

DIAGNOSE - VON DER ENTWICKLUNG
BIS ZUM SERVICE

Fahrzeugdiagnose — vom lästigen Übel zum gewollten Muss (Teil 1)

Die Fahrzeugdiagnose befindet sich bereits seit einigen Jahren in einem rapiden Umbruch. Die Anzahl der Steuergeräte im Fahrzeug steigt nach einer kurzen Phase der Konsolidierung wieder stetig an und bewegt sich nun mit großen Schritten auf 100 ECUs im Fahrzeug zu. Parallel dazu steigt auch die Komplexität sowohl der Funktionen in einer ECU als auch die des Fahrzeugnetzwerks selbst erheblich. Die Diagnose stellt damit eine eigene Klasse von Anforderungen an das Gesamtfahrzeug dar, die parallel zu den eigentlichen Funktionen implementiert werden muss.

In Mittelklassefahrzeugen findet man heute bereits vier Bussysteme, in der Oberklasse kommen noch mal zwei bis drei weitere hinzu, die alle untereinander vernetzt werden müssen. Für die Diagnose ergeben sich daraus mehrere Konsequenzen: Eine reine Beschränkung der Diagnose auf die vom Gesetzgeber vorgeschriebenen Umfänge bezüglich der Einhaltung von Abgasstandards genügt seit Langem nicht mehr. Eine Reparatur von Fahrzeugen ist in einer Werkstatt heute ohne Diagnose nicht mehr möglich. Gleiches gilt für die Fahrzeugproduktion. Und um die Komplexität beherrschen zu können, muss die Diagnose sehr früh im Entwicklungsprozess integriert und als begleitender Prozess durch die gesamte Entwicklung mitgezogen werden – beim Fahrzeughersteller, aber auch unter Einbeziehung der beteiligten Steuergeräteelieferanten.

Damit das Zusammenspiel aller Beteiligten funktioniert, sind Standards unabdingbar. Sie stellen das Ineinandergreifen von Daten und Tools an den Schnittstellen von Steuergeräteentwicklung, Prüffeld, Produktion und der Serviceorganisation sicher und ermöglichen dadurch die Konzentration auf die eigentliche Aufgabe. Im Folgenden werden die zugrundeliegenden Standards und die Absi-

cherung der Diagnose im Entwicklungsprozess ausführlich beleuchtet.

Entlang der Wertschöpfungskette des Fahrzeugs spielt die Diagnose heute in allen Bereichen von der Entwicklung über die Produktion bis in den Service eine Rolle. Entscheidend dabei ist, dass die Mechanismen der Diagnose – die Kommunikation eines Testgeräts außerhalb des Fahrzeugs mit einem Steuergerät – heute für eine ganze Reihe von zusätzlichen Aufgaben verwendet werden (**Bild 1**).

Entwicklungsbereich

In der Steuergeräteentwicklung wird die Diagnose für die nachfolgenden Bereiche zur Verfügung gestellt. Die Basis ist die Diagnosekommunikation, bei neuen Steuergeräten in der Regel auf Basis des Standards UDS. Die klassische Diagnose als Funktion im Steuergerät dient der Erkennung von unerwartetem Verhalten innerhalb des Steuergeräts und in seiner Umgebung, einer Bewertung und dem Eintragen dieser Auffälligkeiten im Fehlerspeicher. Von dort können die Einträge dann durch ein externes Testsystem ausgelesen und weiterverarbeitet werden. Neben den klassischen Fehlerspeicherfunktionen – dazu gehören ebenso die Ermittlung von Randbedingungen für den Feh-

lereintrag („Umgebungsbedingungen“) und das Löschen des Fehlerspeichers – kommt dieselbe Basisfunktionalität heute auch für verwandte Anwendungsfälle zum Einsatz. Beispiele sind:

- Flash-Programmierung: Austausch von Code oder Daten im Steuergerät.
- Varianten-Codierung: Anpassung des Verhaltens eines Steuergeräts entsprechend gesetzlicher Vorschriften (z. B. Tagfahrlicht in Skandinavien) oder der kommerziellen Randbedingungen („SW als Produkt“).
- Routinen starten: Funktionen des Steuergerätes werden von außen angestoßen und laufen dann selbstständig ab, beispielsweise ein Selbsttest.
- Messen: Auslesen der durch das Steuergerät mithilfe von Sensoren ermittelten Größen.

Die Diagnosekommunikation wird also immer dann eingesetzt, wenn ein präziser Daten- und Freigabeprozess nötig ist (Flash-Programmierung, Varianten-Codierung) oder sie einen einfachen und preisgünstigen Zugang zur Blackbox Steuergerät erlaubt.

Im Entwicklungsbereich kommt die Diagnose in zahlreichen Anwendungen zum Einsatz, sei es bei der Kommunikationsentwicklung, der Entwicklung von Steuergerätefunktionen, der Entwicklung von Flash- und Variantencodierfunktionen, im HiL, in Systemprüfplätzen oder bei der Steuergeräteintegration. Anschließend wird auch im Fahrzeugtest vielfältig darauf zurückgegriffen.

Produktion

Gerade der einfache Zugang zum Steuergerät ermöglicht in der Produktion eine Vielzahl von Anwendungen, die sonst nur mit komplexen Lösungen kostenintensiv zu realisieren wären. Die heutigen mechatronischen Systeme können ohne Zugang zur Elektronik kaum einer Prüfung unterzogen werden, sollten andererseits aber so früh wie möglich getestet werden. Auch die Variantencodierung des Fahrzeugs, die Schlüsselprogrammierung und die Initialisierung der Wegfahrsperr werden hier durchgeführt. Schließlich erfolgt im „End-of-line“-Test eine finale Kontrolle der Fahrzeugfunktionen und das Löschen des Fehlerspeichers.

Eine besondere Herausforderung stellt in der Produktion die Unterscheidung der „echten“ Fehler von den „systematischen“ Fehlern dar. Letztere ergeben sich aus der Tatsache, dass die Steuergeräte zahlreiche Fehlereinträge führen, die sich aus der zum Verbauezeitpunkt fehlenden Systemumgebung ergeben. Beispielsweise werden Fehler eingetragen, wenn auf dem CAN-Bus ein Signal nicht verfügbar ist, das von einem noch nicht verbauten Steuergerät gesendet werden sollte. Der Eintrag ist selbstverständlich richtig, er verweist aber eben nicht auf ein fehlerhaftes (Teil-)System.

Service

Die Ursprünge der Diagnose finden sich im Service, genauer in der nüchternen Erkenntnis, dass ein Fahrzeug auf der elektronischen Basis von aktuell 30 bis 80 Steuergeräten, bei der eine Vielzahl der Fahrzeugfunktionen in Software realisiert sind, in einer Werkstatt nicht mehr repariert wer-



den kann. Dem Fachmann in der Werkstatt muss für den Fall, dass der Kunde unerwartete Symptome meldet, der passende „Schraubenschlüssel“ zur Hand gegeben werden. Das entsprechende Werkzeug ist der Service-Tester. Es verbindet die gemeldeten Symptome mit Fehlereinträgen in den Steuergeräten und Messwerten, die parallel ermittelt werden können. Expertensysteme kombiniert mit der Erfahrung des Fachmanns in der Werkstatt führen dann in aller Regel zu einer Reparaturempfehlung. Zusätzlich ist es in der Regel möglich, Steuergeräte mit neuen Software-Ständen zu programmieren.

In allen genannten Anwendungsfällen kommen heute ODX-Daten zum Einsatz. Außer in Testsystemen werden diese auch für die Parametrierung von Steuergeräten und für die automatische Erzeugung von Testprozeduren verwendet. Es liegt auf der Hand, dass die Qualität der ODX-Daten – alleine, aber auch in Verbindung mit dem Steuergerät – für die Gesamtqualität eine entscheidende Einflussgröße darstellt. Neben der Durchführung von intensiven Reviews wird dies durch eine aufwendige Datenverifikation erreicht. Die Datenverifikation ist ein wichtiger Anwendungsfall eines modernen Entwicklungstesters. Darüber hinaus kommt dieser im ganzen Entwicklungsprozess zum Einsatz: als Testumgebung für die wichtigen Steuergerätefunktionen Diagnosekommunikation, Diagnosefunktion, Flash-Programmierung und Variantencodierung. Auch als preiswerter „Debugger“ wird der Entwicklungstester immer wieder eingesetzt. In späteren Phasen spielt er eine große Rolle bei der Fahrzeugüberprüfung in der Versuchswerkstatt und teilweise sogar im Fahrversuch beim Test einzelner Fahrzeugfunktionen und in der Fehleranalyse, speziell bei Problemen in der Kommunikation. Und auch bei der Prüfvorbereitung spielt der Entwicklungstester als Inbetriebnahmewerkzeug eine große Rolle. Die Kombination aus ODX-Daten und Steuergerät in Verbindung mit dem D-Server stellt für den Testingenieur im Wesentlichen eine Blackbox dar, die für ihn nur schwer in Betrieb zu nehmen ist. Mit dem Entwicklungstester kann hier sehr einfach die Kommunikation in Betrieb genommen werden und das korrekte Verhalten des Testablaufs simuliert werden.

OB

Die On-board-Diagnose stellt einen besonderen Bereich der Diagnose dar, da sie vom Gesetzgeber vorgeschrieben ist. Er verlangt von den Fahrzeugherstellern – ursprünglich nur von den Pkw-Herstellern, inzwischen aber auch von Lkw-Herstellern –, dass alle abgasrelevanten Systeme kontinuierlich einem Selbsttest unterzogen werden. Im Falle einer Auffälligkeit muss dies dem Fahrer durch eine Warnleuchte unverzüglich mitgeteilt werden, damit er das Problem beheben lassen kann. Für den Fall Kalifornien ist dieses „kann“ insofern eine Verniedlichung, da der Fahrer empfindliche Strafen zahlen muss, wenn er das Abgassystem nicht reparieren lässt. Dieser Selbsttest wird durch die Forderung unterstützt, dass jederzeit mit einem einfachen Dia-

gnosetester, dem sogenannten Scantool, die Erkenntnisse der On-board-Diagnose ausgelesen werden können. Sowohl eine funktionierende On-board-Diagnose als auch eine funktionierende Kommunikation mit den Scantools stellen für die Hersteller eine Marktzugangsvoraussetzung dar. Entsprechend genau müssen die Tests dieser Funktionalitäten sein. Diese sind mit der Fahrzeugzulassung auch nicht beendet, da eine Häufung von Meldungen bei bereits verkauften Fahrzeugen schnell auch zu Fahrzeugrückrufen führen kann.

Anforderungen an Diagnosesysteme

Wenn man sich moderne Testsysteme im Automotive-Umfeld anschaut, erkennt man generelle Blöcke, die immer wiederkehren. Zunächst ist entscheidend, dass man in den meisten Fällen ein Erstellsystem und eine Laufzeitumgebung vorfindet. Das Erstellsystem dient der Konfiguration des Laufzeitsystems, in ihm konfiguriert der Testingenieur Testabläufe und die Visualisierung für die spätere Verwendung. In vielen Fällen wird auch die Dokumentation des Tests bereits hier definiert. Das Erstellsystem arbeitet oft in einem Administratormodus. In der Laufzeitumgebung sind die Eingriffsmöglichkeiten des Testers häufig stark eingeschränkt. Der Tester hat nur noch den Anwendungsmodus im Zugriff. Das Ziel ist, dass er sich nur noch auf die Testumgebung konzentriert und das Tool mit wenigen Handgriffen funktioniert.

Die Testsysteme haben für die Konfiguration und Testdurchführung Eingangsdaten, die Konfigurationen und Randbedingungen parametrieren. Standardisierte Eingangsgrößen sind die heute geforderten ODX-Daten und zunehmend auch OTX-Daten (**Bild 2**). Daneben kommen häufig proprietäre Daten zum Einsatz, zum Beispiel Logistikinformationen wie Freigabestände und Verbaulisten. Diese Größen sind stark prozessabhängig, teilweise auch infrastrukturabhängig und können schlecht standardisiert werden.

Daneben werden auch entsprechende Ausgangsdaten erzeugt. Diese können ebenfalls standardisiert oder proprietär sein. In der Diagnose gibt es heute, anders als in der Messtechnik, (noch) kein standardisiertes Ausgabeformat. Ein typisches Ausgabeformat ist die Testdokumentation, in

der erfolgreiche und fehlgeschlagene Testfälle aufgezeichnet sind. Diese können entweder durch Spezialisten analysiert oder später für Regressionstests wieder ins System zurückgespielt werden. Auch Fehlerstatistiken werden vielfach mit den Testergebnissen erzeugt.

Anforderungen an Entwicklungstester

Speziell an Entwicklungstester, die ja im gesamten Prozess zum Einsatz kommen, müssen noch eine Reihe zusätzliche Anforderungen gestellt werden. Dies liegt in erster Linie an den Einsatzfällen, aber auch an den zugrundeliegenden Verarbeitungsschichten. Insbesondere werden ODX-Daten durch einen sogenannten D-Server verarbeitet. Dieser stellt die Daten aufbereitet zur Verfügung. Es ist aber für das Testsystem kaum möglich festzustellen, welchem Zweck die Daten dienen. D. h., Diagnosedienste für die Kommunikationskontrolle unterscheiden sich in der Praxis an der Schnittstelle meist nicht von Diagnosediensten zum Messen oder zum Parametrieren eines Steuergerätes. Der Diagnosetester muss den Anwender aber durch anwendungsspezifische Darstellungen unterstützen, die entsprechend konfiguriert werden müssen. Erst dadurch wird die Darstellung mit den passenden Diagnosediensten verbunden.

Die Arbeit am Fehlerspeicher stellt vielleicht die wichtigste Diagnosefunktion dar. Er muss gelesen, aber auch gelöscht werden, um den korrekten Wiedereintrag eines Fehlers prüfen zu können. Je nach Anwender interessieren die Fehler des ganzen Fahrzeugs oder eines Steuergerätes, nur die Liste der Fehler oder die Fehler mit den abgespeicherten Umgebungsbedingungen und den Statusinformationen. Gerade die Statusinformationen sind für eine tiefer gehende Analyse von Bedeutung, da sie den Fehler näher beschreiben. Man erfährt beispielsweise, ob ein Fehler dauerhaft eingetragen wurde oder nur temporär, ob er sporadisch auftritt oder dauerhaft. Eine weitere wichtige Information ist das Readiness-Flag. Es gibt an, ob die zur Erkennung des Fehlers im Steuergerät implementierte Routine bereits vollständig durchlaufen wurde. Für eine ganze Reihe von Fehlern müssen dazu Vorbedingungen einge-

halten werden, wie „Motortemperatur muss Mindestwert erreicht haben“, „vorgegebene Drehzahl muss Wert überschritten haben“, usw. Die Anforderungen an die Visualisierung der Flash-Programmierung sind gänzlich anderer Natur. Diese beschränkt sich im Wesentlichen auf die Auswahl der Programmierdaten (z. B. Programmcode, Kennfelder,...) einen Start-Button und möglichst eine Fortschrittsanzeige, da bei Datengrößen von mehreren MByte durchaus nennenswerte Programmierzeiten entstehen. Die eigentliche Herausforderung steckt bei der Flash-Programmierung im Ablauf. Hier wird zunächst eine Authentifizierung durchgeführt, in die Programmiersession geschaltet, die restlichen Busteilnehmer aufgefordert, im Folgenden keine Busfehler einzutragen und anschließend die Buskommunikation zwischen den Steuergeräten beendet, um ausreichend Bandbreite zur Verfügung zu haben. Danach

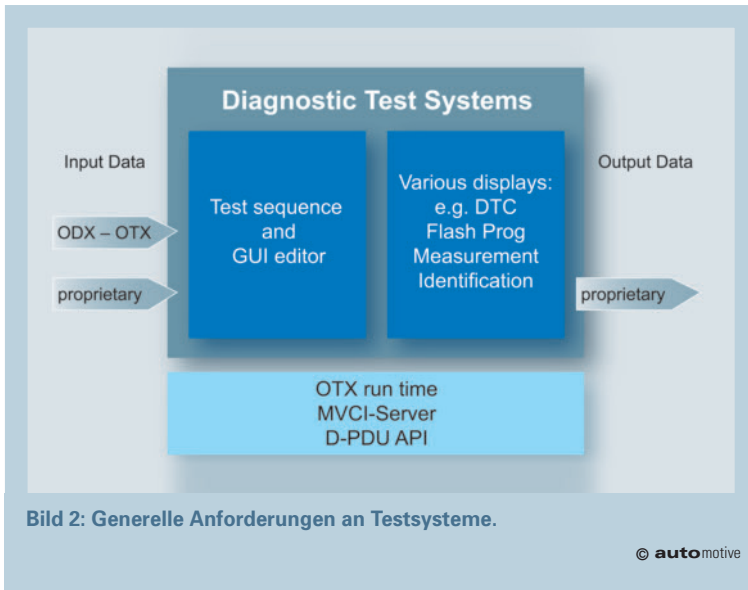


Bild 2: Generelle Anforderungen an Testsysteme.

© automotive

beginnt die möglichst performante Übertragung der Daten. Am Ende werden Checksummen überprüft, das programmierte Steuergerät reinitialisiert und die Buskommunikation wieder freigeschaltet. Abschließend sind manchmal noch Systemanpassungen durchzuführen.

Beim Messen muss hauptsächlich unterschieden werden zwischen graphischer Darstellung der einzelnen Messwerte und textueller Darstellung. Bei der graphischen Darstellung sind die Grenzen im Wesentlichen durch die Bedienbarkeit gesetzt, von der Art der Darstellung (Zeigerinstrument, Balkendiagramm, x-t-Darstellung, ...) über die verwendeten Farben bis zum Darstellungsbereich (Wertebereich, Gültigkeitsbereich mit Farbumschlag, ...) ist hier fast alles möglich. Daneben müssen häufig Messwertblöcke dargestellt werden. Diese werden auf einmal vom Steuergerät ausgelesen, um bei zusammenhängenden Größen keine Latenzen durch die Busübertragung in die Messung einfließen zu lassen.

Bei der Ausführung von Steuergeräteroutinen ist wiederum ein sehr breites Anwendungsspektrum abzudecken. Das eine Ende wird durch Steuergeräteroutinen gebildet, bei denen das Feedback allein von Aktuatoren gegeben wird. Beispiele sind der Zeigertest am Kombiinstrument oder Ventile am ABS-Steuergerät, die öffnen und schließen. Für diese Art von Routinen muss der Tester eigentlich nur einen Start-Button anbieten sowie vielleicht ein Beschreibungsfeld für den Test. Das entgegengesetzte Ende stellen Selbsttests dar, die als Ergebnis eine Ergebnisliste über den Bus zurückliefern. Hier muss der Tester die relevanten Teile der Steuergeräteantwort auf dem Bildschirm darstellen. Ähnlich funktioniert auch die wichtige Darstellung der Fahrzeug- oder Steuergeräteidentifikation, die in der Regel aus einer Liste von Informationen wie der VIN (Vehicle Information), HW-Stand, SW-Stand, usw. gebildet wird.

Der Einsatzfall Datenverifikation erfordert einen erheblich weiteren Darstellungsraum. Der Zugriff sowohl auf die Parametrierung von Diagnosediensten als auch auf die Ergebnisse muss so vollständig sein, dass alle Methodiken, die in automatischen Tests angewendet werden, mit dem Entwicklungstester auch manuell überprüft werden können. Im Entwicklungstester müssen auch Werte ein-

stellbar sein, die eigentlich nicht erlaubt sind, um das korrekte Verhalten an den Grenzen verifizieren zu können. Bei der Überprüfung ist besonders wichtig, dass nicht nur die Werte analysiert werden können, sondern auch die Strukturierung der Ergebnisse. Diese ist für den Zugriff durch eine Testautomatisierung eine Fehlerquelle, die nicht unterschätzt werden darf. Es sind übrigens nicht alle Tests automatisierbar, da zwar die Plausibilität automatisch geprüft werden kann, nicht aber die richtige Umsetzung von hexadezimalen (Bus-)Daten in symbolische Daten. Dies ist eine Aufgabe für Spezialisten, die massiven Einfluss auf die Gesamtqualität des Systems hat.

Varietencodierung

Mit Hilfe der Variantencodierung werden Funktionen im Steuergerät ein- und ausgeschaltet. Dies hat erhebliche Kostenvorteile, da nicht für jede Variante der Steuergerätekodierung einzeln frei getestet werden muss. Die Funktion Variantencodierung muss dafür umso genauer getestet werden. Dies ist mit hohem Aufwand verbunden, da die einzelnen Codierbits nicht unabhängig sind. Dies gilt innerhalb eines Steuergeräts, umso mehr aber auch für das Gesamtfahrzeug. Offensichtlich wird dies an der Kombination von Motor und Getriebe. Die Motorleistung wird heute in der Regel für einen Motor codiert, muss aber mit bestimmten Getrieben kombiniert werden, da das eine Getriebe für ein Drehmoment nicht spezifiziert ist, andererseits bei hohen Leistungen häufig nur noch ein Automatikgetriebe angeboten wird. Der Diagnosetester muss für die Funktionsprüfung die Kombination verschiedener Codierungen möglichst einfach erlauben. Gleichzeitig sollte auch hier das Setzen fehlerhafter Kodierkombinationen möglich sein, um das Fehlverhalten überprüfen zu können.

Restbussimulation

Eigentlich kein Thema der Diagnose ist die Restbussimulation. Dennoch hat sie einen großen Einfluss, da in vielen Fällen – insbesondere, wenn nur ein einzelnes Steuergerät oder ein Teilnetzwerk in Betrieb genommen werden sollen – die Diagnose nicht funktioniert. Oftmals ist die Diagnose in einem Steuergerät nicht aktiv, solange bestimmte Signale nicht auf dem Bus übertragen werden. Ein typisches Beispiel ist die Zündungserkennung, die von einem Steuergerät durchgeführt wird und dann als Signal allen anderen ECUs auf dem CAN-Bus zur Verfügung gestellt wird. Gleiches gilt für Signale, die für eine plausible Diagnose nötig sind (z. B. einem Geschwindigkeitssignal). In jedem Fall muss der Entwicklungstester diese Signale simulieren können, indem er einzelne CAN-Nachrichten auf den Bus zyklisch sendet. Sinnvollerweise sollten die einzelnen Signale auch änderbar sein, da sich beispielsweise das Diagnoseverhalten von niedrigen zu hohen Fahrzeuggeschwindigkeiten ändern kann.

Auch die Anforderung Analysefähigkeit erzeugt Anforderungen an den Entwicklungstester. Zum einen dient er der Datenvisualisierung. Er muss also die Daten geeignet auf-

bereiten und Messwerte auf passenden Instrumenten darstellen, den Fehlerspeicher auflisten und die Daten zur Flashprogrammierung so präsentieren, dass sie fehlerfrei ins Steuergerät programmiert werden können. Zum anderen dient er aber der Inbetriebnahme und soll zu diesem Zweck einen möglichst genauen Zugriff auf die ODX-Informationen und eventuell auftretende Kommunikationsfehler ermöglichen. Falls in der Kommunikation selbst Probleme auftreten, muss aber auch der Busverkehr so dargestellt werden, dass das jeweilige Problem effizient auffindbar und dokumentierbar ist.

Für die Prüfung der OBD-Funktion müssen die OBD-Modes, es handelt sich dabei um spezielle Diagnosedienste, in der jeweils adäquaten Form präsentiert werden. Grob gegliedert dienen sie zum Lesen der Messwerte, zur Fehlerspeicherungsbearbeitung und zur Auswertung von Testergebnissen.

Die Adressierung erfolgt dabei immer auf die Funktion „OBD“, es können also mehrere Steuergeräte betroffen sein. Das VCI muss dazu die funktionale Adressierung beherrschen, das Diagnosesystem mit mehreren Antworten von verschiedenen Steuergeräten umgehen können. Welche Teilfunktionen bei den einzelnen Modes unterstützt werden, wird bitcodiert von den Steuergeräten ans Testsystem gemeldet. Hier ist es besonders wichtig, dass der Tester erlaubt, Anfragen an die Steuergeräte zu stellen, die erst einmal falsch erscheinen, da die Scan-tools die Spezifikation teilweise unterschiedlich interpretieren.

Welche wichtigen Standards muss man beachten?

Heute in der Diagnose verwendete Standards können grundsätzlich in drei Kategorien eingeteilt werden: Protokolle, Datenbeschreibungen und Programmierschnittstellen. Die älteste Kategorie stellt sicher die der Protokolle dar. Eingeführt ursprünglich um die Forderung des Gesetzgebers nach einer Überwachungsmöglichkeit des Abgasverhaltens zu genügen, erfüllen diese Protokolle heute zahllose weitere Aufgaben. Die Vielzahl der herstellerspezifischen Protokolle wurde heute durch standardisierte Protokolle abgelöst, allerdings sollte die Bedeutung der Altprotokolle für Servicetester nicht unterschätzt werden, schließlich sind die Fahrzeuge noch viele Jahre auf der Straße aufzufinden.

Im zweiten Teil dieses Beitrags werden die verschiedenen Protokolle und Programmierschnittstellen aufgezeigt. Schließlich wird am Beispiel des DTS-Monaco von Softing Automotive ein Entwicklungstester vorgestellt, der D-PDU API, ODX, MCD-3D und OTX beherrscht.



Markus Steffelbauer leitet das Produktmanagement und verantwortet Entwicklung und Marketing der Hardware- und Softwareprodukte bei der Softing Automotive Electronics GmbH. Daneben vertritt Steffelbauer seit vielen Jahren Softing in verschiedenen Standardisierungsgremien, seit 2008 als Vertreter im ASAM TSC (Technical Steering Committee).

@ Softing Automotive Electronics GmbH
www.softing.com