Machine Automation Platform

TECH BRIEF

KINGSTAR

Introduction

Historically IntervalZero has been a company focused on Windows based machine controllers. For over twenty years IntervalZero developed and maintained the RTX product line, providing a real-time extension for Windows. RTX is used with control and communication boards to build machine controllers. As the power of processors increase and new Ethernet-based fieldbus protocols emerge, customers continue to ask for software protocols and software-based control logic. Therefore, IntervalZero developed the KINGSTAR automation software platform for building smart machine controllers.

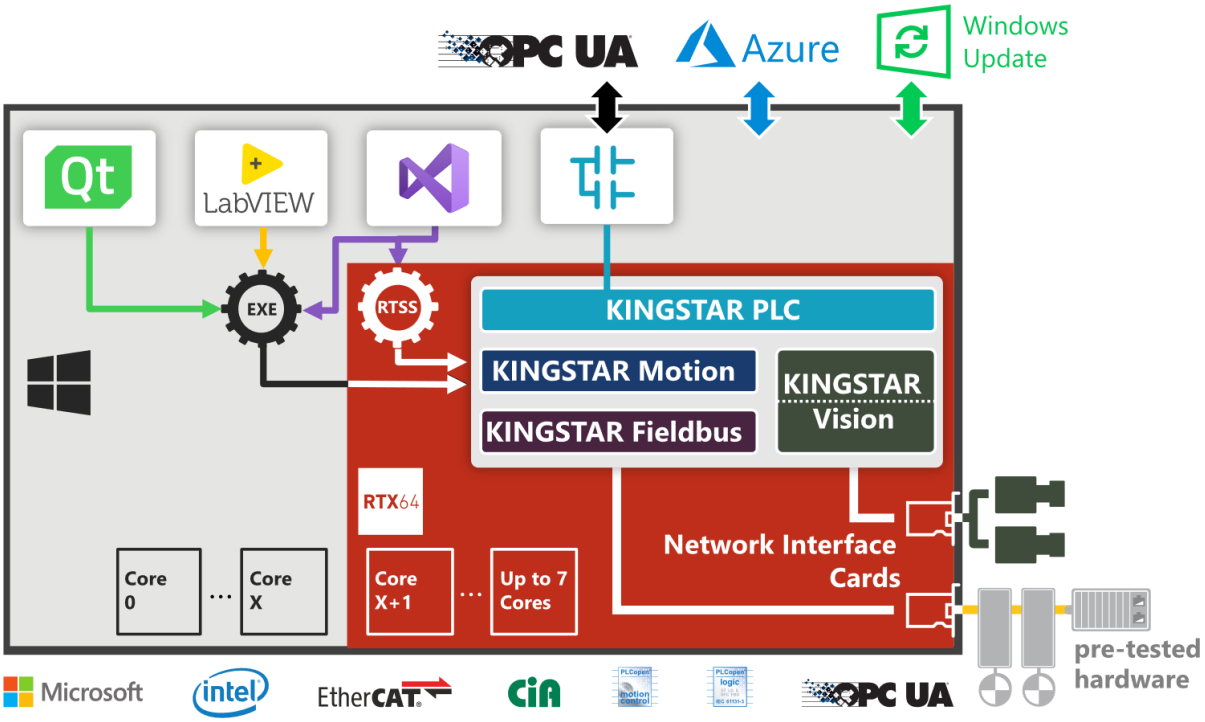
The KINGSTAR offer consists of five products:

- **KINGSTAR Fieldbus** *(real-time EtherCAT® Master)*
- **KINGSTAR Motion** *(motion control)*
- **KINGSTAR PLC** *(software Programmable Logic Controller)*
- **KINGSTAR Vision** *(real-time GigE Vision solution)*
- **KINGSTAR IoT** *(IoT-enabled platform)*

KINGSTAR runs on the rearchitected 64-bit version of the real-time extension, RTX64, which is the core component of the KINGSTAR platform and transforms Windows into a real-time operating system.
Running with Windows 10 64-bit RTX64 allows real-time applications to be developed with C/C++ in Visual Studio. It can be used on a wide range of general-purpose computers from 2 to 64 cores and is deployed in many different industries, such as automation and robotics, but also medical, defense and simulators.

The next component of KINGSTAR is the Fieldbus. KINGSTAR implements a Plug & Play EtherCAT® stack on RTX64. As discussed in a whitepaper that compares the five most important fieldbuses on the market, IntervalZero believes EtherCAT® is the best protocol for machine automation and the KINGSTAR product is based on it. To allow more flexibility to applications, we took advantage of the EtherCAT® bus scan capabilities, to build an automatic configuration feature, which allows the same application to run with different hardware configurations. The benefit of this automatic configuration is, that it supports all the major servo-drive and I/O hardware brands and users can add support for new EtherCAT-based hardware without updating KINGSTAR. Additionally, the fieldbus layer provides direct access to variables, as if they were local, completely hiding the fieldbus from applications.

To further round out the platform for smart machine control, KINGSTAR also provides a software motion component. The motion component comply with the PLCopen Motion Control standard specifications for point to point, synchronized and group motion. With modern processors and the optimized motion equations in KINGSTAR, it is possible to control a large number of axes at fast cycle times. For example, applications can use 20 axes with 125us cycle time or 60 axes with 500us cycle time. Each axis can use a different brand of hardware and have its own control mode. Communication with the drives are based on cyclic synchronous modes, the interpolation is done in the controller, but the PID can be either in the controller or in the drive. The motion algorithms allow modifications of the motion profile while the axis is moving. The synchronization supports electronic camming, gearing and group motion with linear, circular and helical moves. These KINGSTAR Motion options are very flexible as a CAM or gear master axis can have multiple slaves and itself be a virtual axis or even the slave of another axis. These motion features are available to both real-time and Windows applications.

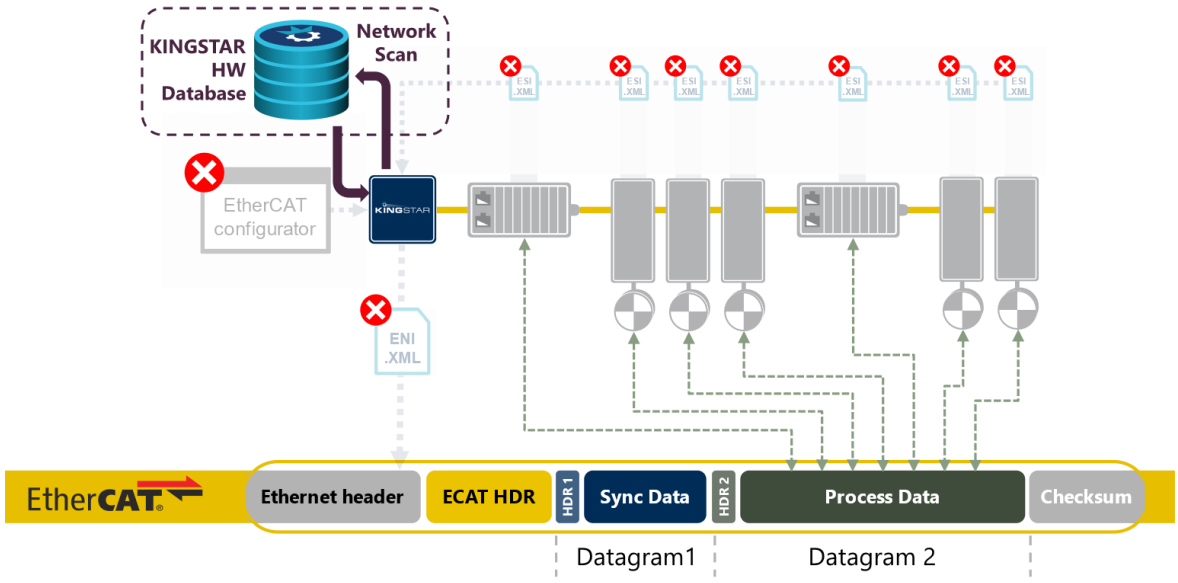


KINGSTAR Architecture

Applications

Motion & CNC controller

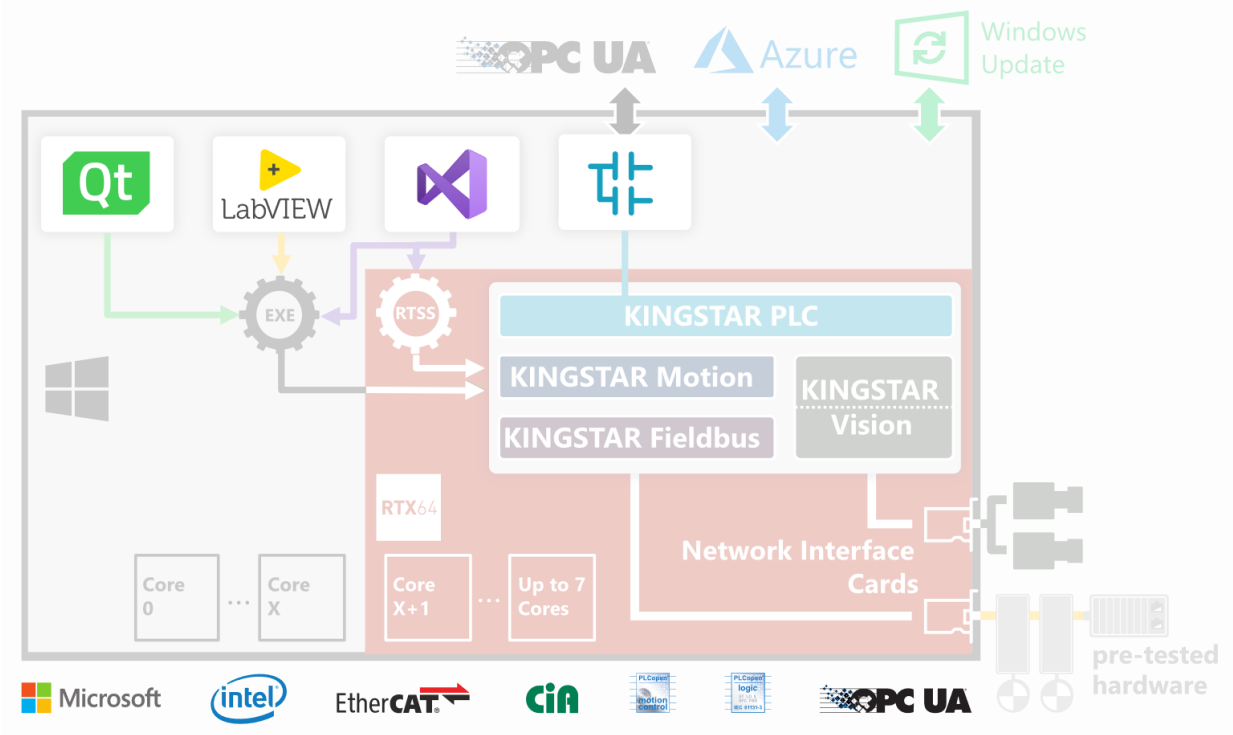
The first customers addressed by the KINGSTAR Fieldbus component were Motion and CNC controller builders. Our customers previously used analog cabling or fieldbuses and wanted to move to EtherCAT®. But controller builders do not know which hardware will be controlled when they develop, so the machine builder – their customer has to create the EtherCAT® Network Information (ENI) file. This means, their customer would need to learn and understand EtherCAT® or that they would have to send integration engineers to every site to do the configuration, which is clearly a downside and a critical issue. This is why we added a layer above the standard EtherCAT® master interface, defined by the EtherCAT® Technology Group (ETG), using the network scanning feature of EtherCAT®, we created an automatic configuration feature with an open database.



KINGSTAR Auto-Configuration

With KINGSTAR Fieldbus, the controller application selects EtherCAT® options, such as the cycle time, Distributed Clocks (DC) synchronization and drive variables to be mapped for motion. When starting KINGSTAR Fieldbus, it will scan the EtherCAT® network to detect the hardware, compare it with the database and enable the requested features if they are supported by the devices. Once started, the list of devices is returned to the application with the features enabled for each device. This way, the controller application automatically has access to the list of devices and their variables, without the need for any EtherCAT® configuration when building the machine. In contrary to many other vendors, the hardware database provided by KINGSTAR is open. It contains devices from a large number of vendors, it is saved in an independent file, so it can be updated independently from the application, a tool is provided to browse and update it by importing new EtherCAT® Slave Information (ESI) files and customers can have their own database file that gets updated independently from the one provided by KINGSTAR.

The other major difference of KINGSTAR Fieldbus is, that it is built as a subsystem instead of a driver, this means that the EtherCAT® master runs in a process independent of the application. This allows the developer to debug their applications with breakpoints, while the EtherCAT® master continues to run and the watchdog protections in the devices are not triggered. Additionally, multiple applications in Windows and Real-time can access the EtherCAT® network at the same time, so the Windows application tool can configure drives and sensors or diagnose issues directly and the Real-time program only handles the process. While commands coming from Windows are buffered through a proxy, the commands from Real-time applications run directly in the application multi-threaded context, so there is no performance reduction compared to a driver.



Develop using your preferred language

Machine Builders

Many builders of testing, inspection, packaging and other machines have converted from analog cabling to proprietary protocols before EtherCAT® became popular. But this came with some restrictions, as they had to buy the controller and hardware of the same brand and it was not possible to add other applications on the controller. A software solution with motion and EtherCAT® allowed them to mix devices from different brands and integrate the control software directly in their application. The KINGSTAR solution with Fieldbus and Motion allows the machine application to poll data at fast cycles and directly diagnose or configure EtherCAT® devices. Machine builders with PCI motion boards, who developed in Windows MFC, .NET or even proprietary applications like LabView, can keep their application and development environment, linking it to the KINGSTAR Subsystem. The KINGSTAR Software simply uses one or more of the CPU cores and a network port to perform the real-time control tasks that the board used to run. If the application needs part of the program to run in real-time, it can program this part in RTX64 in C++ or through the software PLC.

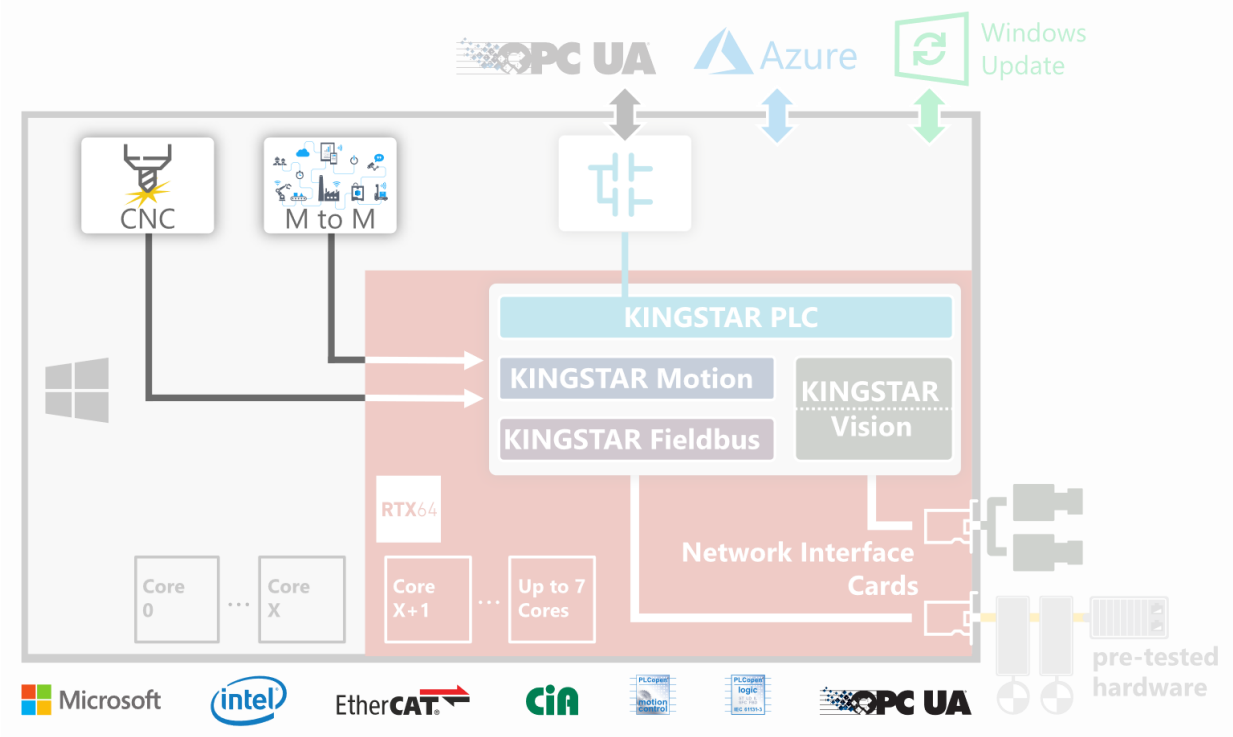
"EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany."

To provide motion features from low to high end while keeping its principle of being an open platform, KINGSTAR looked for a powerful motion control standard. Many motion boards have similar interfaces, but they are too limited, as they do not allow commands to be queued or blended. The most flexible standard we found, is the PLCopen Motion Control standard. It is defined as function blocks for PLC programs, but we extended it to C++ and .NET interfaces. Using this standard, we provide single axis and axis group motion, such as absolute, relative, linear, circular moves and camming or gearing synchronization with the possibility to queue commands, blend them and update them while running. The standard is also flexible enough to allow the use of features, which run directly inside the drive, such as homing and latching encoder positions based on inputs.

For companies who build modular machines, and have an independent development and commissioning teams, we provide a tool to compare the connected hardware with a predefined project. Once the hardware of the machine is assigned to axes and modules defined in the project, it is possible to test and tune the devices directly inside our tool. The tool also allows signals to be mapped to variable names. Once the work is done, the commissioning engineer can save the new configuration and pass it to the application, to allow it to start the devices properly and map the machine hardware to the application variables dynamically.

System Integrators

Given its simple programming languages and reduced testing efforts, many machine builders prefer to use a PLC as their controller. Even though many hardware PLCs are now an embedded computer running a software PLC and controlling modules through local buses and fieldbuses, their design does not allow the users to extend the features directly. This means, when the machine needs market specific machine interfaces, robots or vision, it requires multiple controllers to be integrated together. This task is complex, because the different controllers come from different vendors and are not designed to be synchronized together, or can even break the compatibility with an update. This is why system integrators are turning to software PLC solutions. They can keep the same development, maintenance and HMI components as before, but they can also build their own plugins for the PLC.



KINGSTAR Plugins

The KINGSTAR PLC option is ideal for this, as it includes EtherCAT®, Motion, Modbus and OPC UA by default and allows users to add features to the runtime or development environment themselves Because it runs on RTX64 and Windows, it can use the Windows drivers provided by the vision or robot controller vendors to easily link them with the PLC application. For newer controllers, it is even possible to define functions and function blocks in the PLC, which send commands to a different controller.

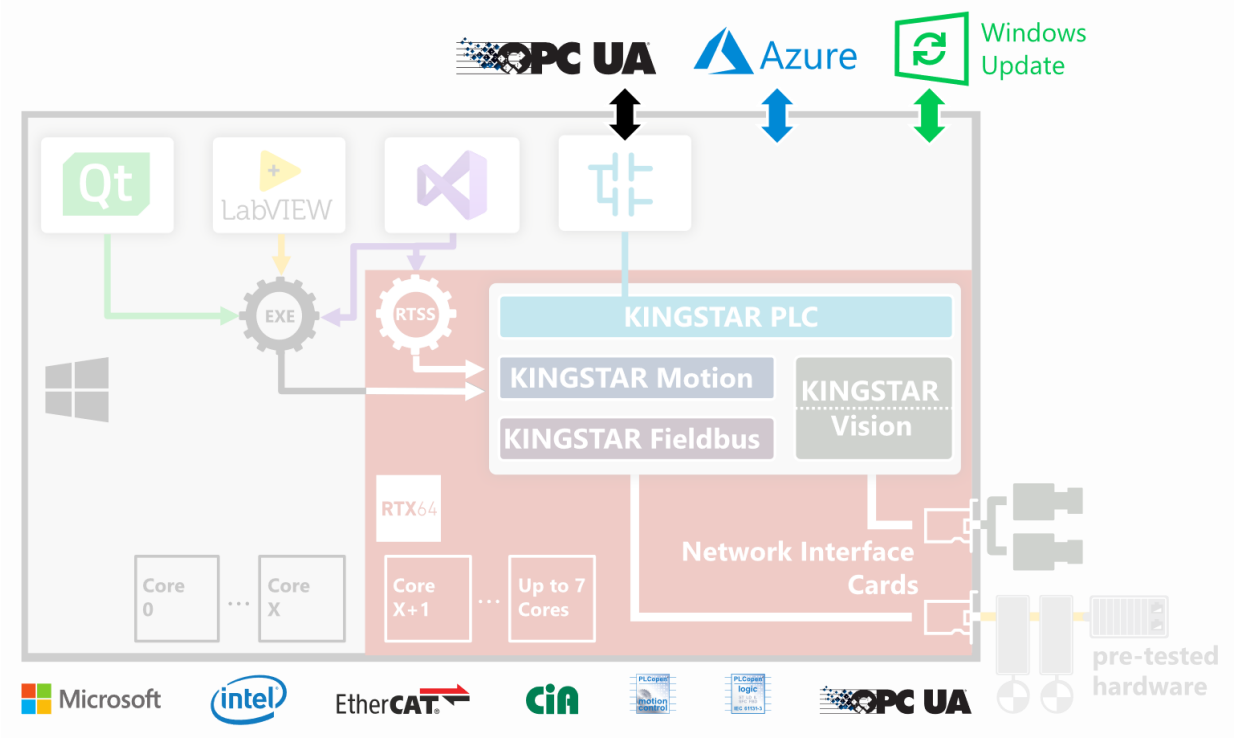
We have built a demo application to prove a single PLC application can control EtherCAT® devices connected to the PLC, along with a Robot and a CNC controller. No application was needed inside the controllers as function blocks inside the PLC could send the instructions directly. As this demo application was using normal Ethernet to communicate with the other controllers, the communication was slow and hardware I/O signals were used for safety around shared resources. But with the publication of the TSN standards, it will be possible to send time-sensitive OPC UA signals between the controllers and thus achieve safety and synchronization of multiple controllers with a PLC using standard protocols.

Industry 4.0 Robot & CNC controllers

Due to its flexibility, the KINGSTAR platform is often used by controller builders who want to develop innovative features, as explained in the KINGSTAR Industry 4.0 and IoT whitepaper. One of the most representative system is the Industry 4.0 Robot controller.

Robot developers write their control algorithms in Real-time C++. Taking advantage of the powerful x86 CPUs, they can build torque control and hand teaching, but also collision detection, comparing the simulated and actual torque and position values. Their robot can be programmed with a script or PLC languages, as their user interface integrates with the PLC development environment and their control algorithm is a PLC plugin. Since KINGSTAR provides auto-configured EtherCAT®, they can allow the end users to connect extra EtherCAT® devices, such as conveyor belts and sensors to the robot controller. The users will then be able to control these devices through the PLC with the standard PLCopen motion control and link them to robot variables, without any extra effort from the robot developers. In addition, the end user can configure, update or diagnose issues in these devices using the standard EtherCAT® tools. The robot application itself will provide status and diagnosis information through the OPC UA server included in the PLC runtime.

The robot controller can even pre-integrate a remote-control mode, where it receives commands from another PLC. The communication is done through a proprietary protocol, OPC UA or even EtherCAT®. By using an EtherCAT® master to master device, the robot can appear as an EtherCAT® slave to other controllers. With this feature, the program inside a single robot controller can control multiple robots of the same brand, allowing them to cooperate easily. As standard machine profiles for OPC UA are being defined, it will even be able to control robots of other vendors, or even CNC or vision controllers.



KINGSTAR – Designed for IoT and Industry 4.0

Conclusion

Industry 4.0 is reshaping manufacturing in the same way that Amazon transformed retailing. The initiative seeks to drive manufacturers and machine builders to integrate all machine automation and machine controller information. By doing so, machine builders can discover new information-based products and identify opportunities to build smarter factories capable of automatically taking corrective or optimizing actions. KINGSTAR has been designed in considering Industry 4.0 requirements and market feedback. Each of its components brings a new piece to the final puzzle. The Windows PC based platform is by far the best option for cloud connectivity and Industry 4.0. KINGSTAR's EtherCAT® master provides some new approaches to implement EtherCAT® quickly and to facilitate the cooperation between development teams and commissioning teams. KINGSTAR is open in terms of development language and can even support mixed development, such as a UI developed with Qt, some specific motion control algorithm in real time with C/C++ and the rest of the application in one of the IEC 61131-C languages. KINGSTAR's architecture was designed so it can be enhanced with plugins for vision, CNC, or other features. Last but not least, thanks to RTX64, KINGSTAR delivers breakthrough machine control performances such as high-speed cycle times down to 100 us or processing scalability from 1 to 7 dedicated cores.