**HoneyWell蓝牙红外编址器Android客户端**

**技术文档**

|  |  |
| --- | --- |
| **产品负责人** | **祁宏涛** |
| **文档编写人** | **祁宏涛** |
| **文档审核人** |  |
| **开始 时间** | **2017-09-01** |

**文档历史记录**

注：后期所加内容均绿色背景字体标注

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **日期** | **版本** | **描述** | **作者** | **审阅者** |
| 01 | 2018-03-10 | 0.1 | 技术文档 | 祁宏涛 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

**[一 说明 6](#_Toc31656)**

[1.1 注意： 6](#_Toc22131)

[1.2 APP简介与环境说明： 6](#_Toc29061)

**[二 技术方案（build.gradle文件compile的库） 7](#_Toc12094)**

[2.1 蓝牙框架： 7](#_Toc9717)

[2.2 本地数据库框架： 7](#_Toc23577)

[2.3 View注入框架： 7](#_Toc11517)

[2.4 事件总线框架： 8](#_Toc22995)

[2.5 RecycleView Adapter 封装框架： 8](#_Toc18558)

[2.6 Execl操作框架： 8](#_Toc25065)

[2.7 其他工具、库： 8](#_Toc22167)

**[三 核心点： 9](#_Toc8748)**

[3.1 项目整体结构： 9](#_Toc15681)

[3.2 蓝牙收发技术： 10](#_Toc17015)

**[四 数据库表： 11](#_Toc32676)**

[4.1 说明： 11](#_Toc12580)

[4.2：表结构： 11](#_Toc13113)

[4.3：SharePrefrence数据表： 14](#_Toc13352)

**[五 核心点业务流程图： 14](#_Toc31489)**

[5.1 单次编址： 14](#_Toc27638)

[5.2 连续编址： 16](#_Toc8524)

**[六 后期改动或实践可能出现问题： 16](#_Toc7834)**

[6.1 库版本修改： 16](#_Toc27884)

[6.2 中英文版本： 16](#_Toc5303)

[6.3 增加/修改功能涉及到6.0动态权限： 17](#_Toc29892)

[6.4 增加手机号注册功能/迁移本地数据库至网络： 17](#_Toc24384)

**[七 图示： 18](#_Toc3349)**

# 一 说明

## 注意：

1. 最新项目下载地址：

http://202.200.112.149:10010/tree/HoneyWellAPP.git/master。

2、此文档适用于0.1版本APP的维护和0.1+版本的开发与维护，务必详读。

3、文档中提到的第三方库、框架请进行一定的技术学习，并学会使用。

## APP简介与环境说明：

1. 本APP分为两个部分：蓝牙（BLE）编址器和红外（IR）编址器，通过手机发送蓝牙和红外命令给对应设备（所有硬件由甲方提供），运行环境说明如下：

|  |  |  |
| --- | --- | --- |
| **序号** | **名称** | **说明** |
| 1 | 安卓手机 | 安卓4.3+、API 18+以上版本的安卓手机才开始支持BLE |
| 2 | 蓝牙模块 | 图1所示，绿色的为蓝牙开发板，白色为探头，采用电池或者usb供电，长按黑色按钮3S以上开机。  对应蓝牙名称：HoneyWellToolkit |
| 3 | 红外模块（EL） | 图X所示，插到手机Micro口上，手机只提供供电，最终命令交互采用BLE。手机发送命令给设备，设备内部通过电路转化为红外信号，并发送出去控制对应的灯板。 |
| 4 | 红外灯板 | 图X所示，有指示灯和照明灯，可接收由图X设备发送的红外信号。 |
| 5 | BLE | 蓝牙4.0协议，低功耗蓝牙，手机通过蓝牙连接图X的设备，发送蓝牙命令给设备，控制白色探头。BLE支持一个中心设备（手机）连接多个从机（蓝牙设备）。 |
| 6 | IR | 红外线，使用NEC 协议。38KHz载波，940nm中心频率，一对一操作。 |

# 二 技术方案（build.gradle文件compile的库）

## 2.1 蓝牙框架：

采用github上一个优质的开源库FastBLE作为本项目蓝牙功能的快速开发框架，可以很方便的进行蓝牙的扫描、连接、读写等操作，作者一直在进行维护。

地址：<https://github.com/Jasonchenlijian/FastBle。>

## 2.2 本地数据库框架：

经过比较，采用LitePal作为项目的数据库框架，避免手写Sql语句，很方便的操作各个表，作者一直在进行维护。

地址：<https://github.com/LitePalFramework/LitePal。>

## 2.3 View注入框架：

ButterKnife，避免手写大量的findViewById，和避免手写onClicker点击事件，使用android studio 插件 “Android ButterKnife Zelezny”，一键生成view对象和点击事件。作者一直在进行维护。

地址：<https://github.com/JakeWharton/butterknife。>

## 2.4 事件总线框架：

eventBus，一款成熟的事件总线框架，简化各组件间的通信，方便的在activity和adapter之间传递消息，避免大量的接口，解耦代码。作者一直在进行维护。

地址：<https://github.com/greenrobot/EventBus。>

## 2.5 RecycleView Adapter 封装框架：

BaseRecyclerViewAdapterHelper，封装了recycleView一系列的操作，简化adapter的书写。20+ 位作者在进行维护。

地址：<https://github.com/CymChad/BaseRecyclerViewAdapterHelper。>

## 2.6 Execl操作框架：

jxl，java操作excel表格的工具类库。创建 .xls文件、创建sheet、设置属性等。

demo地址：<https://www.cnblogs.com/biehongli/p/6497653.html。>

## 2.7 其他工具、库：

1） 数据库调试福利：

[https://github.com/amitshekhariitbhu/Android-Debug-Database](https://github.com/amitshekhariitbhu/Android-Debug-Database" \t "http://blog.csdn.net/ss1168805219/article/details/_blank)。

2）nineoldandroids：兼容3.0下的属性动画库

见lib文件夹下。

3）zxing：二维码扫描库

见Util文件夹下，io.github.xudaojie.qrcodelib。

1. 百度TTS：语音库

百度TTS（百度语音合成），支持离线语音，讯飞不支持。

登录网址：http://yuyin.baidu.com/app

用户名：1003077897@qq.com

密码：qht.19931001

# 三 核心点：

## 3.1 项目整体结构：



1）项目定义了3个基类Activity，一个基类Application，一个基类Fragment。

statusActivity管理状态栏，ToolBarActivity管理ToolBar，所有的Activity需要继承自ToolBarActivity。除非不需要ToolBar。与此同时，所有Activity的xml文件需要include toolbar布局。在具体的实现上，只需要重写getContentViewId方法，返回布局id即可。

1. 项目分包按照module分包，基类单独分包，BleModule、ELModule、LoginModule、Util包、自定义View包等。每个包下文件夹按照adapter、data、widge的方式区分Adapter、Javabean、Activity。
2. 所有的资源文件在drawable大文件夹下，每种图片存在一套，存于drawable-xhdpi目录。动画文件、selecter文件、shape文件等存于drawable目录下。
3. 修改litepal数据库的表结构时，注意维护assets文件夹下litepal.xml文件。

## 3.2 蓝牙收发技术：

FastBle使用方法详情见：[https://github.com/Jasonchenlijian/FastBle](https://github.com/Jasonchenlijian/FastBle。)

1）项目使用固定的蓝牙名称进行连接，两个蓝牙名称和对应的UUID定义在ConstantUtil 类下。

2）连接成功之后进行Write 和Notify 操作。Write成功不能保证真正的写成功，需要Notify 到下位机的消息验证是否成功。

3）对于Write 数据的封装和Notify消息的解析全部在DataHandler类中，方法有注释。

4）在Util包下的CommonBleUtil类中定义了OnResultListtener接口，可以将原本分割的Write操作和Notify操作结果统一管理，回调给SingleFreeAddressActivity。

5）蓝牙空中包数据最大20字节，超出需要分包发送与接收（项目中暂未超出）。FastBle新版本有分包接口，如需接入，请先进行风险评估。

6）蓝牙操作中经常出现 Thread.sleep(100) 等代码，是为了避免蓝牙操作中的失败和异常情况，慎重使用，因为直接Thread.sleep(100)是会阻塞主线程的。

7）待补充。

# 四 数据库表：

## 4.1 说明：

数据库采用litepal 开源数据库框架，有一定的学习成本。碰见问题先看issue，找有没有人和你碰到一样的问题，其次提问作者。

private 修饰代表映射到数据库中，public 不会被映射到数据库中。

## 4.2：表结构：

表1 logininfobean表字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段 | 功能 | 修饰符 |
| 1 | id | 不声明也可以 | private |
| 2 | phone | 蓝牙用户注册手机号 | private |
| 3 | password | 对应密码 | private |
| 4 | username | 蓝牙登录用户名 | private |
| 5 | usertype | 用户类型，开发人员/现场人员 | private |

表2 projectbean表字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 方法 | 功能 | 修饰符 |
| 1 | id | 不声明也可以 | private |
| 2 | isselect | Item是否被选择（编辑功能使用） | private |
| 3 | isexpand | Item是否被展开 | public |
| 4 | tasknumber | 项目编号（肉眼看到的编号） | private |
| 5 | logininfo\_id | 登录id | private |
| 6 | projectname | 项目名称 | private |
| 7 | projectdescription | 项目描述 | private |
| 8 | projectdata | 项目日期 | private |
| 9 | controlleramount | 下级菜单-->控制器数量 | private |
| 10 | controllerlist | 控制器list，懒加载 | private |

表3 controllerbean表字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 方法 | 功能 | 修饰符 |
| 1 | id | 不声明也可以 | private |
| 2 | isselect | Item是否被选择（编辑功能使用） | private |
| 3 | isexpand | Item是否被展开 | public |
| 4 | tasknumber | 项目编号（肉眼看到的编号） | private |
| 5 | logininfo\_id | 登录id | private |
| 6 | project\_id | 项目id | private |
| 7 | controllername | 控制器名称 | private |
| 8 | controllerdescription | 控制器描述 | private |
| 9 | controllerdata | 控制器日期 | private |
| 10 | fathertaskamount | 下级菜单-->回路数量 | private |
| 11 | fathertasklist | 回路list，懒加载 | private |

表4 fatherbean表字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 方法 | 功能 | 修饰符 |
| 1 | id | 不声明也可以 | private |
| 2 | isselect | Item是否被选择（编辑功能使用） | private |
| 3 | isexpand | Item是否被展开 | public |
| 4 | tasknumber | 回路编号（肉眼看到的编号） | private |
| 5 | logininfo\_id | 登录id | private |
| 6 | project\_id | 项目id | private |
| 7 | controller\_id | 控制器id | private |
| 8 | taskname | 回路名称 | private |
| 9 | taskdescription | 回路描述 | private |
| 10 | taskdata | 回路日期 | private |
| 11 | fathertaskamount | 下级菜单-->回路数量 | private |
| 12 | sontasklist | 序列号任务list，懒加载 | private |
| 13 | progressnpercent | 子任务进度 | private |

表5 sonbean表字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 方法 | 功能 | 修饰符 |
| 1 | id | 不声明也可以 | private |
| 2 | isselect | Item是否被选择（编辑功能使用） | private |
| 3 | isexpand | Item是否被展开 | public |
| 4 | tasknumber | 任务编号（肉眼看到的编号） | private |
| 5 | logininfo\_id | 登录id | private |
| 6 | project\_id | 项目id | private |
| 7 | controller\_id | 控制器id | private |
| 8 | fathertask\_id | 回路id | private |
| 9 | taskname | 任务名称 | private |
| 10 | taskdescription | 任务描述 | private |
| 11 | taskdata | 任务日期 | private |
| 12 | sondescription | 任务描述 | private |
| 13 | taskdigitaladdress | 数字地址 | private |
| 14 | taskserialnumber | 序列号 | private |
| 15 | taskloopnumber | 任务回路地址 | private |
| 16 | process | 地址下载进度 | private |

表6 logbean表字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 方法 | 功能 | 修饰符 |
| 1 | id | 不声明也可以 | private |
| 2 | user | 登录用户 | private |
| 3 | data | 日期 | private |
| 4 | protocoltype | 协议类型 | private |
| 5 | operatetype | 操作类型 | private |
| 6 | address | 编写的地址 | private |
| 7 | result | 编址的结果 | private |
| 8 | detail | 编址的详情 | private |

## 4.3：SharePrefrence数据表：

表7 shareprefrence字段（使用Android debug database查看）

|  |  |  |
| --- | --- | --- |
| 序号 | 字段 | 描述 |
| 1 | apptype | 当前使用APP的类型（ble/el） |
| 2 | currentphone | 注册蓝牙APP的手机号 |
| 3 | currentphone\_el | 注册红外APP的手机号 |
| 4 | currentpsaaword | 注册蓝牙APP的密码 |
| 5 | currentpsaaword\_el | 注册红外APP的手机号 |
| 6 | currentusername | 注册蓝牙APP的用户名 |
| 7 | currentusername\_el | 注册红外APP的手机号 |
| 8 | currentusertype | 当前蓝牙APP的用户类型 |
| 9 | currentusertype\_el | 当前红外APP的用户类型 |
| 10 | notfirstusebleapp | 是否第一次使用APP |

# 五 核心点业务流程图：

1）不论是单次编址、还是连续编址，每次write完成后哪怕notify到的数据是正确的，也不能保证地址真正的写入成功了。而是需要再次发送read命令，根据再次notify到的数据验证地址是否被写入成功。

2）连续编址输入开始地址，将整个连续的过程分解为单次编址，选择好地址累加数，当编写成功一次后间隔2s地址自动累加，直到手动点击结束或者到达地址边界值停止。

## 5.1 单次编址：



## 5.2 连续编址：

详见文档：蓝牙编址器方案设计调整\_2018.1.19(1).docx

# 六 后期改动或实践可能出现问题：

## 6.1 库版本修改：

对于build文件中的complie库，如果不熟悉此库的使用和不清楚项目中对该库的使用方式，请勿修改版本。

## 6.2 中英文版本：

后期如果甲方考虑加入英文版本，注意修改项目中所有出现汉字的地方，包括但不限于以下：

1）包下所有.java文件。

2）所有.xml文件。

3）检查所有的dialog，所有的toast。

4）检查所有的value文件。

5）注意英文版本必然导致文本长度增长，所以某些地方有必要进行缩写。

建议：动用3-5名同学，将所有出现的中文改为在string.xml中定义，在其他使用的地方引用string文件即可。

## 6.3 增加/修改功能涉及到6.0动态权限：

如果增加的功能涉及到申请权限，请务必注意是否是危险权限，若是危险权限必须在activity中动态申请，如果只在manifest中注册可能会导致APP崩溃。

## 6.4 增加手机号注册功能/迁移本地数据库至网络：

如果后期甲方购买了数据库，则需要进行一定工作的开发，需要将手机本地的数据库迁移到PC端，迁移时务必理清数据库关系，并掌握litepal的使用，将所有的对象操作（实为sql语句，litepal进行了封装）在后台代码汇中改为sql语句（或者使用hibernate）。

如果加入了手机号注册功能，这需要一定的学习成本，碰到不会或无法解决的问题请教张华龙老师。

# 七 图示：

图1 蓝牙模块，白色为探头

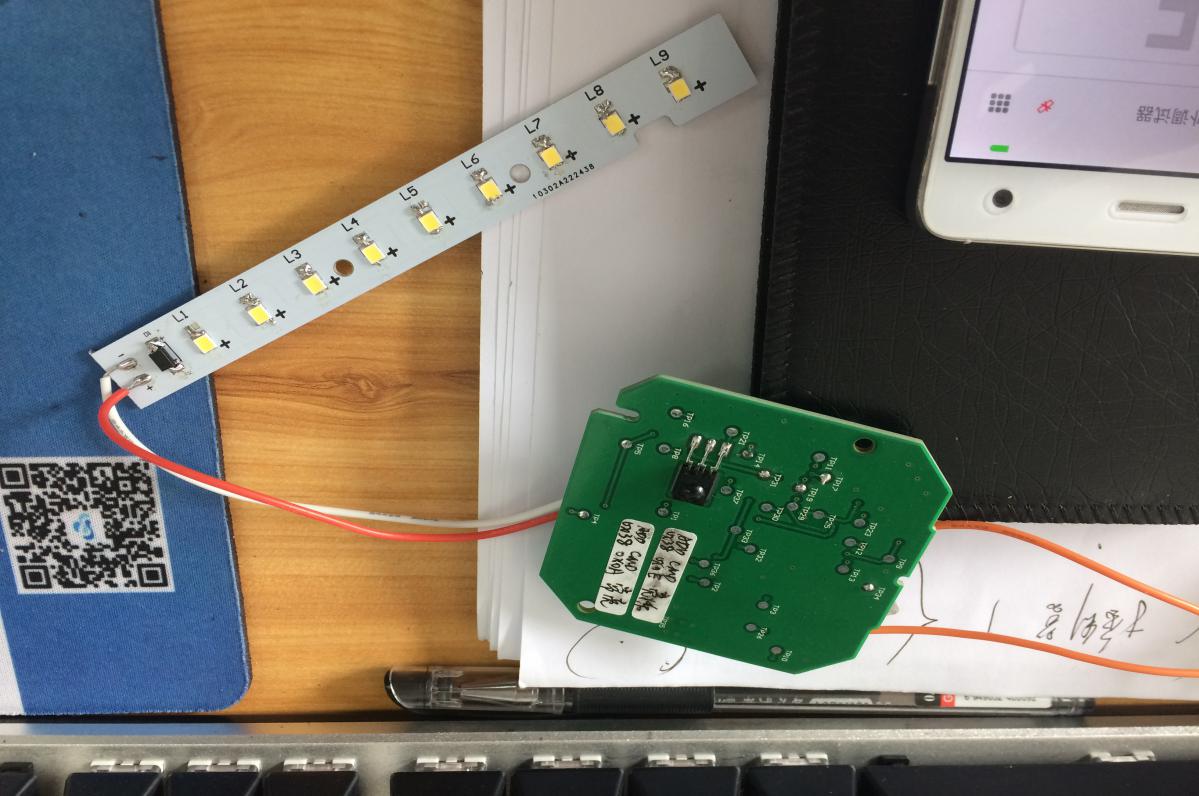
图2 红外照明灯

图3 红外指示灯