Kithara | The Pulse of Real-Time

Product Overview

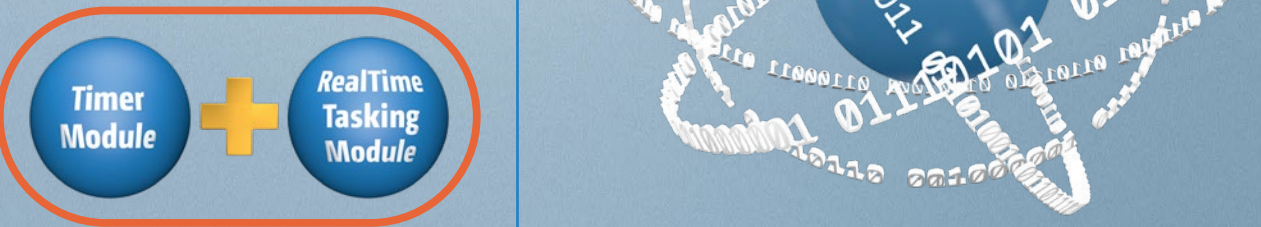
Kithara RealTime Suite

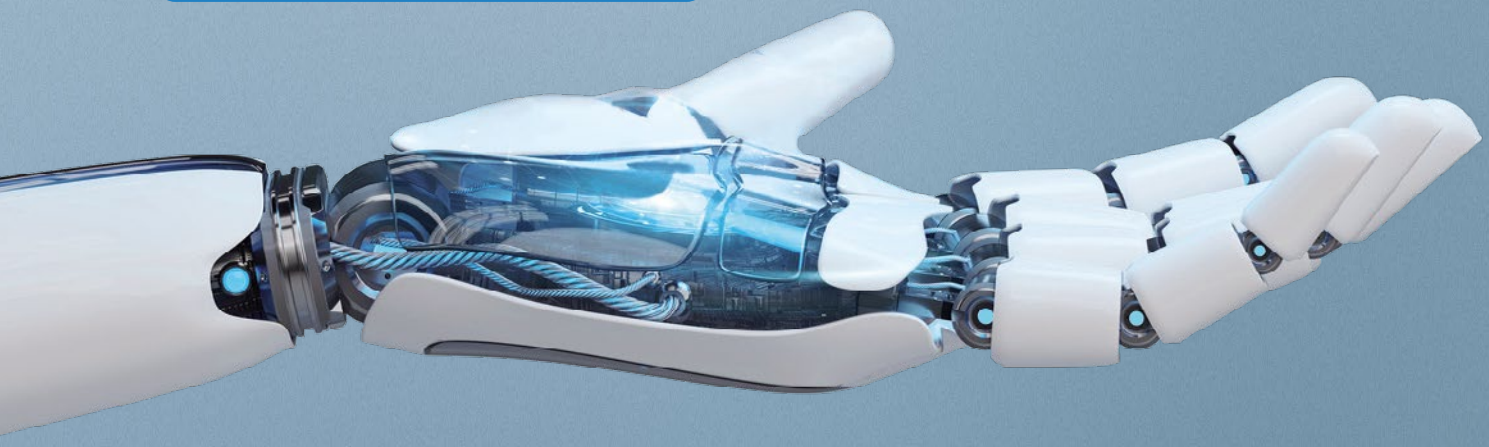
The real-time solution for automation, communication, machine vision, automotive and data storage

NEW!

Timer Module
Highly accurate detection of system time as well as short time delays: › Page 15

RealTime Tasking Module
High frequency real-time timer routines and priority-driven, preemptive real-time multitasking: › Page 16





Precise timing

Real-time for Windows

In the modern world, time is a crucial factor. Automated processes on an industrial level require precise communication between application and machine. High-performance devices generate enormous amounts of data that need to be immediately processed.

Kithara develops software for accurate control, monitoring and analysis down to the microsecond for processes in industry and research. Our products can be found in the automotive sector, the aerospace industry, in food manufacturing as well as medical engineering. Even in the cutting-edge field “autonomous driving”, Kithara is an important contributor.

Problem and Solution: Operating systems such as Microsoft Windows sometimes override priorities of running processes in order to execute their own programs and maintenance functions. This can lead to unwanted interruptions in the sequence of applications.

Kithara basically creates its own protected area, which is independent and unrestricted by the operating system.

Kithara RealTime Suite

Windows real-time extension for PC-based automation

Kithara RealTime Suite is a modular real-time extension for Windows, combining hardware-dependent programming, communication, automation protocols as well as image capturing and image processing into a single high-performance real-time system. Due to seamless integration of modules, users are provided with all the necessary functions in one piece.

The efficient real-time drivers are the basis for socket communication via TCP and UDP, including image capturing with GigE-Vision-compatible Kameras as well as industrial Ethernet protocols. The software is further supplemented by real-time drivers for CAN and UART interfaces. Among the supported automation protocols is a comprehensive EtherCAT® master implementation including Distributed Clock, Safety-over-EtherCAT, hot-plug capability and cable redundancy.

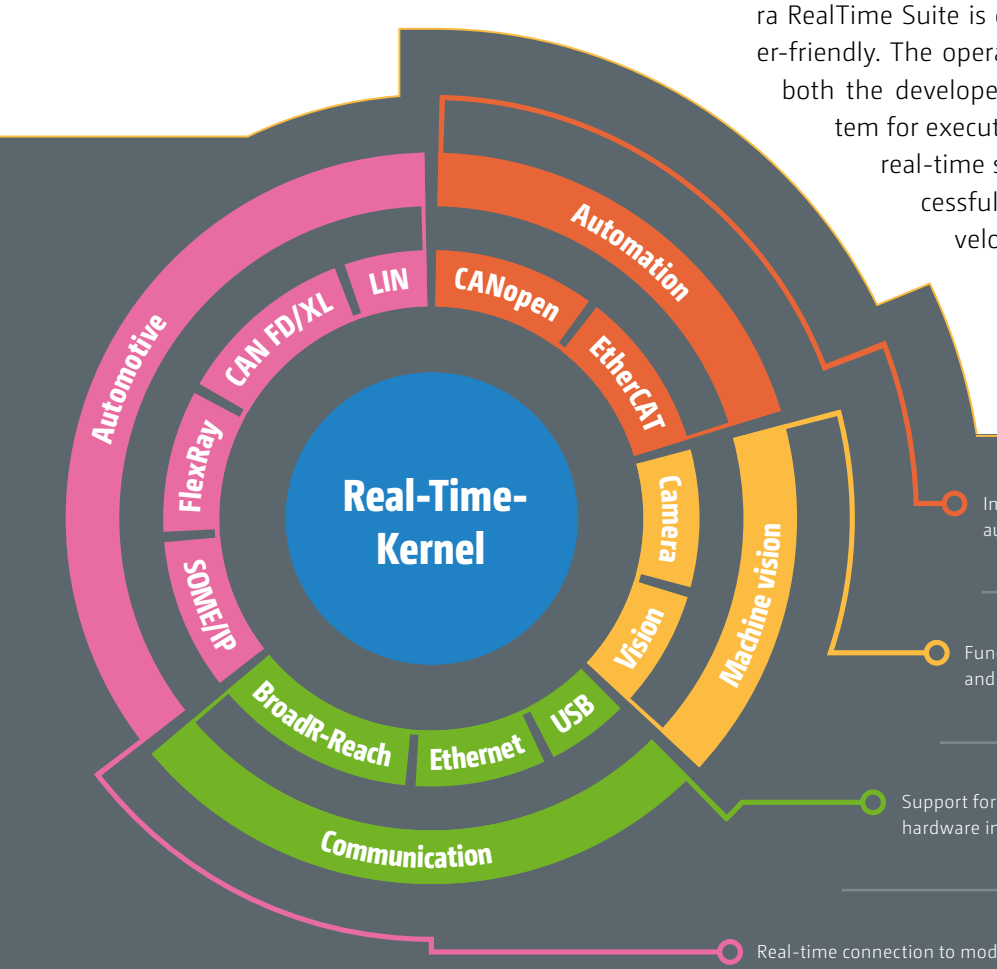
Furthermore, a CANopen® master is provided via a variety of different CAN interfaces, which can also be integrated into EtherCAT®.

Hardware-dependent programming includes direct I/O access, memory access and interrupt handling as well as a convenient high level interface for multifunction boards.

For processing image data captured by GigE Vision® or USB3 Vision®, it is possible to run comprehensive image processing libraries, such as Halcon or OpenCV, within the real-time environment.

For fast real-time data storage, SSDs with NVMe interface (can be multiplied with RAID 0) are supported. Additionally, data sets can be immediately stored as hierarchically structured files in MDF or PCAPng format.

The software is easily accessible and intuitive. Due to the use of familiar programming languages and development environments, Kithara RealTime Suite is designed to be highly user-friendly. The operating system Windows is both the developer platform and host system for execution, rendering a separate real-time system obsolete. This successfully minimizes time and development costs.



Important standards for production automation and quality control | Page 20

Functions for industrial cameras and image processing libraries | Page 28

Support for all relevant hardware interfaces | Page 36

Real-time connection to modern automotive bus systems | Page 50

Our products for your real-time project

Windows is optimally suited for the implementation of industrial applications due to the following traits ...

- popular: intuitive and familiar user interface
- modern: access to the latest communication interfaces
- productive: powerful development tools
- cost efficient: low implementation and training costs
- future-proof: long-living due to its high market share and continual further development

The necessary real-time capabilities, which Windows is missing, is complemented by Kithara RealTime Suite and allows for the execution of time-critical applications due to ...

- deterministic behavior
- extremely low maximum response times
- familiar programming language and tools
- powerful and easy-to-apply functions
- short training time—cost efficient development
- broad support for industrial protocols and standards

Kithara RealTime Suite makes it possible to have both real-time control and visualization within the same system. The application runs in dedicated mode on separate CPU cores, whose quantity can be configured freely. At the same time, Windows is responsible for visualization and user interaction on the remaining CPU cores. This way, both parts run separately without influencing each other.

The combination of Windows and Kithara RealTime Suite allows for the implementation of a diversity of solutions, among them the following fields:

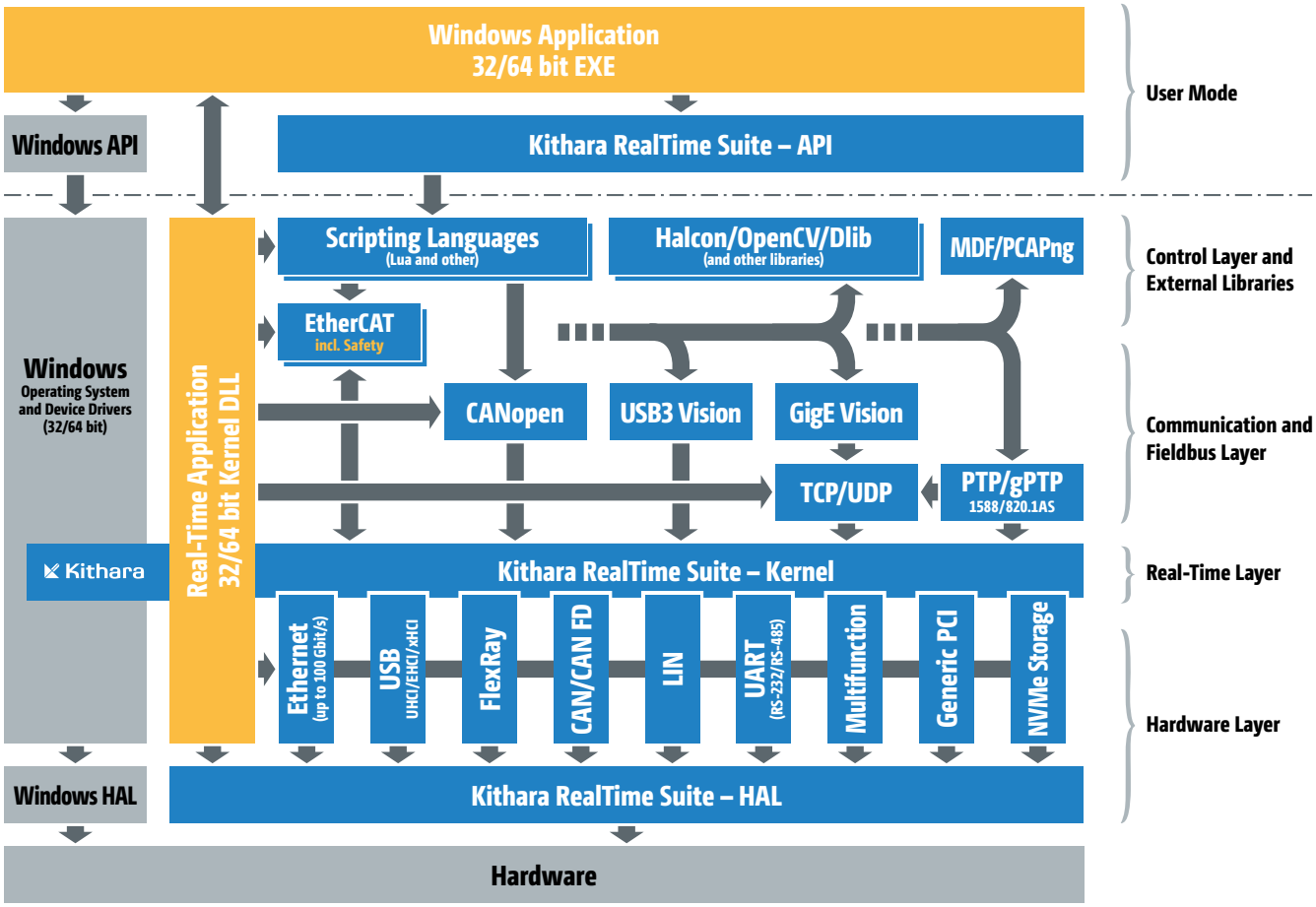
- mechanical engineering, special machine construction and testing systems
- production automation and quality assurance
- lab automation and mobile applications
- measurement and test stations in automotive engineering and aerospace industry
- medical engineering
- laboratories in science and research

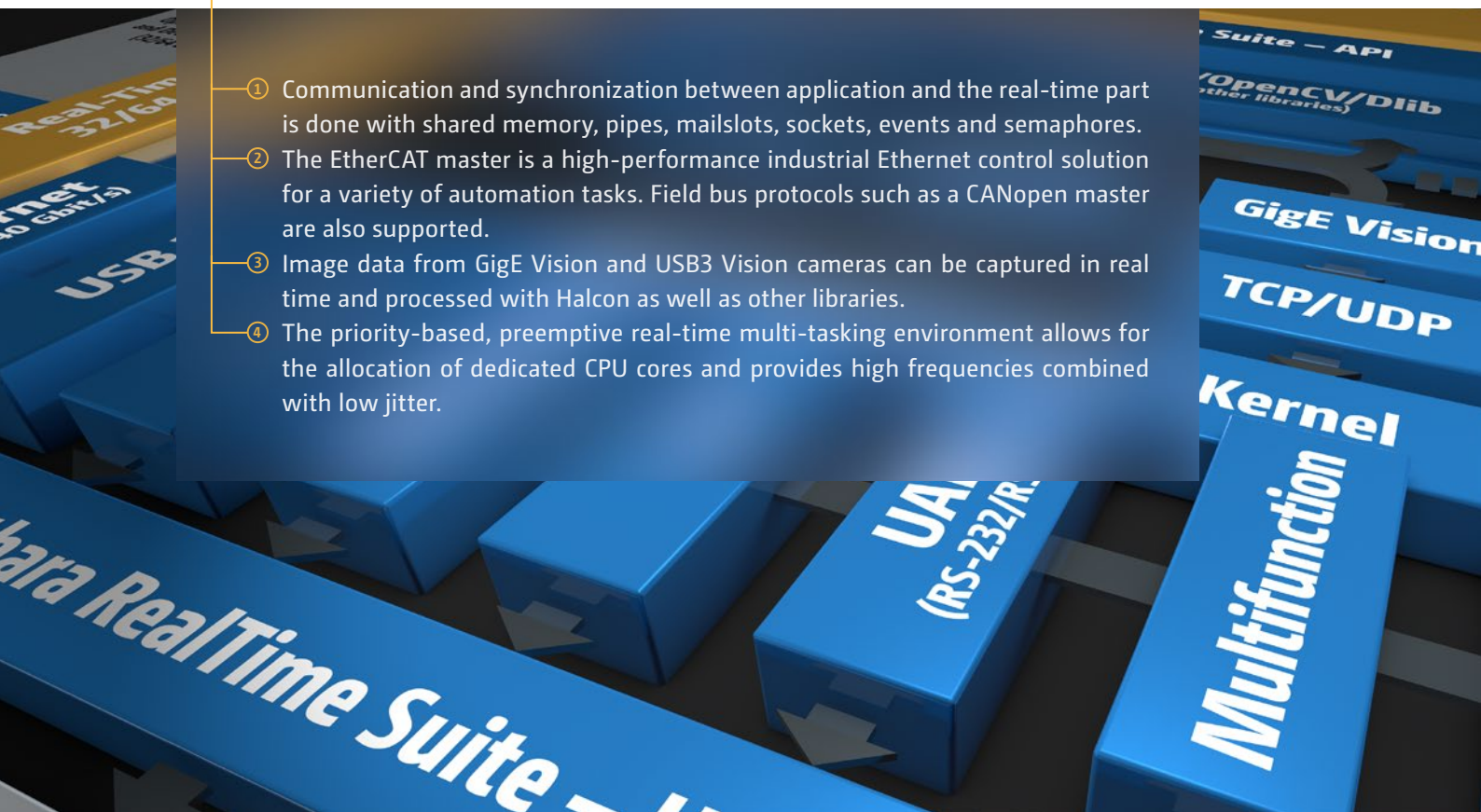
Conclusion

Kithara simplifies your project significantly. You will not need a separate real-time system, thus avoiding an expensive implementation of communication between different systems. Instead, you can utilize the advantages of a uniform solution and benefit from drastically reduced development efforts, coherent programming tools and toolchain. This shortens time to market, lowers costs and results in reduced resource consumption. That makes Kithara RealTime Suite together with Windows the ideal foundation for successful real-time solutions.

By courtesy of: Renk Test System GmbH, Augsburg

The following diagram shows with a layer model how Kithara RealTime Suite is connected to the different software and hardware components of a system.





① Communication and synchronization between application and the real-time part is done with shared memory, pipes, mailslots, sockets, events and semaphores.
② The EtherCAT master is a high-performance industrial Ethernet control solution for a variety of automation tasks. Field bus protocols such as a CANopen master are also supported.
③ Image data from GigE Vision and USB3 Vision cameras can be captured in real time and processed with Halcon as well as other libraries.
④ The priority-based, preemptive real-time multi-tasking environment allows for the allocation of dedicated CPU cores and provides high frequencies combined with low jitter.

How does real time actually work under Windows?

Anyone familiar with the concept of real time knows how important it is nowadays in numerous IT areas. Whether for parallelly operating servomotors in a robotic arm, precise measurement applications or camera-based driving assistance systems—the reaction times between hardware and software components are a crucial factor in many modern electronic technologies. What once was achieved either through special stand-alone real-time operating systems (RTOS) or hardware (DSP and FPGA), can by now also be implemented by utilizing regular, commercially available Windows PCs. However, Windows itself is not a real-time system, as it frequently interrupts running applications in order to execute other programs and processes, i. e., for maintenance. So how can these supposed opposites be brought together? How does real time function with Windows?

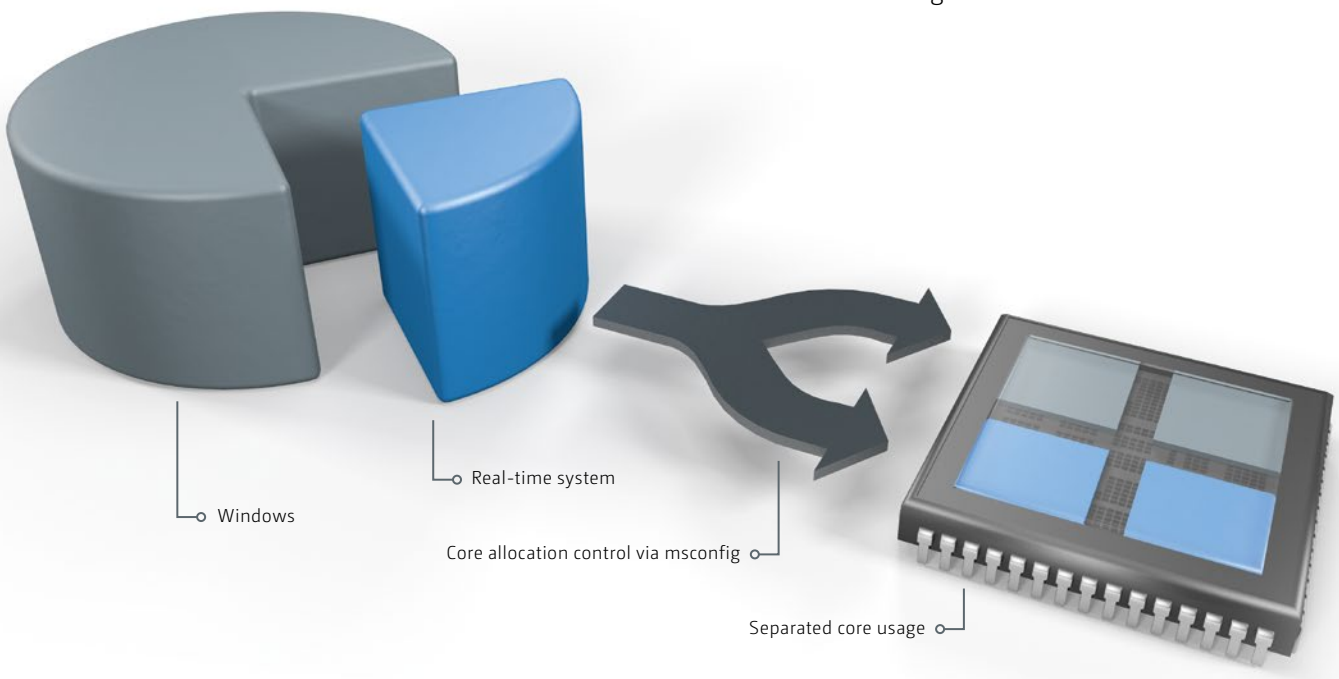
Real time and Windows – The solution

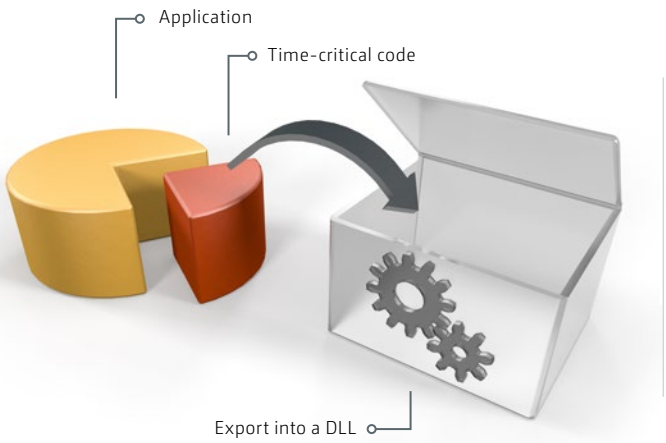
The only requirement for real time on Windows PCs is a processor with at least two cores. The basic operating principle is to instruct Windows to only use a limited number of CPU cores of a multicore processor. On the now vacant cores, the real-time system is booted, which, from here on out, functions just like a separate RTOS, while Windows retains its full operability on the remaining CPU cores. This means that from this point on, Windows and the real-time system run simultaneously and parallel to each other on the same computer, without one restricting the other. The instruction for Windows to boot on a limited amount of cores can be carried out relatively simple by using the configuration program msconfig. A guide for this can be found here: **› Setting up Dedicated CPUs**.

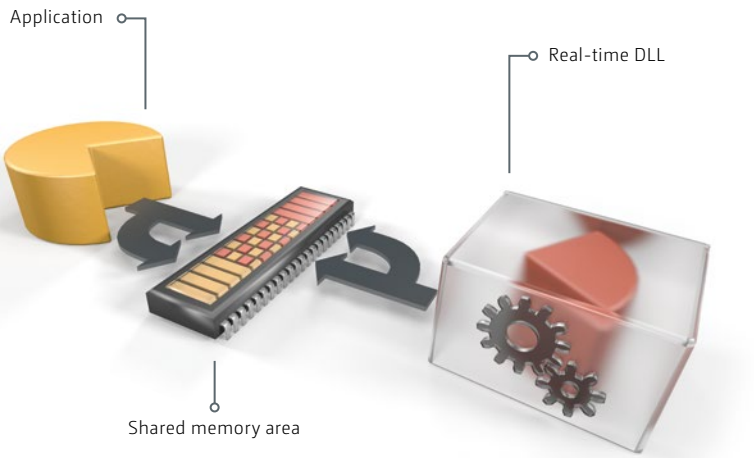
Thanks to this procedure, the real-time system gains a protected area, therefore ensuring that Windows processes do not obtain a higher priority over time-critical operations and thus preventing it from negatively impacting real-time performance. The ultimate goal is to achieve specifically “hard” real-time capabilities on Windows PCs. But what exactly does “hard” mean in this context?

What is the difference between “soft” and “hard” real time?

When it comes to real time, we need to distinguish between different demands and how the timeframe for executing processes is handled. While “soft” real time only approximates to an average of reaction times, “hard” real time needs to guarantee that a set period of time is not exceeded. Due to these higher requirements, achieving “hard” real-time capabilities is significantly more complex and challenging. This deterministic time behavior, however, is indispensable in automated industrial areas, especially in measurement, testing and control technologies.







So how can Windows and real-time context be brought back together?

The real-time system is implemented as a device driver and provides its own API, which allows the user to export the time-critical code parts of a Windows application into a DLL. This DLL is then loaded into the real-time context of the RTOS. In order for Windows and the DLL to communicate with each other from their different contexts – besides using indirect methods such as pipes or sockets – a shared memory area can be created, which both have access to. Despite the utilization of a real-time operating system, Windows remains as both programming interface and host system for execution.

Why use particularly Windows as RTOS?

In contrast to pure real-time operating systems as well as real-time hardware solutions, a Windows PC offers specific advantages. Windows itself is a familiar user interface with broad driver support, that is frequently updated, as well as a wide range of executable programs. A pure RTOS, on the other hand, is heavily specialized and therefore oftentimes much more restricted when it comes to function range or operability. Hardware solutions such as FPGAs and DSPs can be adjusted according to different specifications. However, their programming is complicated and time-consuming. To get "hard" real time working under Windows, only a dual-core PC is required.

Conclusion

Windows is optimally suited for industrial applications due to the following properties:

- wide-spread, intuitive and familiar user interface
- access to the latest communication interfaces
- efficient developer tools
- low implementation and training cost
- long-lasting due to high market share and continuous further development

The missing real-time capabilities of Windows are supplemented by real-time systems such as Kithara RealTime Suite, thus allowing for the implementation of time-critical applications in the first place due to:

- deterministic behavior thanks to high-frequency real-time timers
- extremely low maximum response times
- familiar programming language and tools
- efficient and easy-to-apply functions
- short training time and cost efficient development
- broad support for industrial protocols and standards

Windows and real time are not a contradiction. The seamless interaction of both worlds allows for optimal solutions and often makes the utilization in industrial fields possible to begin with.

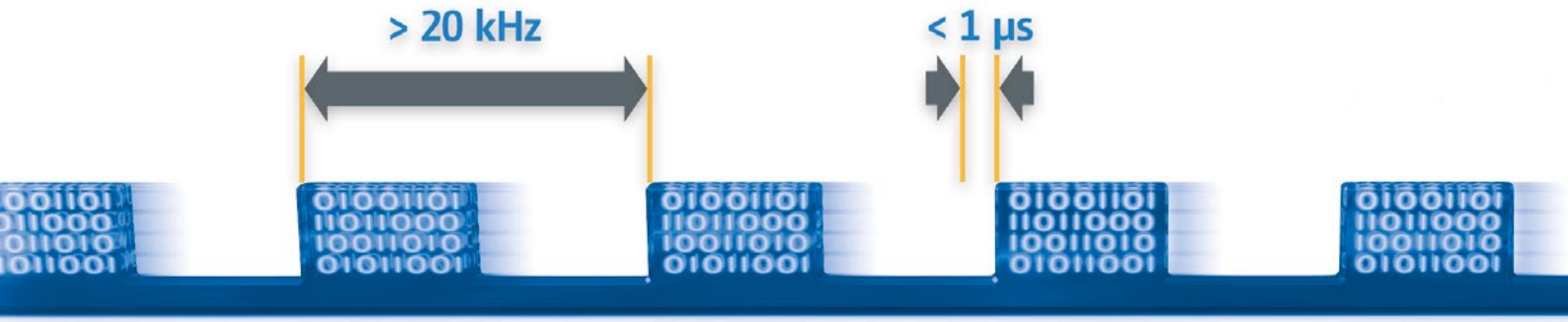
See for yourself how real time can be used under Windows. Just test the **›free trial version** of Kithara RealTime Suite with full function range and many samples that help you get started.

“Hard” Real Time for the Windows Platform

Due to the modular-built system with its broad range of possible implementations, Kithara RealTime Suite is the optimal base for developing efficient applications in automation, hardware communication and machine vision, since these fields are often dependent on time-critical functions.

Kithara RealTime Suite uses the reliable dedicated mode, where single logical CPUs (CPU cores) are run completely autonomous and without any influence from Windows. This way, by using appropriate hardware, cyclical timer calls with frequencies in the upper double-digit kilohertz range can be implemented with deviations (jitters) of only a few microseconds.

At the real-time start, the system first analyzes the hardware, such as all available clocks, and calibrates them. However, not only timer routines can be run with high frequencies. The preemptive real-time multitasking system provides tasks (threads) with up to 255 priority levels so that the action with the highest priority is always executed first. Less prioritized actions are interrupted immediately and are continued only after the completion of higher prioritized actions.



For the synchronization between kernel DLL and Windows application, events, semaphores, mutexes, data and message pipes, sockets as well as shared memory are provided. Individual tasks can be specifically allocated to different logical CPUs in order to further optimize performance and integrity of specialized tasks as well as the scalability of the entire system.

Built around the kernel are a variety of other modules for the connection of external devices and systems via different communication and access interfaces as well as for specialized tasks. One example is the Storage Module, which allows for fast data storing on SSDs with a sustained throughput rate of several gigabytes per second. Captured measurement data can also be stored in future-proof Measurement Data Format (MDF) as terabyte-sized files. Furthermore, powerful diagnosis and programming tools such as Kithara Kernel Tracer and Kithara Performance Analyzer are at the user's disposal, making the whole development process even more flexible and efficient.

Customizable Software Combination

Freely combine your individal set of functions or contact us so we can help you choose the right modules. This custom driver gives you highest possible flexibility and has the following features:

- Kithara RealTime Suite is built modularly and allows for creating an individually customized real-time operating system tailored to your needs.
- The name of development and runtime files can can be chosen freely.
- Solid API, no modifications necessary in case of new versions.
- With the purchase of the module you will automatically receive a developer's license.
- Favourable price scale conditions for runtime licenses.
- Multiple updates provided over a period of 12 months after initial purchase
- Update service extendable by an additional 12 months.
- Extendable with additional modules at any time.
- One-off 10 hours of developer support included, beyond that different support packages are available (20, 50, 100 hours)
- **›Support** is handled in the Kithara ticket system, enables fast and direct contact with developers.

Our qualified support team provides you with the best possible solution for integrating the software into your project and our development support also assists you with questions beyond the software.

Platforms

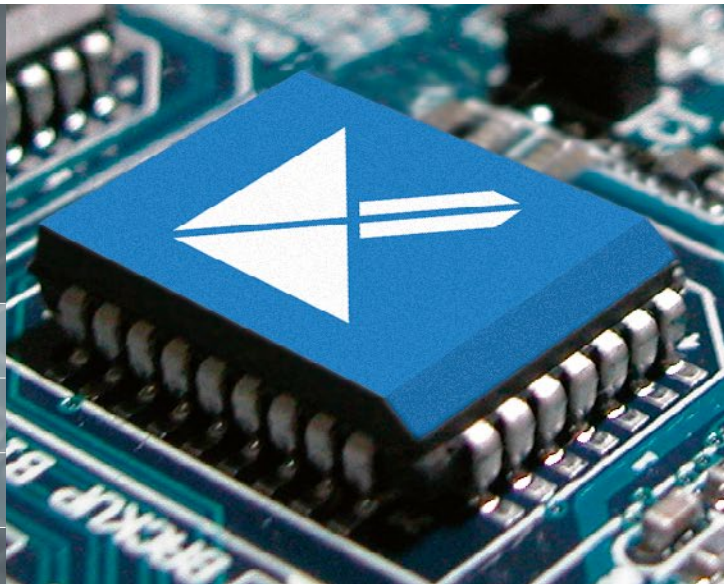
Real-time capability can only be achieved on the kernel level. For this purpose a programming language is required that is able to generate native machine code, e.g. C/C++ or Delphi. Nevertheless Kithara RealTime Suite supports various platforms, for example a .NET environment. The solution is, to transfer the time-critical code into a DLL, which will be loaded directly into the real-time context at the kernel level. Kithara RealTime Suite provides all this functions. Instantly usable program frameworks for the platforms mentioned above are part of every software delivery.

Operating system

The software supports the following operating systems:

- Windows 10 and 11 (32 and 64 bit), dedicated mode
- Windows Server 2016, 2019 and 2022, dedicated mode

System Requirements
Kithara RealTime Suite supports a broad range of hardware and software combinations as well as industrial standards. For further questions, please contact us.



Hardware

The software runs on the following hardware:

- CPU: AMD (Athlon) or Intel (Pentium 2), 32 or 64 bit
- Dual- or multi-core; optional hyper-threading with multi-core, currently up to 48 logical CPUs, above on request
- ACPI recommended (Advanced Control and Power Interface)
- The PC must be able to support PAE (Page Address Extension)
- Gigabit-Ethernet up to 100 Gbit/s, almost every controller of Intel and RealTek is supported

Compiler

Immediately usable program frameworks for the following programming languages/compilers are included in the supplied software (other programming languages can be supported on request):

- Visual Studio 20xx C++ with MFC user interface (provided project files: VS 2010/12/13/15/17/19/22)
- C++ Builder (Embarcadero, formerly Borland) VCL user interface
- Delphi (Embarcadero, formerly Borland) VCL user interface
- Visual Studio 20xx C# with WPF user interface (incl. C++ DLL for real-time execution, supplied project files: VS 2010/12/13/15/17/19/22)

In order to execute code in a real-time context, generally any compiler that can generate a DLL with native machine code, is usable. Furthermore, the Windows application can also be generated with other programming languages, such as C#.

Packages

For common fields of application, we have combined the most important modules into packages. These already cover the requirements of the majority of our customers, include additional tools and are much more cost-efficient than acquiring single modules.

Kithara RealTime EtherCAT – EtherCAT in real time for Windows

Kithara RealTime EtherCAT is a modular real-time extension for Windows operating systems, with specific focus on real-time automation with EtherCAT.

Kithara RealTime EtherCAT Vision Set – EtherCAT and vision in real time under Windows

Kithara RealTime EtherCAT Vision Set is a real-time extension for Windows operating systems, with focus on automation as well as image capture and processing.

Kithara RealTime Vision – Image capture and processing in real time for Windows

Kithara RealTime Vision is a modular real-time extension for Windows operating systems, with focus on image capture and image processing in real time with GigE Vision and USB3 Vision.

Kithara RealTime Automotive – Real time for the automotive field

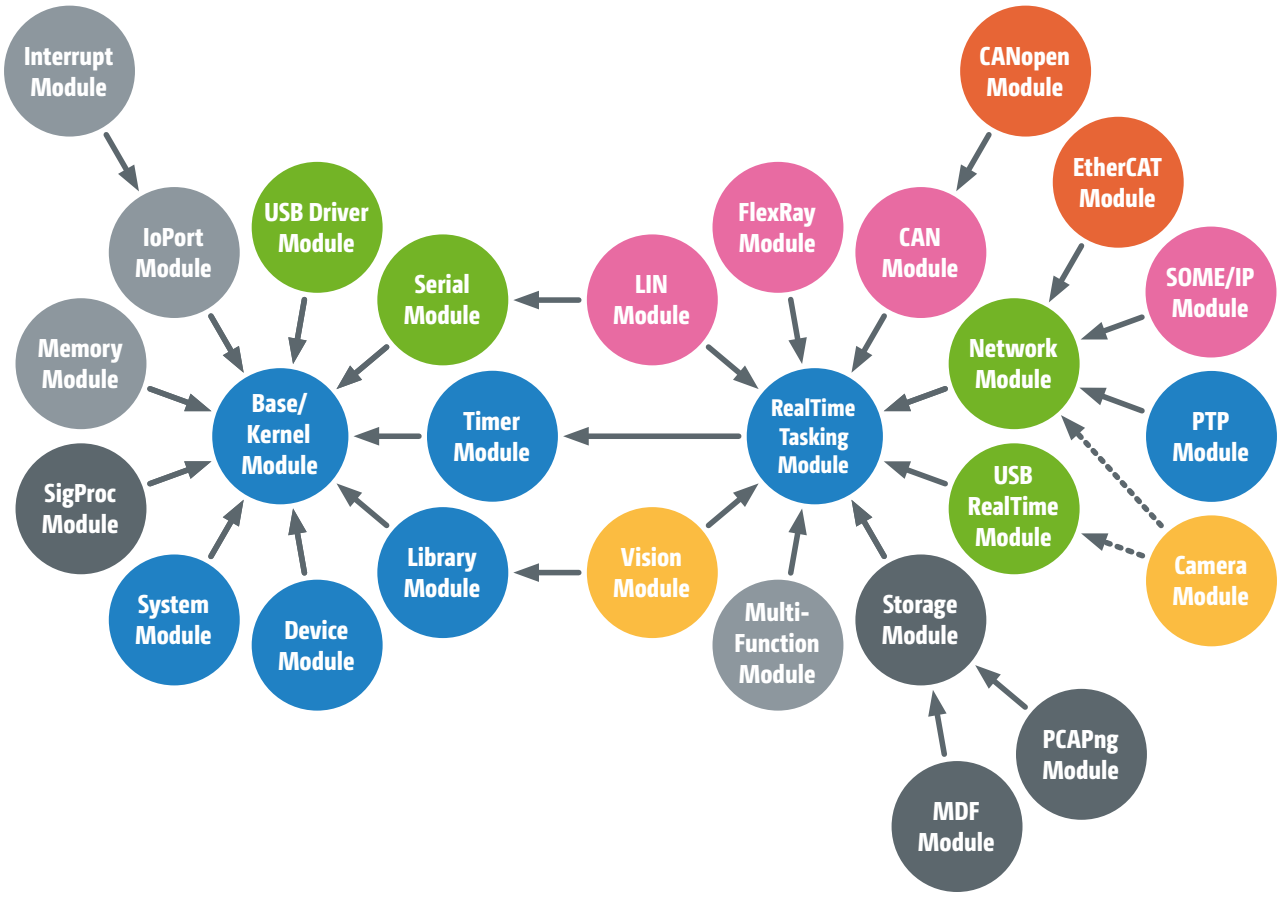
Supports the most successful bus systems in automotive engineering, relevant for driver safety, driving comfort, and driving assistance: CAN/CAN FD, FlexRay and LIN.

Packages	RealTime EtherCAT	RealTime CANopen	RealTime Vision	RealTime Automotive
Basic Functions				
Base/Kernel Module	✓	✓	✓	✓
System Module	✓	✓	✓	✓
Library Module	✓	✓	✓	✓
Real-Time System				
Timer Module	✓	✓	✓	✓
RealTime Tasking Module	✓	✓	✓	✓
Storage Module				
PCAP Module / MDF Module				
PTP Module				
Hardware Access				
IoPort Module				
Memory Module				
Interrupt Module				
MultiFunction Module				
Communication				
Network Module	✓		✓	
USB RealTime Module			✓	
Serial/UART Module				✓
Device Module				
Development Tools				
Kernel Tracer	✓	✓	✓	✓
Performance Analyzer	✓	✓	✓	✓
Master Monitor	✓			
Automation				
EtherCAT Module	✓			
CANopen Module		✓		
Machine Vision				
Camera Module			✓	
Dlib Extension			✓	
Halcon Extension*			✓	
OpenCV Extension*			✓	
Automotive				
FlexRay Module				✓
CAN Module		✓		✓
CAN FD				✓
LIN Module				✓
BroadR-Reach (100BASE-T1/1000BASE-T1)				

Choose a suitable package and combine it with additional modules if desired. A detailed list of all available modules can be found below.
The Halcon license needs to be acquired separately from MVTec. OpenCV is a free programming library.

Modules

Kithara RealTime Suite consists of various modules. The software can be customer-specifically composed of these modules or acquired as ready-to-go packages which can be extended with additional modules. The following diagram illustrates the dependencies between the modules:



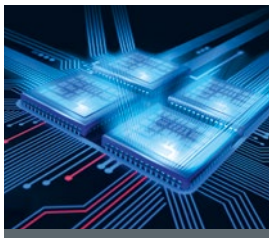
Basic Functions

Base/Kernel Module

General management, execution at kernel level, real-time memory management, debugging support, system information, basic resources, device handling.

Common

The Base/Kernel Module is generally the basis for every real-time system of Kithara RealTime Suite and thus is always required. It provides functions for opening the real-time driver from the Windows application as well as for the execution of real-time code at kernel level in order to reach the real-time context. It enables functions for general management tasks, version information and helper functions for debugging and for determining system information. It also contains mechanisms for fundamental resources such as application threads, events, callbacks, shared memory, data and message



pipes and fast mutex objects. Additionally, it provides functions for real-time memory management, memory copy as well as generic WDM drivers for plug-and-play installation. The integrated KiK64 function allows for 32-bit code to run on 64-bit systems (only with the 64-bit version of Kithara RealTime Suite)

Features

- Functions for opening the driver
- Execution of real-time code at kernel level
- Retrieve error descriptions
- Debugging support
- Determining system information
- Application threads
- Functions for event and callback objects, fast mutex objects
- Functions for shared memory
- Data and message pipes with automatic decoupling between writing and reading side
- Device information, driver management
- Real-time memory management
- Memory copy function for real-time context
- Generic WDM driver for plug-&-play installation included
- Generated log messages compatible to Kithara Kernel Tracer
- KiK64 function: 32-bit code executable on 64-bit systems (only with the 64-bit version of Kithara RealTime Suite)
- Only languages that can generate native machine code (C/C++ or Delphi)
- Supports the following operating systems: Windows 10 and 11 (32 and 64 bit) as well as Windows Server 2016, 2019 and 2022, each with Dedicated Mode
- Supports multi-core processor, Hyper-threading and NUMA multi-processor PCs
- Base/Kernel Module Addons: Kernel CPU Extension, Kernel Memory Extension

Supported Hardware

The following system requirements are necessary:

- Support of the CMPXCHG8B/CMPXCHG16B CPU instruction

Extensions

Kernel NUMA Extension

Use of multi-socket systems with “non-uniform memory access” architecture in real time, automatic optimization of memory allocation.

Thunderbolt Extension

Support for devices connected with Thunderbolt.

System Module

Interception of system events, e. g. protection faults and system crashes at the kernel level (e. g. FailSafe-Handler/“BlueScreen-Handler”)

Common

The System Module provides the interception of system events, e. g. protection faults and system crashes at the kernel level. For that purpose pre-registered handler (Callback functions or real-time tasks) can be started. Thus, FailSafe handler (“BlueScreen handler”) can be implemented and in case of an error a predefined reaction can be executed.

Features

- Interception of system events with Callback functions and real-time tasks in the case of protection faults and system crashes at the kernel level
- Implementation of FailSafe handler (“BlueScreen handler”)

Real-Time System

Timer Module

NEW!

Highly accurate and calibrated detection of system time as well as short time delays.

Common

The Timer Module with its clock functions represents the basis for all real-time tasks. It provides a calibrated access to all hardware time bases in the system. Time information can be freely converted into any time formats, even user specific ones. Short time delays are accurate within a few nanoseconds depending on the utilized hardware. Additionally, it can generate simple timers without real time in millisecond resolution based on Windows mechanisms.

Features

- Detection of all operating time bases in the system
- Calibration of different timer/clocks
- Long-term synchronization of system clocks
- Monitoring the system time in different time and user-specific formats, resolution up to 0.1 µs
- Highly accurate short time delays in 0.1 µs steps
- Programming of user-specific time formats
- Reliable prevention of overflows caused by internal 96-bit operations
- Creation of simple timers in millisecond resolution based on Windows-mechanisms (no real time)

NEW! **RealTime Tasking Module**

Programming of high frequency real-time timer routines and priority-driven, preemptive real-time multitasking

Common

The RealTime Tasking Module complements the ›**Timer Module** and enables the programming of timers, which can trigger the application code in different ways. The context of the program execution can be selected. Signalable objects can be events or callbacks at application or kernel level, as well as real-time tasks. Also included is a priority-driven, preemptive real-time multitasking system for the mutual prioritization of single real-time tasks. This allows for detailed handling of which processes come prior to others, halting less important ones as well as which tasks can be interrupted by more important ones, which is equivalent to the mechanisms of a pure real-time operating system (RTOS).

Features

Real Time

- Development of high frequency real-time timer routines
- Cyclical or one-time timer programmable
- Simple debug and test option with callbacks at application level
- Option for timers to directly start, exit or kill
- Easy implementation of watchdog mechanisms
- Start time can be set in a 0.1 µs resolution (e.g. for synchronization with other processes)
- Timer frequencies of up to 20 kHz and more
- Jitter in the one-digit microsecond range
- Timer period dynamically adjustable

Multitasking

- Preemptive real-time system
- Real-time tasks with up to 255 different priority levels programmable
- Dynamic modification of priority levels
- Multiple tasks with the same priority level as “round-robin”
- Including priority inheritance to avoid priority inversion
- Real-time semaphores for synchronization between real-time tasks
- Real-time events for triggering of external tasks
- Tasks can be suspended, continued, triggered, exited and killed
- Tasks can be delayed (0.1 µs resolution)
- Operation of CPU cores exclusively in real-time mode in order to avoid Windows influence
- Very short task switch delays
- Speedloop mode for high-precision cyclical execution

Requires the ›**Timer Module**

PTP Module

Real-time synchronization with Precision Time Protocol

Common

The PTP Module allows for multiple network participants in a distributed system to be precisely synchronized. The Kithara real-time system enables the generating of precise timestamps with deviations in the sub-microsecond range, in order to match the clock of all participants. The API provides access to the BMCA (Best Master Clock Algorithm), which determines the Master clock according to configurable parameters. Alternatively, Master and Slaves can be set manually.

Features

- Precise synchronization of multiple network participants
- Accurate timestamps with deviations in the sub-microsecond range
- Configurable BMCA (Best Master Clock Algorithm)
- Manual setting of master and slaves
- Requires › **Network Module**

Extensions

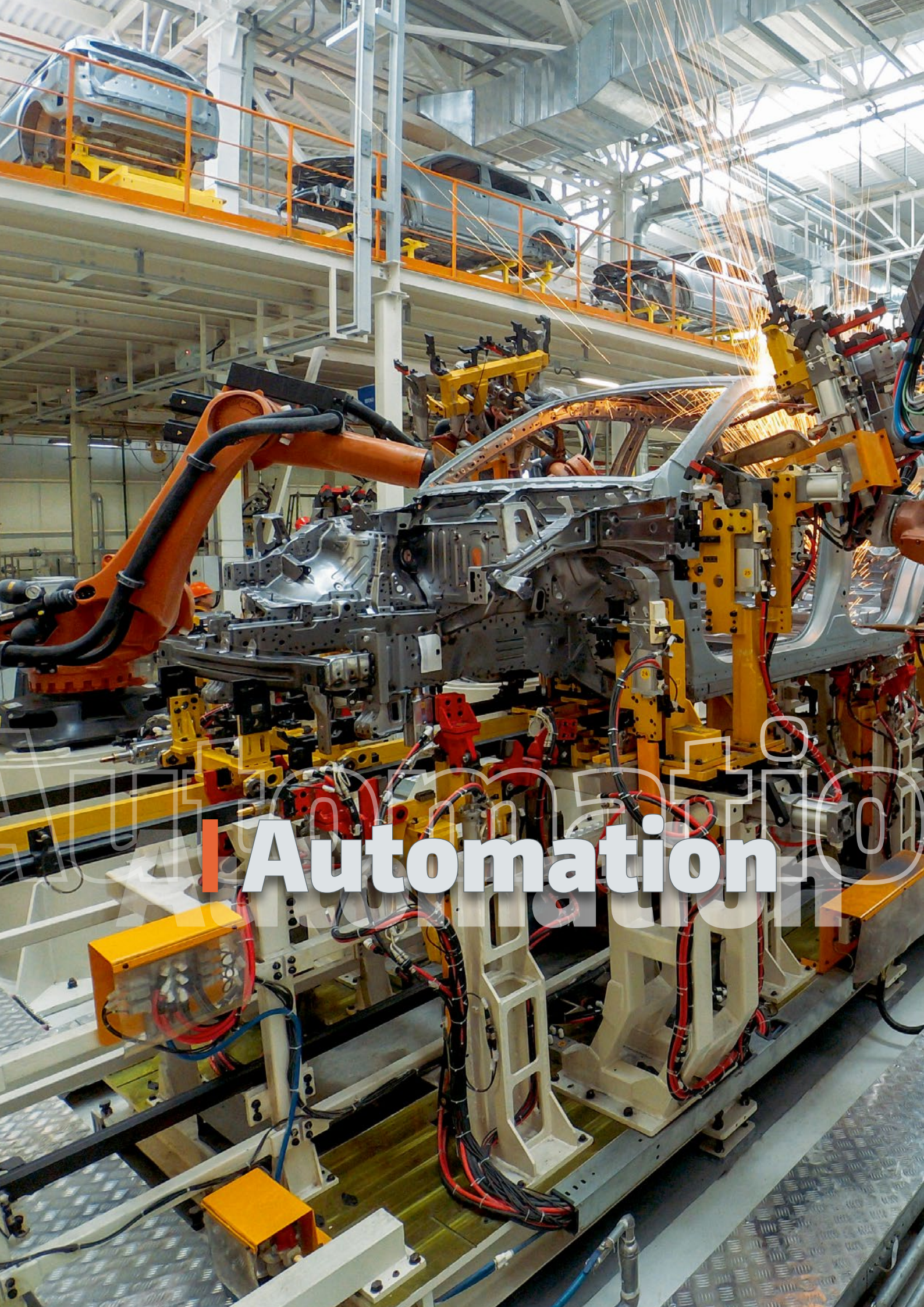
NMEA Extension

Real-time communication and synchronization with NMEA 0183.

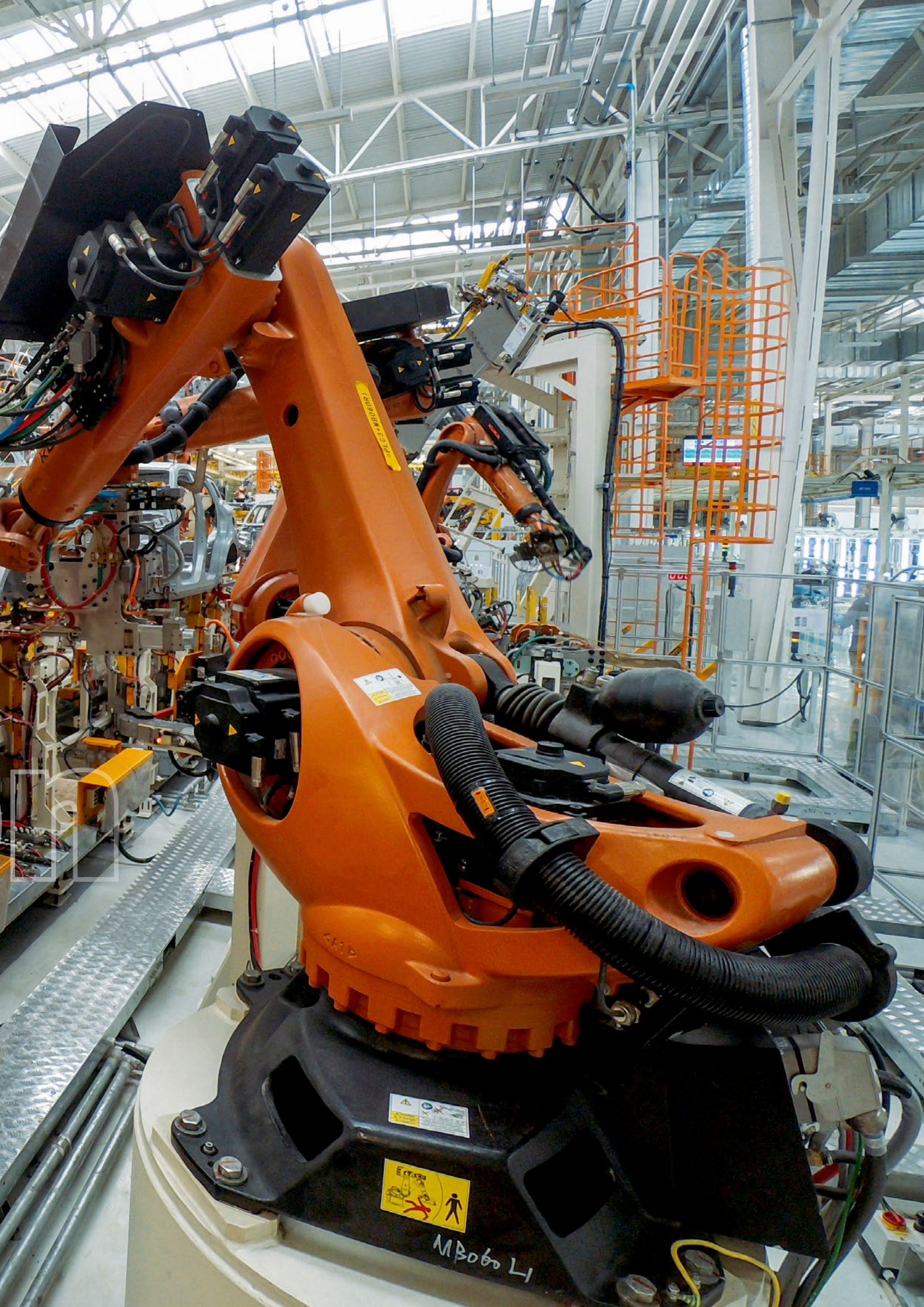
gPTP Extension

The gPTP Extension complements the PTP Module with support for the “generalized Precision Time Protocol” (defined in IEEE 802.1AS), a simplified PTP profile. The gPTP Extension allows for highly accurate synchronization of all clocks within a distributed system but limits itself to a more straightforward set of functions, making it suitable for a broader range of applications. For example, clocks do not have to be configured individually but are all standardized.

- Requires PTP Module
- Precise synchronization of multiple network participants
- Simplified PTP profile
- All clocks standardized
- Easier configuration of the BMCA (Best Master Clock Algorithm)



Automation


MB060 L1

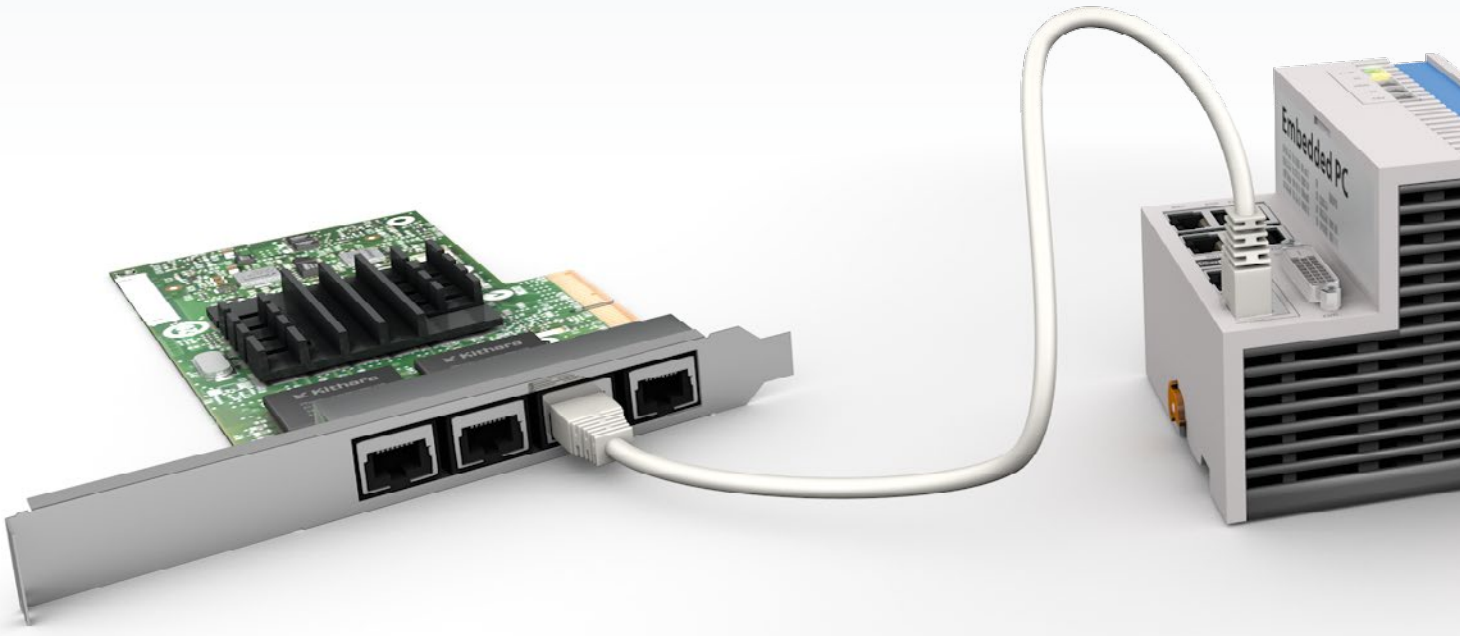
Software solutions for PC-controlled automation

In automation technology, standardized protocols have become mandatory by now. As one of the most popular field busses, the open and fast industrial Ethernet variant EtherCAT has established itself and has since enjoyed considerable growth. For this, Kithara RealTime Suite also includes a high-performance EtherCAT master.

Its “hard” real-time capabilities allow for plant automation with cycle times of only 50 µs or less. The master also provides all necessary mechanisms ranging from automatic determination of the connected topology, support for slaves as distributed clocks (DC) to special features such as hot-plug capability and cable redundancy.

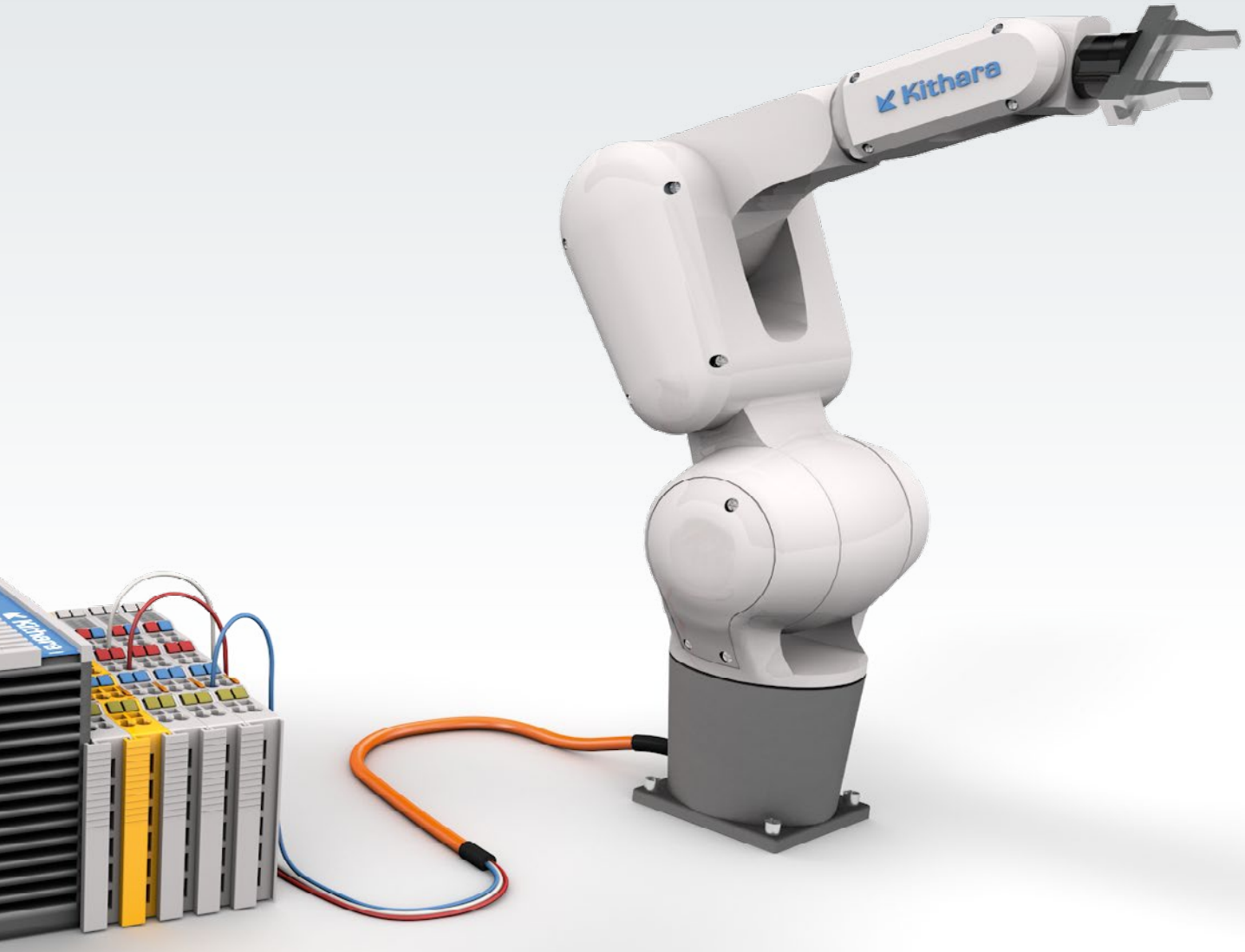
For the implementation of safety applications, the Safety-over-EtherCAT (FSoE) can be integrated as well. In this case, the EtherCAT master handles the data package exchange between safety input, output as well as safety logic components. This way, applications up to SIL3 can be implemented—an otherwise separate wiring, such as for an emergency deactivation, is not necessary.

Furthermore, the integration of PCs as EtherCAT slaves is possible in order to create flexible nodes for scalable EtherCAT networks. To achieve this, special PCIe cards can be used to embed PCs into EtherCAT topologies and operate them as slaves. Such EtherCAT PC slaves benefit from the graphical interface and input options of regular PCs and can be specialized for various tasks, such as the allocation or scaling of processing power.



Also available is the CANopen master which can be utilized to operate CAN interfaces or to embed into higher-level EtherCAT networks.

For embedding EtherCAT topologies into higher levels and even company networks, the EtherCAT Automation Protocol (EAP) is available. It enables devices and whole production segments of identical as well as different levels to exchange data with up to 100 Gbit/s. This allows not only for master-master communication, but also for the connection to higher-level production systems (MES, ERP) and departments (logistics, distribution etc.). This universal connecting of automated facilities makes EAP the base for the execution of cyber-physical systems.



Modules

EtherCAT Module

EtherCAT Master in real time

Common

Kithara EtherCAT Master is internally based on a priority-based, preemptive real-time multi-tasking system. Through utilization of hardware parallelism with multiple CPU cores, it is possible to disseminate several real-time tasks to CPU cores. Thus, a high scalability of real-time processing up to extreme real time on exclusively used CPU cores in dedicated mode is possible.

Features

- Independent EtherCAT Master in real time
- Automatic identification of EtherCAT topologies with XML files or SII protocol
- Process data communication (cyclic I/O data exchange) in real time
- Short cycle time: down to 50 microseconds or less
- Hot connect and cable redundancy
- Network interface from RealTek and Intel are supported
- I/O terminals, servo inverters etc. of all manufacturers are supported (e.g. Beckhoff)
- Modular structure of the EtherCAT Master: basis module + options
- Process data communication (PDO)
- Service date communication (SDO)
- Mailbox communication
- Requires ›**Network Module**

Extensions

EtherCAT PC Slave Device Extension

With the EtherCAT PC Slave Device Extension, a regular PC can be utilized as EtherCAT slave and be implemented into EtherCAT networks. Prior to the introduction of PCIe slave cards, only the master side was able to execute complex PC-based communication with the EtherCAT network. With the implementation of PC slaves, sophisticated automation processes can be conceptualized, integrated and adjusted with better focus.

- Common API with EtherCAT Master and EtherCAT EAP
- Process data and service data communication (PDO/SDO)
- File transfer (FoE)
- Creation of custom PDO mappings
- SII data (EEprom) can be initialized to custom values (e.g. Vendor ID, Product ID, Revision)

Supported Hardware

Beckhoff – EtherCAT Slave devices

- FC1100, PCI EtherCAT Slave Card
- FC1121, PCIe EtherCAT Slave Card
- CX5000 CCAT (Beckhoff CX50xx, CX20xx, CX51xx Embedded PCs)

ESD – EtherCAT Slave card

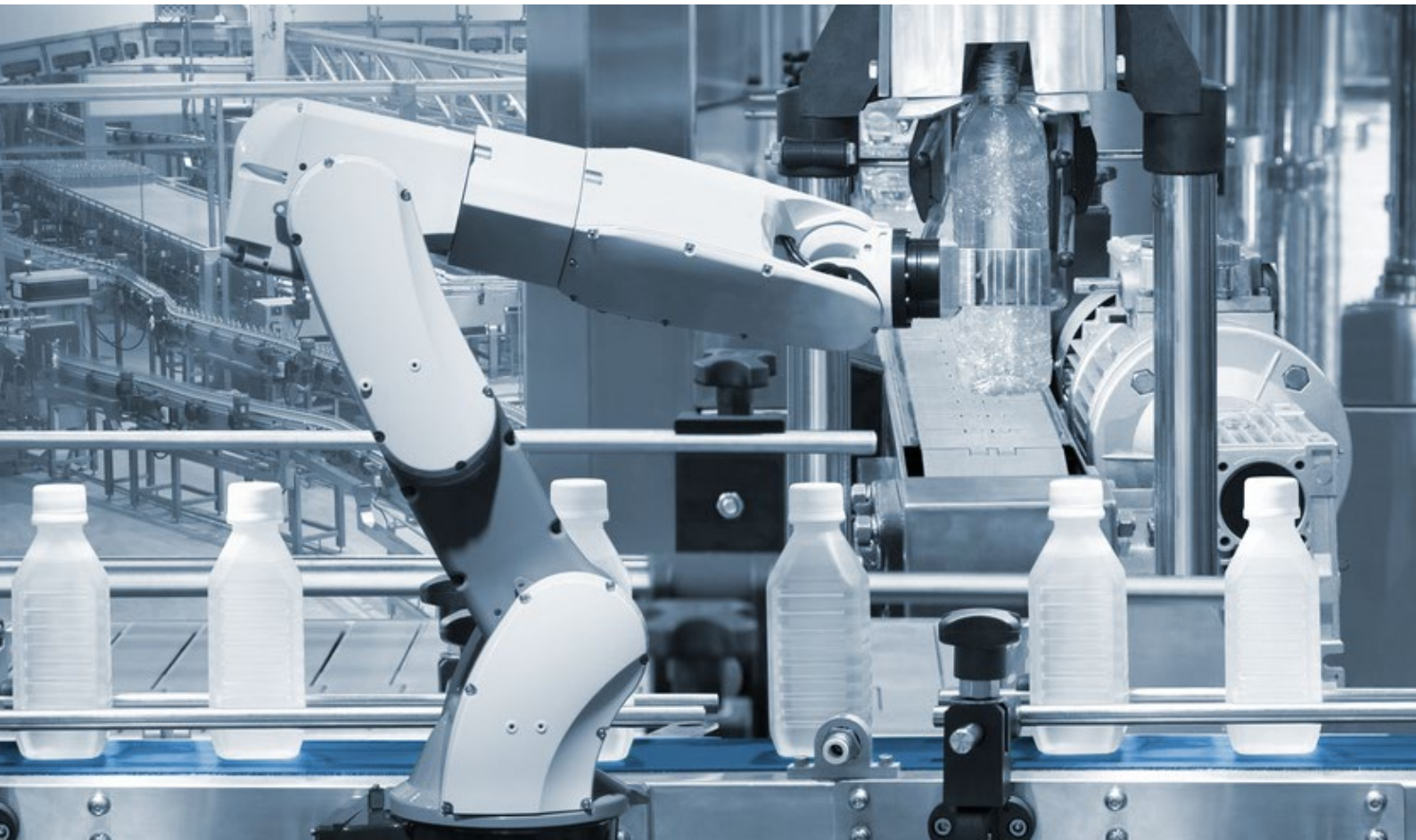
- ECS-PCIe 1100, PCIe EtherCAT Slave Card

Distributed Clocks (DC)

Distributed Clocks is used for software-based time synchronization between all bus devices. During initialization, the master determines delays to individual slaves and sets their clocks accordingly. For drift correction, a message is regularly sent to all slaves via the ring topology in order to synchronize them precisely over the long term.

Ethernet over EtherCAT (EoE)

With Ethernet over EtherCAT, any Ethernet device can be connected within EtherCAT via switch ports. Standard Ethernet communi-



cation is tunneled through EtherCAT, which enables the master to optimize it without affecting process data communication. Some slaves also provide a web interface, e.g. for configuration, which can be accessed via EoE.

File access over EtherCAT (FoE)

File access over EtherCAT allows for simple file access to network devices and can be used, for example, to upload uniform firmware to several devices in the EtherCAT network. The protocol is deliberately kept simple in order to also support bootloaders.

Servo Profile over EtherCAT (SoE)

The Servodrive Profile over EtherCAT provides support for the Sercos interface, which can be used to implement sophisticated motion control applications. The Sercos profile for servo drives and its mapping to EtherCAT are standardized in IEC 61800-7.

Hot Connect (HC)

Hot Connect allows preconfigured segments (slave groups or individual slaves) to be removed from or added to the data traffic before or during operation. This allows for flexible adjustments to the topology.

Cable Redundancy (CR)

With the cable redundancy functionality, operation continues even in the event of unintentional communication interruptions such as broken wires. For this purpose, the process data is sent redundantly via a second Ethernet port through the topology.

Safety over EtherCAT (FSoE):

The TÜV-certified Safety over EtherCAT protocol enables the functionality of EtherCAT to be implemented with safety level SIL 3 (in accordance with IEC 61508). The safety protocol does not cause any restrictions in terms of transfer speed or cycle time, since EtherCAT is used as a single-channel communication medium.

EtherCAT Autom. Protocol

Real-time communication over EtherCAT from the command level

Common

The EtherCAT Automation Protocol allows for real-time communication between all terminals within a network, which achieves a whole new level of connectivity in the field of automation systems. This can be all PC-based controls, machines, testing rigs, conveyor belts, robots, facilities for quality assurance or MES (Manufacturing Execution Systems).

Features

- Transfer rate of e.g. 1000 Mbit/s, 10 Gbit/s or 100 Gbit/s
- Process data and service data exchange (PDO/SDO), file transfer
- Requires ›**EtherCAT Module**

CANopen Module

CANopen Master in real time

Common

The CANopen Modules of Kithara RealTime Suite includes a PC-based master for the CANopen automation protocol. It is based on the CAN Module and requires appropriate hardware.

Features

- Automatic determnitation of CANopen topologies
- Management of CANopen slave states
- Process data and service data communication (PDO+SDO)
- Mailbox communikation
- Requires ›**CAN Module**

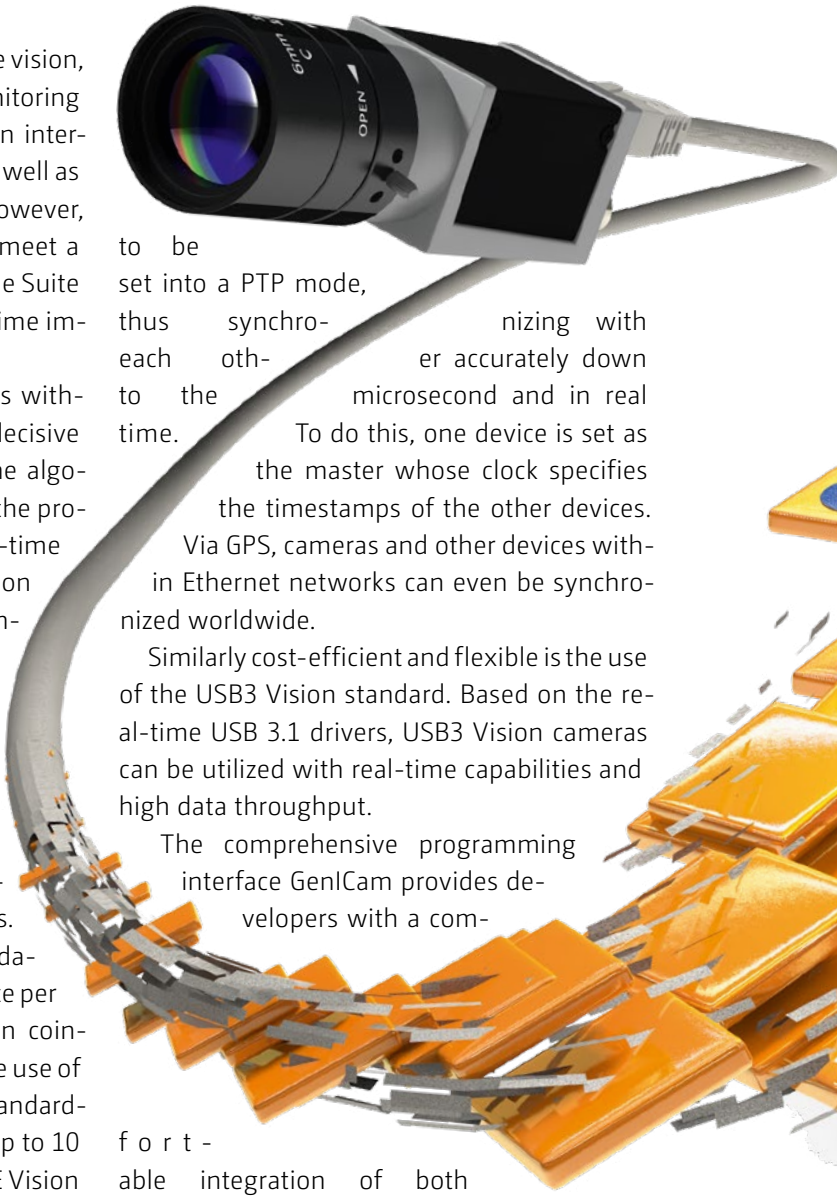
Machine Vision

Machine vision in real time

Many automated plants rely on machine vision, be it for manufacturing, packaging, monitoring or quality assurance. The applied vision interface standards for industrial cameras as well as the use of image processing libraries, however, require low reaction times in order to meet a variety of specific tasks. Kithara RealTime Suite is a leading pioneer in the field of real-time image capturing and processing.

The support of automation protocols within a closed real-time cycle offers the decisive benefit of transmitting the result of the algorithm-based image analysis directly to the process without having to leave the real-time context. This means that even systems on sensor/actor level can be directly controlled by processed image data.

For Ethernet-based camera systems, a real-time-capable GigE Vision driver is provided, which can respond to an incoming complete camera image within a few microseconds. Link aggregation is also supported, enabling parallel transmission on multiple channels. Depending on the deployed hardware, data rates can go up to more than a gigabyte per second. The advantages of GigE Vision coincide with those of Ethernet, meaning the use of cost-efficient and interchangeable, standardized hardware as well as data rates of up to 10 Gbit/s and long cable lengths. With GigE Vision 2.0, the Precision Time Protocol (IEEE 1588) has been introduced. It allows for multiple cameras to be set into a PTP mode, thus synchronizing with each other accurately down to the microsecond and in real time. To do this, one device is set as the master whose clock specifies the timestamps of the other devices. Via GPS, cameras and other devices within Ethernet networks can even be synchronized worldwide.

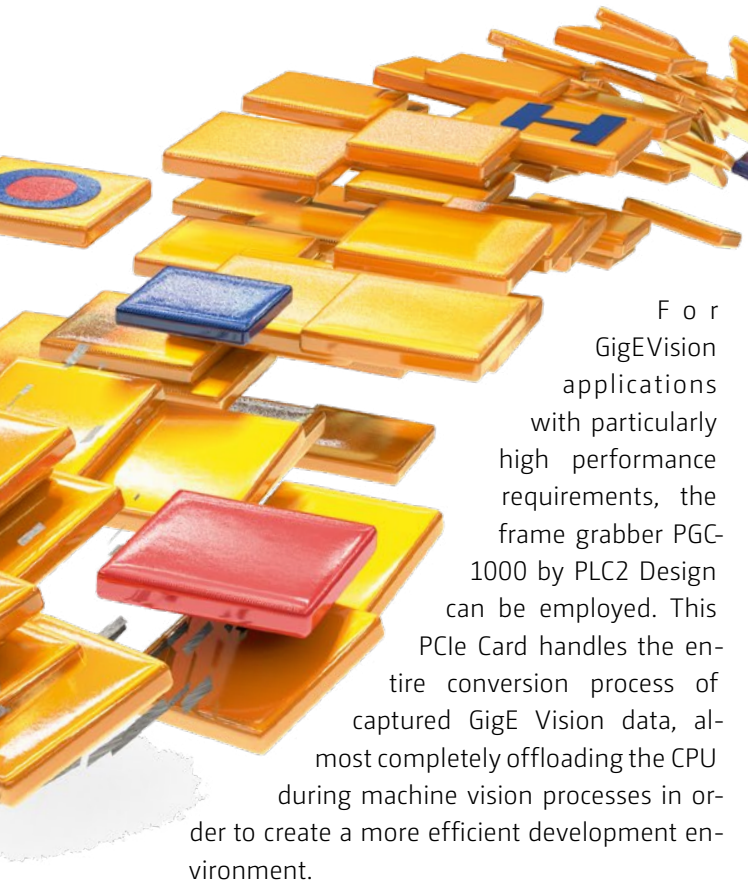


Similarly cost-efficient and flexible is the use of the USB3 Vision standard. Based on the real-time USB 3.1 drivers, USB3 Vision cameras can be utilized with real-time capabilities and high data throughput.

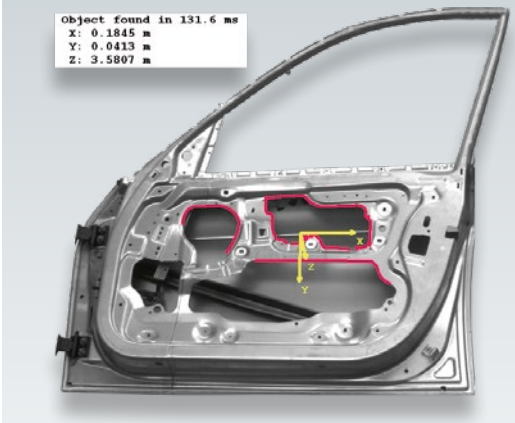
The comprehensive programming interface GenICam provides developers with a comfortable integration of both standards. With optional hardware, even CameraLink can be supported.

By deploying an external frame grabber, which converts CL control information and data packages to GigE Vision, all CL configurations including “Full” can be integrated via a 10 gigabit Ethernet connection into the real-time environment. This also increases the previously short cable lengths to Ethernet levels. For immediate further processing and analysis of image data, certain algorithms

or external image processing libraries such as Halcon or OpenCV are usable. In order to do so, the respective library is loaded into the real-time context. When using Halcon, procedures created with the development environment HDevelop can be executed directly in real time.



For GigEVision applications with particularly high performance requirements, the frame grabber PGC-1000 by PLC2 Design can be employed. This PCIe Card handles the entire conversion process of captured GigE Vision data, almost completely offloading the CPU during machine vision processes in order to create a more efficient development environment.



For manufacturing and quality assurance, especially in the automotive field, the use of camera systems has increased significantly. This often includes high-precision recognition and accurate interpretation of captured images. Without “hard” real-time capabilities for image capturing and processing, this would not be feasible. Kithara provides the necessary technical basis for these tasks.

Modules

Camera Module
Vision Module

Camera Module

Image capturing with GigE Vision and USB3 Vision cameras in real time

Common

The Camera Module serves the purpose of capturing image data of industrial cameras according to GigE Vision or USB3 Vision standards in real time. It provides all necessary functions for detection and management and for creating real-time streams.

Large buffer memory along with real-time network drivers reliably prevent any loss of data packets. The reaction to incoming GigE Vision or USB3 Vision images takes place instantly in the context of the real-time environment within a few microseconds. An immediate real-time task can be triggered, in order to execute an instant control reaction.

Thanks to the real-time drivers, the Camera Module allows for maximal data throughputs according to interface and system configuration, even up to the theoretical maximum. For the processing of image data see **› Halcon Extension** und **› OpenCV Extension**.

The results of the image processing can be given to the process even in the real-time context without any delay, e. g. over EtherCAT or CANopen. Accordingly, automation solutions can be easily combined with several other functions within Kithara RealTime Suite.

Features

General features

- Image acquisition with GigE Vision or USB3 Vision cameras in real time
- Event or request based
- Multiple simultaneous cameras, hot-plugging, configuration in accordance with GenICam 2.0
- Every GigE Vision compliant camera is supported

Control

- Automatic connection handling
- Full access to camera configuration space
- Read and write camera memory
- Set custom heartbeat timeout
- Start and stop acquisition of images by software
- Set acquisition mode by software
- GenICam-XML configuration file automatically downloaded
- Access to GenICam features to configure your camera
- Error handling

Streams

- One or multiple streams per camera possible
- Acquisition of images in real time
- Memory-efficient image capture, no unnecessary copying
- Set your own buffer size and buffer count
- Provides information about missed or incomplete images

GenICam features

- Numbers all categories
- Set and retrieve all accessible camera features like image format, pixel format, frame rate, exposure time and more

The Camera Module requires additional modules for the real-time interface, such as the **› Network Module** for GigE Vision and/or the **› USB RealTime Module** for USB3 Vision.

Extensions

Camera Acceleration Extension

The Camera Acceleration Extension provides efficient real-time drivers for the GigE Vision framegrabber acceleration card PGC-1000 by PLC2. Embedded in Kithara RealTime Suite, the PGC-1000 is able to almost completely offload the CPU for image acquisition processes within machine vision applications. This allows, for instance, to simultaneously capture up to four GigE Vision camera streams each with 10 Gbit/s in a real-time context. The real-time synchronization of multiple cameras via the PTP Module by Kithara is also possible.

- Real-time support for PGC-1000
- Almost completely offloads the CPU for GigE Vision image acquisition
- Connection speeds of up to 4×10 GBit/s or 64×1 GBit/s
- Synchronisation of multiple cameras via PTP Module
- Requires › **Camera Module** and › **Network Module**
- Requires PGC-1000 by PLC2 Design GmbH

Vision Module

Real-time image processing with Halcon or OpenCV

Common

Halcon and OpenCV are programming libraries with algorithms for image processing and machine vision, integrated into the Kithara real-time system. They allow for immediate control reactions of processed images and their implementation into automation systems such as EtherCAT or CANopen.

Features

- Utilization of image processing libraries such as Halcon or OpenCV in real time
- The Halcon license needs to be acquired separately from MVTec
- Supports Halcon versions 10 to 13, 18.11, 20.11 as well as OpenCV 3.0, 3.4 and 4.1
- Includes › **Halcon Extension** and › **OpenCV Extension**
- Requires › **RealTime Tasking Module** und › **Library Module**

Communication

Hardware access and communication in real time

Several modules of Kithara RealTime Suite are used for the connection of PC hardware. Via access to I/O registers of a PC, the physical memory as well as interrupt handling, drivers for individual hardware components and plug-in cards can be created. For many common communication interfaces, efficient, ready-to-use drivers are already available.

Especially in industrial fields, the significance of Ethernet applications with high time-critical requirements has increased substantially. The specially developed drivers for this field support all common Ethernet network controllers by Intel and RealTek with data rates of up to 100 Gbit/s. All functions, traits and benefits of Ethernet, such as the use of flexible and cost-efficient standard hardware, can be utilized as usual but also complemented with real-time capabilities. In this manner, special hardware features of applied controllers, such as jumbo frames, advanced data flow control due to interrupt optimization, can be used effectively.

In order to achieve real-time capability, network controllers are accessed directly. This way, the whole bandwidth of a connection is available and the reaction to incoming data packages takes place immediately. In order to guarantee high-performance network communication, all redundant Windows mechanisms that otherwise would impair the transmission rate or negatively impact immediate reactions, are bypassed. For the implementation of socket communication, a special driver layer for datagram (UDP) or stream-oriented (TCP) communication applications is provided.

For fast communication with USB up to version 3.1, Kithara provides a specially programmed, easy-to-apply access to the Extensible Host Controller Interface (xHCI). Thanks to its cost-efficient implementation with stan-

dardized hardware combined with good transfer rates, USB has become one of the most widespread communication standards. The SuperSpeed mode, introduced with USB 3.0, which achieves even higher data rates and full-duplex communication, has since enabled USB to be applied productively even in advanced industrial fields.

Thunderbolt can also be used in real time, allowing for an immediate communication of components, that are connected via this interface at the lowest possible delay. This Thunderbolt implementation also enables the use of appropriate hardware extensions in real time such as for connecting PCIe cards to laptops. This helps improve the mobility of software-based real-time solutions, for example when testing or measuring directly at the machine on site.

Furthermore, serial UART and COM interfaces can be addressed as well, supporting the common standards RS-232, RS- 422 and RS-485; with baud rates of 115.200 and 921.600 bit/s. This also includes running UART functions from real-time tasks on dedicated CPUs as well as directly controlling handshake and signal lines.

A special additional programming interface allows for the handling of operating system function calls for the communication of device drivers at kernel level. Even “virtual” COM interfaces, for example, can be generated this way. Furthermore, we also support a range of high-performance multifunction cards by National Instruments, which can be accessed via digital and analog input/output. Prime examples for the practical application of time-critical hardware communication, among others, are machine vision with industrial cameras (GigE Vision, USB3 Vision) or sophisticated automation tasks with EtherCAT.

Modules

Network Module

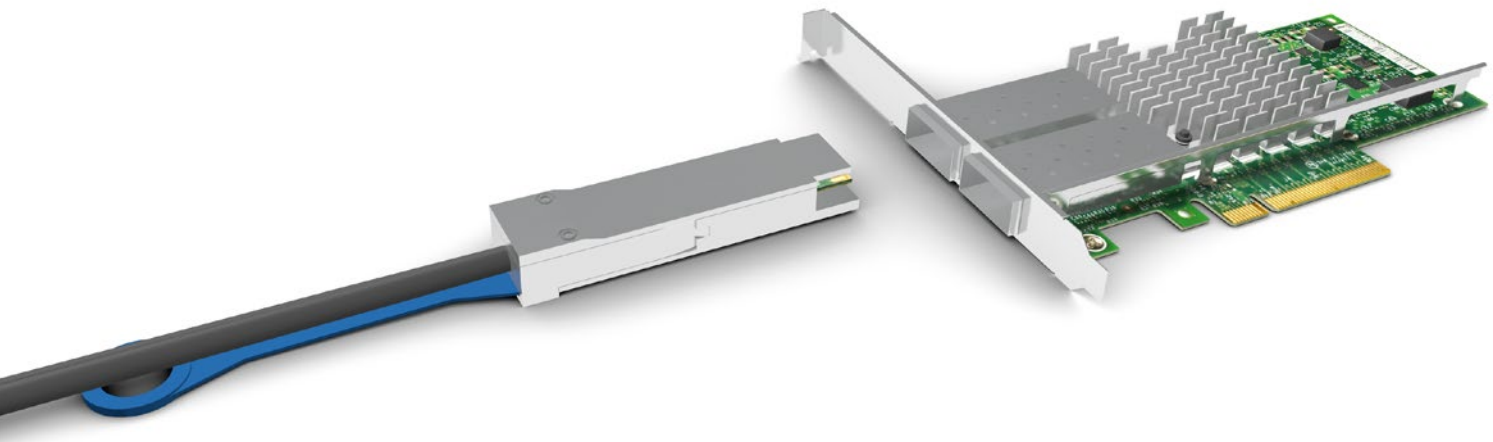
Native real-time driver for fast Ethernet communication

Common

The Network Module provides native real-time Ethernet drivers for the supported network controller, for communication via raw Ethernet as well as UDP and TCP. By preempting all interferences of Windows mechanisms, high transmission rates and immediate reaction are possible. This leads to a high performance network communication. Almost all network controller of Intel and RealTek are supported (see: Supported Hardware).

Features

- Ethernet communication up to 100 Gbit/s in real time
- Instantaneous transmission and reaction at receiving
- Transmission and receiving straight from the real-time context possible
- Raw Ethernet as well as various Ethernet protocols, including IP, UDP and TCP
- Event or request-based Ethernet communication
- Jumbo-Frames up to 16128 bytes depends on NIC
- IP and MAC multicast, broadcast, promiscuous mode for receiving of all packets
- Receiving of data packets either trough Callback functions or directly at the interrupt context or polling process
- Up to 4 priority levels possible for transmission of data packets
- Automatic determination of memory addresses through ARP support
- Functions for CRS calculation and byte order conversion
- Network Module Addons: Network Port Extension, ›**Network PTM Extension**, Network Data Rate Extension, Network BroadR-Reach Extension
- Requires ›**RealTime Tasking Module**



Supported Hardware

In order to achieve real-time capabilities, the standard Windows driver of the network card has to be replaced by the special developed real-time driver of Kithara RealTime Suite. Nearly all available network controllers by Intel and RealTek are supported.

100 Gbit/s

Intel
- E810

40 Gbit/s

Intel
- XL710

25 Gbit/s

Intel
- XXV710

10 Gbit/s

Intel
- 82598, 82599
- x540, x550
- X710

1 Gbit/s

Intel
- 82540, 82541, 82544, 82545, 82546, 82547
- 82566, 82567
- 82571, 82572, 82573, 82574
- 82575, 82576
- 82577, 82578, 82579
- 82580, 82583, i350
- i210, i211, i217, i219

RealTek
- 8100E, 8101E, 8102E, 8110S
- 8168B/8111B, 8168C/8111C, 8168CP/8111CP, 8168D/8111D+DL
- 8168DP/8111DP, 8168E/8111E, 8168EVL/8111EVL
- 8168F/8111F, 8168G/8111G, 8168H/8111H
- 8168EP/8111EP
- 8169, 8169S, 8169SB/8110SB, 8169SC/8110SC

ASIX – USB 3.0 Gigabit Ethernet Controller
- AX88179 (beispielsweise Digitus USB 3.0 Ethernet Adapter)

100 Mbit/s

Intel
- 8255x
- 82562

RealTek
- 8139B, 8139C, 8139D

Beckhoff
- CX5000 CCAT(Beckhoff CX50xx, CX20xx, CX51xx Embedded PCs)

Extensions

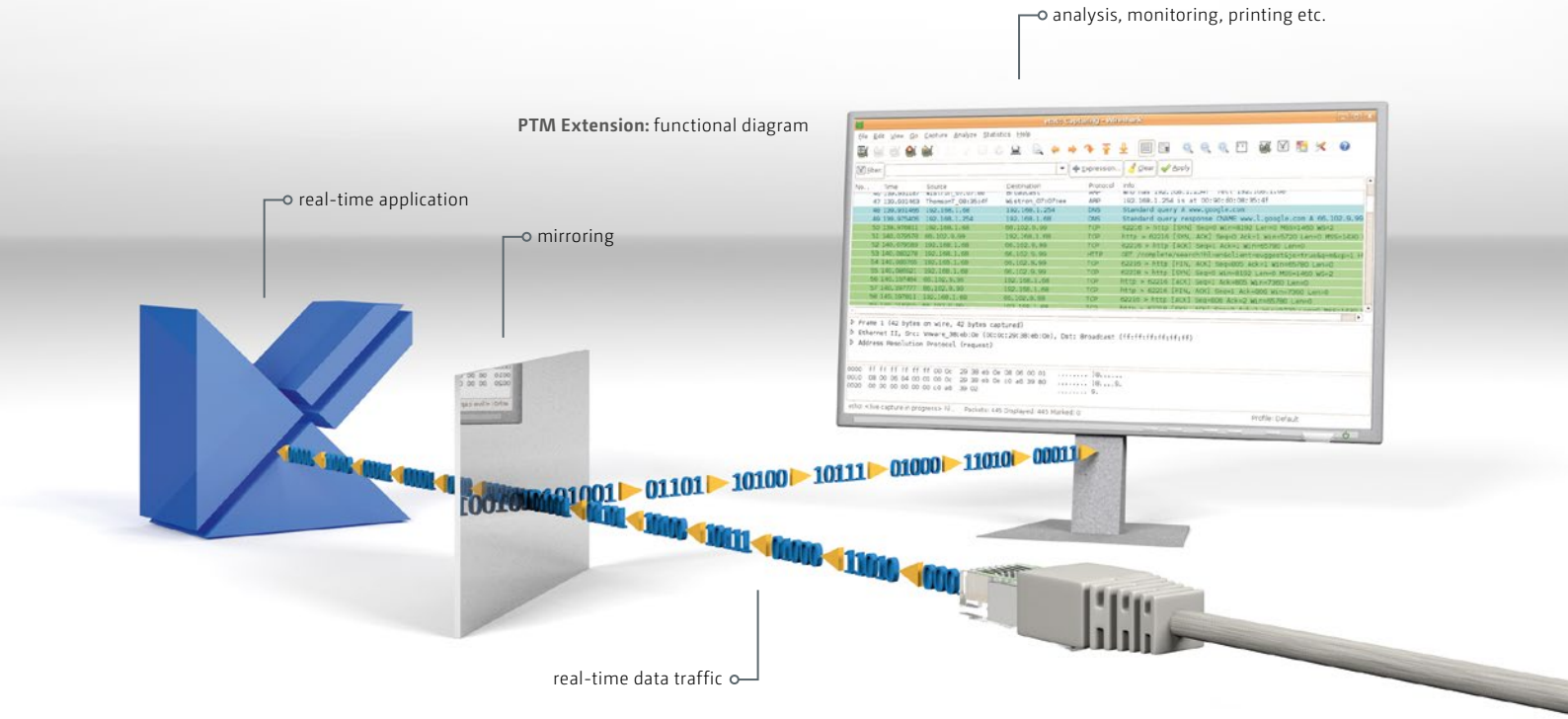
Network PTM Extension

The Network PTM Extension allows for the generation of virtual USB-Ethernet adapters via UDE (USB Device Emulation) for the communication between real time and Windows with port passthrough or port mirroring. Passthrough enables Windows to directly communicate with the Hardware or the Kithara real-time environment, whereas mirroring is used for logging real-time data traffic and providing it to Windows for analysis.

- Creation of virtual USB-Ethernet adapters via UDE
- Port passthrough for direct communication between Windows and Hardware or real time
- Port mirroring for logging of real-time data traffic and displaying in Windows with Wireshark
- Supports all Ethernet-based technologies: socket communication, EtherCAT, GigE Vision, PTP and BroadR-Reach

Network BroadR-Reach Extension

BroadR-Reach (100BASE-T1) is a pyhsical Ethernet transport layer specially for automotive networking.



PTM Extension: functional diagram

USB RealTime Module

Real-time communication with USB devices via direct xHCI access

Common

The USB RealTime Module of Kithara RealTime Suite ensures that all USB-based interfaces and devices, such as cameras, can be used with fast reaction times at the highest possible data rates. For this purpose, Kithara RealTime Suite provides a specially developed and easy-to-apply gateway to the Extensible Host Controller Interface (xHCI). The basic operating principle is to directly access the xHCI controller exclusively within the real-time context.

Features

- Real-time communication up to USB 3.1 devices via xHCI access
- xHCI controller used exclusively
- Direct access from the real-time multitasking context
- Lowest possible reaction time
- Highest possible data throughput
- Reliable prevention of data loss due to customizable buffer sizes
- Low-, full-, high-speed and super-speed
- Control-, bulk-, interrupt- and isochronous transfer
- Response to all plug and play events as well as power management events
- Requires **›RealTime Tasking Module**

Supported Hardware

Concerning the USB RealTime module, all USB 3.1 host controller can be used, that are compatible to the xHCI 1.1 specification.

xHCI controller of the following vendors are currently supported by Kithara RealTime Suite: Intel, AMD, Renesas, Etron, NEC and VIA.

USB Driver Module

Communication with USB devices via Windows driver stack

Common

The USB Driver Module of Kithara RealTime Suite enables the communication with USB devices over simple adaptable functions. It is based on the Windows USB driver stack with only limited real-time capabilities. For “hard” real time, we recommend the **›USB RealTime Module**. All USB devices are supported, including USB 3.1.

Features

- Development of USB device drivers
- Access to internal or external USB devices (including USB 3.1) from application or kernel level
- Low-, full-, high-speed
- Control-, bulk-, interrupt- and isochronous transfer
- Response to all plug and play events and power management events

Serial/UART Module

Easy-to-apply functions for fast communication

- Serial communication in real time by special hardware drivers on UART16550 compatible hardware
- Signal and handshake lines can be controlled directly
- Handler for all interface events in real time
- Alternative API for all Windows-based COM ports provided without real time

Common

The Serial Module of Kithara RealTime Suite provides easy-to-apply functions for fast communication via the serial UART as well as COM interfaces. The UART functions require appropriate UART-16550-compatible hardware as well as a real-time driver to be present. In real-time mode, the UART functions can also be used from real-time tasks as well as on dedicated CPUs.

For COM communication, any interface for which a Windows driver is installed, is supported. In this case the COM functions run via Windows mechanisms, which means real-time capabilities cannot be applied.

Features

- Serial communication in real time at kernel level by utilizing a special hardware driver
- Handshake lines and signal lines directly controllable
- Handler for all interface events can also be registered in a real-time context and on dedicated CPUs
- Regarding UART functions, UART-16550 compatible hardware is necessary
- Regarding COM functions, serial communication on application level allows for using all COM interfaces in the system

To achieve real-time capabilities, special real-time drivers of Kithara RealTime Suite are required for applied serial interfaces. The following interface cards are currently supported:

- Standard COM Interface (PNP0501)

PCIe

- StarTech PEX2S952, 2 × RS232
- Longshine LCS6321O, 2 × RS232
- Delock 89220, 2 × RS232
- Delock 89236, 1 × RS232
- ADDI-DATA APCIe-7300, 1 × serial
- ADDI-DATA APCIe-7420, 2 × serial
- ADDI-DATA APCIe-7500, 4 × serial
- ADDI-DATA APCIe-7800, 8 × serial

ExpressCard

- i-tec EXRS232, 1 × RS-232
- StarTech EC1S1P55254, 1 × Serial, 1 × Parallel (Netmos 9912 chipset)

PCI

- MOXA 4-port RS-232 C104H/PCI
- MOXA 8-port RS-232 C168H/PCI
- MOXA 2-port RS-422/485 CP132/PCI
- MOXA 4-port RS-422/485
- EXSYS 41052 serial card
- EXSYS 43092 serial card
- Oxford Serial Card Quad Port
- SUNIX 2-port RS-232 parallel
- SUNIX 2-port RS-232 PCI
- Meilhaus ME9000 8-port RS-485
- EXAR (XR17C152) 2-port UART PCI
- ADDI-DATA APCI-7300-3, 1 × serial
- ADDI-DATA APCI-7420-3, 2 × serial
- ADDI-DATA APCI-7500-3, 4 × serial
- ADDI-DATA APCI-7800-3, 8 × serial

CardBus

- Socket PCMCIA serial card
- Delock PCMCIA serial card

Device Module

Generation of virtual interfaces

Common

The Device Module of Kithara RealTime Suite provides the end-user with an API for device drivers via operating system functions. The Device Module allows for the handling of such function calls at kernel level in the context of the device driver.

The Device Module also allows for the generating of virtual interfaces–in a special case even as virtual COM interface.

Features

- Provides a Windows API for device communication (ReadFile, WriteFile, DeviceIoControl)
- Device names freely configurable, for example virtual “COM ports”

Hardware Access

IoPort Module

Direct access to I/O-Ports

Common

The IoPort Module of Kithara RealTime Suite enables a direct access to the I/O ports of the PC immediately out of a Windows application. With the help of the IoPort Module users can unlock a direct access to the required I/O ports.

The direct access to I/O ports in no way means reducing the stability. A separate development of the kernel driver is not necessary, which allows for the user to better focus on the application itself while accessing hardware only over defined entry points.

Features

- Access to all I/O registers of the PC from application or DLL
- Determine PCI configuration data
- Retrieve resource information of devices

Memory Module

Access to physical memory

Common

The Memory Module of Kithara RealTime Suite enables a direct access to the physical memory.

In order to allow direct access, the physical memory is displayed within the application space. The Memory Module provides two different mechanisms to access physical memory. A distinction is made between wether the memory is located on external hardware or on the PC main memory.

Features

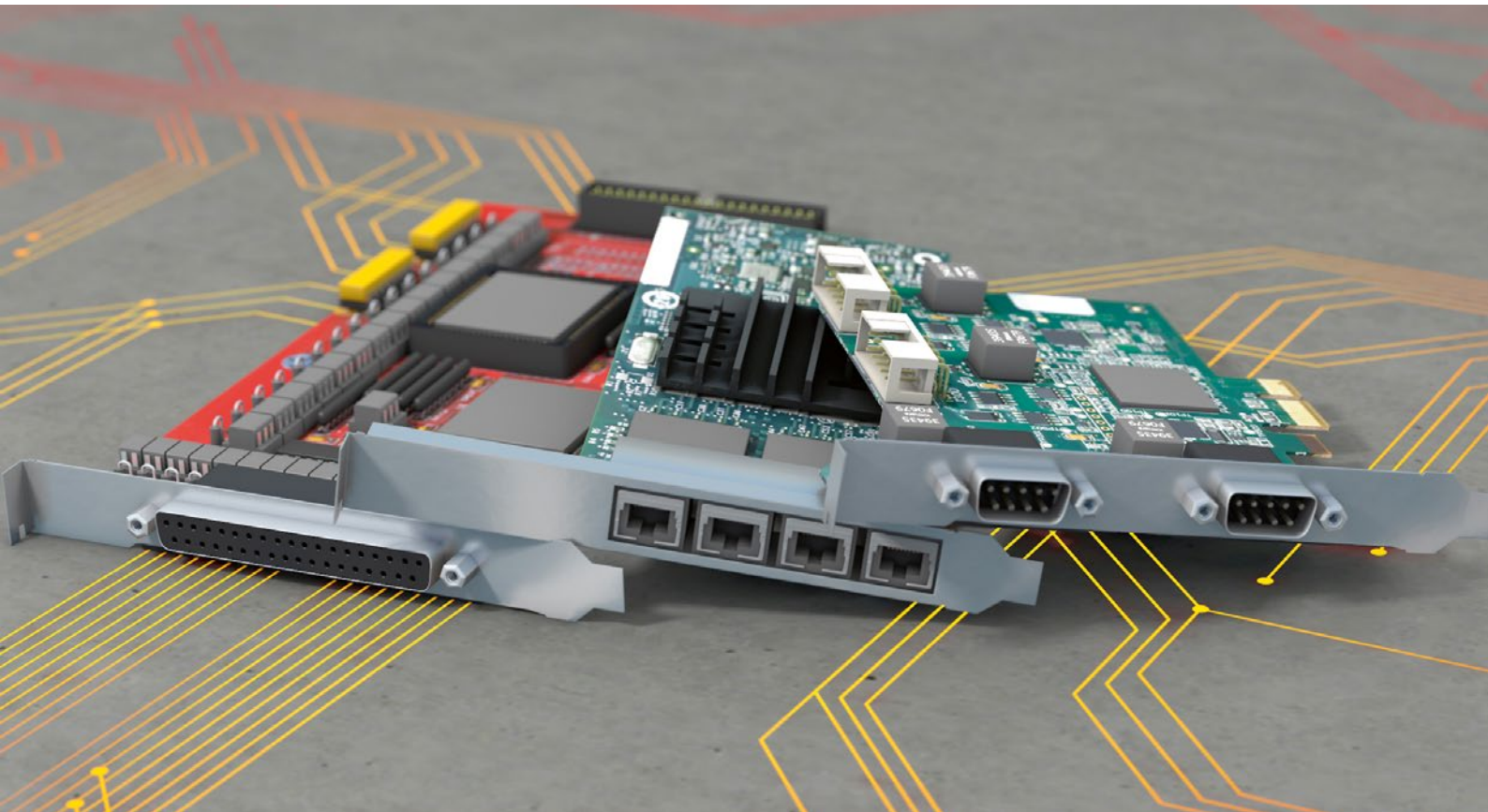
- Access to external physical memory (Dual-Port-RAM)
- Provision of PC memory for external hardware (DMA memory)

Interrupt Module

External hardware control

Common

In order to control external hardware, interrupts are necessary. They are used, for example, to communicate states to PC expansion cards. To register such events, device drivers and technical applications require the installation of an appropriate interrupt handler.



Features

- Handling of hardware interrupts at application or kernel level
- Use of operating system mechanisms for interrupt handling
- By using the ›**RealTime Tasking Module**, real-time interrupts on PCI or PCIe hardware can be applied
- Requires ›**IoPort Module**

MultiFunction Module

Access to multifunction expansion cards in real time

Common

Support of multifunction cards with a vendor-neutral API, digital IO bit wise or word wise, analog I/O as single value, channel sequence, limited series of sequences or continuous mode with a switch buffer interrupt.

Features

- Support of multifunction cards with a vendor-neutral API
- Functions for A/D-, D/A- and digital I/O for controlling and data acquisition
- Digital I/O bit wise or word wise
- Analog I/O as single value, channel sequence, limited series of sequences or continuous mode with a switch buffer interrupt
- Real-time drivers for cards from several manufactures
- Multifunction cards with PCI or PCIe bus
- Requires ›**RealTime Tasking Module**


Automotive

Automotive

Automotive technologies have seen rapid advancements in the past few years. The software part in these technologies, such as for driving safety, driving comfort and driving assistance, has been consistently increasing, as is computer technology itself as well as standardized bus systems for the connection of individual components. The most effective bus systems for this field are CAN, CAN FD, FlexRay, BroadR-Reach and LIN, for which Kithara provides a series of real-time functions to support the development of such software-based automotive systems.

CAN (Controller Area Network) is one of the most successful bus systems in automotive networking due to its universal and flexible applicability. Further, the extension CAN FD (Flexible Data Rate) allows for higher data rates and more information per message.

Thanks to high data rates via two channels, FlexRay is well suited for even more data intensive driving functions and is often applied in safety-critical applications alongside CAN.

The UART-based LIN bus, on the other hand, allows for the development of more cost-efficient single wire networks for rather simple functions within automotive components and is often integrated into higher-level systems such as CAN.

For many manufacturers, Ethernet is the next step for the future of vehicle wiring. The cutting-edge BroadR-Reach is an Ethernet transmission layer which was developed with special focus on the automotive field. Its advantages are particularly high transfer speeds, high scalability as well as cost efficiency regarding wiring and programming.

The real-time capability of these busses is especially relevant in product development and integration, industrial manufacturing as well as for test system engineers in the quality control of integrated mechatronic systems by enabling direct PC-based control and diagnosis.

Kithara RealTime Suite with its modular system provides a variety of mechanisms making it the perfect solution for such tasks.

CAN/CAN FD

The CAN Module enables real time for the field bus CAN (compatible with CAN-2.0B protocol). The API is vendor-independed and thus only requires a one-time initial training in order to program any type of supported CAN hardware.

The immediate response to incoming messages, for example, also allows for a callback functions react instantly to error events. Due to large transmitting and receiving buffer sizes, no packet loss occurs even at fast baud rates and high bus load. Additionally, it is possible to create your own filter routines which can be executed directly at receiving time.

In case of analysis tasks where the CAN interface needs to behave completely passive, a listen-only mode can be used. Furthermore, by using hardware with SJA1000 CAN controllers, the “Error Code Capture” register may be read out to acquire additional information for error analysis.

Flexray

As part of the Kithara real-time extension, the FlexRay Module enables the implementation of real-time capabilities with accurate cycle times for high-precision procedures, making it an essential link between test software and test bench. The module can be used to turn a Windows PC into a full-featured FlexRay node.

This includes the utilization as leading or following node, i. e. as active or passive communication point within the network. Both FlexRay channels can be configured independently from each other and the baud rate may be set flexibly (2,5/5/10 Mbit/s).

BroadR-Reach

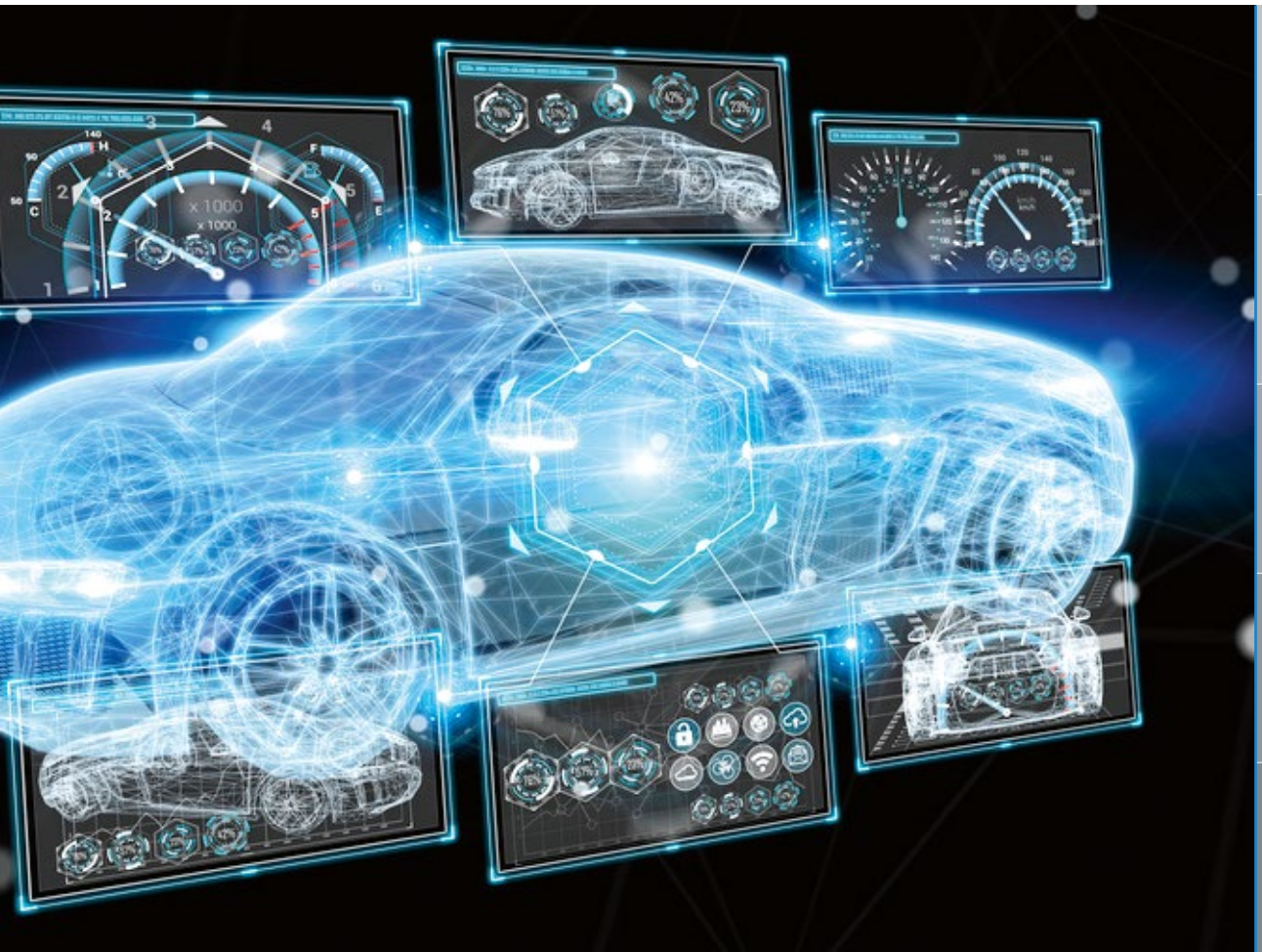
With real-time capabilities for the BroadR-Reach standard (100BASE-T1, 1000BASE-T1), automotive developers can directly control and test physical Ethernet networks within vehicles. With guaranteed reaction times provided by the Kithara real-time system, data within the Ethernet transfer layer can be accurately gathered and embedded into sophisticated testing processes. This bridges the gap between the high-performance Ethernet drivers of Kithara RealTime Suite and modern vehicle networks.

Many of the Kithara real-time functions are applied in the development of cutting-edge automotive technologies. They are, for instance, used for hardware-in-the-loop applications where simulation data needs to be precisely captured and reproduced without any data loss, which can only be achieved with real time.

LIN

By using the LIN Module of Kithara RealTime Suite it is possible to attain real time with LIN. A master or slave node can be created easily using a common PC with a UART interface. Unlike the case of using a microcontroller, this allows

for easier logging of existing LIN networks as well as more efficient debugging procedures. Sending nodes will automatically recheck their transmitted data on a bus, which enables the detection of collisions.



CAN

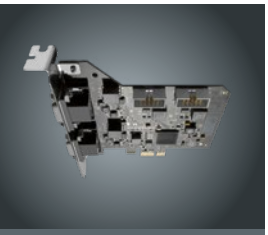
CAN FD

LIN
LOCAL INTERCONNECT NETWORK

FlexRay

BroadR-Reach

Modules



FlexRay Module

FlexRay communication in real time

Common

As part of the Kithara real-time extension, the FlexRay Module represents an essential link between test software and physical test system, which allows for the application of real-time capabilities with accurate cycles for high-precision operations.

With the FlexRay Module, a Windows PC can be utilized as an actual FlexRay node. This includes the assignment as leading or following node, as active or passive point of communication within the network. Both FlexRay channels can be configured independently from each other while it is possible to flexibly adjust their baud rate (2,5/5/10 Mbit/s).

Features

- Real-time FlexRay communication
- Independent configuration of FlexRay channels
- Windows PC used as FlexRay node
- Assignment as leading or following-cold-start node
- Baud rate flexibly adjustable (2,5/5/10 Mbit/s)
- Requires **›RealTime Tasking Module**

Supported Hardware

The following FlexRay cards are supported:

- FlexCard PMC II (Star Cooperation)

CAN Module

CAN communication in real time

Common

The CAN Module allows for real-time capabilities for the serial bus system CAN (Controller Area Network) up to and including CAN 2.0B. It enables developers within the Kithara real-time system to transmit and receive CAN messages without delay. Custom filter routines can be triggered by callback functions. Signaling of received messages via callback or the polling of messages is possible as well. Send and receive routines can be executed directly from the the real-time context.

Additionally, the CAN extension CAN FD (Flexible Data Rate) is supported as well, which, by extending the user data length from 8 to 64 bytes, allows for up to 8 times the data rate. Furthermore, due to improvements of test values, the potential data security of transmission increases, thus expanding possible implementations for safety-critical applications. For analyzing tasks, where the CAN

interface has to remain completely passive, a “listen-only” mode is available. Also, the “Error Code Capture” register can be read out for hardware with SJA1000 CAN controllers in order to provide additional information for debugging.

Features

- Real-time communication with CAN up to 2.0B
- Optional support for CAN FD (Flexible Data Rate)
- Delay-less sending and immediate response on reception with callback function or asynchronous message polling
- Requires **›RealTime Tasking Module**

Supported Hardware

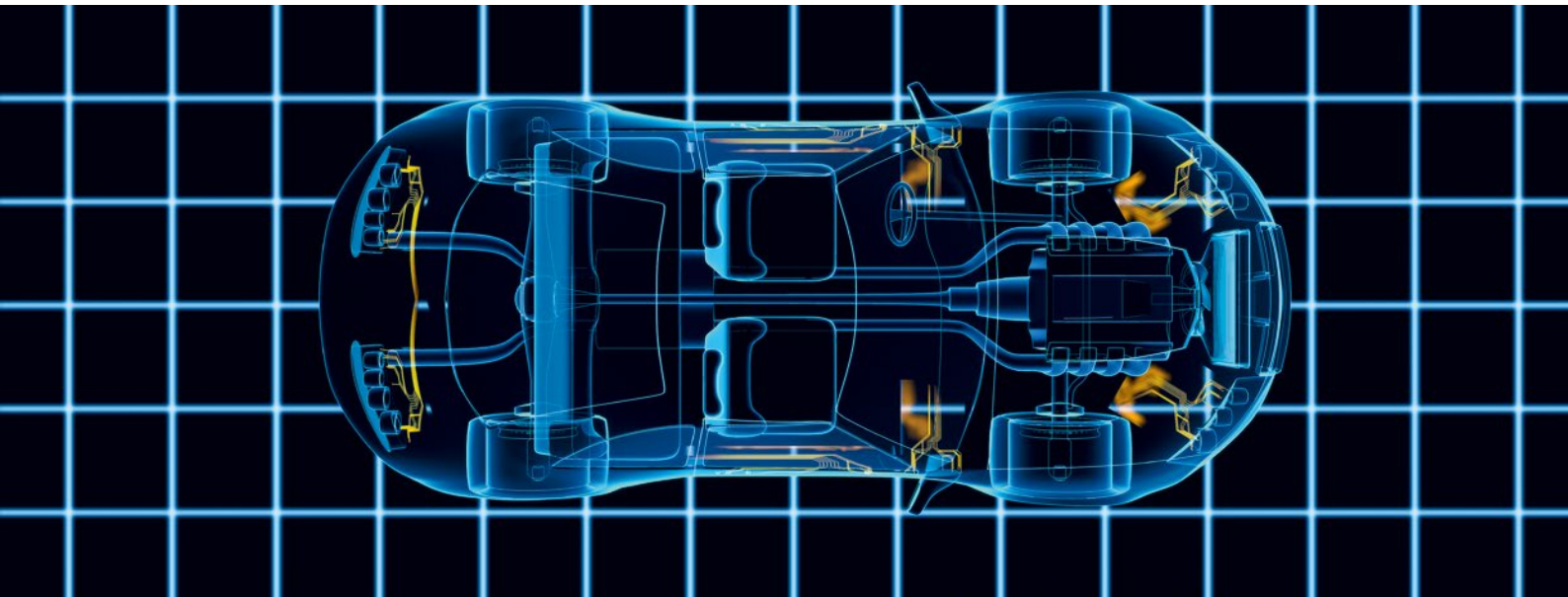
The CAN Module currently supports PCI-based cards with 1, 2, or 4 channels by Peak, Star Cooperation, EMS Dr. Wünsche, ESD, Ixxat and Kvaser (see list below). On request, we will gladly implement support for your preferred card.

Peak System

- PCAN-PCI Express FD, single-, dual-channel, opto-uncoupled
- PCAN-PCI single-, dual-channel, opto-uncoupled
- PCAN-PCI Express, single-, dual-channel, opto-uncoupled
- PCAN-miniPCI, single-, dual-channel, opto-uncoupled
- PCAN-cPCI, dual-, four-channel, opto-uncoupled
- PCAN-USB, opto-uncoupled and PCAN-USB Hub
- PCAN-USB Pro, dual-channel, opto-uncoupled

EMS Dr. Thomas Wünsche

- CPC-PCI, CPC-PCIe (single-, dual-, four-channel) and CPC-104P with SJA1000



ESD – Electronic System Design CAN-PCI

- CAN-PCI /200/266, opto-uncoupled, single and dual-channel
- CAN-PCIe /200, opto-uncoupled, single and dual-channel
- CAN-PCI-104 /200, opto-uncoupled, single and dual-channel
- CPCI-CAN /200, opto-uncoupled, single and dual-channel
- PMC-CAN /266, opto-uncoupled, single and dual-channel

Ixxat

- PC-I 04/PCI, dual-channel

Kvaser Advanced CAN Solutions

- PCIcan HS (single-, dual-, four-channel)
- PCIcanx HS (single-, dual-, four-channel)
- PCIEcan HS (single-, dual-channel)

Extensions

CAN FD Extension

RealTime communication over CAN FD (Flexible Data Rate)

LIN Module

Real-time communication over LIN (Local Interconnect Network)

Common

By using the LIN Module of Kithara RealTime Suite it is possible to attain real time with LIN. A master or slave node can be created easily using a common PC with a UART interface. This enables easy logging of existing LIN networks. It is also comfortable for the debugging of LIN nodes instead of using a microcontroller. Sending nodes will recheck its sent data on a bus so collisions can be detected.

Features

- LIN data exchange in real time
- Easy implementation of a master or slave node using a common PC
- Bus collisions detection
- Automatic checksum validation (LIN version 1.x and 2.x)
- Data rates up to 20 Kbit/s
- Guaranteed latency times
- Receive LIN messages either using callbacks or polling
- In order to adapt to the electrical parameters of the LIN bus, a LIN transceiver is required.
- Requires ›**Serial Module** und ›**RealTime Tasking Module**

Supported Hardware

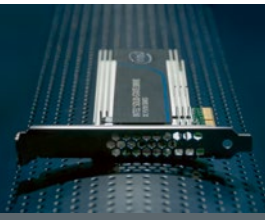
- UART (the use of common RS232 interfaces requires a level converter)
- Peak PCAN-USB Pro
- More on request

Support for PCAN USB FD adapter

Kithara RealTime Suite also supports PCAN USB FD adapters by Peak Systems. This allows for the connection of CAN FD and LIN networks with up to 2 ports on USB interfaces of a computer.

Real-time capabilities for interfaces such as CAN, CAN FD as well as FlexRay are often used for the development and implementation of automotive testing systems. This way, single components or entire hardware systems can be tested and analyzed under real-life conditions by utilizing extremely fast and guaranteed reaction times.

Real-Time Data Storage



In many industrial fields, Big Data has become increasingly important. Numerous applications already benefit from the every growing data quantities and some are made possible to begin with through them. Especially in fields such as measurement and test technologies, these data can be utilized effectively in order to generate high-precision results and evaluations.

Besides the capturing of data, such as from industrial cameras, storage is a crucial element for their efficent use. For this, data sets not only require to be written on storage mediums without loss at high throughput rates, they often also need to be fetched at the same speed. Based on the Storage Module this field of development at Kithara focuses specifically on the storaging of data in real time in order to ensure their profitable implementation into demanding projects.

The Storage Module allows for real-time data storage via SSDs with NMVe interface. This way, data can be written and read with throughput rates of several gigabytes per second at continuous operation. With UDF, an effective data system is provided as well, enabling optimal use of packet writing.

With RAID (Redundant Array of Independent Disks), the already high throughput rate can be increased even further by interconnecting multiple SSDs. This way, the data rate can be multiplied by the number of utilized SSDs (RAID 0), with speeds in the double-digit Gbyte/s range. With Kithara RealTime Suite, read and write accesses are also still processed in real time context.

In order to efficiently storage data sets, a real-time connection to MDF (Measurement Data Format) is available. Due to the support of MDF 4.1 within the Kithara real-time environment, large amounts of complex hierarchically structured measurement data can be immediately stored and accessed. Thanks to the real-time capabilities, losslessly captured raw measurement data is made available as MDF files for further usage. MDF is the defacto standard for measurement tasks, most prominently in the automotive industry, and is characterized by its fast and space-saving storage of captured data. The format also includes high-performance functions for efficient sorting and compressing of data sets.

Furthermore, the Wireshark-supported file format PCAPng can be utilized. In conjunction with real-time capabilities, data packets within networks can be directly captured and evaluated. This also means that the free, popular Wireshark can be used as a convenient tool for later analysis of stored data.

Modules

Storage Module

Real-time data storage via solid-state drives

Common

The Storage Module introduces real-time data storage via SSD with the most recent capacities. This way, via the NVMe interface, big amounts of data can be written as well as read with speeds of several Gbyte/s. UDF is provided as data system, which also allows for packet writing.

Features

- Real-time data storage via NVMe SSD
- High-speed reading and writing with several Gbyte/s
- UDF file system
- Requires › **RealTime Tasking Module**

Supported Hardware

In connection with the Storage Module, all storage devices, which are compatible to the NVMe-specification 1.0 can be used on request. Tested devices:

- Intel Solid-State Drive P3700/P3600/P3520/750 Series
- Intel Solid-State Drive DC P3520 Series
- Intel Solid-State Drive 600p Series
- Samsung 950 SSD
- HGST Ultrastar SN200 Series
- VirtualBox Solid State Disk

Extensions

Storage RAID Extension
Real-time data storage with multiple NVMe SSDs in RAID 0 groups, multiplied storage capacity and throughput rates.

RAID Extension

Connection of multiple SSDs

Allgemein

The RAID Extension expands the Storage Module with the functionality to combine SSDs as well as providing real-time capability. This allows for throughput rate as well as storage capacity to be multiplied by the number of interconnected SSDs.

Features

- Real-time implementation for multiple interconnected SSDs
- Multiply throughput rate and storage capability by the number of connected SSDs
- Throughput of several gigabyte per second up to the double-digit range
- support for RAID 0
- Stripe depth freely configurable
- Display RAID disk in Windows via pass-through mechanism
- Can be used with or without external RAID controller (e. g. HighPoint SSD7101A-1)
- Requires › **Storage Module**

PCAPng Module

Real-time data storage in PCAPng format

Common

The PCAPng Module (PCAP Next Generation) allows for real-time data storage of complex hierarchically structured measurement data in PCAPng format. PCAPng is a Wireshark-supported data format for capturing data packets within networks. This way, Wireshark is used for the later analysis of stored measurement data.

Features

- Real-time data storage in PCAPng format
- Wireshark used for reading stored data sets such as measurement data or even image data
- Based on **› Storage Module**

MDF Module

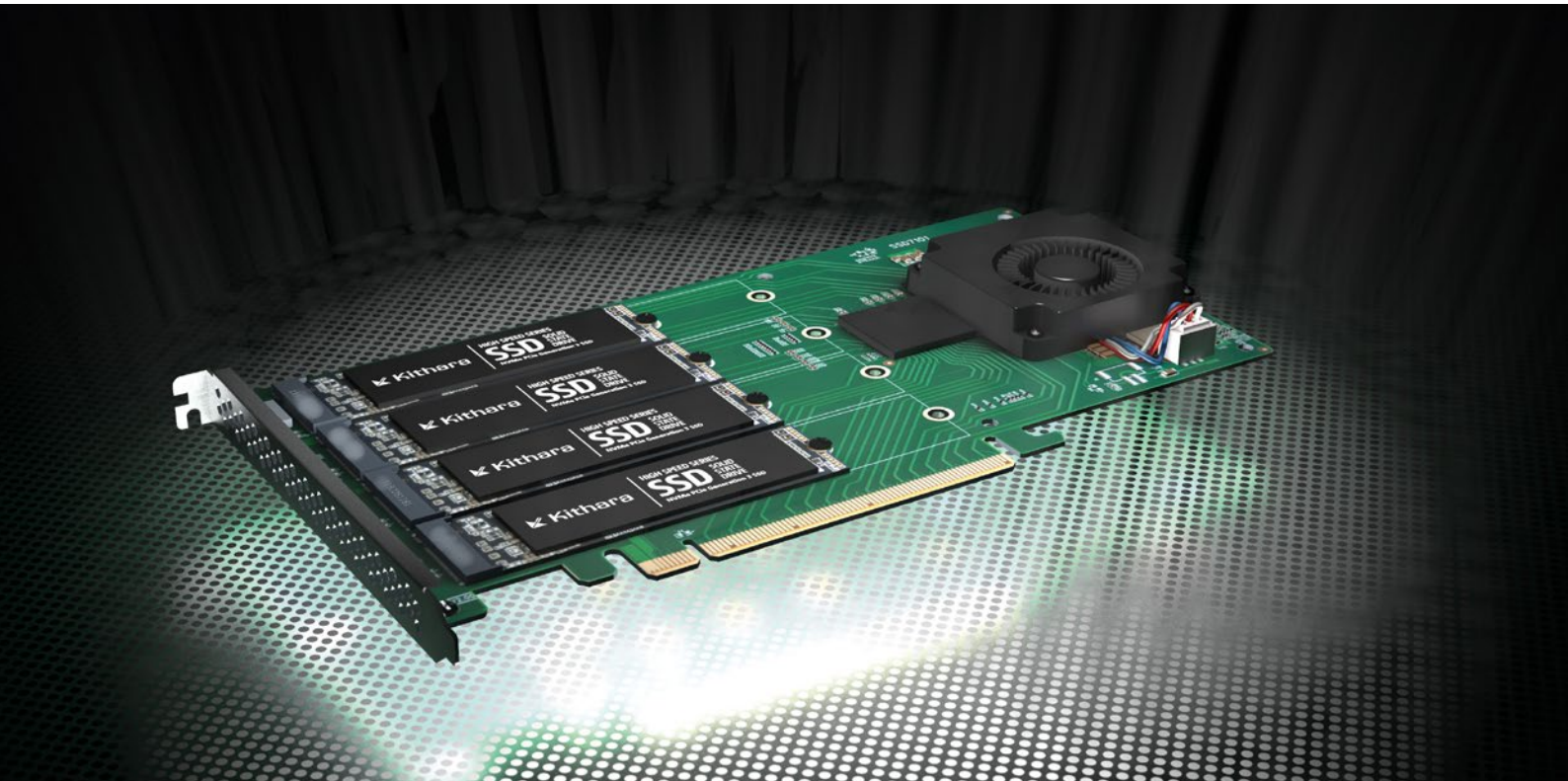
Real-time data storage in MDF

Common

The MDF Module allows for real-time data storage of complex hierarchically structured measurement data in Measurement Data Format. MDF 4.1 is a binary file format provided by ASAM for the evaluation or long-term storage of recorded and calculated data sets. The file size depends solely on the used hardware (meaning up to terabyte range).

Features

- Real-time data storage in MDF
- Support for MDF 4.1, downward compatible to earlier versions
- Writing of files with virtually any size (2^{64} Bytes)
- Storing of raw messages from the busses CAN, LIN, FlexRay and automotive Ethernet
- Based on **› Storage Module**



Real-time Libraries

Kithara generally recommends using the real-time-capable runtime library.

Library Module

Mathematical/trigonometric and String-Memory functions in real time

Common

For the execution of application codes in a real-time context, functions of the common runtime library are often required. However, they are possibly not qualified for real-time execution. Therefore, the Runtime Library provides real-time-adjusted functions of the C standard library, including mathematical/trigonometric and string/memory functions in real time.

Features

- Specific real-time-capable implementation of the C standard library functions
- More than 20 mathematical/trigonometric functions (e.g. sin, cos, tan, arcus and hyperbolicus functions, exp, log, pow, sqrt, floor, ceil)
- More than 20 string/memory functions (e.g. memcpy, memset, strlen, strcmp, strcpy, strcat, strtok)
- Supports application and kernel level
- Optional: Image processing—real time with Halcon or OpenCV
- Optional: Machine learning—real time with Dlib

Extensions

Dlib Extension

Dlib is an open program library with algorithms and tools for the practical application of machine learning tasks. The library includes various software components for dealing with networking, threads, graphical user interfaces, data structures, linear algebra, machine learning with artificial neural networks and deep learning. With the Dlib Extension, these functions can be used within the real-time context of Kithara RealTime Suite.

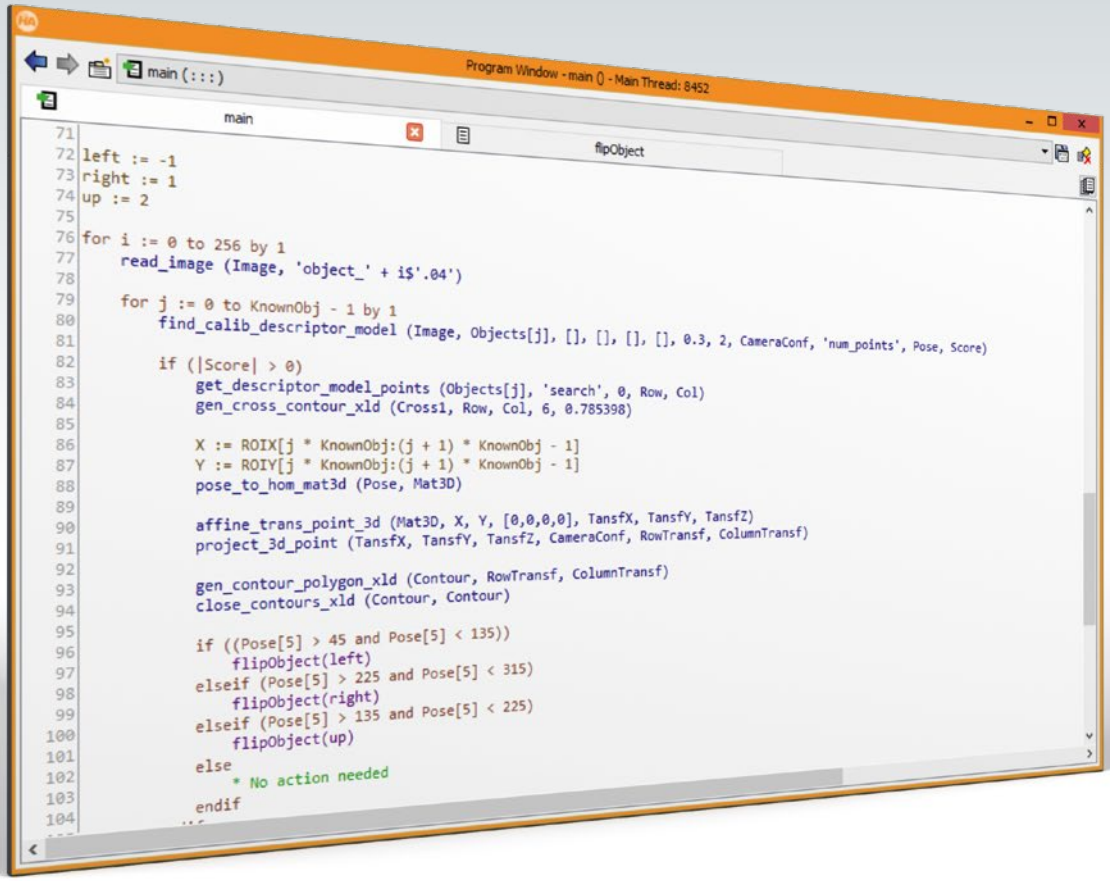
- Dlib in real time
- Algorithms and tools for machine learning
- Functions for networking, threads, graphical user interfaces, data structures, linear algebra, machine learning with artificial neural networks and deep learning
- Free program library

OpenCV Extension

Real-time image processing with OpenCV

Common

OpenCV is a free program library with algorithms for image processing and machine vision. Among others it includes algorithms for face recognition, 3D functionality, hair segmentation, various fast filters as well as functions for camera calibration. The OpenCV Extension allows for the implementation of this program library into the Kithara real-time system. This enables the triggering of immediate control reactions to processed images as well as their integration into automation systems with EtherCAT or CANopen.



Features

- Real-time image processing with OpenCV
- Algorithms for face recognition, 3D functionality, hair segmentation, various fast filters as well as functions for camera calibration
- Automatic parallelization
- Immediate control reaction to processed image data, for example with EtherCAT or CANopen
- Supports OpenCV Versions OpenCV 3.0, 3.4 und 4.1
- Requires **› Library Module**

Halcon Extension

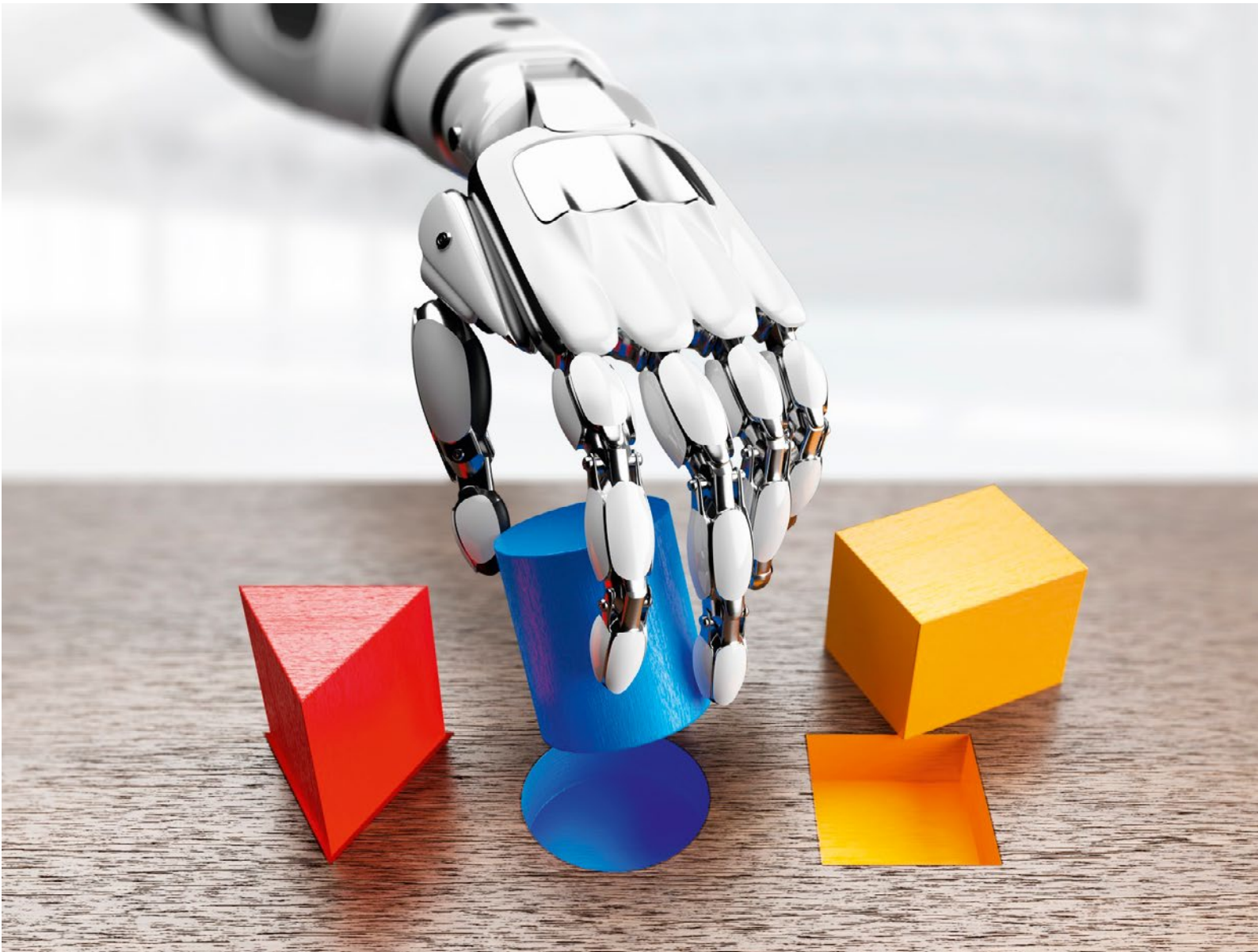
Real-time image processing with Halcon

Common

Halcon by Munich company MVTec is a program library with algorithms for image processing and machine vision. It includes its own integrated development environment HDevelop as well as over 2000 operators. The Halcon Extension allows for the implementation of this program library into the Kithara real-time system. This enables the triggering of immediate control reactions to processed images as well as their integration into automation systems with EtherCAT or CANopen.

Features

- Real-time image processing with Halcon
- Over 2000 operators
- Integrated development environment HDevelop
- Automatic parallelization
- Execute the HDevEngine directly from the real-time context
- Immediate control reaction to processed image data, for example with EtherCAT or CANopen
- Supports Halcon Versions 10 to 13, 18.11 as well as 20.11
- the Halcon license needs to be acquired by MVTec
- Requires **› Library Module**



SigProc Module

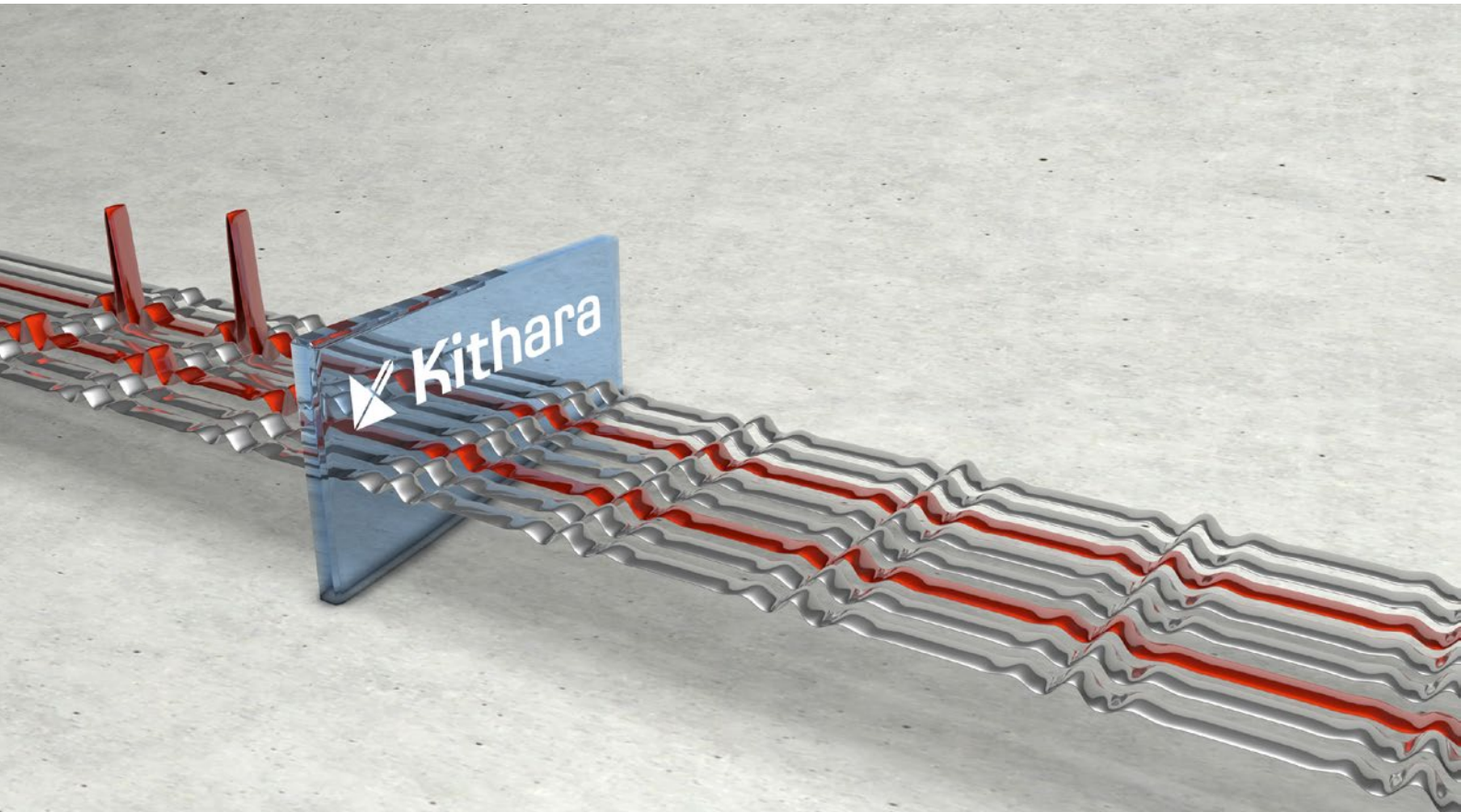
Digital signal processing in real time

Common

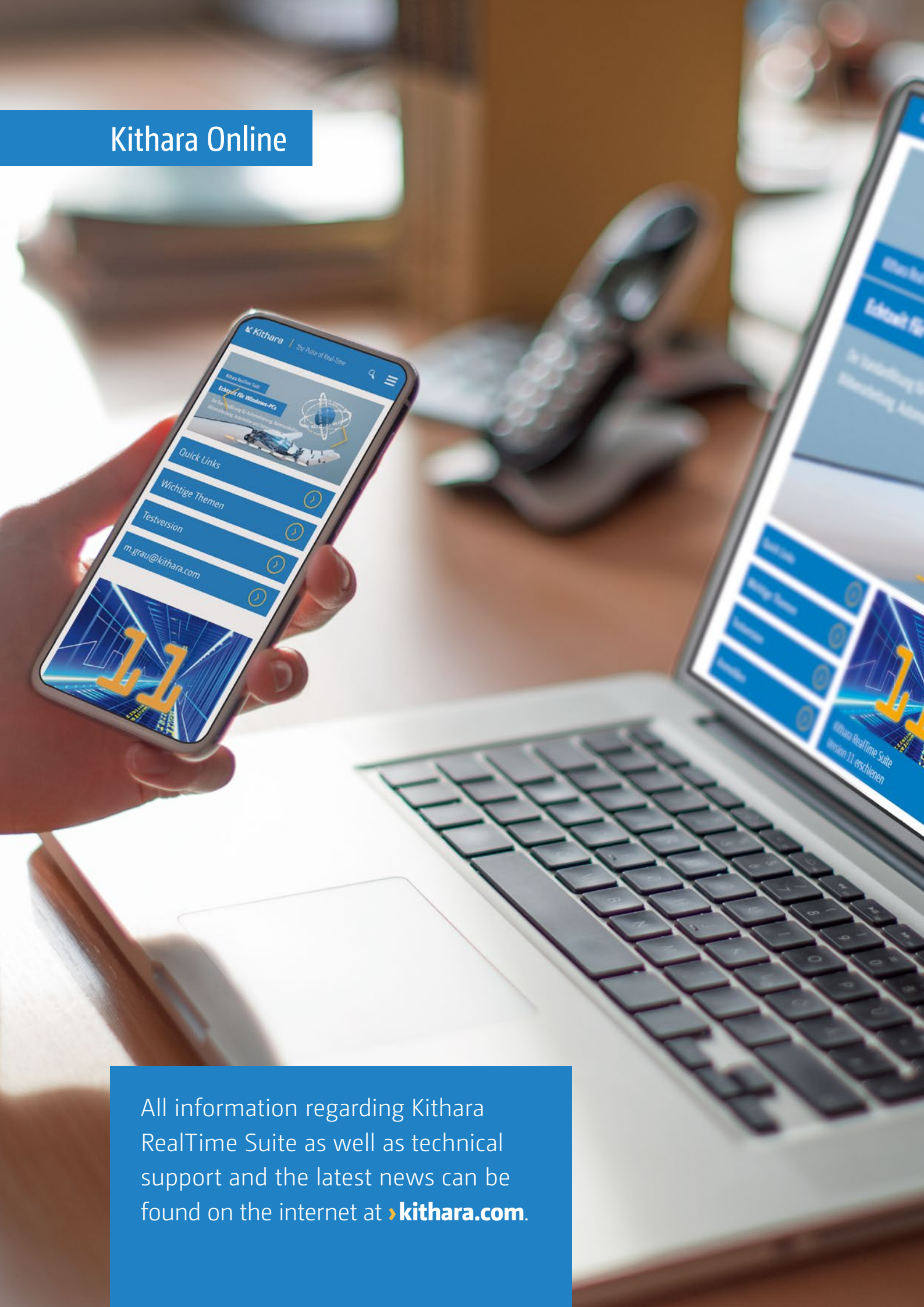
Signal processing and controlling of process variables in real time are often required for a wide field of engineering applications. Therefore, the SigProc Module provides a PID controller and a selection of digital FIR/IRR filters, including different window functions like Hamming, Hanning and Rectangular, as well as different transfer function polynomials like Chebyshev I, Chebyshev II and Butterworth.

Features

- Digital signal processing in real time
- More than 20 different filters (low-pass high-pass, band-pass, band-stop)
- 3 different window functions (Hamming, Hanning, rectangular)
- 3 different transfer function polynomials (Butterworth, Chebyshev I and II)
- User-defined digital FIR/IRR filters
- Feedback control in real time with PID algorithm



Kithara Online



All information regarding Kithara RealTime Suite as well as technical support and the latest news can be found on the internet at **›kithara.com**.

Development Tools

Kithara Performance Analyzer

System Resource Visualizer for Monitoring Computer Performance

Developing real-time applications always demands specific performance requirements for the development platform as well as the target platform. Kithara Performance Analyzer allows for the visualization and efficient monitoring of real-time capacity and system integrity in conjunction with the functions of Kithara RealTime Suite. The Windows task manager only provides limited insight when it comes to system efficiency while also lacking access to relevant data regarding real-time functions. In addition to access to system resources such as memory usage and CPU load, Kithara Performance Analyzer is also able to gather details about the data throughput of all connected interfaces that are supported by Kithara RealTime Suite. This way, various different data values can be gathered and displayed simultaneously, which allows for better identification of internal system interactions and problem sources.

Overview

Kithara Performance Analyzer is a development-related visualization assistant for simultaneous display and monitoring of different relevant system resources. It is therefore an important tool in determining a system's real-time performance. With the execution of Kithara Performance Analyzer, the internal real-time system is instructed to share all necessary information regarding system resources and interface throughput, which are then graphically displayed in real time. The user has full control over which values are displayed and in what format they are monitored.

The data communication with Kithara Performance Analyzer as well as the graphical display has no impact on system performance. Even in case of a large number of displayed values, the CPU load is only at one to two percent.

Features of Kithara Performance Analyzer

Kithara Performance Analyzer allows for the gathering and display of the following data:

- Memory usage for all memory categories such as internal real-time memory or application memory
- CPU load including Task and Interrupt (specific CPU cores can be displayed separately)
- Data throughput (send & receive) of all interfaces supported by Kithara, including (Gigabit) Ethernet, USB3/xHCI
- Protocols that are based on interfaces: EtherCAT, CANOpen, LIN, GigE Vision, USB3 Vision
- Jitter

Visualization functions

- Time-parallel display of multiple values
- Flexible time resolution (15 s to 2 min.)
- Linear or logarithmic display
- Graphical user interface freely adjustable: Display colors, graph positioning, axis scale, custom value designation, and more

Kithara Kernel Tracer

Multi-source real-time message tracer and logger

By using Kithara Kernel Tracer it is possible to trace and log internal processes of real-time applications in detail. During program development effective tools are necessary, for example for debugging. Previously, debuggers were used for this in most cases. However, concerning multithread-programming, a debugger is often limited in use, because it has an effect on the process. When using multiple processes and threads, especially regarding PCs with multiple CPUs (multi-processor and multi-core systems, hyper-threading), simultaneously, the result of the program process depends on the capability to execute particular program parts at the same time. A debugger cannot assure this. Well-known problems:

- By using a debugger the program flow is negatively impacted.
- Even though a program does not behave as expected, a debugger might make it seem like everything is working fine.
- The debugger does not have the capability to run in real-time context.

The solution is a tracer tool, that is able to log the actual process of simultaneous routines and analyze the log.

Brief description

Kithara Kernel Tracer is a multi-source real-time message logger. It supports debugging, quality assurance, and in general detecting interactions, that otherwise would be hard to determine. Kithara Kernel Tracer gathers every trace message that might appear system-wide from different sources, ordered by their time of appearance. Although messages from different sources appear on different tabs, it is possible to merge them.

Kithara Kernel Tracer is able to handle hundreds of thousands of messages per second. In order to find the important information in the huge amount of data, Kithara Kernel Tracer provides several mechanisms for filtering and searching messages. It provides amazing insights in the actual processes of a PC, for example at the kernel level. Besides the visualization through Kithara functions, you have the possibility to create your own messages, which can be displayed with useful additional information.

Features of Kithara Kernel Tracer in Detail

- Multi-source: Support of any number of threads and routines from application and kernel level (real time)
- Multi-core: Accurate allocation of messages from several CPUs (with SMP, HyperThread, Multi-Core, etc.) according to the time point they were generated
- High processing speed (hundreds of thousands of messages per second possible, if performance allows it)
- Every function call of Kithara RealTime Suite is displayed with function arguments
- User-specific messages can be generated by tools of Kithara RealTime Suite (similar to printf with C/C++)
- Messages are displayed with a lot of details including the system time in a 0.1 microsecond resolution
- It is possible to filter, merge, save and reload messages
- Filtering enables: Permanent deleting, transient hiding, bold, cursive and colored display of messages.
- Colored visualization enables accentuation of errors and warnings
- Message filtering from single source or global
- Further context details: Logical CPU, thread and/or process-ID, real-time task-ID, consecutive message number and type, thread and task priority, content of data blocks (of selected messages)
- Search for messages by text, date and all other fields
- Messages masks prevent unnecessary generation of non-required messages
- Messages are saved across BSOD (Crash-Dump will be automatically scanned)
- Different coloring for different processes, threads and task for better distinction
- Messages can be changed retroactively (text, foreground and background color) for documentation
- Programmed filters can be set on or off
- Switch visibility of data columns
- USB, UART, CAN and network devices can be monitored specifically
- Memory viewer to analyze physical memory
- PCI viewer for analyzing PCI devices
- Included SDK allows for the programming of unique trace sources

Customized Messages

You can forward your own messages to the tracer, for example, to display variable contents or to signal the occurrence of special program states.

For this, the tools of Kithara RealTime Suite provide the function KS_logMessage and KS_vprintK. KS_logMessage can be used with all supported languages; KS_vprintK uses an argument list and therefore can be controlled by C/C++.

Note
The correct functioning of the software in conjunction with specific components cannot be entirely guaranteed due to the large number of possible hardware variations. For questions regarding the compatibility with individual hardware configurations, please contact us. On request we will gladly send you a list of hardware supported by Kithara. Any information always refers to the latest version of our software.

Notes

Distribution
For any questions regarding distribution, please contact us via e-mail at **›info@kithara.com** or by phone under +49 30 2789673-0.

Support
You can reach the **›technical support** via e-mail at **›support@kithara.com**.

Notes

Kithara RealTime Suite

Test it for free right now!

Applications in
32/64 bit
with a uniform code base

Ready, steady
Quick Start
Ready-to-use programming frameworks

Fully functional test version with
90 days
free trial period

Function modules freely
combinable
with only few dependencies

Software Package
extendable
anytime with additional modules

Due to special conditions
affordable
Low-priced runtime licenses, as needed

References

The following companies and institutions are among our customers:

Some of our partners:

For further questions regarding features, supported hardware, operating systems or programming languages, please visit our website ›kithara.com. We recommend downloading the free trial version. Please contact us for any questions about your project!

QR Code | Kithara on the Internet:

Kithara

Kithara Software GmbH
Alte Jakobstr. 78, 10179 Berlin, Germany, Phone: +49 30 2789673-0, Fax: +49 30 2789673-20
Email: info@kithara.com (general inquiries), support@kithara.com (technical inquiries), Internet: kithara.com