(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2012-155682
(P2012-155682A)
(43)公開日　平成24年8月16日(2012.8.16)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 11/00	(2006.01)	G06F	9/06	630C	5B376

審査請求 未請求　請求項の数 9　ＯＬ　（全 13 頁）

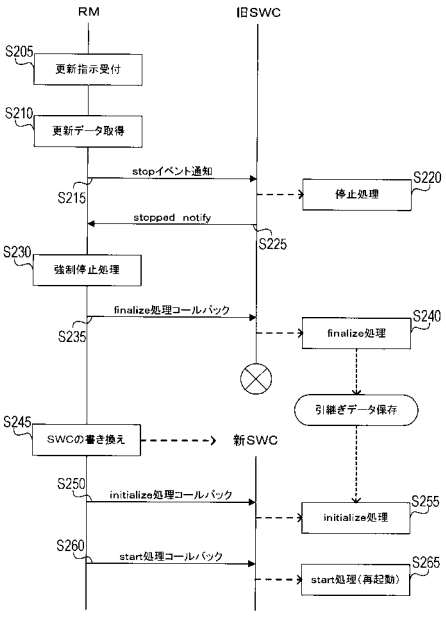
(21)出願番号　特願2011-16852(P2011-16852)
(22)出願日　平成23年1月28日(2011.1.28)

(71)出願人　000004260
株式会社デンソー
愛知県刈谷市昭和町１丁目１番地
(74)代理人　110000578
名古屋国際特許業務法人
(72)発明者　佐藤　洋介
愛知県刈谷市昭和町１丁目１番地　株式会社デンソー内

Ｆターム(参考)　5B376 AE30　CA22　CA76　FA11　GA08

(54)【発明の名称】組み込みシステム用のプラットフォーム、アプリケーション、該プラットフォームと該アプリケーションを備える制御プログラム、電子装置、及び、アプリケーションの更新方法

(57)【要約】
【課題】要求される機能を完全に停止させること無く、プログラムの一部を動的に書き換えることができる組み込みシステム用の制御プログラム等を提供する。
【解決手段】ＡＵＴＯＳＡＲの基本ソフトウェアにおける複合デバイスに含まれるＲＭ（Reconfiguration Management）は、更新指示を受け付けると、更新対象のＳＷＣに対しｓｔｏｐイベントを通知する（Ｓ２１５）。該イベントを受け取ったＳＷＣは、当該ＳＷＣを停止させると共に、ＲＴＥ（Runtime Environment）を介しての当該ＳＷＣへの送信を停止する停止処理を実行し（Ｓ２２０）、その後、ＲＭにｓｔｏｐｐｅｄ　ｎｏｔｉｆｙを通知する（Ｓ２２５）。該通知を受け取ったＲＭは、更新データにより更新対象のＳＷＣを書き換え（Ｓ２４５）、その後、該ＳＷＣを再起動させる。
【選択図】図３



【特許請求の範囲】
【請求項１】
１または複数のアプリケーションと、該アプリケーションを動作させるためのプラットフォームとを備える組み込みシステム用の制御プログラムであって、
前記プラットフォームは、
前記アプリケーションとの間の送受信を中継する中継手段と、
動作中の前記アプリケーションを更新する更新指示を受け付ける受付手段と、
前記アプリケーションを更新するための更新データを外部から取得する取得手段と、
前記受付手段が前記更新指示を受け付けると、該更新指示により更新される前記アプリケーションに対し、動作の正常な停止と、前記中継手段を介しての該アプリケーションへの送信の停止とを指示する停止指示を行う停止指示手段と、
前記停止指示により、前記アプリケーションの動作が正常に停止すると共に、前記中継手段を介しての該アプリケーションへの送信とが停止した後に、前記取得手段が取得した前記更新データに基づき該アプリケーションを更新する更新手段と、
してコンピュータを動作させ、
前記アプリケーションは、当該アプリケーションに対しての前記停止指示がなされると、当該アプリケーションの動作を正常に停止させると共に、前記中継手段を介しての当該アプリケーションへの送信を停止させる停止手段としてコンピュータを動作させること、
を特徴とする制御プログラム。
【請求項２】
請求項１に記載の制御プログラムにおいて、
前記アプリケーションは、さらに、
当該アプリケーションに対しての前記停止指示がなされると、当該アプリケーションの動作状態を示す引継ぎデータをメモリに保存する保存手段と、
前記更新手段により当該アプリケーションが更新された後に、前記保存手段により保存された前記引継ぎデータを読み出し、該引継ぎデータに基づき、前記停止指示がなされた時点の前記動作状態を復元した後に当該アプリケーションの動作を再開させる再開手段と、
してコンピュータを動作させることを特徴とする制御プログラム。
【請求項３】
請求項１または請求項２に記載の制御プログラムにおいて、
前記取得手段は、ネットワークを介して外部サーバから前記更新データを取得すること、
を特徴とする制御プログラム。
【請求項４】
請求項１から請求項３のうちのいずれか１項に記載の制御プログラムにおいて、
当該制御プログラムは、書き換え可能な不揮発性の記憶媒体に記憶された状態でコンピュータを動作させるよう構成されていること、
を特徴とする制御プログラム。
【請求項５】
請求項１から請求項４のうちのいずれか１項に記載の制御プログラムにおいて、
前記プラットフォームは、さらに、前記停止指示がなされた前記アプリケーションの動作が停止しない場合に、該アプリケーションの動作を停止させる強制停止手段としてコンピュータを動作させることを特徴とする制御プログラム。
【請求項６】
１または複数のアプリケーションと、該アプリケーションを動作させるためのプラットフォームであって、
前記アプリケーションとの間の送受信を中継する中継手段と、
動作中の前記アプリケーションを更新する更新指示を受け付ける受付手段と、
前記アプリケーションを更新するための更新データを外部から取得する取得手段と、

前記受付手段が前記更新指示を受け付けると、該更新指示により更新される前記アプリケーションに対し、動作の正常な停止と、前記中継手段を介しての該アプリケーションへの送信の停止とを指示する停止指示を行う停止指示手段と、
前記停止指示により、前記アプリケーションの動作が正常に停止すると共に、前記中継手段を介しての該アプリケーションへの送信とが停止した後に、前記取得手段が取得した前記更新データに基づき該アプリケーションを更新する更新手段と、
してコンピュータを動作させることを特徴とするプラットフォーム。
【請求項７】
当該部位を個別に更新するプラットフォーム上で動作し、該プラットフォームを介して他のソフトウェアモジュールとの間の通信を行う組み込みシステム用のアプリケーションであって、
前記プラットフォームが動作中の当該アプリケーションを個別に更新する際に、該プラットフォームからの停止指示に応じて、当該アプリケーションの動作を正常に停止させると共に、前記プラットフォームを介しての当該アプリケーションへの送信を停止させる停止手段としてコンピュータを動作させることを特徴とするアプリケーション。
【請求項８】
１または複数のアプリケーションと、該アプリケーションを動作させるためのプラットフォームとを備える組み込みシステム用の制御プログラムにより動作するコンピュータを備える電子装置であって、
前記プラットフォームは、
前記アプリケーションとの間の送受信を中継する中継手段と、
動作中の前記アプリケーションを更新する更新指示を受け付ける受付手段と、
前記アプリケーションを更新するための更新データを外部から取得する取得手段と、
前記受付手段が前記更新指示を受け付けると、該更新指示により更新される前記アプリケーションに対し、動作の正常な停止と、前記中継手段を介しての該アプリケーションへの送信の停止とを指示する停止指示を行う停止指示手段と、
前記停止指示により、前記アプリケーションの動作が正常に停止すると共に、前記中継手段を介しての該アプリケーションへの送信とが停止した後に、前記取得手段が取得した前記更新データに基づき該アプリケーションを更新する更新手段と、
して前記コンピュータを動作させ、
前記アプリケーションは、当該アプリケーションに対しての前記停止指示がなされると、当該アプリケーションの動作を正常に停止させると共に、前記中継手段を介しての当該アプリケーションへの送信を停止させる停止手段として前記コンピュータを動作させること、
を特徴とする電子装置。
【請求項９】
１または複数のアプリケーションと、該アプリケーションを動作させると共に、該アプリケーションとの間の送受信を中継するプラットフォームとを備える組み込みシステム用の制御プログラムにおける、該アプリケーションの更新方法であって、
前記プラットフォームにて、動作中の前記アプリケーションを更新する更新指示を受け付ける受付手順と、
前記プラットフォームにて、前記アプリケーションを更新するための更新データを外部から取得する取得手順と、
前記更新指示を受け付けると、前記プラットフォームから、該更新指示により更新される前記アプリケーションに対し、動作の正常な停止と、前記プラットフォームを介しての該アプリケーションへの送信の停止とを指示する停止指示を行う停止指示手順と、
前記停止指示がなされた前記アプリケーションにて、当該アプリケーションの動作を正常に停止させると共に、前記プラットフォームを介しての当該アプリケーションへの送信を停止させる停止手順と、
前記停止指示により、前記アプリケーションの動作が正常に停止すると共に、前記プラ

ットフォームを介しての該アプリケーションへの送信とが停止した後に、前記プラットフォームが、前記更新データに基づき該アプリケーションを更新する更新手順と、
を有することを特徴とする更新方法。
【発明の詳細な説明】
【技術分野】
【０００１】
１または複数のアプリケーションと、該アプリケーションを動作させるプラットフォームを備える組み込みシステム用の制御プログラム等に関する。
【背景技術】
【０００２】
従来、車載装置等を制御する組み込みシステム用のプログラムは、経済性や安全性のため、複数のモジュールから構成されている場合であっても各モジュールが静的にリンクされた状態でＲＯＭに記憶され、メインメモリにロードされた後に実行されるか、或いは、ＲＯＭから直接実行されていた。このため、プログラムの一部を動的に書き換えることはできず、専用装置によりプログラム全体を一括して書き換える等といった必要があったため、ユーザが手軽にプログラムのバージョンアップを行うことができなかった。しかしながら、近年では、プログラムの大規模化に伴い不具合の発生頻度が増加しており、プログラムの書き換えを容易化することが望まれている。
【０００３】
ここで、特許文献１に記載されているように、組み込みシステムのＲＴＯＳ上にさらにもう一層のソフトウェア動作環境を設け、このソフトウェア動作環境上でモジュールの動的なリンクやロードをサポートすることが提案されている。このような構成によれば、ソフトウェア動作環境上で一部のモジュールを動的に書き換えることが可能となるが、その反面、ＣＰＵの処理負荷やメモリの使用量が膨大なものとなり、リソースが限られている車載装置等への適用は困難であった。
【０００４】
また、上記事情に鑑み、ＴＯＰＰＥＲＳのように、プログラムの一部を動的に書き換えるダイナミックローディング機能が搭載されたＲＴＯＳが登場している。このようなＲＴＯＳを用いることで、一部のモジュールのみを更新することができ、より容易に新機能の追加や不具合の修正を行なうことが可能となる。
【先行技術文献】
【特許文献】
【０００５】
【特許文献１】特開２００３－２５６２１６号公報
【発明の概要】
【発明が解決しようとする課題】
【０００６】
ここで、プログラムを構成する各モジュールは、互いに通信を行い協調して動作することで要求される機能を実現しており、プログラムの稼動中に特定のモジュールを書き換えるとした場合には、書き換えの途上にあるモジュールが他のモジュール等からの指示により起動されてしまう可能性がある。そして、このような場合には、ＣＰＵが暴走してしまうおそれがあるため、上述のＴＯＰＰＥＲＳでは、各モジュールを停止させ、機能を完全に停止させた後でなければ、プログラムの一部を動的に書き換えることができなかった。
【０００７】
本願発明は上記課題に鑑みてなされたものであり、要求される機能を完全に停止させること無く、プログラムの一部を動的に書き換えることができる組み込みシステム用の制御プログラム等を提供することを目的とする。
【課題を解決するための手段】
【０００８】
上記課題を解決するためになされた請求項１の発明は、１または複数のアプリケーショ

ンと、該アプリケーションを動作させるためのプラットフォームとを備える組み込みシステム用の制御プログラムに関する。

【０００９】

この制御プログラムにおいて、プラットフォームは、アプリケーションとの間の送受信を中継する中継手段と、動作中のアプリケーションを更新する更新指示を受け付ける受付手段と、アプリケーションを更新するための更新データを外部から取得する取得手段と、受付手段が更新指示を受け付けると、該更新指示により更新されるアプリケーションに対し、動作の正常な停止と、中継手段を介しての該アプリケーションへの送信の停止とを指示する停止指示を行う停止指示手段としてコンピュータを動作させる。また、プラットフォームは、停止指示により、アプリケーションの動作が正常に停止すると共に、中継手段を介しての該アプリケーションへの送信とが停止した後に、取得手段が取得した更新データに基づき該アプリケーションを更新する更新手段としてコンピュータを動作させる。

【００１０】

また、アプリケーションは、当該アプリケーションに対しての停止指示がなされると、当該アプリケーションの動作を正常に停止させると共に、中継手段を介しての当該アプリケーションへの送信を停止させる停止手段としてコンピュータを動作させる。

【００１１】

このような構成によれば、更新指示がなされた際には、更新指示の対象となるアプリケーションの動作と、中継手段を介しての該アプリケーションへの送信が停止された後に該アプリケーションが更新され、更新中のアプリケーションに対してイベント通知等がなされることを防ぐことができる。このため、他のアプリケーションやプラットフォームの動作を停止させずに、特定のアプリケーションのみを更新したとしても、更新途上のアプリケーションが起動されることは無く、ＣＰＵが暴走してしまうことを防ぐことができるのである。したがって、請求項１に記載の制御プログラムによれば、各アプリケーションにより実現される全ての機能を完全に停止させること無く、プログラムの一部を動的に書き換えることができる。

【００１２】

ここで、アプリケーションの更新後、該アプリケーションを初期状態で再起動させるとなると、更新の前後でアプリケーションの動作が大きく変わってしまう可能性があり、制御対象の状態に適合しない処理や、ユーザの意思に反した処理等が行われるおそれがある。

【００１３】

そこで、請求項２に記載の制御プログラムでは、アプリケーションは、さらに、当該アプリケーションに対しての停止指示がなされると、当該アプリケーションの動作状態を示す引継ぎデータをメモリに保存する保存手段と、更新手段により当該アプリケーションが更新された後に、保存手段により保存された引継ぎデータを読み出し、該引継ぎデータに基づき、停止指示がなされた時点の動作状態を復元した後に当該アプリケーションの動作を再開させる再開手段としてコンピュータを動作させる。

【００１４】

こうすることにより、更新後のアプリケーションを更新前と同様に動作させることができ、アプリケーションの更新により、制御対象の状態に適合しない処理や、ユーザの意思に反した処理等が行われてしまうことを防ぐことができる。

【００１５】

また、請求項３に記載されているように、取得手段は、ネットワークを介して外部サーバから更新データを取得しても良い。

こうすることにより、アプリケーションの更新を容易に行うことができる。

【００１６】

また、請求項４に記載されているように、制御プログラムは、書き換え可能な不揮発性の記憶媒体に記憶された状態でコンピュータを動作させるよう構成されていても良い。

こうすることにより、制御プログラムをロードするためのメインメモリを設ける必要が

無くなり、制御プログラムが搭載される装置のコストを抑えることができる。
【００１７】
また、制御プログラムを構成する各アプリケーションは、異なる業者により開発されるという可能性もある。このため、アプリケーションの更新機能が十分にサポートされておらず、プラットフォームからの停止指示に応じてアプリケーションが停止しないという可能性もある。また、アプリケーションの更新機能が十分にサポートされている場合であっても、何らかの理由により、停止指示がなされたアプリケーションが動作を停止できないといった可能性もある。
【００１８】
そこで、請求項５に記載の制御プログラムでは、プラットフォームは、さらに、停止指示がなされたアプリケーションの動作が停止しない場合に、該アプリケーションの動作を停止させる強制停止手段としてコンピュータを動作させる制御プログラム。
【００１９】
こうすることにより、更新を行うアプリケーションの動作を確実に停止させることができ、アプリケーションの更新を確実に行うことができる。
なお、請求項６，７に記載されているように、請求項１に記載の制御プログラムを構成するプラットフォームやアプリケーションを単体で市場に流通させても良い。このような場合であっても、対応するアプリケーションやプラットフォームと組み合わせることで、同様の効果を得ることができる。
【００２０】
また、請求項８に記載されているように、請求項１に記載の制御プログラムが搭載された電子装置を市場に流通させても良い。このような場合であっても、同様の効果を得ることができる。
【００２１】
また、請求項９には、請求項１に記載の制御プログラムにて行われるアプリケーションの更新方法が記載されている。このような方法によれば、アプリケーションにより実現される全ての機能を完全に停止させること無く、プログラムの一部を動的に書き換えることができる。
【図面の簡単な説明】
【００２２】
【図１】ＥＣＵの構成やＡＵＴＯＳＡＲの構造についてのブロック図である。
【図２】制御プログラムの構成を示すブロック図と、ＳＷＣの状態遷移図である。
【図３】ＳＷＣの書き換え処理のシーケンス図である。
【図４】ＳＷＣの書き換え処理のシーケンス図である。
【発明を実施するための形態】
【００２３】
以下、本発明の実施形態について図面を用いて説明する。なお、本発明の実施の形態は、下記の実施形態に何ら限定されることはなく、本発明の技術的範囲に属する限り種々の形態を採りうる。
【００２４】
［構成の説明］
図１（ａ）は、自車両の車速を自動調整するクルーズコントロールや、ヘッドライトの制御や、自車両が車線に追従して走行するようステアリングを自動調整する車線追従等の機能を有する本実施形態のＥＣＵ１０の構成を示すブロック図である。ＥＣＵ１０は、無線通信によりインターネットにアクセスするための無線通信部１１と、書き換え可能な不揮発性メモリ（例えばフラッシュメモリ等）として構成されている記憶部１２と、ユーザからの操作を受け付ける操作部１３と、ＣＰＵ、ＲＯＭ、ＲＡＭ、Ｉ／Ｏ及びこれらを接続するバスライン等からなる周知のマイコンを中心に構成され、記憶部１２に記憶されている制御プログラムにより当該ＥＣＵ１０を統括制御する制御部１４とを有している。なお、図１（ｂ）に記載されているように、ＥＣＵ１０は、無線通信部１１によりインター

ネット３０にアクセスして外部サーバ２０と通信を行い、該外部サーバ２０から記憶部１２に記憶されている制御プログラムを更新するための更新データを取得するよう構成されている。

【００２５】

次に、ＥＣＵ１０の記憶部１２に記憶されている制御プログラム１００の構成について説明する。制御プログラム１００は、車載用の組み込みシステム向けに標準化されたソフトウェアアーキテクチャーであるＡＵＴＯＳＡＲに準拠した構造を有している。

【００２６】

ここで、ＡＵＴＯＳＡＲについて簡単に説明する。図１（ｃ）に記載されているように、ＡＵＴＯＳＡＲは、各種機能を実現するためのアプリケーションが属するアプリケーション層と、ハードウェアを抽象化すると共に、各種サービスを提供する基本ソフトウェアと、基本ソフトウェアからアプリケーションを抽象化し、基本ソフトウェアとアプリケーションとの間や、アプリケーション間の通信を行うＲＴＥ（Runtime Environmentの略，ランタイム環境）から構成される。また、この基本ソフトウェアは、各アプリケーションを動作させるＯＳとして機能すると共に、他のＥＣＵとの通信や、メモリ管理や故障診断等を行うサービス層と、マイコンや各種デバイスを制御するためのＡＰＩを提供するＥＣＵ抽象化層と、マイコン内部のペリフェラルやメモリや各種デバイス等を制御するマイクロコントローラ抽象化層を備えると共に、高速での処理や、標準化できない特殊な処理等をサポートするための複合ドライバを備える。

【００２７】

そして、図２（ａ）に記載されているように、本実施形態の制御プログラム１００は、上記アプリケーション層に含まれる部位であって、１または複数のタスクから構成されたアプリケーションであるＡ～Ｃ－ＳＷＣ１４０～１６０と、上記サービス層に属する部位であって、ＳＷＣを構成するタスクのスケジューリングを行うことで、これらのアプリケーションを動作させるＯＳ１１０と、上記複合ドライバに属する部位であって、動作中のＳＷＣの書き換えを行うＲＭ（Reconfiguration Management）１２０と、ＳＷＣの書き換えに関連する各種インターフェースを提供する拡張Ｉ／Ｆ１３１が設けられた上述のＲＴＥ１３０と、を備える。

【００２８】

なお、ＲＴＥ１３０により提供されるインターフェースを呼び出すことで、各ＳＷＣに対して各種イベントが通知され、対応する処理が実行されると共に、各ＳＷＣの処理がコールバック関数として実行されるよう構成されている。また、ＲＴＥ１３０により提供されるインターフェースを呼び出すことで、ＲＭ１２０とＯＳ１１０との間の通信が行われる。

【００２９】

また、Ａ－ＳＷＣ１４０はクルーズコントロールを実現するためのアプリケーションであり、Ｂ－ＳＷＣ１５０は、ヘッドライトを制御するためのアプリケーションであり、Ｃ－ＳＷＣ１６０は、車線追従を実現するためのアプリケーションである。

【００３０】

また、本実施形態では、制御プログラム１００は、記憶部１２に記憶された状態で制御部１４を構成するマイコンにより実行されるよう構成されているが、これに限定されることは無く、メインメモリにロードされた状態でマイコンにより実行されるよう構成されていても良い。

【００３１】

［動作の説明］

次に、制御プログラム１００の動作について説明する。

（１）Ａ～Ｃ－ＳＷＣの状態について

既に述べたように、本実施形態の制御プログラム１００では、ＳＷＣは、外部サーバ２０から取得された更新データにより書き換え可能に構成されている。

【００３２】

そして、図２（ｂ）に記載されているように、ＳＷＣは、タスクの稼動状態や更新の状態等に基づき以下のような状態に分類される。すなわち、ＳＷＣの状態は、ＳＷＣが記憶部１２から削除された状態であるＵｎｉｎｓｔと、ＳＷＣが記憶部１２に書き込まれた状態であるＩｎｓｔａｌｌｅｄと、ＳＷＣの実行イメージが生成された状態であるＬｏａｄｅｄと、ＳＷＣのインスタンスが生成され、該インスタンスが停止した状態であるＳｔｏｐｐｅｄと、ＳＷＣのインスタンスが起動された状態であるＡｃｔに分類される。さらに、Ｕｎｉｎｓｔ，Ｉｎｓｔａｌｌｅｄが実行ファイルレベルとされ、Ｌｏａｄｅｄ，Ｓｔｏｐｐｅｄ，Ａｃｔがインスタンスレベルとされる。

【００３３】

また、ＳＷＣでは、以下のイベントにより状態遷移が発生する。

すなわち、Ｕｎｉｎｓｔである場合には、ＲＭ１２０がＳＷＣを記憶部１２に書き込むＩｎｓｔａｌｌイベントが発生すると、該ＳＷＣはＩｎｓｔａｌｌｅｄに遷移する。

【００３４】

また、Ｉｎｓｔａｌｌｅｄである場合には、ＲＭ１２０がＳＷＣを記憶部１２から削除するＵｎｉｎｓｔａｌｌイベントが発生すると、該ＳＷＣはＵｎｉｎｓｔに遷移する。また、Ｉｎｓｔａｌｌｅｄである場合には、ＯＳ１１０がＳＷＣにて用いられるデータの初期値をＲＡＭに展開するＬｏａｄイベントが発生すると、該ＳＷＣはＬｏａｄｅｄに遷移する。

【００３５】

また、Ｌｏａｄｅｄである場合には、ＯＳ１１０がＳＷＣを構成するタスクや割込みハンドラを登録するＩｎｉｔイベントが発生すると、該ＳＷＣはＳｔｏｐｐｅｄに遷移する。

【００３６】

また、Ｓｔｏｐｐｅｄである場合には、ＯＳ１１０がＳＷＣを構成するタスクを起動すると共に、該ＳＷＣに対応するデバイスからの割込みを許可するＳｔａｒｔイベントが発生すると、該ＳＷＣはＡｃｔに遷移する。また、Ｓｔｏｐｐｅｄである場合には、ＯＳ１１０がＳＷＣを構成するタスクや割込みハンドラの登録を解除するＦｉｎａｌｉｚｅイベントが発生すると、該ＳＷＣはＬｏａｄｅｄに遷移する。

【００３７】

なお、詳細については後述するが、ＳＷＣを更新する過程において、Ｆｉｎａｌｉｚｅイベントが発生した際には、該ＳＷＣの動作状態を示す引継ぎデータが保存されると共に、Ｉｎｉｔイベントが発生した際には、保存されている引継ぎデータが読み出され、該ＳＷＣの動作状態が復元される。

【００３８】

また、Ａｃｔである場合には、ＯＳ１１０がＳＷＣを構成するタスクを停止させると共に、該ＳＷＣに対応するデバイスからの割込みを禁止するＳｔｏｐイベントが発生すると、該ＳＷＣはＳｔｏｐｐｅｄに遷移する。

【００３９】

また、インスタンスレベルである場合に、ＯＳ１１０がＳＷＣにて用いられるデータを破棄するＥｘｉｔイベントが発生すると、該ＳＷＣはＩｎｓｔａｌｌｅｄに遷移する。

（２）アプリケーションの書き換えについて

次に、外部サーバ２０から取得した更新データにより動作中のＳＷＣを書き換える処理である書き換え処理について、図３に記載のシーケンス図を用いて説明する。なお、本処理は、ＥＣＵ１０の稼動中に、ユーザから動作中のいずれかのＳＷＣを書き換える指令を受け付けた際に開始される。また、以下の説明では、制御プログラム１００を構成する部位が各種処理を行うことが記載されているが、これらの処理は、該部位に従い動作する制御部１４により実現されるということを念のため付言しておく。

【００４０】

制御プログラム１００のＲＭ１２０は、ＥＣＵ１０の操作部１３を介して、ユーザから、動作中のいずれかのＳＷＣを書き換える更新指示を受け付けると（Ｓ２０５）、無線通

信部１１を介して、外部サーバ２０から、書き換えの対象となるＳＷＣ（対象ＳＷＣとも記載）を書き換えるための更新データを取得する（Ｓ２１０）。

【００４１】

そして、Ｓ２１５では、ＲＭ１２０は、ＲＴＥ１３０のインターフェースを呼び出すことで、対象ＳＷＣに対し、動作を正常に停止させるｓｔｏｐイベントを通知する。

そして、対象ＳＷＣは、ｓｔｏｐイベントを受け取ると、当該ＳＷＣを正常に停止させるための処理である停止処理を実行する（Ｓ２２０）。

【００４２】

具体的には、停止処理において、対象ＳＷＣは、実行中の処理を、安全に停止可能な段階まで進行させた上で中断する。なお、対象ＳＷＣに対するイベント通知や、対象ＳＷＣのコールバック関数の呼び出しがなされている場合には、対応する処理の実行完了を待つ。その後、ＲＴＥ１３０のインターフェースを呼び出し、ＯＳ１１０に対し、当該ＳＷＣの動作の停止要求を行う。また、対象ＳＷＣは、ＲＴＥ１３０のインターフェースを呼び出し、ＲＴＥ１３０を介しての当該ＳＷＣへのイベントの通知や、当該ＳＷＣのコールバック関数（当該ＳＷＣの書き換えに関連するコールバック関数を除く）の呼び出しを禁止する。

【００４３】

また、対象ＳＷＣの動作の停止要求を受け取ったＯＳ１１０は、対象ＳＷＣを構成するタスクの稼動を順次停止させる。

停止処理の実行後に移行するＳ２２５では、対象ＳＷＣは、ＲＭ１２０のインターフェースを呼び出し、ＲＭ１２０に対し、停止処理が正常に実行されたことを示すｓｔｏｐｐｅｄ　ｎｏｔｉｆｙを通知し、Ｓ２３０に処理が移行される。なお、ｓｔｏｐｐｅｄ　ｎｏｔｉｆｙが通知される時点では、対象ＳＷＣを構成するタスクのうち、対象ＳＷＣを統括制御するタスク（ｓｔｏｐｐｅｄ　ｎｏｔｉｆｙを通知するタスク）以外の他のタスクは、全て停止された状態となっている。

【００４４】

Ｓ２３０では、対象ＳＷＣが停止しない場合を想定し、ｓｔｏｐｐｅｄ　ｎｏｔｉｆｙを受け取ったＲＭ１２０において、ＲＴＥ１３０のインターフェースを呼び出し、対象ＳＷＣを強制的に停止させる強制停止処理を実行しても良い。なお、該インターフェースは、ＲＴＥ１３０の拡張Ｉ／Ｆ１３１として設けられている。具体的には、強制停止処理において、ＲＭ１２０は、ＲＴＥ１３０のインターフェースを呼び出し、ＯＳ１１０に対し、対象ＳＷＣのタスクを全て強制停止させると共に、対象ＳＷＣが既に受信しているイベントや、既に呼び出された対象ＳＷＣのコールバック関数を全て無効とさせる。強制停止処理を経た後に、Ｓ２３５に処理が移行される。

【００４５】

Ｓ２３５では、ＲＭ１２０は、対象ＳＷＣの動作状態を保存するための処理であるｆｉｎａｌｉｚｅ処理をコールバック関数として呼び出し、Ｓ２４０に処理が移行される。なお、該処理の呼び出しがＦｉｎａｌｉｚｅイベントに相当する。

【００４６】

Ｓ２４０では、対象ＳＷＣはｆｉｎａｌｉｚｅ処理を実行し、該処理において、対象ＳＷＣの動作状態を示すデータを引継ぎデータとしてＲＡＭ（或いは、図示しないＥＥＰＲＯＭの不揮発性記憶媒体等）に保存する。そして、ｆｉｎａｌｉｚｅ処理の実行後、対象ＳＷＣを統括制御するタスクが停止される。

【００４７】

なお、ｆｉｎａｌｉｚｅ処理の実行後、ＯＳ１１０は、対象ＳＷＣを構成するタスクや割込みハンドラの登録を解除する処理を行う。

続くＳ２４５では、ＲＭ１２０は、対象ＳＷＣで用いられるデータをＲＡＭ上から破棄すると共に（Ｅｘｉｔイベントに相当）、対象ＳＷＣを記憶部１２から消去する（Ｕｎｉｎｓｔａｌｌイベントに相当）。その後、ＲＭ１２０は、外部サーバ２０から取得され、制御部１４のＲＡＭに一時的に保存されている更新データを記憶部１２に書き込むことで

、対象ＳＷＣの書き換えを行い（Ｉｎｓｔａｌｌイベントに相当）、さらに、ＯＳ１１０が対象ＳＷＣにて用いられるデータの初期値をＲＡＭに展開する（Ｌｏａｄイベントに相当）。

【００４８】

続くＳ２５０では、ＲＭ１２０は、書き換え直前の対象ＳＷＣの動作状態の復元等を行う処理であるｉｎｉｔｉａｌｉｚｅ処理をコールバック関数として呼び出し、その後、Ｓ２５５に処理が移行される。なお、該処理の呼び出しがＩｎｉｔイベントに相当する。

【００４９】

Ｓ２５５では、対象ＳＷＣはｉｎｉｔｉａｌｉｚｅ処理を実行し、該処理において、ＲＡＭに保存された引継ぎデータを読み出し、書き換え直前の対象ＳＷＣの動作状態を復元する。また、該処理において、対象ＳＷＣは、ＲＴＥ１３０のインターフェースを呼び出し、ＲＴＥ１３０を介しての当該ＳＷＣへのイベントの通知や、当該ＳＷＣのコールバック関数の呼び出しを許可する。また、該処理の実行後、ＯＳ１１０は、対象ＳＷＣを構成するタスクや割込みハンドラを登録する（Ｉｎｉｔイベントに相当）。そして、Ｓ２６０に処理が移行される。

【００５０】

Ｓ２６０では、ＲＭ１２０は、書き換え後の対象ＳＷＣを再起動させるための処理であるｓｔａｒｔ処理をコールバック関数として読み出し、その後、Ｓ２６５に処理が移行される。

【００５１】

Ｓ２６５では、対象ＳＷＣはｓｔａｒｔ処理を実行する。具体的には、対象ＳＷＣは、ＲＴＥ１３０のインターフェースを呼び出してＯＳ１１０に対し当該ＳＷＣの動作の再開要求を行い、該再開要求を受け取ったＯＳ１１０は、対象ＳＷＣを構成するタスクを起動すると共に、対象ＳＷＣに対応するデバイスからの割込みを許可する（Ｓｔａｒｔイベントに相当）。そして、以後、対象ＳＷＣの動作が再開される。

【００５２】

一方、書き換え処理において、対象ＳＷＣからｓｔｏｐｐｅｄ　ｎｏｔｉｆｙの通知がなされない場合には、以下のような処理が行われる（図４参照）。

すなわち、既に述べたように、制御プログラム１００のＲＭ１２０が更新指示を受け付け（Ｓ３０５）、外部サーバ２０から更新データを取得すると共に（Ｓ３１０）、対象ＳＷＣに対しｓｔｏｐイベントを通知すると（Ｓ３１５）、対象ＳＷＣでは停止処理が実行される（Ｓ３２０）。そして、ｓｔｏｐイベントを通知した後、所定時間（例えば１ｓ）が経過しても、対象ＳＷＣからｓｔｏｐｐｅｄ　ｎｏｔｉｆｙが通知されない場合には、ＲＭ１２０は、ＲＴＥ１３０のインターフェースを呼び出し、対象ＳＷＣを強制的に停止させる強制停止処理を実行する（Ｓ３２５）。

【００５３】

続くＳ３３０では、ＲＭ１２０は、対象ＳＷＣのデータをＲＡＭ上から消去すると共に（Ｅｘｉｔイベントに相当）、対象ＳＷＣに対応するプログラムを記憶部１２から消去する（Ｕｎｉｎｓｔａｌｌイベントに相当）。その後、ＲＭ１２０は、外部サーバ２０から取得され、制御部１４のＲＡＭに一時的に保存されている更新データを記憶部１２に書き込むことで、対象ＳＷＣの書き換えを行い（Ｉｎｓｔａｌｌイベントに相当）、さらに、ＯＳ１１０が対象ＳＷＣにて用いられるデータの初期値をＲＡＭに展開する（Ｌｏａｄイベントに相当）。

【００５４】

続くＳ３３５では、ＲＭ１２０は、ｉｎｉｔｉａｌｉｚｅ処理をコールバック関数として呼び出し、その後、Ｓ３４０に処理が移行される。なお、該処理の呼び出しがＩｎｉｔイベントに相当する。

【００５５】

Ｓ３４０では、対象ＳＷＣはｉｎｉｔｉａｌｉｚｅ処理を実行し、該処理において、ＲＴＥ１３０のインターフェースを呼び出し、ＲＴＥ１３０を介しての当該ＳＷＣへのイベ

ントの通知や、当該ＳＷＣのコールバック関数の呼び出しを許可する。また、該処理の実行後、ＯＳ１１０は、対象ＳＷＣを構成するタスクや割込みハンドラを登録する（Ｉｎｉｔイベントに相当）。そして、Ｓ３４５に処理が移行される。

【００５６】

Ｓ３４５では、ＲＭ１２０は、書き換え後の対象ＳＷＣを再起動させるための処理であるｓｔａｒｔ処理をコールバック関数として読み出し、その後、Ｓ３５０に処理が移行される。

【００５７】

Ｓ３５０では、対象ＳＷＣはｓｔａｒｔ処理を実行する。具体的には、対象ＳＷＣは、ＲＴＥ１３０のインターフェースを呼び出してＯＳ１１０に対し当該ＳＷＣの動作の再開要求を行い、該再開要求を受け取ったＯＳ１１０は、対象ＳＷＣを構成するタスクを起動すると共に、対象ＳＷＣに対応するデバイスからの割込みを許可する（Ｓｔａｒｔイベントに相当）。そして、以後、対象ＳＷＣの動作が再開される。

【００５８】

［効果］

本実施形態の制御プログラム１００によれば、更新指示を受け付けた際には、更新の対象となるＳＷＣの動作と、ＲＴＥ１３０を介しての該ＳＷＣへの送信が停止された後に、該ＳＷＣが書き換えられるため、ＳＷＣの書き換え中に、該ＳＷＣに対してのイベント通知や、該ＳＷＣのコールバック関数の実行がなされることを防ぐことができる。このため、全てのＳＷＣの動作を停止させずに特定のＳＷＣのみを更新したとしても、更新途上のＳＷＣが起動されることは無く、ＣＰＵが暴走してしまうことを防ぐことができるのである。したがって、制御プログラム１００によれば、各ＳＷＣにより実現される全ての機能を完全に停止させること無く、特定のＳＷＣを動的に書き換えることができる。

【００５９】

［他の実施形態］

（１）本実施形態では、ユーザからの操作によりＳＷＣの書き換えが行われるが、これに限定されることは無く、例えば、無線通信部１１を介してインターネット経由で外部から受け付けた指示等に応じて、ＳＷＣの書き換えを行っても良い。

【００６０】

また、本実施形態では、外部サーバ２０から更新データが取得されるが、これに限定されることは無く、例えば、ＳＷＣの書き換えを行う際にＥＣＵ１０に記憶装置を接続し、該記憶装置から更新データを取得しても良い。このような場合であっても、同様の効果を得ることができる。

【００６１】

（２）また、本実施形態では、ＳＷＣの書き換えを行う際に、書き換えの対象となるＳＷＣが引継ぎデータの保存を行っている。しかしながら、これに限定されることは無く、例えば、他のＳＷＣとの間で共通化されているデータを引継ぎデータとして保存する場合等には、これらのデータに関しては、ＲＭ１２０やＯＳ１１０等にて引継ぎデータとして保存しても良い。このような場合であっても、同様の効果を得ることができる。

【００６２】

また、本実施形態では、ＳＷＣの書き換えを行う際に、書き換えの対象となるＳＷＣが、当該ＳＷＣへのイベント通知やコールバックを禁止している。この点に関しても、例えば、他のＳＷＣとの間で共通化されているイベントやコールバック関数が存在する場合には、これらに関しては、ＲＭ１２０やＯＳ１１０等でイベント通知やコールバックを禁止しても良い。このような場合であっても、同様の効果を得ることができる。

【００６３】

［特許請求の範囲との対応］

上記実施形態の説明で用いた用語と、特許請求の範囲の記載に用いた用語との対応を示す。

【００６４】

ＳＷＣがアプリケーションに、ＯＳ１１０，ＲＭ１２０，ＲＴＥ１３０がプラットフォームに、ＲＴＥ１３０が中継手段に相当する。

また、書き換え処理のＳ２０５，Ｓ３０５が受付手段，受付手順に、Ｓ２１０，Ｓ３１０が取得手段，取得手順に、Ｓ２１５，Ｓ３１５が停止指示手段，停止指示手順に、Ｓ２２０が停止手段，停止手順に、Ｓ３２５が強制停止手段に、Ｓ２４０が保存手段に、Ｓ２４５，Ｓ３３０が更新手段，更新手順に、Ｓ２５５，Ｓ２６５が再開手段に相当する。

【００６５】

また、ＳＷＣ，ＯＳ１１０，ＲＭ１２０等がソフトウェアモジュールに相当する。

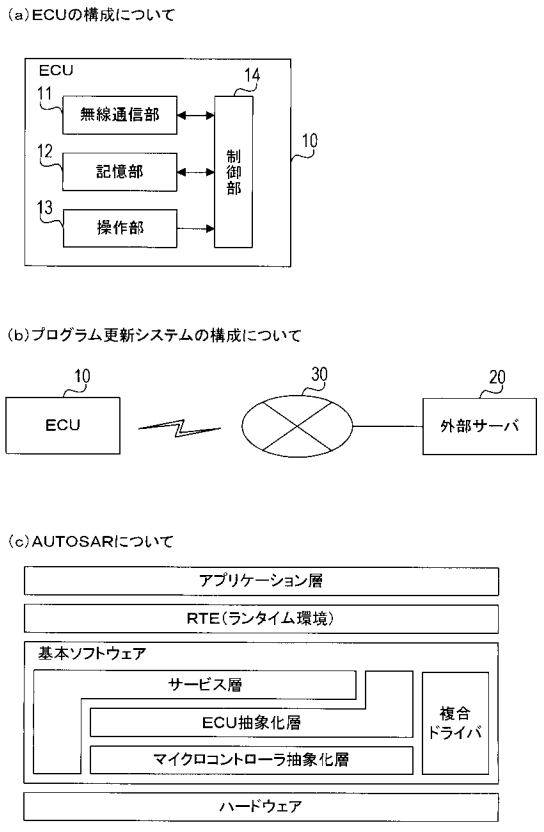
また、ＥＣＵ１０が電子装置に相当する。

【符号の説明】

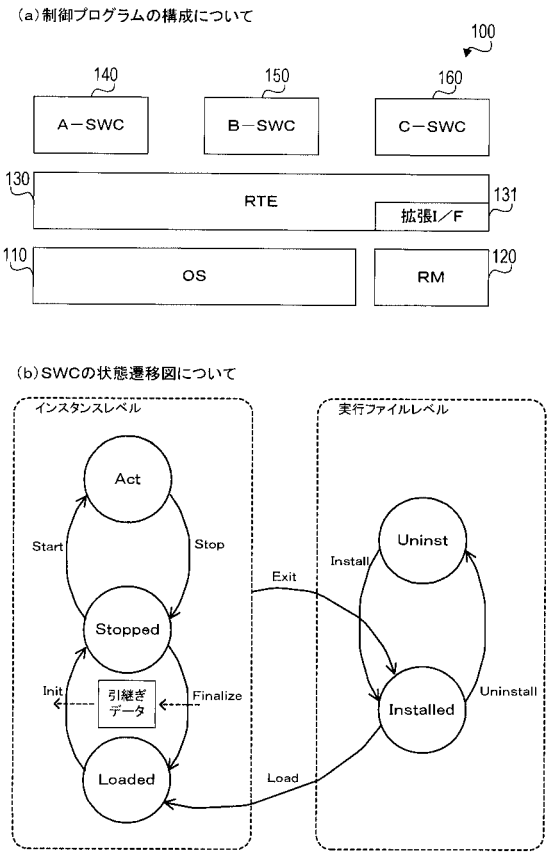
【００６６】

１０…ＥＣＵ、１１…無線通信部、１２…記憶部、１３…操作部、１４…制御部、２０…外部サーバ、３０…インターネット、１００…制御プログラム、１１０…ＯＳ、１２０…ＲＭ、１３０…ＲＴＥ、１３１…拡張Ｉ／Ｆ、１４０…Ａ－ＳＷＣ、１５０…Ｂ－ＳＷＣ、１６０…Ｃ－ＳＷＣ。

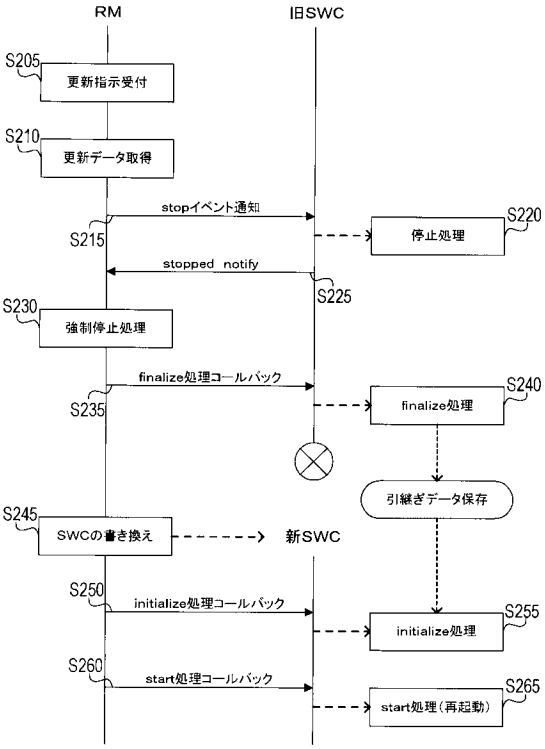
【図１】



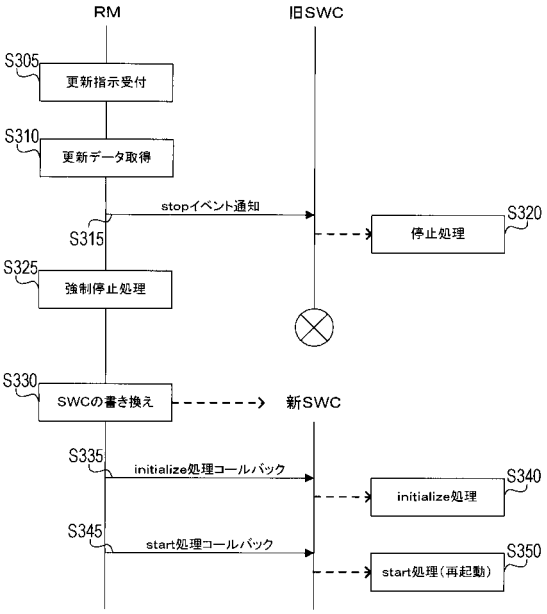
【図２】



【図３】


RM
旧SWC
S205
更新指示受付
S210
更新データ取得
stopイベント通知
S215
S220
停止処理
stopped_notify
S225
S230
強制停止処理
finalize処理コールバック
S235
S240
finalize処理
引継ぎデータ保存
S245
SWCの書き換え
新SWC
S250
initialize処理コールバック
S255
initialize処理
S260
start処理コールバック
S265
start処理(再起動)

【図４】


RM
旧SWC
S305
更新指示受付
S310
更新データ取得
stopイベント通知
S315
S320
停止処理
S325
強制停止処理
S330
SWCの書き換え
新SWC
S335
initialize処理コールバック
S340
initialize処理
S345
start処理コールバック
S350
start処理(再起動)