

© Softing Automotive

STEUERGERÄTE-PROGRAMMIERUNG

Flexibilität in der Flash-Programmierung meistern

Flash-Programmierung schafft heute eine enorme Flexibilität – über alle Steuergeräte hinweg und im gesamten Fahrzeuglebenszyklus. Durch den Einsatz geeigneter Systemarchitekturen lassen sich die entstehenden Herausforderungen bezüglich Performance, Prozessen und Datengrößen gut beherrschen.

Die Flash-Programmierung von Steuergeräten (ECUs) in Fahrzeugen hat sich heute in der Breite durchgesetzt: Die je nach Marktsegment zwischen 3 und 120 ECUs können in der Regel direkt oder über ein Gateway über einen Tester mit neuer Software versorgt werden. Diese Fähigkeit wird vielfältig genutzt, wobei aktuell noch ein Fahrzeuginterface angesteckt wird, um zwischen der Programmieranwendung und den Steuergeräten zu vermitteln. In Zukunft eröffnen sich durch Over-the-air-Updates noch weitergehende Möglichkeiten.

Freiheitsgrade im Lebenszyklus

Ein steigender Teil der Wertschöpfung heutiger Fahrzeuge erfolgt über Software, was sich durch weitere Verbreitung von Fahrerassistenzfunktionen noch beschleunigen wird. Dies hat auch Prozessvorteile, da Software einfach parametrierbar und dadurch leicht an verschiedene Randbedingungen anpassbar ist. Ein Beispiel ist die Variantencodierung, bei der eine Hardware und eine Software parametrierbar wird, um 150 PS und 190 PS als Leistungsabgabe darzustellen. In gleicher Weise werden zahllo-

se Fahrzeugfunktionen über Kennfelder und einzelne Anpassungswerte parametrierbar – und gegebenenfalls geändert. Dies wird sowohl am Anfang des Lebenszyklus eingesetzt, wenn das Verhalten abgestimmt wird, als auch in späteren Phasen. Dadurch kann einem geänderten Geschmack der Kunden Rechnung getragen werden, indem beispielsweise das Soundsystem durch eine leichte Bassanhebung an aktuelle Hörgewohnheiten angepasst wird. Es kann aber auch einem unerwarteten Alterungsverhalten von Sensoren entgegenwirken und die entsprechende

Kennlinie angepasst werden. Genauso können ganze Steuererätefunktionen nachgerüstet werden. Voraussetzung ist natürlich, dass es sich entweder um eine reine Softwarefunktion handelt oder die benötigte Hardware bereits verbaut ist. Ist dies gegeben, muss lediglich die neue Software programmiert werden. Mit Fehlern in der Software kann ebenso verfahren werden. Der Fehler wird gefixt, ein neues Software-Release zur Verfügung gestellt und beizeiten eingespielt.

Anwendung findet dies über den gesamten Lebenszyklus. In der Entwicklung werden Steuererätefunktionen codiert, Varianten gebildet, Funktionen parametrisiert, Fehler behoben. In der Produktion muss der jeweils aktuelle Software-Stand in die Steuergeräte programmiert werden und nach Auslieferung werden in der Werkstatt Bugfixes, Anpassungen und neue Funktionen in Kundenfahrzeuge integriert (siehe Bild 2).

Herausforderung Programmierung

Die Programmierung selbst ist dabei keineswegs ein trivialer Vorgang. Der Zugang zum Fahrzeug erfolgt meist über die OBD-Buchse und ein zentrales Gateway. Manche ECUs sind heute aber auch direkt erreichbar, andere nur über mehrere Gateways. Auch die Busse, über welche die ECUs im Fahrzeugnetzwerk integriert sind, sind bezüglich ihrer Datenraten und Nachrichtenlängen ä-

ßerst heterogen. Es muss berücksichtigt werden, dass bereits heute manche Funktionen nicht auf ein Steuergerät beschränkt sind. Beim Update sind dann mehrere ECUs gleichzeitig auf den neuesten Stand zu bringen. Auch die Datenmengen unterscheiden sich teils erheblich. Bei einfachen Steuergeräten werden Kilobytes übertragen, andere benötigen mehrere GByte, z.B. Navigationsdaten. Dabei stellt nicht nur die Übertragung zum Steuergerät eine Herausforderung dar, auch die Verteilung zur weltweiten Nutzung ist zu meistern. Der Ablauf selbst erfolgt in der Regel dreistufig.

- In der Vorbereitungsphase werden die zu programmierenden ECUs identifiziert und die verfügbaren Flash-Daten ermittelt. Anschließend werden Security-Algorithmen ausgeführt – es darf schließlich nicht jeder Zugriff auf ECUs erhalten – und die Umgebung vorbereitet. Dazu gehört auf dem CAN-Bus Busruhe herzustellen, um die volle Bandbreite zur Verfügung zu haben und die übrigen ECUs über den Programmiervorgang zu unterrichten.
- Im nächsten Schritt erfolgt die eigentliche Programmierung der Steuergeräte.
- Zuletzt wird validiert, dass die Übertragung erfolgreich war, und die Umgebung wird wieder in den ursprünglichen Zustand zurückgesetzt.

Auch von Prozessseite ergeben sich Herausforderungen. Dies beginnt bei der Auswahl der richtigen Flash-Daten, schließlich sind Fahrzeuge meist nur kurz nach der Produktion in einem bekannten Zustand, anschließend werden manche Updates eingespielt, andere nicht. Es muss also zunächst der aktuelle Zustand identifiziert und daraus der zu programmierende Umfang ermittelt werden – unter Berücksichtigung von Freigabeständen über mehrere ECUs hinweg. Auch der Weg zurück muss hier bereits klar sein (Rollback), da in manchen Fällen die Programmierung von einem einzelnen ECU nicht funktioniert, sich am Ende aber alle in einem zulässigen Zustand befinden müssen – funktional und bezüglich der Freigabe.

Ebenfalls unter Robustness fällt eine Infrastruktur, die solche Probleme reduziert. Dazu kann ein stabiles Fahrzeuginterface zählen, die Verwendung von bruchsickeeren Kabelverbindungen (z. B. Mag-Code) oder die Vermeidung von kabelgebundenen Lösungen bei gleichzeitig stabiler WLAN-Ausleuchtung.

Im Hinblick auf die Performance, die bei den hohen Datenvolumina entscheidend ist, ist die ganzheitliche Betrachtung von Prozess und Architektur nötig: welche Daten müssen wirklich geändert werden, welche ECUs kann ich sinnvoll parallel programmieren, kann ich den Datendownload aufs Fahrzeug von der eigentlichen Programmierung trennen. Eine Kombination von Maßnahmen führt hier zu den besten Ergebnissen.

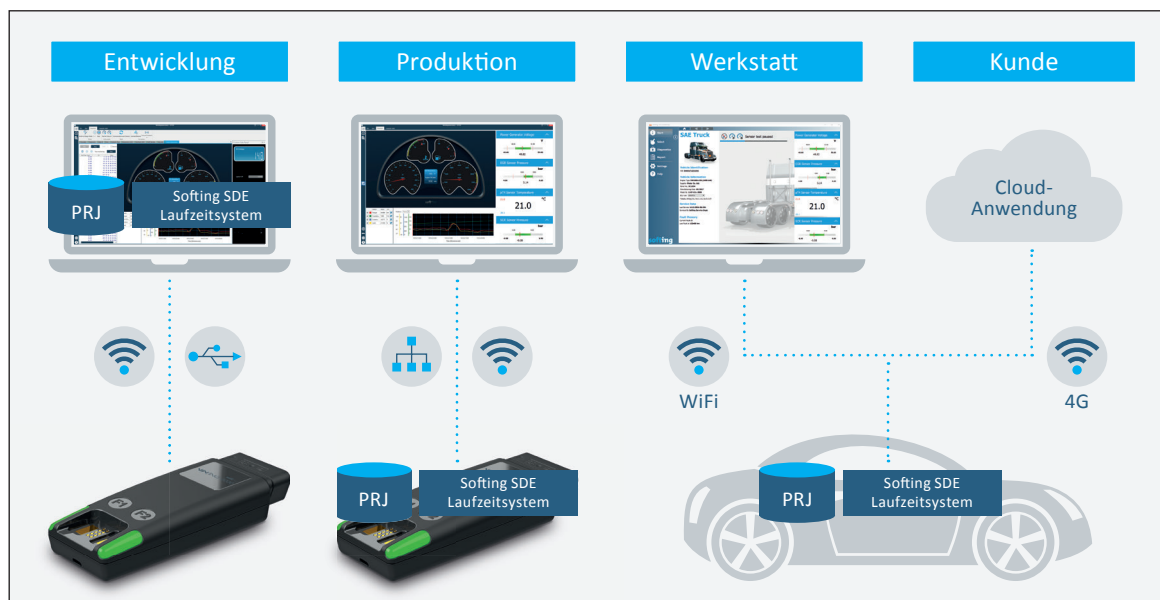


Bild 2: Smarte Architektur ermöglicht Skalierung. © Softing Automotive

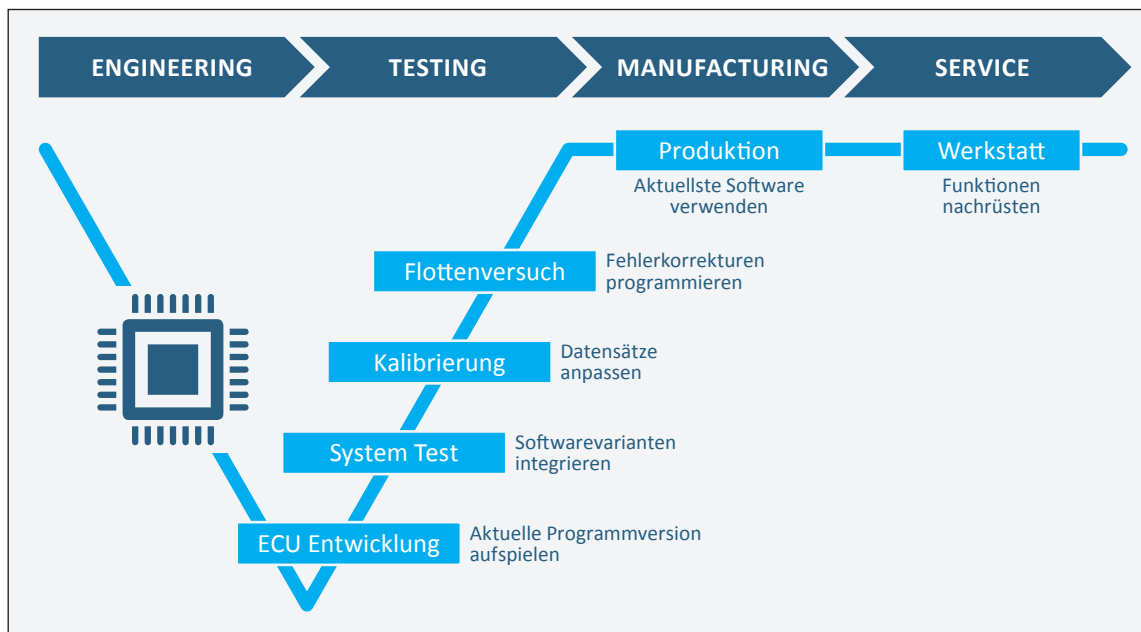


Bild 1: Anwendungsfälle im gesamten Prozess. © Softing Automotive

Praxislösungen

Damit Daten konsistent im gesamten Lebenszyklus verwendet werden können, hat sich für Testsysteme eine dreigeteilte Software-Architektur durchgesetzt. Diese besteht aus einer einheitlichen Protokollimplementierung, einer ebenfalls einheitlichen Laufzeitumgebung für Diagnose und den Anwendungen. Letztere sind spezifisch für den Anwendungsfall umgesetzt. Die Software-schichten können in Lösungen grundsätzlich variabel auf den verwendeten Hardware-Komponenten verteilt werden, also den PCs und VCIs.

In der Steuergeräteentwicklung wird zunächst schnell und ohne große Datenlogistik neuer Code und neue Parametrierungen ins Steuergerät geladen. Dies erfolgt entweder direkt über den Diagnoseentwicklungstester oder über andere Entwicklungstools, die ein Diagnosesystem integriert haben. Letzteres ist dabei in der Regel gemeinsam mit der Applikation auf dem Entwicklungsrechner installiert. Die Verbindung mit dem Steuergerät erfolgt über ein lokales VCI, das über USB oder WLAN mit dem PC verbunden ist (siehe Bild 1).

In Produktionslinien stehen Performance und sichere Datenübertragung im Vordergrund, um Taktzeiten einzuhalten und Nacharbeit zu vermeiden. Da gerade die WLAN-Ausleuchtung hier eine Herausforderung darstellt, hat sich

eine Aufteilung der Softwarestruktur bewährt. Daten und Diagnosesystem liegen direkt auf dem VCI und sind von der Verbindung zum Leitrechner (fast) unabhängig, die Datenversorgung erfolgt an dezidierten Stellen, teilweise sogar über LAN-Kabel.

In Werkstätten kann diese Architektur grundsätzlich übernommen werden. Neueste Fahrzeuge gehen sogar noch einen Schritt weiter. Das VCI wird hier eingespart und das Diagnosesystem direkt ins Fahrzeug integriert, beispielsweise in eine TCU (Telematic Control Unit). Die Daten liegen ebenfalls auf dem Fahrzeug, ihre Aktualisierung erfolgt parallel zu den Steuergeräten. Die Reparaturalgorithmen im Werkstattstester können sich dadurch auf existierende und zum Fahrzeug passende Diagnoseabläufe abstützen.

Remote steigert die Effizienz

Eine Integration des Diagnosesystems im Fahrzeug schafft darüber hinaus noch weitere Freiheitsgrade: in Kombination mit der TCU ist eine einfache Remotediagnose möglich. Das eröffnet schon in der Entwicklung neue Möglichkeiten.

Eine FMU (Functional Mock Up) kann direkt aus dem Büro mit neuer Software versorgt werden, gleiches gilt für Fahrzeuge in der Prototypenwerkstatt. Und auch Prüfstände, die heute vielfach in anderen Regionen betrieben

werden, können sicher aktualisiert werden. Nach Fahrzeugauslieferung an den Kunden ist SOTA (Software over the air) heute die Zauberformel. Gerade kritische Fehler können einfach behoben werden, sobald sich das Fahrzeug in einem sicheren Zustand befindet und der Fahrer zustimmt. Wenn dadurch Rückrufe verhindert werden, ist allen Parteien geholfen.

Fazit

Mit steigendem Software-Anteil im Fahrzeug und wachsender Kritikalität der Software-Funktionen steigt der Bedarf, Steuergeräte anpassen zu können. Flash-Programmierung bietet zahlreiche Freiheitsgrade, stellt dabei einige Herausforderungen an Performance, Sicherheit und Prozessdefinition. Eine geeignete Software-Architektur der Programmierlösung unterstützt signifikant bei der effizienten Nutzung im Feld. ■ (oe)

<https://automotive.softing.com>



Markus Steffelbauer leitet das Produktmanagement und Marketing bei der Softing Automotive und engagiert sich in Standardisierungsgremien.