

車載装置向けソフトウェア更新に関するオブジェクトリンク方式

伊藤益夫[†]

ユーザニーズの多様化に応えるために、車載向けデバイスに対する多様性が求められている。デバイスの多様化に伴い、デバイスを制御する車載電子制御機器も多様化に対応する必要がある。本稿は、デバイスのハードウェア資源管理が必要な制御用ソフトウェアを対象にしたソフトウェアの更新方法を提案する。

Object Link Scheme for Update of Automotive Device Software

Masuo Ito[†]

Automotive devices need diversity to satisfy diversity of user needs. Depending on diversity of the devices, ECUs (Electronic Control Units) need to correspond to it. This paper proposes a scheme to update of control software necessary to manage the hardware resources of the devices.

1. はじめに

自動車に対してユーザニーズの多様化が高まっている。エンジンから照明に至るまで、自動車に搭載されるデバイスは電子化が進んでおり、デバイスの多様化に向けデバイスを制御するソフトウェアへの期待が高まっている。本稿では、運用中においても、ユーザの多様化にあわせたデバイスの制御特性の改善やソフトウェア機能追加が可能な電子制御機器（ECU: Electronic Control Unit）のソフトウェア更新方法を提案する。

2. 研究背景

2.1 多様性への対応

現在の自動車はユーザニーズに対し、車種やグレード、メーカーオプション、ディーラオプションといった定められた範囲内での選択肢を提供することで対応を図っている。ユーザニーズの更なる多様化にきめ細かく対応するには、車両リリース後においても、デバイスの制御特性の改善やソフトウェア機能追加が可能であることが望ましい。また、現在、車両にはサプライヤから提供されたデバイスが搭載されているが、この先の更なる多様性を考えればサードパーティによるデバイスの搭載も考えられる。近年、車両内の機能統合が図られつつあり一つの ECU で複数のデバイスの制御が行われるようになってきた。現在サプライヤが提供する ECU には、サプライヤもしくはカーメカの制御用ソフトウェアが実装されているが、将来サードパーティによるデバイスの提供が始まり、更に一つの ECU への機能統合が進んだ場合、サプライヤが提供する ECU にサプライヤとサードパーティの制御用ソフトウェアが混在して実装されることも起こりえる。

2.2 車載電子制御機器への期待

車両に搭載されるデバイスは、予めデバイスに対応した制御用ソフトウェアが実装された ECU によって制御される。

先でも述べたように、近年車両内の機能統合が図られつつあり一つの ECU で複数のデバイスの制御が行われるようになってきた。今後、多様性の高まりに伴いデバイスの種類が増加すれば、ECU で制御するデバイスの数も増加していくと考えられる。ECU に実装されるソフトウェアのボリュームは情報機器に近づいてくるものの、コストやリアルタイム性といった非機能要件はこれまで通り車載用として対応していか

[†] 三菱電機株式会社 情報技術総合研究所
MITSUBISHI ELECTRIC CORPORATION INFORMATION TECHNOLOGY R&D CENTER

ければならない。

2.3 車載向けソフトウェアの標準規格

近年、自動車業界ではソフトウェア開発コストの低減を目的に AUTOSAR が規格化され、この AUTOSAR の標準プラットフォーム上での開発が進められようとしている。AUTOSAR では、アプリケーション層、RTE (Runtime Environment) 層、BSW (Basic Software) 層に階層化され、アプリケーション層に実装されるソフトウェアはソフトウェアコンポーネント (SW-C) の単位で管理される。SW-C は、その下層に位置する仮想的なファンクションバスである RTE 層により、機能を実現する上では実装される ECU に依存しない。これは、設計段階における SW-C の可搬性が高いことを示している。

2.4 ソフトウェア更新の必要性

多様性の向上への対応として、各々のデバイスに対してあらゆる制御特性や機能を制御用ソフトウェアとして予め全て ECU に実装しておくことが考えられる。ECU に実装される範囲内でユーザが制御特性もしくは機能を選択してデバイスが動作するため、細かなニーズに対応することが可能である。しかし、ECU に実装されるソフトウェアのボリュームの増加に伴い ECU のコスト増加につながる。また、ECU に実装するソフトウェアのボリュームを抑えるために制御特性や機能の選択肢の組み合わせで ECU を用意することも考えられる。しかし、これは ECU の種類が増え管理の問題が発生する。更に、両者においても車両のリリース後の制御特性や機能の変更や追加に対して対応が図れない。

よって、必要なときに必要だけ制御用ソフトウェアを更新することが可能なソフトウェア更新方法が必要になる。車載ソフトウェアの実装においては、近年 AUTOSAR の標準規格の導入が進められており、今後アプリケーションレベルでは可搬性の高い SW-C 構造を用いた開発が主流になると考えられる。標準規格で設計された SW-C は、これまで標準規格なしに設計されてきたアプリケーションソフトウェアと比較すると、車両リリース後の運用場面で求められる制御用ソフトウェアの可搬性に対しても適用し易いと考えられる。多様性によるソフトウェア更新のニーズの高まりに加え、ソフトウェアの更新に対して親和性の高い車載ソフトウェア標準規格の導入が進むと見込み、制御用ソフトウェアを対象にしたソフトウェア更新について検討することにした。

2.5 課題

情報機器の分野においては、制御用ソフトウェアの更新がプラグアンドプレイで実現している。プラグアンドプレイは、制御装置とデバイスに通信機能が備えられ、制御装置とデバイス間で情報がやりとりされる。また、制御装置のドライバは、ライブ

ラリとしてファイルシステム上で管理され DLL(Dynamic Link Library)によって動作する。DLL の動作環境として MMU(Memory Management Unit)が用いられる。このようなプラグアンドプレイの環境を ECU に対しても適用することで、制御用ソフトウェアの更新は可能である。

しかし、制御装置側にはファイルシステムや DLL を動かすハイスpekクなマイコンを必要とする。また、デバイス側にも通信機能が必須となる。これらは ECU とデバイスのコスト要件に見合いにくい。また、MMU はリアルタイム性が確保できないことから、MMU を用いると ECU のリアルタイム性が確保されなくなってしまう。ECU における制御用ソフトウェアを対象としたソフトウェア更新においては、コストとリアルタイム性が課題となる。

2.6 要件の整理

ソフトウェア更新を検討する上での要件を整理する。まず、車載制御装置としてのコストとリアルタイム性は、従来の ECU と同等とする。

搭載されるデバイスは、マイコンを持たないデバイスもあるため、マイコンが必要な機能は用いない。また、インターフェイスが I/O のみのデバイスに対応させるため、デバイスに通信機能は持たせない。

3. 関連研究

組み込み機器を対象としたソフトウェア更新に関しては、不具合やバージョンアップを目的としたソフトウェア更新の方法が提案されている。ここで、これまで提案されているソフトウェア更新に関して大きく二つに分類する。一つは更新するソフトウェアの作成および実行方法、もう一つは作成されたソフトウェアをより少ないデータ数で書き換える方法である。この場合、前者には文献1があり、後者には文献2がある。文献1は、携帯電話機をはじめとする機器を対象としたもので、 μ ITRON仕様OSにダウンロード機能付加し、機器ユーザがネットワーク機能を使って部分的にモジュールをダウンロードし、ソフトウェアをバージョンアップするための仕組みが提案されている。文献2は、車載機器向けの放送を使った差分更新方式で、放送局が新旧ファームウェアの差分情報を放送し、車載機器は受信した差分情報に基づいて自身の持つファームウェアを新しいものに更新する手法である。

本稿は上記の分類では前者に該当するもので、これまでの研究と異なる点は、制御用ソフトウェアを対象としデバイスのハードウェア資源管理を考慮したソフトウェア更新を行う点である。

4. ソフトウェア更新方法

4.1 オブジェクトの管理

デバイスを制御するための制御用ソフトウェアをデバイス自体に格納すればデバイスを通して制御用ソフトウェアを ECU に送ることは可能である。しかし、デバイスに通信機能が必要となるためデバイスの要件を満たさない。従って、ECU はデバイス以外から該当する制御用ソフトウェアを入手する必要がある。そこで、制御用ソフトウェアはメーカーのラインやディーラと接続可能なセンターに格納しておくこととした。ラインやディーラでセンターにつながる外部端末から更新するソフトウェアを ECU へロードすることにする。

4.2 リンカ機能

再配置可能なオブジェクトは、静的もしくは動的なリンク（仮想アドレスの物理アドレスへの変換）で用いられる。静的なリンクは、再配置可能なオブジェクトが生成された後、実行可能ファイル生成時にリンカによって仮想アドレスが物理アドレスに変換される。物理アドレスが決定された実行可能ファイルが ROM に書き込まれ、ROM 上に書き込まれた物理アドレスで動作する。一方、動的なリンクは、仮想アドレスのままの再配置可能なオブジェクトが ROM に書き込まれ、実行時に呼び出された RAM 上で仮想アドレスから物理アドレスに変換される。

動的なリンクでの実行では、オブジェクトの追加時に既に ROM に書き込まれたデータに対して書き換えの影響は及ぼさないものの、実行時にアドレス変換処理を行っているため実行速度に影響を及ぼす。アドレス変換にかかる処理をカバーするためには、従来に比べてハイスペックなマイコンが必要となる。これは ECU のコスト増加につながり ECU の要件に見合わない。

静的なリンクでの実行では、リンカ機能を ECU の外に持たせるか、ECU の中に持たせるかの二通りが考えられる。リンカ機能を ECU の中に持たせた場合、外部から該当するオブジェクトを受け取ることができれば、あとは ECU 内で実行可能ファイルの生成および ROM への書き込みまでが完結する。ECU と外部を含めた全体構成はシンプルであるが、リンカ機能が ECU に実装される分 ECU のコストが上昇してしまい、ECU の要件を満たさない。制御用ソフトウェアの更新といった稀なイベントに対するコスト上昇は避けたい。リンカ機能を ECU の外に持たせ場合、ECU の外で ECU のメモリマップ情報を持ち合わせる仕組みが必要となる。クローズドな環境ではセンターで管理できるが、オープンな環境においては、ECU から外部へ渡す必要がある。ECU の外で作成された実行可能ファイルは、従来 ECU に書き込まれていた実行可能ファイルと同様、ROM 上に書き込まれた物理アドレスで動作するため実行時の影響はない。ECU のコスト上昇にもつながる要素は少ないことから、リンカ機能は ECU の外に持

たせることにする。

4.3 コンフィグレーション情報の管理

対象としているオブジェクトは制御用ソフトウェアであるため、制御するデバイスが必要とするハードウェア資源について管理する必要がある。デバイスは、用途や機能によって入出力する情報が異なるもので、マイコンの I/O、DA、AD といった資源に対して使用する数や使い方が異なるためである。また、デバイスを制御する制御用ソフトウェアにおいても、用途や機能によってマイコンの割り込みやタイマなどといった資源に対して使用する数や使い方が異なるため、ハードウェア資源の管理が必要である。

これまでの ECU 開発は、開発時において対象とするデバイスが特定されているため、その時点でデバイスに対応した制御ソフトウェアも特定される。よって、ECU のハードウェア資源に対する割り当てや設定も ECU 開発時に行い、かつその時点で完結していた。しかし、運用中に制御特性の変更や機能追加に伴いデバイスが管理するハードウェア資源を変更する場合、ECU のハードウェア資源に対する割り当てや設定もその都度実行しなければならない。また、制御用ソフトウェアは様々な機種種の ECU に実装され、更に同じ機種種の ECU においてもユーザのデバイスの選択の仕方によってソフトウェアの実装状態は異なり、これらの点にも対応しなければならない。また、サプライヤもしくはカーメーカーのクローズドな環境で制御用ソフトウェアが開発されている場合は、開発現場における設計仕様によって全てのコンフィグレーション情報が揃い ECU 全体のコンフィグレーション情報を生成することが可能であるが、サードパーティの制御用ソフトウェアも実装可能なオープンな環境では、設計仕様から ECU 全体のコンフィグレーション情報を生成することはできなくなる。

そこで、標準仕様とする、制御用ソフトウェア向けのコンフィグレーション情報と、ECU 向けのコンフィグレーション情報の 2 種類を用意して対応することとする。

制御用ソフトウェア向けのコンフィグレーション情報は、制御用ソフトウェアが動作するのに必要なハードウェア資源の情報であり、制御用ソフトウェアの開発の時点で生成し制御用ソフトウェアと共にセンターで管理する。制御用ソフトウェア向けコンフィグレーション情報は、ソフトウェア更新時において、対象とする ECU に対し制御用ソフトウェアが必要とするハードウェア資源があるかどうかの実装の可否の判定および ECU 全体のコンフィグレーション情報生成に用いる。

ECU 向けのコンフィグレーション情報は、ECU に実装される全ての制御用ソフトウェアのコンフィグレーション情報がまとめられたもので、ECU を動作させるための情報となる。このコンフィグレーション情報は ECU の動作に関わる情報で ECU に実装される。また、ECU 向けのコンフィグレーション情報は、次のソフトウェア更新時に呼び出され、更新可能かどうかの可否判断と ECU 全体のコンフィグレーション情報の

源となる。なお、クローズドな環境で ECU が管理される場合は、ECU 全体のコンフィグレーション情報を ECU から呼び出さず、サーバ内に格納するコンフィグレーション情報を更新可能かどうかの可否判断と ECU 全体のコンフィグレーション情報の生成に用いる。

4.4 ECUの管理

既存の ECU に対し一部の制御用ソフトウェアを更新して実行可能ファイルを生じようとする場合、ECU のハードウェアスペック情報が必要となる。また更新しようとする制御用ソフトウェアが既存の制御用ソフトウェアと相互機能しあう場合は既存の制御用ソフトウェアが必要となる。そこで、ECU には ID を割り当て、ID 毎に各 ECU のスペック情報や制御用ソフトウェアの構成を、制御用ソフトウェアと同様にセンターに格納する。

4.5 デバイスと制御用ソフトウェアの紐付け

コスト要件により、制御用ソフトウェアはデバイスに実装しない。制御用ソフトウェアはセンターに格納するため、デバイスとそのデバイスの制御用ソフトウェアを紐付ける必要がある。よって、デバイスに ID を割り振り、ID で制御用ソフトウェアを紐付ける。

4.6 デバイスとデバイスの装着状態

多様性の観点からデバイスの装着には汎用性が求められるため、数に限りがある装着可能なポートを特定のデバイス専用として確保することはできない。

ECU は、ECU が持つ装着可能なポートに対して、どのデバイスがどのポートに装着されているかを把握しなければならない。デバイス装着ポートと装着されているデバイスの関係は、ECU にてデバイス装着ポート情報として管理する。

5. ソフトウェアの生成と更新手順

5.1 全体構成

本方式でのソフトウェア更新を行う際の全体構成を図 5-1 に示す。車両と、車両と接続可能な外部端末、外部端末と接続可能な ECU 管理サーバ、S/W 管理サーバで構成する（ここでの ECU 管理サーバと S/W 管理サーバは、説明のため便宜上論理的にわけたものであって、構築の際には物理的には一つのサーバでも構わない）。

S/W 管理サーバは、更新するデバイスの制御用ソフトウェアと、デバイスおよび制御用ソフトウェアが必要とする H/W 資源を管理したコンフィグレーション情報をデ

バイスの ID ごとに管理する。

ECU 管理サーバは、ID で管理された各 ECU の H/W スペック、ソフトウェアの実装状態を管理する。

外部端末は、ソフトウェア更新を行う際、車両とのデータ転送手段として用いる。外部端末は、車両に標準装備される OBD II ポートを介して ECU と通信を行う。また、外部端末は、ECU のコンフィグレーション情報の生成、実行可能ファイルの生成処理を行う。

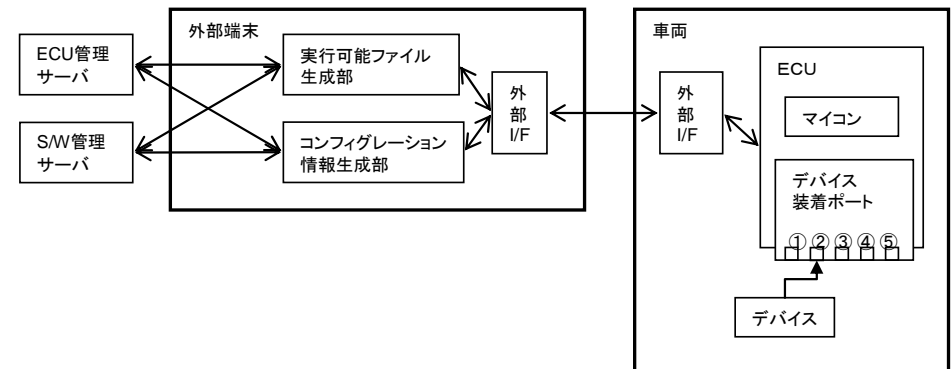


図 5-1 ソフトウェア更新を行うための全体構成

5.2 全体の動作フロー

ソフトウェア更新の全体の動作フローを図 5-2に示す。車両に外部端末を接続すると、車両内のECUと外部端末間で両者間による初期接続を行ったあと、ECUからECU ID、コンフィグレーション情報、メモリマップ、搭載デバイスIDリスト、ECUのデバイス装着ポート情報を送信する。外部端末では、ECU管理サーバより、ECU IDに該当するECUのスペック、ECUに実装されるオブジェクトファイルをダウンロードする。外部端末ではECUから取得し搭載デバイスIDリストより、該当するIDから更新対象となる制御用ソフトウェアのオブジェクトファイルおよびデバイスIDのコンフィグレーション情報をS/W管理サーバよりダウンロードする。

外部端末では、ECU から受信したメモリマップ、ECU 管理サーバから受信した ECU に実装されるオブジェクトファイルおよび ECU のスペック、S/W 管理サーバから受信した更新対象のデバイスのオブジェクトファイルから、実行可能ファイルの生成が可能かの判定後、実行可能ファイルを生成する。

また、外部端末では、ECU から受信した ECU スペックおよびコンフィグレーション情報、S/W 管理サーバから受信したデバイスのコンフィグレーション情報から、更新可能かの判定後、新規に ECU 全体のコンフィグレーション情報を生成する。

外部端末側では実行可能ファイルとコンフィグレーション情報の生成後、ECU にそれぞれを送信する。ECU では受信した実行可能ファイルとコンフィグレーション情報を ROM に書き込む。

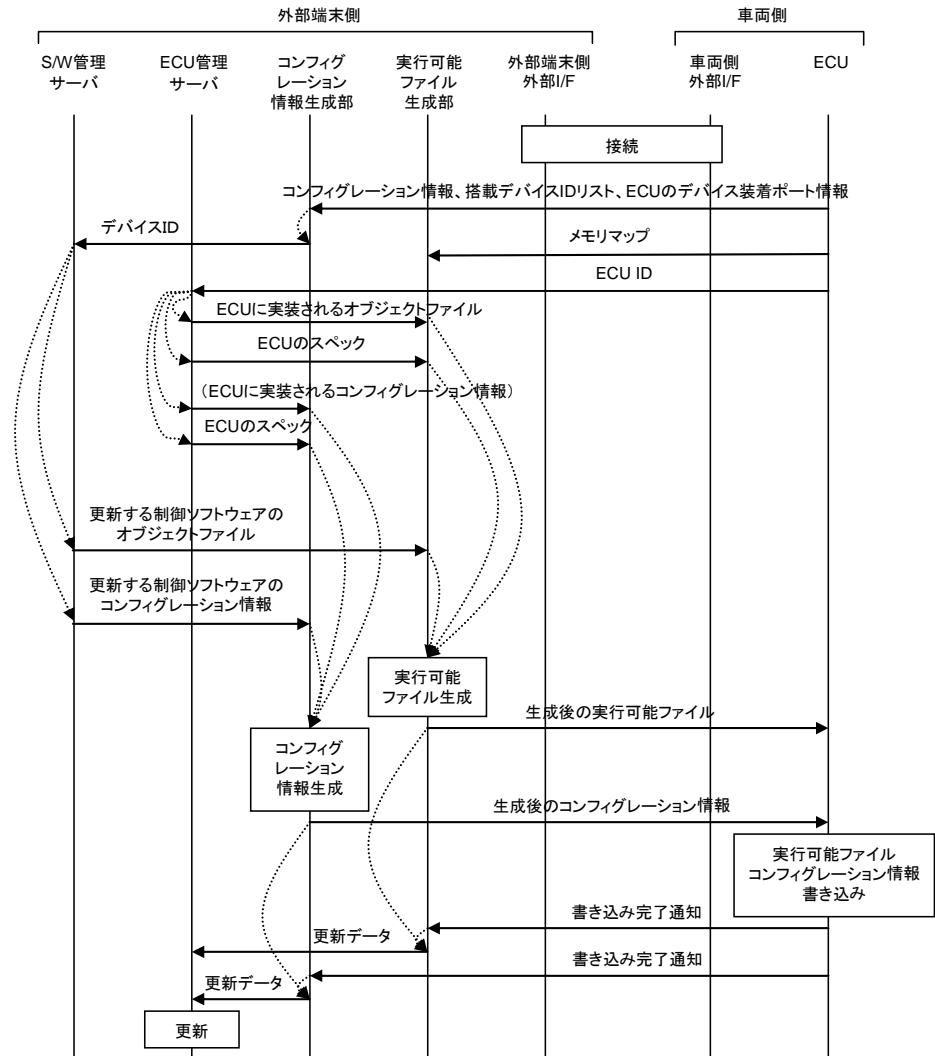


図 5-2 ソフトウェア更新を行うための全体フロー

5.3 ECUのコンフィグレーション情報の生成手順

外部端末でのECUのコンフィグレーション情報の生成手順を図 5-3に示す。外部端末では、EUC管理サーバから受信したECUのスペックを読み込み、ECUのハードウェア資源の情報を揃える。次に、ECUに実装されているコンフィグレーション情報とデバイス装着ポート情報を読み込み、その時のECUのハードウェア資源の使用状況を揃え、更新しようとするデバイスのコンフィグレーション情報を読み出し、更新の対象とするデバイスが必要とするハードウェア資源を揃える。これらの情報により、ECUに対して更新しようとするデバイスが必要とする資源を割り当てることが可能かどうかを判断し、制御用ソフトウェアの更新が可能かどうかを判定する。

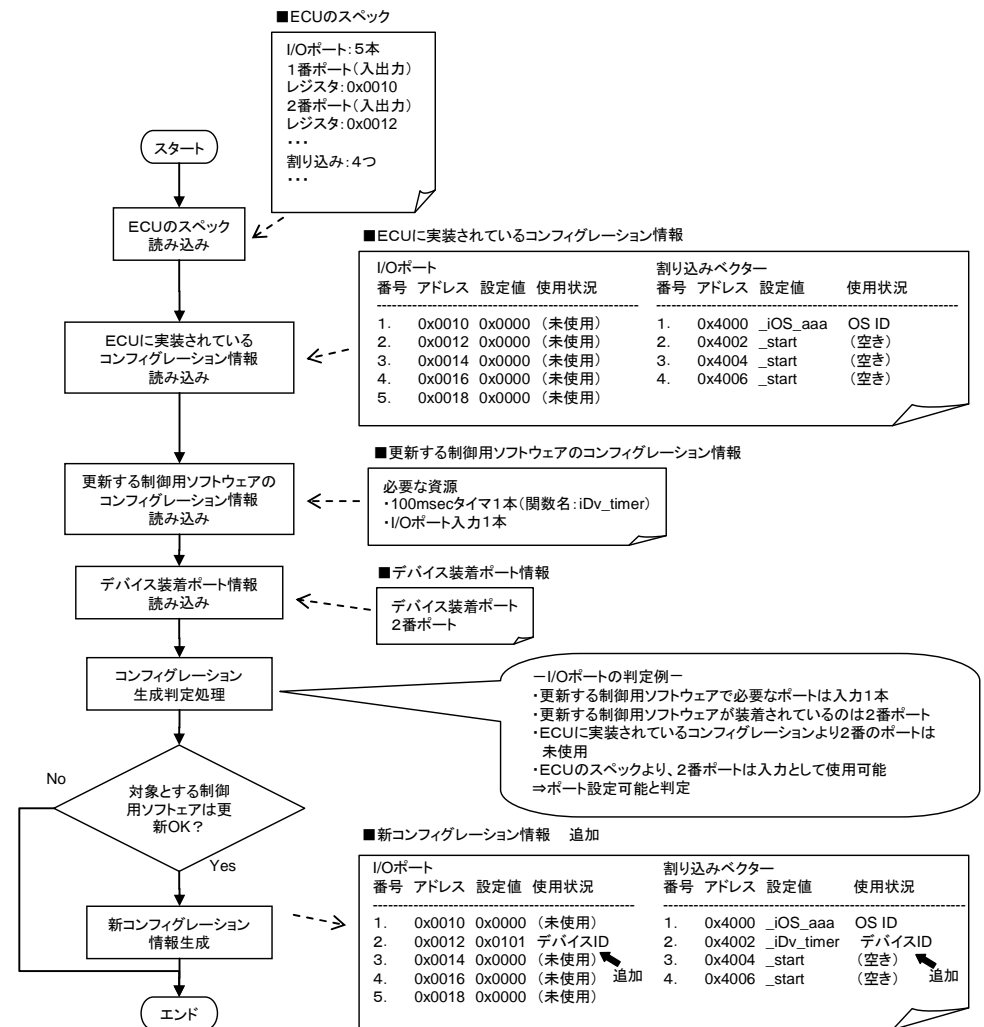


図 5-3 ECU のコンフィグレーション情報の生成手順

5.4 ECUにおけるコンフィグレーション情報の処理

ECUにおけるコンフィグレーション情報の処理を図 5-4に示す。従来、ECUのハードウェア資源に対する割り当てや設定はECU開発時に行われ、実行内容は実行可能ファイルに組み込まれていた。しかし、次の二つの理由から、コンフィグレーション情報は、図 5-4の通りコード領域とは別にコンフィグレーションデータ領域として確保し、この領域に格納する。1つ目の理由は、デバイスの装着においてECUのハードウェア資源に対する割り当てや設定に関して随時更新が必要となるため、他の領域への書き換えの影響を抑えるために、書き換える領域を予め区切っておくことである。2つ目の理由は、外部端末と接続した時にECUのコンフィグレーション情報を送るためである。コンフィグレーションデータ領域にデータとして格納していれば、読み出したデータをそのまま送信することが可能である。

ECUの動作時は、図 5-4の通り電源の投入後の初期化处理において、スタック設定を行った直後にコンフィグレーションデータ領域を読みに行き、ここで読み取った情報を元に、割り込みベクタ設定、レジスタ設定、ポート入出力設定を行う。

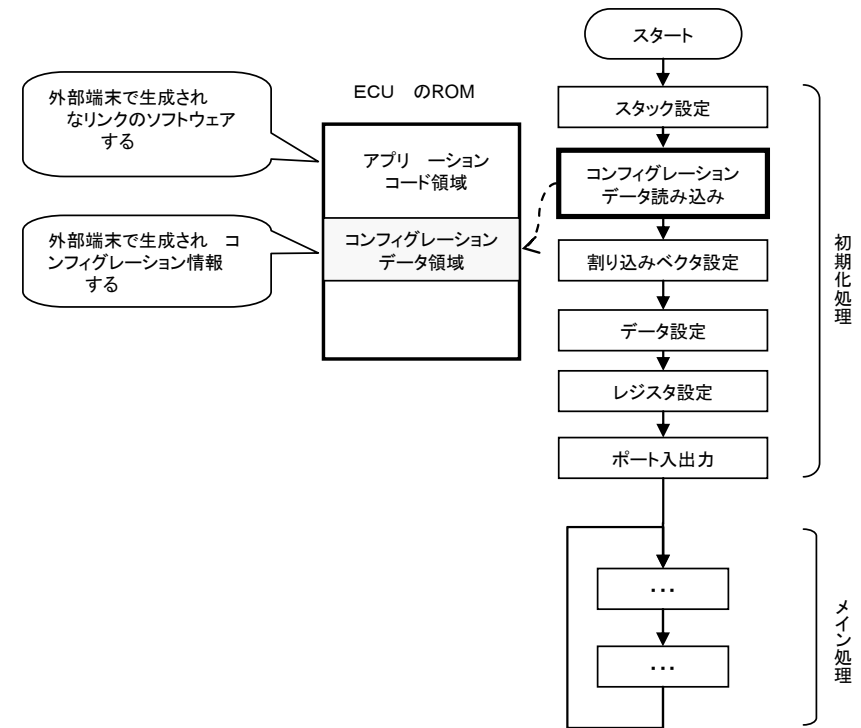


図 5-4 ECU におけるコンフィグレーション情報の処理

6. まとめ

本稿では、車載向けにデバイスのハードウェア資源の管理が必要な制御用ソフトウェアに関するソフトウェア更新の方式を述べた。本方式では、車載向けの装置を対象としているため、コストとリアルタイム性を要件とした。これを実現するために、外部端末を用いて更新しようとする制御用ソフトウェアが必要とするハードウェア資源の情報を得て、外部端末にて物理アドレスを固定した実行可能ファイルと更新する制御用ソフトウェアのコンフィグレーションを加えた ECU 全体のコンフィグレーション情報を生成することを提案した。

今後は試作を行い，機能評価を行っていく．また，異常に関する課題の抽出と対策の検討を進めていく．

参考文献

- 1) 原田 雅章：組込みソフトウェアのバージョンアップ機能を持った μ ITRON仕様OS，IPA，2002年度（平成14年度）成果報告集
- 2) 施 欣漢，中西 恒夫，久住 憲嗣，福田 晃：放送による車載機器向けソフトウェア差分更新方式，情報処理学会，Vol.2011-EMB-20 No.23