



(12) 发明专利

(10) 授权公告号 CN 111225434 B

(45) 授权公告日 2021.04.09

(21) 申请号 201911253733.5

(22) 申请日 2019.12.09

(65) 同一申请的已公布的文献号  
申请公布号 CN 111225434 A

(43) 申请公布日 2020.06.02

(73) 专利权人 厦门骏俊物联科技股份有限公司  
地址 361009 福建省厦门市思明区塔埔东  
路167号观音山国际商务营运中心8楼  
803D单元

(72) 发明人 卢潇

(74) 专利代理机构 厦门仕诚联合知识产权代理  
事务所(普通合伙) 35227  
代理人 乐珠秀

(51) Int. Cl.

H04W 52/02 (2009.01)

(56) 对比文件

US 2017199696 A1, 2017.07.13

CN 107125888 A, 2017.09.05

CN 109147300 A, 2019.01.04

CN 109511133 A, 2019.03.22

CN 109640383 A, 2019.04.16

CN 107592665 A, 2018.01.16

审查员 易水英

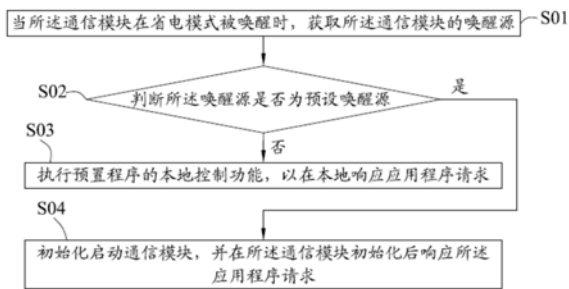
权利要求书2页 说明书8页 附图3页

(54) 发明名称

一种应用程序响应方法、系统、存储介质及通信模块

(57) 摘要

本发明提供一种应用程序响应方法、系统、存储介质及通信模块,该方法包括:当通信模块从处于省电模式唤醒时,获取所述通信模块的唤醒源;判断所述唤醒源是否为预设唤醒源;当判断结果为否,则执行预置程序的本地控制功能,以在本地响应应用程序请求。本发明通过在获取到预置唤醒源以外的唤醒源时,采用预置程序的本地控制功能,以在本地快速响应应用程序请求,因预置程序运行时只使用通信模块上的处理器,不使用通信功能,因此大大缩短了通信模块唤醒后的系统初始化时间,把App响应时延从通常的2-3秒缩短到5毫秒以内,基本可以满足所有低功耗IoT系统实时控制的需求,使低功耗IoT产品在硬件设计上可以完全去掉系统控制MCU,降低了低功耗IoT产品的硬件成本。



1. 一种应用程序响应方法,其特征在于,应用于通信模块当中,所述方法包括如下步骤:

当所述通信模块在省电模式下被唤醒时,由Bootloader获取所述通信模块的唤醒源;  
Bootloader判断所述唤醒源是否为预设唤醒源,所述预设唤醒源是TAU Alarm唤醒;  
当Bootloader判断结果不为TAU Alarm唤醒时,则Bootloader调用并执行BootTest;  
BootTest运行时只使用通信模块上的处理器,执行本地控制功能,以在本地响应应用请求,而不需要使用通信功能。

2. 如权利要求1所述的应用程序响应方法,其特征在于,在所述执行BootTest功能之前,还包括:

BootTest关闭所有不需要的时钟和电源。

3. 如权利要求1所述的应用程序响应方法,其特征在于,所述方法还包括:

当BootTest对于唤醒源判断结果为TAU Alarm时,则初始化启动所述通信模块,并在所述通信模块初始化后响应所述应用程序数据通信请求。

4. 如权利要求1所述的应用程序响应方法,其特征在于,在执行BootTest程序之后,还包括:

判断所述BootTest是否完成所有工作;

当判断结果为否,则BootTest控制CPU进入WFI状态。

5. 如权利要求4所述的应用程序响应方法,其特征在于,在所述判断BootTest是否完成所有工作后,还包括:

如果BootTest完成所有控制任务后,则BootTest控制系统再次进入PSM。

6. 如权利要求5所述的应用程序响应方法,其特征在于,判断BootTest是否完成所有工作的步骤包括:

判断所述BootTest的执行返回值是否为0;

当BootTest返回值为0,则判定BootTest完成所有工作;

当所述BootTest返回值不为0时,则判定BootTest未完成所有工作,BootTest设置CPU进入WFI状态,等待被中断再次唤醒后直接执行BootTest。

7. 如权利要求1所述的应用程序响应方法,其特征在于,在所述BootTest的本地控制功能,以在本地响应所述BootTest的步骤之后,还包括:

将所述BootTest的执行数据保存于通信模块的文件系统中。

8. 一种应用程序响应系统,其特征在于,应用于通信模块当中,所述系统包括:

唤醒源获取模块,用于当所述通信模块在省电模式下被唤醒时,获取所述通信模块的唤醒源;

唤醒源判断模块,用于判断所获取的唤醒源是否为预设的TAU Alarm唤醒源;

应用程序响应模块,用于当判断到所述唤醒源不为预设唤醒源时,执行BootTest;

BootTest运行时只使用通信模块上的处理器,执行本地控制功能,以在本地响应应用请求,而不需要使用通信功能。

9. 一种计算机可读存储介质,其上存储有计算机程序BootTest、BootTest和通信协议栈以及文件系统,其特征在于,BootTest和BootTest被执行时实现如权利要求1-7任一所述的应用程序响应方法。

10. 一种通信模块,包括存储器、处理器以及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行Bootloader、BootTest时实现如权利要求1-7任一所述的应用程序响应方法。

## 一种应用程序响应方法、系统、存储介质及通信模块

### 技术领域

[0001] 本发明属于物联网产品技术领域,尤其涉及一种应用程序响应方法、系统、存储介质及通信模块。

### 背景技术

[0002] 物联网(Internet of things,简称IoT)产品为了尽可能降低产品成本以及挖掘通信模块的计算潜力,通过OpenCPU技术把MCU的工作转移到通信模块上并以APP的形式完成MCU的所有任务,使用户的应用程序(Application,简称APP)运行于通信模块当中,这样就可以省掉物联网产品上的MCU,以在不缩减功能也不降低性能的情况下降低物联网设备的硬件成本。

[0003] 现有OpenCPU产品都是在RTOS(Real Time Operating System,实时操作系统)正常运行起来、并且通信模块完成系统初始化能够正常工作后,再加载基于OpenCPU API开发的App,这时App软件才能开始运行,导致App开始运行的时间通常要比通信功能晚2-3秒以上(取决于系统初始化时间)。这种方式在普通应用中不会有问题,但是对于功耗要求非常严格并且控制的实时性要求高的场景有明显缺陷,例如NB-IoT产品。

[0004] NB-IoT产品的通信模块支持PSM状态(PowerSaving Mode,省电模式),为了保证此状态下取得极低功耗,在PSM状态下会关闭除RTC(Real-time clock,实时时钟)以外包括存储器在内的所有片上设备,保留RTC相关模块的供电及时钟以便在需要工作时能够退出PSM状态进入正常工作状态。由于进入PSM状态前关闭了大容量存储器而丢失了部分系统数据,所以退出PSM模式后需要对系统重新初始化,此初始化过程差不多需要2-3秒才能完成,导致App的响应因系统初始化工作而延误了2-3秒钟,这个延误所增加的功耗在低功耗物联网产品中无法接受,因此在低功耗IoT产品上,依然无法直接去掉MCU。

### 发明内容

[0005] 本发明的目的在于提供一种应用程序响应方法、系统、存储介质及通信模块,旨在解决现有低功耗IoT OpenCPU产品无法去掉MCU的技术问题。

[0006] 本发明实施例是这样实现的,一种应用程序响应方法,应用于通信模块当中,所述方法包括如下步骤:

[0007] 当所述通信模块在省电模式下被唤醒时,获取所述通信模块的唤醒源;

[0008] 判断所述唤醒源是否为预设唤醒源;

[0009] 当判断结果为否,则执行预置程序的本地控制功能,以在本地响应应用程序请求。

[0010] 更进一步的,在所述执行预置程序的本地控制功能,以在本地响应所述应用程序的步骤之前,还包括:

[0011] 关闭预设时钟和电源之外的其它时钟和电源,所述预设时钟和电源为所述预置程序运行所需要的时钟和电源。

[0012] 更进一步的,所述方法还包括:

[0013] 当判断结果为是,则初始化启动所述通信模块,并在所述通信模块初始化后响应所述应用程序请求。

[0014] 更进一步的,在所述执行预置程序的本地控制功能,以在本地响应所述应用程序的步骤之后,还包括:

[0015] 判断所述预置程序是否完成所有预设工作;

[0016] 当判断结果为是,则控制所述通信模块进入休眠模式。

[0017] 更进一步的,在所述判断所述预置程序是否完成所有预设工作的步骤之后,还包括:

[0018] 当判断结果为否,则控制所述通信模块进入WFI状态,并等待所述通信模块被中断唤醒后再次执行所述预置程序。

[0019] 更进一步的,判断所述预置程序是否完成所有预设工作的步骤包括:

[0020] 判断所述预置程序的执行返回值是否为预设值;

[0021] 当所述执行返回值为所述预设值时,则判定所述预置程序完成所有预设工作;

[0022] 当所述执行返回值不为所述预设值时,则判定所述预置程序未完成所有预设工作。

[0023] 更进一步的,在所述执行预置程序的本地控制功能,以在本地响应所述应用程序的步骤之后,还包括:

[0024] 将所述预置程序的执行数据保存于通信模块的文件系统中。

[0025] 此外,本发明实施例还提出一种应用程序响应系统,应用于通信模块当中,所述系统包括:

[0026] 唤醒源获取模块,用于当所述通信模块从省电模式被唤醒时,获取所述通信模块的唤醒源;

[0027] 唤醒源判断模块,用于判断所获取的唤醒源是否为预设唤醒源;

[0028] 应用程序响应模块,用于当判断到所述唤醒源不为预设唤醒源时,执行预置程序的本地控制功能,以在本地响应应用程序请求。

[0029] 此外,本发明实施例还提出一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现如上述的应用程序响应方法。

[0030] 此外,本发明实施例还提出一种通信模块,包括存储器、处理器以及存储在存储器上并可在处理器上运行的计算机程序,所述处理器执行所述程序时实现如上述的应用程序响应方法。

[0031] 与现有技术相比:通过在获取到预置唤醒源以外的唤醒源时,采用预置程序的本地控制功能,以在本地响应应用程序请求,因预置程序运行时只使用通信模块上的处理器而不需要使用通信功能,因此大大缩短了通信模块唤醒后的系统初始化时间,把App响应时延从通常的2-3秒缩短到5毫秒以内,基本可以满足所有低功耗IoT系统控制的需求,使低功耗IoT产品在硬件设计上也可以完全去掉MCU,降低了低功耗IoT产品的硬件成本。

## 附图说明

[0032] 图1是本发明实施例一提供的应用程序响应方法的流程示意图;

[0033] 图2是本发明实施例二提供的应用程序响应方法的流程示意图;

[0034] 图3是本发明实施例三提供的应用程序响应系统的模块示意图；

[0035] 图4是本发明实施例四提供的通信模块的模块示意图。

### 具体实施方式

[0036] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0037] 本发明针对现有低功耗IoT产品因其通信模块的系统初始化工作会使App的响应延误较长时间，导致在低功耗IoT产品上，还无法将App运行于通信模块当中，导致无法直接去掉MCU的问题，提出一种应用程序响应方法、系统、存储介质及通信模块，以通过在获取到预置唤醒源以外的唤醒源时，采用预置程序的本地控制功能，以在本地运行App，大大缩短了通信模块唤醒后的系统初始化时间，从而使App可以运行于通信模块当中，这样在硬件设计上就可以去掉MCU。

#### [0038] 实施例一

[0039] 请参阅图1，是本发明实施例一提供的应用程序响应方法的流程示意图，应用于通信模块当中，所述通信模块可通过软件和/或硬件来实现所述方法，所述方法具体包括步骤S01-步骤S03。

[0040] 步骤S01，当所述通信模块在省电模式下被唤醒时，获取所述通信模块的唤醒源。

[0041] 需要说明的是，省电模式即为PSM低功耗状态，通信模块在PSM状态下会关闭除RTC以外包括大容量存储器在内的所有片上设备，仅保留RTC相关模块的供电及时钟，以便在通信模块需要唤醒工作时，控制通信模块退出PSM状态，以进入正常工作状态。

[0042] 其中，所述唤醒源可以为唤醒通信模块的触发事件，所述唤醒源用于唤醒通信模块，以使通信模块退出PSM状态进入正常工作状态。所述唤醒源具体可以为外部中断唤醒、RTC唤醒、TAU (Tracking Area Update, 跟踪地区最新情况) Alarm唤醒等。

[0043] 步骤S02，判断所述唤醒源是否为预设唤醒源。

[0044] 其中，所述唤醒源为当前获取的唤醒源，当判断到所述唤醒源不为预设唤醒源时，执行步骤S03；当判断到所述唤醒源为预设唤醒源时，则执行步骤S04。具体地，所述预设唤醒源可以为TAU Alarm唤醒，一般来说，非TAU Alarm唤醒表明系统有实时控制任务请求（如按键、系统掉电、定时数据采样等），此时可调用执行通讯模块当中的预置程序来进行系统所需的检测、处理和控制在控制等工作，以响应应用程序的运行。而TAU Alarm唤醒表明系统有更新跟踪区域的请求，此时必须借助通讯模块的通讯功能，以联网更新地区最新情况。

[0045] 步骤S03，执行预置程序的本地控制功能，以在本地响应应用程序请求。

[0046] 其中，所述预置程序可以为BootApp，具体可以在通信模块上增加Bootloader（若已经有Bootloader的通信模块则不需要再增加Bootloader），通过Bootloader调用执行BootApp，以执行BootApp的本地控制功能来进行系统所需的检测、处理和控制在控制等工作，以在本地快速响应应用程序的请求。

[0047] 在具体实施时，BootApp代码可存储于独立的存储空间，使用独立于Bootloader的SRAM空间，BootApp与Bootloader共享堆和栈空间，BootApp可由专门的独立于App的SDK编译得到，并且BootApp可由专门的下载工具下载到通信模块当中，也可以使用与App相同的

下载工具下载到独立的BootApp分区当中。

[0048] 步骤S04,初始化启动通信模块,并在所述通信模块初始化后响应所述应用程序请求。

[0049] 需要指出的是,当通信模块完成系统初始化能够正常工作后,将重新启动通信模块的通讯功能,并借助通讯模块的通讯功能,以联网更新地区域,并且响应应用程序的请求。

[0050] 综上,本实施例当中的应用程序响应方法,通过在获取到预置唤醒源以外的唤醒源时,采用预置程序的本地控制功能,以在本地响应应用程序请求,因预置程序运行时只使用通信模块上的处理器,而不需要使用通信功能,因此大大缩短了通信模块唤醒后的系统初始化时间,把App响应时延从通常的2-3秒缩短到5毫秒以内,基本可以满足所有低功耗IoT系统控制的需求,使低功耗IoT产品在硬件设计上也可以完全去掉MCU,降低了低功耗IoT产品的硬件成本。

[0051] 实施例二

[0052] 请参阅图2,是本发明实施例二提供的一种应用程序响应方法的流程示意图,应用于通信模块当中,所述通信模块可由软件和/或硬件来实现所述方法,所述方法具体包括步骤S11至步骤S19。

[0053] 步骤S11,当所述通信模块在省电模式下被唤醒时,获取所述通信模块的唤醒源。

[0054] 步骤S12,判断所述唤醒源是否为预设唤醒源。

[0055] 其中,所述唤醒源为当前获取的唤醒源,所述预设唤醒源为TAU Alarm唤醒。当判断到所述唤醒源不为预设唤醒源时,执行步骤S13-步骤S18,当判断到所述唤醒源为预设唤醒源时,则直接跳转至执行步骤S19。在具体实施时,可以在通信模块上增加Bootloader(若已经有Bootloader的通信模块则不需要再增加Bootloader),并由Bootloader来区分不同的唤醒源。

[0056] 步骤S13,关闭预设时钟和电源之外的其它时钟和电源,所述预设时钟和电源为预置程序运行所需要的时钟和电源。

[0057] 可以理解的,通过关闭所有不用设备的时钟和电源,以最低系统功耗来执行预置程序BootApp,可以有效降低通信模块的功耗,使通信模块处于极低功耗状态下运行。

[0058] 步骤S14,执行所述预置程序的本地控制功能,以在本地响应应用程序请求。

[0059] 其中,所述预置程序可以为BootApp,具体可通过Bootloader调用BootApp的入口函数来执行BootApp,以进行系统所需的检测、处理和控制在等工作。

[0060] 步骤S15,将所述预置程序的执行数据保存于通信模块的文件系统中。

[0061] 其中,所述预置程序的执行数据可包含在预置程序的执行返回值当中,或从所述预置程序的执行日志中提取得到,通过将BootApp执行中生产的有用数据保存于通信模块的文件系统中,以供App进行访问。

[0062] 步骤S16,判断所述预置程序是否完成所有预设工作。

[0063] 其中,当判断到所述预置程序完成所有预设工作时,则执行步骤S17,当判断到所述预置程序未完成所有预设工作时,则执行步骤S18。

[0064] 在具体实施时,步骤S16可按以下细化步骤进行具体实施,细化步骤具体包括:

[0065] 判断所述预置程序的执行返回值是否为预设值;当所述执行返回值为所述预设值

时,则判定所述预置程序完成所有预设工作,当所述执行返回值不为所述预设值时,则判定所述预置程序未完成所有预设工作。

[0066] 其中,所述预设值可以为0,即当BootApp执行完所有应用程序的请求后,将向Bootloader返回执行状态值0,以表征BootApp执行完所有工作;否则向Bootloader返回非0的执行状态值,以表征BootApp执行完部分应用程序的请求。

[0067] 步骤S17,控制所述通信模块进入所述省电模式。

[0068] 步骤S18,控制所述通信模块进入WFI状态,并等待所述通信模块被中断唤醒后再次执行所述预置程序。

[0069] 具体地,通过BootApp返回值确定是否再次直接进入PSM状态还是WFI状态,如果进入PSM状态则相当于BootApp完成了所有工作等待下一次任务,如果进入WFI状态则相当于BootApp完成了部分工作,通信模块进入低功耗WFI状态并等待通信模块被中断唤醒后再次执行BootApp。

[0070] 步骤S19,初始化启动通信模块,并在所述通信模块初始化后响应所述应用程序请求。

[0071] 需要指出的是,当通信模块完成系统初始化能够正常工作后,将启动通信模块的通讯功能,并借助通讯模块的通讯功能,更新跟踪区域信息,并且响应应用程序的请求。

[0072] 另外,需要进一步说明的是,在本实施例当中,上述各步骤均可由通讯模块的Bootloader来执行,但本发明并不限于此,在其它实施例当中,也可以用BootApp替代Bootloader,以在BootApp中根据唤醒源判断启动通信模块还是执行BootApp的本地控制功能。

[0073] 实施例三

[0074] 本发明另一方面还提出一种应用程序响应系统,请参阅图3,是本发明实施例三提供的一种应用程序响应系统的模块示意图,应用于通信模块当中,所述通信模块可由软件和/或硬件来实现,所述应用程序响应系统具体包括:

[0075] 唤醒源获取模块11,用于当所述通信模块在省电模式下被唤醒时,获取所述通信模块的唤醒源;

[0076] 唤醒源判断模块12,用于判断所获取的唤醒源是否为预设唤醒源;

[0077] 应用程序响应模块13,用于当判断到所述唤醒源不为所述预设唤醒源时,执行预置程序的本地控制功能,以在本地响应应用程序请求。

[0078] 需要说明的是,省电模式即为PSM低功耗状态,通信模块在PSM状态下会关闭除RTC以外包括大容量存储器在内的所有片上设备,仅保留RTC相关模块的供电及时钟,以便在通信模块需要工作时,控制通信模块退出PSM状态,以进入正常工作状态。

[0079] 其中,所述唤醒源可以为唤醒通信模块的触发事件,所述唤醒源用于唤醒通信模块,以使通信模块退出PSM状态进入正常工作状态。所述唤醒源具体可以为外部中断唤醒、RTC唤醒、TAU (Tracking Area Update,更新跟踪区域Alarm)唤醒等。

[0080] 具体地,所述预设唤醒源可以为TAU Alarm唤醒,一般来说,非TAU Alarm唤醒表明系统有实时控制请求(如按键、系统掉电、定时数据采样等),此时可调用通讯模块当中的预置程序来进行系统所需的检测、处理和控制在等工作,以响应应用程序的运行。而TAU Alarm唤醒表明系统有跟踪区域更新的请求,此时必须借助通讯模块的通讯功能,联网更新跟踪



区域信息。

[0081] 其中,所述预置程序可以为BootApp,具体可以在通信模块上增加Bootloader(若已经有Bootloader的通信模块则不需要再增加Bootloader),通过Bootloader调用执行BootApp,以执行BootApp的本地控制功能来进行系统所需的检测、处理和控制等工作。在具体实施时,BootApp代码可存储于独立的存储空间,使用独立于Bootloader的SRAM和Flash空间,BootApp与Bootloader共享堆和栈空间,BootApp可由专门的独立于App的SDK编译得到,并且BootApp可由专门的下载工具下载到通信模块当中,也可以使用与App相同的下载工具下载到独立的BootApp分区当中。

[0082] 综上,本实施例当中的应用程序响应系统,通过在获取到预置唤醒源以外的唤醒源时,采用预置程序的本地控制功能,以在本地响应应用程序请求,因预置程序运行时只使用通信模块上的处理器,而不需要使用通信功能,因此大大缩短了通信模块唤醒后的系统初始化时间,把App响应时延从通常的2-3秒缩短到5毫秒以内,基本可以满足所有低功耗IoT系统控制的需求,使低功耗IoT产品在硬件设计上也可以完全去掉MCU,降低了低功耗IoT产品的硬件成本。

[0083] 进一步地,在本发明一些可选实施例当中,所述系统还包括:

[0084] 时钟和电源控制模块,用于开/关预设时钟和电源之外的其它时钟和电源,所述预设时钟和电源为所述预置程序运行所需要的时钟和电源。

[0085] 进一步地,在本发明一些可选实施例当中,应用程序响应模块13还用于当判断到所述唤醒源为所述预设唤醒源时,初始化启动通信模块,并在所述通信模块初始化后响应所述应用程序请求。

[0086] 需要指出的是,当通信模块完成系统初始化能够正常工作后,将启动通信模块的通讯功能,并借助通讯模块的通讯功能,联网更新跟踪区域信息,并响应应用程序的数据收发请求。

[0087] 进一步地,在本发明一些可选实施例当中,所述系统还包括:

[0088] 完成判断模块,用于判断所述预置程序是否完成所有预设工作;

[0089] 模式控制模块,用于当判断到所述预置程序完成所有预设工作时,控制所述通信模块进入所述省电模式。

[0090] 进一步地,在本发明一些可选实施例当中,模式控制模块还用于当判断到所述预置程序未完成所有预设工作时,控制所述通信模块进入WFI状态,并等待所述通信模块被中断唤醒后再次执行所述预置程序。

[0091] 进一步地,在本发明一些可选实施例当中,完成判断模块还用于判断所述预置程序的执行返回值是否为预设值;当所述执行返回值为所述预设值时,则判定所述预置程序完成所有预设工作;当所述执行返回值不为所述预设值时,则判定所述预置程序未完成所有预设工作。

[0092] 进一步地,在本发明一些可选实施例当中,所述系统还包括:

[0093] 数据保存模块,用于将所述预置程序的执行数据保存于通信模块的文件系统中。

[0094] 上述各模块、单元被执行时所实现的功能或操作步骤与上述方法实施例大体相同,在此不再赘述。

[0095] 实施例四

[0096] 本发明另一方面还提出一种通信模块,请参阅图4,所示为本发明第四实施例当中的通信模块的模块结构图,包括存储器20、处理器10以及存储在存储器上并可在处理器上运行的计算机程序30,所述处理器执行所述程序时实现如上述的应用程序响应方法。

[0097] 具体地,所述通信模块可以为IoT产品的无线通信模块,例如射频通信模块、蓝牙通信模块、wifi通信模块等。所述处理器10在一些实施例中可以是一中央处理器(Central Processing Unit,CPU)、控制器、微控制器、微处理器或其他数据处理芯片,用于运行存储器20中存储的程序代码或处理数据。

[0098] 其中,存储器20至少包括一种类别的可读存储介质,所述可读存储介质包括闪存、硬盘、多媒体卡、卡型存储器(例如,SD或DX存储器等)、磁性存储器、磁盘、光盘等。存储器20在一些实施例中可以是装置的内部存储单元,例如该装置的硬盘。存储器20在另一些实施例中也可以是装置的外部存储设备,例如装置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,存储器20还可以既包括装置的内部存储单元也包括外部存储设备。存储器20不仅可以用于存储安装于装置的应用软件及各类数据,还可以用于暂时地存储已经输出或者将要输出的数据。

[0099] 可选地,所述通信模块还可以包括用户接口、网络接口、通信总线等,用户接口可以包括显示器(Display)、输入单元比如遥控器、实体按键等,可选的用户接口还可以包括标准的有线接口、无线接口。可选地,在一些实施例中,显示器可以是LED显示器、液晶显示器、触控式液晶显示器以及OLED(Organic Light-Emitting Diode,有机发光二极管)触摸器等。其中,显示器也可以适当的称为显示屏或显示单元,用于显示装置中信息以及用于显示可视化的用户界面。网络接口可以包括标准的有线接口、无线接口(如WI-FI接口),通常用于在该装置与其他电子设备之间建立通信连接。通信总线用于实现这些组件之间的连接通信。

[0100] 需要指出的是,图4示出的结构并不构成对装置的限定,在其它实施例当中,该装置可以包括比图示更少或者更多的部件,或者组合某些部件,或者不同的部件布置。

[0101] 综上,本实施例当中的通信模块,通过在获取到预置唤醒源以外的唤醒源时,采用预置程序的本地控制功能,以在本地响应应用程序请求,因预置程序运行时只使用通信模块上的处理器,而不需要使用通信功能,因此大大缩短了通信模块唤醒后的系统初始化时间,把App响应时延从通常的2-3秒缩短到5毫秒以内,基本可以满足所有低功耗IoT系统控制的需求,使低功耗IoT产品在硬件设计上也可以完全去掉MCU,降低了低功耗IoT产品的硬件成本。

[0102] 本实施例还提供了一种计算机可读存储介质,其上存储有上述通信模块中所使用的计算机程序30,该程序在被处理器执行时实现上述的应用程序响应方法。

[0103] 其中,所述的存储介质可以为但不限于ROM/RAM、磁碟、光盘等。

[0104] 本领域技术人员可以理解,在流程图中表示或在此以其他方式描述的逻辑和/或步骤,例如,可以被认为是用于实现逻辑功能的可执行指令的定序列列表,可以具体实现在任何计算机可读介质中,以供指令执行系统、装置或设备(如基于计算机的系统、包括处理器的系统或其他可以从指令执行系统、装置或设备取指令并执行指令的系统)使用,或结合这些指令执行系统、装置或设备而使用。就本说明书而言,“计算机可读介质”可以是任何可以

包含、存储、通信、传播或传输程序以供指令执行系统、装置或设备或结合这些指令执行系统、装置或设备而使用的装置。

[0105] 计算机可读介质的更具体的示例(非穷尽性列表)包括以下:具有一个或多个布线的电连接部(电子装置),便携式计算机盘盒(磁装置),随机存取存储器(RAM),只读存储器(ROM),可擦除可编程只读存储器(EPROM或Flash),光纤装置,以及便携式光盘只读存储器(CDROM)。另外,计算机可读介质甚至可以是可在其上打印所述程序的纸或其他合适的介质,因为可以通过对纸或其他介质进行光学扫描,接着进行编辑、解译或必要时以其他合适方式进行处理后以电子方式获得所述程序,然后将其存储在计算机存储器中。

[0106] 应当理解,本发明的各部分可以用硬件、软件、固件或它们的组合来实现。在上述实施方式中,多个步骤或方法可以用存储在存储器中且由合适的指令执行系统执行的软件或固件来实现。例如,如果用硬件来实现,和在另一实施方式中一样,可用本领域公知的下列技术中的任一项或它们的组合来实现:具有用于对数据信号实现逻辑功能的逻辑门电路的离散逻辑电路,具有合适的组合逻辑门电路的专用集成电路,可编程门阵列(PGA),现场可编程门阵列(FPGA)等。

[0107] 在本说明书的描述中,参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包含于本发明的至少一个实施例或示例中。在本说明书中,对上述术语的示意性表述不一定指的是相同的实施例或示例。而且,描述的具体特征、结构、材料或者特点可以在任何一个或多个实施例或示例中以合适的方式结合。

[0108] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,均应包含在本发明的保护范围之内。

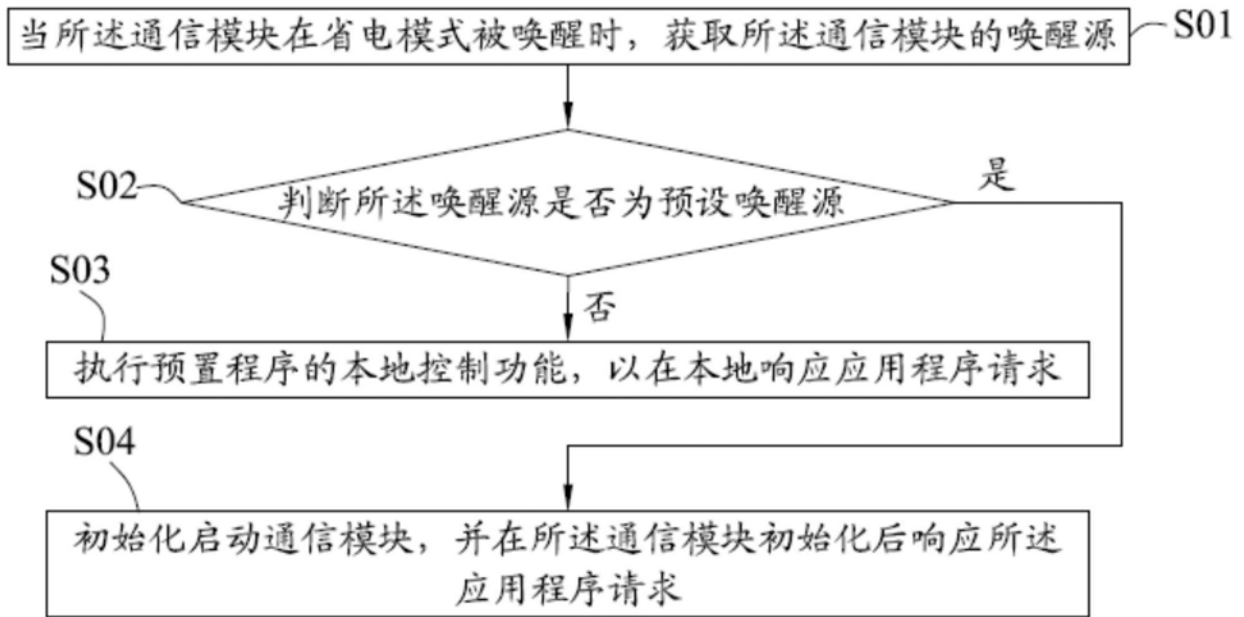


图1

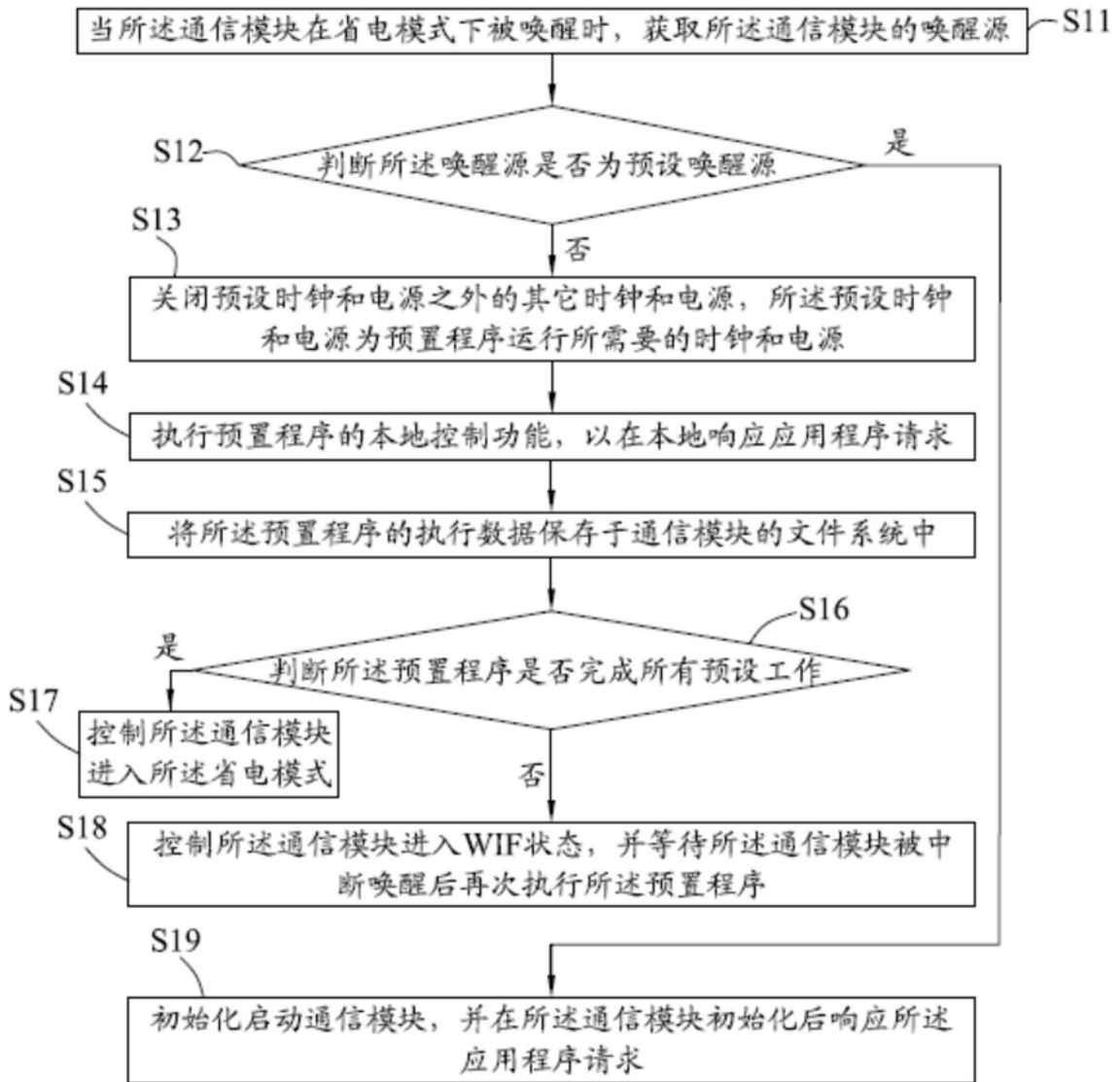


图2

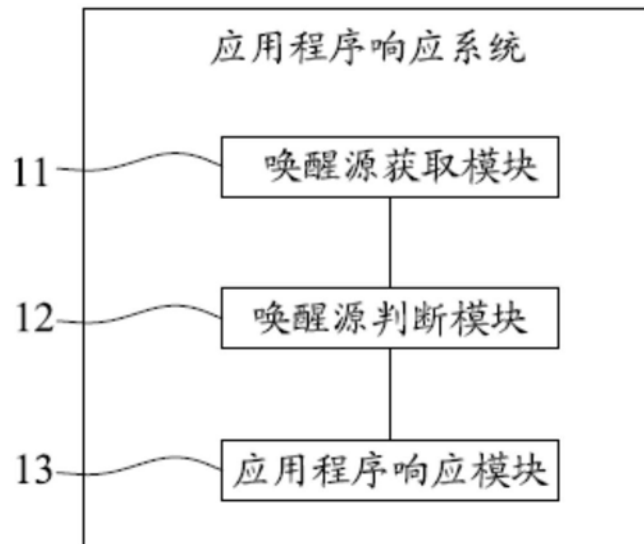


图3

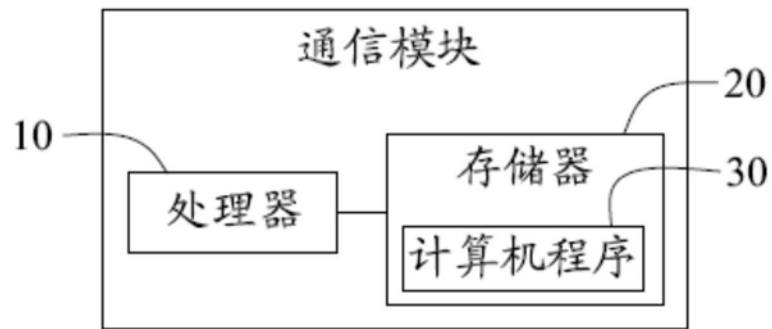


图4