



System Architecture

DEEP DIVE



WHITE PAPER

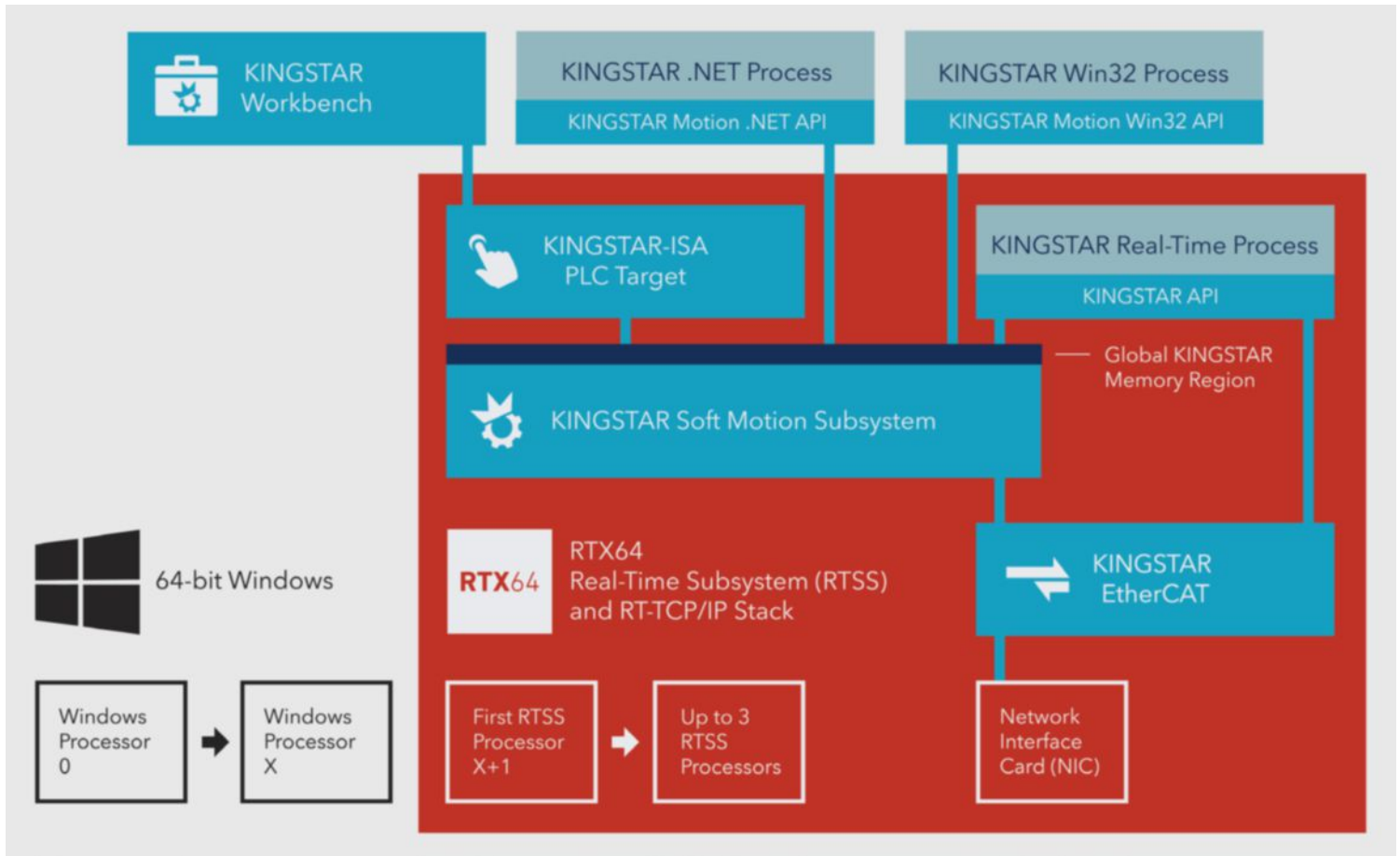
Introduction

At KINGSTAR, we've partnered with over 600 world-class machine automation vendors to identify automation architectures and procedures that enable businesses to stay ahead of the technology curve while eliminating enormous financial and market risk. In every case, these organizations started by building a componentized platform that allowed them to quickly adapt to changing market requirements and advancements. We created KINGSTAR to do the same thing: deliver a pre-integrated, pre-tested motion control framework with exceptional performance, proven scalability, and lower costs.

KINGSTAR was born from the best practices we observed in our most successful customers. As a component-based platform purpose-built for EtherCAT-enabled machine automation, KINGSTAR can adapt to each innovation or demand from the market while protecting your existing investment. It is as easy as adding a new component that plugs into the platform to address the new feature. Whether you need to replace your hardware PLC with an EtherCAT-enabled software PLC or add a new servo drive, machine vision, motion control, real-time operating system (RTOS) for fast processing, a platform approach makes it fast, simple, and cost-effective to achieve best practices.

Machine builders must embrace the speed of digital innovation and move from proprietary technology into open and pre-integrated platforms. To remain competitive, it is critical to be able to integrate with what you currently use today and may use tomorrow without sacrificing performance or budget.

Hardware Architecture



KINGSTAR is a 64-bit Windows-based platform for real-time motion control application development. It is a subsystem that runs on RTX64, a real-time operating system from IntervalZero. This section describes the components of the KINGSTAR subsystem and how they interact with each other.

Like all motion control systems, KINGSTAR requires a computer and hardware like drives and I/O. Any computer that can run 64-bit Windows 7+ operating systems will run KINGSTAR. A standard computer, however, is typically designed with features to save power and enable mobility, like sleep mode and turbo boost. These features can be detrimental to real-time control because the CPU can be stalled or slowed down, affecting the determinism of instructions. As a result, we recommend using an industrial computer. Industrial computers have custom BIOS and driver settings that allow you to disable features that save power and subsequently cause delays in motion control.



To use KINGSTAR,
you must fulfill the following additional hardware requirements:

Processor

The system must have at least two cores but can support up to 64 cores. At least one core must be dedicated to the Windows operating system and at least one core for RTX64 and KINGSTAR. If your system has more than two cores and hyper-threading is enabled, you should assign an even number of processors to Windows so a physical core is not shared between logical processors running Windows and RTX64.

Operating System

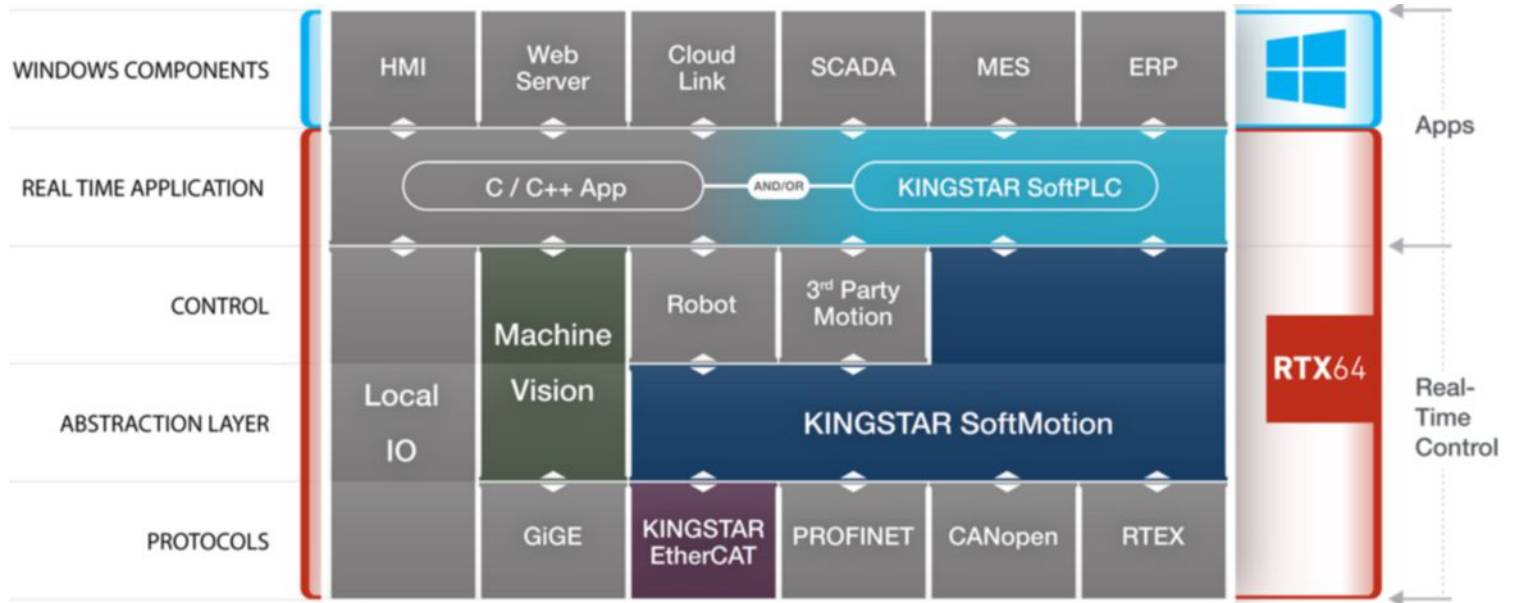
64-bit Windows, which can be Windows 7 with SP1, 8.1 or 10 with anniversary update 1607.

Network

KINGSTAR requires a supported NIC that is suitable for EtherCAT (see the Supported NIC document for a list of supported controllers.) EtherCAT requires packets to be sent at precise times so all delays must be avoided. This means the network port should have its own controller and not use the CPU as a controller. We also recommend that the card be connected to an independent PCI bus with its power saving features disabled.



Software Architecture



KINGSTAR only works with RTX64 on a 64-bit Windows operating system. Most users use an embedded version of the Windows OS. Using an embedded version for real-time control allows you to save resources by removing unnecessary components, disable any unnecessary updates, and use the **power on** feature that lets you switch

the computer on and off using the power cable **power on** convenient for people who are used to switching hardware controllers on and off with a button and want the same behavior from a PC. It is possible, however, to use a standard Windows OS provided it has a proper network card.

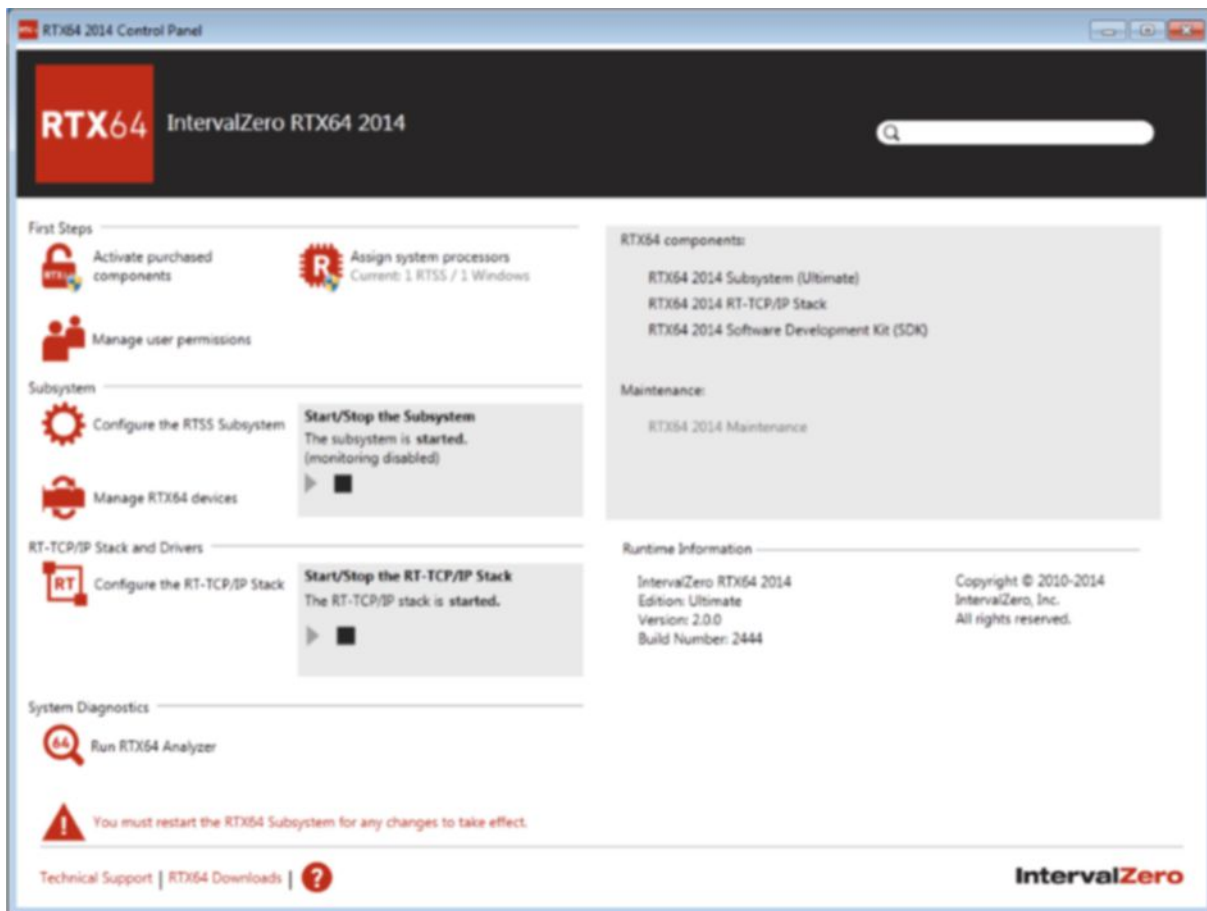


IntervalZero RTX64

RTX64 extends but does not modify the Windows hardware abstraction layer (HAL) by adding a real-time subsystem (RTSS) to Windows that delivers deterministic, hard real-time performance. With the exception of its real-time handler, RTSS is conceptually similar to other Windows subsystems in that it supports its own execution environment and API. RTX64 functions like a separate computer or hardware card, but runs on one of the cores of your normal CPU so you don't require separate hardware. Windows applications can communicate with RTX64 through shared memory, interrupts, and IPC objects like mutexes and events.

RTX64 is not a full operating system. It includes its own hardware abstraction layer, scheduler, name space, and TCIP stack, but relies on Windows for memory allocation and file system.

RTX64 maintains interrupt isolation between RTSS and Windows while providing inter-processors interrupt (IPI) communication between the two. The RTSS schedules its tasks to execute on separate processors, without any interference from Windows or Windows processes, but still allows for communication with Windows processes (RTAPI) and drivers (RTKAPI) through standard inter-process communication (IPC) and shared memory. Further information about RTX64 can be found at www.intervalzero.com/.



KINGSTAR Subsystem

KINGSTAR is comprised of several components. We'll look at them one by one.



EtherCAT

Integrate Customized Motion Control on an Auto-Configured EtherCAT-Enabled Windows PC Framework



KINGSTAR EtherCAT

KINGSTAR EtherCAT is the base component of the KINGSTAR subsystem. It is a communication and configuration library that encapsulates the EtherCAT features of KINGSTAR.

KINGSTAR EtherCAT links directly to and controls the network interface card (NIC) that will be used for all EtherCAT network activity. It is responsible for sending and receiving packets on the NIC. When KINGSTAR EtherCAT starts, it scans its EtherCAT network, finds the hardware that is available on the network and configures devices automatically. Once KINGSTAR EtherCAT is started, the hardware is ready for use.

The EtherCAT real-time data is stored in cyclic packets. You can use the KINGSTAR EtherCAT provided API to access the variables within the packets that contain real-time data. KINGSTAR EtherCAT hides the complexity of EtherCAT and presents the CANopen interface of CANopen over EtherCAT (CoE) drives. With KINGSTAR EtherCAT you don't need to know where the variables are, the differences among vendors' hardware or how the synchronization is handled.





Replace Your Motion Control Hardware with Precision-Performance Software at Half the Cost

KINGSTAR Soft Motion

The KINGSTAR Soft Motion subsystem acts like a motion control card. The difference is that it is software, which can be used with a variety of hardware and allows for the use of a common programming language to develop your motion applications. The interface provided by KINGSTAR Soft Motion to control the axes follows the PLCopen Motion Control standard.

The KINGSTAR Soft Motion architecture allows multiple applications to run using the same EtherCAT link. Some of them can be Windows, some of them can be real-time. This means you can control multiple hardware stations in one machine. You can separate your machine in multiple stations and have a different application controlling each station. All of them run on top of KINGSTAR Soft

Motion and the same EtherCAT bus on the same computer. This way, you can develop applications in different environments using different languages for each part of your machine depending on your need. Different engineers can work on different programs without interference. KINGSTAR offers a few interfaces for motion application development: C++, .NET, Win32 (Windows API) and PLC. Each of them can access the motion library. All of these applications can synchronize using the global KINGSTAR memory region.





Replace Your PLC with an EtherCAT-enabled Soft PLC for Real-Time Motion Control and Machine Vision

KINGSTAR Soft PLC

The KINGSTAR Soft PLC is a software PLC. It has the same features as a hardware PLC (following the IEC 61131 standard) but it is in software and controls the EtherCAT hardware instead of local modules. Using the PLC has multiple advantages. It is simple to use. You can program the PLC program using the graphic interface. You don't need to write C/C++, C#, BASIC or any programming language. KINGSTAR Workbench is the application for PLC program development. It supports all the PLC programming languages defined in the IEC standard. Once you finish programming and compile and download your program to the KINGSTAR Soft PLC runtime, the logic in that program runs in real-time. You don't need to learn or understand how real-time, RTX64 or EtherCAT works. In other words, you can use a PLC language to create a real-time application controlling EtherCAT hardware, just like you do it using a hardware PLC with EtherCAT features.

KINGSTAR Workbench can be installed on a different computer from the KINGSTAR Soft PLC runtime. The KINGSTAR Workbench powered by ISaGRAF generates a PLC target named KINGSTAR-ISA PLC that can be used with the KINGSTAR Soft PLC runtime on your target system.





Global KINGSTAR Memory Region

Because KINGSTAR allows you to connect to the subsystem through multiple applications developed in different languages, you need a simple way for these applications to communicate with each other. KINGSTAR offers a global KINGSTAR memory region, which is controlled by the KINGSTAR Soft Motion subsystem. Inside this memory space you can read and write any variable you want; no need to know where the variable is. It's like a memory buffer. All the interfaces - PLC, C/C++, Win32 and Windows .NET - have APIs to access the KINGSTAR global memory region. This way, you can have multiple applications sharing variables, even though they are on different subsystems and using different languages.





Third-party applications

As a plug-and-play platform, KINGSTAR is able to work with third-party applications. Depending on the KINGSTAR interface you use, your application has different abilities to connect to the third-party application.

- ❑ **C/C++, .NET, Win32:** The application can connect to the KINGSTAR Soft Motion subsystem, a third-party application on RTX64 and an application on Windows at the same time.
- ❑ **PLC:** The application can connect to the KINGSTAR Soft Motion subsystem and an application on Windows at the same time. The reason that the PLC application can't connect to a third-party application on RTX64 is because the third-party application doesn't have the API for the PLC.

The following are examples of how KINGSTAR works with a third-party application.

Matrox MIL

Matrox Image Library (MIL) is a machine vision library made by a Canadian company called Matrox. Both MIL and KINGSTAR can run on RTX64 at the same time. If you use the C/C++, .NET or Win32 library to create your application, you can synchronize and access the data from MIL and KINGSTAR at the same time. That is, you can use machine vision information to control your servo drive.

LabVIEW

If you are making a machine that has an EtherCAT drive controlled by a KINGSTAR Soft PLC program, and has a vision and/or a robot component controlled by a LabVIEW program, you can add logic to the KINGSTAR Soft PLC program and LabVIEW because KINGSTAR has a .NET API that can be connected to LabVIEW. They can share variables through the .NET API and make the machine work on single computer.

Cloud

You can also use the .NET API to send the data remotely. You can use Windows Network to download the logic from the KINGSTAR Workbench to the KINGSTAR Soft PLC runtime. And the same time, data can be connected to the cloud. A Windows application retrieves data from the KINGSTAR Soft PLC runtime and KINGSTAR Soft Motion and puts it to a cloud, so you can set up an HMI or a remote device to use this data. The data in the cloud can be shared by ERP (Enterprise Resource Planning), MES (Manufacturing Execution System) and any other application. No matter what cloud, ERP, MES you use, as long as they have a Windows client, they can retrieve the data from the cloud.

All of the KINGSTAR APIs and variables can be accessed in Windows and can be easily synchronized with any third-party application.

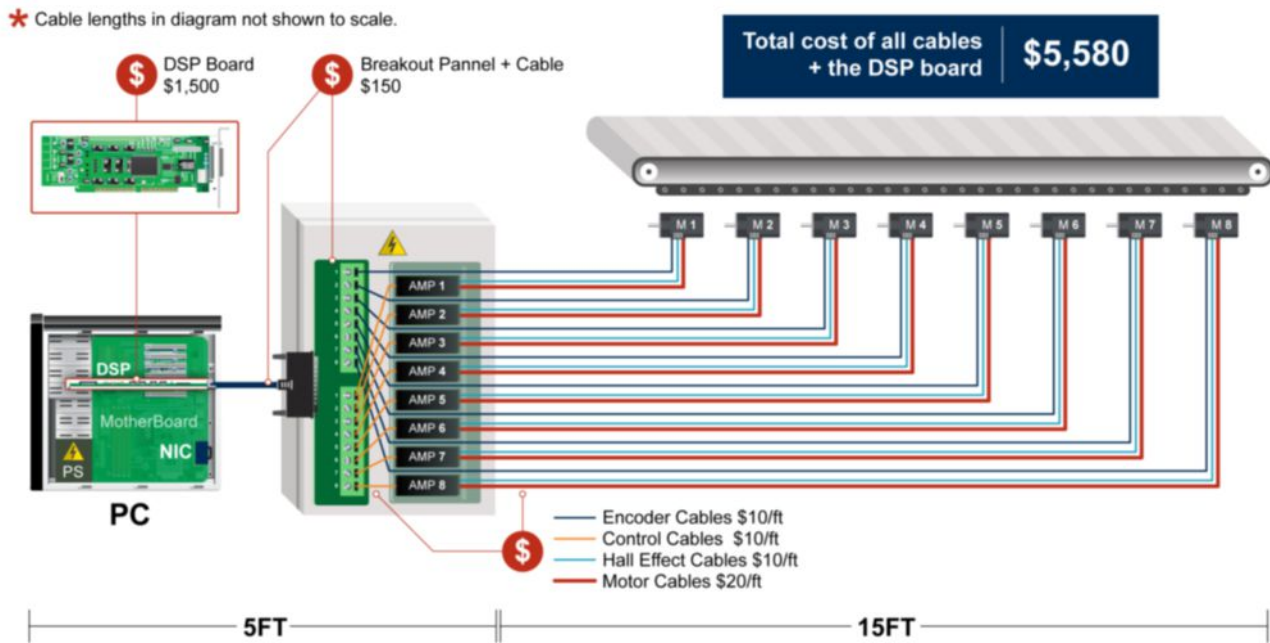


Features and Use Cases

Let's take a closer look at how an application can use KINGSTAR.

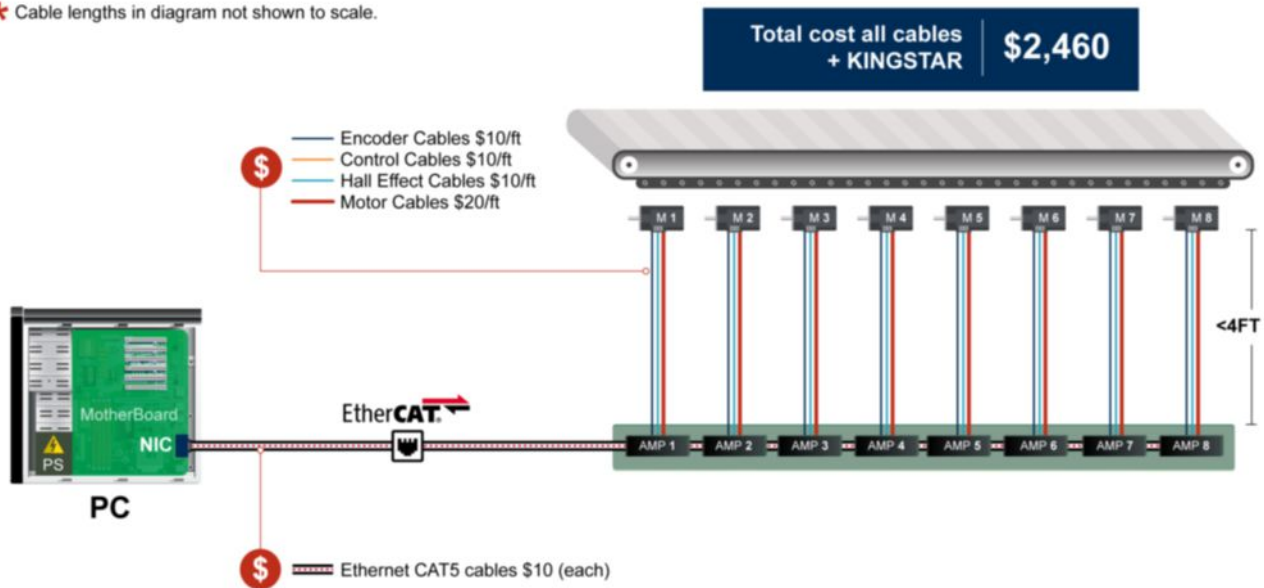
Use cases

The simplest way to use the KINGSTAR Soft Motion platform is to replace a DSP or FPGA motion board. With a motion board, you would plug a PCI card on your computer, and that PCI card would include a real-time system with motion algorithms and communication for local hardware. It performs all the real-time tasks, interpolation, and communication with the drives.



KINGSTAR does the exact same thing, except the DSP or FPGA PCI hardware becomes software. The only additional hardware needed would be a dedicated NIC for EtherCAT and at least one dedicated core of a multiple core system to run the KINGSTAR subsystem.

* Cable lengths in diagram not shown to scale.



You would use a Windows HMI non-real-time control logic like you would do with a motion board, and send commands using KINGSTAR Soft Motion through a .NET or Win32 API. The interpolation PID would be run by KINGSTAR Soft Motion on RTX64, and it would read and write the data through KINGSTAR EtherCAT to the slaves and the I/O. It works exactly like it would with a motion board or an I/O board that is controlled from Windows. .NET, C#, visual basic, LabVIEW, or any other component can be used for the HMI logic, which will send commands to KINGSTAR Soft Motion that will run in real-time on RTX64 dedicated processor(s).



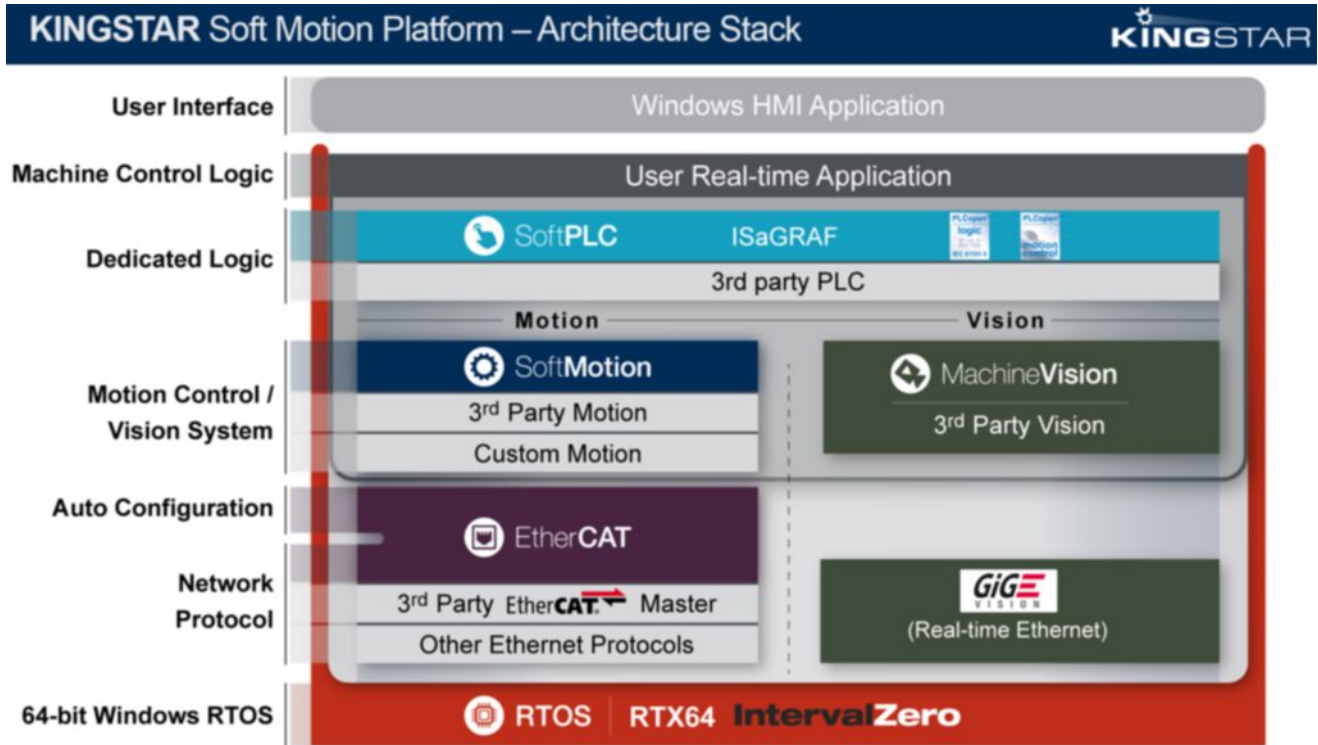
Alternatively, you may want your application logic to run in real-time. Why? Real-time is deterministic and you can have fixed holding times. You can also control the delays, which is only possible in real-time. How fast is your application going to respond to a problem once that problem is detected? That speed can only be guaranteed in real-time.

KINGSTAR Soft Motion gives you 2 options to do this:

- ☐ Write your own C/C++ real-time application on RTX64. That application will connect with KINGSTAR Soft Motion using the same APIs as it did in Windows. But because the application is real-time, you can have fixed timing to check things like data and reaction commands, error commands, cancelling, and alarms.
- ☐ If you don't want to write C++, you can also use KINGSTAR just like a PLC. It works just like a real-time hardware PLC, but all running on your computer; instead of a separate machine, it is running on Windows, using one core and one network card. To develop you will use the same tools as a hardware PLC. You will have the KINGSTAR Workbench, a software GUI to develop your applications. Once you download this logic to the runtime, it will be loaded and run in RTX64, which means in real-time.



Features



Flexible Development Environment

KINGSTAR allows you to work with the development environment of your choice, so your team doesn't need to spend resources learning a new language or interface. If you want to use the PLC interface, the development is done through the KINGSTAR Workbench graphical user interface. If you want to write your own real-time C/C++ application,

the development is done through Microsoft Visual Studio. The APIs are similar as well, since RTX64 only works with Windows. If you're using a Win32 or .NET interface, you can use any development environment that you want, provided that it can link with a Win32 or .NET library.

Auto-Configuration

The KINGSTAR Soft Motion subsystem provides settings for auto-configuration. Within KINGSTAR you can choose which variables you want to use and which control mounts you want to use, as well as do input/outputs and configure PID profiles; everything needed for your conversions and axes. We provide two different motion APIs. The first is

a motion board standard API for use with simple applications. We also support PLCopen motion control. PLCopen is the standard for motion control interface and has many more APIs for many different use cases, such as single axis and multi axis configuration.



Debugging

Another advantage of KINGSTAR Soft Motion is debugging. Since multiple applications can be connected to KINGSTAR Soft Motion, it was designed as a separate process from KINGSTAR EtherCAT. This allows a developer to debug their application in Microsoft Visual Studio, setting

breakpoints or stepping through their application without impacting the KINGSTAR subsystem. The subsystem will continue running, supporting all other processes and EtherCAT traffic.

Interpolation and PID

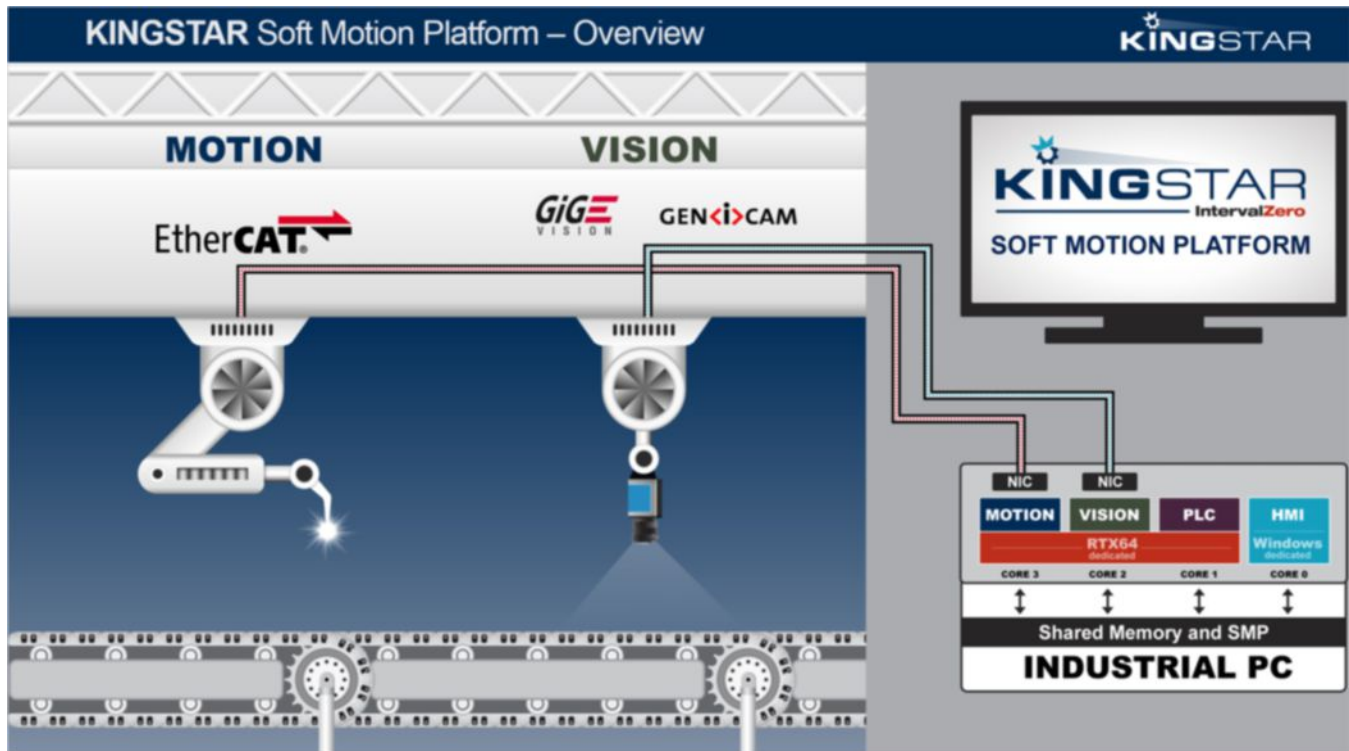
Machine or robot controllers need interpolation and PID functions to control the motors. With KINGSTAR Soft Motion, you can decide for each axis if these features are done in your application, the KINGSTAR subsystem or the motor drive. If you do not want to do the interpolation and PID in your application, using the features inside the KINGSTAR subsystem allows you to have the same configuration options for all the motors no matter the type (linear, servo or stepper) or brand.

For example, you want to build a robot, which means you need a robot controller. This controller can be a real-time application developed in C/C++ using KINGSTAR Soft Motion that does interpolation

itself to control the six axes plus one or two I/O boards. There is a PLC on the side that can be used by the end user or system integrator to add more hardware, I/O or motors to the system, and control them through the PLC and synchronize them with some other applications that are on Windows or on a different machine.

Note: Interpolation and PID are normally real-time functions (which run every cycle) therefore they should be running in a real-time application rather than a Windows application.





Plug-and-play Platform

KINGSTAR was designed as a platform, not a point product, which means that you have the flexibility to add the features you need. You can build your own applications and libraries, use the features built into KINGSTAR and/or get libraries from third-party vendors. For example, you can add robot control or customized motion that can run in RTX64 and be called by KINGSTAR, either in real-time or non-real-time depending on what APIs they expose.

You can also add real-time components that are not related to motion control or EtherCAT but can run on RTX64 at the same time. For instance, for machine vision, you can run the Matrox Imaging Library (MIL) real-time version on RTX64. Your RTX64 applications or Windows applications can link to both KINGSTAR Soft Motion and Matrox Machine Vision, to control and synchronize them together.

Because EtherCAT is only for remote I/O, you can also have local I/O, PCI boards with I/O or any other kinds of protocols through PCIe in RTX64. They can be controlled and connected in RTX64, which means they can be used by your real-time applications at the same time. You can therefore have EtherCAT with motion, machine vision, local I/O, some robot application – the list goes on. Your system can run multiple real-time and non-real-time applications on one single computer, distributed across multiple Windows and RTX64 cores.

Because we provide a standard Windows solution with no modifications, it's not only your non-real-time and HMI applications that you can have on Windows. You can have web services, MES, ERP, any application can be on Windows and can link with KINGSTAR or with your C/C++ applications or with your PLC. You can also have a server or cloud link that would transfer the data from KINGSTAR or other third-party apps to your cloud for remote analysis and access.



Conclusion

With the architecture described above, KINGSTAR delivers true “plug-and-play” functionality that outperforms the traditional hardware approach on performance and cost. A soft motion control platform like KINGSTAR gives machine builders the flexibility, scalability, productivity, and cost-effectiveness to keep pace with technology and outperform the competition.

