特集論文

AUTOSARを利用したECUソフトウェア開発

姫路事業所　開発部　開発第5課
岡村 知哉

1．まえがき

近年、自動車に搭載される機能は年々高度になり、それを実現する車載用電子制御装置(Electronic Control Unit：ECU)の種類やソフトウェアの開発規模が大幅に増加している。ECUのソフトウェア資産の共有や再利用による省力化を目的にAUTOSAR(AUTomotive Open System ARchitecture)開発パートナーシップが制定され業界全体での採用が加速している。

本稿では、AUTOSARリリース3.2からAUTOSARリリース4.2へのマイグレーション作業を通して得たAUTOSAR開発の流れ、およびツール利用について紹介する。

2．AUTOSARとは

AUTOSARとは、車載ECUソフトウェアの標準化アーキテクチャである。

自動車メーカーは、ECU開発におけるソフトウェアの開発規模や複雑さの増加にともない、新技術のリードタイム短縮、ソフトウェア開発費の抑制が課題となっている。また、ECUサプライヤーは、制御ソフトウェア開発の効率化のために、既存制御ソフトウェアの再利用、自動車メーカーに依存しないプラットフォーム化の対応が課題となっている。

これらの課題対処として、ソフトウェアの標準化や再利用を可能にするため、世界の主要な自動車メーカー、ECUサプライヤー、ツールベンダーが、ECU標準ソフトウェアアーキテクチャの確立を目的として「AUTOSAR」を2003年に設立した。

AUTOSARは、何度かのバージョンリリースを行いながら、図1のように、ECUごとに異なるプラットフォームとインターフェース構造を標準化し、制御ソフトウェアの再利用を可能にした。また、プラットフォーム階層の異なりを抽象化し、AUTOSARによるECUソフトウェアの標準化を行った。

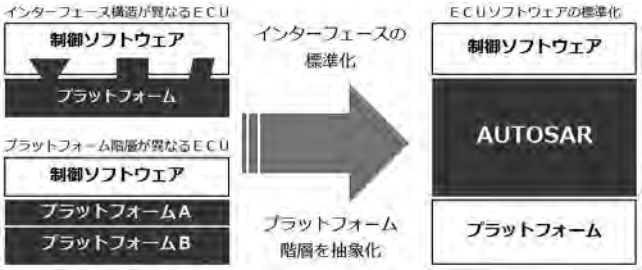

図1．AUTOSARによるECUソフトウェアの標準化

2.1　AUTOSARの概要

AUTOSARでは、図2のように階層化されたソフトウェアアーキテクチャが定義されており、各階層のソフトウェアをモジュール化することで、再利用性を高めている。

各階層のソフトウェア(1)～(6)について説明する。

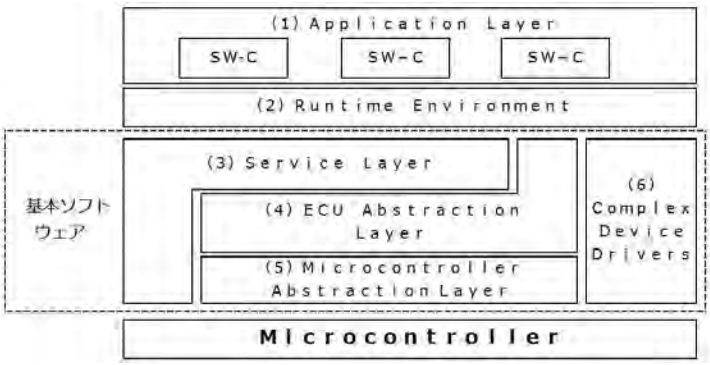

図2．AUTOSARソフトウェアアーキテクチャ

(1) Application Layer

Application Layerは、自動車の制御ソフトウェアで構成される。制御ソフトウェアは機能別に細分化しAUTOSARソフトウェアアーキテクチャで規定されたソフトウェアコンポーネント(SW-C)を用いて実装する。

(2) Runtime Environment

Runtime Environmentは、AUTOSARソフトウェアアーキテクチャのランタイム環境で、図3のようにSW-CとSW-Cの間、SW-Cと基本ソフトウェアの間をつなぐインターフェースである。Runtime Environmentを用いることで、AUTOSARは他のSW-CやECUを意識することなく、SW-Cを開発する仕組みが提供される。

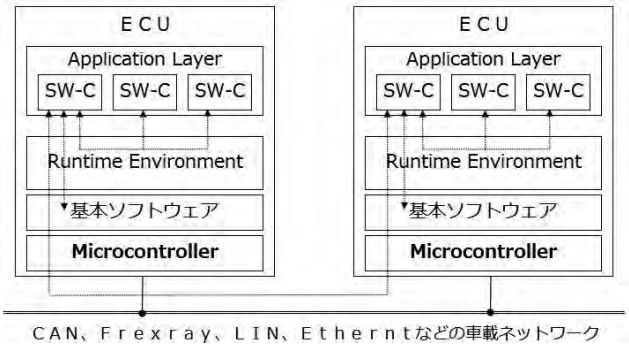

図3．Runtime Environment

(3) Service Layer

Service Layerは、Application Layerが利用するオペレーティングシステム、車載ネットワーク、不揮発性メモリー、故障診断機能、キャリブレーション機能などのサービスを提供する。

(4) ECU Abstraction Layer

ECU Abstraction Layerは、I/O、不揮発性メモリなどのデバイスが内蔵/外付け等のECUハードウェアに依存しないプログラミングインターフェースを提供する。

(5) Microcontroller Abstraction Layer

Microcontroller Abstraction Layerは、ECU Abstraction LayerやService LayerをMicrocontrollerのハードウェア依存から独立させるため、Microcontrollerにアクセスするドライバーを提供する。

(6) Complex Device Drivers

Complex Device Drivers(複合デバイスドライバ)は、マイクロ秒単位での応答性を求められる場合や、基本ソフトウェアの階層や複数のモジュールを経由した処理が間に合わない時に、Microcontrollerへ直接アクセスし制御する。

2.2 業界動向

欧州自動車メーカーでは、既にAUTOSARリリース4によるECUソフトウェア開発を行っている。

国内自動車メーカーでは、一部のメーカーがAUTOSARによるECUソフトウェア開発を行っており、AUTOSAR導入検討が本格化している。

ECUサプライヤーは、欧州自動車メーカー対応で、AUTOSARによるECUソフトウェア開発を導入しており、今後、定着すると予想される。

AUTOSARによるECUソフトウェア開発は、AUTOSAR開発ツールが必要である。AUTOSAR開発ツールベンダーは、Vector社、ETAS社、Elektrobit社、イーソル社やオーバス社などがある。特にVector社はシェアが高く、車載ネットワーク関連の開発･テストツールや、組込みOSなどを提供しており、有力なツールベンダーである。

Vector社のAUTOSARソリューションは、図4のようなMICROSAR、DaVinci Configurator Pro、DaVinci Developerで構成される。MICROSARは、AUTOSAR組み込みソフトウェアである。DaVinci Configurator Proは、基本ソフトウェアとRuntime Environmentの設計、検証、ソースコード生成を行うツールである。また、DaVinci Developerは、SW-Cのインターフェース設計を行うツールである。

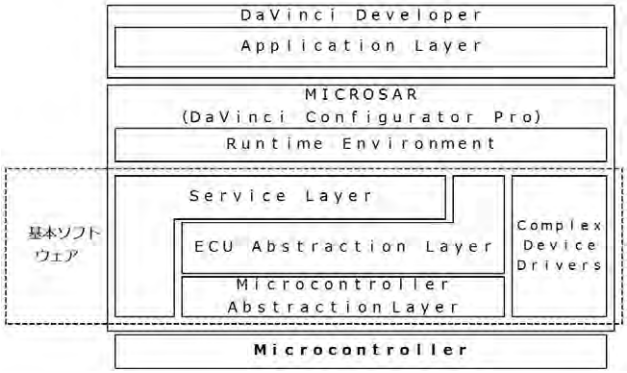

図4．Vector社のAUTOSARソリューション

3．AUTOSARリリース4.2の開発

今回、AUTOSARリリース4.2で採用の車載通信セキュリティ機能を検証するため、AUTOSARリリース3.2で開発されたECUソフトウェアをAUTOSARリリース4.2へ移行する開発を行った。

AUTOSAR開発は、以下のような(1)－(5)の手順で開発を行う。

(1)要求分析
(2)コンフィグレーション
(3)ValidationとGeneration
(4)コンパイルとリンク
(5)実行とデバッグ

3.1 要求分析

要求分析では、ECUに搭載する機能を実現するため、ECUに搭載するマイクロコントローラの選定を行う。また、ECUハードウェア構成に対応したAUTOSARバージョンや利用する基本ソフトウェア、AUTOSAR開発ツールベンダーの選定を行う。

今回、AUTOSARリリース4.2への移行開発は、ツールによる自動変換を検討した。

しかし、AUTOSARリリース4.2は、マルチコアプロセッサ対応、自動車の機能安全規格(ISO 26262)対応や車載通信セキュリティ機能などの追加により、アーキテクチャ改

良、基本ソフトウェアのモジュール追加･変更やパラメータ変更が行われたため、AUTOSARリリース3.2との互換性が無く、ツールによる自動変換ができなかった。

3.2　コンフィグレーション

コンフィグレーションとは、AUTOSAR開発ツールを用いて、AUTOSARのパラメータを設定する作業である。

今回の開発は、図5のようなVector社のDaVinci Configurator Proを用いて、AUTOSARリリース3.2から抽出したパラメータ情報を参考に、以下の(1)－(6)のコンフィグレーションを行った。

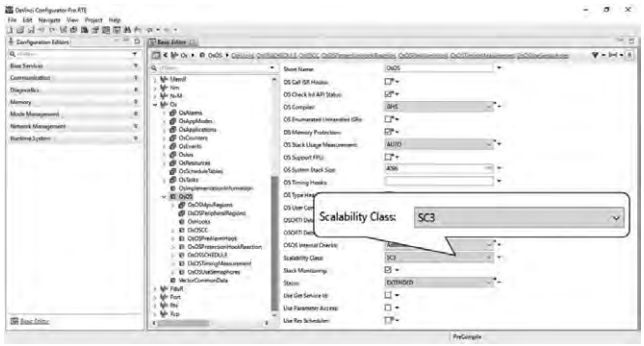

図５．DaVinci Configurator Proのパラメータ設定

(1)車載ネットワーク設定の構築

AUTOSARリリース3.2で使用していた車載ネットワーク記述データファイルをDaVinci Configurator Proに取り込み、車載ネットワーク設定を構築した。

(2)Microcontroller Abstraction Layerの設定

図6のように基本ソフトウェアのMicrocontroller Abstraction Layerは、Microcontrollerのマイクロコントローラドライバー、I/Oドライバー、メモリドライバー、通信ドライバーなどを搭載する。

今回、Microcontroller Abstraction Layerのドライバーに対して、以下のパラメータを設定した。

(a)マイクロコントローラドライバー

Microcontrollerの動作周期、分周器、位相同期回路などを設定した。

(b)ポートドライバー

MicrocontrollerのI/Oポートや通信ポートを設定した。

(c)メモリドライバー

EEPROMメモリ、Flashメモリを設定した。

(d)通信ドライバー

Controller Area Network(CAN)通信やFlexray通信の通信タイミングやコントローラを設定した。

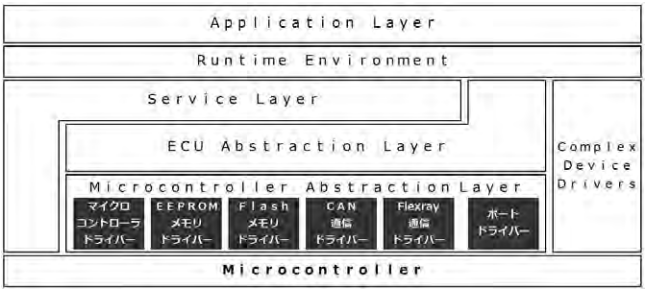

図６．Microcontroller Abstraction Layerの設定

(3)ECU Abstraction Layerの設定

図7のように基本ソフトウェアのECU Abstraction Layerは､メモリハードウェア抽象化モジュール､通信ハードウェア抽象化モジュールなどを搭載する。

今回、ECU Abstraction Layerのモジュールに対して、以下のパラメータ設定をした。

(a)メモリハードウェア抽象化モジュール

EEPROMメモリドライバーやFlashメモリドライバーを抽象化し、1つのインターフェースでEEPROMメモリやFlashメモリにアクセスできるように設定した。

(b)通信ハードウェア抽象化モジュール

CANやFlexray通信ドライバーを抽象化し、1つのインターフェースで複数の通信コントローラやチャンネルにアクセスできるように設定した。

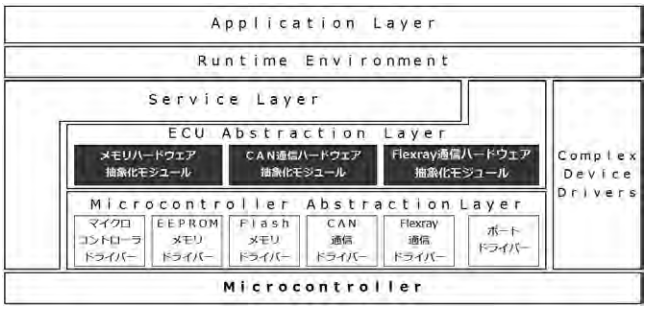

図７．ECU Abstraction Layerの設定

(4)Service Layerの設定

図8のように基本ソフトウェアのService Layerは、システムサービス、メモリサービス、通信サービスなどを搭載する。

今回、Service Layerのサービスに対して、以下のパラメータ設定を実施した。

(a)システムサービス

OSのスケーラビリティクラスを設定し、アラーム、タスク、イベントやECUステートマシン管理を利用できるように設定した。

(b)メモリサービス

メモリハードウェア抽象化モジュールを用いて、故障診断やSW-Cなどのデータを不揮発性メモリに読書きで

きるように設定した。

(c)通信サービス

CANやFlexray通信ハードウェア抽象化モジュールを用いて、通信フレーム、故障診断プロトコル、キャリブレーションプロトコルのデータ構築や抽出、通信状態の取得が利用できるように設定した。

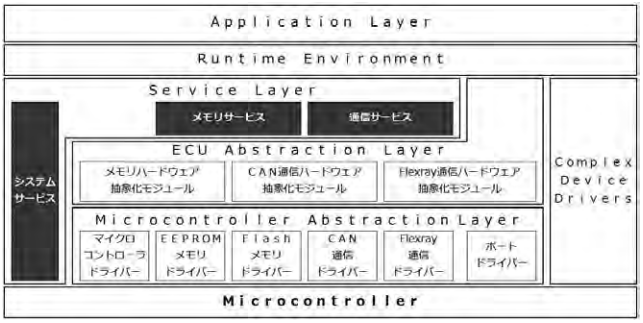

図８．Service Layerの設定

(5)Application Layerの設定

DaVinci Developerを用いて、制御ソフトウェアを機能別に細分化し、図9のようにSW-Cを作成し、SW-Cのインターフェースとなる Runnable(関数名や関数コール方法)やPort(関数の戻り値や引数となるデータ)を設定した。

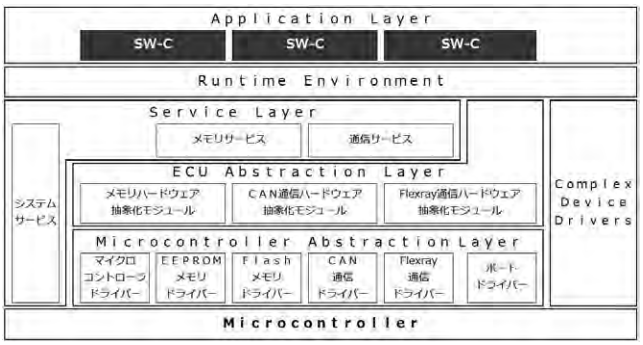

図９．Application Layerの設定

(6)Runtime Environmentの設定

DaVinci Configurator Proを用いて、SW-CのRunnableやPortとRuntime Environmentの関連付けを設定した。

3.3 ValidationとGeneration

(1)Validation

全パラメータを設定後に、DaVinci Configurator ProのValidation機能(パラメータ検証機能)を用いて、パラメータのミスマッチや過不足設定などを検証した。

今回、AUTOSARリリース3.2からの移行開発を行ったため、AUTOSARリリース4.2で追加/削除されたパラメータやパラメータ不一致による漏れや誤りが発生した。問題が発生したパラメータは、AUTOSAR仕様書、AUTOSAR開発ツール仕様書、マイクロコントローラマニュアルを参考にパラメータを設定した。

(2)Generation

パラメータ設定の検証後に、DaVinci Configurator ProのGenerate機能(コード生成機能)を用いて、ソースコードを自動生成した。自動生成されたソースコードにECU起動の初期化処理をハンドコードで実装した。また、制御ソフトウェアをハンドコードでSW-Cに実装した。

3.4 コンパイルとリンク

DaVinci Configurator ProのGenerate機能で自動生成したソースコードをMakeファイルに設定、コンパイル、リンクを行った。

3.5 実行とデバッグ

評価用ボードやデバッガーを用いてデバッグする。デバッグは、ECU起動時の割込み禁止設定、タスク周期、通信の送受信、不揮発性メモリの保持や読出し、状態遷移などの動的要素も確認した。

4．AUTOSAR開発における課題

AUTOSARによるECUソフトウェア開発は、数千項目のパラメータ設定が必要である。AUTOSAR規格は、パラメータを定義しているが、パラメータ仕様や処理が曖昧となっているため、AUTOSAR開発ツールによってパラメータの解釈が異なり、同じパラメータでもECU動作が異なる場合がある。

AUTOSAR開発は、パラメータの組合せによる相関が多く、ValidationとGenerationのパラメータ変更が多く発生し、ソースコード生成までに時間を要する。また、ソースコード生成後のデバッグも、通信、状態遷移や不揮発性メモリなどの動的要素を要求仕様にあわせるため、パラメータ変更が発生する。

AUTOSAR開発の特徴は、数千項目に及ぶパラメータをAUTOSAR仕様書、AUTOSAR開発ツール仕様書、マイクロコントローラマニュアルから抽出し、必要なパラメータや設定を事前に特定することが困難な点である。実際のAUTOSAR開発では、過去に用いたパラメータを参考にして、パラメータを設定する必要がある。

AUTOSAR開発の人材面は、従来の組込み系ソフトウェア開発のノウハウや経験などの恩恵を受けるケースが少なく、AUTOSAR開発ツールを利用するためのトレーニングや、AUTOSAR開発のデバッグ/評価方法の習熟など、AUTOSAR開発を行うまでの準備に多くの時間を要する。

5．今後の展開

AUTOSARリリース4.2への移行開発後、CAN通信モジュールを利用した故障診断機能(故障コード記録する不揮発性メモリへのアクセス、ECUリセット要求に対するモードマネージメントとECUシャットダウン)やECUモニタ/キャリブレーション機能の対応を行っていく。

また、AUTOSARに関するツール開発として、AUTOSAR共通基盤パラメータにアクセスできるARTOP(AUTOSAR Tool Platform)(注1)を利用したARTOP開発環境を構築したので、今後、この開発環境を用いて、AUTOSAR開発における支援ツールの開発を行っていく。

6．むすび

今回、三菱電機㈱自動車機器開発センター(車開セ)からAUTOSARリリース4.2への移行開発の依頼を受け、AUTOSAR開発に携わることができた。今後も、車開セのAUTOSARチームに携わりながらAUTOSAR開発・サポートの対応を進めていく。

最後に、AUTOSAR開発に際し、ご支援ご協力いただいた関係者の方々に深く感謝を申し上げる。

(注1) AUTOSARディスクリプションファイルを読込み、AUTOSAR共通基盤パラメータにアクセスできるソフトウェアモジュール

執筆者紹介

岡村　知哉　　オカムラ　トモヤ
1998年入社。主にECUのソフトウェア開発に従事。現在、姫路事業所開発部開発第5課。