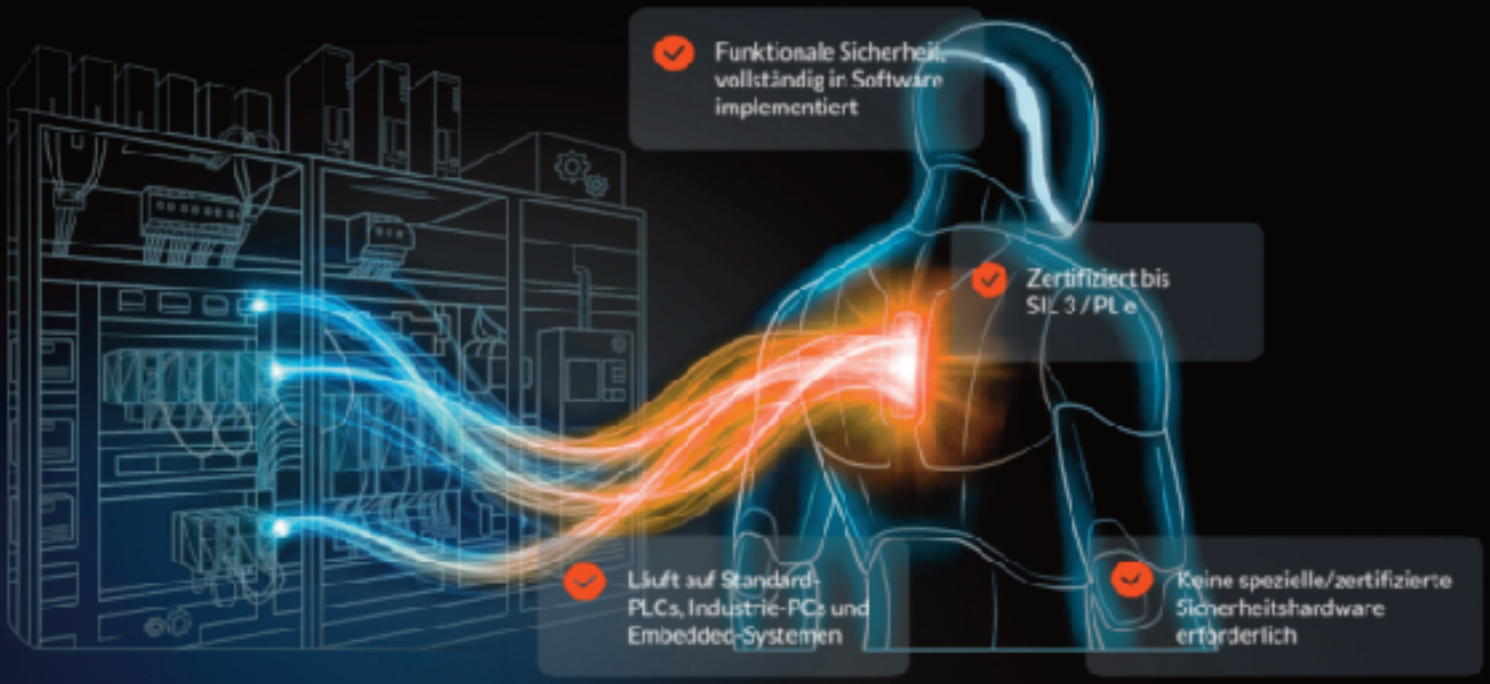

Bild: Neuron GmbH

Durch vordefinierte Architekturen, wiederverwendbaren Referenzdesigns sowie einer einheitlichen Runtime- und Engineering-Basis müssen Unternehmen mit dem Safety-Konzept von Neuron Automation bei einem Projekt nicht jedes Mal bei null beginnen, sondern können auf zertifizierten, skalierbaren Fundamenten aufbauen.

Warum Software die funktionale Sicherheit in der Robotik neu definiert

Ende der getrennten Welten

Artikel anhören!

Auch wenn kollaborierende Robotik als Treiber der Automatisierung gilt, wächst der Aufwand für funktionale Sicherheit aufgrund veralteter Systemarchitekturen überproportional. Da Safety in vielen Robotersystemen noch als separates, hardwarebasiertes Parallelsystem umgesetzt wird, führt dies zu doppelter Hardware, hohem Engineering-Aufwand und komplexen Zertifizierungsprozessen. Bei dem integrierten Ansatz von Neuron Automation, bei dem Safety als softwarebasierte Funktion auf einer gemeinsamen Plattform mit der funktionalen Steuerung ausgeführt wird, reduzieren sich Hardwareaufwand, Entwicklungszeit und Systemkomplexität deutlich.

Während klassische Industrieroboter in abgeschirmten Bereichen betrieben werden, erfordert kollaborierende Robotik eine kontinuierliche Überwachung von Bewegung, Geschwindigkeit und Kraft. Safety wird damit zur zentralen Systemfunktion. Doch hier stößt die klassische Trennung von funktionaler Steuerung und Safety zunehmend an ihre Grenzen.

Status-quo: Zwei getrennte Welten

Der aktuelle Stand in Robotersystemen ist klar: Funktionale Steuerung und Safety sind strikt voneinander getrennt – sowohl architektonisch als auch organisatorisch. Die non-safe-Steuerung berechnet Bewegungen und gibt Sollwerte an lokale Controller weiter, während eine separate Safety-SPS sicherheitsrelevante Parameter wie Position, Geschwindigkeit und Drehmoment überwacht. Daraus wird der Zustand des Tool Center Points bestimmt. Bei Grenzwertüberschreitungen oder beim Verlassen des Arbeitsraums greift die Safety ein und überführt das System in einen sicheren Zustand. Beide Bereiche sind als eigenständige Systeme mit jeweils eigener Hardware, Logik und oft eigener Engineering-Umgebung realisiert.

Safety als limitierender Faktor

Die Trennung von funktionaler Steuerung und Safety hat unmittelbare Auswirkungen auf Aufwand, Kosten und Time-to-Market. Auf Hardwareebene führt sie zu einer doppelten Auslegung: separate Controller, redundante Strukturen und zusätzliche Kommunikationsschnittstellen erhöhen nicht nur die Materialkosten, sondern auch die Systemkomplexität. Parallel dazu entstehen zwei Applikationswelten, die unabhängig voneinander entwickelt, getestet und zertifiziert werden müssen – mit entsprechendem Mehraufwand über den gesamten Entwicklungszyklus hinweg.

Der Integrationsaufwand steigt zusätzlich, da Daten zwischen funktionaler Steuerung und Safety konsistent gehalten werden müssen. Schnittstellen, Synchronisation und Validierung werden zu kritischen Faktoren, die einen erheblichen Abstimmungsbedarf verursachen – sowohl im Engineering als auch später im Betrieb und bei Änderungen. Jede Anpassung zieht potenziell Auswirkungen auf beide Systeme nach sich und erhöht den Aufwand für Tests und Re-Zertifizierung.

Besonders kritisch ist dabei die mangelnde Skalierbarkeit dieser Architektur: Mit zunehmender Funktionalität steigt der Aufwand überproportional. Genau dort, wo moderne Robotiksysteme immer komplexer werden – etwa durch kollaborierende Anwendungen, dynamische Umgebungen oder erweiterte Sicherheitsfunktionen – stößt das klassische Konzept an seine strukturellen Grenzen. Der wachsende Funktionsumfang sicherheitsgerichteter Systeme trifft auf eine Architektur, die ursprünglich für klar getrennte Zuständigkeiten ausgelegt war und diese Komplexität nur unzureichend abbilden kann.

„Zwei vollständig getrennte Systeme, die parallel entwickelt und betrieben werden müssen, sind ein strukturelles Hindernis für moderne Roboterapplikationen“, erklärt Robert Mühlfellner, CTO bei Neuron Automation. „Die klassische Architektur erhöht Kosten, verlängert Entwicklungszeiten und erschwert die Skalierung – genau in dem Moment, in dem Flexibilität und Geschwindigkeit entscheidend sind. Das bremst Innovation.“

Architekturwechsel hin zur Virtual Safety

Mit seinem Safety-Toolkit verfolgt Neuron Automation einen konsequent integrierten und standardisierten Ansatz. Dabei wird Safety nicht mehr als separates Hardware-Subsystem realisiert, sondern als integrierte, softwaredefinierte Funktion innerhalb der Systemarchitektur. Kern dieses Ansatzes ist Virtual Safety beziehungsweise Hardware Independent Safety (HIS). Dabei werden funktionale und sicherheitsgerichtete Funktionen auf derselben Hardware ausgeführt. Die notwendige Trennung erfolgt nicht mehr physisch über separate Systeme, sondern logisch innerhalb einer Runtime-Umgebung.

Technologische Grundlage sind moderne Multicore-SoCs, die eine parallele und deterministische Ausführung von Safe- und Non-safe-Anwendungen ermöglichen. Sicherheitsfunktionen werden in getrennten, softwarediversen Kanälen implementiert und überwachen sich zyklisch gegenseitig. Auf dieser Basis lassen sich Systeme gemäß IEC 61508 bis SIL3 bzw. Performance Level e auslegen und normgerecht nachweisen.

„Eine zentrale Rolle spielt dabei unsere hardwareunabhängige Safety-Runtime, die den Einsatz auf geeigneter industrieller Standardhardware ermöglicht,“ erklärt Robert Mühlfellner. „Damit entkoppeln wir Safety von dedizierten Controllern und machen sie als integrierbare Softwarefunktion verfügbar. Das vereinfacht die Systemarchitektur grundlegend und schafft die Basis für skalierbare, zukunftsfähige Automatisierungslösungen.“

Weniger Hardware – geringere Komplexität

Durch den Wegfall der klassischen Parallelarchitektur lässt sich der Hardwareeinsatz signifikant reduzieren. Separate Safety-Controller und redundante Strukturen entfallen. Daraus ergibt sich ein Einsparpotenzial von größer 50 Prozent bei den Hardwarekosten.

Gleichzeitig sinkt die Systemkomplexität spürbar. Schnittstellen zwischen funktionaler Steuerung und Safety werden reduziert, Daten müssen nicht mehr über getrennte Systeme hinweg synchronisiert und validiert werden. Das vereinfacht sowohl das Systemdesign als auch die Integration, Inbetriebnahme und spätere Anpassungen

Auch auf Engineering-Seite entstehen klare Effizienzgewinne. Statt getrennter Toolchains für Safe- und Non-safe-Anwendungen können beide in einer gemeinsamen Entwicklungsumgebung erstellt werden. Änderungen lassen sich konsistenter umsetzen und Abstimmungsaufwände sowie potenzielle Fehlerquellen werden reduziert.

KI beschleunigt das Safety-Engineering

Ein weiterer Baustein des Konzepts von Neuron Automation ist der Einsatz von KI im Engineering-Prozess. Auf Basis strukturierter Anforderungen – beispielsweise aus der Risikoanalyse – kann man Applikationscode und Testfälle teilweise automatisiert vorbereiten. Ein KI-gestützter Assistent generiert Vorschläge, unterstützt bei Konsistenzprüfungen und reduziert manuelle Arbeitsschritte.

Gerade in den Bereichen Programmierung, Test und Verifikation, die einen wesentlichen Anteil am Gesamtaufwand von Safety-Projekten ausmacht, ergibt sich daraus ein erhebliches Effizienzpotenzial und mögliche Zeitersparnis von bis zu 90 Prozent. „Wichtig bleibt: Die Verantwortung für Validierung und Freigabe liegt weiterhin vollständig beim Entwickler“, sagt Mühlfellner.

Mit diesem integrierten Ansatz verschiebt sich Functional Safety grundlegend: weg von projektspezifischen Einzellösungen hin zu standardisierten, wiederverwendbaren Architekturen. Das durchgängige Safety-Toolkit von Neuron besteht aus Multicore-Referenzarchitekturen, vorzertifizierter Firmware, hardwareunabhängiger Safety-Runtime sowie integrierten Engineering-Tools, Kommunikationskomponenten und Zertifizierungs-Assets. ■

Neuron Automation
www.neuron-automation.eu