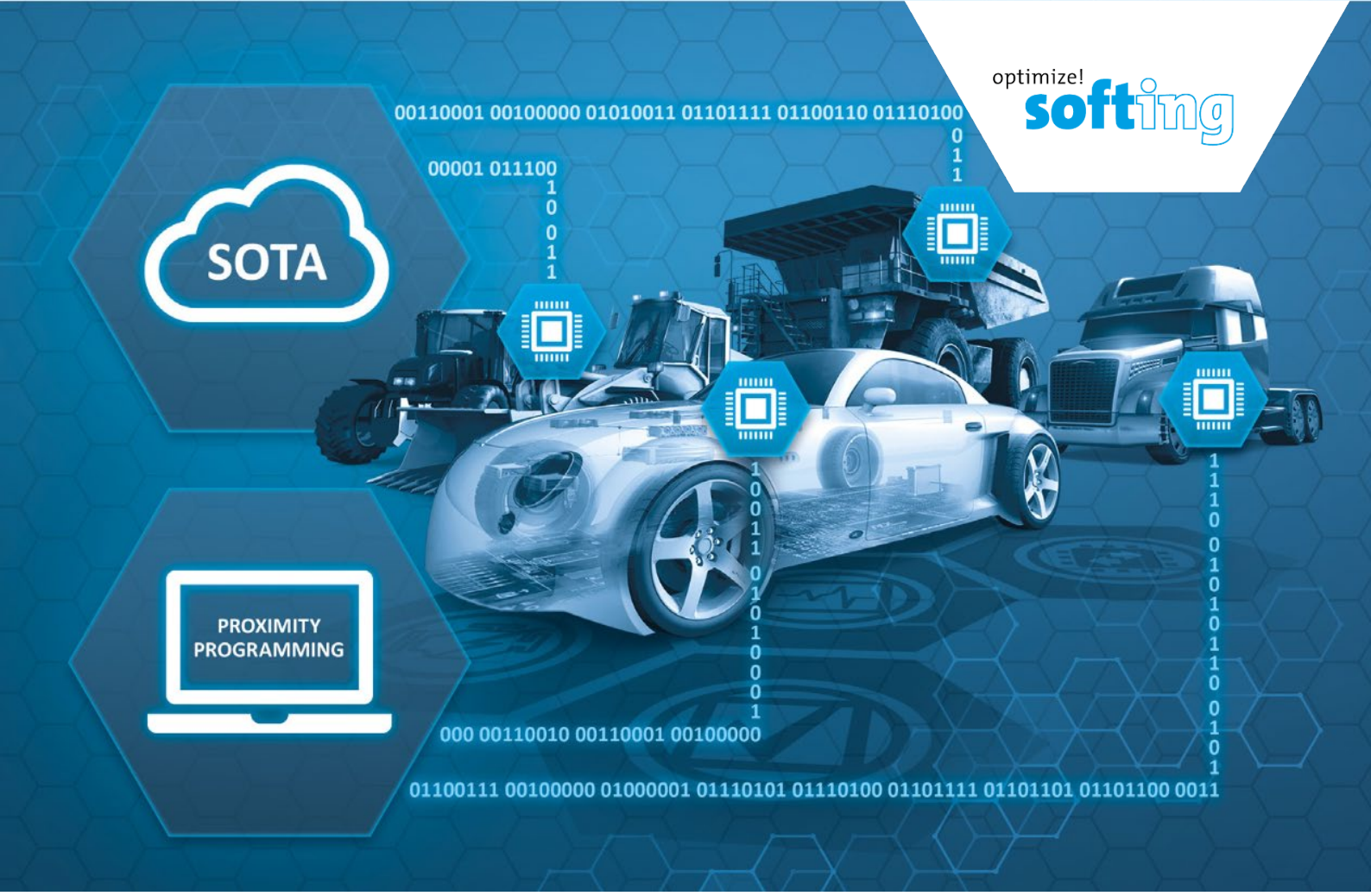

フラッシュ・プログラミングの柔軟性を極める

今日、フラッシュ・プログラミングは、すべての ECU と車両のライフサイクル全体にわたって、非常に高い柔軟性を確保しています。性能、プロセス、データ量などの面で発生する課題は、適切なシステムアーキテクチャによって容易に管理することができます。自動車に搭載されている ECU（Electronic Control Unit）のフラッシュ・プログラミングは、今や広く普及しています。市場によっては、3〜120 個の ECU が存在します。これらの ECU には通常、新しいソフトウェアを直接、またはテスターを使ってゲートウェイ経由で供給することができます。この機能はさまざまな形で利用されていますが、現在、プログラミングアプリケーションと ECU の間には車両インターフェースを接続する必要があります。将来的には、OTA(Over-the-Air)アップデートにより、さらなる可能性が広がるでしょう。

ライフサイクルにおける自由

現在の自動車の付加価値のうち、ソフトウェアが占める割合は増加しており、運転支援機能の普及に伴い、この傾向はさらに強まると考えられます。また、ソフトウェアはパラメータ化が容易であるため、さまざまな基本条件に容易に対応できるというプロセス上のメリットもあります。例えば、ハードウェアとソフトウェアをパラメータ化して、出力として 150hp と 190hp を表現するバリアントコーディングがその一例です。同じように、無数の車両機能も特性マップと個別の調整値によってパラメータ化され、必要に応じて変更される、これは車両のライフサイクルの初期段階で動作を調整する際にも、後期段階でも使用されます。これにより、例えばドライバーの好みに合わせてサウンドシステムの低音を大きくするなど、お客様の嗜好

Richard-Reitzner-Allee 6, 85540 Haar, Germany | Tel +49 (89) 45 65 6-420 | Fax -499 | automotive.softing.com
Managing Directors: René Schneider, Oliver Fieth | Haar near Munich, Munich Local Court, HRB 184930
Contact Japan: Ms. Sunnam Pei | Email: pei@gailogic.co.jp

の変化を考慮することができます。一方で、センサーの予期せぬ経年変化にも対応し、その特性曲線を調整することができます。同じように、コントロールユニットの機能全体を後付けすることもできます。もちろん、純粋なソフトウェア機能であるか、必要なハードウェアがすでにインストールされていることが前提条件です。そうであれば、新しいソフトウェアをプログラムするだけでよく、ソフトウェアのエラーも同じように処理されます。エラーは修正され、新しいソフトウェアのリリースが可能になり、適切な時期にインストールされます。

これは、ライフサイクル全体で使用されます。エンジニアリングでは、ECU の機能をコード化し、バリアントを作成し、機能をパラメータ化し、エラーを修正します。製造工程では、最新バージョンのソフトウェアを ECU にプログラムし、納品後は修理工場でバグ修正や調整、新機能をお客様の車に搭載します。

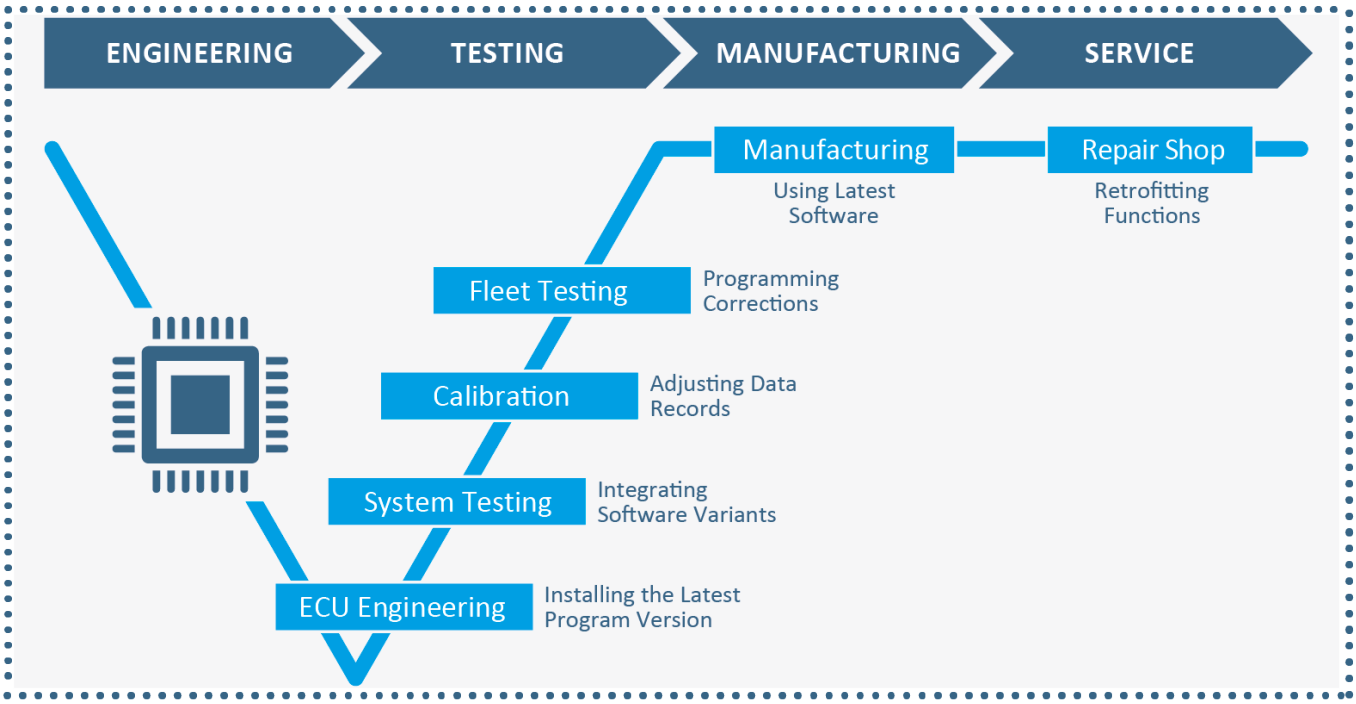

Fig. 1: Use Cases in the Entire Process

プログラミングへの挑戦

プログラミング自体は、決して簡単な作業ではありません。車両へのアクセスは通常、OBD ジャックとセントラルゲートウェイを介して行われます。しかし今日では、直接アクセスできる ECU もあれば、複数のゲートウェイを経由しなければアクセスできない ECU もあります。また、各 ECU が車両ネットワークに組み込まれている各種バスは、データレートやメッセージの長さの点で完全に異なります。今日では、1 つの ECU に限定されない機能がすでに存在していることも忘れてはならない事実です。このような場合、アップデートは複数の ECU を同時に最新の状態にすることになります。また、データ量にも大きな違いがあります。単純な ECU では数キロバイトのデータを転送しますが、ナビゲーションデータなど数 GB のデータを必要とする ECU もあります。ECU への転送だけでなく、全世界に向けた配信にも工夫が必要です。

Richard-Reitzner-Allee 6, 85540 Haar, Germany | Tel +49 (89) 45 65 6-420 | Fax -499 |automotive.softing.com
Managing Directors: René Schneider, Oliver Fieth | Haar near Munich, Munich Local Court, HRB 184930
Contact Japan: Ms. Sunnam Pei | Email: pei@gailogic.co.jp

プログラミングのプセスは、通常 3 段階で行われます。

・準備段階では、プログラムを行う ECU を特定し、利用可能なフラッシュデータを決定します。その後、セキュリティアルゴリズムを実行し、誰もが ECU にアクセスできるわけではないので、環境を整えます。これには、全帯域幅を利用できるようにするための CAN バス上のバスサイレンスの確立や、他の ECU にプログラミングプロセスを知らせることなどが含まれます。

・次のステップでは、実際に ECU のプログラミングを行います。

・最後に、転送が正常に行われたことを確認し、環境を元の状態に戻します。

また、プロセス面でも多くの課題があります。これは、正しいフラッシュデータを選択することから始まります。結局のところ、自動車は通常、製造後しばらくの間は既知の状態にしかならず、その後はアップデートが適用されるものとそうでないものがあるからです。そのため、まず現在の状態を確認する必要があります。その上で、複数の ECU のリリース状態を考慮して、プログラムする範囲を決定する必要があります。1 つの ECU をプログラミングしてもうまくいかない場合があるため、ここでは元に戻す方法（ロールバック）も明確にしておく必要がありますが、最終的にはすべての ECU が機能面でもリリース面でも許容できる状態になっている必要があります。ロバスト性には、このような問題を軽減するインフラも含まれます。これには、安定した車両通信インターフェース（VCI）、壊れにくいケーブル接続（例：Mag-Code）の使用、または同時に安定した WLAN フットプリントを持つ有線ソリューションの回避が含まれます。

パフォーマンスに関しては、特にデータ量が多い場合に重要となりますが、プロセスとアーキテクチャを全体的に捉えることが必要です。どのデータを本当に変更する必要があるのか、どの ECU を意味のある方法で並行してプログラムできるのか、車両へのデータダウンロードと実際のプログラミングを分離できるのか。これらの対策を組み合わせることで、最良の結果を得ることができます。

実用的なソリューション

ライフサイクルを通してデータを一貫して使用できるようにするために、テストシステムでは 3 つのソフトウェアアーキテクチャが確立されています。これは、統一されたプロトコルの実装、同様に統一された診断用のランタイム環境、そしてアプリケーションで構成されています。後者はユースケースに合わせて実装されます。ソリューションでは、ソフトウェア層は基本的に使用されるハードウェアコンポーネント（PC や VCI）に多様に分散させることができます。

ECU 開発では、新しいコードや新しいパラメータ設定を、主要なデータロジスティックを行わずに、迅速に ECU に初期搭載します。この作業は、診断エンジニアリングテスターから直接行うか、診断ランタイムシステムが統合された他のエンジニアリングツールから行います。この診断ランタイムは通常、アプリケーションと一緒にエンジニアリングコンピュータにインストールされます。ECU への接続は、USB または WLAN 経由で PC に接続されたローカル VCI を介して行われます。

Richard-Reitzner-Allee 6, 85540 Haar, Germany | Tel +49 (89) 45 65 6-420 | Fax -499 |automotive.softing.com
Managing Directors: René Schneider, Oliver Fieth | Haar near Munich, Munich Local Court, HRB 184930
Contact Japan: Ms. Sunnam Pei | Email: pei@gailogic.co.jp

Fig. 2: Smart Architecture Enables Scaling

生産ラインでは、サイクルタイムを守り、再作業を防ぐために、パフォーマンスと安全なデータ伝送が重視されます。ここでは特に WLAN のフットプリントが課題となりますが、ソフトウェア構造を分割することでその価値が証明されました。データと診断ランタイムシステムは VCI 上に直接配置され、ホストコンピュータへの接続とは（ほぼ）独立しています。データは専用のポイントで供給され、場合によっては LAN ケーブルを経由することもあります。

このアーキテクチャは、修理工場でも原理的に採用可能です。最新の車両ではさらに一歩進んでいます。ここでは VCI を廃止し、診断ランタイムシステムを車に(例 TCU:Telematic Control Unit)に直接組み込んでいます。また、データは車両に保存され、複数 ECU が並行して更新されます。そのため、修理工場用テスターの修理アルゴリズムは、車両に適合した既存の診断シーケンスに基づいて行うことができます。

リモートによる効率化

診断用ランタイムシステムを車両に組み込むことで、自由度がさらに高まります。TCU と組み合わせることで、簡単なリモート診断が可能になります。これにより、エンジニアリングの段階で新たな可能性が生まれます。FMU（Functional Mock Up）に新しいソフトウェアをオフィスから直接供給することができますし、修理工場の試作車にも同様のことが言えます。また、現在は他の地域で運用されている多くのテストベンチも、安全にアップデートすることができます。

車両がお客様に届けられた後は、SOTA（Software over the air）が今日のマジックフォーミュラです。特に重大なエラーは、車両が安全な状態になり、ドライバーが同意すれば、すぐに修正することができます。リコールを防ぐことができれば、すべての関係者にとってメリットがあります。

Richard-Reitzner-Allee 6, 85540 Haar, Germany | Tel +49 (89) 45 65 6-420 | Fax -499 | automotive.softing.com
Managing Directors: René Schneider, Oliver Fieth | Haar near Munich, Munich Local Court, HRB 184930
Contact Japan: Ms. Sunnam Pei | Email: pei@gailogic.co.jp

We Can Do It!

車両に搭載されるソフトウェアの割合が増加し、ソフトウェア機能の重要性が高まるにつれ、ECU を適合させることができる必要性が高まっています。フラッシュ・プログラミングは、多くの自由度を提供する一方で、性能、セキュリティ、プロセス定義の面でいくつかの課題を抱えています。プログラミングソリューションに適したソフトウェアアーキテクチャは、現場での効率的な使用を大きくサポートします。

Richard-Reitzner-Allee 6, 85540 Haar, Germany | Tel +49 (89) 45 65 6-420 | Fax -499 |automotive.softing.com
Managing Directors: René Schneider, Oliver Fieth | Haar near Munich, Munich Local Court, HRB 184930
Contact Japan: Ms. Sunnam Pei | Email: pei@gailogic.co.jp