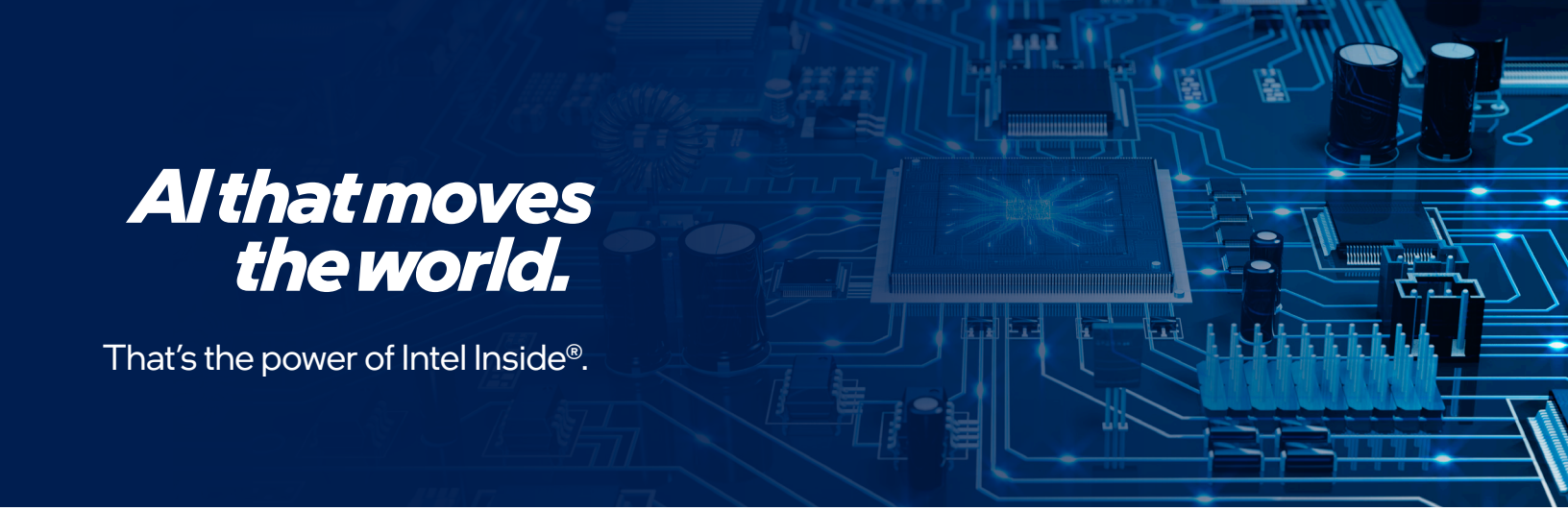

QNX x intel®

Booth 4-544

When Intelligence Needs Determinism

Written by Winston Leung, QNX

Robots were once deployed into environments designed around their limitations. Aisles were fixed. Traffic patterns were prescribed. Human access was restricted. Safety was achieved through separation, and when uncertainty appeared, motion was removed. The robot stopped, the system reset, and work resumed. In those environments, stopping was an acceptable outcome. That assumption no longer holds.

Today's robots operate in spaces where humans are always present and conditions are constantly changing. They share corridors, work areas, and decision-making space with people whose behavior cannot be scripted. Paths are dynamic. Objects appear and disappear. Human intent must be inferred rather than enforced. In these environments, separation is no longer the norm.

At the same time, autonomy has become more capable and more complex. AI-based perception and inferencing now shape how robots understand their surroundings and decide how to move through them. These systems enable robots to function in unstructured environments, but they also introduce uncertainty. Decisions are based on probability, confidence, and learned behavior rather than fixed rules. When treated as authoritative, that uncertainty becomes a safety risk. When treated as input, it becomes manageable. The challenge is not intelligence, it's control.

In many robotic systems today, safety still lives outside the autonomy stack. AI plans motion, navigation executes it, and safety intervenes only when something clearly goes wrong. When timing slips, perception degrades, or confidence drops, the system does what it can reliably do: it stops. This response is safe in isolation, but fragile in practice. Frequent stops reduce availability, disrupt workflows, and create new hazards as humans and robots negotiate around immobile machines. Over time, the robot becomes predictable but unproductive. What's missing is not another algorithm. It is a foundation that allows safety to be enforced continuously, at runtime, under uncertainty, and it begins with how software executes.

For safety to shape behavior rather than override it, safety-critical functions must operate with bounded latency, explicit priority, and freedom from interference. They cannot compete opportunistically with perception pipelines, AI inferencing, or graphics workloads for CPU time and memory. They must be isolated, deterministic, and always able to assert authority over motion. Without those guarantees, safety becomes best-effort, effective only when the system is already behaving well. When blending traditional safety with next-generation probabilistic perception and reasoning, this is where the combination of a QNX software and Intel hardware solution becomes essential rather than incidental.

Intel's embedded processors provide the compute headroom modern robots require. High core counts, hardware-assisted virtualization, and scalable memory bandwidth allow perception, AI inferencing, simulation, visualization, and control to coexist on a single platform. Consolidation is no longer a compromise; it is a necessity. But raw performance alone does not make a system safe. Without deterministic execution and strong isolation, consolidation simply concentrates risk. Intel provides also features to guarantee determinism (i.e., Intel® Time Coordinated Computing), and features to check for silicon integrity allowing for fault detection and error reporting (i.e., Intel® Silicon Integrity Technology).

QNX addresses that risk at the architectural level. Its microkernel design minimizes the trusted computing base and enforces strict isolation between system components. Safety-critical functions execute with guaranteed timing, while complex autonomy workloads are confined to controlled domains. Faults are contained rather than propagated. Failures are observable rather than silent. Most importantly, safety authority is never ambiguous. Together, Intel and QNX turn performance into predictability.

AI inferencing can fully utilize available compute without starving safety functions. Perception and navigation can evolve, scale, and improve without being granted unchecked control. Mixed-criticality workloads can share a platform without sharing failure modes. Safety is no longer a binary response triggered after the fact, it becomes a continuous influence on behavior, shaping speed, trajectory, and motion decisions in real time.

This architectural clarity is what allows robots to remain dependable in the face of uncertainty. Transient faults do not immediately result in shutdowns. Degraded perception does not force a hard stop unless necessary. Motion can be constrained, slowed, or rerouted as confidence changes. Emergency stops remain part of the system, but they are no longer the sole tool for managing risk.

As robots take on more responsibility in human-shared environments, this distinction matters. Availability becomes inseparable from safety. A robot that is technically safe but operationally unreliable will not be trusted, no matter how advanced its autonomy appears.

Building robots that are both intelligent and trustworthy requires more than better AI. It requires a hardware and software foundation designed to enforce safety under load, under uncertainty, and over time. Intel provides the scalable compute to support modern robotics workloads. QNX provides the deterministic, safety-certified execution environment that keeps those workloads under control.

In the end, safe behavior is not something a robot decides. It is something the system is built to guarantee. And that guarantee begins at the foundation.