

組込み機器用アプリケーションへのソフトウェア追加方法

川上 武[†] 片山 吉章[†] 伊藤 益夫[†] 徳永 雄一[†]

[†]三菱電機株式会社 情報技術総合研究所

1. 背景

近年、自動車における電子制御化は、ボディ制御、エンジン制御、パワートレイン制御、車両制御、情報システムなど、多岐にわたり急速に進んでいる。これに伴い、自動車向け電子制御ユニット(Electronic Control Unit : ECU)の数が飛躍的に増加し、ECU 開発に占める車載組込みソフトウェアの規模と開発工数が増加している。このように、自動車全体の統合制御の実現に向けて ECU を協調制御する必要があることから、車載ソフトウェアの複雑性も増しており、車載ソフトウェアの規模及び複雑性の増大に対処するための手段として、車載ソフトウェアの標準化が AUTomotive Open System ARchitecture (AUTOSAR) ですすめられている [1][2]。

AUTOSAR ソフトウェアプラットフォームは、車載ソフトウェアの共通化を目的としており、アプリケーション層をソフトウェアコンポーネント (Software Component : SW-C) 化し、SW-C の下位層は仮想機能バス (Virtual Functional Bus : VFB) とするアーキテクチャで構成する、階層化されたソフトウェア構造を特徴としている。

ハードウェアの違いを吸収する VFB は SW-C にハードウェアやネットワークを意識させることのない環境を提供しており、SW-C のインターフェースが決まれば、あとは VFB が SW-C をどのように接続するかを設計し、実装するだけで済む。

VFB と SW-C との間の接続(設計)作業は、AUTOSAR に準拠したソフトウェア開発ツール(アプリケーション設計ツール)を用いて行われる。

このように AUTOSAR が提唱するソフトウェアプラットフォームを用いると、アプリケーションソフトウェアの組み換えが容易になるなどのメリットがあるため、欧州カーメーカを中心に導入が進んでおり、サプライヤにとっても AUTOSAR 規格への対応が早急に求められている。

2. 課題

アプリケーション設計ツール(GUI 対応)を用いることにより、アプリケーションソフトウェアそのものの開発及び、組み換えが容易になる反面、新たに SW-C の追加を必要とするソフトウェアの更新作業には、下位層にあたる VFB の変更が伴う。このため、SW-C を追加する度に VFB の生成を再度設計し直す必要が生じるため、開発工数が増加してしまう課題がある。

また、アプリケーション設計ツールを用いるには、ソフトウェア更新前の状態でのハードウェア構成及び SW-C の設計方法を認識している必要があるため、各 SW-C 単独での機能向上のためのソフトウェア更新を行うことが困難である。

VFB により、ハードウェアを意識せず SW-C の開発を行えるものの、製品出荷後に新しいハードウェアを接続し、構成が変わってしまう場合、それを利用するためのアプリケーションソフトウェアの追加が難しいという課題がある。

3. アプリケーションサブレイヤ(ASL)

3.1. ASL の構成

既存アプリケーションの SW-C(既存アプリ SW-C)及び VFB に対し、後から追加するアプリケーション(追加アプリモジュール)を可能とし、かつ、追加アプリモジュールの動作が既存アプリ SW-C の妨げとならないための保護機能を有するアプリケーションサブレイヤ(Application Sub Layer : ASL)の検討を行った。構成を図 1 に示す。

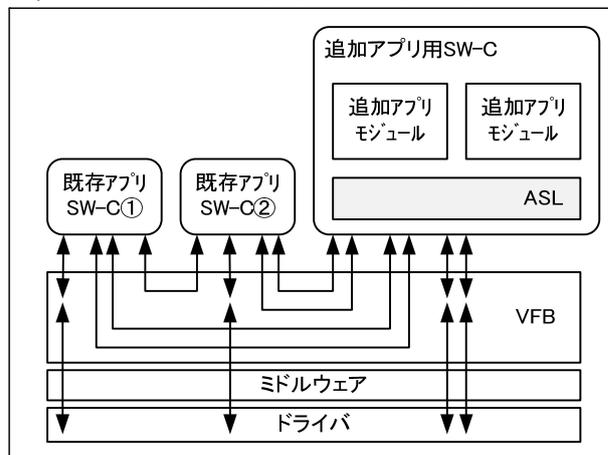


図1: ASL の構成

Installing Additional Software to Embedded Applications
Isamu Kawakami[†], Yoshiaki Katayama[†], Masao Ito[†] and
Yuichi Tokunaga[†]

[†]Information Technology R&D Center, Mitsubishi Electric
Corporation

5-1-1 Ofuna Kamakura Kanagawa Japan 247-8501

ASL は、システム全体に追加された、新たな SW-C の中に位置しており、追加アプリモジュールから既存アプリ SW-C 及びハードウェアへのアクセスは全て ASL を経由して行われる。

3.2. ASL の機能

アプリケーション設計ツールを用いてアプリケーションの設計を行う際、他の SW-C 及びハードウェアをアクセスするためのライブラリ関数が VFB で提供される (VFB-I/F)。同時に SW-C のソースコードもアプリケーション設計ツールによって出力されるが、このソースコードには、ハードウェアやメモリ等のリソース及び他の SW-C へのアクセスなど、実際に処理を行う部分が抜けているため、アプリケーションの開発者は VFB I/F を用いて、SW-C をコーディングする。

しかし、追加アプリモジュールのコーディングを行う際、既存アプリ SW-C と同様に VFB I/F を用いてしまうと、これまで動作していた既存アプリ SW-C に遅延を生じたり、最悪の場合、動作を停止させてしまう可能性が発生する。

ASL では、既存アプリ SW-C がアクセスしているメモリやハードウェアに影響が及ばないようにするための機能を拡張した、拡張ライブラリ関数(拡張 VFB-I/F)を提供する(図 2)。

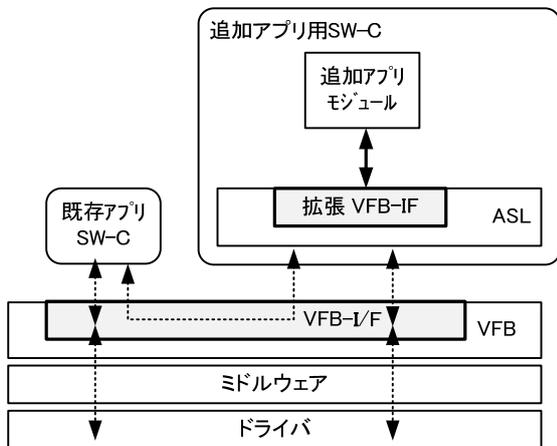


図2: 拡張 VFB-I/F

3.3. ASL の生成

ASL は、他の既存のアプリケーションと同じ、SW-C の一つとして実装される(追加アプリ用 SW-C)。このため、ASL を生成するには、アプリケーション設計ツールを用いる必要がある。その際、追加アプリ用 SW-C と、ツール上に表示されている既存アプリ SW-C 及びハードウェアのコンポーネント全てと接続する。このような手順を行うことにより、実在するハードウェア及び拡張用に用意されているハードウェアに対し、アクセスが可能となるソースコードの枠組み及び、VFB

側にはこれら既存アプリ SW-C 及びハードウェアをアクセスするための I/F が生成される。

アプリケーション設計ツールを用いて ASL を設計する場合の例を図 3 に示す。

図の例では、既存アプリ SW-C が 2 つあった場合を想定している。ASL はこれら既存アプリ SW-C と同様、一つの SW-C として取り扱っており、システム上に実在する全ての SW-C 及びハードウェア (D1~Dn)、また今後接続される可能性がある拡張用ハードウェア (R1~R3) と接続されている。

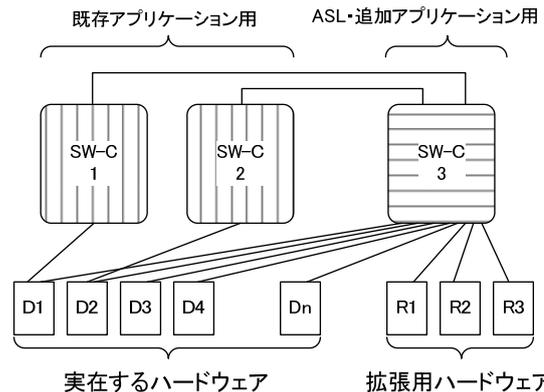


図3: アプリケーション設計ツールを用いた ASL の生成方法

4. ソフトウェアの追加方法

追加アプリ用 SW-C 内には、接続されているハードウェア毎に、メイン処理用のルーチンが用意される。追加アプリモジュールの開発者には、このメイン処理ルーチンへ登録を行うための関数及び、本来アプリケーションを生成する際 SW-C 内で記載するのに用いる VFB-I/F に対し、既存アプリケーションに影響を与えないための変更(特定のメモリやハードウェアに対する保護)を施した、拡張 VFB-I/F を提供する。

従来、新たなアプリケーションを追加するには、アプリケーション設計ツールを用い、VFB の生成を再度設計し直す必要があった。ASL を用いることにより、メイン処理ルーチンへ登録するための関数及び拡張 VFB-I/F を利用し、新たなアプリケーションモジュールのコーディングのみ行うことで追加が可能となる。

5. 今後の課題

VFB を変更せず、アプリケーションモジュールの追加を可能とする ASL の検討を行った。今後は、ASL の実装開発を行う予定である。

6. 参考文献

- [1] <http://www.autosar.org/>
- [2] 車載ソフトウェアの標準化と AUTOSAR の動向, 徳田昭雄・田村太一, 『立命館経営学』2007年1月