

# Aktuelle Trends bei der Steuergeräte-Reprogrammierung

## Flexibilität steht an oberster Stelle

Software-Aktualisierungen in der Werkstatt werden künftig zu einem selbstverständlichen Vorgang werden. Deshalb werden standardisierte Datenformate, leistungsfähige Werkzeuge zur Verwaltung von Flash-Daten und die Organisation der Programmieraufgaben immer wichtiger. Der vorliegende Beitrag beleuchtet diese und weitere Trends rund um die Reprogrammierung der Steuergeräte.

Von Peter Liebscher

Die Anzahl elektronischer Steuergeräte in den Automobilen steigt seit Jahren kontinuierlich an. Dabei verändert sich das Verhältnis von reprogrammierbaren zu nicht reprogrammierbaren Steuergeräten teilweise sehr sprunghaft. Bei DaimlerChrysler z. B. ist der Anteil der reprogrammierbaren Komponenten in den Jahren 2002 bis 2005 von etwa 30 % auf über 80 % bei der aktuellen S-Klasse gewachsen (Bild 1). Noch deutlicher als der Anstieg der Steuergeräte macht sich im Entwicklungsprozess die überproportional komplexer werdende Software be-

merkbar. Um Entwicklungsanstrengungen und Kosten beherrschbar zu halten, hat man unter anderem die neuen herstellerübergreifenden Standards HIS (Hersteller-Initiative Software) und AUTOSAR (Automotive Open System Architecture) ins Leben gerufen. Diese vereinfachen über einheitliche Schnittstellen den Datenaustausch zwischen Zulieferern, Entwicklern, Testern usw. Erste Pilot-Implementierungen gemäß AUTOSAR gibt es seit Mitte 2005; der HIS-Standard wurde Ende 2005 freigegeben.

Die Reprogrammierung der Steuergeräte, kurz: das „Flashen“ der nicht-flüchtigen Speicherbausteine, ist dabei zu einem wesentlichen Teil des Gesamtprozesses geworden, der keinesfalls isoliert betrachtet werden darf. Der rein technische Vorgang lässt sich recht schnell skizzieren: Via PC oder Laptop schreibt man mit Hilfe des Flash-Tools die Flash-Daten in die einzelnen Steuergeräte. Dabei dient CAN bzw. künftig vermehrt FlexRay als Physical Layer, auf dessen Basis die entsprechenden Transport- und Diagnoseprotokolle implementiert sind.

Die eigentliche Herausforderung besteht jedoch darin, Prozesssteuerung, Dokumentation und Organisation aller mit der Reprogrammierung in Verbindung stehenden Teilaspekte bestmöglich zu vereinen. Dies gestal-

tet sich in jeder Situation anders: vom Zulieferer, der einzelne Komponenten für verschiedene Kunden entwickelt, über den Automobilhersteller in den Entwicklungs-, Test- und Integrationsphasen oder der End-of-Line-Programmierung bis hin zum After-Sales-Flashing. In der Praxis ist deshalb ein Automatismus nicht einfach zu realisieren.

## Flash-Bundles: Abhängigkeiten berücksichtigen

Zahlreiche Abhängigkeiten gilt es zu beachten bzw. zu definieren. Welche Software-Stände setzen welche Hardware-Stände voraus – und umgekehrt? Welche Anforderungen stellen sich an andere Steuergeräte, beziehungsweise welche Auswirkungen hat es auf Steuergerät A, wenn Steuergerät B geflasht wird? Welche Gesamtfunktionalität soll jeweils implementiert werden? Für jedes Fahrzeug müssen die richtigen „Bundles“ zusammengestellt und alle Abhängigkeiten in einem Flash-Archiv gespeichert werden. Derzeit noch Zukunftsmusik, aber durchaus denkbar ist die Idee der Flash-Tankstelle. Sie würde Möglichkeiten bieten, neue Funktionalitäten via Internet herunterzuladen, vor Urlaubsfahrten ein paar PS mehr freizuschalten oder das Fahrzeug vor einem Eigentümerwechsel auf den neuesten Software-Stand zu bringen.

Um eine eindeutige Reproduzierbarkeit sicherzustellen und damit auch die Haftungsfrage zu klären, dürfen nur ermächtigte Personen oder Systeme zur richtigen Zeit das richtige Steuergerät in einem ganz bestimmten Fahrzeug flashen. Vor dem Flash-Vorgang muss das Fahrzeug in einen sicheren Zustand gebracht werden, durch Feststellen der Bremsen, Ausschalten des Motors usw.

Entwicklung, Dokumentation und Freigabe von Flash-Daten richten sich

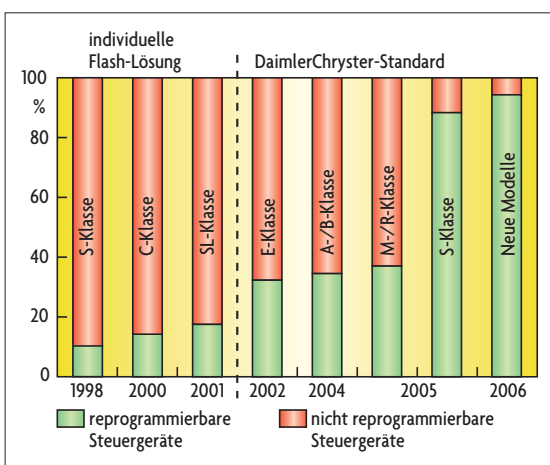


Bild 1. Der Anteil der reprogrammierbaren Steuergeräte bei DaimlerChrysler steigt rapide an. Den bisherigen Höhepunkt mit 80 % reprogrammierbaren Steuergeräten bildet die neue S-Klasse. (Quelle: DaimlerChrysler)

nach den jeweils gültigen Standards der Hersteller. Ein Fahrzeug-Flash-Konzept ist abhängig von der Fahrzeug-Architektur, dem Gateway-Routing, der Steuerung der Buskommunikation usw. Schickt man die Flash-Daten z.B. über diverse Gateways, muss sichergestellt sein, dass dort keine Informationen verlorengehen können. In jedem Fall ist eine Verifikation der Daten unmittelbar nach dem Flash-Vorgang unverzichtbar. Daimler-Chrysler speichert die Flash-Daten zusammen mit dem Flash-Job in einem Flash-Datencontainer. Für diese Aufgaben nutzt der Automobilhersteller die Tools CANDelaStudio und CANDelaFlash von Vector Informatik.

#### ■ Erstellen standardisierter ODX-Flash-Datencontainer

Ein Flash-Datencontainer speichert z.B. neben dem Hex-File alle herstellerspezifischen Metadaten, Datenformate und Wertebereiche. Für einen effizienten Einsatz sind außerdem Homogenität über Projektgrenzen hinaus sowie die Verwendung einheitlicher Standards notwendig. Das ODX-Flash-Format wurde von ASAM als Teil des ODX-Standards (Open Diagnostic Data eXchange) für den Austausch von Diagnosedaten definiert. Es speichert Informationen im XML-Format, kann die Diagnosedaten von ODX-D-Containern nutzen und ist bestens an die Prozessanforderungen der Hersteller anpassbar. So ist mit dem ODX-Flash-Format als künftigen Standard eine signifikante Verbesserung des gesamten Flash-Prozesses verbunden.

Das Tool CANDelaFlash bedient sich zur Erstellung prozesskonformer ODX-F-Datencontainer eines cleveren Vorlagenkonzepts: Fixe Projektdaten und herstellerspezifische Informationen sind bereits in der Vorlage abgelegt und müssen nicht jeweils neu eingegeben werden, was eine zügige und fehlerfreie Containergenerierung ermöglicht. Die mit CANDelaStudio erstellten Diagnosedaten lassen sich direkt wiederverwenden, und es steht ein Plug-in-Interface für Security-DLLs zur Verfügung, da CRC-Check, Signatur- und Schlüsselberechnungen bei jedem Hersteller unterschiedlich sind. Abgerundet werden die Möglich-

keiten durch Export- und Import-Funktionen für ODX-Flash-Daten.

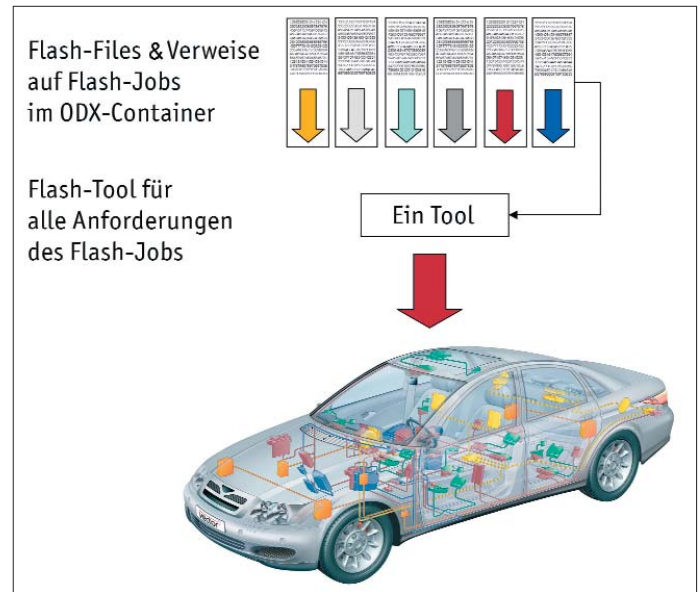
Neben der Erzeugung des Hex-Files bzw. des Flash-Datencontainers erfordern die Programmierung und die Verwaltung von Flash-Jobs viel Aufmerksamkeit. Fahrzeughersteller z.B. brauchen für jedes Flash-File einen speziellen Flash-Job, während Zulieferer wiederum für jeden Kunden einen anderen Job erstellen müssen. Die benötigten Flash-Jobs sind sehr vielfältig und unterscheiden sich von Fahrzeug zu Fahrzeug und von Hersteller zu Hersteller. Da eine umfassende Vereinheitlichung sämtlicher Flash-Formate, Prozesse und Steuergeräte auf absehbare Zeit Wunschtraum bleiben wird, bieten die Werkzeuge von Vector Informatik zur klassischen hardcodierten Methode eine effiziente Alternative. Flash-Files und Verweise auf die zugehörigen Flash-Jobs werden gemeinsam in einem Datencontainer mit Standardformat gespeichert. So kann man durch den Einsatz eines einzigen Flash-Tools den Anforderungen aller Flash-Jobs gerecht werden (Bild 2).

#### ■ Programmierprozess: hardcodiert oder datengesteuert?

Bei der herkömmlichen hardcodierten Methode ist der Ablauf des Flash-Prozesses fest im Flash-Tool integriert. Man entwickelt das Tool mit einem normalen C/C++-Compiler, der keine diagnose- und flash-spezifische Funktionalität aufweist. Zur Programmierung jedes weiteren Steuergeräts ist ein individuelles Tool erforderlich. Am Ende müssen Tool, Hex-File und Steuergerät genau zusammenpassen. Diese starre Vorgehensweise bietet keine Möglichkeit zur flexiblen Konfiguration des Flash-Prozesses, um die steuerungsspezifischen Anforderungen zu erfüllen, wie beispielsweise die

Einleitung des Flash-Ablaufs, die sich von Projekt zu Projekt unterscheiden kann. Beim eigentlichen Flashen ist darüber hinaus noch die unterschiedliche Handhabung von Authentifizierung, Verschlüsselung, Fingerprint und CRC-Check zu berücksichtigen.

Die datenbasierte Methode dagegen liefert größtmögliche Flexibilität bei den genannten variablen Parametern und Abhängigkeiten. Als Grundlage



! Bild 2. Durch die gemeinsame Speicherung der zu programmierenden Daten und der Verweise auf die zugehörigen Flash-Jobs als Datencontainer im Standardformat kann ein Tool alle Flash-Jobs ausführen.

für die Erstellung von Flash-Jobs mit den Werkzeugen CANape Graph und CANDito dienen die mit CANDelaStudio und CANDelaFlash erstellten und in ODX-D- und ODX-F-Containern abgelegten Diagnose- bzw. Flash-Daten. Durch die Verwendung des flash- und diagnosespezifischen Entwicklungssystems einschließlich einer problemorientierten, an C angelehnten Skriptsprache ergeben sich klare Vorteile. CANape und CANDito ermöglichen Debug- und Protokollanalysen und unterstützen so die effiziente Entwicklung der Flash-Jobs. Die fertigen und getesteten Skripte werden zusammen mit den Diagnosedaten in einem Diagnose-File abgelegt.

Die optimal aufeinander abgestimmten Werkzeuge des Stuttgarter Spezialisten verwalten und speichern die Daten redundanzfrei in denselben Datenbasen; so sind Inkonsistenzen prinzipiell ausgeschlossen. Während

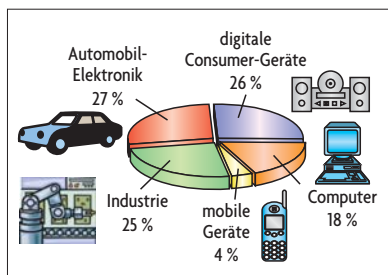
beim klassischen Flash-Prozess der Übergang von der Vorserie zur Serienproduktion mit erheblichem Aufwand verbunden ist, bietet das datenbasierte Konzept durch frühzeitiges Entwickeln und Testen der serienrelevanten Flash-Tools eine sanfte Migration vom Anfang der Entwicklung bis zur Serienprogrammierung.

**Post-Build-Prozess: Daten nachträglich verändern**

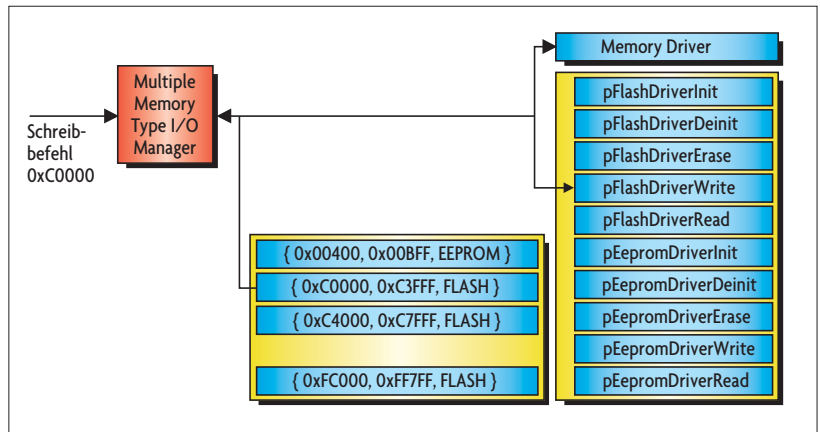
Mit CANape Graph und CANDito lassen sich auch so genannte Post-Build-Prozesse elegant beherrschen, wenn man Flash-Daten im Nachhinein verändern will. Das ist in solchen Fällen sinnvoll, wenn nur wenige Informationen zu modifizieren sind. Man spart sich dann den kompletten Durchlauf von Compiler und Linker. Grundsätzlich wird zwischen Offline- und Online-Post-Building unterschieden.

Beweggründe für Offline-Post-Build-Aktivitäten gibt es beispielsweise, wenn Parameter gezielt verändert werden müssen oder wenn Hersteller bestimmte CAN-IDs durch andere bzw. eigene IDs ersetzen wollen. Die genannten Werkzeuge von Vector Informatik überführen Parameter-Sets, die sich aktuell im Steuergerät befinden oder als Dateien abgelegt sind, in das neue Hex-File. Mit Hilfe der A2L-Datei lassen sich die neuen Parameter korrekt auf ihre Adressbereiche map-pen. Gleichzeitig kümmern sich die Systeme darum, dass neben den veränderten Parametern auch Checksummen neu berechnet und an entsprechender Stelle im Hex-File aktualisiert werden.

Während durch Offline-Modifikationen bestehende Hex-Files direkt



**Bei Mikrocontrollern mit integriertem Flash-Speicher hat sich die Automotive-Industrie zum größten Abnehmer entwickelt.** (Quelle: Renesas Technology Europe GmbH)



**Einheitlicher Zugriff auf unterschiedliche Arten von Flash-Speicher mit dem „Multiple Memory Type I/O Manager“.**

verändert werden, geht es bei der Online-Modifikation um die Berechnung zusätzlicher Fingerprint- und Signatur-Informationen zur Flash-Laufzeit. Meistens sendet das Flash-Tool mit den Hex-Daten Fingerprints an das Steuergerät, wo Angaben über Datum und Identifikation der Flash-Station enthalten sind. Da man Signaturen niemals im Hex-File speichert, sondern zur Laufzeit extern berechnet, gibt es in CANape Graph und CANDito die Möglichkeit, eigene DLLs einzubinden. Diese dienen z.B. zur nichtöffentlichen Schlüsselberechnung im Seed- und Key-Verfahren und zur Signaturberechnung. Auch bei den Post-Build-Aktivitäten profitiert der Anwender von der auf Flash- und Diagnoseaufgaben optimierten Entwicklungsumgebung und Skriptsprache.

**Unterschiedliche Speichertypen rationell verwalten**

Durch die Verschiedenartigkeit der verfügbaren nichtflüchtigen Speichertypen nimmt ferner deren rationelles Verwalten einen wichtigen Platz ein. Die steigende Komplexität der Steuergeräte im Automobil geht einher mit der Verwendung von unterschiedlichen Speichertypen und Multiprozessorsystemen. Die gebräuchlichen nichtflüchtigen Speichereinheiten haben teils sehr unterschiedliche physikalische Charakteristika. Zu den wesentlichen Unterscheidungsmerkmalen nichtflüchtiger Speichertypen zählen die Größe des Schreibsegments, die Größe des Löschsegments, die maximale Anzahl der Programmierzyklen sowie die

zum Programmieren und Löschen benötigten Zeiten.

Interessant ist, dass der Automotive-Markt nach Informationen von Renesas mit 27 % das größte Volumen von Controllern mit integriertem Flash-Speicher einnimmt (Bild 3). Die Anteile der Consumer-Elektronik und des Industriesektors liegen allerdings nur geringfügig darunter. Die gegenwärtigen Flash-Speicher nutzen Technologien mit NOR Stacked Gate und MONOS (Metal Oxide Nitride Oxide Silicon). Eine Renesas-SH2A-32-bit-RISC-Implementierung mit MONOS-Flash-Technologie erreicht 80 MHz „single cycle random access“ und stellt aktuell bis zu 1,5 Mbyte Flash-Speicher zur Verfügung. Als On-Chip-Peripherie sind neben CAN weitere Module speziell für Anwendungen im Antriebsstrang vorhanden. Ab 2008 rechnet man mit der Einführung von neuen nichtflüchtigen Speichertechnologien auf Basis von MRAM, die sowohl Flash als auch RAM ersetzen könnten.

Mit dem Ziel einer einheitlichen Speicherverwaltung wurde von der HIS-Automotive-Gruppe (Hersteller-Initiative Software) ein Standard für das Memory-Driver-Interface definiert, der die zunehmende Unterstützung der Halbleiterhersteller findet. Das Interface stellt die Funktionen zum Initialisieren, De-Initialisieren, Löschen, Programmieren und Lesen der Daten zur Verfügung. Auf Basis des HIS-Interfaces kann durch einen „Multiple Memory Type I/O Manager“ der Zugriff auf unterschiedliche Speicher realisiert werden (Bild 4). Die Speicherkonfiguration lässt sich

komfortabel über das Vector-Tool GENy konfigurieren. Damit gewinnt der Anwender maximale Flexibilität beim Zugriff auf verschiedene Speichertypen.

Bei Konfigurationen mit internem und externem Flash-Speicher nutzt man den physikalischen Adressraum des Mikrocontrollers auch zum Ansprechen des externen Speichers. Dazu wird die Memory-Lookup-Tabelle für den externen Flash-Treiber erweitert und der Speicher direkt in den physikalischen Bereich des Mikrocontrollers eingeblendet. Anders verhält es sich bei externem EEPROM, das über serielle Schnittstellen wie SPI (Serial Peripheral Interface) angebunden ist. Da sich serielle EEPROMs nicht direkt in den physikalischen Adressbereich des Controllers einblenden lassen, verlagert man das SPI-EEPROM mit Hilfe der Memory-Lookup-Tabelle in einen virtuellen Adressraum. Ein ähnliches Problem stellt sich bei der Reprogrammierung von LIN-Slaves und vergleichbaren Subknoten, weil dort prinzipbedingt keine direkte CAN-Verbindung zur Verfügung steht. Hier wird der LIN-Slave beim Flashen in einen virtuellen Adressraum der Memory-Lookup-Tabelle des LIN-Masters gemappt.

### ■ Speziallösungen für Sonderfälle

Nicht immer reichen die Standardwerkzeuge zur Realisierung einer gewünschten Lösung aus, wie z.B. im Fall des Automobilzulieferers Conti-

ental, bei dem ein Mehrprozessorsystem mit Master- und Slave-Controller sowie externen Sensoren zu reprogrammieren ist. Während der Master-Controller direkt über CAN erreichbar ist, lassen sich Slave und Sensoren beim Flashen nur indirekt ansprechen. Um den gesamten Flash-Prozess unter den genannten Randbedingungen in den Griff zu bekommen, wurde von Vector Informatik eine spezielle Erweiterung implementiert. Sie ist in der Lage, die serielle IPC-Verbindung (Interprozessor-Kommunikation) zwischen Master und Slave beim Flashen zu nutzen.

In der Serienproduktion nimmt die Übertragung aller Flash-Daten je nach Zahl der Steuergeräte inzwischen eine volle Stunde und mehr in Anspruch. Aufgrund des weiter steigenden Flash-Daten-Aufkommens hat deshalb auch die Optimierung der Flash-Zeiten eine hohe Priorität. Dazu gehört einerseits die Bereitstellung hoher Bandbreite durch Versetzen möglichst vieler Buskomponenten in den Ruhezustand und andererseits die Verwendung hoher Taktfrequenzen. Auch über Datenkompression denken Fahrzeughersteller und Tool-Lieferanten derzeit nach.

Die bekannten Kompressionsmethoden lassen sich grob in Frequenzbereichs- und kopiergestützte Methoden unterteilen. Zur erstgenannten Gruppe gehören Verfahren nach Huffman, Burrows-Wheeler und das Arithmetic Coding, deren Kompressionsfaktor durch den Entropiefaktor begrenzt ist. Die kopiergestützten Ver-

fahren werden durch die LZ77-Familie (Lempel-Ziv) und LZW (Lempel-Ziv-Welsh) repräsentiert. Aufgrund der Codierung von Strings eignen sie sich gut, wenn identische Substrings häufig im Datenpaket vorkommen. Durch ihr vergleichsweise einfaches Prinzip ermöglichen die kopiergestützten Verfahren eine schnelle Dekompression. Nach wissenschaftlichen Untersuchungen würde sich zur Kompression von Flash-Daten eine Kombination des LZ77-Verfahrens mit einer arithmetischen Codiermethode am besten eignen, so dass das Datenvolumen bzw. die Flash-Zeiten sich bis etwa auf die Hälfte reduzieren. *jk*

### Weiterführende Links

[1] Download-Möglichkeit aller Vortragsfolien des Vector Embedded Symposiums: [www.vector-informatik.de/embedded](http://www.vector-informatik.de/embedded)



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

hat Nachrichtentechnik an der FH in Esslingen studiert. Seit 2002 betreut er als Business Development Manager bei der Vector Informatik GmbH die Produktlinie Embedded Software Components.

[peter.liebscher@vector-informatik.de](mailto:peter.liebscher@vector-informatik.de)