Driver Manual

cifX Device Driver

Windows 2000/XP/Vista/7/8/10

V2.5

Hilscher Gesellschaft für Systemautomation mbH

www.hilscher.com

DOC060701DRV29EN | Revision 29 | English | 2020-05 | Released | Public

Table of contents

1 Introduction

1.1 About this document

This manual describes the **cifX Device Driver** for the Microsoft Windows desktop operating systems Windows® 2000, Windows® XP, Windows® Vista, Windows® 7, Windows® 8 and Windows® 10.

Both versions of the cifX driver are offering the same functionality and also the same application programming interface (API) to access a netX based hardware (e.g. cifX, comX boards and the netX chip).

In general, the drivers are supporting various netX based hardware designs described under *Requirements* for each of the driver.

The API (CIFX API) is designed to give the user an easy access to all of the communication board functionalities. This manual also includes a detailed description of the CIFX API functions.

In addition, Hilscher also offers a free of charge *cifX Toolkit* (C-source code based) which allows to write own drivers based on the Hilscher netX DPM (dual-port memory) definitions including the CIFX API functions. The toolkit is described in a separate manual *cifX/netX Toolkit.*

1.2 List of revisions

Rev	Date	Name	Chapter	Revision
23	2015-12-17	LC		Updated to CIFX Device Driver V1.3
			2	Windows 10 support added.
24	2018-09-20	LC	-	Updated to CIFX Device Driver V1.4
				Windows 10 “Secure Boot” support added.
25	2018-09-28	LC	-	Updated to CIFX Device Driver V1.5
26	2019-08-13	LC		Updated to CIFX Device Driver V2.1
			2.3	Information about cifX 4000 limitations added.
			2.8	Note added about firmware update of CIFX 4000.
27	2019-08-21	HHE		Updated to CIFX Device Driver V2.2
				CIFX M223090 (M.2 format) support added.
			2.3	Information about CIFX M223090 limitations added.
			2.8	Note added about firmware update of CIFX M223090.
28	2019-12-02	LCO		Updated to CIFX Device Driver V2.3
			2.6	Section added about Windows Device Manager and note about CIFX M223090 (multi-function device) added.
29	2020-05-27	LCO		Updated to CIFX Device Driver V2.5
			4	Section *Error codes* updated.

Table 1: List of revisions

1.3 Terms, abbreviations and definitions

Term	Description
cifX	**C**ommunication **I**nter**f**ace based on netX
comX	**Co**mmunication **M**odule based on netX
PCI	**P**eripheral **C**omponent **I**nterconnect
WDM	**W**indows **D**river **M**odel
DLL	**D**ynamic **L**ink **L**ibrary
API	**A**pplication **P**rogramming **I**nterface
SDO	**S**ervice **D**ata **O**bject
PDO	**P**rocess **D**ata **O**bject
DPM	**D**ual-**P**ort **M**emory Physical interface to all communication board (DPM is also used for PROFIBUS-**DP M**aster).

Table 2: Terms, abbreviations and definitions

1.4 References to documents

This document refers to the following documents:

[1] Hilscher Gesellschaft für Systemautomation mbH: Programming reference guide, netX Dual-Port Memory, Revision 2, DOC160904PRG02EN, English, 2019.

[2] Hilscher Gesellschaft für Systemautomation mbH: Dual-Port Memory Interface Manual, netX Dual-Port Memory Interface, Revision 16, DOC060302DPM16EN, English, 2019.

[3] Hilscher Gesellschaft für Systemautomation mbH: Programming reference guide, CIFX API, Revision 8, DOC121201PR08EN, English, 2019.

Table 3: References

2 Windows 2000, XP, Vista, 7, 8 and Windows 10

IMPORTANT NOTE: Windows® is not a deterministic real-time operating system. Any response times to specific hardware or driver functions can not be guaranteed and may differ between different versions of the Windows® operating systems. Furthermore, response times are also depending on the used host hardware, host performance, running services and installed software components.

2.1 Overview

- The cifX Device Driver for the Microsoft desktop operating systems is a kernel mode WDM driver, running in Ring 0 of the operating system. This driver is designed to support the Windows Plug & Play mechanism
- Communication between a user application and the driver is handled by an API DLL. This DLL can be statically or dynamically linked to the application.

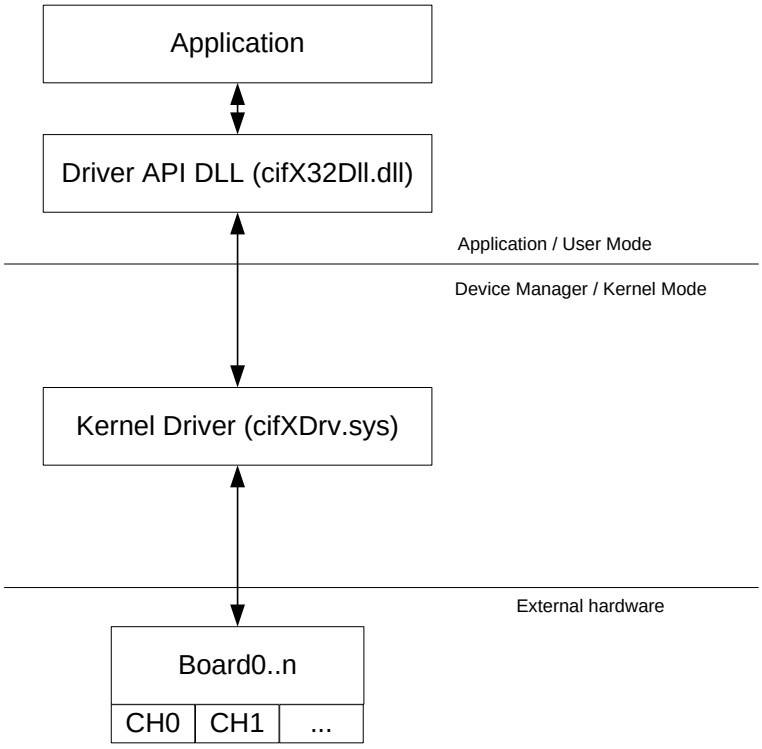


Figure 1: CifX Device Driver - Architecture

2.2 Features

Requirements

- Operating System: Windows 2000, XP, Vista (32/64 bit), Windows 7 (32/64 bit), Windows 8 (32/64 bit), Windows 10 (32/64 bit)
- cifX board or NXSB-PCA / NXSB100 / NXHX board or NX-PCA-PCI / NXHX

Features

- Based on the cifX Toolkit source
- Driver architecture based on the Microsoft KMDF (Kernel Mode Driver Framework) architecture
- Compatible to the Plug&Play mechanism of Windows 2K, XP, Vista, 7, 8 and Windows 10
- x86 and x64 (also known as AMD64) support for Vista, 7, 8 and Windows 10
- Support for PCI, PCIe, PC/104, PCI-104, ISA netX based hardware
- Unlimited number of cifX boards supported
- Support for NXSB-PCA or NX-PCA-PCI boards included (PCI-Adapter to a netX DPM)
- DMA data transfer for I/O data
- Interrupt notification for applications
- Support of second Memory Window for PCI based device (e.g. MRAM)
- Setting the device time during start-up if time handling is supported by the device

2.3 Limitations

- Windows Plug & Play power save mode supported by the driver
  Attention: The actual netX hardware states are not stored and will be lost during power down!
  On system wake-up the hardware is re-started like on system power-on.
- No IA64 support
- Response times of driver calls are operating system and system load depending. Deterministic response times can't be guaranteed
- No DMA (Direct Memory Access) and no MSI (Message-Signaled Interrupts) support for CIFX 4000
- No DMA (Direct Memory Access), no PLC functions and no xDriverMemory support for CIFX M223090 (because of internal SPI communication)

IMPORTANT NOTE: Windows® is not a deterministic real-time operating system. Any response times to specific hardware or driver functions can not be guaranteed and may differ between different versions of the Windows® operating systems. Furthermore, response times are also depending on the used host hardware, host performance, running services and installed software components.

2.4 Windows operating system timing behaviour

IMPORTANT NOTE: Windows® is not a deterministic real-time operating system. Any response times to specific hardware or driver functions can not be guaranteed and may differ between different versions of the Windows® operating systems. Furthermore, response times are also depending on the used host hardware, host performance, running services and installed software components.

Depending on the system layout and system load the processing speed of driver calls are more or less deterministic. Under specific circumstances the Windows operating will re-schedule running processes which could lead to very long function call durations (factor 10 to 100 higher than average time).

Researching this behavior shows a possible re-scheduling during transition of the driver function call from “User-Space” to “Kernel-Space” or during processing the IRP in kernel mode.

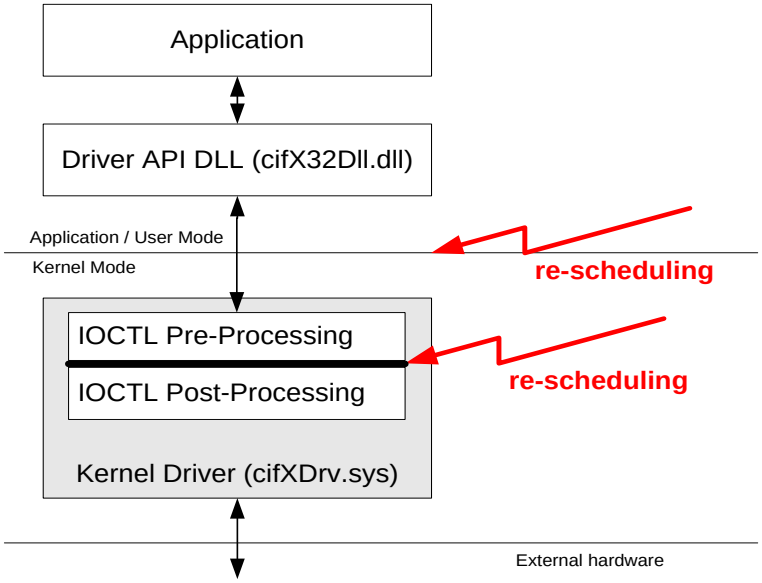


Figure 2: CifX Device Driver – System architecture

At least, re-scheduling could appear at all stages during the call into the driver. A User-Space, important applications is able to increase its process and thread priority to achieve better performance and lower the impact of other running processes.

At driver level, some of the CIFX API functions, usually used during cyclic device handling, are executed directly at pre-processing stage to prevent re-scheduling.

Both measures are helpful in getting more deterministic function call durations, but there is no 100% guarantee of a deterministic program flow.

Note: Specially handled CIFX API functions are marked in function overview tables of chapters 3.1.

Access time measurements

Test System	Windows 7 / 64Bit, Intel Core2Duo E6550 2,33 GHz 1 GB RAM
Process Priority	NORMAL_PRIORITY_CLASS
Thread Priority	THREAD_PRIORITY_TIME_CRITICAL
I/O - Cycles	100000, cycle time 1ms
CIFX Device	CIFX50-DP (PROFIBUS Master V2.3.22.5 / Slave: CB-AB32 (2 Bytes In/Out))

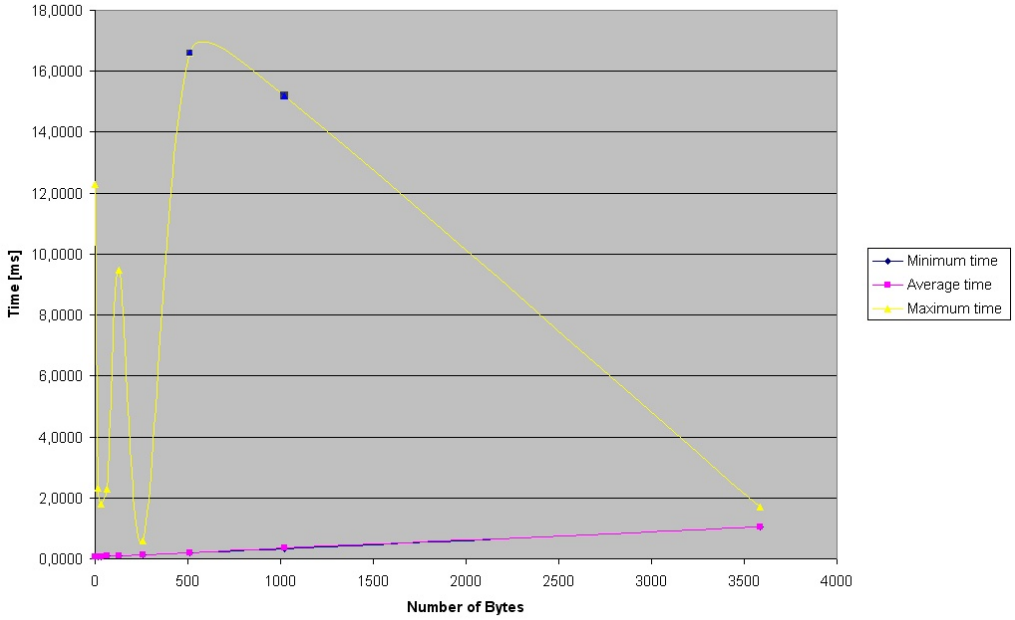


Figure 3: Windows 7 64-bit with standard IOCTL handling

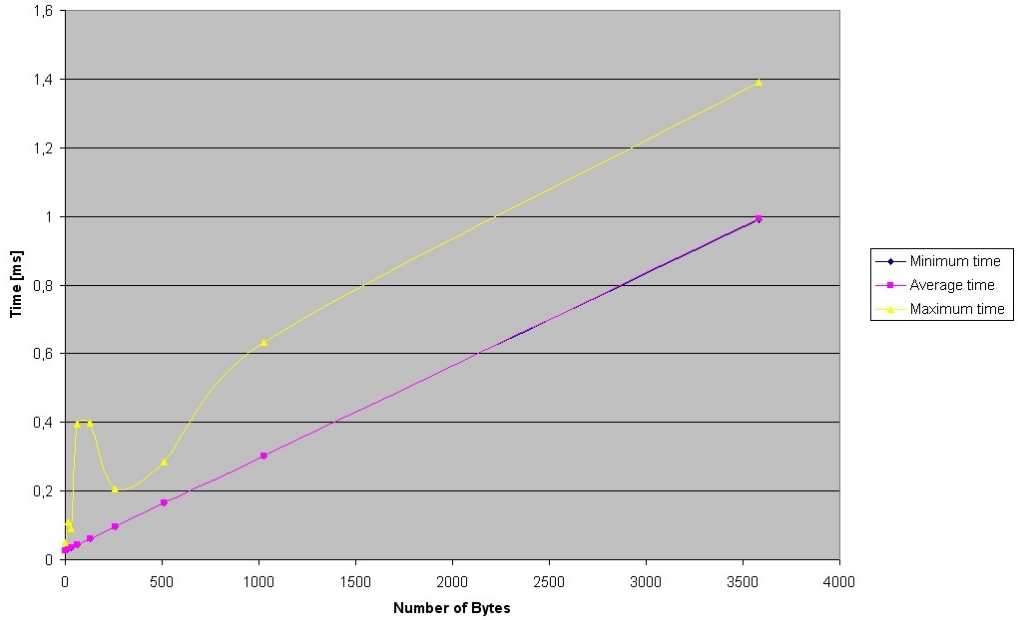


Figure 4: Windows 7 64-bit with direct IOCTL handling

2.5 Installation

The cifX Device Driver is compatible to the Plug and Play functionalities from Windows and offers two types of installation methods.

Note: A detailed step by step installation guide for the CIFX Device Driver for Windows can be found in the '*CIFX Device Driver Installation for Windows OI EN.pdf*' manual

Installation Methods:

- Installation using the driver *Setup.exe* application (**preferred method**)
  The setup application allows a "driver pre-installation" (software first) without hardware and also offers a un-installation.
- Installation using an INF file
  This assumes a connected hardware and does not allow an un-installation of the driver and its components (uninstall under Vista, Win7 and later by Windows device manager)

Both methods are creating several directories on the PC system partition and registry entries to start the driver.

Following steps are processed by the driver setup and INF file:

- Copy necessary driver files to the target system

File name	Description	Destination
cifXDrv.sys	Device driver	.\Windows\System32\drivers
cifX32DLL.dll	Driver API	.\Windows\System32
cifXDrv.cpl	Control applet to start the driver setup or driver test program from the Windows control panel	.\Windows\System32
cifXSetup.exe	Driver setup program	.\Program Files\CIFx Device Driver
cifXTest.exe	Driver test program	.\Program Files\CIFx Device Driver
NETX100-BSL.bin	cifX / netX 100 bootloader	.\Program Files\CIFx Device Driver
NETX50-BSL.bin	netX 50 bootloader	.\Program Files\CIFx Device Driver
x64 only		
cifX32DLL.dll	Compatibility dll for 32 Bit applications running on a 64 Bit Windows	.\Windows\SysWow64

Table 4: cifX Device Driver - Files installed by the INF file

- Creating driver specific registry entries

Destination
HKLM\System\CurrentControlSet\Services\CIFxDrv

Table 5: cifX Device Driver - Registry keys created by the INF file

2.6 cifX Communication Interface in the Device Manager

CIFX devices (PC cards) are located in class 'cifX Communication Interfaces' of the Windows Device Manager.

CIFX M.2 in the Windows Device Manager

Note: CIFX M.2 (e.g. CIFX M223090) is a multi-function device and the Windows Device Manager shows two entries.

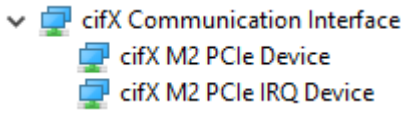
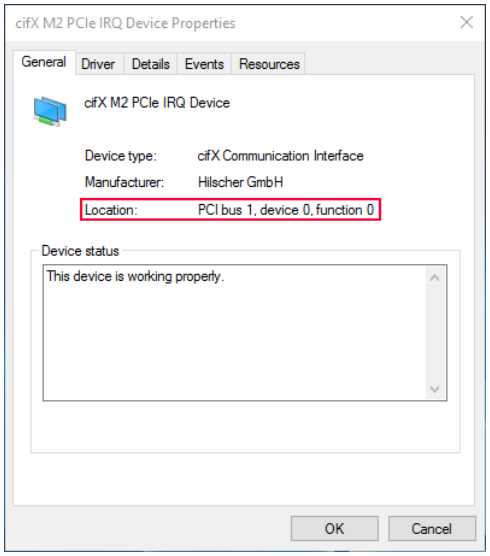
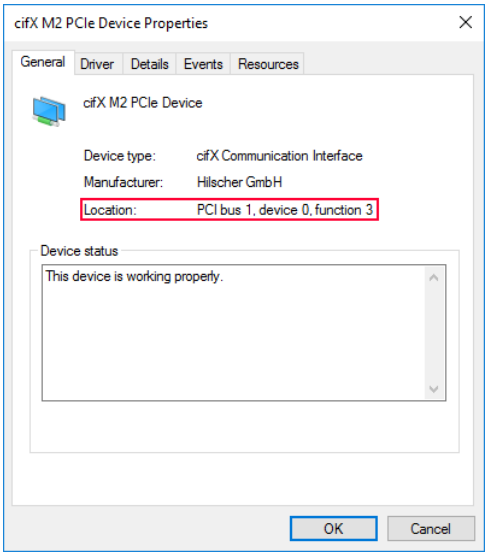


Figure 5: cifX M.2 in the Device Manager

Note: In case, multiple CIFX M.2 devices are installed in the system, the corresponding pair can be determined by checking the PCI location for equal **PCI(e) Bus** and PCI(e) **Device numbers** in device properties dialogue. In the following figures, the two entries showing **PCIbus 1, device 0** belong to one CIFX M.2 device.



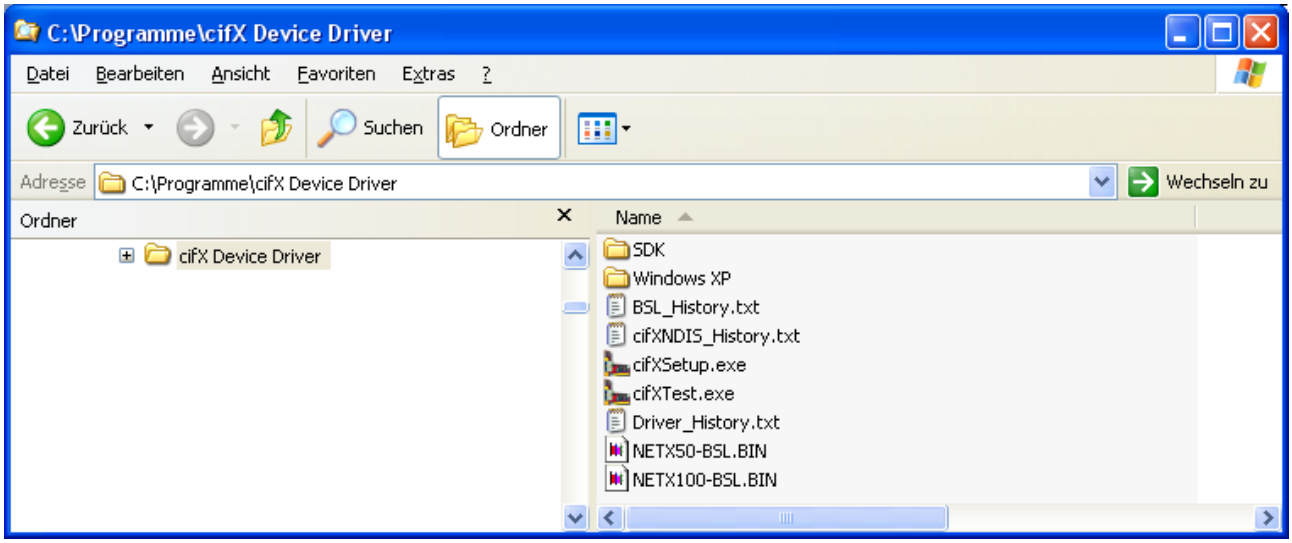


2.7 cifX Windows Driver SDK (Software Development Kit)

The driver installation will also create a SDK directory under the driver installation directory.

This SDK directory contains the C-header files of the drivers user interface and the necessary libraries separated into x86 (32Bit) and x64 (64Bit) versions of the files.

Example driver installation directory:



- SDK directory content

Subdirectory	File name	Description
includes	cifXUser.h	C header containing the CIFX Driver API definition
	cifXErrors.h	C header containing the driver function error number definition
	stdint.h	C header containing ISO C9x data type definition
libs\x86	cifx32dll.lib	DLL library file for 32Bit applications
libs\x64	cifx32dll.lib	DLL library file for 64Bit applications

Table 6: cifX Device Driver - SDK directory content

2.8 Driver Setup and Test Program

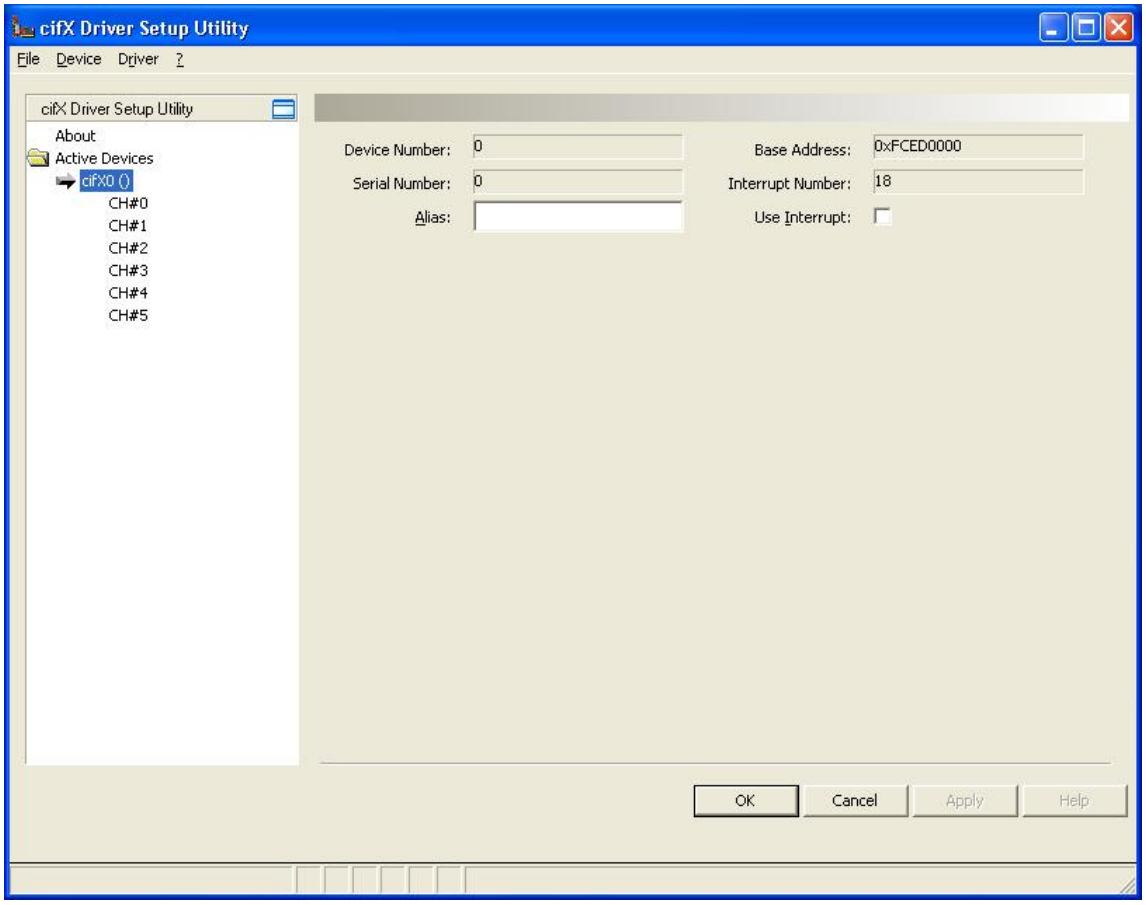
The *cifX Device Driver* for the Microsoft desktop operating systems is delivered wit a separate '*Setup*' and '*Test*' program. Both programs are automatically installed during the driver installation.

Note: A detailed step by step guide on how to setup the CIFX device and how to use the cifX driver setup and test program can be found in the '*CIFX Device Driver Installation for Windows OI EN.pdf*' manual

cifX Driver Setup Utility

The cifX Driver Setup Utility allows you to configure the cifX Device Driver e.g. to set interrupt mode or poll mode. Furthermore, you can use the cifX Driver Setup Utility to specify the firmware file to be loaded into the CIFX device.

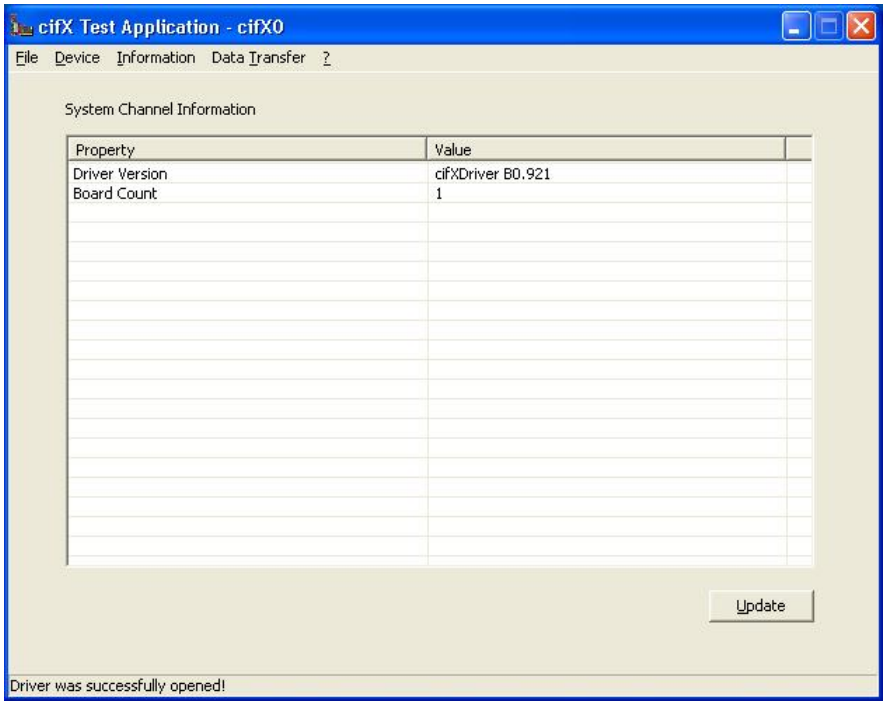
Note: The firmware update handling for a CIFX M223090 and CIFX 4000 differs from previous cifX devices (e.g. CIFX 50).
To update the firmware of a CIFX M223090 or CIFX 4000, use the **Device Explorer** software.



Note: For CIFX M223090 (which is a MSI capable device), a negative interrupt number is displayed.

Driver Test Program

The driver test program can be used to verify the driver functions.



2.9 Device time setting if device supports time handling

The driver will automatically setup the device time if the device signals the time handling feature in the system status information structure (*SYSTEM_CHANNEL_SYSTEM_STATUS_BLOCK / ulHWFeatures*) of the system channel.

Time setting is only executed if the time module of the device signals a not already set device clock. The device clock can be software driven or a physical RTC, which depends on the hardware assembly option and the firmware implementation.

Setting the device time is handled by an asynchronous command sent to the hardware after the firmware is successfully started and the information in the dual port memory signals an available time handling and a not already set time state.

2.10 Timer resolution

The following table lists some operating systems with its standard minimum timer resolution.

Operating system	Timer resolution in ms
DOS	54,95 ms (18.2 ticks per second)
Windows 2000/XP/Vista/7/8/10	10 ms
Windows CE	Platform dependent (default 1ms)

Table 7: Timer resolution

Note: Smaller timer resolution needs deeper knowledge of the operating system.

3 CIFX API (Application Programming Interface)

The CIFX API description can be found in the *CIFX API Application Programming Interface* manual

3.1 Additional information for Windows 7

Windows 7 uses a changed scheduling mechanism which can lead into a delayed processing of driver functions.

Therefore cyclic driver functions are specially processed in the driver to prevent such delays.

The following table lists the CIFX API functions which are specially handled in the driver.

Function	Description	Info
xSysdeviceInfo	Get System device specific information (e.g. mailbox size)	**X**
xSysdeviceGetMBXState	Retrieves the system mailbox state	**X**
xSysdeviceGetPacket	Retrieves a pending packet from the system mailbox	**X**
xSysdevicePutPacket	Send a packet to the system mailbox	**X**
Asynchronous services (Packets)		
xChannelGetMBXState	Retrieve the channels mailbox state	**X**
xChannelGetPacket	Retrieve a pending packet from the channel mailbox	**X**
xChannelPutPacket	Send a packet to the channel's mailbox	**X**
Device administrational/informational functions		
xChannelInfo	Retrieve channel specific information	**X**
xChannelWatchdog	Activate/Deactivate/Trigger Channel Watchdog	**X**
xChannelCommonStatusBlock	Access to the common status block	**X**
xChannelExtendedStatusBlock	Access to the extended status block	**X**
xChannelUserBlock	Access user block (not implemented yet!)	
Cyclic data services (I/O's)		
xChannelIORead	Instructs the device to place the latest data into the DPM and passes them to the user	**X**
xChannelIOWrite	Copies the data to the DPM and waits for the firmware to retrieve them	**X**
xChannelIOReadSendData	Reads back the last send data	**X**
Cyclic data services (I/O's, PLC optimized)		
xChannelPLCActivateRead	Instruct the firmware to place the latest input data into the dual port (no wait for completion)	**X**
xChannelPLCActivateWrite	Instruct the firmware to retrieve the latest output data from the dual port (no wait for completion)	**X**
xChannelPLCIsReadReady	Checks if the last Read Activation has finished	**X**
xChannelPLCIsWriteReady	Checks if the last Write Activation has finished	**X**
Bus synchronous operation		
xChannelSyncState	Wait for synchronization event or trigger/acknowledge sync	**X**

Table 8: Additional information for Windows 7

(X) Marked functions are handled with higher priority under Windows Vista/7 and later.

4 Error codes

Error code	Symbol / Description
0x00000000	CIFX_NO_ERROR No error

General error codes (0x800Axxxx)

Error code	Symbol / Description
0x800A0001	CIFX_INVALID_POINTER Invalid pointer (e.g. NULL was passed to the function)
0x800A0002	CIFX_INVALID_BOARD No board with the given name / index available
0x800A0003	CIFX_INVALID_CHANNEL No channel with the given index available
0x800A0004	CIFX_INVALID_HANDLE An invalid handle was passed to the function
0x800A0005	CIFX_INVALID_PARAMETER Invalid parameter passed to the function
0x800A0006	CIFX_INVALID_COMMAND Command parameter is invalid
0x800A0007	CIFX_INVALID_BUFFERSIZE The supplied buffer does not match the expected size
0x800A0008	CIFX_INVALID_ACCESS_SIZE Invalid access size (e.g. I/O area size is exceeded by given offset and length)
0x800A0009	CIFX_FUNCTION_FAILED Generic function failure
0x800A000A	CIFX_FILE_OPEN_FAILED A file cannot not be opened
0x800A000B	CIFX_FILE_SIZE_ZERO File size is zero
0x800A000C	CIFX_FILE_LOAD_INSUFF_MEM Insufficient memory to load file a file to RAM
0x800A000D	CIFX_FILE_CHECKSUM_ERROR File checksum comparison failed
0x800A000E	CIFX_FILE_READ_ERROR Error while reading file
0x800A000F	CIFX_FILE_TYPE_INVALID The given file is invalid for the operation
0x800A0010	CIFX_FILE_NAME_INVALID Invalid filename given
0x800A0011	CIFX_FUNCTION_NOT_AVAILABLE Function is not available on the driver
0x800A0012	CIFX_BUFFER_TOO_SHORT The passed buffer is too short to receive all of the requested data

Error code	Symbol / Description
0x800A0013	CIFX_MEMORY_MAPPING_FAILED Error mapping the dual port memory for later memory access
0x800A0014	CIFX_NO_MORE_ENTRIES No more entries available. Returned by enumeration functions (e.g. xDriverEnumBoards(), directories etc.)
0x800A0015	CIFX_CALLBACK_MODE_UNKNOWN Unknown callback handling mode
0x800A0016	CIFX_CALLBACK_CREATE_EVENT_FAILED Creation of callback events failed
0x800A0017	CIFX_CALLBACK_CREATE_RECV_BUFFER Creation of callback receive buffer failed
0x800A0018	CIFX_CALLBACK_ALREADY_USED Another application has already registered a callback for the given event
0x800A0019	CIFX_CALLBACK_NOT_REGISTERED A callback was not registered before
0x800A001A	CIFX_INTERRUPT_DISABLED Device interrupt is disabled. The executed function expects an enabled hardware interrupt (depending on the driver this must be done either by the device configuration or driver setup program).

Table 9: General Error Codes (0x800Axxxx)

Driver-related error codes (0x800Bxxxx)

Error code	Symbol / Description
0x800B0001	CIFX_DRV_NOT_INITIALIZED Driver was not correctly initialized during startup or driver is already closed
0x800B0002	CIFX_DRV_INIT_STATE_ERROR Initialization state error. Hardware does not show correct or expected states and information in the DPM after a reset or boot start
0x800B0003	CIFX_DRV_READ_STATE_ERROR Driver read state error
0x800B0004	CIFX_DRV_CMD_ACTIVE The called function is in use by another program instance or application
0x800B0005	CIFX_DRV_DOWNLOAD_FAILED General error during download (e.g. boot loader could not be downloaded or started)
0x800B0006	CIFX_DRV_WRONG_DRIVER_VERSION Wrong driver version
0x800B0030	CIFX_DRV_DRIVER_NOT_LOADED CIFX driver is not loaded / running. Failed to open or start the driver, returned by xDriverOpen()
0x800B0031	CIFX_DRV_INIT_ERROR Failed to initialize the driver
0x800B0032	CIFX_DRV_CHANNEL_NOT_INITIALIZED Channel not initialized (e.g. xChannelOpen() not called)
0x800B0033	CIFX_DRV_IO_CONTROL_FAILED Function call into the driver failed (e.g. used by the Windows API DLL to signal problems in IO-Control driver calls)
0x800B0034	CIFX_DRV_NOT_OPENED Driver was not opened by calling xDriverOpen()
0x800B0040	CIFX_DRV_DOWNLOAD_STORAGE_UNKNOWN Unknown download storage type (RAM/FLASH based) found
0x800B0041	CIFX_DRV_DOWNLOAD_FW_WRONG_CHANNEL Channel number for a firmware download not supported
0x800B0042	CIFX_DRV_DOWNLOAD_MODULE_NO_BASEOS Modules are not allowed without a Base OS firmware

Table 10: Driver-related error codes (0x800Bxxxx)

Device / Communication-related error codes (0x800Cxxxx)

Error code	Symbol / Description
0x800C0010	CIFX_DEV_DPM_ACCESS_ERROR Dual port memory not accessible (e.g. board not found, wrong dual port memory content)
0x800C0011	CIFX_DEV_NOT_READY Device is not ready (NSF_READY or NCF_READY flag is not set) The system device or communication channel is not working
0x800C0012	CIFX_DEV_NOT_RUNNING Device is not running (NCF_RUNNING flag is not set). The communication channel is not configured
0x800C0013	CIFX_DEV_WATCHDOG_FAILED Watchdog test failed
0x800C0015	CIFX_DEV_SYSERR Error in handshake flags
0x800C0016	CIFX_DEV_MAILBOX_FULL Send mailbox is full. The PutPacket() function was not able to write a packet to the device mailbox. Either the mailbox state does not show empty or no more resources on the device available. (NSF_SEND_MBX_ACK / HSF_SEND_MBX_CMD or NCF_SEND_MBX_ACK / HCF_SEND_MBX_CMD flags in wrong state or mailbox counter usPackagesAccepted = 0)
0x800C0017	CIFX_DEV_PUT_TIMEOUT Send packet timeout. The PutPacket() function was not able to write a packet to the device mailbox and the wait time in PutPacket() has expired. Either the mailbox state does not show empty or no more resources on the device available. (NSF_SEND_MBX_ACK / HSF_SEND_MBX_CMD or NCF_SEND_MBX_ACK / HCF_SEND_MBX_CMD flags in wrong state or mailbox counter usPackagesAccepted = 0)
0x800C0018	CIFX_DEV_GET_TIMEOUT Receive packet timeout. GetPacket() function was not able to read a packet from the device and the wait time in GetPacket() has expired. Either the mailbox state does not show a packet available or the device has not sent a packet. (NSF_RECV_MBX_CMD / HSF_RECV_MBX_ACK or NCF_RECV_MBX_CMD / HCF_RECV_MBX_ACK flags in wrong state or mailbox counter usWaitingPackages = 0)
0x800C0019	CIFX_DEV_GET_NO_PACKET No packet available. The GetPacket() function was called with timeout = 0 and the function was not able to read a packet from the device. Either the mailbox state does not show a packet available or the device has not sent a packet. (NSF_RECV_MBX_CMD / HSF_RECV_MBX_ACK or NCF_RECV_MBX_CMD / HCF_RECV_MBX_ACK flags in wrong state or mailbox counter usWaitingPackages = 0)
0x800C001A	CIFX_DEV_MAILBOX_TOO_SHORT Mailbox is to short for the given packet. The packet send by PutPacket() does not fit into the mailbox.
0x800C0020	CIFX_DEV_RESET_TIMEOUT Reset command timeout. The device was not reaching READY state, in the given reset timeout, after the application has initiated a reset (RCX_COMM_COS_READY flag not set).
0x800C0021	CIFX_DEV_NO_COM_FLAG Communication flag not set. The fieldbus protocol stack has no communication to the fieldbus devices. Either the cable is disconnected or no other device is connected to the wire (NCF_COMMUNICATING flag not set).

Error code	Symbol / Description
0x800C0022	CIFX_DEV_EXCHANGE_FAILED I/O data exchange failed. Function xChannelIORead() or xChannelIOWrite() fails, because the device does not allow to access the I/O data image. (NCF_PDIN / NCF_PDOUT flags are not in the state allowing access to the I/O process data image)
0x800C0023	CIFX_DEV_EXCHANGE_TIMEOUT I/O data exchange timeout. The given timeout in xChannelIORead() / xChannelIOWrite() expires while the function is waiting to get access to the process data image. (NCF_PDIN / NCF_PDOUT flags are not in the state allowing access to the I/O process data image)
0x800C0024	CIFX_DEV_COM_MODE_UNKNOWN Unknown I/O data exchange mode (mode is not within 0..5)
0x800C0025	CIFX_DEV_FUNCTION_FAILED Device function failed
0x800C0026	CIFX_DEV_DPMSIZE_MISMATCH DPM size differs from configuration, The firmware signals a communication channel size which does not fit into the maximum DPM size defined by the hardware or defined by the user.
0x800C0027	CIFX_DEV_STATE_MODE_UNKNOWN Unknown state mode
0x800C0028	CIFX_DEV_HW_PORT_IS_USED Device is accessed either by another application or another instance. - Driver / device can't be unloaded, open connection to the system device or a communication channels still active - xChannelOpen() can't be executed because it is currently used by another application
0x800C0029	CIFX_DEV_CONFIG_LOCK_TIMEOUT Failed lock the communication channels configuration within the given time. xChannelConfigLock() wait time expired (RCX_COMM_COS_CONFIG_LOCKED flag not set).
0x800C002A	CIFX_DEV_CONFIG_UNLOCK_TIMEOUT Failed to unlock the communication channel configuration within the given time. xChannelConfigLock() wait time expired (RCX_COMM_COS_CONFIG_LOCKED flag not cleared)
0x800C002B	CIFX_DEV_HOST_STATE_SET_TIMEOUT Wait time expires during xChannelHostState() without reaching CIFX_HOST_STATE_READY. (The function was not able to set the RCX_APP_COS_APP_READY flag or the device has not acknowledged the new status in time)
0x800C002C	CIFX_DEV_HOST_STATE_CLEAR_TIMEOUT Wait time expires during xChannelHostState() without reaching CIFX_HOST_STATE_NOT_READY (The function was not able to clear the RCX_APP_COS_APP_READY flag or the device has not acknowledged the new status in time)
0x800C002D	CIFX_DEV_INITIALIZATION_TIMEOUT Timeout during device / channel initialization
0x800C002E	CIFX_DEV_BUS_STATE_ON_TIMEOUT Wait time expires during xChannelBusState() without reaching CIFX_BUS_STATE_ON (RCX_COMM_COS_BUS_ON flag not set) Using a timeout, the function will activate fieldbus communication and waits until communication to another fieldbus device is available (NCF_COMMUNICATION flag is set)

Error code	Symbol / Description
0x800C002F	CIFX_DEV_BUS_STATE_OFF_TIMEOUT Wait time expires during xChannelBusState() without reaching CIFX_BUS_STATE_OFF. (The function was not able to clear the RCX_APP_COS_BUS_ON flag or the device has not acknowledged the new status in time and still signals bus communication is active by RCX_COM_COS_BUS_ON).
0x800C0040	CIFX_DEV_MODULE_ALREADY_RUNNING Firmware module (NXO) download and start failed because a module is already running
0x800C0041	CIFX_DEV_MODULE_ALREADY_EXISTS Firmware module (NXO) download was skipped because the module already exists
0x800C0050	CIFX_DEV_DMA_INSUFF_BUFFER_COUNT Number of configured DMA buffers insufficient (at least 8 buffers are expected) Or xChannelDMAState() is used without previously configured DMA buffers.
0x800C0051	CIFX_DEV_DMA_BUFFER_TOO_SMALL DMA buffers size too small (min. size 256 Byte)
0x800C0052	CIFX_DEV_DMA_BUFFER_TOO_BIG DMA buffers size too big (max. size 63,75 KByte)
0x800C0053	CIFX_DEV_DMA_BUFFER_NOT_ALIGNED DMA buffer alignment failed (must be 256 Byte)
0x800C0054	CIFX_DEV_DMA_HANSHAKEMODE_NOT_SUPPORTED I/O process data exchange mode "uncontrolled" not allowed when DMA transfer is activated
0x800C0055	CIFX_DEV_DMA_IO_AREA_NOT_SUPPORTED I/O process data area index in DMA mode not supported (only area 0 possible)
0x800C0056	CIFX_DEV_DMA_STATE_ON_TIMEOUT Failed to set DMA transfer to "ON" within the given wait time in xChannelDMAState(). (The device has not acknowledged the new status or not set the RCX_COM_COS_DMA flag)
0x800C0057	CIFX_DEV_DMA_STATE_OFF_TIMEOUT Failed to set DMA transfer to "OFF" within the given wait time in xChannelDMAState(). (The device has not acknowledged the new status or not cleared the RCX_COM_COS_DMA flag)
0x800C0058	CIFX_DEV_SYNC_STATE_INVALID_MODE Device is in invalid mode for the command initiated by xChannelSyncState(). The mode must be either "SYNC Host Controlled" (RCX_SYNC_MODE_HST_CTRL) or "SYNC Device Controlled" (RCX_SYNC_MODE_DEV_CTRL)
0x800C0059	CIFX_DEV_SYNC_STATE_TIMEOUT Wait time expired during xChannelSyncState(..,CIFX_SYNC_WAIT_CMD,). Device does not signal the expected synchronization handshake flag state

Table 11: Device / Communication-related error codes (0x800Cxxxx)

5 Appendix

5.1 List of tables

5.2 List of figures

5.3 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.

5.4 Contacts

Headquarters

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

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

Subsidiaries

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

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

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

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

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

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

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

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

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

Korea
Hilscher Korea Inc.
Seongnam, Gyeonggi, 463-400
Phone: +82 (0) 31-789-3715
E-Mail: info@hilscher.kr

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

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

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

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