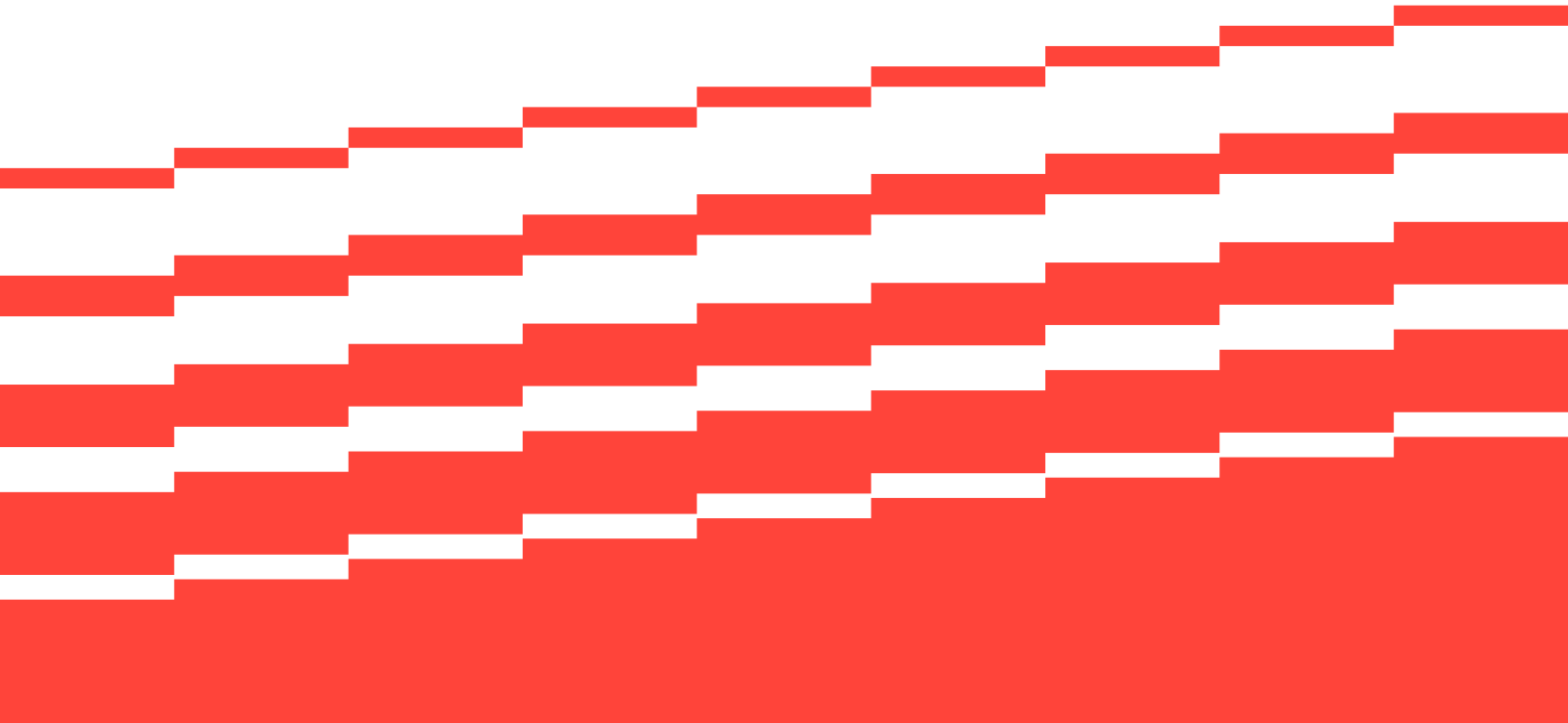
QNX

Product Brief

QNX Hypervisor 8.0

Empowering Embedded Systems with Trusted Virtualization and Mixed-Criticality Support



QNX Hypervisor 8.0

QNX® Hypervisor is an embedded virtualization solution with a microkernel architecture that enables multiple operating systems, including Android™, Linux®, and QNX, to safely operate on the same system-on-a-chip (SoC).

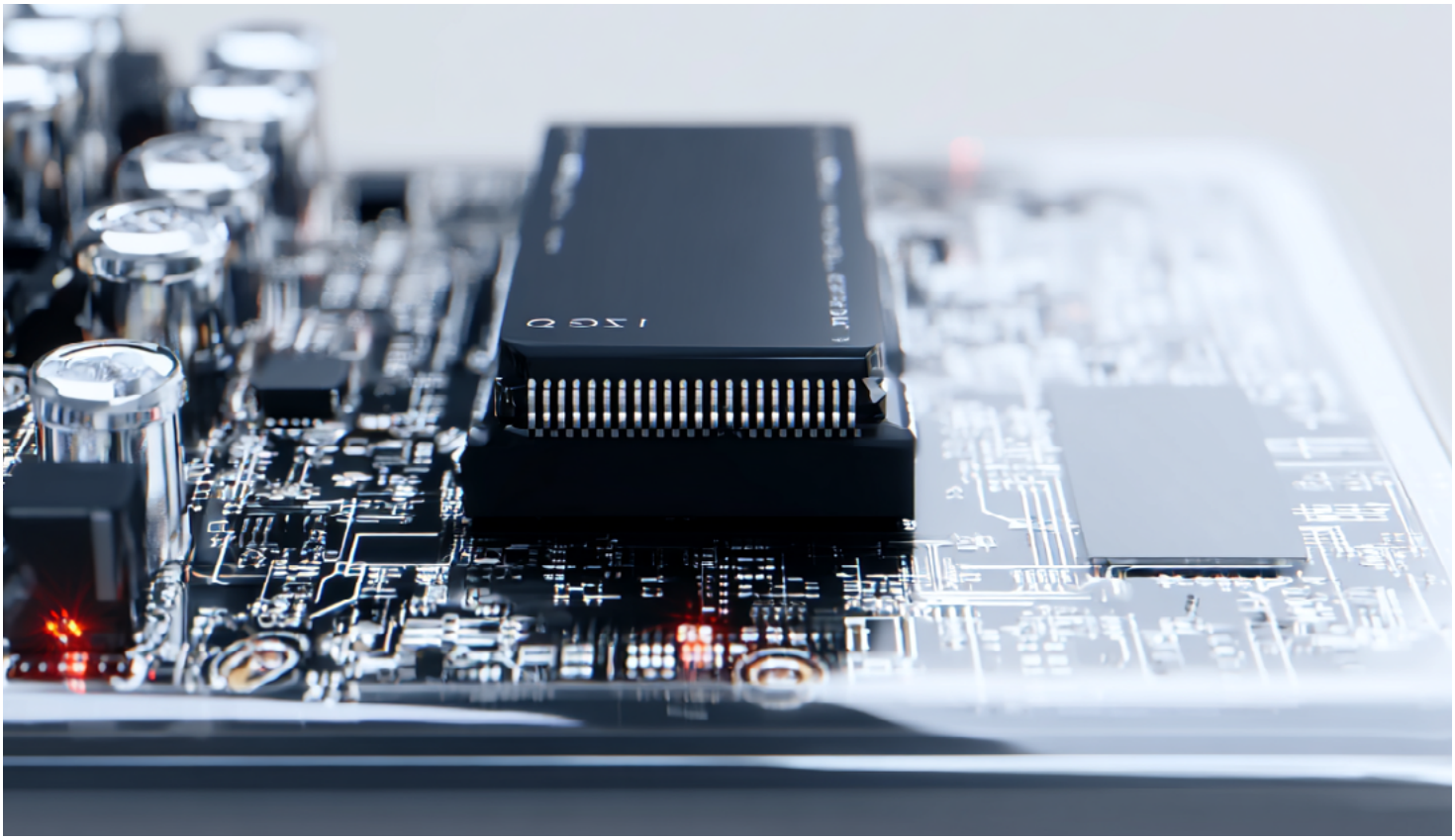
The Evolving Role of Hypervisors in Modern Embedded Systems

The growing demand for hypervisors and virtualization in embedded systems is driven by the need to manage increasing complexity, consolidate hardware, and meet stringent safety and security requirements. As embedded platforms evolve to support mixed-criticality workloads, where safety-critical and non-safety applications must coexist, hypervisors provide the isolation and control necessary to ensure system integrity. They enable developers to run multiple operating systems on a single system-on-chip (SoC), reducing hardware costs, power consumption, and thermal output. Additionally, hypervisors support legacy code migration, accelerate development cycles, and simplify certification by allowing safety-critical components to be isolated and independently validated. With the rise of software-defined architectures in industries like automotive, medical, defense, and industrial automation, hypervisors have become essential for building scalable, secure, and future-ready embedded systems.

Consolidate Diverse Systems on a Single SoC

QNX Hypervisor 8.0 enables the consolidation of diverse operating systems with varying reliability and security requirements, including safety-critical and non-safety workloads, onto a single SoC, supporting secure mixed-criticality system designs. You can, for example, build a safety-critical system certified to standards such as IEC 61508 and ISO 26262 that includes one or more safety-certified virtual machines. These virtual machines can contain non-safety guest systems and run alongside safety-certified software components executing mission-critical work.

No matter your system requirements, QNX Hypervisor 8.0 lets you implement the features you need on your preferred operating systems, while reducing power consumption, minimizing thermal output, and lowering both initial development costs and long-term ownership expenses.



Proven Reliability

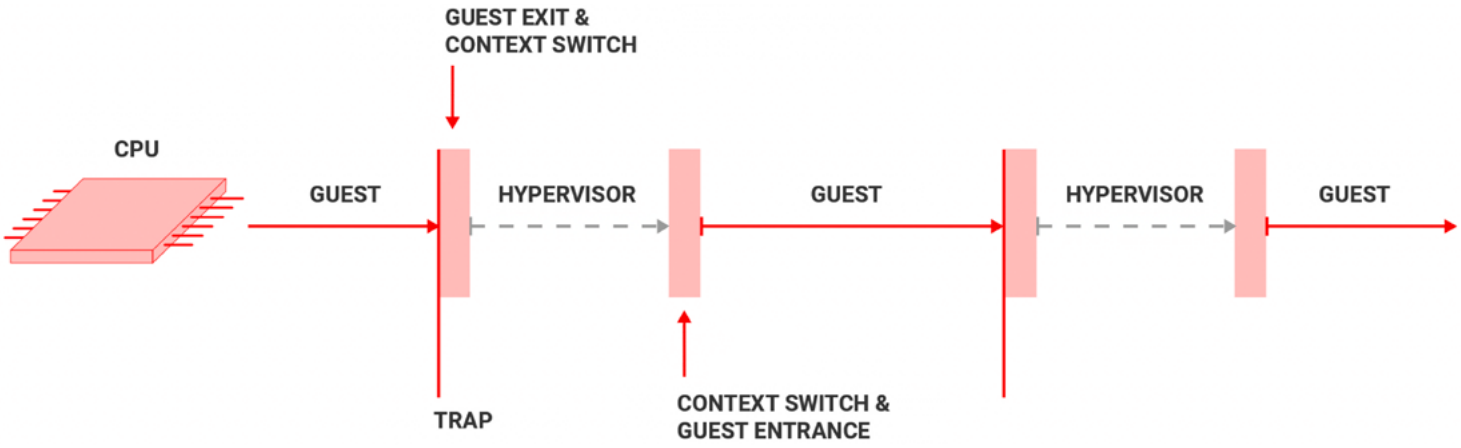
The QNX Hypervisor implements virtualization extensions to the QNX OS microkernel, so when you design a system with the QNX Hypervisor, you are building on a foundation whose reliability and performance have been proven over more than four decades in hundreds of millions of mission-critical systems.

Ensure System Isolation

System integrity and isolation are core strengths of the QNX Hypervisor, enabled by its microkernel architecture. It protects both itself and your system from internal faults and external interference, including from guest virtual machines.

When configuring virtual machines, you define each guest's memory allocation and access to physical and virtual devices. The hypervisor strictly enforces these boundaries. If a guest attempts to exceed its permissions—whether due to error or malicious intent—the QNX Hypervisor blocks the action.

To further enhance isolation, the QNX Hypervisor includes the SMMU manager service (SMMUMAN). This service works in tandem with your hardware's System Memory Management Units (SMMUs) to ensure that Direct Memory Access (DMA) devices remain contained. These devices cannot access memory beyond the limits defined by the SMMUs.



A Lahav Line illustrating how in a QNX Hypervisor system execution alternates between the hypervisor and its guests.

Flexible Virtualization Models

The QNX Hypervisor supports the latest ARMv8 and x86-64 hardware virtualization extensions, providing a highly adaptable solution for embedded systems.

Guest operating systems and their applications can run in virtual machines, with the hypervisor responsible only for handling events and exceptions. Alternatively, a complete native system can be implemented directly on the hypervisor, including resource managers, drivers, and applications, alongside one or more guest systems.

Regardless of the selected model, whether the priority is reliability, performance, security, or cost, the QNX Hypervisor maintains strong isolation between guest systems, the hypervisor itself, and other components on the SoC.

QNX Hypervisor Virtual Machines

A QNX Hypervisor virtual machine is configured much as a physical board is assembled, with memory and devices assigned to the guests or to the hypervisor itself, as required by your implementation. Devices can be physical devices (including pass-through devices) or virtual devices, including emulation and para-virtualized devices from our virtual device library.

Leverage QNX OS Technologies

Virtualization Host Extensions (VHE) are supported on AArch64 processors, enabling more efficient guest transitions to the QNX OS host backends.

Because virtual machines are implemented as QNX Hypervisor processes with virtual CPUs represented as threads within those processes, both symmetrical and bound multiprocessing (SMP and BMP) technologies are used to either pin virtual CPUs to specific physical cores or migrate among core clusters, depending on system requirements.

Develop Custom Virtual Devices

The QNX Hypervisor includes a comprehensive virtual device developer API reference and guide, complete with source code examples that can be used as models for building custom virtual devices. These resources support the development of para-virtualized devices that conform to VirtIO standards, enabling efficient and standardized communication between guest systems and virtual hardware.

Familiar QNX OS API

The QNX Hypervisor is fully compatible with the QNX OS API, allowing developers to begin hypervisor-based development without additional ramp-up time.

Both safety-critical and non-safety applications can be built on the same foundational technologies. Development continues within the POSIX-compliant environment of the QNX Software Development Platform, using the QNX Momentics® Tool Suite for efficient coding, debugging, and analysis.

The QNX Hypervisor at a Glance

Guest OS Support

Run unmodified guests:
- Linux®
- Android™
- QNX OS
- QNX OS for Safety

Processor Support

- 64-bit support for the latest ARMv8 and x86-64 SoCs

Security

- Access control lists (ACLs)
- Mandatory Access Control (process privileges)
- Isolation and separation of guests
- No root mode vulnerable to malicious exploitation

Configuration

- Supports thin (event handler) implementation or full-featured OS implementation
- Guest access to Advanced Configuration and Power Interface (ACPI) tables and Flattened Device Trees (FDTs)
- Easily-modified, cascading configuration files

Networking

- Guest-to-guest
- Guest-to-hypervisor
- Guest-to-world

Memory Sharing

- Guest-to-guest
- Guest-to-hypervisor

Devices

- Assignable to a guest or to the hypervisor
- Configurable device sharing
- Pass-through physical devices
- Extensive library of virtual devices for ARMv8 and x86-64 platforms
- Para-virtualized devices built to VirtIO specifications
- Support for custom virtual devices

Development

- Fully-documented virtualization API
- Virtual device examples with source code
- VirtIO para-virtualization support

Documentation

- User's Guide (includes virtualization basics, virtual machine and device configuration, debugging, and performance tuning)
- Virtual Device Developer's Guide
- Virtual Device Developer's API Reference

For more details, see the user guide →

https://www.qnx.com/developers/docs/7.1/#com.qnx.doc.hypervisor.user/topic/about.html

Related Products

QNX OS

Not building a system that needs certification? The QNX OS powers hundreds of millions of embedded systems in every industry where reliability matters, including automotive, medical devices, robotics, transportation, and industrial automation.

QNX OS for Safety

Don't need a hypervisor system, but need a safety-certified system? The QNX OS for Safety is certified by TÜV Rheinland to IEC 61508 SIL 3, ISO 26262 ASIL D, and IEC 62304 Class C, so you can focus your talents and efforts on developing the systems your customers need.

QNX Momentics Tool Suite

Work with a mix of languages (e.g., C, C++, and Python), and develop for multiple SoC architectures (ARM and x86) simultaneously in a familiar Eclipse-based environment.

QNX Professional Services

We've helped thousands of clients build safe, secure, and reliable systems on the QNX OSs. BlackBerry® QNX system architects and engineers are here to guide you through the complex process of aligning software, hardware, and processes to achieve your project goals.

Safety Services

We offer functional safety training, consulting, custom development, root cause analysis and troubleshooting, and system-level optimization and onsite services across a range of industries and systems. Let us help you with your certification journey.

Virtualization Assessment

If you built your prototype on Linux or another OS but don't know how to proceed with virtualization, we will help you better understand the effort and resources required to port your prototype or project to a QNX Hypervisor system.

Learn More

Find out more about QNX Hypervisor 8.0 →

https://blackberry.qnx.com/en/products/foundation-software/qnx-hypervisor

October 2025

About QNX

QNX, a division of BlackBerry Limited, enhances the human experience and amplifies technology-driven industries, providing a trusted foundation for software-defined businesses to thrive. The business leads the way in delivering safe and secure operating systems, hypervisors, middleware, solutions, and development tools, along with support and services delivered by trusted embedded software experts. QNX® technology has been deployed in the world's most critical embedded systems, including more than 255 million vehicles on the road today. QNX® software is trusted across industries including automotive, medical devices, industrial controls, robotics, commercial vehicles, rail, and aerospace and defense. Founded in 1980, QNX is headquartered in Ottawa, Canada.

Learn more at qnx.com →