

# CAN を用いた階層統合型車載ネットワークの提案

徳永雄一<sup>1</sup> 西山博仁<sup>1</sup> 千田修一郎<sup>1</sup>

**概要:** 自由な拡張、未知の進化に対応する車載システムアーキテクチャ実現に向け、ECU とデバイス間の I/O 配線をネットワーク化し、これを ECU 間の情報共有手段である CAN に統合した階層統合型ネットワークを提案する。I/O 配線量を削減し、機器の設置自由度を増すことで、交換や追加が可能となり、H/W の進化に対応できる。本稿では、提案を実現する通信方式を説明するとともに、階層統合の実現性をシミュレーション評価し、自動車への適合が可能であることを示す。さらに、AUTOSAR 上への実装方式を考察し、H/W の進化に S/W が追従できることを示す。

## Proposal of In-vehicle Network Integrating Multiple Control Levels on CAN

YUICHI TOKUNAGA<sup>1</sup> HIROHITO NISHIYAMA<sup>1</sup> SHUICHIRO SENDA<sup>1</sup>

### 1. はじめに

自動車は近年、自動運転、コネクテッド、人工知能など新たな技術が次々と導入され、安全性、快適性、利便性の全ての分野において機能進化が加速している。一方で、ユーザの買い換え周期は伸びる傾向にあり[1]、保有する車の機能進化へのニーズが高まっている。その一例として、テスラモーターズは、OTA (Over The Air) を使ったソフトウェアアップデートにより保有する車が進化できることを実証している[2]。しかし、この例では、アップデートを想定した配線やセンサなどのハードウェアがあらかじめ備えられており、これらを制御するソフトウェアが後からリリースされたにすぎない。自由な拡張、未知の進化に対応するためには、システムアーキテクチャからの見直しが必要である。

我々は、CAN (Controller Area Network) で結ばれている自動車の制御ネットワーク構成に着目した。CAN は車両全体の各所に配置した ECU (Electronic Control Unit) の情報共有手段であり、多重伝送により省配線化が図られ、ECU の増減に柔軟に対応できる。一方、その下の階層、すなわちセンサやアクチュエータなどの I/O デバイスの制御階層においては、専用配線で繋がれ、I/O デバイスの多い車種では、その重量は 50kg にも及ぶ[3]。大量の配線の引き回しにより車両のハードウェア配置は固定化され、自由な変更や増設を妨げているといえる。

これを解決するために、ECU と I/O デバイス間の配線をネットワークで仮想化し、これを ECU 間の情報共有手段である CAN に統合する階層統合型ネットワークの提案を行う。車両の要所にデバイス HUB と呼ぶ I/O デバイスの接

続ポイントとなる装置を置き、これを CAN で結ぶことで、I/O 配線を CAN で仮想化する。I/O デバイスは ECU の位置に関係なく、近傍 HUB に接続できるので、I/O 配線量を削減できるだけでなく、設置自由度が増し、交換や追加が可能となり、ハードウェアの進化に対応できる。既存 CAN を使うことから、アーキテクチャの構造変化が少なく、実績による信頼性を担保しながら進化させることができるのも特徴として挙げられる。

以下、2 節でターゲットシステムの定義と課題整理を行い、3 節で課題を解決するための通信方式を示す。4 節で階層統合の実現性をシミュレーションによるネットワーク負荷および応答遅延特性から評価し、自動車への適合が可能であることを示す。さらに 5 節で AUTOSAR[4]への実装方式を考察し、H/W の進化に S/W が追従できることを示す。

### 2. ターゲットシステムと課題

車載システムには、エンジンやトランスミッションなど走りの制御に関わるパワートレイン制御系、ブレーキやステアリングなど止まる、曲がるの制御に関わるシャーシ系、ワイパー、ドアロックなど運動制御以外の電装系制御に関わるボディ系の 3 つの独立したドメインが存在する。各ドメインが独立したネットワークシステムで構成され、各ドメインシステムはゲートウェイを通じて情報連携される。この他に、カーナビゲーションや音・映像に関わる情報系、緊急ブレーキなどの安全系が車種により追加される。

本稿では、ボディ系ドメインをターゲットとする。他ドメインに比べ、制御デバイスの数・種類が多く、車種・グレード・年式によるデバイス構成が多様なことから、自由

1 三菱電機株式会社 情報技術総合研究所

な拡張、未知の進化への対応ニーズが最も高いことがその理由である。

ボディ系システムは、上述したワイパー、ドアロック等を制御する ECU の他、エアコン ECU、電子キーシステム ECU、ステアリングロック ECU 等、それぞれ機能毎に ECU があり、機能に関連した I/O デバイスが、該当する ECU と I/O 配線で接続されている。ECU 間は CAN で接続され、イグニッションや車速等の情報を共有している[5][6]。ボディ系システムは、モデルチェンジ毎に新たなデバイスが導入される。当初は独立した ECU で実現されることがあるが、やがて低コスト化のため他の ECU と統合される。このとき、ECU と I/O デバイスの位置関係が問題となる。専用 ECU で制御している場合には、ECU を I/O デバイス近傍に設置することで、I/O 配線長はミニマムとなるが、ECU を統合すると、配置の自由度は制限され、I/O 配線長が増大する。I/O デバイスによっては数十点の I/O を備えており、ハーネスコストの増加と車重増による燃費低下に加え、組み立て生産性や保守時の作業性悪化にもつながる。このような統合による弊害を最小限に食い止めるため、配線臙装を考慮した ECU の再配分や設置位置決めを行う必要があり、I/O デバイスの変更や増設の自由度を妨げている。

### 3. 提案方式

前節で述べた課題を解決するために、ネットワーク技術を用いて多重化した信号を伝送することが有効であると考えられる。ただし、I/O デバイスそのものを変更することはできないので、図 1 に示すように、車体の要所にアクセスポイントとしてデバイス HUB を設置し、近傍の I/O デバイスを接続、デバイス HUB と統合 ECU をネットワーク接続する形態を考える。本節では、ネットワーク要件を定義し、これを実現する通信方式を導く。

#### 3.1 ネットワーク要件

ネットワーク技術の適用にあたり、車載システムとして満たさなければならない必須要件について以下に記す。

##### (1) リアルタイム性

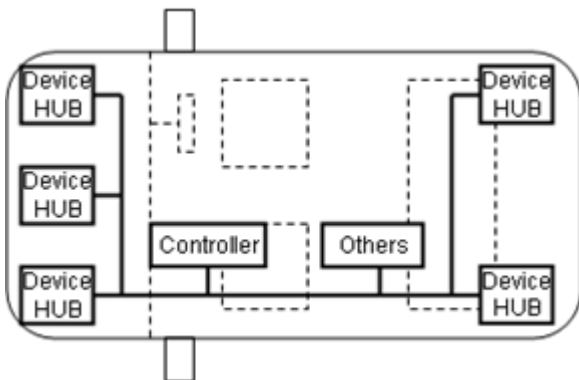


図 1. デバイス HUB を用いたネットワーク構成

ワイパーやドアロックなどの車載システムの多くの機能は、スイッチ操作してアクチュエータを駆動するものである。操作から駆動まで人間工学的に不快感を与えない応答時間は 100ms 程度といわれている[7]。スイッチ操作には、チャタリング除去のための 30~50ms 程度のフィルタを設けることが一般的であることを考慮すると、50ms 程度のリアルタイム性が必要となる。

##### (2) 扱う I/O デバイス数と種別

現状の車載システムに搭載されている I/O デバイス数は、100~150 程度である[8][9][10]。各 I/O デバイスは、複数の I/O 信号を有しており、I/O 点数の総計は 400 点前後となる。インタフェースの種別としては、約 90% が接点 I/O で、センサ類のアナログ入力、モータや照明などのアクチュエータの PWM (Pulse Width Modulation) が残り 10% を占める。データ長は接点 I/O が 1bit、それ以外は 10~16bit である。

##### (3) 通信規格

本稿目的から、CAN を第一候補とする。なお、実績のある車載ネットワーク規格として、信頼性とリアルタイム性を強化した FlexRay や、I/O デバイス階層でのネットワークである LIN (Local Interconnect Network) が挙げられる。FlexRay は、X-by-Wire を目的とした通信規格であり、高レベルの機能安全に対応できる。反面、実装コストが高く[11]、今回ターゲットとしたボディ系では使われることが少ない。LIN は、その対極にあり、低コストにデバイスを省配線に繋ぐことを目的としており、低性能なため、ECU 間通信には適さない。

以上をもとに、提案実現に必要なネットワーク要件を表 1 に示す。

#### 3.2 通信方式の検討

表 1 の条件を満たす ECU とデバイス HUB 間の通信方式を検討する。

車載システムでは、装置間の通信は、定期的にデバイスの入力情報や制御情報を交換するのが一般的である。例えば、ヘッドランプ制御では、コラムスイッチが接続された

表 1. 提案実現に必要なネットワーク要件

項目	条件
通信規格	CAN(伝送レート 500Kbps). 標準フォーマット. スタッフビットはデータ 5bit に対してスタッフビット 1bit として計算.
通信周期	10ms
I/O デバイス数	150
I/O 点数	400 点
デバイス割合	(A)90%, (B)10%
ペイロード長	(A)3 バイト, (B)4 バイト (安全通信ヘッダ 16bit を含む)

(A)は接点入力デバイスの条件

(B)はアナログ入力、PWM 出力デバイスの条件

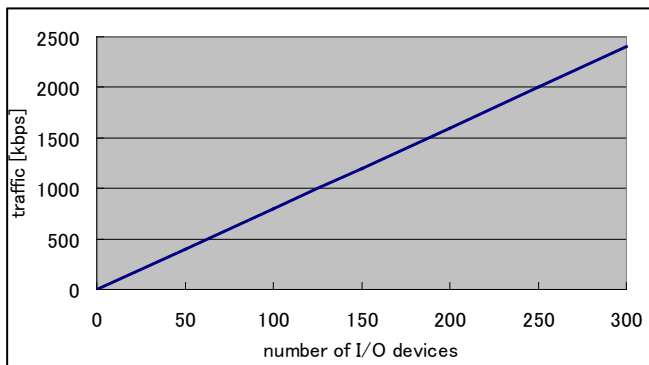


図 2. I/O デバイス数と伝送性能の関係

デバイス HUB から統合 ECU に定期的にコラムスイッチ情報が送信され、統合 ECU でその情報を使って計算したヘッドライト制御情報を、ヘッドライトが接続されたデバイス HUB に定期的に送信する。

定期的に情報交換を行うのは、エラーに伴うデータの取りこぼしの防止や、アライブ・チェックによる故障検出のためである。リアルタイム性要件を満たすために、通信周期は 10ms 程度である必要がある。

図 2 は、表 1 の条件下において I/O デバイス毎に CAN フレームを送信することを想定した場合に、I/O デバイス数に対する通信帯域の関係を示したグラフである。これによれば、I/O デバイス数の要件である 150 デバイスを扱うため必要な通信帯域は 1200Kbps となり、1 本の CAN (500Kbps) には収容できないことがわかる。

そこで、共有メモリ方式を適用した効率化を図る。これは、FA システム用のネットワークである FL-net[12]や CC-Link[13]に適用されている制御情報を共有メモリにおいて、サイクリック伝送によりデータ伝送する方式を、CAN の上位層に適用した方式である。

本方式を図 3 で説明する。ECU とデバイス HUB は、デバイスの I/O 情報を共有する共有メモリを有する。I/O デバイスへの出力は、ECU が共有メモリに制御情報を出力する。ネットワーク機能は、一定周期で定期的にデバイス HUB に共有メモリの内容を転送する。デバイス HUB は、共有メモリの内容に従って I/O デバイスに制御出力を行う。同様に I/O デバイスからの入力情報は、入力デバイスの信号を共有メモリに出力し、ネットワーク機能が一定周期で ECU に送信する。ECU 上の制御アプリケーション S/W は、共有メモリを読みだすことで、デバイスからの入力情報を得ることが可能となる。

共有メモリ方式により、I/O 情報は I/O デバイスに依存しない共有メモリ上のビットイメージとして転送されるため、情報をペイロード限界データ長まで詰め込むことができる。I/O デバイス単位にフレームを生成、伝送する方式に比べ高効率に伝送することができる。

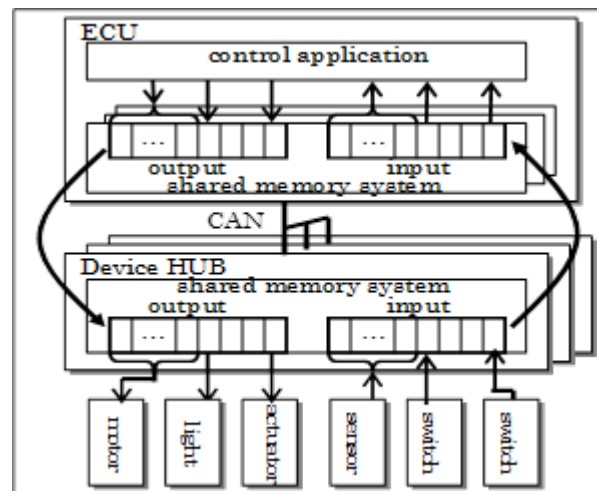


図 3. 共有メモリ方式を用いた伝送効率化

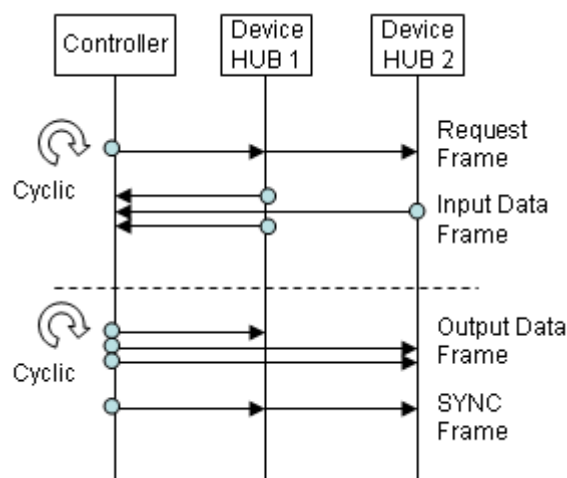


図 4. 通信方式

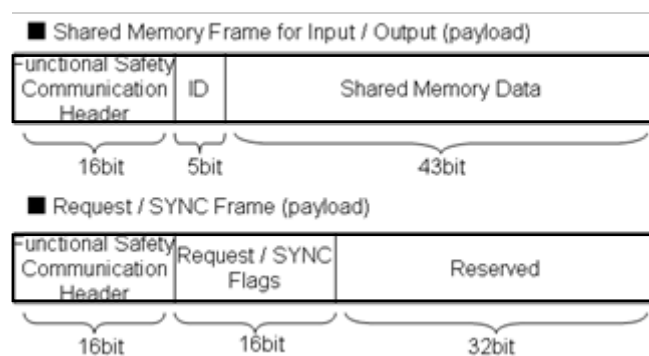


図 5. フレームフォーマット

### 3.3 提案する通信方式

以上の検討結果を踏まえ、CAN を用いた I/O 情報の通信プロトコルを図 4 に示す。入力情報の取得は、ECU が全デバイス HUB に対して入力データの共有メモリの送信を要求する Request フレームを周期的にブロードキャストすることで開始する。デバイス HUB は Request フレームへの応答として、コントローラに自身の入力用の共有メモリデータを送信する。一つのデバイス HUB がペイロードサイズ

を超えるデータサイズを持つ場合は、順に送信する。

出力情報の伝達に対しては、コントローラが各デバイス HUB に出力用の共有メモリデータを順次送信した後に SYNC フレームをブロードキャストする。デバイス HUB はあらかじめ受信していた出力情報を SYNC フレーム受信時に出力することで、異なるノード間で出力タイミングを同期できる。

図 5 に、CAN フレームフォーマットを示す。上図は入力／出力データフレームを表し、共有メモリデータを送受信する際に用いる。先頭から 16bit の機能安全通信ヘッダ、5bit の共有メモリ ID、43bit の共有メモリデータで構成される。機能安全通信ヘッダは CRC、メッセージ ID、シーケンスカウンタなどの情報から生成されるもので、通信経路上でのメッセージの消失や改ざんを検知する仕組みである。機能安全通信に関してはヘッダの領域は確保するが本稿では評価しない。

共有メモリ ID は 1 フレームで送信可能な 43bit のデータを 1 組の共有メモリとして区別するものである。通常 CAN 通信ではフレームヘッダ内の CAN ID によりメッセージの内容を判別するが、1 組の共有メモリにつき 1 つの CAN ID を利用すると CAN ID が不足する恐れがあるため、ペイロード内に ID を設け共有メモリは同一の CAN ID を利用可能とした。実際に CAN ID を決定するときは、本来のコントローラ間情報伝送の ID と衝突しないように割り振る。

共有メモリデータ部分は任意の接点 I/O (ON/OFF の 1bit) と拡張 I/O (アナログ値や Duty 比の 16bit) の情報を合わせて 43bit 分割り付けることができる。

図 5 の下図は Request/SYNC フレームを表す。16bit の Request/SYNC フラグは先頭 bit から順に共有メモリ ID と対応する。デバイス HUB は、Request フラグが True の共有メモリを ECU に返し、SYNC フラグが True の共有メモリを ECU からの情報で更新し I/O デバイスへ出力する。フラグを使い分けることで、周期時間の倍数で制御周期の異なる I/O を制御することができる。

## 4. 性能評価

提案する階層統合ネットワークは、CAN の本来の用途である ECU 間通信を実施しながら、同じ媒体を用いた I/O 通信の混在を特徴とする。そこでシミュレーション環境にネットワークモデルを作り、ネットワーク負荷として I/O デバイス通信と ECU 間通信が混在可能かを評価する。さらに、I/O 通信の応答遅延が実用に耐えるかを評価する。

### 4.1 評価環境

#### (1) シミュレーション環境

シミュレーションは Vector 社の CAN シミュレーションツール「CANoe」上に、各ノードのモデルを生成して行う。

#### (2) システムモデル

システム構成は、図 1 に示すように、デバイス HUB を車体の 4 隅とスイッチ類の多いダッシュボード付近に配置することで各機器への配線長を短く抑えることを想定する。すなわち、5ヶ所に配置したデバイス HUB 経由で接続された I/O デバイスを 1 つの ECU が CAN 経由に制御する構成とする。また、I/O 通信とは別に、ECU 間通信を想定したネットワークトラフィックを CAN に加える。

#### (3) 通信周期

I/O 通信においては、表 1 に示したとおり通信周期は 10ms とする。また各共有メモリの送信周期はメイン周期と同一とする。すなわち Request/SYNC フレームにおいて登録された共有メモリに対応するフラグは全て True とする。

#### (4) 通信優先度制御

CAN は CAN ID の大小によってメッセージ衝突時にどのメッセージを優先するかを決定する仕組みを持つ。すなわち、複数の CAN メッセージが同時に通信を開始しようとした場合、最も優先度の高いメッセージだけが通信を継続し、他の通信は再送を行うため、遅延が発生する。

本稿では ECU 間通信を I/O 通信よりも高い優先度に設定し、I/O 通信により ECU 間通信が遅延しない制御とする。

## 4.2 I/O 点数に対するネットワーク負荷

4.1 節の評価環境下において、接続する I/O 点数と通信量の関係性を評価した。その結果を図 6 に示す。I/O 通信の特性を見るため、ECU 間通信負荷はゼロとしている。

表 1 の要件に示した 400 点の I/O 点数におけるネットワーク負荷は 28% を占めることが分かる。特性が段階的な形状を取る理由は、I/O 点数が CAN メッセージに収容できる限界を超えたときに周期あたりの CAN メッセージが増えるためである。430 点を過ぎたところで負荷上昇率が高まっている理由は、デバイス HUB からの入力信号情報送信に、CAN メッセージを 2 回送る必要が出てくるためである。430 点より前は、ECU の出力送信情報送信の CAN メッセージのみが増加し、デバイス HUB からの入力信号情

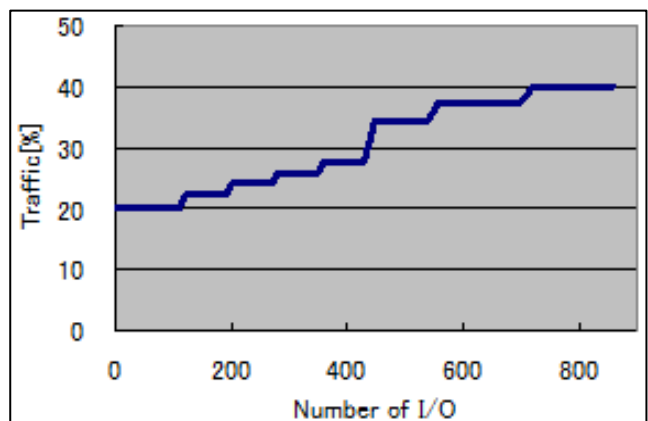


図 6. トラフィック特性

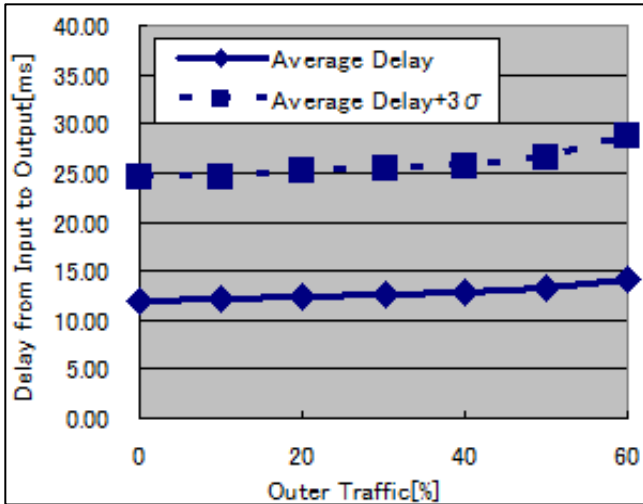


図 7. 遅延特性

送信は各 1 回であった。

CAN はイベント通信を前提とするネットワークであるため、エラー発生時など通信料が増大したときもリアルタイム性を確保できるよう帯域使用量が 60%程度になるよう設計される。すなわち、本評価結果は、ECU 間通信の最大負荷が 30%程度であれば、CAN を増設せずに I/O 通信を統合できることを示している。

#### 4.3 遅延性能

次に、ECU 間通信と統合した場合の、I/O 応答遅延を評価した。ここで I/O 応答遅延とは、デバイス HUB が入力デバイスから入力信号を得るところを起点に、ECU に情報伝送され、ECU での演算を経て出力指示がデバイス HUB から出力デバイスに伝達するまでの時間を指す。図 7 は、400 点の I/O 点数に対する伝送遅延である。横軸は、ECU 間通信の負荷を示す。

図 7 より、ECU 間通信負荷が 60%に達してもデバイスネットワークの遅延時間は 30ms に留まっていることが分かる。ボディ系ドメインでは、スイッチやセンサの入力に対しライトやワイパーが動作するまでの応答時間であり、100ms 以内に動作すれば違和感を与えないことを考えると、I/O 応答遅延として実用に耐えるといえる。

### 5. 実装方式

提案する階層統合ネットワークは、自由な拡張、未知の進化に対応することを目的としているので、I/O デバイスの変更や新規追加に対して、これを制御する S/W も容易に変更できることが望ましい。そこで、ECU の標準的な S/W プラットフォームである AUTOSAR に対し、容易な制御 S/W の変更を可能とする階層統合ネットワークの実装方式を示す。

図 8 に、AUTOSAR の基本構成を示す。アプリケーション

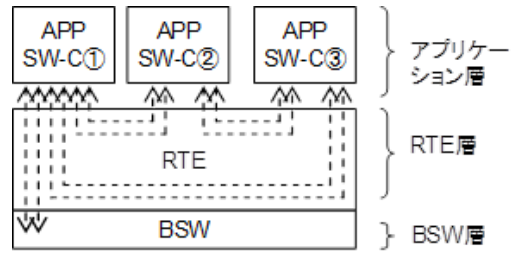


図 8. AUTOSAR の S/W 構成

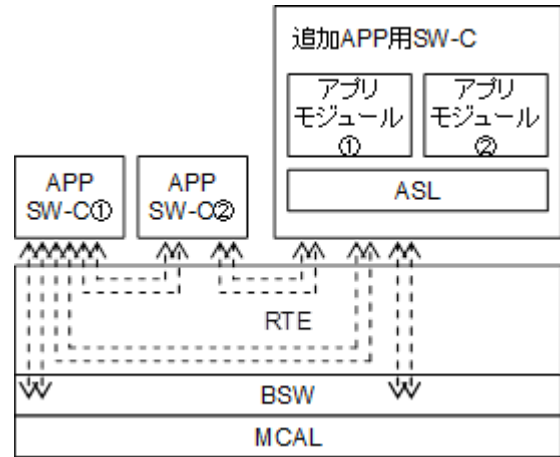


図 9. 階層統合型ネットワークの S/W 構成

ン層に実装される S/W は、S/W コンポーネント (SW-C) の単位で管理される。SW-C は、その下に位置する RTE (Runtime Environment) 層の仮想的なファンクションバスに合わせた設計をすることで、H/W や伝送経路を意識せず実装でき、別のマイコンへの載せ替え、ネットワークの変更時にも再利用可能としている。RTE で提供される SW-C のファンクションバスは、①他の SW-C ②BSW (Basic S/W) が提供するサービス ③CAN などのネットワーク ④自身のマイコンの I/O の情報通信に大別される。RTE は、構成管理情報をもとに各 SW-C のファンクションバスを①～④に割り付ける。②～④は BSW に伝達され、③は BSW 内でネットワークメッセージに生成され通信媒体へ、④は MCAL (Microcontroller Abstraction Layer) 経由で H/W の制御に至る。

図 9 に、階層統合ネットワークを適用した場合の実装構成を示す。SW-C と RTE の間に ASL (Application Sub Layer) を設け、ASL とその上位にある複数の SW-C を、RTE から見た場合に 1 つの SW-C として扱う構成とする。なお、SW-C の中には、初期化シーケンスや演算機能など、I/O デバイスの制御を伴わないものがある、これらは従来通り RTE 上に実装する。従来の RTE 上の SW-C と ASL 上の SW-C を区分するため、ASL 上の SW-C をアプリモジュールと呼ぶ。

ASL は、各アプリモジュールにおける前記④の I/O 情報の通信を管理する。各アプリモジュールの I/O 情報の通信は、ASL 上の共有メモリの読み書きとして処理される。ま

た、ASL は、RTE に対し前記③の CAN 情報の通信を要求し、本通信によって共有メモリ内容の送付と更新を行う。RTE 以下の層では、ECU 間通信と I/O 通信は同等の CAN メッセージとして処理される。

以上のように、I/O デバイスの直接制御から I/O 通信経由の制御への変更は ASL が吸収するので、従来 SW-C 資産が流用できる。I/O デバイスの追加に対して、ホスト ECU の S/W に対象デバイスの SW-C を加え、ASL の管理情報を更新することで追加デバイスを機能させることができる。

## 6. まとめ

本稿では、自由な拡張、未知の変化に対応する車載システム実現に向け、ネットワーク提案を行った。

多数の機器を制御する各 ECU は、CAN で結ばれ省配線での情報共有が実現されている一方、ECU と機器内のセンサやアクチュエータなどの I/O デバイスとは専用配線で結ばれ、大量の配線の引き回しにより車両のハードウェア配置は固定化され、自由な変更や増設を妨げている。そこで ECU と I/O デバイス間をネットワークで仮想化し、さらにネットワークの省配線化のために ECU 間通信で用いている CAN にこれを統合する。デバイス HUB を車上各所に配置し、CAN 経由で ECU と I/O 通信を行う。I/O デバイスは近傍のデバイス HUB に接続され、離れた場所に位置する ECU から制御される。I/O 配線の省配線化が進み、I/O デバイスの設置自由度が大幅に増す。

ECU 間通信用途の CAN に I/O 通信が加わることでネットワーク負荷が増大する。負荷低減に向け、I/O 情報を共有メモリに配置し、ECU とデバイス HUB 間で共有メモリのビットイメージを周期的に同期する通信方式を提案した。I/O デバイスの多いボディ系ドメインを例にシミュレーション評価した結果、ドメインで必要とされる 400 点の I/O を 30% 以下の負荷で伝送でき、ECU 間通信と統合可能であることを示した。また、I/O 通信の優先度を ECU 間通信よりも落とした設定で混在通信を行った時のデバイス I/O の応答遅延は 30ms 以下であり、実用に耐えることを示した。

AUTOSAR 上への実装を想定し、RTE と SW-C の間に、ASL を定義し、ここで I/O デバイス制御の仮想化を管理する提案を行った。I/O デバイスの追加に対して、ホスト ECU の S/W に対象デバイスの SW-C を加え、ASL の管理情報を更新することで追加デバイスを機能させることができる。

コネクテッドカーの時代となり、自動車がクラウドと常時接続できる環境を利用すると、デバイス HUB に接続される I/O デバイスの変化をクラウドに通知し、クラウド上で ECU の S/W をリビルド、車上 ECU を S/W アップデートし機能開始させることが可能となる。このような車両ごとの個別進化に向け、仮想環境下での S/W 管理、検証および車上での機能安全対応の検討を進めていく予定である。

## 参考文献

- [1] 内閣府経済社会総合研究所景気統計部: "消費動向調査",平成 29 年 3 月実施調査結果",内閣府, pp.11 (2017)
- [2] Susan Beardslee, James Hodgson, Dominique Bonte: "Automotive Over the Air Updates: Technologies, Applications, and Business Models", Application Analysis Report, AN-2252, ABI Research (2016)
- [3] 佐藤弘明, 神谷有弘: "自動車 ECU の市場・技術動向徹底解説", 電子ジャーナル, Electronic Journal Archives, No.125, pp.52 (2011)
- [4] <http://www.autosar.org/>
- [5] 加藤光治, デンソーカーエレクトロニクス研究会: "図解カーエレクトロニクス[上]システム編",日経 BP 社, 第 5 章 (2010)
- [6] 佐藤道夫: "車載ネットワーク・システム徹底解説", CQ 出版社 (2005)
- [7] Jakob Nielsen: "Response Times: The 3 Important Limits", <http://www.nngroup.com/articles/response-times-3-important-limits/>, January 1 (1993)
- [8] LEAF(ZAA-E0), T00SM3N1J, 日産自動車株式会社
- [9] i-MiEV(HA-3W), 1034U92, サービスマニュアル, 三菱自動車工業株式会社
- [10] Accord(DBA-CU2), 60TL160, サービスマニュアル, 本田技研工業株式会社
- [11] 伊東維年: "カーエレクトロニクス化の進展とその課題",熊本学園大学産業経営研究, 29 号, pp.65-88 (2010)
- [12] ネットワーク推進特別委員会 FL-net 仕様サブワーキング: "FL-net Ver3.00 プロトコル仕様書", 社団法人日本電機工業会 (2009)
- [13] 楠和浩, 中野宜政: "ファクトリーオートメーション(FA)用オープンフィールドバス", 計測と制御, Vol.44, No.6, pp.365-370 (2005)