# Automatisiertes Fahren
## Notwendiger Wandel der Infrastruktur

Aktuelle Fahrerassistenzfunktionen wie selbstständiges Einparken oder Stau- und Spurhalteassistenten für die Autobahn sind der erste Schritt auf dem Weg zum automatisierten Fahren. Doch die aktuellen Systemarchitekturen stoßen für die in den nächsten Jahren erwarteten Fahrfunktionen bereits an ihre Grenzen. Die zunehmende Vernetzung der Steuergeräte untereinander und die immer komplexeren Funktionen erfordern einen integrativen Systemansatz bei der Entwicklung und eine funktionsübergreifende, fahrzeugweite Softwareplattform. Vorschläge für künftige Ansätze kommen von Elektrobit.


© EB

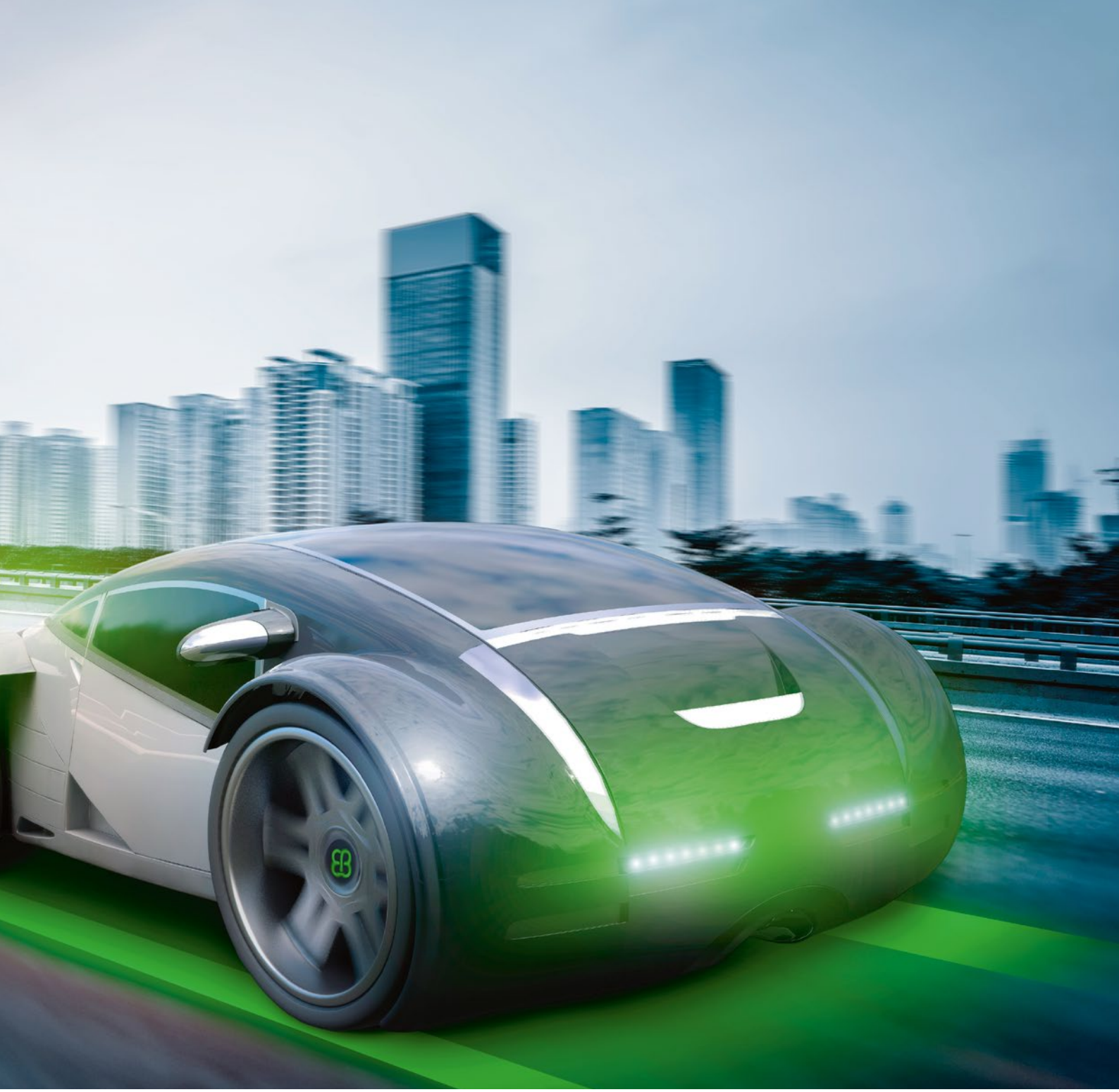
AUTOR

**Dipl.-Ing. (M. Sc.) Karsten Hoffmeister**
ist Senior Manager bei Elektrobit in Erlangen.

### KLASSISCHE ENTWICKLUNG FUNKTIONIERT NICHT MEHR

Eine Vielzahl von Fahrerassistenzfunktionen ist bereits heute verfügbar. Doch bislang ist es dem Fahrer gesetzlich nicht erlaubt, sich auf diese Systeme hundertprozentig zu verlassen. Im Fehlerfall (zum Beispiel beim Ausfall eines Sensors) muss er „sofort“ die Kontrolle wieder übernehmen können. Zukünftige Fahrerassistenzfunktionen werden dem Fahrer eine neue Freiheit geben. Es wird erlaubt sein, das Lenkrad loszulassen und beispielsweise im Stau bei aktivierter Assistenzfunktion ein Videotelefonat zu führen, statt seine volle Aufmerksamkeit dem Stauverkehr widmen zu müssen.

Systeme wie der von Audi für das Jahr 2017 angekündigte Staupilot [1] lassen dem Fahrer mehr Zeit zur Übernahme. Zunächst wird der Fahrer bis zu 10 s Zeit haben, wieder die Kontrolle über das



Fahrzeug zu übernehmen. Dabei ist es dem Fahrer explizit erlaubt, während der aktiven Assistenzfunktion seine Aufmerksamkeit auf andere Dinge zu lenken. Er ist damit für einen eventuellen Fehlerfall nicht mehr verfügbar, was bedeutet, dass das Fahrzeugsystem in dieser Zeit eine gefährliche Situation vermeiden oder bewältigen muss. Bereits dieser scheinbar kurze Zeitraum von 10 s erfordert tiefgreifende Veränderungen in der Systemarchitektur. Die Systeme müssen sich von „Fail Silent“ zu „Fail Degraded“ oder sogar „Fail Operational“ verändern. Das bedeutet, dass sie sich im Fehlerfall nicht einfach abschalten dürfen, ohne die Kontrolle an den Fahrer übergeben zu haben. Sondern es gilt, ein Mindestmaß an Funktionalität aufrechtzuerhalten, um die Sicherheit des Fahrzeugs und seiner Insassen zu gewährleisten, **BILD 1**.

Fahrzeuginfrastruktur, Hardware und Software müssen nun so ausgelegt sein, dass sie selbst im Fehlerfall arbeitsfähig sind. Für die ersten Funktionen kann dieser Notfall-Funktionsumfang rudimentär sein, beispielsweise ein sicheres, langsames Anhalten mit Warnblinkanlage. Dieser „sichere Zustand“ ist im Stau ausreichend, im fließenden Verkehr wird das System aber sicher bis zur nächsten Nothaltebucht weiterfahren oder die nächste Abfahrt nehmen und dort sicher zum Stillstand kommen müssen. Je nach

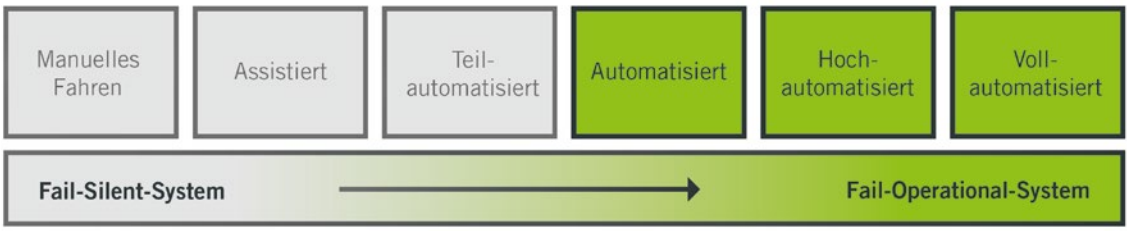



**BILD 1** Die unterschiedlichen Stufen zum automatisierten Fahrzeug erfordern eine Veränderung in der Systemarchitektur (© EB)

geforderter Funktionaliät im Fehlerfall sind viele Teilfunktionen beteiligt, mit all ihren Sensoren, Aktoren und Softwareberechnungen.

Um Funktionen mit der durch das automatisierte Fahren geforderten, oben beschriebenen Verfügbarkeit und Zuverlässigkeit implementieren zu können, bedarf es einer veränderten Systemarchitektur. Denn noch wird ein Steuergerät klassisch entwickelt: Die Implementierung der Fahrerassistenzfunktionen ist statisch und folgt einem festgelegten Muster (Sensordaten einlesen, Regelsollwerte berechnen, Aktoren ansteuern). In Zukunft ist die Entwicklung gezwungen, Lösungen verstärkt auf Systemebene umzusetzen. Einzelne Steuergeräte geraten aus dem Fokus: Kommunikationspfade müssen sich verändern können, Funktionen in Software müssen beim Ausfall von Hardware in Echtzeit auf einer anderen Hardware weiter ausgeführt werden.

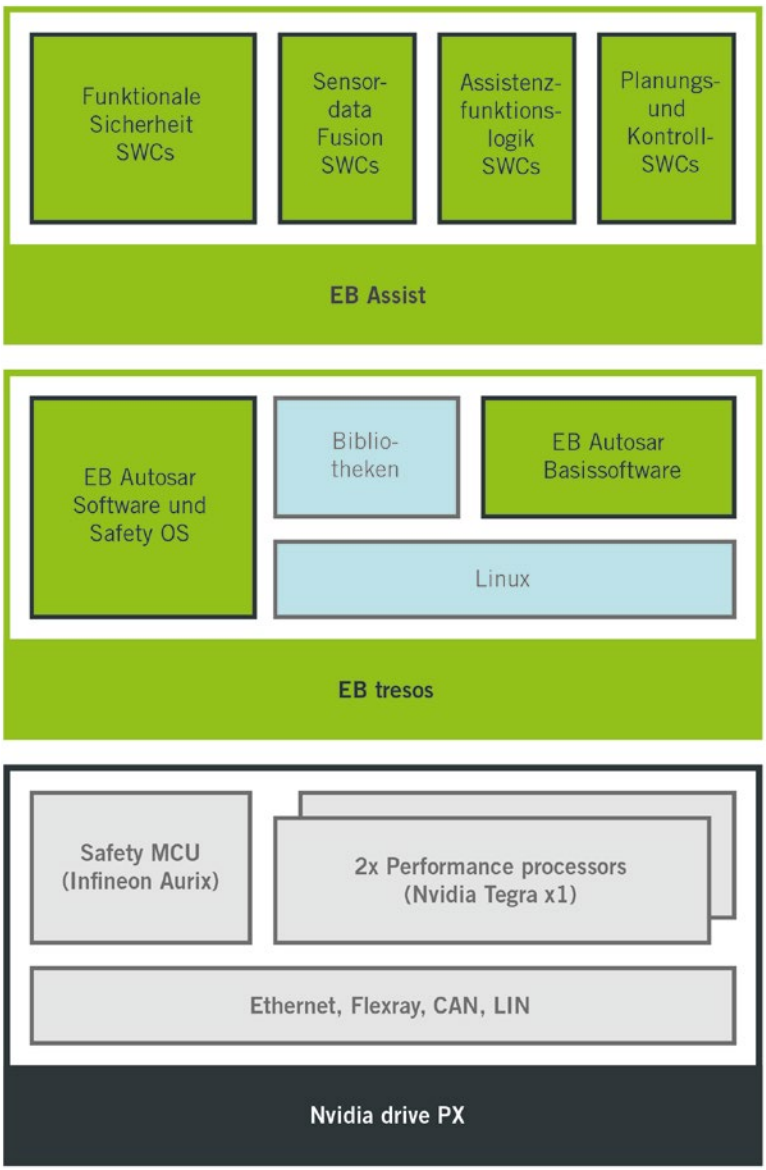



**BILD 2** Softwareschichten in Kombination mit leistungsstarken Hardware-Plattformen (zum Beispiel die Drive PX von Nvidia) bilden die Basis für einen integrativen Systemansatz (© EB)

Hinzu kommen die steigenden Anforderungen für die Umsetzung der eigentlichen Assistenzfunktionen, deren Komplexität im Verbund der Sensoren und Aktoren stetig zunimmt. Der Bedarf an leistungsstarken Prozessoren, die komplexe Berechnungen in Echtzeit durchführen können, wird größer. Teile der Software müssen zusätzlich hohe Sicherheitsanforderungen erfüllen, bis hin zum höchsten Sicherheitsniveau ASIL D (Automotive Safety Integrity Level). Die aber sind auf solchen leistungsstarken Prozessoren nicht ohne Weiteres umsetzbar. Eine Mischung verschiedener Kontrollerarchitekturen als Ausführungsplattform für zukünftige Fahrerassistenzfunktionen ist notwendig. Im optimalen Fall werden diese heterogenen Rechnerarchitekturen in der Automobilindustrie so umgesetzt, dass die Funktionen hardwareunabhängig sind und die Entwickler keine Kenntnisse davon haben müssen, wo und wie ihre Funktionsanteile ausgeführt werden.

## VERÄNDERTE SOFTWAREARCHITEKTUR

Ein Beispiel für eine solche Hardware-Plattform ist die „Drive PX“ von Nvidia, bestehend aus einem Infineon-Aurix-Prozessor als Safety-Kontroller und zwei Tegra-K1-Prozessoren als Performance-Kontroller. Die zugehörige Software liefert Elektrobit mit der Basissoftware

# Keysight TrueIR Wärmebildkamera!

## 4-mal bessere Auflösung und für kurze Zeit 5 Jahre Garantie kostenlos.

Gehen Sie Wärmeproblemen auf den Grund – mit den Keysight TrueIR Wärmebildkameras. Vermeiden Sie ungewollte Ausfallzeiten mit diesem leichten, benutzerfreundlichen, tragbaren Gerät. Mit der hohen kamerainternen Auflösung von 320 x 240 Pixel lassen sich Probleme schnell und klar bis auf 10 cm Entfernung erkennen. Die Auflösung ist 4-mal besser als bei anderen Wärmebildkameras – und das ohne Mehrkosten. Ausfallzeiten können Sie sich nicht leisten – aber die Keysight TrueIR Wärmebildkamera schon.

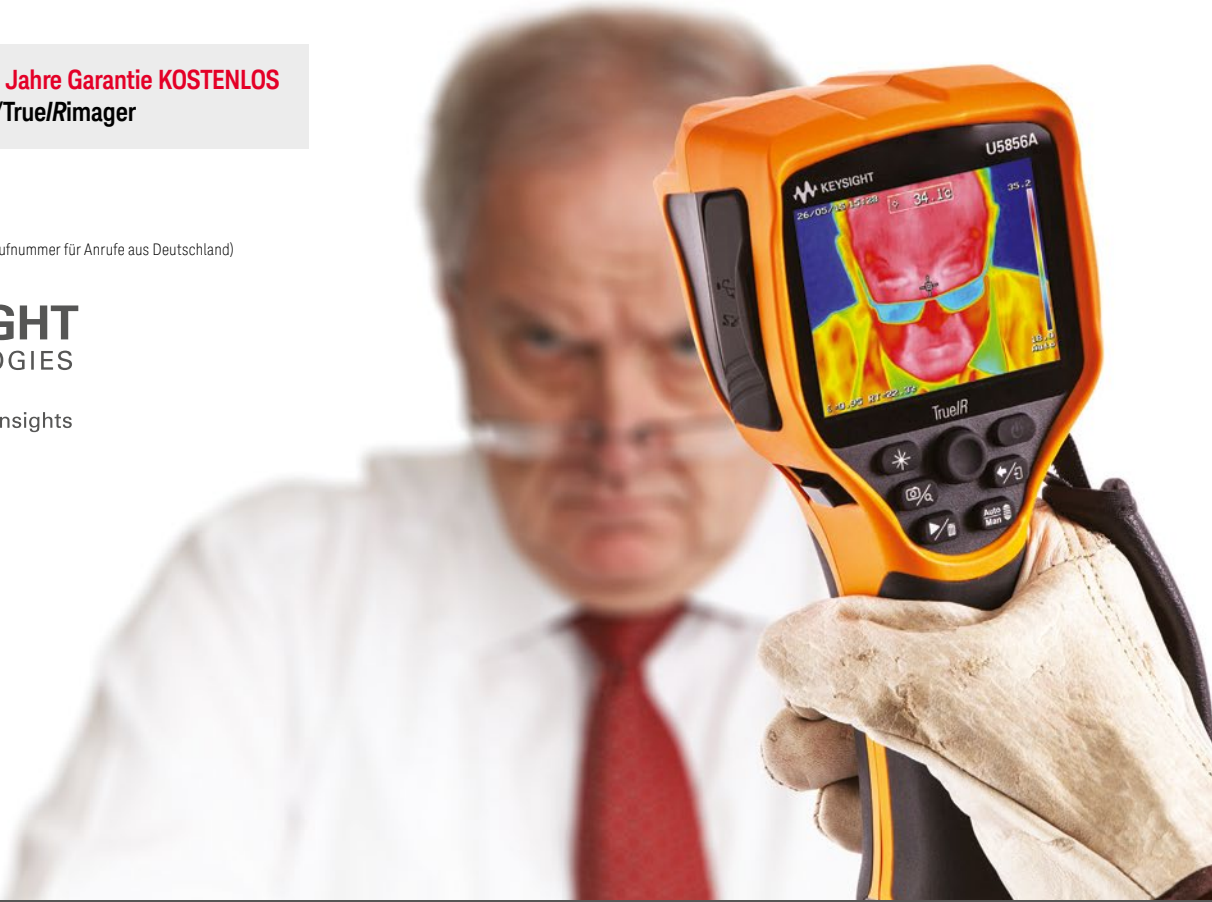
| | U5855A | U5856A | U5857A |
|---|---|---|---|
| Detektorauflösung | 160 x 120 (19.200 Pixels) | | |
| Kamerainterne Auflösung | 320 x 240 (76.800 Pixels) | | |
| Temperatur-Messbereich | –20 bis +350 °C | –20 bis +650 °C | –20 bis +1200 °C |
| Räumliche Auflösung (IFOV) | 3,1 mrad (2,1 mrad bei hoher Auflösung) | | |
| Empfindlichkeit | 0,07 °C bei 30 °C | | |

**Nur für begrenzte Zeit: 5 Jahre Garantie KOSTENLOS**
**www.keysight.com/find/TrueIRimager**

Kontakt:
+49 (0)7031 464 6333
0800 6270999 (kostenfreie Rufnummer für Anrufe aus Deutschland)

KEYSIGHT TECHNOLOGIES

Unlocking Measurement Insights




ECN 2015 WINNER iMPACT Awards
Cooling and Thermal Management
U5855A TrueIR Thermal Imager



**Embedded World 2016** Halle 4 / Stand Nr. 208

EB tresos, die in diesem Fall sowohl auf dem klassischen Autosar basiert als auch auf einer Variante dieses Standards, die erstmals im Sinne des neuen „Adaptive Autosar“ mit einem dynamischen Betriebssystem (in diesem Fall Linux) auf dem Tegra-Prozessor die Laufzeitumgebung bildet.

Als Anwendung läuft auf diesem System ein von EB entwickeltes Framework für Fahrerassistenzanwendungen. Dieses Framework implementiert Module zur Fusion von Sensordaten, Trajektorenplanung und -regelung sowie Standardmechanismen zur Synchronisation und Koordination mehrerer Fahrerassistenzfunktionen im Rahmen eines Situations- und Entscheidungs-Moduls. Die im Framework laufenden Assistenzfunktionen wie eine automatische Einparkfunktion, stützen sich auf die Sicherheitsmechanismen, die von der darunterliegenden Infrastruktur zur Verfügung gestellt werden: Funktionales Management und die Überwachung der Einhaltung funktionaler Randbedingungen werden als Softwaremodule rückwirkungsfrei auf dem Safety-Kontroller ausgeführt, **BILD 2**. Das gewährleistet einen zuverlässigen Betrieb. Zusätzlich ist auf einem Steuergerät gleichzeitig die Ausführung von Software mit hohem ASIL-Sicherheitslevel möglich sowie von nicht sicherheitskritischen Algorithmen mit hohen Leistungsanforderungen (sogenannte „Mixed-Criticality-Systeme“).

Auf dynamischen Softwareplattformen, wie die von Elektrobit auf der Drive PX umgesetzte EB-tresos-Umgebung, sind die Kommunikationswege zwischen den einzelnen Funktionen transparent und flexibel. Ob Anwendungen über eine „Classic Autosar RTE“ mit ihrer Umgebung auf demselben Steuergerät kommunizieren oder diese Kommunikation über die passenden Softwareschichten und Kanäle zu beliebigen Sendern und Empfängern dynamisch auf- und abgebaut wird, ist für die Anwendung zukünftig nicht relevant. Diese Flexibilität in der Kommunikation ermöglicht die Umsetzung von Redundanz-Mechanismen in der Funktionsausführung für Fail-Operational-Systeme, beispielsweise durch die Umsetzung von Hot- oder Cold-Standby-Funktionalität. Das bedeutet, dass Funktionen redundant auf einer weiteren Ausführungsumgebung im

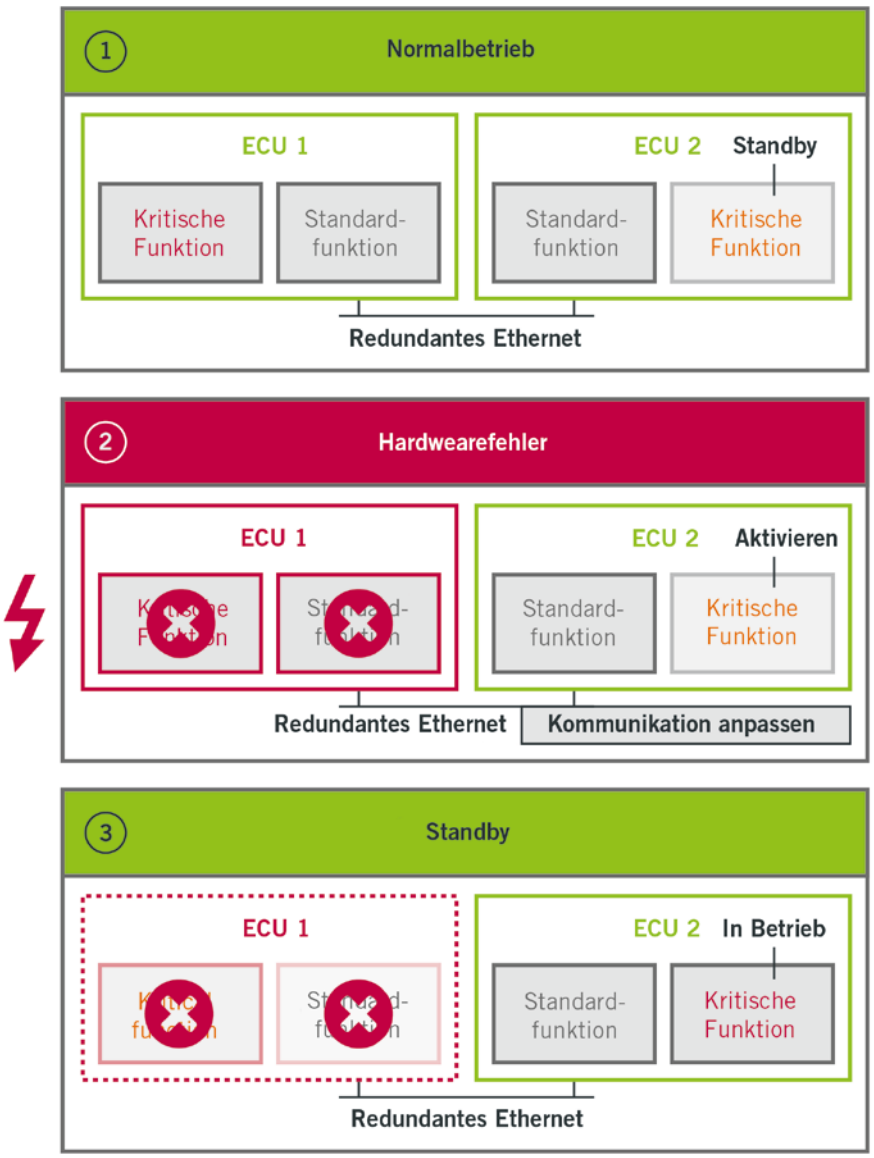



**BILD 3** Dynamische Steuergeräte-Architekturen gewährleisten die Ausfallsicherheit dedizierter Funktionsumfänge (© EB)

Standby sind und dann aktiviert werden, wenn die Hauptfunktion ausfällt. Beim Cold-Standby werden sie erst bei Ausfall der Hauptfunktion aktiviert, sie verdrängen dann gegebenenfalls unwichtigere Anwendungen auf diesem Steuergerät. Beim Hot-Standby laufen sie parallel zur Hauptfunktion, was ein sehr schnelles Umschalten ermöglicht, jedoch dauerhaft Ressourcen wie CPU-Zeit und Speicher bindet. Damit einher geht ein dynamisches Umleiten der Sensor- und Aktor-Signale, je nach Ausführungsort der Funktion, **BILD 3**.

In einer Steuergerätearchitektur, die diese Dynamik zulässt, kann man durch intelligente Verknüpfung mehrerer solcher Steuergeräte auf der Systemebene eine Ausfallsicherheit von dedizierten Funktionsumfängen gewährleisten. Hierzu müssen natürlich auch physikalische Redundanzen in der Stromversorgung und in den Kommunikationskanälen umgesetzt werden. Mit dem Einzug von Ethernet und mehreren Batterien, beispielsweise in einem Elektrofahrzeug, sind hier erste Grundlagen vorhanden, die es ermöglichen werden, eine Software-Infrastruktur-Plattform für zukünftige hochautomatisierte Fahrfunktionen bereitzustellen.

Die Herausforderung bei der Umsetzung liegt darin, dass sie einen system-

weiten Ansatz für die Implementierung einer Softwareinfrastruktur erfordert. Die Funktionen werden auf mehreren Steuergeräten ausgeführt, und die Logik zur Veränderung der Verteilung musssystemweite,dynamischeAspektebe-rücksichtigen. Informationen der einzelnen Steuergeräte über ihre Funktionstüchtigkeit müssen zentral gesammelt werden. Aus diesen Daten können Entscheidungen abgeleitet werden: Je nach Zustand der Hardware beziehungsweise der Integrität der Laufzeitumgebung müssen der Ort der Funktionsausführung und die damit entstehende Umleitung der Signale von Sensoren und Aktoren neu konfiguriert werden. Ziel ist es, dedizierte Funktionsumfänge zuverlässig ausführen zu können, unabhängig davon, welcher Fehler im System auftritt oder welche Hardware ausfällt. Dies kann durch eine serviceorientierte Architektur mit zentralen Kontroll- und Verteilungsmechanismen erfolgen. Es muss eine einzige Plattform entstehen, gewissermaßen ein „Fahrzeug-Gesamt-Betriebssystem“.

Der Bedarf hierfür wird noch offensichtlicher, wenn zu der Fail-Operational-Anforderung weitere neue Technologien hinzukommen, zum Beispiel die permanente Vernetzung des Fahrzeuges mit der Umgebung. Eine solche Online-Verbindung erfordert ebenfalls eine Veränderung in der Architektur, um die erhöhten Security-Anforderungen zu erfüllen. Firewalls, Zugangskontrollen und die Beobachtung des Systems von innen heraus, um potenzielle Manipulationen an sicherheitskritischen Funktionen erkennen zu können (Intrusion Detection), sind weitere Beispiele für steigende Anforderungen an eine im Fahrzeug umzusetzende Software-Infrastruktur-Plattform.

## GANZHEITLICHE ENTWICKLUNG

Ein solches Gesamtsystem aus Hardware und Software ist mit dem heutigen Entwicklungsansatz nur schwer effizient umzusetzen. Noch besteht das Fahrzeug aus kommunizierenden „Einzelkämpfern“, den Steuergeräten mit ihren lokalen Funktionen. Diese wurden teilweise unabhängig voneinander definiert, von unterschiedlichen Abteilungen beim Automobilhersteller spezifiziert und von verschiedenen Zulieferern entwickelt. Die Software basiert teilweise auf Standards wie Autosar, aber von einer universellen Plattform ist man noch weit entfernt.

Neue Player und diverse Start-ups für Elektrofahrzeuge, die sich selbst als „nicht-traditionelle Fahrzeughersteller“ verstehen, sowie natürlich die großen IT-Anbieter haben den Vorteil, dass sie in den neuen Systembereichen wie den Fahrerassistenzfunktionen die Entwicklung von Grund auf starten und strukturieren können. Dadurch werden sie ganz andere Architekturentscheidungen treffen und die Veränderung von Fahrzeug-System-Architekturen beschleunigen.

Automobilhersteller, Tier-1-Zulieferer und Softwareentwickler stehen hier vor der Herausforderung, ihre langfristig gewachsenen Entwicklungs- und Zuliefererstrukturen zu adaptieren. Nur so läßt sich die Systemarchitektur völlig neu gestalten und der Weg zu einer softwaredefinierten Fahrzeug-Infrastrukturplattform ebnen.

**LITERATURHINWEIS**

**[1]** http://www.pcwelt.de/news/Audi-zeigt-Stau-Pilot-Autonom-bis-Tempo-60-Serienreife-ab-2017-9683696.html