Protocol API

Ethernet interface

Packet interface

Hilscher Gesellschaft für Systemautomation mbH

www.hilscher.com

DOC060901API12EN | Revision 12 | English | 2022-08 | Released | Public

Table of contents

1 Introduction

1.1 About this document

This manual describes the application interface of the Ethernet inteface component.

1.1.1 List of revisions

Rev	Date	Name	Revisions
11	2020-02-11	BME	Section *Limitations* added.
			Section *Identifying the Ethernet Packet interface API compliant DPM Communication* Channels added.
			Section *Clear All Multicast Addresses service* added.
12	2022-08-01	HHE, BME	Table 4: Values for “Protocol Class” and “Conformance Class (bitmask)” corrected.
			Clarification regarding Register Application and following indications.
			Add section *Event service*.
			Add hint to technical datasheet in specification section.

Table 1: List of revisions

1.1.2 Intended readers

This manual is intended for software developers who have knowledge of:

- the C programming language
- the Ethernet (specification IEEE 802.3)

This manual is for experts that have knowledge about Ethernet and the structure of Ethernet frames.

1.2 Specifications

The firmware/stack component is based on the Ethernet specification IEEE 802.3.

The following technical data shows principal possibilities, please see related firmware technical datasheet and/or Protocol API Manual for potential differences.

Technical data

Maximum frame length	1518 Bytes, including source and target addresses and Ethertype
Size of receive/transmit queue	4 telegrams each
Data transport layer	Ethernet II, IEEE 802.3
Amount of multicast MAC addresses	32

Features

- Send and receive of Ethernet frames
- Send and receive of Ethernet multicast frames (after activating the specific multicast MAC address)

1.3 Limitations

The following general limitations apply:

- The underlying switch in the netX firmware might apply filtering to the frames to protect the Industrial Ethernet protocol from any disturbance due to frames sent or received by the host application. This means that specific multicast or broadcast frames received by the netX may not be forwarded to the Ethernet application. In addition, specific frames generated by the Ethernet application may not be sent to the network. Typically only protocol specific frames are affected by this issue.
- The size of the receive and transmit queue is limited (see section *Specifications*). If more frames are received by the underlying switch these frames are silently dropped.
- The handling of multicast MAC addresses is implemented protocol-specific in the underlying switch. Conflicts are possible to occur (e.g. unexpected multicast frames are handed over to the Ethernet application).
- This API is not designed to be used by Ethernet application to implement any real-time capable protocol or application. This is due to its design and internal handling in the netX firmware whose main purpose is always executing the Industrial Ethernet protocol.
- Depending on the industrial protocol stack further restrictions or limitations may apply. Please consult the Protocol API Manual of the protocol stack / firmware.

1.4 Terms, abbreviations and definitions

Term	Description
DPM	Dual-Port Memory
NDIS	Network Driver Interface Specification
Industrial Ethernet protocol	The (main) Real-Time Ethernet protocol running on the netX

Table 2: Terms, abbreviations and definitions

1.5 References to documents

This document refers to the following documents:

[1] Hilscher Gesellschaft für Systemautomation mbH: Dual-Port Memory Interface Manual, netX Dual-Port Memory Interface, revision 17, English, 2020.

[2] Hilscher Gesellschaft für Systemautomation mbH: Packet API, netX Dual-Port Memory, Packet-based services (netX 90), revision 6, English, 2021.

[3] Hilscher Gesellschaft für Systemautomation mbH: Operating Instruction Manual, Tag List Editor, Viewing and Editing Tags in NXF/NXI/NXO/BSL Files, V1.5, revision 8, 2020.

Table 3: References

2 Getting started

2.1 Send and receive Ethernet frames

The application can use the Ethernet interface API to

- send and receive Ethernet frames
- send and receive Multicast Ethernet frames

The application can use the Ethernet interface and the Real-Time Ethernet stack simultaneously. Compared to the Industrial Ethernet protocol executed in the netX, the Ethernet interface stack uses its own MAC address. Typically, four MAC addresses are assigned to a netX-based device. The Ethernet interface uses the fourth MAC address. The application is “visible” in the network due to this dedicated MAC address and, as a result, appears as an individual device.

In order to protect the Industrial Ethernet protocol operation, the Ethernet switch may filter certain Ethernet frames sent or received by the application. Details of this filtering depend on the used firmware.

Note: The name also used for the Ethernet interface is NDIS.

Requirements

Requirements for using the Ethernet interface:

1. The Ethernet interface component must be available/integrated in the firmware to be used.
2. The Ethernet interface support must be enabled. For a description, see section *Event service* on page 14.

Header file

Name of the header file: `EthernetIF_API.h`

2.2 Identifying the Ethernet Packet interface API compliant DPM Communication Channels

The identification of the Ethernet Packet Interface using values in the dual-port memory depends on the firmware. Two different identification value sets exists.

- “Ethernet Interface”-based firmware, typically firmware for netX 51, netX 100 and netX 500
- “GCI”-based firmware, typically firmware for netX 90

Due to further development a “Generic Common Interface” (GCI) was designed and implemented for netX 90-based firmware.

Although in both cases the same API is supported, there is the need to distinguish between the two use cases and clearly identify the hardware and software used.

Table 4 lists the identification values and fields used in the dual-port-memory.

Identification values	Identifies
Communication Class: HIL_COMM_CLASS_MESSAGING (0x06) Protocol Class: HIL_PROT_CLASS_ETHERNET (0x24) Conformance Class (bitmask): 0	“Ethernet Interface”-based firmware
Communication Class: HIL_COMM_CLASS_MESSAGING (0x06) Protocol Class: HIL_PROT_CLASS_NETWORK_SERVICES (0x2E) Conformance Class (bitmask): HIL_CONF_CLASS_FLAG_NDIS_AWARE (0x01)	GCI-based firmware

Table 4: Identification values for Ethernet Packet API interface

If the dual-port memory shows a combination of values not described in Table 4, the API and features described in this document cannot safely be applied. Especially early netX 90-based firmware use different identification parameters and is thus not useable for the API documented in this document.

3 The application interface

This chapter describes the application interface of the Ethernet interface component. The application uses the dual-port interface as described in *Dual-Port Memory Interface Manual* (reference [1] and [2]).

3.1 Overview

The following table lists the services of the Ethernet interface (NDIS):

Service	Command code	Page
Register Application	0x00002F10, HIL_REGISTER_APP_REQ, 0x00002F11, HIL_REGISTER_APP_CNF	Documented in reference [2].
Unregister Application	0x00002F12, HIL_UNREGISTER_APP_REQ, 0x00002F13, HIL_UNREGISTER_APP_CNF	Documented in reference [2].
Link Status Change	0x00002F8A, HIL_LINK_STATUS_CHANGE_IND, 0x00002F8B, HIL_LINK_STATUS_CHANGE_RES	Documented in reference [2].
Send Ethernet Frame service	0x00003B22, ETHERNET_IF_SEND_ETH_FRAME_REQ, 0x00003B23, ETHERNET_IF_SEND_ETH_FRAME_CNF	9
Received Ethernet Frame service	0x00003B24, ETHERNET_IF_RECV_ETH_FRAME_IND, 0x00003B25, ETHERNET_IF_RECV_ETH_FRAME_RES	10
Set Multicast Address service	0x00003B26, ETHERNET_IF_SET_MULTICAST_ADDR_REQ, 0x00003B27, ETHERNET_IF_SET_MULTICAST_ADDR_CNF	11
Clear Multicast Address service	0x00003B28, ETHERNET_IF_CLR_MULTICAST_ADDR_REQ, 0x00003B29, ETHERNET_IF_CLR_MULTICAST_ADDR_REQ	12
Clear All Multicast Addresses service	0x00003B2A, ETHERNET_IF_CLR_ALL_MULTICAST_ADDR_REQ, 0x00003B2B, ETHERNET_IF_CLR_ALL_MULTICAST_ADDR_REQ	13
Event service	0x00003B20, ETH_INTF_CMD_EVENT_IND, 0x00003B21, ETH_INTF_CMD_EVENT_RSP	14

Table 5: Ethernet interface – Packet overview

The following topics have to be taken into account when using the Ethernet interface API in Ethernet (NDIS) mode:

The application has use the *Register Application service* (HIL_REGISTER_APP_REQ) in order to receive indications. After the firmware has received the *Register Application Service*, it informs the application about the Link status. Depending on the implementation, either

- a *Link Status Change* indication (GCI-based firmware) is sent, or
- an *Event Indication* (“Ethernet Interface”-based firmware) is sent

in order to synchronize the link status.

3.2 Common services

The Ethernet interface component supports the following common services that are described in reference [2].

- **Register Application service**: Used to register the application.
- **Unregister Application service**: Used to unregister the application.
- **Identify Firmware service**: Used to retrieve identification information.

3.3 Send Ethernet Frame service

The application can use this service to send an Ethernet frame. The minimum length of an Ethernet frame is 60 bytes and the maximum length is 1518 bytes.

3.3.1 Send Ethernet Frame request

Packet description

Variable	Type	Value / Range	Description
ulDest	uint32_t		
ulLen	uint32_t	60 - 1518	Packet data length in bytes
ulCmd	uint32_t	0x00003B22	ETHERNET_IF_SEND_ETH_FRAME_REQ
Data			
abDestMacAddr[6]	uint8_t		Ethernet destination MAC address
abSrcMacAddr[6]	uint8_t		The source MAC address has to be set to the fourth MAC address assigned to the device.
abData[1506]	uint8_t		Data payload of Ethernet frame (starting from Ethertype field) The application has to transfer at least 48 bytes.

Table 6: ETHERNET_IF_SEND_ETH_FRAME_REQ – Send Ethernet Frame request

3.3.2 Send Ethernet Frame confirmation

Packet description

Variable	Type	Value / Range	Description
ulDest	uint32_t		
ulLen	uint32_t	0	Packet data length in bytes
ulSta	uint32_t		See section *Status/error codes* (page 14).
ulCmd	uint32_t	0x00003B23	ETHERNET_IF_SEND_ETH_FRAME_CNF

Table 7: ETHERNET_IF_SEND_ETH_FRAME_CNF – Send Ethernet Frame confirmation

3.4 Received Ethernet Frame service

The Ethernet interface component uses this service to indicate a frame received from Ethernet to application. As a prerequisite the application must register with the Ethernet interface component.

3.4.1 Received Ethernet Frame indication

Packet description

Variable	Type	Value / Range	Description
ulDest	uint32_t		
ulLen	uint32_t	60 - 1518	Packet data length in bytes
ulCmd	uint32_t	0x00003B24	ETHERNET_IF_RECV_ETH_FRAME_IND
Data			
abDestMacAddr[6]	uint8_t		Ethernet destination MAC address
abSrcMacAddr[6]	uint8_t		The source MAC address.
abData[1506]	uint8_t		Data payload of Ethernet frame (starting from Ethertype field)

Table 8: ETHERNET_IF_RECV_ETH_FRAME_IND – Received Ethernet Frame indication

3.4.2 Received Ethernet Frame response

To save computing time, the application need not send a response for this indication.

Packet description

Variable	Type	Value / Range	Description
ulDest	uint32_t		
ulLen	uint32_t	0	Packet data length in bytes
ulSta	uint32_t	0	Packet status ignored by Ethernet interface component. Set to zero for future compatibility.
ulCmd	uint32_t	0x00003B25	ETHERNET_IF_RECV_ETH_FRAME_RSP

Table 9: ETHERNET_IF_RECV_ETH_FRAME_RSP – Received Ethernet Frame response

3.5 Set Multicast Address service

The application has to use this service to configure the reception of Ethernet frames from a specific IPv4 multicast group, i.e. to enable frame reception which targets the given destination multicast MAC address. A netX device can participate in a minimum of 32 multicast groups. Once set, frames targeting the specified destination MAC address are forwarded by means of the *Received Ethernet Frame service* (on page 10).

The multicast filtering uses the non-collision-free hashing, multicast traffic for other group addresses with the same hash value may also pass the filter once a given group address is set via this service.

Depending on the industrial protocol stack used, there may be further restrictions on certain frames not to be forwarded towards the Ethernet interface in order to avoid disturbing realtime data exchange in the Industrial Ethernet protocol.

3.5.1 Set Multicast Address request

Packet description

Variable	Type	Value / Range	Description
ulDest	uint32_t		
ulLen	uint32_t	6	Packet data length in bytes
ulCmd	uint32_t	0x00003B26	`ETHERNET_IF_SET_MULTICAST_ADDR_REQ`
Data			
abMacAddr[6]	uint8_t		Ethernet multicast group address

Table 10: `ETHERNET_IF_SET_MULTICAST_ADDR_REQ` – Set Multicast Address request

3.5.2 Set Multicast Address confirmation

Packet description

Variable	Type	Value / Range	Description
ulDest	uint32_t		
ulLen	uint32_t	0	Packet data length in bytes
ulSta	uint32_t		See section *Status/error codes* (page 14).
ulCmd	uint32_t	0x00003B27	`ETHERNET_IF_SET_MULTICAST_ADDR_CNF`

Table 11: `ETHERNET_IF_SET_MULTICAST_ADDR_CNF` – Set Multicast Address confirmation

3.6 Clear Multicast Address service

The application has to use this service to stop receiving frames from a specific IPv4 multicast group, i.e. to disable frame reception which targets the given destination multicast MAC address. A netX device can participate in a minimum of 32 multicast groups. Once cleared, frames targeting the specified destination MAC address which has previously been set with the Set Multicast Single service are no longer forwarded.

Since the current implementation of multicast filtering may make use of non-collision-free hashing, multicast traffic for other group addresses with the same hash value may also stop passing the filter once a given group address is cleared via this service.

3.6.1 Clear Multicast Address request

Packet description

Variable	Type	Value / Range	Description
ulDest	uint32_t		
ulLen	uint32_t	6	Packet data length in bytes
ulCmd	uint32_t	0x00003B28	ETHERNET_IF_CLR_MULTICAST_ADDR_REQ
Data			
abMacAddr[6]	uint8_t		Ethernet multicast group address

Table 12: ETHERNET_IF_CLR_MULTICAST_ADDR_REQ – Clear Multicast Address request

3.6.2 Clear Multicast Address confirmation

Packet description

Variable	Type	Value / Range	Description
ulDest	uint32_t		
ulLen	uint32_t	0	Packet data length in bytes
ulSta	uint32_t		See section *Status/error codes* (page 14).
ulCmd	uint32_t	0x00003B29	ETHERNET_IF_CLR_MULTICAST_ADDR_CNF

Table 13: ETHERNET_IF_CLR_MULTICAST_ADDR_CNF – Clear Multicast Address confirmation

3.7 Clear All Multicast Addresses service

The application can use this service to stop the receiving of multicast frames. All previously activated Multicast MAC Addresses (via *Set Multicast Address request*) will be cleared at once. For further details about Multicast Addresses, see sections *Set Multicast Address service* (page 11) and *Clear Multicast Address service* (page 12).

Note: This specific service is only available for GCI-based communication channels. For details, see information in section *Identifying the Ethernet Packet interface API compliant DPM Communication Channels* (page 7) to be able to decide whether the service is supported by the used firmware.

3.7.1 Clear Multicast Addresses request

Packet description

Variable	Type	Value / Range	Description
ulDest	uint32_t		
ulLen	uint32_t	0	Packet data length in bytes
ulCmd	uint32_t	0x00003B2A	ETHERNET_IF_CLR_ALL_MULTICAST_ADDR_REQ

Table 14: ETHERNET_IF_CLR_ALL_MULTICAST_ADDR_REQ – Clear All Multicast Addresses request

3.7.2 Clear Multicast Addresses confirmation

Packet description

Variable	Type	Value / Range	Description
ulDest	uint32_t		
ulLen	uint32_t	0	Packet data length in bytes
ulSta	uint32_t		See section *Status/error codes* (page 14).
ulCmd	uint32_t	0x00003B2B	ETHERNET_IF_CLR_ALL_MULTICAST_ADDR_CNF

Table 15: ETHERNET_IF_CLR_ALL_MULTICAST_ADDR_CNF – Clear All Multicast Addresses confirmation

3.8 Event service

The Ethernet protocol task uses the Event service to notify the host application about state changes. As a prerequisite the host application has to register to the Ethernet protocol task. A simple locking mechanism is used to prevent flooding the host application with Event indications. Thus, the host application must send a valid Event response for each received Event indication. If the host application fails in this, the Ethernet protocol task will not sent further Event indications.

Note: This specific service is only available for “Ethernet Interface”-based communication channels. For details, see section *Identifying the Ethernet Packet interface API compliant DPM Communication* Channels (page 7) to be able to decide whether the service is supported by the used firmware.

3.8.1 Event indication

Packet Description

Variable	Type	Value / Range	Description
ulLen	UINT32	4	Packet Data Length in bytes
ulCmd	UINT32	0x00003B20	ETH_INTF_CMD_EVENT_IND
tData			
uiEventCnt[2]	UINT16		Array of Event Counters. Each counter contains the number of corresponding events occurred since the last Event indication had been sent.

Table 16: ETH_INTF_CMD_EVENT_IND - Event indication

Currently the following events are defined:

Constant	Event id	Description
ETH_INTF_EVENT_LINKCHANGED	0	Link Status has changed. Current Link Status can be derived from the Extended Status Block.
ETH_INTF_EVENT_IPCHANGED	1	netX internal TCP/IP stack configuration has changed. Current settings can be derived from the Extended Status Block.

Table 17: Events

3.8.2 Event response

Packet Description

Variable	Type	Value / Range	Description
ulLen	UINT32	0	Packet Data Length in bytes
ulSta	UINT32	0	Packet Status. Ignored by Ethernet protocol task. Set to zero for future compatibility
ulCmd	UINT32	0x00003B21	ETH_INTF_RECV_ETH_FRAME_RES

Table 18: ETH_INTF_CMD_EVENT_RSP – Event response

4 Firmware resource and feature configuration

Loadable firmware supports the feature to configure firmware parameters. During startup of the firmware, it reads the configuration parameters from the tag list of the firmware.

The firmware reads the tag list parameters

- to customize the resource allocation, and
- to configure features.

For viewing and editing the Tag List of a firmware file, the Tag List Editor software is to be used. For a description see reference [3].

Enable or disable the Ethernet interface (NDIS)

By default, the Ethernet interface (NDIS) is disabled in a firmware. To activate the Ethernet interface (NDIS), “Ethernet NDIS Support” has to be set to “enabled” using the Tag List Editor software.

Figure 1 shows the setting for the enabled Ethernet NDIS support.

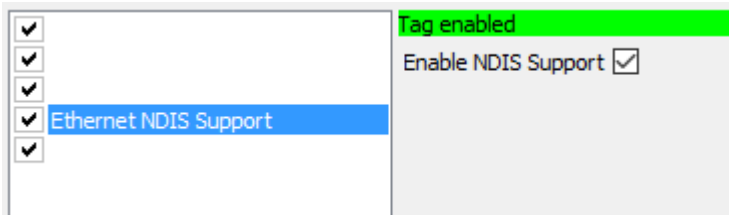

Figure 1: “Ethernet NDIS Support” is “enabled”

Figure 2 shows the setting for the disabled Ethernet NDIS support.

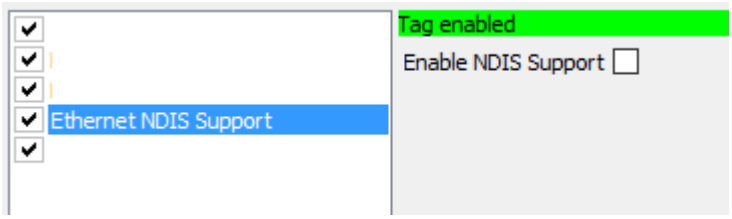

Figure 2: “Ethernet NDIS Support” is “disabled” (default)

5 Status/error codes

5.1 Packet status/error of Ethernet interface

The following status and error codes are used by the Ethernet interface component in the `ulSta` field of confirmation packets:

Hexadecimal value	Definition and description
0x00000000	SUCCESS_HIL_OK Status ok.
0xC05D0001	ERR_ETH_INTF_COMMAND_INVALID Invalid command received.
0xC05D0002	ERR_ETH_INTF_CONFIG_LOCK Configuration is locked.
0xC05D0003	ERR_ETH_INTF_INVALID_PACKET_LENGHT Invalid packet length.
0xC05D0004	ERR_ETH_INTF_INVALID_MODE Invalid mode in request.
0xC05D0005	ERR_ETH_INTF_PARAM_AUTO_NEGOTIATION_PORT_0 Invalid parameter for auto-negotiation port 0.
0xC05D0006	ERR_ETH_INTF_PARAM_AUTO_NEGOTIATION_PORT_1 Invalid parameter for auto-negotiation port 1.
0xC05D0007	ERR_ETH_INTF_PARAM_DUPLEX_MODE_PORT_0 Invalid parameter for duplex mode port 0.
0xC05D0008	ERR_ETH_INTF_PARAM_DUPLEX_MODE_PORT_1 Invalid parameter for duplex mode port 1.
0xC05D0009	ERR_ETH_INTF_PARAM_TRANSMISSION_RATE_PORT_0 Invalid parameter for transmission rate port 0.
0xC05D000A	ERR_ETH_INTF_PARAM_TRANSMISSION_RATE_PORT_1 Invalid parameter for transmission rate port 1.
0xC05D000B	ERR_ETH_INTF_PARAM_AUTO_CROSSOVER_PORT_0 Invalid parameter for auto cross-over port 0.
0xC05D000C	ERR_ETH_INTF_PARAM_AUTO_CROSSOVER_PORT_1 Invalid parameter for auto cross-over port 1.
0xC05D000D	ERR_ETH_INTF_NO_CONFIGURATION Task is not configured.
0xC05D000E	ERR_ETH_INTF_APP_NOT_REGISTERED No application registered.
0xC05D000F	ERR_ETH_INTF_APP_SET_NOT_READY Application set not ready.
0xC05D0010	ERR_ETH_INTF_LINK_DOWN No Ethernet link.
0xC05D0011	ERR_ETH_INTF_GET_SEND_BUFFER Failed to get send buffer.
0xC05D0012	ERR_ETH_INTF_SEND_FRAME Failed to send Ethernet frame.
0xC05D0013	ERR_ETH_INTF_SET_DRV_EDD_CFG Failed to set driver EDD configuration.
0xC05D0014	ERR_ETH_INTF_INVALID_ETH_PORT Invalid parameter for Ethernet port.
0xC05DFFFF	ERR_ETH_INTF_UNKNOWN_ERROR Unknown error detected.

Table 19: Packet status/error of Ethernet interface

6 Appendix

6.1 List of figures

6.2 List of tables

6.3 Legal notes

Copyright

Important notes

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

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

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

Liability disclaimer

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

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

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

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

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

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

Warranty

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

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

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

Costs of support, maintenance, customization and product care

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

Additional guarantees

Although the hardware and software was developed and tested in-depth with greatest care, Hilscher Gesellschaft für Systemautomation mbH shall not assume any guarantee for the suitability thereof for any purpose that was not confirmed in writing. No guarantee can be granted whereby

the hardware and software satisfies your requirements, or the use of the hardware and/or software is uninterruptable or the hardware and/or software is fault-free.

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

Confidentiality

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

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

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

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

Export provisions

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

6.4 Contacts

Headquarters

Germany
Hilscher Gesellschaft für Systemautomation mbH
Rheinstraße 15
D-65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-mail: info@hilscher.com
Support
Phone: +49 (0) 6190 9907-990
E-mail: hotline@hilscher.com

Subsidiaries

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

France
Hilscher France S.a.r.l.
69800 Saint Priest
Phone: +33 (0) 4 72 37 98 40
E-mail: info@hilscher.fr
Support
Phone: +33 (0) 4 72 37 98 40
E-mail: fr.support@hilscher.com

India
Hilscher India Pvt. Ltd.
Pune, Delhi, Mumbai, Bangalore
Phone: +91 8888 750 777
E-mail: info@hilscher.in
Support
Phone: +91 8108884011
E-mail: info@hilscher.in

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

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

Republic of Korea
Hilscher Korea Inc.
13494, Seongnam, Gyeonggi
Phone: +82 (0) 31-739-8361
E-mail: info@hilscher.kr
Support
Phone: +82 (0) 31-739-8363
E-mail: kr.support@hilscher.com

Austria
Hilscher Austria GmbH
4020 Linz
Phone: +43 732 931 675-0
E-mail: sales.at@hilscher.com
Support
Phone: +43 732 931 675-0
E-mail: at.support@hilscher.com

Switzerland
Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-mail: info@hilscher.ch
Support
Phone: +41 (0) 32 623 6633
E-mail: support.swiss@hilscher.com

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