

非AUTOSARからAUTOSARへの移行

三田事業所 開発部 開発1課
横田 浩之

1. まえがき

近年、自動車機器開発における電気電子(E/E)システムの複雑さが増大し続けており、技術革新を必要とするレベルに達している。この複雑さに対処するため、自動車業界全体における車載ソフトウェアの再利用や授受を促進するべく、2003年に自動車業界のグローバル開発パートナーシップとしてAUTOSAR(AUTomotive Open System ARchitecture)が設立された。

AUTOSARでは、階層化されたソフトウェアアーキテクチャを定義している。AUTOSARの階層は、アプリケーション層、AUTOSARランタイム環境(RTE:RunTime Environment)^(注1)、基板ソフトウェア(BSW:Basic SoftWare)^(注2)に分かれている。また、AUTOSARは、ソフトウェアをモジュール化し、モジュール間のインタフェースを標準化することにより、再利用性を高めている。

現在、海外の自動車メーカーを中心にAUTOSARの採用が進められており、国内メーカーも導入を始めている。この状況の中、当所は、三菱電機(株)三田製作所で行ったIIP

(Intelligent Information Panel)の試作機開発にて、従来のプロジェクトを参考に、AUTOSARに対応した開発を進めてきた。

本稿では、AUTOSAR対応開発の概要と、試作機開発で実施したAUTOSARへの移行方法と課題について紹介する。

2. AUTOSARとは

2.1 AUTOSAR対応開発の流れ

AUTOSAR対応開発での自動車メーカー及びECU(Electronic Control Unit)サプライヤでの設計・開発の流れを図1に示す。

各開発ステップ間の設計情報の授受の媒体は、AUTOSAR XML(arxml)形式のファイルが用いられる。

図1に示すとおり、ECUの開発では、自動車メーカーから提供されるEcuEx(ECU Extract of System Description)をインプットとして、設計・開発を行う。

一方、自動車メーカーから入手したEcuExで設計情報が

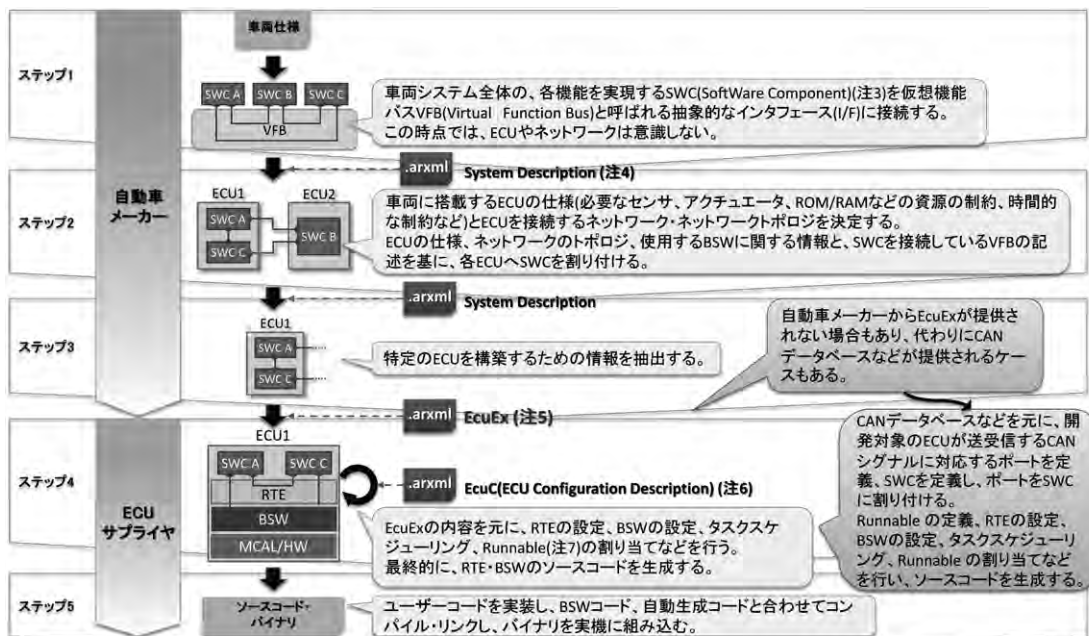


図1. AUTOSAR開発の流れ

(注1) VFB上の仮想的なSWC間通信をECU内通信、またはECU間通信として実装する。また、SWC内のRunnableを実際のOS上で動作するTASKに割り付け動作させる。
(注2) AUTOSARの規格に従って設計された、OS、通信スタック、診断、入出力などの各機能を提供するソフトウェアモジュール群。
(注3) AUTOSARで標準化されたタイプやI/Fに従い定義された、マイコンに非依存で基本的に再利用可能なRunnableの集合体。

(注4) 車両システム全体の設計情報。
(注5) System Descriptionの内、特定のECUを構成するための情報。自動車メーカーとECUサプライヤ間の設計情報の授受媒体。
(注6) ECU上のSWCの情報、RTEやBSWの設定情報。
(注7) 周期イベントなどの各種イベント発生時にSWCで実行される処理。

不足する場合や、提供されない場合にはサプライヤが不足を補っている。

3章で述べる試作機開発では、自動車メーカーからのインプットがないため、CAN(Controller Area Network)のデータベース自体の定義・作成を行い、作成したCANデータベースをインプットとしてAUTOSAR対応開発を実施した。

3. AUTOSARへの移行

3.1 IIP試作機開発

当所では、三菱電機(株)三田製作所とともに、IIPの試作機開発を行った。

本試作機開発において当所は、AUTOSARを搭載したマイコンのソフトウェア開発を担当した。

IIPは、従来別々のユニットであるHU(Head Unit)とIC(Instrument Cluster)を統合したユニットである。

HUには、機能安全^(注8)の対応が要求されないナビゲーションやオーディオ機能などのエンターテインメント機能が搭載される。一方、ICには、機能安全の対応を求められる方向指示器や警告灯などの機能が搭載される。

機能安全の対象機能と非対象機能を共存させるには、機能安全の非対象機能から機能安全の対象機能への干渉・侵害を防止する必要がある。IIPでは、機能安全の対象機能と非対象機能を分離するアーキテクチャを採用し、1つのSoC(System on Chip)上に両方の機能を実現した。

本試作機開発は、採用したアーキテクチャでの性能評価、課題抽出、自動車メーカーへのデモなどを目的として実施した。

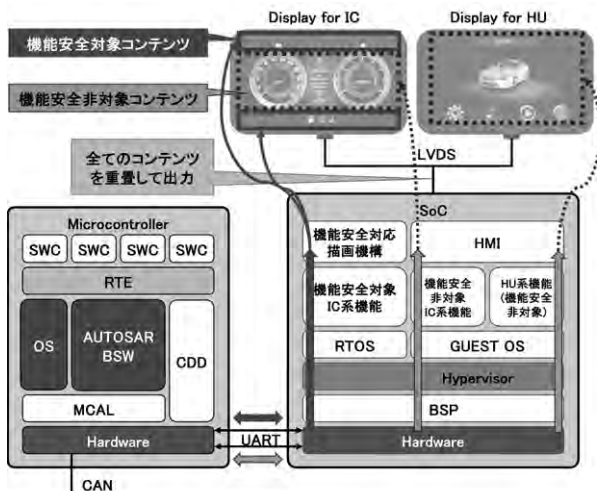


図2. IIP構成図

本試作機は、マイコンとSoCの2CPUで構成されている。マイコンでは、システムの起動終了制御、電源制御、CAN通信、AudioDSPの制御などを行う。例えば、CANから受信した各ホイールの回転数の情報から車速を計算し、UART(Universal Asynchronous Receiver Transmitter)

経由のCPU間通信でSoCに通知する。

SoCでは、マイコンから受信した情報を元に、メータや警告灯などのICの情報を描画する。また、HUの情報としてオーディオ・ビデオプレイヤーなども描画する。SoC内部では、機能安全系の機能はRTOS(Real Time OS)上で処理され、機能安全に対応する描画機構により映像が生成される。一方、機能安全が求められない機能は、LinuxなどのGUEST OS上で処理され、映像が生成される。

最終的に、SoC内のDisplay Controllerにより、それぞれの映像が重畳され、ディスプレイへ出力される構成となっている。

3.2 実施内容

本試作機のマイコンのソフトウェア開発では、AUTOSARを使用していない既存機種のソフトウェアアーキテクチャ設計を参考に、新たにAUTOSARを導入したソフトウェアを構築した。

図3が既存機種のソフトウェアブロック図、図4が本試作機のソフトウェアブロック図である。

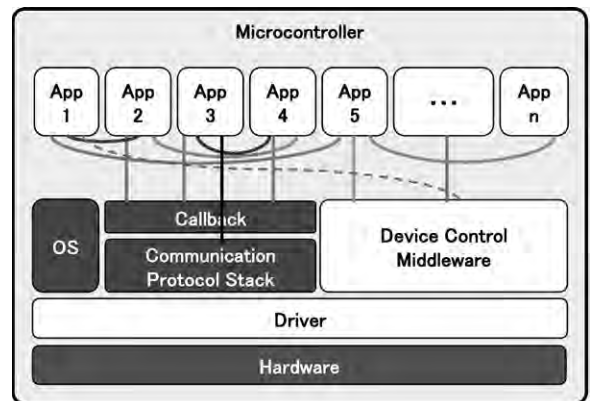


図3. 既存機種のソフトウェアブロック図

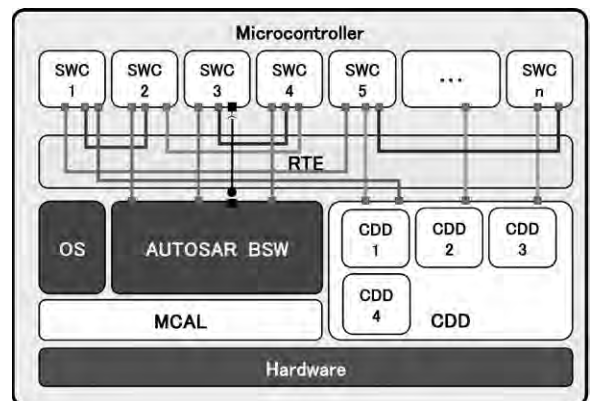


図4. 本試作機のソフトウェアブロック図

(注8) 自動車機器の機能、部品が故障しても、E/Eシステムの安全性を確保する考え方で、国際規格のISO26262を指す。

既存機種と本試作機の違いは2点ある。

1つ目は、通信プロトコルスタックの違いである。既存機種では3rd Party製のソフトウェアスタックを導入していたが、本試作機ではAUTOSARのBSWを導入した。このBSWを実装するための設定は、既存機種でのソフトウェアスタックと比較して、設定すべきパラメータ自体が変わっていること、設定項目数が大幅に増加していることなど、大きな違いがある。

2つ目は、既存機種での各アプリケーションをAUTOSARに適合させるため、SWC(SoftWare Component)として実装した点である。

AUTOSARのSWCは、CANシグナルの入出力、SWC間の通信、BSWが提供するサービスへのアクセスなどを全て、RTEを経由して行う。

この違いに対応するために実施した主な変更内容は、以下4点である。

- (1)CANシグナル受信方法
- (2)CANシグナル送信方法
- (3)プロトコルスタック提供サービスの実行方法
- (4)アプリケーションの実装方法

本試作機開発では、設計ツールとしてVector社製のDaVinci Developer及びDaVinci Configurator Proを使用した。設計ツール上で各SWCやCDD(Complex Device Driver)^(注9)及びポート^(注10)を定義し、接続した例を図5に示す。

図5で実線で囲った部分が、SWCやCDDである。IC機能SWCsは、設計ツール上で階層化されており、図5上吹き出しに表すように、内部に詳細なSWCを複数定義している。

全てのアプリケーションやドライバをSWCやCDD化する必要はない。しかし、今回はAUTOSARのポートやI/F仕様の把握、使いこなしを目的として、全てのアプリケーションとMCAL(MicroController Abstraction layer)^(注11)として提供されないUARTやI2C(Inter Integrated Circuit)などのドライバをSWCやCDDとして実装した。

図5でIC機能SWCsと表記しているSWCは、CANシグナルを送受信し、CANから取得した情報を処理するSWCである。その他のSWCは、Audio機能やSystem状態管理機能などのアプリケーションである。

それぞれのSWCやCDD間の線は、SWC及びCDD間の通信I/Fであるポートの接続を表している。ポートの接続により、SWCやCDD、BSW間の通信が定義可能となり、必要な通信毎に設定する。

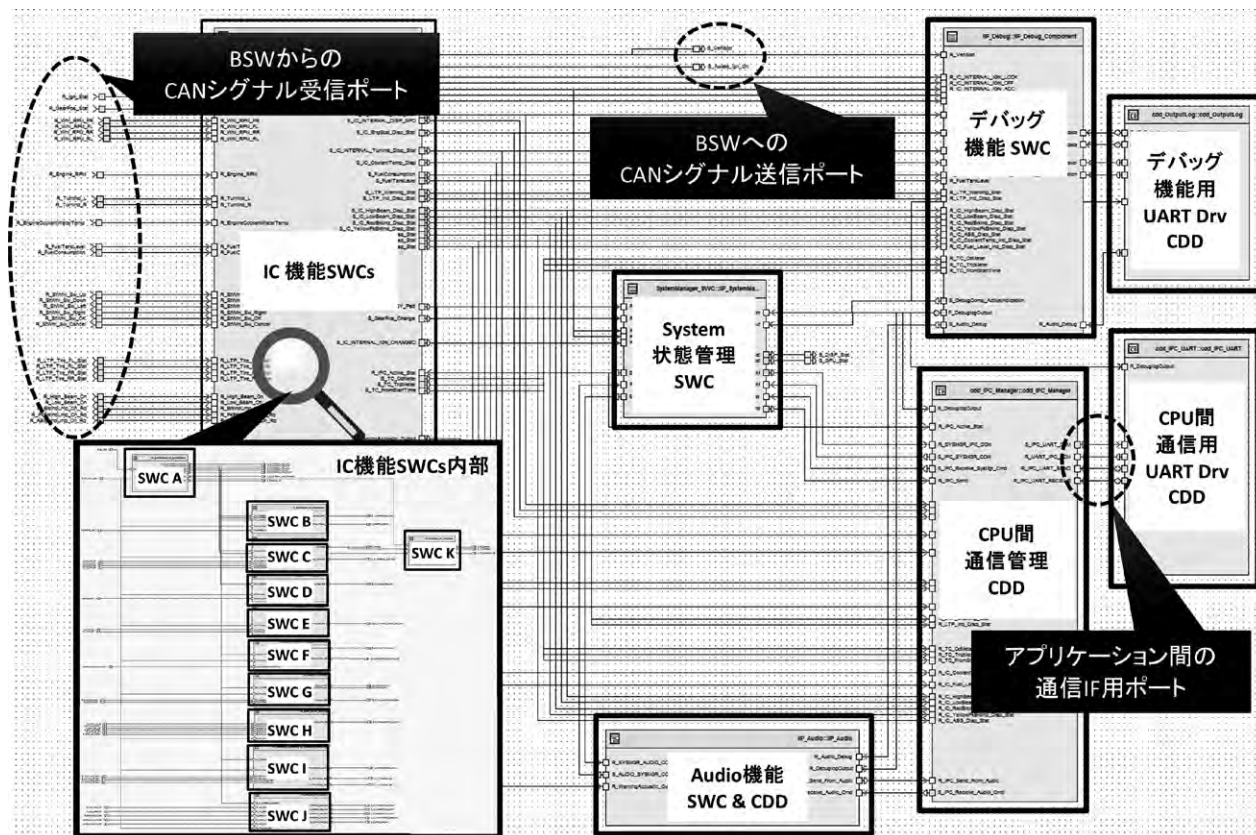


図5. SWC・CDD・ポート定義・接続例

(注9) AUTOSARの規格でサポートされないドライバや、時間的制約が厳しい処理、従来型の既存ソフトウェアを組み込む場合の分類。

(注10) 標準化された形式に従って定義されたSWC間I/F。CDDとBSW、BSWとSWC間のI/Fとしても使用可能。

(注11) マイコン固有のドライバ (Port,DIO,MCU,GPTなど)。一般的にマイコンメーカーが提供する。

本試作機開発では、客先からのEcuEx提供がないため、SWCの定義、CANシグナルや通信毎のポート定義、通信先のSWCやCDD、BSWとの接続、全てのポートや通信毎のパラメータ設定などの作業を設計ツールで実施した。

この作業には、AUTOSARの知識、設計ツールの使用方法や各パラメータの設定内容に対する知識が必要となる。

また、AUTOSARのバージョン更新に伴い、設計ツールに設定するパラメータも変化する。

そのため、プロジェクトで使用するAUTOSARのバージョンに追従して、作業も継続的な知識習得が必要となる。

以降に、実施した各変更内容について述べる。

(1) CANシグナル受信方法の変更

既存機種と本試作機での受信したCANシグナルをアプリケーションへ通知する方法、アプリケーションでシグナル値を取得する方法を表1に示す。

表1. CANシグナル受信方法

種別	実装方法
既存機種	プロトコルスタックからのCANシグナル毎のCallback関数実行により、シグナル受信を通知。Callback関数内で、プロトコルスタック提供I/FによりCANシグナル値を取得。
本試作機	CANシグナル毎にポートを定義、受信ポートをSWCに割り付け、CANシグナルを処理するRunnableを定義。Runnableの駆動トリガに受信ポートのデータ受信を設定。CANシグナル受信時にRunnableが駆動され、Runnable内で、RTEにより生成されるI/FによりCANシグナル値を取得。

(2) CANシグナル送信方法の変更

既存機種と本試作機でのアプリケーションからCANシグナルを送信する方法を表2に示す。

表2. CANシグナル送信方法

種別	実装方法
既存機種	プロトコルスタック提供I/Fを直接呼出しにより、CANシグナル値を設定し、送信。
本試作機	CANシグナル毎にポートを定義、送信ポートをSWCに割り付け、CANシグナルを送信するRunnableを定義。Runnable内で、RTEにより生成されるI/FによりCANシグナル値を設定し、送信。

(3) プロトコルスタック提供サービスの実行方法の変更

既存機種と本試作機でのアプリケーションからBSWのサービスを実行する方法を表3に示す。

BSWの提供するサービスとは、例えば通信モードの変更や読み出しなどが該当する。

表3. プロトコルスタック提供サービスの実行方法

種別	実装方法
既存機種	プロトコルスタック提供I/Fをアプリケーションから直接呼出し、実行。
本試作機	サービスを実行するRunnableを定義。サービスが提供するサービスポートをRunnableに割り付ける。Runnable内で、RTEにより生成されるI/Fによりサービスを実行。

(4) アプリケーションの実装方法の変更

既存機種と本試作機でのアプリケーションの実装方法を表4に示す。

表4. アプリケーションの実装方法

種別	実装方法
既存機種	主にOSのTASKとして実装。OS提供I/F (EventやMessageなど)の使用、公開関数の直接呼出しなどにより、アプリケーション間の通信やアプリケーションの駆動を実施。
本試作機	SWCとして定義し実装。アプリケーション間の通信用ポートを定義、ポートをSWCに割り付け、Runnableを定義。Runnable内で、SWCに割り付けたポートI/Fに応じてRTEが生成するI/Fにより、アプリケーション間の通信を実施。ただし、モジュール内部処理の駆動にOS提供I/Fも直接実行。

3.3 AUTOSARへの移行における課題

今回の試作機開発では、Audio機能やSystem状態管理などのアプリケーション、UARTやI2Cなどのドライバも全て、SWC、CDDとして定義し、それぞれのモジュール間の通信はRTEを経由する設計とした。

しかし、一方、Audio機能やSystem状態管理などのアプリケーション、UARTやI2CなどのドライバをAUTOSARに対応させたSWC、CDDでは、従来と同じくOS提供I/Fも使用した。

その結果、本来のSWCの駆動方式とは異なる設計となったため、設計ツールにより自動生成されるコードに手を加えなければならなくなり、コードを生成する度に手修正を行う必要があった。

AUTOSARでは、アプリケーションを全てSWCとして定義しなければならない、という制約はないため、今後の開発においては、モジュールの特性に応じて最適なSWCやCDDの定義方法や運用方法の検討が必要であるという課題が明確になった。

4. AUTOSAR対応の現状と今後

国内・海外問わず自動車メーカーから、AUTOSAR対応の要求が増えてきている。

それに伴い当所も関連する量産開発の受注が増加することが見込まれる。一方、AUTOSAR有識者、経験者が不足しており、AUTOSAR関連技術の早期獲得が必要である。

AUTOSARへの対応力を強化するため、AUTOSARに対する知識習得、設定に使用する設計ツール類の導入と、その使いこなし、ノウハウ蓄積を推進する。

また、事業所の枠を超えた情報共有により、社内全体でAUTOSAR対応力の底上げを行っていく。

5. むすび

IIPの試作機開発を通じて、AUTOSAR対応開発に触れ、設計方法、設定内容、設計ツールの使用方法や、課題の把握ができた。

今後の車載機器開発では必須要件となってくるAUTOSAR対応に向け、当所として組織的な対応力強化を進めていく。

最後に、今回のIIP試作機開発に際し、共同開発を行った三菱電機(株)三田製作所はじめ社内外の関係各位に深く感謝を申し上げる。

執筆者紹介



横田 浩之 ヨコタ ヒロユキ
2006年入社。主に車載情報ネットワークのソフトウェア開発に従事。現在、三田事業所開発部開発第1課。