



(12)发明专利申请

(10)申请公布号 CN 109800565 A

(43)申请公布日 2019.05.24

(21)申请号 201711136495.0

(22)申请日 2017.11.16

(71)申请人 深圳市优必选科技有限公司

地址 518000 广东省深圳市南山区学苑大
道1001号南山智园C1栋16、22楼

(72)发明人 熊友军 张礼富

(74)专利代理机构 深圳中一联合知识产权代理
有限公司 44414

代理人 张全文

(51)Int.Cl.

G06F 21/51(2013.01)

G06F 21/12(2013.01)

G06F 8/65(2018.01)

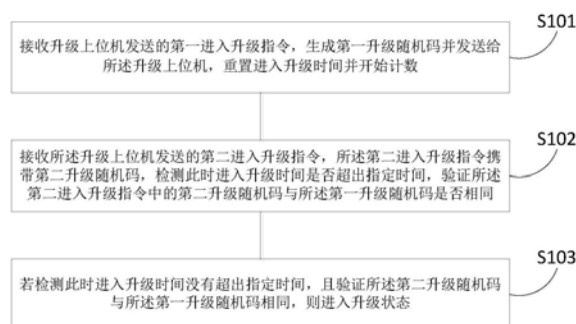
权利要求书2页 说明书8页 附图3页

(54)发明名称

软件升级方法及终端设备

(57)摘要

本发明适用于软件升级技术领域,提供了一种软件升级方法及终端设备,其中软件升级方法包括:接收升级上位机发送的第一进入升级指令,生成第一升级随机码并发送给升级上位机,重置进入升级时间并开始计数,接收升级上位机发送的第二进入升级指令,若检测此时进入升级时间没有超出指定时间,且验证第二进入升级指令携带的第二升级随机码与第一升级随机码相同,则进入升级状态。本发明提供的一种软件升级方法及终端设备降低了其他指令因误码变成进入升级指令而让软件进入升级状态的风险,避免出现软件误入升级状态、误改程序内容的问题,还使整个升级过程更不容易被破解,大大提高了软件升级过程的稳定性和安全性。



1. 一种软件升级方法,其特征在于,包括以下步骤:

接收升级上位机发送的第一进入升级指令,生成第一升级随机码并发送给所述升级上位机,重置进入升级时间并开始计数;

接收所述升级上位机发送的第二进入升级指令,所述第二进入升级指令携带第二升级随机码,检测此时进入升级时间是否超出指定时间,验证所述第二进入升级指令中的第二升级随机码与所述第一升级随机码是否相同;

若检测此时进入升级时间没有超出指定时间,且验证所述第二升级随机码与所述第一升级随机码相同,则进入升级状态。

2. 根据权利要求1所述的软件升级方法,其特征在于,还包括:

在所述升级状态下,接收所述升级上位机发送的升级数据帧,所述升级数据帧携带加密升级数据;

对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级。

3. 根据权利要求2所述的软件升级方法,其特征在于,在所述接收所述升级上位机发送的升级数据帧之后,还包括:

生成回应升级数据指令,根据所述回应升级数据指令发送下一帧要接收的数据地址和上一帧接收到的加密数据至所述升级上位机。

4. 根据权利要求2所述的软件升级方法,其特征在于,所述对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级包括:

对所述加密升级数据进行解密,将解密后的数据存放于接收缓存中;

当接收缓存中存储数据的大小达到预设值时,将接收缓存中存储的数据写入程序存储空间,对软件进行升级。

5. 根据权利要求2所述的软件升级方法,其特征在于,在所述将解密后的数据写入程序存储空间之后,所述对软件进行升级之前,还包括:

对写入程序存储空间中的数据进行全局验证。

6. 根据权利要求2所述的软件升级方法,其特征在于,在所述对软件进行升级之后,还包括:

若升级成功,则生成升级成功提示;

若升级失败,则重新执行对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级的步骤。

7. 一种软件升级系统,其特征在于,包括:

第一进入升级指令接收模块,用于接收升级上位机发送的第一进入升级指令,生成第一升级随机码并发送给所述升级上位机,重置进入升级时间并开始计数;

第二进入升级指令接收模块,用于接收所述升级上位机发送的第二进入升级指令,所述第二进入升级指令携带第二升级随机码,检测此时进入升级时间是否超出指定时间,验证所述第二进入升级指令中的第二升级随机码与所述第一升级随机码是否相同;

升级状态判定模块,用于当检测此时进入升级时间没有超出指定时间,且验证所述第二升级随机码与所述第一升级随机码相同时,进入升级状态。

8. 根据权利要求7所述的软件升级系统,其特征在于,还包括:

升级数据帧接收模块,用于在所述升级状态下,接收所述升级上位机发送的升级数据帧,所述升级数据帧携带加密升级数据;

数据解密模块,用于对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级。

9.一种软件升级终端设备,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现如权利要求1至6任一项所述软件升级方法的步骤。

10.一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至6任一项所述软件升级方法的步骤。

软件升级方法及终端设备

技术领域

[0001] 本发明属于软件升级技术领域,尤其涉及一种软件升级方法及终端设备。

背景技术

[0002] 现有的软件升级协议,往往会出现为了追求实现简单使程序无验证进入升级,致使升级程序整体的稳定性及安全性欠佳。比如无验证进入升级一般做法是收到升级指令后进入升级状态,但实际的传输总是无法做到100%可靠,加之实际应用中存在着各种干扰因素,传输数据存在着误码情况,这样便不可避免地存在着其它功能指令被误码成升级指令的情况,进而出现软件误入升级状态、误修改程序内容情况。

发明内容

[0003] 有鉴于此,本发明实施例提供了一种软件升级方法及终端设备,以解决现有技术中软件升级协议不存在进入升级验证,容易出现软件误入升级状态、误改程序内容的问题。

[0004] 为解决上述问题,本发明实施例的第一方面提供了一种软件升级方法,包括以下步骤:

[0005] 接收升级上位机发送的第一进入升级指令,生成第一升级随机码并发送给所述升级上位机,重置进入升级时间并开始计数;

[0006] 接收所述升级上位机发送的第二进入升级指令,所述第二进入升级指令携带第二升级随机码,检测此时进入升级时间是否超出指定时间,验证所述第二进入升级指令中的第二升级随机码与所述第一升级随机码是否相同;

[0007] 若检测此时进入升级时间没有超出指定时间,且验证所述第二升级随机码与所述第一升级随机码相同,则进入升级状态。

[0008] 进一步地,所述软件升级方法还包括:

[0009] 在所述升级状态下,接收所述升级上位机发送的升级数据帧,所述升级数据帧携带加密升级数据;

[0010] 对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级。

[0011] 进一步地,在所述接收所述升级上位机发送的升级数据帧之后,所述软件升级方法还包括:

[0012] 生成回应升级数据指令,根据所述回应升级数据指令发送下一帧要接收的数据地址和上一帧接收到的加密数据至所述升级上位机。

[0013] 进一步地,所述对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级包括:

[0014] 对所述加密升级数据进行解密,将解密后的数据存放于接收缓存中;

[0015] 当接收缓存中存储数据的大小达到预设值时,将接收缓存中存储的数据写入程序存储空间,对软件进行升级。

[0016] 进一步地,在所述将解密后的数据写入程序存储空间之后,所述对软件进行升级之前,所述软件升级方法还包括:

[0017] 对写入程序存储空间中的数据进行全局验证。

[0018] 进一步地,在所述对软件进行升级之后,所述软件升级方法还包括:

[0019] 若升级成功,则生成升级成功提示;

[0020] 若升级失败,则重新执行对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级的步骤。

[0021] 本发明实施例的第二方面,提供了一种软件升级系统,包括:

[0022] 第一进入升级指令接收模块,用于接收升级上位机发送的第一进入升级指令,生成第一升级随机码并发送给所述升级上位机,重置进入升级时间并开始计数;

[0023] 第二进入升级指令接收模块,用于接收所述升级上位机发送的第二进入升级指令,所述第二进入升级指令携带第二升级随机码,检测此时进入升级时间是否超出指定时间,验证所述第二进入升级指令中的第二升级随机码与所述第一升级随机码是否相同;

[0024] 升级状态判定模块,用于当检测此时进入升级时间没有超出指定时间,且验证所述第二升级随机码与所述第一升级随机码相同时,进入升级状态。

[0025] 进一步地,所述软件升级系统还包括:

[0026] 升级数据帧接收模块,用于在所述升级状态下,接收所述升级上位机发送的升级数据帧,所述升级数据帧携带加密升级数据;

[0027] 数据解密模块,用于对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级。

[0028] 本发明实施例的第三方面提供了一种软件升级终端设备,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现上述第一方面的软件升级方法。

[0029] 本发明实施例的第四方面提供了一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现上述第一方面的软件升级方法。

[0030] 本发明实施例与现有技术相比存在的有益效果是:通过对进入升级指令进行进入升级时间和升级随机码双重验证,降低了其他指令因误码变成进入升级指令而让软件进入升级状态的风险,避免出现软件误入升级状态、误改程序内容的问题,还使整个升级过程更不容易被破解,大大提高了软件升级过程的稳定性和安全性。

附图说明

[0031] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0032] 图1是本发明实施例提供的一种软件升级方法的示意图;

[0033] 图2是本发明实施例提供的一种软件升级系统的示意性框图;

[0034] 图3是本发明实施例提供的一种软件升级终端设备的结构示意图;

[0035] 图4是本发明实施例提供的一种验证进入升级指令的示意图；

[0036] 图5是本发明实施例提供的一种升级数据帧的结构示意图；

[0037] 图6是本发明实施例提供的一种回应升级数据帧的结构示意图。

具体实施方式

[0038] 以下描述中,为了说明而不是为了限定,提出了诸如特定系统结构、技术之类的具体细节,以便透彻理解本发明实施例。然而,本领域的技术人员应当清楚,在没有这些具体细节的其它实施例中也可以实现本发明。在其它情况中,省略对众所周知的系统、装置、电路以及方法的详细说明,以免不必要的细节妨碍本发明的描述。

[0039] 为了说明本发明所述的技术方案,下面通过具体实施例来进行说明。

[0040] 参照图1,本发明实施例提供的一种软件升级方法,包括以下步骤:

[0041] S101,接收升级上位机发送的第一进入升级指令,生成第一升级随机码并发送给所述升级上位机,重置进入升级时间并开始计数。

[0042] 具体地,升级上位机发送的第一进入升级指令中含有预设的或者随机生成的升级随机码,在生成第一升级随机码后,将第一升级随机码发送给升级上位机,也可以是将携带有第一升级随机码的第一进入升级指令发送给升级上位机,升级上位机识别第一进入升级指令,并获取第一进入升级指令中携带的第一升级随机码。

[0043] S102,接收所述升级上位机发送的第二进入升级指令,所述第二进入升级指令携带第二升级随机码,检测此时进入升级时间是否超出指定时间,验证所述第二进入升级指令中的第二升级随机码与所述第一升级随机码是否相同。

[0044] S103,若检测此时进入升级时间没有超出指定时间,且验证所述第二升级随机码与所述第一升级随机码相同,则进入升级状态。

[0045] 具体地,正常升级流程要求升级上位机程序在指定时间内先后发送两次升级指令才能顺利进入升级状态,第一次发送进入升级指令用于获取升级随机码,并重置进入升级时间并开始计数,第二次在指定时间内一并发送进入升级指令和得到的升级随机码,验证升级随机码通过后最终进入升级状态。

[0046] 如图4所示,以舵机升级为例,一个实施例中舵机进行升级时验证进入升级指令的具体流程包括:接收升级上位机发送的第一进入升级指令,若检测进入升级时间计数为零,表明接收该进入升级指令之前没有接收其他升级指令,给该进入升级指令匹配一个升级随机码,并重置进入升级时间计数器开始计数;若检测进入升级时间计数不为零,表明接收该进入升级指令之前接收过其他升级指令,此时判断进入升级时间计数是否在指定时间内,如果进入升级时间计数不在指定时间内,则指令超时,给该进入升级指令重新匹配一个升级随机码,并重置进入升级时间计数器开始计数,如果进入升级时间计数在指定时间内,则进一步验证该进入升级指令中的升级随机码是否与上一进入升级指令中的升级随机码相同,如果升级验证码验证相同,则舵机进入升级状态,如果升级验证码验证不相同,则给该进入升级指令重新匹配一个升级随机码,并重置进入升级时间计数器开始计数。

[0047] 接收升级上位机发送的第二进入升级指令,第二进入升级指令携带第二升级随机码,如果在接收第一进入升级指令后,舵机没有进入升级状态,则此时进入升级时间计数不为零,判断进入升级时间计数是否在指定时间内,如果进入升级时间计数在指定时间内,则

进一步判断第二进入升级指令中的升级随机码是否与上一进入升级指令中的升级随机码相同,即判断第二进入升级指令中的升级随机码是否与第一进入升级指令中的升级随机码相同,如果升级随机码验证相同,则舵机进入升级状态。

[0048] 具体地,可以利用计数器对进入升级时间进行计数。

[0049] 该流程的限时验证机制很大程度地降低了其它指令因误码变成进入升级指令让程序进入升级状态的风险,同时加入了随机码匹配功能,提高安全性的同时还使整个升级过程更不容易被破解。

[0050] 进一步地,所述软件升级方法还包括:

[0051] 在所述升级状态下,接收所述升级上位机发送的升级数据帧,所述升级数据帧携带加密升级数据。

[0052] 具体地,在升级过程中,升级上位机以升级数据帧的格式传输数据,如图5所示,升级数据帧包括升级数据指令、数据地址和加密数据,这里的数据地址并非程序数据在存储器中的绝对地址,而是相对于整个升级数据内容的相对地址,如升级数据总大小为4096字节,其对应数据地址范围是0-4096。加密数据指原始的升级数据按某一约定的加密方法进行加密后的升级数据。

[0053] 对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级。

[0054] 通过上述描述可知,本发明实施例通过对进入升级指令进行进入升级时间和升级随机码双重验证,降低了其他指令因误码变成进入升级指令而让软件进入升级状态的风险,避免出现软件误入升级状态、误改程序内容的问题,还使整个升级过程更不容易被破解,大大提高了软件升级过程的稳定性和安全性。

[0055] 进一步地,在所述接收所述升级上位机发送的升级数据帧之后,所述软件升级方法还包括:

[0056] 生成回应升级数据指令,根据所述回应升级数据指令发送下一帧要接收的数据地址和上一帧接收到的加密数据至所述升级上位机。

[0057] 具体地,在升级过程中,如果成功接收到一个升级数据帧,根据升级数据帧,生成回应升级数据指令,要求对所接收到的升级数据帧进行回应,回应内容包括回应升级数据指令、下一帧接收的数据地址和上一帧接收到的加密数据,如图6所示,可以以回应升级数据帧的格式将回应内容发送给升级上位机,根据该回应升级数据帧的内容,升级上位机可以知道上一帧数据是否被成功接收,并决定是重发上一帧数据还是下发下一帧要数据,这样,发送程序及接收程序均可以很清楚地知道所要处理的数据内容,数据传输的同步性高,使整机升级过程更加稳定可靠。

[0058] 进一步地,所述对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级包括:

[0059] 对所述加密升级数据进行解密,将解密后的数据存放于接收缓存中。

[0060] 当接收缓存中存储数据的大小达到预设值时,将接收缓存中存储的数据写入程序存储空间,对软件进行升级。

[0061] 具体地,现将接收到的加密数据解密后存放在接收缓存中,当接收缓存中存储的解密后的数据的大小达到预设值,或者全部升级数据接收完成时,再将接收缓存中存储的

数据写入到程序存储空间,其中接收缓存空间的大小可以根据实际应用制定,定量存储机制可以减少对程序存储空间的写入操作次数,保证程序的稳定性,提高升级效率。

[0062] 进一步地,在所述将解密后的数据写入程序存储空间之后,所述对软件进行升级之前,所述软件升级方法还包括:

[0063] 对写入程序存储空间中的数据进行全局验证。

[0064] 具体地,在接受完全部升级数据后,还要对写入到程序存储空间中的升级数据进行全局验证,例如:对升级数据进行全局CRC (Cyclic Redundancy Check,循环冗余校验) 校验,可以进一步确保升级的完整性和正确性。

[0065] 进一步地,在所述对软件进行升级之后,所述软件升级方法还包括:

[0066] 若升级成功,则生成升级成功提示。

[0067] 若升级失败,则重新执行对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级的步骤。

[0068] 参照图2,本发明的另一实施例提供了一种软件升级系统,包括:

[0069] 第一进入升级指令接收模块201,用于接收升级上位机发送的第一进入升级指令,生成第一升级随机码并发送给所述升级上位机,重置进入升级时间并开始计数。

[0070] 第二进入升级指令接收模块202,用于接收所述升级上位机发送的第二进入升级指令,所述第二进入升级指令携带第二升级随机码,检测此时进入升级时间是否超出指定时间,验证所述第二进入升级指令中的第二升级随机码与所述第一升级随机码是否相同。

[0071] 升级状态判定模块203,用于当检测此时进入升级时间没有超出指定时间,且验证所述第二升级随机码与所述第一升级随机码相同时,进入升级状态。

[0072] 进一步地,所述软件升级系统还包括:

[0073] 升级数据帧接收模块,用于在所述升级状态下,接收所述升级上位机发送的升级数据帧,所述升级数据帧携带加密升级数据。

[0074] 数据解密模块,用于对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级。

[0075] 进一步地,所述软件升级系统还包括:

[0076] 回应升级数据指令生成模块,用于生成回应升级数据指令,根据所述回应升级数据指令发送下一帧要接收的数据地址和上一帧接收到的加密数据至所述升级上位机。

[0077] 进一步地,所述软件升级系统还包括:

[0078] 全局验证模块,用于对写入程序存储空间中的数据进行全局验证。

[0079] 进一步地,所述数据解密模块包括:

[0080] 解密数据缓存模块,用于对所述加密升级数据进行解密,将解密后的数据存放于接收缓存中。

[0081] 数据写入模块,用于当接收缓存中存储数据的大小达到预设值时,将接收缓存中存储的数据写入程序存储空间,对软件进行升级。

[0082] 进一步地,所述软件升级系统还包括:

[0083] 升级成功处理模块,用于当升级成功,生成升级成功提示。

[0084] 升级失败处理模块,用于当升级失败,重新执行对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级的步骤。

[0085] 参照图3,本发明又一实施例提供了一种软件升级终端设备,如图3所示,该实施例的软件升级终端设备3包括:处理器30、存储器31以及存储在所述存储器31中并可在所述处理器30上运行的计算机程序32,例如软件升级程序。所述处理器30执行所述计算机程序32时实现上述各个软件升级方法实施例中的步骤,例如图1所示的步骤S101至S103。或者,所述处理器30执行所述计算机程序32时实现上述各装置实施例中各模块/单元的功能,例如图2所示模块201至203的功能。

[0086] 示例性的,所述计算机程序32可以被分割成一个或多个模块/单元,所述一个或者多个模块/单元被存储在所述存储器31中,并由所述处理器30执行,以完成本发明。所述一个或多个模块/单元可以是能够完成特定功能的一系列计算机程序指令段,该指令段用于描述所述计算机程序32在所述软件升级终端设备3中的执行过程。例如,所述计算机程序32可以被分割成同步模块、汇总模块、获取模块、返回模块(虚拟装置中的模块),各模块的具体功能如下:

[0087] 接收升级上位机发送的第一进入升级指令,生成第一升级随机码并发送给所述升级上位机,重置进入升级时间并开始计数。

[0088] 接收所述升级上位机发送的第二进入升级指令,所述第二进入升级指令携带第二升级随机码,检测此时所述进入升级时间是否超出指定时间,验证所述第二进入升级指令中的第二升级随机码与所述第一升级随机码是否相同。

[0089] 若检测此时所述进入升级时间没有超出指定时间,且验证所述第二升级随机码与所述第一升级随机码相同,则进入升级状态。

[0090] 进一步地,各模块的具体功能还包括:

[0091] 在所述升级状态下,接收所述升级上位机发送的升级数据帧,所述升级数据帧携带加密升级数据。

[0092] 对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级。

[0093] 进一步地,各模块的具体功能还包括:

[0094] 生成回应升级数据指令,根据所述回应升级数据指令发送下一帧要接收的数据地址和上一帧接收到的加密数据至所述升级上位机。

[0095] 进一步地,各模块的具体功能还包括:

[0096] 对所述加密升级数据进行解密,将解密后的数据存放于接收缓存中。

[0097] 当接收缓存中存储数据的大小达到预设值时,将接收缓存中存储的数据写入程序存储空间,对软件进行升级。

[0098] 进一步地,各模块的具体功能还包括:

[0099] 对写入程序存储空间中的数据进行全局验证。

[0100] 进一步地,各模块的具体功能还包括:

[0101] 若升级成功,则生成升级成功提示。

[0102] 若升级失败,则重新执行对所述加密升级数据进行解密,并将解密后的数据写入程序存储空间,对软件进行升级的步骤。

[0103] 所述软件升级终端设备3可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。所述软件升级终端设备可包括,但不仅限于,处理器30、存储器31。本领域技术人

员可以理解,图3仅仅是软件升级终端设备3的示例,并不构成对软件升级终端设备3的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件,例如所述软件升级终端设备还可以包括输入输出设备、网络接入设备、总线等。

[0104] 所称处理器30可以是中央处理单元(Central Processing Unit,CPU),还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0105] 所述存储器31可以是所述软件升级终端设备3的内部存储单元,例如软件升级终端设备3的硬盘或内存。所述存储器31也可以是所述软件升级终端设备3的外部存储设备,例如所述软件升级终端设备3上配备的插接式硬盘,智能存储卡(Smart Media Card,SMC),安全数字(Secure Digital,SD)卡,闪存卡(Flash Card)等。进一步地,所述存储器31还可以既包括所述软件升级终端设备3的内部存储单元也包括外部存储设备。所述存储器31用于存储所述计算机程序以及所述软件升级终端设备所需的其他程序和数据。所述存储器31还可以用于暂时地存储已经输出或者将要输出的数据。

[0106] 所属领域的技术人员可以清楚地了解到,为了描述的方便和简洁,仅以上述各功能单元、模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能单元、模块完成,即将所述装置的内部结构划分成不同的功能单元或模块,以完成以上描述的全部或者部分功能。实施例中的各功能单元、模块可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中,上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。另外,各功能单元、模块的具体名称也只是为了便于相互区分,并不用于限制本申请的保护范围。上述系统中单元、模块的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0107] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0108] 在本发明所提供的实施例中,应该理解到,所揭露的装置和方法,可以通过其它的方式实现。例如,以上所描述的系统实施例仅仅是示意性的,例如,所述模块或单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通讯连接可以是通过一些接口,装置或单元的间接耦合或通讯连接,可以是电性,机械或其它的形式。

[0109] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0110] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0111] 所述集成的模块/单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明实现上述实施例方法中的全部或部分流程,也可以通过计算机程序来指令相关的硬件来完成,所述的计算机程序可存储于一计算机可读存储介质中,该计算机程序在被处理器执行时,可实现上述各个方法实施例的步骤。其中,所述计算机程序包括计算机程序代码,所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质可以包括:能够携带所述计算机程序代码的任何实体或装置、记录介质、U盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、电载波信号、电信信号以及软件分发介质等。需要说明的是,所述计算机可读介质包含的内容可以根据司法管辖区内立法和专利实践的要求进行适当的增减,例如在某些司法管辖区,根据立法和专利实践,计算机可读介质不包括是电载波信号和电信信号。

[0112] 以上所述实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围,均应包含在本发明的保护范围之内。

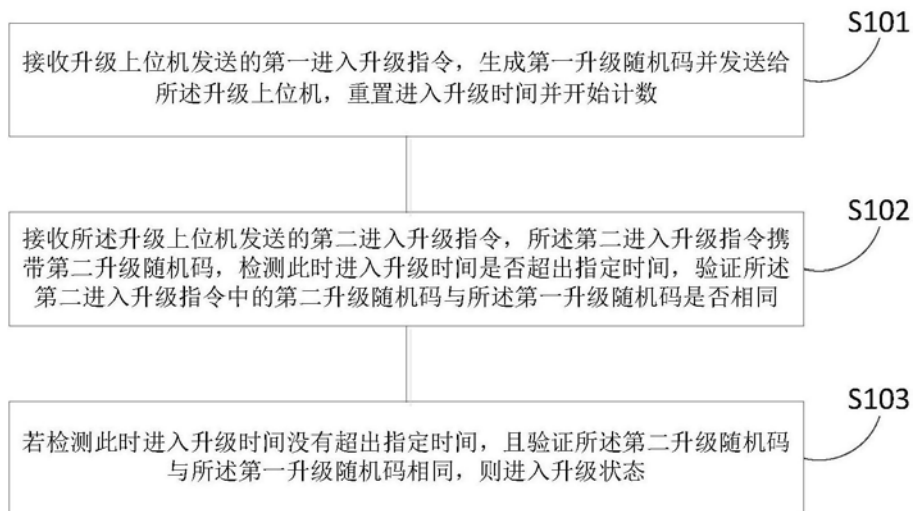


图1

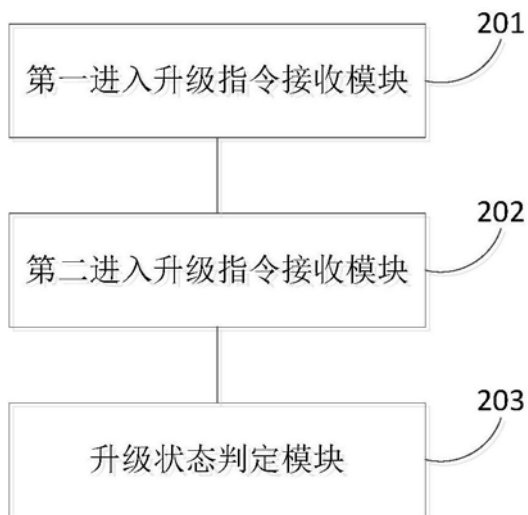


图2

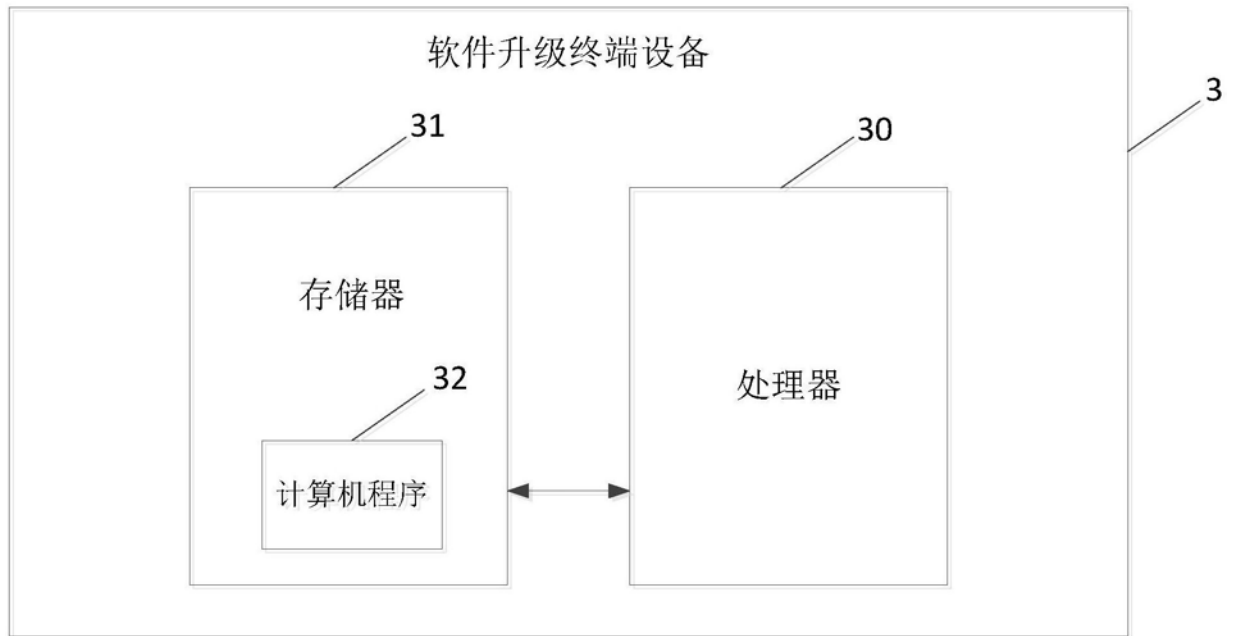


图3

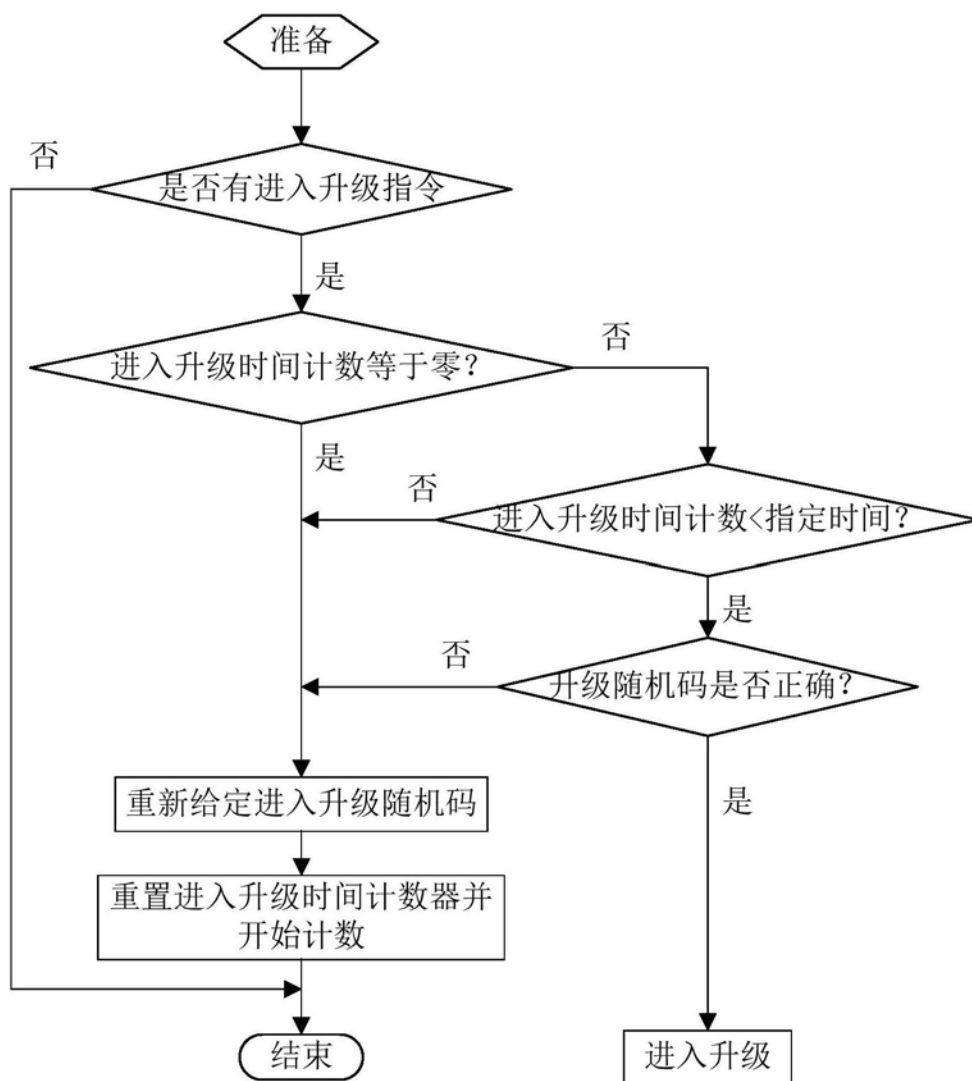


图4

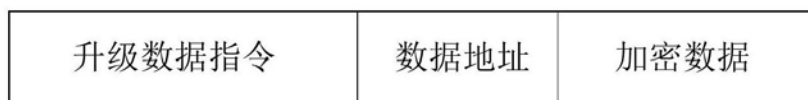


图5

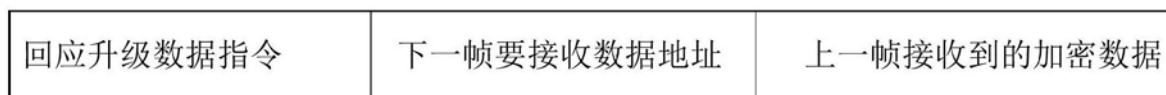


图6