Ansätze zur Integration von energieeffizienten Intelligenten Kommunikationskontrollern für FlexRay in Autosar

Christoph Schmutzler[1], Andreas Krüger[1], Martin Simons[1] and Jürgen Becker[2]

[1]Daimler AG
HPC: 050/G007, 71059 Sindelfingen
Email: {christoph.schmutzler, andreas.k.krueger, martin.simons}@daimler.com

[2]Institute for Information Processing Technology (ITIV)
Karlsruhe Institute of Technology (KIT)
Email: becker@kit.edu

Abstract: Die Energieeffizienz elektrischer Verbraucher im Fahrzeug wird derzeit mit größter Aufmerksamkeit untersucht und optimiert. In diesem Beitrag wird ein Intelligenter Kommunikationskontroller (ICC) vorgestellt, der bei inaktivem Steuergerät anhand der Buskommunikation - und damit auf Basis des Fahrzeugzustandes - entscheiden kann, wann das Steuergerät wieder benötigt wird. Zusätzlich können ICCs vordefinierte Statusnachrichten senden. Durch den ICC ist im Idealfall der Schlafzustand eines Steuergerätes nicht anhand der Buskommunikation erkennbar. Um die ICC-Erweiterungen auf breiter Basis einsetzen zu können, wird die Standardisierung im Betriebssystemstandard Autosar angestrebt. Diese Arbeit beschreibt Ansätze und notwendige Änderungen für die Einbindung, Konfiguration und Verwaltung von ICCs, sowie die Abbildung des durch den ICC ermöglichten neuen Kommunikationszustandes in der Autosar Basissoftware. Dabei beschreiben wir insbesondere, welche existierenden Mechanismen und Konfigurationsparameter wir verwenden können und welche Erweiterungen notwendig sind, um das ICC-Konzept in Autosar zu integrieren.

1 Einführung

Die Anzahl und der Funktionsumfang von Steuergeräten in Fahrzeugen ist über die vergangenen Fahrzeuggenerationen hinweg stetig gewachsen. Dabei stand primär der Energieverbrauch bei abgestelltem Motor im Fokus von Optimierungen und Beschränkungen: durch die Einhaltung von sehr niedrigen Ruhestromanforderungen von Steuergeräten ($< 500\ \mu$A) ist die Startfähigkeit eines Fahrzeuges auch nach längeren Standzeiten gewährleistet. Während der Fahrt, d.h. bei laufenden Motor und Generator, galt Energie lange Zeit als frei verfügbar. Durch die steigende Anzahl von Steuergeräten wuchs jedoch auch die elektrische Grundlast der Fahrzeuge. Da der Energieverbrauch eines Fahrzeuges einen direkten Einfluss auf Kraftstoffverbrauch und damit auch Abgasemissionen hat [RB96, SKSS11], sowie insbesondere bei elektrifizierten Antriebssträngen die Reichweite des Fahrzeuges reduziert, gewann die Optimierung der Leistungsaufnahme bei aktivem Fahrzeug in den letzten Jahren immer stärkere an Bedeutung [WLMS10].

Die gestiegene Bedeutung der Energieeffizienz spiegelt sich auch bei der Entwicklung von neuen *Elektrik/Elektronik-Architekturen* (E/E-Architekturen) wieder [Hud09]. Alter-

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

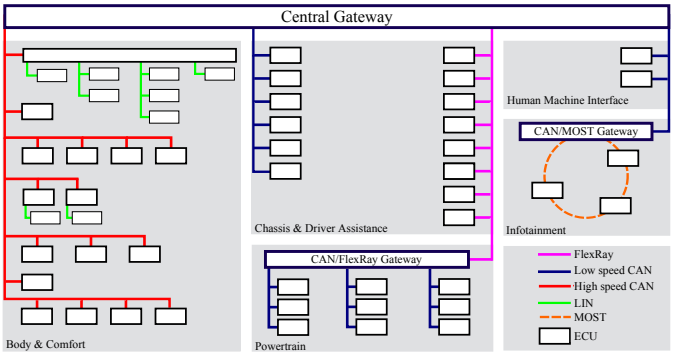


Abbildung 1: Nach funktionalen Domänen aufgeteilte E/E-Architektur.

nativ zu aufwendigen lokalen Optimierungen jedes einzelnen Steuergerätes schlagen wir hier eine Deaktivierung von temporär nicht benötigten Steuergeräten vor. Die verbreiteten Automotive-Busse CAN und FlexRay bieten dazu jedoch bei aktivem Fahrzeug keine Unterstützung. Daher hat sich zur Deaktivierung von Steuergeräten der Einsatz von dedizierten Weckleitungen oder Klemmen durchgesetzt. Beide Ansätze erlauben aber nur eine unflexible Adaption an die bedarfsorientierte Abschaltung von Steuergeräten und erhöhen aufgrund der zusätzlich benötigten Verkabelung Fahrzeuggewicht und Produktionskosten.

Aus diesem Grund arbeiten wir an dem Konzept eines *Intelligenten Kommunikationskontrollers* (ICC) für FlexRay, der es erlaubt, temporär nicht benötigte Steuergeräte in einen energiesparenden Zustand zu versetzten. Auch bei inaktivem Steuergerät erkennt der ICC anhand der Buskommunikation, wann das Steuergerät wieder benötigt wird. Zusätzlich kann der ICC weiterhin vordefinierte Statusnachrichten senden. Durch den ICC ist damit im Idealfall der Schlafzustand eines Steuergerätes nicht von anderen Knoten erkennbar. Um die ICC-Erweiterungen auf breiter Basis einsetzen zu können, wird eine Standardisierung im Embedded-Betriebssystemstandard *Autosar* angestrebt. In dieser Arbeit diskutieren wir auftretende Fragestellungen und stellen Lösungsansätze zur Integration des ICC-Konzeptes in Autosar vor. Nach einer kurzen Einführung in die Grundidee einer E/E-Architektur und in die Motivation und Entstehung von Autosar, beschreiben wir in Kapitel 2 die Herausforderungen, sowie bestehende und neuartige Ansätze zur Energieoptimierung von E/E-Architekturen. Hier wird auch das ICC-Konzept erläutert. In Kapitel 3 werden notwendige Erweiterungen der Autosar Systembeschreibung erklärt und mögliche Ansätze zur Konfiguration und Verwaltung von ICCs in der standardisierten Basissoftware behandelt. Dabei gehen wir auch auf Anknüpfungspunkte an die Autosar-Entwurfsmethodologie ein. In Abschnitt 4 werden mehrere Varianten zur Abbildung des durch den ICC ermöglichten neuen Kommunikationszustandes beschrieben. In Kapitel 5 schließen wir mit den nächsten geplanten Schritten zur weiteren Konzeptabsicherung.

1.1 Elektrik/Elektronik-Architekturen und Automotive-Bustechnologien

Die E/E-Architektur eines Fahrzeuges beschreibt die logische und physikalische Aufteilung von Steuergeräten, sowie von vernetzten Sensoren und Aktoren in einem Fahr-

zeug. Auf logischer Ebene werden die Kommunikationsbeziehungen zwischen funktionalen Komponenten spezifiziert. Die physikalische Beschreibung gibt den Verbauort von Steuergeräten, Verlauf des Leitungssatzes und Anbindung an die Spannungsversorgung wieder. Diese beiden Sichten werden durch eine Abbildung der funktionalen Komponenten auf Steuergeräte und der Kommunikationsbeziehungen auf Fahrzeugbusse miteinander verknüpft. Abbildung 1 zeigt eine nach funktionalen Domänen unterteilte E/E-Architektur.

Zur Fahrzeugvernetzung werden heute vor allem CAN, LIN, MOST und FlexRay eingesetzt [ZS08]. Da MOST hauptsächlich für den Infotainment-Bereich konzipiert wurde, unterscheiden sich der Protokollstack und die Bitübertragungsschicht deutlich von CAN und FlexRay. Daher ist der hier vorgestellte Mechanismus zur selektiven Abschaltung von Steuergeräten nicht für MOST geeignet. CAN wird in den meisten andern anderen Domänen verwendet. In sicherheits- und bandbreitenintensiven Bereichen ersetzt FlexRay aber zunehmend den CAN-Bus. LIN wird als low-cost Bus mit niedriger Bandbreite zur Verbindung von intelligenten Sensoren oder Aktoren, wie z. B. Stellmotoren oder Tastenfeldern, sowie zur Verbindung von Slave-Steuergeräten mit sehr begrenzten Funktionsumfang an ein CAN- oder FlexRay-Steuergerät verwendet. Da LIN damit primär als Peripheriebus eingesetzt wird, ist er ebenfalls nicht Bestandteil der vorgestellten Untersuchungen und wir beschränken uns bei der Abschaltung von Steuergeräten nur auf CAN- und FlexRay.

1.2 Autosar

Innovationen in der Automobiltechnik werden seit den 90 Jahren zunehmend durch E/E-Anteile ermöglicht [WR06]. Während anfangs eine solitäre Entwicklung einzelner Steuergeräte ausreichend war, gewann durch die zunehmend komplexer werdenden E/E-Architekturen eine strukturierte Entwicklung auf Systemebene, sowie die Vereinheitlichung von Softwaremechanismen und -infrastrukturen zunehmend an Bedeutung. In Tabelle 1 ist die Anzahl der durchschnittlich verbauten Steuergeräte, die Anzahl der Kommunikationssignale, sowie die Anzahl der Fahrzeugbusse für verschiedene Generationen der Mercedes-Benz E-Klasse angegeben. Ausgehend von nur 2 Steuergeräten werden heutzutage in einem Fahrzeug bis zu 70 Steuergeräte verbaut. Diese Zahl nimmt nochmals deutlich zu, wenn man auch vernetzte Sensoren und Aktoren dazu zählt. Die Netzwerkkommunikation erfolgt über tausende Signale, die in mehreren hundert Botschaften zusammengefasst und übertragen werden. Getrieben durch diese große Komplexität wurde 2003 von Automobilherstellern und Zulieferern das *AUTomotive Open System ARchitecture*-Konsortium (Autosar) gegründet. Eine gute Übersicht über die Historie, sowie den gesetzten und erreichten Zielen geben [H$^+$04, H$^+$06, F$^+$09]. Ausführliche Beschreibungen über Autosar selbst kann man direkt den Autosar-Spezifikationen entnehmen [AUT03].

Eines der Hauptziele des Autosar-Konsortiums ist die Standardisierung von Softwareschnittstellen, sowie die Erarbeitung von Entwicklungsmethodologien, die eine komponentenorientierte, von der genauen Implementierung der Hardware und Buskommunikation unabhängige Entwicklung ermöglichen. Teilkomponenten einer Fahrzeugfunktion sind in der Applikationsschicht in *Software Components* (SW-Cs) gekapselt. Die Entwicklung der SW-Cs erfolgt losgelöst von der verwendeten Hardware, die von den *Basic Software Modules* (BSW) abstrahiert wird. Die BSW-Module beinhalten zudem eine Reihe von standardisierten Diensten, z. B. zur Diagnose oder zur Kommunikation. Die *Runtime En-*

Tabelle 1: Zuwachs an Komplexität von E/E-Architekturen am Beispiel der Mercedes-Benz E-Klasse

Baureihe	Baujahr (Limousine)	Anzahl Fahrzeugbusse (CAN. MOST, FlexRay)	Steuergeräte	Kommunikations-signale
W124	1984 - 1995	1	7	unter 100
W210	1995 - 2002	3	30	ca. 200
W211	2002 - 2009	5	52	ca. 4100
W212	seit 2009	9	ca. 70	ca. 6000

vironment (RTE) dient als standardisierte Schnittstelle zwischen BSW und SW-Cs. Die Kommunikation zwischen SW-Cs, sowie zwischen SW-Cs und Diensten der BSW erfolgt über gerichtete Punkt-zu-Punkt Verbindungen (*Ports*). Ports werden logisch gesehen über einen abstrakten *Virtual Functional Bus* (VFB) miteinander verbunden. Damit ist es für SW-Cs unerheblich, ob sich der Ziel-Port auf dem gleichen Steuergerät befindet, oder die Datenübertragung über einen Kommunikationsbus erfolgt. Das Routing von Signalen zwischen Ports erfolgt durch die RTE. Die Abhängigkeiten zwischen Ports werden anhand einer globalen Systembeschreibung zur Generierzeit des Systems aufgelöst. Die Spezifikation und Entwicklung von SW-Cs, sowie die Abstraktion der Kommunikation bieten interessante Ansätze für die Anpassung bestehender und der Entwicklung neuer Werkzeugketten [VF10], Testmethoden [KBR09] und Adaption von modellbasierten Entwicklungsansätzen [RJV09] auf Autosar. Ansätze zur Portierung bestehender Software, die für eine proprietäre Plattform entwickelt wurde, werden in [KPLJ08] diskutiert.

Die notwendige Modularisierung wird durch die Vereinheitlichung von Schnittstellen, sowie einer vorgegebenen Entwurfsmethodologie für die relevanten Komponenten der E/E-Architektur ermöglicht. Bei der Integration neuer Funktionen und Hardwarekomponenten - wie z. B. eines ICCs - müssen die bestehenden Vorgehensweisen und Systembeschreibung erweitert werden, um ein Autosar-konformes Entwurfsparadigma zu unterstützen. Diese Arbeit beschreibt wichtige Randbedingungen und Eigenschaften die bei der Integration und Vereinheitlichung eines neuen Hardwarekonzeptes beachtet werden müssen.

2 Energieeffiziente Auslegung von E/E-Architekturen

Die Energieoptimierung einer E/E-Architektur kann lokal, Geräte-spezifisch erfolgen (beispielsweise durch Optimierung von Aktoren oder der Leistungsverteilung), oder allgemein das Ziel verfolgen, beliebige, nicht benötigte Knoten komplett abzuschalten. Die lokale Optimierung bietet insbesondere bei Komponenten mit hohem Leistungsbedarf, wie beispielsweise der elektrischen Servolenkung oder der Klimaanlage, ein hohes Einsparpotential. Haben Steuergeräte jedoch keine dominierenden Hauptverbraucher, kann die lokale Optimierung in Relation zum Einsparpotential schnell kosten- und zeitintensiv werden. Hier ist der Ansatz einer Knotenabschaltung lohnenswert, Steuergeräte bei Nichtbedarf in einen energiesparenden Zustand zu versetzen.

Heute wird ein großer Anteil von Fahrzeugsystemen verteilt ausgeführt. Dabei sind die Teilfunktionen eines Systems, wie z. B. der Elektronischen Stabilitätskontrolle, oder dem Lademanagement für Hybridfahrzeuge, auf mehreren Steuergeräten platziert. Andere Systeme, z. B. zur Steuerung der automatisch schließenden Heckklappe führen sämtliche

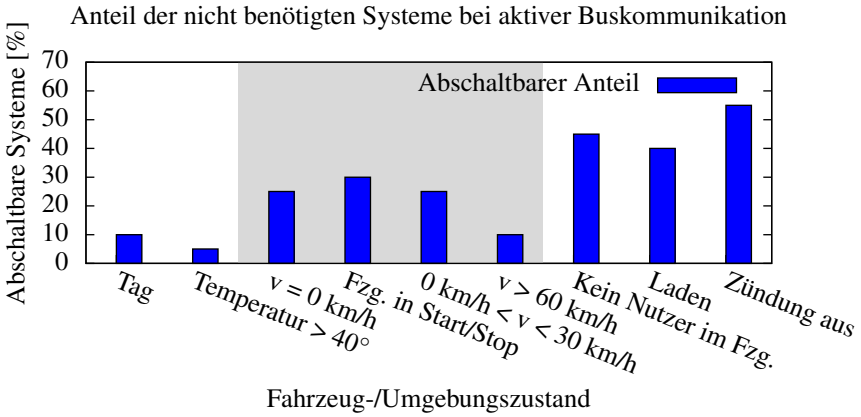


Abbildung 2: Nach Fahrzeug- und Umgebungszustand unterteilter Anteil von abschaltbaren Fahrzeugsystemen bei aktiver Buskommunikation.

Funktionen auf einem Steuergerät aus. Zur Untersuchung von Abschaltszenarien bietet sich eine Klassifizierung von nicht benötigten Systemen nach Fahrzeug- und Umgebungszustand an. Steuergeräte, die nur Funktionen momentan nicht benötigter Systeme ausführen, können deaktiviert werden. In Abbildung 2 ist der vorläufig geschätzte Anteil von abschaltbaren Fahrzeugsystemen die sich im Ruhezustand befinden (z. B. Einparkhilfe über einer Geschwindigkeit von ca. 20 km/h oder eine nicht aktive Sitzheizung) eines aktuellen Fahrzeugs im Premiumsegment dargestellt. Insbesondere im Stand, in gewissen Geschwindigkeitsbereichen, sowie während des Ladens der Hochvoltbatterie eines Hybrid- oder Elektrofahrzeugs können bis zu 55% der Systeme abgeschaltet werden. Die restlichen Klassifizierungsstufen ergeben deutlich weniger nicht benötigte Systeme.

Das für einen Großteil der Fahrzeugdomänen relevante Bussystem CAN, und der an Bedeutung gewinnende FlexRay-Bus, unterstützten bei aktiver Kommunikation jedoch keine selektive Knotenabschaltung, die für eine anforderungsabhängige Deaktivierung von Steuergeräten benötigt wird. Bei CAN wacht ein schlafendes Steuergerät bei jeglicher Busaktivität auf. FlexRay spezifiziert zwar ein WakeUp-Pattern, durch das ein busweites Wecken angefordert wird, der Fokus bei der Spezifikation des Patterns lag jedoch auf einem zuverlässigen Wecken und nicht auf der Vermeidung eines Fehlweckens. Das Pattern wird daher mit hoher Wahrscheinlichkeit auch in regulärer Buskommunikation fehlerkannt, ein schlafendes Steuergerät würde nach kurzer Zeit geweckt werden. Da bei aktivem Fahrzeug überwiegend zyklisch kommuniziert wird, sind bei aktivem Bus momentan also alle Steuergeräte wach [SKSS11].

2.1 Technologien zur bedarfsorientierten Abschaltung von Steuergeräten

Aufgrund der soeben erläuterten Einschränkungen heutiger Bussysteme werden die in [EH10] beschriebenen busunabhängige Mechanismen zur Knotenabschaltung über Weckleitungen und Klemmen verwendet: bei der Abschaltung über *Weckleitungen* steuert ein

Master-Steuergerät über diskrete Leitungen die Spannungsversorgung angeschlossener Steuergeräte und entscheidet, wann diese nicht benötigt werden. Dies erhöht die Komplexität des Systemdesigns, da der Master auch die funktionalen Anforderungen der kontrollierten Steuergeräte kennen muss. Ändern sich die Anforderungen, da z. B. Funktionen eines Steuergerätes verschoben, oder auf einem Steuergerät zusammengefasst werden, muss auch die Master-Komponenten angepasst werden. Insbesondere bei räumlich stark verteilten Knoten erfordert dieser Ansatz zudem zusätzlichen Verkabelungsaufwand.

Mit der Einführung von standardisierten *Klemmen* (DIN 72552), welche in unterschiedlichen Fahrzeugzuständen bestromt sind, können Knoten durch entsprechende Aufteilung an die Leistungsverteilung abgeschaltet werden. So ist die Klemme 15 beispielsweise nur bei aktiver Zündung bestromt; die Klemme 30 ist dauerbestromt. Damit können Steuergeräte, die nur bei aktivem Fahrzeug benötigt werden, durch eine passende Anbindung ohne zusätzlichen Optimierungsaufwand deaktiviert werden. Die Abschaltung über Klemmen ermöglicht jedoch nur eine sehr grobe Einteilung von Fahrzeugzuständen und erzeugt ebenfalls einen erhöhten Verkabelungsaufwand.

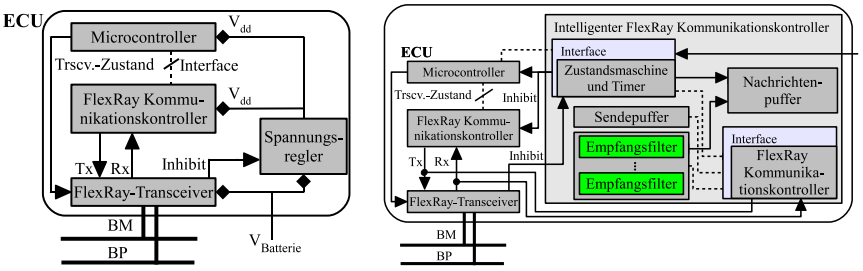
Für CAN wurde in den vergangenen Jahren der *Transceiver-basierte Teilnetzbetrieb* entwickelt [FSG10, EH10]. Eine neue Generation von CAN-Transceivern ist in Vorbereitung, die zusätzlichen zum bisherigen Ruhezustand, der bei jeglicher Busaktivität beendet wird, auch einen Teilnetzbetrieb-Schlafzustand unterstützen. Dabei reagiert der Transceiver nur noch auf eine vorher definierte Wecknachricht. Steuergeräte können damit bei aktivem CAN-Bus schlafen gehen; andere Teilnehmer können bei Bedarf schlafende Knoten durch den Versand der Wecknachricht aufwecken.

Der Transceiver-basierte Teilnetzbetrieb weist zwei Nachteile auf. Zum einen hat er die Anpassung aller funktional abhängigen Knoten zur Konsequenz, auch wenn nur ein Teil der Knoten den neuen Schlafzustand verwendet. Außerdem ist die Erweiterung durch die zusätzliche Filtermöglichkeit des Transceivers nur für CAN und nicht für FlexRay geplant. Dies liegt zum einen an der geringeren Komplexität des CAN-Protokolls, und zum anderen am geringeren Systemtakt des Protokolldecoders: 8 MHz für CAN und 80 MHz für FlexRay. Beides würde wesentlich komplexere FlexRay-Transceiver erfordern, um einen vergleichbaren Mechanismus abzubilden.

2.2 Intelligente Kommunikationskontroller für FlexRay

In Abbildung 3a ist ein stark vereinfachter Aufbau eines FlexRay-Steuergerätes dargestellt. Eine detaillierte Beschreibung des FlexRay-Bussystems, sowie des Funktionsumfangs der benötigten Hardware findet man in [Rau07]. Der FlexRay-Transceiver steuert über seinen *Inhibit*-Pin einen Spannungsregler, über den der Microcontroller (μC) sowie der FlexRay-Kommunikationskontroller (FlexRay-CC) versorgt werden. Bei gestoppter Buskommunikation schaltet der μC den Transceiver über separate Pins in den Schlafmodus. In diesem Zustand wird der Inhibit-Pin hochohmig geschaltet, der Spannungsregler wird deaktiviert. Empfängt der Transceiver ein WakeUp-Pattern, wird der Inhibit-Pin auf V_{dd} gesetzt und der μC und FlexRay-CC werden bestromt - das Steuergerät ist wieder aktiv. Da FlexRay-Transceiver bei Busaktivität das WakeUp-Pattern häufig in regulären Nachrichten fehlerkennen, sind heute bei aktivem Bus alle Steuergeräte aktiv.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html



(a) Abschaltung über Transceiver. (b) Abschaltung über FlexRay-ICC

Abbildung 3: Vereinfachte Strukturdarstellung eines FlexRay-Steuergerätes mit und ohne ICC.

Das Konzept eines *Intelligenten Kommunikationskontrollers* (ICC) für FlexRay kann wie folgt beschrieben werden. Erkennt die auf dem μC ausgeführte Applikation, dass sie aufgrund des aktuellen Fahrzeugzustandes nicht mehr ausgeführt werden muss, konfiguriert sie den ICC, übergibt die Kommunikation an ihn und schaltet den μC in einen energieeffizienten Zustand. Der ICC kann den Busverkehr anhand von vorher definierten Filtern auswerten und Weckbedingungen erkennen. Wird ein Zustand erkannt, in dem die Software wieder benötigt wird, informiert der ICC den μC, und die Kommunikation wird wieder von der Software übernommen.

So können sich beispielsweise Steuergeräte, die geschwindigkeitsabhängige Systeme ausführen (z. B. das Anhängeranschlußsteuergerät), über einer bestimmten Geschwindigkeit abschalten. Der ICC erkennt durch Filterung der entsprechenden Signale, wenn das Fahrzeug unter eine Grenzgeschwindigkeit fällt und aktiviert das Steuergerät wieder. Es ist also keine explizite externe Anforderung anderer Knoten nötig, um ein schlafendes Steuergerät aufzuwecken. Relevante Frames werden im ICC gespeichert und stehen der Software sofort nach der Aktivierung zur Verfügung. Damit muss nicht auf den erneuten Empfang der Nachricht gewartet werden und das Steuergerät kann mit minimaler Verzögerung auf das Weckereignis reagieren, was insbesondere bei Nachrichten mit langen Zykluszeiten (> 200 ms) wichtig ist. Zusätzlich zur Datenfilterung unterstützt der ICC einen Timeout-Mechanismus, der bei Ausbleiben einer Nachricht ein Wecken veranlasst. Damit kann die Fehlerbehandlung wieder in Software erfolgen.

Fast alle Steuergeräte senden im Betrieb zyklisch Statusnachrichten, durch die beispielsweise der Öffnungszustand der Heckklappe, oder das Vorhandensein eines Anhängers an andere Knoten kommuniziert wird. Ein Ausbleiben der Statusnachrichten wird von anderen Teilnehmern als Fehler des Senders gewertet. Daher würden auch nicht gesendete Nachrichten eines schlafenden Teilnehmers zu Fehlereinträgen führen. Um dies zu vermeiden, könnte der Knoten vor seiner Abschaltung den anderen Knoten mitteilen, dass seine Signale aufgrund des Schlafzustandes ausbleiben. Dies würde jedoch Anpassungen in *allen* Empfangsknoten erfordern, was einen erheblichen Aufwand darstellt. Daher ist der ICC zusätzlich in der Lage, weiterhin vorkonfigurierte Statusnachrichten zu senden. Da die Verwendung des ICCs nur geplant ist, wenn sich der Zustand des Steuergerätes nicht ändern kann, ist die Versendung von statischen Nachrichten ausreichend.

Idealerweise ist damit zusammenfassend der durch den ICC ermöglichte Schlafzustand transparent, d. h. andere Teilnehmer können auf Basis der Buskommunikation ein schlafendes Steuergerät nicht erkennen.

Wie in Abbildung 3b dargestellt, wird bei der Verwendung eines ICCs der Inhibit-Pin anstelle des Transceivers direkt von der ICC-Zustandsmaschine angesteuert. Zur besseren Übersicht ist der Spannungsregler nicht eingezeichnet. Durch die Entkoppelung vom Transceiver wird der μC bei aktivem ICC nun nicht mehr durch jegliche Buskommunikation geweckt.

Je nach erforderlicher Reaktionszeit nach einem Weckereignis kann der μC zwei verschiedene Energiesparmodi einnehmen. Der Zielzustand wird dabei durch die Applikation vorgegeben. Ist eine sehr geringe Latenz zwischen Eingang eines Weckgrunds (beispielsweise Betätigung eines Tasters, welche von einem anderen Steuergerät über den Kommunikationsbus übertragen wird) bis zur ersten Reaktion des μC (z. B. Ansteuerung einer LED) erforderlich, bleibt der Inhibit-Pin aktiv. Die Applikation stoppt ihre Ausführung, indem sie den μC in den *Halt-Modus* schaltet. Hier wird der μC, also insbesondere die CPU und Speicher weiterhin bestromt, aber nicht getaktet. Meldet der ICC ein Weckereignis, wird die Ausführung fortgesetzt. Da Speicherinhalt und damit der Zustand der Applikation erhalten bleiben und der genaue Weckgrund im ICC gespeichert ist, ergeben sich bei optimiertem Design der SW-Cs Reaktionszeiten von deutlich unter einer Millisekunde.

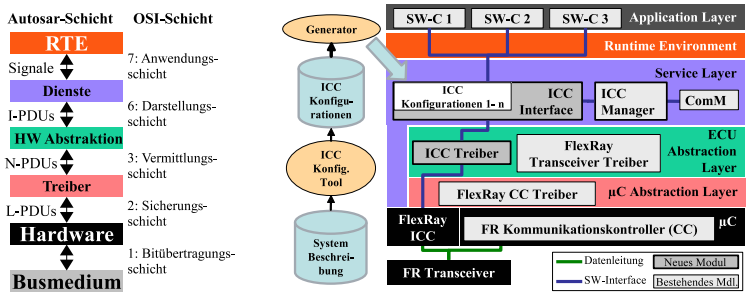
Für Zustände, in denen längere Reaktionszeiten gültig sind (beispielsweise beim Ladevorgang der Hochspannungsbatterie), kann der μC über den Inhibit-Pin des ICCs alternativ auch komplett stromlos geschaltet werden. Im Vergleich zur ersten Variante sind bei aktivem ICC dadurch deutlich höhere Energieeinsparungen erreichbar. Nach einem Weckereignis aktiviert der ICC den Spannungsregler wieder, der μC wird gestartet. Durch die notwendige Initialisierung der BSW sind die zu erwartenden Reaktionszeiten jedoch wesentlich höher (je nach Steuergerät ab ca. 25 Millisekunden).

Für beide Varianten wird die Kommunikation in Senderichtung nach dem Wecken weiterhin vom ICC übernommen. Dies ist erforderlich, da der Zustand der BSW nach Aktivierung aus dem Halt-Modus veraltet ist, und bei der kompletten Deaktivierung des μC neu initialisiert werden muss. Nachdem die Applikation wieder gültige Sendewerte bereitstellt, kann die Kommunikation vom ICC wieder komplett an die Software übergeben werden.

Die entstehende Übergangszeit hängt dabei stark von der jeweiligen Konfiguration des Steuergerätes ab, ein pauschaler Wert kann nicht angegeben werden. Die Applikation muss vor der Aktivierung des ICCs also sicherstellen, dass für den erkannten Fahrzeugzustand ein statisches Sendeverhalten für die entsprechende Dauer nach dem Wecken gültig ist.

Der hier beschriebene ICC-Ansatz gibt die logische Aufteilung der beteiligten Komponenten wieder. In der von uns prototypisch implementierten Variante teilen sich ICC und das restliche System den Kommunikationskontroller [SLSB11]. Ebenso würde ein auf dem μC integrierter ICC den existierenden Kommunikationskontroller sehr wahrscheinlich wiederverwenden. Die restlichen Komponenten auf dem μC (wie z. B. CPU oder A/D-Wandler) könnten dann über Takt- oder Power-Gating-Mechanismen deaktiviert werden.

Bei der zu erwartenden Energieeinsparung berücksichtigen wir nur die durch den μC entstehende Einsparung. Diese liegt je nach μC-Leistungsklasse zwischen 1 - 2,5 W pro Knoten. Eine pauschale Aussage zu weiteren Einsparungen, die bei aktivem ICC durch Deaktivierung der restlichen Komponenten des Steuergerätes entstehen, kann nicht getroffen werden. Eine detaillierte Vorstellung der durch die durch selektive Knotenabschaltung zu erwartenden Energieeinsparungen, sowie deren Auswirkungen auf Kraftstoffverbrauch und Abgasemissionen, werden in [SKSS11] vorgestellt.



(a) Aufteilung von Autosar-Signalen, Interaction Layer-Protocol Data Units (I-PDUs), Network Layer-PDUs und Data Link Layer-PDUs nach den Schichten des OSI-Modells.

(b) ICC-Konfigurationen werden vor der Systemgenerierung auf Signalbasis festgelegt. Generiertools wandeln die Bus- und Hardwareunabhängige Beschreibung in ICC-spezifischen Code um. Zur Laufzeit kann zwischen den statischen Konfigurationssätzen umgeschaltet werden.

Abbildung 4: Abstraktion der Kommunikation in Autosar und Einbindung der ICC-Konfiguration.

3 Konfiguration der Intelligenten Kommunikationskontroller

Die Konfiguration von Steuergeräten, verwendeten Bussen und die Spezifikation der gesamten Kommunikation erfolgt unter Autosar über ein standardisiertes Austauschformat, der *Systembeschreibung*. Die Autosar Entwurfsmethodologie sieht dabei vor, dass sich die Konfiguration stark an den unterschiedlichen Abstraktionsschichten orientiert: die von einem Steuergerät empfangenen und gesendeten Daten werden auf oberster Ebene über Busprotokoll- und Kanalunabhängige Signale definiert. Die wichtigsten Attribute eines Signals sind Sendeart (z. B. spontan oder zyklisch), Größe und Datentyp. Erst beim Durchlaufen der niederen Schichten der BSW werden Signale schrittweise ihren vorgegebenen Positionen im Dateninhalt von busspezifischen Frames zugeordnet. Der Datenfluss mit den entsprechenden Ebenen des OSI-Modells ist in Abbildung 4a dargestellt.

Einzelne Module, wie z. B. der Treiber des FlexRay-Kommunikationskontrollers, werden einzeln konfiguriert. Durch diese Aufteilung können die Module bei Bedarf ausgetauscht werden, beispielsweise bei einer Änderung des Transceivers oder des μCs, ohne Anpassungen in höhere Schichten zu erfordern. Die Erstellung der BSW und RTE erfolgt größtenteils automatisiert durch Generiertools, welche die durch die Systembeschreibung vorgegebene Konfiguration in hardwarespezifischen Code umwandeln.

Hinsichtlich der Integration der ICC-Konfiguration in Autosar sind damit die in den folgenden Abschnitten diskutierten Fragestellungen von Interesse:

- Festlegung von zusätzlich benötigten und bereits vorhandene Konfigurationsdaten.
- Platzierung der zur ICC-Konfiguration benötigten Schritte in die verschiedenen Abschnitte des Autosar-Systemdesigns.
- Verwaltung der ICC-Konfiguration zur Laufzeit.

ISBN 978-3-88579-286-4

Tabelle 2: Übersicht über für die ICC-Konfiguration benötigten Parameter

Attribut	aktueller Stand
Businformationen (z. b. Timing, Anzahl Slots)	vorhanden
ICC-spezifische Konfiguration (z. B. für Initialisierung)	neu
Daten für Konfiguration der Empfangsfilter	
Zuordnung von Signalen zu Frame und Position in Payload	vorhanden
Verleichswert und -operation, Wert für Timeout-Überwachung	neu
Daten für Konfiguration der Sendepuffer	
Zuordnung von Signalen zu Position in Frame, Zykluszeit des Frames	vorhanden
Statischer Wert für zu sendendes Signal	neu
Default-Wert für die restlichen Signale im Frame	vorhanden

3.1 Erweiterung der Systembeschreibung für den Einsatz von ICCs

Zur ICC-Konfiguration werden im Wesentlichen die auf empfangene Frames anzuwendenden Filter, sowie die zu sendenden Frames benötigt. Die dafür notwendigen Parameter können zum Großteil aus der existierenden Systemkonfiguration hergeleitet werden. In Tabelle 2 ist dargestellt, welche für den ICC benötigten Autosar-Konfigurationsdaten bereits vorhanden sind und welche hinzukommen.

Die ICC Filterkonfiguration kann basierend auf der bisherigen Systembeschreibung signalbasiert erfolgen. Pro Filter müssen neben dem zu filternden Signal (z. B. Fahrzeuggeschwindigkeit) zusätzlich Vergleichswert und -operation angegeben werden. Die Konfiguration von zu sendenden Signalen mit einem vorgegebenen Standard-Wert kann ebenfalls auf Basis vorhandener Konfigurationsdaten erfolgen. Da jedoch nicht sichergestellt ist, dass immer alle in einem Frame übertragenden Signale in der ICC-Konfiguration enthalten sind, können Datenbereiche im Frame unbelegt bleiben. Diese können mit dem bereits vorhandenen Default-Wert belegt werden, der momentan bei deaktiviertem Kommunikationskanal verwendet wird. Alle benötigten ICC-Parameter können als reine Erweiterung in die bereits vorhandene Systembeschreibung eingebracht werden, es ist also keine Änderung an bestehenden Datenstrukturen notwendig.

Zusätzlich muss die Toolkette, die einen Großteil der Steuergerätesoftware erzeugt, erweitert werden. Wie in Abbildung 4b dargestellt, erzeugen dabei Generiertools aus der Systembeschreibung den implementierungsabhängigen Code zur ICC-Konfiguration.

3.2 Erstellung und Verwaltung der ICC-Konfiguration zur Design- und Laufzeit

Um Fehler bei der Bedatung des ICCs frühzeitig zu erkennen, verfolgen wir den in Abbildung 4b gezeigten statischen Konfigurationsansatz. Die komplette ICC-Beschreibung enthält eine variable Anzahl von ICC-Konfigurationssätzen. Ein Konfigurationssatz enthält alle benötigten Empfangsfilter und Sendedaten. Die Konfigurationssätze werden statisch zur Designzeit erstellt, d. h. ein dynamisches Umladen oder Kombinieren einzelner ICC-Parameter zur Laufzeit ist nicht möglich. Dadurch können Konfigurationsfehler, wie z. B. nicht unterstützte Filteroperationen, oder eine zu große Anzahl von Empfangsfiltern,

bereits zur Generierzeit des Systems erkannt werden und müssen nicht zur Laufzeit behandelt werden.

Durch die unterschiedlichen Konfigurationen kann zur Laufzeit das Verhalten des ICC flexibel an den aktuellen Fahrzeugzustand angepasst werden. Erkennt die Applikationssoftware eines Steuergerätes, dass sie nicht benötigt wird, lädt sie die passende ICC-Konfiguration, aktiviert den ICC und versetzt den μC in einen energiesparenden Zustand. Dabei muss sichergestellt sein, dass die Filterbedingungen den μC wieder aufwecken, sobald ein dynamisches Verhalten des Steuergerätes nötig ist (z. B. unter einer bestimmten Geschwindigkeit), und dass der Versand von statischen Statusnachrichten ausreichend ist.

In Abbildung 4b sind die zur Einbindung wichtigsten Schnittstellen und zum Teil neuen BSW-Module dargestellt. Über ein standardisiertes ICC-Interface wird die zu ladende ICC-Konfiguration übergeben. Da SW-Cs gleichberechtigt Konfigurationen anfordern können, kann eine vorher geladene Konfiguration überschrieben werden. Um zu vermeiden, dass SW-Cs konkurrierende Anforderungen stellen, ist es sinnvoll, eine Master-Komponente zu definieren, die in Abhängigkeit vom Fahrzeugzustand eine passende Konfiguration lädt. Die restlichen SW-Cs sollten dann anhand der aktuellen Konfiguration nur noch den im nächsten Kapitel beschriebenen erforderlichen Kommunikationsmodus anfordern.

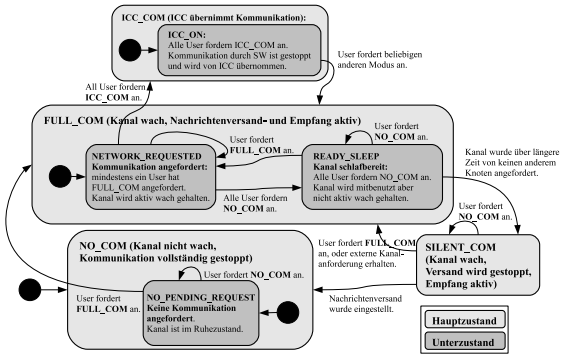
Der ICC-Manager arbitriert den ICC-Zustand anhand der von SW-Cs eingehenden Anfragen. Die Synchronisierung der Kommunikationsrechte zwischen ICC und Software erfolgt durch den *Communication Manger* (ComM). Hardwareabhängigkeiten werden durch einen ICC-Treiber abstrahiert. Da, wie in Abschnitt 2.2 beschrieben, der ICC logisch als externe Komponente behandelt wird, ist der ICC-Treiber im Gegensatz zum Treiber des FlexRay-CC dem ECU- und nicht dem μC-Abstraction Layer zugeordnet.

4 Umschaltung zwischen herkömmlicher und ICC-Kommunikation

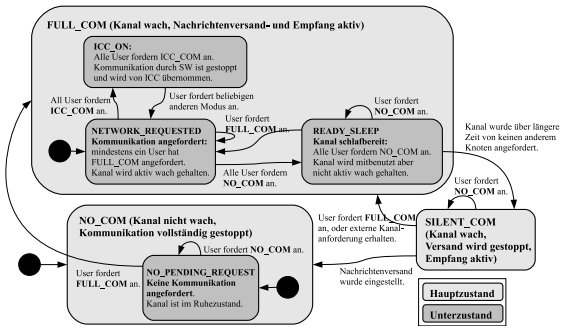
Autosar bietet verschiedene Möglichkeiten, interne Modi anzufordern, zu arbitrieren und zu verwalten. Dabei wird allgemein zwischen Steuergeräte-, Kommunikations-, und implementierungsabhängigen Zuständen unterschieden.

Die Verwaltung und Bestimmung des aktuellen Kommunikationszustandes übernimmt der *Communication Manager* (ComM). Über *User-Handles* können SW-Cs die Kommunikationsmodi *NO_COM* (SW-C benötigt keine Kommunikation), oder *FULL_COM* (SW-C benötigt Kommunikation) anfordern. In der BSW sind User statisch einem oder mehreren Kommunikationskanälen zugeordnet. Damit wird die genaue Kommunikationsanbindung gegenüber den SW-Cs abstrahiert, d. h. diese können allgemein ihren Kommunikationsbedarf melden, ohne die genaue Busanbindung des Steuergerätes zu kennen.

Die Arbitrierung des ComM für einen Kanal erfolgt aktuell wie folgt: solange für einen Kanal mindestens ein User Kommunikation anfordert, befindet er sich im FULL_COM-Modus. Der Kanal wird aktiv wach gehalten, indem zyklisch *Netzwerkmanagment*-Nachrichten (NM-Nachrichten) verschickt werden. Wenn keine lokale Anforderung vorliegt, wird der Kanal zuerst in den Unterzustand *Ready Sleep* geschaltet. Solange mindestens ein anderer Busteilnehmer NM-Nachrichten verschickt, bleibt der Kanal aktiv, die Kommunikation muss aufrecht erhalten werden. Wenn keine NM-Nachrichten mehr empfangen



(a) Abbildung von ICC_ON als neuer Hauptzustand. SW-Cs müssen an den neuen ICC_ON-Zustand angepasst werden.



(b) Abbildung von ICC_ON als Unterzustand von FULL_COM. Bei Abfrage des aktuellen Kommunikationsmodus wird immer ein bekannter Zustand zurückgegeben.

Abbildung 5: Mögliche Abbildung des neuen ICC-Kommunikationszustandes in der vereinfachten Zustandsmaschine des Autosar Communication Managers für einen Kanal [AUT10b].

werden, wird der Kanal nach einer definierten Zeit abgeschaltet.

Die ICC-Integration in Autosar erfordert eine Anpassung der ComM-Zustandsmaschine. Zur Umschaltung der Kommunikationsrechte zwischen ICC und Software müssen dabei

- der neue ICC-Kommunikationszustand im Autosar-Stack abgebildet werden;
- und Mechanismen zur Anforderung und Arbitrierung des gewünschten Kommunikationszustandes erweitert werden.

In den nächsten Abschnitten werden mögliche Implementierungsvarianten vorgestellt und bezüglich ihres Änderungsaufwandes und ihrer Rückwärtskompatibilität bewertet.

4.1 Implementierungsvarianten

Für die Abbildung der ICC-Kommunikation ergeben sich 3 unterschiedliche Ansätze:

Abbildung als neuer Kommunikationszustand (Var. I): die ICC-Kommunikation kann als eigenständiger, neuer Zustand dargestellt werden. Wie Abbildung 5a gezeigt, wird da-

bei aus dem vorhandenen Zustand FULL_COM in den neuen Modus ICC_COM gewechselt, wenn alle Komponenten ICC_COM anfordern. Solange mindestens eine Komponente einen anderen Zustand anfordert, wird die Kommunikation nicht an den ICC übergeben.

Die Einführung eines neuen Zustandes ist jedoch nicht vollständig rückwärtskompatibel, da SW-Cs, die nur passiv den aktuellen Kommunikationsmodus abfragen, an den neuen Zustand angepasst werden müssen.

Abbildung als Unterzustand von FULL_COM (Var. II): wie in Abbildung 5b dargestellt, kann die ICC-Kommunikation auch als Unterzustand von FULL_COM abgebildet werden. Wie in der vorherigen Variante wird nie in ICC_COM gewechselt, solange dieser nicht von allen berechtigten Komponenten angefordert wird. Zusätzlich erhalten unmodifizierte Komponenten bei der Abfrage des aktuellen Kommunikationsmodus über bestehende Schnittstellen weiterhin gültige Werte und müssen den neuen ICC-Modus nicht kennen. Damit sind zwei wichtige Kriterien zum Erhalt der Rückwärtskompatibilität erfüllt.

Abbildung als gemischter Steuergerätezustand (Var. III): alternativ können Kommunikations- und μC-Zustand auch kombiniert dargestellt werden. Im Gegensatz zu den vorher beschriebenen Varianten, kann dabei direkt nach Aktivierung des ICCs ein Übergang in einen energiesparenden Zustand erfolgen.

Zur Verwaltung solcher gemischten Zustände kann der *Basic Software Mode Manager* (BswM) verwendet werden [AUT10a]. Dieser verwaltet abstrakte Anforderung anhand von konfigurierbaren Arbitrierungsregeln. Damit können Modi definiert werden, die die ICC-Konfiguration und den Steuergerätezustand direkt bestimmen.

4.2 Vergleich

Für die Varianten I und II stellt der ICC-Modus immer einen reinen Kommunikationszustand dar. Weitere Aktionen, wie z. B. das Wechseln in einen energiesparenden Zustand, erfolgen nachgelagert, wenn die Applikationssoftware erkennt, dass die Kommunikation vom ICC übernommen wurde. Beide Ansätze bieten eine standardisierte Schnittstelle, der ICC-Modus ist wie die anderen Zustände des ComM eindeutig definiert. Obwohl bestehende Mechanismen wie die Verwaltung der Kommunikationsanforderungen über User zum Teil wiederverwendet werden können, erfordert die Einführung eines neuen Zustandes oder Unterzustandes im ComM größere BSW-Anpassungen. Zusätzlich muss die Übergabe von Kommunikationsrechten zwischen Software und ICC implementiert werden.

Bei der Verwendung des BswM (Var. III) muss die Applikation die Bedeutung der implementierungsabhängigen BswM-Zustände kennen. Dieser Ansatz bietet damit keine standardisierte Schnittstelle zu den SW-Cs. Als weiterer Nachteil können, je nach Wahl der Arbitrierungsregeln, Übernahmekomponenten bei aktivem ICC ausgeführt werden. Es lässt sich damit keine allgemeine Aussage über die Rückwärtskompatibilität treffen. Andererseits werden durch die Verwendung des BswM aufwendige Anpassungen existierender Autosar-Module vermieden. Insbesondere als Übergangslösung, oder bei der Verwendung von nicht standardisierten Funktionen ist der BswM damit eine gute Alternative.

Aufgrund der höheren Rückwärtskompatibilität und der Verwendung von standardisierten Zuständen bewerten wir für eine langfristige Integration des ICC eine Abbildung als

Unterzustand des bestehenden ComM-Modus FULL_COM als sinnvollste Lösung.

Bei allen diskutierten Varianten soll der durch den ICC ermöglichte Schlafzustand nicht von anderen Knoten erkennbar sein. Andere Teilnehmer sollen also auf Basis der Buskommunikation nicht feststellen können, wann der ICC eines Knotens aktiv ist. Das bedeutet insbesondere, dass Signale eines schlafenden Knotens nicht fehlen oder unplausibel sein dürfen und dass er selbständig, ohne externe Anforderung aufwacht, wenn nötig.

Damit liegt die Verantwortlichkeit für die Auswahl der geeigneten Konfiguration und der Aktivierung des ICC steuergerätelokal bei der Applikation. Alle SW-Cs auf einem Steuergerät müssen entsprechend angepasst werden, damit sie - in Abhängigkeit des Fahrzeugzustandes - die erforderliche ICC-Konfiguration laden, den ICC aktivieren und den passenden Energiesparmodus wählen. Eine verteilte, steuergeräteübergreifende Abstimmung im Netzwerk ist jedoch nicht nötig.

5 Zusammenfassung und nächste Schritte

In diesem Beitrag haben wir den Einsatz von Intelligenten Kommunikationskontrollern (ICCs) zur selektiven Knotenabschaltung temporär nicht benötigter Knoten vorgestellt. ICCs dienen der Steigerung der Energieeffizienz aktueller E/E-Architekturen und erfordern Gegensatz zu busunabhängigen Mechanismen keine zusätzliche Verkabelung.

Um ICCs auf breiter Basis einsetzen zu können, streben wir die Integration in den für eingebettet Systeme im Automotive Umfeld ausgelegten Betriebsystemstandard Autosar an. Dazu haben wir Ansätze vorgestellt, die die Autosar-konforme Konfiguration von ICCs und die Verwaltung des entstehenden neuen ICC-Kommunikationsmodus ermöglichen. Bei der Einbindung in den Autosar-Entwicklungsprozess unterscheiden wir dabei zwischen statischen Anteilen und Umfängen, die zur Laufzeit kontrolliert werden müssen.

Die Integration des ICC-Konzeptes wird unter dem Autosar-Konzept *Pretended Networking* mit verschiedenen Autosar-Partnern verfolgt. Ziel der entstehenden Pretended Networking-Spezifikation ist die Unterstützung verschiedener ICC-Implementierungen, insbesondere für verschiedene Busse oder mit variierendem Funktionsumfang.

Zur weiteren Absicherung bereiten wir eine prototypische Hardware- und Softwareimplementierung vor. Der Hardwaredemonstrator verwendet einen herkömmlichen μC, der über einen FPGA um die ICC-Funktionen erweitert. Angepasst auf diesen Hardwaredemonstrator wird die Softwareimplementierung einen bestehenden Autosar-Stack um die in diesem Papier vorgestellten Ansätze erweitern. Zusätzlich arbeiten wir an einer Klassifizierung und Bewertung von Fehlerzuständen, die durch den neuen ICC-Kommunikationszustand auftreten können.

Literatur

[AUT03] AUTOSAR. *Automotive Open System Architecture*, 2003.

[AUT10a] AUTOSAR. *Specification of Basic Software Mode Manager V. 1.1.0*, Oktober 2010.

[AUT10b] AUTOSAR. *Specification of Communication Manager V. 3.1.0*, November 2010.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

[EH10] Bernd Elend und Holger Huber. Reduktion des CO_2 Ausstoßes durch Teilnetzbetrieb in der Fahrzeugvernetzung. In *2. Elektronik automotive congress*, Mai 2010.

[F+09] Simon Fürst et al. AUTOSAR - A Worldwide Standard is on the road. In *14th International VDI Congress Electronic Systems for Vehicles, Baden-Baden*, 2009.

[FSG10] Martin Fuchs, Patrick Scheer und Andreas Grzemba. Selektiver Teilnetzbetrieb im Fahrzeug: Eine Realisierung für den CAN-Bus und Adaption auf andere Bussysteme. In *AmE 2010 - Automotive meets Electronics*, Seiten 15–18, 2010.

[H+04] Harald Heinecke et al. AUTomotive Open System ARchitecture - An Industry-Wide Initiative to Manage the Complexity of Emerging Automotive E/E Architectures. In *Convergence International Congress & Exposition On Transportation Electronics*, Seiten 325–332, 2004.

[H+06] Harald Heinecke et al. AUTOSAR - Current results and preparations for exploitation. In *7th EUROFORUM conference Software in the vehicle Stuttgart*, 2006.

[Hud09] R. Hudi. Die E/E Entwicklung im Wandel auf dem Weg zur Elektromobilität. In *13. Internationaler Fachkongress Fortschritte in der Automobil-Elektronik*, 2009.

[KBR09] Matthias Krause, Oliver Bringmann und Wolfgang Rosenstiel. Verification of AUTOSAR Software by SystemC-Based Virtual Prototyping. In Wolfgang Ecker, Wolfgang Müller und Rainer Dömer, Hrsg., *Hardware-dependent Software*, Seiten 261–293. Springer Netherlands, 2009. 10.1007/978-1-4020-9436-1_10.

[KPLJ08] Daehyun Kum, Gwang-Min Park, Seonghun Lee und Wooyoung Jung. AUTOSAR migration from existing automotive software. In *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, Seiten 558–562, Oktober 2008.

[Rau07] Mathias Rausch. *FlexRay. Grundlagen, Funktionsweise, Anwendung*. Hanser Fachbuch, 2007.

[RB96] Klaus Rohde-Brandenburger. Verfahren zur einfachen und sicheren Abschätzung von Kraftstoffverbrauchspotentialen. Haus der Technik, Essen, November 1996.

[RJV09] Devendra Rai, T. K. Jestin und Lev Vitkin. Model-Based Development of AUTOSAR-Compliant Applications: Exterior Lights Module Case Study. *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, 1(1):84–91, April 2009.

[SKSS11] Christoph Schmutzler, Andreas Krüger, Fred Schuster und Martin Simons. Energy Effcient Automotive Networks: State of the Art and Challenges Ahead. *Int. J. Communication Networks and Distributed Systems*, 2011. Im Druck.

[SLSB11] Christoph Schmutzler, Abdallah Lakhtel, Martin Simons und Jürgen Becker. Increasing Energy Efficiency of Automotive E/E-Architectures with Intelligent Communication Controllers for FlexRay. In Vorbereitung, 2011.

[VF10] Stefan Voget und P. Favrais. How the Concepts of the Automotive Standard "AUTOSAR" Are Realized in New Seamless tool-chains. In *Embedded Real-Time Software and Systems, Toulouse*, 2010.

[WLMS10] Thomas Weber, Vera Lauer, Dieter Mann und Martin Simons. The Comprehensive Energy Management - From the Conventional Powertrain to the Full e-Drive. In *4. VDI-Tagung: Baden-Baden Spezial 2010 Elektrisches Fahren machbar machen*, 2010.

[WR06] Henning Wallentowitz und Konrad Reif. *Handbuch Kraftfahrzeugelektronik: Grundlagen, Komponenten, Systeme, Anwendungen*. Vieweg, Wiesbaden, 2006.

[ZS08] Werner Zimmermann und Ralf Schmidgall. *Bussysteme in der Fahrzeugtechnik: Protokolle und Standards*. Vieweg, Wiesbaden, 3.. Auflage, 2008.