

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7207818号
(P7207818)

(45)発行日 令和5年1月18日(2023.1.18)

(24)登録日 令和5年1月10日(2023.1.10)

(51)国際特許分類

F I

G 0 6 F 15/78 (2006.01)

G 0 6 F 15/78 5 5 0

H 0 2 J 7/02 (2016.01)

H 0 2 J 7/02 H

G 0 6 F 12/00 (2006.01)

G 0 6 F 12/00 5 9 7 U

G 0 6 F 8/65 (2018.01)

G 0 6 F 8/65

請求項の数 13 (全11頁)

(21)出願番号 特願2021-513394(P2021-513394)
(86)(22)出願日 令和2年1月10日(2020.1.10)
(65)公表番号 特表2022-501706(P2022-501706
A)
(43)公表日 令和4年1月6日(2022.1.6)
(86)国際出願番号 PCT/KR2020/000511
(87)国際公開番号 WO2020/145742
(87)国際公開日 令和2年7月16日(2020.7.16)
審査請求日 令和3年3月19日(2021.3.19)
(31)優先権主張番号 10-2019-0003998
(32)優先日 平成31年1月11日(2019.1.11)
(33)優先権主張国・地域又は機関
韓国(KR)

(73)特許権者 521065355
エルジー エナジー ソリューション リ
ミテッド
大韓民国 ソウル ヨンドゥンボーグ ヨ
イーデロ 1 0 8 タワー1
(74)代理人 110000877
弁理士法人 R Y U K A 国際特許事務所
(72)発明者 リ、ホジョーン
大韓民国 0 7 3 3 6 ソウル, ヨンドゥ
ンボーグ, ヨイーデロ 1 2 8 エルジー
・ケム・リミテッド内
審査官 田中 幸雄

最終頁に続く

(54)【発明の名称】 エラー復旧方法、これを用いたマイクロコントローラユニット、および前記マイクロコン
トローラを含むバッテリー装置

(57)【特許請求の範囲】

【請求項1】

マイクロコントローラユニット (Micro Controller Unit、MCU)
において、

フラッシュメモリを含み、

前記フラッシュメモリは、フラッシュブートローダー (Flash Boot Loader) と、アプリケーション (application) ブロックと、アプリケーション
キーブロックと、を含み、

前記MCUのパワーオン時に、前記フラッシュブートローダーがアプリケーションのキ
ー値を確認し、

前記アプリケーションのキー値が有効でなければ、前記フラッシュメモリは、前記アプ
リケーションが保存されたアプリケーション領域を前記フラッシュブートローダーによっ
て再プログラミングする、フラッシュブートローダーモードに入るMCU。

【請求項2】

前記アプリケーションのキー値が有効でない場合、前記アプリケーションキーブロック
が前記アプリケーションに対応する領域に有効なキー値を書き込み、

前記アプリケーションのキー値が有効である場合、前記フラッシュブートローダーから
移動した命令により、前記アプリケーションブロックが前記アプリケーション領域に保存
された前記アプリケーションを実行する、

請求項1に記載のMCU。

【請求項 3】

前記アプリケーションキーブロックは、

前記再プログラミングが完了した後、有効なアプリケーションのキー値を書き込む、請求項 2 に記載の M C U。

【請求項 4】

前記フラッシュメモリは、

再プログラミング要請が受信されれば、前記再プログラミング要請に対応するアプリケーションのキー値を消し、前記フラッシュブートローダーモードに入る、請求項 1 から 3 のいずれか一項に記載の M C U。

【請求項 5】

前記フラッシュメモリは、

前記フラッシュブートローダーモードで、受信されたアプリケーションデータを再プログラミングし、アプリケーションのキー値を書き込む、請求項 4 に記載の M C U。

【請求項 6】

前記フラッシュメモリは、

前記アプリケーションのキー値が有効であれば、対応するアプリケーションを実行するアプリケーション実行モードに入り、

前記対応するアプリケーションの実行中に、再プログラミング要請が受信されれば、前記対応するアプリケーションのキー値を消す、

請求項 1 から 5 のいずれか一項に記載の M C U。

【請求項 7】

前記フラッシュメモリは、

前記アプリケーションのキー値を消した後、

前記アプリケーションのキー値を確認し、前記アプリケーションのキー値が有効でなければ、前記フラッシュメモリが前記フラッシュブートローダーモードに入る、

請求項 6 に記載の M C U。

【請求項 8】

フラッシュブートローダーと、アプリケーションブロックと、アプリケーションキーブロックと、を有するフラッシュメモリを含むマイクロコントローラユニット (Micro Controller Unit、M C U) のエラー復旧方法において、

前記 M C U のパワーオン時に、前記フラッシュブートローダーがアプリケーションキーブロックからアプリケーションのキー値を確認する段階と、

前記アプリケーションのキー値を確認する段階で、前記アプリケーションのキー値が有効でなければ、前記フラッシュメモリは、前記アプリケーションが保存されたアプリケーション領域を前記フラッシュブートローダーによって再プログラミングする、フラッシュブートローダーモードに入る段階と、を含む M C U のエラー復旧方法。

【請求項 9】

前記アプリケーションのキー値が有効でない場合、前記アプリケーションキーブロックが前記アプリケーションに対応する領域に有効なキー値を書き込み、

前記アプリケーションのキー値が有効である場合、前記フラッシュブートローダーから移動した命令により、前記アプリケーションブロックが前記アプリケーション領域に保存された前記アプリケーションを実行する、

請求項 8 に記載の M C U のエラー復旧方法。

【請求項 10】

受信されたアプリケーションデータを再プログラミングする段階と、

前記再プログラミングが完了した後、アプリケーションのキー値を書き込む段階と、をさらに含む、請求項 8 または 9 に記載の M C U のエラー復旧方法。

【請求項 11】

前記アプリケーションのキー値を確認する段階で、前記アプリケーションのキー値が有効であれば、対応するアプリケーションを実行する段階と、

10

20

30

40

50

前記対応するアプリケーションの実行中に、再プログラミング要請を受信する段階と、
前記再プログラミング要請を受信すれば、前記対応するアプリケーションのキー値を消す段階と、をさらに含む、請求項 8 から 10 のいずれか一項に記載の M C U のエラー復旧方法。

【請求項 12】

前記アプリケーションのキー値を消した後、

前記フラッシュブートローダーが前記アプリケーションキーブロックからアプリケーションのキー値を確認する段階から行われる、請求項 11 に記載の M C U のエラー復旧方法。

【請求項 13】

電氣的に連結された複数のバッテリーセルを含むバッテリーセル組立体と、

バッテリー管理のためのアプリケーションを保存したフラッシュメモリを含むマイクロコントローラユニット (Micro Controller Unit、MCU) と、

前記 MCU で行われるアプリケーションによりバッテリー管理を行うバッテリー管理システムと、を含み、

前記フラッシュメモリは、フラッシュブートローダー (Flash Boot Loader) と、アプリケーション (application) ブロックと、アプリケーションキーブロックと、を含み、

前記 MCU のパワーオン時に、前記フラッシュブートローダーがアプリケーションのキー値を確認し、

前記アプリケーションのキー値が有効でなければ、前記フラッシュメモリは、前記アプリケーションが保存されたアプリケーション領域を前記フラッシュブートローダーによって再プログラミングする、フラッシュブートローダーモードに入るバッテリー装置。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願との相互参照

【0002】

本出願は、2019 年 1 月 11 日付の韓国特許出願第 10-2019-0003998 号に基づく優先権の利益を主張し、当該韓国特許出願の文献に開示されたすべての内容は本明細書の一部として含まれる。

【0003】

本開示は、エラー復旧方法、これを用いたマイクロコントローラユニット、およびマイクロコントローラを含むバッテリー装置に関する。

【背景技術】

【0004】

アプリケーションキー (application key) 領域を消す (erase) 時、またはアプリケーションキーを書き込む (write) 時に、意図せぬパワーオフ (power off) などによるメモリ割れ現象が発生しうる。すると、後にアプリケーションキーを読み出す時に発生するエラー訂正コード (Error Correction Code、ECC) エラーによる CPU コア例外 (core exception) が発生してソフトウェアリセット (software reset) が発生する。

【0005】

これにより、初期パワーオン (power on) と有効キー (valid key) を確認するためのアプリケーションキー領域を読み出す過程の間に無限反復が発生する。すると、アプリケーションとフラッシュブートローダー (Flash Boot Loader、FBL) が全て行われない状況に陥ってしまう。

【発明の概要】

【発明が解決しようとする課題】

【0006】

フラッシュメモリのエラーを復旧できる方法、これを用いたマイクロコントローラユニ

10

20

30

40

50

ット、およびマイクロコントローラユニットを含むバッテリー装置を提供しようとする。

【課題を解決するための手段】

【0007】

発明の一つの特徴によるマイクロコントローラユニット (Micro Controller Unit、MCU) は、フラッシュメモリを含み、フラッシュメモリは、MCUのパワーオン時に、アプリケーションのキー値を確認し、アプリケーションのキー値が有効でなければ、再プログラミング可能なモードに入ることができる。

【0008】

フラッシュメモリは、フラッシュブートローダー (Flash Boot Loader) と、アプリケーション (application) ブロックと、アプリケーションキーブロックとを含み、MCUのパワーオン時に、フラッシュブートローダーがアプリケーションキーブロックからアプリケーションのキー値を確認し、アプリケーションのキー値が有効でなければ、フラッシュブートローダーモードに入ることができる。

10

【0009】

アプリケーションブロックは、フラッシュブートローダーモードで、受信されたアプリケーションデータを再プログラミングできる。

【0010】

アプリケーションキーブロックは、再プログラミングが完了した後、有効なアプリケーションのキー値を書き込むことができる。

【0011】

フラッシュメモリは、再プログラミング要請が受信されれば、再プログラミング要請に対応するアプリケーションのキー値を消し、モードに入ることができる。

20

【0012】

フラッシュメモリは、モードで、受信されたアプリケーションデータを再プログラミングし、アプリケーションのキー値を書き込むことができる。

【0013】

発明の他の特徴によるフラッシュブートローダーと、アプリケーションブロックと、アプリケーションキーブロックとを含むフラッシュメモリを含むMCUのエラー復旧方法は、MCUのパワーオン時に、フラッシュブートローダーがアプリケーションキーブロックからアプリケーションのキー値を確認する段階と、確認段階で、アプリケーションのキー値が有効でなければ、再プログラミング可能にフラッシュブートローダーにとどまる段階とを含む。

30

【0014】

MCUのエラー復旧方法は、受信されたアプリケーションデータを再プログラミングする段階と、再プログラミングが完了した後、アプリケーションのキー値を書き込む段階とをさらに含むことができる。

【0015】

MCUのエラー復旧方法は、確認段階で、アプリケーションのキー値が有効であれば、対応するアプリケーションを実行する段階と、対応するアプリケーションの実行中に、再プログラミング要請を受信する段階と、再プログラミング要請を受信すれば、対応するアプリケーションのキー値を消す段階とをさらに含むことができる。

40

【0016】

MCUのエラー復旧方法は、アプリケーションのキー値を消した後、フラッシュブートローダーがアプリケーションキーブロックからアプリケーションのキー値を確認する段階から始まることができる。

【0017】

本発明のさらに他の特徴によるバッテリー装置は、電氣的に連結された複数のバッテリーセルを含むバッテリーセル組立体と、バッテリー管理のためのアプリケーションを保存したフラッシュメモリを含むマイクロコントローラユニット (Micro Controller Unit、MCU) と、MCUで行われるアプリケーションによりバッテリー管理を行うバ

50

ッテリ管理システムとを含む。フラッシュメモリは、MCUのパワーオン時に、アプリケーションのキー値を確認し、アプリケーションのキー値が有効でなければ、再プログラミング可能なモードに入ることができる。

【発明の効果】

【0018】

フラッシュメモリのエラーを復旧できる方法、これを用いたマイクロコントローラユニット、およびマイクロコントローラユニットを含むバッテリー装置を提供する。

【図面の簡単な説明】

【0019】

【図1】一実施例によるマイクロコントローラユニットを示す図である。

10

【図2】一実施例によるフラッシュメモリ2の構成を示す図である。

【図3】一実施例による復旧方法を示すフローチャートである。

【図4】一実施例によるバッテリー装置を示す図である。

【発明を実施するための形態】

【0020】

以下、添付した図面を参照して、本発明の実施例について、本発明の属する技術分野における通常の知識を有する者が容易に実施できるように詳しく説明する。しかし、本発明は種々の異なる形態で実現可能であり、ここで説明する実施例に限定されない。そして、図面において本発明を明確に説明するために説明上 unnecessaryな部分は省略し、明細書全体にわたって類似の部分については類似の図面符号を付した。

20

【0021】

図1は、一実施例によるマイクロコントローラユニットを示す図である。

【0022】

図1に示されているように、マイクロコントローラユニット (Micro Controller Unit、MCU) 1は、フラッシュメモリ2と、プロセッサ3と、周辺装置4とを含む。

【0023】

フラッシュメモリ2は、MCU1が適用される装置に必要なアプリケーションを保存し、アプリケーションのソフトウェアを実行させ、要請があった場合、ソフトウェアを再プログラミング (reprogramming) し、ECCエラーによるCPUコア例外の発生時、エラーの発生位置がアプリケーションキー領域の場合、アプリケーションキー領域を消し、リセットして再プログラミング可能な状態で動作できる。

30

【0024】

プロセッサ3は、フラッシュメモリ2および周辺装置4とインターフェースを介してネットワークングし、周辺装置4を介して入力された情報をフラッシュメモリ2に伝送でき、フラッシュメモリ2および周辺装置4の動作を制御する命令 (command) を伝送できる。MCU1がパワーオンされると、プロセッサ3にパワーオン信号P_ONが入力され、プロセッサ3は、フラッシュメモリ2にアプリケーション実行のための命令を伝達できる。

【0025】

周辺装置4は、CAN通信、UART通信などのための通信モジュールを含み、プロセッサ3に再プログラミング要請および再プログラミングのためのソフトウェアコードを伝送できる。

40

【0026】

以下、図2を参照して、一実施例によるフラッシュメモリを説明する。

【0027】

図2は、一実施例によるフラッシュメモリ2の構成を示す図である。

【0028】

図2に示されているように、フラッシュメモリ2は、フラッシュブートローダー (Flash Boot Loader、FBL) 21と、アプリケーションブロック22と、ア

50

アプリケーションキーブロック 23 とを含む。

【0029】

FBL21 に命令がとどまる FBL モードで、FBL21 は、アプリケーション領域のソフトウェアを消し、再プログラミングできる。FBL21 は、プロセッサ 3 から伝達された命令によりアプリケーションのキー値が有効であるかを確認できる。FBL21 の確認結果、アプリケーションのキー値が有効でない場合、アプリケーションブロック 22 のアプリケーションは実行されず、FBL モードに入る。FBL21 は、プロセッサ 3 にアプリケーションのキー値が有効でないことを知らせることができる。

【0030】

FBL21 によってアプリケーションのキー値が有効であると判断されれば、命令によりアプリケーションブロック 22 がアプリケーションの領域に保存されたソフトウェアを動作させることができる。動作中に、プロセッサ 3 から再プログラミング要請がフラッシュメモリ 2 に伝送されれば、アプリケーションキーブロック 23 は、動作中のアプリケーションに対応するアプリケーションキー領域を消し、リセットした後、FBL モードに入る。

10

【0031】

このように、アプリケーションのキー値が有効でないか、再プログラミング要請があった場合、再プログラミングのために FBL21 に命令がとどまる状態である FBL モードに入る。

【0032】

FBL21 は、CAN 通信または UART 通信により命令に対応するアプリケーションに対するデータを、周辺装置 4 を介して受信し、受信されたデータをアプリケーションブロック 22 領域に再プログラミングする。受信されたデータが再プログラミングによりアプリケーション領域に書き込まれた (writing) 後、アプリケーションキーブロック 23 は、アプリケーションに対応する領域に有効なキー値を書き込む (writing)。アプリケーションキーブロック 23 は、書き込み後にリセットを行い、リセットによりアプリケーションブロック 22 が再プログラミングされたアプリケーションを実行できる。

20

【0033】

このような動作とともに、FBL21 は、ECC エラーによる CPU コア例外の発生時、ECC エラーの発生した位置がアプリケーションキーブロック 23 である場合、アプリケーションのキー値を消し、リセットして FBL モードに入る。つまり、フラッシュメモリ 2 は FBL モード状態で、再プログラミング可能な状況に復旧される。

30

【0034】

このような動作を行うために、FBL21、アプリケーションブロック 22、およびアプリケーションキーブロック 23 は、相互間で必要な情報を送受信できる。

【0035】

図 3 は、一実施例による復旧方法を示すフローチャートである。

【0036】

図 3 に示されているように、MCU1 がパワーオン (power-on) またはリセットされる (段階 S1)。

40

【0037】

FBL21 は、パワーオン後、プロセッサ 3 の命令により実行されるべきアプリケーションのキー値が有効であるかを確認できる (段階 S2)。

【0038】

段階 S2 の判断結果、アプリケーションのキー値が有効な場合、命令は、FBL21 からアプリケーションブロック 22 に移動できる (段階 S3)。

【0039】

アプリケーションブロック 22 は、命令によりアプリケーションを実行できる (段階 S4)。

50

【0040】

アプリケーションの実行中に、再プログラミング要請があるかを判断できる（段階S5）。

【0041】

段階S5の判断結果、プロセッサ3から再プログラミング要請がフラッシュメモリ2に伝送されれば、アプリケーションキーブロック23は、アプリケーションのキー値を消すことができる（段階S6）。

【0042】

アプリケーションのキー値が消された後、ソフトリセット（soft reset）が行われる（段階S7）。ソフトリセットによりアプリケーションの実行が中断される。

【0043】

段階S5の判断結果、再プログラミング要請がなければ、段階S4によりアプリケーションの実行が維持できる。

【0044】

段階S7の後、段階S1によりFBLモードに入り、再びS2段階から実行できる。すると、アプリケーションのキー値が有効でないので、FBLモードに入って、再プログラミングが行われて有効なアプリケーションのキー値が書き込まれる。

【0045】

段階S2の判断結果、アプリケーションのキー値が有効でない場合、命令は、FBL21にとどまる（stay）（段階S8）。つまり、FBLモードに入る。同時に、ECCエラーによるCPUコア例外が発生しうる。

【0046】

プロセッサ3から再プログラミングのためのデータがフラッシュメモリ2に受信されれば、FBL21は、アプリケーション領域を再プログラミングできる（段階S9）。

【0047】

アプリケーションブロック22は、再プログラミングが完了したかを判断できる（段階S10）。

【0048】

段階S10の判断結果、再プログラミングが完了していなければ、段階S9の実行が維持できる。

【0049】

段階S10の判断結果、再プログラミングが完了したならば、アプリケーションキーブロック23が有効なアプリケーションのキー値を書き込むことができる（段階S11）。

【0050】

段階S11の後、ソフトリセット（soft reset）が行われる（段階S12）。ソフトリセット後、段階S1によりFBLモードに入り、再びS2段階から実行できる。

【0051】

本開示の一実施例によれば、アプリケーションのキー値が有効でない場合、再プログラミング可能なFBLモードに入ることによって、従来技術においてECCエラーによるCPUコア例外が発生した時、パワーオンと有効キー値を読み出す過程の間に無限反復が発生する問題を解決できる。

【0052】

以下、先に説明した一実施例によるMCUが適用されたバッテリー装置に関する実施例を説明する。

【0053】

図4は、一実施例によるバッテリー装置を示す図である。

【0054】

図4に示されているように、バッテリー装置100は、MCU120を含み、MCU120は、バッテリーのモニタリングおよび管理に必要な複数のアプリケーションを保存できる。MCU120で実行されるアプリケーションにより、バッテリー装置100が駆動できる

10

20

30

40

50

。例えば、MCU120は、バッテリーパックの充電状態および寿命を推定するためのアプリケーション、バッテリーパックおよびバッテリーセルの電流、電圧、温度などの状態情報を検知するためのアプリケーションなどを含むことができる。

【0055】

一例として、MCU120でバッテリーパックの充電状態(State of Charge、SOC)を推定するためのアプリケーションが実行され、バッテリーパックの充電状態推定のためのバッテリーの状態情報を検知するためのアプリケーションが実行できる。

【0056】

SOC推定のためのアプリケーションが動作する時の一例を、以下に説明する。

【0057】

図4に示されているように、バッテリー装置100は、バッテリー管理システム(Battery Management System、BMS)110と、MCU120と、バッテリーセル組立体130と、リレー140、150とを含む。

【0058】

バッテリーセル組立体130は、電氣的に連結されている複数のバッテリーセルを含む。所定数のバッテリーセルが直列連結されてバッテリーモジュールを構成し、所定数のバッテリーモジュールが直列および並列連結されて所望の電力を供給することができる。

【0059】

バッテリーセル組立体130を構成する複数のバッテリーセルそれぞれは、BMS110と複数の配線を介して電氣的に連結されている。

【0060】

BMS110は、複数のバッテリーセルに関する情報を含むバッテリー装置に関する多様な情報を取り合いおよび分析してバッテリー装置の充放電、バッテリーセルバランシング、保護動作などを制御できる。

【0061】

リレー140、150は、リレーやnチャネルタイプのトランジスタなどで実現可能であり、バッテリーセル組立体130と外部装置との間に直列連結されている。リレー140、150それぞれは、BMS110から供給される信号によって動作できる。

【0062】

MCU120は、先に説明した一実施例により実現できる。MCU120でアプリケーションが行われ、BMS110にアプリケーションによる制御信号が伝送される。すると、BMS110は、MCU120で行われるアプリケーションにより動作できる。

【0063】

例えば、MCU120に保存された複数のアプリケーションのうちSOC推定アプリケーションが行われると、BMS110は、アプリケーションによりSOCを推定する。SOCの推定方式は、アプリケーションにより決定される。

【0064】

一例として、SOC推定アプリケーションは、第1バッテリーモデリング方式を利用してSOCを推定でき(以下、第1SOC)、第2バッテリーモデリング方式を利用するSOCを推定でき(以下、第2SOC)、第1および第2バッテリーモデリング方式は互いに異なってもよい。バッテリーモデリング方式は、非線形的な特性を有するバッテリーの状態を推定するために利用されるバッテリーモデリング方法であって、電気回路モデル(Electrical Circuit Model)、電気化学モデル(Electrochemical Model)、分析的モデル(Analytical Model)および確率的モデル(Stochastic Model)などが含まれる。

【0065】

具体的には、BMS110は、SOC推定アプリケーションにより、第1バッテリーモデリング方式として電気回路モデルを用いて直前段階で推定されたSOCおよびバッテリーパックの状態情報に含まれている情報に基づいて現段階の第1SOCを推定できる。第2バッテリーモデリング方式として電気化学モデルが利用可能である。

10

20

30

40

50

【0066】

電気回路モデルは、電気回路で実現された等価回路でバッテリーの入力および出力特性をモデリングする方法である。電気回路モデルは、SOC推定のための演算過程が比較的簡単に演算による所要時間が長くなく、演算のための負荷が多くかからないという利点を有する。しかし、このような電気回路モデルの場合、正確性にやや劣る問題がある。

【0067】

これに対し、電気化学モデルは、バッテリーの内部で発生する化学的作用に基づいてバッテリーの特性をモデリングする方法である。このような電気化学モデルの代表例として、DFN (Doyle-Fuller-Newman) モデルが挙げられる。DFNモデルは、多孔性の電極内に存在するリチウムイオン濃度の時空間的な変化、電位、インターカレーションキネティック (intercalation kinetic)、固体相と電解液相との間の電流密度などをモデリングすることができる。このような電気化学モデルは、正確性が非常に高いという利点を有する。

10

【0068】

BMS110は、SOC推定アプリケーションによりDFNモデルのような電気化学モデルを用いてバッテリーパックの状態情報に基づいた正確度の高いSOC推定値 (第2SOC) を得ることができる。そして、BMS110は、SOC推定アプリケーションによりこのように高い正確度を有するSOC推定結果 (第2SOC) を入力パラメータに反映することによって、SOC (第1SOC) 推定の正確度を高めることができる。

【0069】

例えば、BMS110は、SOC推定アプリケーションにより第2SOC値を直前段階の第1SOCとして認識し、受信されたバッテリーパックの状態情報および直前段階の第1SOCを第1バッテリーモデリング方式に適用して現段階の第1SOCを推定できる。すると、第1SOC値が周期的に補正されて、第1SOC推定の誤差が大きくなるのを周期的に防止でき、これによってSOC推定の正確性を向上させることができる。

20

【0070】

以上、本発明の実施例について詳しく説明したが、本発明の権利範囲はこれに限定されるものではなく、以下の特許請求の範囲で定義している本発明の基本概念を利用した当業者の様々な変形および改良形態も本発明の権利範囲に属する。

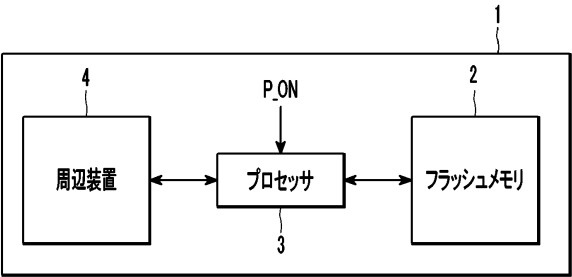
30

40

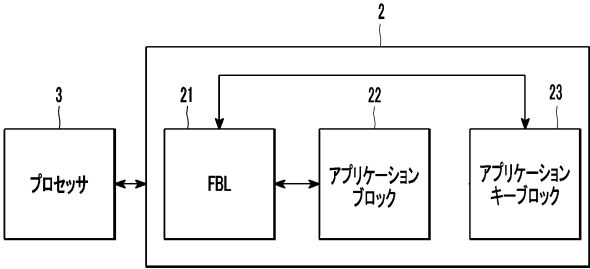
50

【図面】

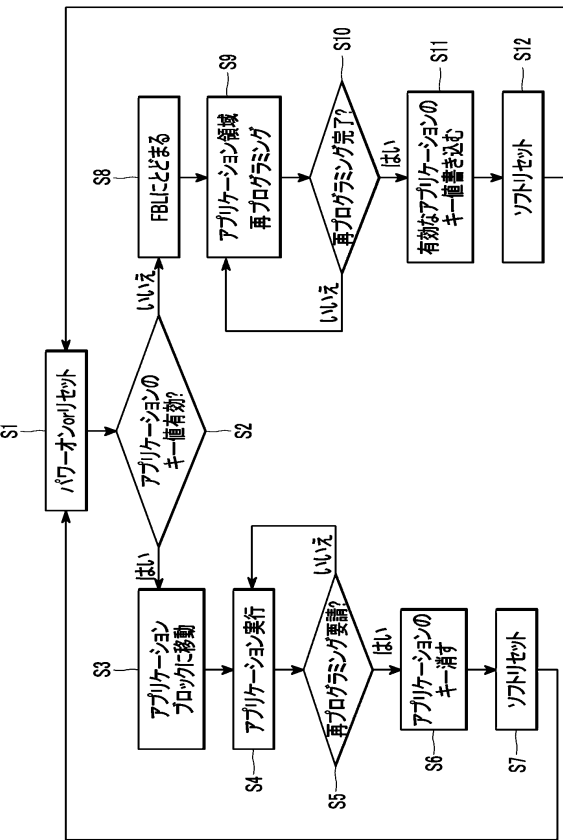
【図 1】



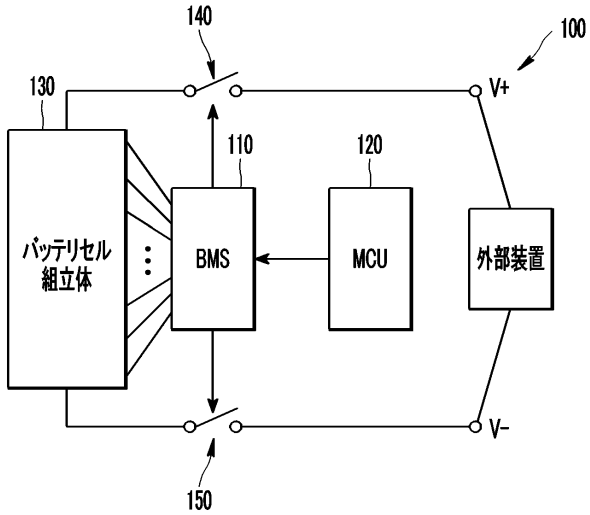
【図 2】



【図 3】



【図 4】



10

20

30

40

50

フロントページの続き

- (56)参考文献 米国特許出願公開第2017/0024221 (US, A1)
米国特許出願公開第2008/0086631 (US, A1)
特開2011-164827 (JP, A)
特開2011-83147 (JP, A)
- (58)調査した分野 (Int.Cl., DB名)
G06F 15/78
H02J 7/02
G06F 12/00
G06F 8/65