

Entwickeln mit J1939

Steuergeräte-Vernetzung in Nutzfahrzeugen

Bei der Vernetzung von Steuergeräten im Nutzfahrzeug spielt das J1939-Protokoll eine zentrale Rolle, was aber einige Herausforderungen mit sich bringt. Geeignete Tools helfen jedoch bei der Entwicklung mit dieser Netzwerk-Architektur.

Von Peter Fellmeth und Thomas Löffler

Die J1939-Netzwerke basieren auf dem CAN-Bus (CAN-Highspeed nach ISO 11898) und finden in Antriebsstrang- und Fahrgestellkomponenten bevorzugten Einsatz. Das Protokoll schafft eine einheitliche Grundlage für die Kommunikation zwischen den elektronischen Steuergeräten und arbeitet nach dem Plug-and-Play-Prinzip. Für Entwicklungen auf Basis dieses Protokolls gibt es spezielle J1939-Tools und Software-Komponenten, die den Ingenieur von der detaillierten Einarbeitung in das J1939-Protokoll befreien und die Qualität des Entwicklungsprozesses steigern.

Warum J1939 sinnvoll ist

Das aus den USA stammende und von der Society of Automotive Engineers

(SAE) definierte J1939-Protokoll dient vor allem dem Ziel, eine einheitliche Behandlung der gebräuchlichen Fahrzeugkomponenten bei verschiedenen Fahrzeugtypen bzw. Herstellern sicherzustellen. In diesem Zusammenhang ist interessant, dass es zwischen dem europäischen und dem amerikanischen Nutzfahrzeugmarkt ausgeprägte Unterschiede gibt. So schreiben die Nfz-Käufer in den USA den OEMs in der Regel vor, welche Komponenten sie in den jeweiligen Fahrzeugen zu verbauen haben. In Europa hingegen bestimmen die OEMs vollständig das Design des kompletten Fahrzeugs einschließlich der Komponenten und deren Konfiguration.

Wichtig neben der Kommunikation über einheitlich definierte Signale und Datenformate ist natürlich, dass die Empfänger wissen, wie die Informa-

tionen jeweils zu interpretieren sind. Im Idealfall lassen sich die einzelnen J1939-Komponenten dann nach dem Plug-and-Play-Schema "zusammenstecken". Trotz aller Vereinheitlichungen bietet J1939 genügend Freiheit für eine herstellerspezifische Erweiterung der Kommunikation. Das ist insbesondere für Innovationen wichtig, weil kein Hersteller diese vor der Implementierung zuvor in Gremien ausbreiten und diskutieren möchte.

ISO-Schichtmodell entkoppelt Applikation von Übertragungsphysik

Aus Sicht des ISO/OSI-Schichtmodells baut J1939 im Wesentlichen auf dem Application Layer (Schicht 7), dem Network Layer (Schicht 3), dem Data Link Layer (Schicht 2) und dem Physical Layer (Schicht 1) auf (Bild 1). Dies erlaubt den Entwicklern, nur mit Signalsequenzen arbeiten zu können, ohne sich auf Applikationsebene um Details der Kommunikation wie beispielsweise die Transportprotokolle kümmern zu müssen. Auch die Dokumentation/Definition von J1939 orientiert sich an den einzelnen Schichten, was sich in der Bezeichnung der insgesamt 14 Werke des Standards aus-

drückt. So behandeln die Dokumente des 7er-Nummernkreises wie z.B. J1939/71 den Application Layer, das Dokument 21 den Data Link Layer usw. (einige Details hierzu im Kasten).

Das CAN-Botschaftsformat bei J1939

J1939 nutzt zwar normale 29-bit-CAN-Botschaften mit bis zu 8 byte Daten, jedoch wird über den CAN-Identifizierer quasi die Maske eines einheitlichen J1939-Schemas gelegt (Bild 2). Dies ist aufgrund der Plug-and-Play-Eigenschaften notwendig. So enthält der CAN-Identifizierer neben der Identifizierung der Nutzdaten mit Hilfe der Parameter Group Number (PGN) auch die J1939-Steuergeräte-Adressen des Senders und gegebenenfalls auch des Empfänger.

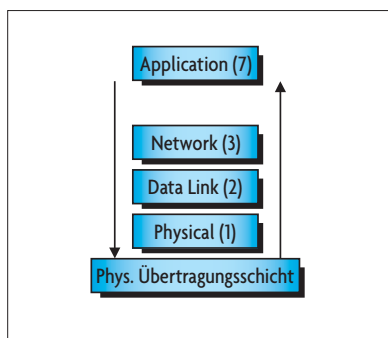


Bild 1. Das J1939-Protokoll baut im Wesentlichen auf dem Application Layer (Schicht 7), dem Network Layer (Schicht 3), dem Data Link Layer (Schicht 2) und dem Physical Layer (Schicht 1) auf, so dass man sich auf der Applikationsebene nicht mehr mit den Details der Kommunikation beschäftigen muss.

(Quelle der Grafiken: Vector Informatik)

Außerdem sind die drei höchstwertigen Bits des CAN-Identifizierers als Prioritätsfeld reserviert, die zwar nicht die im CAN-Protokoll implizite Priorität ersetzen, jedoch die Zuordnung der Parameter Groups in bis zu acht J1939-spezifische Prioritätsstufen ermöglichen. Mit Hilfe dieser Prioritäten kann zum Zeitpunkt der Übertragung der Parameter Group oder während einer optionalen Konfigurationsphase des Steuergeräts die Priorität an die aktuellen Anforderungen der Anwendung angepasst werden. Dies ermöglicht eine feine Abstimmung der Kommunikation auf die

Wesentliche Einzeldokumente des J1939-Standards

J1939	Generelle Beschreibung des Netzwerks	J1939/31	Bridge, Router, Gateway, Filter
J1939/0X	Beschreibung der Anwendung	J1939/21	Data Link Layer (Transportprotokolle, Acknowledge, ...)
J1939/01	Nutzfahrzeuge (Truck und Bus)	J1939/1X	Beschreibung Physical Layer
J1939/02	Landwirtschaft (Agricultural Equipment)	J1939/11	250 kbit/s, Twisted pair, geschirmt
J1939/7X	Application Layer	J1939/12	250 kbit/s, Twisted quad
J1939/71	Fahrzeug	J1939/13	Diagnose-Stecker
J1939/73	Diagnose		

Anwendung, ohne dass die SAE zum Zeitpunkt der Definition der Parameter Group die Priorität festlegt.

Nomen est omen: die J1939-Gerätenamen

J1939 definiert Gerätenamen, die jeweils durch eine 64-bit-Zahl repräsentiert sind. Schaltet man ein Steuergerät im Plug-and-Play-Netzwerk aktiv, dient der Gerätenamen zur Identifikation des Geräts und seiner Funktion. Der Gerätenamen unterteilt sich in verschiedene Elemente, zwischen denen teilweise Abhängigkeiten bestehen. Zu den unabhängigen Feldern gehören unter anderem die Industry Group und der Manufacturer Code. Über die Industriegruppe legt man die im Netzwerk benötigten Funktionen fest, da das J1939-Protokoll nicht nur in herkömmlichen Nutzfahrzeugen zum Einsatz kommt, sondern auch in der Landtechnik oder im Maritim-Bereich. Jedes Steuergerät trägt einen von der SAE zugewiesenen Hersteller-Code, der dort zu beantragen ist. Da jedes Gerät zusätzlich eine Seriennummer hat, ist der gesamte Name weltweit eindeutig: Es kommen keine Überschneidungen vor.

Da die Adressierbarkeit der Geräte über den 64 bit langen Gerätenamen in der Praxis mit CAN ineffizient ist, sind den einzelnen Fahrzeugkomponenten im Nutzfahrzeug-Umfeld durch die SAE feste 8-bit-Adressen zugewiesen; diese ändern sich in der Regel zur Laufzeit nie. Für die Landtechnik und den Maritim-Bereich trifft dies nicht zu, dort werden die Adressen beim Start unter Berücksichtigung des Gerätenamens dynamisch ausgehandelt. Die Adressen 0 bis 127 sind den gängigsten Steuergeräten wie Motor, Getriebe, Retarder, Bremsen usw. zugeordnet, während der Bereich von 128 bis 247 für Landtechnik, Schifffahrt, Baumaschinen usw. reserviert ist. Die Adressen von 248 bis 253 belegen die Adressen von 248 bis 253.

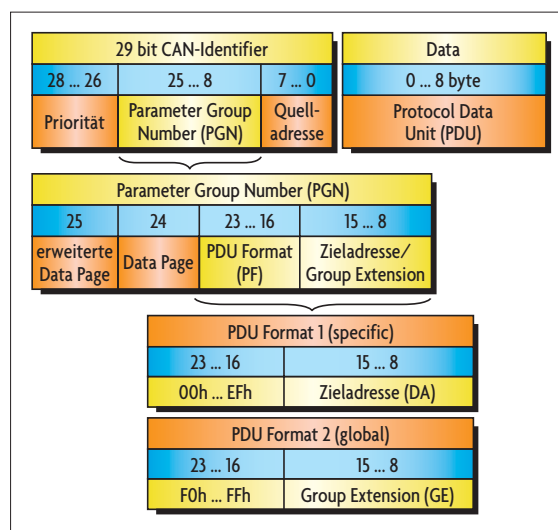


Bild 2. Aufgrund der Plug&Play-Fähigkeit bedarf das auf normalen 29-bit-CAN-Botschaften aufbauende J1939-Botschaftsformat einiger Ergänzungen. Über den CAN-Identifizierer wird quasi die Maske eines einheitlichen J1939-Schemas gelegt.

Übrig bleiben als Sonderadressen die 254, mit der man Geräte kennzeichnet, die keine eigene Adresse haben, sowie die 255 als globale Adresse zur Adressierung von Broadcast Messages.

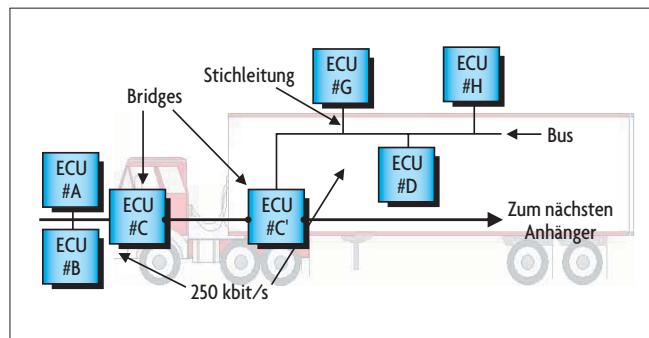
▣ Kommunikationsarten, Netzwerk-Topologie und Timing

Das J1939-Protokoll unterstützt zwei Kommunikationsarten. Punkt-zu-Punkt-Übertragungen (1:1) richten sich genau an eine Zieladresse, man verwendet sie beispielsweise zur Gerätekonfiguration oder für Steuergerätekommmandos. Broadcast-Botschaften (1:n) hingegen sind an sämtliche Bus Teilnehmer gleichzeitig adressiert, was praktisch ist für den Versand von Messwerten, zur Fehlerbehandlung und für Diagnosezwecke.

J1939 arbeitet mit einem passiven Bus, der an beiden Enden mit dem Wellenwiderstand von 120 Ω abgeschlossen ist. Das hat den Vorteil, dass man einzelne Steuergeräte über Stichleitungen mit einer Länge von 1 bis 3 m anschließen kann. Kabelbäume lassen sich somit flexibel gestalten, solange man eine Gesamtbuslänge von 40 m nicht überschreitet. Je nach physikalischer Übertragungsschicht sind zwischen zehn und maximal 30 Knoten an das Netzwerk anschließbar. Für Werkstatt-Tester und On-Board-Diagnosen stellt J1939 einen einheitlichen Diagnosezugang zur Verfügung. Gesetzliche Vorgaben verlangen, dass hierfür eine Stichleitung mit bis zu 5 m Länge möglich ist, etwa für Straßenkontrollen des

Abgassystems. Mit Bridges lassen sich J1939-Netzwerke außerdem auf Anhänger erweitern, um dort ein separates Netz zu realisieren (Bild 3). In der EU hat sich für diesen Zweck die ISO 11992 durchgesetzt, während in den USA „Power Line Carrier“ Stand der Technik ist.

Beim Design von J1939-Steuergeräten ist zu beachten, dass keine Botschaft durch Hardware- oder Design-Einschränkungen verlorengehen darf. Bei einer Übertragungsrate von 250 kbit/s dauert die Übertragung eines Bits 4 µs. Mit acht Datenbytes erhält man eine typische Botschaftslänge von ungefähr 128 bit, was zu einer Übertragungszeit von etwa 500 µs



▣ Bild 3. Hinsichtlich der Netzwerk-Topologie erlaubt J1939 eine flexible Gestaltung der Kabelbäume. Einzelne Steuergeräte lassen sich mit Stichleitungen von bis zu 3 m Länge anschließen, und Bridges ermöglichen es, separate Netzwerke auf Zugmaschinen und Anhänger zu realisieren.

pro CAN-Botschaft führt. Die kürzeste Botschaftslänge liegt jedoch bei 64 bit, d.h., man muss in der Lage sein, Botschaften im Abstand von 250 µs zu empfangen und entsprechend schnell zu verarbeiten. Dies führt in der Praxis aufgrund der oft limitierten CAN-Identifizier-Filtereigenschaften der CAN-Controller zu einer hohen Interrupt-Last. Zudem muss die Filterung deshalb meist in der Software realisiert werden.

▣ Test und Diagnose von J1939-Komponenten und -Systemen

Angesichts der steigenden Zahl von J1939-Steuergeräten und komplexer werdenden Software-Architekturen im Nutzfahrzeug gewinnt eine systematische Test- und Diagnosestrategie auch im J1939-Umfeld immer größere Bedeutung. Tests sind in allen Entwicklungsphasen von Funktionstests über Integrationstests bis hin zur Fahr-Erprobung im Gesamtfahrzeug unverzichtbar. Je später Fehler detektiert werden, desto aufwendiger und teurer ist bekanntlich deren Beseitigung. Allerdings sind Steuergeräte in der Regel erst dann umfassend testbar, nachdem sie in den Netzwerkverbund integriert wurden. So offenbaren sich die Schwachstellen erst sehr spät, es sei denn, man setzt von Anfang an auf die Unterstützung durch bewährte Software-Werkzeuge.

Vor diesem Hintergrund können sicherlich spezialisierte Tools den Entwicklern bei Test- und Diagnose-Aufgaben wesentliche Erleichterungen bringen. Mit einer praxiserfahrenen Werkzeugpalette für alle J1939-Vorhaben, wie sie im Programm von Vector Informatik zu finden ist, lassen sich beispielsweise die Aufgaben bei der Vernetzung und Kommunikation im Nutzfahrzeugbereich effizient lösen [1]. Neben Entwicklungs-, Test- und Analyse-Tools sind auf die speziellen Anforderungen J1939-basierter Anwendungen zugeschnittene Embedded-Software-Komponenten verfügbar, zusätzlich auch individuelle Projektarbeit und Schulungen.

Für das verbreitete Entwicklungs- und Testwerkzeug CANoe ist z.B. eine J1939-Erweiterung erhältlich, die den Nfz-Entwicklern die detaillierte Einarbeitung in das J1939-Protokoll erspart. Diese Erweiterung stockt die Funktionen der Basis-Software um alle notwendigen protokollspezifischen Besonderheiten auf. So dienen die in der Designphase damit erstellten Modelle und Datenbasen nicht nur als Grundlage für die Simulation während der Entwicklung, sondern auch für sämtliche entwicklungsbegleitenden Tests bis hin zu späteren Diagnose-

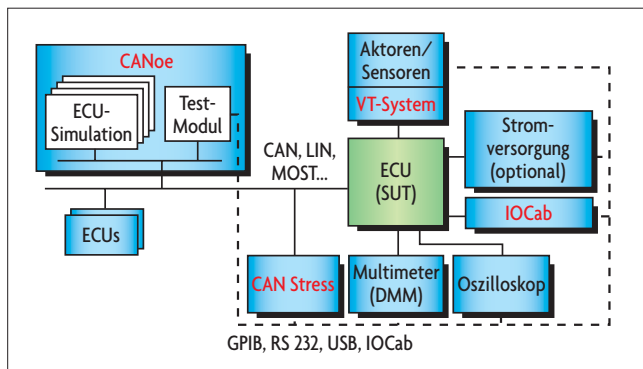


Bild 4. Durch entwicklungsbegleitende Tests mit Hilfe von Simulationen lassen sich in allen Entwicklungsphasen Fehler frühzeitig detektieren und beseitigen. Das „CANoe Test Feature Set“ stellt umfangreiche Test- und Analysemöglichkeiten zur Verfügung.

Aufgaben (Bild 4). Somit können mit Hilfe simulierter Knoten frühzeitig Tests für das zu entwickelnde Steuergerät aufgesetzt und durchgeführt werden. Die Tests werden während der Entwicklung weiter verfeinert und bei der Verifikation des Gesamtsystems eingesetzt.

■ Umfangreiche J1939 Test Library

Das „CANoe Test Feature Set“ umfasst CAPL-Testmodule, XML-Testmodule und .NET-Testmodule. Je nach Komplexität der Testaufgabe lassen sich damit die Herausforderungen von einfachen bis schwierigen Testfällen abdecken. Während die C-ähnliche Skriptsprache CAPL zum Erstellen umfangreicher Testszenarien prädestiniert ist, stehen bei XML eine einfache Parametrierung von Testpattern und eine einfache Generierung von Testabläufen im Vordergrund. Das bietet die Möglichkeit, anwendungsspezifische Testmodule (Funktionsbibliotheken) in CAPL zu implementieren und anschließend die Teststeuerung individuell passend zur Konfiguration des Steuergeräts generieren zu lassen. Die J1939-spezifischen Erweiterungen in den Test Service Libraries ermöglichen dem System die Reaktion auf Parameter Groups (PG) anstatt der typischen CAN-Identifizier, außerdem bieten sie Testmuster für J1939-Protokoll-Funktionen und Checks (Hintergrundüberwachungen) auf Protokollverletzungen. Somit lässt sich beispielsweise prüfen, ob das Steuergerät bei hoher Buslast in der Lage ist, alle Parameter Groups mit der konfigurierten Zykluszeit zu senden. Ferner können die Transportprotokolle BAM (Broadcast Announce Message) und CMDT (Connection Mode Data Transfer) für Testzwecke auch fehlerhaft gesendet werden.

Zur Erstellung der Testmodule ist neben dem J1939 Test Module Manager und dem komfortablen Test Automation

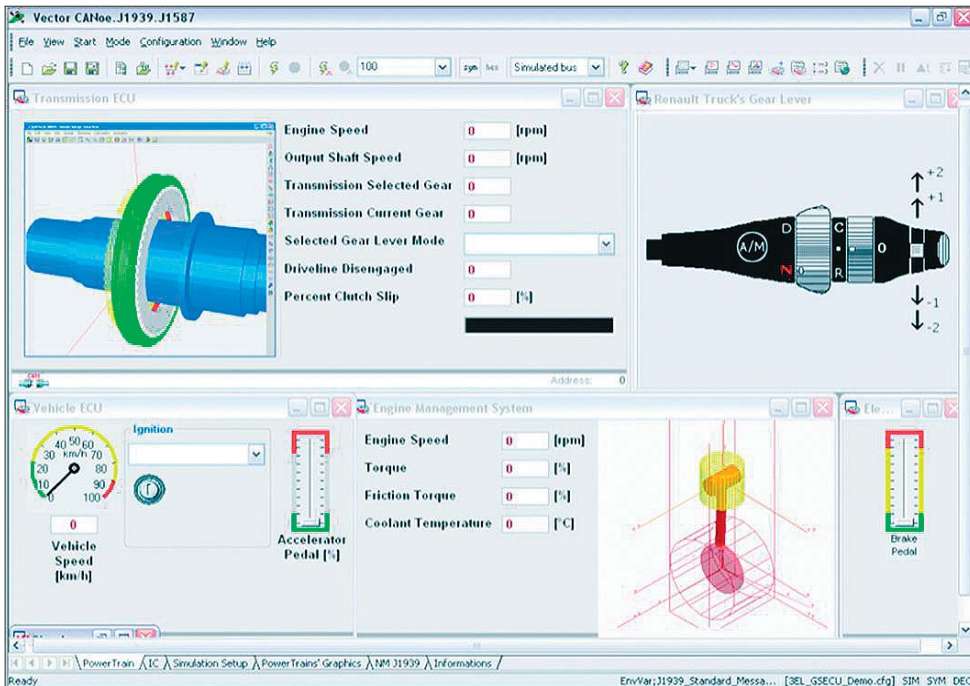


Bild 5. CANoe ist nicht nur in der Lage, Funktionsmodelle von Steuergeräten entlang des Entwicklungsprozesses zu simulieren und dabei mit Matlab/Simulink erstellte Modelle in die Szenarien einzubinden, sondern dient hier gleichzeitig als komfortable Benutzerschnittstelle. (Bild: Renault Trucks)

Editor die Option „DiVa“ nützlich. Sie schafft eine Verbindung zwischen CANoe und dem Diagnosespezifikations-Tool „CANdelaStudio“, so dass dort erstellte Spezifikationen für steuergerätespezifische Diagnose-Tests weiterverwendbar sind.

Zu den weiteren Funktionen des Test Feature Sets gehören eine Testablaufsteuerung und die automatische Reportgenerierung einschließlich statistischer Informationen im XML- oder HTML-Format nach individuellen Vorgaben. Weitergehende Optionen zur Automatisierung der Testvorgänge eröffnet die COM-Schnittstelle, zum Beispiel hinsichtlich Ablaufsteuerung, Parameteränderungen oder Statusabfragen. Die CANoe-Option J1939 stellt für Diagnose-Zwecke auch Trace-Fenster, einen J1939-Diagnose-Monitor und den J1939-Diagnose-Speicherzugriff bereit. Der Diagnose-Monitor unterstützt die verschiedenen J1939-Diagnose-Messages, wie beispielsweise DM1 und DM2, und dient zur Anzeige und zum Löschen von aktiven Fehlern. Möglich ist darüber hinaus ein Zugriff auf Speicherbereiche, Objekte und Parameter sowie eine zyklische Objektaktualisierung für Monitoring-Zwecke.

■ Matlab/Simulink-Modelle in J1939-Netzwerk-Simulationen einbinden

Normalerweise werden während der verschiedenen Nutzfahrzeug-Entwicklungsphasen verschiedene Funktionsmodelle für mechanische Komponenten wie Getriebe, Antriebsstrang oder sogar das komplette Fahrzeug erstellt. Steuergeräte-Architek-

turen bildet man zunächst in virtuellen Funktionsmodellen ab und realisiert sie Schritt für Schritt auf der endgültigen Hardware-Zielpattform. CANoe.J1939 integriert auch Matlab/Simulink-Modelle in die Steuergeräte- und Netzwerk-Simulationen (Bild 5). Hierzu erzeugt der Anwender mit dem Real-Time Workshop von Mathworks eine *.DLL für CANoe, so dass Variablen-Namen und Units kompatibel sind.

Entlang der verschiedenen Stufen des V-Entwicklungsmodells sind nun Einzeltests und Teilsystemtests möglich, bis hin zur endgültigen Verifikation des Gesamtsystems. So lassen sich Fehler frühzeitig aufspüren und beseitigen. Wird ein Fehler gefunden, können die automatisierten Tests jederzeit erneut gestartet werden und minimieren das Risiko auf Seiteneffekte bei der Fehlerbehebung. Die Entwicklung ist somit durch kurze Verifikationszyklen gekennzeichnet und erlaubt einen nahtlosen Übergang vom MiL (Model In the Loop) über SiL (Software In the Loop) zum realen Steuergerät (HiL – Hardware In the Loop). Bei besonderen Anforderungen an die Echtzeit-Fähigkeit der Simulationsplattform ist mit CANoe RT eine spezielle Echtzeit-Version verfügbar.

Zu schnellen Entwicklungsergebnissen führen die J1939-Software-Komponenten des Pakets CANbedded J1939. Sie entbinden die Entwickler

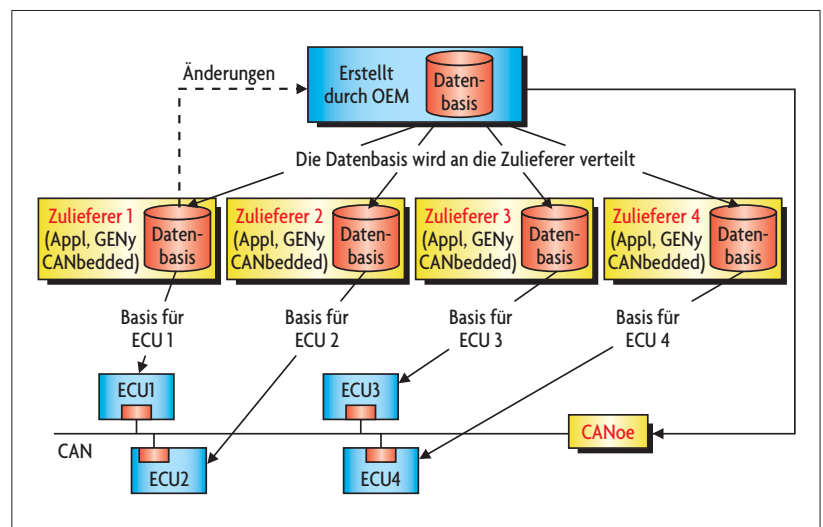


Bild 6. Die Standard-Software-Komponenten des CANbedded-J1939-Pakets führen rasch zu Entwicklungsergebnissen, ohne dass sich Entwickler mit allen Details des J1939-Standards befassen müssen. Eine zentral verwaltete Datenbank vermeidet Mehrfachentwicklungen und erlaubt die Arbeitsteilung.

zu einem Großteil von der Notwendigkeit, sich mit allen Details des J1939-Standards auseinanderzusetzen, und vermeiden Mehrfachentwicklungen. Im Mittelpunkt steht eine vom OEM verwaltete zentrale Datenbasis, in der alle elementaren Informationen der Steuergeräte-Kommunikation enthalten sind. Die Informationen werden – je nach Bedarf – an die beteiligten Partner verteilt, so dass sich eine flexible Arbeitsteilung zwischen OEM, den Netzwerkspezialisten

von Vector sowie den Zulieferern ergibt (**Bild 6**). Letztere können mit dem Konfigurationswerkzeug GENy ihre spezifischen Einstellungen und Parametrierungen vornehmen. Das Ergebnis ist eine Reduzierung von Kosten und Zeit für Implementierung und Test, Kompatibilität auf dem CAN-Bus aufgrund einer unmissverständlichen Signalinterpretation und Integrität sowie auch Flexibilität im J1939-Kommuni-

kations-Stack. Unterstützt werden alle relevanten Mikrocontroller bei geringem Bedarf an ROM- und RAM-Kapazität sowie mit hoher Laufzeiteffizienz. *ha*

Links

- [1] J1939-Lösungen der Vector Informatik GmbH: www.j1939-solutions.de
- [2] Download der Vorträge des „J1939 User Day“: www.vector-informatik.de/ud



**Dipl.-Ing. (FH)
Peter Fellmeth**

studierte an der FH Esslingen im Fachgebiet Technische Informatik mit Schwerpunkt auf Automatisierungstechnik. Als Teamleiter und Produktmanager bei der Vector Informatik GmbH ist er verantwortlich für die Entwicklung von Produkten und kundenspezifischen Projekten im Umfeld von J1939, ISOBUS, Ethernet und DeviceNet.



**Dipl.-Ing. (FH)
Thomas Löffler**

studierte Automatisierungstechnik an der Fachhochschule in Reutlingen. Er arbeitet seit 2000 bei der Vector Informatik GmbH, zu-nächst im Umfeld DeviceNet und seit 2002 im Bereich J1939 und ISOBUS. Seine Schwerpunkte sind Konfigurations- und Generierungswerkzeuge für Embedded Software sowie Betreuung von Kundenprojekten, Produkt- und Protokollschulungen.