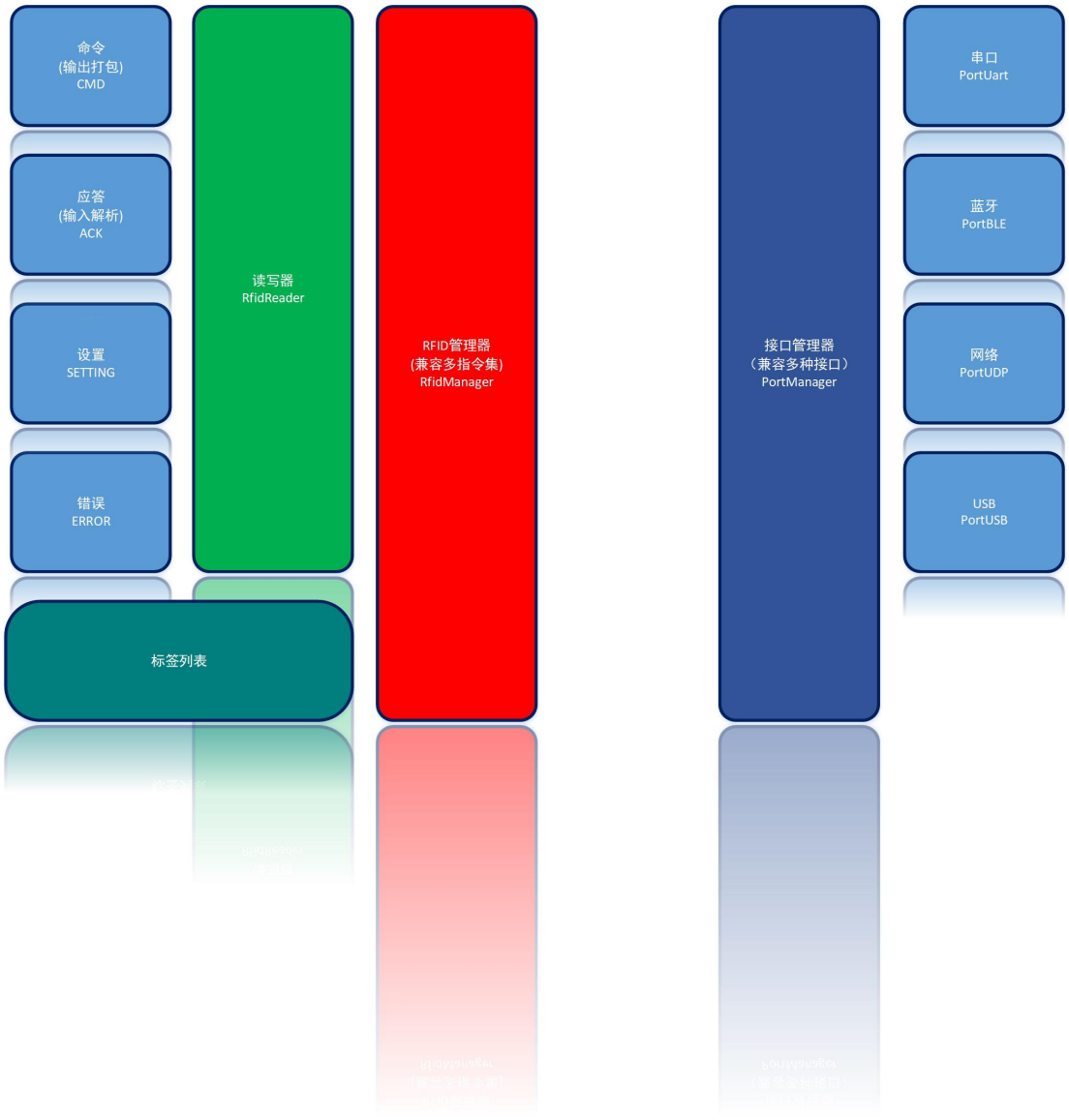
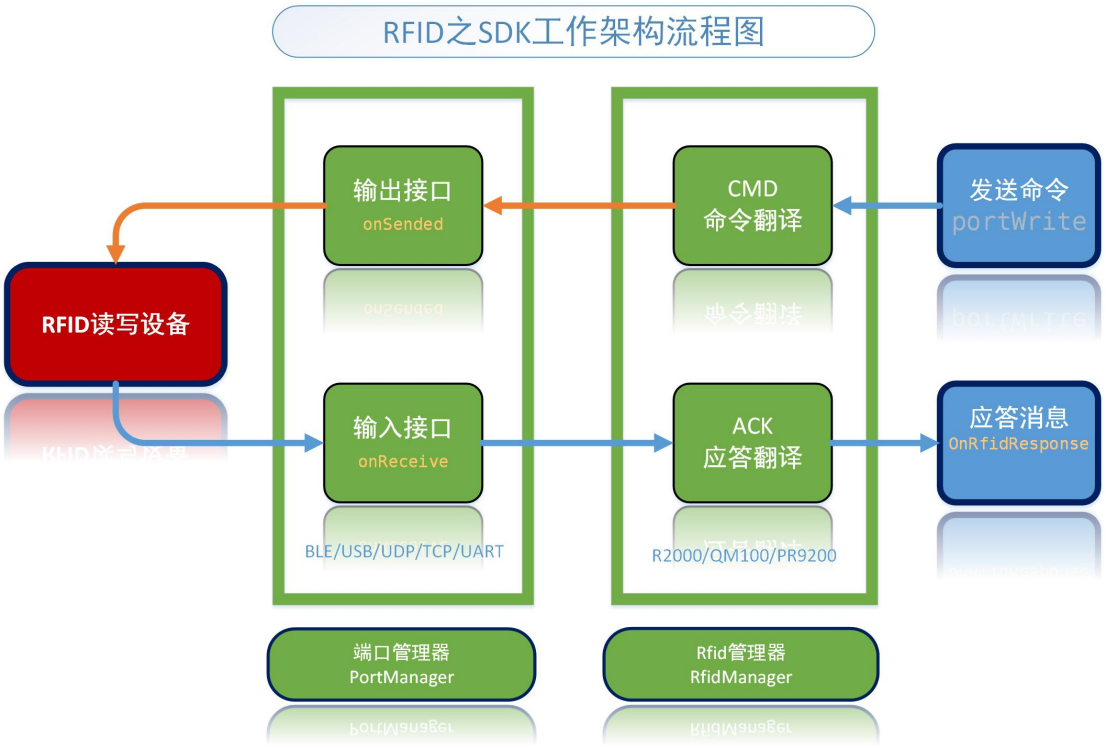
安卓开发包使用说明

版本:1.01

1. 环境要求:Android Studio 3.0 以上版本
2. 系统要求:安卓 5.0 以上版本
3. 系统架构:



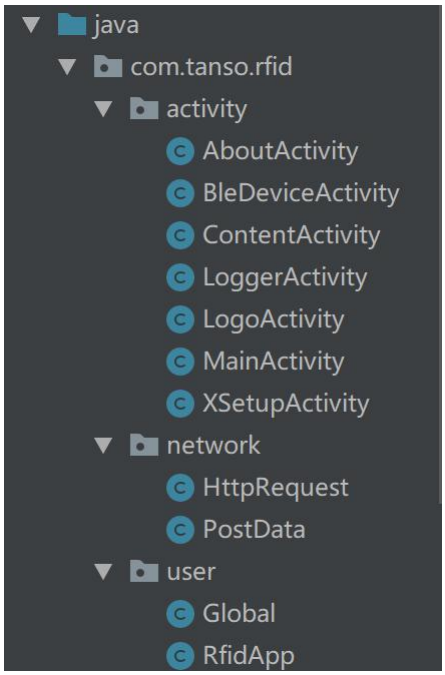


本开发包架构主要包含两大块:

1. 接口管理器(PortManager)
   目前支持:串口,网口,USB,蓝牙.

2. 电子标签管理器(RfidManager)
   目前支持:M100,R2000,J2000,...
   支持多种通道:1/2/4/8/16.

演示代码结构如下:



主要内容就在 RfidApp.java,MainActivty.java.设置信息在 XSetup.java

demo 程序使用库如下：

名称	修改日期	类型
devlib-release.aar	2019-05-25 17:11	AAR 文件
guilib-release.aar	2019-06-10 12:30	AAR 文件
rfidlib-release.aar	2019-06-10 12:30	AAR 文件

基本用法只需要 **rfidlib-release.aar** 一个库，

先创建两个管理器.（指定读写器类型，通道数，指定接口类型）

```
//  标签管理器
rfidManager = new RfidManager( context: this, USE_RFID_READER, USE_RFID_CHANNELS);

//  接口管理器
portManager = new PortManager( context: this, USE_RFID_PORT);
```

关联当前接口

```
//  关联 - 接口
rfidManager.setPort(portManager.getPort());
```

关联消息处理，调用连接接口.

```
//  应用
RfidApp theApp = (RfidApp) this.getApplication();
//  管理器
PortManager portManager = theApp.portManager;

//  消息接口(RFID)
theApp.rfidManager.setRfidEvent(this);
//  消息接口(PORT)
portManager.setPortEvent(this);
//  端口 - 连接
portManager.getPort().connect();
```

获取当前读写器

```
//  当前读写器
BaseReader reader = theApp.rfidManager.getReader();
//  当前命令接口
BaseCmd cmd = reader.cmd;
//  检查设置数据范围
reader.set.checkData();
```

发送指令接口

```
/**
 * 发送数据
 *
 * @param array : 数据
 */
public void portWrite(byte[] array) {
    rfidManager.putCmd(array);
}
```

定期发送轮询指令，**不发指令，不会自动读取**.

```
//============================================
//  轮询inventory_times次
//============================================
theApp.portWrite(cmd.rfid_inventory(reader.set.inventory_times));
```

如果是多通道的读写器，需要定期切换天线.

```
//  多通道模块，可以切换天线
if (reader.getChannels() > 1) {
    //  下一个天线
    int ant = reader.getNextAntenna();
    if (DEBUG) {
        LoggerActivity.Log_e(TAG,  msg: "天线:" + (ant + 1));
    }
    //  设置天线(0 ~ (N - 1))
    theApp.portWrite(cmd.rfid_set_antenna(ant));
} else {
    if (DEBUG) {
        LoggerActivity.Log_d(TAG,  msg: "单天线模式,不切换!");
    }
}
```

收到标签，或是其他消息，进如下回调接口：

```
/**
 * RFID - 消息响应
 *
 * @param type  : 类型
 * @param cmd   : 命令
 * @param param : 参数
 * @param obj   : 对象
 */
@Override
public void OnRfidResponse(int type, int cmd, byte[] param, Object obj) {
```

以上响应接口会收到 **obj** 对象，就是**解析之后**的对象.（不同指令，返回不同对象）

TagItem 对象对应的就是一个标签. 基本结构如下：

```
public class TagItem implements Comparable<TagItem>, ByteReadWrite {

    /**
     * 序号
     */
    public int index;

    /**
     * 信号强弱(1 byte,0~127dBm)
     */
    public int rssi;

    /**
     * PC-通讯协议(2 bytes)
     */
```

```
    public short pc;

    /**
     * 校验码(2 bytes)
     */
    public int crc;

    /**
     * EPC-电子标签码(12 bytes)
     */
    public byte[] epc;

    /**
     * RFU 区域
     */
    public byte[] rfu;

    /**
     * TID 区域-唯一码
     */
    public byte[] tid;

    /**
     * USER 区域-用户区
     */
    public byte[] user;

    /**
     * 天线(1byte)
     */
    public int ant;

    /**
     * 累计
     */
    public int count;

    /**
     * 激活时间
     */
    public long time;

    /**
```

```
     * 首次时间
     */
    public long first;

    /**
     * 关联信息
     */
    public Object tag;
```

rfidManager 直接获取标签列表调用如下接口

```
/**
 * 标签列表
 *
 * @return : 列表
 */
public ArrayList<TagItem> getList() {
    return listTags;
}
```

读写器对象有 4 个模块组成,对应常见的处理.

```
/**
 * 应答
 */
public BaseAck ack;

/**
 * 命令
 */
public BaseCmd cmd;

/**
 * 错误
 */
public BaseErr err;

/**
 * 设置
 */
public BaseSet set;
```

基本用法介绍完毕,欢迎大家探索深度用法.

备注:

附带的库还有比较好用的东西.比如:对话框接口,设置列表接口,对象字节化接口,设置字节化接口等.都是本公司的研发成果,欢迎大家使用!

作者:施探宇
微信:18680399436
邮箱:Alecksty@163.com
网站:http://www.ts-rfid.com
深圳探索智能科技有限公司
2019 年 6 月 13 日