

百度智能手环 ROM 设计

模块名称	百度智能手环 ROM
所属系统	
模块负责人	郑亚军
项目负责人	孙鹤飞
文档提交日期	2014 年 8 月
版本号	1.0

百度在线网络技术（北京）有限公司

目录

1. 智能手环开源介绍	3
2. 开发环境	3
3. 手环 ROM 介绍	3
3.1 正常模式	3
3.2 升级模式（OTA 模式）	4
4. 交互设计	6
4.1 交互设计原则(改交互设计，需要先阅读本节内容)	6
5. 功能自定义	6
5.1 硬件相关修改	6
5.2 OTA 的客制化	7
6. ROM 打包与烧写	7
6.1 打包步骤如下:	7
6.2 烧写步骤	7

1. 智能手环开源介绍

智能手环基于 Nordic 公司 nrf51822 蓝牙 soc 芯片开发，使用 LIS3DH 作为加速度传感器，进行运动和睡眠监测。

开源遵循 Apache License, Version 2.0, 详情参见：

<http://www.apache.org/licenses/LICENSE-2.0>

2. 开发环境

操作系统：Windows XP, windows7

开发工具：Keil uVision4.0

Nordic 官方网站：<https://www.nordicsemi.com/>

Nordic SDK 版本：4.4.2

SoftDevice 版本：5.2.1

仿真器：J-Link

开发环境的搭建参考 nRF51822_Developer_Kit_User_Guide

3. 手环 ROM 介绍

智能手环有两种工作模式：正常模式和升级模式。

正常模式下，手环提供产品定义中的所有功能；

升级模式下，手环等待手机端把要升级的 ROM 通过蓝牙发过来，然后进行升级。空中升级模式（Over-the-Air）给手环的 bug 修复、产品优化提供了实现的途径，这也是智能设备必备的功能。

3.1 正常模式

智能手环从软件结构上分为应用层、中间件、硬件抽象层，如图所示。

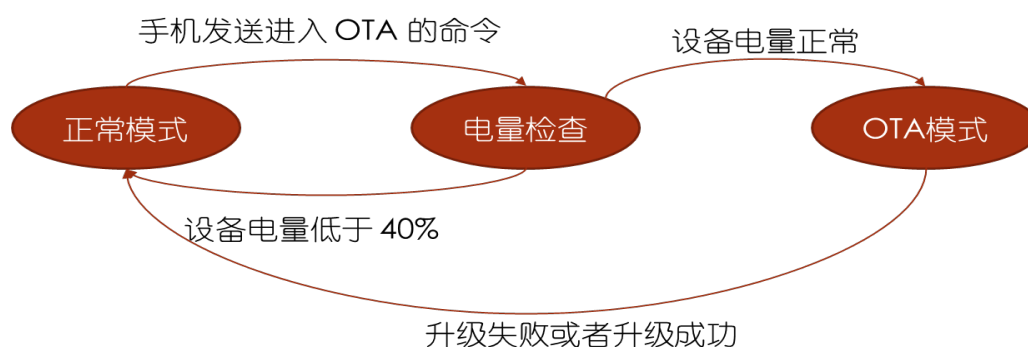
- 其中闹钟、时间、提醒、计步、睡眠等用户可以直接接触的功能属于应用层。
- 中间件为应用层提供支撑，从底层获取数据和事件，然后进行处理分发给不同的应用模块。
- 硬件抽象层包含了底层硬件的驱动和暴露给中间件的调用接口，抽象层的目的硬件设备可以比较容易的更换厂家和型号，只要传递给上层的数据格式不变即可。
- 蓝牙部分硬件抽象层封装在 Nordic 提供的 softdevice 中，在此之上，我们使用了两个标准 profile：Battery profile 来传递电池信息给手机客户端，Device info profile 传递设

备信息给客户端；其他自定义的数据，如步数、睡眠信息，我们按照数据通信私有协议组帧后，通过串口协议发送给手机客户端，手机客户端再依据数据通信私有协议进行解析。



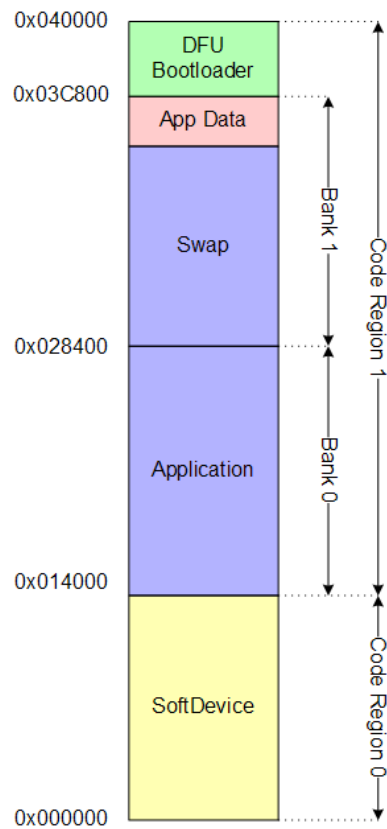
3.2 升级模式（OTA 模式）

升级过程如下图所示：

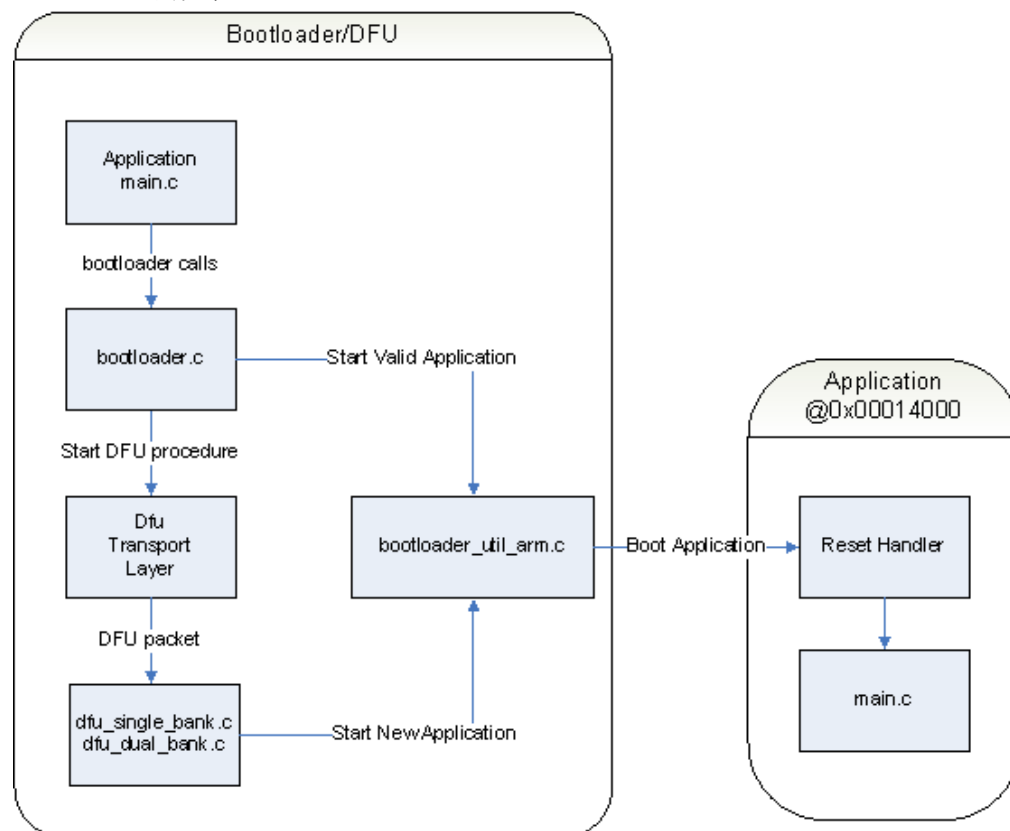


要了解 OTA 模式需要知道 ROM 的内存分布，如下图所示，我们使用 256KB Flash 的 51822 芯片，分为四个区。

进入 OTA 后，系统运行在 DFU BootLoader 中，它从手机端收到新的 ROM 放在 Swap 区，接收完毕、校验成功后，将新的 ROM 从 Swap 区复制到 Application 区，然后重新启动进入 Application。



OTA 的工程单独提供，编译出来一个 **bootloader**。其工作流程图如下，具体可以参看 Nordic SDK 的文档。



4. 交互设计

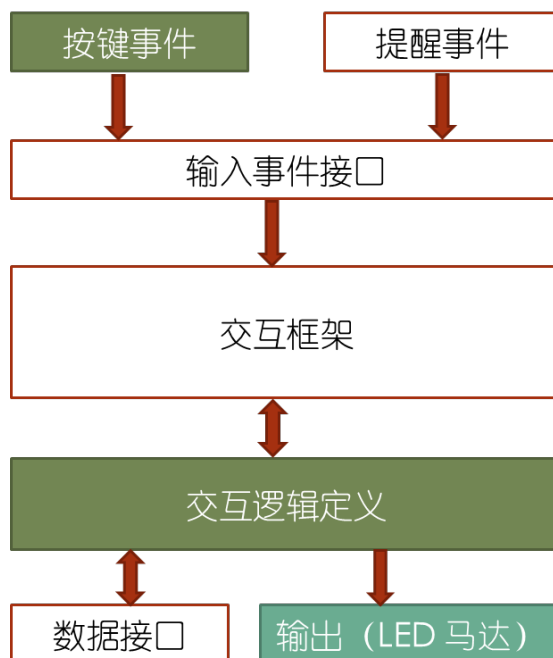
4.1 交互设计原则(改交互设计，需要先阅读本节内容)

交互设计秉承抽象分层的理念，输入设备抽象出来，对应到手环产品的按键事件。也可以对接到敲击识别，语音识别等，这些输入事件发生后传递给输入事件接口层，然后发送给交互框架。

交互框架负责依据交互定义分派交互事件；

交互定义根据具体的产品形态定义，以手环为例分别定义了以下交互事件，并定义了优先级和优先级的处理规则。这些规则的具体实现可以根据输出设备的选择而不同，比如目前使用的显示设备是 LED 点阵，也可以换成 LCD 或者其他的显示设备。

交互实现参考如下结构图：



5. 功能自定义

5.1 硬件相关修改

公共功能的定义在 config.h，不建议修改。和板子功能相关的定义可以参考下面文件：
dvk6310_board_config_pins.h

设备信息可以自定义：

```
#define DEVICE_NAME "DuLife" /**< Name of device. Will  
be included in the advertising data. */  
#define MANUFACTURER_NAME "Baidu Inc" /**< Manufacturer.
```

```
Will be passed to Device Information Service. */
#define FW_REV_STR            "0.9.0"
#define MODULE_NUM            "DVK6310"    /**< Model Number String. */
#define SERIAL_NUM            "0123456789abcde"    /**< Serial Number String. */
#define HW_REV_STR            "1.0"        /**< Hardware Revision String. *
```

GPIO 引脚定义可根据实际需求修改：

```
#define DISPLAY_SCL_PIN      9//10 /*!< GPIO pin number for display module AS1130 */
#define DISPLAY_SDA_PIN      10//11 /*!< GPIO pin number for display module AS1130 */
```

5.2 OTA 的客制化

OTA 的客制化主要是进入 OTA 模式之后手环上提示灯设计。这部分代码会另外提供，修改主要在 boot loader 的 main 函数。

如果修改了 IO 定义，也需要在 bootloader 工程中对应修改。

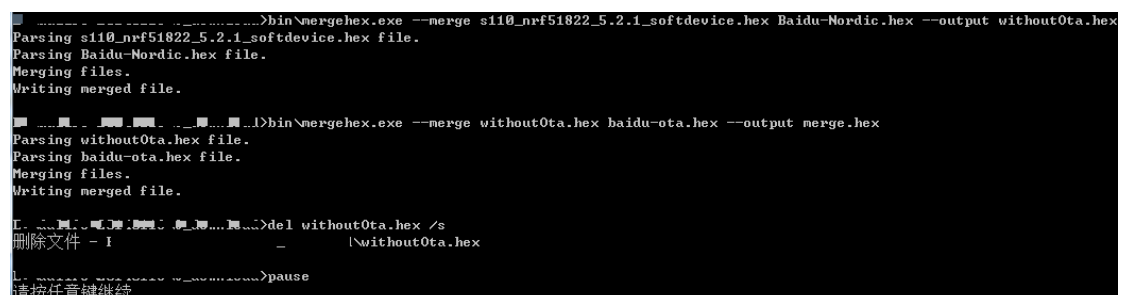
6. ROM 打包与烧写

开源工程有提供打包烧写工具，在 yunFlash 中。

其中 baidu-ota.hex 由 boot loader 工程生成，简约版和增强版是不同的，需要注意。

6.1 打包步骤如下：

1. 把编译出来的 Baidu-Nordic.hex 和 baidu-ota.hex 复制到，yunFlash.exe 同级目录；
2. 运行 merge.bat 会把上面的两个 hex 文件和 s110_nrf51822_5.2.1_softdevice.hex 一起合并生成 merge.hex，这个就是我们要用的烧入产线的 ROM。
3. 如果运行成功可以看到下面的画面：



```
>bin\mergehex.exe --merge s110_nrf51822_5.2.1_softdevice.hex Baidu-Nordic.hex --output withoutOta.hex
Parsing s110_nrf51822_5.2.1_softdevice.hex file.
Parsing Baidu-Nordic.hex file.
Merging files.
Writing merged file.

>bin\mergehex.exe --merge withoutOta.hex baidu-ota.hex --output merge.hex
Parsing withoutOta.hex file.
Parsing baidu-ota.hex file.
Merging files.
Writing merged file.

C:\Users\user>del withoutOta.hex /s
删除文件 - I          \withoutOta.hex

C:\Users\user>pause
请按任意键继续. . .
```

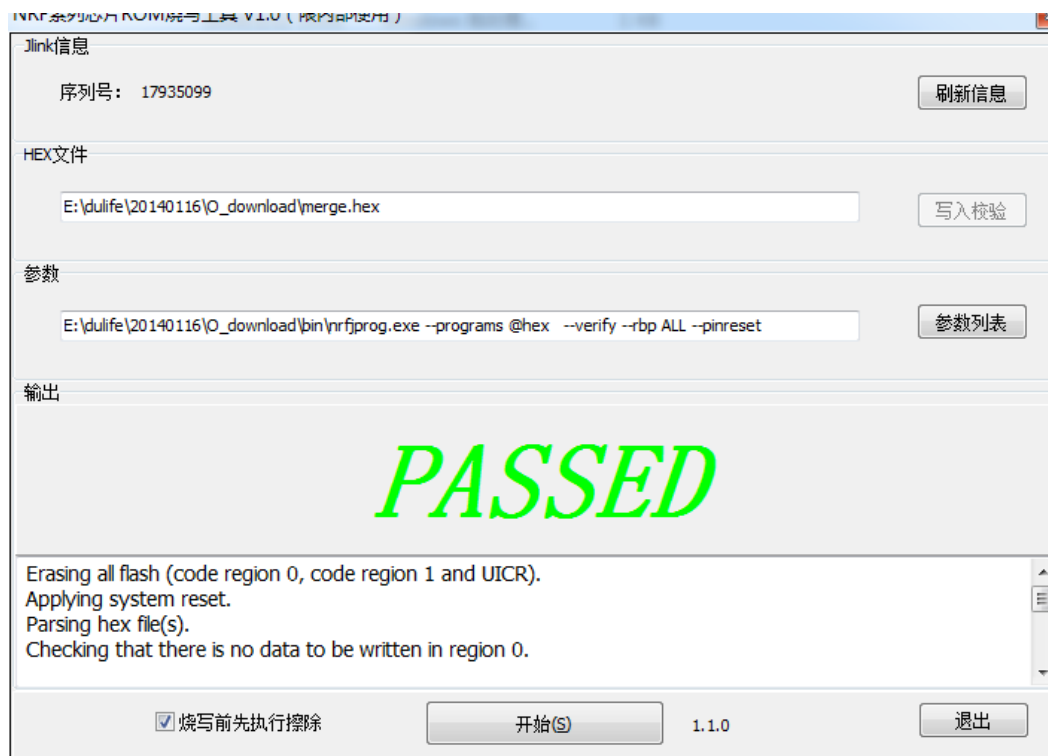
6.2 烧写步骤

1. 连接 J-Link 调试器和要烧入的电路板

2. 打开 YunFlash.exe 点击“写入校验”；
3. 成功后点击开始，如下图



4. 等待烧写校验完成；



烧写完成后可以用厂测 apk 进行测试。

厂测部分见工厂测试手册。