

产品概述

Kithara RealTime Suite

面向自动化、通讯、机器视觉、
汽车领域及数据存储的实时解决方案

NEW!

速率翻倍

以太网支持200 Gbit/s: ▶第11页

Timer Module

极高精度的系统时间检测与极短时间的延迟控制: ▶第15页

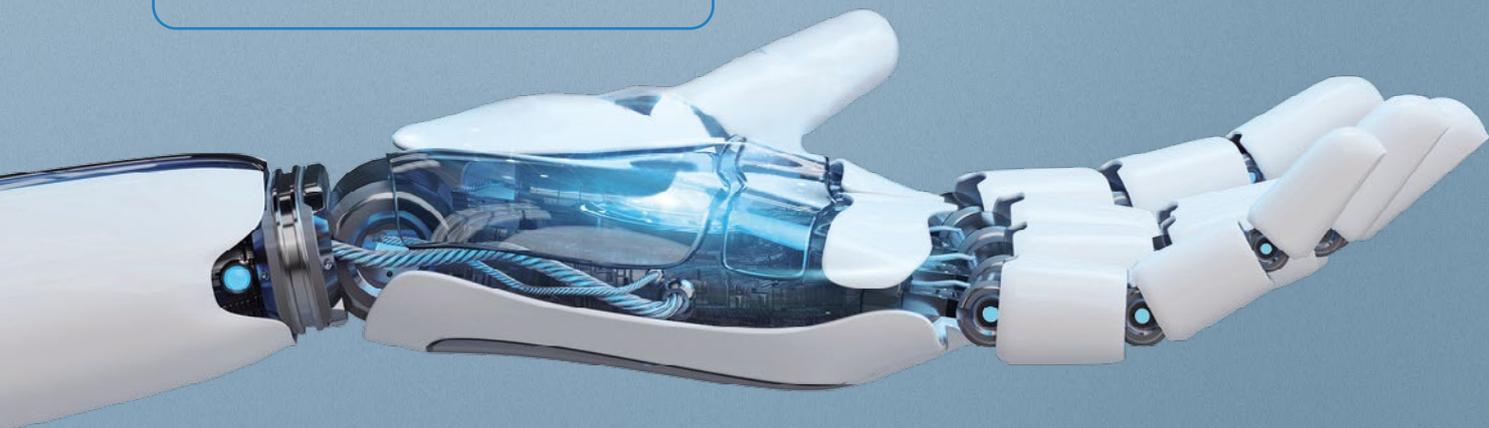
RealTime Tasking Module

高频实时定时器例程和优先级驱动的抢占式实时多任务处理: ▶第16页

Timer
Module



RealTime
Tasking
Module





精确计时

适用于Windows的实时技术

在当代社会，时间是一个至关重要的因素。工业级别的自动化流程需要应用程序与机器之间实现精准的通信。高性能设备会产生大量需要被立即处理的数据。

Kithara致力于为工业与研究领域开发和提供可以精确至微秒级的控制，并且具有监控与分析功能的软件。我们的产品广泛应用于汽车制造、航空航天、食品生产以及医疗工程等领域。即便是在“自动驾驶”这一前沿领域，Kithara也有非常重要的贡献。

问题与解决方案：Microsoft Windows 有时会覆盖正在运行的进程的优先级，以便执行自己的程序和维护功能。这可能会导致应用程序出现不必要的中断。

为此，Kithara创建自己的保护区，该保护区不受操作系统限制，确保了程序运行环境的平稳运行。

Kithara RealTime Suite

面向基于PC自动化的Windows实时扩展自动化

KitharaRealTimeSuite是一款模块化的Windows实时扩展，它将硬件相关的编程、通讯、自动化协议以及图像采集和图像处理整合为一个高性能的实时系统。得益于各模块间的无缝集成，用户得以在一个统一的平台上获取所有必需的功能。

高效的实时驱动程序构成了通过TCP和UDP进行套接字通讯的基础，同时支持兼容GigE-Vision标准的摄像机进行图像采集，以及工业以太网协议的应用。此外，该软件还配备了用于CAN和UART接口的实时驱动，增强了通讯能力。

在支持的自动化协议中，包含了一套全面的EtherCAT®主站实施方案，具备分布式时钟、Safety-over-EtherCAT安全技术、热插拔能力和电缆冗余功能。

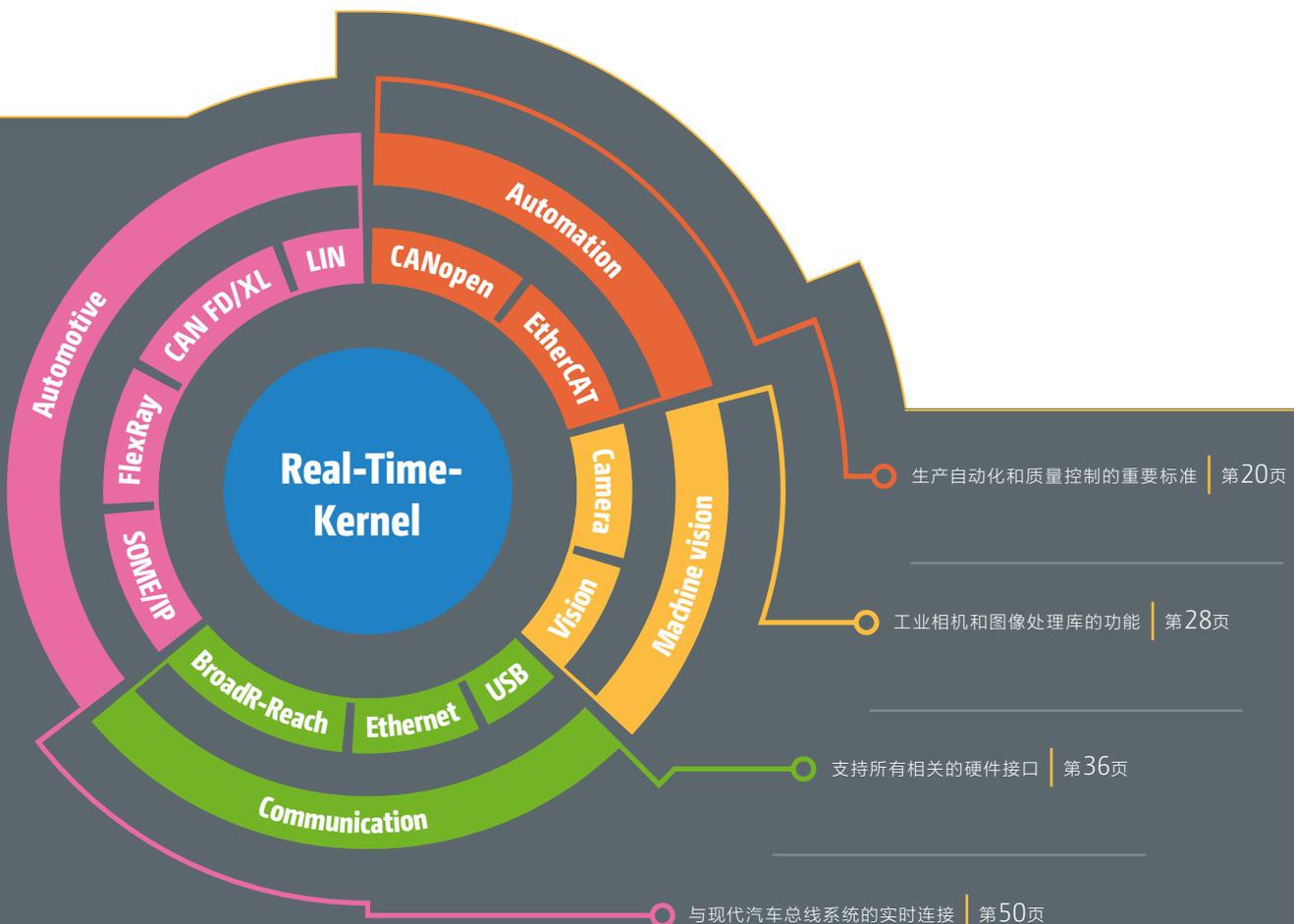
同时，通过多种不同的CAN接口提供了CANopen®主站功能，这些接口亦可融入EtherCAT®体系。

硬件相关的编程功能涵盖了直接I/O访问、内存访问与中断处理，并为多功能板卡提供了一个便捷的高级接口。

对于由GigE Vision®或USB3 Vision®标准采集的图像数据，可以在实时环境中运行如Halcon或OpenCV等综合图像处理库，进行高效处理。

为了实现快速的实时数据存储，Kithara RealTime Suite支持NVMe接口的SSD（可通过RAID 0阵列进一步提升速度），并能即时将数据集以分层结构的文件形式保存为MDF或PCAPng格式。

而借助熟悉的编程语言和开发环境，Kithara RealTime Suite使用直观易用的接口将实时系统与Windows操作系统结合在一起，使得Windows系统既作为开发平台也作为执行主机，避免了额外添加单独的实时系统，最大限度的减少了开发时间和成本。



我们的产品将为您的项目量身定制

Windows凭借以下特质，成为工业应用的理想选择：

- 广受欢迎：直观且熟悉的用户界面
- 现代化：适配最新的通讯接口
- 高效生产力：强大的开发工具
- 成本效益：低廉的实施与培训成本
- 长远无忧：因高市场份额与持续发展而历久弥新

Kithara RealTime Suite补充了Windows所缺乏的实时性能，凭借以下特点，使一些对实时性较高的应用得以执行：

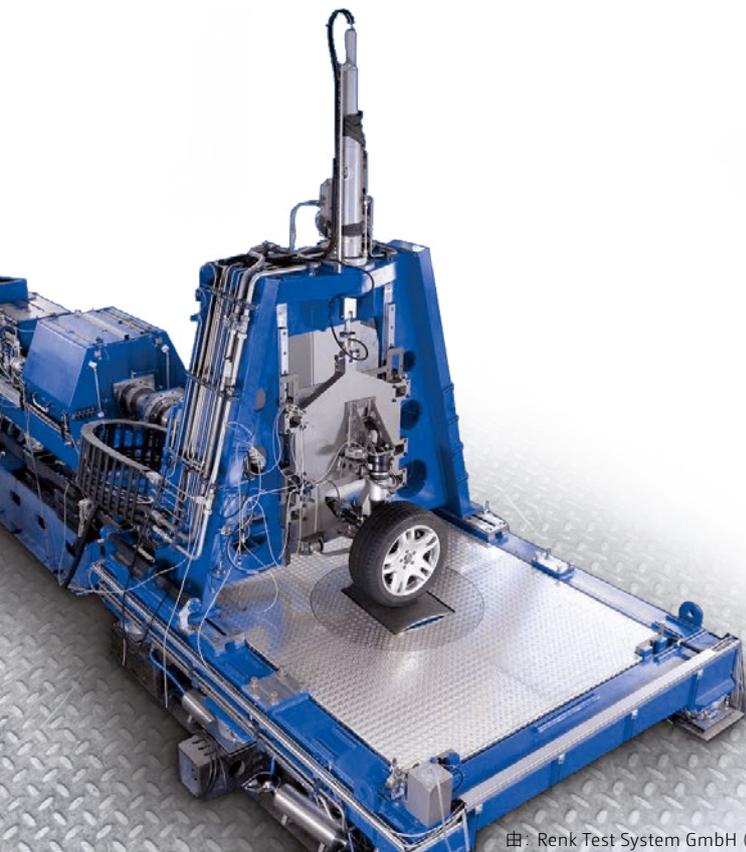
- 确定性的行为表现
- 极低的最大响应时间
- 熟悉的编程语言与工具
- 强大且易于应用的功能
- 短暂的学习周期——成本高效的开发
- 广泛支持工业协议与标准

Kithara RealTime Suite使得在同一系统内实现实时控制与可视化成为可能。应用程序在独立模式下运行，可自由配置的CPU核心数量。与此同时，Windows负责剩余CPU核心上的可视化与用户交互。如此一来，两部分独立运行，互不影响。Windows与Kithara RealTime Suite的结合，为多样化的解决方案提供了可能，涵盖如下领域：

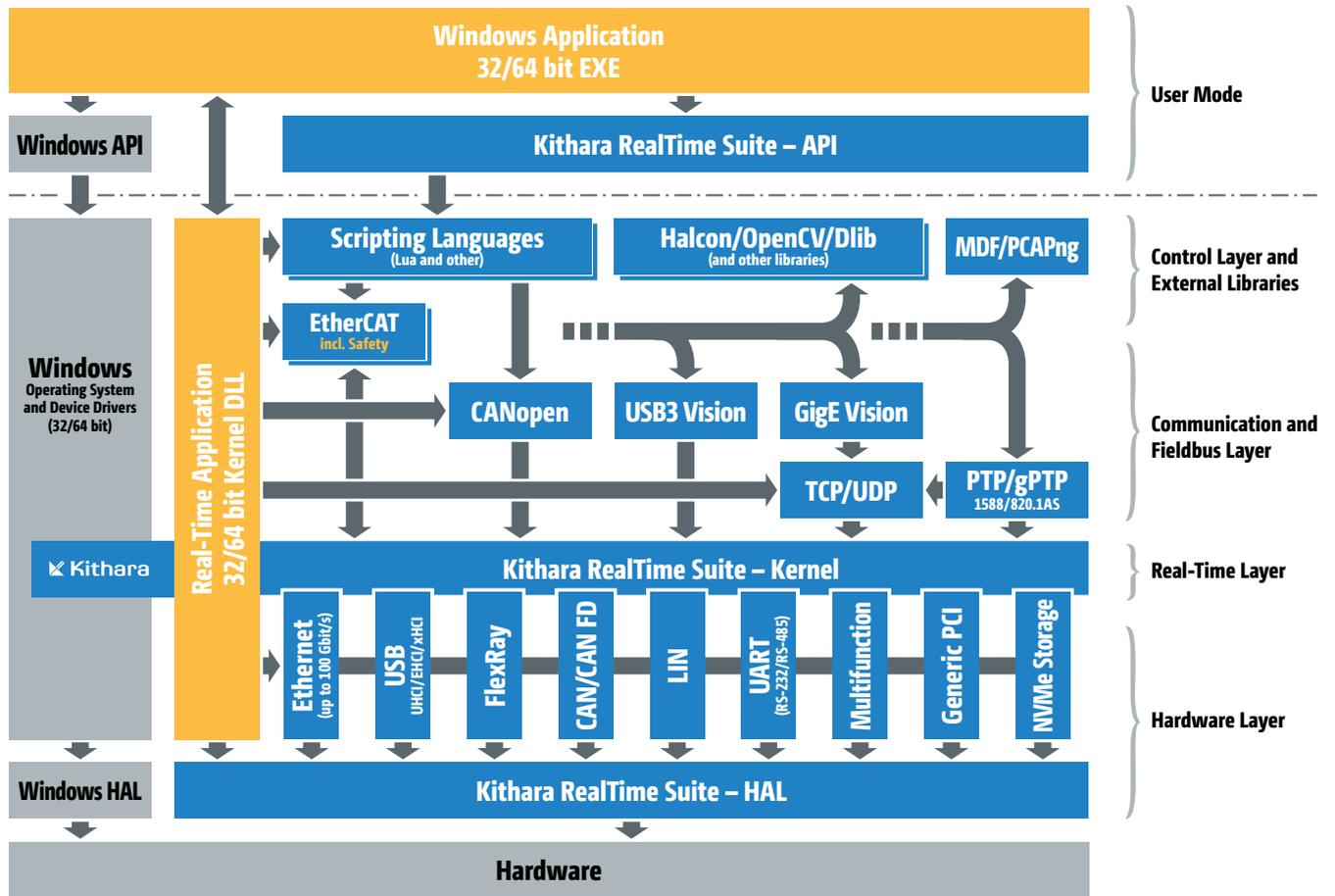
- 机械工程、特种机械制造与测试系统
- 生产自动化与质量保证
- 实验室自动化与移动应用
- 汽车工程与航空航天行业的测量与测试站
- 医疗器械
- 科学研究实验室

总结

Kithara极大简化了您的项目实施。您无需再构建一个独立的实时系统，从而规避了不同系统间通信的高昂实现成本。相反，您可以充分利用统一解决方案的优势，享受显著减少的开发工作量、连贯一致的编程工具链所带来的便利。这不仅缩短了产品上市时间，降低了成本，还有效减少了资源消耗。因此，Kithara RealTime Suite与Windows的结合，成为了构建成功实时解决方案的理想基石。



下图显示了使用层模型将Kithara RealTime Suite连接到系统的不同软件和硬件组件。



- ① 应用与实时部分之间的通信与同步，通过共享内存、管道、进程间通信、套接字、事件和信号量来实现。
- ② EtherCAT主站是一种高性能的工业以太网控制方案，适用于多种自动化任务。同时，也支持如CANopen主站等现场总线协议。
- ③ 来自GigE Vision与USB3 Vision相机的图像数据能够被实时捕捉，并借助Halcon及其他库进行处理。
- ④ 基于优先级的抢占式实时多任务环境，允许分配专属的CPU核心，提供高频处理能力的同时确保低抖动。

在Windows系统下, 实时技术如何真正发挥作用?

对实时概念有所了解的人都深知, 它在当今日新月异的信息技术领域占据着举足轻重的地位。不论是机器人手臂中伺服电机的并行操控、精密测量应用, 还是基于摄像头的驾驶辅助系统, 硬件与软件组件间的即时响应时间都是诸多现代电子技术不可或缺的关键要素。以往, 为了实现这一需求, 人们要么采用独立的实时操作系统 (RTOS), 要么依托专业的硬件设备 (如DSP和FPGA)。而今, 这一切同样能在标准的、市面上可购的Windows个人电脑上得以实现。尽管如此, Windows系统本身并不具备实时特性, 因为它会频繁中断正在运行的应用程序, 以便执行其他程序和维护任务等。那么, 如何将这看似矛盾的两者融为一体? Windows环境下实时功能的实现机制又是什么呢?

实时技术与Windows——解决方案

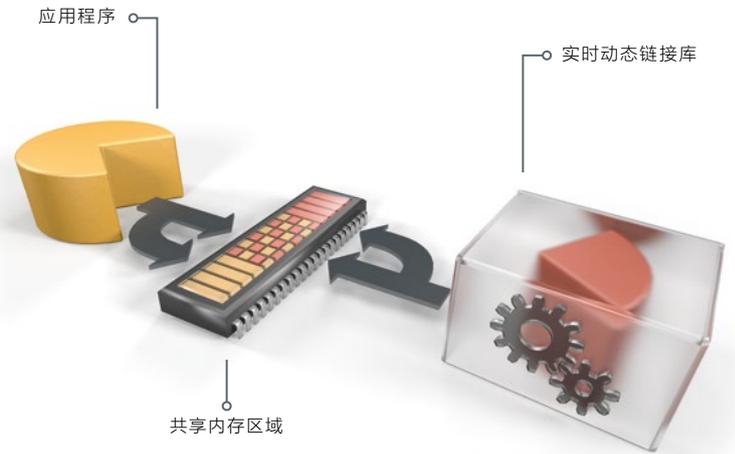
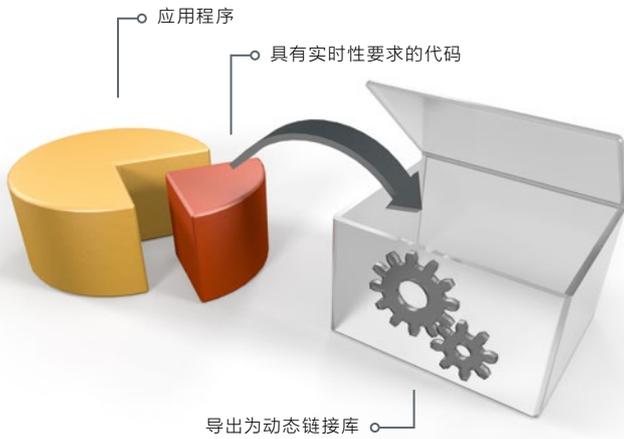
在Windows个人电脑上实现实时处理的唯一硬件需求是一个至少包含两个核心的处理器。其基本运行原理是, 让Windows仅使用多核处理器中的有限几个CPU核心。于是, 空闲出来的核心上便能启动实时系统, 自此如同一个独立的实时操作系统 (RTOS) 般运行, 与此同时, Windows在剩余的核心上继续保持全面的操作能力。这意味着从此刻起, Windows系统与实时系统在同一台计算机上并行不悖、同时运行, 互不影响。通过配置程序msconfig来限制Windows使用的核心数量, 这一操作相对简便易行。关于此设置的具体指引可参见: [配置专用CPU](#)。

得益于这一流程, 实时系统获得了一个受保护的运行环境, 确保 Windows 进程不会在具有高实时性任务中获得更高的优先级, 从而防止它对实时性能产生负面影响。最终目标是在 Windows PC 上实现特定的“硬”实时功能。但是, 在这种情况下, “硬”究竟意味着什么?

“软”实时与“硬”实时的区别何在?

谈及时实技术, 我们需区分不同的需求以及进程执行时间的处理方式。“软”实时追求的是反应时间的大致平均值, 而“硬”实时则必须确保不超过预定的时间限制。鉴于这些更高的要求, 实现“硬”实时功能显著更为复杂且富有挑战。然而, 这种确定性的时间行为在自动化工业领域, 尤其是测量、测试与控制技术中, 是不可或缺的。





那么, 如何让Windows与实时环境再度协同工作呢?

实时系统以设备驱动的形式实现, 并提供了自己的API, 用户可以借此将Windows应用程序中的具有实时性要求的代码段导出到一个动态链接库 (DLL) 中。随后, 这个DLL将在RTOS的实时环境中加载运行。为了让处于不同执行环境的Windows与该DLL能够相互通讯——除了使用管道或套接字等间接方式外——可以创建一个双方均可访问的共享内存区域。即便采用了实时操作系统, Windows依旧作为编程接口和宿主系统发挥作用, 确保了应用的开发便利性和系统兼容性。

为何选中Windows作为实时操作系统?

相较于纯粹的实时操作系统及实时硬件解决方案, Windows电脑展现出特有的优势。Windows本身作为一种深入人心的用户界面, 不仅拥有广泛的支持驱动, 频繁迭代更新, 还涵盖了丰富多样的可执行程序。而纯正的RTOS则高度专业化, 导致其在功能范畴与操作便捷性方面常常局限颇多。诸如FPGA与DSP之类的硬件解决方案能够依据不同规范进行定制, 然而, 它们的编程过程复杂且耗时良多。要在Windows平台上实现“硬”实时性能, 仅仅需要一台双核配置的电脑即可。

总结

Windows之所以能完美贴合工业应用的需求, 得益于如下特质:

- 普及广泛、直观且熟稔于心的用户界面
- 连通最新通信接口的便捷途径
- 高效的开发者工具加持
- 实施成本与培训开销俱低
- 因市场占有率高及持续迭代更新而历久弥新

Kithara RealTime Suite实时系统补充了Windows缺少的实时功能, 使得实时性要求较高的应用的实现, 主要因素有以下几点:

- 凭借高频实时计时器确保的确定性行为
- 极其短暂的最大响应时间
- 熟悉的编程语言与工具
- 高效且易于应用的功能
- 短暂的学习周期与成本效益高的开发
- 广泛支持工业协议与标准

Windows与实时并非水火不容, 两者的无缝融合促成了最佳解决方案, 并常为工业领域的应用开启了先决条件。亲自见证一下Windows下实时技术的运用吧。只需试用Kithara RealTime Suite的 **▶ 全能免费试用版**, 内含众多帮助快速上手的示例, 让您亲身体会实时性能的魅力。

专为Windows平台打造的"硬"实时

凭借模块化构建系统及其大范围实施的可能性，Kithara RealTime Suite成为了开发自动化、硬件通讯及机器视觉领域高效应用的理想基石，因为这些领域常常需要依赖较高的实时环境。

Kithara RealTime Suite采用了可靠的专用模式，使单一逻辑CPU（CPU核心）完全独立运行，不受Windows系统的任何影响。利用适当的硬件，可实现上万赫兹级别的周期性计时器调用，其偏差（抖动）仅为几微秒，确保了极高的精度。

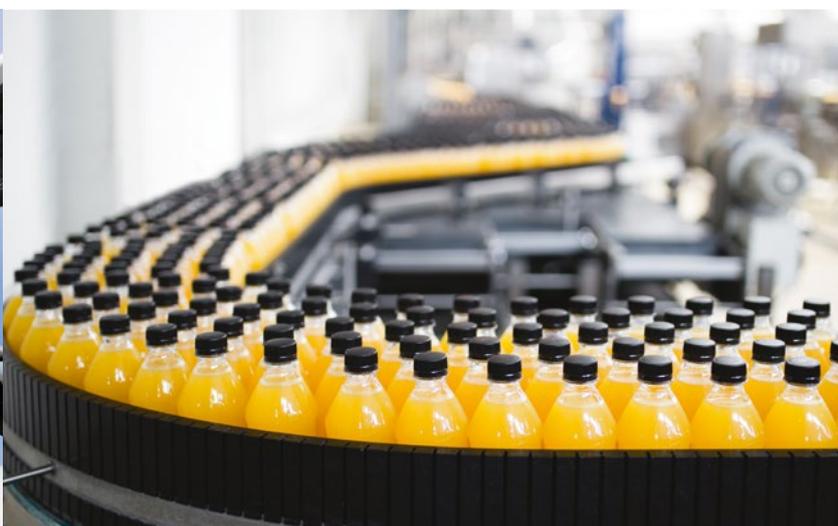
在实时启动之初，系统首先会对硬件进行全面分析，包括所有可用的时钟，并进行校准。不仅计时器例程能够以高频率运行，抢先式实时多任务系统还为任务（线程）提供了高达255级的优先级设定，确保优先级最高的动作总是优先执行。较低优先级的动作会被立即中断，直至更高优先级的动作完成后才会继续执行。这一设计保证了在严苛的时间约束下，关键任务的可靠与高效执行。



为了实现内核DLL与Windows应用程序间的同步，提供了事件、信号量、互斥锁、数据及消息管道、套接字以及共享内存等多种机制。个别任务可被特定分配至不同的逻辑CPU上，以便进一步优化专门任务的性能与完整性，以及整个系统的可扩展性。

围绕内核，还有多种其他模块用于通过不同的通信和访问接口连接外部设备与系统，以及应对

特殊任务。例如，Storage Module能够以每秒数GB的持续吞吐率在SSD上实现高速数据存储。采集的测量数据也可以是面向未来的测量数据格式(MDF)，存储为TB大小的文件，确保长期适用性。此外，诸如Kithara Kernel Tracer和Kithara Performance Analyzer等强大诊断与编程工具也供用户使用，使得整个开发流程更加灵活高效。



个性化软件组合

- 根据您的需求自由搭配功能模块，或联系我们，我们将协助您挑选合适的组件。这款定制化驱动程序旨在提供最大灵活性，具备以下特点：
- Kithara RealTime Suite采用模块化构建，能够创建一个根据您的需求量量身定制的个性化实时操作系统。
- 开发与运行时文件的命名可自由选择。
- 稳固的应用程序接口(API)，在新版本发布时无需进行修改。
- 购买模块后，您将自动获得开发者许可。
- 运行时许可享有优惠的价格阶梯条件。
- 初始购买后12个月内提供多次更新。
- 更新服务可额外延长12个月。
- 随时可添加更多模块进行扩展。
- 包含一次性10小时开发者支持，超出部分可选择不同支持包（20、50、100小时不等）。
- **支持服务通过Kithara票据系统处理，确保快速直接与开发人员沟通。**

我们的专业支持团队将为您的项目集成软件提供最佳解决方案，并且我们的开发支持还将在软件之外的问题上为您提供帮助。

平台

实时性能仅能在内核层面实现。为此，需要一种能生成本地机器代码的编程语言，如C/C++或Delphi。尽管如此，Kithara RealTime Suite仍兼容多种平台，包括.NET环境。解决之道在于，将时间敏感代码封装进一个动态链接库(DLL)，该库将在内核级实时环境中直接加载。Kithara RealTime Suite提供了所有这些功能，并为上述平台预置了即用型程序框架，作为软件交付的一部分。

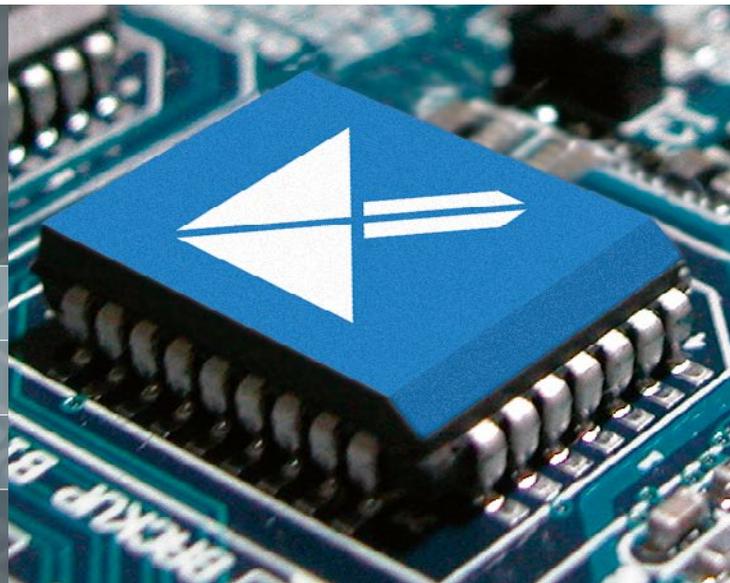
操作系统

本软件支持以下操作系统：

- Windows 10及11 (32位与64位)，专属模式
- Windows Server 2016、2019及2022，专属模式

系统要求

Kithara RealTime Suite支持广泛的硬件和软件组合以及工业标准。如有其他疑问，请联系我们。



硬件

本软件兼容以下硬件配置：

- 处理器：AMD (速龙系列) 或Intel (奔腾2代及以上)，支持32位与64位架构
- 双核或多核处理器；多核处理器可选配超线程技术，当前支持最多达48个逻辑处理器，如有更高需求可特别申请
- 建议采用ACPI (高级配置与电源接口)
- 计算机需支持PAE (页地址扩展)
- 千兆以太网至200吉比特/秒，广泛支持Intel与RealTek的网络控制器

NEW!

编译器

随软件提供的可立即使用的程序框架涵盖了以下编程语言/编译器（如有其他编程语言需求，可另行申请支持）：

- Visual Studio 20xx C++，配备MFC用户界面（已提供项目文件：VS 2010至2022版）
- C++ Builder (Embarcadero公司，前身为Borland) VCL用户界面
- Delphi (Embarcadero公司，前身为Borland) VCL用户界面
- Visual Studio 20xx C#，配备WPF用户界面（含用于实时执行的C++ DLL，已提供项目文件：VS 2010至2022版）

为了在实时环境中执行代码，原则上任何能够生成包含原生物理码的DLL的编译器均可使用。此外，Windows应用程序亦可采用C#等其他编程语言开发。

套餐方案



针对常见的应用领域，我们已将最重要的模块整合成一系列套餐。这些套餐不仅满足了多数客户的基本需求，还包含了额外的工具，相较于单独购买模块更为划算。

Kithara RealTime EtherCAT——专为Windows设计的实时EtherCAT技术

Kithara RealTime EtherCAT是一款面向Windows操作系统的模块化实时扩展，特别专注于利用EtherCAT实现自动化控制的实时应用。

Kithara RealTime EtherCAT Vision Set——Windows平台下的EtherCAT与视觉实时处理

Kithara RealTime EtherCAT Vision Set是专为Windows操作系统设计的实时扩展，集成了自动化控制与图像采集处理功能，实现了二者的实时协同。

Kithara RealTime Vision——专为Windows打造的实时图像采集与处理

Kithara RealTime Vision是一个面向Windows操作系统的模块化实时扩展，专注于利用GigE Vision和USB3 Vision技术实现实时的图像采集与处理。

Kithara RealTime Automotive——汽车领域的实时技术支持

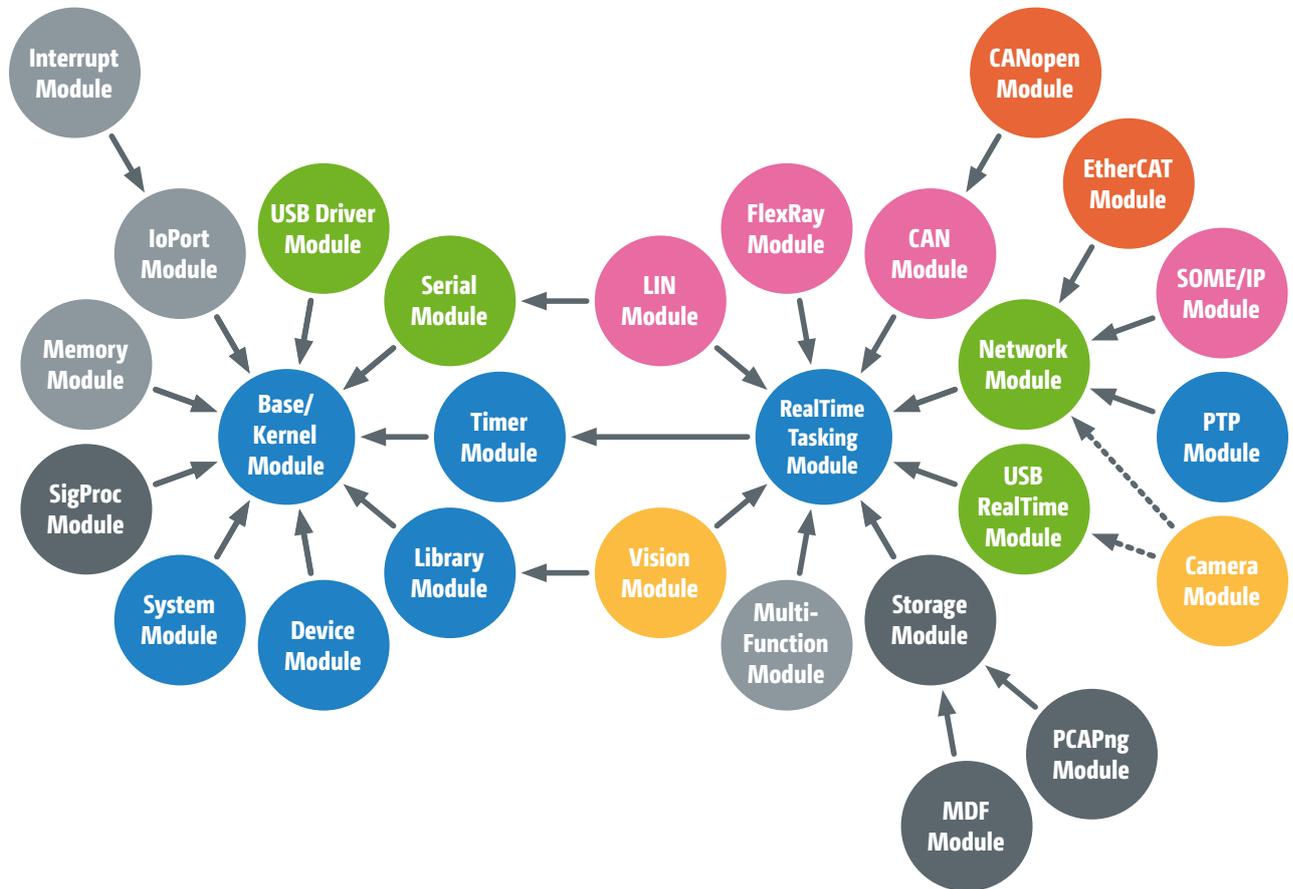
全面支持汽车工程中至关重要的总线系统，这些系统对于驾驶员安全、驾驶舒适度以及驾驶辅助至关重要：CAN/CAN FD、FlexRay和LIN。

套餐	RealTime EtherCAT	RealTime CANopen	RealTime Vision	RealTime Automotive
基础功能				
Base/Kernel Module	✓	✓	✓	✓
System Module	✓	✓	✓	✓
Library Module	✓	✓	✓	✓
实时系统				
Timer Module	✓	✓	✓	✓
RealTime Tasking Module	✓	✓	✓	✓
Storage Module				
PCAP Module / MDF Module				
PTP Module				
硬件访问				
IoPort Module				
Memory Module				
Interrupt Module				
MultiFunction Module				
通讯技术				
Network Module	✓		✓	
USB RealTime Module			✓	
Serial/UART Module				✓
Device Module				
开发工具				
Kernel Tracer	✓	✓	✓	✓
Performance Analyzer	✓	✓	✓	✓
Master Monitor	✓			
自动化				
EtherCAT Module	✓			
CANopen Module		✓		
机器视觉				
Camera Module			✓	
Dlib Extension			✓	
Halcon Extension*			✓	
OpenCV Extension*			✓	
汽车领域				
FlexRay Module				✓
CAN Module		✓		✓
CAN FD				✓
LIN Module				✓
BroadR-Reach (100BASE-T1/1000BASE-T1)				

您可以挑选一个合适的套餐，并根据实际需要搭配额外的模块。所有可用模块的详细列表参见下方。
 Halcon许可证需单独从MVTec获取。OpenCV则是一个免费的编程库。

模块

Kithara RealTime Suite由多种模块组成。软件可以根据客户需求特定地从这些模块中进行组合，或以即用型套餐形式获得，并可进一步搭配额外模块进行扩展。下图展示了各模块之间的依赖关系：



基本功能

Base/Kernel Module

负责整体管理，在操作系统内核层级执行，实现实时内存管理，提供调试支持，系统信息展示，基础资源管理以及设备操控功能。

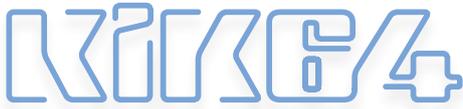
概述

Base/Kernel Module通常是Kithara RealTime Suite每个实时系统的基础，因此始终是必备组件。它提供了从Windows应用程序启动实时驱动的功能，以及在内核级别执行实时代码以达到实时环境的能力。该模块支持完成一般管理任务的函数、版本信息查询以及用于调试和获取系统信息的帮助函数。它还包含了应用线程、事件、回调、共享内存、数据和消息管道及快速互斥对象等基本资源的管理机制。此外，模块还提供了实时内存管理、内存复制功能以及即插即用安装所需的通用WDM驱动。集成的KiK64功能允许32位代码在64位系统上运行（仅限64位版本的Kithara RealTime Suite）。



[Base/Kernel Module](#)
[System Module](#)
[Timer Module](#)
[RealTime Tasking Mod.](#)
[PTP Module](#)

特征



- 驱动加载功能：用于启动实时驱动的函数。
- 内核级实时代码执行：在操作系统内核层面执行实时代码。
- 错误描述获取：检索详细的错误描述信息。
- 调试辅助：内置调试支持工具。
- 系统信息获取：收集并展示系统详细信息。
- 应用线程管理：创建和管理应用线程。
- 事件与回调处理：事件对象与回调函数的调用机制。
- 快速互斥体：用于同步的快速互斥对象功能。
- 共享内存操作：共享内存区域的分配与访问方法。
- 数据与消息管道：带自动读写解耦的数据与消息传输管道。
- 设备与驱动管理：查询设备信息及驱动管理工具。
- 实时内存管理：针对实时需求优化的内存管理方案。
- 实时内存复制：适用于实时环境的高效内存复制功能。
- 通用WDM驱动：内置即插即用安装所需的通用WDM驱动。
- 生成与 Kithara Kernel Tracer 兼容的日志消息
- KiK64功能：允许32位代码在64位系统上运行（仅适配Kithara RealTime Suite的64位版本）。
- 支持语言：仅限能生成原生机器代码的语言（如C/C++或Delphi）。
- 操作系统兼容：支持Windows 10及11（含32位与64位），以及 Windows Server 2016、2019与2022，均能在专用模式下运行。
- 处理器与架构适配：兼容多核处理器、超线程技术及NUMA多处理器架构的个人电脑。
- 基础/内核模块扩展：包含内核CPU扩展与内核内存扩展功能。
- Base/Kernel Module Addons: Kernel CPU Extension, Kernel Memory Extension

硬件支持

以下是必要的系统要求：

- CMPXCHG8B/CMPXCHG16B CPU指令集支持：系统需支持该CPU指令集。

扩展功能

Kernel NUMA Extension

内核NUMA扩展：实现在实时环境中利用具有“非统一内存访问”(NUMA)架构的多插槽系统，自动优化内存分配，提升多处理器性能。

Thunderbolt Extension

Thunderbolt扩展：对通过Thunderbolt接口连接的设备提供支持，确保高速数据传输与设备兼容性。

System Module

系统模块负责在内核级别拦截系统事件，例如保护错误和系统崩溃（即故障安全处理器/“蓝屏处理器”）。

概述

System Module旨在捕获系统事件，特别是内核级别的保护错误与系统崩溃情形。为此，可预先注册处理器（回调函数或实时任务），以便在事件发生时启动。这样就能够实现故障安全处理器（即“蓝屏处理器”），并在出现错误时执行预定义的应对措施。

特征

- 内核级保护错误与系统崩溃事件的拦截：利用回调函数和实时任务进行捕捉。
- 故障安全处理器（“蓝屏处理器”）的实现：确保在系统遇到错误时，能够触发并执行预定的响应策略。

实时系统

Timer Module

实现系统时间的高精度校准检测以及短时间延迟。

NEW!

概述

Timer Module通过其时钟功能，构成了所有实时任务的基础。它提供了对系统中所有硬件时间基准的校准访问。时间信息能够自由转换为任意时间格式，包括用户自定义格式。根据所使用的硬件，短时间延迟可精确到几纳秒之内。此外，它还能基于Windows机制生成非实时的、毫秒级分辨率的简单计时器。

特征

- 检测系统中的所有运行时钟基准。
- 不同计时器/时钟的校准。
- 系统时钟的长期同步。
- 监控系统时间，支持多种时间格式及用户自定义格式，解析度高达0.1微秒。
- 提供0.1微秒步进的高精度短时间延迟。
- 支持用户自定义时间格式编程。
- 通过内部96位运算可靠防止溢出。
- 基于Windows机制创建毫秒级分辨率的简单计时器（非实时功能）。

NEW!

RealTime Tasking Module

用于编写高频实时计时器例程及基于优先级、可抢占式的实时多任务处理。

概述

RealTime Tasking Module是对“[Timer Module](#)”的补充，它使得程序员能够设置触发应用程序代码的不同方式的计时器。可以选择程序执行的上下文环境。可信号对象可以在应用层或内核层的事件、回调，或是实时任务。此外，模块还包括了一个基于优先级的、可抢占式的实时多任务系统，用于单个实时任务之间的相互优先级排序。这使得能够精细控制哪些进程优先于其他进程执行，暂停不那么重要的进程，以及哪些任务可以被更重要任务中断，这一机制等同于纯实时操作系统（RTOS）中的机制。

特征

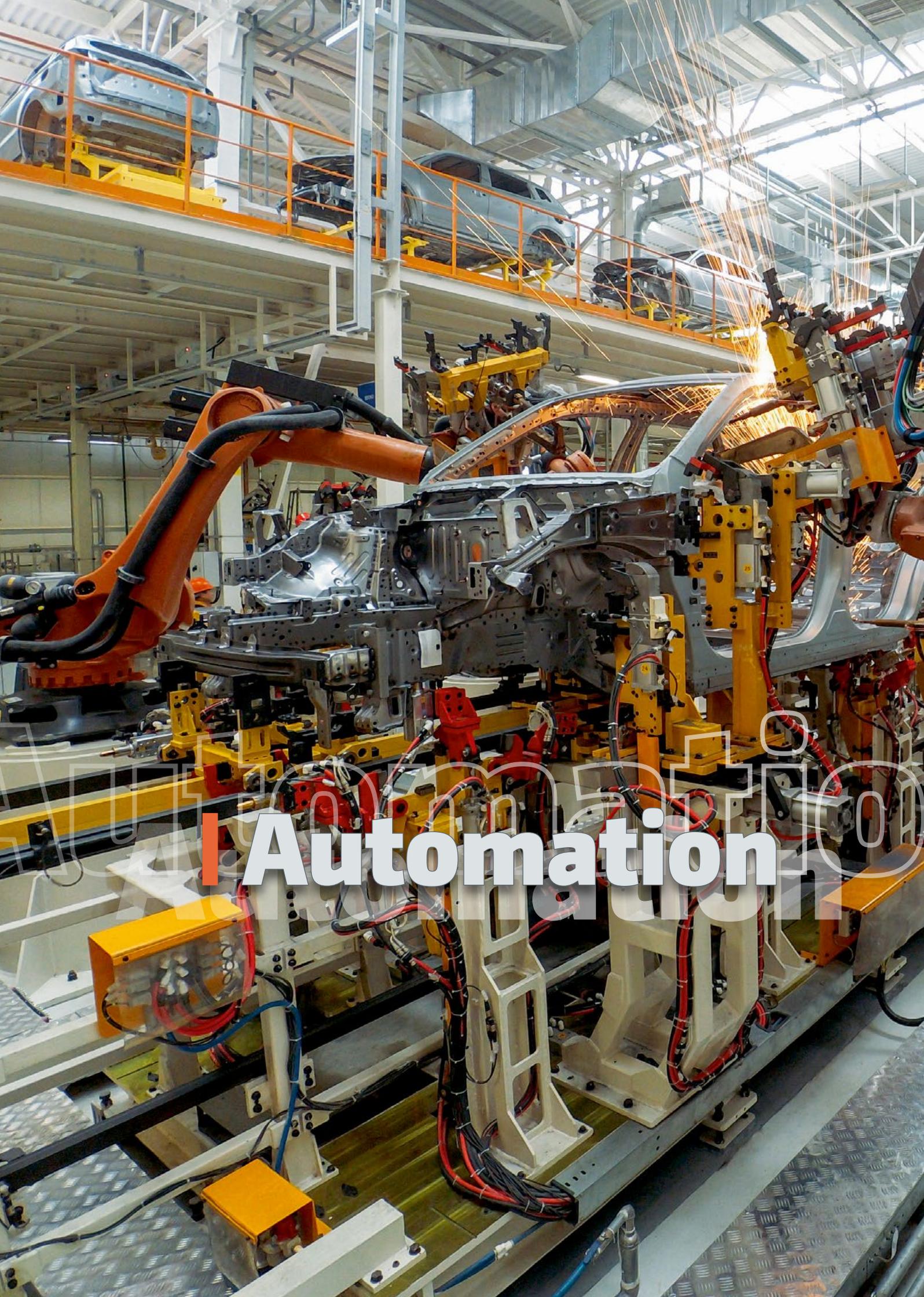
实时功能

- 开发高频实时计时器例程
- 周期性或一次性计时器编程选项
- 应用层级回调实现简易调试与测试
- 计时器直接启动、退出或终止功能
- 看门狗机制轻松部署
- 启动时间设定精度至0.1微秒（例如，与其他进程同步）
- 计时器频率高达20千赫兹乃至更高
- 微秒单位内低抖动
- 计时周期动态可调

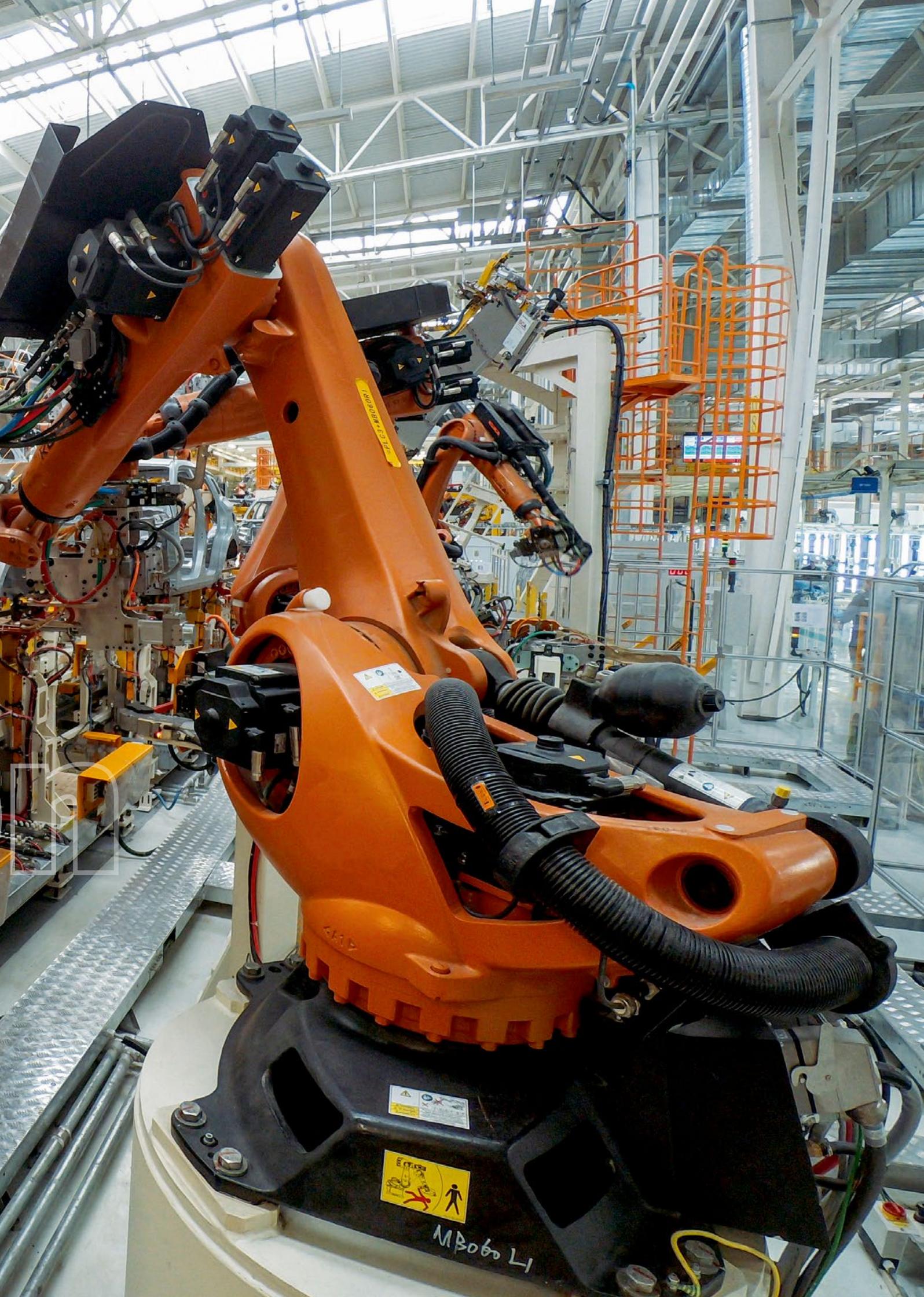
多任务处理

- 抢占式实时系统
- 支持编程高达255级不同优先级的实时任务
- 优先级动态调整
- 同优先级任务以“轮询”方式执行
- 包含优先级继承以防优先级倒置
- 实时信号量，用于实时任务间同步
- 实时事件，触发外部任务
- 任务可挂起、继续、触发、退出及终止
- 任务延迟功能（0.1微秒精度）
- CPU核心独占实时模式运行，避免Windows系统影响
- 极短的任务切换延迟
- 速度闭环模式，实现高精度循环执行
- 需搭配[Timer Module](#)使用

PTP Module	采用精密时间协议实现实时同步
概述	<p>PTP Module使得分布式系统中的多个网络参与者能够实现精准的时间同步。Kithara实时系统能够生成偏差在亚微秒范围内的精确时间戳，以此来统一所有参与者的时钟。API提供了对BMCA（最佳主时钟算法）的访问权限，该算法会根据可配置参数确定主时钟。另外，主时钟和从时钟也可手动设定。</p>
特征	<ul style="list-style-type: none"> ■ Precise synchronization of multiple 多网络参与者的精密同步 ■ 亚微秒级偏差的准确时间戳 ■ 可配置的BMCA（最佳主时钟算法） ■ 手动指定主从时钟 ■ 需搭配 Network Module 使用
扩展功能	<p>NMEA Extension 实现实时通信与NMEA 0183协议的同步。</p> <p>gPTP Extension gPTP Extension在PTP Module基础上增加了对“广义精密时间协议”（IEEE 802.1AS标准定义）的支持，这是一种简化的PTP配置文件。gPTP Extension允许分布式系统内所有时钟实现高度精确的同步，但其功能设置更为直接，适用于更广泛的应用场景。例如，无需单独配置每个时钟，它们都遵循标准化设置，并且BMCA（最佳主时钟算法）的配置更为简便。</p> <ul style="list-style-type: none"> ■ 需依赖PTP Module ■ 多网络参与者的精密同步 ■ 简化的PTP配置文件 ■ 所有时钟标准化 ■ BMCA（最佳主时钟算法）配置更为简易



Automation



MELCO AMB060R

E-STOP

MB060 L1

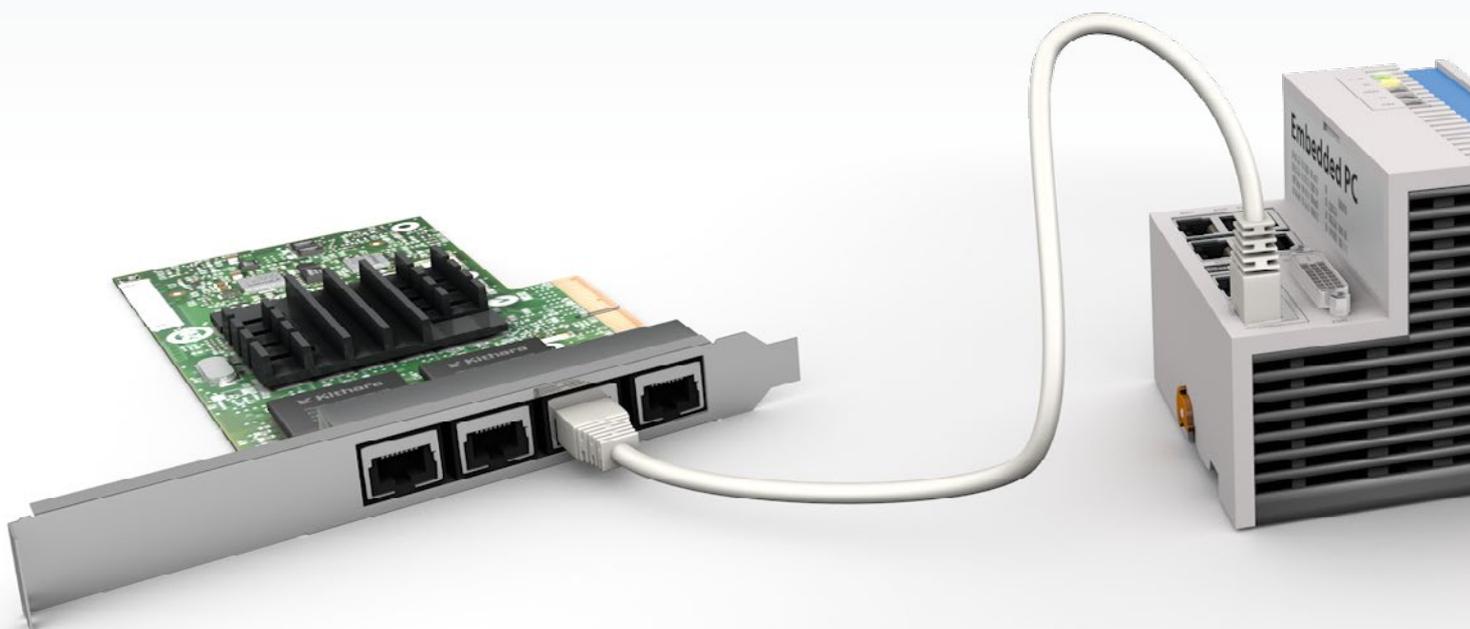
面向PC控制的自动化软件解决方案

在自动化技术领域，标准化协议已成为不可或缺之标准。作为最受欢迎的现场总线之一，开放且高速的工业以太网变体EtherCAT已确立其地位，并持续展现出强劲的增长势头。为此，Kithara Real-Time Suite融入了一款高性能的EtherCAT主站。

其“硬”实时能力使得工厂自动化得以实现仅50微秒或更短的循环周期。该主站还提供了所有必需机制，包括自动检测连接拓扑、支持从站作为分布式时钟(DC)，以及热插拔能力和电缆冗余等特殊功能。

在实施安全应用时，可集成Safety-over-EtherCAT (FSoE)。在此情况下，EtherCAT主站负责在安全输入、输出及安全逻辑组件间的数据包交换。这样，即可实现高达SIL3级别的应用——无需额外布线，如紧急停机线路。

此外，还能将PC作为EtherCAT从站集成，以构建灵活节点，从而实现可扩展的EtherCAT网络。为此，可采用特殊PCIe卡将PC嵌入EtherCAT拓扑中并作为从站运行。此类EtherCAT PC从站得益于常规PC的图形界面和输入选项，可根据不同任务进行专业化配置，如处理能力的分配或调整。

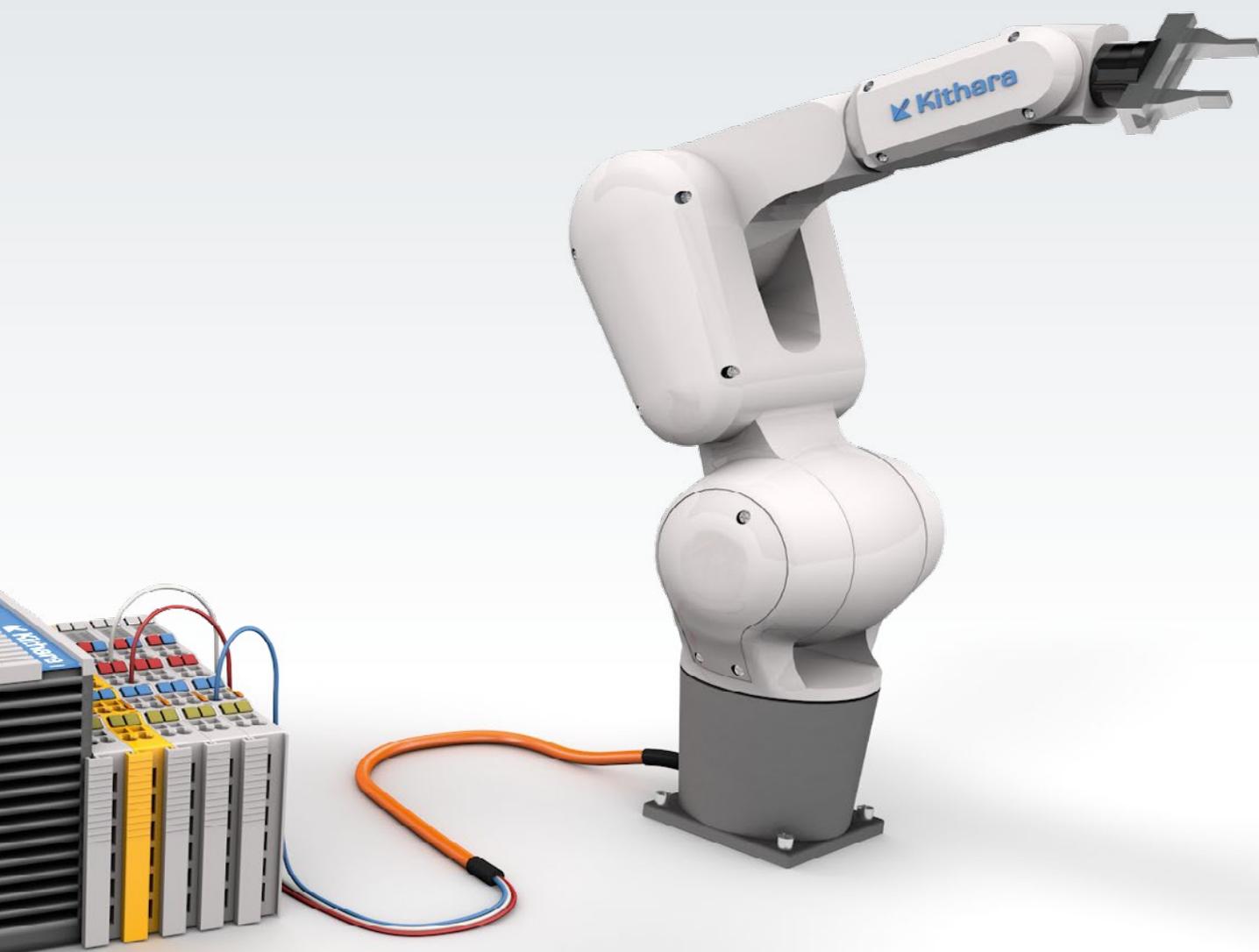


同时提供的CANopen主站功能，能够让用户操作CAN接口，或是将其融入更高级别的EtherCAT网络中。

为了将EtherCAT拓扑结构嵌入更高级别乃至企业网络，我们提供了EtherCAT自动化协议(EAP)。这一协议使得相同或不同层级的设备及整个生产环节能够以高达200 Gbit/s的速度交

换数据，不仅支持主站间的通信，还能够连接更高级别的生产管理系统（MES、ERP）及企业部门（物流、分销等）。通过这种自动化设施的连接，EAP成为了实现信息物理系统运行的基础。

NEW!



模块



EtherCAT Module
 ECAT PC Slave Dev. Ex.
 ECAT Autom. Protocol
 CANopen Module

EtherCAT Module

实时EtherCAT主站

概述

Kithara EtherCAT Master基于优先级驱动、可抢占式的实时多任务系统构建。利用多CPU核心的硬件并行性，可将多个实时任务分配至不同的CPU核心上执行。因此，无论是在专用模式下完全占用的CPU核心上，还是在高度可扩展的实时处理环境中，都能达到极致的实时性能。

特征

- 独立实时EtherCAT主站
- 借助XML文件或SII协议自动识别EtherCAT拓扑
- 实时过程数据通信（循环I/O数据交换）
- 短周期时间：低至50微秒或以下
- 热插拔与线缆冗余支持
- 支持RealTek与Intel网络接口
- 全面支持各厂商的I/O端子、伺服逆变器等设备（如Beckhoff）
- EtherCAT主站模块化结构：基础模块+附加选项
- 过程数据通信（PDO）
- 服务数据通信（SDO）
- 邮箱通信
- 需搭配 [Network Module](#) 使用

扩展功能

EtherCAT PC Slave Device Extension

借助EtherCAT PC Slave Device Extension，标准PC可作为EtherCAT从站加入EtherCAT网络。在此之前，仅限于PCIe从卡形式的主站侧能执行与EtherCAT网络的复杂PC端通信。集成PC从站后，可更聚焦地构思、整合与调整精密的自动化流程。



- 统一API兼容EtherCAT主站与EtherCAT EAP
- 过程数据与服务数据通信 (PDO/SDO)
- 文件传输 (FoE)
- 自定义PDO映射创建
- SII数据 (EEPROM) 可初始化为自定义值 (如供应商ID、产品ID、版本号)

硬件支持

Beckhoff – EtherCAT 从设备：

- FC1100, PCI EtherCAT 从站卡
- FC1121, PCIe EtherCAT 从站卡
- CX5000 CCAT (适用于Beckhoff CX50xx、CX20xx、CX51xx系列嵌入式PC)

ESD – EtherCAT从站卡

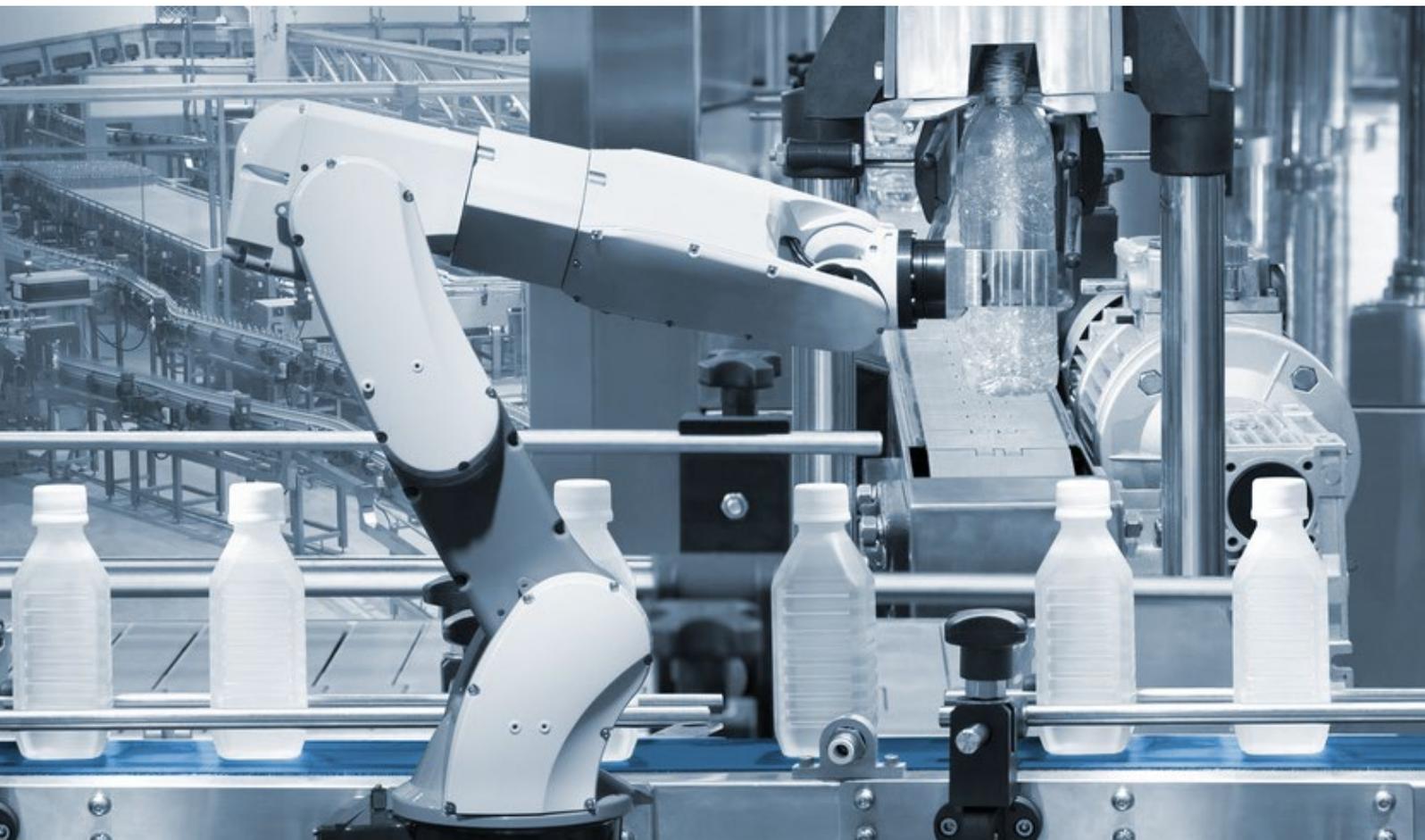
- ECS-PCIe 1100, PCIe EtherCAT从站卡

Distributed Clocks (DC)

Distributed Clocks用于实现总线上所有设备的软件时间同步。初始化过程中，主站会测定至各个从站的延时并相应设置其时钟。为了修正漂移，主站会定期通过环形拓扑向所有从站发送消息，确保长期精确同步。

Ethernet over EtherCAT (EoE)

通过Ethernet over EtherCAT，任何以太网设备均可通过交换机端口接入EtherCAT网络。标准以太网通信通过EtherCAT隧道化传输，使得主站能够在不影响过程数据通信的前提下对其进行优化。部分从站还提供通过EoE访问Web界面（如配置界面）。



File access over EtherCAT (FoE)

File access over EtherCAT允许对网络设备进行简单的文件操作，例如，可用于将统一固件上传到 EtherCAT 网络中的多个设备。该协议特意保持简单，以便也支持引导加载程序。

Servo Profile over EtherCAT (SoE)

The Servodrive Profile over EtherCAT提供了对Sercos接口的支持，可用于实现复杂的运动控制应用。伺服驱动器的Sercos配置文件及其到EtherCAT的映射遵循IEC 61800-7标准。

Hot Connect (HC)

Hot Connect功能允许在操作前或操作期间，预配置的段（从站组或单个从站）被移除或添加到数据传输中，从而实现拓扑的灵活调整。

Cable Redundancy (CR)

Cable Redundancy功能使得即使发生意外的通信中断（如线路断裂），操作仍能继续。为此，过程数据通过第二个以太网端口经由拓扑结构冗余发送。

Safety over EtherCAT (FSoE):

经过TÜV认证的Safety over EtherCAT安全协议，实现了符合IEC 61508标准的安全等级SIL 3的EtherCAT功能。安全协议不会对传输速度或周期时间造成任何限制，因为EtherCAT被用作单通道通信媒介。

EtherCAT Autom. Protocol	从命令层开始的EtherCAT实时通信
概述	EtherCAT Automation Protocol实现了网络内所有终端间的实时通信，使自动化系统领域达到了前所未有的互联水平。这涵盖了基于PC的控制系统、机器、测试台、输送带、机器人、质量保证设施或MES（制造执行系统）。
特征	<ul style="list-style-type: none"> ■ 传输速率为1000 Mbit/s、10 Gbit/s或200 Gbit/s ■ 过程数据和服务数据交换 (PDO/SDO)、文件传输 ■ 需要实时 EtherCAT Module

NEW!

CANopen Module

实时CANopen Master

概述

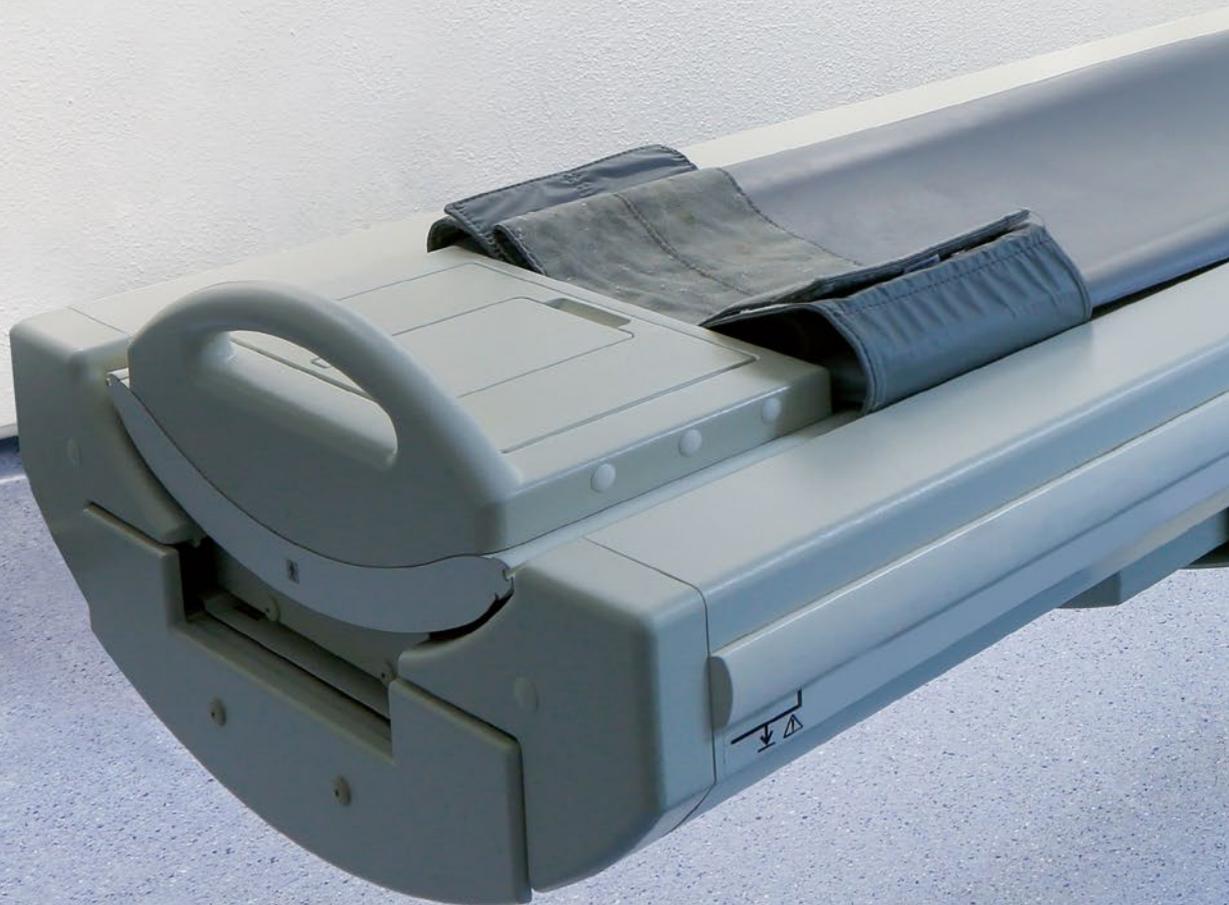
Kithara RealTime Suite的CANopen Module包括一个基于PC的CANopen自动化协议主站。它基于CANopen Module, 需要适当的硬件。

特征

- CANopen拓扑结构的自动识别
- CANopen识别状态的管理
- 过程数据和服务数据通信 (PDO+SDO)
- 邮箱通信
- 需要 [CAN Module](#)



Machine Vision





实时机器视觉

众多自动化产线均倚重机器视觉技术，无论是用于制造、包装、监控还是质量保证。然而，应用于工业相机的视觉接口标准及图像处理库的使用，要求极低的响应时间以应对多样化的特定任务。Kithara RealTime Suite在实时图像采集与处理领域处于领先地位。

在封闭的实时循环中集成自动化协议的支持，其决定性优势在于能将基于算法的图像分析结果直接传输至流程中，无需脱离实时环境。这意味着即使是传感器/执行器级别的系统也能直接由处理过的图像数据进行操控。

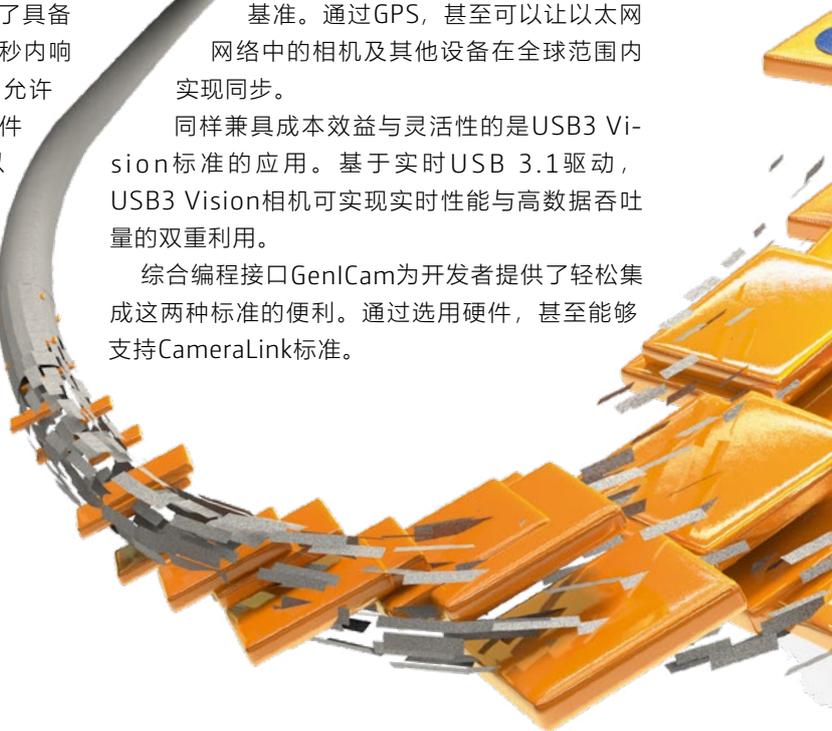
针对基于以太网的摄像系统，我们提供了具备实时能力的GigE Vision驱动，能够在数微秒内响应传入的完整图像。同时支持链路聚合，允许在多个通道上并行传输。依据所采用的硬件配置，数据传输速率可高达每秒吉字节以上。GigE Vision的优势与以太网技术相契合，意味着能够采用经济高效、可互换的标准化硬件，以及高达10 Gbit/s的数据传输速率和长距离的电缆长度。随着GigE Vision 2.0的推出，引入了精确时间协议（IEEE 1588）。



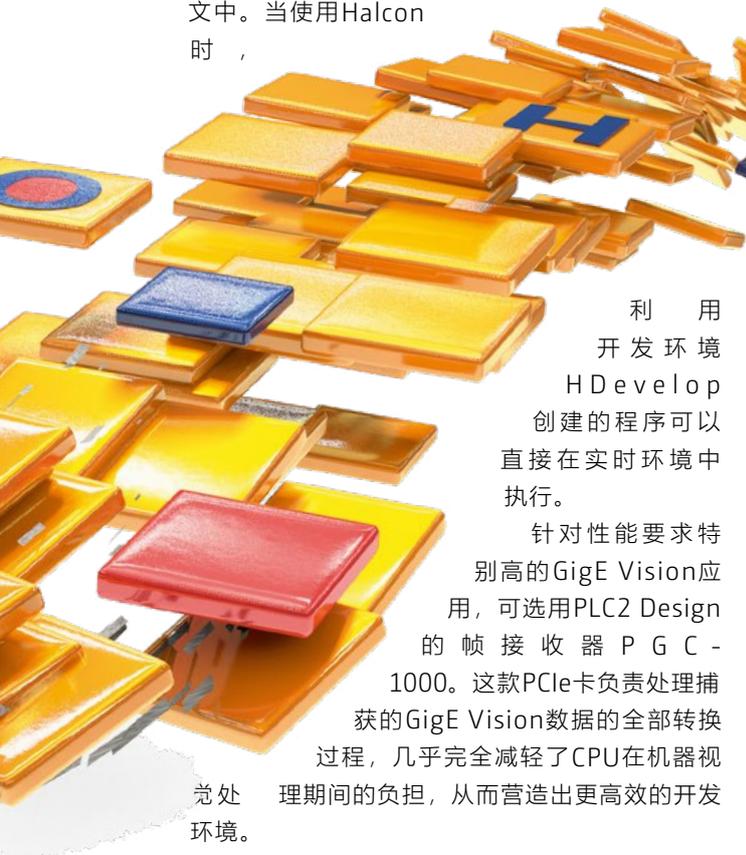
这一协议使得多台相机能够设置为PTP模式，实现彼此间精确到微妙级且实时的同步。实现此功能的方式是设定一台设备作为主设备，其时钟为其他设备的时间戳设定基准。通过GPS，甚至可以让以太网网络中的相机及其他设备在全球范围内实现同步。

同样兼具成本效益与灵活性的是USB3 Vision标准的应用。基于实时USB 3.1驱动，USB3 Vision相机可实现实时性能与高数据吞吐量的双重利用。

综合编程接口GenICam为开发者提供了轻松集成这两种标准的便利。通过选用硬件，甚至能够支持CameraLink标准。

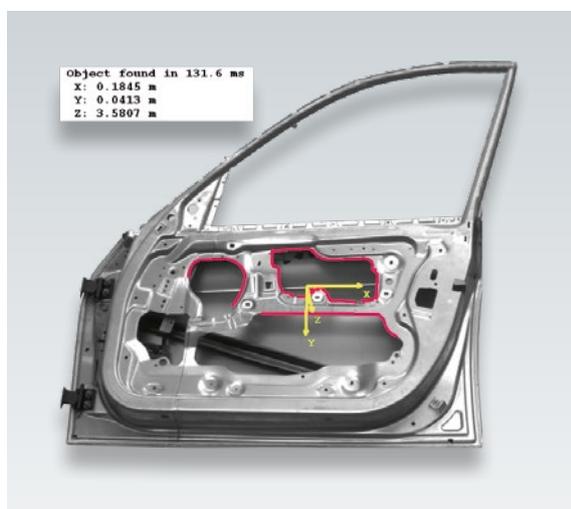


借助外置帧接收器，它能将CL控制信息和数据包转换为GigE Vision格式，所有CL配置（包括“全”配置）均可通过10 Gbit以太网连接整合进实时环境，同时也将原本较短的电缆长度提升至以太网级别。为了即时进一步处理和分析图像数据，可运用特定算法或诸如Halcon、OpenCV等外部图像处理库。为此，相应库会被加载到实时上下文中。当使用Halcon时，



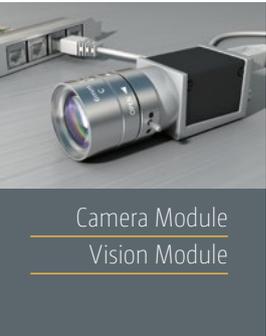
利用开发环境HDevelop创建的程序可以直接在实时环境中执行。

针对性能要求特别高的GigE Vision应用，可选用PLC2 Design的帧接收器PGC-1000。这款PCIe卡负责处理捕获的GigE Vision数据的全部转换过程，几乎完全减轻了CPU在机器视觉处理期间的负担，从而营造出更高效的开发环境。



在制造业和质量保证领域，特别是在汽车行业中，相机系统的应用显著增加。这通常涉及到对所拍摄图像的高精度识别与准确解读。若缺乏“硬”实时能力来执行图像捕获与处理，这些将无法实现。Kithara为此类任务提供了必要的技术支持基础。

模块



Camera Module

实时捕获GigE Vision与USB3 Vision相机图像

概述

Camera Module旨在实现实时捕获遵循GigE Vision或USB3 Vision标准的工业相机图像数据。它提供了所有必要的检测、管理功能以及创建实时流的功能。大容量缓冲内存结合实时网络驱动程序，有效防止数据包丢失。对接收到的GigE Vision或USB3 Vision图像的响应在实时环境的微秒级瞬间即刻进行，从而能够立即触发实时任务，执行即时控制反应。

得益于实时驱动程序，Camera Module根据接口和系统配置，可实现最大数据吞吐量，甚至达到理论上限。关于图像数据处理，请参阅 [Halcon Extension](#) 和 [OpenCV Extension](#)。

图像处理的结果可以在无延迟的情况下直接在实时环境中传递给进程，例如通过EtherCAT或CANopen。因此，自动化解决方案可以轻松地与Kithara RealTime Suite中的其他多项功能相结合。

特征

一般特征

- 实时状态下使用GigE Vision或USB3 Vision相机进行图像采集
- 基于事件或请求触发
- 支持多相机同时工作、热插拔及GenICam 2.0规范配置
- 全面兼容所有GigE Vision标准相机

控制功能

- 自动连接管理
- 完全访问相机配置空间
- 读写相机内存
- 自定义心跳超时设定
- 软件控制图像采集的启动与停止
- 软件设定采集模式
- 自动下载GenICam-XML配置文件
- 访问GenICam特性以配置您的相机
- 错误处理机制

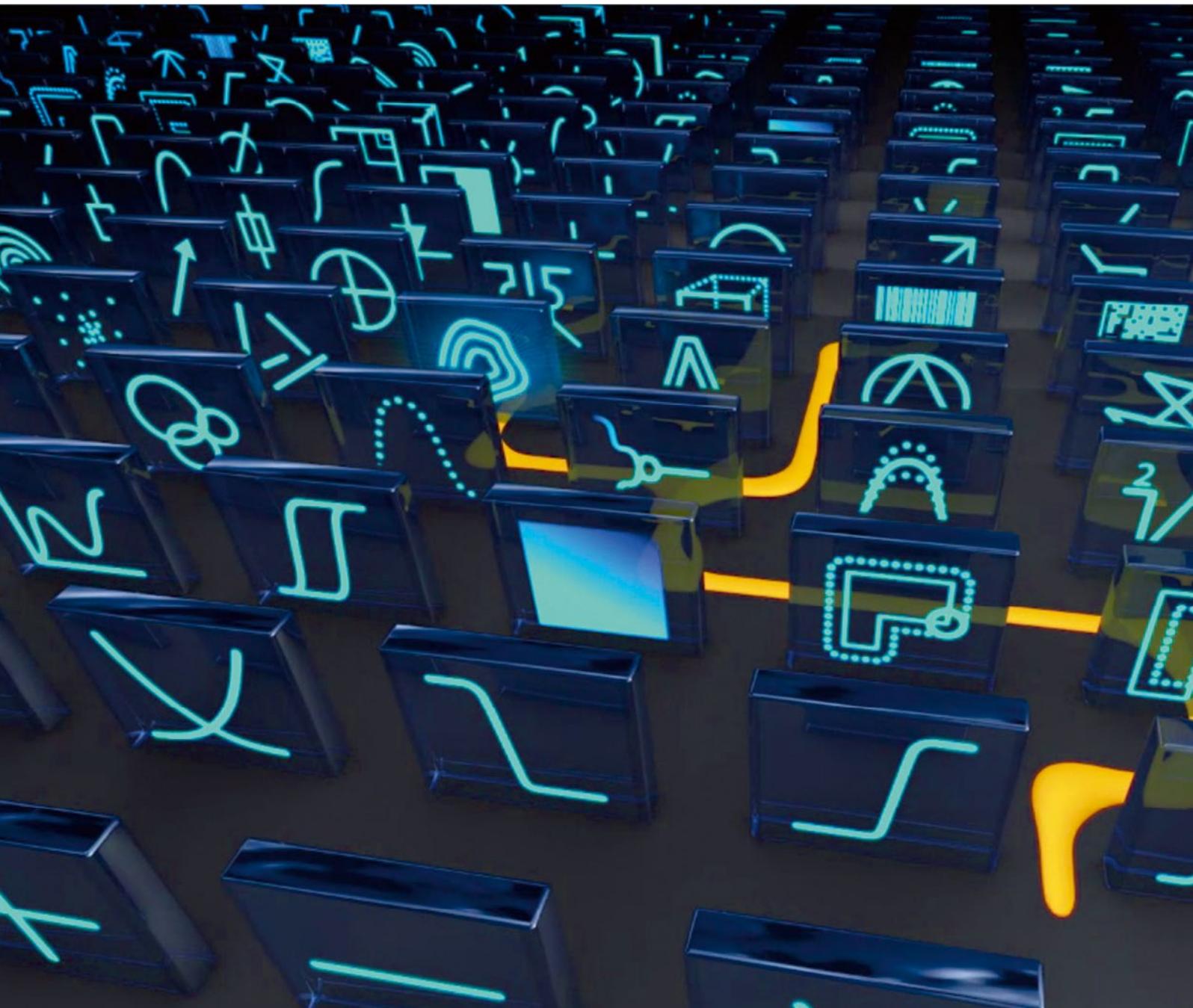
数据流特性

- 每个相机可支持一个或多个数据流
- 实时图像采集
- 高效利用内存进行图像捕获，避免不必要的复制
- 自定义缓冲区大小与缓冲区数量
- 提供错过或不完整图像的相关信息

GenICam功能

- 对所有类别进行编号
- 设置与获取所有可访问的相机特性，包括图像格式、像素格式、帧率、曝光时间等

Camera Module需配合实时接口的其他模块使用，例如，GigE Vision需结合 [Network Module](#) 而USB3 Vision则需搭配 [USB RealTime Module](#)。



扩展功能

Camera Acceleration Extension

Camera Acceleration Extension为PLC2的GigE Vision采集卡PGC-1000提供了高效的实时驱动程序。将其嵌入Kithara RealTime Suite后，PGC-1000能够在机器视觉应用中几乎完全减轻CPU在图像采集过程中的负担。例如，它能够实现在实时环境下同时捕获多达四路每路10 Gbit/s的GigE Vision相机视频流。通过Kithara的PTP Module同步多台相机的实时功能亦得以实现。

- 对PGC-1000提供实时支持
- 几乎完全卸载CPU在GigE Vision图像采集上的负担
- 连接速度高达4×10 Gbit/s或64×1 Gbit/s
- 通过PTP模块同步多台相机
- 需要配套使用 [Camera Module](#) 和 [Network Module](#)
- 要求使用PLC2 Design GmbH的PGC-1000

Vision Module

基于Halcon或OpenCV的实时图像处理

概述

Halcon和OpenCV是集成于Kithara实时系统中的编程库，包含用于图像处理和机器视觉的算法。它们允许对处理后的图像做出即时控制反应，并将其整合进自动化系统，如EtherCAT或CANopen。

特征

- 实时利用Halcon或OpenCV等图像处理库
- Halcon许可证需单独向MVTec获取
- 支持Halcon版本10至13、8.11、20.11及OpenCV 3.0、3.4、4.1
- 包含 [Halcon Extension](#) 和 [OpenCV Extension](#)
- 需要 [RealTime Tasking Module](#) 和 [Library Module](#)









Communication

实时硬件访问与通讯

Kithara RealTime Suite的多个模块用于PC硬件的连接。通过访问PC的I/O寄存器、物理内存及中断处理，可为单个硬件组件及插卡创建驱动。对于许多常见的通讯接口，已有高效、即用型的驱动程序准备就绪。

尤其是在工业领域，对具有高实时性的以太网应用的需求显著增长。为此专门开发的驱动程序支持Intel和RealTek的所有常见以太网网络控制器，数据传输速率高达200 Gbit/s。以太网的所有功能、特性和优势，比如使用灵活且经济的标准硬件，均可照常利用，同时增添实时能力。如此一来，可有效利用所应用控制器的特殊硬件特性，例如巨型帧、因中断优化而增强的数据流控制等。

NEW!

为了实现实时性能，网络控制器被直接访问，从而充分利用连接的全部带宽，并立即对传入的数据包做出响应。为了确保高性能的网络通信，所有可能降低传输速率或负面影响即时反应的冗余Windows机制都被绕过。在实施套接字通信时，提供了针对数据报(UDP)或面向流(TCP)通信应用的特殊驱动程序层。

为了实现与USB 3.1版本及以下的快速通信，Kithara提供了一种特别编写的、易于应用的可扩展主机控制器接口(xHCI)访问方式。凭借与标准化硬件结合的成本效益实现以及良好的传输速率，USB已成为最广泛传播的通信标准之一。自USB 3.0引入的SuperSpeed模式，实现了更高的数据传输速率和全双工通信，使得USB即便在先进的工业领域也能得到高效应用。



Thunderbolt技术同样支持实时使用，可通过此接口连接的组件以最低延迟实现即时通讯。这种Thunderbolt实现还允许实时使用相应的硬件扩展，如将PCIe卡连接到笔记本电脑上，从而提升了基于软件的实时解决方案的机动性，例如在现场直接对机器进行测试或测量时。

此外，串行UART和COM接口亦可被访问，支持通用标准RS-232、RS-422和RS-485；波特率可达115,200及921,600位/秒。这还包括从专用CPU上的实时任务运行UART功能，以及直接控制握手和信号线路。

一个特殊的附加编程接口允许处理操作系统函数调用，以便在内核级别进行设备驱动程序的通信。甚至像“虚拟”COM接口这样的功能，也可以通过这种方式生成。此外，我们也支持National Instruments的一系列高性能多功能卡，这些卡可通过数字和模拟输入/输出进行访问。

在时间敏感硬件通信的实际应用方面，典型实例包括但不限于：采用工业摄像机（GigE Vision、USB3 Vision）的机器视觉，或是结合EtherCAT技术的复杂自动化任务。



模块



Network Module

用于快速以太网通信的本机实时驱动程序

概述

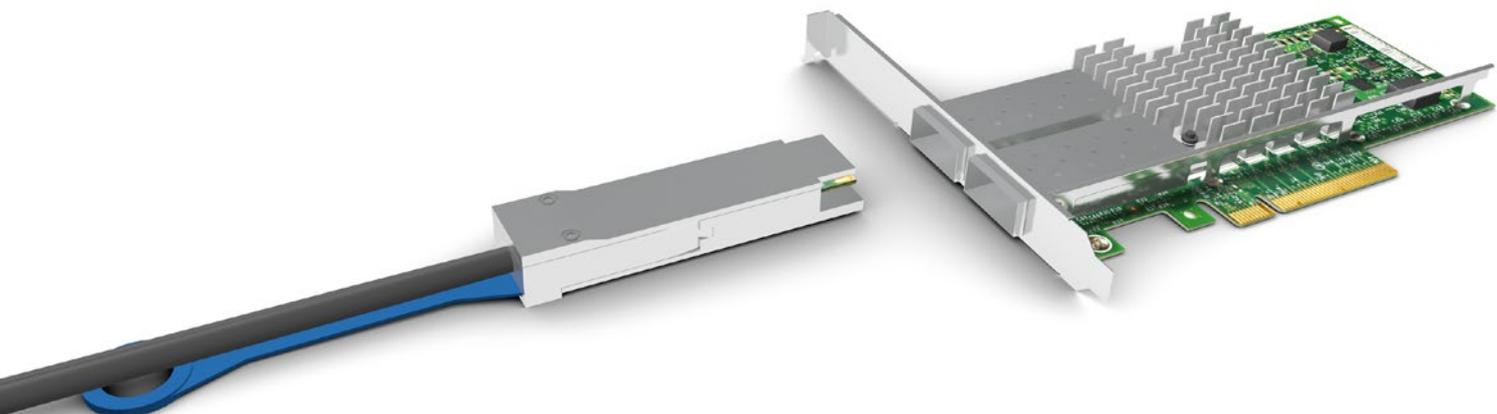
Network Module为支持的网络控制器提供了本机实时以太网驱动，以实现通过原始以太网、UDP 和 TCP 的通信。通过预先阻止所有Windows机制的干扰，可实现高速传输和即时响应，从而达到高性能的网络通信效果。几乎支持所有英特尔和RealTek的网络控制器（参见：支持的硬件列表）。

特征

- 实时以太网通信，速率高达200 Gbit/s
- 接收时即刻传输与响应
- 可直接在实时环境中进行传输与接收
- 支持原始以太网及多种以太网协议，包括IP、UDP和TCP
- 基于事件或请求的以太网通信模式
- 巨型帧支持，最大可达16128字节（依据网卡而定）
- 支持IP和MAC多播、广播以及混杂模式接收所有数据包
- 数据包接收可通过回调函数、中断上下文或轮询过程实现
- 数据包传输最多可设置4个优先级
- 自动通过ARP支持确定内存地址
- 提供CRS计算与字节序转换功能
- Network Module 附加组件：Network Port Extension,
 - › Network PTM Extension, Network Data Rate Extension,
 - › Network BroadR-Reach Extension
- 需要配合使用 › RealTime Tasking Module

- Network Module
- USB RealTime Module
- USB Driver Module
- Serial/UART Module
- Device Module
- IoPort Module
- Memory Module
- Interrupt Module
- MultiFunction Module

NEW!



为了实现实时性能，需将网络卡的标准Windows驱动替换为Kithara RealTime Suite特别开发的实时驱动。几乎涵盖了所有英特尔和RealTek提供的网络控制器。

NEW!

200/100 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 (例如Digitus USB 3.0 Ethernet Adapter)

100 Mbit/s

Intel

- 8255x
- 82562

RealTek

- 8139B, 8139C, 8139D

Beckhoff

- CX5000 CCAT(Beckhoff CX50xx, CX20xx, CX51xx 嵌入式电脑)

扩展功能

Network PTM Extension

Network PTM Extension 通过UDE (USB设备仿真) 生成虚拟USB以太网适配器, 以便在实时环境与Windows之间进行通信, 并支持端口穿透或端口镜像功能。端口穿透使Windows能够直接与硬件或Kithara实时环境通信, 而镜像则用于记录实时数据流量, 并将其提供给Windows以便使用Wireshark等工具进行分析。

- 通过UDE创建虚拟USB以太网适配器
- 端口穿透实现Windows与硬件或实时环境间的直接通信
- 端口镜像记录实时数据流量, 并在Windows中展示 (如使用Wireshark)
- 支持所有基于以太网的技术: 套接字通信、EtherCAT、GigE Vision、PTP及BroadR-Reach

Network BroadR-Reach Extension

BroadR-Reach (100BASE-T1) 是一种专为汽车网络设计的物理层以太网传输技术。



USB RealTime Module

通过直接访问xHCI实现与USB设备的实时通信

概述

Kithara RealTime Suite的USB RealTime Module确保了所有基于USB的接口和设备（如摄像头）能够在最快的数据传输速率下实现快速响应。为此，Kithara RealTime Suite提供了一个特别开发且易于应用的接口，通往可扩展主机控制器接口（xHCI）。其基本工作原理是在实时环境下直接且唯一地访问xHCI控制器。

特征

- 通过xHCI访问实现对USB 3.1设备的实时通信
- 独占使用xHCI控制器
- 直接从实时多任务上下文访问
- 达到可能的最低响应时间
- 实现最高数据吞吐量
- 通过可自定义缓冲区大小可靠防止数据丢失
- 支持低速、全速、高速和超高速模式
- 支持控制、批量、中断和等时传输类型
- 响应所有即插即用事件及电源管理事件
- 需要配合使用 [RealTime Tasking Module](#)

硬件支持

关于USB RealTime module，所有符合xHCI 1.1规范的USB 3.1主控制器均可使用。

Kithara RealTime Suite当前支持以下厂商的xHCI控制器：英特尔、AMD、Renesas、Etron、NEC和VIA。

USB Driver Module

通过Windows驱动堆栈与USB设备通信

概述

Kithara RealTime Suite的USB Driver Module通过简单可适应的功能实现了与USB设备的通信。它基于Windows USB驱动堆栈，仅具有有限的实时能力。对于“严格”的实时应用，我们推荐使用 [USB RealTime Module](#)。支持所有USB设备，包括USB 3.1。

特征

- USB设备驱动开发
- 应用程序或内核级别访问内部或外部USB设备（包括USB 3.1）
- 支持低速、全速、高速
- 支持控制、批量、中断和等时传输
- 响应所有即插即用事件及电源管理事件

Serial/UART Module

易于应用的快速通信功能

- 通过针对UART16550兼容硬件的特殊硬件驱动，在实时环境中实现串行通信
- 直接控制信号线和握手线
- 所有接口事件的处理程序均在实时环境中注册
- 提供不依赖实时性的适用于所有基于Windows的COM端口的备用API

概述

Kithara RealTime Suite的Serial Module提供了易于使用的功能，以实现通过串行UART和COM接口的快速通信。UART功能需要相应的UART-16550兼容硬件以及实时驱动程序的支持。在实时模式下，UART功能既可以在实时任务中使用，也可以在专用CPU上运行。

对于COM通信，任何已安装Windows驱动程序的接口都受支持。在这种情况下，COM功能通过Windows机制运行，这意味着不能应用实时能力。

特征

- 通过特殊硬件驱动在内核级别实现串行实时通信
- 直接可控的手动线和信号线
- 所有接口事件的处理程序也可在实时上下文和专用CPU上注册
- 使用UART功能时，需要UART-16550兼容硬件
- 使用COM功能时，应用级别的串行通信允许系统中所有COM接口的使用

支持硬件列表

为了实现实时功能，应用于串行接口的Kithara RealTime Suite需要特殊的实时驱动程序。当前支持以下接口卡：

- Standard COM Interface (PNP0501)

PCIe

- StarTech PEX2S952, 2×RS232
- Longshine LCS63210, 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	
	生成虚拟接口
概述	<p>Kithara RealTime Suite的Device Module通过操作系统功能为终端用户提供设备驱动程序的API。Device Module允许在设备驱动程序的上下文中，在内核级别处理此类函数调用。</p> <p>Device Module还允许生成虚拟接口——在特殊情况下甚至可以作为虚拟COM接口。</p>
特征	<ul style="list-style-type: none"> ■ 为设备通信提供Windows API (ReadFile, WriteFile, DeviceIoControl) ■ 设备名称可自由配置，例如虚拟“COM端口”

Hardware Access

IoPort Module	直接访问I/O端口
概述	<p>Kithara RealTime Suite的IoPort Module允许直接从Windows应用程序访问PC的I/O端口。借助IoPort Module, 用户可以解锁对所需I/O端口的直接访问。</p> <p>直接访问I/O端口绝不意味着降低稳定性。无需单独开发内核驱动程序, 这使得用户能够更专注于应用程序本身, 仅通过定义的入口点访问硬件。</p>
特征	<ul style="list-style-type: none"> ■ 从应用程序或DLL直接访问PC的所有I/O寄存器 ■ 确定PCI配置数据 ■ 获取设备资源信息
Memory Module	物理内存访问
概述	<p>Kithara RealTime Suite的Memory Module允许直接访问物理内存。</p> <p>为了实现直接访问, 物理内存会在应用空间中映射出来。内存模块提供了两种不同的物理内存访问机制, 区别在于所访问的内存是位于外部硬件上还是PC的主内存上。</p>
特征	<ul style="list-style-type: none"> ■ 外部物理内存访问 (双端口RAM) ■ 为外部硬件提供PC内存 (DMA内存)
Interrupt Module	外部硬件控制
概述	<p>为了控制外部硬件, 中断是必不可少的。它们常用于将状态信息传达给PC扩展卡等设备。为了记录这些事件, 设备驱动程序和技术应用需要安装合适的中断处理程序。</p>

特征

- 在应用或内核级别处理硬件中断
- 使用操作系统机制进行中断处理
- 通过使用 **RealTime Tasking Module**，可在PCI或PCIe硬件上应用实时中断
- 需要 **IoPort Module**

MultiFunction Module

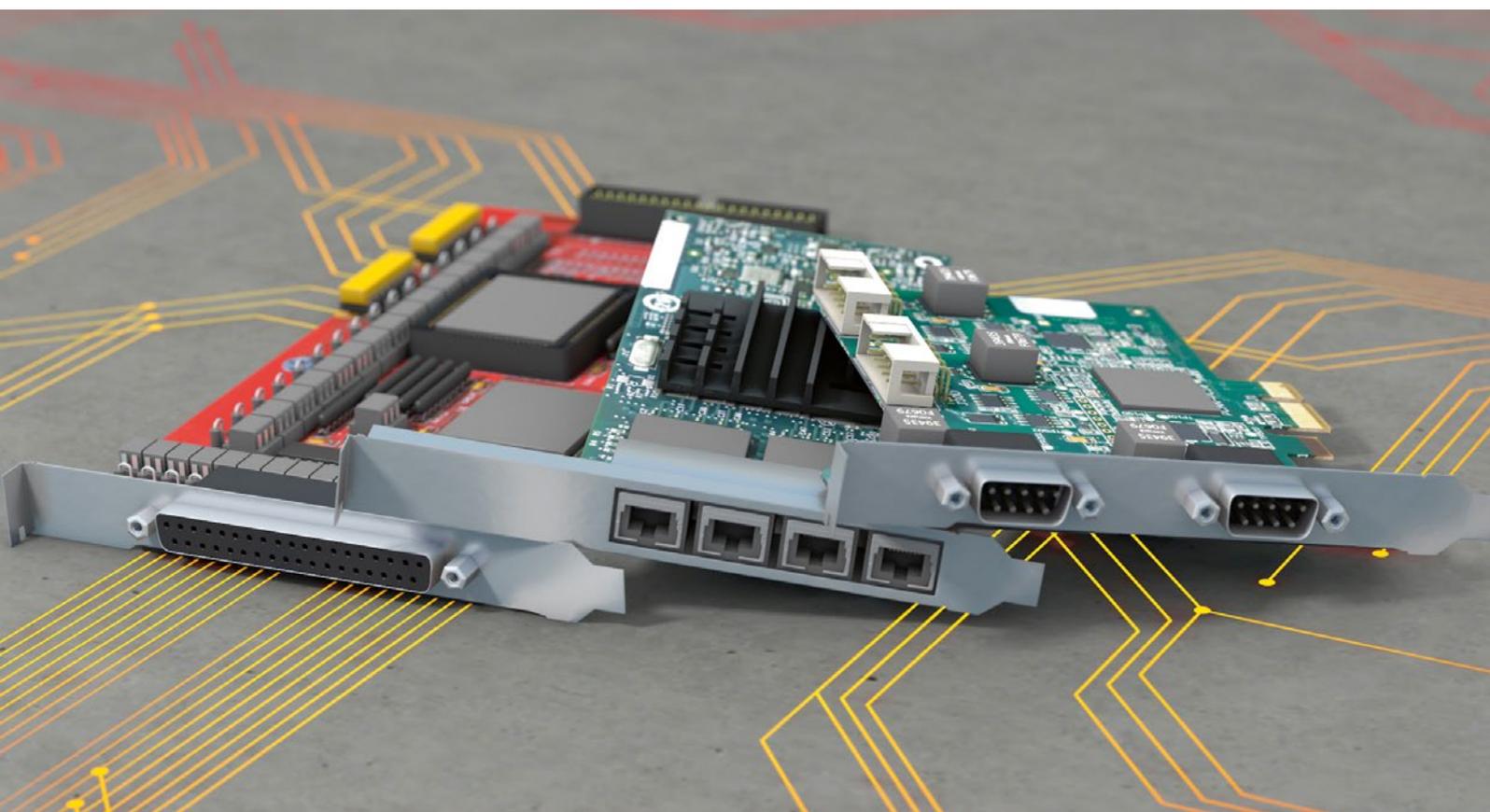
实时访问多功能扩展卡

概述

支持多功能卡并提供与供应商无关的API，支持数字IO按位或按字操作，模拟I/O作为单个值、通道序列、有限序列系列或带有切换缓冲中断的连续模式。

特征

- 支持带有与供应商无关API的多功能卡
- 控制和数据采集的A/D、D/A及数字I/O功能
- 数字I/O支持按位或按字操作
- 模拟I/O支持单值、通道序列、有限序列或带切换缓冲中断的连续模式
- 为多家制造商的卡片提供实时驱动程序
- 支持PCI或PCIe总线的多功能卡
- 需要 **RealTime Tasking Module**





Automotive



汽车领域



近年来，汽车技术经历了飞速的发展。在这些技术中，关乎驾驶安全、驾驶舒适度及驾驶辅助的软件部分持续增长，正如计算机技术本身及其用于各部件连接的标准化总线系统一样。该领域中最有效的总线系统包括CAN、CAN FD、FlexRay、BroadR-Reach和LIN，对此，Kithara提供了一系列实时功能，以支持基于此类软件的汽车系统开发。

CAN（控制器局域网）由于其普遍适用性和灵活性，成为了汽车网络中最为成功的总线系统之一。此外，CAN FD（灵活数据速率）的扩展使得每条消息能承载更高的数据速率和更多信息。

FlexRay凭借双通道下的高速数据传输，非常适合数据密集型的驾驶功能，常与CAN一同应用于安全关键型场景中。

另一方面，基于UART的LIN总线则支持开发更为经济的单线网络，用于汽车部件中较为简单的功能，并且常被整合进CAN等更高级别的系统中。

对于众多制造商而言，以太网是汽车布线未来发展的下一步。尖端的BroadR-Reach是一种专为汽车行业设计的以太网传输层，其优点在于极高的传输速度、高可扩展性，以及在布线和编程上的成本效率。

这些总线的实时能力在产品开发与集成、工业制造以及通过直接基于PC的控制与诊断来对集成机电系统进行质量控制的测试系统工程师工作中尤为重要。

Kithara RealTime Suite凭借其模块化系统，提供了多种机制，成为了此类任务的理想解决方案。

CAN/CAN FD

CAN Module使得现场总线CAN（兼容CAN-2.0B协议）能够在实时环境中工作。该API不受供应商限制，因此只需一次初始培训即可编程任何受支持的CAN硬件类型。

例如，对于传入消息的即时响应也允许回调函数立即对错误事件作出反应。由于拥有大容量的发送和接收缓冲区，即使在快速波特率和总线负载下也不会发生数据包丢失。此外，还可以创建自己的过滤例程，并在接收时直接执行。

在需要CAN接口完全处于被动状态的分析任务中，可以使用只听模式。此外，通过使用配备SJA1000 CAN控制器的硬件，可以读取“错误码捕获”寄存器，以获取额外的错误分析信息。

Flexray

作为Kithara实时扩展的组成部分，FlexRay Module能够实现高精度程序所需的实时功能，确保精确的循环时间，成为测试软件与测试平台间不可或缺的桥梁。借助此模块，可将Windows PC转变为具备全方位功能的FlexRay节点。

这涵盖了作为主控节点或从属节点的应用，即在网络中担当主动或被动通讯点的角色。两个FlexRay通道能够独立配置，波特率设定灵活，可选范围为2.5、5或10兆比特每秒。



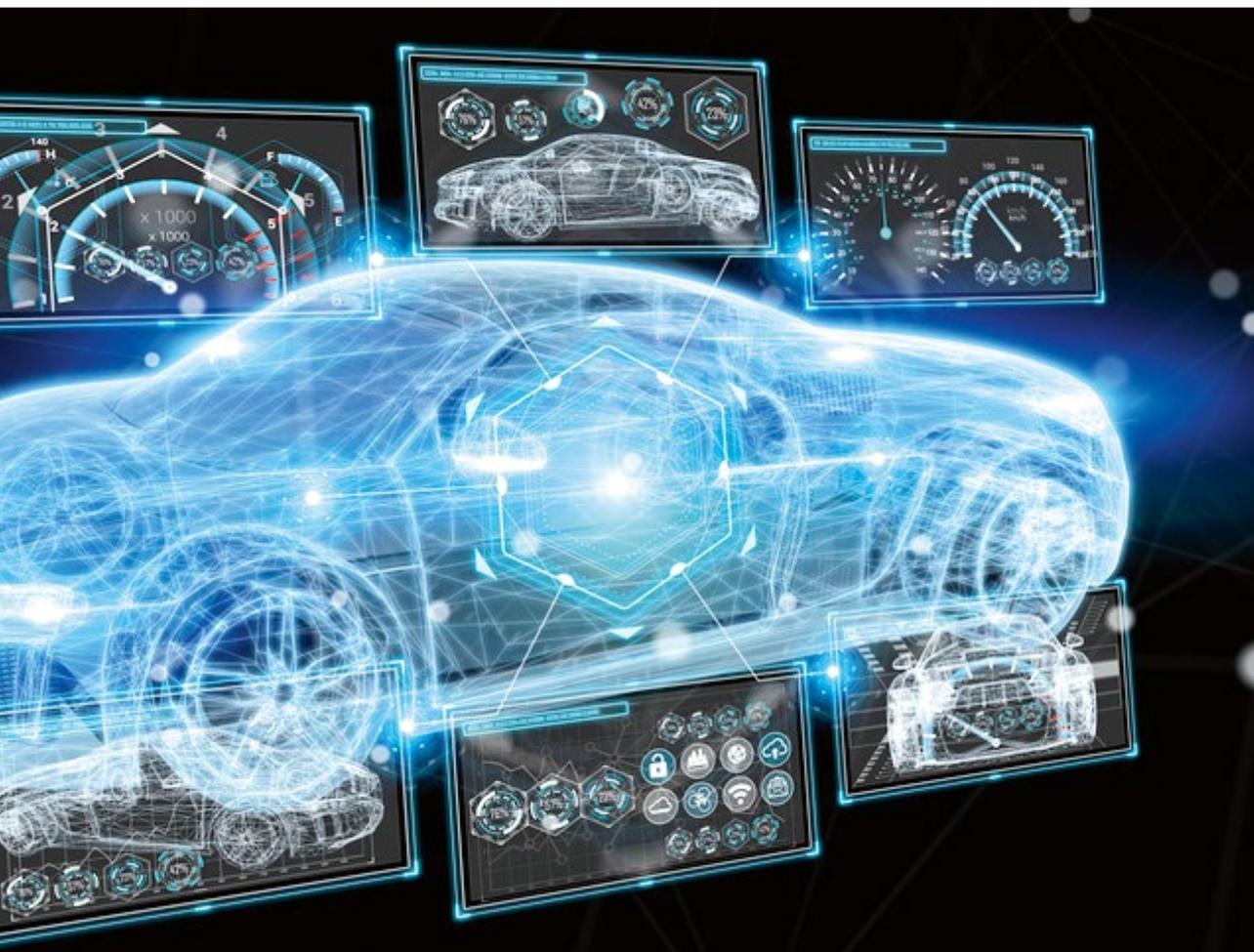
BroadR-Reach

针对BroadR-Reach标准（100BASE-T1, 1000BASE-T1）的实时功能使汽车开发者能够直接控制和测试车内物理以太网网络。Kithara实时系统提供的保证响应时间，确保了以太网传输层中的数据能够被准确收集并嵌入到复杂的测试流程中，从而架起了Kithara RealTime Suite的高性能以太网驱动程序与现代车辆网络之间的桥梁。

Kithara实时功能中的许多都在先进汽车技术的开发中得到应用。例如，在硬件在环应用中，需要精确捕获并重现仿真数据而不产生任何数据丢失，这只有通过实时技术才能实现。

LIN总线技术

借助Kithara RealTime Suite中的LIN Module，可以实现LIN通信的实时性。仅需一台配备UART接口的普通电脑，就能简便地建立主节点或从节点。相较于使用微控制器，这一方式更加便于现有LIN网络的监控日志记录，并能提升调试流程的效率。发送节点会自动复核其在总线上的传输数据，从而确保能及时发现并处理数据碰撞问题。



CAN

can^{FD}

lin
LOCAL INTERCONNECT NETWORK

FlexRay

BroadR_{xc}Reach

模块



FlexRay Module

实时FlexRay通信

概述

作为Kithara实时扩展的一部分，FlexRay Module是测试软件与物理测试系统之间的重要链接，它允许在高精度操作中应用具有精确周期的实时功能。

通过FlexRay Module，Windows PC可以用作实际的FlexRay节点。这包括被指定为主导节点或跟随节点，以及在网络中的主动或被动通信点。两个FlexRay通道可以独立配置，并且可以灵活调整它们的波特率（2.5/5/10 Mbit/s）。

特征

- 实时FlexRay通信
- 灵活配置FlexRay通道
- 将Windows PC用作FlexRay节点
- 被指定为主导节点或跟随冷启动节点
- 波特率可灵活调整（2.5/5/10 Mbit/s）
- 需要 **RealTime Tasking Module**

硬件支持

目前支持以下FlexRay卡：

- FlexCard PMC II (Star Cooperation)

CAN Module

实时CAN通信

概述

CAN Module为CAN (Controller Area Network) 串行总线系统，包括CAN 2.0B版本，提供了实时功能。它使Kithara实时系统内的开发者能够在无延迟的情况下发送和接收CAN消息。自定义过滤例程可以通过回调函数触发。可以通过回调信号或消息轮询来指示接收到的消息。发送和接收例程可以直接从实时上下文中执行。

此外，该模块也支持CAN扩展CAN FD (Flexible Data Rate)，通过将用户数据长度从8字节扩展到64字节，使得数据速率最高可达原来的8倍。而且，由于测试值的改进，传输的潜在数据安全性得到提高，从而为安全关键型应用扩展了可能的实施方式。对于分析任务，当CAN接口必须保持完全被动时，提供了“监听”模式。此外，对于配备SJA1000 CAN控制器的硬件，可以读取“错误代码捕获”寄存器，以便为调试提供更多信息。

特征

- 支持至CAN 2.0B版本的实时通信
- 可选支持CAN FD (Flexible Data Rate)
- 发送无延迟，并通过回调函数或异步消息轮询立即响应接收
- 需要 [RealTime Tasking Module](#)

硬件支持

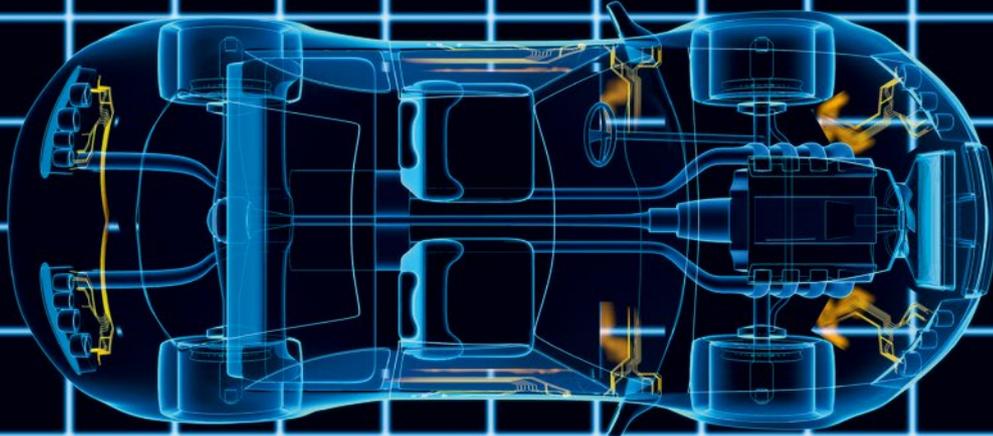
CAN模块当前支持Peak、Star Cooperation、EMS Dr. Wünsche、ESD、Ixxat和Kvaser品牌的基于PCI的1、2或4通道卡（见下列表）。如有要求，我们很乐意为您选择的卡提供支持。

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-PCle (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-PCle /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

lxxat

- PC-104/PCI, dual-channel

Kvaser Advanced CAN Solutions

- PClcan HS (single-, dual-, four-channel)
- PClcanx HS (single-, dual-, four-channel)
- PCIEcan HS (single-, dual-channel)

Extensions

CAN FD Extension

通过CAN FD (Flexible Data Rate) 实现实时通信

LIN Module

通过LIN (Local Interconnect Network) 实现实时通信

概述

使用Kithara RealTime Suite的LIN Module, 可以实现LIN上的实时功能。使用具有UART接口的普通PC可以轻松创建主节点或从节点。这使得现有LIN网络的记录变得简单。对于LIN节点的调试而言, 相比使用微控制器也更加方便。发送节点会在总线上重新检查其发送的数据, 因此可以检测到冲突。

特征

- 实时LIN数据交换
- 使用普通PC轻松实现主节点或从节点的配置
- 检测总线冲突
- 自动校验和验证 (LIN版本1.x和2.x)
- 数据速率高达20 Kbit/s
- 确保延迟时间
- 通过回调或轮询接收LIN消息
- 为了适应LIN总线的电气参数, 需要一个LIN收发器。
- 需要 [Serial Module](#) 和 [RealTime Tasking Module](#)

硬件支持

- UART (使用常见的RS232接口需要电平转换器)
- Peak PCAN-USB Pro
- 更多硬件支持可根据请求提供

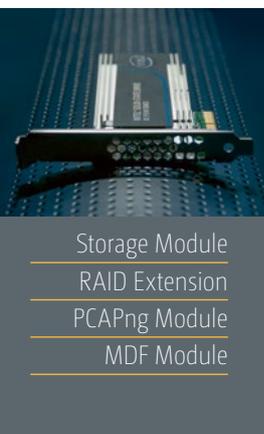


对PCAN USB FD适配器的支持

Kithara RealTime Suite亦涵盖了Peak Systems推出的PCAN USB FD适配器的支持，这一支持使得用户能够通过计算机的USB接口连接多达双端口的CAN FD与LIN网络。

诸如CAN、CAN FD及FlexRay等接口的实时性能，在汽车测试系统的开发与部署中扮演着至关重要的角色。凭借这些技术，无论是单一部件还是整个硬件系统，都能在近乎真实的环境下，借由极快且稳定的响应时间进行细致的测试与分析。

实时数据存储



在众多工业领域中，大数据的重要性日益凸显。许多应用已经从不断增长的数据量中受益，有的甚至因之而打大获成功。特别是在测量与测试技术等领域，大量的数据被有效利用来生成准确的结果和评估。

除了数据采集，来自工业相机的数据，存储环节也是高效利用它们的关键环节。为此，数据集不仅需要高吞吐率下无损地写入存储介质，往往还需要以相同的速度被检索。基于此，Kithara在Storage Module这一领域的发展重点特别放在实时数据存储上，以确保它们能在高要求项目中得到有效实施。

Storage Module支持通过NVMe接口的SSD进行实时数据存储。这样一来，数据可以在持续运行中以每秒数GB的速度进行写入和读取。通过UDF（通用磁盘格式），还提供了一个高效的数据系统，使得包写入得以最优利用。

通过RAID（独立磁盘冗余阵列），在已经通过互联多块SSD之后达到的很高的吞吐率还能进一步提升。如此一来，数据速率可通过所使用的SSD数量倍增（RAID 0模式下），速度可达数十GB/s的范围。借助Kithara RealTime Suite，读写访问依然在实时环境中处理。

为了高效存储数据集，提供了与MDF（测量数据格式）的实时连接。得益于Kithara实时环境对MDF 4.1的支持，大量复杂的层次化结构测量数据能够即刻存储并访问。利用实时功能，无损捕获的原始测量数据作为MDF文件供后续使用。MDF已成为测量任务的实际标准，尤其在汽车行业最为突出，其特点在于对采集数据的快速且节省空间的存储。该格式还包括高效排序和压缩数据集的强大功能。

此外，还支持Wireshark兼容的文件格式PCAPng。结合实时能力，可在网络中直接捕获并评估数据包。这意味着免费且广受欢迎的Wireshark可作为存储数据后续分析的便利工具。

模块

Storage Module	通过固态硬盘实现实时数据存储
概述	存储模块引入了通过最新容量的SSD进行实时数据存储的功能。通过NVMe接口，可实现大数据量以每秒数GB的速度进行高速写入与读取。同时，采用UDF（通用磁盘格式）作为文件系统，支持高效的分包写入操作。
特征	<ul style="list-style-type: none">■ 通过NVMe SSD实现实时数据存储■ 高速读写，速度可达每秒数GB■ 支持UDF文件系统■ 需要 RealTime Tasking Module
硬件支持	结合Storage Module，所有符合NVMe规范1.0的存储设备均可按需使用。已测试设备包括： <ul style="list-style-type: none">■ 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



扩展

Storage RAID Extension

通过RAID 0组的多个NVMe SSD实现实时数据存储，存储容量和吞吐率成倍增加。

RAID Extension

连接多个SSD

概述

RAID Extension增加Storage Extension了将SSD组合并提供实时功能的能力，从而使得通过互联的SSD数量，吞吐率和存储容量均可成倍提升。

特征

- 多个互联SSD的实时实现
- 通过连接的SSD数量乘以吞吐率和存储容量
- 吞吐率高达每秒数GB，直至两位数GB范围
- 支持RAID 0
- 条带深度自由配置
- 通过透传机制在Windows中显示RAID磁盘
- 可搭配或不搭配外部RAID控制器使用（例如HighPoint SSD7101A-1）
- 需要 [Storage Module](#)

PCAPng Module

以PCAPng格式进行实时数据存储

概述

PCAPng Module（PCAP下一代）允许以PCAPng格式实现实时存储复杂且层次化的测量数据。PCAPng是一种Wireshark支持的数据格式，用于捕获网络中的数据包。借此，Wireshark可用于存储测量数据的后期分析。

特征

- 以PCAPng格式进行实时数据存储
- 使用Wireshark读取存储的数据集，包括测量数据甚至图像数据
- 基于 [Storage Module](#)

MDF Module

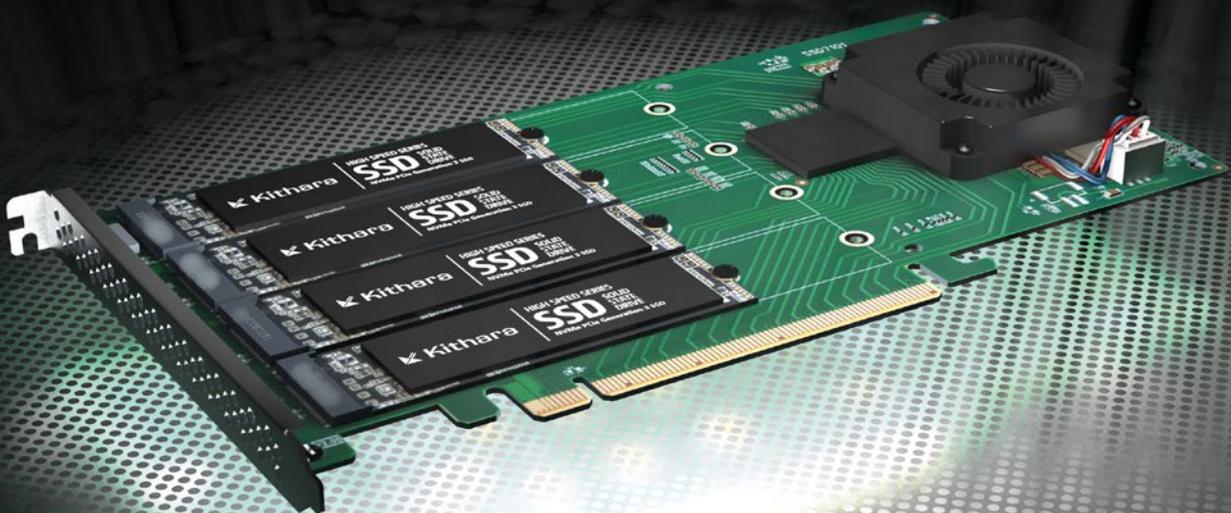
以MDF格式进行实时数据存储

概述

MDF Module允许采用测量数据格式（MDF）实现实时存储复杂且层次化的测量数据。MDF 4.1是由ASAM提供的一种二进制文件格式，用于评估或长期存储记录和计算的数据集。文件大小仅受限于所使用的硬件（即理论上可达TB级别）。

特征

- 以MDF格式进行实时数据存储
- 支持MDF 4.1，向下兼容早期版本
- 几乎可写入任意大小的文件（至 2^{64} 字节）
- 存储来自CAN、LIN、FlexRay和车载以太网总线的原始消息
- 基于 [Storage Module](#)

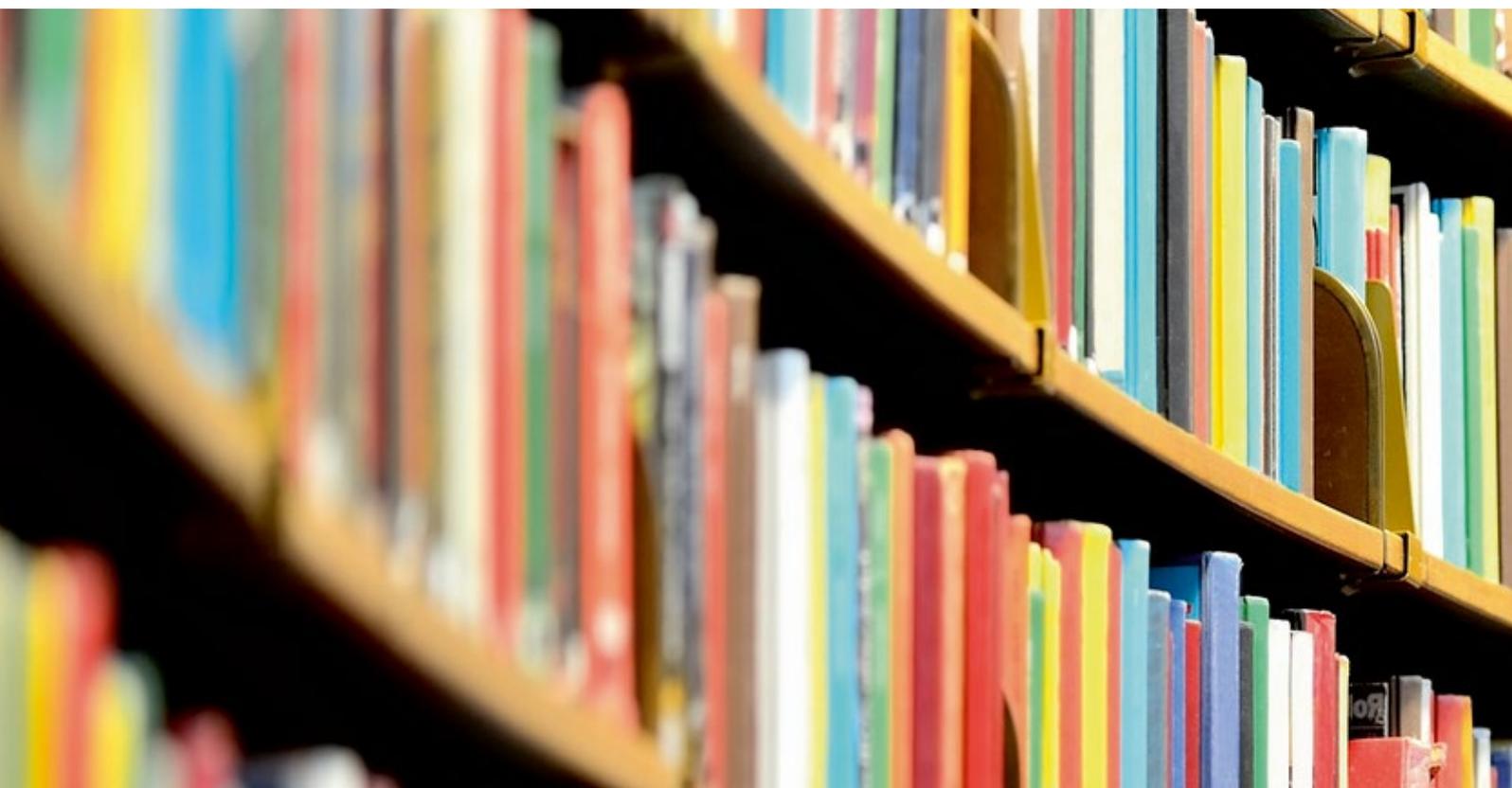


实时库



Kithara 通常建议使用支持实时的运行时库。

Library Module	
	数学/三角函数和字符串记忆函数
概述	为了在实时上下文中执行应用程序代码，通常需要公共运行时库的功能。但是，它们可能不符合实时执行的条件。因此，运行时库提供了 C 标准库的实时调整函数，包括实时的数学/三角函数和字符串/内存函数。
特征	<ul style="list-style-type: none">■ C 标准库函数的特定实时实现■ 超过 20 个数学/三角函数（例如 sin、cos、tan、arcus 和双曲线函数、exp、log、pow、sqrt、floor、ceil）■ 超过 20 个字符串/内存函数（例如 memcpy、memset、strlen、strcmp、strcpy、strcat、strtok）■ 支持应用程序和内核级别■ 可选：图像处理 - 使用 Halcon 或 OpenCV 实时处理■ 可选：机器学习 - 使用 Dlib 实时学习



扩展

Dlib Extension

Dlib是一个开放程序库，包含用于机器学习任务实际应用的算法和工具。该库包含了处理网络、线程、图形用户界面、数据结构、线性代数、基于人工神经网络的机器学习及深度学习等各种软件组件。通过Dlib Extension，这些功能可以在Kithara RealTime Suite的实时环境中使用。

- 实时中的Dlib
- 机器学习的算法和工具
- 网络、线程、图形用户界面、数据结构、线性代数、基于人工神经网络的机器学习及深度学习功能
- 免费程序库

OpenCV Extension

使用OpenCV进行实时图像处理

概述

OpenCV是一个免费程序库，拥有图像处理和机器视觉的算法。它包括人脸识别、3D功能、头发分割、多种快速滤波器以及相机校准等功能。OpenCV Extension 使得这个程序库能被整合到Kithara实时系统中。这使得处理后的图像能立即触发控制反应，并且能够将它们集成到带有EtherCAT或CANopen的自动化系统中。



```
71
72 left := -1
73 right := 1
74 up := 2
75
76 for i := 0 to 256 by 1
77   read_image (Image, 'object_' + i$.04')
78
79   for j := 0 to KnownObj - 1 by 1
80     find_calib_descriptor_model (Image, Objects[j], [], [], [], [], 0.3, 2, CameraConf, 'num_points', Pose, Score)
81
82     if (|Score| > 0)
83       get_descriptor_model_points (Objects[j], 'search', 0, Row, Col)
84       gen_cross_contour_xld (Cross1, Row, Col, 6, 0.785398)
85
86       X := ROIY[j * KnownObj:(j + 1) * KnownObj - 1]
87       Y := ROIY[j * KnownObj:(j + 1) * KnownObj - 1]
88       pose_to_hom_mat3d (Pose, Mat3D)
89
90       affine_trans_point_3d (Mat3D, X, Y, [0,0,0,0], TansfX, TansfY, TansfZ)
91       project_3d_point (TansfX, TansfY, TansfZ, CameraConf, RowTransf, ColumnTransf)
92
93       gen_contour_polygon_xld (Contour, RowTransf, ColumnTransf)
94       close_contours_xld (Contour, Contour)
95
96       if ((Pose[5] > 45 and Pose[5] < 135))
97         flipObject(left)
98       elseif (Pose[5] > 225 and Pose[5] < 315)
99         flipObject(right)
100      elseif (Pose[5] > 135 and Pose[5] < 225)
101        flipObject(up)
102      else
103        * No action needed
104      endif
105    endfor
106  endfor
```

特征

- 使用OpenCV进行实时图像处理
- 包含人脸识别、3D功能、头发分割、多种快速滤波器及相机校准功能的算法
- 自动并行化处理
- 对处理后的图像数据立即触发控制反应，例如通过EtherCAT或CANopen
- 支持OpenCV版本：OpenCV 3.0、3.4和4.1
- 需要> [Library Module](#)

HALCON Extension

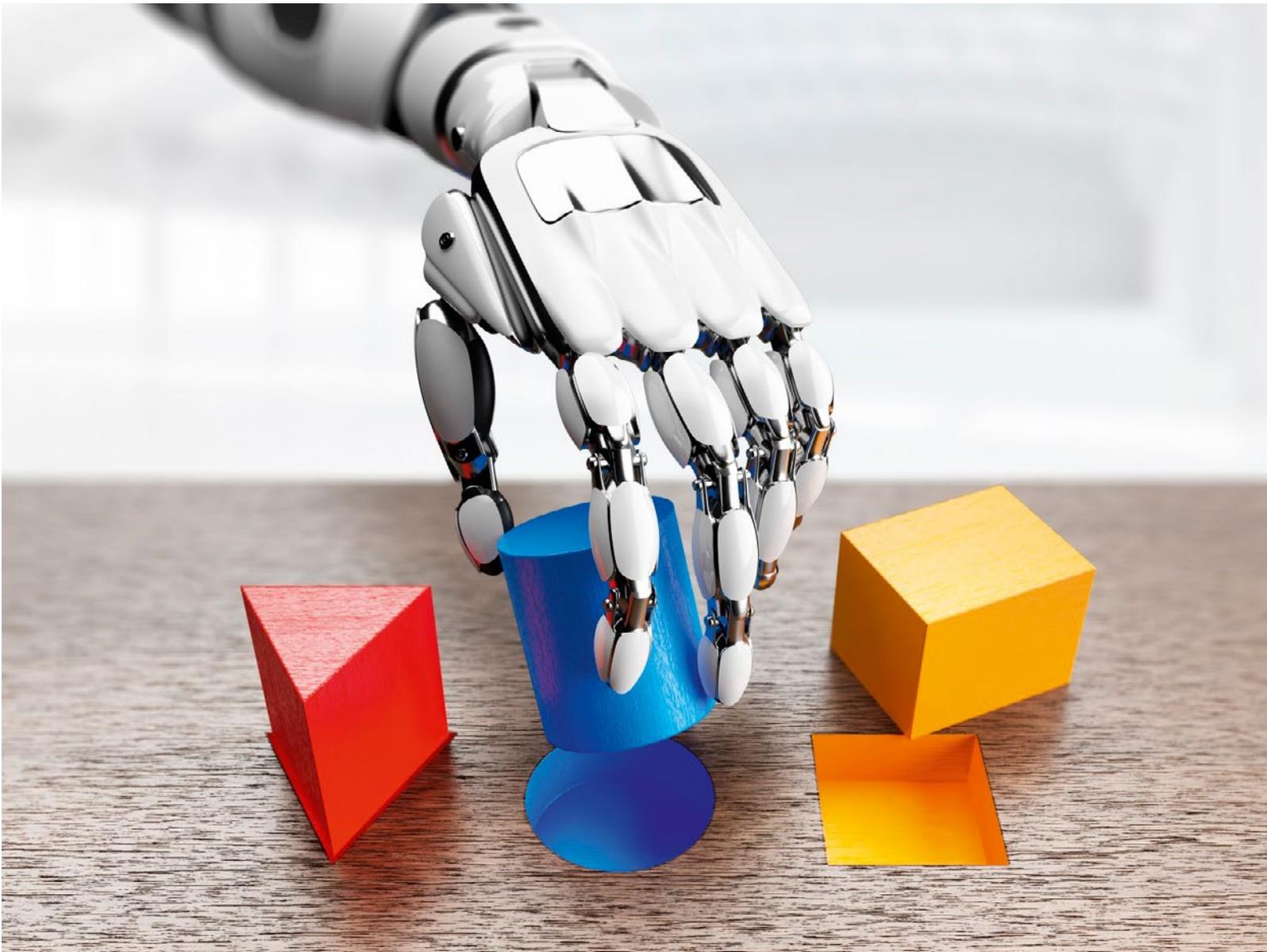
使用HALCON进行实时图像处理

概述

HALCON是由慕尼黑公司MVTec提供的程序库，包含图像处理和机器视觉的算法。它拥有自己的集成开发环境HDevelop及超过2000个操作符。HALCON Extension 使得该程序库能被整合到Kithara实时系统中。这使得处理后的图像能立即触发控制反应，并且能够将它们集成到带有EtherCAT或CANopen的自动化系统中。

特征

- 使用Halcon进行实时图像处理
- 超过2000个操作符
- 集成开发环境HDevelop
- 自动并行处理
- 直接从实时环境中执行HDevEngine
- 对处理后的图像数据立即触发控制反应，例如通过EtherCAT或CANopen
- 支持Halcon版本10至13，以及18.11和20.11
- Halcon许可证需通过MVTec获取
- 需要 [Library Module](#)



SigProc Module

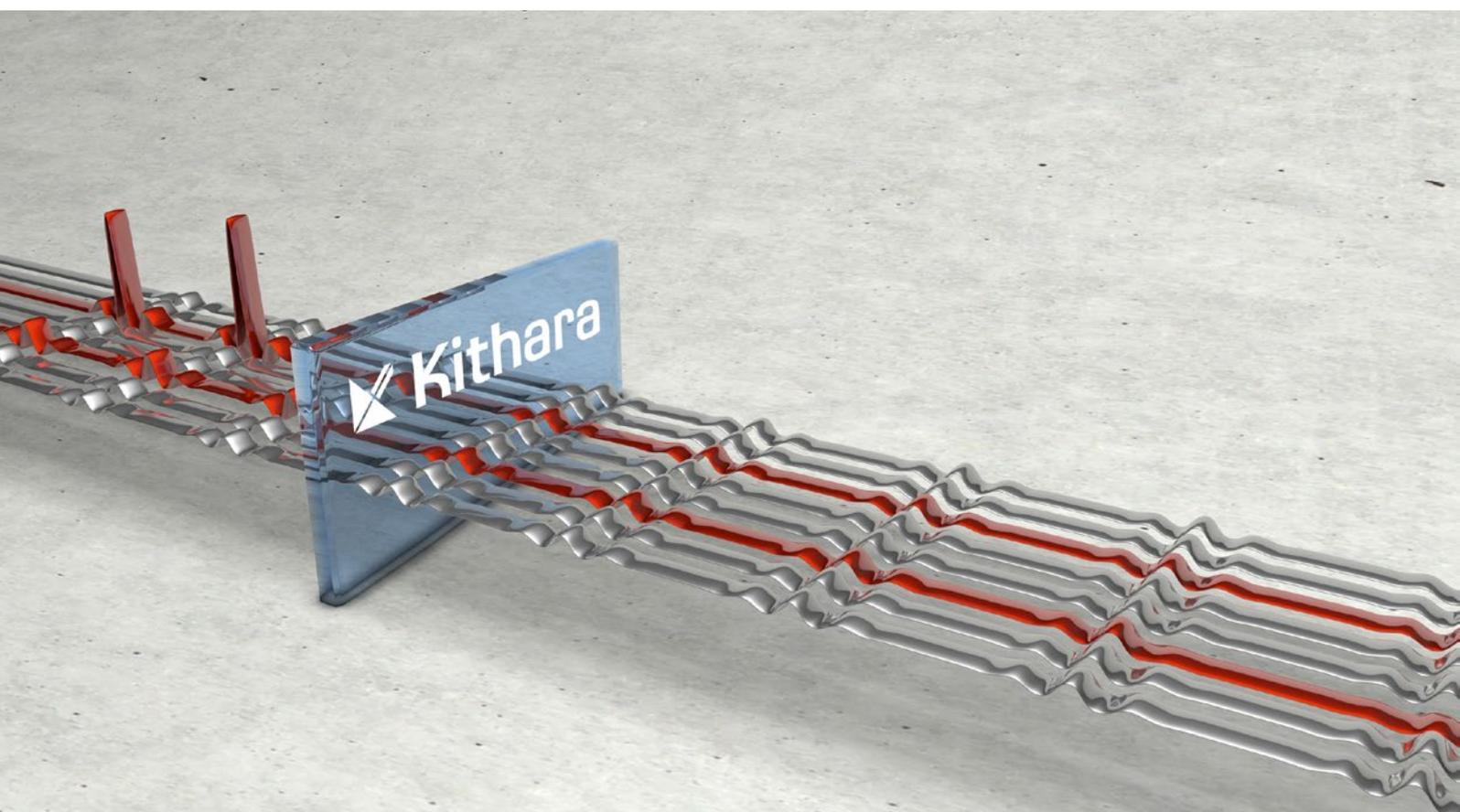
实时数字信号处理

概述

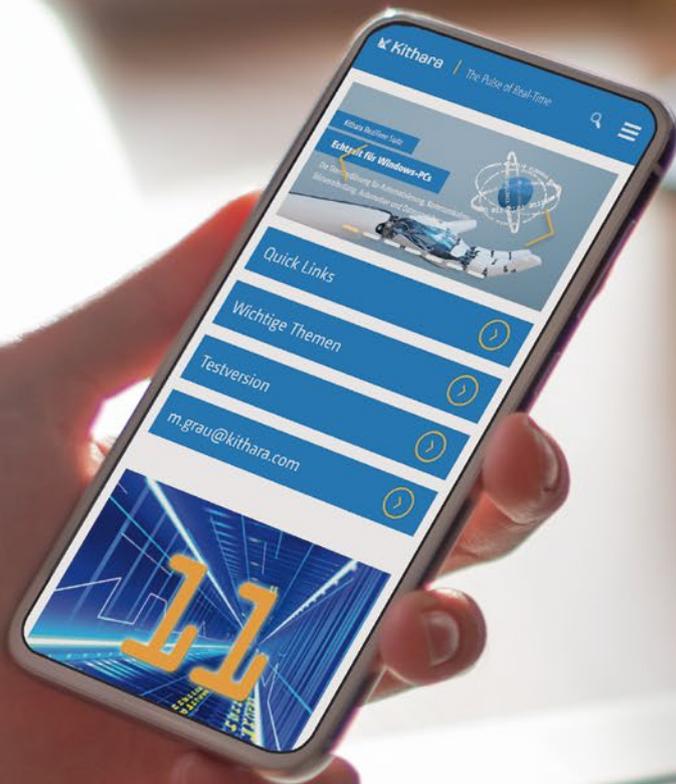
对于广泛的工程应用而言，实时信号处理和过程变量控制通常是必需的。因此，SigProc Module提供了一个PID控制器以及一系列数字FIR/IRR滤波器选择，包括不同的窗函数（如Hamming, Hanning和Rectangular），以及不同的传递函数多项式（如Chebyshev I, Chebyshev II和Butterworth）。

特征

- 实时数字信号处理
- 超过20种不同滤波器（低通、高通、带通、带阻）
- 3种不同的窗函数（Hamming, Hanning, rectangular）
- 3种不同的传递函数多项式（Butterworth, Chebyshev I和II）
- 用户自定义数字FIR/IRR滤波器
- 基于PID算法的实时反馈控制



Kithara在线



关于Kithara RealTime Suite实时套件的所有信息、技术支持及最新动态，您都可以在互联网上访问 [kithara.com](https://www.kithara.com) 获得。

开发工具

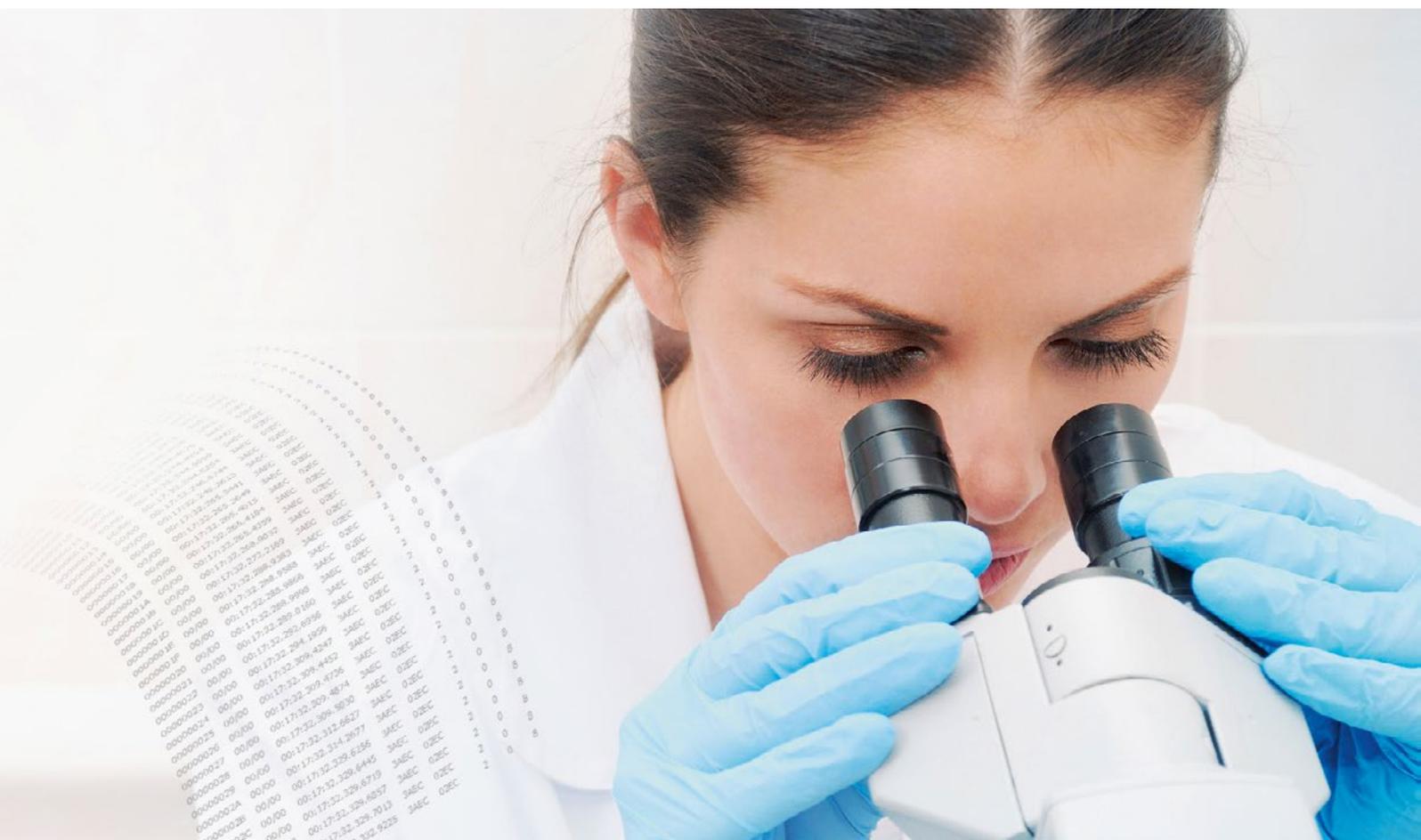
Kithara Performance Analyzer

用于监视计算机性能的系统资源可视化工具

开发实时应用程序始终需要对开发平台和目标平台提出特定的性能要求。Kithara Performance Analyzer结合Kithara RealTime Suite的功能，可以对实时容量和系统完整性进行可视化和高效监控。Windows任务管理器在系统效率方面仅提供有限的见解，同时也无法访问有关实时功能的相关数据。除了访问内存使用率和CPU负载等系统资源外，Kithara Performance Analyzer还能够收集有关Kithara RealTime Suite。支持的所有连接接口的数据吞吐量的详细信息。这样，可以同时收集和显示各种不同的数据值，从而更好地识别内部系统交互和问题源。

概述

Kithara Performance Analyzer是一个与开发相关的可视化助手，用于同时显示和监控不同的相关系统资源。因此，它是确定系统实时性能的重要工具。随着Kithara Performance Analyzer的执行，内部实时系统被指示共享有关系统资源和接口吞吐量的所有必要信息，然后以图形方式实时显示。用户可以完全控制显示哪些值以及以何种格式监视这些值。与Kithara Performance Analyzer的数据通信以及图形显示对系统性能没有影响。即使在显示大量值的情况下，CPU负载也只有1%到2%。



Kithara Performance Analyzer 特征

Kithara Performance Analyzer能够收集并展示以下数据：

- 所有内存类别的使用情况，比如内部实时内存或应用程序内存
- CPU负载，包括任务和中断（可单独显示特定CPU核心）
- Kithara支持的所有接口的数据吞吐量（发送与接收），涵盖(Gigabit)以太网、USB3/xHCI
- 基于接口的协议：EtherCAT、CANOpen、LIN、GigE Vision、USB3 Vision
- 抖动（Jitter）

可视化功能

- 多个数值的时间并行显示
- 灵活的时间分辨率（15秒至2分钟）
- 线性或对数显示模式
- 图形用户界面完全可自定义：显示颜色、图表位置、轴比例尺、自定义值标识等



Kithara Kernel Tracer

多源实时消息跟踪器与记录器

通过使用Kithara Kernel Tracer，可以详细地跟踪和记录实时应用程序的内部进程。在程序开发过程中，有效的工具是必要的，例如用于调试。以往，大多数情况下会使用调试器来实现这一目的。然而，在多线程编程中，调试器的使用常常受到限制，因为它会影响进程。当使用多个进程和线程时，特别是在涉及具有多CPU（多处理器和多核系统、超线程）的PC上并行运行时，程序进程的结果取决于能否同时执行特定的程序部分。调试器无法保证这一点。常见的问题包括：

- 使用调试器会负面影响程序流程。
- 即使程序表现不如预期，调试器也可能使其看起来一切正常。
- 调试器不具备在实时上下文中运行的能力。

解决方案是一个跟踪工具，能够记录并行例程的实际进程并分析日志。

简述

Kithara Kernel Tracer是一个多源实时消息记录器。它支持调试、质量保证，并且通常用于检测那些否则难以确定的交互。Kithara Kernel Tracer按出现时间顺序收集来自不同源的每个可能出现的跟踪消息。尽管来自不同源的消息出现在不同的标签页上，但可以将它们合并。这样，开发者可以全面了解系统的实时行为，有助于优化程序性能和解决并发问题。

Kithara Kernel Tracer每秒能处理数十万条消息。为了从庞大的数据量中寻找重要信息，它提供了多种消息过滤和搜索机制。它能深度揭示计算机实际运作的奥秘，尤其是在内核层面。除了通过Kithara功能进行直观展示外，你还可以创建自定义消息，并附带更多有益的附加信息进行展示。

Features of Kithara Kernel Tracer in Detail

- 多源支持：实时环境下，无论是应用程序还是内核级别，均可支持无限数量的线程和程序流程。
- 多核精准：根据生成时间点，准确地从多个CPU（包括SMP、超线程、多核等技术）分配消息。
- 高速处理：若系统性能允许，每秒可处理数十万条消息。
- 功能调用全览：Kithara RealTime Suite中的每次函数调用均会展示其参数。
- 用户自定义消息：利用Kithara RealTime Suite的工具，用户可生成个性化消息，类似于C/C++中的printf功能。
- 详尽信息展示：消息展示包含系统时间（精确至0.1微秒）在内的诸多细节。
- 消息管理灵活：支持过滤、合并、保存及重载消息。
- 高级过滤功能：可实现永久删除、临时隐藏、加粗、斜体及彩色显示消息。
- 色彩视觉强化：通过着色突出错误与警告信息。
- 灵活筛选来源：支持从单一源头或全局范围筛选消息。
- 丰富上下文信息：提供逻辑CPU、线程及/或进程ID、实时任务ID、连续消息编号与类型、线程及任务优先级，以及选定消息的数据块内容。
- 全方位搜索：可通过文本、日期及所有其他字段查找消息。
- 消息屏蔽机制：避免生成不必要的非必需消息。
- 系统崩溃保护：消息可在蓝屏死机（BSOD）后恢复，崩溃转储将自动扫描。
- 差异化色彩配置：为不同进程、线程和任务赋予不同色彩，以便清晰区分。
- 后期编辑消息（文本、前景与背景色）以辅助文档记录。
- 编程过滤器可启用或禁用。
- 数据列显示切换。
- 监控特定的USB、UART、CAN及网络设备。
- 内存查看器：用于物理内存分析。
- PCI查看器：便于PCI设备分析。
- 集成SDK：支持开发独特追踪源。

自定义消息

您可以将自己的消息转发给追踪器，例如用来展示变量内容或标记特殊程序状态的发生。

为此，Kithara RealTime Suite的工具提供了KS_logMessage和KS_vprintK函数。KS_logMessage适用于所有受支持的语言环境；而KS_vprintK则采用参数列表，因此可以通过C/C++进行控制。

鉴于硬件变化的多样性，软件与特定组件的完全兼容性无法得到绝对保证。对于有关个别硬件配置兼容性的问题，请联系我们。如有需要，我们很乐意向您提供Kithara支持的硬件列表。所有提供的信息均针对我们软件的最新版本。

注：鉴于硬件变化的多样性，软件与特定组件的完全兼容性无法得到绝对保证。对于有关个别硬件配置兼容性的问题，请联系我们。如有需要，我们很乐意向您提供Kithara支持的硬件列表。所有提供的信息均针对我们软件的最新版本。



参考资料

以下公司和机构均为我们的客户：



我们的一些合作伙伴：



如需了解更多关于功能、支持硬件、操作系统或编程语言的信息，请访问我们的网站 kithara.com。我们推荐下载免费试用版。针对您的项目有任何疑问，欢迎随时联系我们！

QR Code | Kithara on the Internet:

