

Zuverlässigkeit gestalten

Worauf es bei der Softwarebasis moderner Robotersysteme ankommt



Eine Mitarbeiterin prüft eine Roboter-Arbeitsstation im Produktionsprozess

Industrielle Robotersysteme sind längst nicht mehr auf einzelne Automatisierungseinseln und abgetrennte Prozessbereiche beschränkt. In Fertigung und Logistik agieren sie heute als integraler Bestandteil vernetzter und zunehmend softwaredefinierter Prozesse. Die Roboter übernehmen dabei immer häufiger auch sicherheitskritische Aufgaben, etwa bei der präzisen Materialhandhabung oder der Paradedisziplin schlechthin, der Mensch-Roboter-Kollaboration (MRK). In diesen anspruchsvollen Umgebungen reicht mechanische Robustheit allein nicht mehr aus. Gefordert ist vor allem ein vorhersehbares, fehlertolerantes und auditierbares Softwareverhalten über den gesamten Produktlebenszyklus hinweg.

Betriebssystem als Basis

Die Basis dafür bildet das zugrundeliegende Betriebssystem. Insbesondere Echtzeitbetriebssysteme spielen eine zentrale Rolle, da sie deterministisches Verhalten ermöglichen und eine klare Trennung von sicherheitskritischen Funktionen unterstützen. Mit steigender Systemkomplexität, wachsender Vernetzung und zunehmenden regulatorischen Anforderungen wird die Softwareplattform damit zu einem entscheidenden Faktor für Sicherheit, Skalierbarkeit und Zukunftsfähigkeit moderner Robotiklösungen.

Echtzeitfähigkeit ist Voraussetzung, nicht Option

In vielen industriellen Anwendungen ist nicht maximale Rechenleistung entscheidend, sondern die garantierte Einhaltung definierter Zeitfenster. Bewegungssteuerung, Sensorik und Aktorik müssen innerhalb exakt vorgegebener Zeit reagieren. Echtzeitbetriebssysteme stellen sicher, dass zeitkritische Prozesse deterministisch ausgeführt werden und priorisierte Aufgaben jederzeit zuverlässig abgearbeitet werden können.

Unerwartete Latenzen und Prioritätskonflikte können andernfalls zu instabilen oder sogar gefährlichen Zuständen führen. Besonders in kollaborativen Robotikanwendungen, in denen Menschen und Maschinen in unmittelbarer Nähe arbeiten, ist eine konsistente Reaktionszeit sicherheitskritisch.

Modularisierung

Softwarearchitekturen mit klarer Trennung zwischen Kernel und Systemdiensten bieten hier entscheidende Vorteile. Durch diese Modularisierung wird verhindert, dass Fehler in einzelnen Komponenten das Gesamtsystem beeinträchtigen. Gerade bei präzisen Bewegungssteuerungen in der industriellen Automatisierung ist dies ein zentrales Kriterium. Gleichzeitig erleichtert ein modularer Aufbau die Wartung, Erweiterung und Aktualisierung laufender Systeme, ohne den deterministischen Kontrollfluss zu unterbrechen.

Mixed-Criticality: Sicherheitszonen intelligent trennen

Moderne Robotikplattformen vereinen unterschiedlichste Funktionen auf einer gemeinsamen Hardwarebasis. Dazu zählen sicherheitskritische Steuerungslogik,

Bildverarbeitung, Benutzeroberflächen, Sensorfusion, Kommunikationsmodule oder Edge KI-Anwendungen. Diese Funktionen folgen unterschiedlichen Sicherheits- und Echtzeitanforderungen und müssen dennoch parallel betrieben werden.

Virtualisierungstechnologien und speziell für Embedded-Systeme entwickelte Hypervisoren ermöglichen eine saubere Trennung dieser Funktionsbereiche. Sicherheitskritische Komponenten können isoliert neben weniger kritischen Anwendungen betrieben werden, ohne gegenseitige Beeinflussung. Ressourcen, Kommunikationswege und Zugriffsrechte lassen sich klar definieren und kontrollieren.

Komplexe Systemarchitekturen realisieren

Dadurch lassen sich komplexe Systemarchitekturen realisieren, in denen mehrere Entwicklungsteams parallel arbeiten und klar abgegrenzte Verantwortlichkeiten bestehen. Dies erhöht nicht nur die Systemsicherheit, sondern verbessert auch die Effizienz, Skalierbarkeit und Planbarkeit im Entwicklungsprozess. Gleichzeitig können neue Funktionen schrittweise integriert werden, ohne bestehende sicherheitsrelevante Module zu gefährden.



Die Embedded-World-2025-Demo von QNX zeigte einen Roboterarm mit fortschrittlichen Sicherheits- und Leistungsfunktionen, der nahtlos in eine virtuelle Fabrik sowie cloudbasiertes Daten-Streaming integriert war.

Autor:

João Pereira

Director EMEA

General Embedded Markets

QNX

www.qnx.software

Zertifizierbarkeit als struktureller Vorteil

In sicherheitsrelevanten Branchen ist funktionale Sicherheit Voraussetzung für Marktzugang und Betrieb. Normen wie IEC 61508 SIL 3 definieren hohe Anforderungen an Softwarearchitektur, Entwicklungsprozesse, Testabdeckung und Dokumentation. Die Einhaltung dieser Vorgaben ist komplex, zeitaufwendig und erfordert eine lückenlose Nachvollziehbarkeit aller sicherheitsrelevanten Funktionen.

Eine Softwarebasis, die diese Anforderungen von Beginn an berücksichtigt und über klar strukturierte Sicherheitsmechanismen verfügt, kann den Zertifizierungsaufwand erheblich reduzieren. Standardisierte Architekturen, wiederverwendbare Komponenten und unterstützende Toolchains helfen dabei, Auditprozesse zu vereinfachen und Entwicklungszeiten zu verkürzen.

Gerade bei Robotiksystemen mit langen Produktlebenszyklen oder bei Projekten, die mehrere Normen gleichzeitig adressieren müssen, wird Zertifizierbarkeit zu einem entscheidenden wirtschaftlichen Faktor. Sie beeinflusst nicht nur die Markteinführung, sondern auch Wartung, Weiterentwicklung und internationale Skalierung.

Praxisnahe Tools

Auch im akademischen Umfeld und bei Studierendenprojekten zeigt sich, dass praxisnahe Tools für deterministische Steuerungslogik, modulare Architekturen und zertifizierungsfreundliche Softwarestrukturen zunehmend zum Standard gehören. Entwicklungsplattformen mit freiem Zugang zu sicherheitsrelevanten Basistechnologien und vorgefertigten Toolchains ermöglichen bereits in der Ausbildung einen realitätsnahen Umgang mit funktionaler Sicherheit.

Skalierbarkeit und Offenheit als Investitionsschutz

Robotiksysteme entwickeln sich kontinuierlich weiter, sowohl funktional als auch hinsichtlich der eingesetzten Hardware. Eine zukunftsfähige Softwareplattform muss daher skalierbar sein, von kompakten Steuerungseinheiten bis hin zu leistungsfähigen Multicore-Systemen mit virtualisierten Umgebungen.

Offene und standardkonforme Schnittstellen erleichtern die Wiederverwendung bestehender Anwendungen und fördern die Zusammenarbeit zwischen unterschiedlichen Entwicklungsteams und Partnern. Moderne Entwicklungsumgebungen senken zudem die Einstiegshürden für neue Projekte und ermög-

lichen eine konsistente Toolchain über verschiedene Produktgenerationen hinweg.

Diese Offenheit unterstützt den Aufbau von Entwicklerökosystemen, in denen Innovation kollaborativ entsteht und Wissen langfristig gesichert wird. Für Hersteller bedeutet dies einen wirksamen Investitionsschutz, da Softwareentwicklungen nicht bei jeder neuen Hardwareplattform von Grund auf neu beginnen müssen.

Cybersicherheit beginnt in der Architektur

Mit wachsender Vernetzung steigt die Angriffsfläche moderner industrieller Robotiksysteme. Sicherheitskonzepte müssen deshalb tief in der Systemarchitektur verankert sein und dürfen sich nicht auf einzelne Anwendungen beschränken. Schutzmechanismen sollten bereits auf Betriebssystemebene ansetzen und den Zugriff auf Ressourcen klar regeln.

Architekturen mit minimalem Kernelumfang und klar definierten Kommunikationswegen reduzieren den gemeinsamen Codepfad und ermöglichen fein granulierte Zugriffskontrollen. Ergänzt durch sichere Update-Mechanismen, Integritätsprüfungen und rollenbasierte Zugriffskonzepte entsteht ein robustes Fundament, das auch regulatorischen Anforderungen wie IEC 61508 SIL 3 gerecht wird.

Entscheidend ist dabei, dass Sicherheitsmechanismen von Beginn

an Teil des Systemdesigns sind und nicht erst nachträglich ergänzt werden. Nur so lassen sich langfristig stabile und vertrauenswürdige Robotiklösungen realisieren.

Vertrauen und Zukunftsfähigkeit

Die Softwareplattform entscheidet über Vertrauen und Zukunftsfähigkeit. Robotiksysteme sind heute hochintegrierte und softwaregetriebene Plattformen mit langen Produktlebenszyklen und klaren Sicherheitsanforderungen. Die Wahl der Softwareplattform ist dabei keine technische Detailscheidung, sondern eine strategische Weichenstellung. Sie beeinflusst maßgeblich Zertifizierbarkeit, Skalierbarkeit, Wartbarkeit und Cybersicherheit.

Echtzeitbetriebssysteme mit modularer Architektur, klaren Sicherheitskonzepten und offenen Entwicklungsansätzen schaffen die Grundlage für robuste und interoperable Robotiklösungen. Sie reduzieren Komplexität, senken Entwicklungsrisiken und ermöglichen es, Innovation und Sicherheit in Einklang zu bringen.

In einem Umfeld steigender Normendichte, zunehmender Vernetzung und wachsendem Innovationsdruck wird Vertrauen zu einer zentralen Währung. Eine belastbare Softwareplattform bildet dafür das Fundament, heute ebenso wie für die nächste Generation intelligenter, vernetzter Robotiksysteme. ◀



Sichere Nähe braucht Vertrauen:
QNX schafft das Softwarefundament für Mensch-Roboter-Kollaboration.



Hardware teilen, Sicherheit trennen:
Mit QNX lassen sich Mixed-Criticality-Architekturen effizient umsetzen.