

(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2012-198929

(P2012-198929A)

(43) 公開日 平成24年10月18日(2012.10.18)

(51) Int.Cl.			F I	テーマコード (参考)		
<b>G06F</b>	<b>9/445</b>	<b>(2006.01)</b>	G06F 9/06	610K	5B027	
<b>G06F</b>	<b>11/14</b>	<b>(2006.01)</b>	G06F 11/14	310J	5B376	
<b>G06F</b>	<b>11/00</b>	<b>(2006.01)</b>	G06F 9/06	630B		

審査請求 有 請求項の数 1 O L (全 20 頁)

(21) 出願番号 特願2012-137193 (P2012-137193)  
 (22) 出願日 平成24年6月18日 (2012. 6. 18)  
 (62) 分割の表示 特願2009-40291 (P2009-40291) の分割  
 原出願日 平成21年2月24日 (2009. 2. 24)

(71) 出願人 000003207  
 トヨタ自動車株式会社  
 愛知県豊田市トヨタ町1番地  
 (74) 代理人 100100549  
 弁理士 川口 嘉之  
 (74) 代理人 100085006  
 弁理士 世良 和信  
 (74) 代理人 100113608  
 弁理士 平川 明  
 (74) 代理人 100123319  
 弁理士 関根 武彦  
 (74) 代理人 100123098  
 弁理士 今堀 克彦  
 (74) 代理人 100143797  
 弁理士 宮下 文徳

最終頁に続く

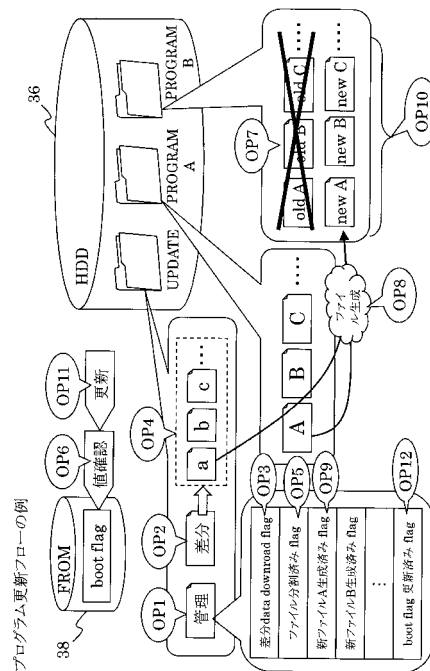
(54) 【発明の名称】 情報処理装置

(57) 【要約】

【課題】プログラムの更新処理に失敗した場合でも、正常時の動作に復帰することができるプログラム更新方法を提供する。

【解決手段】工場出荷時のバージョンのプログラムと第1のバージョンのプログラムとを格納した補助記憶装置と、第1のバージョンのプログラムを保持する主記憶装置と、電源投入時に主記憶装置内の第1のバージョンのプログラムを起動するCPUと、を備える装置であって、CPUは、主記憶装置内の第1のバージョンのプログラムにエラーがある場合、及び、主記憶装置内の第1のバージョンのプログラムが所定回数以上動作を中断させていた場合には、補助記憶装置内の第1のバージョンのプログラムを主記憶装置にローディングし、主記憶装置内の第1のバージョンのプログラムが所定回数以上ローディングに失敗している場合には、補助記憶装置内の工場出荷時のバージョンのプログラムを主記憶装置にローディングする。

【選択図】 図2



**【特許請求の範囲】****【請求項 1】**

工場出荷時のバージョンのプログラムと前記工場出荷時のバージョンよりも新しいバージョンである第 1 のバージョンのプログラムとを格納した補助記憶装置と、

前記補助記憶装置からローディングされた前記第 1 のバージョンのプログラムを保持する主記憶装置と、

電源投入時に前記主記憶装置に保持される第 1 のバージョンのプログラムを起動する CPU と、を備える装置であって、

前記 CPU は、

前記主記憶装置に保持される第 1 のバージョンのプログラムにエラーがある場合、及び、前記主記憶装置に保持される第 1 のバージョンのプログラムが今回の電源投入時までの間に所定回数以上動作を中断させていた場合には、前記補助記憶装置に格納された前記第 1 のバージョンのプログラムを前記主記憶装置にローディングし、

前記主記憶装置に保持される第 1 のバージョンのプログラムが今回の電源投入時までの間に所定回数以上ローディングに失敗している場合には、前記補助記憶装置に格納された前記工場出荷時のバージョンのプログラムを前記主記憶装置にローディングする情報処理装置。

10

**【発明の詳細な説明】****【技術分野】****【0001】**

本発明は、プログラムの更新処理に関する。

20

**【背景技術】****【0002】**

カーナビゲーションシステムを搭載する車載器等は、例えば、道路の新設等に合わせて、地図情報等のアプリケーションプログラムを更新する必要がある。例えば、車載器のオペレーティングシステム及びアプリケーションプログラム等を含むプログラム（以下、まとめてプログラムと称する）を更新する方法として以下の 2 つが挙げられる。

**【0003】**

（1）専門家による更新処理の実施

車載器を保有するユーザがメーカへハードディスク（以下、HDD（Hard Disk Drive））又は車載器そのものを送り、メーカがプログラムの更新処理を実施する方法である。若しくは、自動車の販売業者に自動車を預けて、販売員がプログラムの更新処理を実施する方法である。

30

**【0004】**

この方法では、専門家がプログラムの更新処理を実施するので、プログラムの更新処理の手順、手段等がある程度自由に実施可能である。専門家による更新処理の実施における利点は、例えば、以下の点が挙げられる。

**【0005】**

（1）プログラムの個別の更新から、全プログラムの更新まで更新するプログラムの選択が自由に実施可能である。（2）プログラムの更新処理に使用する治具等の専用の接続機器を使用できる。また、機器を分解して記憶媒体を取り出すことも可能である。プログラムの更新に失敗してもやり直すことが可能である。（3）データのバックアップを取ることが可能である。（4）リカバリによって工場出荷時に戻すことも可能である。

40

**【0006】**

専門家によるプログラムの更新処理の実施の以上のような利点に対して、例えば、以下に示す欠点もある。

**【0007】**

（1）車載器、又は HDD をメーカに送らなければならない、もしくは、ディーラーへ出向いて自動車を預ける必要があるなどの、ユーザの手間がかかる。（2）車載器、HDD、又は自動車を預けている間は、ユーザはカーナビゲーションシステムの使用ができな

50

い。(3)専用の治具の使用、機器の分解等、プロでしかできない作業内容である。

【0008】

(2)ユーザによる更新処理の実施

CD-ROM (Compact Disk - Read Only Memory) や DVD-ROM (digital Versatile disc) 及びその他の記憶媒体に記憶された更新プログラムを利用して、ユーザ自身がプログラムの更新処理を実施する方法である。若しくは、携帯電話や車載器の通信機能等を使用して、外部との通信により更新プログラムをダウンロードして、ユーザ自身がプログラムの更新処理を実施する方法である。ユーザによる更新処理の実施における利点は、例えば、以下の点が挙げられる。

【0009】

(1)ユーザ自身で実施できるため、メーカへHDDを送ったり、販売店へ自動車を預けたりする必要がない。

【0010】

ユーザによるプログラムの更新処理の実施の以上のような利点に対して、例えば、以下に示す欠点もある。

【0011】

(2)車載器は復帰手段を持たないため、プログラムの更新処理に失敗した場合に、最悪の場合、起動が不可能になる事態に陥る可能性がある。(2)CD-ROMやDVD-ROM等を用いる場合、それらを手に入れる手間がかかる。(3)携帯電話や車載器の通信機能を用いて更新プログラムのデータをダウンロードする場合、パケット料金などの通信量が発生する。ダウンロードするデータが大きければ、通信料も高額になり、時間もかかる。

【0012】

ユーザによるプログラムの更新処理の方法において、重大な欠点は、プログラムの更新処理に失敗した場合に復帰できない可能性があるという点である。プログラムの更新処理が失敗する原因としては、例えば、プログラムの更新処理中の電源断(自動車のアクセサリ電源の切断等)などが考えられる。

【0013】

従って、ユーザによってプログラムの更新処理ができ、且つ、更新に失敗した場合に復帰手段を持つプログラム更新方法が望まれていた。

【先行技術文献】

【特許文献】

【0014】

【特許文献1】特開2006-301960号公報

【発明の概要】

【発明が解決しようとする課題】

【0015】

本発明は、プログラムの更新処理に失敗した場合でも、正常時の動作に復帰することができるプログラム更新方法を提供することを課題とする。

【課題を解決するための手段】

【0016】

本発明は前記課題を解決するために、以下の手段を採用した。すなわち、プログラム更新装置は、第1のバージョンのプログラムを格納する第1格納部と、

前記第1のバージョンと同じ又は前記第1のバージョンよりも新しいバージョンである第2のバージョンの前記プログラムを格納する第2格納部と、

前記第2のバージョンのプログラムと前記第2のバージョンのプログラムよりもバージョンが新しい第3のバージョンの前記プログラムとの差分を取得する取得部と、

前記第2格納部に格納される前記第2のバージョンのプログラムと前記取得部によって取得した差分とから前記第3のバージョンのプログラムを生成し、前記生成した第3のバージョンのプログラムを前記第1格納部に格納する更新部とを備える。

10

20

30

40

50

## 【 0 0 1 7 】

本発明によれば、第 1 格納部に格納される第 1 のバージョンのプログラムが第 3 のバージョンのプログラムに更新される一方で、第 2 格納部には、第 2 のバージョンのプログラムが保持され続ける。例えば、第 3 のバージョンのプログラムを生成するステップ、又は、第 3 のバージョンのプログラムを第 1 格納部に格納するステップにおいて、コンピュータの電源が断絶する場合でも、第 2 のバージョンのプログラムが保持され続けているので、第 2 のバージョンのプログラムを使用して、復帰することができる。

## 【 0 0 1 8 】

また、本発明の態様は、工場出荷時のバージョンのプログラムと前記工場出荷時のバージョンよりも新しいバージョンである第 1 のバージョンの前記プログラムとを格納した補助記憶装置と、

前記補助記憶装置からローディングされた前記第 1 のバージョンのプログラムを保持する主記憶装置と、

電源投入時に前記主記憶装置に保持される第 1 のバージョンのプログラムを起動する CPU と、を備える装置であって、

前記 CPU は、

前記主記憶装置に保持される第 1 のバージョンのプログラムにエラーがある場合、及び、前記主記憶装置に保持される第 1 のバージョンのプログラムが今回の電源投入時までの間に所定回数以上動作を中断させていた場合には、前記補助記憶装置に格納された前記第 1 のバージョンのプログラムを前記主記憶装置にローディングし、

前記主記憶装置に保持される第 1 のバージョンのプログラムが今回の電源投入時までの間に所定回数以上ローディングに失敗している場合には、前記補助記憶装置に格納された前記工場出荷時のバージョンのプログラムを前記主記憶装置にローディングする。

## 【 0 0 1 9 】

本発明によれば、主記憶装置にローディングされた第 1 のバージョンのプログラムデータに異常が発生している場合でも、補助記憶装置から再度第 1 のバージョンのプログラムのローディングを実行し正常な状態に復帰することができる。また、補助記憶装置からの第 1 のバージョンのプログラムのローディングが失敗した場合でも、工場出荷時のバージョンのプログラムを使用して装置を起動することができる。このようにして、段階的に処理を行うので、より効率的に復旧処理を行うことができる。また、最終的には、工場出荷時のバージョンのプログラムによって装置の起動が保証される。

## 【 発明の効果 】

## 【 0 0 2 0 】

本発明によれば、プログラムの更新処理に失敗した場合でも、正常時の動作に復帰することができるプログラム更新方法を提供することができる。

## 【 図面の簡単な説明 】

## 【 0 0 2 1 】

【 図 1 】 プログラム更新システムの構成例を示す図である。

【 図 2 】 プログラム更新システムのプログラム更新処理における動作例を示す図である。

【 図 3 A 】 プログラム更新処理中に電源断が発生した場合の復帰動作の例について示す図である。

【 図 3 B 】 プログラム更新処理中に電源断が発生した場合の復帰動作の例について示す図である。

【 図 4 】 FROM 内のブートフラグの値の例を示す図である。

【 図 5 】 車載器のリカバリデータの保持の例を示す図である。

【 図 6 】 正常時の起動動作の流れの例を示す図である。

【 図 7 】 起動シーケンスの例を示す図である。

【 図 8 】 異常時シーケンス A で実行される処理の流れの例を示す図である。

【 図 9 】 異常時シーケンス A の処理フローを示す図である。

【 図 1 0 】 異常時シーケンス B で実行される処理を示す図である。

【図 1 1】異常時シーケンス B の処理フローを示す図である。

【発明を実施するための形態】

【0022】

以下、図面に基づいて、本発明の実施の形態を説明する。以下の実施形態の構成は例示であり、本発明は実施形態の構成には限定されない。

【0023】

< 第 1 実施形態 >

本発明のプログラム更新方法は、機器の 1 つの記憶媒体に、同一のプログラム群を格納するフォルダ（以降、プログラムフォルダと称する）を 2 個保持する。プログラムの更新処理の際には、一方のプログラムフォルダ内のプログラム群を更新し、他方のプログラムフォルダ内のプログラム群はバックアップとしてそのまま保持する。

10

【0024】

2 個のプログラムフォルダについて交互にプログラム更新処理を実施することによって、一方のプログラムフォルダはプログラム群を更新して最新のバージョンのプログラム群を保持することができる。また、他方のプログラムフォルダは更新処理の直前まで使用していたバージョンのプログラム群を保持することができる。プログラムの更新処理の後に、最新のバージョンのプログラム（更新済みのプログラム）が動作しない場合には、プログラム更新処理を実施しなかった（バックアップとしての）プログラムフォルダを選択して、当該プログラムを起動することが可能となる。

20

【0025】

< プログラム更新システムの構成例 >

図 1 は、プログラム更新システムの構成例を示す図である。プログラム更新システムは、車載器 1 と、更新プログラムを保持する、例えば、メーカのサーバであるセンタ C 1 とを含む。

【0026】

センタ C 1 は、例えば、車載器 1 に搭載されているプログラムのメーカ等のサーバである。センタ C 1 は、プログラムの更新情報を発信する。センタ C 1 は、バージョン比較部 C 2 を備える。バージョン比較部 C 2 は、車載器 1 に搭載されているプログラムのバージョンとセンタ C 1 が保持する最新のプログラムのバージョンを比較して差分データを生成する。

30

【0027】

車載器 1 は、ユーザの自動車に搭載されているカーナビゲーションシステムを備える機器である。車載器 1 は、携帯電話 / D C M ( D a t a C o m m u n i c a t i o n M o d u l e ) 2 と、マスタ機器と 3 と、スレーブ機器 4 と、デッキ 5 とを備える。

【0028】

携帯電話 / D C M 2 は、外部のセンタ C 1 と無線通信を行うインターフェイスである。カーナビゲーションシステムを備える車載器では、携帯電話と（専用ケーブル等で）接続して、携帯電話の通信機能を用いてプログラムの差分データをダウンロードする機種もある。また、D C M と呼ばれる車載器専用の通信機を備える機種もある。図 1 では、センタ C 1 との通信インターフェイスとして、携帯電話とのインターフェイスと D C M とをまとめて携帯電話 / D C M 2 として表示している。

40

【0029】

デッキ 5 は、C D ( C o m p a c t D i s k )、又は、D V D ( D i g i t a l V e r s a t i l e D i s c ) などの外部記憶媒体からプログラムの更新データを読み出す機器である。

【0030】

マスタ機器 3 は、マイクロコンピュータ（以下、マイコンと称す）を少なくとも 1 つ備えており、スレーブ機器 4 を取りまとめる中心的役割を果たす、車載器 1 に 1 台のみ存在する機器である。マスタ機器 3 のような車載器専用の制御コンピュータを E C U ( E l e c t r o n i c C o n t r o l U n i t ) と呼ぶこともある。

50

## 【0031】

マスタ機器3は、通信部31と、ダウンロード選択部32と、バージョン情報収集部33と、プログラム更新部34と、スレーブ通信部35と、HDD(Hard Disk Drive)36と、DRAM(Dynamic Random Access Memory)37と、FROM(Flash Read Only Memory)38とを備える。

## 【0032】

通信部31(「取得部」に相当)は、携帯電話/DCM2又はデッキ5と通信を行う。通信部31は、携帯電話/DCM2又はデッキ5を通じて更新データを受信し、HDD36に格納する。

10

## 【0033】

ダウンロード選択部32は、携帯電話/DCM2又はデッキ5のどちらから更新データをダウンロードするかを選択する。ダウンロード選択部32は、例えば、デッキ5にCD又はDVDが挿入されたことを検知し、デッキ5から更新データのダウンロードを行うことを選択する。

## 【0034】

バージョン情報収集部33は、現在のマスタ機器3に搭載されるプログラムのバージョンを管理する。

## 【0035】

プログラム更新部34(「更新部」に相当)は、ダウンロードされた更新データから実際にプログラムの更新処理を実施する。プログラム更新部34の詳細については、後述する。

20

## 【0036】

スレーブ通信部35は、スレーブ機器4と接続し、通信を行う。マスタ機器3とスレーブ機器4とは、例えば、LAN(Local Area Network)配線やバス接続によって接続されている。

## 【0037】

HDD36は、車載器1上で実行されるプログラムや関連するデータ等を格納する補助記憶装置である。図1においては、マスタ機器3がHDD36を備える図を示している。HDD36は、マスタ機器2及びスレーブ機器4とは独立に車載器1内に備えられてもよい。

30

## 【0038】

DRAM37は、主記憶装置である。DRAM37は、HDD36に格納されているプログラムをローディングするロードプログラムを格納する。ロードプログラムは、あるプログラムを起動するときに、HDD36に格納されているプログラムデータをDRAM37上に展開する。

## 【0039】

FROM38は、不揮発性の記憶装置である。FROM38は、ブートプログラムとブートフラグとを格納する。ブートプログラムは車載器1が電源投入等により起動する際の起動プロセスを格納したプログラムである。ブートフラグは、プログラムをローディングするプログラムフォルダを示すフラグを保持する。詳細は、後述する。

40

## 【0040】

スレーブ機器4は、それぞれが独自の機能を果たす。スレーブ機器4には、例えば、デジタルテレビ装置、DVD再生装置等がある。スレーブ機器4は、車載器1の仕様によって、備えられる個数が変わる。

## 【0041】

スレーブ機器4の中でも、例えば、デジタルテレビ装置等のように扱うデータ量が多い場合には、1つのマイコンにかかる負荷が大きくなるため、マイコンを複数備えることがある。スレーブ機器4がマイコンを複数備える場合には、複数のマイコンの内の1つのマイコンを親マイコン、その他のマイコンを子マイコンというように分ける。子マイコンは

50

親マイコンを通じてマスタ機器から更新データの受信等を行う。

【0042】

スレーブ機器4は、共通して、機器通信部411、プログラム更新部412、バージョン情報収集部413を有する。マイコンを複数備えるスレーブ機器4では、親マイコンが機器通信部411、プログラム更新部412、バージョン情報収集部413を有する。スレーブ機器4がマイコンを複数備える場合には、親マイコン、子マイコンそれぞれが互いに通信するための通信部を備える。子マイコンは、親マイコンとの通信インターフェイスである通信部421とプログラム更新部422とを備える。

【0043】

機器通信部411は、マスタ機器3と接続するインターフェイスである。機器通信部411は、例えば、LAN配線やバス接続等でマスタ機器3と接続する。

【0044】

バージョン情報収集部413は、スレーブ機器4内の親マイコン及び子マイコンが使用するプログラムのバージョン情報を管理する。バージョン情報収集部413は、マスタ機器3がダウンロードしたプログラムの更新データとスレーブ機器が実行するプログラムのバージョンとを比較する。

【0045】

プログラム更新部412及びプログラム更新部422は、マスタ機器3によってダウンロードされた更新データから、スレーブ機器4の親マイコン及び子マイコンそれぞれのプログラム更新処理を行う。プログラム更新部412及びプログラム更新部422については、後述する。

【0046】

マスタ機器3の通信部31、ダウンロード選択部32、バージョン情報収集部33、プログラム更新部34、及びスレーブ通信部35は、マスタ機器3が備えるマイコンのCPU(Central Processing Unit)、及びそれぞれの機能部の専用の電子回路を含んだICチップ等によって実現される。スレーブ機器4の機器通信部411、プログラム更新部412、422、バージョン情報収集部413、通信部414、422は、スレーブ機器4が備える親マイコン及び子マイコンのCPU、電子回路を含んだICチップ等で実現される。

【0047】

<プログラムの更新システムの動作例>

図2は、プログラム更新システムのプログラム更新処理における動作例を示す図である。図2では、代表してマスタ機器3がプログラム更新処理を実行する場合を例として説明する。HDD36には、アップデートフォルダとプログラムフォルダAとプログラムフォルダBとが設定されている。アップデートフォルダは、プログラム更新に関するファイルが格納される。プログラムフォルダA及びプログラムフォルダBは、マスタ機器3が使用する全プログラム(以下、プログラム群と称す)が格納されている。図2では、プログラムフォルダAには、バージョン2のプログラム群が格納されており、プログラムフォルダBには、プログラムフォルダAに格納されているプログラム群のバージョンよりも古いバージョン1のプログラム群が格納されているとして説明する。

【0048】

マスタ機器3のプログラム更新部34は、HDD36のアップデートフォルダ内に管理ファイルを作成する(OP1)。この時点では、管理ファイルは空である。

【0049】

ダウンロード選択部32は、携帯電話/DCM2からダウンロードするか、デッキ5から更新データと読み出すかを選択し、通信部41に指示を出す。通信部41は、ダウンロード選択部32から指示された方を選択し、差分データファイルをダウンロードする(OP2)。ダウンロードされた差分データファイルはHDD36のアップデートフォルダに保存される。

【0050】

10

20

30

40

50

尚、差分データとは、センタC 1又はCD、DVDなどの外部記憶媒体が保持するプログラムと、プログラムフォルダA又はプログラムフォルダBが格納するプログラム群の内最も新しいバージョンのプログラム群との差分データである。バージョン情報収集部33は、プログラムフォルダA及びプログラムフォルダBに格納されているプログラム群のバージョンを管理している。携帯電話/DCM2経由でセンタC 1から差分データをダウンロードする場合には、バージョン情報収集部33で管理している内で最も新しいバージョンをセンタC 1に通知し、センタC 1において最新のバージョンのプログラムとの差分データを生成する。デッキ5を経由して、CD-ROM/DVD-ROM等の外部記憶媒体から差分データを取得する場合には、通信部31は、バージョン情報収集部33で管理している内で最も新しいバージョンのプログラムと、外部記憶媒体に格納されているバージョンのプログラムとの差分のみを読み出す。

10

**【0051】**

差分データファイルのアップデートフォルダへの保存が完了すると、プログラム更新部34は、管理ファイルに差分データのダウンロード済みを記録する(OP3)。管理ファイルへの記録方法は、例えば、管理ファイル内の差分データのダウンロード済みを示す「差分データダウンロードフラグ」を立てる。

**【0052】**

次に、プログラム更新部34は、ダウンロードされた差分データファイルをモジュール毎の差分データファイルに分割する(OP4)。プログラム更新部34は、モジュール毎の差分データファイルを生成すると、管理ファイルにファイル分割済みであることを記録する(OP5)。管理ファイルへの記録方法は、例えば、管理ファイル内のファイル分割済みを示す「ファイル分割済みフラグ」を立てる。

20

**【0053】**

プログラム更新部34は、FROM38内のブートフラグを確認して、更新対象のプログラムフォルダを決定する(OP6)。ブートフラグは、プログラムフォルダA又はプログラムフォルダBの何れが現在有効になっているかを示すフラグを保持する。例えば、プログラムフォルダAが、現在有効になっているとする。現在プログラムフォルダAが有効になっている場合には、プログラムフォルダAに格納されているプログラム群が、マスター機器3が搭載するプログラムの最新バージョンであることを示す。従って、ブートフラグでプログラムフォルダAが現在有効なフォルダである場合には、プログラム更新部34は、より古いバージョンのプログラム群を格納するプログラムフォルダBを更新対象のフォルダに決定する。

30

**【0054】**

プログラム更新部34は、更新対象のプログラムフォルダB内に格納されるファイルを全て削除する(OP7)。プログラム更新部34は、更新対象でないプログラムフォルダA内のファイルと差分データファイルとから新たにプログラムファイルをプログラムフォルダBに生成する(OP8)。例えば、プログラムフォルダA内のファイルAと、差分データファイルaとから新ファイルAをプログラムフォルダB内に生成する。新ファイルAの生成が完了すると、プログラム更新部34は、管理ファイル内に新ファイルAを生成済であることを記録する(OP9)。

40

**【0055】**

プログラム更新部34は、プログラムフォルダAに格納されているファイル分だけOP8からOP9の処理を繰り返す(OP10)。

**【0056】**

プログラムフォルダAに格納されている全ファイルについて新ファイルの生成が完了すると、プログラム更新部34は、FROM38内のブートフラグの有効プログラムフォルダをプログラムフォルダBに設定する(OP11)。プログラム更新部34は、管理ファイル内にブートフラグ更新済みを記録する(OP12)。

**【0057】**

プログラム群を格納するプログラムフォルダを2つ用意し、古いバージョンのプログラ

50

ム群を格納するプログラムフォルダのみを更新することによって、その直前まで使用していた動作確認済みのプログラムフォルダをバックアップとして保持することができる。

【 0 0 5 8 】

図 3 A 及び図 3 B は、図 2 で説明した O P 1 から O P 1 2 の各処理中に電源断が発生した場合の復帰動作の例について示す図である。図 3 では、車載器 1 の電源（自動車のアクセサリ電源）が何らかの要因でオフとなり、プログラム更新処理が強制的に終了され、その後、車載器 1 が起動した場合のプログラム更新処理の復帰動作を示す。

【 0 0 5 9 】

車載器 1 が再起動した後、アップデートフォルダ内の管理ファイルの中が初期状態（空）である場合には、プログラム更新部 3 4 は、O P 1 から処理を開始する。管理ファイルの中身が初期状態である場合には、電源断が発生した時の状態として、管理ファイルの生成処理中（O P 1）、差分データのダウンロード中（O P 2）、管理ファイルへ差分データのダウンロード済み記録中（O P 3）、であったと考えられる。若しくは、O P 1 から O P 3 までの各処理の間であったと考えられる。そのため管理ファイルの中身が初期状態である場合には、プログラム更新部 3 4 は、O P 1 から処理を開始し、生成途中のファイル（管理ファイル、差分データファイル）が存在する場合には、一旦その生成途中のファイルを破棄して再生成する。

10

【 0 0 6 0 】

管理ファイルの中身が、差分データダウンロード済み記録している場合には、すなわち、「差分データダウンロードフラグ」が立っている場合には、プログラム更新部 3 4 は、O P 4 から処理を開始する。この場合には、電源断が発生した時の状態として、差分データファイルを分割してモジュール毎の差分データファイル生成中（O P 4）、管理ファイルへモジュール毎に分割した差分データファイル生成済み記録中（O P 5）、の何れかであったと考えられる。若しくは、O P 3 から O P 5 までの各処理の移行時であったと考えられる。そのため管理ファイルの中身が差分データダウンロード済み記録している場合には、プログラム更新部 3 4 は、O P 4 から処理を開始する。

20

【 0 0 6 1 】

管理ファイルの中身が、差分データダウンロード済みとファイル分割済みとを記録している場合には、すなわち、「差分データダウンロードフラグ」と「ファイル分割済みフラグ」とが立っている場合には、プログラム更新部 3 4 は、O P 6 から処理を開始する。この場合には、電源断が発生した時の状態として、FROM 3 8 内のブートフラグのチェック中（O P 6）、更新対象プログラムフォルダ内の全ファイルの削除中（O P 7）、更新対象プログラムフォルダ内に新ファイルの生成中（O P 8）、管理ファイルへ新ファイルの生成済み記録中（O P 9）、の何れかであったと考えられる。若しくは、O P 5 から O P 9 までの各処理の移行時であったと考えられる。そのため管理ファイルの中身が差分データダウンロード済みとファイル分割済みとを記録している場合には、プログラム更新部 3 4 は、O P 6 から処理を開始する。

30

【 0 0 6 2 】

管理ファイルの中身が、差分データダウンロード済みと、ファイル分割済みと、新ファイル X 生成済み（X：変数）とを記録している場合には、すなわち、「差分データダウンロードフラグ」と「ファイル分割済みフラグ」と「新ファイル X 生成済みフラグ」とが立っている場合には、プログラム更新部 3 4 は、O P 1 0 から処理を開始する。即ち、プログラム更新部 3 4 は、新ファイル X の次のファイルの新ファイル生成から処理を開始する。この場合には、電源断が発生した時の状態として、更新対象プログラムフォルダ内に新ファイルの生成中、管理ファイルへ新ファイルの生成済み記録中、の何れかであったと考えられる。そのため管理ファイルの中身が差分データダウンロード済みと、ファイル分割済みと、新ファイル X 生成済みとを記録している場合には、プログラム更新部 3 4 は、ファイル X の次のファイルの新ファイルの生成から処理を開始する。

40

【 0 0 6 3 】

管理ファイルの中身が、差分データダウンロード済みと、ファイル分割済みと、全ファ

50

イル生成済みとを記録している場合には、すなわち、「差分データダウンロードフラグ」と、「ファイル分割済みフラグ」と、「新ファイルX生成済みフラグ」が全ファイルについて立っている場合には、プログラム更新部34は、FROM38内のブートフラグを参照してプログラムのロードを開始する。この場合には、電源断が発生した時の状態として、FROM38内のブートフラグの有効フォルダを更新中(OP11)、管理ファイルへブートフラグの更新済みとを記録中、の何れかであったと考えられる。そのため管理ファイルの中身が差分データダウンロード済みと、ファイル分割済みと、全ファイル生成済みとを記録している場合には、プログラム更新部34は、ブートフラグの内容を参照してプログラムをロードする。

#### 【0064】

管理ファイルにおいて、プログラム更新処理の進捗状況を管理することによって、プログラム更新処理が電源断等によって強制終了された場合でも、管理ファイルを参照することによって、強制終了された(と予想される)処理から再開することができる。

#### 【0065】

<ブートフラグの値の例>

図4は、FROM38内のブートフラグの値の例を示す図である。例えば、プログラムフォルダ毎に、ブートフラグを保持する。すなわち、FROM38内にブートフラグA及びブートフラグBを保持し、これによって、ローディング時にロードすべきプログラムフォルダの選択を行う(有効なプログラムフォルダを示す)。

#### 【0066】

図4では、プログラムフォルダAの状態を示すブートフラグAと、プログラムフォルダBの状態を示すブートフラグBとが、それぞれ16ビットで構成される例を示す。図4では、各ブートフラグの値を16進数表記している。各ブートフラグは、「0x0000」が無効、「0x0001」が有効を示す。FROM38の特性として、ブートフラグを書き換える(更新する)場合には、いったんブートフラグのデータを無効にする必要があるため、「0xFFFF」としてデータをクリアする。また、図4では、ブートフラグの更新は、プログラムフォルダAの更新がプログラムフォルダBの更新に優先して行われる場合を示す。ブートフラグA及びブートフラグBの状態遷移は、以下に示す状態1から状態2,3,4を経て状態5、若しくは、状態5から状態6,7,8を経て状態1に遷移するとする。

#### 【0067】

状態1は、ブートフラグAが「0x0001」、ブートフラグBが「0x0000」である状態である。すなわち、プログラムフォルダAが現在有効なプログラムファイルとして選択されている状態を示す。

#### 【0068】

状態2は、ブートフラグAが「0xFFFF」、ブートフラグBが「0x0000」である状態である。この場合には、ブートフラグAは書き換えられている最中であることを示し、ブートフラグBの値は無効を示すものの、ブートフラグAが更新中であることを鑑みて、現在有効なプログラムフォルダはプログラムフォルダBであると判断する。

#### 【0069】

状態3は、ブートフラグAが「0x0000」、ブートフラグBが「0x0000」である状態である。この場合には、ブートフラグAおよびブートフラグBの双方が更新中であることを示し、状態1から状態5への過渡期であることを鑑みて、プログラムフォルダBが有効なプログラムフォルダであると判断する。

#### 【0070】

状態4は、ブートフラグAが「0x0000」、ブートフラグBが「0xFFFF」である状態である。ブートフラグBは書き換えられている最中であることを示す。ブートフラグAが優先的に更新されることを鑑みて、プログラムフォルダBが現在有効なプログラムフォルダであることを判断する。

#### 【0071】

10

20

30

40

50

状態 5 は、ブートフラグ A が「0 x 0 0 0 0」、ブートフラグ B が「0 x 0 0 0 1」である状態である。この場合、プログラムフォルダ A は無効で、プログラムフォルダ B が有効である。

【0072】

状態 6 は、ブートフラグ A が「0 x F F F F」、ブートフラグ B が「0 x 0 0 0 1」である状態である。ブートフラグ A は書き換えられている最中であり、状態 5 から遷移することを鑑みて、プログラムフォルダ A は有効で、プログラムフォルダ B は無効であると判断する。

【0073】

状態 7 は、ブートフラグ A が「0 x 0 0 0 1」、ブートフラグ B が「0 x 0 0 0 1」である状態である。この場合には、ブートフラグ A およびブートフラグ B の双方が更新中であることを示し、状態 5 から状態 1 への過渡期であることを鑑みて、プログラムフォルダ A が有効なプログラムフォルダであると判断する。

10

【0074】

状態 8 は、ブートフラグ A が「0 x 0 0 0 1」、ブートフラグ B が「0 x F F F F」である状態である。ブートフラグ B は書き換えられている最中であることを示す。ブートフラグ A が優先的に更新されることを鑑みて、プログラムフォルダ A が現在有効なプログラムフォルダであることを判断する。

【0075】

車載器 1 が起動したときに、ブートフラグの値が状態 2 から状態 4 の場合は状態 5 の値に、状態 6 から状態 8 の場合は状態 1 の値に書き換えてから起動する。

20

【0076】

< 起動シーケンスの例 >

プログラム更新処理が完了すると、車載器 1、マスタ機器 3、もしくは各スレーブ機器 4 を再起動して、更新した最新バージョンのプログラムをローディングする処理を実行する。このとき、プログラム更新処理は成功したとしても、更新した最新バージョンのプログラムのローディング処理に異常が発生する等の可能性がある。このようなプログラムのローディングの異常の原因としては、更新プログラムにバグがある、更新プログラムを D R A M 3 7 上にローディングするときにビット化けが生じる等が考えられる。このような、更新後のプログラムが起動しない場合に、復帰する手段として、第 1 実施形態の車載機 1 では、H D D 3 6 にリカバリデータを保持する。

30

【0077】

図 5 は、車載器 1 のリカバリデータの保持の例を示す図である。H D D 3 6 は、ローダフォルダとリカバリフォルダを保持している。ローダフォルダには、ローディング処理を実行するローダプログラムが格納されている。リカバリフォルダには、リカバリプログラムと各機器（マスタ機器、スレーブ機器）の工場出荷状態のプログラムが格納されている。リカバリプログラムは、リカバリ処理を実行するプログラムである。

【0078】

<< 正常時の起動動作の流れの例 >>

図 6 は、正常時の起動動作の流れの例を示す図である。図 6 は、車載器 1 に電源が投入されて、マスタ機器 3 がスレーブ機器 4 を含む起動シーケンスを実行する場合について説明する。マスタ機器 3 は、車載器 1 又はマスタ機器 3 に電源が投入されると、F R O M 3 8 内に格納されているブートプログラムが起動する（O P 2 1）。ブートプログラムを実行する機器の C P U は、D R A M 3 7 にロードされているアプリケーションプログラムのエラー検出を行う（O P 2 2）。

40

【0079】

アプリケーションプログラムのエラーチェックは、チェックサムによるエラー検出、異常リセット発生回数のチェック、異常ローディングの発生回数のチェックが行われる。異常リセットは、アプリケーションプログラムにバグがあると自動的にリセットを行うことがある。異常リセット回数は、ログとして記録された、今回の起動までにアプリケーショ

50

ンプログラムがリセットされた回数である。異常ローディングとは、HDD 36に格納されたアプリケーションプログラムをDRAM 37にコピーする処理に失敗することである。異常ローディング回数とは、今回の起動までに異常ローディングが発生した回数である。

【0080】

アプリケーションプログラムのエラーチェックにおいて、異常がない場合には、その後、DRAM 37にロードされているアプリケーションプログラムを実行する(OP 23)。

【0081】

正常時には、CPUは、HDD 36に格納されているプログラムフォルダを参照せず、DRAM 37にロードされているアプリケーションプログラムのデータを使用して起動する。

10

【0082】

<< 異常が発生した場合の起動時の動作の例 >>

図7は、起動シーケンスの例を示す図である。図7も、図6と同様に、マスタ機器3が起動シーケンスを行う場合について説明する。

【0083】

車載器1又はマスタ機器3に電源が投入される(又は、再起動される)と、FROM 38に格納されたブートプログラムを起動する(OP 31)。

【0084】

マスタ機器3のCPUは、ログに記録されたアプリケーションプログラムの異常リセット回数Xをチェックする(OP 32)。アプリケーションプログラムの異常リセット回数Xが予め設定された所定回数以上の値か否かを確認する。アプリケーションの異常リセット回数Xが所定回数以上の値である場合には(OP 32: Y)、異常時シーケンスAに処理が移る(OP 38)。異常時シーケンスAについては、後述する。

20

【0085】

アプリケーションプログラムの異常リセット回数Xが所定値より小さい値である場合には(OP 32: N)、異常ローディング回数Yが予め設定された所定回数以上の値か否かを判定する(OP 33)。異常ローディング回数Yが所定回数以上の値である場合には(OP 33: Y)、異常時シーケンスBに処理が移る(OP 37)。異常時シーケンスBについては、後述する。

30

【0086】

アプリケーションプログラムの異常ローディング回数Yが所定回数よりも小さい場合には(OP 33: N)、チェックサムを確認する(OP 34)。チェックサムに異常がある場合には(OP 34: Y)、異常時シーケンスAに処理が移る(OP 38)。チェックサムに異常がない場合には(OP 34: N)、アプリケーションプログラムを起動する(OP 35)。

【0087】

CPUは、アプリケーションプログラムの異常リセットの発生を監視しており、異常リセットが発生した場合には(OP 36)、異常リセット回数Xに1を加え、異常リセットの発生を記録する(OP 39)。

40

【0088】

<< 異常時シーケンスA >>

図8は、異常時シーケンスA、すなわち、異常リセット回数Xが所定回数以上の値である場合と、チェックサムに異常がある場合とに実行される処理の流れの例を示す図である。図9は、異常時シーケンスAの処理フローを示す図である。図8と図9とは、同じ処理には同じ符号を付して示す。

【0089】

異常リセット回数Xが所定回数以上の値である場合、及び、チェックサムに異常がある場合には、DRAM 37にロードされているロードプログラムをHDD 36のロードフォ

50

ルダから再度ローディングする（OP41）。次に、ローディングされたローダプログラムを起動する（OP42）。

【0090】

起動されたローダプログラムに従って、CPUは、FROM38内のブートフラグをチェックして、現在有効なプログラムフォルダを取得する（OP43）。CPUは、HDD36に格納されている現在有効なプログラムフォルダからアプリケーションプログラムを再度ローディングする（OP44）。

【0091】

アプリケーションプログラムのローディング時に異常が発生した場合には（OP45：Y）、異常ローディング回数Yに1を加え、異常ローディングの発生を記録する（OP46）。その後、再起動して図7に示す処理を再度実行する（OP47）。

10

【0092】

アプリケーションプログラムのローディングに成功した場合にも（OP45：N）、再起動をして、図7に示す処理を再度実行する（OP47）。

【0093】

<<異常時シーケンスB>>

図10は、異常時シーケンスB、すなわち、異常ローディング回数Yが所定回数以上の値である場合に実行される処理を示す図である。図11は、異常時シーケンスBの処理フローを示す図である。図10と図11とは、同じ処理には同じ符号を付して示す。

【0094】

異常ローディング回数Yが所定回数以上の値である場合には、まず、車載器1のユーザに対して、リカバリ処理を実行するか否かを問い合わせるコーションを表示する（OP51）。表示されるコーションは、例えば、「プログラムファイル異常のため、起動できません。工場出荷状態に戻します。よろしいですか。（YES/NO）」など、ユーザにリカバリ処理の実行の確認を求める内容である。

20

【0095】

ユーザは、コーション表示に対して、入力を行う（OP52）。ユーザがリカバリ処理を実行することを選択すると（OP53：Y）、CPUはリカバリプログラムをHDD36内のリカバリファイルからローディングする（OP54）。ローディングされたリカバリファイルを起動し（OP55）、HDD36のリカバリフォルダから工場出荷状態のアプリケーションプログラムをローディングする（OP56）。その後、再起動して図7に示す処理を再度実行する（OP57）。

30

【0096】

ユーザがリカバリ処理の実行を選択しなかった場合には（OP53：N）、再起動を行う旨のコーションを表示する（OP58）。このとき表示されるコーションは、例えば、「システムを再起動します。」等、ユーザに再起動の確認を行う内容である。その後、再起動して図7に示す処理を再度実行する（OP57）。

【0097】

<第1実施形態の作用効果>

プログラムを格納するプログラムフォルダを2個保持し、プログラムの更新処理を行う場合には、どちらか一方のみを更新する。もう一方はバックアップとして保持する。これによって、プログラム更新処理中に電源断等で処理が強制終了された場合でも、元の動作保障されたプログラムのデータを失うことがない。

40

【0098】

プログラム更新処理において、進捗状況を管理ファイルに記録することによって、プログラム更新処理が電源断等により強制終了された場合でも、管理ファイルに完了が記録された処理の次の処理から再開することができる。したがって、プログラム更新処理を最初からやり直す必要がなく、時間を省くことができる。

【0099】

マスタ機器3がスレーブ機器4の更新プログラムのダウンロードも一括して行う。スレ

50

ーブ機器4のプログラム更新処理を行う際には、専用の治具が必要となる場合があった。しかし、プログラム更新の窓口をマスタ機器3に一本化されることで、専用の治具がなくてもユーザがスレーブ機器4のプログラム更新処理を行うことができる。

【0100】

更新プログラムをセンタC1からダウンロードする際には、車載器1が保持するプログラムと、センタC1が保持する新しいバージョンのプログラムとの差分データのみをダウンロードする。これによって、センタC1と車載器1との間の通信量を少なく抑えることができ、データをダウンロードする時間も短くすることができる。パケット通信等従量課金制の通信の場合には、通信量を抑えることができる。

【0101】

10

起動時には、DRAM内のアプリケーションプログラムのチェックサム、異常リセット回数、異常ローディング回数といった、3種のパラメータをチェックすることによって、起動方法を判断する。これによって、段階的にリカバリを実行でき、効率がよい。また、リカバリフォルダ内には工場出荷状態の全プログラムが格納されているので、最終的には、工場出荷状態にフルリカバリすることができ、起動が保証される。

【0102】

第1実施形態によれば、ユーザ自身がプログラム更新処理を行うことができ、且つ、プログラム更新処理の中断や、プログラム更新処理の失敗により起動不可能な状態になった場合でも、正常な状態に復帰することができる。

【0103】

20

<変形例>

管理ファイルを、プログラムファイルのように2個用意し、二重化してもよい。例えば、プログラム更新処理が開始されると、アップデートフォルダ内の管理ファイルAのコピーを生成する。この管理ファイルAのコピーファイルを管理ファイルBとする。管理ファイルAはそのまま保持し、プログラム更新処理では管理ファイルBを更新する。プログラム更新処理が完了した時点で、管理ファイルAを削除する。このようにすることによって、プログラム更新処理中に電源断が発生した場合でも、管理ファイルAをバックアップとして保持できるため、管理ファイルの内容が失われることを防ぐことができる。

【0104】

30

また、管理ファイルBの更新中に電源断等が発生した場合には、例えば、以下のように判断することで、処理を再開した場合に適切な管理ファイルを選択することができる。

(1) 管理ファイルAが存在し、且つ管理ファイルBが存在しない場合には、管理ファイルAを参照する。

(2) 管理ファイルA及び管理ファイルBが存在する場合には、管理ファイルAを参照し、管理ファイルBは削除する。

(3) 管理ファイルAが存在せず、且つ、管理ファイルBが存在する場合には、管理ファイルBを参照する。

【0105】

40

第1実施形態では、プログラムフォルダ内に格納されているプログラム群を一括して更新する処理について説明した。これに代えて、個々のプログラムを更新することも可能である。個々のプログラムを指定して更新する場合は、ブートフラグで示される有効なプログラムフォルダに格納されている方のプログラムを更新する。更新が完了したら、更新されたプログラムについて、更新済みであることを管理ファイルに記録しておく。

【0106】

第1実施形態では、異常時シーケンスBにおいて、リカバリフォルダから工場出荷状態のプログラムをローディングした。これに代えて、異常時シーケンスBにおいて、ブートフラグをチェックし、現在有効ではないプログラムフォルダ、すなわち、更新前の動作実績のある旧バージョンのプログラムをローディングしてもよい。

【0107】

50

また、第1実施形態では、マスタ機器3が各スレーブ機器4のプログラム更新処理、及

び、リカバリ処理を行う場合について説明した。本発明は、これに限られるものではなく、自己が使用するアプリケーションプログラムについて、各スレーブ機器 4 でプログラム更新処理、及び、リカバリ処理を行ってもよい。

【0108】

<その他>

以上の実施形態に関し、更に以下の付記を開示する。

(付記 1)

第 1 のバージョンのプログラムを格納する第 1 格納部と、  
前記第 1 のバージョンと同じ又は前記第 1 のバージョンよりも新しいバージョンである第 2 のバージョンのプログラムを格納する第 2 格納部と、  
前記第 2 のバージョンのプログラムと前記第 2 のバージョンのプログラムよりもバージョンが新しい第 3 のバージョンのプログラムとの差分を取得する取得部と、  
前記第 2 格納部に格納される前記第 2 のバージョンのプログラムと前記取得部によって取得した差分とから前記第 3 のバージョンのプログラムを生成し、前記生成した第 3 のバージョンのプログラムを前記第 1 格納部に格納する更新部と  
を備えるプログラム更新装置。

10

(付記 2)

前記第 1 格納部と前記第 2 格納部とのどちらに格納されているプログラムを使用するかを示す情報を保持する記憶部をさらに備え、

前記更新部は、前記取得した差分と、前記第 2 格納部に格納される前記第 2 のバージョンのプログラムと、から前記第 3 のバージョンのプログラムを生成し、前記生成した第 3 のバージョンのプログラムを前記第 1 格納部に格納した場合に、前記記憶部によって保持される前記情報を前記第 1 格納部に格納されているプログラムを使用することを示す情報に更新する

20

請求項 1 に記載のプログラム更新装置。

(付記 3)

前記差分を取得する処理、前記第 3 のバージョンのプログラムを生成する処理、及び、前記生成した第 3 のプログラムを第 1 格納部に格納する処理のそれぞれについて処理の完了を示す管理情報を格納する管理情報格納部を

さらに備え、

前記更新部は、前記差分を取得する処理、前記第 3 のバージョンのプログラムを生成する処理、及び、前記生成した第 3 のプログラムを第 1 格納部に格納する処理の各処理が完了した場合に、該処理の完了を示す管理情報を前記管理情報格納部に格納し、前連の処理が中断された場合に、前記管理情報格納部に格納されている管理情報に基づいて、処理の完了が示されていない処理を実行する

30

請求項 1 に記載のプログラム更新装置。

(付記 4)

第 1 のバージョンのプログラムを格納する第 1 格納部と、

前記第 1 のバージョンと同じ又は前記第 1 のバージョンよりもバージョンが新しい第 2 のバージョンのプログラムを格納する第 2 格納部と、を備えるコンピュータが、

40

前記第 2 のバージョンのプログラムと前記第 2 のバージョンよりもバージョンが新しい第 3 のバージョンのプログラムとの差分を取得するステップと、

前記第 2 格納部に格納される前記第 2 のバージョンのプログラムと前記取得した差分とから前記第 3 のバージョンのプログラムを生成するステップと、

前記第 1 格納部に格納される前記第 1 のバージョンのプログラムを削除して、前記生成した第 3 のバージョンのプログラムを前記第 1 格納部に格納するステップと、を実行するプログラム更新方法。

(付記 5)

工場出荷時のバージョンのプログラムと前記工場出荷時のバージョンよりも新しいバージョンである第 1 のバージョンのプログラムとを格納した補助記憶装置と、

50

前記補助記憶装置からローディングされた前記第 1 のバージョンのプログラムを保持する主記憶装置と、

電源投入時に前記主記憶装置に保持される第 1 のバージョンのプログラムを起動する CPU と、を備える装置であって、

前記 CPU は、

前記主記憶装置に保持される第 1 のバージョンのプログラムにエラーがある場合、及び、前記主記憶装置に保持される第 1 のバージョンのプログラムが今回の電源投入時までの間に所定回数以上動作を中断させていた場合には、前記補助記憶装置に格納された前記第 1 のバージョンのプログラムを前記主記憶装置にローディングし、

前記主記憶装置に保持される第 1 のバージョンのプログラムが今回の電源投入時までの間に所定回数以上ローディングに失敗している場合には、前記補助記憶装置に格納された前記工場出荷時のバージョンのプログラムを前記主記憶装置にローディングする  
情報処理装置。

10

【符号の説明】

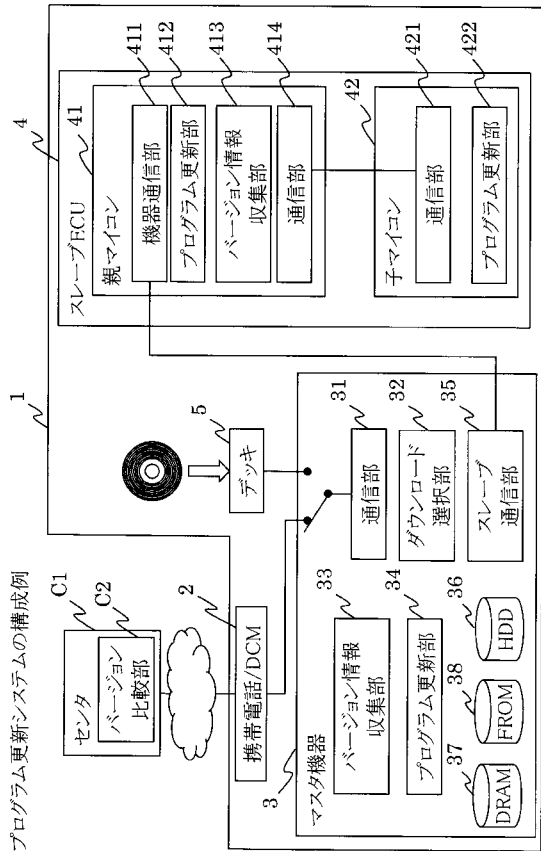
【 0 1 0 9 】

- 1 車載器
- 2 携帯電話 / D C M
- 3 マスタ機器
- 4 スレーブ機器
- 5 デッキ
- 3 1 通信部
- 3 2 ダウンロード選択部
- 3 3 バージョン情報収集部
- 3 4 プログラム更新部
- 3 5 スレーブ通信部
- 3 6 H D D
- 3 7 D R A M
- 3 8 F R O M
- 4 1 親マイコン
- 4 2 子マイコン
- 4 1 1 機器通信部
- 4 1 2 プログラム更新部
- 4 1 3 バージョン情報収集部
- 4 1 4 通信部
- 4 2 1 通信部
- 4 2 2 プログラム更新部

20

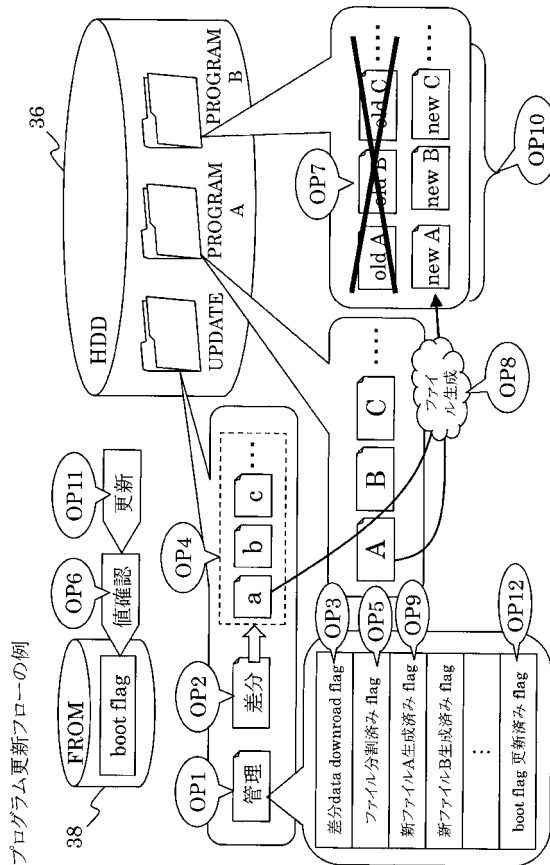
30

【 図 1 】



プログラム更新システムの構成例

【 図 2 】



プログラム更新フローの例

【 図 3 A 】

各処理更新中に電源断が発生した場合の復帰動作内容の例

起動後の管理ファイルの状態	電源復旧後の動作	電源断時のステータス
初期状態	OP1 から開始。 生成途中のファイルは破棄して再生成。	OP1 UPDATEフォルダ管理ファイル生成中 OP2 更新差分データダウンロード中 OP3 管理ファイル記録中 (差分データダウンロード済みflag)
差分データDL済:OK	OP4 から開始。	OP4 差分ファイル分割中 OP5 管理ファイル記録中(ファイル分割済みflag)
差分データDL済:OK ファイル分割済:OK	OP6 から開始。	OP6 FROM内boot flag チェック中 OP7 更新対象プログラムフォルダ内ファイル全削除中 OP8 更新対象プログラムフォルダ内に新ファイルを生成中 OP9 管理ファイル記録中(新Xファイル生成済みflag)

【 図 3 B 】

各処理更新中に電源断が発生した場合の復帰動作内容の例 (続き)

起動後の管理ファイルの状態	電源復旧後の動作	電源断時のステータス
差分データDL済:OK ファイル分割済:OK 新Xファイル生成済:OK	OP10 から開始。	OP10 -
差分データDL済:OK ファイル分割済:OK 新全ファイル生成済:OK	boot flag の内容を参照して、プログラムをロード。	OP11 FROM内boot flag 更新中 (有効プログラムフォルダ) OP12 管理ファイル更新中 (boot flag 更新済みflag)

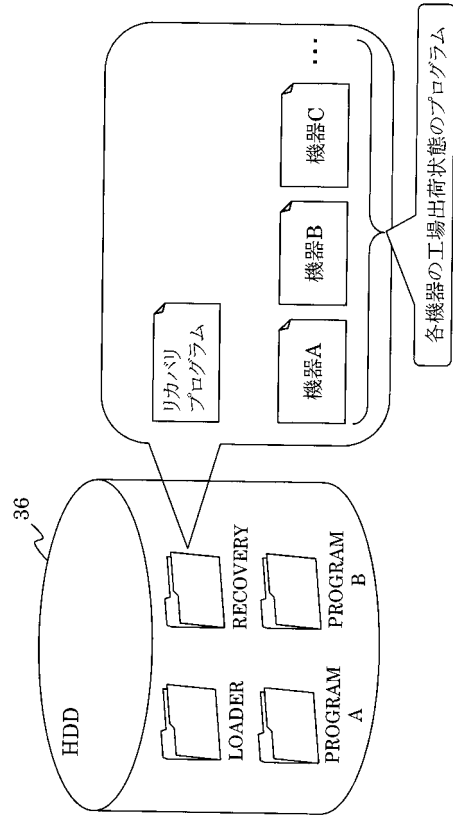
【 図 4 】

boot flag の値の例

状態	flag A	flag B	有効フォルダ	
1	0001	0000	A	—
2	FFFF	0000	B	boot flag A 書き換え中でフォルダB更新済みと判断
3	0000	0000	B	boot flag 書き換え中でフォルダB更新済みと判断
4	0000	FFFF	B	boot flag B 書き換え中でフォルダB更新済みと判断
5	0000	0001	B	—
6	FFFF	0001	A	boot flag A 書き換え中でフォルダA更新済みと判断
7	0001	0001	A	boot flag 書き換え中でフォルダA更新済みと判断
8	0001	FFFF	A	boot flag B 書き換え中でフォルダA更新済みと判断

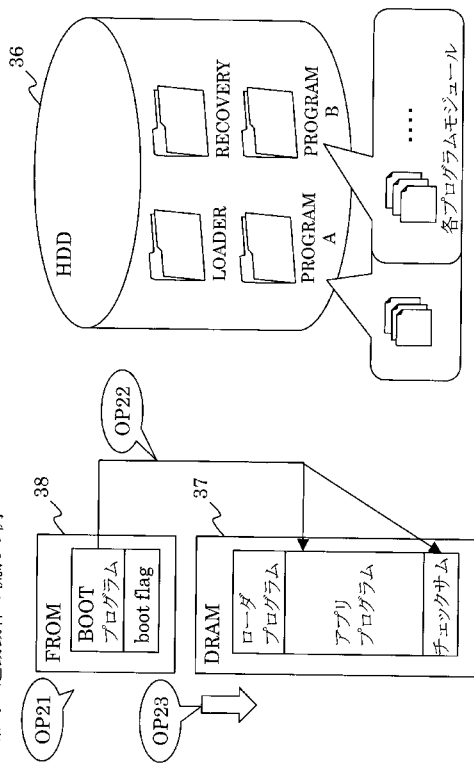
【 図 5 】

リカバリデータの保持の例



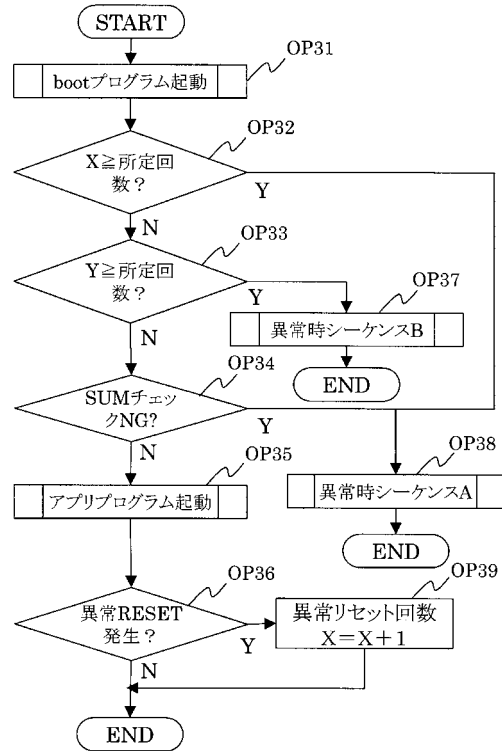
【 図 6 】

正常時の起動動作の流れの例

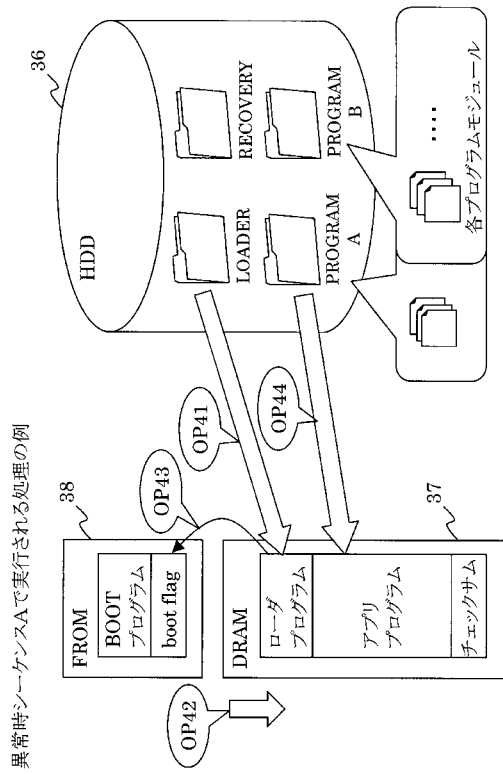


【 図 7 】

起動シーケンスの例

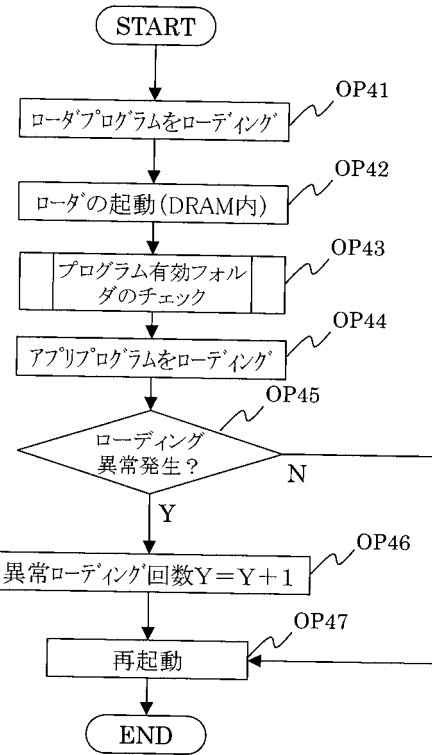


【図8】

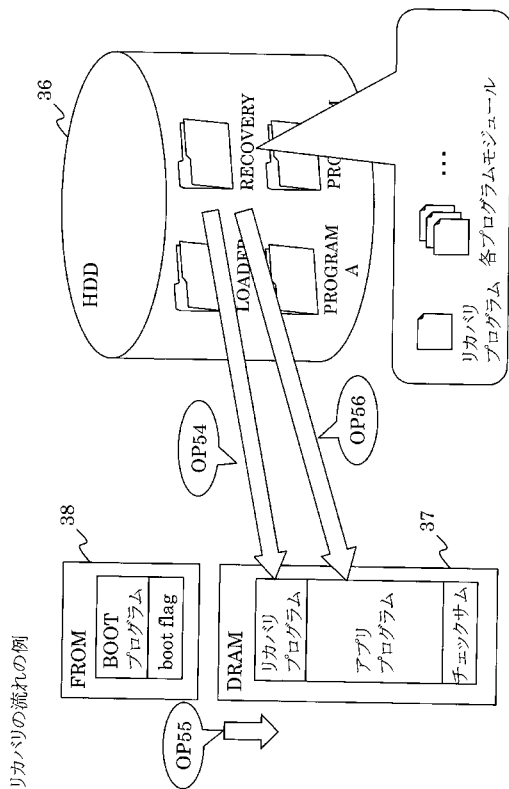


【図9】

異常時シーケンスAのフローの例

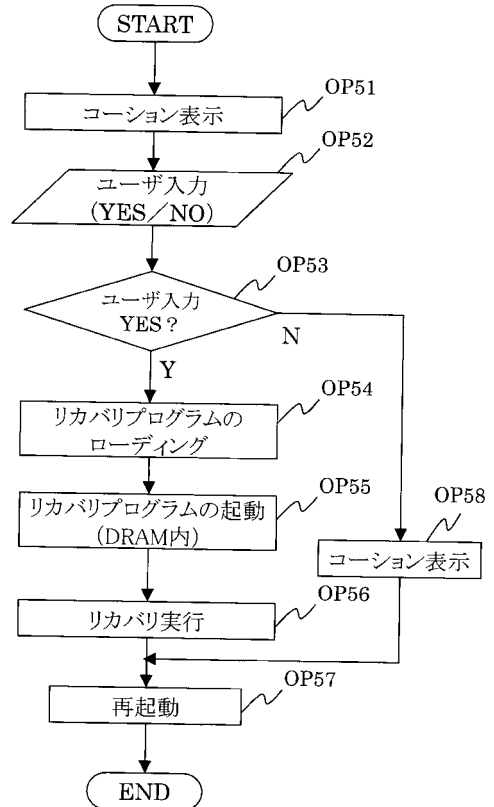


【図10】



【図11】

異常時シーケンスBのフローの例



---

フロントページの続き

- (72)発明者 石田 泰久  
兵庫県神戸市兵庫区御所通1丁目2番28号 富士通テンテクノロジー株式会社内
- (72)発明者 籠谷 成彦  
兵庫県神戸市兵庫区御所通1丁目2番28号 富士通テン株式会社内
- (72)発明者 杉本 浩伸  
愛知県豊田市トヨタ町1番地 トヨタ自動車株式会社内
- (72)発明者 小段 友紀  
愛知県豊田市トヨタ町1番地 トヨタ自動車株式会社内

Fターム(参考) 5B027 CC02

5B376 AB06 AB15 AB25 AB28 AB47 AC07 AE05 AE18 AE44 AE62  
AE63 CA06 CA15 CA19 CA35 CA51 CA52 CA74 CA76 DA03  
DA08 EA03 EA13 FA01 FA13 GA08 GA09