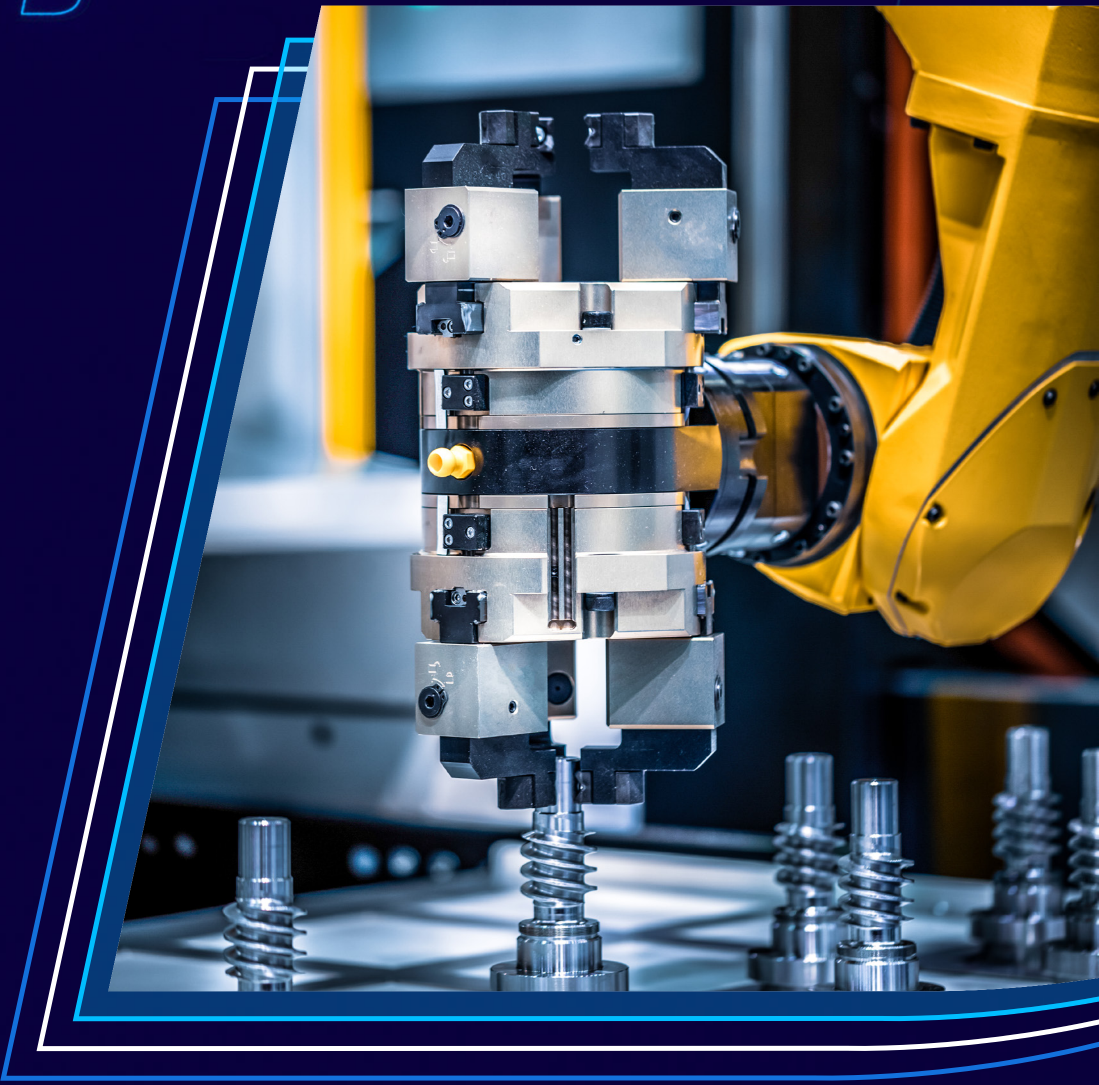


BlackBerry | QNX

OPTIMIZING ROBOTIC PRECISION: UNLEASHING REAL-TIME PERFORMANCE WITH ADVANCED FOUNDATIONAL SOFTWARE SOLUTIONS

WHITE PAPER

The field of robotics is a dynamic and swiftly evolving industry, with an emphasis on the development of advanced robotic systems and components. This evolving landscape underscores the increasing demand for systems that are not only cutting-edge but also marked by accuracy, precision, and reliability. At the heart of these systems are robotic devices (e.g., SCARA, Cartesian, Delta, Articulated Arm, Automated Guided Vehicles, Autonomous Mobile Robots), essential for automating operations. As we delve deeper, we encounter the perennial challenge of achieving the desired levels of precision and accuracy in the world of industrial automation and robotic devices.

One of the most formidable adversaries in this type of software development is “jitter” response. This term alludes to the deviation in timing, manifesting as rapid, small, and often unpredictable variations or fluctuations in a robot’s movements, positions, or behaviours. Jitter is highly undesirable as it can result in imprecise actions, leading to erroneous outcomes, and in the worst-case scenarios, it may induce instability in robotic systems, compelling them to revert to predefined safe states. The importance of minimizing jitter cannot be overstated, especially in applications where precision and accuracy are paramount.

Yet, jitter represents just the tip of the iceberg of performance challenges. The second protagonist in this narrative is “latency.” Latency refers to delay or time lag between an input or action and the corresponding response or output within a robotic system. It is an inevitable aspect of any robotic system since instantaneous response with zero latency is a technological impossibility. The crux of the matter is to determine how much latency is acceptable in any given context. The higher or more unbounded the latency, the greater its potential to disrupt the reliability, accuracy, and precision of robotic systems.

Let us explore the profound impact of latency on specific industries. Take industrial robotics as an example, where latency affects processes that require high precision and synchronization, such as manufacturing and assembly lines, or material handling. In these settings, even a modest degree of latency between the issuance of a command and its execution by the robot can result in misalignment, inaccuracies, and lack of precision in the production process. The consequences can range from the production of defective items to diminished throughput and increased operational downtime.

In the realm of medical robotics, particularly in applications like robot-assisted surgery, latency is not merely a concern – it is a critical factor. In these scenarios, surgeons operate devices remotely, and any delay between their movements and the robotic device’s response can have dire consequences. Increased or unbound latency can lead to imprecise movements and reduced control, ultimately jeopardizing patient safety. Thus, the imperative is to minimize latency, ensuring real-time precision and accurate manipulation in medical robotic systems.

In environments where real-time and deterministic response is a non-negotiable requirement, the underlying operating system that manages the entire software ecosystem that is running the robotic system must be meticulously designed to minimize both jitter and latency. A sluggish or inefficient software framework can lead to delayed actions and jittery movements, which are not conducive to optimal robotic performance. Achieving efficiency in the operational core of the system is paramount, as any shortcomings here can cascade into compromised robot performance and ultimately suboptimal outcomes.

In the following sections, we will embark on a comprehensive journey through the intricate terrain of performance challenges in robotics across a spectrum of industries. We will explore the foundational software strategies, innovations, and best practices employed by developers to minimize jitter and latency, reinforcing our commitment to address the performance challenges that underpin the robotics revolution across various sectors.

Navigating Performance Challenges in Robotics Across Industries

Manufacturing

Logistics and Warehousing

Healthcare and Surgical Robotics

Defense

Aerospace

Space Exploration

As robots become integral to various sectors, the need to address performance challenges has never been more critical. In the following, we delve into the impact of robotic performance, specifically focussing on the issues of jitter and latency that impact the many diverse industries.

Precision in Production

In the manufacturing sector, robotics are the backbone of assembly lines, welding processes, and material handling. The imperative for precise synchronization cannot be overstated. Yet, in this domain, high jitter in high-speed robotic movements can lead to the misalignment of components, the creation of defective products, and a decrease in overall throughput. Notably, in automotive assembly, jitter and latency issues can disrupt the seamless integration of components, resulting in quality control complications and amplified production costs.

Fulfilling Efficiency Demands

The logistics and warehousing industry relies extensively on robots for functions like order picking, package sorting, and inventory management. Jitter or latency-related disruptions can severely impact the efficient flow of goods. For instance, in e-commerce fulfilment centers where robots rely on the accuracy and grip strength of end-effectors, jitter during item picking can slow down order processing or damage products, leading to extended delivery times, diminishing customer satisfaction, and increasing operational costs.

Life and Death Precision

Within the healthcare sector, the integration of medical robotics has revolutionized the landscape of surgeries, particularly in the realm of minimally invasive procedures. Surgeons now employ robotic systems for tasks such as robot-assisted surgery. In this context, latency becomes a matter of life and death. Any delay in the robot's response to a surgeon's movements can result in imprecise incisions, directly influencing patient outcomes. Reducing latency is an utmost priority in medical robotics, and developers persistently strive to guarantee real-time control.

Precision in Operations

Autonomous drones play a pivotal role in reconnaissance, surveillance, and even tactical operations. The demand for flawless performance is undeniable. Nevertheless, in this arena, deviations in robotic movements can have dire consequences. High jitter during critical manoeuvres can lead to inaccurate targeting, collateral damage, and mission failure. This is especially problematic in covert operations where precision and stealth are non-negotiable. Any deviations in robotic actions can compromise the success of the mission, risking national security and the safety of personnel.

Zero Margin for Error

The aerospace industry relies on robotics for tasks ranging from aircraft assembly to the inspection and maintenance of aircraft structures. In this high-stakes environment, the impact of jitter is amplified, particularly in precision tasks like welding aircraft components. Even the minutest deviation in welding position can jeopardize the structural integrity of an aircraft, leading to safety concerns and the imposition of substantial rework costs. Additionally, high latency in robotic responses during critical maintenance operations may lead to extended aircraft downtimes, affecting both operational efficiency and costs.

Precision in the Cosmos

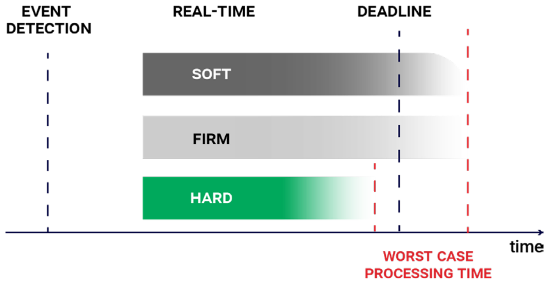
The unforgiving environment of space exploration demands robotics that are free from jitter and latency. During missions such as planetary exploration and satellite servicing, the importance of minimizing these issues cannot be overstated. For instance, a high-latency scenario in a satellite servicing mission can hinder the precise capture and manipulation of a satellite in orbit, potentially jeopardizing the entire mission.

Across these diverse industries, developers diligently attempt to address jitter and latency challenges specific to their applications. Employing advanced control algorithms, robust sensor technologies, and optimized real-time operating systems, they work towards achieving unprecedented levels of precision, safety, and efficiency. In the following pages, we explore the importance of real-time for simplifying and minimizing jitter and latency to address the performance challenges that underpin the robotics evolution across these various sectors.

REAL-TIME COMPUTING AND THE CRITICAL ROLE OF REAL-TIME SYSTEMS IN ROBOTICS

The demand for real-time computing and real-time systems is paramount. These systems go far beyond mere speed; they are the embodiment of precision, predictability, and reliability. Real-time computing is about meeting specific timing requirements, ensuring that actions are not just swift but are carried out within predetermined deadlines. This concept of real-time computing encompasses a spectrum of applications where timing is critical. The significance of real-time computing becomes evident when we delve into the domain of real-time systems.

To comprehend this, we categorize real-time systems into three distinct types:



1. Hard Real-time: Strict time constraints with guaranteed response times. Missing a deadline is not an option as the consequences are nothing short of catastrophic, especially in high-safety critical applications. Consider an autonomous mobile robot (AMR) for example navigating a high-traffic warehouse. Any delay in its ability to respond to obstacles and change direction could lead to collisions, potentially causing damage to goods and posing a safety risk to personnel.

2. Soft Real-time: These systems introduce a measure of flexibility where system operations degrades if it cannot meet specific timing requirements. While they aim to meet deadlines, they can occasionally tolerate

minor deviations without disastrous outcomes. In an industrial setting, vision systems for inspection play a role here. These systems ensure the quality and accuracy of manufactured products, where minor delays in inspection may impact production efficiency but not result in severe consequences.

3. Firm Real-time: Firm real-time systems are akin to soft real-time but with a slight difference. Data arriving after the deadline is often deemed invalid. A prime example in robotics is automated 3D printing systems. In 3D printing, if a layer isn't deposited precisely on time, it can result in defects in the final product. While minor deviations might not be catastrophic, they could lead to the rejection of a printed part, which can impact production efficiency and waste materials.

The significance of these distinctions becomes clear when we consider the intricacies of robotics.

Real-time computing and systems play a pivotal role in meeting these diverse timing requirements. The key to achieving these goals is a real-time operating system (RTOS), which distinguishes itself from conventional operating systems through specialized scheduling algorithms and minimal interrupt/task switching latency. It ensures that critical tasks are executed with precision and adherence to their specific time constraints. With an RTOS, robotic systems can navigate the complex interplay of hard, soft, and firm real-time requirements, delivering reliability and predictability in the face of ever-changing environments and inputs.

THE ROLE OF RTOS IN REDUCING JITTER AND LATENCY

In this intricate landscape, software takes centre stage in the quest to minimize latency and jitter. Operating systems play a pivotal role in managing and mitigating these challenges, and the RTOS emerges as the unsung hero of this narrative. The "Real-Time" designation is not mere rhetoric; it is rooted in the RTOS' unique characteristics – predictability and determinism – that set it apart from its general-purpose OS counterparts. RTOS tasks are executed within predefined and unforgiving time boundaries, a guarantee that has made it the cornerstone of modern robotics software.

Real-Time Scheduling: a RTOS introduces a sophisticated array of real-time scheduling mechanisms, akin to the conductors of a well-orchestrated symphony. These mechanisms schedule threads on each CPU, factoring in their priority, precise timing constraints and an array of other parameters. This orchestration ensures that critical processes adhere rigorously to their specified deadlines, leaving no room for deviation and, in turn, minimizing latency in time-sensitive applications.

Determinism: At the heart of the RTOS is determinism. It enforces deterministic timing behaviour, meticulously ensuring that processes adhere to predefined timing intervals and meet specified time constraints with unwavering commitment. This adherence to schedules forms the bedrock for reducing variability and unpredictability in task execution times, laying the foundation for the minimization of jitter.

Task Prioritization: In the complex web of robotic applications where multiple threads or processes operate concurrently, proper task prioritization ensures that critical threads, such as control loops, are granted the utmost priority, allowing them to execute within their predefined time slots without interruption. Lower-priority processes or threads can be pre-empted, yielding the state to higher-priority processes, a dynamic that is crucial for minimizing latency.

Priority Inheritance: An RTOS stands as protective barrier from intrusion of priority inversion, a scenario where a low-priority task holds a resource urgently required by a higher-priority task. In this context, the RTOS ensures that high-priority processes are never held hostage by their lower-priority counterparts, fortifying the defences against latency.

Interrupt Handling: Interruptions are an inescapable reality, typically stemming from hardware or software issues, and the RTOS' role is to effectively navigate through these disruptions. Its mechanisms for handling interrupts are characterized by precision and predictability. When an interrupt occurs, the RTOS responds swiftly, deterministically, and decisively. This responsiveness is paramount for managing latency and jitter, ensuring that the rhythm of robotic performance remains unwavering.

Deadlock Avoidance: Deadlocks akin to traffic gridlocks in the world of robotics, are situations where threads find themselves hopelessly immobilized due to resource conflicts. The RTOS steps in with tools to alleviate these conundrums. By preventing deadlocks, the RTOS empowers a continuous flow of task execution, an essential facet in the overarching mission to reduce latency.

In this intricate world where precision is the watchword, the RTOS emerges as a beacon of predictability and determinism. It meticulously manages robotic operations, ensuring that tasks are executed seamlessly and efficiently. These features encapsulated within an RTOS underpin the mission to minimize latency and jitter, guaranteeing that robots operate with precision and reliability across an array of applications. It is without a doubt, the foundation for optimizing robotic performance, and BlackBerry QNX is the driver of this technology.

LEVERAGING MICROKERNEL ARCHITECTURE FOR UNPARALLELED JITTER AND LATENCY MITIGATION

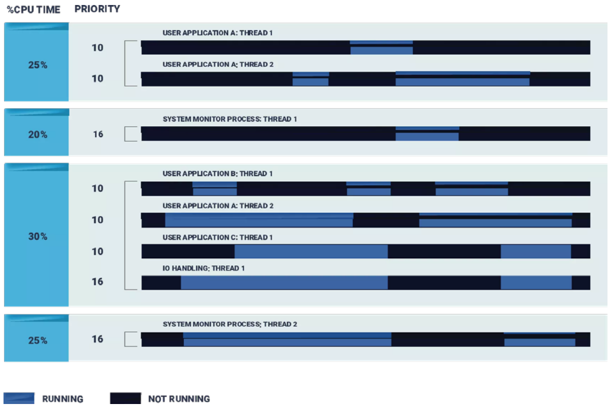
Where precision and real-time responsiveness are the twin pillars of success, the choice of an operating system becomes paramount. In this realm, the BlackBerry® QNX® RTOS stands as a beacon of performance and efficiency, uniquely equipped to address the relentless challenges of jitter and latency. Anchoring this capability is the microkernel architecture, a foundational element that sets the stage for the attainment of unparalleled precision.

"Jitter is the enemy of stability when it comes to our clients' surgical robotics control models. In our engagements, we must reconcile strict timing requirements with the realities of hardware limitations, proposed software architectures, and data throughput. QNX allows us to conduct full, meaningful system characterization and take a targeted approach to reduce jitter and therefore increase stability, performance, and ultimately, safety."

– Royce Stubbs, Senior Software Engineer, MedAcuity

REAL-TIME DETERMINISM

The QNX® RTOS goes a step further, offering hard real-time determinism where a missed deadline is unequivocally considered a failure or fault. This level of determinism is paramount. It ensures that control loops and sensor feedback are processed with unwavering punctuality, adhering precisely to specified time constraints. This deliberate consistency in task execution times leaves no room for variability, thereby mitigating jitter and fortifying the system's stability and precision.

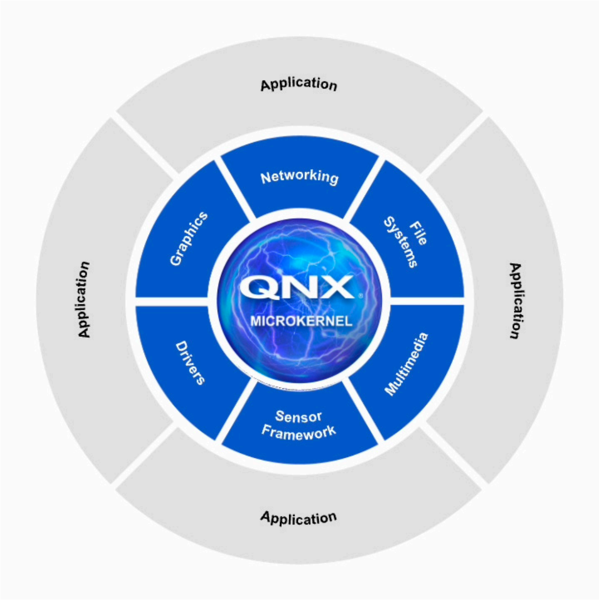


RESOURCE ALLOCATION PRECISION

The QNX RTOS is fortified by sophisticated mechanisms and scheduling algorithms that epitomize the precise allocation of CPU resources. This meticulous allocation guarantees that critical processes receive the computational resources they require without contention. The result is an environment of enhanced predictability, as essential processes are granted unhindered access to the resources they demand. This capacity to guarantee resources contributes significantly to the system's consistency and reliability in robotic control.

PROTECTED MEMORY ISOLATION

Within the QNX RTOS, the microkernel architecture introduces a paradigm of protected memory isolation. It meticulously segregates user applications, device drivers, and core OS services into their own discrete memory spaces. This meticulous isolation serves as an impenetrable fortress, effectively limiting the ripple effects of software failures and faults. Should any software service encounter a fault, the microkernel leaps into action, ensuring prompt containment and potential service restoration - all without introducing disruption or system-wide latency. This fortification is a linchpin in preserving the system's real-time performance.



SWIFT FAULT RECOVERY

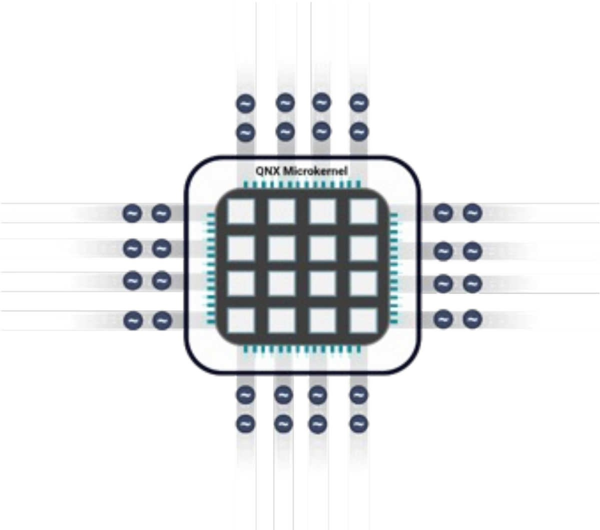
Failure, while unwelcome, is an occasional visitor in any operational environment. In robotics, its disruptive potential can be particularly significant. When a component falters, the QNX microkernel promptly restarts only the affected component without impacting the rest of the RTOS. This swift recovery mechanism effectively mitigates the downtime associated with failures, thus reducing the overall impact on the system's real-time performance. These rapid restarts serve as a potent antidote to the jitter that can stem from halted or delayed components.

NEXT GENERATION RTOS MICROKERNEL ARCHITECTURE PERFORMANCE

While these features are available in the QNX® RTOS, QNX® OS for Safety, QNX® Hypervisor, and QNX® Hypervisor for Safety, the next-generation QNX® OS 8.0 introduces further innovative functionalities that enhance overall performance for robotic systems. The redesigned microkernel architecture optimizes thread scheduling, particularly benefiting high-performance computing, artificial intelligence, and machine learning applications. This optimization is critical for enabling the development of tomorrow's functionally safe human-machine collaborative robotic systems and in any devices where performance and reliability work in lockstep to ensuring maximized uptime.

Multi-Threaded Scaling Architecture

Part of QNX OS 8.0's exceptional performance lies in its multi-threaded scaling architecture. Diverging from many other operating systems, this architecture uniquely allows the incorporation of additional cores without any resulting performance deterioration. Notably, it grants a significantly higher degree of parallelism compared to existing microkernel and monolithic kernels. What sets it apart is the fine-grained execution of context switching, eliminating the need for kernel locks entirely. This precision is a significant boon in the pursuit of optimizing system performance. What's crucial is that this architecture does not compromise on the in-field reliability and certification experience that BlackBerry QNX has long been known for. It underpins a robust foundation, ensuring that the next generation of robotic systems benefits from both innovation and a track record of proven reliability and certification.



Resource Allocation Versatility

QNX OS 8.0 is a robust solution designed with specific features to optimize robotic system performance. Its many schedulers including the round-robin, priority-based (FIFO), and sporadic scheduler - offer versatility in managing threads, ensuring equitable resource distribution, execution predictability, and efficient handling of both periodic and aperiodic tasks. Coupled with the multi-threaded scaling architecture, QNX OS 8.0 provides a reliable and adaptable foundation for the development of cutting-edge robotic systems, catering to the precise requirements of the robotics industry.

BUILDING BLOCKS FOR ROBOTIC SYSTEMS

Where precision is not just an aspiration but a necessity, the QNX RTOS and its microkernel architecture stand as indispensable allies. Together, they empower robotics applications to elevate their precision, efficiency, and reliability to new heights. It underpins a robust foundation, ensuring that the next generation of robotic systems are optimized for high performance operations.

For a deeper exploration of how QNX's microkernel architecture can transform your robotic endeavours, we welcome you to reach out and experience the potential for yourself. Unleash the power of real-time control and precision with us.

> ”
>
> *The growing desire for robotics automation has created a large ecosystem of traditional robotics OEMs and budding startups. Compared to the traditional OEMs, these startups are nimble and quick at designing dedicated hardware systems that address specific pain points. Still, they have limited resources to address software-level challenges. As robots are increasingly being deployed in mission-critical and business-critical applications, there is no surprise that end users expect them to consistently perform at high precision and stability over their useful lives. Through its secure hypervisor and OS, precise resource allocation, fault recovery, and real-time determinism, robotics software solutions like QNX are expected to bring clarity and order to this wild West of robotics safety.*
>
> *– Lian Jye – Chief Analyst, Applied Intelligence, OMDIA.*

About BlackBerry: BlackBerry (NYSE: BB; TSX: BB) provides intelligent security software and services to enterprises and governments around the world. The company secures more than 500M endpoints including over 235M vehicles. Based in Waterloo, Ontario, the company leverages AI and machine learning to deliver innovative solutions in the areas of cybersecurity, safety and data privacy solutions, and is a leader in the areas of endpoint security, endpoint management, encryption, and embedded systems. BlackBerry's vision is clear - to secure a connected future you can trust.

.For more information, visit ***BlackBerry.com*** *and follow* ***@BlackBerry.***