

悦动圈 硬件开放sdk文档

名词定义：

应用（代表悦动圈应用）

插件（代表硬件插件化部分）

概要说明：

插件 与 应用 独立打包 不存在直接代码交互

硬件插件 需要做的事情

- 1 以一个Activity做为入口，展示硬件的数据及功能
- 2 用户 绑定蓝牙设备之后
 - 插件向应用注册 获取device_identify 并保存
 - 根据需求注册 Action （唤起，蓝牙状态发生改变，关闭蓝牙等）
- 3 使用ContentProvder 进行数据的读写 （需要提供device_identify）
- 4 接收Action 执行相应操作
 - 唤起（尝试连接蓝牙，与硬件同步数据）
 - 蓝牙状态发生改变 （需要长连接设备 开启蓝牙等）
 - 关闭蓝牙 （其他硬件需要使用蓝牙，暂时关闭）
- 3 用户 解绑蓝牙设备之后 插件向应用解绑设备

应用 需要做的事情

- 1 提供入口 根据插件提供的Activity（通过类名）打开插件界面
- 2 提供ContentProvder支持插件数据读写
- 3 支持插件绑定 Action
- 4 根据注册的Action，在合适的时候向插件发送Action 维持插件生命周期

Action 需要用Service接收，注册的时候要把该Service的class信息传递过去
后续 应用会向该Service发送Action，从而触发插件的活动

应用提供的组件：

- 1 YDHardwarePlugService 提供AIDL接口允许插件执行设备注册、取消注册等
其中接口

```
interface YDHardwarePlugInterface{
// 绑定硬件设备后需要想用用注册
    String registerDevice(String deviceId, String plugName);
// 解除绑定设备后要注销
    void unRegisterDevice(String deviceId, String plugName);
// 向应用注册Action通知回调 具体值参照YDHardwarePlugConst
```

```
void registerServiceAction(String intentUri, String action, String deviceIdentify);  
// 获取到用户信息的json 格式的字符串  
String userInfoJsonStr();  
}
```

2 ContentProvider 提供给插件使用 用于插件所持有的设备的数据的读写，其中数据库结构参见YDHardwarePlugConst文件

应用与插件交互流程

1 初次使用

1) 应用通过插件的类名 唤起插件界面

demo中对应插件界面(com.yuedong.sdkapplication.SdkActivity)

Intent 中会指定uid

2) 插件执行绑定蓝牙设备操作，蓝牙设备绑定成功后 通过AIDL接口向应用申请 deviceIdentify(SdkActivity.bindDevice)；获取到deviceIdentify后插件需要保存下来

3) 根据插件需要使用AIDL接口向应用注册自己需要的Action(SdkActivity.registerAction)

2 持续使用

1) 应用会在特定时候向插件发送插件自己注册的Action，数据的传递使用intent

2) 插件读取到硬件的数据 然后使用ContentProvider存储下来(SdkActivity.writeData)

3) 插件展示历史数据 使用ContentProvider读取(SdkActivity.readData)

3 用户解除绑定蓝牙设备

插件使用AIDL接口向应用注销deviceIdentify(SdkActivity.unbindDevice)

注意事项：

插件资源 要求使用特定前缀 防止与宿主程序 资源冲突

悦动圈硬件开放平台可以做的：

支持插件Activity, Service, BroadcastReceiver, 支持插件在不增加三种组件的前提下在线更新插件不需要安装悦动新版

悦动圈开放平台不能直接做到的：

插件新增Activity, Service, BroadcastReceiver, 不能在线更新,需要悦动发版本
新的插件发布,需要悦动发版本

建议：

插件可以预先声明Android三种组件占位,则后续可以不需要悦动发版,发布新的功能 更新插件