

推薦論文

# 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,a)</sup> 西山 博仁<sup>1</sup> 伊藤 益夫<sup>1</sup> 千田 修一郎<sup>1</sup>

受付日 2018年2月9日, 採録日 2018年8月9日

**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 配線を多重化し, 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, ハードウェアの進化に対応できる. 本稿では, 提案を実現する通信方法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, I/O 制御に必要な帯域, 応答性を確保できていることを示す. さらに, AUTOSAR 上への実装とシステム評価によって, 車載システムに適合できることを示す.

**キーワード:** 車載ネットワーク, 省配線, CAN, AUTOSAR, リアルタイムシステム

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

YUICHI TOKUNAGA<sup>1,a)</sup> HIROHITO NISHIYAMA<sup>1</sup> MASUO ITO<sup>1</sup> SHUICHIRO SENDA<sup>1</sup>

Received: February 9, 2018, Accepted: August 9, 2018

**Abstract:** For realization of in-vehicle system architecture corresponding to free extension and future evolution, we propose an in-vehicle network which multiplex the I/O wiring between an ECU and the devices and integrates it into CAN, which is information sharing instrument between ECUs. It can cope with the evolution of hardware by reducing the amount of I/O wiring and increasing the degree of freedom of installation of I/O devices. In the paper, we explain the communication method to realize the proposal and show that it satisfies the bandwidth and responsiveness required for I/O control by evaluating the feasibility of hierarchical integration by simulation. In addition, we show that it is compatible with in-vehicle system by the implementation on AUTOSAR and system evaluation.

**Keywords:** in-vehicle network, cable reduction, CAN, AUTOSAR, real-time systems

### 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 1 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ものである. 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

<sup>1</sup> 三菱電機株式会社  
Mitsubishi Electric Corporation, Chiyoda, Tokyo 100-8310, Japan

<sup>a)</sup> Tokunaga.Yuichi@ds.MitsubishiElectric.co.jp

本稿の内容は 2017 年 6 月のマルチメディア, 分散, 協調とモバイル (DICOMO2017) シンポジウムで報告され, 高度交通システムとスマートコミュニティ研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である.

と ECU 間の I/O 制御階層においては、デバイスごとに専用線でつながれており、I/O デバイスの多い車種の配線重量は 50 kg にもおよぶ [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多重化し、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。車両の各所にデバイス 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 で結び、I/O 信号を多重化する。I/O デバイスの配線は近傍 HUB に接続されるため、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あげられる。

以下、2 章でターゲットシステムの定義と課題整理を行い、3 章で課題を解決するための通信方式を示す。4 章でシミュレーション評価から車載システム要件を満たすことを示し、5 章で AUTOSAR [4] プラットフォーム上への実装と実機評価により車載システムに適合できることを示す。

## 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、曲がる制御に関わるシャーシ系、ワイパ、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。このほかに、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、ドアロックなどを制御する ECU のほか、エアコン ECU、電子キーシステム ECU、ステアリングロック ECU など、それぞれ機能ごとに 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などの情報を共有している [5], [6]。ボディ系システムは、モデルチェンジごと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

は I/O デバイス近傍に設置されることが多く、I/O 配線長も短い。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、配線構築を考慮した 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

2 章で述べた ECU の統合化における配線増の課題を解決するためには、ネットワーク技術を用いて信号を多重化し、信号本数を削減することが有効である。ただし、各 I/O デバイスのインタフェースを変更することはコスト面から難しい。そこで、図 1 に示すように、車体の各所にアクセスポイントとしてデバイス HUB を設置し、周辺にある I/O デバイスを現在のインタフェースでこれに接続する。デバイス HUB が各 I/O デバイスのインタフェースをネットワークプロトコルに変換し、同じネットワーク上の統合 ECU と情報交換することで I/O デバイスの制御を実現する。本構成は、現行構成に対しても配線量を削減できる。ある車両を例にボディ系の 3 種類の ECU とこれに接続される I/O デバイスとの信号配線距離を調査し、デバイス HUB への接続に変えたときの配線距離を試算した結果、現行に比べ配線総距離は 48%削減できる見通しを得た。

本章では、デバイス HUB を介した I/O デバイスと統合 ECU のネットワーク接続に焦点を当て、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 ネットワーク条件

車載システムへの適用にあたり、ネットワークが満たさなければならない必須要件を表 1 に示し、以下で説明する。

#### (1) リアルタイム性

ワイパ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

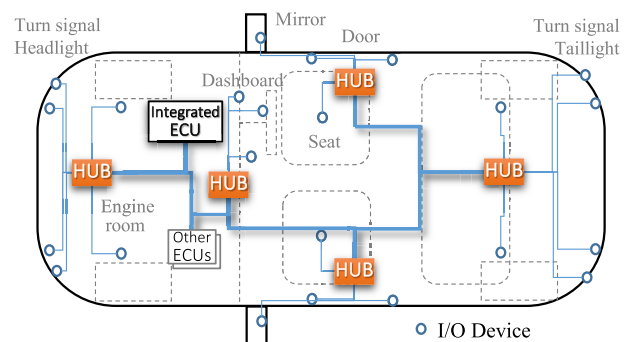


図 1 デバイス HUB を用いたネットワーク構成  
Fig. 1 Network structure using device HUB.

表 1 提案実現に必要なネットワーク要件

Table 1 Network requirement to realize the proposal.

項目	条件
通信規格	CAN2.0 標準フォーマット(11bit-CAN ID) ISO-11898(伝送レート 500kbps)
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%
データ長	(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

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 100 ms 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 30~50 ms 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、50 ms 程度の入出力応答時間が必要となる。

(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%が導通/開放の 2 つの状態を示す接点 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 1 bit, それ以外は 10~16 bit である。

(3) 通信規格

一般的な車載ネットワークは CAN 2.0 [11] に準拠している。ID フィールドが 11 bit から 29 bit に拡張された仕様も規定されているが、ボディ系では標準的な 11 bit を適用することが多い。物理層は ISO11898 [12] 規定の Hi speed CAN が適用されている、規定上の最大転送速度は 1 Mbps であるが、実用的には 500 kbps で使われている。

なお、車載ネットワーク規格としては、CAN のほかにも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) が使われている。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 [13]、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つな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。

3.2 標準 CAN による I/O 通信

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

CAN による ECU 間通信では、データを持つ ECU が非同

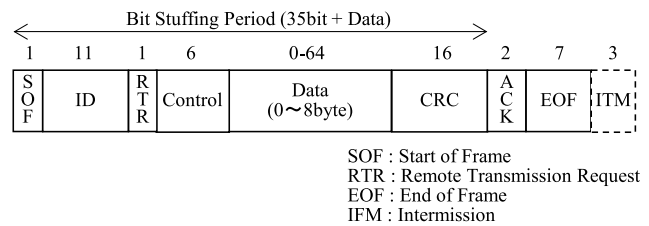


図 2 CAN フレームフォーマット

Fig. 2 CAN frame format.

期にメッセージを送信する。図 2 の CAN メッセージに含まれる 11 bit の CAN ID は、ネットワーク上での送信権の調停とメッセージの識別子として用いられる。送信権の調停は、CSMA/CR (Carrier Sense Multiple Access/Collision Resolution) と呼ばれるアクセス制御を用いる。すなわち、複数の同時送信が発生した場合に、CAN ID の送信フェーズで調停が行われ、最も大きな数値を示す CAN ID のメッセージが優先送信される。各 ECU は CAN ID に基づき受信判断し、アプリケーションで規定したフォーマットに従いデータを参照する。

これを統合 ECU とデバイス HUB 間の通信に適用することを考える。ヘッドライト制御を例とすると、コラムスイッチが接続されたデバイス HUB は、スイッチ接点情報を CAN メッセージにして統合 ECU へ送信する。統合 ECU はスイッチ情報をもとにヘッドライト制御情報を CAN メッセージにし、ヘッドライトが接続されたデバイス HUB に送信する。なお、データのとりこぼし防止やアラブ・チェックのため、定期的に通信を繰り返す。通信周期は制御周期と同じ 10 ms が一般的である。

ここで、I/O デバイスごとに上記通信を行う場合に必要な帯域を求める。CAN は図 2 の情報に加え、SOF から CRC までの領域に対しスタッフビットと呼ばれる冗長ビットが付加されるため、1 回の CAN メッセージの最大送信時間  $C_i$  はデータバイト数  $s_i$  とビットあたりの送信時間  $\tau_{bit}$  を用いて式 (1) で示される [14] :

$$C_i = \left( \left\lfloor \frac{35 + 8s_i - 1}{4} \right\rfloor + 46 + 8s_i \right) \tau_{bit} \quad (1)$$

ここから、通信周期  $T$  [ms] ごとに  $m$  個の CAN メッセージを送信した場合のネットワーク負荷  $L$  [kbps] を求めると式 (2) で示される :

$$L = \frac{\sum_{i=1}^m \left( \left\lfloor \frac{35 + 8s_i - 1}{4} \right\rfloor + 46 + 8s_i \right)}{T} \quad (2)$$

式 (2) を用いて、I/O デバイスごとに CAN フレームを送信した場合の I/O デバイス数に対するネットワーク負荷の関係を図 3 に示す。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 1,200 kbps を超え、1 本の CAN (500 kbps) には収容できないことが分かる。

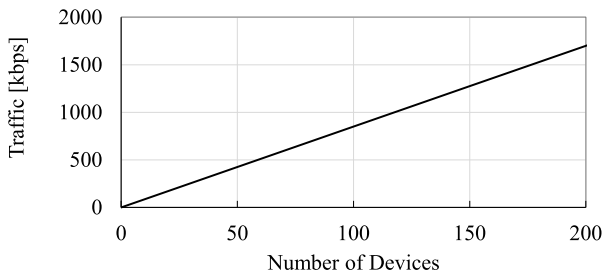


図 3 I/O デバイス数と伝送性能の関係

Fig. 3 Network traffic for number of I/O devices.

### 3.3 通信方式の検討

CAN の標準的な通信方式を適用した場合、表 1 で要求される 150 デバイスの I/O 制御に CAN の 2.4 倍の通信帯域が必要なが分かった。安定して通信するためには帯域の 60%程度で設計する必要があることを考えると、4 倍以上の通信帯域が必要となる。

帯域を増やす方法として、BOSCH 社から CAN FD (CAN with Flexible Data rate) [15] が提案されている。フレーム内のデータフェーズのみを高速化し、Hi-Speed CAN の 4 倍となる 4Mbps まで対応する。CAN FD は、ECU の S/W 書き込みの高速化ニーズにともない S/W コードサイズの多い高機能な ECU 向けに徐々に適用されつつあるが、高価な専用トランシーバが必要となるためすべての ECU への普及には至っていない。また、各 I/O デバイスはただか 1~2byte の情報で構成されるためデータフェーズのみを高速化する効果は低い。

別の方法として、1 つのデータフレームに収容する情報量を増やし、通信量を減らす方法が、FA システム用のネットワークに適用されている [16], [17]。ネットワークを介してコントローラと I/O デバイスが接続される分散システムにおいて、各ノードはそれぞれ共有メモリを備え、周期的に各共有メモリを同期することで I/O 情報を共有する。共有メモリの同期において、コントローラは全 I/O デバイスへの出力制御値をビットイメージデータとしてメッセージにまとめ、ネットワークへブロードキャストする。各 I/O デバイスは受信したメッセージから自身のデバイスに割り当てられたビット情報を取得し、デバイスを制御する。I/O デバイスごとに伝送する方法に比べ、1 メッセージで複数の I/O デバイス情報を伝送できる。

本方式を車載システムに適用した場合、図 4 に示すように、統合 ECU とデバイス HUB に、デバイスの I/O 情報を共有する共有メモリを備える。統合 ECU は、デバイスごとに割り当てられた共有メモリビットに出力制御値を書き込み、デバイス HUB は、入力デバイスのステータスを共有メモリに書き込む。各共有メモリデータを定期的に CAN メッセージで同期することで、統合 ECU は各デバイス HUB 下の入力デバイス値を、デバイス HUB は統合コントローラからの出力デバイス制御値を自身の共有メモリ

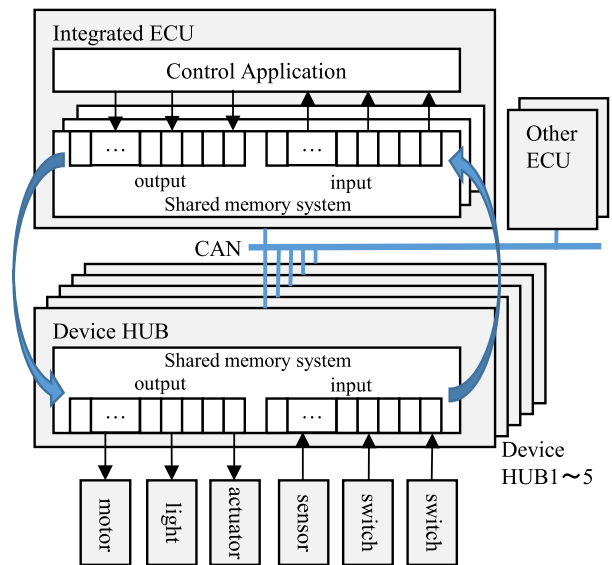


図 4 共有メモリ方式

Fig. 4 Shared memory method.

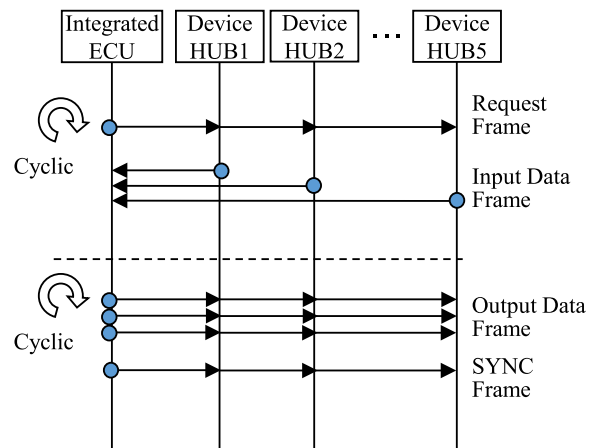


図 5 通信方式

Fig. 5 Communication protocol.

から取得することができる。CAN メッセージで伝送できるデータ量は 64 ビットであるのに対し、3.1 節で示したように 90%の I/O デバイスは 1~数ビットの接点情報であるので、複数の I/O デバイスをまとめて伝送することによる CAN メッセージの削減効果は高い。一方、変化の有無に関係なく全情報を周期的に伝送する必要があるため、I/O の変化が少ない状況においては、I/O デバイスの変化に応じて非同期に伝送する従来方式の方がネットワーク負荷を減少できることもあるが、I/O の変化が重なるピーク時の負荷を削減できるので、状況の変化によらずネットワークを平準化でき、全 I/O 情報を確実に伝送することができる。

### 3.4 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 5 に示す。入力情報の取得は、統合 ECU が全デバイス HUB に対して共有メモリ上の入力データの

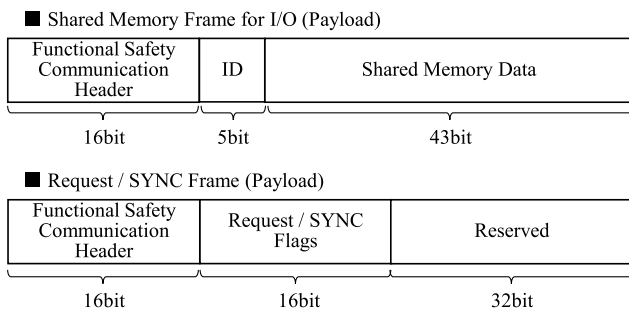


図 6 データフレームフォーマット  
Fig. 6 Data frame format.

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、統合 ECU に自身の共有メモリ上の入力データを送信する。1つのデバイス HUB がペイロードサイズを超える入力データを持つ場合は、共有メモリ ID の異なる複数のフレームにデータを分割し、順に送信する。

出力情報の伝達に対しては、統合 ECU が各デバイス HUB に共有メモリ上の出力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 I/O デバイスへ出力することで、異なるノード間で出力タイミングを同期できる。

図 6 に、CAN データフィールドにおける本通信方式のデータ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 16 bit を機能安全通信ヘッダ、5 bit を共有メモリ ID として確保し、残りの 43 bit を共有メモリデータ領域とする。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。

共有メモリ ID は、共有メモリ上に配置される 43 bit 単位のデータの識別子である。5 bit = 32 種の 43 bit データを識別し、合計 1,376 bit までの I/O 情報に対応し、将来の I/O 点数の増加に備える。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、共有メモリごとに CAN ID を消費すると ECU 間通信との統合において CAN ID が不足する恐れがあるため、入力/出力データフレームにおいては、ペイロード内に ID を設け、CAN ID とは非依存とする。本通信方式の CAN ID は、送信権の調停のため、各送信者を区分する ID として用いる。共有メモリ別に CAN ID を使う場合に比べ、CAN ID の消費を最大 32 から 5 つに減らすことができる。

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

図 6 の下図は Request/SYNC フレームのデータフォーマットを表す。16 bit 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 の共有メモリを統合 ECU に返し、SYNC フラグが True の共有メモリを統合 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 を制御できる。

### 3.5 提案方式のリアルタイム性

3.2 節で述べたとおり、CAN における CAN ID は、送信衝突時の送信者を定めるための優先度の意味を持つ。I/O 情報の通信を CAN に統合するうえで、CAN ID の割当てはシステムのリアルタイム性に影響を与える課題である。

CAN 上に 1 つの統合 ECU と複数の HUB が存在し、I/O 情報通信だけが実行される場合、Request フレームに回答する複数 HUB からの共有メモリデータ送信で衝突が発生する。このとき、CSMA/CR に従い最も大きな CAN ID を持つ HUB の送信データが伝送される。Request フレームへの各 HUB の応答数は固定であるため、高い CAN ID を持つ HUB の送信が終わると次に高い CAN ID を持つ HUB の送信が実行され、一定時間内に全応答が完了する。したがって、本ケースの場合は CAN ID の割当てに依存せず、リアルタイム性は維持される。

次に、ECU 間通信と混在する場合を考える。この場合、ECU 間の非同期通信と統合 ECU-HUB 間の通信で衝突が発生するため、それぞれの優先度を考えた CAN ID の割当てが必要となる。とくに I/O 情報通信のタイミングトリガおよび出力デバイスの制御を司る統合 ECU の CAN ID はリアルタイム要求の厳しい ECU 間通信の次に高い優先度とすべきである。以下、ECU 間通信と I/O 情報通信を統合する際の CAN ID の割当て方針を示す。番号が小さい程、高い CAN ID を示す。

- ① ECU 間通信の高リアルタイム要求 CAN メッセージ
- ② 統合 ECU が送信する I/O 通信メッセージ (Request/SYNC/出力データフレーム)
- ③ 高リアルタイム要求の入力デバイスを持つ HUB が送信する I/O 通信メッセージ (入力データフレーム)
- ④ ECU 間通信の ① 以外の CAN メッセージ
- ⑤ HUB が送信する ③ 以外の I/O 通信メッセージ (入力データフレーム)

上記に従い CAN ID を割り当てた場合のジッタについて検討する。ある通信周期  $T$  [ms] の ①、④ に該当するメッセージ伝送負荷 [kbps] をそれぞれ  $L_1$ ,  $L_2$ 、ビットあたりの送信時間を  $\tau_{bit}$  とすると、I/O 通信のジッタが最大となるのは ECU 間通信が I/O 通信の Request から SYNC フレーム送信までのシーケンス中に連続発生する場合である。すなわち、各フレームのジッタ最大値は以下で示される。

$$\begin{aligned} \text{Request フレーム送信ジッタ} & \Delta D_r = T L_1 \tau_{bit} \\ \text{入力データ送信ジッタ} & \Delta D_i = T(L_1 + L_2) \tau_{bit} \end{aligned}$$

出力データ送信ジッタ  $\Delta D_o = T(L_1 + L_2) \tau_{bit}$   
 SYNC フレーム送信ジッタ  $\Delta D_s = T(L_1 + L_2) \tau_{bit}$   
 $(L_1 + L_2) \tau_{bit}$  は ECU 間通信による CAN の占有率であるので、入出力データ、SYNC フレーム送信ジッタ最大値は、ECU 間通信負荷と周期時間の積で示される。すなわち、 $T = 10 \text{ ms}$ 、 $L_1 = 50 \text{ kbps}$  (CAN 占有率 10%)、 $L_2 = 250 \text{ kbps}$  (同 50%) の入出力データおよび SYNC フレームの送信ジッタの最大値は 0~6 ms となる。

#### 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

Vector 社 [18] 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、センサやスイッチ類の多い前方、エンジンルーム、ダッシュボード、両ドア、後方の 5 か所にデバイス HUB を配置することを想定し、各デバイス HUB に接続された I/O デバイスを 1 つの統合 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負荷を CAN に加える。

統合 ECU の処理モデルは、10 ms 間隔の周期処理とし、各周期の先頭で Request フレーム送信を起動、5 ms 後に共有メモリ上の入力値を読み出し出力値を生成し、出力データフレームおよび SYNC フレーム送信起動を行う。デバイス HUB は 10 ms 間隔で共有メモリ上の入力値を更新する。Request フレーム受信から入力値の送信要求、SYNC フレーム受信から出力デバイスへの出力値の伝達は遅延なく実行されるものとする。

###### (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 10 ms 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すべて True とする。

###### (4) I/O 情報伝送

パラメータとして設定する I/O 点数は、5 つのデバイス HUB に均等に割り振られるものとする。また、入力と出力の点数も均等とする。図 6 に従い 1 つのメッセージで伝送可能な 43 bit までは周期あたりそれぞれ 1 メッセージで I/O 情報を伝送し、これを超える場合にはメッセージを

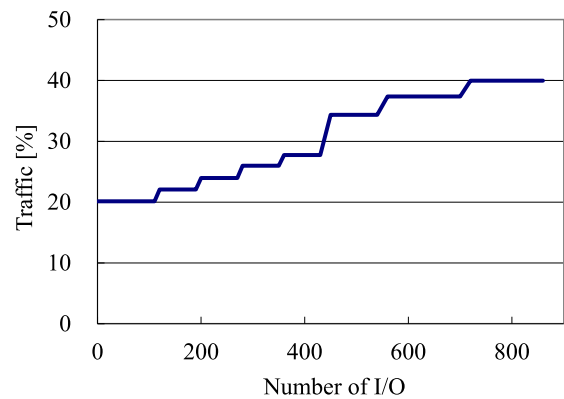


図 7 トラフィック特性

Fig. 7 Result of traffic evaluation.

追加する。たとえば、I/O 点数が 200 の場合、統合 ECU が送信する出力情報は 100 bit となり、周期あたり 3 つの CAN メッセージで送信し、各デバイス HUB の入力データはそれぞれ 20 bit で 43 bit に満たないため、各デバイス HUB は 1 つの CAN メッセージで I/O 情報を伝送する。

###### (5) 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どのメッセージを優先するかを決定する仕組みを持つ。CAN ID の割当て方針は 3.5 節の設定に従う。I/O 通信のジッタを評価するため、すべての ECU 間通信を方針 ① の高リアルタイム要求とし、ECU 間通信、統合 ECU 送信の I/O 通信、デバイス HUB 送信の I/O 通信の順に CAN ID を割り当てる。

##### 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 7 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とる理由は、統合 ECU が送信する出力情報が、CAN メッセージに収容できる 43 bit を超え、周期あたりの CAN メッセージが増えるためである。同じ理由で、430 点を過ぎたところで各デバイス HUB が送信する入力情報の CAN メッセージが増え、ネットワーク負荷が上昇する。

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量が增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 リアルタイム性能

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、入力値の変化を起点

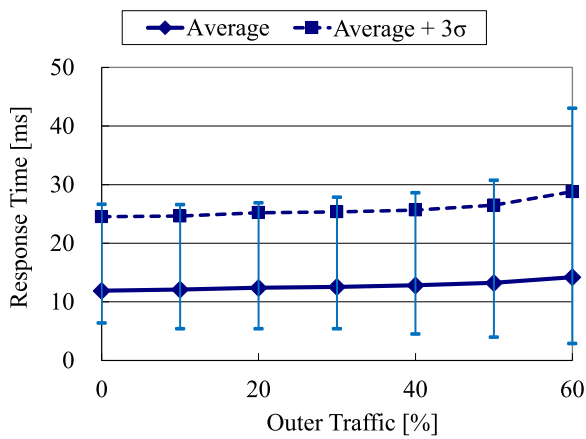


図 8 遅延特性

Fig. 8 Result of delay evaluation.

に、入力値がデバイス HUB 経由で統合 ECU に伝送され、統合 ECU から出力値としてデバイス HUB 経由で出力デバイスに伝達するまでの時間を指す。I/O 点数を 400 点に固定し、ECU 間通信によるネットワーク負荷をパラメータに、各負荷で 20,000 周期のシミュレーションを実行させたときの I/O 応答遅延を図 8 に示す。遅延値は平均と  $3\sigma$  に加え、縦線で最小値と最大値を示している。図 8 より、ECU 間通信負荷 50% までの I/O 応答遅延時間は平均、 $3\sigma$ 、変動幅とも負荷とともに線形に増加し、負荷 50% で 4.0~30.7ms の応答遅延時間となった。I/O 応答遅延が変動する主な原因は、統合 ECU およびデバイス HUB の周期処理によるものである。すなわち、デバイス HUB は周期的に入力値を取得するため、入力値の変化がデバイス HUB の共有メモリに反映されるまでに最大 1 周期の遅延が発生する。また統合 ECU とデバイス HUB の周期ずれにより、デバイス HUB の入力データが統合 ECU に伝達されるまでに最大 1 周期の遅延が発生する。以上の 2 つの遅延により、負荷によらず 0~20ms の入力情報の伝達遅延が発生する。最大値の負荷に対する増分は 10% ごとに 1ms となっており、3.5 節で示した送信ジッタの最大値と一致する。

一方、負荷 60% 時の I/O 応答遅延時間の最大値は 43.0ms となり、50% までと比べ 10ms 以上増大した。3.5 節で示した入力データ送信ジッタが 5ms 以上となり、統合 ECU の入力値読み出しタイミングに間に合わなくなることが原因である。その結果、入力値は次の処理周期で読み出され、出力値への反映が処理周期を示す 10ms だけ遅延する。出力の周期ずれは I/O 間の協調動作に影響が及ぶ。したがって、ECU 間通信と I/O 通信を統合する場合の ECU 間通信負荷は 50% が限界といえる。

## 5. 実装評価

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

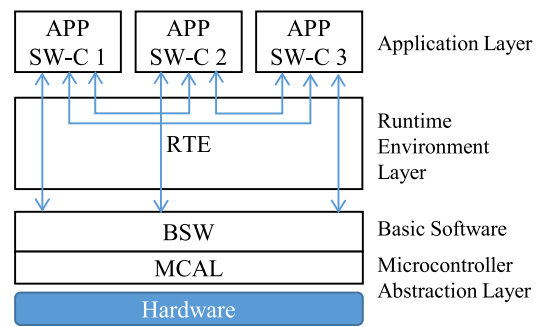


図 9 AUTOSAR の S/W 構成

Fig. 9 S/W structure of AUTOSAR.

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 [19]。また、これを統合 ECU 相当のマイコンに実装し、ネットワーク制御のリソースを評価する。

### 5.1 AUTOSAR 上への実装方式

図 9 に、AUTOSAR の基本構成を示す。アプリケーション層に実装される 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、ハードウェア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、② BSW (Basic S/W) が提供するサービス、③ CAN 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) 経由でハードウェアの制御に至る。

次に、階層統合ネットワークを適用した場合の実装構成を図 10 に示す。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、演算ライブラリやセキュリティ機能など、I/O デバイスの制御をともなわないものがある、これらは従来どおり 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

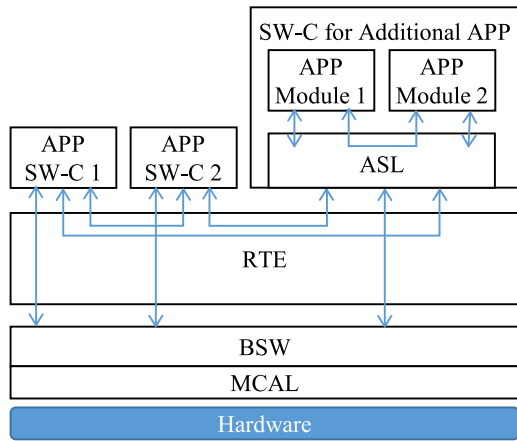


図 10 提案方式の実装

Fig. 10 Implementation of proposed method.

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、統合 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。

5.2 評価ボードへの実装

統合 ECU を想定した評価ボード上に S/W を実装し、実機による評価を行う。なお、コスト観点から統合 ECU の演算リソース、メモリリソースは限られるので、実装リソースについても評価する。

表 2 に、実装環境および実装する S/W の基本要件を示す。マイコンは車載 ECU 用マイコンとして実績のある Cypress 社 [18] 製 FR81S を使い、メモリリソースはマイコン内蔵のみを使う。S/W プラットフォームは Vector [19] 製 AUTOSAR Ver.3.0 に準拠した MICROSAR を使う。ASL 上には最大 8 個のアプリモジュールを追加できる構成とし、各モジュールは起動時に 1 回だけ実行されるシステム起動時間関数 (Runnable) と最少周期 10ms の周期実行関数からなる。

図 11 にタスク構成を示す。アプリモジュールの異常を監視するため、システムはアプリ実行時間を監視し、指定時間内に終了しない場合にフェールセーフ機能タスクを呼び出す機能を備える。本評価では障害ログ記録を実装しているが、将来的には機能縮退などの管理をここで行う。

ASL のリソース使用量を表 3 に示す。使用量の測定は、MICROSAR の統合開発環境で実行ファイルを作成した際に生成されるメモリマップより算出した。表 2 に示したすべての S/W に使用したプログラム Flash サイズは 288 KB、使用 RAM サイズは 39 KB であった。ASL を適用せず、同機能 S/W を RTE 上に実装した場合と比べ、Flash で 7.40 KB、RAM で 2.74 KB の増分があり、これが ASL に

表 2 実装 S/W 諸元

Table 2 Specification of S/W implementation.

項目	諸元	備考
1. 実装環境		
マイコン	FR81S	32bitRISC SoC
CPU 動作周波数	80MHz	
RAM 容量	128KB	
Flash メモリ容量	1MB/64KB	プログラム/データ
S/W プラットフォーム	MICROSAR	AUTOSAR 3.0 準拠
2. 追加 APP 用 S/W-C 仕様		
ASL 数	1	
追加アプリモジュール数	8	
Runnable 数	2/Module	初期化+周期処理
最少周期	10ms	整数倍で周期設定

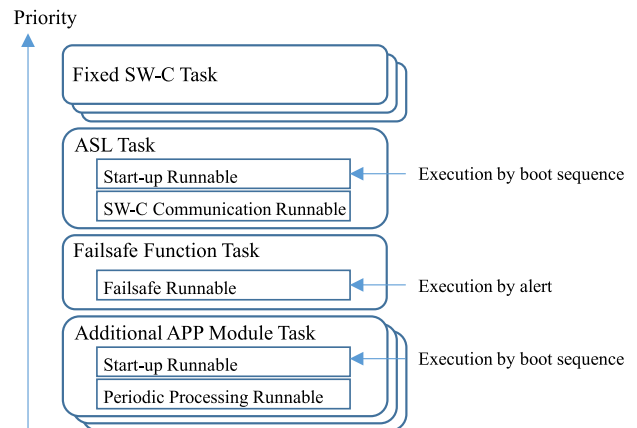


図 11 タスク構成

Fig. 11 Task structure.

表 3 ECU S/W のリソース使用量

Table 3 Resource usage of ECU S/W.

リソース	Size(Kbyte)		
	ASL 無	ASL 有	差分
実行 RAM	35.92	38.65	2.74
プログラム Flash	280.39	287.79	7.40

よるオーバーヘッドと見なされる。本差分には、実行コードのほかに、AUTOSAR プラットフォームの管理データ増分も含まれる。マイコンリソースに対する ASL の占有率は RAM で 2.1%、Flash で 0.7% であり、実用的には問題ないといえる。

次に、S/W 実行時間の計測結果を図 12 に示す。ASL を適用した場合の実行時間オーバーヘッドは大きく、その増分は主に追加モジュールと ASL の間の API 数に依存することが分かる。遅延の原因は、モジュール間通信に対する AUTOSAR の付加処理にある。すなわち、ASL 上のモジュールは、ASL が備えた通信用バッファを介して通信されるが、その際にソフトウェア保護機能によるバッファのアクセス権チェックが API の数だけ実行される。モジュールの周期起動ごとにこれが実行され、遅延増の原因となった。



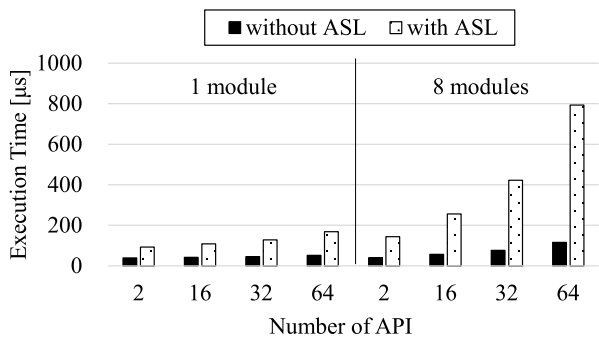


図 12 S/W 実行時間

Fig. 12 Execution time of S/W.

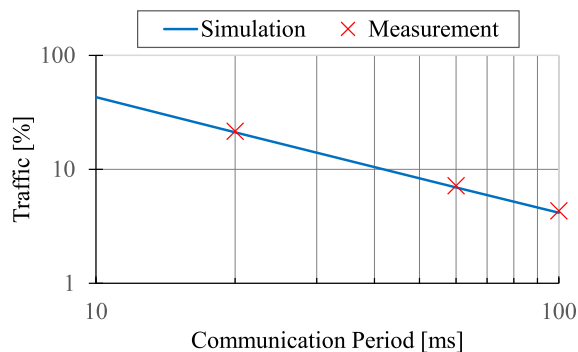


図 14 ネットワーク負荷の実測

Fig. 14 Measurement of network traffic.

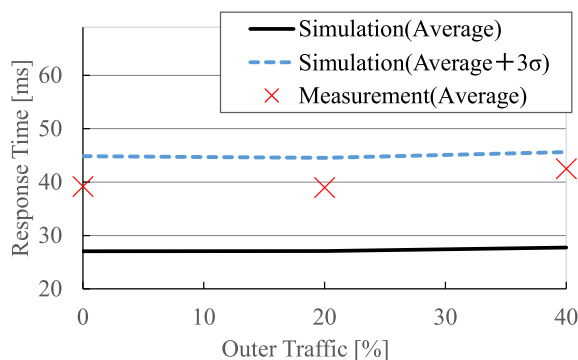
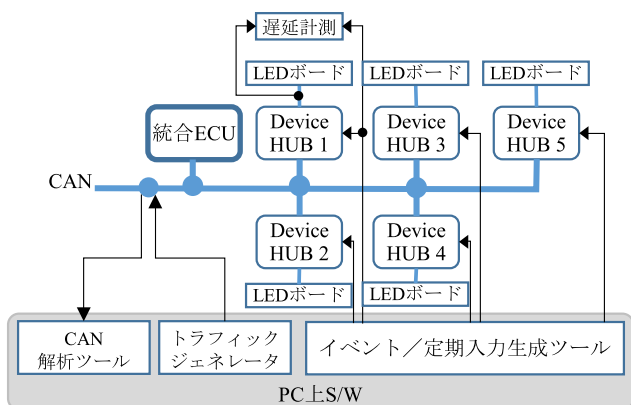


図 15 入出力遅延の実測

Fig. 15 Measurement of I/O delay.

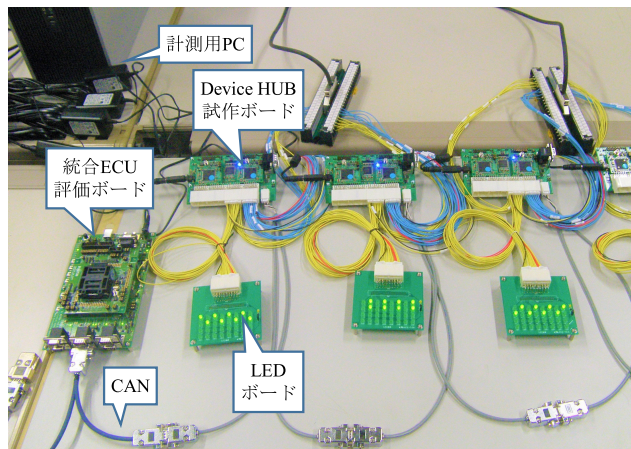


図 13 評価システム構成

Fig. 13 Evaluation system structure.

ASLがない場合、SW-C間の通信はAUTOSARプラットフォームのバッファ領域を介してアクセスされるだけなので、このチェック処理がない。ASLを適用する場合には、性能の要件に合わせ、追加モジュールの総API数に上限を加える必要がある。

### 5.3 システム評価

ASLを実装した統合ECU評価ボードを用いてシステム実機評価を行う。図13に評価システムの構成図を示す。シミュレーション評価と同様に、5つのデバイスHUBと1つの統合ECUで構成する。I/Oデバイスの代わりにPC

上で生成した信号を各デバイスHUBへデバイス入力信号として印加し、これをネットワーク経由で統合ECUへ伝達する。統合ECUで各I/Oデバイス出力値へ変換した後、デバイスHUBへ返送し、デバイスHUBに接続したLEDボード上のLED点灯パターンで動作を確認する。

本構成でI/O点数400点の制御を行ったときのネットワーク負荷の実測結果を図14に示す。ASL実装時の遅延増などが影響し、周期10msで制御することが困難であったため、周期20, 60, 100msで実測した。シミュレーションと実測の誤差は4%以下であり、実機上で期待どおりの結果を得ることができた。

次に、周期20msで制御実行中にトラフィックジェネレータで負荷を与えたときの制御遅延の実測結果を図15に示す。PCからデバイスHUB1への入力信号と、デバイスHUB1からLEDボードへの出力信号を観測し、入力変化が出力に反映されるまでの遅延を計測している。シミュレーションより平均値で12~15ms程度の遅延増が見られた。シミュレーションでは考慮していない統合ECUやデバイスHUB内のS/W処理遅延によるものと考えられる。なお、外部ネットワーク負荷を60%とした場合、動作が不安定となった。4.3節に示した入力値の周期ずれによる処理不整合の影響に加え、ネットワークの通信負荷が80%を超えパケット衝突が頻繁に発生したことが原因と推測される。

## 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。

各 ECU 間は CAN により省配線での情報共有が実現されている一方、ECU と I/O デバイスの間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課題に着目し、ECU と I/O デバイス間の通信を CAN に統合した、CAN と I/O 信号の変換を行うデバイス HUB を車上各所に配置し、I/O デバイスを近傍のデバイス HUB に接続することで I/O 配線の省配線化が進み、設置自由度が大幅に増す。

一方、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ため、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ときのデバイス I/O の応答遅延は、ECU 間通信による負荷が 50%以下の条件で 30ms であり、実用に耐えることを示した。

なお、ここでは CAN 2.0 を前提として評価したが、共有メモリを介してビットイメージを伝送する方式は、より高速なネットワークにおいても有効に機能する。すなわち、CAN 2.0 の 8 倍のデータ長を同等の時間で伝送できる CAN FD であれば、本評価と同等のネットワーク負荷、I/O 応答遅延のまま、I/O 点数を 8 倍に拡張できる。さらに、今後増大するイメージセンサなど、情報量多いセンサデータの伝達にも活用が期待できる。

さらに、AUTOSAR 上への実装に向け、RTE と SW-C の間に ASL と呼ぶ管理層を定義し、ここで I/O デバイス制御を仮想化する提案を行うとともに ECU 評価ボード上に実装し、メモリリソースが車載適用可能なレベルであることを示した。一方、ASL に起因した実行時間増大の課題も明らかとなり、性能改善を今後の課題とした。

今後、OTA が普及すると、各車両に搭載される I/O デバイスの変更に合わせて S/W を更新す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] 消費動向調査：平成 29 年 3 月実施調査結果，内閣府経済社会総合研究所景気統計部，p.11 (2017).

[2] Beardslee, S., Hodgson, J. and Bonte, D.: 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, p.52 (2011).

[4] AUTOSAR, available from (<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, 第 5 章, 日経 BP 社 (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] Nielsen, J.: Response Times: The 3 Important Limits (1993), available from (<http://www.nngroup.com/articles/response-times-3-important-limits/>)

[8] LEAF (ZAA-E0), T00SM3N1J, 日産自動車株式会社 (2010).

[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 (2010).

[10] Accord (DBA-CU2), 60TL160, サービスマニュアル, 本田技研工業株式会社 (2008).

[11] Robert Bosch GmbH: CAN Specification, Version 2.0 (1991).

[12] ISO: Road vehicles – Controller area network (CAN) – Part 5: High-speed medium access unit with low-power mode, International Organization for Standardization (2007).

[13] 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, No.29, pp.65-88 (2010).

[14] 倉地 亮, 高田広章, 西村政信, 堀端啓史:最大遅れ時間解析によるスケーラブル CAN プロトコルの性能評価, 情報処理学会論文誌, Vol.53, No.8, pp.1962-1975 (2012).

[15] BOSCH: CAN with flexible data-rate - Specification 1.0 (2012).

[16] 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).

[17] 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).

[18] Vector, available from (<http://vector.com>)

[19] 川上 武, 片山吉章, 伊藤益夫, 徳永雄一: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, 情報処理学会第 75 回全国大会講演論文集, Vol.2013, No.1, pp.33-34 (2013).

[20] Cypress, available from (<http://www.cypress.com>)

## 推薦文

本稿は自動車の機能進化に対応する車載ネットワークの実現に向けて、CAN ネットワークを用いた階層統合型車載ネットワークを提案している。自動車の機能進化に向けては小型コンピュータである ECU とセンサ、アクチュエータなどのデバイスとを追加する必要が生じる。これらを容易に行うために CAN 上で ECU・デバイス間の配線をネットワーク化する手法を示している。高度交通システムに関する研究者にとって有用な手法であると思います。

(高度交通システムとスマートコミュニティ研究会主査 清原良三)



徳永 雄一 (正会員)

三菱電機(株)情報技術総合研究所.  
産業用計算機のフォールトトレラント技術, 組み込み機器の省電力設計技術, 無線アドホックネットワーク技術, ネットワーク時刻同期技術, 動態監視技術の研究に従事. 工学博士. 情報処理学会高度交通システム研究会, システム制御情報学会, 電子情報通信学会各会員.



西山 博仁

三菱電機(株)情報技術総合研究所.  
産業用計算機システム技術, 無線アドホックネットワーク技術, 制御ネットワーク技術の研究に従事. 自動車技術会会員.



伊藤 益夫 (正会員)

三菱電機(株)情報技術総合研究所.  
車載情報機器・車載制御装置の開発, 車載制御システム開発技術の研究に従事. 自動車技術会会員.



千田 修一郎

三菱電機(株)情報技術総合研究所.  
自動車向け制御機器のセキュリティ分析および機能開発, 無線アドホックネットワーク技術の研究に従事.