



(12) 发明专利

(10) 授权公告号 CN 115509153 B

(45) 授权公告日 2024.09.27

(21) 申请号 202110697429.0

审查员 童阳

(22) 申请日 2021.06.23

(65) 同一申请的已公布的文献号

申请公布号 CN 115509153 A

(43) 申请公布日 2022.12.23

(73) 专利权人 中移物联网有限公司

地址 401336 重庆市南岸区玉马路8号

专利权人 中国移动通信集团有限公司

(72) 发明人 卢林 李蒙 许睿 庞超 范海鹏

周林海

(74) 专利代理机构 北京银龙知识产权代理有限

公司 11243

专利代理师 许静 姜精斌

(51) Int. Cl.

G05B 19/042 (2006.01)

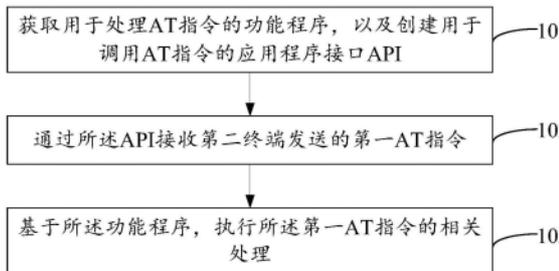
权利要求书2页 说明书10页 附图3页

(54) 发明名称

一种窄带物联网NB-IoT通信模组的二次开发方法及装置

(57) 摘要

本发明提供了一种窄带物联网NB-IoT通信模组的二次开发方法及装置,所述方法包括:获取用于处理AT指令的功能程序,以及创建用于调用AT指令的应用程序接口API;通过所述API接收第二终端发送的第一AT指令;基于所述功能程序,执行所述第一AT指令的相关处理。本发明方案,通过获取用于处理AT指令的功能程序,以及创建用于调用AT指令的应用程序接口API,可以实现在NB-IoT通信模组使用AT指令做二次开发,通过所述API接收第二终端发送的第一AT指令,基于所述功能程序,执行所述第一AT指令的相关处理,降低开发成本和开发时长。



1. 一种窄带物联网NB-IoT通信模组的二次开发方法,应用于第一终端,其特征在于,包括:

获取用于处理AT指令的功能程序,以及创建用于调用AT指令的应用程序接口API;

通过所述API接收第二终端发送的第一AT指令;

基于所述功能程序,执行所述第一AT指令的相关处理;

其中,所述执行所述第一AT指令的相关处理,包括:

获取与所述第一AT指令对应的目标数据;

通过所述API向所述第二终端发送所述目标数据;

其中,所述通过所述API向所述第二终端发送所述目标数据,包括:

在所述NB-IoT通信模组处于标准模式的情况下,通过所述API将所述目标数据传输至第一物理串口,所述第一物理串口是第一终端与所述第二终端之间进行通信的串口;

在所述NB-IoT通信模组处于OpenCPU模式的情况下,通过所述API将所述目标数据传输至第一AT指令线程,所述第一AT指令线程是第一终端向所述第二终端发送所述AT指令的线程。

2. 根据权利要求1所述的方法,其特征在于,所述获取用于处理AT指令的功能程序,包括:

根据NB-IoT通信模组的资源配置,生成软件开发工具包SDK;

通过所述SDK,获取所述用于处理AT指令的功能程序。

3. 根据权利要求2所述的方法,其特征在于,所述通过所述SDK,获取所述用于处理AT指令的功能程序,包括:

通过所述SDK,移植所述第一终端中的微控制单元MCU的所述功能程序;其中,所述NB-IoT通信模组属于所述第一终端;

和/或,

通过所述SDK,创建所述功能程序。

4. 根据权利要求2所述的方法,其特征在于,所述根据NB-IoT通信模组的资源配置,生成软件开发工具包SDK之前,所述方法还包括:

接收对所述NB-IoT通信模组的图形配置界面的配置操作;

响应于所述配置操作,确定所述NB-IoT通信模组的资源配置。

5. 根据权利要求1所述的方法,其特征在于,所述基于所述功能程序,执行所述第一AT指令的相关处理,包括:

基于所述功能程序,对所述第一AT指令进行格式化处理,得到格式化处理后的第一AT指令;

根据所述格式化处理后的第一AT指令,执行所述第一AT指令的相关处理。

6. 根据权利要求1所述的方法,其特征在于,所述获取与所述第一AT指令对应的目标数据之后,所述方法还包括:

在所述NB-IoT通信模组连接日志工具的情况下,通过所述NB-IoT通信模组将所述第一AT指令和所述目标数据传输至所述日志工具;

通过所述日志工具对所述AT指令和所述目标数据进行存储。

7. 一种窄带物联网NB-IoT通信模组的二次开发装置,应用于第一终端,其特征在于,包

括:

处理模块,用于获取用于处理AT指令的功能程序,以及创建用于调用AT指令的应用程序接口API;

发送模块,用于通过所述API接收第二终端发送的第一AT指令;

执行模块,用于基于所述功能程序,执行所述第一AT指令的相关处理;

其中,所述处理模块还包括:

第二获取单元,用于获取与所述第一AT指令对应的目标数据;

发送单元,用于通过所述API向所述第二终端发送所述目标数据;

其中,所述发送单元具体用于:

在所述NB-IoT通信模组处于标准模式的情况下,通过所述API将所述目标数据传输至第一物理串口,所述第一物理串口是第一终端与所述第二终端之间进行通信的串口;

在所述NB-IoT通信模组处于OpenCPU模式的情况下,通过所述API将所述目标数据传输至第一AT指令线程,所述第一AT指令线程是第一终端向所述第二终端发送所述AT指令的线程。

8. 根据权利要求7所述的装置,其特征在于,所述处理模块包括:

生成单元,用于根据NB-IoT通信模组的资源配置,生成软件开发工具包SDK;

获取单元,用于通过所述SDK,获取所述用于处理AT指令的功能程序。

9. 一种终端设备,所述终端设备为第一终端,其特征在于,包括:处理器、存储器及存储在所述存储器上并可在所述处理器上运行的程序,所述程序被所述处理器执行时实现如权利要求1至6中任一项所述的窄带物联网NB-IoT通信模组的二次开发方法的步骤。

10. 一种可读存储介质,其特征在于,所述可读存储介质上存储程序或指令,所述程序或指令被处理器执行时实现如权利要求1至6中任一项所述的窄带物联网NB-IoT通信模组的二次开发方法的步骤。

一种窄带物联网NB-IoT通信模组的二次开发方法及装置

技术领域

[0001] 本发明涉及通信技术领域,特别涉及一种窄带物联网NB-IoT通信模组的二次开发方法及装置。

背景技术

[0002] 随着物联网技术的快速发展进步,基于蜂窝的窄带物联网(Narrow Band Internet of Things,NB-IoT)成为万物互联网络的一个重要分支。由于用户在使用NB-IoT通信模组的过程中将产生需求的变化,因此需要对NB-IoT通信模组进行二次开发,从而使得NB-IoT通信模组更贴合用户需求。

[0003] 现有的NB-IoT通信模组将单片机与NB-IoT通信模组进行连接的操作过程复杂,而且连接单片机必然需要占据更多的空间,外围电路冗杂,另外单片机的使用也增加了进行二次开发的成本,随着物联网应用复杂度的提高,低价微控制单元(Microcontroller Unit,MCU),又称单片机的硬件资源已经不能满足开发需求,需要更换价格更高的MCU,使得设备成本增加;并且,现有技术中,对NB-IoT通信模组的开发,在虚拟机中将程序翻译成机器语言,但是其影响中央处理器(central processing unit,CPU)的运行效率,同时习惯使用MCU开发模式的人员一般不具备解释性语言开发能力。由于基带厂商设计OpenCPU方案的历史较短,一方面提供的应用程序接口(Application Programming Interface,API)不够丰富,API接口设计也不够合理,另一方面对于原来习惯使用MCU开发的用户,已经具备一套成熟的AT(Attention)处理框架,更换成复杂的API方式需要较长的学习时间,不利于产品的快速上市。

[0004] 因此需要设计一种NB-IoT通信模组的二次开发方法,降低MCU的成本和开发时间成本,使产品快速上市。

发明内容

[0005] 本发明实施例提供一种窄带物联网NB-IoT通信模组的二次开发方法及装置,用以解决现有技术中,MCU的开发成本高、开发时间长的问题。

[0006] 为了解决上述技术问题,本发明实施例提供如下技术方案:

[0007] 本发明实施例提供一种窄带物联网NB-IoT通信模组的二次开发方法,应用于第一终端,包括:

[0008] 获取用于处理AT指令的功能程序,以及创建用于调用AT指令的应用程序接口API;

[0009] 通过所述API接收第二终端发送的第一AT指令;

[0010] 基于所述功能程序,执行所述第一AT指令的相关处理。

[0011] 可选地,所述获取用于处理AT指令的功能程序,包括:

[0012] 根据NB-IoT通信模组的资源配置,生成软件开发工具包SDK;

[0013] 通过所述SDK,获取所述用于处理AT指令的功能程序。

[0014] 可选地,所述通过所述SDK,获取所述用于处理AT指令的功能程序,包括:

- [0015] 通过所述SDK,移植所述第一终端中的微控制单元MCU的所述功能程序;其中,所述NB-IoT通信模组属于所述第一终端;
- [0016] 和/或,
- [0017] 通过所述SDK,创建所述功能程序。
- [0018] 可选地,所述根据NB-IoT通信模组的资源配置,生成软件开发工具包SDK之前,所述方法还包括:
- [0019] 接收对所述NB-IoT通信模组的图形配置界面的配置操作;
- [0020] 响应于所述配置操作,确定所述NB-IoT通信模组的资源配置。
- [0021] 可选地,所述基于所述功能程序,执行所述第一AT指令的相关处理,包括:
- [0022] 基于所述功能程序,对所述第一AT指令进行格式化处理,得到格式化处理后的第一AT指令;
- [0023] 根据所述格式化处理后的第一AT指令,执行所述第一AT指令的相关处理。
- [0024] 可选地,所述执行所述第一AT指令的相关处理,包括:
- [0025] 获取与所述第一AT指令对应的目标数据;
- [0026] 通过所述API向所述第二终端发送所述目标数据。
- [0027] 可选地,所述通过所述API向所述第二终端发送所述目标数据,包括:
- [0028] 在所述NB-IoT通信模组处于标准模式的情况下,通过所述API将所述目标数据传输至第一物理串口,所述第一物理串口是第一终端与所述第二终端之间进行通信的串口;
- [0029] 在所述NB-IoT通信模组处于OpenCPU模式的情况下,通过所述API将所述目标数据传输至第一AT指令线程,所述第一AT指令线程是第一终端向所述第二终端发送所述AT指令的线程。
- [0030] 可选地,所述获取与所述第一AT指令对应的目标数据之后,所述方法还包括:
- [0031] 在所述NB-IoT通信模组连接日志工具的情况下,通过所述NB-IoT通信模组将所述第一AT指令和所述目标数据传输至所述日志工具;
- [0032] 通过所述日志工具对所述AT指令和所述目标数据进行存储。
- [0033] 本发明实施例还提供一种窄带物联网NB-IoT通信模组的二次开发装置,应用于第一终端,包括:
- [0034] 处理模块,用于获取用于处理AT指令的功能程序,以及创建用于调用AT指令的应用程序接口API;
- [0035] 发送模块,用于通过所述API接收第二终端发送的第一AT指令;
- [0036] 执行模块,用于基于所述功能程序,执行所述第一AT指令的相关处理。
- [0037] 可选地,所述处理模块包括:
- [0038] 生成单元,用于根据NB-IoT通信模组的资源配置,生成软件开发工具包SDK;
- [0039] 第一获取单元,用于通过所述SDK,获取所述用于处理AT指令的功能程序。
- [0040] 可选地,所述获取单元具体用于:
- [0041] 通过所述SDK,移植所述第一终端中的微控制单元MCU的所述功能程序;其中,所述NB-IoT通信模组属于所述第一终端;
- [0042] 和/或,
- [0043] 通过所述SDK,创建所述功能程序。

- [0044] 可选地,所述处理模块还包括:
- [0045] 接收单元,用于接收对所述NB-IoT通信模组的图形配置界面的配置操作;
- [0046] 确定单元,用于响应于所述配置操作,确定所述NB-IoT通信模组的资源配置。
- [0047] 可选地,所述执行模块包括:
- [0048] 处理单元,用于基于所述功能程序,对所述第一AT指令进行格式化处理,得到格式化处理后的第一AT指令;
- [0049] 执行单元,用于根据所述格式化处理后的第一AT指令,执行所述第一AT指令的相关处理。
- [0050] 可选地,所述处理模块还包括:
- [0051] 第二获取单元,用于获取与所述第一AT指令对应的目标数据;
- [0052] 发送单元,用于通过所述API向所述第二终端发送所述目标数据。
- [0053] 可选地,所述发送单元具体用于:
- [0054] 在所述NB-IoT通信模组处于标准模式的情况下,通过所述API将所述目标数据传输至第一物理串口,所述第一物理串口是第一终端与所述第二终端之间进行通信的串口;
- [0055] 在所述NB-IoT通信模组处于OpenCPU模式的情况下,通过所述API将所述目标数据传输至第一AT指令线程,所述第一AT指令线程是第一终端向所述第二终端发送所述AT指令的线程。
- [0056] 可选地,所述处理模块还包括:
- [0057] 传输单元,用于在所述NB-IoT通信模组连接日志工具的情况下,通过所述NB-IoT通信模组将所述第一AT指令和所述目标数据传输至所述日志工具;
- [0058] 存储单元,用于通过所述日志工具对所述AT指令和所述目标数据进行存储。
- [0059] 本发明实施例还提供一种终端设备,所述终端设备为第一终端,包括:处理器、存储器及存储在所述存储器上并可在所述处理器上运行的程序,所述程序被所述处理器执行时实现如上任一项所述的窄带物联网NB-IoT通信模组的二次开发方法的步骤。
- [0060] 本发明实施例还提供着一种可读存储介质,所述可读存储介质上存储程序或指令,所述程序或指令被处理器执行时实现如上任一项所述的窄带物联网NB-IoT通信模组的二次开发方法的步骤。
- [0061] 本发明的有益效果是:
- [0062] 本发明方案,通过获取用于处理AT指令的功能程序,以及创建用于调用AT指令的应用程序接口API,可以实现在NB-IoT通信模组使用AT指令做二次开发,通过所述API接收第二终端发送的第一AT指令,基于所述功能程序,执行所述第一AT指令的相关处理,降低开发成本和开发时长。

附图说明

- [0063] 图1表示本发明实施例提供的窄带物联网NB-IoT通信模组的二次开发方法的流程图;
- [0064] 图2表示本发明实施例提供的NB-IoT通信模组的组件架构图;
- [0065] 图3表示本发明实施例提供的NB-IoT通信模组的OpenCPU开发模式的流程图;
- [0066] 图4表示本发明实施例提供的NB-IoT通信模组的硬件结构示意图;

[0067] 图5表示本发明实施例提供的窄带物联网NB-IoT通信模组的二次开发装置的结构示意图;

[0068] 图6表示本发明实施例提供的终端设备的结构示意图。

具体实施方式

[0069] 为使本发明的目的、技术方案和优点更加清楚,下面将结合附图及具体实施例对本发明进行详细描述。

[0070] 本发明针对现有技术中,MCU的开发成本高、开发时间长的问题,提供一种窄带物联网NB-IoT通信模组的二次开发方法及装置。

[0071] 如图1所示,本发明实施例提供一种窄带物联网NB-IoT通信模组的二次开发方法,应用于第一终端,包括:

[0072] 步骤101:获取用于处理AT指令的功能程序,以及创建用于调用AT指令的应用程序接口API。

[0073] 需要说明的是,终端设备可以是电子设备,该电子设备可以是能接收、发送数据,且与其他设备进行通信的智能终端,如手机、平板电脑、掌上电脑等,也可以是具有数据收发与处理功能的处理器。

[0074] 在本发明实施例中,提供一种AT虚拟通道,增加一个用于处理和传输AT指令的API接口。

[0075] 需要说明的是,本发明实施例的获取用于处理AT指令的功能程序的步骤,以及创建用于调用AT指令的应用程序接口API的步骤可以同时进行,也可以先后进行,且进行顺序在本发明实施例中不做限定。

[0076] 本发明实施例提供一套虚拟AT通道,提出了在NB-IoT通信模组上使用AT指令做OpenCPU开发的方式。

[0077] 步骤102:通过所述API接收第二终端发送的第一AT指令。

[0078] 在本发明实施例中,通过API接口,接收终端发送的第一AT指令。本发明实施例可以实现通过虚拟AT通道调用AT指令。

[0079] 步骤103:基于所述功能程序,执行所述第一AT指令的相关处理。

[0080] 在本发明实施例中,基于功能程序,执行对第一AT指令的相关处理,可以实现在NB-IoT通信模组上的设置功能程序模块,通过虚拟AT通道调用AT指令,并执行对AT指令的相关处理,降低开发成本和开发时间。

[0081] 可选地,所述获取用于处理AT指令的功能程序,包括:

[0082] 根据NB-IoT通信模组的资源配置,生成软件开发工具包SDK;

[0083] 通过所述SDK,获取所述用于处理AT指令的功能程序。

[0084] 在本发明实施例中,通过对NB-IoT通信模组进行资源配置,以此形成满足用户需求的专用SDK,将功能程序移植到NB-IoT通信模组的SDK的APP目录。

[0085] 可选地,所述通过所述SDK,获取所述用于处理AT指令的功能程序,包括:

[0086] 通过所述SDK,移植所述第一终端中的微控制单元MCU的所述功能程序;其中,所述NB-IoT通信模组属于所述第一终端;

[0087] 和/或,

[0088] 通过所述SDK,创建所述功能程序。

[0089] 在本发明实施例中,可以将原来运行在第一终端中的MCU中的功能程序进行简单的移植,和/或直接通过SDK,创建功能程序,使用本NB-IoT通信模组提供的构建系统对该功能程序进行管理。

[0090] 在本发明实施例中,将原MCU中调用AT指令的接口替换为NB-IoT通信模组提供的虚拟AT通道API,将原MCU的功能程序进行快速一直与构建,已放线可以为终端厂家节省一个MCU,减少电路设计的复杂程度,降低开发成本,降低开发时间。

[0091] 举例来说,在对功能程序进行移植之前,第一终端中存在一个MCU,在对功能程序进行移植,以及创建API之后,该MCU可以去掉。

[0092] 可选地,所述根据NB-IoT通信模组的资源配置,生成软件开发工具包SDK之前,所述方法还包括:

[0093] 接收对所述NB-IoT通信模组的图形配置界面的配置操作;

[0094] 响应于所述配置操作,确定所述NB-IoT通信模组的资源配置。

[0095] 在本发明实施例中,对NB-IoT通信模组进行二次开发时,首先根据NB-IoT通信模组的资源进行配置和剪裁。

[0096] 为NB-IoT通信模组的二次开发方法增加图形配置界面,利用图形配置界面选择业务所需的组件,包括但不限于:

[0097] 通信协议组件:轻量级机器到机器(Lightweight Machine-To-Machine,LwM2M)协议、网络时间协议(Network Time Protocol,NTP)、受限应用协议(Constrained Application Protocol,CoAP)、域名系统(Domain Name System,DNS)等;

[0098] 驱动框架组件:互联集成电路(Inter-Integrated Circuit,I2C)、串行外设接口(Serial Peripheral Interface,SPI)、安全数字输入输出(Secure Digital Input and Output,SDIO)、闪存(FLASH)、脉冲宽度调制(Pulse width modulation,PWM)等;

[0099] 其他应用组件:空中下载技术(Over-the-Air Technology,OTA)、编程语言JS对象简谱(JavaScript Object Notation,JSON)、计算机排除故障(Debug)、通道追踪(BankTrace)等。

[0100] 本NB-IoT通信模组的支持的组件如图2所示。

[0101] 图形配置界面可以实现对模组的功能进行参数配置,以达到用户需求的定制化,以及可以实现对模组提供的组件进行剪裁,以减少固件的大小,节约FLASH空间。

[0102] 具体来说,部分组件需要进行参数的配置,例如:OTA通信模组支持下载协议、检测周期、重试次数等。用户配置和剪裁完成后,系统根据配置生成定制化的配置头文件config.h头文件,该头文件中包含各组件的宏开关和宏配置参数。

[0103] 可选地,所述基于所述功能程序,执行所述第一AT指令的相关处理,包括:

[0104] 基于所述功能程序,对所述第一AT指令进行格式化处理,得到格式化处理后的第一AT指令;

[0105] 根据所述格式化处理后的第一AT指令,执行所述第一AT指令的相关处理。

[0106] 在本发明实施例中,基于功能程序,提供的API接口用于格式化处理和传输AT指令,对AT指令进行格式化处理包括:对AT指令进行语法检测、格式校验、去除指令内的空格,末尾换行符处理,将格式化后的AT指令传入AT功能模块,由AT功能模块分发至对应的应用

程序。

[0107] 需要说明的是,该AT功能模块是处理和分发AT指令的模块。

[0108] 可选地,所述执行所述第一AT指令的相关处理,包括:

[0109] 获取与所述第一AT指令对应的目标数据;

[0110] 通过所述API向所述第二终端发送所述目标数据。

[0111] 在本发明实施例中,基于功能程序,在触发AT指令的任务中,等待并截获AT功能模块通过API返回的处理数据。

[0112] 可选地,所述通过所述API向所述第二终端发送所述目标数据,包括:

[0113] 在所述NB-IoT通信模组处于标准模式的情况下,通过所述API将所述目标数据传输至第一物理串口,所述第一物理串口是第一终端与所述第二终端之间进行通信的串口;

[0114] 在所述NB-IoT通信模组处于OpenCPU模式的情况下,通过所述API将所述目标数据传输至第一AT指令线程,所述第一AT指令线程是第一终端向所述第二终端发送所述AT指令的线程。

[0115] 在本发明实施例中,在截获AT功能模块通过API返回的处理数据的过程中,在数据处理的位置增加判断,若判断出NB-IoT通信模组处于标准模式,发送数据的过程操作不变,直接通过API将数据传输至第一物理串口;

[0116] 若判断出NB-IoT通信模组处于OpenCPU模式,通过事件的方式,通过API将数据传输至发送第一AT指令的线程。

[0117] 可选地,所述获取与所述第一AT指令对应的目标数据之后,所述方法还包括:

[0118] 在所述NB-IoT通信模组连接日志工具的情况下,通过所述NB-IoT通信模组将所述第一AT指令和所述目标数据传输至所述日志工具;

[0119] 通过所述日志工具对所述AT指令和所述目标数据进行存储。

[0120] 在本发明实施例中,将NB-IoT通信模组连接日志工具时,实时地将接收到的AT指令和AT功能模块返回的数据传输到日志工具,方便用户的开发过程,同时,将这些信息通过离线日志功能存储到日志分区,方便用户进行问题的定位与分析。

[0121] 下面结合图3,具体说明将MCU程序快速移植到NB-IoT通信模组进行OpenCPU开发方法。

[0122] 首先根据NB-IoT通信模组的资源进行配置和剪裁,利用图形化配置界面,选择所需的组件,之后对一些主要进行参数配置的组件进行配置,用户配置和裁剪组件完成后,系统根据配置完成定制化的config.h头文件。将原MCU功能程序移植到本NB-IoT通信模组的SDK的APP目录中,按照本NB-IoT通信模组提供的SCONS构建方式书写构建脚本。

[0123] 本NB-IoT通信模组提供有微控制器软件接口标准(Cortex Microcontroller Software Interface Standard,CMSIS)接口、可移植操作系统接口(Portable Operating System Interface,POSIX)、免费实时操作系统(Free Real-time Operation System,FreeRTOS)接口、实时操作系统(Micro-Controller Operating System,uCOS)接口,能适配大多数的MCU应用程序,程序移植后,对于系统的调用不需要做修改,只需要把原调用的AT指令发送接口替换成本NB-IoT通信模组提供的虚拟AT通道的发送接口,在第二终端向第一终端发送AT指令发送完毕后,第一终端阻塞等待AT功能模块处理完成返回的数据。本NB-IoT通信模组提供有统一的驱动框架,开发人员需要将原MCU上外围驱动的调用替换为驱动

框架的接口。驱动接口部分实现简单,替换较容易。

[0124] 业务功能移植完成后,即可以使用本NB-IoT通信模组提供的SCONS构建方式进行代码编译生成可运行的固件。本NB-IoT通信模组的二次开发方法还提供有编译链接检查,可以排查开发者的编程语法异常等错误。

[0125] 生成固件后即可通过串口烧录工具烧写到本NB-IoT通信模组提供的存储器,由处理器进行执行。执行过程中,如果调试串口连接到实时日志工具上,AT调用相关的日志会输出到工具上,用户可以根据日志输出进行功能验证与调试。同时,AT调用的相关信息也会记录到FLASH上的日志分区,方便事后审查与分析。

[0126] 本发明实施例,通过虚拟AT通道对AT指令进行传输与处理,实现将MCU的功能程序快速移植与构建,一方面为终端厂家节省一颗MCU的成本,减少电路设计复杂度;另一方面降低了开发成本,复用已有的业务程序,加快产品的快速上市。

[0127] 本发明实施例还提供一种NB-IoT通信模组,如图4所示,包括存储器401、处理器402、射频芯片403、外围控制器404,存储器401用于保存可运行的计算机程序,处理器402用于在本NB-IoT通信模组执行计算机程序,射频芯片403用于控制功放的性能、外围控制器404用于连接外围的传感器等。

[0128] 需要说明的是,在本申请实施例中,存储器401是用于存储NB-IoT通信模组的OpenCPU方法制作的固件;处理器402是用于执行NB-IoT通信模组的OpenCPU方法制作的固件。

[0129] 如图5所示,本发明实施例还提供一种窄带物联网NB-IoT通信模组的二次开发装置,应用于第一终端,包括:

[0130] 处理模块501,用于获取用于处理AT指令的功能程序,以及创建用于调用AT指令的应用程序接口API;

[0131] 发送模块502,用于通过所述API接收第二终端发送的第一AT指令;

[0132] 执行模块503,用于基于所述功能程序,执行所述第一AT指令的相关处理。

[0133] 本发明实施例,通过获取用于处理AT指令的功能程序,以及创建用于调用AT指令的应用程序接口API,可以实现在NB-IoT通信模组使用AT指令做二次开发,通过所述API接收第二终端发送的第一AT指令,基于所述功能程序,执行所述第一AT指令的相关处理,降低开发成本和开发时长。

[0134] 可选地,所述处理模块包括:

[0135] 生成单元,用于根据NB-IoT通信模组的资源配置,生成软件开发工具包SDK;

[0136] 第一获取单元,用于通过所述SDK,获取所述用于处理AT指令的功能程序。

[0137] 可选地,所述获取单元具体用于:

[0138] 通过所述SDK,移植所述第一终端中的微控制单元MCU的所述功能程序;其中,所述NB-IoT通信模组属于所述第一终端;

[0139] 和/或,

[0140] 通过所述SDK,创建所述功能程序。

[0141] 可选地,所述处理模块还包括:

[0142] 接收单元,用于接收对所述NB-IoT通信模组的图形配置界面的配置操作;

[0143] 确定单元,用于响应于所述配置操作,确定所述NB-IoT通信模组的资源配置。

[0144] 可选地,所述执行模块包括:

[0145] 处理单元,用于基于所述功能程序,对所述第一AT指令进行格式化处理,得到格式化处理后的第一AT指令;

[0146] 执行单元,用于根据所述格式化处理后的第一AT指令,执行所述第一AT指令的相关处理。

[0147] 可选地,所述处理模块还包括:

[0148] 第二获取单元,用于获取与所述第一AT指令对应的目标数据;

[0149] 发送单元,用于通过所述API向所述第二终端发送所述目标数据。

[0150] 可选地,所述发送单元具体用于:

[0151] 在所述NB-IoT通信模组处于标准模式的情况下,通过所述API将所述目标数据传输至第一物理串口,所述第一物理串口是第一终端与所述第二终端之间进行通信的串口;

[0152] 在所述NB-IoT通信模组处于OpenCPU模式的情况下,通过所述API将所述目标数据传输至第一AT指令线程,所述第一AT指令线程是第一终端向所述第二终端发送所述AT指令的线程。

[0153] 可选地,所述处理模块还包括:

[0154] 传输单元,用于在所述NB-IoT通信模组连接日志工具的情况下,通过所述NB-IoT通信模组将所述第一AT指令和所述目标数据传输至所述日志工具;

[0155] 存储单元,用于通过所述日志工具对所述AT指令和所述目标数据进行存储。

[0156] 需要说明的是,本发明实施例提供的窄带物联网NB-IoT通信模组的二次开发装置是能够执行上述的窄带物联网NB-IoT通信模组的二次开发方法的装置,则上述的窄带物联网NB-IoT通信模组的二次开发方法的所有实施例均适用于该装置,且能够达到相同或者相似的技术效果。

[0157] 如图6所示,本发明实施例还提供一种终端设备,所述终端设备为第一终端,包括:处理器600;以及通过总线接口与所述处理器600相连接的存储器610,所述存储器610用于存储所述处理器600在执行操作时所使用的程序和数据,处理器600调用并执行所述存储器610中所存储的程序和数据。

[0158] 其中,所述终端设备还包括收发机620,收发机620与总线接口连接,用于在处理器600的控制下接收和发送数据;处理器600用于读取存储器610中的程序。

[0159] 具体的,处理器600用于,获取用于处理AT指令的功能程序,以及创建用于调用AT指令的应用程序接口API;

[0160] 收发机620用于,通过所述API接收第二终端发送的第一AT指令;

[0161] 处理器600用于,基于所述功能程序,执行所述第一AT指令的相关处理。

[0162] 可选地,处理器600具体用于,根据NB-IoT通信模组的资源配置,生成软件开发工具包SDK;

[0163] 收发机620具体用于,通过所述SDK,获取所述用于处理AT指令的功能程序。

[0164] 可选地,处理器600具体用于,通过所述SDK,移植所述第一终端中的微控制单元MCU的所述功能程序;其中,所述NB-IoT通信模组属于所述第一终端;和/或,通过所述SDK,创建所述功能程序。

[0165] 可选地,处理器600具体还用于,接收对所述NB-IoT通信模组的图形配置界面的配

置操作;以及,响应于所述配置操作,确定所述NB-IoT通信模组的资源配置。

[0166] 可选地,处理器600具体用于,基于所述功能程序,对所述第一AT指令进行格式化处理,得到格式化处理后的第一AT指令;以及,根据所述格式化处理后的第一AT指令,执行所述第一AT指令的相关处理。

[0167] 可选地,处理器600具体用于,获取与所述第一AT指令对应的目标数据;

[0168] 收发机620具体用于,通过所述API向所述第二终端发送所述目标数据。

[0169] 可选地,收发机620具体用于,在所述NB-IoT通信模组处于标准模式的情况下,通过所述API将所述目标数据传输至第一物理串口,所述第一物理串口是第一终端与所述第二终端之间进行通信的串口;以及,在所述NB-IoT通信模组处于OpenCPU模式的情况下,通过所述API将所述目标数据传输至第一AT指令线程,所述第一AT指令线程是所述第一终端向所述第二终端发送所述AT指令的线程。

[0170] 可选地,处理器600具体用于,在所述NB-IoT通信模组连接日志工具的情况下,通过所述NB-IoT通信模组将所述第一AT指令和所述目标数据传输至所述日志工具;以及,通过所述日志工具对所述AT指令和所述目标数据进行存储。

[0171] 其中,在图6中,总线架构可以包括任意数量的互联的总线和桥,具体由处理器600代表的一个或多个处理器和存储器610代表的存储器的各种电路链接在一起。总线架构还可以将诸如外围设备、稳压器和功率管理电路等之类的各种其他电路链接在一起,这些都是本领域所公知的,因此,本文不再对其进行进一步描述。总线接口提供接口。收发机620可以是多个元件,即包括发送机和收发机,提供用于在传输介质上与各种其他装置通信的单元。针对不同的终端,用户接口630还可以是能够外接内接需要设备的接口,连接的设备包括但不限于小键盘、显示器、扬声器、麦克风、操纵杆等。处理器600负责管理总线架构和通常的处理,存储器610可以存储处理器600在执行操作时所使用的数据。

[0172] 本发明实施例还提供着一种可读存储介质,所述可读存储介质上存储程序或指令,所述程序或指令被处理器执行时实现如上任一项所述的窄带物联网NB-IoT通信模组的二次开发方法的步骤。

[0173] 此外,需要指出的是,在本发明的装置和方法中,显然,各部件或各步骤是可以分解和/或重新组合的。这些分解和/或重新组合应视为本发明的等效方案。并且,执行上述系列处理的步骤可以自然地按照说明的顺序按时间顺序执行,但是并不需要一定按照时间顺序执行,某些步骤可以并行或彼此独立地执行。对本领域的普通技术人员而言,能够理解本发明的方法和装置的全部或者任何步骤或者部件,可以在任何计算装置(包括处理器、存储介质等)或者计算装置的网络中,以硬件、固件、软件或者它们的组合加以实现,这是本领域普通技术人员在阅读了本发明的说明的情况下运用他们的基本编程技能就能实现的。

[0174] 因此,本发明的目的还可以通过在任何计算装置上运行一个程序或者一组程序来实现。所述计算装置可以是公知的通用装置。因此,本发明的目的也可以仅仅通过提供包含实现所述方法或者装置的程序代码的程序产品来实现。也就是说,这样的程序产品也构成本发明,并且存储有这样的程序产品的存储介质也构成本发明。显然,所述存储介质可以是任何公知的存储介质或者将来所开发出来的任何存储介质。还需要指出的是,在本发明的装置和方法中,显然,各部件或各步骤是可以分解和/或重新组合的。这些分解和/或重新组合应视为本发明的等效方案。并且,执行上述系列处理的步骤可以自然地按照说明的顺序

按时间顺序执行,但是并不需要一定按照时间顺序执行。某些步骤可以并行或彼此独立地执行。

[0175] 以上所述的是本发明的优选实施方式,应当指出对于本技术领域的普通人员来说,在不脱离本发明所述的原理前提下还可以作出若干改进和润饰,这些改进和润饰也在本发明的保护范围内。

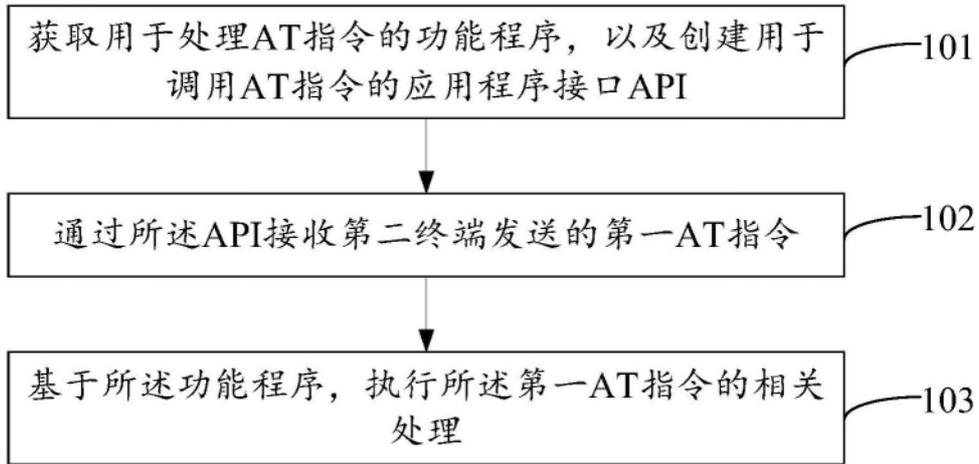


图1



图2

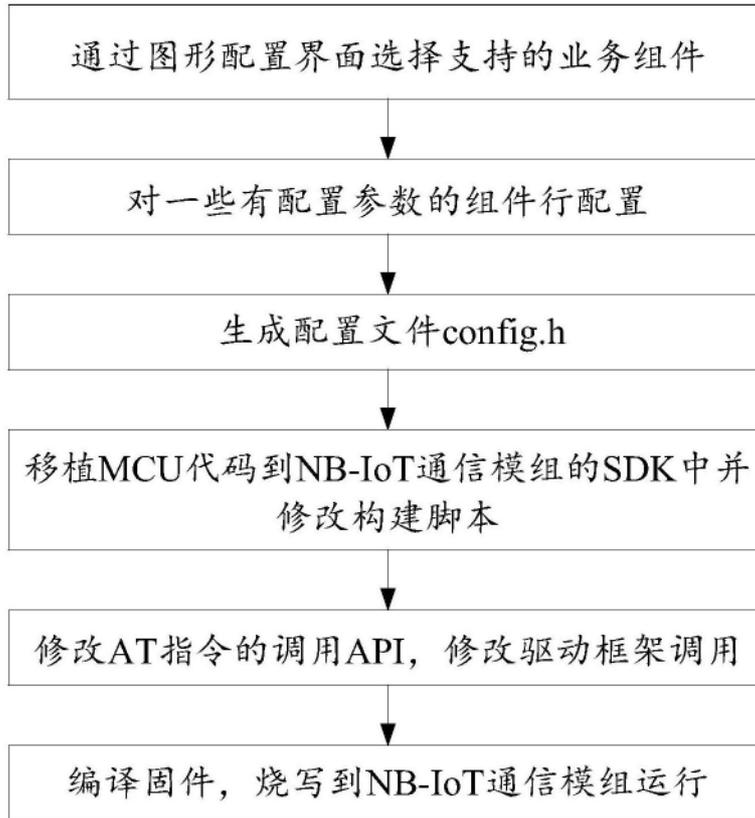


图3

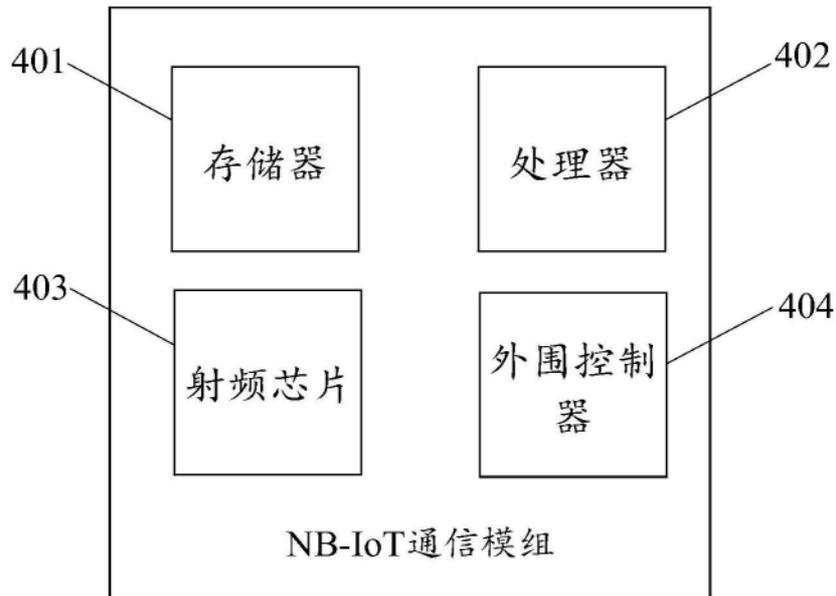


图4

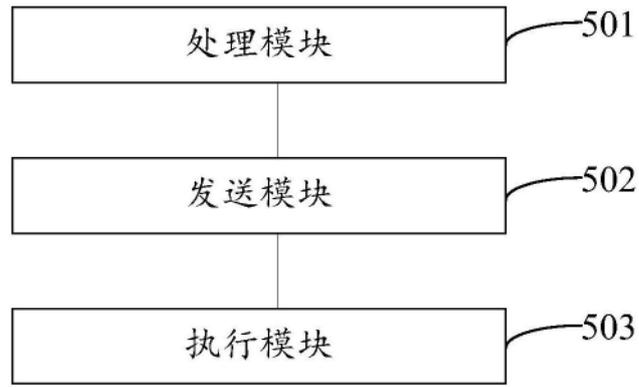


图5

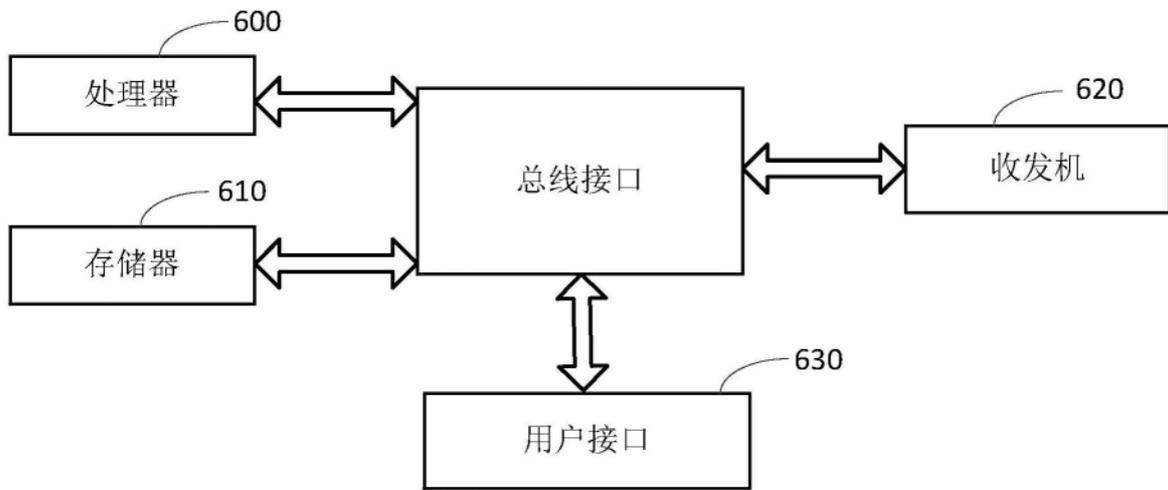


图6