

## AUTOSAR 4.1 IP-Stack – bereit für den Serieneinsatz!

---

In November 2011 wurde am ersten Ethernet Tech-Day bei BMW eindrucksvoll dargelegt, dass mit Broadcom's BroadR-Reach PHYs die Ethernet Hardware die automotive Hardwareanforderungen für eine Kommunikation via Ethernet mit 100 MBit/s über ein ungeschirmtes, verdrilltes Leitungspaar erfüllen kann. Bezüglich der Basissoftware wird nun mit der im März 2013 erscheinenden neuen AUTOSAR 4.1 Release auch eine aktualisierte und für den Serieneinsatz validierte Spezifikation der Basissoftware für den Einsatz der bekannten TCP/IP Protokollsuite mit einem Switched-Ethernet Netzwerk bereitstehen. Dieser Artikel gibt einen Überblick über die wichtigsten Elemente dieses AUTOSAR 4.1 IP-Stacks, den EB in enger Zusammenarbeit mit BMW spezifiziert und implementiert hat.

### Automotive Anforderungen an den AUTOSAR IP-Stack

Die Automotive Anforderungen an den AUTOSAR IP-Stack für die fahrzeuginterne Kommunikation und den direkten Diagnosezugang können in drei Hauptkategorien unterteilt werden:

- (1) Vollständige Integration in die AUTOSAR Architektur:** AUTOSAR Anwendungen sollen in der Lage sein über das neue Netzwerk zu kommunizieren ohne, dass eine Änderung auf Applikationsebene nötig ist. Des Weiteren müssen die allgemeinen Anforderungen an die AUTOSAR Basissoftware eingehalten sowie der AUTOSAR Konfigurationsprozess unterstützt werden.
- (2) Kommunikationsprotokolle:** Während die vertikale Kommunikation, das heißt die Schnittstelle zwischen den Modulen des AUTOSAR IP-Stacks selbst und zu den angrenzenden Modulen, durch die AUTOSAR Architektur geprägt ist, soll für die horizontale Kommunikation beziehungsweise die Kommunikationsprotokolle möglichst stark auf bekannte und weit verbreitete „Internet Standards“, wie zum Beispiel UDP, TCP, IP, ARP, ICMP und DHCP aufgebaut werden.
- (3) Ressourcenverbrauch:** Der Ressourcenverbrauch des AUTOSAR IP-Stacks muss vorhersagbar, beschränkbar und vor Allem optimiert sein. Dazu muss die Vorabberechnung des Speicherbedarfs möglich sein, eine einzelne Anwendung darf nicht den gesamten Pufferspeicher monopolisieren können und der Bedarf an Zwischenpuffern und Datenkopieroperationen muss möglichst gering sein.

Bereits mit AUTOSAR 4.0.1 wurde eine erste Version des AUTOSAR IP-Stacks spezifiziert. Dabei wurde versucht einen Commercial-Off-The-Shelf (COTS) TCP/IP-Stack in die AUTOSAR Basissoftware zu integrieren. Das Ergebnis war jedoch aus mehreren Gründen unzureichend. So ist die Verwendung der bei COTS TCP/IP-Stacks verbreiteten BSD-Socket-API in typischen AUTOSAR Steuergeräten auf Grund der benötigten OS-Ressourcen ineffizient und nur schwer in die AUTOSAR Architektur integrierbar. Außerdem werden die meisten der oben beschriebenen Anforderungen von einem COTS TCP/IP-Stack nicht ausreichend unterstützt beziehungsweise ist dessen Verhalten nicht ausreichend detailliert spezifiziert oder uneinheitlich.

Damit der AUTOSAR IP-Stack in der Serie einsetzbar wird, wurden in enger Zusammenarbeit von BMW und EB die Spezifikationen im Rahmen von AUTOSAR 4.1 signifikant überarbeitet und erweitert.

## Integration in AUTOSAR

Die Architektur des um die TCP/IP-Protokollsuite und Ethernet erweiterten AUTOSAR Kommunikationsstacks ist in Abbildung 1 dargestellt. Hier ist die nahtlose Integration des neuen Kommunikationsstacks parallel zu den bestehenden Stacks für CAN, LIN und FlexRay gut zu erkennen.

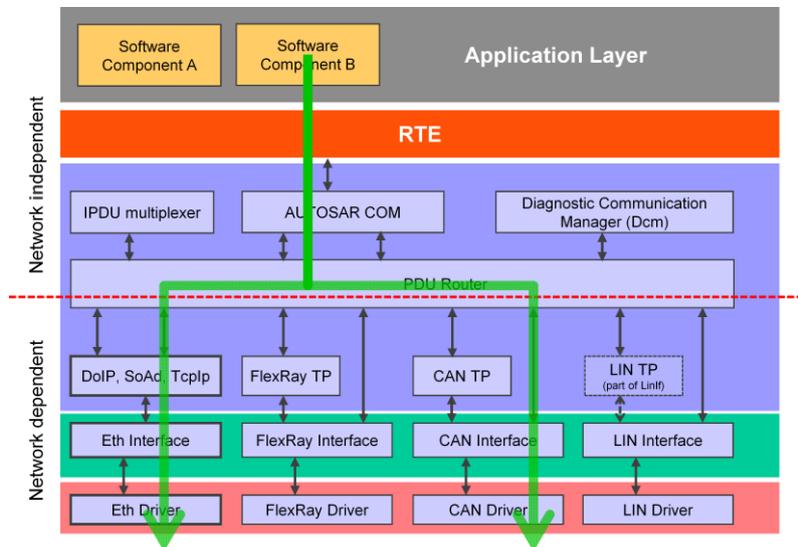


Abbildung 1: AUTOSAR Kommunikationsstack

Auf dem Ethernet Treiber (**Eth**), der vom Ethernet-Kontroller abstrahiert, setzt das Ethernet-Interface (**EthIf**) auf. Das **EthIf** abstrahiert von der konkreten MCU-Anbindung des Ethernet-Kontrollers (zum Beispiel memory-mapped versus SPI) und unterstützt virtuelle LANs (VLANs) durch die Bereitstellung virtueller Controller und die Nutzung von VLAN-Tags gemäß IEEE 802.1Q.

Die Funktion der IP-basierten Kommunikation sowie der Anbindung an den AUTOSAR PDU Router (**PduR**) werden durch die Module **TcpIp**, **DoIP** und **SoAd** bereitgestellt. Aufgrund der netzwerkunabhängigen PDU-basierten Datenschnittstelle des **PduR** ist es für Module oberhalb des **PduR** oder Softwarekomponenten (zum Beispiel die Softwarekomponente B in Abbildung 1) unerheblich und ausschließlich eine Frage der Konfiguration, ob die Kommunikation nun über CAN oder über Ethernet erfolgt.

## Architektur des AUTOSAR IP-Stacks

Eine detailliertere Darstellung der inneren Struktur des AUTOSAR IP-Stacks sowie der Interaktion zwischen den Modulen liefert Abbildung 2. Zur besseren Unterscheidung ist hier der Datenpfad schwarz und der Kontrollpfad orange dargestellt.

Aufbauend auf der **EthIf**-Schnittstelle ist das **TcpIp** Modul für die Umsetzung der TCP/IP-Protokolle und der lokalen IP-Adressverwaltung verantwortlich.

Der Socket Adaptor (**SoAd**) stellt eine PDU-basierte Datenschnittstelle sowie eine Kontrollschnittstelle bereit und abstrahiert von TCP/IP-Spezifika wie IP-Adressen und Ports, sowie von Details des Verbindungsauf- und -abbaus. Für die Übertragung und den Empfang von PDUs erfolgt im **SoAd** eine Umsetzung dieser PDUs (identifiziert durch eine eindeutige PDU-ID) in Socket Connections und vice versa. Socket Connections definieren die Endpunkte von TCP-Verbindungen beziehungsweise Sender und Empfänger von UDP-Datagrammen (identifiziert durch ein 4-Tupel aus lokaler/entfernter IP-Adresse und Portnummer) sowie zugehörige Kommunikationsparameter.

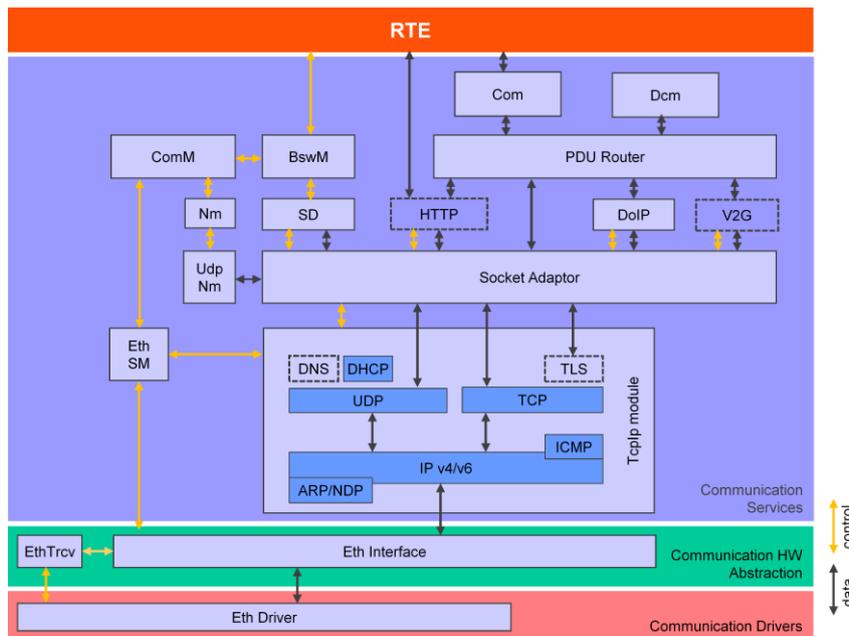


Abbildung 2: Architektur des AUTOSAR IP-Stack

Für die Diagnosekommunikation zwischen externem Diagnosegerät und Steuergerät entsprechend des ISO Standards 13400 dient das Diagnostics over IP (**DoIP**) Modul.

Der Ethernet State Manager (**EthSM**) ist für das Anlaufen und das Abschalten einer IP/Ethernet-Netzanbindung verantwortlich, und das UDP Network Management (**UdpNm**) Modul dient zur Koordinierung der Steuergeräteabschaltung eines Ethernet-Netzes.

Das Service Discovery (**Sd**) Modul verwaltet die Verfügbarkeit von Diensten und Informationsquellen.

## Optimierte/skalierbare Ressourcenverwaltung

Einen wesentlicher Punkt für die AUTOSAR Basissoftware stellt die optimierte und skalierbare Ressourcenverwaltung dar, deren Umsetzung im AUTOSAR IP-Stack sich anhand der folgenden drei Beispiele exemplarisch zeigt.

- (1) **Socket Connection Sharing:** Um ein gutes Verhältnis zwischen den Längen von Protokollheadern und Daten zu erzielen und um den Overhead für den Verbindungsauf- und -abbau zu reduzieren, können mehrere unterschiedliche AUTOSAR PDUs eine Socket-Connection teilen und darüber hinaus auch gemeinsam in einem UDP-Datagramm beziehungsweise TCP-Segment verschickt werden. Hierzu führt der **SoAd** anhand eines zusätzlichen PDU-Headers ein Multiplexing auf Senderseite und ein Demultiplexing auf Empfängerseite durch.
- (2) **Optimierter Datenpfad:** Die Minimierung der Anzahl an Zwischenpuffern und Kopieroperationen am Datenpfad durch den Stack ist eine wesentliche Anforderung für Steuergeräte, da hiervon der RAM Bedarf, die CPU-Last und die Nachrichtenlatenz abhängt. Bezüglich der Schnittstellenauslegung zur Optimierung des Datenpfads galt es zumindest folgende drei Ausprägungen des Datenpfads zu berücksichtigen: die Datenbereitstellung über IF- oder TP-API, die Verwendung des PDU Headers (Socket Connection Sharing) und das verwendete Transportprotokoll (TCP oder UDP). Mit der aktualisierten Schnittstelle wird beispielsweise für den häufigsten Datenpfad „IF/PDU-Header/UDP“ kein zusätzlicher Datenpuffer und nur eine einzige Kopieroperation im AUTOSAR IP-Stack vom beziehungsweise zum Ethernet Treiber benötigt.

- (3) Flexible Allokation von Sendepuffern:** Um flexibel auf dynamische Änderungen des Pufferbedarfs mehrerer TCP Verbindungen eingehen zu können, ist eine dynamische Allokation aus einem geteilten Pufferpool vorteilhaft. Eine solche Allokationsstrategie birgt jedoch die Gefahr, dass eine Verbindung sämtliche Puffer des Pools monopolisiert und somit alle anderen Verbindungen blockiert. Um dies zu verhindern kann über Quotas die maximale Anzahl an allozierbaren Puffern pro Verbindung beschränkt werden.

## Anwendungsempfehlungen

Während die aktuell genutzte signalbasierte Kommunikation, das heißt der Broadcast von dicht gepackten Nachrichten, für CAN, LIN und FlexRay gut funktioniert, ist sie nicht optimal für IP/Ethernet.

Bei einem Switched-Ethernet wird bei einer 1:1 Kommunikation (Unicast) nur Bandbreite auf den Verbindungen zwischen Sender und Empfänger benötigt. Dieser Vorteil geht jedoch bei der Verwendung von 1:n Kommunikation (Broadcast oder Multicast) verloren. Des Weiteren erhöht die intensive Verwendung von Multicasts die CPU-Last, da die hardwarebasierten Multicast-Adressfilter von Ethernet-Kontrollern üblicherweise wesentlich schlechter sind als die Adressfilter bei CAN- oder FlexRay-Kontrollern und eine nachgelagerte softwarebasierte Adressfilterung bei IP/Ethernet schon deshalb benötigt wird, weil der IP-Multicastadressraum (28 bit) nicht eindeutig auf den MAC-Multicastadressraum (23 bit) abgebildet werden kann.

Die Effizienz, das heißt das Verhältnis von Nutzdatenlänge zur Länge des gesamten Ethernet Frames inklusive Preamble und Inter-Frame Gap, ist bei IP/Ethernet schlecht für kurze Nachrichten. So liegt die Effizienz von IP/Ethernet für 8 Byte PDUs bei unter 10% während sie bei 200 Byte PDUs bereits auf über 75% steigt. Im Vergleich dazu liegt die Effizienz bei CAN für 8 Byte PDUs bei circa 58% und sinkt für längere PDUs, da dann zur Übertragung das CAN Transport Protokoll benötigt wird und hierdurch zusätzliche Protokoll-Overheads entstehen.

Um das volle Potential von IP/Ethernet auszunutzen, sollten folgende Anwendungsempfehlungen berücksichtigt werden:

- Unicasts mit langen Nachrichten, welche die Daten in unkomprimierter, natürlicher Form enthalten anstatt Multicasts von kurzen dicht gepackten Nachrichten. Auf diese Weise wird einerseits eine gute Effizienz erzielt und andererseits die CPU-Last reduziert, weil die Rechenzeit für das Ver- und Entpacken sowie die softwarebasierte Adressfilterung entfällt.
- Explizite Bekanntgabe der Verfügbarkeit von entfernten Diensten und Informationsquellen mittels Service Discovery statt implizite Schlussfolgerung über deren Verfügbarkeit basierend auf empfangener Information (zum Beispiel Ungültigkeitswerte von Signalen)
- Explizite Aktivierung von entfernten Diensten mittels Remote Procedure Calls (RPCs) statt implizite Schlussfolgerung basierend auf empfangener Information (zum Beispiel spezieller Signalwert)
- Explizite Aufforderung zur Informationsübertragung durch konsequente Umsetzung des Publish/Subscribe Kommunikationsmodells statt unbedingter (periodischer) Übertragung von Information
- Aggregation von mehreren PDUs in einem Ethernet-Frame mittels konsequenter Nutzung des Socket-Connection-Sharing

## Zusammenfassung

Die im Rahmen von AUTOSAR 4.1 in enger Zusammenarbeit zwischen BMW und EB stattgefundene Überarbeitung der AUTOSAR IP-Stack Spezifikationen wurde mit Erreichen des AUTOSAR-

Meilensteins MS3 „Technical Readiness“ mit Ende Juli 2012 erfolgreich finalisiert. Damit ist das Ziel erreicht, den Einsatz der bekannten TCP/IP Protokollsuite mit einem Switched-Ethernet Netzwerk für die fahrzeuginterne Kommunikation und den direkten Diagnosezugang von außen für den Serieneinsatz bereit zu machen.

Die Validierung der Spezifikationen erfolgte durch einen gemeinsam von BMW und EB entwickelten Demonstrator, der die fahrzeuginterne Kommunikation und den direkten Flash-Download von einem externen Tester sowie die Trennung der internen und externen Kommunikation über VLANs zeigt. Als AUTOSAR-Stack wurde hierbei der EB tresos AutoCore 6 eingesetzt. Der darin enthaltene AUTOSAR 4.1 IP-Stack von EB wurde parallel zum Spezifikationsupdate entwickelt und basiert im Kern auf dem bereits 2008 in der Serie eingesetzten TCP/IP-Stack von EB.

#### Autoren:



Dr. Michael Ziehensack ist bei Elektrobit als Senior Manager für die AUTOSAR Kommunikationssoftware verantwortlich.



Dr. Thomas M. Galla ist bei Elektrobit Senior Software Architekt für die gesamte AUTOSAR Software.