



(12) 发明专利

(10) 授权公告号 CN 112530056 B

(45) 授权公告日 2023. 07. 25

(21) 申请号 202011299093.4

(22) 申请日 2020.11.18

(65) 同一申请的已公布的文献号
申请公布号 CN 112530056 A

(43) 申请公布日 2021.03.19

(73) 专利权人 深圳TCL新技术有限公司
地址 518052 广东省深圳市南山区西丽街
道中山园路1001号国际E城D4栋9楼

(72) 发明人 董文杰 王志红

(74) 专利代理机构 深圳市世纪恒程知识产权代
理事务所 44287
专利代理师 宋朝政

(51) Int. Cl.
G07C 9/00 (2020.01)

(56) 对比文件

CN 105225316 A, 2016.01.06

CN 109448178 A, 2019.03.08

WO 2019001331 A1, 2019.01.03

审查员 周红静

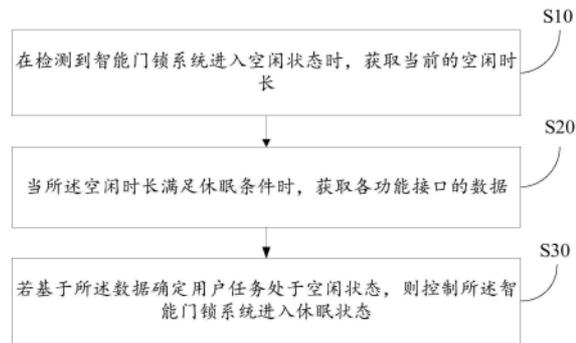
权利要求书2页 说明书10页 附图5页

(54) 发明名称

降低智能门锁功耗的方法、智能门锁和存储
介质

(57) 摘要

本申请公开了一种降低智能门锁功耗的方法、智能门锁和存储介质,降低智能门锁功耗的方法包括:在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长;当空闲时长满足休眠条件时,获取各功能接口的数据;若基于数据确定用户任务处于空闲状态,则控制智能门锁系统进入休眠状态。解决了由于智能门锁功能模块较多,导致智能门锁的功耗较高,需要频繁更换电池的技术问题,有效降低了智能电门锁的功耗,减少电池的更换次数以及提高了智能门锁的寿命。



1. 一种降低智能门锁功耗的方法,其特征在于,所述方法包括:

在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长,所述空闲时长为系统空闲预计维持的时长;

当所述空闲时长满足休眠条件时,获取各功能接口的数据;

若基于所述数据确定用户任务处于空闲状态,则控制所述智能门锁系统进入休眠状态;

所述控制所述智能门锁系统进入休眠状态的步骤包括:

对各外部模块进行低功耗配置;以及

设置唤醒时长,并控制主控芯片进入休眠状态;

所述对各外部模块进行低功耗配置的步骤包括:

对各外部模块的引脚电平和I/O口的电平状态进行低功耗配置;

所述设置唤醒时长,并控制主控芯片进入休眠状态的步骤包括:

获取第二预设休眠时长,将所述第二预设休眠时长与所述空闲时长进行比较;

若所述第二预设休眠时长小于所述空闲时长,则确定所述第二预设休眠时长为所述唤醒时长,并控制主控芯片进入休眠状态;或者

若所述第二预设休眠时长大于所述空闲时长,则确定所述空闲时长为所述唤醒时长,并控制主控芯片进入休眠状态;

所述控制所述智能门锁系统进入休眠状态的步骤之后,还包括:

在当前休眠时长达到所述唤醒时长时或存在外部事件发生时,唤醒所述智能门锁系统,并将所述智能门锁系统的系统节拍加上休眠时长对应的系统节拍数。

2. 根据权利要求1所述的降低智能门锁功耗的方法,其特征在于,所述当所述空闲时长满足休眠条件时,获取各功能接口的数据的步骤包括:

获取第一预设休眠时长;

若所述第一预设休眠时长小于所述空闲时长,则获取所述各功能接口的数据。

3. 根据权利要求1至2中任一项所述的降低智能门锁功耗的方法,其特征在于,在当前休眠时长达到所述唤醒时长时,唤醒所述智能门锁系统的步骤之后,还包括:

获取当前的第一空闲时长,若所述第一空闲时长满足所述休眠条件且所述用户任务处于空闲状态,则控制所述智能门锁系统再次进入所述休眠状态。

4. 根据权利要求1至2中任一项所述的降低智能门锁功耗的方法,其特征在于,所述在当前休眠时长达到所述唤醒时长时或存在外部事件发生时,唤醒所述智能门锁系统的步骤包括:

在当前休眠时长达到所述唤醒时长时或存在外部事件发生时,对所述主控芯片进行唤醒;以及

对所述各外部模块进行唤醒配置。

5. 根据权利要求4所述的降低智能门锁功耗的方法,其特征在于,在存在外部事件发生时,所述对所述主控芯片进行唤醒的步骤包括:

当检测到目标按键处于按压状态时,生成高电平信号;

根据所述高电平信号对所述主控芯片进行唤醒。

6. 根据权利要求4所述的降低智能门锁功耗的方法,其特征在于,在存在外部事件发生

时,所述对所述各外部模块进行唤醒配置的步骤之后,还包括:

获取调度指令,所述调度指令用于指示所要执行的所述用户任务;

根据所述调度指令启动所述智能门锁系统处理所述用户任务。

7.一种智能门锁,其特征在于,所述智能门锁包括存储器、处理器及存储在存储器上并在所述处理器上运行的降低智能门锁功耗的程序,所述处理器执行所述降低智能门锁功耗的程序时实现如权利要求1至6中任一项所述的方法的步骤。

8.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有降低智能门锁功耗的程序,所述降低智能门锁功耗的程序被处理器执行时实现如权利要求1至6中任一项所述的方法的步骤。

降低智能门锁功耗的方法、智能门锁和存储介质

技术领域

[0001] 本申请涉及智能门锁技术领域,尤其涉及一种降低智能门锁功耗的方法、智能门锁和存储介质。

背景技术

[0002] 随着科学技术的不断发展,家居行业也逐步向智能化发展,普通的门锁在安全性能上已不能满足人们的要求,而智能门锁在安全性能和用户体验感上有很大的优势,智能门锁具有很大的发展前景,在家庭应用领域中将会占据重要的地位。智能门锁多是用电池供电,由于智能门锁功能模块繁多,导致整机功耗增加、电池更换周期过短,这不仅增加了成本,而且也对环境造成污染,同时会影响用户的体验。

发明内容

[0003] 本申请实施例通过提供一种降低智能门锁功耗的方法、智能门锁和存储介质,旨在解决由于智能门锁功能模块较多,导致智能门锁的功耗较高,需要频繁更换电池的问题。

[0004] 为实现上述目的,本申请一方面提供一种降低智能门锁功耗的方法,所述降低智能门锁功耗的方法包括以下步骤:

[0005] 在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长;

[0006] 当所述空闲时长满足休眠条件时,获取各功能接口的数据;

[0007] 若基于所述数据确定用户任务处于空闲状态,则控制所述智能门锁系统进入休眠状态。

[0008] 可选地,所述控制所述智能门锁系统进入休眠状态的步骤包括:

[0009] 对各外部模块进行低功耗配置;以及

[0010] 设置唤醒时长,并控制主控芯片进入休眠状态。

[0011] 可选地,所述当所述空闲时长满足休眠条件时,获取各功能接口的数据的步骤包括:

[0012] 获取第一预设休眠时长;

[0013] 若所述第一预设休眠时长小于所述空闲时长,则获取所述各功能接口的数据。

[0014] 可选地,所述设置唤醒时长的步骤包括:

[0015] 获取第二预设休眠时长,将所述第二预设休眠时长与所述空闲时长进行比较;

[0016] 若所述第二预设休眠时长小于所述空闲时长,则确定所述第二预设休眠时长为所述唤醒时长;或者

[0017] 若所述第二预设休眠时长大于所述空闲时长,则确定所述空闲时长为所述唤醒时长。

[0018] 可选地,所述控制所述智能门锁系统进入休眠状态的步骤之后,还包括:

[0019] 在当前休眠时长达到所述唤醒时长时,自动唤醒所述智能门锁系统;

[0020] 获取当前的第一空闲时长,若所述第一空闲时长满足所述休眠条件且所述用户任

务处于空闲状态,则控制所述智能门锁系统再次进入所述休眠状态。

[0021] 可选地,所述控制所述智能门锁系统进入休眠状态的步骤之后,还包括:

[0022] 在接收到唤醒信号时,对所述主控芯片进行唤醒;以及

[0023] 对所述各外部模块进行唤醒配置,并修正所述智能门锁系统节拍。

[0024] 可选地,所述在接收到唤醒信号时,对所述主控芯片进行唤醒的步骤包括:

[0025] 当检测到目标按键处于按压状态时,生成高电平信号;

[0026] 根据所述高电平信号对所述主控芯片进行唤醒。

[0027] 可选地,所述对所述各外部模块进行唤醒配置,并修正所述智能门锁系统节拍的步骤之后,还包括:

[0028] 获取调度指令,所述调度指令用于指示所要执行的所述用户任务;

[0029] 根据所述调度指令启动所述智能门锁系统处理所述用户任务。

[0030] 此外,为实现上述目的,本申请另一方面还提供一种智能门锁,所述智能门锁包括存储器、处理器及存储在存储器上并在所述处理器上运行的降低智能门锁功耗的程序,所述处理器执行所述降低智能门锁功耗的程序时实现如上所述降低智能门锁功耗的方法的步骤。

[0031] 此外,为实现上述目的,本申请另一方面还提供一种计算机可读存储介质,所述计算机可读存储介质上存储有降低智能门锁功耗程序,所述降低智能门锁功耗的程序被处理器执行时实现如上所述降低智能门锁功耗的方法的步骤。

[0032] 本实施例在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长;当空闲时长满足休眠条件时,获取各功能接口的数据;若基于数据确定用户任务处于空闲状态,则控制智能门锁系统进入休眠状态。通过在智能门锁系统和各用户任务都处于空闲状态时,对各外部模块进行低功耗设置,以及设置唤醒时间并使主控MCU进入休眠状态,解决了由于智能门锁功能模块较多,导致智能门锁的功耗较高,需要频繁更换电池的问题,有效降低了智能电门锁的功耗,减少电池的更换次数以及提高了智能门锁的寿命。

附图说明

[0033] 图1为本申请实施例方案涉及的硬件运行环境的智能门锁结构示意图;

[0034] 图2为本申请降低智能门锁功耗的方法第一实施例的流程示意图;

[0035] 图3为本申请降低智能门锁功耗的方法第二实施例的流程示意图;

[0036] 图4为本申请降低智能门锁功耗的方法中控制所述智能门锁系统进入休眠状态的流程示意图;

[0037] 图5为本申请降低智能门锁功耗的方法中设置唤醒时长的流程示意图;

[0038] 图6为本申请降低智能门锁功耗的方法中控制所述智能门锁系统进入休眠状态的步骤之后的流程示意图;

[0039] 图7为本申请降低智能门锁功耗的方法中在接收到唤醒信号时,对所述主控芯片进行唤醒的流程示意图;

[0040] 图8为本申请降低智能门锁功耗的方法中对所述各外部模块进行唤醒配置,并修正所述智能门锁系统节拍的步骤之后的流程示意图;

[0041] 图9为本申请降低智能门锁功耗的方法的软件流程示意图。

[0042] 本申请目的的实现、功能特点及优点将结合实施例,参照附图做进一步说明。

具体实施方式

[0043] 应当理解,此处所描述的具体实施例仅用以解释本申请,并不用于限定本申请。

[0044] 本申请实施例的主要解决方案是:在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长;当所述空闲时长满足休眠条件时,获取各功能接口的数据;若基于所述数据确定用户任务处于空闲状态,则控制所述智能门锁系统进入休眠状态。

[0045] 由于现有的智能门锁多是用电池供电,且功能模块繁多,导致整机功耗增加、电池更换周期过短,这不仅增加了成本,而且也对环境造成污染,同时影响用户的体验。本申请在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长;当空闲时长满足休眠条件时,获取各功能接口的数据;若基于数据确定用户任务处于空闲状态,则控制智能门锁系统进入休眠状态。通过在智能门锁系统和各用户任务都处于空闲状态时,对各外部模块进行低功耗设置,以及设置唤醒时间并使主控MCU进入休眠状态,解决了由于智能门锁功能模块较多,导致智能门锁的功耗较高,需要频繁更换电池的问题,有效降低了智能电门锁的功耗,减少电池的更换次数以及提高了智能门锁的寿命。

[0046] 如图1所示,图1为本申请实施例方案涉及的硬件运行环境的智能门锁结构示意图。

[0047] 如图1所示,该智能门锁可以包括:处理器1001,例如CPU,网络接口1004,用户接口1003,存储器1005,通信总线1002。其中,通信总线1002用于实现这些组件之间的连接通信。用户接口1003可以包括显示屏(Display)、输入单元比如键盘(Keyboard),可选用户接口1003还可以包括标准的有线接口、无线接口。网络接口1004可选的可以包括标准的有线接口、无线接口(如WI-FI接口)。存储器1005可以是高速RAM存储器,也可以是稳定的存储器(non-volatile memory),例如磁盘存储器。存储器1005可选的还可以是独立于前述处理器1001的存储装置。

[0048] 可选地,智能门锁还可以包括摄像头、RF(Radio Frequency,射频)电路,传感器、遥控器、音频电路、WiFi模块、检测器等等。当然,所述智能门锁还可配置陀螺仪、气压计、湿度计、温度传感器等其他传感器,在此不再赘述。

[0049] 本领域技术人员可以理解,图1中示出的智能门锁结构并不构成对智能门锁设备的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。

[0050] 如图1所示,作为一种计算机可读存储介质的存储器1005中可以包括操作系统、网络通信模块、用户接口模块以及降低智能门锁功耗程序。

[0051] 在图1所示的智能门锁中,网络接口1004主要用于连接后台服务器,与后台服务器进行数据通信;用户接口1003主要用于连接客户端(用户端),与客户端进行数据通信;而处理器1001可以用于调用存储器1005中降低智能门锁功耗程序,并执行以下操作:

[0052] 在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长;

[0053] 当所述空闲时长满足休眠条件时,获取各功能接口的数据;

[0054] 若基于所述数据确定用户任务处于空闲状态,则控制所述智能门锁系统进入休眠状态。

[0055] 参考图2,图2为本申请降低智能门锁功耗的方法第一实施例的流程示意图。

[0056] 本申请实施例提供了降低智能门锁功耗的方法的实施例,需要说明的是,虽然在流程图中示出了逻辑顺序,但是在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤。

[0057] 降低智能门锁功耗的方法包括:

[0058] 步骤S10,在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长;

[0059] 本实施例提供的方法的应用环境为应用在智能门锁中,该智能门锁的硬件采用LPC55S6x系列芯片作为主控,软件采用RTOS系统,其中,实时操作系统(Real-time operating system,RTOS),又称即时操作系统,与一般的操作系统相比,其特点是实时性,如果有一个任务需要执行,实时操作系统会在较短时间内执行该任务,不会有较长的延时。

[0060] 智能门锁上电初始化后,执行当前的用户任务,当用户长时间与智能门锁没有交互时,RTOS系统则会进入空闲状态(即待机状态);此时,RTOS系统会执行空闲任务,在执行空闲任务中,首先由RTOS系统接口读取系统空闲预计维持的时间 T_i ,即空闲时长。

[0061] 步骤S20,当所述空闲时长满足休眠条件时,获取各功能接口的数据;

[0062] 为避免RTOS系统在短时间内频繁进行休眠与唤醒,用户预先设置有最短休眠时间,即第一预设休眠时长,该时长一般设置为几秒即可,如5秒;同时,还可以根据用户的实际需求设置,在此不做限定。将第一预设休眠时长与空闲预计维持的时间 T_i 进行比较,若第一预设休眠时长大于空闲预计维持的时间 T_i ,则保持RTOS系统继续处于激活状态;若第一预设休眠时长小于空闲预计维持的时间 T_i ,则满足第一预设休眠时长的条件,进一步判断各用户任务是否均处于空闲状态,其中,所述用户任务包括日常任务、突发任务、监控任务以及分析等,基于用户的需求不同,可产生不同的用户任务。在第一预设休眠时长小于空闲预计维持的时间 T_i 时,智能门锁向任务调度模块发起各功能接口数据的采集请求;任务调度模块在接收到采集请求后,将对应的请求数据存储至数据库中,新建一张该任务对应的任务采集表;同时,根据任务相关数据生成任务采集模块需要的包括任务启动参数的任务启动命令,并发送给任务采集模块,任务采集模块根据启动参数进行对应的各功能接数据的采集工作,以获取各功能接口的数据。其中,所述请求数据包括:任务名称、数据类型、任务并发数、抽取方式以及抽取频率等等。

[0063] 步骤S30,若基于所述数据确定用户任务处于空闲状态,则控制所述智能门锁系统进入休眠状态。

[0064] 智能门锁在获取到任务采集模块反馈的各功能接口的数据时,对该数据进行分析,以确定当前各用户任务是否处于空闲状态。具体地,获取该数据对应的任务执行表,判断任务执行表中是否存在等待执行的用户任务,若不存在,则判断正在执行的用户是否已执行完毕,若均已执行完毕,说明当前所有的用户任务均处于空闲状态。此时,智能门锁在进行相关的低功耗配置后,控制系统进入低功耗的休眠状态。当系统进入休眠状态后,各外部模块、CPU、定时器、串行口等全部停止工作,只有外部中断继续工作。同时,使系统进入休眠状态的指令将成为休眠前系统执行的最后一条指令,进入休眠状态后,MCU中程序未涉及到的数据存储器和特殊功能寄存器中的数据都将保持原值,可由外部中断低电平触发或由下降沿触发中断或者硬件复位模式唤醒系统。需要说明的是,使用中断唤醒单系统时,程序从原来停止处继续运行,当使用硬件复位唤醒系统时,程序将从头开始执行。

[0065] 进一步地,参考图4,所述控制所述智能门锁系统进入休眠状态的步骤包括:

[0066] 步骤S31,对各外部模块进行低功耗配置;以及

[0067] 步骤S32,设置唤醒时长,并控制主控芯片进入休眠状态。

[0068] 低功耗是MCU的一项非常重要的指标,比如某些可穿戴的设备,其携带的电量有限,如果整个电路消耗的电量特别大,就会经常出现电量不足的情况,影响用户体验。因此,当RTOS系统在进入休眠时,需要对各外部模块以及主控MCU进行低功耗配置。例如:配置外部模块进入低功耗模式,关闭与低功耗无关的偏上外设及时钟,配置合适引脚电平,避免漏电。同时,还需要配I/O口的电平状态,一般的I/O口的内部或者外部会有上下拉电阻,若在MCU进入休眠状态时,未设置各I/O口的电平状态,则会消耗掉一部分电量。例如:若某个I/O口有个10K Ω 的上拉电阻,把引脚拉到4V,当MCU进入休眠状态时,该I/O口被设置为输出低电平,根据欧姆定律,此引脚会消耗 $4V/10K=0.4mA$ 的电流,若存在多个这样的I/O口,则会造成电量消耗。因此,在MCU进入休眠状态前,逐一检测各I/O口的状态,若I/O口带上拉电阻,则设置为高电平输出或者高阻态输入;若I/O口带下拉电阻,则设置为低电平输出或者高阻态输入。

[0069] 在成功设置低功耗模式后,需要周期性的运行程序,然后进入低功耗,间隔一定时间后,继续运行程序。此时,需要用到RTC时钟来进行对低功耗模式的唤醒,设置周期唤醒函数,如每间隔50s,RTC闹钟自动唤醒一次低功耗。同时,还需要设置RTC/GINT中断,GINT初始化2个以上引脚为同一组中断,当同一组中无任一极性有效信号时,任意一个引脚产生极性有效信号,则产生中断。在完成各外部模块、I/O口、RTC唤醒以及GINT中断等匹配后,主控MCU进入休眠状态。

[0070] 进一步地,参考图5,所述设置唤醒时长的步骤包括:

[0071] 步骤S320,获取第二预设休眠时长,将所述第二预设休眠时长与所述空闲时长进行比较;

[0072] 步骤S321,若所述第二预设休眠时长小于所述空闲时长,则确定所述第二预设休眠时长为所述唤醒时长;或者

[0073] 步骤S322,若所述第二预设休眠时长大于所述空闲时长,则确定所述空闲时长为所述唤醒时长。

[0074] 智能门锁中预先设置有最大休眠时长,即第二预设休眠时长,该最大休眠时长可设置为几分钟、几小时或者几天,在MCU进入休眠状态前,需要基于最大休眠时长确定唤醒时长。具体地,将空闲预计维持的时间 T_i 与最大休眠时长进行比较,若空闲预计维持的时间 T_i 小于最大休眠时长,则以空闲预计维持的时间 T_i 作为唤醒时长;若空闲预计维持的时间 T_i 大于最大休眠时长,则以最大休眠时长作为唤醒时长。

[0075] 本实施例在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长;当空闲时长满足休眠条件时,获取各功能接口的数据;若基于数据确定用户任务处于空闲状态,则控制智能门锁系统进入休眠状态。通过在智能门锁系统和各用户任务都处于空闲状态时,对各外部模块进行低功耗设置,以及设置唤醒时间并使主控MCU进入休眠状态,解决了由于智能门锁功能模块较多,导致智能门锁的功耗较高,需要频繁更换电池的问题,有效降低了智能电门锁的功耗,减少电池的更换次数以及提高了智能门锁的寿命。

[0076] 进一步地,参考图3,提出本申请降低智能门锁功耗的方法第二实施例。

[0077] 所述降低智能门锁功耗的方法第二实施例与所述降低智能门锁功耗的方法第一

实施例的区别在于,所述控制所述智能门锁系统进入休眠状态的步骤之后,还包括:

[0078] 步骤S33,在当前休眠时长达到所述唤醒时长时,自动唤醒所述智能门锁系统;

[0079] 步骤S34,获取当前的第一空闲时长,若所述第一空闲时长满足所述休眠条件且所述用户任务处于空闲状态,则控制所述智能门锁系统再次进入所述休眠状态。

[0080] 智能门锁存在两种被唤醒的情况,一种是达到休眠时长自动唤醒,另一种是接收到唤醒信号进行唤醒。当智能门锁的休眠时长到达唤醒时且未接收到唤醒信号时,智能门锁会被内部的RTC唤醒;醒来后系统恢复调度,若当前无事件需要处理(用户任务空闲),则系统会调用空闲任务,在空闲任务中可再次进行休眠。也就是说,在检测到RTOS系统空闲且所有用户任务均空闲的情况下,则确定满足休眠条件,MCU再次进入休眠状态。例如:唤醒时间为1小时,智能门锁在1小时内未接收到唤醒信号,则进行自动唤醒,在唤醒后检测各功能接口的数据,确定当前各用户任务均处于空闲状态时,判断当前满足休眠条件,控制RTOS系统再次进入休眠状态,并等待下一个周期性唤醒。

[0081] 进一步地,参考图6,所述控制所述智能门锁系统进入休眠状态的步骤之后,还包括:

[0082] 步骤S35,在接收到唤醒信号时,对所述主控芯片进行唤醒;以及

[0083] 智能门锁在接收到唤醒信号时,即在外部事件发生或RTC唤醒后,主控MCU将会从掉电模式唤醒,其中,智能门锁的休眠模式一般包括:睡眠模式、深度睡眠模式、掉电模式以及深度掉电模式,而对于每种模式的选择,包括两个指标:休眠电流低和唤醒速度快。

[0084] 进一步地,参考图7,所述在接收到唤醒信号时,对所述主控芯片进行唤醒的步骤包括:

[0085] 步骤S350,当检测到目标按键处于按压状态时,生成高电平信号;

[0086] 步骤S351,根据所述高电平信号对所述主控芯片进行唤醒。

[0087] 在智能门锁中,按键模块与主控MCU之间通过SPI口通信连接,用户预先设置有多个目标按键,该目标按键用于唤醒系统。同时,目标按键与主控MCU连接,当按压目标按键后,会产生高电平信号唤醒主控MCU,其中,高电平信号可由人为操作触发或自主智能触发。可选地,目标按键还可以为虚拟按键,例如:用户按压手机上的目标虚拟按键时,产生一高电平信号,手机将该高电平信号发送至智能门锁,以实现主控MCU的唤醒。

[0088] 可选地,还可以通过用户发送的语音信息进行主控MCU的唤醒操作,例如:用户预先设置唤醒词“小T小T”,将该唤醒词存储至智能门锁中。当用户向智能门锁发送“小T小T”时,智能门锁自动对该语音信息进行识别,获取语音信息中的文本内容,若该文本内容与预先存储的唤醒词一致,则唤醒主控MCU。

[0089] 步骤S36,对所述各外部模块进行唤醒配置,并修正所述智能门锁系统节拍。

[0090] 在唤醒主控MCU后,进一步对各外部模块进行唤醒配置,并修正智能门锁的系统节拍,具体地,在唤醒主控MCU后,将各外部模块的工作模式设置为运行模式,同时,设置各外部模块的电源、相关引脚以及时钟等,在设置时钟时,需要设置时钟参数,如打开外部高速时钟晶振、设置系统时钟等等。其次,智能门锁在休眠时,系统时钟(系统节拍)会停掉,所以休眠的这段时间里系统节拍是停止的,在被唤醒后,系统节拍要加上休眠时间对应的系统节拍数。其中,所有的操作系统都需要提供一个系统时钟节拍,以供系统处理如延时,超时等与时间相关的事件(时基)。系统时钟节拍是特定的周期性中断,中断间的时钟间隔取决

于不同的应用,设定时基,一般是1ms左右,同时,时钟的节拍中断使得可以将任务延迟若干个时钟节拍(释放CPU给其他任务,若干时间后再执行);在任务等待事件发生的时候,也提供等待超时等待,当时钟节拍频率越快,系统额外开销越大。因此,智能门锁在被唤醒后,需要及时修正系统节拍,以避免系统节拍频率过快,导致系统开销大。

[0091] 进一步地,参考图8,所述对所述各外部模块进行唤醒配置,并修正所述智能门锁系统节拍的步骤之后,还包括:

[0092] 步骤S360,获取调度指令,所述调度指令用于指示所要执行的所述用户任务;

[0093] 步骤S361,根据所述调度指令启动所述智能门锁系统处理所述用户任务。

[0094] 智能门锁在接收到唤醒信号后,自动产生调度指令或者从客户端接收由调度员下达的调度指令,该调度指令用于指示所要执行的所述用户任务,根据调度指令启动智能门锁系统处理用户任务。具体地,智能门锁预先将用户任务分为三个低级,如:第一等级(紧急),第二等级(重要)以及第三等级(普通),进一步根据系统的负载情况,对不同优先级的用户任务采用不同的调度方法。例如:对于第一等级的用户任务,首先从系统中查找空闲的工作线程,将该用户任务调度到空闲的工作线程上执行;若未找到空闲的工作线程,则创建新的空闲的工作线程,并将该用户任务调度至新创建的工作线程上。对于第二等级的用户任务,首先从系统中获取等待队列中各用户任务的预期等待时间,将预期等待时间与当前用户任务的最迟执行时间作比较,获取预期等待时间小于当前用户任务的最迟执行时间的任务队列,将当前用户任务调度至该任务队列的末端;若未找到匹配的任务队列,则查找系统中空闲的工作线程,将当前用户任务调度至空闲的时间线程上。对于第一等级的用户任务,获取该用户任务对应执行的内容,在系统中查找与该内容匹配的非空闲的工作线程,将用户任务调度至该非空闲的工作线程中。通过将用户任务分为三个不同的等级,基于等级的不同,采用不同的调度方案,使得可以及时执行用户任务,满足用户对实时性的需求。

[0095] 本实施例在外部事件发生或RTC唤醒后,主控MCU将会从掉电模式唤醒,对各外部模块进行唤醒配置以及修正系统节拍,同时,还通过获取调度指令,通过该调度指令及时启动系统处理所述用户任务,使得系统被唤醒后能够及时处理各项事务,保证了系统的正常运行。

[0096] 为了更好的说明本申请实施例的方案,降低智能门锁功耗的方法在软件方面的操作流程如下所示:

[0097] 参考图9,在智能门锁上电后,RTOS系统进入初始化,在初始化完成后,处理当前的用户任务。当用户长时间(如2小时)与智能门锁没有交互时,检测RTOS系统是否进入空闲状态(即待机状态),若是,则执行空闲任务,若否,则返回执行检测RTOS系统是否进入空闲状态的步骤。在执行空闲任务的过程中,获取系统空闲预计维持的时间 T_i ,即空闲时间,判断该空闲时间是否满足休眠要求,若空闲预计维持的时间 T_i 大于最短休眠时间,则判断空闲时间满足休眠要求;若空闲预计维持的时间 T_i 小于最短休眠时间,则返回执行检测RTOS系统是否进入空闲状态的步骤。在空闲时间满足休眠要求时,获取系统各功能接口的数据,通过该数据判断系统中的用户任务是否均空闲,若均空闲,则对外部模块进行低功耗的休眠配置;若存在任意一个非空闲的用户任务,则返回执行检测RTOS系统是否进入空闲状态的步骤。在进行外部模块的低功耗的休眠配置时,需要配置外部模块进入低功耗模式,关闭与低功耗无关的偏上外设及时钟,配置合适引脚电平,避免漏电等等,进一步还需要配置RTC

唤醒和GINT唤醒,通过将空闲预计维持的时间 T_i 与最大休眠时长进行比较,若空闲预计维持的时间 T_i 小于最大休眠时长,则以空闲预计维持的时间 T_i 作为唤醒时长;若空闲预计维持的时间 T_i 大于最大休眠时长,则以最大休眠时长作为唤醒时长。在完成上述配置后,控制主控MCU进入掉电模式,同时设置RTC/GINT定时中断。在接收到唤醒信号时,对主控MCU进行唤醒,对各外部模块进行唤醒配置,并修正智能门锁系统节拍,同时获取调度指令,根据调度指令启动智能门锁系统处理所述用户任务。系统被唤醒进行正常工作后,继续实时检测RTOS系统是否进入空闲状态(即待机状态)。

[0098] 本实施例在智能门锁系统和各用户任务都处于空闲状态时,对各外部模块进行低功耗设置,以及设置唤醒时间并使主控MCU进入休眠状态,解决了由于智能门锁功能模块较多,导致智能门锁的功耗较高,需要频繁更换电池的问题,有效降低了智能电门锁的功耗,减少电池的更换次数以及提高了智能门锁的寿命。

[0099] 此外,本申请还提供一种降低智能门锁功耗系统,在一实施例中,所述系统包括存储器、处理器及存储在存储器上并可在处理器上运行的降低智能门锁功耗程序,降低智能门锁功耗程序被处理器执行时实现以下步骤:

[0100] 在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长;

[0101] 当所述空闲时长满足休眠条件时,获取各功能接口的数据;

[0102] 若基于所述数据确定用户任务处于空闲状态,则控制所述智能门锁系统进入休眠状态。

[0103] 在一实施例中,所述降低智能门锁功耗系统包括第一获取模块、第二获取模块以及确定模块;

[0104] 所述第一获取模块,用于在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长;

[0105] 所述第二获取模块,用于当所述空闲时长满足休眠条件时,获取各功能接口的数据;

[0106] 所述确定模块,用于若基于所述数据确定用户任务处于空闲状态,则控制所述智能门锁系统进入休眠状态。

[0107] 进一步地,所述确定模块包括配置单元;

[0108] 所述配置单元,用于对各外部模块进行低功耗配置;以及

[0109] 所述配置单元,还用于设置唤醒时长,并控制主控芯片进入休眠状态。

[0110] 进一步地,所述第二获取模块包括获取单元;

[0111] 所述获取单元,用于获取第一预设休眠时长;

[0112] 所述获取单元,还用于若所述第一预设休眠时长小于所述空闲时长,则获取所述各功能接口的数据。

[0113] 进一步地,所述配置单元包括获取子单元和确定子单元;

[0114] 所述获取子单元,用于获取第二预设休眠时长,将所述第二预设休眠时长与所述空闲时长进行比较;

[0115] 所述确定子单元,用于若所述第二预设休眠时长小于所述空闲时长,则确定所述第二预设休眠时长为所述唤醒时长;或者

[0116] 所述确定子单元,还用于若所述第二预设休眠时长大于所述空闲时长,则确定所

述空闲时长为所述唤醒时长。

[0117] 进一步地,所述确定模块还包括唤醒单元;

[0118] 所述唤醒单元,用于在当前休眠时长达到所述唤醒时长时,自动唤醒所述智能门锁系统;

[0119] 所述唤醒单元,还用于获取当前的第一空闲时长,若所述第一空闲时长满足所述休眠条件且所述用户任务处于空闲状态,则控制所述智能门锁系统再次进入所述休眠状态。

[0120] 进一步地,所述唤醒单元,还用于在接收到唤醒信号时,对所述主控芯片进行唤醒;以及

[0121] 所述唤醒单元,还用于对所述各外部模块进行唤醒配置,并修正所述智能门锁系统节拍。

[0122] 进一步地,所述唤醒单元包括检测子单元和唤醒子单元;

[0123] 所述检测子单元,用于当检测到目标按键处于按压状态时,生成高电平信号;

[0124] 所述唤醒子单元,用于根据所述高电平信号对所述主控芯片进行唤醒。

[0125] 进一步地,所述唤醒单元还包括获取子单元和调度子单元;

[0126] 所述获取子单元,用于获取调度指令,所述调度指令用于指示所要执行的所述用户任务;

[0127] 所述调度子单元,用于根据所述调度指令启动所述智能门锁系统处理所述用户任务。

[0128] 上述的降低智能门锁功耗系统各个模块功能的实现与上述方法实施例中的过程相似,在此不再一一赘述。

[0129] 此外,本申请还提供一种智能门锁,所述智能门锁包括存储器、处理器及存储在存储器上并在所述处理器上运行的降低智能门锁功耗程序,所述智能门锁在检测到智能门锁系统进入空闲状态时,获取当前的空闲时长;当空闲时长满足休眠条件时,获取各功能接口的数据;若基于数据确定用户任务处于空闲状态,则控制智能门锁系统进入休眠状态。通过在智能门锁系统和各用户任务都处于空闲状态时,对各外部模块进行低功耗设置,以及设置唤醒时间并使主控MCU进入休眠状态,解决了由于智能门锁功能模块较多,导致智能门锁的功耗较高,需要频繁更换电池的问题,有效降低了智能电门锁的功耗,减少电池的更换次数以及提高了智能门锁的寿命。

[0130] 此外,本申请还提供一种计算机可读存储介质,所述计算机可读存储介质上存储有降低智能门锁功耗程序,所述降低智能门锁功耗程序被处理器执行时实现如上所述降低智能门锁功耗的方法的步骤。

[0131] 本领域内的技术人员应明白,本申请的实施例可提供为方法、系统、或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0132] 本申请是参照根据本申请实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流

程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0133] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0134] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0135] 应当注意的是,在权利要求中,不应将位于括号之间的任何参考符号构造成对权利要求的限制。单词“包含”不排除存在未列在权利要求中的部件或步骤。位于部件之前的单词“一”或“一个”不排除存在多个这样的部件。本申请可以借助于包括有若干不同部件的硬件以及借助于适当编程的计算机来实现。在列举了若干装置的单元权利要求中,这些装置中的若干个可以是通过同一个硬件项来具体体现。单词第一、第二、以及第三等的使用不表示任何顺序。可将这些单词解释为名称。

[0136] 尽管已描述了本申请的可选实施例,但本领域内的技术人员一旦得知了基本创造性概念,则可对这些实施例作出另外的变更和修改。所以,所附权利要求意欲解释为包括可选实施例以及落入本申请范围的所有变更和修改。

[0137] 显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样,倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内,则本申请也意图包含这些改动和变型在内。

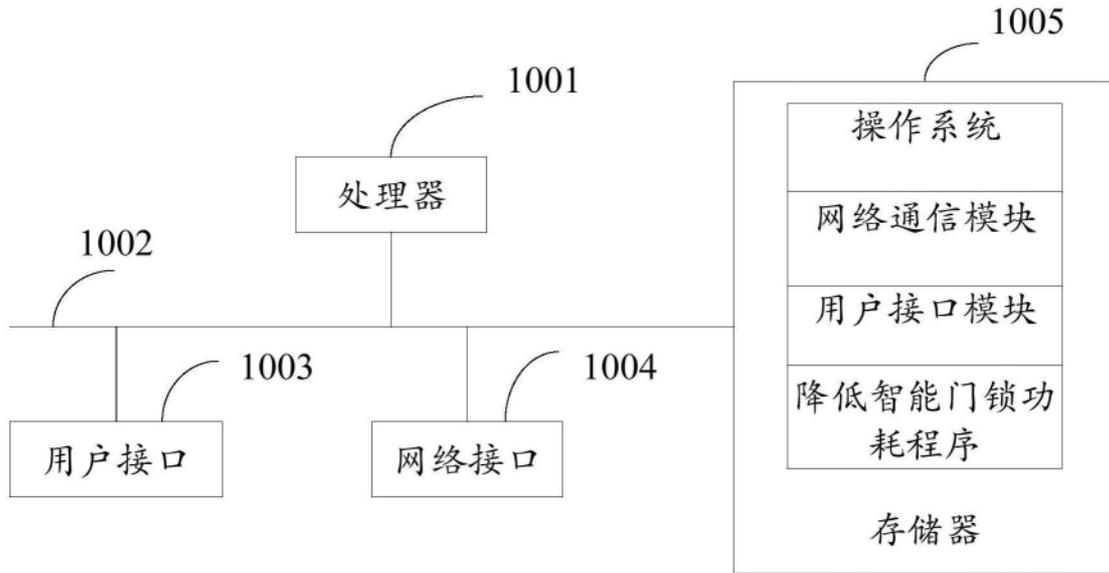


图1

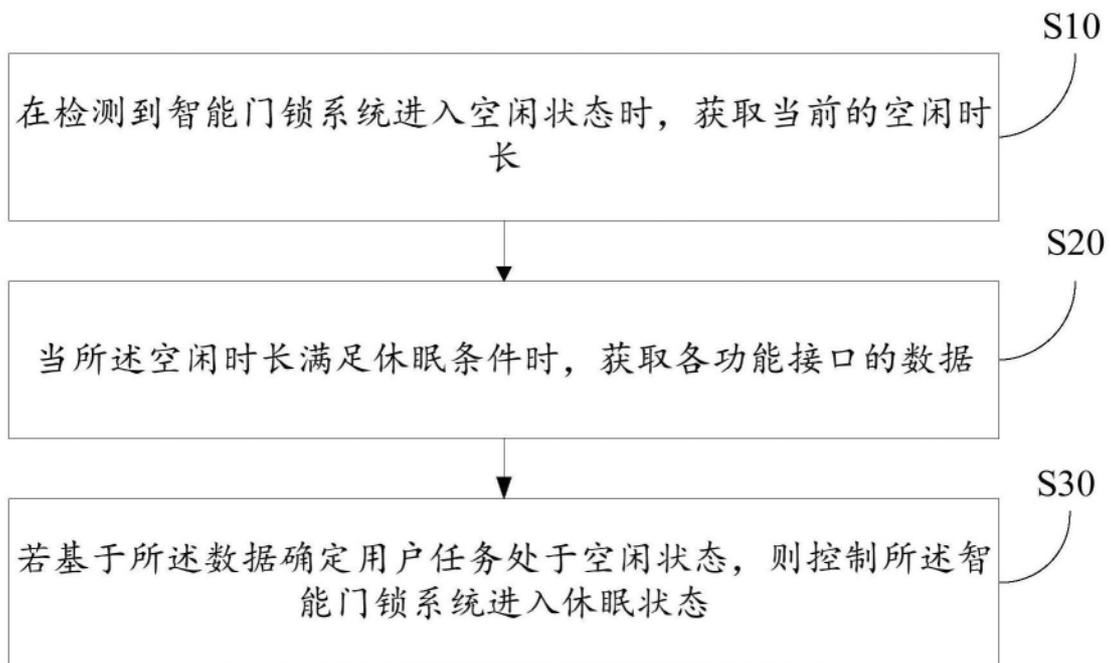


图2

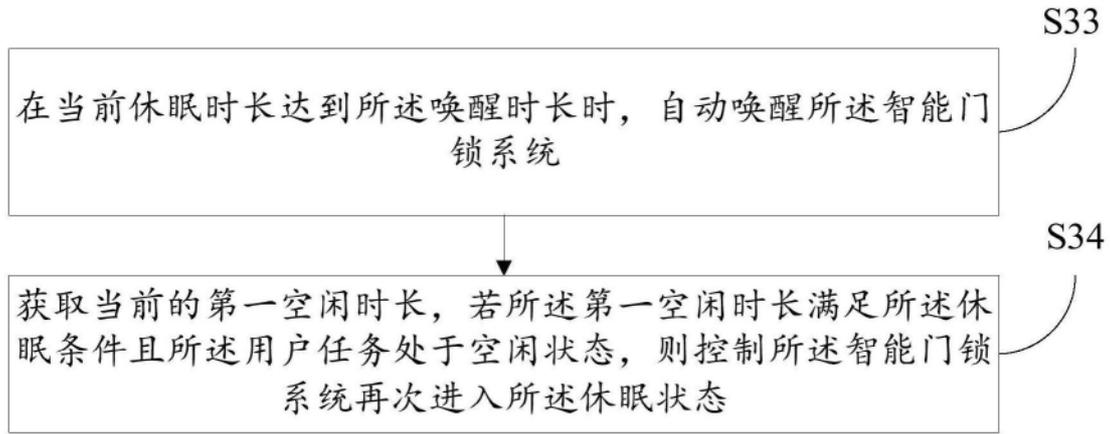


图3

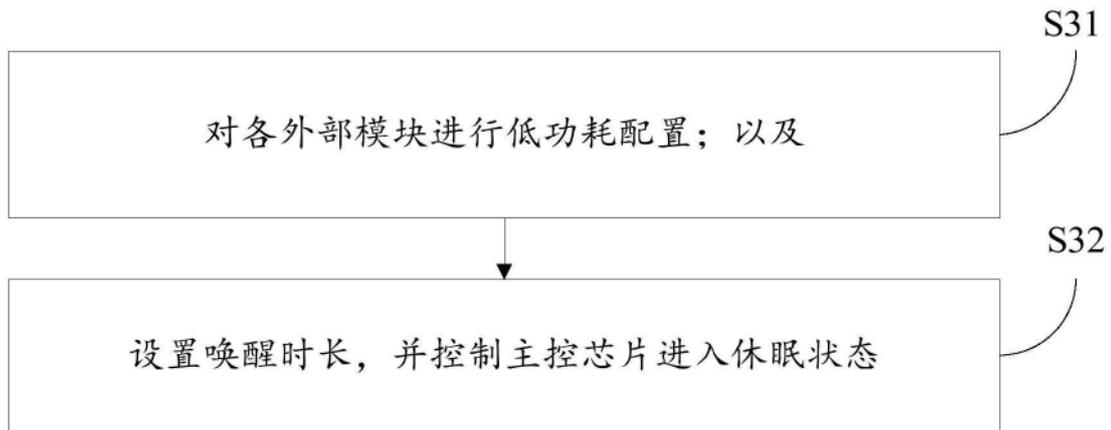


图4

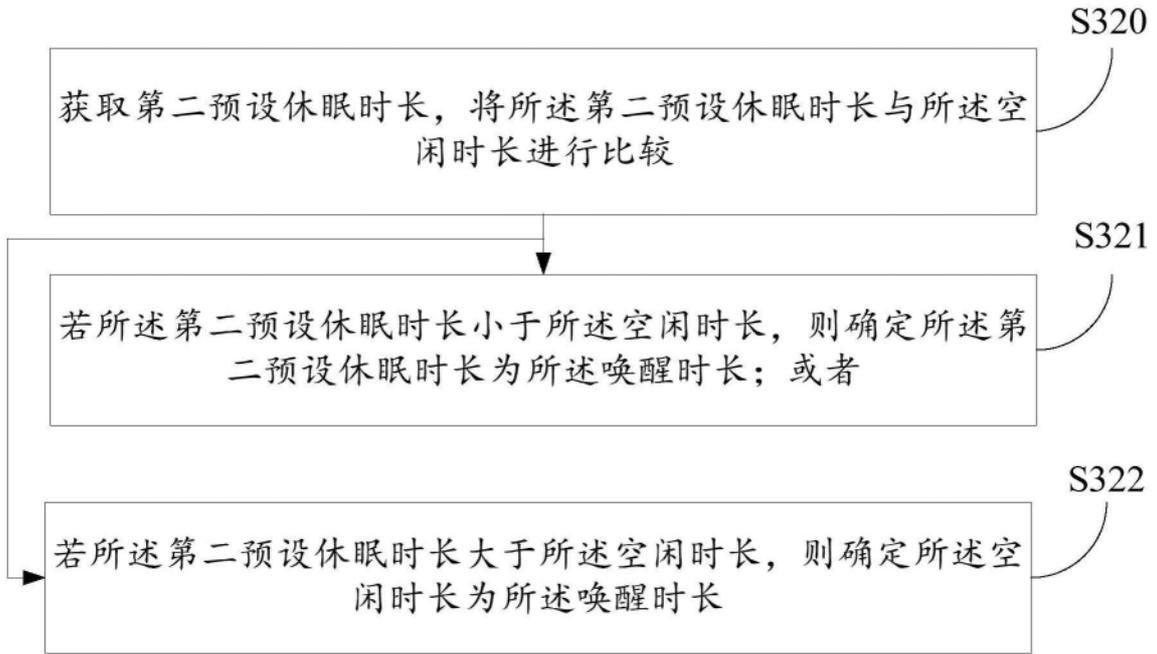


图5

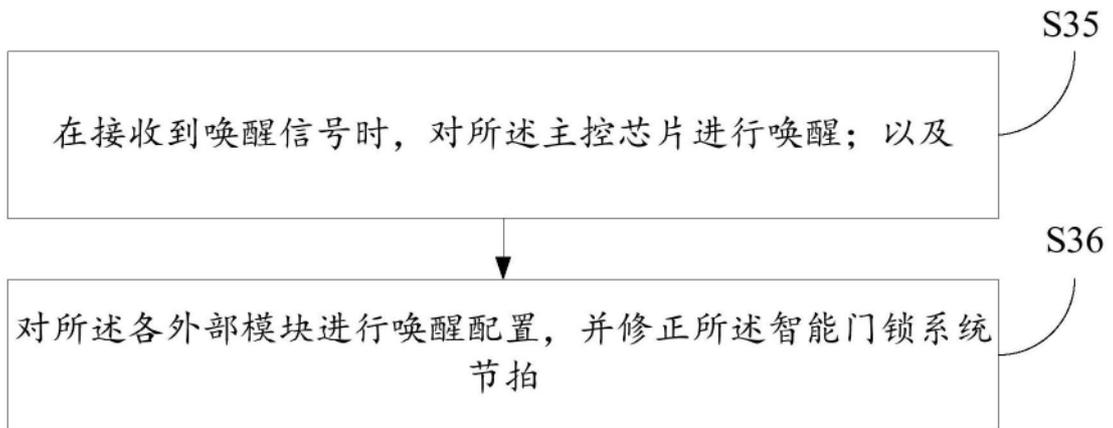


图6

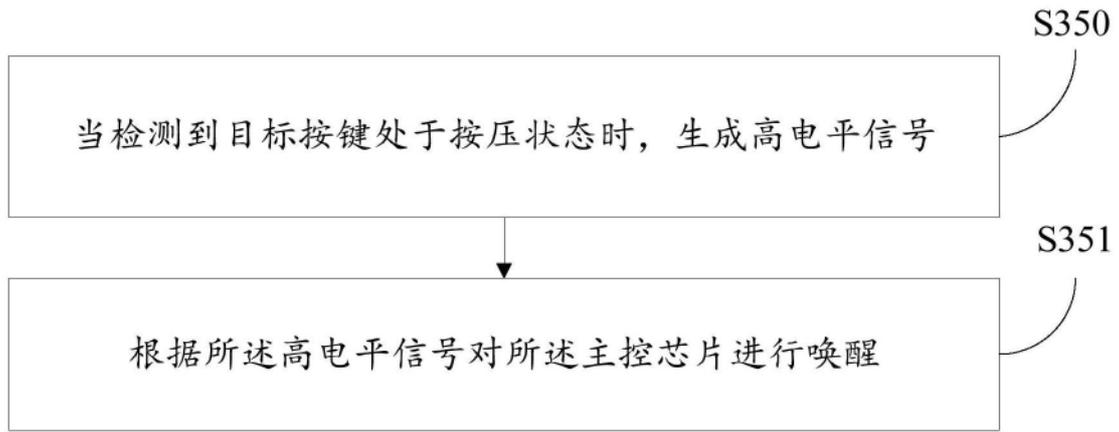


图7

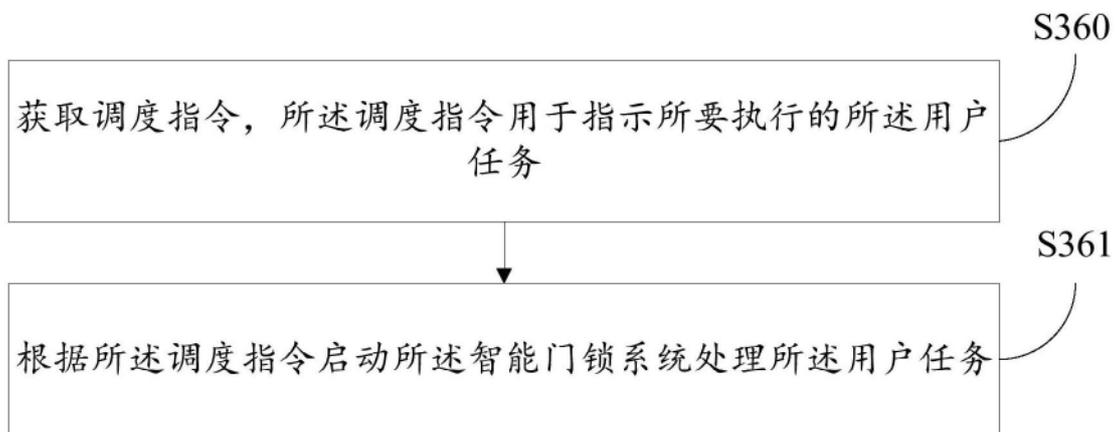


图8

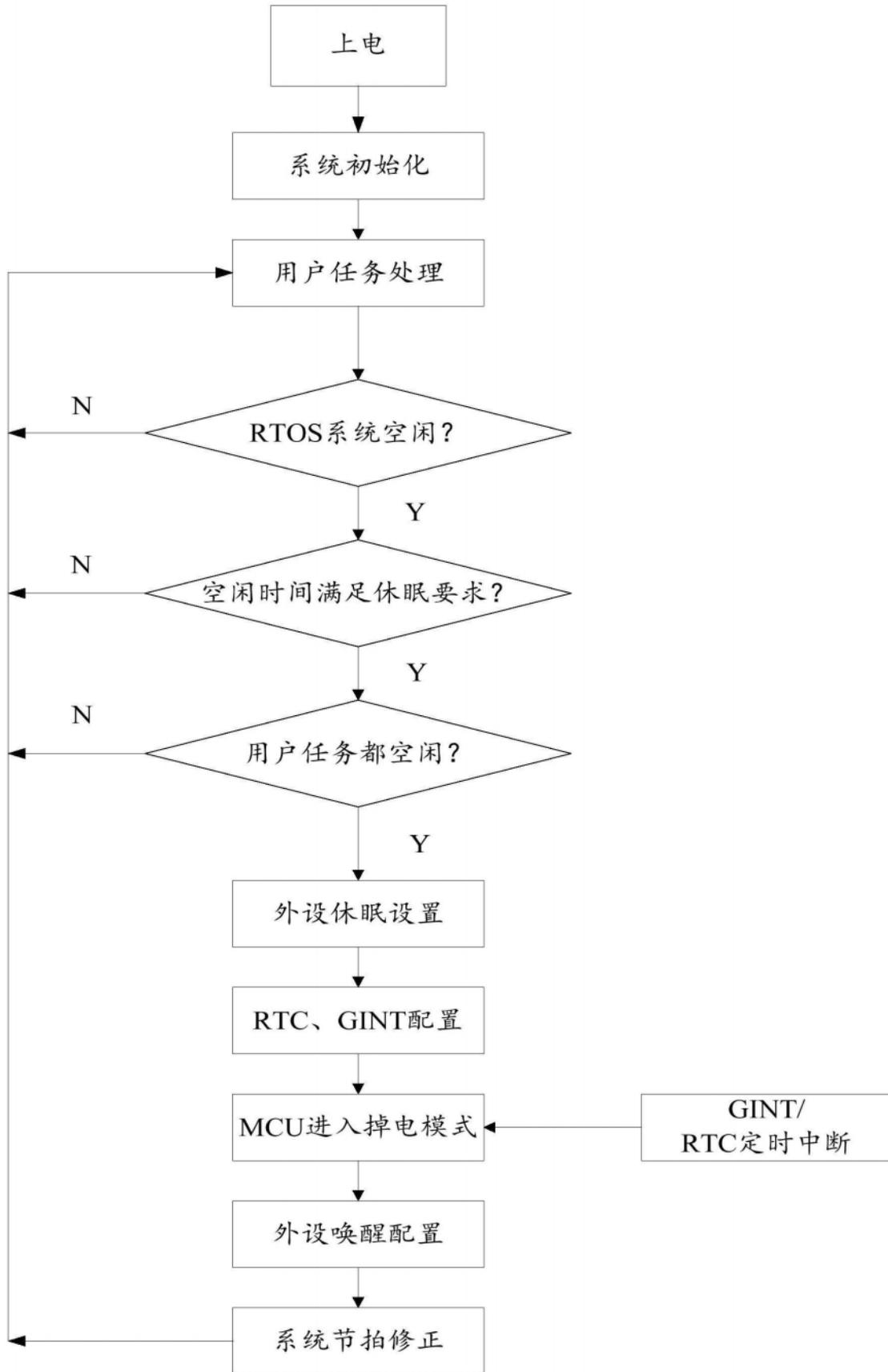


图9