

Besonderheiten und Nutzen der Implementierung modularisierter Software

# Embedded Software für FlexRay-Systeme

Standardisierte Softwarekomponenten helfen die zunehmende Komplexität im Zusammenspiel aller Softwareteile eines Steuergeräts zu beherrschen. Auf dem FlexRay-Symposium der Vector Informatik im März dieses Jahres zeigten die Referenten Wege auf, wie eine moderne Steuergeräte-Software für FlexRay zu entwickeln ist.

Mit dem Busprotokoll FlexRay geht gerade ein Fahrzeugnetzwerk an den Start, das eine große Übertragungsbandbreite für schnelle Regelsysteme zur Verfügung stellt. Es ermöglicht eine 20-fach höhere Bandbreite als CAN und reduziert die Komplexität durch weniger Gateways. Durch seine Zeitsteuerung eignet es sich besonders als Kommunikationssystem für verteilte, fehlertolerante Systeme und sicherheitsrelevante Anwendungen. Gleichzeitig hofft man durch die Standardisierung der Software-Systemarchitektur mit AUTOSAR, die steigende Komplexität der Fahrzeugelektronik eindämmen zu können.

Wie muss eine FlexRay-Steuergeräte-Software aussehen?

Um die Vorteile der auf FlexRay basierenden Kommunikation zu nutzen, ist es sinnvoll die zugehörige Basissoftware grundsätzlich entsprechend der AUTOSAR-Spezifikation zu entwickeln. Mit AUTOSAR wurde eine neue Entwicklungsmethodik, Softwarearchitektur und Basissoftware spezifiziert. Diese wird sukzessive von den OEM für neue Fahrzeugbaureihen eingesetzt. Die standardkonforme Steuergeräte-Software ist modular aufgebaut (Bild 1) und ermöglicht dadurch einerseits die Trennung in die Softwarekomponenten

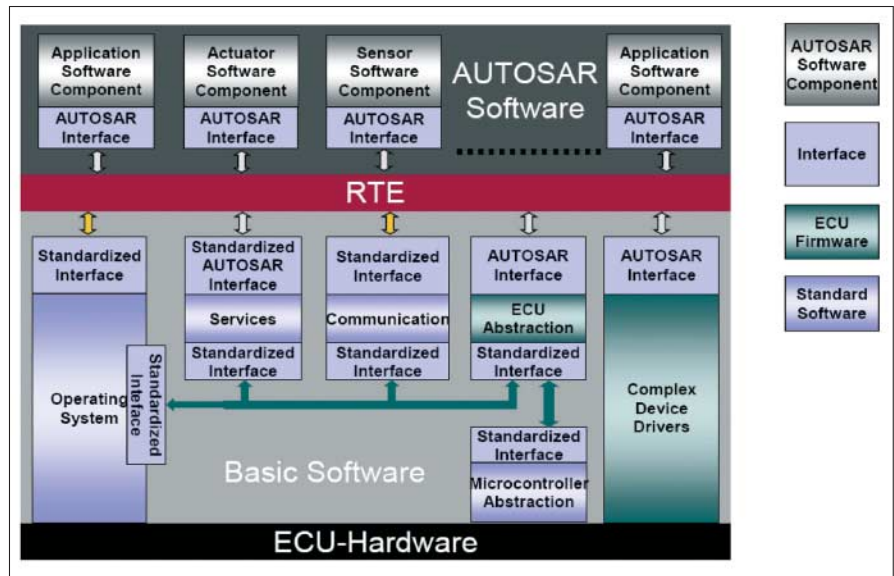


Bild 1: AUTOSAR-Schichtenmodell der Steuergerätesoftware mit modularem Aufbau

(Quelle: AUTOSAR GbR)

oberhalb der RTE und die Basissoftware unterhalb der RTE. Andererseits ist auch die Basissoftware in sich modularisiert und durch klar definierte Schnittstellen spezifiziert, so dass bei der Integration auch Software beliebiger Herkunft eingesetzt werden kann. Dazu definiert der Standard, welche Austauschformate verwendet werden können und wie die Schnittstellen zwischen einzelnen Modulen zu arbeiten haben.

Die Modularität vereinfacht es, den Softwareumfang jeweils für die Anforderungen einer Fahrzeugvariante oder -generation zu skalieren, z. B. Steuergeräte ohne Netzwerkmanagement. Sowohl für OEMs als auch für die Steuergerätezulieferer sinkt der Entwicklungsaufwand für die Basissoftware, da die einzelnen Softwaremodule sogar vom Softwarezulieferer komplett vorkonfiguriert geliefert werden können. Somit kann der Fokus bei der Entwicklung deutlich stärker als bisher auf Innovationen und der eigentlichen Funktionsentwicklung liegen.

Embedded Architektur für FlexRay

Der schematische Aufbau einer FlexRay-Basissoftware der Vector Informatik ist in Bild 2 dargestellt. Im FlexRay-Stack sind die FlexRay-spezifischen Komponenten wie Interface, Treiber, Netzwerkmanagement (NM) oder das Transportprotokoll (TP) zusammengefasst. Der Treiber abstrahiert die Hardware und ermöglicht es, verschiedene Kommunikations-Controller (CC) zu bedienen. Er initialisiert den Controller, sendet und empfängt Frames und erkennt Controller-Fehler. Das Interface kommuniziert mit darüber liegenden Schichten, verarbeitet PDUs (Protocol Data Units) zu Frames – und umgekehrt. Außerdem erteilt es Sende- und Empfangsbestätigungen an die betroffenen Schichten. Das Netzwerkmanagement übernimmt die Koordination aller Steuergeräte im Cluster im Hinblick auf den Kommunikationsbedarf für das Bussystem. Liegt von keinem Busteilnehmer mehr Kommunikationsbedarf vor, wird der synchrone Übergang in den Bus-Sleep-Steuergerätemodus eingeleitet.

**AUTOR**

*Dipl.-Ing. Dirk Großmann* ist für die Entwicklung der FlexRay-Embedded-Softwarekomponenten und *Dipl.-Ing. Oliver Kitt* für die Entwicklung von Mess- und Kalibrierprotokollen in CANape bei Vector Informatik in Stuttgart verantwortlich.

Auch das FlexRay-Transportprotokoll setzt auf dem FlexRay-Interface auf. Es übernimmt die Aufgabe, große Datenpakete, die nicht in einer einzigen PDU verschickt werden können, zu zerteilen und auf der Gegenseite wieder zusammen zu setzen. Wie sich die modulare Anpassung in der Praxis auswirkt, zeigt das Beispiel zweier FlexRay-Treiber für einen FlexRay-Controller von Freescale und von NEC. Der Treiber ist an die jeweils verwendete Hardware optimal angepasst und nutzt deren vorhandene Features. Trotzdem stellt er der darüber liegenden Schicht eine unveränderte „Sicht“- und Verhaltensweise bereit. Im Falle des 16-Bit-Controllers S12X von Freescale muss der FlexRay-Treiber das Buffer-RAM verwalten, da in diesem Fall der benötigte Speicherplatz in das System-RAM ausgelagert werden muss. Beim 32-Bit-Controller NEC V850 ist ein großes RAM für die Puffer bereits im FlexRay-Controller enthalten. Hier übernimmt der Treiber dessen effiziente Partitionierung und Nutzung.

## Steuergeräte-Kalibrierung mit XCP on FlexRay

Auf einfache Weise lassen sich auch später entwickelte Komponenten, die beispielsweise durch neue Protokolle oder erweiterte Standards notwendig werden, in die Architektur integrieren. Sie müssen dazu an ihren Schnittstellen dem AUTOSAR-Standard entsprechen. So wird z. B. dem zuvor beschriebenen FlexRay-Stack XCP-Funktionalität hinzugefügt, um interne Signale der FlexRay-Steuergeräte messen und kalibrieren zu können.

XCP ist ein universelles Kommunikationsprotokoll zur Optimierung der Systemparameter eines Steuergeräts. Wegen der Trennung von Protokollschicht und Transportschicht kann XCP in verschiedenen Kommunikationsnetzen betrieben werden (XCP on CAN, FlexRay, Ethernet, USB, RS232 oder SPI/SCI). Die klare Tren-

nung der Schichten spiegelt sich auch in der Integration im FlexRay-Stack wider. Die universelle Protokollschicht XCP sitzt über dem FlexRay-spezifischen Transport Layer (FrXCP), der wiederum den Signalaustausch mit dem FlexRay-Interface ermöglicht (Bild 3).

Aufgrund der dynamischen Bandbreitenzuweisung muss der Treiber mit der Pufferkonfiguration während der Messung oder Kalibrierung eine zusätzliche Aufgabe bewältigen. Deshalb wird dieses Modul durch eine erweiterte Version des Standard-AUTOSAR-Treibers ersetzt.

XCP ist ein adressorientiertes Protokoll. Die Kommunikation findet zwischen einer Steuergerätekomponente und einer ähnlich ausgelegten Softwarekomponen-

te im XCP-Master statt. In der Regel ist der XCP-Master ein Mess- und Kalibriertool wie bspw. CANape von Vector Informatik. Er greift adressorientiert auf Messdaten zu, die Task-synchron (event driven) in den Steuergeräten erfasst werden.

## Dynamische Bandbreitenverwaltung

Der Verbindungsaufbau erfolgt über einen initialen Kommunikationskanal, der in der Steuergeräte-Beschreibungsdatei (A2L) festgelegt ist. Über Transport-Layer-Kommandos steuert der XCP-Master die Zuteilung freier Slots des dynamischen Teils eines Zyklus und ermöglicht so eine Erweiterung des Kanals für die Übertragung der Mess- und Kalibrierdaten. Diese „Last-

verteilung“ geschieht dynamisch zur Laufzeit und sorgt für eine optimale Bandbreitenausnutzung. Da mehrere Steuergeräte über den gleichen Bus kommunizieren, kann ein Slot nicht nur exklusiv belegt werden, sondern steht durch so genanntes Slot-Multiplexing auch mehreren Steuergeräten zur Verfügung. Dies ist sinnvoll, wenn weniger Bandbreite für eine Messung benötigt wird, ein Steuergerät also z. B. nur in jedem 2. oder 4. Zyklus eine Nachricht senden muss. Dafür erhält jede Nachricht eine eindeutige Adresse (LPDU-Id, Data Link Layer Protocol Data Unit Identifier), die Slot, Zyklus und Kanal präzisiert. So kann der gleiche Slot in jedem Sendezyklus durch ein anderes Steuergerät mit Daten beaufschlagt werden. Die Messdaten sind außerdem mit einem Zeitstempel versehen. Auf diese Weise ist es möglich, Messwerte in kürzeren Abständen als der Zykluszeit abzufragen. Beispielsweise kann eine Messung alle 2,5 ms erfolgen auch wenn

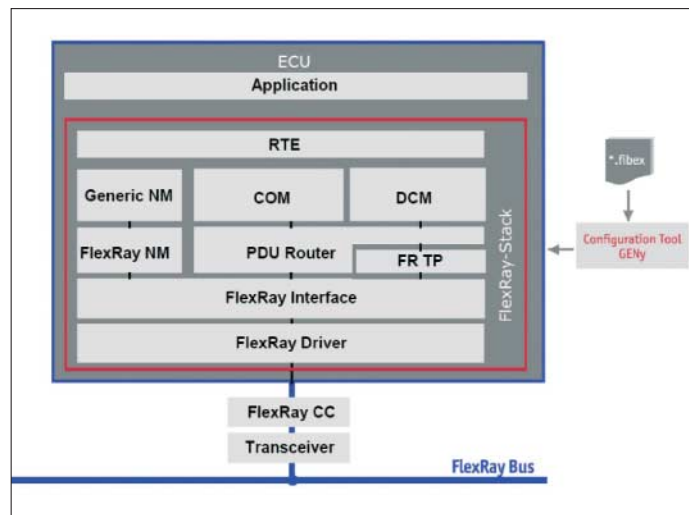


Bild 2: Schematischen Aufbau einer FlexRay-Basissoftware von Vector Informatik (Quelle der Bilder 2 bis 4: Vector Informatik GmbH)

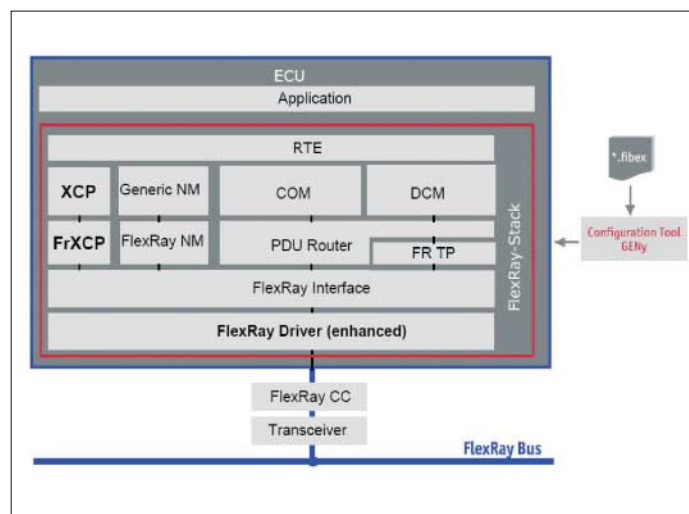


Bild 3: Integration von XCP im FlexRay-Stack mit klarer Trennung der Protokollschicht (XCP) und Transportschicht (FrXCP).

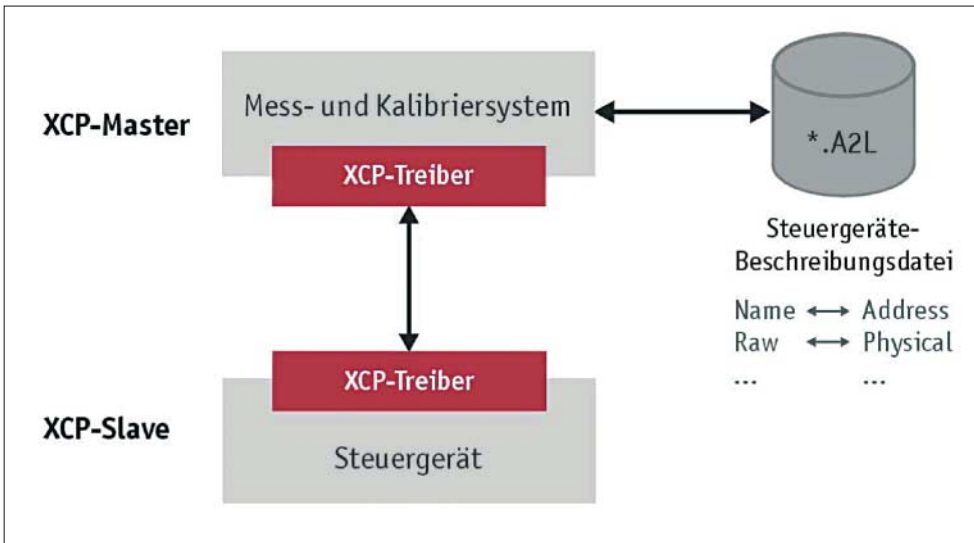


Bild 4: XCP ist als Master/Slave-Struktur realisiert. Der XCP-Slave befindet sich dabei im Steuergerät, der XCP-Master im Mess- und Kalibriertool.

ein System mit 5-ms-Zyklus vorliegt. Es muss lediglich gewährleistet sein, dass die Bandbreite ausreicht, um die Daten auch innerhalb einer geforderten Zeit überhaupt übertragen zu können.

Das Steuergerät muss für diese Kommunikation Sende- und Empfangspuffer in Form von RAM reservieren und dynamisch konfigurieren. Entweder ist das RAM im Controller enthalten oder es wird externer Speicher benötigt, der über den FlexRay-Treiber verwaltet wird.

Wie viele Slots für XCP zur Verfügung stehen und wie die Slotverteilung auszusehen hat, muss rechtzeitig – nämlich schon während der Systemdefinition – festgelegt werden. Dazu werden in der FIBEX-Datei, neben allen anderen Systemdaten, auch die für XCP reservierten Slots definiert. Hier sind verschiedene Szenarios möglich:

- ▶ jedes Steuergerät belegt einen eigenen Slot
- ▶ mehrere Steuergeräte verwenden den gleichen Slot. Dafür hat jedes Steuergerät eine eigene Adresse (NAX, Node Adress for XCP). Über Kommandos wird jeweils der Puffer aktiviert oder deaktiviert
- ▶ Puffer sind konfigurierbar, das entspricht der dynamischen Zuordnung

Losgelöst vom verwendeten Transportprotokoll sollte die Oberfläche eines Tools (XCP-Master) die eigentliche Mess- und Kalibrieraufgabe immer identisch darstellen. Der Applikationsingenieur kann so die Parameter des Steuergerätes optimal abstimmen, unabhängig davon, auf welchem Bussystem er kalibriert.

Damit die Parameter und Messgrößen im Steuergerät gefunden werden, ohne dass man die kryptische Adresse der jeweiligen Variablen wissen muss, arbeitet der Applikateur mit symbolischen Namen. Die Verknüpfung des Namens mit der physikalischen Adresse stellt im Hintergrund der Anwendung die Steuergeräte-Beschreibungsdatei her (Bild 4).

### Durchgängige Unterstützung in allen Entwicklungsphasen

Von der Systembeschreibung über die Implementierung der Basissoftware bis hin zur Kalibrierung der Steuergeräte erhält der FlexRay-Entwickler eine durchgängige Unterstützung aus dem Hause Vector Informatik. Dazu gehören auch Werkzeuge für Design, Entwicklung, Simulation, Analyse, Test von Steuergeräten und verteilten Netzwerken sowie die passenden Businterfaces. Als erstes Mess-, Kalibrier- und Diagnose-Tool unterstützt CANape XCP on FlexRay. Mit gebrauchsfertigen Software-Stacks für Steuergeräte und der XCP on FlexRay-Unterstützung auf Seiten des Kalibriertools erhält der Anwender perfekt aufeinander abgestimmte Komponenten, die durch AUTOSAR-Konformität universell aufgebaut sind und damit flexibel eingesetzt werden können. (jj)

 **infoDIRECT** 500eio607  
[www.elektronik-industrie.de](http://www.elektronik-industrie.de)  
 ▶ Link zu Vector Informatik