

## 車載 ECU ソフト更新におけるデータ転送方式 Transmission of Delta Data for Automotive ECU

小沼 寛† 福田 勇輔‡ 寺島 美昭\* 清原 良三‡  
Yutaka Onuma Yusuke Fukuda Yoshiaki Terashima Ryozo Kiyohara

### 1. はじめに

自動車の電子制御システムは、複数の ECU(Electronic Control Unit)と呼ばれる電子制御装置が、エンジンやブレーキ、トランスミッションなどの各システムを制御し、互いに通信しあうことで成り立っている。Advanced Driving Assistant System(ADAS) や Advanced Emergency Braking System(AEBS)、自動運転機能などの搭載により、自動車に搭載される ECU の数は増加、ソフトウェアの規模は増大している。その結果、不具合を修正しきれず出荷後に不具合が発覚する場合がある。また、自動車システムの高度化・電子化が進む一方、セキュリティに対する懸念が高まっている。ネットワークにつながることによる自動車のセキュリティに問題があった実例として、リコールを行っている[1]。この様に不具合や問題が発覚した際、早急に修正・更新を行うことが重要となる。

一般に、ECU ソフトウェアの更新は、自動車の所有者がディーラーに自動車を持ち込むことで行っている。更新中はディーラーに自動車を預けることになるため、所有者は更新中に自動車を使うことができない。また、Over-The-Air(OTA)を使用した更新技術が開発されている[2]。OTA を使用した更新の場合、走行中に更新を行うことは危険なため、自動車を停止しておく必要がある。OTA による更新においても、更新中に所有者は自動車を使うことはできない。よって、更新時間を短くすることが求められる。

ECU ソフトウェアの更新には低速な車載ネットワーク[3]を介して更新データを送る必要があるため、更新データの車載ネットワーク上の転送時間が課題となる。更新時間短縮の一手法として更新データサイズを小さくし、ダウンロード時間を削減する研究がされている[4][5][6]。これらの研究では、データサイズの削減方法として差分更新による圧縮を用いている。差分更新はソフトウェアのバージョン間の差分の抽出を行い、更新データ全体ではなく差分情報のみを送信し、旧バージョンのプログラムに差分を適用することで更新する。データの完全な状態ではなく差分のみを扱うため、大幅なデータ量の削減が期待できる。

そこで、本論文では、更新時間の短縮を目的とし、差分更新を用いた車載 ECU のソフトウェア更新における効率的な転送方式を提案する。

### 2. ECU ソフトウェアの更新時間

本論文では、更新データは差分更新により圧縮されると前提する。ECU ソフトウェアの更新時間は、フラッシュメモリの書き換え時間と更新データの転送時間に分けられる。それぞれの時間の要素について以下に述べる。

#### 2.1 フラッシュメモリの書き換え時間

ECU の多くは NOR 型フラッシュメモリを採用している。NOR 型フラッシュメモリの書き込み手順は、一度イレースブロック単位で消去した後、1 バイト単位で書き込みを行う。NOR 型フラッシュメモリは読み出し速度が 100ns 程度と高速であるが、消去は単位サイズにより 1 秒から数十秒、書き込みに 10  $\mu$ s 程度と、読み出しに比べ書き換えには時間を要し、更新時間は消去の時間に依存する[7]。

フラッシュメモリの書き換え時間を次式に示す。

$$T_r = D + E_t \times N \quad (1)$$

ここで  $T_r$  はフラッシュメモリ書き換え時間、 $D$  は圧縮された更新データの復号化時間、 $E_t$  はイレースブロックの書き換え時間、 $N$  は書き換えが必要なブロック数とする。自動車に搭載されている ECU はメーカーや性能の違いにより多種混在する。それぞれの変数は、更新対象となる ECU の性能や更新データのサイズに依存する。

#### 2.2 車載ネットワーク上の転送時間

各 ECU は車載ネットワークを利用して通信しており、更新データも車載ネットワークを介して各 ECU に送信する。車載ネットワークの業界標準とされている Controller Area Network(CAN)[3]はバス型トポロジで構成され、最大ビットレートは 1Mbps、一度に送信可能なデータ長は最大 8 バイトとなっている。ただし、ビットレートは CAN のバス長に依存し、実際の運用では最大で 500Kbps 程度となっており、携帯電話などのブロードバンド回線に比べ低速・低帯域のため、大量のデータの転送には時間を要する。

CAN ではフレーム単位で通信を行う。フレームによるデータの送信例を図 1 に示す。バス上へのデータ送信には図 2 に示した構造を持つデータフレームが使用される。データフレームは最大 8 バイトのデータの送信に使用できるフィールドを持つ。データの要求には、データフレームからデータフィールドを除いたリモートフレームが使用される。データを必要とするノードは、データを要求するためにリモートフレームを送信し、その ID に該当するノード

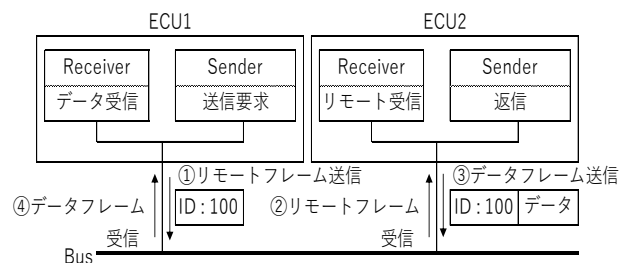


図 1 フレームの通信

フィールド名	SOF	ID	RTR	コントロール	データ	CRC	ACK	EOF
ビット数	1	11	1	6	0-64	16	2	7

図 2 データフレーム

† 神奈川工科大学大学院, Graduate School of Kanagawa Institute of Technology

‡ 神奈川工科大学, Kanagawa Institute of Technology

\* 創価大学, Soka University

がデータフレームを返すことでデータをやりとりする。同時に複数のノードからメッセージが送信され衝突が起きた場合、メッセージに付与された ID を用いた優先度による調停が行われる。優先度の高いメッセージがバスを使用している間、優先度の低いメッセージは送信できないため遅延が発生する。また、各ノードは ID を使用し、受信したデータがどのようなデータであるか識別する。

以上のことを踏まえ、転送時間を次式で示す。

$$T_d = \frac{(F_r + F_d) \times \left(\frac{U_s}{W}\right)}{B} \quad (2)$$

ここで  $T_d$  は転送時間、 $F_r$  と  $F_d$  はそれぞれリモートフレームとデータフレームの送信時間、 $U_s$  は更新データサイズ、 $W$  は使用可能なデータフィールドの長さ、 $B$  はビットレートとする。データフィールド長は  $1 \leq W \leq 8$  の値をとる。一度のデータフレームの送信により最大 8 バイトのデータを送ることができるが、データフィールドの一部はエラー訂正やセキュリティのため使用されるべきである。従って、 $W$  が小さくなり送信すべきフレーム数は増加する。

### 3. 提案手法

#### 3.1 更新データの転送

差分更新を用いた ECU ソフトウェアの更新の流れを図 3 に示す。差分更新を用いる場合、更新を適用する際に更新対象となる ECU 内で復号化を行う必要がある。復号化を行うためには更新データを全て受信する必要がある。

また、CAN では複数の ECU に対して同時に異なるデータを送信することができない。そのため、複数の ECU に対して更新を行う場合、データの受信を行っていない ECU には待機時間が発生する。

そこで、更新データを分割し、各 ECU の更新を分散して行う。図 4 のように、ある ECU のフラッシュメモリ書き換え中に他方の ECU に更新データを送信する。復号化時間  $D$  を分散させることで、更新時間に含まれる ECU のフラッシュメモリ書き換え時間  $T_r$  を軽減できる。

#### 3.2 更新データ分割

更新データの分割方法について、フラッシュメモリの書き換え単位であるイレースブロックのサイズ単位で分割することが望ましい。また、送信する更新データのサイズを  $W$  単位にし、余分なフレームを送信しないことで、分割による  $T_d$  の増加を抑える。対象とする ECU ごとにイレースブロックや更新データサイズが異なるため、それぞれの ECU について分割数やサイズを計算する。

次に、各 ECU への更新データの送信順序を決定する必要がある。送信順序は各 ECU の更新データの分割数、サイズ、書き換え時間、優先度を考慮して決定する。更新するタイミングにより対象となる ECU や更新データは変わるため、送信順序は対象 ECU に合わせ、更新を行うごとに決定する。

#### 3.3 ECU による影響

分割更新を行うためには ECU が更新方法に対応している必要がある。しかし、自動車内には多種の ECU が存在するため、更新方法に対応した ECU と未対応の ECU が混在することが予想される。また、対象 ECU の更新中に他

#### ①更新データのダウンロード

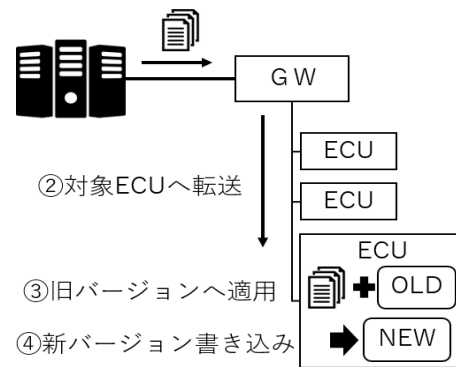


図 3 差分更新

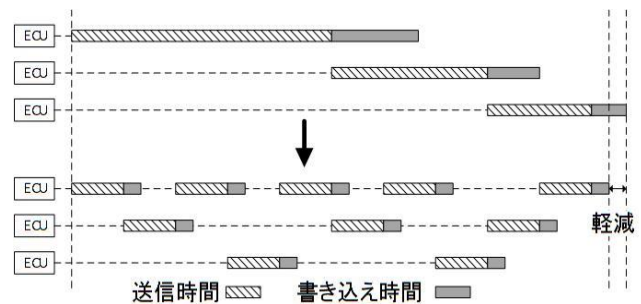


図 4 分割による更新時間軽減

の優先度の高い ECU がメッセージを送信している場合、更新データのダウンロードがその間停止してしまう。更新対象にそれぞれの ECU が混在するケースについての送信順序の決定方法や、更新中の他の ECU のメッセージによる影響についてシミュレーションを行う。

### 4. まとめ

本論文では、差分更新を前提とした ECU ソフトウェア更新の効率的なデータ転送方法について提案をした。提案方式では、更新データを分割して送信・書き換えを行うことにより、更新時間を短縮できる可能性を示した。更新データの分割と送信順序の決定には、更新対象となる ECU に依存するため、様々な値に対し考慮する必要がある。今後、更新時間に関する ECU の値をパラメータ化し、想定されるケースにおいてシミュレーションをし、評価を行う。

#### 参考文献

- [1] “Chrysler、車の遠隔操作問題で 140 万台のリコール発表”, <http://www.itmedia.co.jp/enterprise/articles/1507/27/news038.html>
- [2] 日立, <http://www.hitachi.co.jp/New/cnews/month/2016/04/0428.html>
- [3] Bosch CAN 2.0 Specification, <http://esd.cs.ucr.edu/webres/can20.pdf>
- [4] Yutaka Onuma, Masanao Nozawa, Yoshiaki Terashima, Ryozo Kiyohara, “Improved Software Updating for Automotive ECUs - Code Compression -,” The 4th IEEE International Workshop on Consumer Devices and Systems conjunction with COMPSAC 2016
- [5] 野澤 優尚, 小沼 寛, 清原 良三, “車載 ECU 向けソフト更新のためのデータ圧縮方式”, 研究報告マルチメディア通信と分散処理 (DPS), Vol.2015-DPS-165, No.4 (2015)
- [6] 寺岡 秀敏, 中原 章晴, 黒澤 憲一, “車載 ECU 向け差分更新方式”, 研究報告コンシューマ・デバイス&システム (CDS), Vol. 2016-CDS-16, No.5 (2016)
- [7] Spansion Japan, フラッシュメモリガイドブック (2008)