

AUTOSAR – so funktioniert es

Mit einem Entwicklungs-Framework sowie weitreichenden Methoden und Tools will die AUTOSAR-Entwicklungspartnerschaft den Wildwuchs der Software im Auto eindämmen.

Ziel ist eine drastische Reduzierung der Varianten und die Entwicklung wiederverwendbarer Software. Der Weg dahin führt über eine radikale Trennung von Hard- und Software.

Wichtige Ziele des AUTOSAR-Konsortiums sind

- Wiederverwendbarkeit der Software (Software Reuse): Software, die einmal geschrieben ist, lässt sich in anderen Entwicklungsprojekten wieder einsetzen, weil sie keine hardware-spezifische Implementierung darstellt. Gegebenenfalls sind Modifikationen auf hoher Ebene leicht durchzuführen. Die Anpassung konventionell erstellter Software an eine neue Prozessorarchitektur dagegen erfordert einen erheblichen Aufwand.
- Verschiebbarkeit: AUTOSAR-konforme Funktionen lassen sich innerhalb eines Fahrzeugs von einer ECU (Electronic Control Unit) zu einer anderen verschieben. Damit gewinnen Entwickler erheblich an Flexibilität. Beispielsweise lässt sich so die Leistungsfähigkeit von Multicore-Prozessoren besser ausschöpfen. Vordenker skizzieren bereits zentralisierte Rechner im Auto – je einen pro Domain (Antriebsstrang, Karosserie, Infotainment etc.). Die Verschiebung der Funktionen findet zum Zeitpunkt der Konstruktion statt; an eine dynamische Verschiebung nach der Auslieferung des Fahrzeugs oder gar zur Laufzeit der Programme ist einstweilen nicht gedacht.
- Standardisierung: Statt für jede Hardware die Software neu zu schreiben oder zumindest anzupassen, werden standardisierte Funktionsmodule verwendet. Das ist nicht nur erheblich billiger, es schränkt auch die Vielzahl der Varianten ein. Letztlich kommt die damit erzielte Vereinfachung auch der Sicherheit der Software und damit der Fahrzeuge zugute.

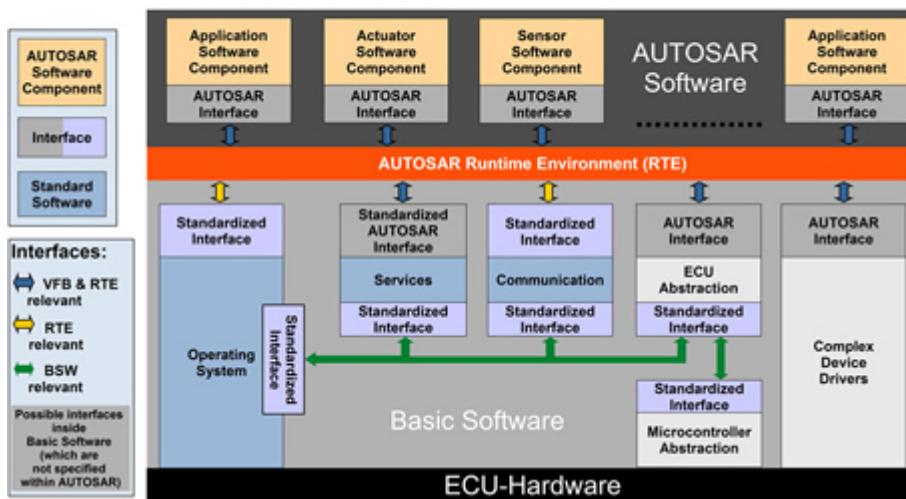
Der AUTOSAR-Standard umfasst eine Reihe architektonischer Elemente und Schnittstellendefinitionen:

- Basic Software (BSW): Die Basic Software enthält einen Hardware Abstraction Layer (HAL), einen Services Layer und das AUTOSAR Runtime Environment (RTE).

- Das RTE stellt die geordnete Verbindung zwischen den funktionalen Softwarekomponenten (Applikationen) und den Basis-Softwarekomponenten her (siehe Abbildung). Damit hat das RTE gleichzeitig die Aufgabe, diese beiden Welten zu entkoppeln. Die Applikationssoftware enthält dann nicht mehr eine Reihe prozessorspezifischer Maschinencode-Anweisungen, sondern beschreibt eine Funktion aus der Sicht des Autokonstruktors. Zwar kostet die Umsetzung ein gewisses Maß an Rechenleistung, doch gehen Experten davon aus, dass die Computing Power in ausreichendem Maße zur Verfügung steht.
- Virtual Function Bus (VFB): Der VFB ist eine Funktion des RTE in Bezug auf eine spezifische ECU. Er gewährleistet die Kommunikation zwischen Applikationen unabhängig von der (Hardware-)Infrastruktur. Die Kommunikation erfolgt über dezidierte Ports. Über den VFB läuft sowohl die Kommunikation innerhalb eines Steuergeräts als auch die Verständigung zwischen unterschiedlichen ECUs (Electronic Control Unit). Von darunterliegenden Abhängigkeiten oder Technologien muss aus Applikationssicht keine detaillierte Kenntnis mehr vorhanden sein. Dies ermöglicht die hardwareunabhängige Entwicklung und Nutzung der Anwendungssoftware. Das Runtime Environment realisiert den Virtual Functional Bus auf jedem Steuergerät. Es ist skalierbar und wird für eine ECU statisch erstellt, zugeschnitten auf die Applikationen dieser ECU.

Ein praktisches Beispiel für den rationelleren Einsatz der Software liefert der aktuelle 7er BMW: Die Entwickler haben hier Funktionen wie Scheibenwischer-, Licht- und Fensterhebersteuerung als AUTOSAR-Softwarekomponenten entwickelt und in zentralen Steuergeräten (ECUs) integriert. Die jeweiligen Softwarefunktionen werden künftig auf neuentwickelten Steuergeräten wieder zum Einsatz kommen, wenn auch in einer anderen Partitionierung. So plant der Fahrzeughersteller, die Zahl der hier relevanten Steuergeräte von vier auf zwei zu reduzieren – ohne dass sich an der Software etwas ändert. Und vor allem, ohne dass sie neu geschrieben werden muss.

Interfaces
Components and interfaces view (simplified)



AUTOSAR-Framework.

Einmal entwickelt – mehrfach verwendet: Strikte Trennung von Hard- und Software verbessert die Möglichkeit, Komponenten und Module wiederzuverwenden. Zudem soll die Qualität und damit die Sicherheit erhöht werden. (Quelle: AUTOSAR)