

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 022 540**

51 Int. Cl.:

G06F 8/71 (2008.01)

G06F 8/65 (2008.01)

H01M 10/42 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **10.06.2022 PCT/KR2022/008244**

87 Fecha y número de publicación internacional: **05.01.2023 WO23277386**

96 Fecha de presentación y número de la solicitud europea: **10.06.2022 E 22833442 (1)**

97 Fecha y número de publicación de la concesión europea: **05.03.2025 EP 4191404**

54 Título: **Método para actualizar software mediante confirmación de información de versión de hardware**

30 Prioridad:

28.06.2021 KR 20210084131

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

28.05.2025

73 Titular/es:

**LG ENERGY SOLUTION, LTD. (100.00%)
Tower 1, 108, Yeoui-daero, Yeongdeungpo-gu
Seoul 07335, KR**

72 Inventor/es:

**JUNG, JIN KYU y
YI, YEON OK**

74 Agente/Representante:

VEIGA SERRANO, Mikel

ES 3 022 540 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Método para actualizar software mediante confirmación de información de versión de hardware

5 **Sector de la técnica**

La presente invención se refiere a un método de confirmación de una versión de hardware para actualización de software cuando se realiza una actualización de software de un BMS. La presente invención se refiere a un dispositivo y un método para aumentar la utilización de los canales de los microcontroladores al tiempo que se identifican con precisión las versiones de hardware.

Estado de la técnica

15 La versión de hardware varía en función de los dispositivos de aplicación y los circuitos que constituyen el sistema de gestión de baterías (BMS). Para compensar los problemas o pérdidas que se produjeron en la versión anterior, la versión de hardware se actualiza a medida que se fabrica un nuevo BMS complementando la fabricación de la placa de circuito impreso y los dispositivos de aplicación del BMS. Esto también puede denominarse actualización del hardware.

20 La actualización de la versión de software procede según la actualización del diseño del controlador digital para el funcionamiento del circuito de control y protección de cada aplicación del BMS. Esto también puede denominarse actualización del software.

25 Existe una versión de hardware a la que puede aplicarse una versión de software, y existe una versión de hardware que puede aplicarse incluso al actualizar el software. Por tanto, al actualizar el software, debe comprobarse la versión del hardware.

30 La literatura anterior de la patente coreana n.º 10-0624723 (8 de septiembre de 2006) es una técnica anterior de la presente invención, y almacena la versión del hardware en el código del cargador de arranque, y cuando se comprueba la versión del hardware, se comprueba a través del código del cargador de arranque.

Por tanto, si se desea comprobar la versión del hardware al actualizar el software, accediendo al código del cargador de arranque y comparándolos, se comprueba el hardware y, tras ello, se procederá a la actualización del software.

35 En este caso, como la versión del hardware se introduce directamente y se almacena en el código del cargador de arranque, pueden producirse errores y accidentes debido a entradas incorrectas.

(Documento de patente 1) Registro de patentes coreano 10-0624723 (2006.09.08.).

40 El documento KR 2013 0045508 A se refiere a un método de instalación de software para instalar información de versión de un dispositivo de gestión de baterías e instalar el software correspondiente almacenando la información de versión en el dispositivo integrado.

Objeto de la invención

45

Problema técnico

50 Para resolver los problemas anteriores, la presente invención proporciona un dispositivo y un método para actualizar el software de un sistema de gestión de baterías (BMS) utilizando información de versión de hardware almacenada en un dispositivo de identificación de información de versión de hardware independiente, de forma que no se produzca un error durante la identificación de versiones del hardware.

Solución técnica

55 Para lograr el objeto anterior, la presente invención proporciona un dispositivo de gestión de baterías según la reivindicación 1.

60 Además, el microcontrolador está configurado para incluir un terminal de control para emitir una orden de control de encendido/apagado a los conmutadores primero y segundo, en el que el terminal de control emite una señal de control para encender el primer conmutador y apagar el segundo conmutador en un modo normal o en un modo de funcionamiento de aplicación.

65 Además, cuando se recibe el software de actualización, el microcontrolador compara una lista de versiones de hardware compatibles incluida en el software de actualización con la información de versión de hardware recibida de la unidad de identificación de versiones de hardware, y controla el encendido/apagado de los conmutadores primero y segundo según un resultado de comparación.

Según la reivindicación 4, se proporciona un método de control de un dispositivo de gestión de baterías según un segundo aspecto de la presente invención.

5 Efectos ventajosos

En la presente invención, mediante el uso de un dispositivo de identificación de información de versión de hardware independiente que almacena información de versión de hardware, es posible evitar errores debidos a la introducción de una versión de hardware incorrecta durante la actualización del software.

Además, estableciendo un canal de aplicación predeterminado como canal designado y conectando el elemento de aplicación y el elemento que tiene información de versión de hardware a un canal designado, se proporcionan un método y un dispositivo que eliminan la necesidad de comprobar la versión a través del código de arranque durante la actualización del software y no requieren una gestión de versión de hardware independiente al realizar el modo de actualización de software y el modo de aplicación a través de un canal.

Descripción de las figuras

La FIG. 1 muestra una placa 100 de BMS según un ejemplo útil para la comprensión de la presente invención.

La FIG. 2 muestra una placa 200 de BMS según una realización de la presente invención.

La FIG. 3 es una vista que muestra una lista de versiones de hardware compatibles con la actualización del software.

La FIG. 4 es un diagrama de flujo de una actualización de software de la presente invención.

Descripción detallada de la invención

En lo sucesivo, se describirán en detalle las realizaciones de la presente invención con referencia a los dibujos adjuntos, de modo que los expertos en la técnica puedan aplicar fácilmente la presente invención. Sin embargo, la presente invención puede implementarse de diversas formas y no se limita a las realizaciones descritas en el presente documento. En los dibujos se omiten partes irrelevantes para la descripción con el fin de describir con claridad la presente invención, y los números de referencia similares se refieren a elementos similares en toda la memoria descriptiva.

1. Técnica anterior

A continuación se describirá en detalle la actualización de versión de hardware, por ejemplo, en un sistema BMS. El sistema BMS consta de una serie de elementos que constituyen un dispositivo de medición de temperatura, un microcontrolador, un circuito de protección y un convertidor de potencia, y estos elementos constituyen el circuito. En este momento, la pérdida del sistema BMS aumenta o la estabilidad se determina en función de la mejora del rendimiento de cada elemento o de la configuración del circuito. Para mejorar este problema, a medida que se modifican algunos elementos y circuitos del sistema, se cambia el hardware, lo que se denomina actualización de versión de hardware.

La mejora de la forma en que los conmutadores controlan y codifican el funcionamiento de un sistema de hardware se denomina actualización de software.

Por esta razón, convencionalmente, para la actualización del software, la propia versión del hardware se almacena en el código de arranque (código es un término genérico para el lenguaje de programación que hace funcionar el microcontrolador) y se almacena en la memoria flash del microcontrolador. En este momento, dado que la versión de hardware se introduce y gestiona directamente en el código de arranque, es posible que el sistema de hardware no funcione correctamente debido a un error en la versión, lo que puede provocar un accidente. Además, al estar almacenada en la memoria del microcontrolador, existe el problema de que es difícil comprobarla de vez en cuando.

2. Dispositivo para comprobar la versión del hardware de la presente invención

La FIG. 1 muestra un ejemplo útil para la comprensión de la presente invención en el que la información de versión de hardware del BMS se registra en el hardware mejorando el problema descrito anteriormente, y luego se proporciona directamente a través del canal del microcontrolador cuando se actualiza el software.

Una unidad 130 de identificación de versiones de hardware está configurada para proporcionar la información de versión de hardware del BMS desde la unidad 130 de identificación de versiones de hardware cuando se actualiza el software, y comparar la información de versión de hardware recibida con la lista de versiones de hardware compatibles incluidas en el software para la actualización con el fin de determinar si debe procederse con la

actualización del software. Es decir, la lista de versiones de hardware es una lista de versiones de hardware para las que debe procesarse el software.

El procedimiento de actualización del software sigue el mismo procedimiento que en la segunda realización que se describirá más adelante.

Por otro lado, el microcontrolador tiene un número limitado de canales, y a medida que se actualiza la versión de hardware del BMS, se necesitan más canales del microcontrolador. El método analógico de introducir directamente la versión de hardware a través del microcontrolador utiliza el canal del microcontrolador solo en el modo de actualización del software (o modo de arranque), por lo que existe el problema de que la utilización del canal es baja.

Para superar el problema de utilización del canal, la realización de la presente invención tiene una configuración de circuito como se muestra en la FIG. 2.

La realización de la FIG. 2 se refiere a un método para aumentar la utilización del canal mientras se recibe información de versión de hardware a través de un canal de un microcontrolador.

Se trata de una invención que permite tanto la identificación de la versión de hardware en el modo de arranque como la ejecución de la aplicación en el modo de aplicación mediante el montaje de dos conmutadores 140 y 150 que funcionan de forma complementaria con un canal del microcontrolador y que controlan el funcionamiento en función de cada modo.

Según la presente invención, el dispositivo de gestión de baterías incluye un primer conmutador para la entrada y salida de información según las operaciones de un microcontrolador, una unidad de ejecución de aplicación, una unidad de identificación de versiones de hardware, y un segundo conmutador para el funcionamiento de la unidad de identificación de versiones de hardware.

2.1. Microcontrolador (MCU) 110

Un microcontrolador (MCU) 110 es un dispositivo de procesamiento aritmético ultrapequeño y es un dispositivo que recibe y almacena datos como la corriente de la batería, el voltaje, la temperatura, etc. detectados por múltiples sensores que comprueban el SOC de la batería en el BMS y permite el funcionamiento de un circuito de protección y el control de encendido/apagado de la batería.

En consecuencia, para mejorar la función de control del BMS, se actualiza el software del microcontrolador 110.

Además, el microcontrolador 110 sirve para controlar un modo de arranque y un modo de aplicación para la actualización del software del BMS. (El modo de aplicación es un modo en el que funcionan generalmente las aplicaciones necesarias para el BMS).

Se trata de un dispositivo que ordena el funcionamiento de un elemento de aplicación a través del canal 111 designado o transmite una señal de control al terminal 170 de control.

Además, el microcontrolador recibe el software para la actualización cuando se actualiza la versión del software. El software para la actualización incluye una lista de versiones de hardware compatibles (también denominada lista de versiones de hardware compatibles). La FIG. 3 muestra una lista 300 de versiones de hardware compatibles recibida por el microcontrolador en la presente invención.

Tras recibir la información de versión de hardware introducida a través de la unidad 130 de identificación de versiones de hardware que se describirá más adelante desde el microcontrolador a través del canal 111 designado, la información de versión de hardware recibida se compara con la lista 300 de versiones de hardware. La unidad 130 de identificación de versiones de hardware almacena la información de versión de hardware de un BMS que incluye un microcontrolador que debe actualizarse mediante software, y emite la información de versión de hardware al microcontrolador según una operación de encendido de un segundo conmutador que se describirá más adelante.

En consecuencia, si la versión de hardware introducida a través de la unidad 130 de identificación de versiones de hardware no existe en la lista 300 de versiones de hardware, no se procede a la actualización del software.

Por el contrario, si la versión de hardware introducida a través de la unidad 130 de identificación de versiones de hardware existe en la lista 300 de versiones de hardware, se procede a la actualización del software.

Un dispositivo que realiza este proceso es el microcontrolador 110.

2.2. Unidad de ejecución de aplicación 120

La unidad de ejecución de aplicación es un dispositivo que está conectado al microcontrolador (MCU) 110 y que

opera el elemento de aplicación. La unidad 120 de ejecución de aplicación es un dispositivo que funciona en el modo de aplicación (o modo normal), y es un dispositivo que se detiene cuando se introduce una orden de actualización del software y vuelve a funcionar cuando se completa la actualización del software. La unidad 120 de ejecución de aplicación está conectada al primer conmutador 140 para introducir información o emitir una operación al microcontrolador a través del canal 111 designado.

2.3. Primer conmutador 140

Se trata de un conmutador para conectar el canal designado de la unidad de ejecución de aplicación y el microcontrolador, y es un dispositivo que permite conectar la unidad 120 de ejecución de aplicación y el microcontrolador 110 en un modo de aplicación (o modo normal) en el que la actualización del software se completa o no se realiza.

2.4. Unidad de identificación de versiones de hardware 130

La unidad 130 de identificación de versiones de hardware es un dispositivo que proporciona datos sobre la versión de hardware al microcontrolador 110. La unidad 130 de identificación de versiones de hardware almacena información de hardware del BMS y está conectada a un segundo conmutador 150 que se describirá más adelante, y cuando se introduce la orden de actualización del software, el segundo conmutador 150 se enciende y se proporciona información de versión de hardware del BMS al microcontrolador 110. La diferencia en la primera realización es que la unidad 130 de identificación de versiones de hardware emite directamente la información de versión de hardware al microcontrolador a través de un canal independiente, como se muestra en la FIG. 1.

La unidad 130 de identificación de versiones de hardware puede introducir en el microcontrolador 110 una combinación de señales altas o bajas codificadas según la información de versión de hardware, o introducir información de versión de hardware en el microcontrolador 110 cambiando el valor de corriente aplicado al microcontrolador 110.

La salida de la unidad 130 de identificación de versiones de hardware está conectada a un canal del microcontrolador y emite la versión del BMS al microcontrolador, y puesto que en la primera realización está configurada para ocupar un canal independiente y la identificación de versiones solo se utiliza para las actualizaciones del software, en la segunda realización, en términos de utilización de recursos del canal, el circuito está configurado de tal manera que se comparte el canal del microcontrolador ocupado por otras aplicaciones.

2.5. Segundo conmutador 150

Es un conmutador que funciona cuando se aplica externa o internamente una orden de actualización del software del BMS. Conecta eléctricamente el canal 111 designado del microcontrolador y la unidad 130 de identificación de versiones de hardware, y permite que la información de versiones de hardware aplicada desde la unidad 130 de identificación de versiones de hardware se introduzca en el microcontrolador 110 solo durante la actualización del software. El primer conmutador 140 y el segundo conmutador 150 funcionan de forma complementaria entre sí, y cuando el segundo conmutador 150 se enciende, el primer conmutador 140 se apaga.

Un extremo del segundo conmutador 150 está conectado a la unidad 130 de identificación de versiones de hardware, y el otro extremo está conectado a la parte 160 de conexión.

2.6. Terminal de control 170

Sirve para controlar el primer conmutador 140 y el segundo conmutador 150, y emite una señal de control para apagar el primer conmutador 140 y encender el segundo conmutador 150 a través de un terminal 170 de control cuando se introduce una orden de actualización del software desde el microcontrolador 110.

Se trata de un terminal para emitir una señal de control para encender el primer conmutador 140 y apagar el segundo conmutador 150 durante el funcionamiento del modo de aplicación.

2.7. Parte de conexión 160 del primer conmutador y del segundo conmutador

Uno de los extremos del primer conmutador 140 y del segundo conmutador 150 están conectados respectivamente a la unidad 120 de ejecución de aplicación y a la unidad 130 de identificación de versiones de hardware, y el otro extremo de cada uno del primer conmutador 140 y el segundo conmutador 150 está conectado a un extremo de la parte 160 de conexión.

El otro extremo de la parte 160 de conexión se conecta a un canal 111 designado del microcontrolador 110.

Mediante una configuración de circuito de este tipo, en función del encendido/apagado del primer conmutador 140 y del segundo conmutador 150, uno de los datos de la unidad 120 de ejecución de aplicación y de la unidad de

identificación de versiones de hardware se introduce selectivamente en el canal 111 del microcontrolador 110.

Según una realización de la presente invención, el conmutador se controla de manera que en el modo normal o modo de aplicación del dispositivo de gestión de baterías, el primer conmutador 140 se controla encendido, y las señales y la información se introducen/emiten a la unidad 120 de ejecución de aplicación por el canal 111 designado, y en el modo de arranque, el primer conmutador 140 se controla apagado y el segundo conmutador 150 se controla encendido y así la información de versiones de hardware introducida a través de la unidad 130 de identificación de versiones de hardware se introduce en el microcontrolador 110 a través del canal 111 designado.

3. Método de comprobación de versiones de hardware de la presente invención

La presente invención se refiere a un método de comprobación de una versión de hardware cuando se realiza una actualización de software de un microcontrolador. La FIG. 4 muestra un ejemplo del método de la presente invención.

Un modo para realizar una actualización de software se denomina modo de arranque, y un modo en el que cada aplicación funciona según una situación necesaria se denomina modo de aplicación (o modo normal).

3.1. Etapa de ejecución de modo de aplicación S410

Se trata de una etapa de entrada o salida de información de aplicación mediante el encendido del primer conmutador 140 de la FIG. 2 para ejecutar una aplicación mediante la unidad de ejecución de aplicación.

Se trata de una etapa en la que el segundo conmutador está APAGADO, de modo que la información de versiones de hardware proporcionada desde la unidad de identificación de versiones de hardware no se transmite al microcontrolador.

3.2. Etapa de introducción de instrucciones de actualización de software S420

Se trata de una etapa en la que se introduce una orden para proceder a la actualización del software del BMS desde el exterior o el interior y, en consecuencia, es una etapa para prepararse para el funcionamiento en modo de arranque. Esta etapa de introducción de la orden de actualización del software recibe el software de actualización para la actualización del software del BMS, que también se denomina etapa de recepción de software de actualización.

En esta etapa se apaga el primer conmutador 140 y se enciende el segundo conmutador 150. El encendido/apagado de los conmutadores primero y segundo se controla a través del terminal 170 de control del microcontrolador anteriormente descrito.

Además, es una etapa de recepción de la lista 300 de versiones de hardware para actualizarse con la orden de actualización de software introducida junto con la orden. En este momento, la lista 300 de versiones de hardware (también denominada lista de versiones de hardware compatibles) se almacena como un archivo de imagen para facilitar el acceso y la gestión.

3.3. Etapa de introducción de versión de hardware S430

Se trata de una etapa en la que la versión de hardware de la unidad 130 de identificación de versiones de hardware se introduce en el microcontrolador 110 cuando se enciende el segundo conmutador 150. Se trata de una etapa en la que la información de versión de hardware del BMS almacenada en la unidad 130 de identificación de versiones de hardware se introduce en el microcontrolador 110 a través del canal 111 designado. La etapa de introducción de versión de hardware es para obtener información de versión de hardware del BMS, y también se denomina etapa de adquisición de información de versión de hardware.

El método de identificación de versiones del hardware transmite una señal de identificación de hardware al microcontrolador 110 mediante la emisión de una corriente o voltaje altos/bajos, y un cambio de valor de corriente a través de una unidad de identificación de versiones de hardware.

En la primera realización, al recibir la orden de actualización del software descrita anteriormente, la información de versión de hardware se recibe directamente desde un canal conectado a la unidad 130 de identificación de versiones de hardware.

3.4. Etapa de comprobación de lista de versiones de hardware S440

Se trata de una etapa para determinar si se debe proceder a la actualización del software comparando la información de versión de hardware del BMS introducida en el canal 111 designado a través de la etapa de introducción de versión de hardware descrita anteriormente con la lista 300 de versiones de hardware incluida en el

archivo de actualización de software recibido.

La lista 300 de versiones de hardware incluida en el archivo de actualización de software puede recibirse por adelantado y almacenarse como un archivo de imagen en un área predeterminada del microcontrolador.

Cuando la información de versión de hardware adquirida en la etapa de introducción de versión de hardware o en la etapa de adquisición de información de versión de hardware se encuentra en la lista de versiones de hardware, se inicia el modo de arranque.

Si la información de versión de hardware no se encuentra en la lista de versiones de hardware, conmuta al modo de aplicación.

Si la versión de hardware autorizada a través del canal 111 designado se encuentra en la lista 300 de versiones de hardware, se realiza la actualización del software.

Para proceder a la actualización del software, en la etapa de inicio del modo de arranque posterior, el primer conmutador 140 mantiene un estado apagado y el segundo conmutador 150 mantiene un estado encendido.

Si la información de versión de hardware autorizada a través del canal 111 designado no existe en la lista 300 de versiones de hardware, no se realiza la actualización del software. En consecuencia, el segundo conmutador 150 se apaga y el primer conmutador 140 se enciende para proceder al modo de aplicación (o modo normal). La información de versión de hardware ya no se transmite al microcontrolador 110.

3.5. Conmutación del modo de arranque

Cuando la información de versión de hardware obtenida a través de la etapa de introducción de versión de hardware existe en la lista de versiones de hardware a actualizar, se realiza la actualización del software.

3.5.1. Etapa de inicio del modo de arranque S450

Se trata de una etapa en la que el software y el hardware del BMS se conmutan a un modo para la actualización del software, y solo se ponen en funcionamiento los dispositivos necesarios para la actualización del software, y los dispositivos no necesarios para la actualización del software se conmutan al modo de reposo (o se apagan). A continuación, se procede a la actualización del software del BMS.

Cuando se introduce información sobre el inicio del modo de arranque en la etapa S450 de inicio del modo de arranque, el microcontrolador 110 ordena la conmutación de modo según el modo de arranque. A través del terminal 170 de control, se controla el primer conmutador 140 para mantenerlo encendido, y se controla el segundo conmutador 150 para mantenerlo apagado. Además, otras aplicaciones y software también proceden al modo de arranque.

3.5.2. Etapa de finalización del modo de arranque S460

Se trata de una etapa para detener el modo de arranque cuando se recibe desde el interior del microcontrolador 110 una orden que indica que se ha completado la actualización del software del BMS.

A través del terminal 170 de control, el segundo conmutador 150 se apaga y el primer conmutador 140 se enciende.

3.6. Etapa de conmutación del modo de aplicación S470

Tras salir del modo de arranque, se inicia el modo de aplicación original (o modo normal). En consecuencia, el segundo conmutador 150 se apaga, y el canal 111 designado del microcontrolador se conecta a la unidad 120 de ejecución de aplicación para introducir o emitir la información necesaria para la ejecución de la aplicación.

Por otra parte, aunque la idea técnica de la presente invención se ha descrito específicamente según la realización anterior, debe tenerse en cuenta que las realizaciones anteriores son a título explicativo y no limitativo.

REIVINDICACIONES

1. Un dispositivo (200) de gestión de baterías que comprende:

una unidad (130) de identificación de versiones de hardware que incluye información de versión de hardware de un sistema de gestión de baterías y está configurada para emitir la información de versión de hardware; y

un microcontrolador (110) que incluye una pluralidad de canales de entrada/salida, y está configurado para recibir la información de versión de hardware de la unidad (130) de identificación de versiones de hardware a través de al menos un canal de entrada/salida,

en el que el microcontrolador (110):

recibe un software de actualización que incluye una lista (300) de versiones de hardware compatibles, y

compara la lista (300) de versiones de hardware compatibles con la información de versión de hardware del sistema de gestión de baterías recibida de la unidad (130) de identificación de versiones de hardware para determinar si se debe proceder a la actualización del software del sistema de gestión de baterías,

en el que al menos un canal de entrada/salida del microcontrolador (110) está configurado como canal (111) designado,

en el que el canal (111) designado está conectado simultáneamente a un primer conmutador (140) que tiene un extremo conectado a una unidad (120) de ejecución de aplicación predeterminada y un segundo conmutador (150) que tiene un extremo conectado a la unidad (130) de identificación de versiones de hardware,

en el que cada uno del primer conmutador (140) y el segundo conmutador (150) tiene un estado de encendido/apagado establecido de forma diferente según un control del microcontrolador (110);

en el que, cuando hay una instrucción de actualización de software, el microcontrolador (110) enciende el segundo conmutador (150) y apaga el primer conmutador (140) para recibir la información de versión de hardware introducida desde la unidad (130) de identificación de versiones de hardware a través del canal (111) designado; y

en el que el microcontrolador (110) está configurado de manera que, si la información de versión de hardware recibida a través del canal (111) designado no existe en la lista (300) de versiones de hardware, no se realiza la actualización de software y el segundo conmutador (150) se apaga y el primer conmutador (140) se enciende.

2. El dispositivo (200) de gestión de baterías según la reivindicación 1, en el que el microcontrolador (110) está configurado para incluir un terminal (170) de control para emitir una orden de control de encendido/apagado a los conmutadores (140, 150) primero y segundo,

en el que el terminal (170) de control emite una señal de control para encender el primer conmutador (140) y apagar el segundo conmutador (150) en un modo normal o en un modo de funcionamiento de aplicación.

3. El dispositivo (200) de gestión de baterías según la reivindicación 1, en el que, cuando se recibe el software de actualización, el microcontrolador (110) compara una lista (300) de versiones de hardware compatibles incluida en el software de actualización con la información de versión de hardware recibida de la unidad (130) de identificación de versiones de hardware, y controla el encendido/apagado de los conmutadores (140, 150) primero y segundo según un resultado de comparación.

4. Un método de control de un dispositivo (200) de gestión de baterías, comprendiendo el método:

una etapa (S420) de recepción de software de actualización de recepción de software de actualización para actualizar el software de un sistema de gestión de baterías;

una etapa (S430) de adquisición de información de versión de hardware de adquisición de información de versión de hardware del sistema de gestión de baterías almacenada en una unidad (130) de identificación de versiones de hardware del sistema de gestión de baterías;

una etapa (S440) de comprobación de lista de versiones de hardware de comparación de información de versión de hardware adquirida con una lista (300) de versiones de hardware compatibles incluida en el software de actualización para determinar si debe procederse con la actualización del software; y

una etapa (S450) de inicio del modo de arranque de inicio de un modo de arranque para actualizar el software cuando la información de versión de hardware del sistema de gestión de baterías está incluida en la lista de

versiones de hardware compatibles; en el que el dispositivo (200) de gestión de baterías incluye al menos un canal de entrada/salida configurado como canal (111) designado,

5 en el que el canal (111) designado está conectado simultáneamente a un primer conmutador (140) que tiene un extremo conectado a una unidad (120) de ejecución de aplicación predeterminada y un segundo conmutador (150) que tiene un extremo conectado a la unidad (130) de identificación de versiones de hardware,

10 en el que cada uno del primer conmutador (140) y el segundo conmutador (150) tiene un estado de encendido/apagado configurado de forma diferente según un control del microcontrolador (110);

15 en el que, cuando hay una instrucción de actualización de software, el método incluye además encender el segundo conmutador (150) y apagar el primer conmutador (140) para recibir la información de versión de hardware introducida desde la unidad (130) de identificación de versiones de hardware a través del canal (111) designado; y en el que

20 si la etapa (S440) de comprobación de lista de versiones de hardware revela que la información de versión de hardware recibida a través del canal (111) designado no existe en la lista (300) de versiones de hardware, no se realiza la actualización de software y el segundo conmutador (150) se apaga y el primer conmutador (140) se enciende.

Fig. 1

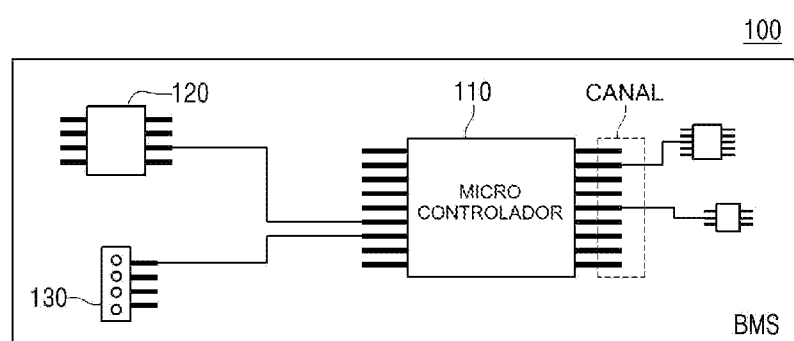


Fig. 2

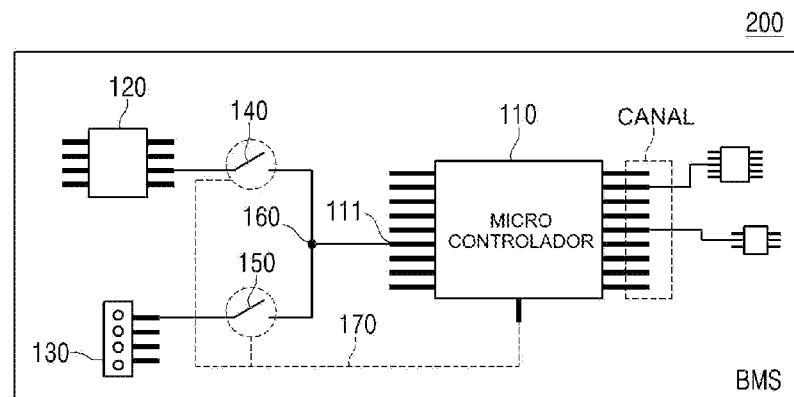


Fig. 3

300

00000070	D7	81	00	08	D9	81	00	08	D8	81	00	08	D0	81	00	08
00000080	DF	81	00	08	E1	81	00	08	E3	81	00	08	E5	81	00	08
00000090	E7	81	00	08	E9	81	00	08	E8	81	00	08	ED	81	00	08
000000A0	EF	81	00	08	F1	81	00	08	E3	81	00	08	F5	81	00	08
00000009	57	81	00	08	E9	81	00	08	E9	81	00	08	50	81	00	08
00000F40	41	EA	01	21	41	EA	01	41	20	2A	12	D3	A2	F1	20	02
00000F50	2D	E9	F0	03	08	46	0C	46	00	46	0E	46	0F	46	88	46
00000F60	89	46	A0	E8	FA	03	20	3A	FB	D2	BD	E8	F0	03	20	32
00000F70	8B	D0	04	2A	05	D3	40	F8	04	18	04	3A	05	D0	04	2A
00000F80	F9	D2	00	F8	01	1B	01	3A	FB	D1	G0	4G	70	47	00	00
00000F90	00	00	00	00	00	00	00	01	02	03	04	06	07	08	09	
00000FA0	00	24	F4	00	.											

Fig. 4

