[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 11/08 (2006.01)
G06F 7/58 (2006.01)

[12] 发明专利说明书

专利号 ZL 02159862.2

[45] 授权公告日 2007 年 8 月 29 日

[11] 授权公告号 CN 100334555C

[22] 申请日 2002.12.27 [21] 申请号 02159862.2

[73] 专利权人 技嘉科技股份有限公司

地址 台湾省台北县

[72] 发明人 黄添寿 陈允迪 郭仲轩

[56] 参考文献

US5210854A 1993.5.11

EP1256865A3 2003.1.2

审查员 陈晓华

[74] 专利代理机构 隆天国际知识产权代理有限公司

代理人 潘培坤 楼仙英

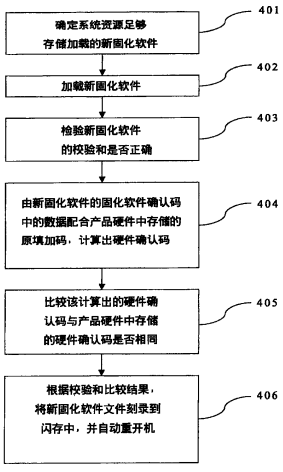
权利要求书 3 页 说明书 10 页 附图 4 页

[54] 发明名称

智能型固化软件的升级控制方法

[57] 摘要

本发明公开了一种智能型固化软件升级控制方法，包括：生成硬件确认码；生成固化软件确认码；及根据固化软件确认码与硬件确认码的校验比较，升级固化软件程序。其通过一套可管理产品软、硬件版本的编码方式，根据编码数据，检验固化软件确认码并比较硬件确认码，对用户升级的固化软件的完整性和正确性进行检查，以确保固化软件文件没有因疏忽或人为蓄意被修改，并且与其硬件规格相匹配，防止将不当的固化软件版本烧录进产品中，同时，在进行固化软件升级前可先检测系统资源是否足以完成升级的工作，避免升级动作遭到中途失败。



1．一种智能型固化软件升级控制方法，在一硬件上记录有一硬件确认码和一填加码，在一固化软件文件上记录有一固化软件确认码，该固化软件确认码由一软件版本码、一硬件版本码、该填加码、该硬件确认码、一校验和与一随机数所组成，该智能型固化软件升级控制方法包括下列步骤：

加载新固化软件；

校验该新固化软件文件，根据校验结果产生一新校验和，并判断该新校验和与该校验和是否相同；

如果该新校验和与该校验和不同，则显示警告讯息通知使用者；

如果该新校验和与该校验和相同，则根据该新固化软件的固化软件确认码的数据和该硬件上的该填加码，计算出一新硬件确认码，并对比该新硬件确认码与该硬件确认码是否相同；

如果该新硬件确认码与该硬件确认码不同，则显示警告讯息通知使用者；以及

如果该新硬件确认码与该硬件确认码相同，则将该新固化软件文件烧录到一闪存。

2．如权利要求 1 所述的智能型固化软件升级控制方法，其特征在于，该硬件版本码的数据格式包含下列各字段：

Vender/Product ID 字段：用于识别产品制造者及产品名称；

V 字段：使用 1 字节记录制造者数据；

C 字段：使用 1 字节记录 CPU 数据，由于 CPU 是产品中的重要元件，不同 CPU 将会使用不同的固化软件版本，故以 1 字节记录；

P 字段：使用 1 字节记录产品代号数据；

H 字段：使用 1 字节记录硬件(电路板)数据；及

R 字段：保留 1 字节作为以后延伸用途。

3．如权利要求 1 所述的智能型固化软件升级控制方法，其特征在于，

所述软件版本码的数据格式包含下列各字段：

A 字段：以一个字节记录主要的版本；

BBB 的第一个字节字段：修正重要 BUG；

BBB 的第二、第三个字节字段：修正次要 BUG；及

CC 字段：以一个字节记录特别版本。

4．如权利要求 1 所述的智能型固化软件升级控制方法，其特征在于，所述固化软件确认码的数据格式包含下列各字段：

Vender/Product ID 字段：用于标识产品制造者及产品名称；

V 字段：使用 1 字节记录制造者数据；

C 字段：使用 1 字节记录 CPU 数据；

P 字段：使用 1 字节记录产品代号数据；

H 字段：使用 1 字节记录硬件(电路板)数据；

R 字段：保留 1 字节作为以后延伸用途；

CS 字段：使用 1 字节记录“校验和”数据；

Salt 字段：使用 1 字节记录“填加码”数据；

H()字段：使用 1 字节记录硬件确认码数据；

Software dynamic version 字段：使用 6 字节记录软件版本数据；及

Rand1,Rand2 字段：在未使用空间填上随机数。

5．如权利要求 1 所述的智能型固化软件升级控制方法，其特征在于，该硬件确认码由一算法计算所产生，其中该算法使用到的数据包含该硬件版本码、该填加码及一密码。

6．如权利要求 5 所述的智能型固化软件升级控制方法，其特征在于，所述填加码为随机自动产生，且每运算一次就随机产生一次，每次各不相同。

7．如权利要求 5 所述的智能型固化软件升级控制方法，其特征在于，所述计算使用的算法是 XOR 运算。

8．如权利要求 5 所述的智能型固化软件升级控制方法，其特征在于，所述计算使用的算法是 MD5 运算。

9．如权利要求 1 所述的智能型固化软件升级控制方法，其特征在于，

该硬件确认码和该填加码一同烧录到该硬件中的一非易失性内存。

10. 如权利要求 9 所述的智能型固化软件升级控制方法，其特征在于，所述非易失性的内存是复杂可编程逻辑设备。

11. 如权利要求 9 所述的智能型固化软件升级控制方法，其特征在于，所述非易失性的内存是电子可擦可编程只读存储器。

12. 如权利要求 1 所述的智能型固化软件升级控制方法，其特征在于，在加载该新固化软件之前还包含一检测系统资源的步骤，如检测到系统资源不足时，则显示警告信息通知使用者。

13. 如权利要求 1 所述的智能型固化软件升级控制方法，其特征在于，在校验该新固化软件文件的步骤中还包含一检查该新固化软件的固化软件确认码的硬件确认码和填加码与该硬件上记录的该硬件确认码和该填加码是否相符合的步骤。

智能型固化软件的升级控制方法

技术领域

本发明涉及一种程序的升级控制方法，特别涉及一种智能型固化软件的升级控制方法。

背景技术

固化软件即被写入设备硬件中的只读存储器上的软件，载有在用户环境中不能加以改变的计算机程序及数据等。是计算机系统中最关键的组件之一。

一般而言，某一产品推出后，往往由于加入新功能或是要修正原有错误，需要更新产品内的固化软件。因此，对硬件产品内的固化软件进行升级已成为目前发展的趋势，是延长硬件产品生命期的有效方法，对用户是一种投资上的保护，也会让硬件变得更耐用。

然而，对于固化软件的升级有两个相互冲突的要求：（1）它应该很好的被保护，一旦他被修改后破坏，整个系统将无法工作；（2）它应该很容易的被修改，从而允许现场升级以进行性能提高或清除软件故障。

因此，当进行固化软件的升级时，应当非常的小心和谨慎，因为升级修改的过程本身是很容易的，但如果在这个过程中，不小心出现操作失误，或是升级了不正确的固化软件版本，那么后果又是十分严重的，有时会造成整个系统的无法工作。

然而，在现有的固化软件升级过程中，并没有针对上述危险提供任何的预防机制和控制方法，用户常常会因此操作过程中的疏忽，或是选用了错误的固化软件版本，而造成升级失败，甚至是系统瘫痪。

发明内容

本发明针对上述问题而提供一种智能型固化软件的升级控制方法，其主要目的在于通过一套可管理产品软、硬件版本的编码方式，根据编码数据，对用户升级的固化软件的完整性和正确性进行检查，以确保固化软件文件没有因疏忽或人为蓄意被修改，并且与其硬件规格相匹配，防止将不当的固化软件版本烧录进产品中。

为解决上述问题本发明提供一种智能型固化软件升级控制方法，在一硬件上记录有一硬件确认码和一填加码，在一固化软件文件上记录有一固化软件确认码，该固化软件确认码由一软件版本码、一硬件版本码、该填加码、该硬件确认码、一校验和与一随机数所组成，该智能型固化软件升级控制方法包括下列步骤：加载新固化软件；校验该新固化软件文件，根据校验结果产生一新校验和，并判断该新校验和与该校验和是否相同；如果该新校验和与该校验和不同，则显示警告讯息通知使用者；如果该新校验和与该校验和相同，则根据该新固化软件的固化软件确认码的数据和该硬件上的该填加码，计算出一新硬件确认码，并对比该新硬件确认码与该硬件确认码是否相同；如果该新硬件确认码与该硬件确认码不同，则显示警告讯息通知使用者；以及如果该新硬件确认码与该硬件确认码相同，则将该新固化软件文件烧录到一闪存。

本发明提供的一种智能型固化软件升级控制方法，包括：生成硬件确认码；生成固化软件确认码；以及根据固化软件确认码与硬件确认码的校验比较，升级固化软件程序。

如上所述的智能型固化软件升级控制方法，其中，所述生成硬件确认码的步骤还包括有如下步骤：

确定硬件版本码；确定密码；确定填加码；确定算法；按照算法，将各码值运算，其结果作为硬件确认码；及将填加码和硬件确认码一同写入硬件。

如上所述的智能型固化软件升级控制方法，其中，所述生成固化软件确认码的步骤还包括有如下步骤：填写软件版本码和硬件版本码；计算并填写硬件确认码和填加码；计算并填写完整固化软件文件的校验和；及在未使用的空间中填入随机产生的随机数。

如上所述的智能型固化软件升级控制方法，其中，所述根据固化软件确认码与硬件确认码的校验比较，升级固化软件程序的步骤还包括有如下步骤：确定系统资源足够存贮加载的新固化软件；加载新固化软件；检验新固化软件的校验和是否正确；由新固化软件的固化软件确认码中的数据配合产品硬件中存储的原填加码，计算出硬件确认码；比较该计算出的硬件确认码与产品硬件中存储的硬件确认码是否相同；及根据校验和比较结果，将新固化软件文件烧录到闪存中，并自动重开机。

本发明的有益效果是，本发明的方法在加载的固化软件文件时，通过对校验和的确认，确保文件内容不会因人为蓄意或疏失而被修改；由比较加载的固化软件与硬件内部的版本确认码，可以确定固化软件版本与硬件规格、产品类型是否相符，防止使用者有意或无意地升级不正确的版本；另外本发明的方法在加载固化软件文件前，会先检测系统资源(如内存空间)是否足够，并主动提醒使用者，避免造成因系统资源不足造成升级失败。

为对本发明的目的、构造特征及其功能有进一步的了解，以下配合附图作详细说明如下。

附图说明

图 1 是本发明的智能型固化软件的升级控制方法流程图；

图 2 是本发明的生成硬件确认码的流程图；

图 3 是本发明的生成固化软件确认码的流程图；

图 4 是本发明根据固化软件确认码与硬件确认码的校验对比，升级固化软件程序的流程图。

其中，附图标记说明如下：

步骤 101：生成硬件确认码

步骤 102：生成固化软件确认码

步骤 103：根据固化软件确认码与硬件确认码的校验对比，升级固化软件程序

步骤 201：确定硬件版本码

步骤 202：确定密码

步骤 203：确定填加码（Salt）

步骤 204：确定算法

步骤 205：按照算法，将各码值运算，其结果作为硬件确认码

步骤 206：将填加码和硬件确认码一同写入硬件

步骤 301：填写软件版本码和硬件版本码

步骤 302：计算并填写硬件确认码和填加码

步骤 303：计算并填写完整固化软件文件的校验值（Checksum）

步骤 304：在未使用的空间中填入随机产生的随机数

步骤 401：确定系统资源足够存贮加载的新固化软件

步骤 402：加载新固化软件

步骤 403：检验新固化软件的校验和（Checksum）是否正确

步骤 404：由新固化软件的固化软件确认码中的数据配合产品硬件中存储的原填加码，计算出硬件确认码

步骤 405：比较该计算出的硬件确认码与产品硬件中存储的硬件确认码是否相同

步骤 406：根据校验值比较结果，将新固化软件文件烧录到闪存（Flash Memory）中，并自动重开机

具体实施方式

本发明所揭露的智能型固化软件升级控制方法设计了一套可管理产品软硬件版本的编码方式。将一个产品的版本分为两类：与硬件有关的“硬件版本”以及与软件升级有关的“软件版本”。以下以一优选实施例来说明：

1、硬件版本：用于辨认硬件和固化软件之间的兼容性，其数据格式包含以下字段：

Vender/ProductID	V	C	P	H	R

Vender/Product ID：用于辨识产品制造者及产品名称。

V：使用 1 字节记录制造者数据。

C：使用 1 字节记录 CPU 数据，由于 CPU 是产品中的重要组件，不同 CPU 将会使用不同的固化软件版本，故以 1 字节记录。

P：使用 1 字节记录产品代号数据。

H：使用 1 字节记录硬件(电路板)数据。

R：保留 1 字节作为以后延伸用途。

例如：

Vender/Product ID											V	C	P	H	R
G	I	G	A	B	Y	T	E	4	0	1	G	C	B	1	0

公司一般往往会发展一系列相关的产品，也许有不同的硬件组件（如 CPU），或是电路板有不同的修改。皆可以由硬件版本中的字段（C：CPU；H：电路版版本）控制。同时，如果为其它厂商代工时，也可由硬件版本中的 V(Vender)字段分别不同的客户，避免为 A 客户生产的固化软件，可被烧录进 B 客户的产品中。以保护自己及代工客户的权益。

2、软件版本：用于在同一硬件上，分别不同固化软件版本之间的差异，其数据格式包含以下字段：

A	B	B	B	C	C

A：以一个字节记录主要的版本(如主要功能的增加)。

BBB 的第一个字节：修正重要 BUG。

BBB 的第二、第三个字节：修正次要 BUG。

CC：以一个字节记录特别版本(如不同的 OEM 客户或是为特定目的制作的版本)。

例如：

A	B	B	B	C	C
1	0	3	4	0	1

请参阅图 1，为本发明的智能型固化软件的升级控制方法流程图。首先，

在步骤 101，产生硬件确认码；然后，在步骤 102，产生固化软件确认码；最后，在步骤 103，根据固化软件确认码与硬件确认码的校验及对比，升级固化软件程序。

如图 2 所示，为本发明生成硬件确认码的流程图。首先，在步骤 201，确定硬件版本码；然后在步骤 202，确定密码；步骤 203，确定填加码（Salt）；接着，在步骤 204 中确定算法；然后，步骤 205 中，按照算法，将各码值运算，其结果作为硬件确认码；最后，步骤 206 中，将填加码和硬件确认码一同写入硬件。

其中所述密码由产品开发者保存，并不公开。

所述填加码（Salt）为随机产生，且每执行运算一次则随机产生一次，每次都不相同，所以计算出来的硬件确认码也随之不同，因此，需要将该次运算所使用的填加码与计算出来的硬件确认码一同写入产品硬件中，以备升级固化软件时作校验对比用。

所述算法可以使用各种的算法，本实施例将“硬件版本码”，“密码”和“填加码”依每一个字节分别异或（XOR）组成最后的硬件确认码，其具有简单，快速，且具有基本的安全性等优点。以下举一运算实例：

硬件版本	G	I	G	A	B	Y	T	E	4	0	1	G	C	B	1	0
(HEX 值)	0x47	0x49	0x47	0x41	0x42	0x59	0x54	0x45	0x34	0x30	0x31	0x47	0x43	0x42	0x31	0x30
	XOR															
密码	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F	0x10
	XOR															
Salt	0x0A	0x0A	0x0A	0x0A	0x0A	0x0A	0x0A	0x0A	0x0A	0x0A	0x0A	0x0A	0x0A	0x0A	0x0A	0x0A
	=															
硬件确认码	0x4C	0x41	0x4E	0x4F	0x4D	0x55	0x59	0x47	0x37	0x30	0x30	0x41	0x44	0x46	0x34	0x2A

所述将填加码和硬件确认码一同写入硬件，可以是将其烧录到硬件中的

某一块非易失性的内存中，例如 CPLD(Complex Programmable Logic Device，复杂可编程逻辑设备）或是 EEPROM（Electrically Erasable Programmable Read Only Memory，电可擦除可编程只读存储器）。

如图 3 所示，为本发明的生成固化软件确认码的流程图。首先，在步骤 301 中，填写软件版本码和硬件版本码；然后在步骤 302，计算并填写硬件确认码和填加码；然后在步骤 303 中，计算并填写完整固化软件文件的校验值（Checksum）；最后在步骤 304 中，及在未使用的空间中填入随机产生的随机数。

其中，可以由固定的硬件版本和密码值，以及随机产生的填加码（Salt）值算出硬件确认码。由于每一次均使用不同的填加码，也就会产生不同的硬件确认码，然后将填加码与硬件确认码同样存于固化软件确认码中，如此可以增加破解硬件确认码的难度。

计算校验和(Checksum)，可以选用不同的校验和(Checksum)算法，本实施例将已填入硬件确认码的整个固化软件内容以异或 XOR 的方式产生一个字节的确认码。

在未使用的空间中填入随机数，其目的在于增加破解硬件确认码的难度，由于每一次的“Salt”皆为随机数，所产生的硬件确认码和随机数就具有不可分辨性，在未使用的空间填入随机数后，试图破解硬件版本的恶意者将很难由单纯观察固化软件文件而破解硬件版本算法。

完整的固化软件确认码的数据格式为：

<table>
<tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td>20h</td><td colspan="11">Vender/Product ID</td><td>V</td><td>C</td><td>P</td><td>H</td><td>R</td></tr>
<tr><td>30h</td><td>CS</td><td colspan="2">Rand1</td><td>salt</td><td colspan="12">H(version(20h-2Fh),key, salt(33h))</td></tr>
<tr><td>40h</td><td colspan="6">Software Dynamic Version</td><td colspan="11">Rand2</td></tr>
</table>

Vender/Product ID：用于辨识产品制造者及产品名称。

V：使用 1 字节记录制造者数据。

C：使用 1 字节记录 CPU 数据，由于 CPU 是产品中的重要组件，不同

CPU 将会使用不同的固化软件版本，故以 1 字节记录。

P：使用 1 字节记录产品代号数据。

H：使用 1 字节记录硬件(电路板)数据。

R：保留 1 字节作为以后延伸用途。

CS：使用 1 字节记录“Checksum”数据。

Salt：使用 1 字节记录“Salt”数据。

H()：使用 1 字节记录硬件确认码数据。

Software dynamic version：使用 6 字节记录软件版本数据。

Rand1, Rand2：在未使用空间填上随机数。

下面以一实例来说明固化软件确认码的产生过程：

1、填入软件明文版本、硬件明文版本

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20h	G	I	G	A	B	Y	T	E	4	0	1	G	C	B	1	0
30h																
40h	1	0	1	2	3	4										

2、计算并填入硬件确认码版本，填加码(Salt)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20h	G	I	G	A	B	Y	T	E	4	0	1	G	C	B	1	0
30h				salt	H(version(20h-2Fh), key, salt(33h))											
40h	1	0	1	2	3	4										

3、计算并填入完整固化软件文件的校验和(checksum)。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20h	G	I	G	A	B	Y	T	E	4	0	1	G	C	B	1	0
30h	CS			salt	H(version(20h-2Fh), key, salt(33h))											
40h	1	0	1	2	3	4										

4、在未使用的空间中填入随机产生的随机数。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20h	G	I	G	A	B	Y	T	E	4	0	1	G	C	B	1	0
30h	CS	Rand1		salt	H(version(20h-2Fh), key, salt(33h))											
40h	1	0	1	2	3	4	Rand2									

请参阅图 4，为本发明根据固化软件确认码与硬件确认码的校验对比，升级固化软件程序的流程图。首先，在步骤 401 中，确定系统资源足够存贮加载新的固化软件；然后，在步骤 402 中，加载新固化软件；然后在步骤 403 中检验新固化软件的校验和（Checksum）是否正确；在接下来的步骤 404 中，由新固化软件的固化软件确认码中的数据配合产品硬件中存储的原填加码，计算出硬件确认码；步骤 405 中，比较该计算出的硬件确认码与产品硬件中存储的硬件确认码是否相同；最后，在步骤 406 中，根据校验和比较结果，将新固化软件文件烧录到闪存（Flash Memory）中，并自动重开机。

其中，如果检测到系统资源不足时，则显示警告信息通知使用者重新开机。

如果校验结果或比较结果不正确，则同样显示警告信息通知使用者重新开机。

本发明为一种智能型固化软件的升级控制方法，其有益效果为：

1、加载的固化软件文件由校验和(Checksum)确认，确保文件内容不会因人为蓄意或疏失而被修改。

2、由比较加载的固化软件与硬件内部的版本确认码，可以确定固化软件版本与硬件规格、产品类型是否相符。防止使用者有意或无意地升级不正确的版本。如，检查固化软件确认码的硬件确认码和填加码是否与硬件上记录的硬件确认码和填加码相符合。

3、加载固化软件文件前，会先检测系统资源(如内存空间)是否足够，并主动提醒使用者，避免造成因系统资源不足造成升级失败。

4、易于辨识固化软件版本，在固化软件中可直接辨识固化软件的版本，

适用的产品及硬件规格，且不会危害固化软件升级的正确性。

5、对于试图破解硬件确认码以升级非法固化软件(使用 A 客户的固化软件，烧录到 B 客户的产品中)的使用者，设计有多重机制以保护固化软件文件。(1)整个固化软件文件具有校验和(Checksum)，任意更动其中的数据将会造成校验和(Checksum)计算错误。(2)计算硬件确认码的算法中需要一组密码，此密码是不公开的。(3)由于每次计算都会使用随机数 Salt，所以每次产生的硬件确认码与随机数不可分辨，并且固化软件中未使用的空间皆填上随机数。除了软硬件版本明文以外，将不易猜出哪几个字节是硬件确认码及 Salt，将更增加破解硬件确认码算法困难。

6、可以弹性地依需要更改其中的硬件确认码算法及校验和(Checksum)算法，如想要有效快速，可以使用异或（XOR）计算。如需要更强的安全性，可以使用更复杂的运算方式，如 MD5（Hash 算法的一种）或其它的密码方法。

虽然本发明以前述的较佳实施例揭露如上，但是并非用以限定本发明，任何本技术领域的普通技术人员，在不脱离本发明的精神和范围内，所做出的等效结构变化，均包含在本发明的专利范围内。

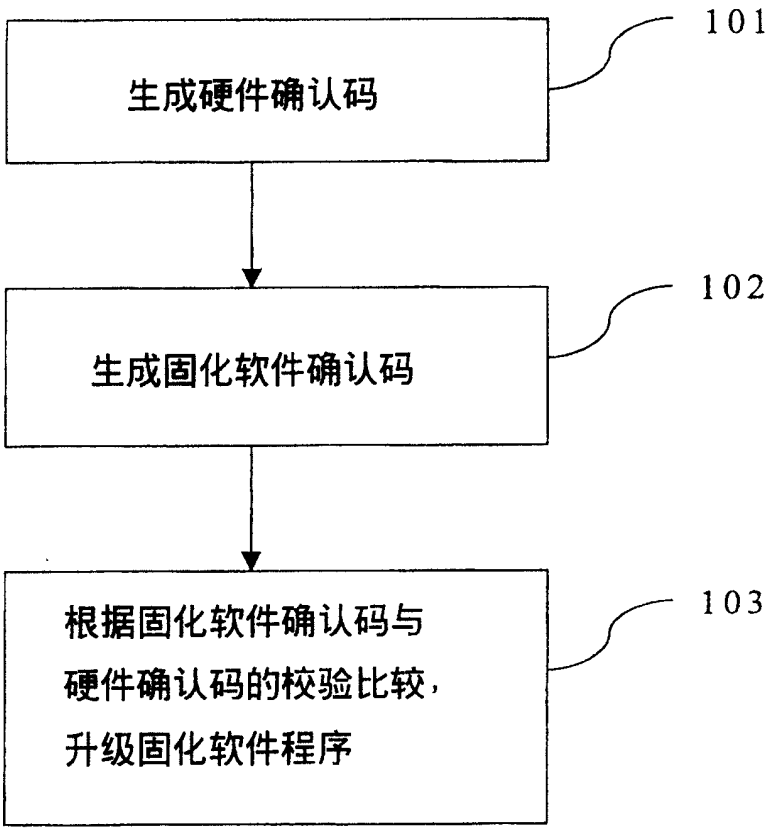

101
生成硬件确认码
102
生成固化软件确认码
103
根据固化软件确认码与
硬件确认码的校验比较，
升级固化软件程序

图 1

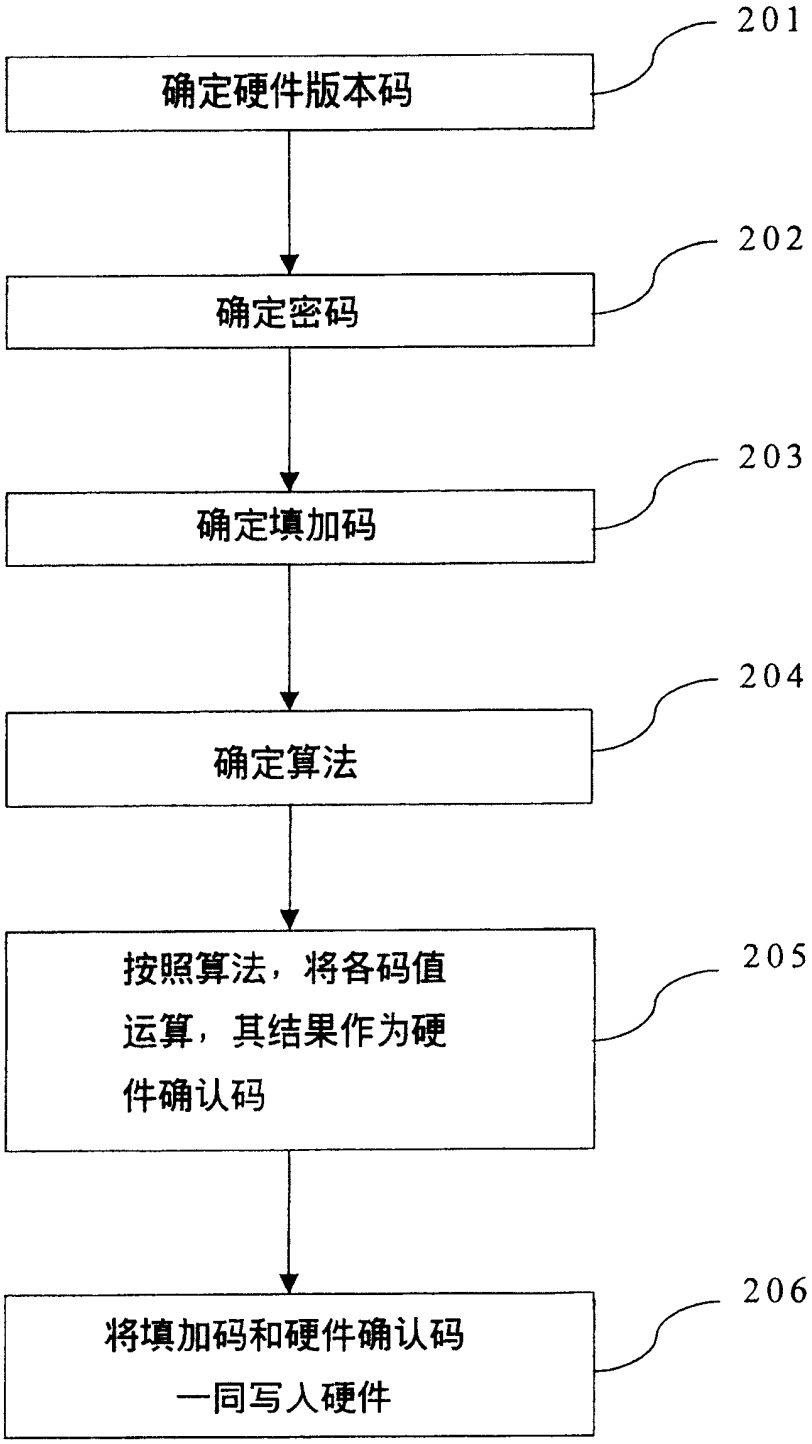

201
确定硬件版本码
202
确定密码
203
确定填加码
204
确定算法
205
按照算法，将各码值运算，其结果作为硬件确认码
206
将填加码和硬件确认码一同写入硬件

图 2

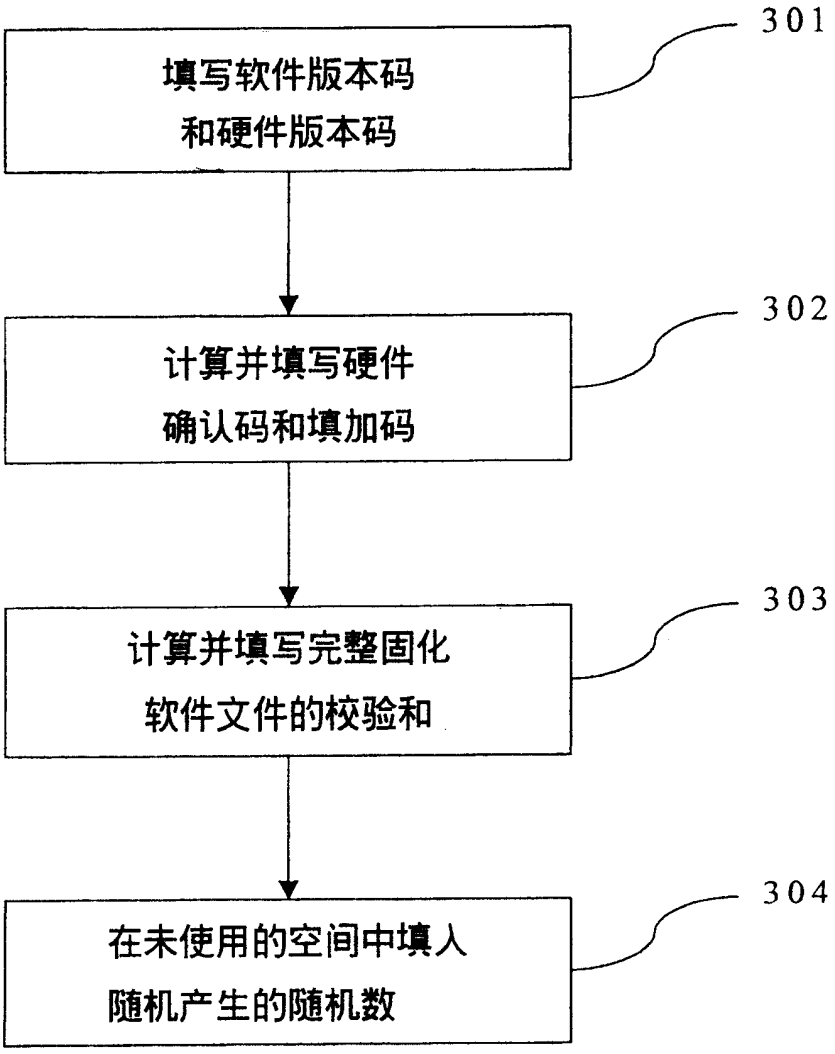

301
填写软件版本码
和硬件版本码
302
计算并填写硬件
确认码和填加码
303
计算并填写完整固化
软件文件的校验和
304
在未使用的空间中填入
随机产生的随机数

图 3

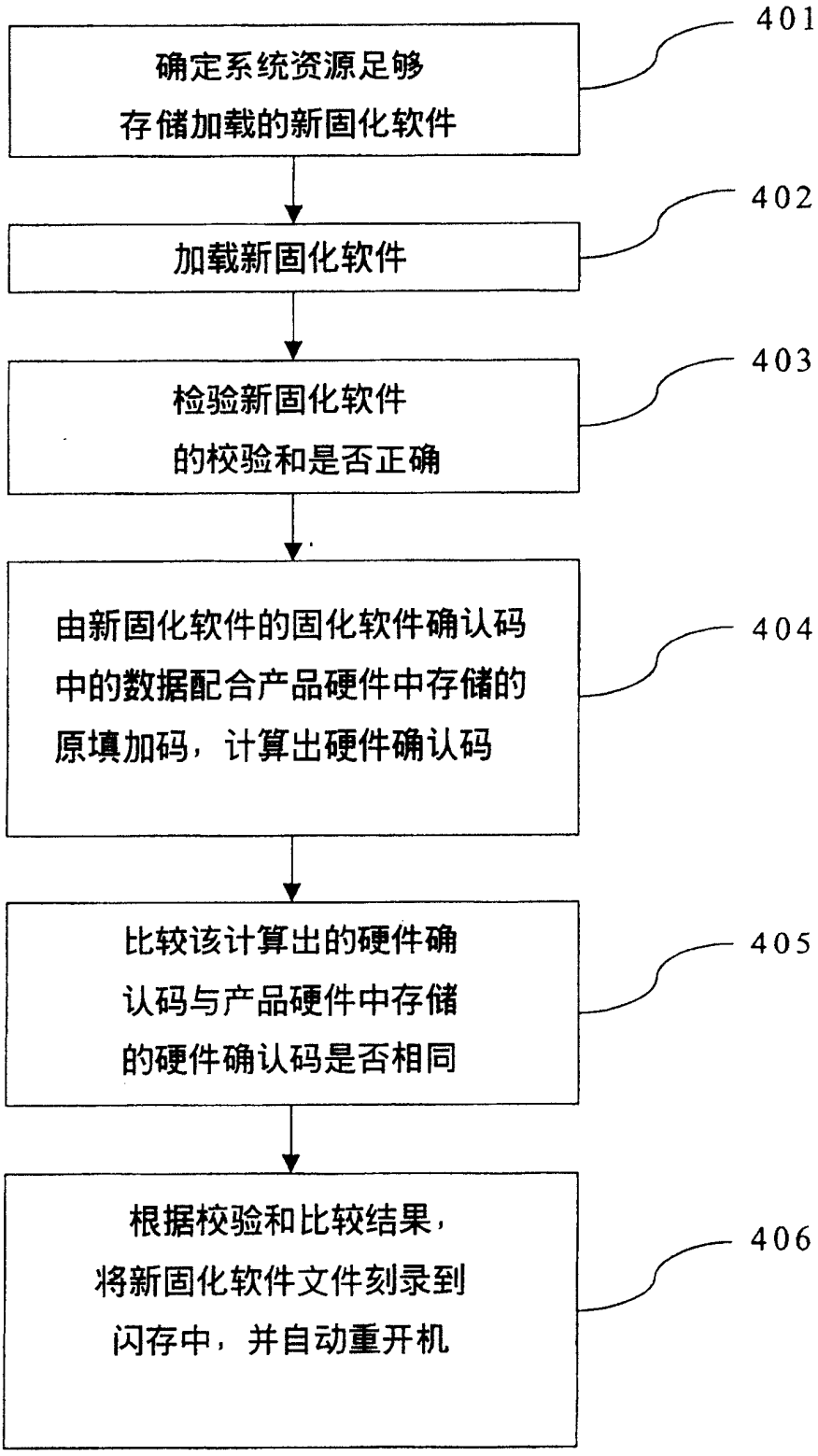

401
确定系统资源足够
存储加载的新固化软件
402
加载新固化软件
403
检验新固化软件
的校验和是否正确
404
由新固化软件的固化软件确认码
中的数据配合产品硬件中存储的
原填加码，计算出硬件确认码
405
比较该计算出的硬件确
认码与产品硬件中存储
的硬件确认码是否相同
406
根据校验和比较结果，
将新固化软件文件刻录到
闪存中，并自动重开机

图 4