BXP_S_SLCTHF 概要设计

版本	说明	修改人/时间	审核
V1.0.0	创建	易焰荣/20221024	

目录

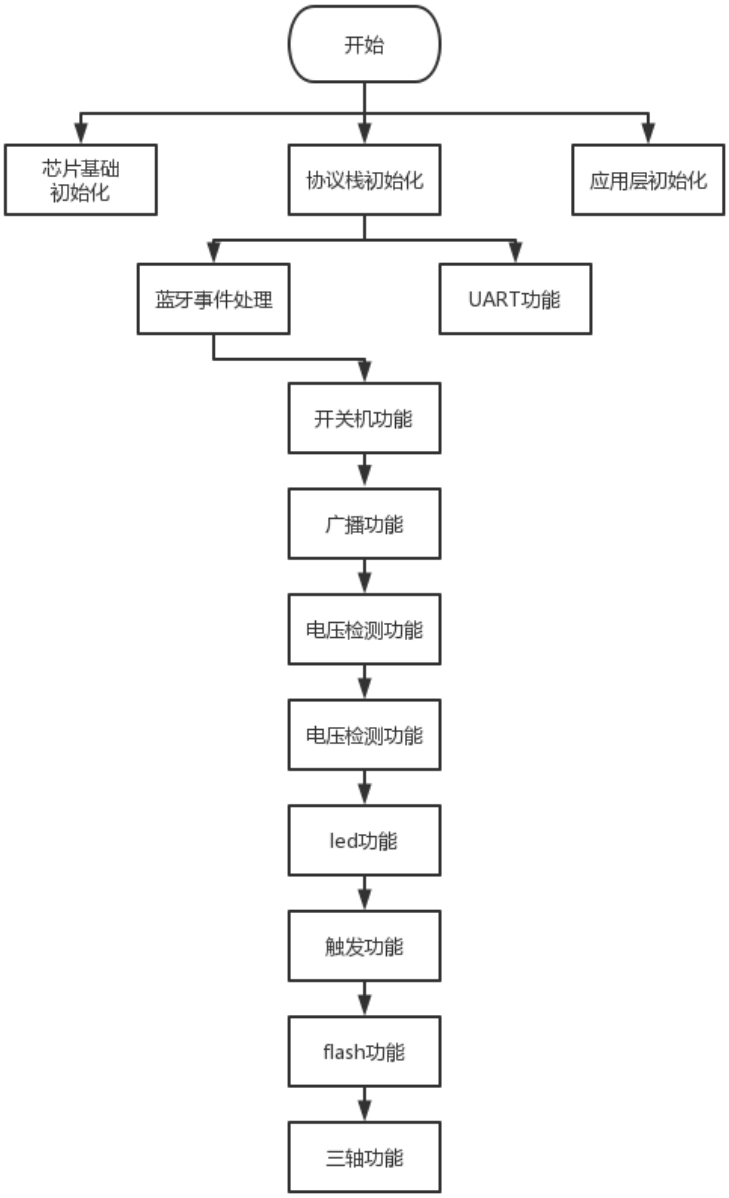
1. 整体功能结构说明

将整体功能分为 2 个部分：芯片系统功能和应用功能。

芯片系统功能为芯片可运行的最简系统功能。可低功耗实现芯片内部硬件的基本运行，可供软件工程师作为基础固件进行应用开发。

应用功能为 BXP_S_SLCTHF 特定需求功能的运行过程，下面会进行一一说明。

1.1. 系统框架流程



1.1.1. 芯片底层初始化

芯片底层指相关芯片内部的硬件模块。芯片上电后，为了能让芯片正常运行，需要对芯片底层硬件模块进行初始化。

此步骤包含：a. NVIC（中断控制器）初始化;；b. DCDC 模块初始化； c. 高低频晶振初始化；d. 系统时钟初始化；e. 低功耗系统初始化；f. NVM 初始化。

模块解释说明：

NVIC 中断控制器：用于系统软件复位，内部中断检测等功能实现。

DCDC 模块：用于输出稳定电压，输入为 VDD 电压，输出为 1.8V，可接入内部模电数电电路电源等。可一定程度地降低功耗。

NVM 存储器：用于存储应用数据，可掉电保留，一般存储用户参数配置以及出厂参数配置。

1.1.2. 协议栈初始化

MK16 应用需要蓝牙功能的实现，需要蓝牙协议栈接入芯片系统，蓝牙协议栈是蓝牙服务应用功能、蓝牙底层协议和芯片射频模块驱动的重要控制模块。

相关蓝牙功能配置可在..\config\btconf\gatt_configuration.btconf 文件进行设置

1.1.3. 应用层初始化

应用层初始化主要为应用功能驱动的创建和初始参数进行配置。包括应用层的**定时唤醒时钟初始化、用户数据的初始值配置和初始存储、看门狗初始配置、蓝牙广播数据初始配置、UART 产测驱动初始化、ADC 驱动初始化、按键识别驱动初始化、三轴芯片驱动、管理类功能定时器初始化（广播间隔管理，led 管理，三轴采样定时）**。

1.2. 应用层功能

1.2.1. Flash 数据管理功能

1.2.1.1. Flash 功能说明

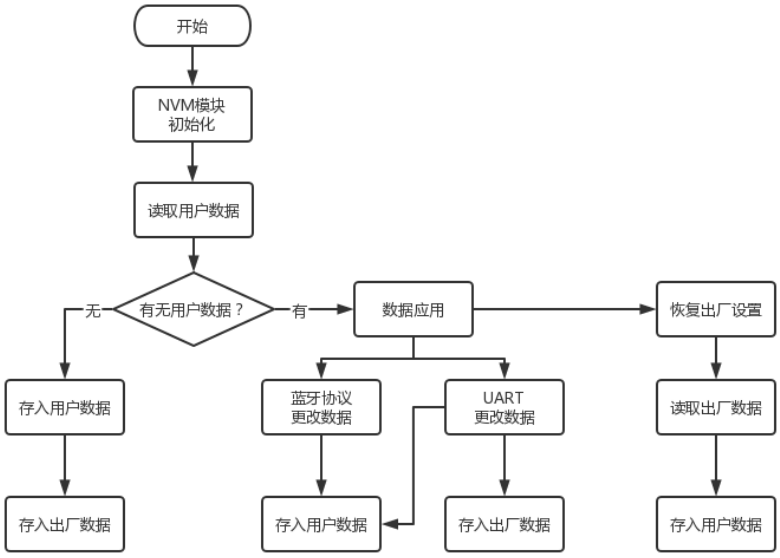
用户数据的参数用于控制需求功能，要求用户数据需要掉电仍处于保存状态。往常主要保存于 flash 中。在芯科芯片平台，有 NVM 模块可替代 flash，以 key 为单位，每个 key 可存 56 个字节数据。

存储的数据参数分用户数据与出厂数据。用户数据是当前应用的设置，通过蓝牙协议修改后，用户数据存入 NVM，同时修改当前应用的参数。

出厂数据为出厂时默认参数。出厂时，用户数据与出厂数据等同，均为默认参数值。出厂数据是通过 UART 协议进行修改的，也支持蓝牙协议命令进行存储出厂设置。

恢复出厂设置由蓝牙协议控制。恢复出厂设置的操作是将出厂数据替换掉当前用户数据，使得应用参数恢复到出厂时的状态。

1.2.1.2. Flash 功能流程图



1.2.1.3. Flash 功能设计注意项

A. 开关机记忆保存功能，需设置在开关机后延迟 3s 再存入 flash

B. 恢复出厂设置为协议恢复出厂设置，mac 和密码不恢复

C. NVM 默认放在 memory 末端，起始地址与 NVM 大小相关，这会影响到 OTA 升级包的大小

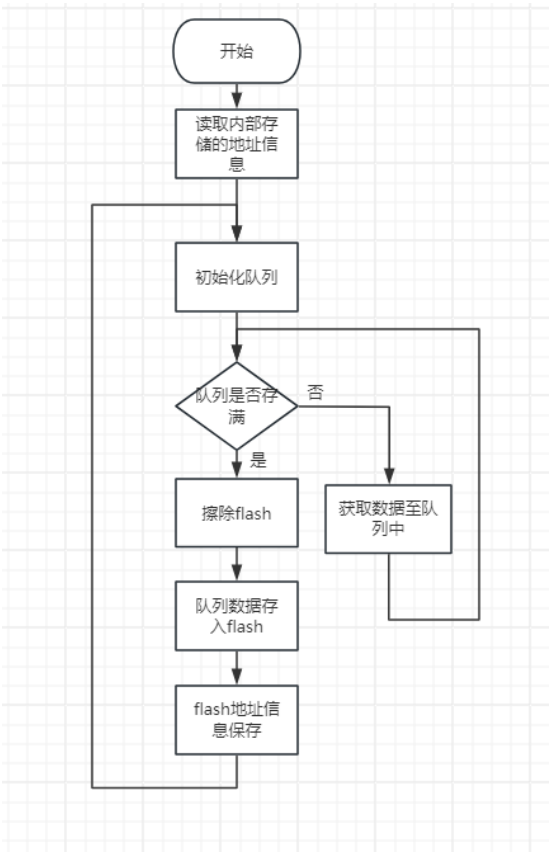
1.2.2. 外部 Flash 功能

芯片的内部 flash 可用来存储相对数据量较小的客户配置参数，而对于数据量很大的数据，则需要用到外部的 flash 进行存储。

同样的，对外部的 flash 进行擦除和写入操作也需要占用很大的功耗。所以程序中，会对数据使用队列进行缓存，达到写入要求（例如：数据量达到 1024byte），然后一次性进行写入。

在每次进行 flash 写入时，需要对 flash 进行擦除，然后再对 flash 进行写入，同时需要对当前 flash 写入的地址和可写入大小等具体参数进行存储，flash 地址等信息则存储于 MCU 芯片的内部 flash 中用于管理 flash。

1.2.2.1. 外部 Flash 功能流程图



1.2.3. 广播功能

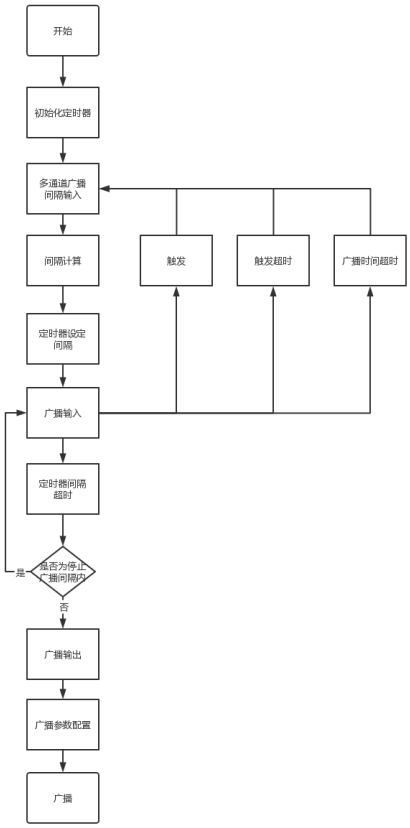
1.2.3.1. 广播流程说明

多帧广播，是通过错开一定时间间隔来实现多帧同时且满足单帧广播间隔的。按当前所需广播帧的最大公约数作为定时器间隔，以定时器间隔为单位，对多帧广播的起始时间依次增加 1 个单位，这样就可以错开所有广播帧实现多帧广播。定时器间隔的更新，只有触发、触发超时、（大）广播间隔超时、广播开启时会执行。

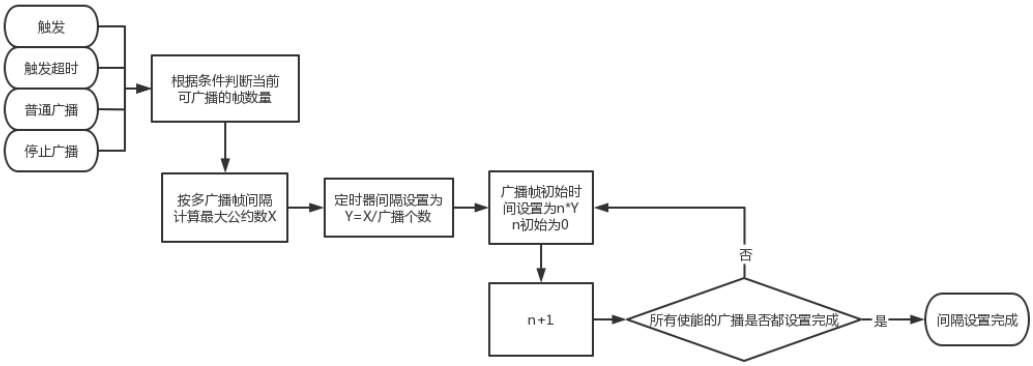
比如开启 4 个广播，第一个为 500ms 间隔，第二个为 1000ms 间隔，第 3 个为 2000ms 间隔，第 3 个为 3000ms 间隔，他们的最大公约数是 500，所以定时器间隔为 500/4=125ms，并且初始通道时间设定为：第一个广播：125*0ms，第二个广播：125*1=125ms，第三个广播：125*2=250ms，第四个广播 125*3=375ms。

之后每帧广播间的最小间隔必然保持 125ms 以上，实现错开广播。

1.2.3.2. 广播流程图



广播功能流程

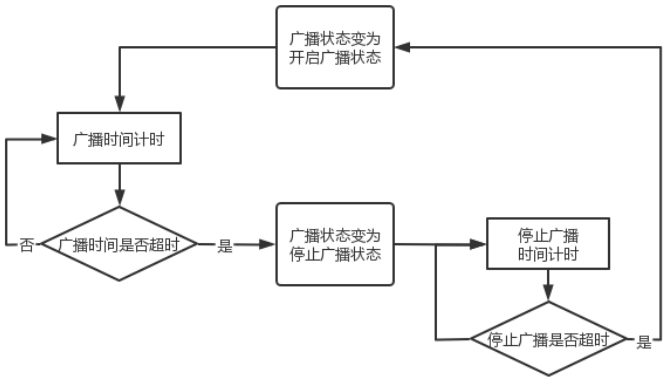


间隔计算及设置流程

1.2.3.3. 大小广播功能说明

按需求，每帧广播都有秒级的广播时间与停止广播时间，广播时间默认为 10s，停止广播时间默认为 0s，即一直广播，不会停止广播。当设置为 x 秒时，广播会按照广播间隔进行 10s 广播，再进入停止广播状态，计时 x 秒后再开启广播，以此循环。

1.2.3.4. 大小广播功能流程图



1.2.3.5. 广播设计注意项

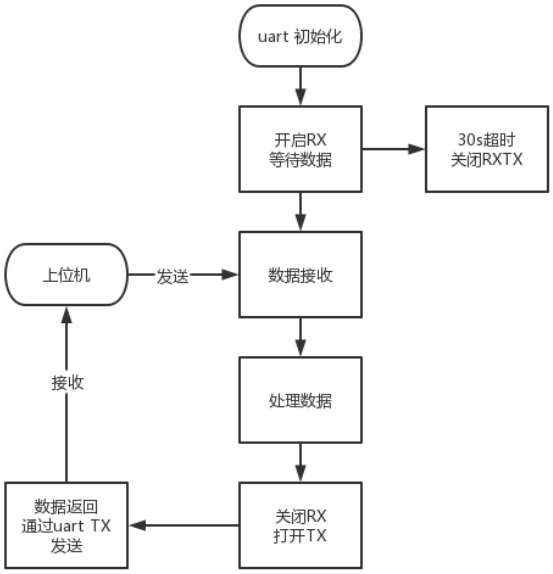
- 优化功耗时关闭自定义定时器需注意，广播间隔太大时导致计时间隔大，单开一个定时器可能会导致一些计时时间不准的情况（慢个几秒）功能需要保证符合需求

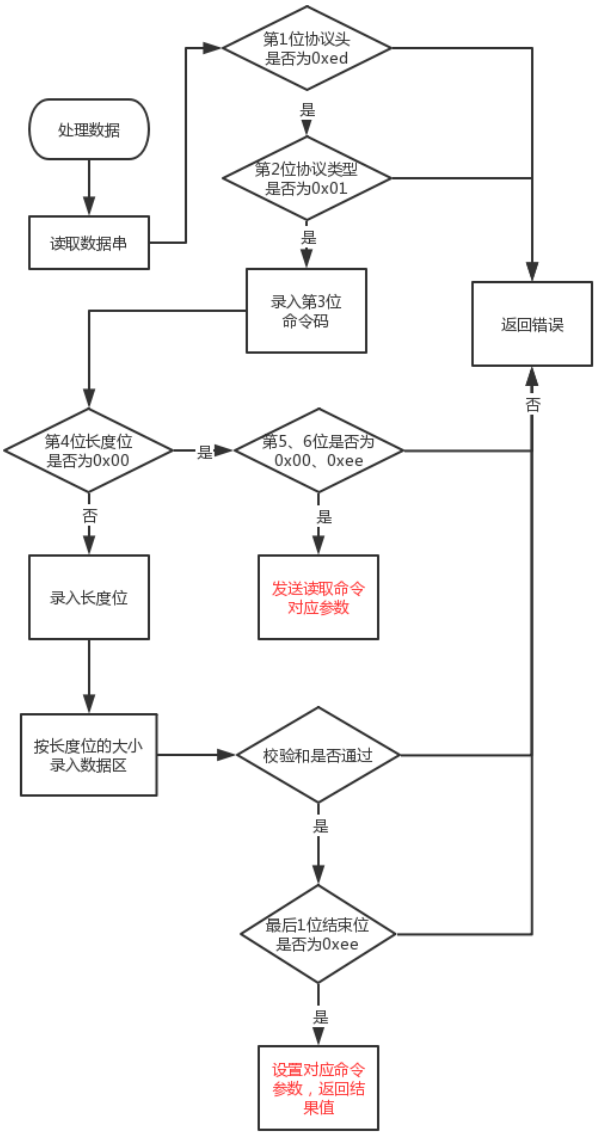
1.2.4. Uart 功能

1.2.4.1. UART 功能描述

- Uart 初始化，对对应 IO 初始化为 UART 接口，以及 DMA 存储初始化。当数据从 RX 输入时，DMA 暂存数据，并输出给应用程序进行处理，处理结束后根据命令，通过 TX 进行回复或回复错误给上位机。
- 数据格式处理如下图，格式按照 0xed（协议头）、0x01（类型）、cmd（命令）、0xXX（长度）、数据区、校验和、0xee（结束码）组成，其处理过程如下图

1.2.4.2. UART 功能流程图



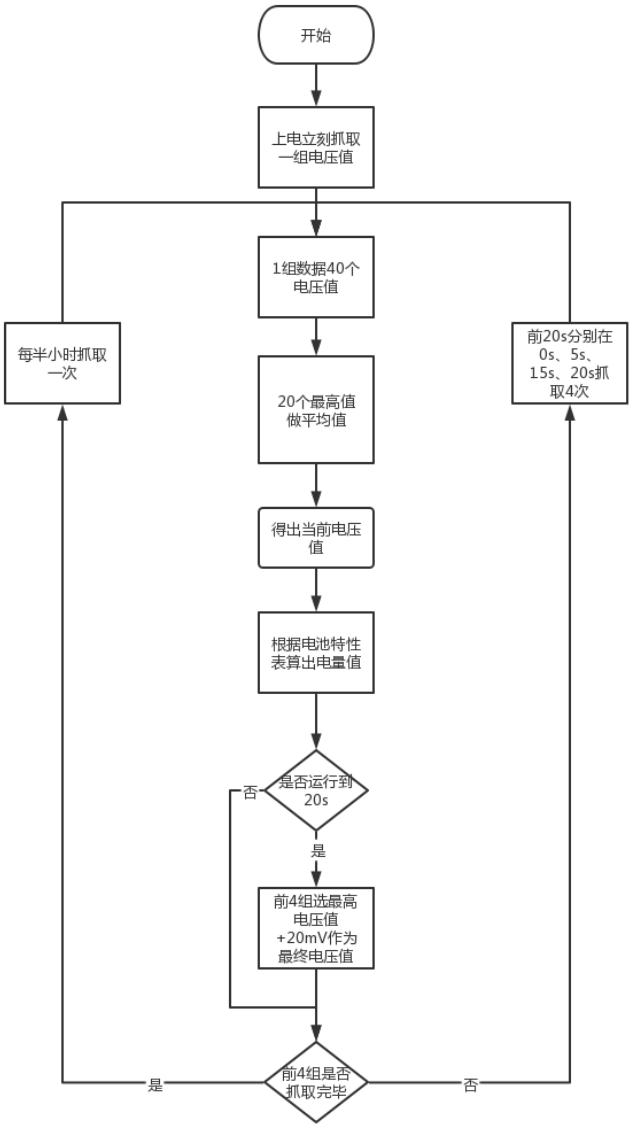

处理数据
读取数据串
第1位协议头
是否为0xed
是
第2位协议类型
是否为0x01
是
录入第3位
命令码
返回错误
否
第4位长度位
是否为0x00
是
第5、6位是否为
0x00、0xee
否
是
录入长度位
发送读取命令
对应参数
按长度位的大小
录入数据区
校验和是否通过
是
最后1位结束位
是否为0xee
是
设置对应命令
参数，返回结
果值

1.2.5. 电量读取任务

1.2.5.1. 电量检测功能描述

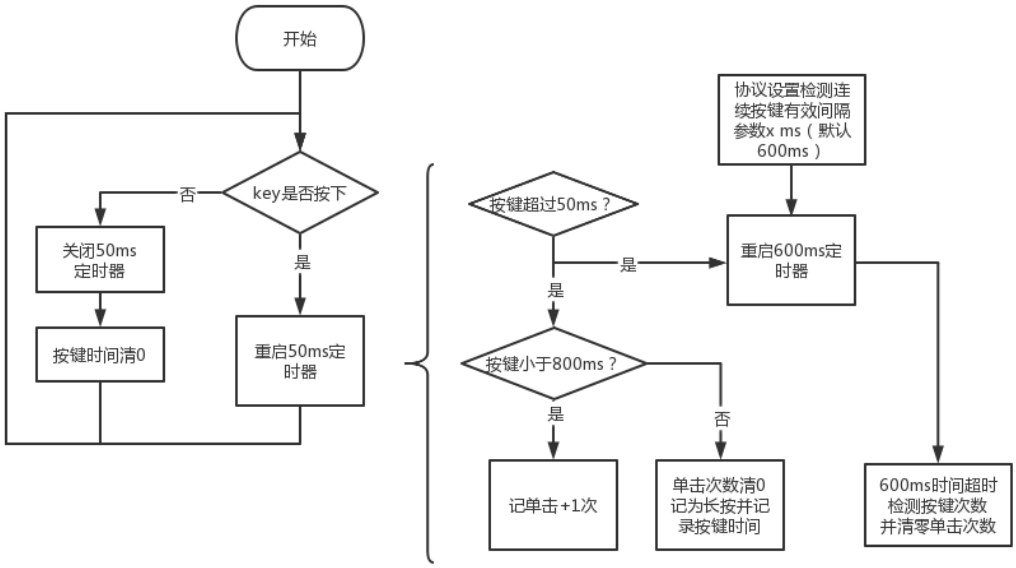
上电后，会先抓取 1 组 40 个电压值进行计算，往后为 1 小时取 1 组。
低电压判断标准为连续 10s 保持在 5%电量以下。

1.2.5.2. 电量检测流程



1.2.6. 按键管理（霍尔）

按键机制如图所示：



按键管理主要由一个 50ms 定时器作为基础，来判定按键的按下时间。而按键操作的结束由一个 600ms 单次计时来判定。通过 2 个定时器的配合工作判定出按键操作是单击、双击、多击、长按以及其他特殊按键操作定义。

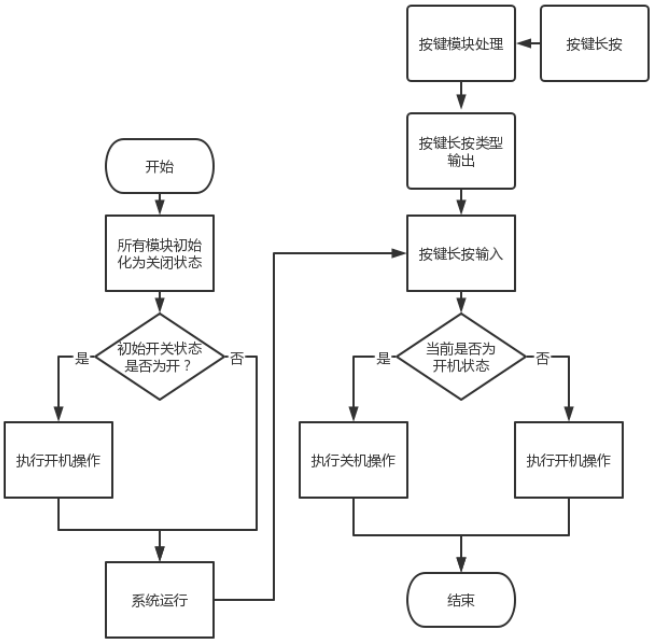
当按键按下时，50ms 定时器开启并开始计时，当按键松开时 50ms 定时器关闭，600ms 定时器开启。

- 单击：按下，超过 50ms 瞬间按键次数+1，松开，600ms 超时，输出按键次数为 1 次
- 多击：按下，超过 50ms 瞬间按键次数+1，松开，600ms 未超时，再按下，循环 n 次，直到按键结束，600ms 超时，输出按键次数 n 次
- 长按：按下，超过 800ms，识别为长按，按键次数清 0

600ms 定时器可协议修改间隔，默认 600ms，即按键次数的可累加条件是任意两次按键按下的时刻之间的时间间隔不得超过 600ms

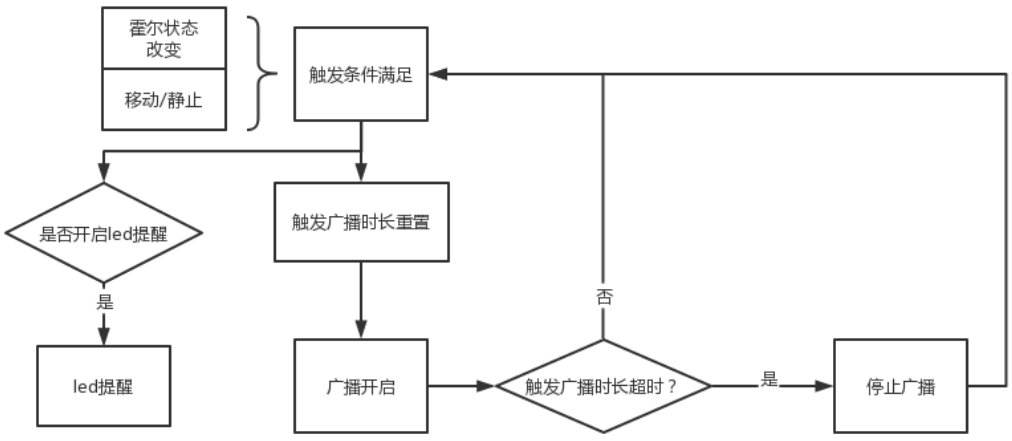
1.2.7. 开关机管理

根据按键管理输出的长按状态，以及当前的开关机状态，来判断执行开关机操作。流程如下：

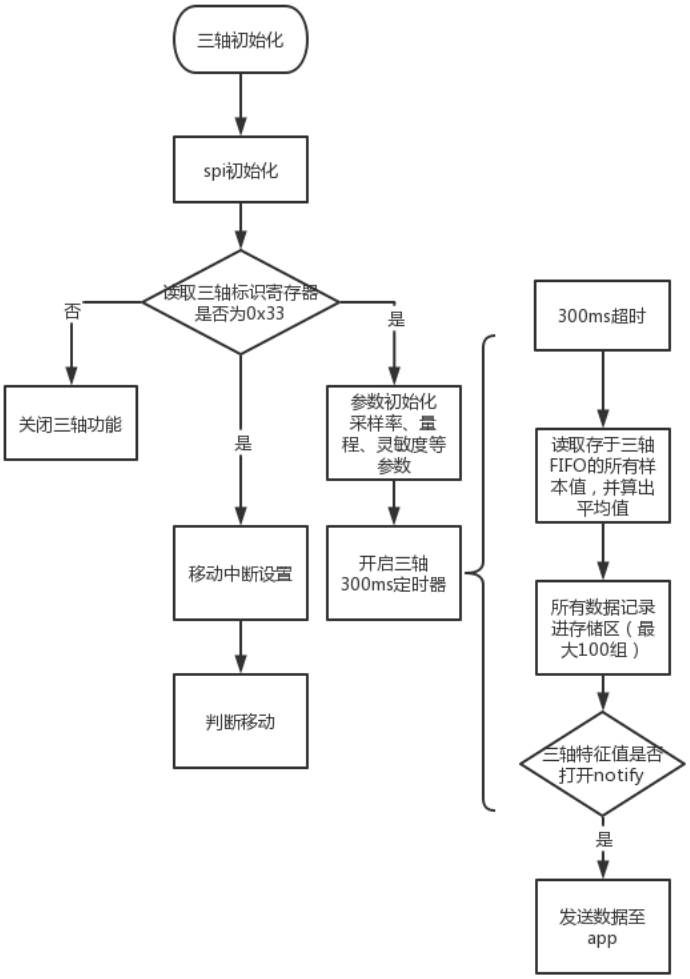


1.2.8. 触发管理

触发机制默认关闭，通过协议更改可设置触发模式，设置完成后即开始判断触发条件是否满足（连接状态下不会触发，等待断开后才触发）。在触发条件未满足时，该广播帧按照设置开启/关闭广播。触发条件满足瞬间，判断 led 是否开启提醒，同时广播按照设置关闭/开启广播，同时触发广播时长超时后重新开启/关闭广播，等待下一次广播。流程如下：

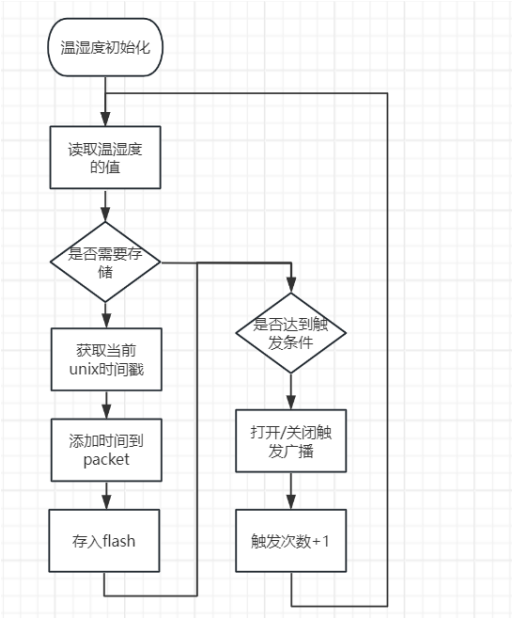


1.2.9. 三轴驱动



- 三轴驱动主要分三轴芯片初始化、移动中断与三轴数据抓取这 3 个步骤。
- 移动中断由芯片计算输出，MCU 无需计算，只需要打开相应的中断输入引脚进行判断即可
- 三轴数据为每 300ms 抓取一次，每次抓取的数量由三轴采样率决定。
- 抓取到的三轴可通过蓝牙服务特征传输曝光

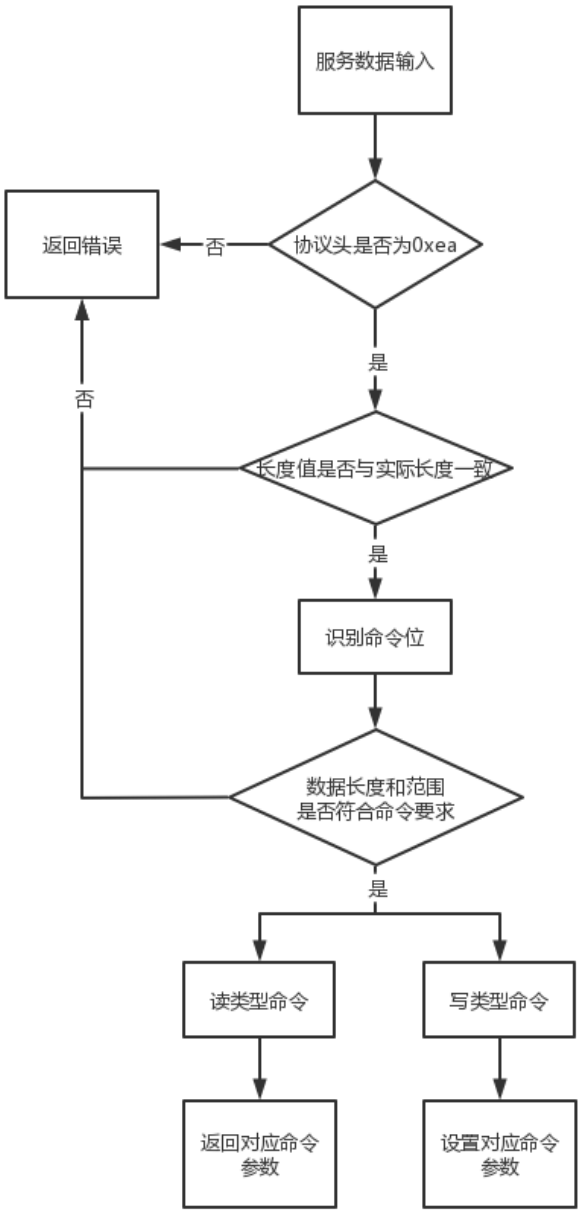
1.2.10. 温湿度驱动



本次设备主要通过温湿度传感器进行温度和湿度的检测，并且可以和当前的时间组成一条数据，存入外部的 flash 中。其中的时间以秒为单位的 Unix 时间轴，须开机进行同步。

通过设置触发广播的温湿度，以控制广播在特定的温度和湿度下，进行特定的广播，或者关闭广播。并且触发次数进行记录加 1。

1.2.11. 蓝牙服务协议

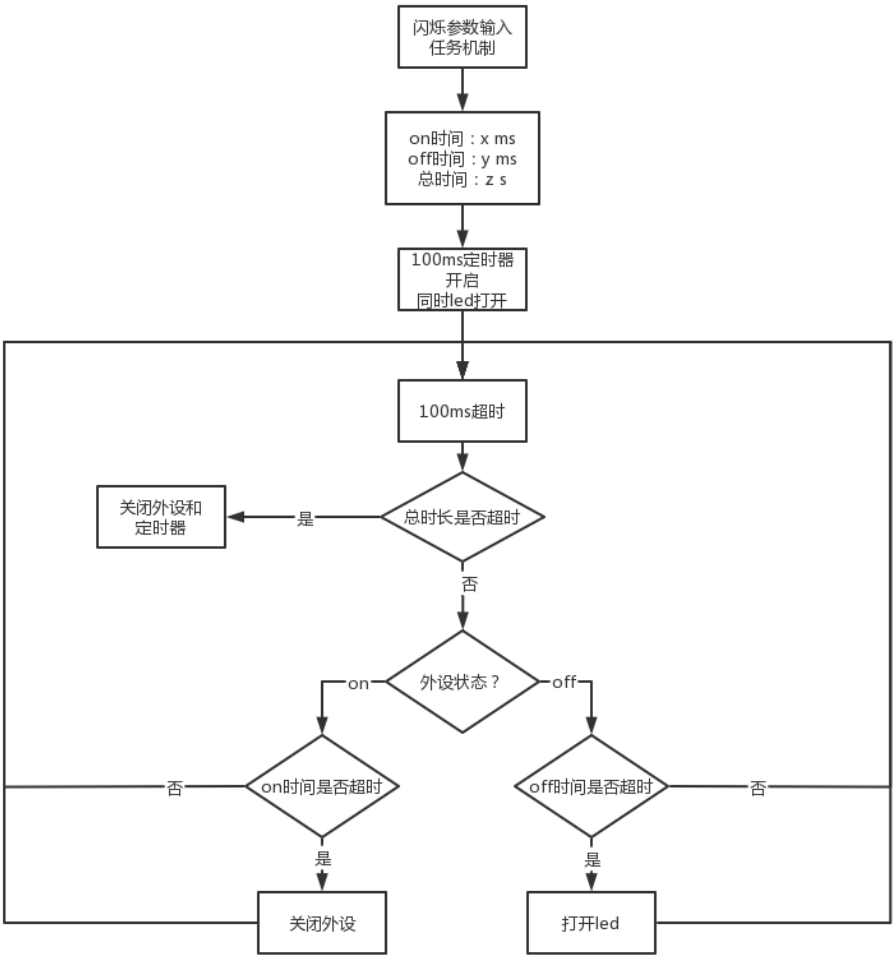


- 蓝牙协议根据协议格式进行判断命令是否有效，格式如下：

格式类型	功能
HEAD	命令头，主->从 0xEA 从->主 0xEB
FLAG	命令类型，0x00 为读命令，0x01 为写命令，0x02 为主动回复命令
CMD	1 字节命令
LEN	1 字节数据长度
DATA	数据，若长度为 0，则此字节为空

1.2.12. Led 闪烁机制

led 闪烁定时器最小间隔为 100ms，当 led 任务下发时，会输入打开时间，关闭时间和闪烁总时间。初始状态为打开 led，从打开时间开始计时，打开时间超时后，关闭 led，计时关闭时间，同样超时后打开 led，以此循环，直至闪烁总时长超时，流程如下图：



2. 应用层整体流程

2.1. 定时器的使用与优化功耗

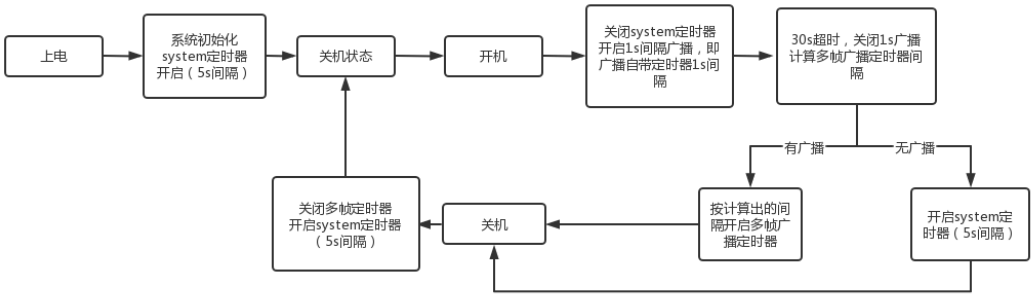
为防止设备出现严重死机的情况发生，固件会开启看门狗以确保固件异常后可执行软件复位进行重启设备。看门狗需要时刻进行投喂，来让看门狗识别出固件正在正常执行中，若长时间未投喂而导致超时，看门狗会认为固件已异常，会立马执行软件复位。因此，在固件设计过程中，保证固件可时刻进行投喂看门狗是非常重要的。一般使用的时定时器来定时投喂看门狗。

由于功耗的优化需求，固件需要减少定时器多开的情况，所以在运行过程中会尽量保持只有一个定时器在运行（除特殊应用需开多个除外）。下面对 BXP-T 固件对于定时器的开关逻辑进行描述说明。

2.1.1. 常开-定时器类型

常开定时器类型有 3 种，广播自带定时器、多帧广播定时器和系统定时器，主要通过这 3 个定时器的切换来实现功耗优化的目的。优化的同时，固件运行过程中其他相关的定时功能也需要保证正常运行。应用影响包括：开机前 30s 广播的 30s 计时、串口关闭的 30s 计时、flash 延时写入计时、触发时间计时、看门狗投喂、电压读取时间计时、密码超时。

定时器切换流程图如下：



从上电后，每个环节会保证有一个定时器的运行。至于时间间隔的运算，在 system time 任务中添加了 RTC 计算时钟功能，可准确计算与上次唤醒的间隔时间，累加至大于 1s 后，输出 tick 供应用计时功能使用