

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 984 946**

51 Int. Cl.:

G06F 8/65 (2008.01)

G06F 11/14 (2006.01)

G06F 1/30 (2006.01)

G06F 9/4401 (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **30.11.2020** **E 20275177 (2)**

97 Fecha y número de publicación de la concesión europea: **01.05.2024** **EP 4006718**

54 Título: **Actualización a prueba de fallos del firmware del cargador de arranque**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
31.10.2024

73 Titular/es:

CARRIER CORPORATION (100.0%)
13995 Pasteur Blvd.
Palm Beach Gardens, FL 33418, US

72 Inventor/es:

WOLNOWSKI, PIOTR y
RAASZ, PAWEL

74 Agente/Representante:

ISERN JARA, Jorge

ES 2 984 946 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

Actualización a prueba de fallos del firmware del cargador de arranque

- 5 La presente descripción se refiere a un método para actualizar firmware de un sistema tal como un sistema de protección contra incendios o un sistema de detección de intrusiones.

10 Sistemas tales como sistemas de protección contra incendios y sistemas de detección de intrusiones comprenden típicamente una pluralidad de dispositivos conectados eléctricamente, por ejemplo, en una configuración en bucle. Por ejemplo, los sistemas de protección contra incendios comprenden típicamente un panel de control de incendios y uno o más de otros dispositivos de protección contra incendios, tales como detectores de incendios (tales como sensores de humo y calor), puntos de llamada manuales, alarmas contra incendios, y sistemas de extinción de incendios (tales como aspersores, barreras contra incendios, extractores de humo, etc.).

15 Cada dispositivo de estos sistemas puede incluir un microcontrolador integrado configurado para ejecutar una aplicación que proporciona las funciones base de ese dispositivo. La aplicación puede proporcionarse como firmware, por ejemplo, cuando el código de aplicación se almacena en una memoria no volátil (por ejemplo, una memoria flash) del microcontrolador.

20 En estos sistemas, también se puede proporcionar un denominado cargador de arranque como firmware, por ejemplo, cuando el código del cargador de arranque también se almacena en la memoria no volátil. El cargador de arranque es responsable de cargar (iniciar) la aplicación, y también proporciona al dispositivo con la capacidad de actualizar o reemplazar el código de aplicación.

25 El documento US 2005/132179 describe un método para aplicar actualizaciones de imagen de software personalizadas a un almacenamiento no volátil de una manera a prueba de fallos. El documento CN 109, 933, 348 describe un método y aparato para actualizar un cargador de arranque en una unidad de control electrónico.

30 El Solicitante cree que sigue habiendo margen para mejorar los métodos de actualización de firmware.

La presente invención proporciona un método para actualizar firmware almacenado en una memoria no volátil de un controlador como se reivindica en la reivindicación 1.

35 La presente invención también proporciona un sistema como se reivindica en la reivindicación 10.

40 Varias realizaciones se dirigen a un método para actualizar firmware almacenado en una memoria no volátil de un controlador tal como un microcontrolador, y en particular a un método para actualizar un firmware de cargador de arranque para el controlador. Según el método, una aplicación de actualización del cargador de arranque se almacena en la memoria no volátil en un conjunto particular (segundo) de bloques de memoria de la memoria no volátil, es decir, donde el segundo conjunto de bloques de memoria incluye un bloque de memoria inicial, opcionalmente uno o más bloques de memoria intermedia, y un bloque de memoria final. La aplicación de actualización del cargador de arranque es una aplicación que es operable para actualizar (por ejemplo, reemplazar) un cargador de arranque existente almacenado en un primer conjunto de bloques de memoria de la memoria no volátil, es decir, donde el primer conjunto de bloques de memoria incluye un bloque de memoria inicial, opcionalmente uno o más bloques de memoria intermedia, y un bloque de memoria final.

50 El Solicitante ha reconocido que cuando un cargador de arranque se actualiza usando técnicas convencionales, puede haber un riesgo significativo de que el (micro)controlador se vuelva inoperable (es decir, "bloqueado"), por ejemplo, si el controlador pierde alimentación o se reinicia de otro modo mientras se actualiza el cargador de arranque.

55 Por lo tanto, como una etapa en la actualización del cargador de arranque (es decir, como una etapa de la aplicación de actualización del cargador de arranque), el bloque de memoria inicial del primer conjunto de bloques de memoria (es decir, el bloque de memoria de la memoria no volátil que almacena un bloque de datos inicial del cargador de arranque) se reemplaza (sobrescribe) con un código de salto, donde el código de salto incluye una instrucción para saltar al bloque de memoria inicial del segundo conjunto de bloques de memoria (es decir, el bloque de memoria de la memoria no volátil que almacena un bloque de datos inicial de la aplicación de actualización del cargador de arranque). El código de salto también puede comprender un código de inicio mínimo (es decir, según se requiera para establecer la tabla de vectores y el puntero de pila) junto con la instrucción para saltar al bloque de memoria inicial del segundo conjunto de bloques de memoria.

65 A este respecto, el Solicitante ha reconocido que cuando el controlador pierde alimentación (externa), puede proporcionarse internamente suficiente energía al controlador (por ejemplo, almacenada en uno o más condensadores del controlador) para permitir que al menos uno (o más) bloque(s) de memoria de la memoria

no volátil (por ejemplo, al menos un grupo (por ejemplo, fila) de bloques de memoria) se borren y escriban con éxito. Esta propiedad garantiza efectivamente que el código de salto se almacenará con éxito en el bloque de memoria inicial del primer conjunto de bloques de memoria, incluso si el controlador pierde alimentación.

- 5 Además, reemplazar el bloque de datos inicial del cargador de arranque con un código de salto significa que, si el controlador pierde alimentación o se reinicia de otro modo mientras los (otros) bloques de memoria que almacenan el cargador de arranque existente se actualizan (reemplazan), cuando el controlador se reinicia, el procesamiento del controlador regresará a la aplicación de actualización del cargador de arranque a través del código de salto, mientras se salta el código del cargador de arranque posiblemente dañado o incompleto. La aplicación de actualización del cargador de arranque puede entonces reanudar su operación y completar su actualización del cargador de arranque.

Se apreciará que varias realizaciones proporcionan en consecuencia un método a prueba de fallos (es decir, resistente a la pérdida de alimentación) para actualizar un firmware del cargador de arranque.

- 15 El controlador puede ser cualquier controlador adecuado tal como un controlador integrado de un dispositivo tal como un dispositivo de protección contra incendios o un dispositivo de detección de intrusiones. El controlador puede proporcionarse con un único chip de circuito integrado (IC), por ejemplo, puede ser un microcontrolador, un sistema en un chip, etc.

- 20 El dispositivo puede formar parte de un sistema tal como un sistema de protección contra incendios o un sistema de detección de intrusiones. El sistema puede comprender una pluralidad de tales dispositivos que pueden estar conectados eléctricamente, por ejemplo, en una configuración en bucle.

- 25 Por ejemplo, cuando el sistema es un sistema de protección contra incendios, el sistema puede comprender un panel de control de incendios y una pluralidad de dispositivos de protección contra incendios conectados al panel de control de incendios. Como tal, el controlador puede ser un (micro)controlador integrado de un dispositivo de protección contra incendios tal como un detector de incendios, un detector de humo, un detector de calor, un punto de llamada manual, una alarma contra incendios, un dispositivo de extinción de incendios, un aspersor, una barrera contra incendios, un extractor de humo, y similares. Cuando el sistema es un sistema de detección de intrusiones, el sistema puede comprender un panel de control de detección de intrusiones y una pluralidad de dispositivos de detección de intrusiones conectados al panel de control de detección de intrusiones. Como tal, el controlador puede ser un (micro)controlador integrado de un dispositivo de detección de intrusiones tal como un sensor de movimiento, sensor de contacto, y similares.

- 35 El controlador (por ejemplo, microcontrolador) puede comprender un núcleo de procesador junto con una o más etapas de entrada/salida. El controlador puede incluir opcionalmente una memoria de acceso aleatorio (RAM) accesible al núcleo del procesador.

- 40 El controlador (por ejemplo, microcontrolador) incluye memoria no volátil. La memoria no volátil puede comprender cualquier memoria no volátil adecuada tal como memoria flash. La memoria no volátil puede comprender una pluralidad de celdas de memoria dispuestas en una pluralidad de bloques de memoria (también conocidos como "páginas" de memoria), es decir, donde cada bloque de memoria (página) incluye un subconjunto (diferente, no superpuesto) de las celdas de memoria. Cada bloque de memoria puede configurarse para almacenar un bloque de datos (un bloque de datos). La memoria no volátil puede configurarse de manera que cada bloque de memoria pueda escribirse independientemente de los demás bloques de memoria. Un bloque de memoria (página) es, por consiguiente, la colección más pequeña de celdas de memoria en las que se puede escribir en una sola operación (escritura). La memoria no volátil puede comprender cualquier número (varios) adecuado de bloques de memoria.

- 50 La memoria no volátil puede configurarse de manera que cada bloque de memoria pueda borrarse independientemente de los demás bloques de memoria. Sin embargo, más típicamente, la memoria no volátil puede configurarse de manera que cada grupo de bloques de memoria plurales (por ejemplo, 4) (tal como cada fila de bloques de memoria) puede borrarse independientemente de los demás grupos (por ejemplo, fila) de bloques de memoria.

- 60 Cada bloque de memoria puede tener una dirección respectiva. Por lo tanto, se puede proporcionar y definir una secuencia de direcciones de memoria para la memoria no volátil, donde cada dirección de memoria de la secuencia corresponde a (se refiere a) un bloque individual de la pluralidad de bloques de memoria. Las direcciones de memoria pueden comprender direcciones virtuales o direcciones físicas.

La memoria no volátil puede configurarse para almacenar (y puede almacenar) firmware del controlador, por ejemplo, en la forma de código adecuado (por ejemplo, legible por máquina u otro nivel bajo).

- 65 El firmware puede incluir en general una o más aplicaciones, junto con un cargador de arranque. Por lo tanto, la memoria no volátil puede configurarse para almacenar (y puede almacenar) una o más aplicaciones y un

cargador de arranque, por ejemplo, en la forma de código de aplicación y código del cargador de arranque.

La una o más aplicaciones pueden comprender cualquier aplicación o aplicaciones adecuadas, tal como una aplicación configurada para proporcionar las funciones base del dispositivo.

5

El cargador de arranque puede comprender cualquier cargador de arranque adecuado, pero en realizaciones particulares el cargador de arranque es un cargador de arranque de una sola etapa. También sería posible que el cargador de arranque fuera la primera etapa (la primera) en un cargador de arranque de múltiples etapas. El cargador de arranque puede configurarse para cargar (para iniciar) la aplicación, y también puede configurarse de manera que la aplicación (código) puede actualizarse o reemplazarse.

10

El controlador puede configurarse de manera que un primer conjunto de bloques de memoria (de la pluralidad de bloques de memoria) se configura para almacenar el cargador de arranque (por ejemplo, de manera que un primer conjunto de bloques de memoria se asigna para almacenar el cargador de arranque). El primer conjunto de bloques de memoria puede incluir un bloque de memoria inicial, opcionalmente uno o más bloques de memoria intermedia, y un bloque de memoria final. El primer conjunto de bloques de memoria puede ser un conjunto contiguo de bloques de memoria (física y/o lógicamente), pero este no tiene por qué ser el caso.

15

El primer conjunto de bloques de memoria puede tener (puede ser dirigido por) un primer conjunto de direcciones de memoria. El primer conjunto de direcciones de memoria puede incluir una dirección de memoria inicial, opcionalmente una o más direcciones de memoria intermedias, y una dirección de memoria final. El primer conjunto de direcciones de memoria puede ser un conjunto contiguo de direcciones de memoria. El primer conjunto de direcciones de memoria puede ser las primeras direcciones de memoria en la secuencia de direcciones de memoria para la memoria no volátil, pero este no tiene que ser el caso.

20

25

El controlador puede configurarse de manera que los bloques de memoria de la pluralidad de bloques de memoria diferentes del primer conjunto (es decir, bloques de memoria con direcciones diferentes del primer conjunto de direcciones) se configuran para almacenar la una o más aplicaciones.

30

El controlador puede configurarse de manera que, al arrancar (encender), el cargador de arranque se ejecuta inicialmente. Esto puede conseguirse de cualquier manera adecuada. En varias realizaciones particulares, el controlador se configura, al arrancar (encender), para ejecutar el código almacenado en el primer conjunto de bloques de memoria, es decir, ejecutando inicialmente el código almacenado en el bloque de memoria inicial del primer conjunto de bloques de memoria. Por ejemplo, cuando el bloque de memoria inicial del primer conjunto de bloques de memoria es el bloque de memoria de la memoria no volátil que tiene la primera (primera) dirección de memoria en la secuencia de direcciones de memoria para la memoria no volátil, el controlador puede configurarse, al iniciar (encender), para ejecutar el código almacenado en el bloque de memoria que tiene la primera (primera) dirección de memoria en la secuencia de direcciones de memoria para la memoria no volátil.

35

40

El controlador también puede configurarse, después de ejecutar el código almacenado en un bloque particular, para ejecutar entonces el código almacenado en el siguiente bloque de memoria (es decir, el bloque de memoria que tiene la siguiente dirección de memoria en la secuencia de direcciones de memoria para la memoria no volátil), y así sucesivamente (excepto cuando el código ejecutado incluye una instrucción de salto, como se describirá más adelante). De esta manera, cuando se desee, el controlador puede ejecutar el cargador de arranque (completo) que puede almacenarse a través del primer conjunto (varios) de bloques de memoria al iniciar (encender).

45

El cargador de arranque puede configurarse para determinar si se desea actualizar (por ejemplo, reemplazar) la aplicación (código), por ejemplo, comprobando una señal (externa) (por ejemplo, que puede enviarse al controlador desde el panel de control) y/o comprobando el estado de un indicador en la memoria no volátil, o de otro modo. Cuando se determina que esto no es (es diferente de) necesario, la aplicación puede entonces ser cargada (iniciada), por ejemplo, ejecutando el código de aplicación apropiado almacenado en la memoria no volátil.

50

55

Sin embargo, cuando se determina que se desea actualizar (por ejemplo, reemplazar) la aplicación (código), la aplicación (código) puede entonces actualizarse (reemplazarse) (en lugar de cargar la aplicación), por ejemplo, escribiendo nuevos bloques de código de aplicación en la memoria no volátil.

60

Tal como se ha descrito anteriormente, el Solicitante ha reconocido que, además de actualizar la aplicación, puede ser necesario actualizar el cargador de arranque, pero que en las técnicas convencionales para actualizar un cargador de arranque, puede haber un riesgo significativo de que el controlador se vuelva inoperable (es decir, "bloqueado"), por ejemplo, si el controlador pierde alimentación o se reinicia de otro modo mientras el cargador de arranque se actualiza o reemplaza (particularmente cuando el cargador de arranque a actualizar es un cargador de arranque de una sola etapa o una primera etapa de un cargador de arranque de múltiples etapas).

65

Por lo tanto, las realizaciones proporcionan una aplicación de actualización del cargador de arranque que puede funcionar para actualizar (reemplazar) un cargador de arranque existente almacenado en la memoria no volátil con un nuevo cargador de arranque. La aplicación de actualización del cargador de arranque puede configurarse para actualizar el cargador de arranque de una manera a prueba de fallos (es decir, resistente a la pérdida de alimentación).

La aplicación de actualización del cargador de arranque puede proporcionarse inicialmente al controlador y almacenarse en la memoria no volátil. Esto puede ser hecho usando el proceso descrito anteriormente de actualización de la aplicación.

Por lo tanto, el cargador de arranque (existente) puede ejecutarse, y una señal externa y/o un indicador puede establecerse para que el cargador de arranque determine que la aplicación debe actualizarse. La aplicación puede entonces actualizarse (por ejemplo, al menos reemplazarse parcialmente) escribiendo nuevos bloques de código para la aplicación de actualización del cargador de arranque en la memoria no volátil (al menos parcialmente en lugar del código de aplicación existente).

La aplicación (código) de actualización del cargador de arranque se almacena en un segundo conjunto de bloques de memoria de la memoria no volátil. El segundo conjunto de bloques de memoria puede comprender un conjunto diferente (no superpuesto) de bloques de memoria al primer conjunto de bloques de memoria. El segundo conjunto de bloques de memoria puede incluir un bloque de memoria inicial, opcionalmente uno o más bloques de memoria intermedia, y un bloque de memoria final. El segundo conjunto de bloques de memoria puede ser un conjunto contiguo de bloques de memoria (física y/o lógicamente), pero este no tiene por qué ser el caso.

El segundo conjunto de bloques de memoria puede tener (puede ser dirigido por) un segundo conjunto de direcciones de memoria. El segundo conjunto de direcciones de memoria puede incluir una dirección de memoria inicial, opcionalmente una o más direcciones de memoria intermedias, y una dirección de memoria final. El segundo conjunto de direcciones de memoria puede ser un conjunto contiguo de direcciones de memoria.

La aplicación (código) de actualización del cargador de arranque que se almacena en el segundo conjunto de bloques de memoria puede comprender una pluralidad de bloques de datos. Además de la aplicación (código) de actualización del cargador de arranque, el código de salto puede almacenarse en el segundo conjunto de bloques de memoria.

Además, el nuevo código del cargador de arranque propiamente dicho (o partes del código del cargador de arranque que se van a reemplazar) puede(n) almacenarse en el segundo conjunto de bloques de memoria. Alternativamente, una instrucción o instrucciones que provocarán que el controlador descargue el nuevo código del cargador de arranque (o partes del código del cargador de arranque que van a ser reemplazadas), por ejemplo, desde un dispositivo externo tal como el panel de control y/o a través de una conexión de red, se puede(n) almacenar en el segundo conjunto de bloques de memoria. El nuevo código del cargador de arranque comprenderá una pluralidad de bloques de datos, incluyendo un bloque de datos inicial, opcionalmente uno o más bloques de datos intermedios, y un bloque de datos final.

Una vez que la aplicación de actualización del cargador de arranque (y el código de salto y opcionalmente el nuevo cargador de arranque) se ha almacenado en el segundo conjunto de bloques de memoria, el controlador puede entonces hacer funcionar (ejecutar) la aplicación de actualización del cargador de arranque, por ejemplo, ejecutando el código de aplicación apropiado almacenado en el segundo conjunto de bloques de memoria.

La aplicación de actualización del cargador de arranque está configurada para almacenar el código de salto en el bloque de memoria inicial del primer conjunto de bloques de memoria. El código de salto puede almacenarse solo en el (único, primerísimo) bloque de memoria inicial del primer conjunto de bloques de memoria, o el código de salto puede almacenarse en un conjunto de varios bloques de memoria del primer conjunto de bloques de memoria que incluye (único, primerísimo) bloque de memoria inicial del primer conjunto de bloques de memoria. Por ejemplo, el código de salto puede almacenarse en el grupo inicial (por ejemplo, fila) de bloques de memoria del primer conjunto de bloques de memoria.

Por lo tanto, el controlador, cuando ejecuta la aplicación de actualización del cargador de arranque, almacenará el código de salto en (al menos) el bloque de memoria inicial del primer conjunto de bloques de memoria. Esto puede comprender que el controlador copie el código de salto que se almacenó en el segundo conjunto de bloques de memoria (como se describió anteriormente) a (al menos) el bloque de memoria inicial del primer conjunto de bloques de memoria. Copiar el código de salto al bloque de memoria inicial puede comprender borrar (al menos) el bloque de memoria inicial (por ejemplo, borrar el grupo inicial (por ejemplo, fila) de bloques de memoria que incluye el bloque de memoria inicial), y escribir el código de salto en (al menos) el bloque de memoria inicial (por ejemplo, escribir el código de salto en el grupo inicial (por ejemplo fila) de bloques de

memoria).

El código de salto puede comprender un código de inicio mínimo (es decir, según se requiera para establecer la tabla de vectores y el puntero de pila) y una instrucción para saltar (bifurcar) al bloque de memoria inicial (primero) (es decir, a la dirección de memoria inicial (primera)) del segundo conjunto de bloques de memoria (es decir, al bloque de memoria en el que se almacena el primer bloque de datos del código de aplicación de actualización del cargador de arranque). Esto significa que, cuando se ejecuta por el controlador, el código de salto hará que el controlador se bifurque a la aplicación de actualización del cargador de arranque que está almacenada en el segundo conjunto de bloques de memoria.

Por lo tanto, como una etapa en la actualización del cargador de arranque (es decir, como una etapa de la aplicación de actualización del cargador de arranque), el bloque de memoria inicial del primer conjunto de bloques de memoria (es decir, el bloque de memoria de la memoria no volátil que almacena un bloque de datos inicial (primero) del cargador de arranque) se reemplaza (sobrescribe) con un código de salto, donde el código de salto incluye una instrucción para saltar al bloque de memoria inicial del segundo conjunto de bloques de memoria (es decir, el bloque de memoria de la memoria no volátil que almacena un bloque de datos inicial (primero) de la aplicación de actualización del cargador de arranque).

A este respecto, el Solicitante ha reconocido que cuando el controlador pierde alimentación (externa), puede proporcionarse internamente suficiente energía al controlador (por ejemplo, almacenada en uno o más condensadores del controlador o de otro modo) para permitir que al menos uno (o más) bloque(s) de memoria de la memoria no volátil (tal como, al menos uno (o más) grupo(s) (fila) de bloques de memoria) se borren y escriban con éxito. Esta propiedad garantiza efectivamente que el código de salto se almacenará con éxito en el bloque de memoria inicial del primer conjunto de bloques de memoria, incluso si el controlador pierde alimentación.

Además, el almacenamiento del código de salto en el bloque de memoria inicial del primer conjunto de bloques de memoria (es decir, la sustitución del bloque de datos inicial del cargador de arranque por un código de salto) significa que, en caso de que el controlador pierda alimentación o se reinicie de otro modo mientras se actualizan los (demás) bloques de memoria que almacena el cargador de arranque existente, cuando se reinicia el controlador, el procesamiento del controlador volverá a la aplicación de actualización del cargador de arranque a través del código de salto, mientras se salta el código del cargador de arranque posiblemente dañado o incompleto. La aplicación de actualización del cargador de arranque puede entonces reanudar su operación y completar su actualización del cargador de arranque.

Se entenderá que varias realizaciones proporcionan en consecuencia un método a prueba de fallos (es decir, resistente a la pérdida de alimentación) para actualizar un firmware del cargador de arranque.

El controlador está configurado de manera que un bloque de memoria, o más de un bloque de memoria de la memoria no volátil puede reprogramarse (borrarse y escribirse) sin que el controlador reciba alimentación externa. Por ejemplo, el controlador puede configurarse de manera que dos, cuatro, seis, ocho, diez o más bloques de memoria de la memoria no volátil puedan reprogramarse (borrarse y escribirse) sin que el controlador reciba alimentación externa. El controlador puede configurarse de manera que un grupo (fila) de bloques de memoria, o más de un grupo (fila) de bloques de memoria puedan reprogramarse (borrarse y escribirse) sin que el controlador reciba alimentación externa. Por ejemplo, el controlador puede configurarse de manera que dos, cuatro, seis, ocho, diez o más grupos (filas) de bloques de memoria puedan reprogramarse (borrarse y escribirse) sin que el controlador reciba alimentación externa. La configuración del controlador de manera que más de un bloque de memoria de la memoria no volátil pueda programarse (borrarse y escribirse) sin que el controlador reciba energía externa proporciona de forma beneficiosa un grado de redundancia.

En realizaciones, antes de almacenar el código de salto en el bloque de memoria inicial del primer conjunto de bloques de memoria, el controlador puede (y la aplicación de actualización del cargador de arranque puede estar configurada para) determinar si el controlador está recibiendo actualmente alimentación (externa). Cuando se determina que el controlador está recibiendo actualmente alimentación (externa), entonces el controlador puede (y la aplicación de actualización del cargador de arranque puede estar configurada para) proceder con la etapa de almacenar el código de salto en el bloque de memoria inicial del primer conjunto de bloques de memoria. Cuando se determina que el controlador no está recibiendo (es diferente de) alimentación (externa), entonces el controlador puede (y la aplicación de actualización del cargador de arranque puede estar configurada para) no proceder con (que no sea) esta etapa, por ejemplo, cancelando el proceso de actualización del cargador de arranque o pausando el proceso de actualización del cargador de arranque (en particular pausando el almacenamiento del código de salto en el bloque de memoria inicial del primer conjunto de bloques de memoria) hasta que se determine que la alimentación está presente. Esto asegura que el controlador tendrá suficiente alimentación para almacenar con éxito el código de salto.

Una vez que el código de salto ha sido almacenado en el (uno o más) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria, el controlador (y la aplicación de actualización del cargador de

- arranque pueden configurarse para) entonces almacena (almacenar) bloques de datos del nuevo cargador de arranque diferentes del (uno o más) bloque(s) de datos inicial (es decir, los bloques de datos intermedio y final, sin incluir el (uno o más) bloque(s) de datos inicial(es)) en bloques de memoria del primer conjunto de bloques de memoria distintos del (los) bloque(s) de memoria inicial(es) (es decir, los bloques de memoria intermedio y final, sin incluir el(los) bloque(s) de memoria inicial(es)). En otras palabras, el controlador almacena los bloques de datos del nuevo cargador de arranque excepto el(los) bloque(s) de datos inicial(es) (primero(s)) en bloques de memoria del primer conjunto de bloques de memoria excepto el(los) bloque(s) de memoria inicial(es) (primero(s)) del primer conjunto de bloques de memoria (en el que se almacena el código de salto).
- Se entenderá que, si el controlador pierde alimentación o se reinicia de otro modo durante este proceso, cuando el controlador se reinicia, volverá a ejecutar la aplicación de actualización del cargador de arranque, mientras se salta el cargador de arranque posiblemente dañado o incompleto. La aplicación de actualización del cargador de arranque puede entonces completar su operación para actualizar completamente el cargador de arranque.
- En varias realizaciones, antes de la etapa descrita anteriormente de determinar si el controlador está recibiendo alimentación actualmente, el controlador puede (y la aplicación de actualización del cargador de arranque puede estar configurada para) determinar si el código de salto ya se ha almacenado en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria. Cuando se determina que el código de salto ya se ha almacenado en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria, entonces el controlador puede (y la aplicación de actualización del cargador de arranque puede configurarse para) proceder con (por ejemplo, continuar con) el proceso descrito anteriormente de almacenar bloques de datos del nuevo cargador de arranque excepto el(los) bloque(s) de datos inicial(es) en bloques de memoria del primer conjunto de bloques de memoria excepto el(los) bloque(s) de memoria inicial(es) (en los que se almacena el código de salto). Esta determinación puede hacerse cuando el controlador pierde alimentación o se reinicia de otro modo durante un intento previo de actualización del cargador de arranque.
- Cuando, por otro lado, se determina que el código de salto no ha sido (ha sido diferente de) ya almacenado en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria, el controlador puede (y la aplicación de actualización del cargador de arranque puede estar configurada para) proceder con la etapa descrita anteriormente de determinar si el controlador está recibiendo actualmente alimentación (externa) (y luego almacenar el código de salto en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria, y así sucesivamente).
- Una vez que los bloques de datos del nuevo cargador de arranque distinto del(de los) bloque(s) de datos inicial(es) se han almacenado en el primer conjunto de bloques de memoria (distintos del(de los) bloque(s) de memoria inicial(es)), el controlador entonces (de nuevo) (y la aplicación de actualización del cargador de arranque se puede configurar para) determina (determinar) si el controlador está recibiendo actualmente alimentación (externa).
- Cuando se determina que el controlador está recibiendo actualmente alimentación (externa), entonces el controlador (y la aplicación de actualización del cargador de arranque pueden configurarse para) almacena (almacenar) el(los) bloque(s) de datos inicial(es) del nuevo cargador de arranque en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria (es decir, sobrescribiendo el código de salto).
- Cuando se determina que el controlador no está recibiendo (es diferente de) alimentación (externa), entonces el controlador puede (y la aplicación de actualización del cargador de arranque puede estar configurada para) no proceder con esta etapa, por ejemplo, cancelando el proceso de actualización del cargador de arranque o pausando el proceso de actualización del cargador de arranque (en particular pausando el almacenamiento del código de salto en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria) hasta que se determine que la alimentación está presente. Esto asegura que el controlador tenga suficiente alimentación para almacenar con éxito el(los) bloque(s) de datos inicial(es) del nuevo cargador de arranque en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria (incluso si el controlador pierde alimentación).
- Una vez que el(los) bloque(s) de datos inicial(es) del nuevo cargador de arranque se han almacenado en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria, el proceso de actualización del cargador de arranque se completa, y la aplicación de actualización del cargador de arranque se puede deshabilitar. Se entenderá que, al reiniciar, el controlador ejecutará ahora el nuevo cargador de arranque.
- En varias realizaciones, antes de la etapa descrita anteriormente de determinar si el código de salto ya se ha almacenado en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria, el controlador puede (y la aplicación de actualización del cargador de arranque puede configurarse para) determinar si el nuevo cargador de arranque se ha almacenado (completamente) en el primer conjunto de bloques de memoria.
- Cuando se determina que el nuevo cargador de arranque se ha almacenado (completamente) en el primer

conjunto de bloques de memoria, el controlador puede (y la aplicación de actualización del cargador de arranque puede configurarse para) proceder con la etapa descrita anteriormente de deshabilitar la aplicación de actualización del cargador de arranque. Esta determinación se puede hacer cuando el controlador perdió alimentación durante un intento previo de actualización del cargador de arranque, por ejemplo, antes o durante la etapa de deshabilitación. Esta determinación también se puede hacer o en su lugar se puede hacer cuando se hace un intento de actualizar el cargador de arranque cuando esto ya se ha hecho.

Cuando, por otro lado, se determina que el nuevo cargador de arranque no ha sido (ha sido diferente de) almacenado completamente en el primer conjunto de bloques de memoria, el controlador puede (y la aplicación de actualización del cargador de arranque puede estar configurada para) proceder con la etapa descrita anteriormente de determinar si el código de salto ya ha sido almacenado en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria (y así sucesivamente).

Todos los aspectos y realizaciones descritos en la presente memoria pueden y en realizaciones incluyen una cualquiera o más o todas las características preferidas y opcionales descritas en la presente memoria, según sea apropiado.

Los métodos descritos en la presente memoria pueden implementarse al menos parcialmente usando software, por ejemplo, programas informáticos. Por lo tanto, se verá que cuando se observa desde otros aspectos, la presente invención proporciona software informático específicamente adaptado para llevar a cabo los métodos descritos en la presente memoria cuando se instala en medios de procesamiento de datos, un elemento de programa informático que comprende partes de código de software informático para realizar los métodos descritos en la presente memoria cuando el elemento de programa se ejecuta en medios de procesamiento de datos, y un programa informático que comprende medios de código adaptados para realizar todas las etapas de un método o de los métodos descritos en la presente memoria cuando el programa se ejecuta en un sistema de procesamiento de datos. El sistema de procesamiento de datos puede ser un microprocesador, un FPGA programable (Matriz de Puertas Programables en Campo), etc.

La invención también se extiende a un portador de software informático que comprende dicho software que cuando se usa para operar un sistema de protección contra incendios o un sistema de detección de intrusos que comprende un procesador de datos que provoca en conjunto con el procesador de datos que el sistema lleve a cabo los etapas de los métodos descritos en la presente memoria. Dicho portador de software informático podría ser un medio de almacenamiento físico tal como un chip ROM, CD ROM o disco, o podría ser una señal tal como una señal electrónica sobre cables, una señal óptica o una señal de radio tal como la de un satélite o similar.

Se apreciará además que no todas las etapas de los métodos descritos en la presente memoria necesitan llevarse a cabo mediante software informático y, por lo tanto, a partir de un aspecto más amplio, la presente invención proporciona software informático y dicho software instalado en un soporte de software informático para llevar a cabo al menos una de las etapas de los métodos descritos en la presente memoria.

Por consiguiente, la presente invención puede realizarse adecuadamente como un producto de programa informático para su uso con un sistema informático. Dicha implementación puede comprender una serie de instrucciones legibles por ordenador ya sea fijas en un medio tangible, tal como un medio legible por ordenador no transitorio, por ejemplo, disquete, CD ROM, ROM o disco duro. También podría comprender una serie de instrucciones legibles por ordenador transmisibles a un sistema informático, a través de un módem u otro dispositivo de interfaz, a través de un medio tangible, incluyendo, pero sin limitarse a, líneas de comunicaciones ópticas o analógicas, o intangiblemente usando técnicas inalámbricas, incluyendo, pero sin limitarse a, microondas, infrarrojos u otras técnicas de transmisión. La serie de instrucciones legibles por ordenador incorpora toda o parte de la funcionalidad descrita en la presente memoria.

Dichas instrucciones legibles por ordenador se pueden escribir en varios lenguajes de programación para su uso con muchas arquitecturas de ordenador o sistemas operativos. Además, dichas instrucciones pueden almacenarse usando cualquier tecnología de memoria, presente o futura, incluyendo, pero sin limitarse a, de semiconductor, magnética u óptica, o transmitirse usando cualquier tecnología de comunicaciones, presente o futura, incluyendo, pero sin limitarse a, óptico, infrarrojo o microondas. Se contempla que dicho producto de programa informático pueda distribuirse como un medio extraíble con documentación impresa o electrónica adjunta, por ejemplo, software retractilado, precargado con un sistema informático, por ejemplo, en una ROM de sistema o disco fijo, o distribuido desde un servidor o tablón de anuncios electrónico a través de una red, por ejemplo, Internet o red mundial de información (World Wide Web).

Ciertas realizaciones preferidas de la presente invención se describirán ahora, a modo de ejemplo únicamente, con referencia a los siguientes dibujos, en los que:

La Figura 1 muestra esquemáticamente un sistema de protección contra incendios según varias realizaciones;

La Figura 2 es un diagrama de flujo que ilustra un método convencional de actualización de un firmware de cargador de arranque;

5 La Figura 3 es un diagrama de flujo que ilustra un método a prueba de fallos para actualizar un firmware de cargador de arranque según varias realizaciones;

La Figura 4 es un diagrama de flujo que ilustra un método según varias realizaciones;

10 La Figura 5 muestra esquemáticamente un firmware de actualización del cargador de arranque según varias realizaciones;

Las Figuras 6A-6E muestran esquemáticamente varios estados de una memoria no volátil durante un proceso de actualización del cargador de arranque según diversas realizaciones; y

15 La Figura 7 muestra esquemáticamente un método para determinar un número de páginas que son programables por un microcontrolador después de una pérdida de alimentación.

20 La Figura 1 muestra esquemáticamente un sistema de protección contra incendios según varias realizaciones. Como se muestra en la Figura 1, el sistema comprende una fuente de alimentación (controlador de bucle) 11 y un conjunto de dispositivos 12 de protección contra incendios conectados eléctricamente en una configuración en bucle. Cada dispositivo 12 de protección contra incendios puede ser, por ejemplo, un detector de incendios, un detector de humo, un detector de calor, un punto de llamada manual, una alarma contra incendios, un componente de extinción de incendios, un aspersor, una barrera contra incendios, un extractor de humo y similares. La fuente de alimentación (controlador de bucle) 11 puede ser controlada por y/o puede ser parte de otra parte del sistema 10 de protección contra incendios tal como un panel de control (contra incendios).

30 Aunque en la realización de la Figura 1 el sistema es un sistema de protección contra incendios, también se entenderá que el sistema puede ser cualquier sistema adecuado tal como un sistema de detección de intrusiones, y similares. Cuando el sistema es un sistema de detección de intrusiones, el sistema puede comprender un panel de control de detección de intrusiones y una pluralidad de dispositivos de detección de intrusiones conectados al panel de control de detección de intrusiones. Cada dispositivo de detección de intrusiones puede ser un sensor de movimiento, un sensor de contacto y similares.

35 Cada dispositivo 12 del conjunto de dispositivos puede conectarse eléctricamente a, y recibir alimentación eléctrica desde, la fuente de alimentación (controlador de bucle) 11 mediante cableado, por ejemplo, que comprende un par de cables 13, 14. Los cables 13, 14 pueden actuar, por ejemplo, como un conductor neutro y un conductor activo, respectivamente. Como puede verse en la Figura 1, el cableado puede estar dispuesto en una configuración en bucle, es decir, de manera que cada cable 13, 14 se inicia y termina en la fuente de alimentación (controlador de bucle) 11. Por consiguiente, la alimentación puede suministrarse en ambos extremos del cableado de bucle por la fuente de alimentación (controlador de bucle) 11.

40 El sistema de protección contra incendios está configurado de manera que cada dispositivo 12 es capaz de comunicarse con el panel 10 de control de incendios, por ejemplo a través de los cables 13, 14 y el controlador 11 de bucle. En particular, como se muestra en la Figura 1, el panel 10 de control puede enviar solicitudes de actualización y datos a uno o más de cada dispositivo 12, por ejemplo, a través del controlador 11 de bucle y los cables 13, 14.

45 Como también se muestra en la Figura 1, uno o más de cada dispositivo 12 incluye(n) un microcontrolador 15 integrado configurado para ejecutar una aplicación que proporciona las funciones de base de ese dispositivo. La aplicación se proporciona como firmware, donde el código 17 de aplicación se almacena en la memoria 16 no volátil (por ejemplo, memoria flash autoprogramable) del microcontrolador 15.

50 También se proporciona un denominado cargador de arranque como firmware, donde el código 18 del cargador de arranque también se almacena en la memoria 16 no volátil. El cargador de arranque es responsable de cargar (iniciar) la aplicación, y también proporciona al dispositivo 12 con la capacidad de actualizar o reemplazar el código 17 de aplicación. Esto permite actualizaciones de firmware de la aplicación en campo. En realizaciones, el cargador de arranque es un cargador de arranque de una sola etapa, pero también sería posible que el cargador de arranque fuera la primera etapa de un cargador de arranque de múltiples etapas.

60 Como también se muestra en la Figura 1, un primer conjunto de bloques de memoria (o "páginas") de la memoria 16 no volátil almacena el código 18 de cargador de arranque, y los bloques de memoria restantes almacenan el código 17 de aplicación. El microcontrolador 15 está configurado de manera que, al arrancar (encender), el código 18 de cargador de arranque se ejecuta inicialmente ejecutando el código almacenado en el primer conjunto de bloques de memoria.

El cargador de arranque puede determinar si se desea actualizar (por ejemplo, reemplazar) el código 17 de aplicación comprobando una señal externa recibida desde el panel 10 de control o en otro lugar y/o comprobando el estado de un indicador en la memoria 16 no volátil. Si esto no es necesario, la aplicación se inicia entonces ejecutando el código 17 de aplicación apropiado almacenado en la memoria 16 no volátil. Sin embargo, cuando se determina que se desea actualizar (reemplazar) el código 17 de aplicación, el código 17 de aplicación puede actualizarse (reemplazarse) escribiendo sobre el código 17 de aplicación existente.

La actualización del código de aplicación siempre es posible con un cargador de arranque de trabajo. Como se ha descrito anteriormente, el Solicitante ha reconocido que, además de actualizar el código 17 de aplicación, puede ser necesario actualizar el código 18 de cargador de arranque, por ejemplo, para corregir errores en el código del cargador de arranque, o para extender su funcionalidad.

Sin embargo, como se ilustra en la Figura 2, en las técnicas convencionales para hacer esto, puede haber un riesgo significativo de que el microcontrolador 15 se vuelva inoperable (es decir, "bloqueado"). Como se muestra en la Figura 2, durante un proceso convencional de este tipo, el microcontrolador 15 se enciende (etapa 20), el cargador de arranque se inicia (etapa 21), el proceso de actualización del cargador de arranque se inicia (etapa 22), y luego el código del cargador de arranque se actualiza (reemplaza) escribiendo sobre el código 18 del cargador de arranque existente (etapa 23). Sin embargo, si el microcontrolador 15 pierde alimentación o se reinicia de otro modo mientras se está copiando el nuevo código del cargador de arranque (etapa 24), el dispositivo puede volverse inutilizable (es decir, "bloqueado") (etapa 25).

Las realizaciones proporcionan una aplicación de actualización del cargador de arranque que es operable para actualizar el código 18 de cargador de arranque existente almacenado en la memoria 16 no volátil con nuevo código del cargador de arranque. Las realizaciones pueden reemplazar el firmware del cargador de arranque de una manera a prueba de fallos, es decir, de una manera que sea resiliente a fallos tales como una pérdida de alimentación o un reinicio inesperado.

Según diversas realizaciones, una aplicación de actualización del cargador de arranque se proporciona inicialmente al microcontrolador 15 y se almacena como una aplicación en la memoria 16 no volátil. Esto se puede hacer de una manera a prueba de fallos usando el proceso convencional de actualización del código 17 de aplicación. El código de aplicación de actualización del cargador de arranque se almacena en un conjunto particular (segundo) de bloques de memoria de la memoria 16 no volátil en un conjunto particular (segundo) de direcciones de memoria para la memoria 16 no volátil.

Una vez que la aplicación de actualización del cargador de arranque se ha almacenado en la memoria 16 no volátil, el controlador puede entonces hacer funcionar (ejecutar) la aplicación de actualización del cargador de arranque. Este proceso se ilustra en la Figura 3. Como se muestra en la Figura 3, en este proceso, el microcontrolador 15 se enciende (etapa 30), el cargador de arranque se inicia (etapa 31), y se inicia el proceso de actualización del cargador de arranque (etapa 32).

A continuación, el bloque de memoria inicial (primerísimo) del primer conjunto de bloques de memoria (es decir, el bloque de memoria de la memoria 16 no volátil que almacena el primerísimo bloque de datos del código 18 de cargador de arranque) (por ejemplo, el bloque de memoria con dirección cero) se reemplaza (se sobrescribe) con un código de salto (etapa 33). Esto puede comprender borrar el grupo (fila) inicial (primerísimo) de bloques de memoria, y luego escribir el código de salto en el grupo (fila) inicial (primerísimo) de bloques de memoria. El código de salto puede almacenarse solo en el bloque de memoria inicial (primerísimo), pero también sería posible que el código de salto se almacenara en más de uno de los bloques de memoria iniciales (primerísimos) (es decir, en más de uno de los bloques de memoria del grupo (fila) inicial (primerísimo) de bloques de memoria). El código de salto incluye un código de inicio mínimo (es decir, según se requiera para establecer la tabla de vectores y el puntero de pila) y una instrucción para saltar a la primera dirección en la que se almacena el código de aplicación de actualización del cargador de arranque en la memoria 16 no volátil.

A este respecto, el Solicitante ha reconocido que si el microcontrolador 15 perdiera alimentación (externa), puede estar presente internamente suficiente energía en el microcontrolador 15 (por ejemplo, almacenada en uno o más condensadores del microcontrolador 15) para permitir que al menos un bloque de memoria (tal como al menos un grupo (fila) de bloques de memoria) de la memoria 16 no volátil se programe con éxito. Esta propiedad garantiza efectivamente que el código de salto se almacenará con éxito, incluso si el microcontrolador 15 pierde alimentación. En otras palabras, la etapa de copiado de código de salto (etapa 33) es efectivamente una operación "atómica", ya que su duración puede ser mucho más corta que el tiempo de descarga del circuito de alimentación del microcontrolador 15.

A continuación, el código del cargador de arranque se actualiza (reemplaza) escribiendo sobre el código 18 del cargador de arranque existente (etapa 34).

Como se muestra en la Figura 3, si el microcontrolador 15 pierde alimentación o se reinicia de otro modo

- mientras se copia el nuevo código del cargador de arranque (etapa 35), al reiniciar el microcontrolador 15 ejecutará inicialmente el código de salto (etapa 36) haciendo de este modo que el microcontrolador 15 ejecute entonces la aplicación de actualización del cargador de arranque y continúe copiando el nuevo código del cargador de arranque (etapa 37), mientras se salta el código del cargador de arranque posiblemente dañado o incompleto. Una vez que este proceso de actualización del cargador de arranque ha sido completado con éxito, la aplicación de actualización del cargador de arranque puede ser invalidada (etapa 38), y el nuevo cargador de arranque puede ser iniciado (etapa 39).
- Se apreciará que las realizaciones proporcionan una aplicación de actualización del cargador de arranque que es operable para actualizar el código 18 del cargador de arranque existente almacenado en la memoria 16 no volátil con un nuevo código del cargador de arranque de una manera a prueba de fallos. En particular, el proceso de actualización del cargador de arranque es resistente a fallos de alimentación o reinicios inesperados.
- La Figura 4 es un diagrama de flujo más detallado que ilustra el funcionamiento de la aplicación de actualización del cargador de arranque.
- Como se muestra en la Figura 4, cuando se inicia la aplicación de actualización del cargador de arranque, como etapa inicial se determina si el nuevo cargador de arranque ya se ha copiado (completamente) al primer conjunto de bloques de memoria (etapa 40).
- Si se determina que el nuevo cargador de arranque ha sido almacenado (completamente) en el primer conjunto de bloques de memoria, la aplicación de actualización del cargador de arranque puede ser deshabilitada (etapa 47) y el proceso puede terminar. Esta determinación se puede hacer cuando el microcontrolador 15 perdió alimentación durante un intento previo de actualización del cargador de arranque, por ejemplo, antes o durante la etapa de deshabilitación (etapa 47). Esta determinación también se puede hacer cuando se hace un intento de actualizar el cargador de arranque cuando esto ya se ha hecho.
- Si, por otro lado, en la etapa 40 se determina que el nuevo cargador de arranque no se ha almacenado completamente en el primer conjunto de bloques de memoria, entonces se puede hacer una determinación en cuanto a si el código de salto ya se ha almacenado en (programado para) el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria (etapa 41).
- Si se determina que el código de salto ya se ha almacenado en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria, entonces el proceso puede almacenar los bloques de datos del nuevo cargador de arranque excepto el(los) bloque(s) de datos inicial(es) en bloques de memoria del primer conjunto de bloques de memoria excepto el(los) bloque(s) de memoria inicial(es) (en los que se almacena el código de salto) (etapa 44). En otras palabras, el microcontrolador 15 almacena los bloques de datos del nuevo cargador de arranque excepto el(los) bloque(s) de datos inicial(es) (primero(s)) en bloques de memoria del primer conjunto de bloques de memoria excepto el(los) bloque(s) de memoria inicial(es) (primero(s)) del primer conjunto de bloques de memoria (en el que se almacena el código de salto). Esta determinación puede hacerse cuando el controlador pierde alimentación o se reinició de otra modo durante un intento previo de actualizar el cargador de arranque.
- Si, por otro lado, se determina que el código de salto no se ha almacenado ya en el bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria, se hace una comprobación en cuanto a si el microcontrolador 15 está recibiendo actualmente alimentación externa (etapa 42).
- Si se determina que el microcontrolador 15 está recibiendo actualmente alimentación externa, entonces el microcontrolador 15 puede proceder con la etapa de almacenar el código de salto en el bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria (etapa 43). En otras palabras, el microcontrolador 15 comprueba en primer lugar que la alimentación está presente, y luego reemplaza la primera página flash de la memoria 16 con un código de salto a la dirección de la aplicación de actualización del cargador de arranque.
- Si se determina que el microcontrolador 15 no está recibiendo alimentación externa actualmente, entonces el microcontrolador 15 puede no proceder con esta etapa, por ejemplo, cancelando el proceso de actualización del cargador de arranque o pausando el proceso de actualización del cargador de arranque (en particular pausando la escritura del código de salto al bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria) hasta que se determine que la alimentación está presente. Esto garantiza que el microcontrolador 15 tenga suficiente alimentación para completar con éxito la escritura del código de salto.
- Una vez que se ha almacenado el código de salto en el bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria, la aplicación de actualización del cargador de arranque puede proceder entonces con el proceso de almacenar bloques de datos del nuevo cargador de arranque distintos del(de los) bloque(s) de datos inicial(es) en bloques de memoria del primer conjunto de bloques de memoria distintos del(de los) bloque(s) de memoria inicial(es) (etapa 44). Se entenderá que, si el microcontrolador 15 pierde alimentación

durante este proceso, cuando se reinicia el microcontrolador 15, volverá a ejecutar la aplicación de actualización del cargador de arranque, mientras se salta el cargador de arranque posiblemente dañado o incompleto. La aplicación de actualización del cargador de arranque puede entonces completar su operación para actualizar (completamente) o reemplazar el cargador de arranque.

5

Por lo tanto, después de que se ha programado el código de salto, la aplicación puede comenzar a copiar de forma segura las nuevas páginas del cargador de arranque al primer conjunto de páginas de memoria, excepto la primera página.

10 Una vez que los bloques de datos del nuevo cargador de arranque distintos del(de los) bloque(s) de datos inicial(es) se han almacenado en el primer conjunto de bloques de memoria, el controlador puede entonces comprobar de nuevo si el controlador está recibiendo actualmente alimentación externa (etapa 45).

15 Si se determina que el microcontrolador 15 está recibiendo actualmente alimentación externa, entonces el microcontrolador 15 puede almacenar el(los) bloque(s) de datos inicial(es) del nuevo cargador de arranque en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria (es decir, sobrescribiendo el código de salto) (etapa 46). En otras palabras, antes de reemplazar la primera página flash, la aplicación comprueba que la alimentación está presente y, a continuación, programa la primera página con nuevos datos del cargador de arranque.

20

Si se determina que el microcontrolador 15 no está recibiendo alimentación externa, entonces el microcontrolador 15 puede no proceder con esta etapa, por ejemplo, cancelando el proceso de actualización del cargador de arranque o pausando el proceso de actualización del cargador de arranque (en particular pausando la escritura del(de los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria) hasta que se determine que la alimentación está presente. Esto garantiza que el microcontrolador 15 tenga suficiente alimentación para completar con éxito la escritura del(de los) bloque(s) de datos inicial(es) del nuevo cargador de arranque en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria.

25

30 Una vez que el(los) bloque(s) de datos inicial(es) del nuevo cargador de arranque se han almacenado en el(los) bloque(s) de memoria inicial(es) del primer conjunto de bloques de memoria, el proceso de actualización del cargador de arranque se completa, y la aplicación de actualización del cargador de arranque se deshabilita (etapa 47). Al reiniciar, el microcontrolador 15 ejecutará el nuevo cargador de arranque (de la manera descrita anteriormente).

35 La Figura 5 ilustra una disposición para proporcionar la aplicación de actualización del cargador de arranque según las realizaciones. Como se muestra en la Figura 5, el código 50 de aplicación de actualización del cargador de arranque puede proporcionarse junto con el código 51 de salto y también el nuevo código 52 del cargador de arranque. El código 50 de actualización del cargador de arranque es un código que ejecuta una lógica de la aplicación de actualización del cargador de arranque descrita anteriormente. El código 51 de salto puede ser el código (por ejemplo, mínimo) que necesita ser ejecutado (en un dispositivo particular) para ejecutar un salto (bifurcación) al código de aplicación de actualización del cargador de arranque.

40

Estos tres pueden combinarse para crear un firmware 53 de actualización del cargador de arranque que se almacena en la (segunda serie de bloques de memoria de la) memoria 16 no volátil (de la manera descrita anteriormente). Como se muestra en la Figura 5, el firmware 53 de actualización del cargador de arranque puede crearse enlazando código del cargador de arranque de actualización compilado con código de salto y nuevos binarios de firmware del cargador de arranque. El enlazador puede colocar los binarios del objeto de salto y del firmware del cargador de arranque en las direcciones predefinidas.

45

50 Sin embargo, son posibles varias otras realizaciones. Por ejemplo, no es necesario proporcionar el código 52 del cargador de arranque como parte del firmware 53 de actualización del cargador de arranque que está almacenado en la memoria 16 no volátil. El firmware 53 de actualización del cargador de arranque puede incluir en su lugar una instrucción o instrucciones para descargar el nuevo código 52 del cargador de arranque, por ejemplo, desde el panel de control y/o a través de una conexión de red, etc.

55

La Figura 6 ilustra esquemáticamente el estado de la memoria 16 no volátil durante las varias etapas del proceso de actualización del cargador de arranque descrito anteriormente. Como se muestra en la Figura 6A, la memoria 16 no volátil almacena inicialmente el código 18 de cargador de arranque antiguo y el código 17 de firmware de la aplicación. Como se muestra en la Figura 6B, la aplicación 17 se sobrescribe entonces por el firmware 53 de actualización del cargador de arranque, que en esta realización incluye la lógica de actualización del cargador de arranque, el código de salto y el nuevo cargador de arranque.

60

Como se muestra en la Figura 6C, el código de salto se copia en la primera dirección de memoria de la memoria 16 no volátil. A continuación, como se muestra en la Figura 6D, el código 18 del cargador de arranque antiguo se actualiza con el nuevo código del cargador de arranque (excepto para el(los) primer(os) bloque(s)). Finalmente, como se muestra en la Figura 6E, el código de salto se sobrescribe con el(los) primer(os) bloque(s)

65

del nuevo código del cargador de arranque. Al reiniciar, el microcontrolador 15 ejecutará ahora el nuevo cargador de arranque, según se desee.

5 Como se ha descrito anteriormente, las realizaciones se basan en el circuito de suministro de alimentación del microcontrolador 15 proporcionando suficiente energía para permitir la programación segura de al menos una (o más de una) página flash después de que el microcontrolador 15 se desconecte de la fuente de alimentación. Basándose en esta propiedad, la lógica de actualización reemplaza inicialmente la primera página del cargador de arranque antiguo con un código de salto a la dirección del código de firmware de actualización, lo que garantiza que el proceso de actualización del cargador de arranque continuará incluso después de una
10 interrupción inesperada (por ejemplo, pérdida de alimentación).

Por lo tanto, para proporcionar un proceso de actualización a prueba de fallos que sea resistente a la pérdida de alimentación, el dispositivo debe configurarse de manera que su fuente de alimentación proporcione suficiente energía para la programación segura de al menos una sola página flash después de la pérdida de
15 alimentación externa.

La Figura 7 ilustra un proceso para determinar el número de bloques de memoria de la memoria 16 no volátil que puede programarse cuando el microcontrolador 15 no está recibiendo alimentación externa. El algoritmo presentado mide la duración del tiempo de apagado y el número de páginas programadas durante ese tiempo.
20 Como se muestra en la Figura 7, en este proceso, el microcontrolador 15 se enciende inicialmente (etapa 60) y la alimentación se deshabilita a continuación (etapa 61), y se hace que el microcontrolador 15 programe un número de páginas (etapa 62). Cuando eventualmente el microcontrolador 15 se reinicia debido a la pérdida de alimentación (etapa 63), se puede determinar el número de páginas programadas con éxito (etapa 64).

25 Si el número de páginas programadas durante el periodo sin alimentación es mayor que algún umbral mínimo deseado (por ejemplo, > 8), el algoritmo de actualización del cargador de arranque puede aplicarse de forma segura en el campo. Sin embargo, si el número de páginas programadas es menor que el umbral deseado, la capacitancia en la placa de circuito impreso puede aumentarse para satisfacer la condición anterior.

30 Se apreciará que varias realizaciones proporcionan un firmware de actualización del cargador de arranque que es resistente a fallos de alimentación durante un proceso de actualización del cargador de arranque.

La descripción detallada anterior se ha presentado a efectos ilustrativos y descriptivos. No pretende ser exhaustiva ni limitar la tecnología a la forma precisa descrita. Son posibles muchas modificaciones y variaciones a la luz de la enseñanza anterior. Las realizaciones descritas fueron elegidas para explicar mejor
35 los principios de la tecnología y su aplicación práctica, para permitir de este modo a otros expertos en la técnica utilizar mejor la tecnología en varias realizaciones y con varias modificaciones que sean adecuadas al uso particular contemplado. Se pretende que el alcance se defina mediante las reivindicaciones adjuntas a la presente memoria.

REIVINDICACIONES

1. Un método para actualizar firmware almacenado en una memoria (16) no volátil de un controlador (15), en donde la memoria (16) no volátil comprende un primer conjunto de bloques de memoria configurados para almacenar un cargador de arranque para el controlador (15), el método que comprende:
 - 5 almacenar una aplicación de actualización del cargador de arranque en un segundo conjunto de bloques de memoria de la memoria (16) no volátil;
 - 10 reemplazar un bloque de memoria inicial del primer conjunto de bloques de memoria con un código de salto, en donde el código de salto comprende una instrucción para saltar a un bloque de memoria inicial del segundo conjunto de bloques de memoria;
 - 15 almacenar un nuevo cargador de arranque para el controlador (15) en la memoria (16) no volátil una vez que el código de salto se ha almacenado en el bloque de memoria inicial del primer conjunto de bloques de memoria almacenando bloques de datos del nuevo cargador de arranque distintos de un bloque de datos inicial del nuevo cargador de arranque en bloques de memoria del primer conjunto de bloques de memoria distintos del bloque de memoria inicial;
 - 20 determinar si el controlador (15) está recibiendo alimentación externa; y
 - almacenar el bloque de datos inicial del nuevo cargador de arranque en el bloque de memoria inicial del primer conjunto de bloques de memoria sobrescribiendo el código de salto;
 - 25 en donde, cuando el controlador (15) pierde alimentación externa, se proporciona energía internamente al controlador (15) para que al menos un bloque de memoria de la memoria (16) no volátil pueda borrarse y escribirse sin que el controlador (15) reciba alimentación externa con el fin de garantizar que el código de salto se almacenará con éxito.
- 30 2. El método de la reivindicación 1, que comprende además determinar si el controlador (15) está recibiendo alimentación externa antes de almacenar el código de salto en el bloque de memoria inicial del primer conjunto de bloques de memoria.
- 35 3. El método de la reivindicación 1 o 2, que comprende además determinar si el código de salto ya se ha almacenado en el bloque de memoria inicial del primer conjunto de bloques de memoria antes de almacenar el código de salto en el bloque de memoria inicial del primer conjunto de bloques de memoria y/o antes de determinar si el controlador (15) está recibiendo alimentación externa.
- 40 4. El método de una cualquiera de las reivindicaciones precedentes, que comprende además determinar si se ha almacenado un nuevo cargador de arranque en el primer conjunto de bloques de memoria antes de almacenar el código de salto en el bloque de memoria inicial del primer conjunto de bloques de memoria y/o antes de determinar si el controlador está recibiendo alimentación externa y/o antes de determinar si el código de salto ya se ha almacenado en el bloque de memoria inicial del primer conjunto de bloques de memoria.
- 45 5. El método de cualquier reivindicación precedente, en donde la energía almacenada en uno o más condensadores del controlador (15) se proporciona internamente al controlador cuando el controlador (15) pierde alimentación externa.
- 50 6. Un método para actualizar firmware almacenado en una memoria no volátil de un controlador integrado de un dispositivo de protección contra incendios o un dispositivo de detección de intrusiones, comprendiendo el método actualizar el firmware del controlador integrado usando el método de una cualquiera de las reivindicaciones precedentes.
- 55 7. Un programa informático que comprende un código de software informático para realizar el método de una cualquiera de las reivindicaciones precedentes cuando el programa se ejecuta en medios de procesamiento de datos.
8. Un controlador configurado para realizar el método de una cualquiera de las reivindicaciones 1-6.
- 60 9. Un dispositivo de protección contra incendios o un dispositivo de detección de intrusiones que comprende el controlador de la reivindicación 8.
10. Un sistema que comprende:

un controlador (15) que comprende una memoria (16) no volátil configurada para almacenar firmware para el controlador (15), en donde la memoria (16) no volátil comprende un primer conjunto de bloques de memoria configurados para almacenar un cargador de arranque para el controlador (15);

- 5 en donde el sistema está configurado para actualizar el firmware mediante:

almacenar una aplicación de actualización del cargador de arranque en un segundo conjunto de bloques de memoria de la memoria (16) no volátil;

- 10 reemplazar un bloque de memoria inicial del primer conjunto de bloques de memoria con un código de salto, en donde el código de salto comprende una instrucción para saltar a un bloque de memoria inicial del segundo conjunto de bloques de memoria;

- 15 almacenar un nuevo cargador de arranque para el controlador (15) en la memoria (16) no volátil una vez que el código de salto se ha almacenado en el bloque de memoria inicial del primer conjunto de bloques de memoria almacenando bloques de datos del nuevo cargador de arranque distintos de un bloque de datos inicial del nuevo cargador de arranque en bloques de memoria del primer conjunto de bloques de memoria distintos del bloque de memoria inicial;

- 20 determinar si el controlador (15) está recibiendo alimentación externa; y

almacenar el bloque de datos inicial del nuevo cargador de arranque en el bloque de memoria inicial del primer conjunto de bloques de memoria sobrescribiendo el código de salto;

- 25 en donde, el sistema está configurado de manera que, cuando el controlador (15) pierde alimentación externa, se proporciona energía internamente al controlador (15) para que al menos un bloque de memoria de la memoria (16) no volátil pueda borrarse y escribirse sin que el controlador (15) reciba alimentación externa con el fin de garantizar que el código de salto se almacenará con éxito.

- 30 11. El sistema de la reivindicación 10, en donde:

el sistema está configurado para determinar si el controlador (15) está recibiendo alimentación externa antes de almacenar el código de salto en el bloque de memoria inicial del primer conjunto de bloques de memoria.

- 35 12. El sistema de una cualquiera de las reivindicaciones 10 u 11, en donde el sistema es un sistema de protección contra incendios o un sistema de detección de intrusiones.

Fig. 1

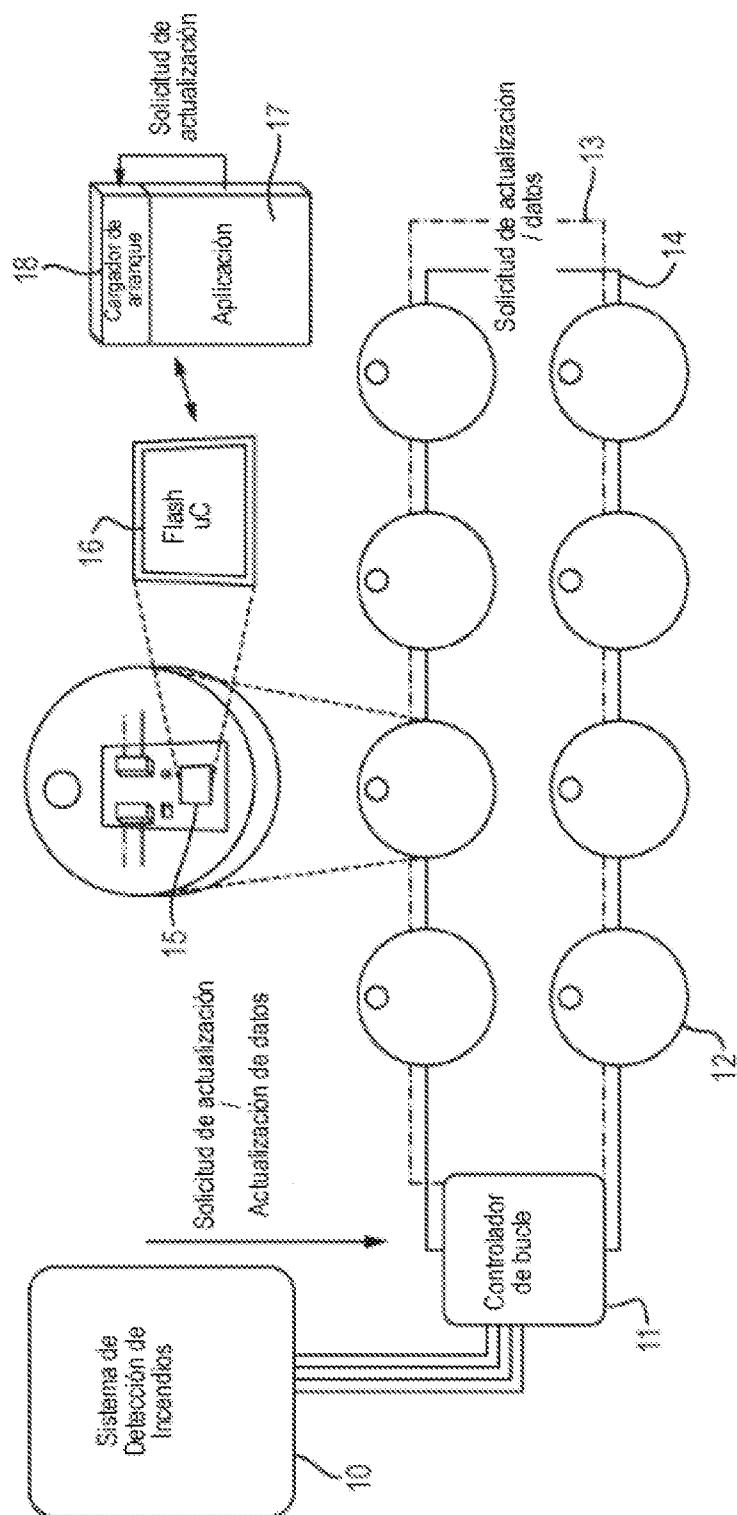


Fig. 2

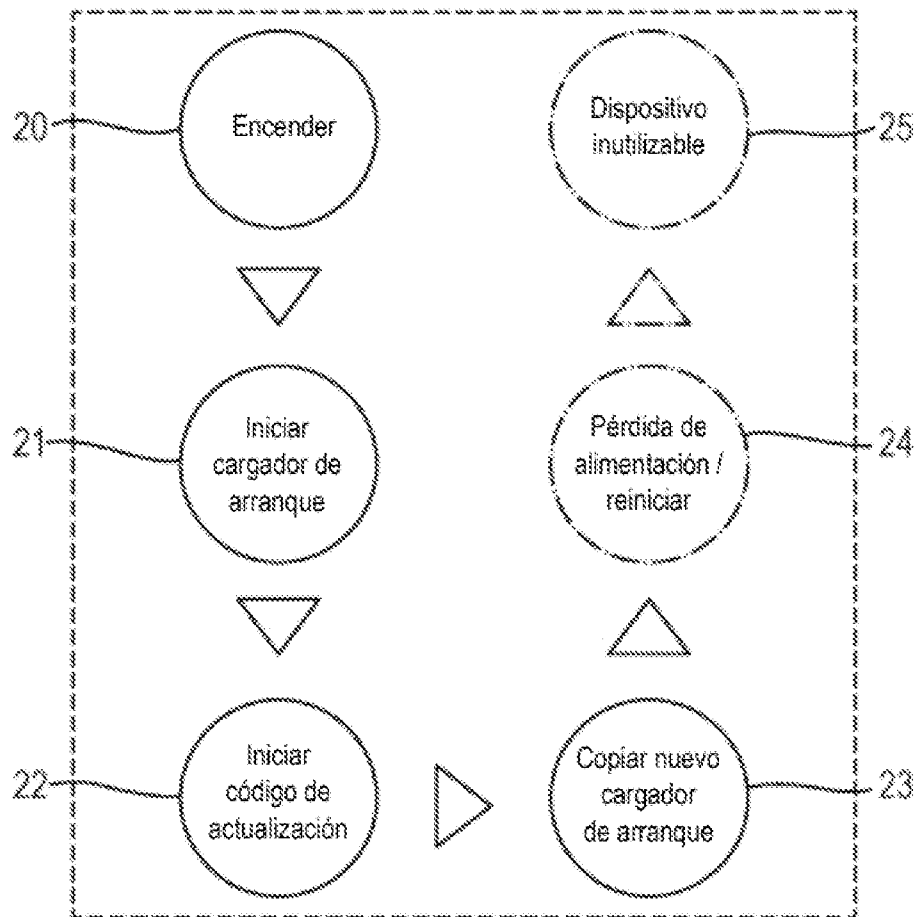


Fig. 3

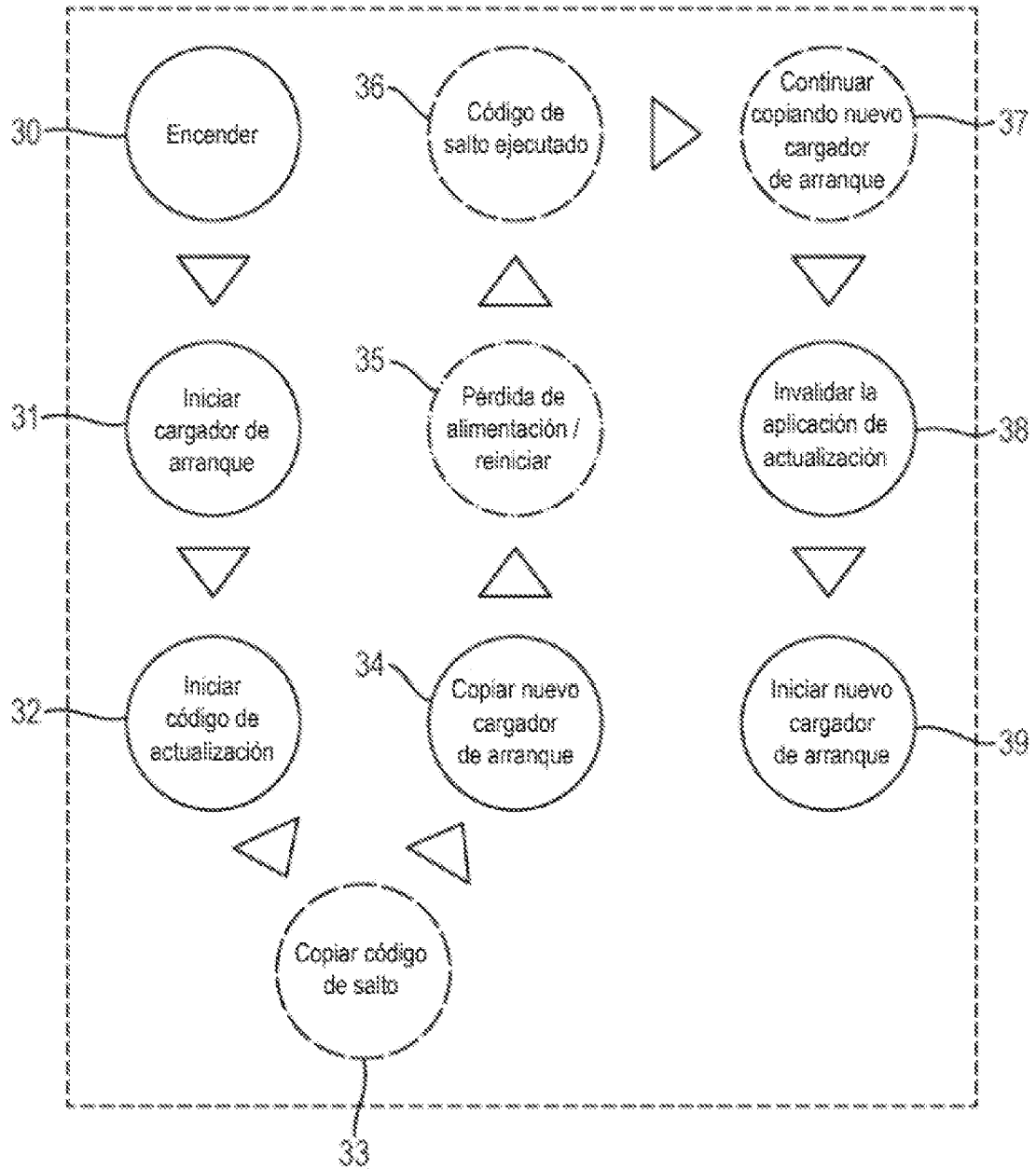


Fig. 4

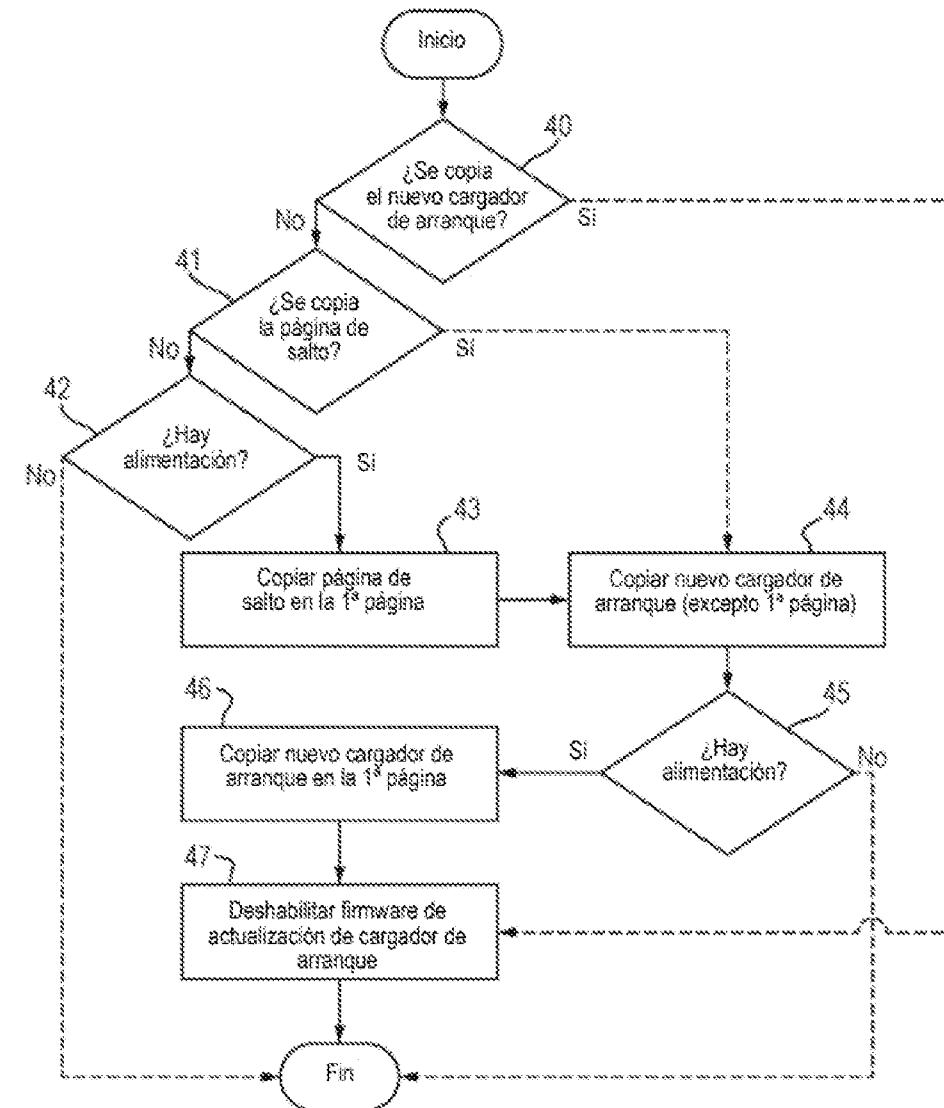


Fig. 5

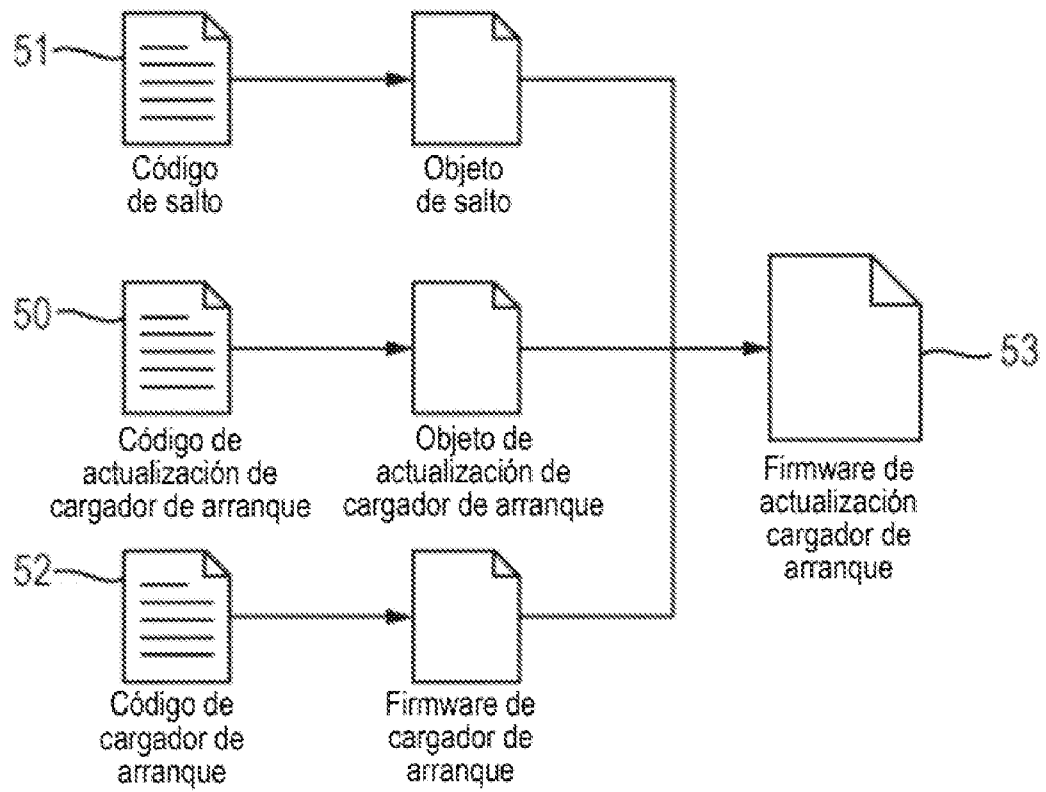


Fig. 6

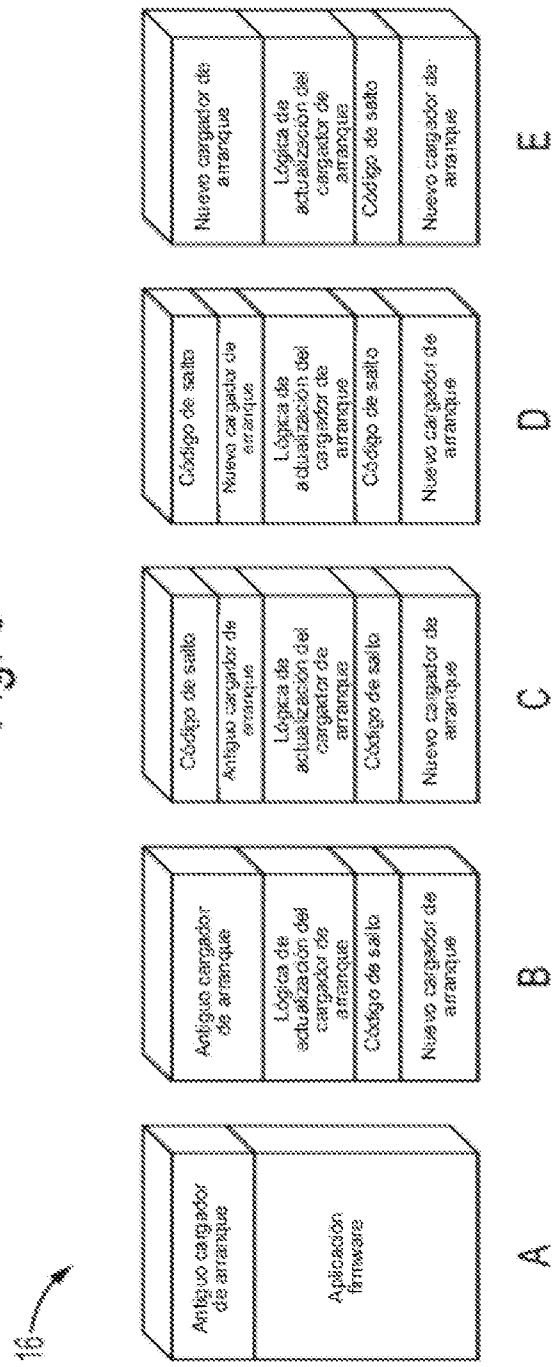


Fig. 7

