



Driver Manual
cifX Device Driver
Windows 2000/XP/Vista/7/8/10
V1.3

Hilscher Gesellschaft für Systemautomation mbH

www.hilscher.com

DOC060701DRV23EN | Revision 23 | English | 2016-02 | Released | Public

Table of contents

1	Introduction.....	3
1.1	About this document	3
1.2	List of revisions.....	4
1.3	Terms, abbreviations and definitions	5
1.4	References	5
1.5	Legal Notes	6
	1.5.1 Copyright.....	6
	1.5.2 Important notes	6
	1.5.3 Exclusion of liability	7
	1.5.4 Export	7
2	Windows 2000, XP, Vista, 7, 8 and Windows 10.....	8
2.1	Overview	8
2.2	Features	9
2.3	Limitations	9
2.4	Windows operating system timing behaviour.....	10
2.5	Installation	12
2.6	CIFX Windows Driver SDK (Software Development Kit)	13
2.7	Driver Setup and Test Program	14
2.8	Device time setting if device supports time handling	16
2.9	Timer resolution.....	16
3	CIFX API (Application Programming Interface).....	17
3.1	Additional information for Windows 7	17
4	Error codes	18
5	Appendix	21
5.1	List of tables	21
5.2	List of figures	21
5.3	Contacts	22

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
19	2011-09-21	RM	5.6.6 5.7.5	xDriverEnumChannel parameter description fixed Updated to CIFX Device Driver V1.1.x.x - xSysdeviceBootstart() function added
20	2011-10-19	RM	2.4 5.3, 5.4	New chapter <i>Windows operating system timing behaviour</i> added - Added information about Windows driver time issues - Added diagrams with access times Added information about function with performance improvements
21	2011-12-12	RM	5.3 5.7.13 2.8	Updated to CIFX Device Driver V1.1.1.x Added new API function xSysdeviceExtendedMemory() Extended SYSTEM_CHANNEL_SYSTEM_STATUS_BLOCK by ulHWFeatures Added information about device time setting during start-up
22	2013-03-12	RM	5.7.7 2.6	Updated to CIFX Device Driver V1.2.x.x Updated description to software reset - Information about driver SDK (Software Development Kit) added - Windows 8 support added - API documentation moved to separate manual (CIFX API Application Programming Interface manual) - Added information about the ' <i>CIFX Device Driver Installation for Windows</i> ' manual
23	2015-12-17	LC	2	Updated to CIFX Device Driver V1.3x.x Windows 10 support added.

Table 1: List of revisions

1.3 Terms, abbreviations and definitions

Term	Description
cifX	Communication Interface based on netX
comX	Communication Module based on netX
PCI	Peripheral Component Interconnect
WDM	Windows Driver Model
DLL	Dynamic Link Library
API	Application Programming Interface
SDO	Service Data Object
PDO	Process Data Object
DPM	Dual-Port Memory Physical interface to all communication board (DPM is also used for PROFIBUS-DP Master).

Table 2: Terms, abbreviations and definitions

1.4 References

This document refers to the following documents:

- [1] Real-time Communication System for netX
- [2] netX Bootstrap Specification
- [3] netX Program Reference Guide
- [4] netX DPM Interface Manual
- [5] CIFX API Application Programming Interface

Table 3: References

1.5 Legal Notes

1.5.1 Copyright

©2006-2016 Hilscher Gesellschaft für Systemautomation mbH

All rights reserved.

The images, photographs and texts in the accompanying material (manual, accompanying texts, documentation, etc.) are protected by German and international copyright law as well as international trade and protection provisions. You are not authorized to duplicate these in whole or in part using technical or mechanical methods (printing, photocopying or other methods), to manipulate or transfer using electronic systems without prior written consent. You are not permitted to make changes to copyright notices, markings, trademarks or ownership declarations. The included diagrams do not take the patent situation into account. The company names and product descriptions included in this document may be trademarks or brands of the respective owners and may be trademarked or patented. Any form of further use requires the explicit consent of the respective rights owner.

1.5.2 Important notes

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

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

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

1.5.3 Exclusion of liability

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

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

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

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

1.5.4 Export

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

2 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.

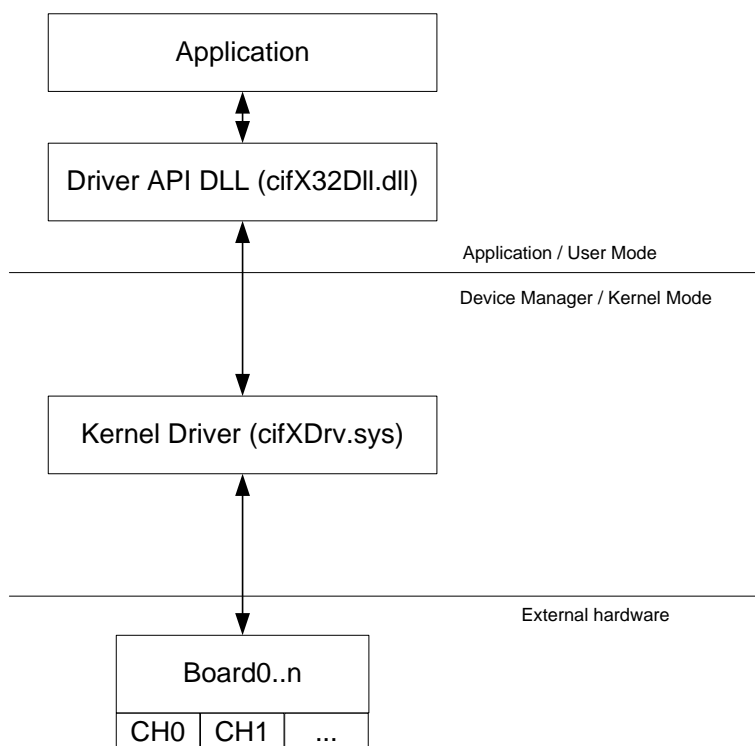


Figure 1: CifX Device Driver - Architecture

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

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.

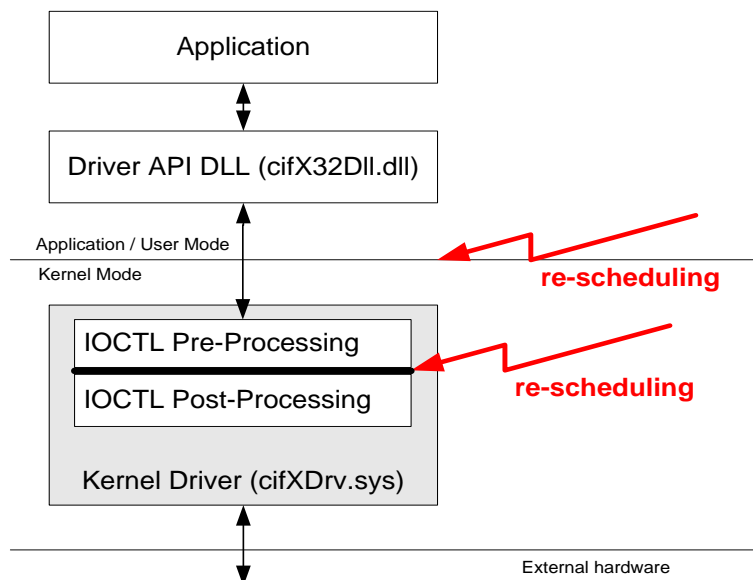


Figure 2: CifX Device Driver – System architecture

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))

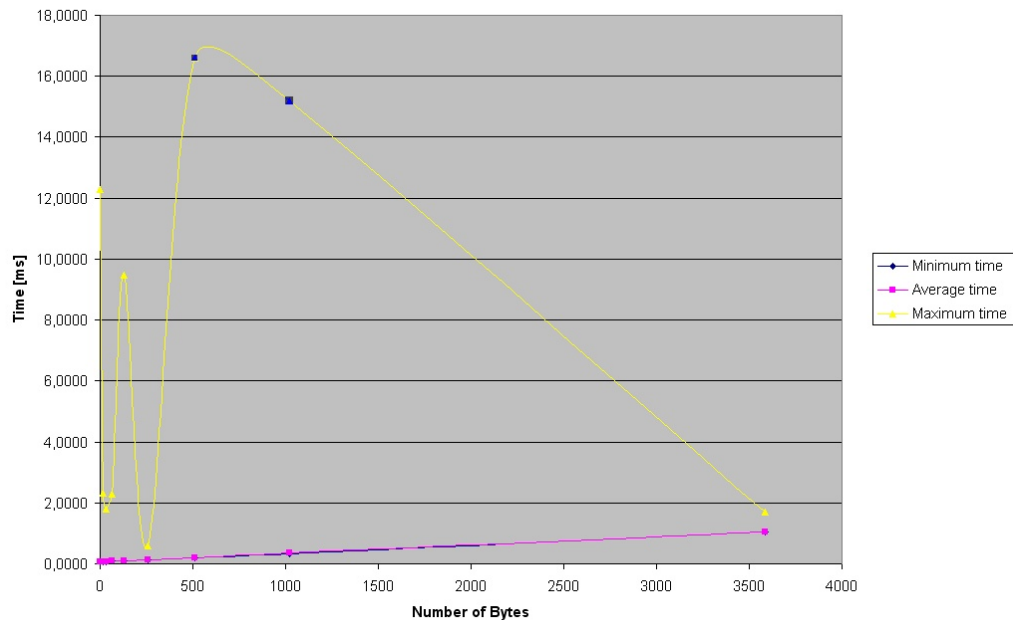


Figure 3: Windows 7 64-bit with standard IOCTL handling

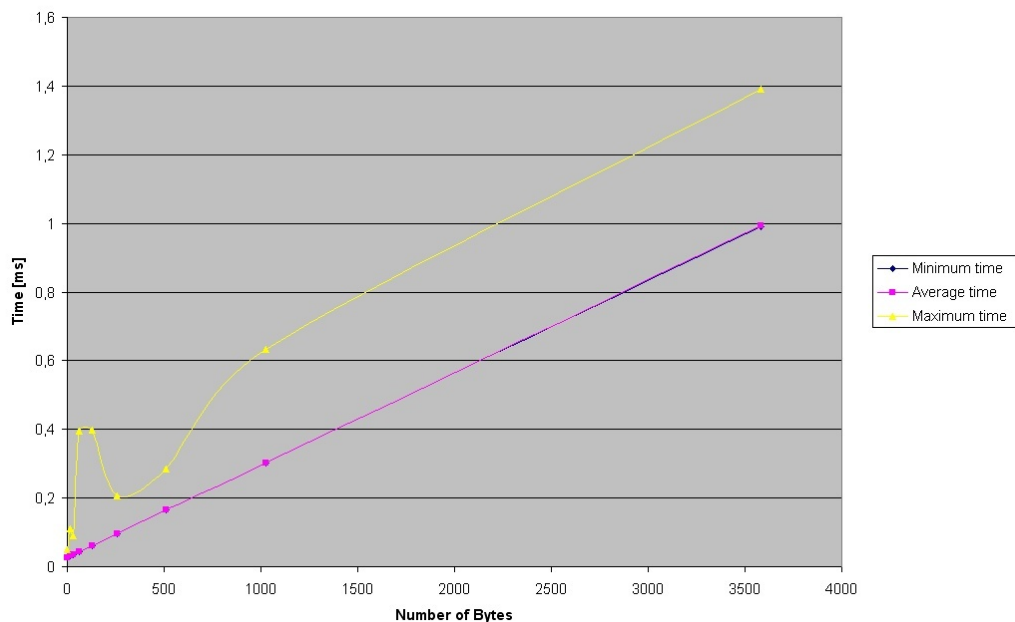


Figure 4: Windows 7 64-bit with direct IOCTL handling

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 nor 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

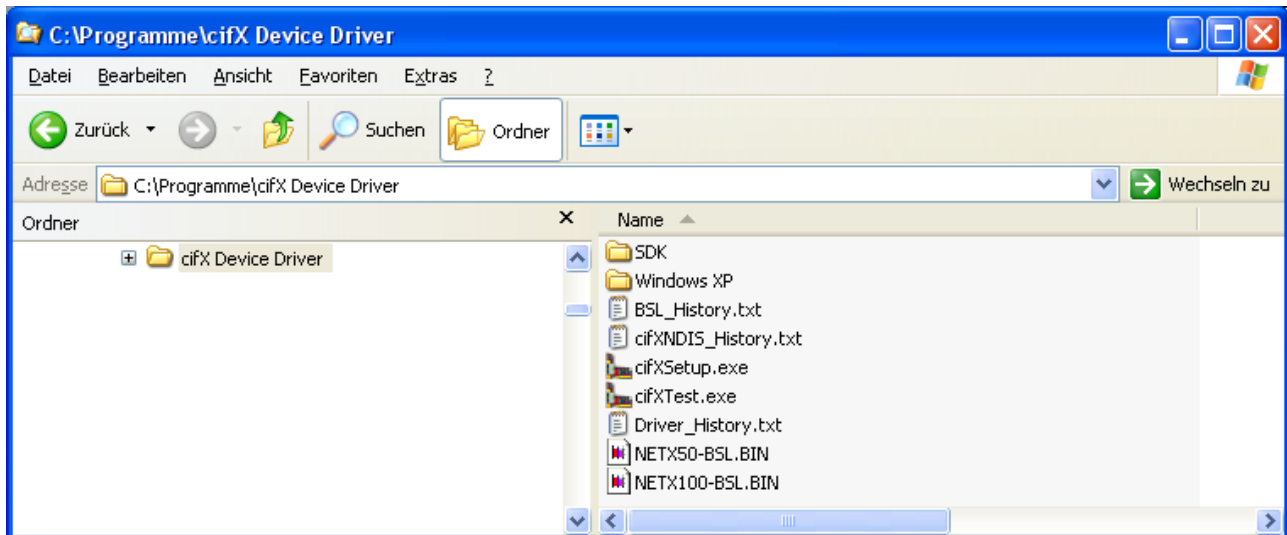
Table 5: cifX Device Driver - Registry keys created by the INF file

2.6 CFX 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 CFX 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

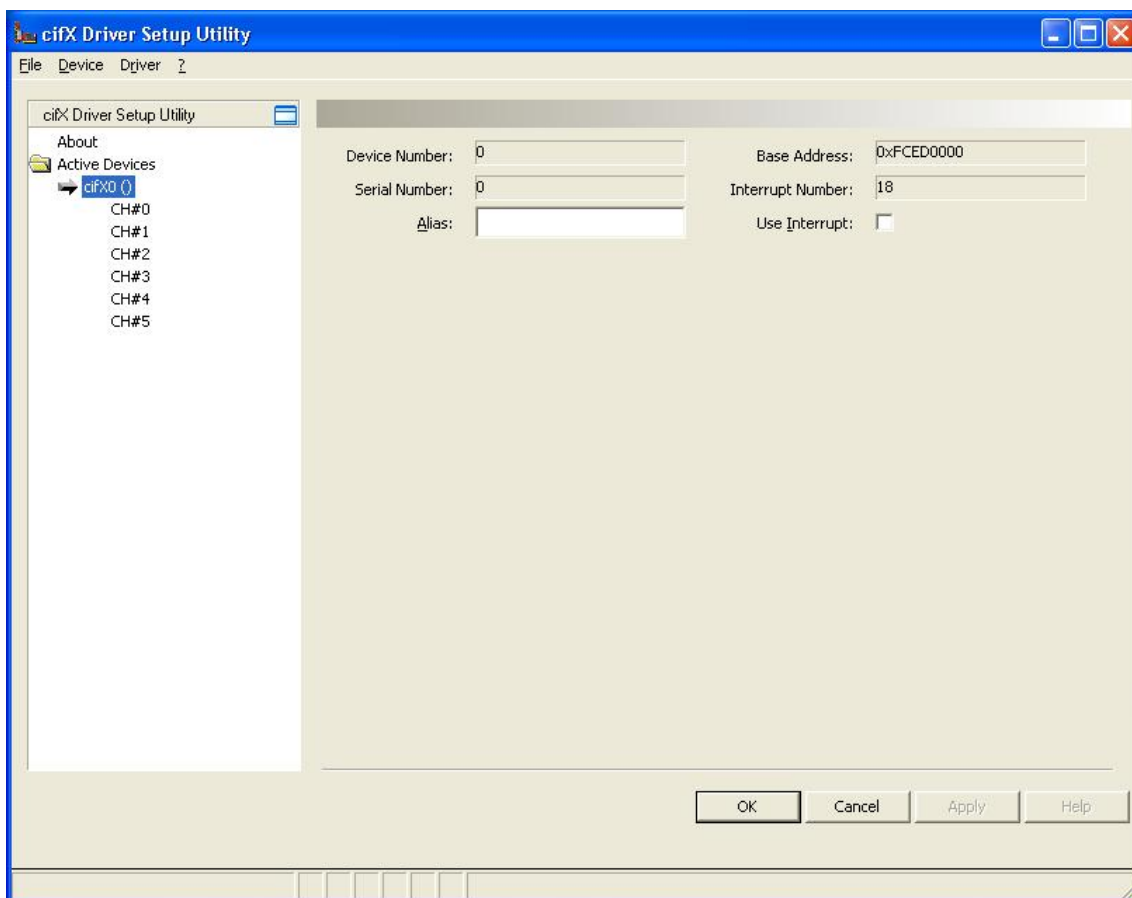
Table 6: cifX Device Driver - SDK directory content

2.7 Driver Setup and Test Program

The *cifX Device Driver* for the Microsoft desktop operating systems is delivered with 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 hardware 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

Driver Setup Program



2.8 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.

2.9 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 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 administrative/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

Value	Symbol	Description
0x00000000	CIFX_NO_ERROR	No error
0x800Axxxx		
0x800A0001	CIFX_INVALID_POINTER	An 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 is available
0x800A0004	CIFX_INVALID_HANDLE	An invalid handle was passed to the function
0x800A0005	CIFX_INVALID_PARAMETER	Invalid parameter passed to 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. IO Area is exceeded by Offset and size)
0x800A0009	CIFX_FUNCTION_FAILED	Generic Function failure
0x800A000A	CIFX_FILE_OPEN_FAILED	A file could not be opened
0x800A000B	CIFX_FILE_SIZE_ZERO	File size is zero
0x800A000C	CIFX_FILE_LOAD_INSUFF_MEM	Insufficient memory to load file
0x800A000E	CIFX_FILE_READ_ERROR	Error reading file data
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 fit the device data
0x800A0013	CIFX_MEMORY_MAPPING_FAILED	Error mapping dual port memory
0x800A0014	CIFX_NO_MORE_ENTRIES	No more entries available (e.g. while enumerating directories)
0x800A0015	CIFX_CALLBACK_MODE_UNKNOWN	Unkown callback handling mode
0x800A0016	CIFX_CALLBACK_CREATE_EVENT_FAILED	Failed to create callback events
0x800A0017	CIFX_CALLBACK_CREATE_RECV_BUFFER	Failed to create callback receive buffer
0x800A0018	CIFX_CALLBACK_ALREADY_USED	Callback already used
0x800A0019	CIFX_CALLBACK_NOT_REGISTERED	Callback was not registered before
0x800A001A	CIFX_INTERRUPT_DISABLED	Interrupt is disabled
0x800Bxxxx		
0x800B0001	CIFX_DRV_NOT_INITIALIZED	Driver not initialized
0x800B0002	CIFX_DRV_INIT_STATE_ERROR	Driver init state error
0x800B0003	CIFX_DRV_READ_STATE_ERROR	Driver read state error
0x800B0004	CIFX_DRV_CMD_ACTIVE	Command is active on device
0x800B0005	CIFX_DRV_DOWNLOAD_FAILED	General error during download
0x800B0006	CIFX_DRV_WRONG_DRIVER_VERSION	Wrong driver version

Table 9: Error codes (1)

Value	Symbol	Description
0x800B0030	CIFX_DRV_DRIVER_NOT_LOADED	CIFx driver is not running
0x800B0031	CIFX_DRV_INIT_ERROR	Failed to initialize the device
0x800B0032	CIFX_DRV_CHANNEL_NOT_INITIALIZED	Channel not initialized (xOpenChannel() not called)
0x800B0033	CIFX_DRV_IO_CONTROL_FAILED	IOControl call failed
0x800B0034	CIFX_DRV_NOT_OPENED	Driver was not opened
0x800C0010	CIFX_DEV_DPM_ACCESS_ERROR	Dual port memory not accessible (board not found)
0x800C0011	CIFX_DEV_NOT_READY	Device not ready (ready flag failed)
0x800C0012	CIFX_DEV_NOT_RUNNING	Device not running (running flag failed)
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
0x800C0017	CIFX_DEV_PUT_TIMEOUT	Send packet timeout
0x800C0018	CIFX_DEV_GET_TIMEOUT	Receive packet timeout
0x800C0019	CIFX_DEV_GET_NO_PACKET	No packet available
0x800C001A	CIFX_DEV_MAILBOX_TOO_SHORT	Mailbox is too short for a packet
0x800C0020	CIFX_DEV_RESET_TIMEOUT	Reset command timeout
0x800C0021	CIFX_DEV_NO_COM_FLAG	Communication flag not set
0x800C0022	CIFX_DEV_EXCHANGE_FAILED	I/O data exchange failed
0x800C0023	CIFX_DEV_EXCHANGE_TIMEOUT	I/O data exchange timeout
0x800C0024	CIFX_DEV_COM_MODE_UNKNOWN	Unknown I/O exchange mode
0x800C0025	CIFX_DEV_FUNCTION_FAILED	Device function failed
0x800C0026	CIFX_DEV_DPMSIZE_MISMATCH	DPM size differs from configuration
0x800C0027	CIFX_DEV_STATE_MODE_UNKNOWN	Unknown state mode
0x800C0028	CIFX_DEV_HW_PORT_IS_USED	Device is still accessed
0x800C0029	CIFX_DEV_CONFIG_LOCK_TIMEOUT	Configuration locking timeout
0x800C002A	CIFX_DEV_CONFIG_UNLOCK_TIMEOUT	Configuration unlocking timeout
0x800C002B	CIFX_DEV_HOST_STATE_SET_TIMEOUT	Set HOST state timeout
0x800C002C	CIFX_DEV_HOST_STATE_CLEAR_TIMEOUT	Clear HOST state timeout
0x800C002D	CIFX_DEV_INITIALIZATION_TIMEOUT	Timeout during channel initialization
0x800C002E	CIFX_DEV_BUS_STATE_ON_TIMEOUT	Timeout setting bus on flag
0x800C002F	CIFX_DEV_BUS_STATE_OFF_TIMEOUT	Timeout setting bus off flag
0x800C0040	CIFX_DEV_MODULE_ALREADY_RUNNING	Module already running
0x800C0041	CIFX_DEV_MODULE_ALREADY_EXISTS	Module already exists

Table 10: Error codes (2)

Value	Symbol	Description
0x800C0050	CIFX_DEV_DMA_INSUFF_BUFFER_COUNT	Number of configured DMA buffers insufficient
0x800C0051	CIFX_DEV_DMA_BUFFER_TOO_SMALL	DMA buffers size too small (min size 256Byte)
0x800C0052	CIFX_DEV_DMA_BUFFER_TOO_BIG	DMA buffers size too big (max size 63,75KByte)
0x800C0053	CIFX_DEV_DMA_BUFFER_NOT_ALIGNED	DMA buffer alignment failed (must be 256Byte)
0x800C0054	CIFX_DEV_DMA_HANSHAKEMODE_NOT_SUPPORTED	I/O data uncontrolled handshake mode not supported
0x800C0055	CIFX_DEV_DMA_IO_AREA_NOT_SUPPORTED	I/O area in DMA mode not supported (only area 0 possible)
0x800C0056	CIFX_DEV_DMA_STATE_ON_TIMEOUT	Set DMA ON Timeout
0x800C0057	CIFX_DEV_DMA_STATE_OFF_TIMEOUT	Set DMA OFF Timeout
0x800C0058	CIFX_DEV_SYNC_STATE_INVALID_MODE	Device is in invalid mode for this operation
0x800C0059	CIFX_DEV_SYNC_STATE_TIMEOUT	Waiting for synchronization event bits timed out

Table 11: Error codes (3)

5 Appendix

5.1 List of tables

Table 1: List of revisions.....	4
Table 2: Terms, abbreviations and definitions.....	5
Table 3: References.....	5
Table 4: cifX Device Driver - Files installed by the INF file.....	12
Table 5: cifX Device Driver - Registry keys created by the INF file	12
Table 6: cifX Device Driver - SDK directory content.....	13
Table 7: Timer resolution.....	16
Table 8: Additional information for Windows 7	17
Table 9: Error codes (1)	18
Table 10: Error codes (2)	19
Table 11: Error codes (3)	20

5.2 List of figures

Figure 1: CifX Device Driver - Architecture.....	8
Figure 2: CifX Device Driver – System architecture	10
Figure 3: Windows 7 64-bit with standard IOCTL handling	11
Figure 4: Windows 7 64-bit with direct IOCTL handling	11

5.3 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