



Q&A with Real-Time Systems

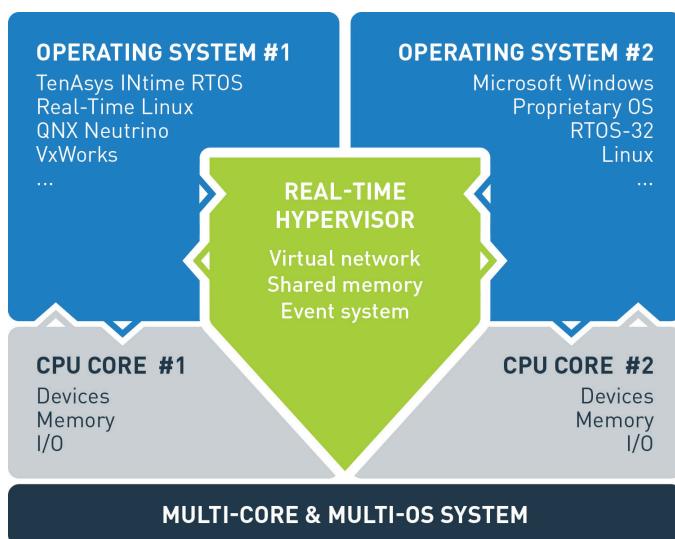
Hypervisors in Robotics

Hypervisors in Robotics

Communications are much faster since you are no longer communicating across cables. Instead, communication happens through a low-latency shared memory space with deterministic time synchronization of operations.

Tell us about Real-Time Systems (RTS) and the history of the company.

Fifteen years ago, when innovators began moving from single-core to affordable, higher-compute, lower-power multicore processors from Intel and AMD, our company's founders realized that workload consolidation was possible. Working then as software architects, designers, and developers at KUKA Robotics, they founded Real-Time Systems (RTS) as a spin-off. They pursued their vision of matching multicore processors' capabilities to the market needs of robotics and industrial automation. Today companies realize the benefits of the RTS hypervisor, including real-time performance and reduced hardware costs.



What is it that sets your hypervisor apart from traditional hypervisors?

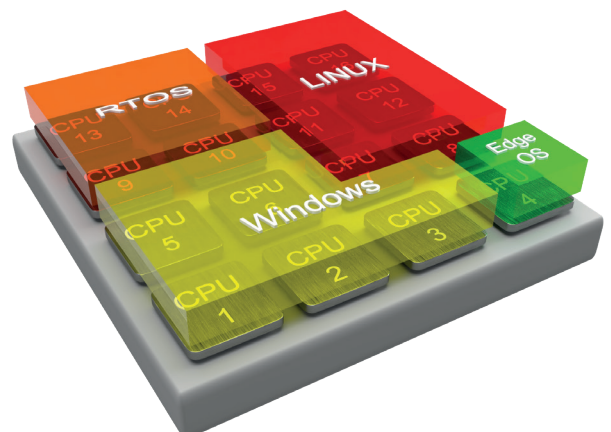
Real-time Systems' hypervisor interacts directly with hardware enabling it to create partitions. This partitioning distinguishes the RTS hypervisor from others in the market, relying on virtualized hardware or emulators. Our competitors have guest operating systems and applications sitting on top of the hypervisor software stack. Even if you are on a bare-metal hypervisor, your applications are still running on top of emulated hardware, not the real hardware. Going through this software layer will add latency and jitter.

If you need real-time performance, you are not going to get it on emulated hardware. Our hypervisor delivers real-time performance.

To do this partitioning when hosting a real-time operating system, we take a multicore computer and first separate the CPU cores, memory, I/O, and PCIe devices. Then we allocate them safely and securely for each operating system.

How are hypervisors being used in robotics?

Rather than having a robot with separate hardware computers for systems such as HMI, vision, motion, data analytics or safety, you can have a robot that places many of these computer systems onto one computer. Our hypervisor makes robotic systems elegant and straightforward by assigning CPU cores, PCIe devices and SATA memory to each system where it runs in its own secure space. Furthermore, communications are much faster since you are no longer communicating across cables. Instead, communication happens through a low-latency shared memory space with deterministic time synchronization of operations. You can also use shared memory for exchanging data. And this is all taking place on one computer where the extra cabling, rack space and computers have vanished, saving substantially per robot system.



You are premium partners with Intel and support TSN and TCC with their new 11th generation processors, formerly known as Elkhart Lake and Tiger Lake. Why is that important and relevant in robotics?

It's important and relevant for the advantages of partnering with Intel and what support for TSN and TCC offers for robotics. For robotics, you need reliable deterministic communications and performance with low latency and minimum jitter. A key component of achieving those goals is the tight integration of partitioning software with TSN and TCC. We have partnered with Intel since 2008, and during the last two years, we have been working closely with Intel's Elkhart Lake and Tiger Lake engineering team to make that tight integration with our hypervisor possible. This effort brings leading-edge silicon capability to robotics with a hypervisor that will support the TSN network interfaces and TCC extensions.

Can you share an example of how a hypervisor can be utilized in a smart factory setting with industrial robots, collaborative robots, IIoT, etc.?

IIoT / Cloud

Traditional Architecture

REAL-TIME
SYSTEMS

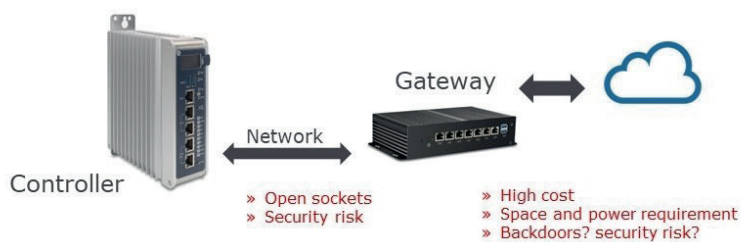


Diagram 1

One example I can share would be using our hypervisor to consolidate two Industrial IoT (IIoT) systems. One system is a controller with a real-time operating system. The other is a gateway running a general-purpose operating system that connects to the cloud, shown in diagram 1. With these two systems running on the same computer, you eliminate redundant computers, cables, spare parts, and reduce security risks. Now communication between the two systems occurs in a shared memory space, which decreases latency, as shown in diagram 2.



Diagram 2

To cite another example, we're also seeing our customers use multiple instances of robotic controllers running on the same hardware. The cooperation between the robots is easily synchronized with TCC on the same computer, eliminating clock drift.

Please give us an idea of what the overall setup entails and what devices are supported.

First, for setup, we ask what operating systems you plan to run on the computer. As a customer or as a company wanting to evaluate our software, you then receive a link to a downloadable installation file. Installation of the RTS Hypervisor requires no additional steps beyond running the installation program. Nor will you need to customize the hypervisor or make any adaptations to the hardware you are already using. You can be up and running quickly with minimum assistance.

As to supported devices, we support any modern x86 architecture, from Intel to AMD, and from mobile chips such as Atom to multi-socket servers.

What does system consolidation look like for your customers?

One RTS customer using our hypervisor for system consolidation is an American conglomerate operating in several market segments worldwide, including power and renewable energy. While system consolidation, in this case, did not involve a complex system, it did lead to cutting their hardware costs in half. The customer had been using two computers, one for a gas turbine controller function, the other for a gateway function. As you can imagine, these two systems individually required more power, rack space, cabling, connectors, and spare parts. Our partitioning software consolidated both functions onto one quad-core processor. The hypervisor partitioned three of the available cores for a QNX real-time operating system, which runs the gas turbine controller. The remaining core was utilized for Linux, which provides the gateway function connecting the system to the cloud.

Author

Tim Willmoth is the Business Development Manager in the U.S. for Real-Time Systems. He holds a bachelor's degree in computer science and a master's degree in business administration. He has 25 years of software experience with embedded systems, working with Fortune 500 companies in the industrial and factory automation market.





Real-Time Systems GmbH is a leading provider of real-time virtualization, hypervisor and operating system technology for embedded and real-time systems. With an extensive knowledge of quality software design, Real-Time Systems GmbH provides a competitive advantage to their customers. The Intel® co-development partner was founded in 2006 and is headquartered in Ravensburg, Germany. Since 2018, Real-Time Systems GmbH is a company of congatec GmbH with partners in Europe, USA, and Asia.

www.real-time-systems.com

Headquarters:
Real-Time Systems GmbH
Gartenstraße 33
88212 Ravensburg
Germany

Phone: +49 751 359 558 0
E-Mail: info@real-time-systems.com

Sales Office:
Real-Time Systems North America
6262 Ferris Square
San Diego CA 92121
United States

Phone: +1 425-802-0240
E-Mail: sales-us@real-time-systems.com

Our partners



© 2021 Real-Time Systems GmbH.
All rights reserved. Subject to change.