Sicherer Software-Update „Over the Air“

© Infineon Technologies

Autohersteller sehen sich mit steigenden Kosten für Rückruf-Aktionen konfrontiert. Dabei entfällt ein Großteil der Kosten auf die Korrektur fehlerhafter Software. So liegt es nahe, eine Mobilfunkverbindung zu nutzen, um Software-Updates Over-the-Air zu implementieren. Deren Umsetzung in Automobilen stehen insbesondere Sicherheits- und Komfortaspekte gegenüber. Es muss sichergestellt werden, dass das Fahrzeug vor Manipulationen geschützt ist (Security), der Update-Prozess zuverlässig und schnell erfolgt und die funktionale Sicherheit (Safety) in keiner Weise beeinträchtig wird.

Die grundlegende Motivation für die Software-over-the-air (SOTA)-Implementierung ist offensichtlich. So schätzt IHS Automotive, dass das Einsparpotenzial mittels SOTA von etwa 2,7 Milliarden US-Dollar in 2015 auf mehr als 35 Milliarden US-Dollar in 2022 anwachsen wird (Bild 1). Reduzierte Kosten bei Rückrufen, schnellere Feature-Updates und eine höhere Kundenzufriedenheit sind gute Beweggründe für OEMs, SOTA einzuführen. Derzeit sind OTA-Updates von Navigationsdaten oder Infotainment-Applikationen bereits Stand der Technik. Eine besondere technische Herausforderung stellt nun die Implementierung von SOTA für Steuergeräte außerhalb des Infotainment-Bereichs dar. Hier sind typischerweise Mikrocontroller mit Embedded-Flash im Einsatz, um die Echtzeit-Anwendungen im Automobil zu steuern. Auch der Anspruch an die Qualität und Sicherheit ist sicherzustellen. Die Safety des Fahrzeugs darf nicht durch mangelnde Security gefährdet werden. Daher liegen die Anforderungen deutlich über denen von Infotainment-Anwendungen, in denen heute oftmals Konsumentenprodukte zum Einsatz kommen.

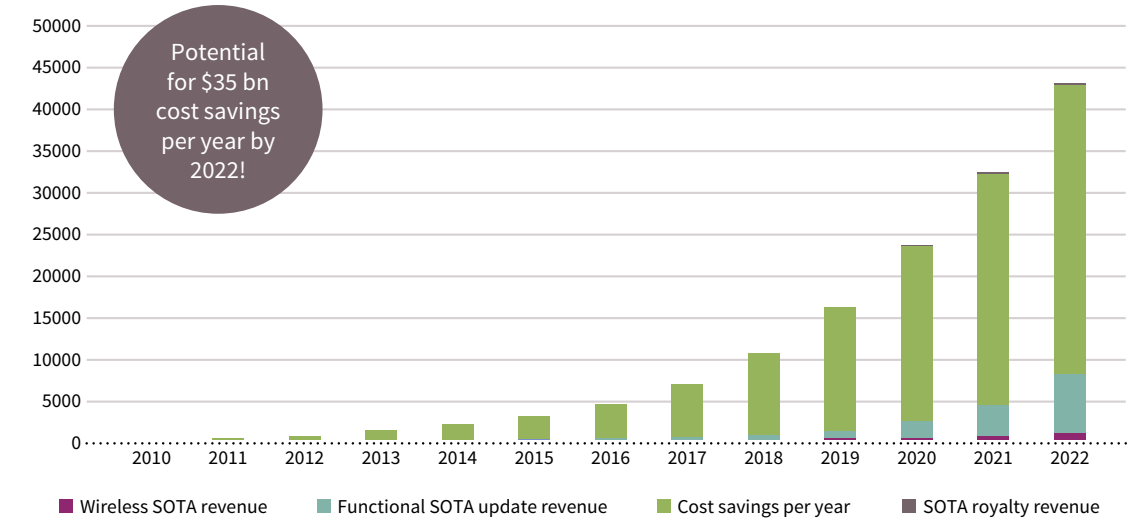

Bild 1: Die wichtigste Motivation für SOTA ist die signifikante Kostenersparnis. (© IHS)

Schutz gegen Cyber-Attacken

Eine unzureichend abgesicherte SOTA-Implementierung, über die es potenziellen Angreifern ermöglicht wird, Manipulationen an Safety-kritischen Anwendungen des Fahrzeugs durchzuführen, kann die gesamte Sicherheit des Fahrzeugs und im schlimmsten Falls das Leben seiner Insassen gefährden. Um dies zu verhindern, bedarf es einer komplexen Security-Architektur, die durch die Verwendung von Zertifikaten und privaten Schlüsseln sowie kryptografischen Operationen unterstützt wird. Entsprechende Kryptografie basiert auf Standard-Algorithmen wie RSA, ECC, AES oder SHA. Infineons Sicherheitscontroller wie die Produkte der AURIX-Familie und der OPTIGA TPM verfügen über derartige Sicherheitsfunktionen und -merkmale.

SOTA-Prozess und die nötige Security-Architektur

Der SOTA-Prozess erfolgt meist in mehreren aufeinander folgenden Schritten: Nach Fertigstellung und anschließender sicherheitstechnischer „Verpackung" (Verschlüsselung und Signierung) eines neuen Software-Pakets erfolgt die Kommunikation zum Zielfahrzeug. Danach wird zwischen dem Fahrzeug (als Client) und dem OEM-Update-Server eine gesicherte Kommunikation etabliert. Das Fahrzeug und die Server-Plattform führen eine wechselseitige Authentifizierung durch und richten einen gesicherten, verschlüsselten Transport-Kanal (TLS, Transport Layer Security) ein, über den das Fahrzeug das neue Softwarepaket empfängt und nach »

Bild 2: Sicherer Flash-Zugriff mit AURIX-Mikrocontroller, inklusive integriertem HSM und zusätzlichem OPTIGA TPM. (© Infineon)

erster Verifikation in einen zentralen Speicher ablegt. Der Empfang und die Speicherung der neuen Software laufen im Hintergrund ab, ohne dass der Fahrer informiert und das Verhalten des Fahrzeugs beeinflusst wird. Der eigentliche Update-Prozess startet erst nach Initiierung durch den Fahrer bei abgestelltem sicheren Zustand des Fahrzeugs und kann je nach Busarchitektur und Lage des Zwischenspeichers unterschiedlich lange dauern (von Sekunden bis zu einigen Minuten).

Architektur für SOTA

Die Fahrzeug-Architektur für SOTA kann prinzipiell in drei ECU-Blöcke gegliedert werden, in denen unterschiedliche Security-Controller unterschiedliche Sicherheitsfunktionen übernehmen: Telematik-Steuergerät, zentrales Gateway und Ziel-Steuergerät (Bild 2).

Die Telematik-Einheit stellt über ihre Mobilfunkschnittstelle eine Verbindung zum OEM-Server her und führt die Service-Authentifizierung aus. Für diese kritische Authentifizierungsfunktion wird aus Security-Gründen die Implementierung eines dedizierten Sicherheitscontrollers, ein OPTIGA TPM (Trusted Platform Module), empfohlen. Ein Mikrocontroller der AURIX-Familie wird üblicherweise neben dem eigentlichen Applikationscontroller für die gesicherte Verbindung zum Fahrzeugnetzwerk genutzt.

Im zentralen Gateway unterstützt ein AURIX-Controller die Verifikation und Zwischenspeicherung der empfangenen Software. Auch bestünde die Möglichkeit, die sicherheitskritischen Authentifizierungsfunktionen von der Telematik-Einheit in das Gateway zu verlagern. In diesem Fall empfiehlt sich die Positionierung des TPMs im Gateway, das hier weitere wichtige Sicherheitsfunktionen wie z. B. das zentrale Schlüsselmanagement übernehmen kann.

In der Ziel-ECU erfolgt nach Initialisierung durch den Fahrer das eigentliche Update. Dazu wird das Datenpaket aus dem Speicher in die Ziel-ECU geleitet, dort entschlüsselt, noch einmal verifiziert und schließlich „geflasht". All diese sicherheitsrelevanten Funktionen werden von AURIX-Controllern unterstützt.

Sichere Authentifizierung und Verifizierung

Wie bereits beschrieben, übernehmen Sicherheitscontroller, sogenannte „Trust Anchors", dedizierte Sicherheitsfunktionen, um Manipulationen und Fehlfunktionen insbesondere beim Update kritischer, Safety-relevanter Anwendungen zu verhindern. Der OPTIGA TPM ist ein zertifizierter Sicherheits-Controller, der speziell für die kritische Authentifizierungsfunktion genutzt werden kann. So soll er sicherstellen, dass nur autorisierte Einrichtungen Daten an das Fahrzeug schicken können. Das TPM führt alle Verschlüsselungs-Algorithmen für die Authentifizierung aus. Dafür speichert es langfristig genutzte Zertifikate und entsprechende private Schlüssel in einer geschützten Umgebung. TPM2.0 unterstützt Algorithmen wie ECC, RSA, AES oder SHA 256. Das TPM kann kryptografisch mit dem Applikationsprozessor

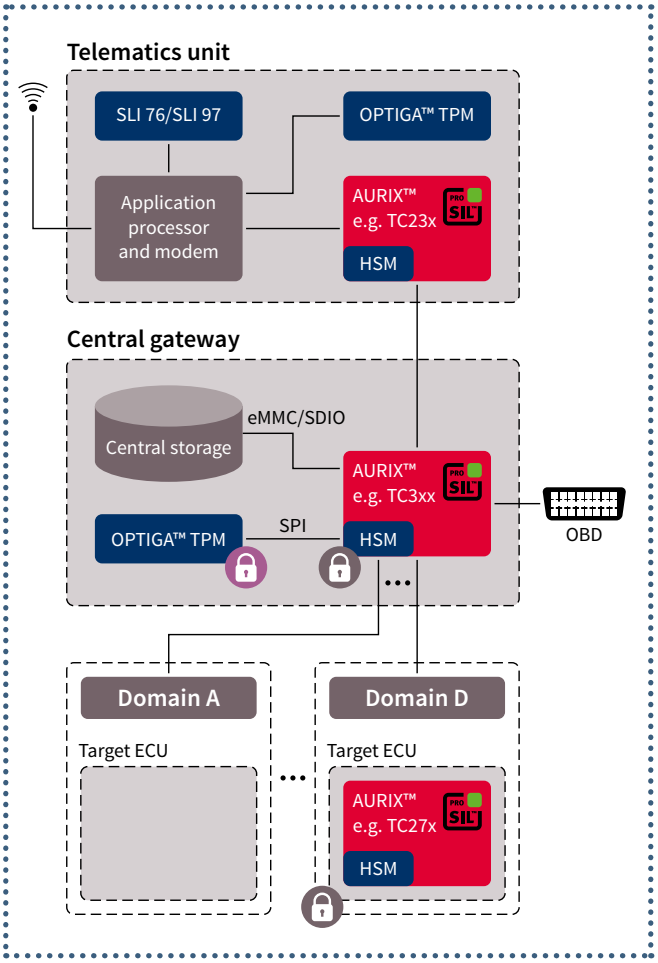

Bild 3: Wesentliche Funktionsblöcke für die SOTA-Implementierung: Telematik-Einheit, zentraler Gateway und Ziel-ECU. (© Infineon)

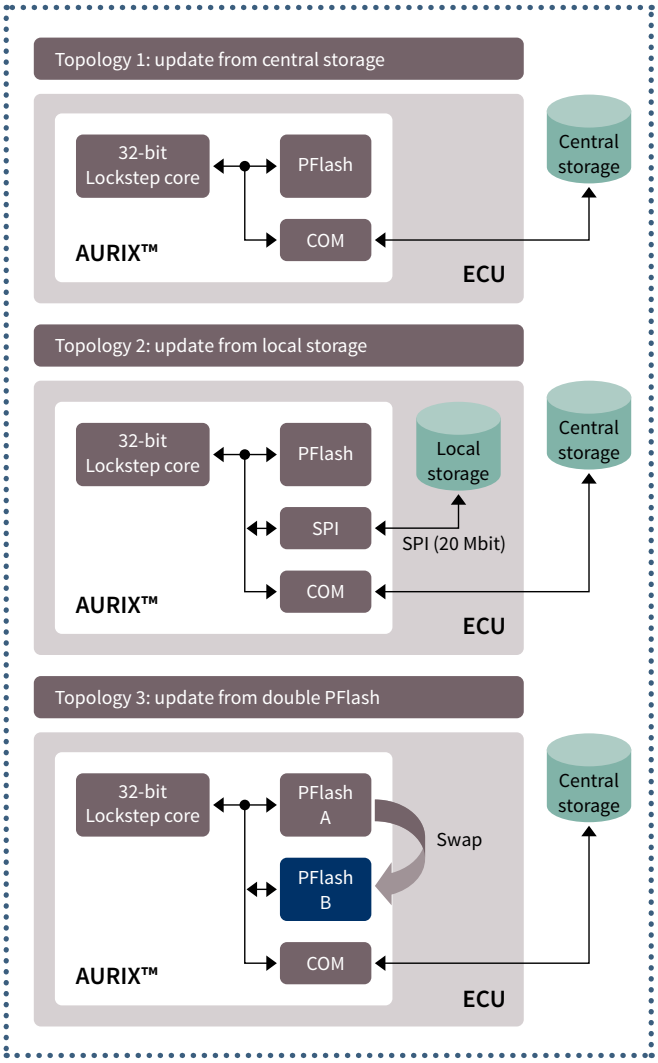

Bild 4: Verschiedene Ansätze für den sicheren OTA Firmware-Update: klassisches Verfahren mit zentralem Gateway-Speicher, A/B-Swap mit zwei Flash-Speicherblöcken und die Methode mit zusätzlichem lokalen Speicher. (© Infineon)

verknüpft werden. Der Schlüsselspeicher des TPM ist skalierbar und kann sicher auf den externen Speicher des Applikationsprozessors geladen werden. Damit können OEMs weitere Authentifizierungs-Zertifikate abspeichern (Bild 3).

Das TPM wird im Rahmen eines sicherheitszertifizierten Herstellungsprozesses gefertigt, bei dem schon ein erster Schlüssel sicher im TPM gespeichert wird. Das Schutzniveau (z. B. gegen Hardware- oder Seitenkanalangriffe) ist bei einem TPM weitaus höher als bei einem SHE (Secure Hardware Extension)-Modul oder einem Hardware Security Module (HSM). Über beide sollten aber alle beteiligten MCUs verfügen, um einen durchgehenden Schutz (End-to-End) zu gewährleisten.

Typische Cyber-Angriffe manipulieren ein System in der Art und Weise, dass dieses nicht spezifizierte Operationen ausführt. Um dies zu verhindern, werden Systeme oftmals in verschiedene, voneinander isolierte Sicherheits-Domains unterteilt. Ein Beispiel ist das TPM, das die asymmetrischen Schlüssel in einer separaten, geschützten Umgebung speichert und für Kryptographieverfahren nützt. Aber auch Mikrocontroller verfügen über isolierte Security-Bereiche. So kann das HSM in den AURIX-Mikrocontrollern Security-Funktionen von der Applikations-Domain isolieren. Ein erster wesentlicher Schritt ist eine Unversehrtheitsprüfung des Programmspeichers der beteiligten Mikrocontroller zu Beginn des Fahrzyklus mittels Secure Boot (SHE oder HSM überprüfen dabei den Speicherinhalt anhand einer kryptografischen Prüfsumme).

Die AURIX-Mikrocontrollerfamilie mit ihrem integrierten HSM übernimmt außerdem die wichtige Verifikation empfangener Software. Dabei profitiert die Verifizierung von den leistungsfähigen Verschlüsselungsbeschleunigern und den schnellen Kommunikationsbussen des HSM. Diese Verifizierung wird durch die Gateway-MCU mittels HSM durchge- »

führt. Da die Firmware-Verifizierung nur öffentliche Zertifikate nutzt, sind die Security-Anforderungen geringer als beim Authentifizierungsprozess.

Im Kontext von SOTA kann das HSM auch für eine On-Demand-Integritätsprüfung genutzt werden. In unserem Beispiel tauschen sowohl die Telematik-Einheit als auch das Gateway ihren Integritäts-Status auf sichere Art und Weise aus und starten erst dann das Software-Update. Entsprechendes kann auf der Ziel-ECU implementiert werden. Dafür nutzt die Ziel-ECU wiederum das HSM, wobei ein sicherer Flash-Boot-Loader (SFBL) für den Empfang und die Verifizierung des Updates verantwortlich ist. Der Unterschied zwischen einem normalen FBL und einem SFBL besteht in zusätzlichen Verschlüsselungs-Algorithmen. Der Boot-Loader selbst sollte von einem SOTA-Update-Prozess ausgenommen sein. Da ein Angriff auf das Fahrzeug während des Fahrbetriebs erfolgen könnte, ist die Möglichkeit einer On-the-Fly-Überprüfung der Applikationssoftware ein wesentlicher Vorteil des HSM gegenüber dem SHE-Modul.

Maximale Verfügbarkeit

Neben dem Thema Sicherheit bei der SOTA-Integration ist für Automobilhersteller von entscheidender Bedeutung, dass die bestehende Systemarchitektur des Automobils möglichst minimal beeinflusst und eine maximale Verfügbarkeit gewährleistet wird, d. h. Minimierung der Updatezeit, in der das Fahrzeug stillstehen muss. Dabei werden im Folgenden insbesondere die vorhandene Bordnetz-Architektur und spezielle Anforderungen auf ECU-Ebene betrachtet.

Bisher wird die Neuprogrammierung einer ECU (oder auch des gesamten Fahrzeugs) üblicherweise in einer Werkstatt ausgeführt. Derartige Updates nutzen ein Diagnose-Tool, das an den OBD-Stecker angeschlossen wird. Das Diagnose-Tool steuert den kompletten Update-Prozess, insbesondere das Herunterladen der neuen Software bzw. des Service-Packs, die Distribution zur Ziel-ECU und die finale Verifizierung. OEMs sind bestrebt, auch für SOTA möglichst nah an diesem Mechanismus zu arbeiten. Für SOTA ist es daher entscheidend, die Funktionalität des Diagnose-Tools auf eine zentrale Stelle in der Bordnetz-Architektur zu übertragen und mit den erforderlichen Funktionen für den zusätzlichen SOTA-Ablauf zu versehen. Diese Funktionen werden üblicherweise innerhalb der Gateway-ECU ausgeführt, in der ein zentraler Speicher das neue Softwarepaket nach dem Download vom OEM-Server zwischenspeichert.

Da die neue Software vom zentralen Speicher des Gateways zur Ziel-ECU gelangen muss, muss die Netzwerk-Topologie beachtet werden, die von OEM zu OEM unterschiedlich ist. Grundsätzlich kann zwischen drei verschiedenen Ansätzen unterschieden werden (Bild 4):

Beim sogenannten „klassischen Verfahren" wird für das Update einer individuellen ECU das entsprechende neue Softwarepaket vom Zentralspeicher über das Bord-Netzwerk in den Embedded-Flash-Speicher des Mikrocontrollers der Ziel-ECU geladen – und zwar in einem Schritt. Hierbei sind keine Hardware-Änderungen seitens der ECUs erforderlich. Die wesentliche Einschränkung liegt bei den Bus-Geschwindigkeiten und damit in der Dauer für die Updates. Tabelle 1 gibt die Datenraten für gängige Bussysteme wider. Nimmt man ein Service Pack mit 4 MByte, wie in der Tabelle angeführt, dann benötigt das Update für eine einzelne ECU über den CANBus fast fünf Minuten und bei 20 ECUs kann das Fahrzeug über 1,5 Stunden nicht genutzt werden. Der Durchsatz kann zwar durch verschiedene Methoden (Cluster von CANbus-Sub-Domains oder Daten-Kompression) erhöht werden, was allerdings höheren Aufwand und Kosten bedingt.

Protocol	Realistic transfer rate	Time to transfer 4 MByte
CAN 500 kbit/s	16.8 kByte/s 50% bus utilization	250 s
CAN FD 2 Mbit/s	88.5 kByte/s 50% bus utilization	47.4 s
FlexRay 10 Mbit/s	300 kByte/s 50% dynamic segment utilization	13.6 s
Ethernet 100baseT 100 Mbit/s	5 to 10 MByte/s (UCP or TCP/IP)	0.8 to 1.6 s

Tabelle 1: Vergleich des Datentransfers für verschiedene Bussysteme. (© Infineon)

	Availability (parking situation)	Cost impact	Impact on power consumption	Impact on ECU performance	Applicable for all MCUs
Classical update from central storage	ca. 30–290 s[1]	Minor (in gateway)	None	None	Yes
A/B Swap Two blocks of program Flash	ca. 100 ms (reset sequence)	Significant[2]	+ ca. 5%	- ca. 0–15%[3]	No[4]
Local storage Extra memory on ECU level	ca. 9 s	Medium	None	None	Yes

1) Criticality essentially depending on the available battery capacity
2) Higher price due increased chip size and test time
3) Depending on implementation and layout constraints
4) Storage limit at the upper end

Tabelle 2: Vor- und Nachteile der verschiedenen Ansätze für den SOTA-Firmware-Update. (© Infineon)

A/B-Swap

Das „A/B-Swap" verfolgt einen anderen Ansatz. Man hat zwei Blöcke (A und B) im Flash-Speicher für die Code-Ausführung innerhalb des Mikrocontrollers. Das Software-Download vom zentralen Speicher in die Ziel-ECU und die Neuprogrammierung des „freien" Speicherteils (z. B. Block B) kann im Hintergrund beim Fahren und so langsam wie erforderlich durchgeführt werden. Währenddessen wird Block A für die Ausführung des aktuellen Codes davon unbeeinflusst genutzt. Sind alle ECUs soweit „vorprogrammiert", wechselt der Controller die Code-Ausführung von Block A zu Block B.

Nach einem Neustart ist der Swap-Vorgang abgeschlossen. Der große Vorteil dieser Methode liegt in praktisch nicht vorhandenen Stillstandzeiten. Der Nachteil besteht in den höheren Kosten durch die größeren Flashspeicher und zusätzlichen Validierungsmaßnahmen, um einen etwaigen Einfluss auf die funktionale Sicherheit auszuschließen. Hinzu kommt, dass viele Mikrocontroller das „Swapping" noch nicht unterstützen.

Ein dritter Ansatz versucht die Vorteile der beiden erst genannten zu kombinieren, bei dem ein zusätzlicher „externer Speicher auf ECU-Ebene" vorgesehen wird, in den während des Fahrzeugbetriebs im Hintergrund das neue Service-Pack geladen wird und dort auf den eigentlichen Updateprozess wartet. Diese Methode nutzt die Tatsache, dass moderne Mikrocontroller wie die AURIX-Familie, ihren Programm-Flashspeicher sehr schnell löschen und neu programmieren können. So können 4 MByte innerhalb von 8 s gelöscht und aus dem externen lokalen Speicher über die SPI-Schnittstelle neu programmiert werden. Die wesentlichen Vorteile dieses Ansatzes sind ein minimaler Eingriff in das bestehende System-Design, überschaubare Zusatzkosten und geringe Abmessungen für das zusätzliche Speicherelement. Tabelle 2 zeigt einen Vergleich der drei besprochenen Methoden.

Fazit

Die Fähigkeit, Software im Auto über Mobilfunk im Feld zu ändern, verspricht ein großes Sparpotenzial für die Automobilindustrie. Jedoch müssen hierbei ausreichende Sicherheitsmaßnahmen ergriffen werden. Die AURIX-Mikrocontroller und zusätzliche dedizierte Sicherheitscontroller wie der OPTIGA TPM an zentral wichtigen Stellen, bieten optimierte Funktionalität für diese wichtige Absicherung von SOTA. Neben konkreten Sicherheitsmaßnahmen müssen sich die OEMs jedoch auch Gedanken machen, wie sie durch eine optimierte Netzwerkarchitektur und Speicherstrategie die Stillstandzeit des Fahrzeugs während des Update-Vorgangs und somit den Einfluss auf den Fahrer möglichst minimieren. ■ (oe)

» **www.infineon.com**

Martin Klimke ist Lead Principal for Technical Marketing Chip Card und Security bei Infineon Technologies.

Björn Steurich ist Senior Marketing Manager Powertrain Systems bei Infineon Technologies.

Ines Pedersen ist Produkt Marketing Manager bei Infineon Technologies.