设备底层软件说明书

设备底层软件的设计按照执行流程划分，可以分为如下几个部分：

1. 初始化阶段；
2. 中断任务；
3. 背景任务：
   1. 定时任务；（仅在设备开机后执行）
   2. 循环任务。（无论设备开机或关机，均执行）

按照上述规划，将每个任务进行细致的说明，以及流程图的示意。

* 初始化任务
  1. 系统时钟

设置为内部时钟源，高频16MHz。

* 1. 看门狗

软件控制开启，定时8s。

* 1. GPIO

除了CCP、UART、AN0外的所有引脚全部设置为输入或输出。

* 1. 定时器2

定时4ms。

* 1. ADC及SOC初始化

AN0配置为模拟通道，对内部FVR进行初始化2.048V，使用内部FVR输出作为电压参考，1MHz转换时钟。连续采样512次电池电压，根据平均值计算电池初始SOC值。

* 1. UART

波特率为19200，异步通信模式。

* 1. PWM

CCP1/CCP2作为PWM模式输出，载波250Hz。

* 1. NRF

使用GPIO模拟SPI接口，配置以下内容：发送地址、接收地址、自动应答、接收通道使能、自动重发、接收数据宽度、信道100、发射功率-18dB、掉电模式。

* 1. 指示灯

所有指示灯全部打开。

* 1. 光耦发射端控制

光耦发射关闭。

* 1. 复用器地址线初始化

全部初始化为0。

* 1. 延时

延时500ms。

* 1. 中断

打开外设中断PIE和全局中断GIE。

* 1. 打开定时器

打开定时器TIMER2，进入背景任务中。

* 中断任务
  1. RCIF接收中断

处理UART接收字符。

如果设备已打开，且没有需要处理的应答任务（rx\_pack\_flag为零），则将接收到的字符保存至接收缓冲区内。

* 1. TIMER2定时中断

首先，建立时基，包括20ms、100ms、1s三个时间基准。然后检查接收缓冲区是否接收到完整的数据包（数据匹配）：

* 当前定时周期内无接收任何字符，将缓冲区复位；
* 接收长度大于3后，开始判断，如果长度匹配，则将rx\_pack\_flag置位，并在背景20ms任务中准备应答；如果长度超过，则将缓冲区复位（同时长度异常数据包递增），如果长度未达到指定长度，则继续接收，不做任何处理。
* 背景循环任务

背景循环任务一直在死循环中不断执行，无论设备开机还是关机。主要包括三个任务：

* 1. 按键扫描任务



按键扫描任务主要处理设备的开机与关机过程，在关机过程中，需要将所有变量全部复位，并关闭蓝牙和无线，然后进入31kHz运行。在开机过程中，将蓝牙打开，并将无线置为搜索态，同时，进入16MHz运行。

* 1. 电池SOC计算任务

SOC的计算在背景函数中不断循环执行，计算方法：

* 每采集512次后，计算一次SOC；
* 求512次的平均值；
* 根据V-SOC表，获取理论SOC值；
* 根据当前充放电状态，和过去的SOC值，更新当前的SOC。
  1. 电池指示灯任务

在背景任务中循环执行，方式如下：

* 5V电源未接入
  + 将充电绿灯熄灭；
  + 如果当前SOC<15%，红灯闪烁；否则红灯熄灭。
* 5V电源接入
  + 将红灯熄灭；
  + 如果当前正在充电，绿灯闪烁；否则，绿灯长亮（充满）。
* 背景定时20ms任务

主要执行三项任务，依次为：动作检测、NRF无线状态查询、蓝牙数据包应答。现分别描述如下：

* 1. 动作检测

读取SGL0~SGl5的数值，为1标明插入，否则，无插入。那么，当前的位置可以写为从0开始的连续的1的个数（遇0则结束）。遇到用户插入后，将自关机定时器复位。计算方法如下：

* 当前位置大于过去位置，判定反向为插入（INSERT）；如果小于，判定方向为拔出（EXTRACT），否则判定为不动（STILL）；
* 计算累计移动长度：L+=ABS(pos-pos\_1)；
* 定义过程标记：如果方向为INSERT则为+1，如果方向为EXTRACT则为0，如果方向为STILL则不变；
* 计算累计抽插次数：如果当前标记为0，而上次标记为+1，则将抽查次数递增，并将上次的位置记为当前抽插深度（瞬时深度）。
  1. NRF无线状态查询

查询NRF的状态，如果为TX，标明发送成功，立即转为接收模式；如果为RX，标明接收成功，立即读取数据；如果为MAX\_TX，标明发送失败。

* 1. 蓝牙数据包应答

蓝牙数据包应答分为三个阶段：

* 数据包检查阶段
* 数据包解析阶段
* 数据包发送阶段

首先给出数据包检查阶段的流程图，如下：



数据包检查阶段主要对控制字、校验码、操作码进行检查，如果出现非法字符，则对相应的统计数据包递增，并返回。如果所有检查均合法，则进入数据包解析阶段，在数据包解析阶段，进行应用数据的处理，和应答数据包的数据准备。

数据包解析按照统一的协议来处理，即：app发送数据包后，由设备给出应答。目前设备底层软件支持10种命令，分别描述如下：

1. 设备信息查询

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 4 | 0x40 | 0xE0 | NULL | CRC |

手机app端下发的数据包

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 14 | 0x50 | 0xE0 | “MC47A8D013” | CRC |

设备端的应答数据包

该命令在设备连接前需要执行一次，app可在任意需要的时刻通过该命令获取设备信息。设备应答数据包长度为14，信息包含10个字节。

MC---男用飞机杯代码；

47A8D0---产品编号；

13---产品软件版本V1.3

1. 连接

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 4 | 0x40 | 0xC0 | NULL | CRC |

手机app端下发的数据包

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 5 | 0x50 | 0xC0 | 0x00 | CRC |

设备端的应答数据包

该命令用于连接设备，应答返回数据为0x00.

1. 断开连接

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 4 | 0x40 | 0xC1 | NULL | CRC |

手机app端下发的数据包

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 5 | 0x50 | 0xC1 | 0x00 | CRC |

设备端的应答数据包

该命令用于断开连接，应答数据返回0x00.

1. 刷新

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 4 | 0x40 | 0xC2 | NULL | CRC |

手机app端下发的数据包

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 5 | 0x50 | 0xC2 | 0x00 | CRC |

设备端的应答数据包

该命令用于维持设备的连接过程，目前该命令处于无用状态，因为app还可以通过定时查询设备状态来维持连接。但是，还是保留了这个命令，但不做任何处理。

应答数据包返回数据位0x00.

1. 设备设置命令

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 5 | 0x40 | 0xA0 | 设定字节 | CRC |

手机app端下发的数据包

该命令包含一个字节，用于对设备的功能进行设置。每一位对应的含义为：

|  |  |
| --- | --- |
| Bit7 | 游戏模式设置。  1：设备游戏模式开启；  0：设备游戏模式关闭。 |
| Bit6 | 动作过程结束。  1：动作过程由app端主动结束；  0：无意义。 |
| Bit5 | 跳蛋1使能。  1：跳蛋1使能；  0：跳蛋1关闭。 |
| Bit4 | 跳蛋2使能。  1：跳蛋2使能；  0：跳蛋2关闭。 |
| Bit3~Bit1 | 保留。 |
| Bit0 | 底座使能。  1：底座使能；  0：底座关闭。 |

如果接收到的数据包长度为5，则应答数据包返回0x00，否则，返回0xFF。其余返回数据无效。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 5 | 0x50 | 0xA0 | 0x00或0xFF | CRC |

设备端的应答数据包

1. 跳蛋设置命令

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 6 | 0x40 | 0xA1 | 跳蛋1控制字  跳蛋2控制字 | CRC |

该命令用于对跳蛋的功能进行设置。其包含两个字节，第一个字节为跳蛋1的控制字，第二个字节为跳蛋2的控制字。两者含义相同，现对控制字节的位定义描述如下：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 控制模式 | | | | 作用时间 | | | |

解释如下：

|  |  |
| --- | --- |
| Bit7-Bit4 | 控制模式，目前暂时定义为强度，依次从0~15，共16种强度。  0x0~0xF：定义了16种强度。 |
| Bit3-Bit0 | 作用时间，以秒为单位，目前定义了16种时间长度。  0x0~0xE：作用时间分别为0~14秒；  0xF：作用时间长期有效。 |

注意，该功能仅在对应跳蛋的功能使能后方有效。

如果接收数据包长度为6，应答数据包返回0x00，否则，返回0xFF，其它应答数据无效。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 5 | 0x50 | 0xA1 | 0x00或0xFF | CRC |

设备端的应答数据包

1. 底座设置命令

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 8 | 0x40 | 0xA2 | 运行模式  频率  保留字节1  保留字节2 | CRC |

该命令用于对底座功能进行设置，包含四个字节，前两个字节依次为运行模式和参考频率，后两个字节保留。该命令仅在底座功能使能后方有效。

如果应答数据包接收数据长度为8，应答数据位0x00，否则为0xFF，其余数据无效。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 5 | 0x50 | 0xA2 | 0x00或0xFF | CRC |

设备端的应答数据包

1. SOCPACK查询（soc+数据包查询）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 4 | 0x40 | 0xD0 | NULL | CRC |

该命令用于对电池SOC和数据包统计参数进行查询，设备的应答为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 应用数据 | 校验码 |
| 10 | 0x50 | 0xD0 | SOC  接收数据包  长度错误包  校验错误包  控制错误包  操作错误包 | CRC |

1. 设备状态查询

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 4 | 0x40 | 0xD1 | NULL | CRC |

该命令用于对设备和底座的状态进行查询，设备应答数据位2个字节。如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 6 | 0x50 | 0xD1 | 设备状态  底座状态 | CRC |

返回的设备状态字节定义如下：

|  |  |
| --- | --- |
| Bit7 | 设备运行状态。  1：设备开机运行中；  0：设备已关闭。 |
| Bit6 | 设备连接状态。  1：设备与app处于连接状态；  0：设备与app处于断开状态。 |
| Bit5 | 动作过程标记。  1：动作过程已开始；（app可以开始使用动作数据查询命令获取设备的动作数据）  0：动作过程已结束。 |
| Bit4 | 设备游戏运行模式。  1：设备处于游戏运行模式；  0：设备处于用户操作模式。 |
| Bit3 | 跳蛋1运行状态。  1：跳蛋1正在运行；  0：跳蛋1已关闭。 |
| Bit2 | 跳蛋2运行状态。  1：跳蛋2正在运行；  0：跳蛋2已关闭。 |
| Bit1 | 设备充电状态。  1：设备正在冲断；  0：设备未充电（未接入电源或充电已完成）。 |
| Bit0 | 5V电源接入状态。  1：5V电源已接入；  0：5V电源未接入。 |

返回的底座状态字节定义如下：

|  |  |
| --- | --- |
| Bit7 | 底座运行状态。  1：底座正在运行；  0：底座运行暂停。 |
| Bit6 | 底座无线连接状态。  1：底座与设备完成连接；  0：底座与设备未连接。 |
| Bit5 | 底座操作模式。  1：底座处于游戏模式；（app操作）  0：底座处于用户操作模式。（用户手动） |
| Bit4 | 设备紧固状态。  1：设备已紧固；  0：设备未紧固。 |
| Bit3-Bit0 | 保留 |

底座状态必须要高4位全部为1方能由app对其进行操作。

1. 动作数据查询

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 4 | 0x40 | 0xD2 | NULL | CRC |

手机app端下发的数据包

该命令用于获取设备底层的动作数据。设备的应答数据为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据包字节长度 | 控制字 | 操作符 | 数据区 | 校验码 |
| 12 | 0x50 | 0xD2 | 时间点数（H：L）  瞬时位置/瞬时方向  最近一次抽插深度  累计抽插次数（H：L）  累计抽插长度（H：L） | CRC |

应答数据包含有8个字节，依次为时间信息的高、低字节，瞬时位置/方向字节、瞬时深度字节、累计抽插次数的高、低字节，累计抽插长度的高、低字节。解释如下：

* 时间信息

包含有高、低两个字节，格式如下：

|  |  |
| --- | --- |
| 时间：高8位 | 时间：低8位 |

时间信息的单位为100ms，时间从0~65535，即0~109分钟。

* 瞬时位置/方向

|  |  |
| --- | --- |
| 瞬时位置：高4位 | 方向：低4位 |

该字节的高4位为瞬时位置，0~6。

该字节的低4位为瞬时方向，其值具有三种状态，分别为：

|  |  |  |
| --- | --- | --- |
| STILL | 0 | 保持不动 |
| INSERT | 1 | 正在插入 |
| EXTRACT | 2 | 正在拔出 |

* 瞬时深度

|  |
| --- |
| 瞬时深度：0~6 |

该字节为瞬时深度信息，范围0~6。

* 累计抽插次数

|  |  |
| --- | --- |
| 累计抽插次数：高8位 | 累计抽插次数：低8位 |

累计抽插次数包含2个字节，依次为高8位字节和低8位字节。

* 累计抽插长度

|  |  |
| --- | --- |
| 累计抽插长度：高8位 | 累计抽插长度：低8位 |

累计抽插长度包含2个字节，依次为高8位字节和低8位字节。

数据包的发送比较简单，将准备发送的数据，包含：

1. 待发送数据包长度；
2. 控制字0x50；
3. 操作符；
4. 数据区复制；
5. 计算校验码；
6. UART发送；
7. 结束。

* 背景定时100ms任务

背景定时100ms任务主要包括：

* 1. 状态刷新；

状态刷新主要是根据运行中的各个部件状态，对设备的整体状态字节进行更新。

* 1. 运行指示灯控制

包含不同用户操作模式、连接/断开状态、游戏模式等进行指示灯的不同颜色、闪烁频率。

* 1. NRF状态机

NRF状态机计算最近16次发送的结果。根据结果对NRF状态进行更新。

* + 搜索态：如果成功次数不低于15次，状态变为连接态；如果连续200次处于搜索态，则变为掉电态。
  + 连接态：如果成功次数不超过3次，状态变为搜索态。
  + 掉电态：无任何操作。设备开机状态时，用户通过按键可以使NRF重新处于搜索态。
  1. 跳蛋控制

跳蛋的控制基于如下两种模式：

* 游戏模式：按照app发送的设置命令对其进行控制，包括强度、作用时间。作用时间结束后，跳蛋自动关闭。App如果需要激活跳蛋，则使用跳蛋的设定命令刷新一次即可。
* 用户模式：用户通过按键切换模式，支持8种基本模式。

跳蛋1和跳蛋2的状态字会根据运行情况实时更新。

* 背景定时1s任务

背景定时1s任务主要包括：

* 1. 动作过程判断

动作过程判断主要处理：

* 动作过程什么时候开始？

该任务会计算连续2秒内的抽插距离总长度，当发现该长度超过3后，立即激活动作过程，将动作过程标记置位。此时，对如下变量进行初始化：

1. 动作过程时间信息清零；
2. 动作过程累计抽插次数清零；
3. 动作过程累计抽插长度清零；
4. 动作过程禁止结束标记有效；（在2min内不进行结束判断）
5. 将2min内的抽插长度数组全部清零；
6. 将2min内抽插总长度清零；
7. 将2min内的抽插长度数组索引号清零。

经过上述初始化后，所有与过程信息有关的变量全部被复位，动作过程所有数据信息开始记录。设备通过该标记（设备状态标志的Bit5）通知app，可以使用动作查询命令获取动作信息。

* 动作过程什么时候结束？

当动作过程开始后，2min内不进行结束判断，2min后，禁止结束标记被释放，此时任务开始判断动作过程是否结束。

任务通过统计2min内的抽插移动总长度来进行。

如果2min内抽插移动总长度低于10，立即将动作过程结束，将动作过程标记清零。

动作过程支持两种方式结束：

1. 用户通过app发送设置命令，将设备设置命令的Bit6位写1，则可以立即将动作过程结束。通过该方式，app将立即对所有数据进行分析，认为当前动作过程已结束。
2. 用户不再使用设备后，经过2min后，设备自动结束，并通过设备状态标志位的Bit5位通知app动作过程已结束，app将立即对所有数据进行分析，认为当前动作过程已结束。
   1. 蓝牙失联判断

设备接收到有效数据包后，会将蓝牙失联定时器复位。

当该定时器溢出后（5s），该任务自动将蓝牙状态变更为断开状态。

* 1. 自动关机

设备插入，或者用户按键后，会将该自动关机定时器复位。

当该定时器溢出后（15min），该任务自动将设备关机。