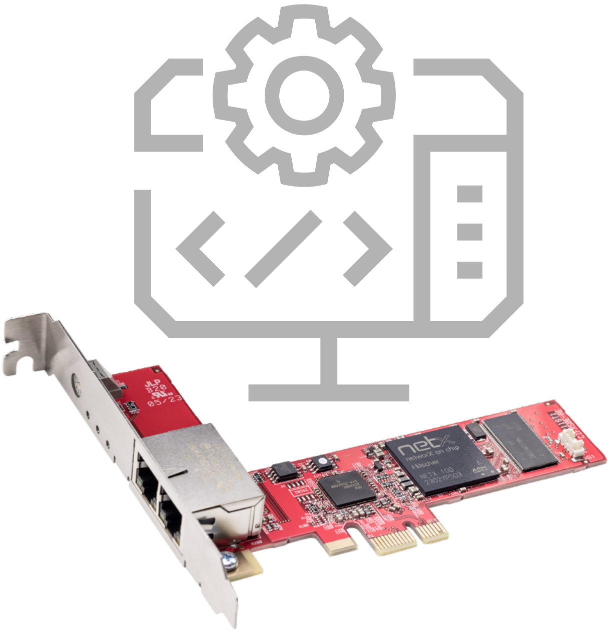
hilscher

cifX / netX

Device Driver

- → Standard application interface
- → Standard code base (C toolkit)
- → Independent of fieldbus system
- → Number of supported devices limited only by operating system



Hilscher offers a series of device drivers for a wide variety of operating systems

Most operating systems require a device driver to integrate hardware components. Hilscher cifX PC Cards and Embedded Modules like comX/netJACK require such device drivers as well.

All Hilscher drivers have a standard user interface (CIFX-API) and are based on a C toolkit also usable for independent device driver development.

The drivers are fully responsible for administering the communication components, including loading firmware and configuration files. The user interface contains all of the functions needed to initialize, parametrize, exchange I/O data and aquire status information.

Depending on the operating system, the user interface is contained in a DLL or library. As far as possible, the drivers are supporting existing bus connections like PCI, PCIe and dual-port memory (DPM) connections.
The C toolkit, and therefore the basis of the drivers code, is designed in such a way that an unlimited number of communication components can be used simultaneously. However, the specific characteristics of each operating system must be taken into account.

Linux
IntervalZero
INtime for Windows
QNX
Windows
Windows Embedded Compact

Device Driver

Windows

Windows Embedded Compact

The cifX Device Driver for Windows is a kernel-mode driver that runs in Ring 0 of the operating system and supports the Windows plug-and play mechanisms. PCI and PCIe components are supported and can be combined arbitrarily using an unlimited number of components.
The cifX Device Driver for Windows CE is a stream driver, running in the kernel of the operating system. Access to the driver functions is offered by a driver API DLL (cifXCEDLL).
The DLL covers the device IO control calls used to communicate with the driver and offers the same CIFX API like on the Windows desktop operating systems.
The driver's function interface is fieldbus system independent and includes functions for the recognition of installed hardware, reading system information, initializing and parametrizing field bus systems, exchanging cyclical and acyclical data and reading status information, as well as time-monitoring and watchdog function.

Linux

The cifX Device Driver for linux, is splitted in a kernel module (free of chargeable) and a user space library (chargeable). The kernel module is based on the generic uio module and is responsible for cifX hardware detection and preparation to allow mapping the device memory (DPM) to user space. The User space driver libcifX is a user mode driver providing the whole device specific functionality. The user space library offers the same API as the cifX driver API for Windows.

QNX

The cifX Device Driver for QNX is available as a static library or a shared object, which is built around the cifX Toolkit.
Any application which needs to access a cifX device can use the device specific functions provided by this driver library.

VxWorks

The cifX Device Driver for VxWorks offers access to the Hilscher netX based hardware (e.g. CIFX 50, comX) with the same functional API as the cifX device driver for Windows and offers transparent access to the different devices.
It is available as a library built around the cifX Toolkit. Any application which needs to access a cifX device can use the device specific functions provided by this driver library.

INtime for Windows

The cifX Device Driver for INtime offers access to the Hilscher netX based hardware (e.g. CIFX 50) with the same functional API as the cifX device driver for Windows. It is also possible to use the cifX INtime driver from normal Windows environment or from inside the INtime real-time kernel.
The cifX INtime driver runs inside the Real-Time extension in a separate address space. A special version of the cifX Application Interface DLL (cifX32DLL.dll) is provided with the INtime driver to allow Windows applications the accesses to the INtime driver from the Windows User Space.

IntervalZero RTX64 Real-time Windows

The cifX Device Driver for IntervalZero RTX® is available as a dynamic library built around the cifX Toolkit. Any application which needs to access a cifX device can use the device specific functions provided by this driver library.
The driver offers access to the Hilscher netX based hardware with the same functional API as the cifX device driver for Windows® and offers transparent access to the different devices.
User processes on RTX till version 2012 gain access to the cifX driver functions by using the LoadLibrary() and GetProcAddress() calls (explicit library load). As RTX64 provides its own image loader, the driver library can also be loaded implicitly here.

WinAC

The cifX Device Driver for WinAC is an extension DLL (CCX) for the Siemens SIMATIC WinAC RTX 2010 SP1 system and allows the Siemens SIMATIC WinAC system to access Hilscher CIFX based communication hardware form the SPS cycle.
Driver functions are accessed from STEP7 / TIA via FBs / Cs. At least two functions are needed to exchange I/O data during the program cycle and the I/O data are mapped into data blocks of the STEP7 program (one DB for input and one DB for output data).

Technical Data	NXDRV WIN	NXDRV CE	NXDRV Linux	NXDRV VxWorks	NXDRV QNX
Associated products	cifX / comX / netPLC / netJACK	cifX / comX / netPLC / netJACK	cifX / comX / netJACK	cifX and comX	cifX and comX / netJACK
Communication controllers	netX 500, 100, 50/51/52, 90	netX 500, 100, 50/51/52	netX 500, 100, 50/51/52, 90	netX 500, 100, 50/51/52	netX 500, 100, 50/51/52, 90
Operating systems	Windows XP / VISTA Windows 7, 8, 10, 11	Windows CE 5.0 / 6.0 Windows EC 7.0 / 2013	Linux kernel 2.6, 3x, 4x, 5x	VxWorks V5.5, 6.2, 6.7, 6.9 VxWorks V7 (rev. 2015)	QNX Neutrino RTOS V6.4.0, 7.x, 8.x
Driver type	32/64-bit kernel-mode driver, based on Microsoft KMDF driver architecture	32-bit kernel-mode driver	32-bit user mode library, based on UIO Kernel Module	Image Component as function interface	32-bit library
User interface	API-DLL, Opening and closing the driver, Initializing the PC cards, Accessing process data map, Writing/reading commands, Writing parameters, Reading status information, Watchdog function	API-DLL, Opening and closing the driver, Initializing the PC cards, Accessing process data map, Writing/reading commands, Writing parameters, Reading status information, Watchdog function	Shared Library, Opening and closing the driver, Initializing PC cards, Accessing process data map, Writing/reading commands, Writing parameters, Reading status information, Watchdog function	API-DLL, Opening and closing the driver, Initializing the PC cards, Accessing process data map, Writing/reading commands, Writing parameters, Reading status information, Watchdog function	Shared Library, Opening and closing the driver, Initializing PC cards, Accessing process data map, Writing/reading commands, Writing parameters, Reading status information, Watchdog function
Operating mode	Polling, Interrupt	Polling, Interrupt	Polling / Interrupt (only for devices accessed via UIO Kernel Module)	Polling, Interrupt	Polling, Interrupt
Hardware support	PCI / PCIe	DPM, PCI/PCIe	DPM, PCI / PCIe	DPM, PCI / PCIe	DPM, PCI / PCIe
of cards/system	Unlimited	10 devices (limited by Windows CE)	Variable	Variable	Variable
Other characteristics	Based on the CIFX toolkit. Support for development boards.	Source code must be compiled for the target system. No simultaneous access to one card. For hardware without flash memory (CIFX50), the driver needs access to the firmware and configuration files during the start phase.	Support for development boards. No simultaneous access to one card from multiple applications.	Support for development boards. Source Code must be compiled for the target system. No simultaneous access to one card from multiple applications.	Support for development boards. Source Code must be compiled for the target system. No simultaneous access to one card from multiple applications.

→ cifX / netX Device Driver
Service-Hotline: +49 (0) 6190 9907-90
www.hilscher.com

empowering communication

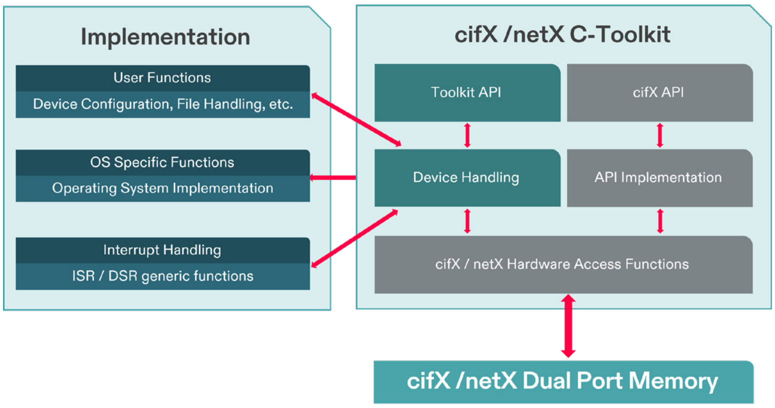
Product information

Technical Data

cifX/netX device driver toolkit source code

The cifX toolkit consists of C-source and header files allowing abstract access to the dual-port memory (DPM) defined by Hilscher and offered by cifX, comX and netX devices.
It contains all of the user interface functions as well as the generic access functions needed to use the Hilscher DPM.

→ ANSI-C source code
→ Operating-system-dependent functions in a separate C module
→ Support of little-endian/big-endian
→ Trace output integrated in the source


Implementation
User Functions
Device Configuration, File Handling, etc.
OS Specific Functions
Operating System Implementation
Interrupt Handling
ISR / DSR generic functions
cifX /netX C-Toolkit
Toolkit API
cifX API
Device Handling
API Implementation
cifX / netX Hardware Access Functions
cifX /netX Dual Port Memory

Technical Data: NXDRV-TKIT
Associated products cifX / comX / netJACK
Communication controllers netX 500, netX 100, netX 50/51/52, netX 90
Operating systems any
Functions Defining the CIFX-API interface, Hardware initialization, Downloading firmware and configuration files during startup, Initializing PC cards, Recognizing flash-based and non-flash-based netX devices

Technical Data
Operating mode Polling / Interrupt via generic ISR and DSR functions
Hardware support PCI / PCIe, dual-port memory connection
Cards per PC Variable
Other characteristics Operating-system-specific hardware functions (such as PCI functions) not included, Support for development boards (NX PCA PCI) without interrupt

Note: All technical data may be changed without further notice.

Artikel	Bestellnummer	Beschreibung
NXDRV-WIN	*	cifX/netX Device Driver for Microsoft Windows
NXDRV-CE	6211.020	cifX/netX Device Driver for Windows CE/EC
NXDRV-LINUX	*	cifX/netX Device Driver for Linux
NXDRV-VXWORKS	6211.040	cifX/netX Device Driver for VxWorks
NXDRV-InTime	6211.050	cifX/netX Device Driver for InTime
NXDRV-QNX	6211.060	cifX/netX Device Driver for QNX
NXDRV-IntervalZero	6211.070	cifX/netX Device Driver for IntervalZero
NXDRV-WinAC	6211.080	cifX/netX Device Driver for WinAC
NXDRV-TKIT	*	cifX/netX Device Driver Toolkit Source Code

** free of charge*