

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5113700号  
(P5113700)

(45) 発行日 平成25年1月9日(2013.1.9)

(24) 登録日 平成24年10月19日(2012.10.19)

(51) Int. Cl. F 1  
**G 0 6 F 11/00 (2006.01)** G 0 6 F 9/06 6 3 0 D

請求項の数 8 (全 13 頁)

(21) 出願番号	特願2008-244754 (P2008-244754)	(73) 特許権者	000233055
(22) 出願日	平成20年9月24日 (2008. 9. 24)		株式会社日立ソリューションズ
(65) 公開番号	特開2010-79440 (P2010-79440A)		東京都品川区東品川四丁目12番7号
(43) 公開日	平成22年4月8日 (2010. 4. 8)	(74) 代理人	100091096
審査請求日	平成23年1月12日 (2011. 1. 12)		弁理士 平木 祐輔
		(74) 代理人	100105463
			弁理士 関谷 三男
		(74) 代理人	100102576
			弁理士 渡辺 敏章
		(74) 代理人	100101063
			弁理士 松丸 秀和
		(72) 発明者	大浜 伸之
			東京都品川区東品川四丁目12番7号 日 立ソフトウェアエンジニアリング株式会社 内

最終頁に続く

(54) 【発明の名称】 ファームウェア更新装置及び方法

(57) 【特許請求の範囲】

【請求項1】

ファームウェアを有し、このファームウェアを更新する機能を備えるファームウェア更新装置であって、

更新プログラムに従って、ファイル単位でファームウェアの更新処理を実行する演算処理部と、

一時的にデータを保存する揮発性メモリと、

持続的にデータを保存する不揮発性メモリと、を備え、

前記不揮発性メモリは、通常時に動作する通常ファームウェアと、この通常ファームウェア書き換え失敗時を含む緊急時に動作する緊急用ファームウェアと、を保持し、

前記緊急用ファームウェアは、前記通常ファームウェアよりサイズが小さく、前記更新プログラムを実行できる状態にする機能を有し、

前記不揮発性メモリは、更新前のファイルを一時的に格納するための一時退避領域と、更新ファイルを格納する更新データ領域と、を有し、

前記演算処理部は、前記通常ファームウェアの更新処理をする場合には、前記更新処理の前に次のブート先を前記緊急用ファームウェアに設定して前記更新処理を実行し、前記更新処理の後に次回ブート先を前記通常ファームウェアに設定し、

前記演算処理部は、ファイルの更新処理を実行する前に、更新前のファイルを前記一時退避領域に格納し、前記更新処理を実行した後、前記一時退避領域に格納した前記更新前のファイルを前記一時退避領域から削除すると共に、対応する前記更新ファイルと入れ替

10

20

えて前記更新データ領域に格納することを特徴とするファームウェア更新装置。

【請求項 2】

前記不揮発性メモリは、更新手順を記載した更新手順記載領域と、前記通常ファームウェアを構成する各ファイルについての更新状況を記載した更新状況記載領域と、を有し、前記演算処理部は、前記更新手順に従って前記更新プログラムを実行し、更新のログを前記更新状況記載領域に記述することを特徴とする請求項 1 に記載のファームウェア更新装置。

【請求項 3】

前記更新手順は、前記通常ファームウェアを構成する各ファイルについて、更新処理が完了した場合に再起動が必要か否かについての情報を含み、

前記演算処理部は、前記更新処理終了後に、再起動が必要とされるファイルを再起動することを特徴とする請求項 2 に記載のファームウェア更新装置。

【請求項 4】

前記演算処理部は、前記緊急用ファームウェアを起動した場合に、前記更新処理において、前記通常ファームウェアのブートに必要なファイルの破損の有無をチェックし、前記通常ファームウェアを構成するファイルの更新を実行した後に、次回ブート先を前記通常ファームウェアに設定することを特徴とする請求項 1 に記載のファームウェア更新装置。

【請求項 5】

ファームウェアを組込んだ組込み装置において、前記ファームウェアを更新するファームウェア更新方法であって、

前記組込み装置は、更新プログラムに従って、ファイル単位でファームウェアの更新処理を実行する演算処理部と、一時的にデータを保存する揮発性メモリと、持続的にデータを保存する不揮発性メモリと、を備え、

前記不揮発性メモリは、通常時に動作する通常ファームウェアと、この通常ファームウェア書き換え失敗時を含む緊急時に動作する緊急用ファームウェアと、を保持し、

前記緊急用ファームウェアは、前記通常ファームウェアよりサイズが小さく、前記更新プログラムを実行できる状態にする機能を有し、

前記不揮発性メモリは、更新前のファイルを一時的に格納するための一時退避領域と、更新ファイルを格納する更新データ領域と、を有し、

前記演算処理部が、前記通常ファームウェアの更新処理をする場合には、前記更新処理の前に次回のブート先を前記緊急用ファームウェアに設定して前記更新処理を実行し、前記更新処理の後に次回ブート先を前記通常ファームウェアに設定するステップと、

前記演算処理部が、ファイルの更新処理を実行する前に、更新前のファイルを前記一時退避領域に格納し、前記更新処理を実行した後、前記一時退避領域に格納した前記更新前のファイルを前記一時退避領域から削除すると共に、対応する前記更新ファイルと入れ替えて前記更新データ領域に格納するステップと、

を備えることを特徴とするファームウェア更新方法。

【請求項 6】

前記不揮発性メモリは、更新手順を記載した更新手順記載領域と、前記通常ファームウェアを構成する各ファイルについての更新状況を記載した更新状況記載領域と、を有し、

前記演算処理部が、前記更新手順に従って前記更新プログラムを実行し、更新のログを前記更新状況記載領域に記述するステップを備えることを特徴とする請求項 5 に記載のファームウェア更新方法。

【請求項 7】

前記更新手順は、前記通常ファームウェアを構成する各ファイルについて、更新処理が完了した場合に再起動が必要か否かについての情報を含み、

前記演算処理部が、前記更新処理終了後に、再起動が必要とされるファイルを再起動するステップを備えることを特徴とする請求項 6 に記載のファームウェア更新方法。

【請求項 8】

前記演算処理部が、前記緊急用ファームウェアを起動した場合に、前記更新処理におい

10

20

30

40

50

て、前記通常ファームウェアのブートに必要なファイルの破損の有無をチェックし、前記通常ファームウェアを構成するファイルの更新を実行した後に、次回ブート先を前記通常ファームウェアに設定するステップを備えることを特徴とする請求項5に記載のファームウェア更新方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ファームウェア更新装置及び方法に関する。

【背景技術】

【0002】

組込み装置をリリースした後、組み込み装置の不揮発性記憶メモリにあらかじめ記憶されている全てのデータ、例えばOSやアプリケーション（以下、「ファームウェア」という）に関して不具合が見つかった場合、当該ファームウェアを不具合を解決したファームウェアに書き換えることによって、不具合を解決するという方法が従来から一般に行われている。

10

【0003】

例えば、特許文献1には、新しいファームウェアと現在のファームウェアの2つのバージョンを保有するという2面構成を採用するシステムが開示されている。ここでは、新しいファームウェアと現在のファームウェアは外部のコンソールにより、手動で切り替えることができるようになっている。特許文献1によれば、更新途中で電源が落ちたり、想定外の操作をしてしまうなど更新中不具合があり、更新が途中で終了してしまった場合でも、外部のコンソールから更新を初めからやり直すことで、更新を完了させることができる。また、更新状況を管理しておけば、初めからではなく不具合発生時点から更新を再開することも可能である。さらに、新しいファームウェアから前のバージョンのファームウェアに切り替えることで、システムを起動することができる。つまり、更新中不具合が発生しても更新を再開できたり、前のバージョンに戻すことができるという利点がある。

20

【0004】

【特許文献1】特開平11-110218号公報

【発明の開示】

【発明が解決しようとする課題】

30

【0005】

しかしながら、特許文献1に開示されたファームウェア更新装置は2面構成を採用しているため、バックアップを持たないものに比べてファームウェア容量が2倍要り、装置の開発コストが高くなるという欠点がある。また、ブートができない状態に陥った場合、それを直すために、必ず外部コンソールが必要であり、一般ユーザが自分で簡単に修正することができない。

【0006】

さらに、該装置では、装置を停止させてファームウェアの更新を行うため、更新後の起動処理を行う必要がある。装置を停止させずにファームウェアを書き換える他のファームウェア更新装置でも、書き換えた後、必ず装置の再起動を必要とする。このように常に再起動を必要とするとファームウェア更新処理に時間が掛かってしまう。

40

【0007】

本発明はこのような状況に鑑みてなされたものであり、更新中不具合時に更新の再開が可能であり、前のバージョンに戻すことができるという従来の2面構成と同等の耐障害性を有しつつも、ファームウェア保存用の不揮発性メモリを節約でき、ブート失敗によるシステム停止を防ぐことができ、かつ必ずしも再起動を必要としないファームウェア更新処理を提供するものである。

【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、本発明は、ファームウェアを有し、このファームウェアを

50

更新する機能を備えるファームウェア更新装置であって、更新プログラムに従って、ファイル単位でファームウェアの更新処理を実行する演算処理部と、一時的にデータを保存する揮発性メモリと、持続的にデータを保存する不揮発性メモリと、を備える。そして、不揮発性メモリは、通常時に動作する通常ファームウェアと、この通常ファームウェア書き換え失敗時を含む緊急時に動作する緊急用ファームウェアと、を保持する。ここで、緊急用ファームウェアは、通常ファームウェアよりサイズが小さく、更新プログラムを実行できる状態にする機能のみを有する。そして、演算処理部は、通常ファームウェアの更新処理をする場合には、更新処理の前に次のブート先を緊急用ファームウェアに設定して更新処理を実行し、更新処理の後に次回ブート先を通常ファームウェアに設定するようにしている。

10

**【0009】**

また、不揮発メモリは、更新前のファイルを一時的に格納するための一時退避領域と、更新ファイルを格納する更新データ領域と、を有する。この場合、演算処理部は、更新前のファイルを一時退避領域に格納し、更新処理を実行した後、一時退避領域に格納した更新前のファイルを一時退避領域から削除すると共に、対応する更新ファイルと入れ替えて更新データ領域に格納するようにする。なお、一時退避領域の容量は、通常ファームウェアを構成する複数のファイルのうち最大サイズのファイルの容量に設定されている。

**【0010】**

さらに、不揮発性メモリは、更新手順を記載した更新手順記載領域と、通常ファームウェアを構成する各ファイルについての更新状況を記載した更新状況記載領域と、を有している。この場合、演算処理部は、更新手順に従って更新プログラムを実行し、更新のログを更新状況記載領域に記述する。ここで、更新手順は、さらに、通常ファームウェアを構成する各ファイルについて、更新処理が完了した場合に再起動が必要か否かについての情報を含んでいる。この場合、演算処理部は、更新処理終了後に、再起動が必要とされるファイルについて再起動する。

20

**【0011】**

演算処理部は、緊急用ファームウェアを起動した場合に、通常ファームウェアのブートに必要なファイルの破損の有無をチェックし、更新処理を実行した後に、次回ブート先を通常ファームウェアに設定する。

**【0012】**

また、演算処理部は、緊急用ファームウェアを起動したとき、通常ファームウェアが起動可能か否かを判断し、起動可能な場合には、次回ブート先を通常ファームウェアに設定する。一方、起動可能でない場合には、一時退避領域に格納された更新前のファイルを用いて前記通常ファームウェアを修復して起動可能な状態にしてから、次回ブート先を前記通常ファームウェアに設定する。

30

**【0013】**

さらなる本発明の特徴は、以下本発明を実施するための最良の形態および添付図面によって明らかになるものである。

**【発明の効果】****【0014】**

本発明によれば、ファームウェア更新において、ファームウェア保存用の不揮発性メモリの容量を節約しながらも、更新中不具合時の更新再開と前のバージョンへの戻すことができるという従来の2面構成と同等の信頼性を保つことができる。また、ブート失敗によるシステム停止を防ぐことができる。また、再起動によるシステム停止時間を必要最小限に留めることができる。

40

**【発明を実施するための最良の形態】****【0015】**

本発明は、ファームウェアを格納するフラッシュメモリの容量を削減しながらも、更新中の不具合にも対応できるなどの耐障害性を有するファームウェア更新装置に関するものである。

50

## 【 0 0 1 6 】

以下、添付図面を参照して本発明の実施形態について説明する。ただし、本実施形態は本発明を実現するための一例に過ぎず、本発明の技術的範囲を限定するものではないことに注意すべきである。

## 【 0 0 1 7 】

< 組込み装置（ファームウェア更新装置）の構成 >

図 1 は本発明の実施形態による組込み装置の概略構成を示す図である。本発明において、この組込み装置として想定しているものは例えば T V などの家電機器である。

## 【 0 0 1 8 】

組込み装置 1 は、M P U ( C P U ) 1 1 と、R A M 1 2 と、F R O M 1 3 と、入力装置 1 4 と、表示装置 1 5 と、通信装置 1 6 と、それぞれを接続するバス 1 7 と、で構成されている。

10

## 【 0 0 1 9 】

F R O M 1 3 はファームウェアを保持している不揮発性メモリである。F R O M 1 3 は、組込み装置 1 の起動時に最初に行われるブートブロック 1 3 0 1 と、組込み装置 1 に不具合のない通常状態で利用される通常ファームウェアブロック 1 3 0 2 と、組込み装置 1 に不具合があった場合に緊急に利用される緊急用ファームウェアブロック 1 3 0 3 と、ファームウェア更新時に利用される更新プログラム保存ブロック 1 3 0 4 と、ファームウェア更新時に利用される更新データ保存ブロック 1 3 0 5 と、更新ファイルを一時的に格納する更新ファイル一時退避ブロック 1 3 0 6 と、によって構成されている。

20

## 【 0 0 2 0 】

ブートブロック 1 3 0 1 には、本装置起動時最初に行われる、起動時に設定（指定）したファームウェアから起動させるためのソフトウェアであるブートローダ 1 3 0 1 A が保存されている。ブートローダは、起動後、通常ファームウェアブロック 1 3 0 2 もしくは緊急用ファームウェアブロック 1 3 0 3 のいずれかを読み込む。ブートローダ 1 3 0 1 A には、通常ファームウェアブロック 1 3 0 2 及び緊急用ファームウェアブロック 1 3 0 3 のいずれから起動するか、あらかじめ次回ブート先が M P U 1 1 によって登録される。

## 【 0 0 2 1 】

通常ファームウェアブロック 1 3 0 2 には、通常基盤ソフトウェア（O S ） 1 3 0 2 A と通常応用ソフトウェア 1 3 0 2 B が保存されている。

30

## 【 0 0 2 2 】

緊急用ファームウェアブロック 1 3 0 3 には、最低限の機能のみ有し、通常基盤ソフトウェア 1 3 0 2 A に比べてサイズを小さく抑えた緊急用基盤ソフトウェア 1 3 0 3 A が保存される。つまり、緊急用基盤ソフトウェア 1 3 0 3 A は、更新プログラム保存ブロック 1 3 0 4 と更新データ保存ブロック 1 3 0 5 と通常ファームウェアブロック 1 3 0 2 とを R A M にマウントして呼び出すだけの機能、即ち更新したプログラムを走らせることができる状態にする機能のみを備えている。従来とは異なり、通常基盤ソフトウェアと同じものを備えている必要はない。

## 【 0 0 2 3 】

更新プログラム保存ブロック 1 3 0 4 には、通常基盤ソフトウェア 1 3 0 2 A と通常応用ソフトウェア 1 3 0 2 B を更新する際に用いる更新プログラムが保存されている。

40

## 【 0 0 2 4 】

更新データ保存ブロック 1 3 0 5 には、新しいファームウェアに更新するための、現在のファームウェアと新しいファームウェアのファイル単位の差分データである更新データ 1 3 0 5 A と、更新手順（どのような手順及び操作で更新を実行するか）を記録する更新手順 1 3 0 5 B と、実行中の更新手順を指定する更新手順格納パスと該更新手順の中のどの部分まで終了しているかを表す更新手順実行終了位置とを記録するための更新状況（l o g f i l e ） 1 3 0 5 C と、が保存されている。従来ではイメージ（バイナリデータの塊）単位で更新していたのに対し、本発明ではファイル単位での更新を行っている。

## 【 0 0 2 5 】

50

< 通常ファームウェアの動作 >

図2は、本発明の実施形態による通常ファームウェアブロック1302の起動直後の動作を説明するためのフローチャートである。

【0026】

最初に、MPU11は、ファームウェア更新処理が途中であるか否かについて判定する(ステップS201)。これは、更新状況1305Cの中の更新手順格納パスがNULLか否かで判定する。更新途中である場合はNULLではなく、更新途中でない場合はNULLとなっている。更新途中と判定された場合、処理はステップ207に進み、更新途中でないと判定された場合、処理はステップS202に移行する。

【0027】

ステップS202において、MPU11は、ファームウェア更新を行うかどうか判定する。例えば、MPU11が入力装置14と表示装置15を利用して、組込み装置1のユーザに問い合わせるか、もしくは更新データ1305Aに更新を新規に行うのか、既に更新が正常終了したのかを示すフラグを保有しておき、それを参照するようにする。更新しない場合には、処理はステップS203に移行し、MPU11は通常の起動手順を実行し(ステップ203)、処理を終了する。

【0028】

ファームウェアを更新する場合には、MPU11は、その更新が順方向か逆方向かを判定する(ステップ204)。これは例えば、MPU11が、組込み装置1のユーザに入力装置14と表示装置15を利用して問い合わせることにより実現される。順方向である場合、処理はステップS205に移行し、MPU11が更新状況1305Cに更新手順書1305Bの格納位置を設定する(ステップS205)。一方、逆方向である場合、処理はステップS206に移行し、MPU11が更新手順書1305Bを元に逆方向の更新手順書を生成し、更新状況1305Cに更新手順書1305Bの格納位置を設定する(ステップS206)。

【0029】

続いて、MPU11は、ブートローダに対し、次回ブート先を緊急用ファームウェアブロックに設定する(ステップ207)。これは、ファームウェアが更新途中になっているので、その先の更新を可能にするための処理である。例えば、通常基盤ソフトウェア1302Aを書き換えている途中で電源が落ちてしまったような場合、ユーザは通常基盤ソフトウェアを起動することできず、メーカーが回収しなければならなくなるからである。よって、このような場合には、緊急用ファームウェアを用いて更新プログラムを実行するようにしている。次回立ち上がる時は緊急用ファームウェアブロックから立ち上がるので、処理としては図3(a)又は(b)の処理が実行されることになる。

【0030】

そして、MPU11は、ファームウェアの更新処理を行進手順書に基づいて実行する(ステップS208)。更新処理の詳細は図4でさらに説明する。

【0031】

次に、MPU11は、ブートローダに対し、次回ブート先を通常ファームウェアブロックに設定する(ステップS209)。

【0032】

さらに、MPU11は、更新したファイルについて再起動が必要かどうかを更新手順書で判定する(ステップS210)。必要な場合は、MPU11は、更新手順書に従って必要なコンポーネントに対し再起動し(ステップS211)、終了する。2面構成の場合には更新した場合には必ず再起動が必要であったが、本発明の場合には正常に更新が完了すれば再起動は不要であり、異常(更新途中で電源が落ちた等)があった場合に再起動が必要となるだけである(OSの書き換えに関しては書き換えの対象によっては正常に更新が終了しても再起動が必要な場合がある)。

【0033】

なお、更新手順書は、例えば、更新前の現バージョン、更新後の新バージョン、更新内

10

20

30

40

50

容からなり、更新内容は対象ファイルごとに操作内容、対象ファイル、再起動が必要な対象を表す再起動対象からなる。更新対象とは例えば、サービス単位での更新で済む場合はそのサービスを指定し、システム全体を再起動する必要がある場合は、システム全体を指定する。ステップS 2 0 6の逆方向の更新手順書の生成について、例えば、ファイルの生成については削除、削除については生成に変更する。

【 0 0 3 4 】

< 緊急用ファームウェアブロックの動作 >

緊急用ファームウェアブロックの起動後の動作としては、例えば、図3 ( a ) 及び ( b ) に示されるように2種類考えられる。

【 0 0 3 5 】

1) 図3 ( a ) は、緊急用ファームウェアブロックの起動直後の動作であり、通常ファームウェアが少なくとも修正すれば起動可能であった場合の動作を説明するためのフローチャートである。

【 0 0 3 6 】

最初に、MPU 1 1 は、ファームウェア更新処理が途中であるか否かについて判定する (ステップS 3 0 1)。これは、前述と同様に、更新状況1 3 0 5 Cの中の更新手順格納パスがNULLか否かで判定する。更新途中である場合はNULLではなく、更新途中でない場合はNULLである。更新途中と判定された場合、処理はステップS 3 0 3に進む。更新途中でないとして判定された場合、処理はステップS 3 0 2に移行し、MPU 1 1 は通常の起動手順を実行し (ステップ3 0 2)、処理を終了する。

【 0 0 3 7 】

ステップS 3 0 3において、MPU 1 1 は、通常ファームウェアブロックをRAM 1 2にマウントする。図3 ( a ) は通常ファームウェアから起動する場合に係るからである。

【 0 0 3 8 】

そして、MPU 1 1 は、通常ファームウェアブロックに欠陥があるか否か検査し、そのまま起動可能か判定する (ステップS 3 0 4)。起動できないと判断された場合、処理はステップS 3 0 5に移行し、MPU 1 1 は通常ファームウェアブロックを更新前の状態に戻して、起動可能にする (ステップS 3 0 5)。例えば、更新前のファイルを更新ファイル一時退避ブロック1 3 0 6に格納した後、更新処理している途中で電源が落ちてしまったような場合には、この更新ファイル一時退避ブロック1 3 0 6から更新前のファイルを取り出して更新前の状態に戻す、といった処理である。一方、起動可能と判断された場合、処理はステップS 3 0 6に移行する。

【 0 0 3 9 】

続いて、MPU 1 1 は、次回ブート先を通常ファームウェアに設定し (ステップS 3 0 6)、再起動する (ステップS 3 0 7)。緊急用ファームウェアから立ち上がった場合には必ず再起動することになる。

【 0 0 4 0 】

2) 図3 ( b ) は、緊急用ファームウェアブロックの起動直後の動作であり、通常ファームウェアブロックが起動可能でない場合の動作を説明するためのフローチャートである。図3 ( b ) の方が基本となる動作を示している。

【 0 0 4 1 】

最初に、MPU 1 1 は、ファームウェア更新処理が途中であるか否かについて判定する (ステップ3 1 1)。更新途中であるか否かの判断については上述した通りである。更新途中と判定された場合、処理はステップS 3 1 3に進む。更新途中でないとして判定された場合は、MPU 1 1 は、通常の起動手順を実行し (ステップ3 1 2)、処理を終了する。

【 0 0 4 2 】

ステップS 3 1 3では、MPU 1 1 は、通常ファームウェアブロックをRAM 1 2にマウントする。

【 0 0 4 3 】

そして、MPU 1 1 は、更新処理を実行する (ステップS 3 1 4)。また、MPU 1 1

10

20

30

40

50

は、次回ブート先を通常ファームウェアに設定し（ステップS315）、再起動する（ステップS316）。

【0044】

なお、ステップS314では、ブートに必要なファイルの破損状況を確認することでブート可能かどうかを判定しているが、その代わりもしくはそれに加え、緊急用ファームウェア領域内に備えた、OS上で別のOSを動作させることのできる仮想化環境により、通常ファームウェアの起動が可能であることを実際に確かめても良い。

【0045】

<ファームウェアの更新処理>

図4は、ファームウェアの更新処理の詳細について説明するためのフローチャートである。

10

【0046】

まず、MPU11は、前回の更新処理において破損ファイルがあれば破棄する（ステップS401）。これは、例えば、更新途中で電源が落ちてしまったような場合には更新処理によって生成されたファイルに破損ファイル（途中までしか更新されていないファイル）が含まれている場合がある。このような破損ファイルを破棄するのがステップS401の処理である。ファイル単位で更新処理が行われ、どのファイルまで更新処理が完了しているかは更新状況1305Cを見れば分かるので更新途中のファイルが検出できる。

【0047】

次に、MPU11は、更新手順1305Bと更新状況1305Cを基に更新開始位置を判定し（ステップS402）、更新対象ファイルの複製を更新ファイル一時退避ブロック1306に退避する（ステップS403）。

20

【0048】

そして、MPU11は、更新手順1305Bに従って、更新対象ファイルを更新する（ステップS404）。

【0049】

続いて、MPU11は、更新データ1305Aの中の該当更新ファイル（更新ファイルA、B、・・・）を削除し、代わりに更新ファイル一時退避ブロック1306に格納されていた退避ファイル（更新前のファイル）を更新データ1305Aに移動する（ステップ405）。従来は単純にバックアップ領域が存在し、該バックアップ領域に更新前のファイルを格納していたが、本発明でステップS405のようにすることで、ファームウェア容量を節約することができる。なお、更新ファイル一時退避ブロック1306の容量は、あらかじめ組み込み装置1の最大ファイルサイズに決定されている。

30

【0050】

さらに、MPU11は、ステップS404の更新処理中に不具合があったかどうか判定をし（ステップ406）、不具合があったと判断された場合、処理はステップS407に移行し、不具合がなかったと判断された場合、処理はステップS408に移行する。

【0051】

ステップS407では、MPU11は、基盤ソフトウェア（通常又は緊急基盤ソフトウェア）を再起動し、図2又は図3の処理を最初から開始する。一方、ステップS408では、MPU11は、全てのファイルについて更新処理が完了したか否か調べる。何れかのファイルについて更新済みで無い場合は、処理はステップS403に戻り、全ファイル更新済みである場合は、処理はステップS409に移行する。ステップS409では、MPU11は、更新状況1305Cを初期化し、更新処理が終了する。更新処理が終了した場合には、処理は、図2のステップS209又は図3（b）のステップS315に移行する。

40

【0052】

なお、ステップS403からステップS405の処理に関しては、更新ファイル一時退避ブロック1306に更新後のファイルを格納し、その後、更新データ中の該当ファイルを削除し、代わりに更新前のファイルを更新データ保存領域へ移動し、更新後ファイルを

50



通常ファームウェアブロックに移動しても良い。

【 0 0 5 3 】

<まとめ>

本発明の実施形態では、不揮発性メモリ内に、ファームウェアを読み込むためのブートローダ、通常動作する通常ファームウェアと通常ファームウェア書き換え失敗時など緊急時に動作する緊急用ファームウェア、両ファームウェアからアクセス可能な更新プログラム、通常ファームウェアのうち変更のあるファイルのみで構成される更新データ、更新方法を記した更新手順、更新途中経過を記録する更新状況、ファイル更新中にその前のファイルまたは更新後のファイルを一時的に格納する更新ファイル一時退避領域を保有する。そして、更新データを通常ファームウェアのうち変更のあるファイルのみで構成すること  
10  
と、更新ファイル一時退避領域をあらかじめ決めた最大ファイルサイズの容量にすること、更新前のファイルを該ファイル更新後に更新後ファイルが占有していた領域に格納することで、不揮発性メモリ容量を小さく抑える。

【 0 0 5 4 】

また、通常ファームウェアと緊急用の両方から更新に必要なデータにアクセス可能な構成と、更新の直前に次回ブート先を緊急用ファームウェアに設定し、更新終了直後に次回ブート先を通常ファームウェアに設定する。これにより、更新中の不具合発生時に、不具合発生時点から更新を再開することができ、退避した更新前ファイルと、更新手順から操作内容を反転させることで自動生成した新たな更新手順によって、更新前の状態に戻すこと  
20  
ができるようになっている。

【 0 0 5 5 】

さらに、緊急用ファームウェア起動中に通常ファームウェアのブートに必要なファイルの破損がないかを検査するようにしている。これにより、その後の通常ファームウェア起動時に起動失敗に陥りシステムが停止してしまうことを回避することができる。

【 0 0 5 6 】

また、緊急用ファームウェア起動中に緊急用ファームウェア領域内に備えた、OS内で別のOSを動作させることのできる仮想化環境を用意し、通常ファームウェアの起動が可能であることを実際に確かめる。これにより、その後の通常ファームウェア起動時に起動失敗に陥りシステムが停止してしまうことを回避することができるようになる。

【 0 0 5 7 】

また、稼働中の通常ファームウェアから書き換え、かつ、更新手順に、更新ファイルごとに必要な再起動の対象として、システム全体またはあるサービスまたは一切再起動が必要がないことを記述しておく。これにより、システムの停止時間を必要最小限に抑えることができるようになる。

【 0 0 5 8 】

なお、本発明は、実施形態の機能を実現するソフトウェアのプログラムコードによっても実現できる。この場合、プログラムコードを記録した記憶媒体をシステム或は装置に提供し、そのシステム或は装置のコンピュータ（又はCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出す。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコード自体、及びそれを記憶した記憶媒体は本発明を構成することになる。このようなプログラムコードを供給するための記憶媒体としては、例えば、フレキシブルディスク、CD-ROM、DVD-ROM、ハードディスク、光ディスク、光磁気ディスク、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどが用いられる。

【 0 0 5 9 】

また、プログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部又は全部を行い、その処理によって前述した実施の形態の機能が実現されるようにしてもよい。さらに、記憶媒体から読み出されたプログラムコードが、コンピュータ上のメモリに書き込まれた後、そのプログラムコードの指示に基づき、コンピュータのCPUなどが実際の処理の一部又は全部を行い、その処  
40  
50

理によって前述した実施の形態の機能が実現されるようにしてもよい。

【0060】

また、実施の形態の機能を実現するソフトウェアのプログラムコードを、ネットワークを介して配信することにより、それをシステム又は装置のハードディスクやメモリ等の記憶手段又はCD-RW、CD-R等の記憶媒体に格納し、使用時にそのシステム又は装置のコンピュータ(又はCPUやMPU)が当該記憶手段や当該記憶媒体に格納されたプログラムコードを読み出して実行するようにしても良い。

【図面の簡単な説明】

【0061】

【図1】本発明の実施形態における組込み装置の概略構成を示す図である。

10

【図2】通常ファームウェアブロックの起動直後の動作を説明するためのフローチャートである。

【図3a】緊急用ファームウェアブロックの起動直後の動作を説明するためのフローチャート(1)である。

【図3b】緊急用ファームウェアブロックの起動直後の動作を説明するためのフローチャート(2)である。

【図4】ファームウェアの更新処理の詳細動作を説明するためのフローチャートである。

【符号の説明】

【0062】

1・・・組込み装置

20

11・・・MPU

12・・・RAM

13・・・FROM

14・・・入力装置

15・・・表示装置

16・・・通信装置

1301・・・ブートブロック

1301・・・通常ファームウェアブロック

1302・・・更新ファームウェアブロック

1303・・・緊急用ファームウェアブロック

30

1304・・・更新プログラム保存ブロック

1305・・・更新データ保存ブロック

1306・・・更新ファイル一時退避ブロック

1301A・・・ブートローダ

1302A・・・通常基盤ソフトウェア

1302B・・・通常応用ソフトウェア

1303A・・・緊急用基盤ソフトウェア

1304A・・・更新プログラム

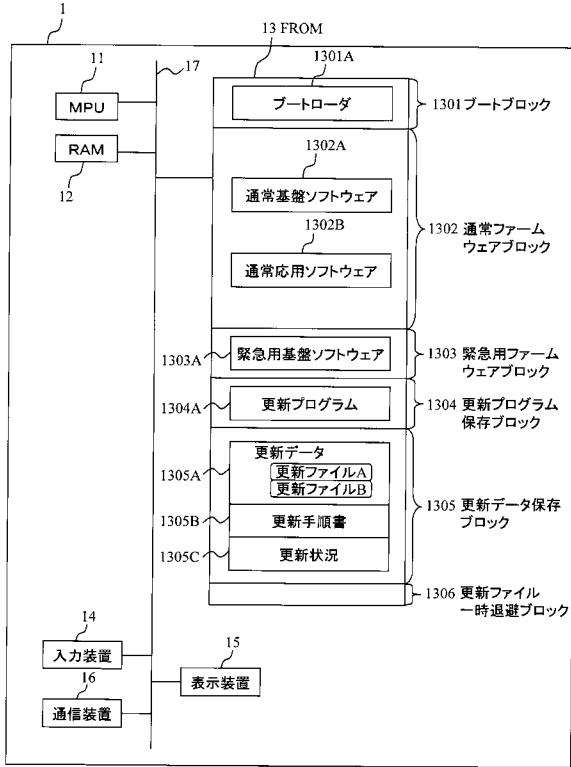
1305A・・・更新データ

1305B・・・更新手順書

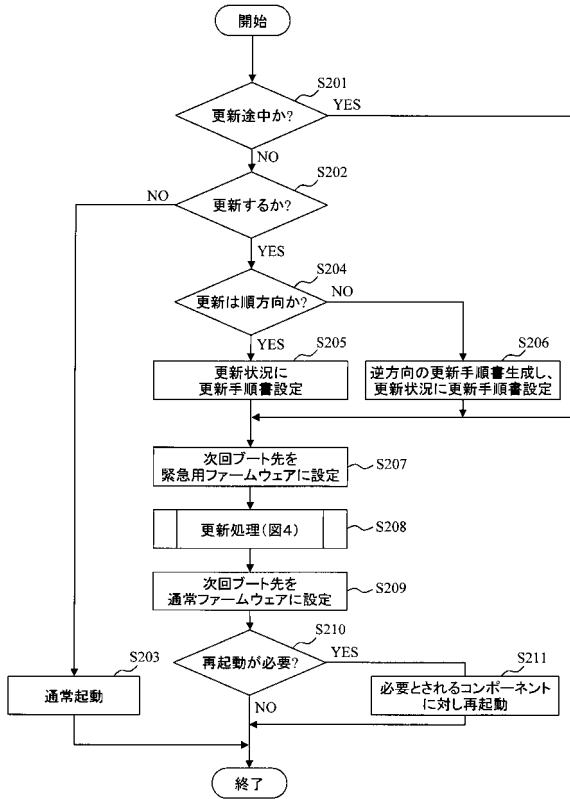
40

1305C・・・更新状況

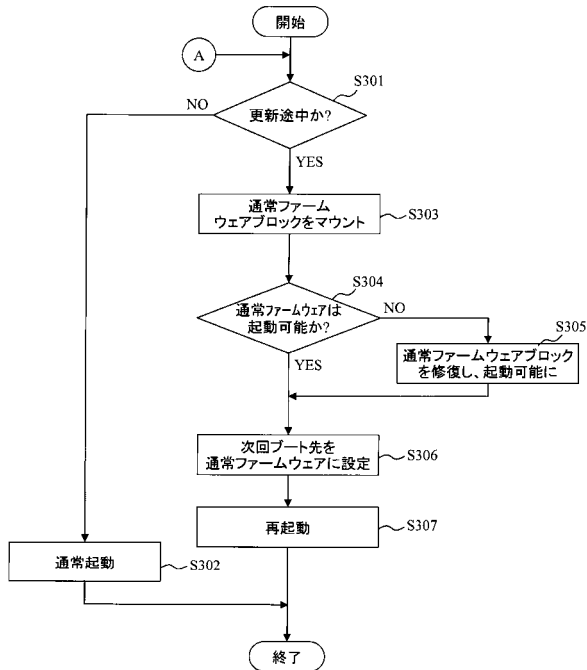
【図1】



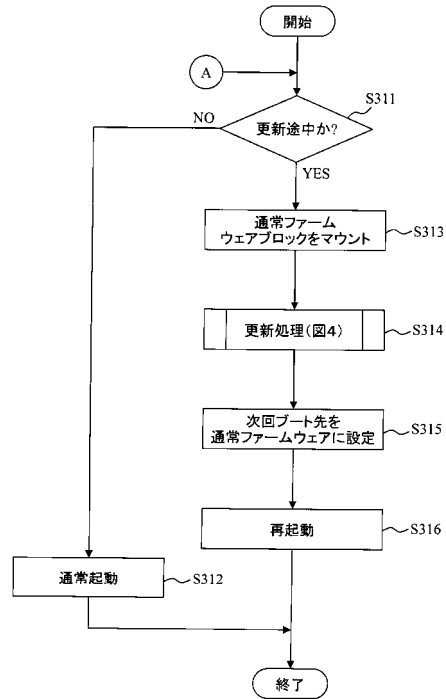
【図2】



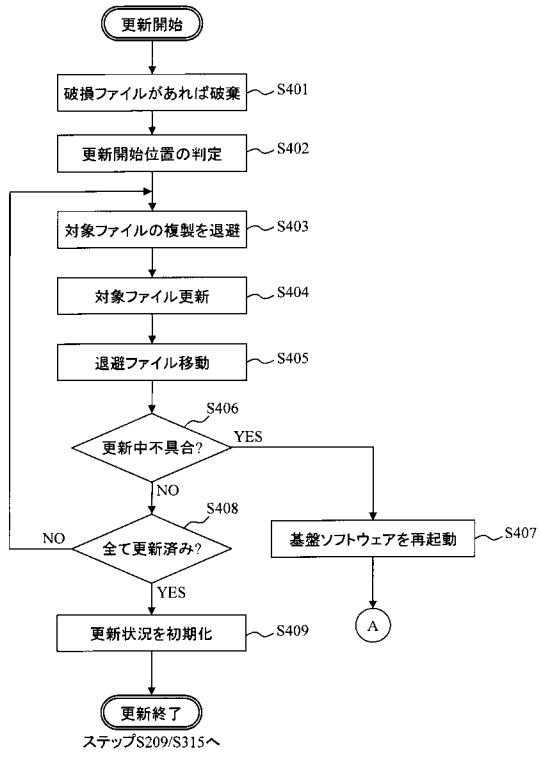
【図3 a】



【図3 b】



【図4】



---

フロントページの続き

(72)発明者 中村 雄一

東京都品川区東品川四丁目12番7号 日立ソフトウェアエンジニアリング株式会社内

審査官 林 毅

(56)参考文献 特開2000-187588(JP,A)

特開2008-084304(JP,A)

特開2001-331379(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 11/00