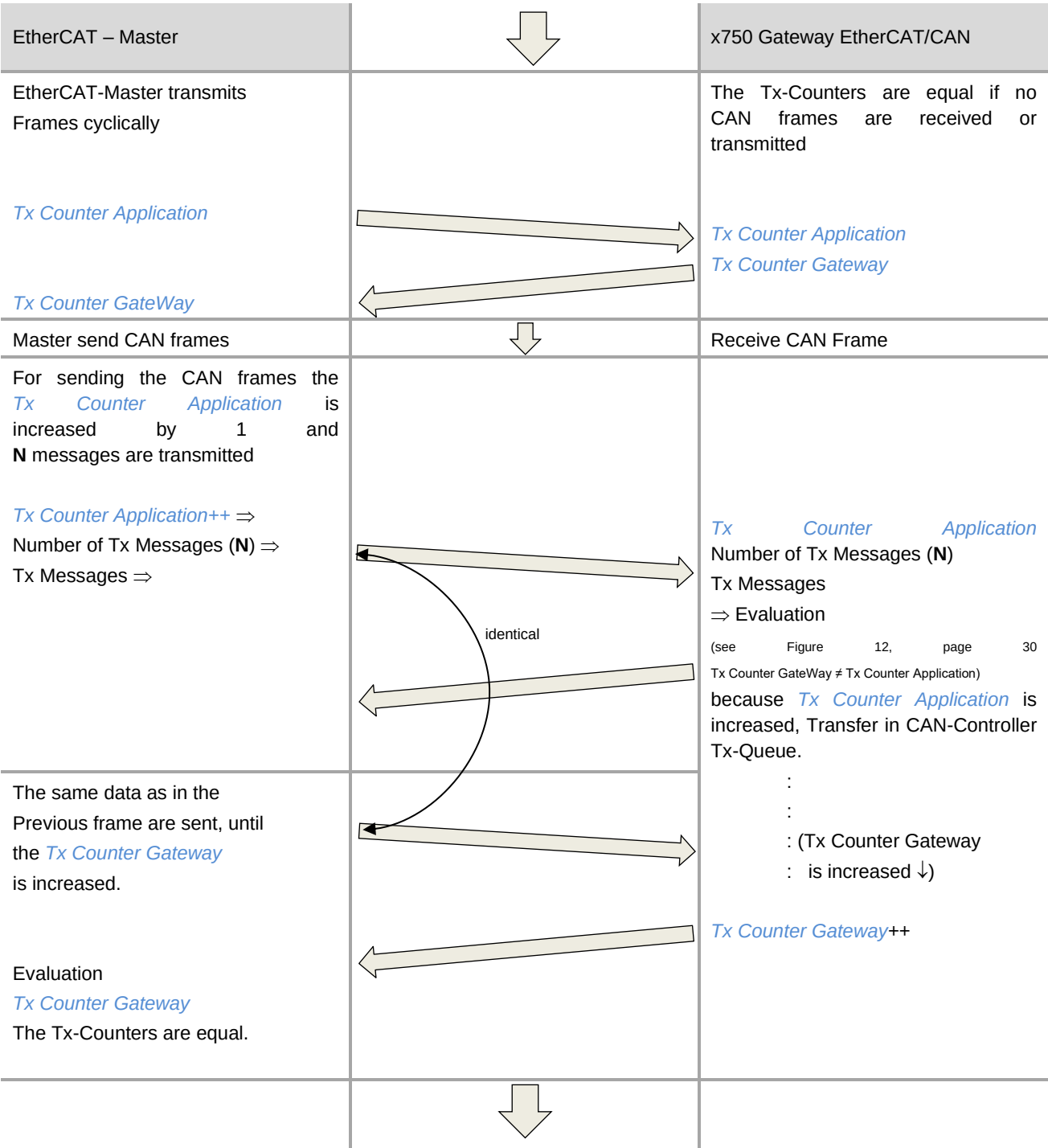


Figure 13: Chronological Sequence Tx-Counter

6.1.5.4

Object 7001 CAN Tx Extended Message Queue

NOTICE

The following CoE objects (Index, Sub-Index) can only be written in *Pre-Operational* state: Index: 7001_h, Sub-Index 4...m_h

Index	SubIndex	Name	Data Type	RW
7001 _h	0	Number of sub-indexes	UINT8	RO
	1 _h	Tx Counter Application	UINT16	RO
	2 _h	Rx Counter Application	UINT16	RO
	3 _h	Number of Tx Messages	UINT16	RO
	4 _h	Tx Message 1	OCTET-STRING [16]	RO
	...	...	...	...
	m _h	Tx Message m-3	OCTET-STRING [16]	RO

This object contains the CAN interface input messages with 29-Bit ID.

See Figure 12 for the relationship of the CAN Tx Message Queues. For 29-Bit-Identifiers objects 6001_h and 7001_h are used instead of objects 6000_h and 7000_h accordingly.

Parameter Description

Tx Counter Application	This counter must be increased when or after writing the CAN Tx message to the output data (see Figure 12).														
Rx Counter Application	This counter must be increased by the EtherCAT Master application for each CAN Rx message list it has received and read. This indicates that the received Rx messages have been read (see Figure 9).														
Number of Tx Messages	Contains the number of CAN Tx messages which are transmitted with every increase of the Tx counter (1...m-3).														
Tx Message 1...(m-3)	CAN Tx messages which are transmitted with every increase of the Tx counter. The message is composed of the following components: <table> <tr> <td>Bit 0-15:</td><td>Transaction Number</td></tr> <tr> <td></td><td>The transaction number of the last transmitted CAN Tx message; readable in the input data.</td></tr> <tr> <td>Bit 16-31:</td><td>CAN message length (0...8 byte)</td></tr> <tr> <td>Bit 32-60:</td><td>CAN Identifier (11- or 29-bit CAN ID)</td></tr> <tr> <td>Bit 62:</td><td>RTR bit</td></tr> <tr> <td>Bit 63:</td><td>0 = 11-bit CAN identifier 1 = 29-bit CAN identifier</td></tr> <tr> <td>Bit 64-127:</td><td>CAN Tx data</td></tr> </table>	Bit 0-15:	Transaction Number		The transaction number of the last transmitted CAN Tx message; readable in the input data.	Bit 16-31:	CAN message length (0...8 byte)	Bit 32-60:	CAN Identifier (11- or 29-bit CAN ID)	Bit 62:	RTR bit	Bit 63:	0 = 11-bit CAN identifier 1 = 29-bit CAN identifier	Bit 64-127:	CAN Tx data
Bit 0-15:	Transaction Number														
	The transaction number of the last transmitted CAN Tx message; readable in the input data.														
Bit 16-31:	CAN message length (0...8 byte)														
Bit 32-60:	CAN Identifier (11- or 29-bit CAN ID)														
Bit 62:	RTR bit														
Bit 63:	0 = 11-bit CAN identifier 1 = 29-bit CAN identifier														
Bit 64-127:	CAN Tx data														

6.1.5.5

Object 8000_h CAN Interface Configuration

Index	Sub-Index	Name	Data Type	RW	Default
8000 _h	0	<i>Number of sub-indexes</i>	UINT8	RO	24 _h (36 _d)
	1 _h	<i>Node Address</i>	UINT16	RO	0000 _h
	20 _h	<i>Flags</i>	UINT16	RO	0000 _h
	21 _h	<i>Rx queue size</i>	UINT8	RO	10 _h (16 _d)
	22 _h	<i>Tx queue size</i>	UINT8	RO	10 _h (16 _d)
	23 _h	<i>Local Rx queue size</i>	UINT16	RO	80 _h (128 _d)
	24 _h	<i>Local Tx queue size</i>	UINT16	RO	80 _h (128 _d)

The CAN interface can be configured with this object.

Parameter Description

Node Address Must be set to 0

Flags Bit 0-2: Reserved for future extensions; must be 0
 Bit 3: 0 = Standard Queue (11-bit identifier),
 1 = Extended Queue (29-bit identifier)

Bit 4-14: Reserved for future extensions; must be 0

Rx queue size Number of Rx messages; max. 63_d

Tx queue size Number of Tx messages; max. 63_d

Local Rx queue size Rx queue size of the internal CAN driver

Local Tx queue size Tx queue size of the internal CAN driver

6.1.5.6

Object 8001 CAN Rx Filter Table

Index	SubIndex	Name	Data Type	RW
8001 _h	0	Number of sub-indexes	UINT8	RO
	1 _h	Identifier Area 1	UINT64	RO
	...	...	...	...
	m _h	Identifier Area m	UINT64	RO

This object assigns the CAN identifier areas, which are filled into the Rx queue and are transmitted with the EtherCAT input data.

INFORMATION

For 29-Bit CAN identifiers bit 31 and 63 must be set!

In case this object is not configured, all received CAN messages will be assigned to the Rx queue and transmitted through the EtherCAT input data.

Parameter Description

Identifier Area 1 Byte 0-3: First identifier to be assigned to the Rx queue
 Byte 4-7: Last identifier to be assigned to the Rx queue
 Identifier Area m Byte 0-3: First identifier to be assigned to the Rx queue
 Byte 4-7: Last identifier to be assigned to the Rx queue
 m = max. 4_h (4_d)

6.1.5.7

Object F000 Modular Device Profile

Usually only needed by configuration tool PLC Designer. See ETG.5100 documents for details.

Index	SubIndex	Name	Data Type	RW	Default
F000 _h	0	Max. sub-index	UINT8	RO	3 _h
	1 _h	Index distance	UINT16	RO	16 _d
	2 _h	Maximum number of modules	UINT16	RO	1 _h
	3 _h	General configuration	UINT32	RO	1 _h

6.1.5.8

Object F108_h CAN Status

Index	SubIndex	Name	Data Type	RW	Default
F108 _h	0	Max. sub-index	UINT8	RO	22 _h
	1 _h	Bus OFF (Read from CAN controller status byte)	BOOL	RO	0, false
	2 _h	Warning Limit reached (Read from CAN controller status byte)	BOOL	RO	0, false
	3 _h	Rx overflow (Read from CAN controller overrun counter)	BOOL	RO	0, false
	5 _h	Tx overflow (Not served, always false)	BOOL	RO	0, false
	6 _h	Ack error (Not served, always false)	BOOL	RO	0, false
	21 _h	Rx error counter (Read from CAN controller Rx error counter byte)	UINT8	RO	0
	34 _h	Tx error counter (Read from CAN controller Tx error counter byte)	UINT8	RO	0

6.1.5.9

Object F800 CAN Bus Parameter

Index	SubIndex	Name	Data Type	RW	Default
F800 _h	0	Number of sub-indexes	UINT8	RO	
	2 _h	Baud rate	UINT8	RO	02 _h
	5 _h	API Baud rate	UINT32	RO	7FFFFFFF _h

This object contains the baud rate.

Parameter Description

Baud rate CAN bit rate according to table below:

Parameter Baud rate [decimal]	CAN Bit rate [kBaud]
0	1000
1	800
2	500 (Default)
3	250
4	125
5	100
6	50

Table 5: Parameter Baud rate

7 Technical Data

7.1 General Technical Data

Power supply voltage	Nominal voltage: typ. 24V/DC, (min.: 18V, max.: 32V) Current consumption (24V, 20 °C): typ. 150mA	
Connectors	24V CAN IN/OUT	24V-power supply voltage (X1, 4-pin. COMBICON connector, spring-cage connection) CAN Bus interface (X2, 5-pin Phoenix Contact MC 1,5/5-GF-3,81) EtherCAT interface (X3A/B, 2x RJ45 socket)
Temperature range	0 °C ... 50 °C ambient temperature	
Humidity	max. 90%, non-condensing	
Pollution degree	maximum permissible according to DIN EN 61131-2: Pollution Degree 2	
Dimensions	Width: 22.5 mm Height: 114.5 mm Depth: 99 mm (Dimensions without connector protrusion)	
Weight	130 g	

Table 6: General data of the Gateway

7.2 Microprocessor and Memory

CPU	ARM9-Prozessor, 240 MHz, AT91SAM9263
Data Flash	1 MB
NAND Flash	256 MB
SDRAM	32 MB

Table 7 Microprocessor and memory

7.3 CAN Interface

Number of CAN interfaces	1x CAN
CAN controller	integrated in CPU
CAN protocol	according to ISO 11898-1
Physical layer	High-speed CAN interface according to ISO 11898-2, bit rate from 50 kBit/s up to 1 Mbit/s
Electrical isolation	Isolation voltage U: 500 V (= withstand-impulse voltage according to DIN EN 60664-1)
Bus termination	terminating resistor must be set externally, if required
Connector	CAN, 5-pin COMBICON (X2)

Table 8: Data of the CAN interface

7.4 EtherCAT Interface

Number of interfaces	1
Controller	Beckhoff ET1100
Typ	EtherCAT Slave
Bit rate	100BASE-TX, 100 Mbit/s
Connection	Twisted Pair (compatible to IEEE 802.3), 100BASE-TX
Electrical isolation	via transformer
Connector	2x RJ-45-socket with integrated LEDs in the front panel IN (X3B), OUT (X3A)
Distributed Clock	No
Max. number of Gateways	c5xx, Firmware V1.09 10 Gateways c5xx, Firmware V1.10 or higher 16 Gateways

Table 9: Data of the EtherCAT interface

7.5 Operating System and License Information

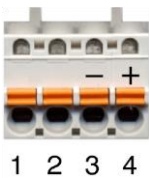
Operating system	QNX 6.5 (Firmware 1.0: QNX 6.4)
Bootloader	U-Boot
License information	This product uses the open source-bootloader "Das U-Boot". The U-Bootsource code is released under the terms of the GNU Public License (GPL). The complete text of the license is contained in Chapter 10. License Terms. Lenze provides the complete bootloader-source code on request. Lenze strives to restore all changes on the bootloader into the official sources. The homepage of the U-Boot project is: http://www.denx.de/wiki/U-Boot .
MD5	md5deep
License information	GPLv2 License The complete text of the license is contained in Chapter 10. License Terms.
Interface	CMSIS - Common Microcontroller Software Interface Standard
License information	BSD License The complete text of the license is contained in Chapter 10. License Terms.
Operating system	FreeRTOS - Free real-time operating system
License information	MIT License The complete text of the license is contained in Chapter 10. License Terms.

8 Interfaces and Connector Assignments

8.1 24V-Power Supply Voltage

The power supply voltage can be fed via connector X1.
Device socket: Phoenix Contact MSTBO 2,5/4-G1L-KMGY
Line connector: Phoenix Contact FKCT 2,5/4-ST, 5.0 mm pitch,
spring cage connection,
Phoenix Contact order No.: 19 21 90 0 (included in the scope of delivery)

Pin Position:



Pin Assignment:

Labelling of the CAN-EtherCAT

Labelling of the Gateway	24V			
	.	.	M	P
Connector label	(free)	(free)	-	+
Pin-Nr.	1	2	3	4
Signal	P24 (+ 24V)	M24 (GND)	M24 (GND)	P24 (+ 24V)

Please refer to the connecting diagram page 8.
The pins 1 and 4 are connected internally.
The pins 2 and 3 are connected internally.

Signal Description:

P24... power supply voltage +24V ± 10 %
M24... reference potential

NOTICE

Feeding through the +24V power supply voltage can cause damage on the modules. It is not permitted to feed through the power supply voltage through the connector X1 and to supply the power supply voltage to another CAN module station!

8.2 CAN

The CAN bus signals are electrically isolated from the other signals via digital isolator and DC/DC converter.

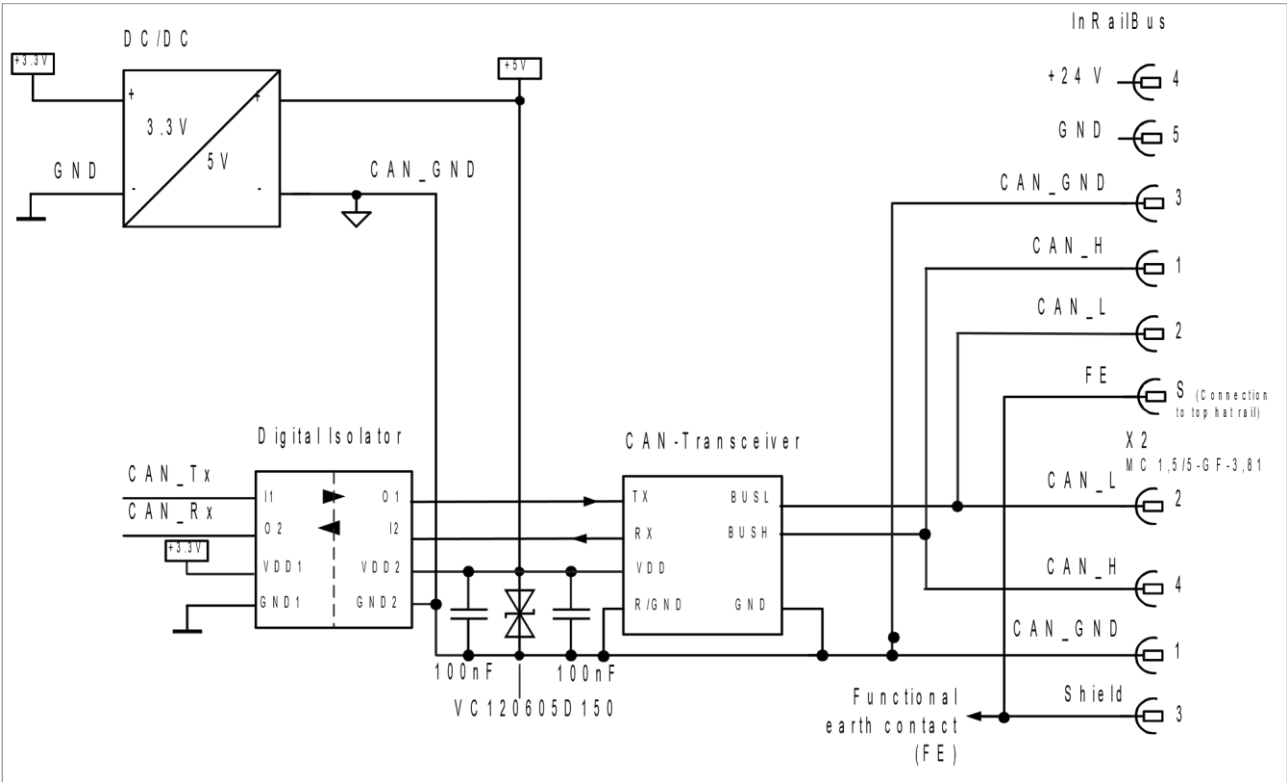


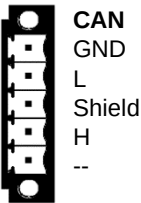
Figure 14: CAN-Interface

The CAN interface can be connected via CAN connector (X2).

8.2.1 CAN Connector

Device connector: Phoenix Contact MC 1,5/5-GF-3,81
Line connector: Phoenix Contact FK-MCP 1,5/5-STF-3,81, spring-cage connection Phoenix Contact Order No.: 1851261 (included in delivery)

Pin Position:
(Device connector view)



Pin Assignment:

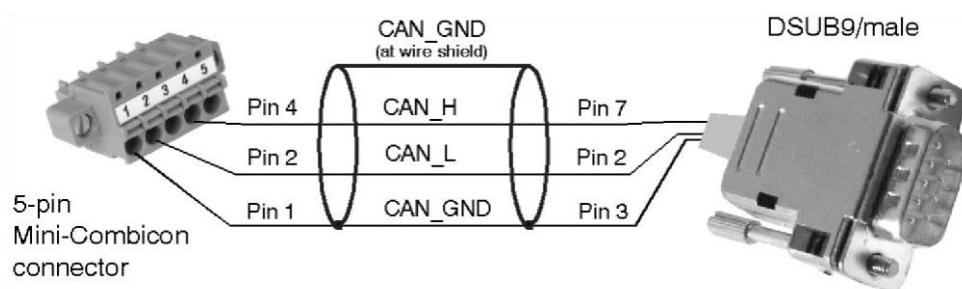
Pin	Signal
1	CAN_GND
2	CAN_L
3	Shield
4	CAN_H
5	-

Signal description:

CAN_L, CAN_H ... CAN signals
CAN_GND ... reference potential of the local CAN physical layer
Shield ... pin for line shield connection (using hat rail mounting direct contact to the mounting rail potential)
- ... not connected

Interfaces and Connector Assignments

Recommendation of an adapter cable from 5-pin COMBICON (here line connector FK-MCP1,5/5-STF_3,81 with spring-cage-connection) to 9-pin D-Sub:



The assignment of the 9pin D-Sub-connector is designed according to CiA DS-102.

The assignment of the 5-pin Mini- COMBICON is designed according to CiA DR-303 Part 1

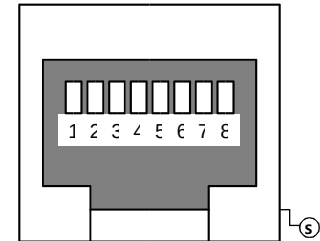
8.3

EtherCAT

Device connector: RJ45 socket, 8-pin, according to IEEE 802.3-2008, Table 25-3 'UTP MDI contact assignment'

The ports have an identical pin assignment.

Pin Position:



Pin Assignment:

Pin	Signal	Meaning
1	MDI0+ (TxD+)	Transmit Data +
2	MDI0- (TxD-)	Transmit Data -
3	MDI1+ (RxD+)	Receive Data +
4	-	-
5	-	-
6	MDI1- (RxD-)	Receive Data -
7	-	-
8	-	-
S	Shield	

Pin 1 to 8 are connected to a line termination.

Signal Description:

MDI0+/-, MDI1+/- ...	EtherCAT data lines
- ...	reserved for future applications, do not connect!
Shield...	line shield connection (using hat rail mounting direct contact to the mounting rail potential)

NOTICE

Permissible cables: To ensure function in networks with up to 100 MBit/s cables of Cat. 5e or better must be used. To ensure the EC Conformity cables with shielding SF/UTP or better must be used.

8.4

Conductor Connection/ Conductor Cross Sections

The following table contains an extract of the technical data of the cable plugs.

Characteristics	Connector Type ¹	
	Power Supply Voltage 24V	CAN Connector
Connector type plug component (Range of articles)	FKCT 2,5/...-ST KMGY	FK-MCP 1,5/5-STF3,81
Connection method	spring-cage connection	screw connection
Stripping length	10 mm	9 mm
Conductor cross section solid min. / max.	0.2 / 2.5 mm ²	0.14 / 1.5 mm ²
Conductor cross section stranded min. / max.	0.2 / 2.5 mm ²	0.14 / 1.5 mm ²
Conductor cross section stranded, with ferrule without plastic sleeve min. / max.	0.25 / 2.5 mm ²	0.25 / 1.5 mm ²
Conductor cross section stranded, with ferrule with plastic sleeve min. / max.	0.25 / 2.5 mm ²	0.25 / 0.5 mm ²
Conductor cross section AWG/kcmil min. / max.	24 / 12	26 / 16
2 conductors with same cross section, without TWIN ferrules with plastic sleeve	not allowed	not allowed
2 conductors with same cross section, stranded, TWIN ferrules with plastic sleeve, min./ max.	0.5 / 1.0 mm ²	not allowed
Minimum AWG according to UL/cUL	26	28
Maximum AWG according to UL/cUL	12	16

¹ Technical Data from Phoenix Contact website, printed circuit board connector, plug component

9 Order Information

Type	Properties	Order No.
CAN-EtherCAT	x750 Gateway EtherCAT/CAN	13626911

Table 10: Order Information

PDF Manuals

Manuals are available in English and in German as well. Available manuals are listed on the Lenze website.

Please download the manuals as PDF documents from our Lenze website www.Lenze.com for free.

10 License Terms

10.1 GPL V1

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, Distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

10.2 GPL V2

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that

users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)
- The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is

License Terms

implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

10.4 MIT Lizenz

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.3 OpenBSD License

Copyright (c) 1990, 1993 The Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT

