RFID（UNI-APP）

开发文档

V2.12

**目录**

[1. 常用方法使用流程 4](#_Toc10079)

[1.1 初始化 4](#_Toc30321)

[1.2 盘点 4](#_Toc18379)

[1.3 读取标签 5](#_Toc15166)

[1.4 写标签 5](#_Toc207)

[1.5 读取功率 5](#_Toc3349)

[1.6 写入功率 6](#_Toc23365)

[1.7 参数设置 6](#_Toc14814)

[1.8 长距（带手柄）设备控制扫描头出光 7](#_Toc7180)

[2. 回调接口 8](#_Toc12101)

[onInventoryTag 8](#_Toc27963)

[onInventoryTagEnd 9](#_Toc7586)

[3. RFID相关接口 9](#_Toc6661)

[3.1 init【初始化】 9](#_Toc22216)

[3.2 release【释放资源】 9](#_Toc15021)

[3.3 disconnect【断开rfid连接】 10](#_Toc28836)

[3.4 isConnected【判断是否连接上rfid模块】 10](#_Toc22846)

[3.5 registerCallback【注册回调】 10](#_Toc12701)

[3.6 startInventory【开始扫描（盘点）】 11](#_Toc7399)

[3.7 stopInventory【停止扫描】 11](#_Toc15625)

[3.8 readTag【读取标签】 11](#_Toc14617)

[3.9 readDataByTid【通过tid读取数据】 12](#_Toc26676)

[3.10 inventorySingle【单标签查询】 13](#_Toc1773)

[3.11 writeTag【写标签】 13](#_Toc9429)

[3.12 writeTagByTid【通过tid向各存储区写数据】 14](#_Toc8789)

[3.13 writeEpc【广播方式随机改写一张标签的EPC号】 15](#_Toc5086)

[3.14 killTag【销毁标签】 15](#_Toc29212)

[3.15 KillbyTID【通过TID销毁标签】 15](#_Toc11267)

[3.16 lockTag【锁定标签】 16](#_Toc30932)

[3.17 LockbyTID【通过TID锁定标签】 17](#_Toc31105)

[3.18 setQueryMode【设置查询模式】 18](#_Toc32271)

[3.19 getQueryMode【获取查询模式】 19](#_Toc26838)

[3.20 getFirmwareVersion【获取固件版本】 19](#_Toc23201)

[3.21 setOutputPower【设置输出功率】 20](#_Toc23696)

[3.22 getOutputPower【查询输出功率】 20](#_Toc18368)

[3.23 setFrequencyRegion【设置频率区域】 21](#_Toc3690)

[3.24 getFrequencyRegion【获取频率区域】 21](#_Toc20529)

[3.25 addMask【添加掩码】（短距不支持） 22](#_Toc32314)

[3.26 clearMask【清除掩码】 23](#_Toc15343)

[3.27 unregisterCallback【注销回调】 23](#_Toc1422)

[3.28 getModuleFirmware【获取模块固件信息】 23](#_Toc20969)

[3.29 setScanInterval【设置盘存间隔时间】 23](#_Toc7394)

[3.30 getScanInterval【读取盘存间隔时间】 24](#_Toc26824)

[3.31 startRead【启动盘点】 24](#_Toc2006)

[3.32 scanRfid【询查标签】 25](#_Toc18145)

[3.33 setInventoryParameter【设置盘点参数】 25](#_Toc2000)

[3.34 getDeviceId【获取序列号】 25](#_Toc4800)

[3.35 setCustomRegion【设置自定义频段】（国标无效） 25](#_Toc28907)

[3.36 getCustomRegion【获取自定义频段】（国标无效） 26](#_Toc24808)

[3.37 setProfile【设置链路配置信息】 26](#_Toc6950)

[3.38 GetReaderType【获取模块类型】 27](#_Toc4288)

[3.39 getReaderTemperature【查询内部温度】 27](#_Toc14886)

[3.40 startInventoryLed【询查ELD标签】 28](#_Toc31176)

[3.41 stopInventoryLed【停止询查ELD标签】 28](#_Toc14357)

[3.42 enableLog【开启或者关闭日志】 29](#_Toc13189)

[3.43 writeTagEpc【写标签EPC区】 29](#_Toc21597)

[3.44 writeTagEpcByTid【通过tid写EPC区】 29](#_Toc21963)

[3.45 findEpc【查找标签】 30](#_Toc3408)

[3.46 power【模块上电、下电】 30](#_Toc15431)

[3.47 connect【连接RFID模块】 31](#_Toc21501)

[附录1：常见错误码 32](#_Toc28541)

[附录2：频段说明 33](#_Toc17345)

[附录3: RSSI参数对照表 35](#_Toc9880)

1. **常用方法使用流程**

说明：

由于RFID模块运行状态相对比较耗电，在APP不使用RFID模块或者APP处于后台（不活动状态的时候），建议APP给RFID模块进行下电处理。

对应方法如下（详情可参考demo代码）：

1>使用RFID功能或APP处于前台时：

（该方法执行完在2秒左右，需要逻辑做对应处理）

rfidManager.init( (ret) => {});//初始化

2>RFID功能不使用或APP处于后台时：

rfidManager.disconnect();//断开连接

rfidManager.power(false , (ret) => {});//模块下电

* 1. **初始化**

// 加载插件 获取 rfidManager

var rfidManager = uni.requireNativePlugin("TH-PlatformRFID")

//初始化

rfidManager.init( (ret) => {

if(ret){

//初始化成功

}else{

//初始化失败

}});}

* 1. **盘点**

当rfidmanager初始化后就可以使用盘点方法。

通过rfidmanager.startInventory()方法开始盘点。盘点的结果可以通过，rfidManager.onInventoryTag((result) => {}); 的result回调中获取相关数据。

**步骤：**

开始盘点：

rfidmanager.startInventory(int session);

回调实现体中的onInventoryTag(回调)

* 1. **读取标签**

通过rfidmanager.readTag方法来读标签，

步骤：

var result=

rfidmanager.readTag(epc,btMemBank,btWordAdd,btWordCnt,btAryPassWord),直接返回标签读到的结果。

* 1. **写标签**

通过rfidmanager.writeTag方法。

步骤：

1. 写操作：

//写标签（任意区和地址）

var ret =

rfidmanager.writeTag(epc,btAryPassWord,btMemBank,btWordAdd,btAryData);

//写EPC区

var ret =rfidmanager.writeTagEpc(epc,strPassword,strData);

方法会直接返回结果，0是成功，其他是失败。

**1.5 读取功率**

通过rfidmanager的getOutputPower获取，

1. 获取功率：rfidmanager.getOutputPower()，该方法会直接返回当前设备的功率。

**1.6 写入功率**

通过rfidmanager.setOutputPower方法。

1. 设置输出功率：rfidmanager.setOutputPower(power),该方法会直接返回int值0是成功，其他是失败。

**1.7 参数设置**

参数接口

public class RfidParameter{

public byte ComAddr;

public int IvtType;

public int Memory;

public String Password;

public int WordPtr;

public int Length;

public int Session;

public int QValue;

public int ScanTime;

public int Antenna;

public int Interval;

public byte MaskMem;

public byte MaskAdr[] = new byte[2];

public byte MaskLen;

public byte[]MaskData=new byte[32];

}

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 备注 |
| IvtType | int | 0-EPC查询;1-混合查询;2-FastID.默认0. |
| Memory | int | 混合查询区域，默认2 |
| Password | String | 标签访问密码，默认00000000 |
| WordPtr | int | 混合查询起始地址，默认0 |
| Length | int | 混合查询长度，默认6 |
| Session | int | 范围0~3.默认0 |
| QValue | int | 范围0~15，默认4 |
| ScanTime | int | 询查命令最大响应时间，默认20（N\*100ms），  比如设置2000ms，则传入 20（20\*100） |
| Interval | int | 询查命令之间的间隔时间,默认0 （N\*10ms）,  比如设置10ms，则传入 1（1\*10） |
| MaskMem | byte | 掩码区域，默认2 |
| MaskAdr | byte[] | 掩码起始地址，默认000 |
| MaskLen | int | 掩码长度，0表示不启用掩码 |
| MaskData | byte[] | 掩码数据 |

**获取当前参数：**

var rfidParameter = rfidManager.getInventoryParameter();

**设置参数：**

var rfidParameter = rfidManager.getInventoryParameter();

rfidParameter.Session = 1;

rfidParameter.Interval = 0;

rfidParameter.QValue = 6;

.......

rfidManager.setInventoryParameter(rfidParameter);

**1.8 长距（带手柄）设备控制扫描头出光**

rfidmanager.enableScanHead(boolean isOpen);

*说明：此方法是针对长距带手柄RFID设备，控制手柄上的按键，是否触发扫码头扫码； 传入true,触发扫码；传入false，不触发扫码。*

1. **回调接口**

public interface IRfidCallback {

void onInventoryTag(String EPC, String mData, String strRSSI) ;

void onInventoryTagEnd();

}

注：下面的函数的参数名和上面是一一对应的，只是给出了具体含义，有一些参数目前没有实际作用。

onInventoryTag

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | rfidManager.onInventoryTag((result) => {}); | | |
| **说明** | 该方法会在开启盘点后，通过回调该方法返回找到标签的信息。 | | |
| **参数** | **名称** | **类型** | **备注** |
| result.epc | String | 盘点到的标签的epc号 |
| result.data | String | 标签的TID |
| result.rssi | String | 标签的信号强度 |

**onInventoryTagEnd**

|  |  |
| --- | --- |
| **定义** | rfidManager.onInventoryTagEnd((ret) => {}); |
| **说明** | 此方法盘点结束的时候调用。 |

1. **RFID相关接口**

3.1 init【初始化】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public void init((ret) => {}); | | |
| **说明** | 初始化RFID。  **init()**方法，等同于同时调用了 **3.46 power(true);** 和 **3.47 connect()**方法的组合。 | | |
| **参数** | **名称** | **类型** | **备注** |
|  |  |  |
| **返回** | ret： true 初始化成功；false 初始化失败。 | | |
| **参考代码** | 无。 | | |

3.2 release【释放资源】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public void release(); | | |
| **说明** | 释放资源 | | |
| **参数** | **名称** | **类型** | **备注** |
|  |  |  |
| **返回** | 无 | | |
| **参考代码** | 无 | | |

3.3 disconnect【断开rfid连接】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public boolean disconnect(); | | |
| **说明** | 断开rfid连接模块。 | | |
| **参数** | **名称** | **类型** | **备注** |
|  |  |  |
| **返回**  **(boolean )** | true:断开成功 false:断开失败 | | |
| **参考代码** | 无 | | |

3.4 isConnected【判断是否连接上rfid模块】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public boolean isConnected(); | | |
| **说明** | 判断是否连接上。 | | |
| **参数** | **名称** | **类型** | **备注** |
|  |  |  |
| **返回**  **(boolean)** | 成功：true ；失败：false ； | | |
| **参考代码** | 无 | | |

**3.5 registerCallback【注册回调】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **rfidManager.onInventoryTag((result) => {});** | | |
| **说明** | **注册回调（标签数据或通知将在注册回调中回调）。** | | |
| **参数** | **名称** | **类型** | **备注** |
| **result** |  | **回调** |
| **返回**  **(result)** | **var epc = result.epc;**  **var tid = result.data;**  **var rssi = result.rssi;** | | |
| **参考代码** | **无** | | |

3.6 startInventory【开始扫描（盘点）】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public int startInventory( int session); | | |
| **说明** | 开始扫描标签。 | | |
| **参数** | **名称** | **类型** | **备注** |
| btSession | **Int** | **值有 0、1、2、3 ，一般用 0或者1.** |
| **返回**  **(int)** | 成功：0；  失败：其他。 | | |
| **参考代码** | 无 | | |

3.7 stopInventory【停止扫描】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public int stopInventory(); | | |
| **说明** | 停止扫描标签。 | | |
| **参数** | **名称** | **类型** | **备注** |
|  |  |  |
| **返回**  **(int)** | 成功：0；  失败：其他。 | | |
| **参考代码** | 无 | | |

**3.8 readTag【读取标签】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public String readTag(String EPC,int btMemBank,int btWordAdd,**  **int btWordCnt,String strPassWord);** | | |
| **说明** | **读取标签TID。** | | |
| **参数** | **名称** | **类型** | **备注** |
| **EPC** | **Stirng** | **标签的EPC号，16进制字符串，长度必须是4的整数倍** |
| **btMemBank** | **int** | **待写入的存储区，**  **0 - 密码区，前2个字是销毁密码，后2个字是访问密码**  **1 - EPC区**  **2 - TID区**  **3 - 用户区** |
| **btWordAdd** | **int** | **读取的起始字地址** |
| **btWordCnt** | **int** | **读取的字长度** |
|  | **strPassWord** | **String** | **标签的访问密码，16进制字符串格式。** |
| **返回**  **(String)** | **成功返回读取的内容，不成功返回null。** | | |
| **参考代码** | **无** | | |

**3.9 readDataByTid【通过tid读取数据】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public String** **readDataByTid(String TIDStr, int Mem, int WordPtr, int Num, String Password)** | | |
| **说明** | **通过标签的tid来读取数据** | | |
| **参数** | **名称** | **类型** | **备注** |
| **TIDStr** | **String** | **标签的TID，16进制字符串** |
| **Mem** | **int** | **存储区，**  **0 - 密码区，前2个字是销毁密码，后2个字是访问密码**  **1 - EPC区**  **2 - TID区**  **3 - 用户区** |
| **WordPtr** | **int** | **待读取数据的起始地址** |
| **Num** | **int** | **需要读取数据的长度** |
| **Password** | **String** | **标签的访问密码，16进制字符串格式。** |
| **返回**  **String** | **成功返回读取的内容，不成功返回null。** | | |
| **参考代码** | **无** | | |

**3.10** **inventorySingle【单标签查询】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public int inventorySingle()** | | |
| **说明** | **每次调用只读取一个标签。** | | |
| **参数** | **名称** | **类型** | **备注** |
|  |  |  |
|  |  |  |  |
| **返回** | **成功：0；**  **失败：其他。**  **读取结果在 onInventoryTag() 回调返回对应结果。** | | |
| **参考代码** | **无。** | | |

**3.11 writeTag【写标签】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public int** **writeTag(String epc,**  **String strPassWord,**  **int btMemBank,**  **int btWordAdd,**  **String wData);** | | |
| **说明** | **写标签。** | | |
| **参数** | **名称** | **类型** | **备注** |
| **epc** | **String** | **标签的EPC号，16进制字符串，长度必须是4的整数倍** |
| **strPassWord** | **String** | **标签的访问密码，16进制字符串格式。** |
| **btMemBank** | **int** | **待写入的存储区，**  **0 - 密码区，前2个字是销毁密码，后2个字是访问密码**  **1 - EPC区**  **2 - TID区**  **3 - 用户区** |
| **btWordAdd** | **int** | **写入的起始字地址 写 EPC 区时，注意 EPC 从地址 02 开始，前两个字为 PC+CRC。** |
| **wData** | **String** | **待写入数据，长度必须是4的整数倍** |
| **返回**  **(int)** | **成功：0；**  **失败：其他；** | | |
| **参考代码** | **无** | | |

**3.12 writeTagByTid【通过tid向各存储区写数据】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义3** | **public int writeTagByTid(String TIDStr,int Mem,**  **int WordPtr,String Password,String wdata);** | | |
| **说明** | **向各存储区写数据.** | | |
| **参数** | **名称** | **类型** | **备注** |
| **TIDStr** | **String** | **标签的tid号，16进制字符串** |
| **Mem** | **int** | **待写入的存储区，**  **0 - 密码区，前2个字是销毁密码，后2个字是访问密码**  **1 - EPC区**  **2 - TID区**  **3 - 用户区** |
| **WordPtr** | **int** | **写入的起始字地址** |
| **Password** | **String** | **标签的访问密码，16进制字符串格式。** |
|  | **WData** | **String** | **待写入的数据，16进制字符串** |
| **返回**  **(int)** | **成功：0；**  **失败：其他；** | | |
| **参考代码** | **无** | | |

**3.13 writeEpc【广播方式随机改写一张标签的EPC号】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义1** | **public int WriteEPC(int epclen,String epc ,String Password);** | | |
| **说明** | **广播方式随机改写一张标签的EPC号** | | |
| **参数** | **名称** | **类型** | **备注** |
| **epclen** | **int** | **要写入的标签EPC号字长度(epc长度除以4)** |
| **epc** | **String** | **标签的EPC号，16进制字符串** |
| **Password** | **String** | **标签的访问密码，16进制字符串格式。** |
| **返回**  **(int)** | **成功：0；**  **失败：其他；** | | |
| **参考代码** | **无** | | |

**3.14 killTag【销毁标签】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public int killTag(String epc,String strPassWord);** | | |
| **说明** | **销毁标签（销毁密码不能为默认密码00000000，如果销毁密码是默认密码00000000,需要先修改成其他，再去销毁）。** | | |
| **参数** | **名称** | **类型** | **备注** |
| **epc** | **String** | **标签的epc号** |
| **strPassWord** | **String** | **标签销毁密码（销毁密码不能为默认密码00000000，如果销毁密码是默认密码00000000,需要先修改成其他，再去销毁）。** |
| **返回**  **(int)** | **成功：0；**  **失败：其他；** | | |
| **参考代码** | **无** | | |

**3.15 KillbyTID【通过TID销毁标签】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public int KillbyTID(int tidlen,String tid, String Password);** | | |
| **说明** | **销毁标签（销毁密码不能为默认密码00000000，如果销毁密码是默认密码00000000,需要先修改成其他，再去销毁）。** | | |
| **参数** | **名称** | **类型** | **备注** |
| **tidlen** | **int** | **TID长度** |
| **tid** | **String** | **TID值。** |
|  | **Password** | **String** | **标签销毁密码（销毁密码不能为默认密码00000000，如果销毁密码是默认密码00000000,需要先修改成其他，再去销毁）。** |
| **返回**  **(int)** | **成功：0；**  **失败：其他；** | | |
| **参考代码** | **无** | | |

**3.16 lockTag【锁定标签】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public int lockTag(String epc,String strPassWord,**  **Int btMemBank,int btLockType);** | | |
| **说明** | **用于锁定标签。** | | |
| **参数** | **名称** | **类型** | **备注** |
| **Epc** | **String** | **标签的EPC** |
| **btAryPassWord** | **String** | **标签的访问密码，4个字节** |
| **btMemBank** | **int** | **0 – 控制Kill密码读写保护设定。**  **1 – 控制访问密码读写保护设定。**  **2 – 控制EPC存储器读写保护设定。**  **3 – 控制TID存储器读写保护设定。**  **4 – 控制用户存储器读写保护设定。** |
| **btLockType** | **int** | **1个字节，**  **当Select为0或1，SetProtect值代表的意义如下:**  **0 – 设置为可读写**  **1 – 设置为永远可读写**  **2 – 设置为带密码可读写**  **3 – 设置为永远不可读写**  **当Select为2、3、4时，SetProtect值代表的意义如下:**  **0 – 设置为可写**  **1 – 设置为永远可写**  **2 – 设置为带密码可写**  **3 – 设置为永远不可写。** |
| **返回**  **(int)** | **成功：0；**  **失败：其他；** | | |
| **参考代码** | **无** | | |

**3.17 LockbyTID【通过TID锁定标签】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public int lockTag(int tidlen, String tid,int select,**  **int setprotect,String Password);** | | |
| **说明** | **用于锁定标签。** | | |
| **参数** | **名称** | **类型** | **备注** |
| **tidlen** | **int** | **标签的TID字长度(tid值除以4)** |
| **tid** | **String** | **输入，标签的TID** |
| **select** | **int** | **1个字节，**  **0 – 控制Kill密码读写保护设定。**  **1 – 控制访问密码读写保护设定。**  **2 – 控制EPC存储器读写保护设定。**  **3 – 控制TID存储器读写保护设定。**  **4 – 控制用户存储器读写保护设定。**  **其它值保留，若出读写器接收到了其他值，将返回参数出错的消息。** |
| **setprotect** | **int** | **1个字节，**  **当Select为0或1，SetProtect值代表的意义如下:**  **0 – 设置为可读写**  **1 – 设置为永远可读写**  **2 – 设置为带密码可读写**  **3 – 设置为永远不可读写**  **当Select为2、3、4时，SetProtect值代表的意义如下:**  **0 – 设置为可写**  **1 – 设置为永远可写**  **2 – 设置为带密码可写**  **3 – 设置为永远不可写。** |
|  | **Password** | **String** | **标签的访问密码，4个字节** |
| **返回**  **(int)** | **成功：0；**  **失败：其他；** | | |
| **参考代码** | **无** | | |

**3.18 setQueryMode【设置查询模式】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public void setQueryMode(int mode)** | | |
| **说明** | **可以通过三种方式来查询标签** | | |
| **参数** | **名称** | **类型** | **备注** |
| **Mode** | **Int** | **0：epc**  **1：epc+tid** |
| **返回**  **(void)** |  | | |
| **参考代码** | **注意：**  **如果要读 EPC + USER 区，则不用该方法，使用如下方法设置：**  **RfidParameter parameter = RFIDSDKManager.getInstance().getRfidManager().getInventoryParameter(); parameter.IvtType = 1; parameter.Memory = 3;//读取USER区 parameter.WordPtr = 0;//读取USER的起始字地址 parameter.Length = 6;//读取USER的字长度  RFIDSDKManager.getInstance().getRfidManager().setInventoryParameter(parameter);** | | |

**3.19 getQueryMode【获取查询模式】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public int getQueryMode()** | | |
| **说明** | **获取设备的查询模式** | | |
| **参数** | **名称** | **类型** | **备注** |
| **返回**  **(int)** | **对应设置模式的值**  **0：epc**  **1：epc+tid**  **2：epc+user**  **3：fasttid** | | |
| **参考代码** | **无** | | |

3.20 getFirmwareVersion【获取固件版本】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public String getFirmwareVersion(); | | |
| **说明** | 获取读卡器固件版本。 | | |
| **参数** | **名称** | **类型** | **备注** |
| **返回**  **(String)** | 返回固件版本信息 | | |
| **参考代码** | 无 | | |

**3.21 setOutputPower【设置输出功率】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public int setOutputPower(int btOutputPower);** | | |
| **说明** | **设置功率** | | |
| **参数** | **名称** | **类型** | **备注** |
| **btOutputPower** | **int** | **射频输出功率：**  **长距功率范围：0-33**  **短距版功率范围：0-26** |
| **返回**  **(int)** | **成功：0；失败：other；** | | |
| **参考代码** | **无** | | |

3.22 getOutputPower【查询输出功率】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public int getOutputPower(); | | |
| **说明** | 查询输出功率。 | | |
| **参数** | **名称** | **类型** | **备注** |
| **返回**  **(int)** | 返回输出功率(小于0则获取失败) | | |
| **参考代码** | 无 | | |

3.23 setFrequencyRegion【设置频率区域】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public int setFrequencyRegion(int btRegion,int btStartRegion,  int btEndRegion); | | |
| **说明** | 设置频率区域（读写器默认支持的频率）。 | | |
| **参数** | **名称** | **类型** | **备注** |
| btRegion | int | **1个字节，频谱频段。**  **1 - "Chinese band2";**  **2 - "US band";**  **3 - "Korean band";**  **8 - "Chinese band1";**  **9 - "EU band";**  **详见：（附录2：频段说明）** |
| btStartRegion | int | **频谱的起始频点。** |
| btEndRegion | int | **频谱的终止频点**  **设置射频输出频谱的范围。**  **规则是：**  **1、起始频率和终止频率应在规定的规定范围内。**  **2、起始频率应等于或低于终止频率。**  **3、结束频率等于起始频率，即使用单频点。** |
| **返回**  **(int)** | 成功：0；失败：other； | | |
| **参考代码** | 无 | | |

**3.24 getFrequencyRegion【获取频率区域】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public RfidDate getFrequencyRegion();** | | |
| **说明** | **获取频率区域。** | | |
| **参数** | **名称** | **类型** | **备注** |
| **返回**  **(RfidDate )** | **RfidDate rfidDate = mRFIDManager.getFrequencyRegion(); rfidDate.getbtRegion();//频谱频段 rfidDate.getbtFrequencyStart();//频谱的起始频点。 rfidDate.getbtFrequencyEnd();//频谱的终止频点**  **(rfidDate 如果为null，则获取失败)** | | |
| **参考代码** | **无** | | |

**3.25 addMask【添加掩码】（短距不支持）**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public void addMask(int mem,int startAddress,int len,String data)** | | |
| **说明** | **添加盘点时候的过滤参数** | | |
| **参数** | **名称** | **类型** | **备注** |
| **mem** | **int** | **过滤区域：**  **0 - 密码区，前2个字是销毁密码，后2个字是访问密码**  **1 - EPC区**  **2 - TID区**  **3 - 用户区** |
| **startAddress** | **int** | **起始地址** |
| **len** | **int** | **过滤长度** |
| **data** | **String** | **过滤数据（偶数长度）** |
| **返回**  **Void** |  | | |
| **参考代码** | 例子:  标签内容 :  Epc:123456789  Tid:234567891  User:345678912  如果要过滤从第二个字母开始的TID区域，则需要使用4567过滤数据。您应该设置:  mem:2,startAddress:2,len：4，data:4567.  startAddress：字符在区域中的位置(位置从0开始)。  len： 字符个数  The code:mRfidManager.addMask(2,2,4,”4567”) | | |

**3.26 clearMask【清除掩码】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public void clearMask();** | | |
| **说明** | **清除掩码数据** | | |
| **参数** | **名称** | **类型** | **备注** |
| **Void** |  | | |
| **参考代码** | **无** | | |

**3.27 unregisterCallback【注销回调】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public void unregisterCallback(IRfidCallback cb);** | | |
| **说明** | **注销回调。** | | |
| **参数** | **名称** | **类型** | **备注** |
| **cb** | **IRfidCallback** | **回调** |
| **返回**  **(void)** | **无** | | |
| **参考代码** | **无** | | |

3.28 getModuleFirmware【获取模块固件信息】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public String getModuleFirmware(); | | |
| **说明** | 获取模块固件信息 | | |
| **参数** | **名称** | **类型** | **备注** |
| **返回**  **(String)** | 模块信息 | | |
| **参考代码** | 无 | | |

3.29 setScanInterval【设置盘存间隔时间】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public int setScanInterval( int interval) | | |
| **说明** | 设置启用盘点后使用的询查参数 | | |
| **参数** | **名称** | **类型** | **备注** |
| interval | int | 参数为 0-6；  分别对应：  0-0ms  1-10ms  2-20ms  3-30ms  4-40ms  5-50ms  6-60ms  即 interval\*10ms为实际间隔时间，单位（毫秒） |
| **返回**  **(int)** | 成功：0；  失败：其他。 | | |
| **参考代码** | 无 | | |

3.30 getScanInterval【读取盘存间隔时间】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public int getScanInterval() | | |
| **说明** | 读取启用盘点后使用的询查参数 | | |
| **参数** | **名称** | **类型** | **备注** |
|  |  |  |
| **返回**  **(**int**)** | 返回0-6，为获取到的间隔时间值 \*10ms 得到实际的间隔时间，单位 ms；  返回-1，调用方法失败 | | |
| **参考代码** | 无 | | |

3.31 startRead【启动盘点】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public int startRead() | | |
| **说明** | 启动盘点 | | |
| **参数** | **名称** | **类型** | **备注** |
| **返回**  **(int)** | 成功：0；  失败：其他。 | | |
| **参考代码** | 无 | | |

3.32 scanRfid【询查标签】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public void scanRfid() | | |
| **说明** | 开启盘点后一段时间自动停止 | | |
| **参数** | **名称** | **类型** | **备注** |
| **返回**  **(void)** |  | | |
| **参考代码** | 无 | | |

3.33 setInventoryParameter【设置盘点参数】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public void setInventoryParameter(Rfidparameter parameter) | | |
| **说明** | 设置盘点参数 | | |
| **参数** | **名称** | **类型** | **备注** |
| **返回**  **(void)** |  | | |
| **参考代码** | 无 | | |

3.34 getDeviceId【获取序列号】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public String getDeviceId() | | |
| **说明** | 获取序列号 | | |
| **参数** | **名称** | **类型** | **备注** |
| **返回**  **(String)** | 序列号rfid的序列号。 | | |
| **参考代码** | 无 | | |

3.35 setCustomRegion【设置自定义频段】（国标无效）

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public int setCustomRegion(int flags, int FreSpace, int FreNum, int StartFre); | | |
| **说明** | 设置自定义频段 | | |
| **参数** | **名称** | **类型** | **备注** |
|  | flags | int | **下电是否保存设置信息标识**  **0：保存**  **1：不保存** |
| int | FreSpace | **频段间隔** |
| int | FreNum | **频点数量** |
| int | StartFre | **起始频段** |
| 序列号rfid的序列号。  0:成功  其他:失败 |  |  |
| **返回**  **(int)** | 成功：0；  失败：其他。 | | |
| **参考代码** |  | | |

3.36 getCustomRegion【获取自定义频段】（国标无效）

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public CustomRegionBean getCustomRegion(); | | |
| **说明** | 获取自定义频段信息，如果设置自定义频段后需用该方法来获取相关设置信息，调用getFrequencyRegion无法获取自定频段的相关信息。 | | |
| **参数** | **名称** | **类型** | **备注** |
| **返回**  **(**CustomRegionBean **)** | 序列号rfid的序列号。  CustomRegionBean 实例:成功  Null:失败  public class CustomRegionBean {  public int [] band=new int[1];//保留信息  public int [] FreSpace=new int[1];//频段间隔  public int [] FreNum=new int[1];//频点数量  public int [] StartFre=new int[1];//起始频点  public CustomRegionBean(){}  } | | |
| **参考代码** | 无 | | |

3.37 setProfile【设置链路配置信息】

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public int setProfile(int param); | | |
| **说明** | 设置链路信息 | | |
| **参数** | **名称** | **类型** | **备注** |
| param | int | **一般这个参数的可选项需要根据设备信息来提供。**  **如果读取器的类型是**  **（0x21,0x23,0x28,0x36,0x37）**  **parm所代表的是：**  **0:40K, FM0,25us**  **1:250K,M4, 25us**  **2:300K,M4, 25us**  **3:400K,FM0,6.25us**  **如果读取器的类型不是上面那些类型**  **则param可供选项为一下**  **11:640K,FM0,7.5u**  **1:640K, M2,7.5us**  **15:640K, M4,7.5us**  **12:320K, M2, 15us**  **3:320K, M2, 20us**  **5:320K, M4, 20us**  **7:250K, M4, 20us**  **13:160K, M8, 20us** |
| **返回（int）** | 成功：0；  失败：其他； | | |
| **参考代码** | 无 | | |

**3.38 GetReaderType【获取模块类型】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | public int GetReaderType() | | |
| **说明** | 获取模块类型 | | |
| **参数** | **名称** | **类型** | **备注** |
| **返回**  **(int)** | 对应设置模式的值  80：短距  其他：长距  （如果小于0，则获取失败） | | |
| **参考代码** | 无 | | |

**3.39 getReaderTemperature【查询内部温度】**

|  |  |  |  |
| --- | --- | --- | --- |
| 定义 | public String getReaderTemperature(); | | |
| 说明 | 查询内部温度。 | | |
| 参数 | 名称 | 类型 | 备注 |
| 返回  (String) | 返回字符串格式的温度值，单位（摄氏度） | | |
| 参考代码 | 无 | | |

**3.40 startInventoryLed【询查ELD标签】**

|  |  |  |  |
| --- | --- | --- | --- |
| 定义 | public int startInventoryLed(int manufacturers, List<String> epcs); | | |
| 说明 | 询查ELD标签 | | |
| 参数 | 名称 | 类型 | 备注 |
| manufacturers | int | 厂商（0或1） |
| epcs | List<String> | 需要询查的epc |
| 返回  (int) | 成功：0；  失败：其他。 | | |
| 参考代码 | 无 | | |

**3.41 stopInventoryLed【停止询查ELD标签】**

|  |  |  |  |
| --- | --- | --- | --- |
| 定义 | public void stopInventoryLed(); | | |
| 说明 | 停止询查ELD标签 | | |
| 参数 | 名称 | 类型 | 备注 |
|  |  |  |
| 返回  (void) |  | | |
| 参考代码 | 无 | | |

**3.42 enableLog【开启或者关闭日志】**

|  |  |  |  |
| --- | --- | --- | --- |
| 定义 | public void enableLog(boolean enable); | | |
| 说明 | 开启或者关闭日志。 | | |
| 参数 | 名称 | 类型 | 备注 |
| enable | boolean | true:开启日志；false:关闭日志 |
| 返回  (void) |  | | |
| 参考代码 | 无 | | |

**3.43 writeTagEpc【写标签EPC区】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public int writeTagEpc(String selectEpc, String password, String writeData**  **);** | | |
| **说明** | **修改标签的EPC值，通过原EPC值改写成写入的EPC值，即把标签EPC号改成writeData。** | | |
| **参数** | **名称** | **类型** | **备注** |
| **selectEpc** | **String** | **选中的标签的epc** |
| **password** | **String** | **进入密码4字节** |
| **writeData** | **String** | **待写入数据（16进制，长度4的倍数）** |
| **返回**  **(int)** | **成功：0；**  **失败：其他；** | | |
| **参考代码** | **无** | | |

**3.44 writeTagEpcByTid【通过tid写EPC区】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public int writeTagByTid(String TIDStr, String password, String writeData**  **);** | | |
| **说明** | **修改标签的EPC值，通过TID值,把标签当前的EPC值改写成写入的EPC值，即把标签EPC号改成writeData。** | | |
| **参数** | **名称** | **类型** | **备注** |
| **TIDStr** | **String** | **标签的tid号** |
| **Password** | **String** | **标签的访问密码，16进制字符串格式。** |
|  | **WData** | **String** | **待写入的数据，16进制字符串，长度必须是4的整数倍** |
| **返回**  **(int)** | **成功：0；**  **失败：其他；** | | |
| **参考代码** | **无** | | |

**3.45 findEpc【查找标签】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public Tag6C findEpc(String selectEpc);** | | |
| **说明** | **查找、定位指定的标签** | | |
| **参数** | **名称** | **类型** | **备注** |
| **selectEpc** | **String** | **标签的epc号** |
| **返回**  **(int)** | **成功返回Tag6C 不为空。**  **通过 Tag6C 的rssi 字段，来判断信号的强弱，以此来判断是否靠近标签，进行定位判断。** | | |
| **参考代码** | **无** | | |

**3.46 power【模块上电、下电】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public void power(boolean enable** **, (ret) => {});** | | |
| **说明** | **对RFID模块进行上电下电的操作.**  如果需要上电，在设置为true之后，需要等待1500ms才能操作连接模块**3.47 connect()** 方法。因为上电需要1500ms的时间。 | | |
| **参数** | **名称** | **类型** | **备注** |
| **enable** | **boolean** | **true：上电 false:下电** |
| **返回**  **(ret)s** | **ret： true 上、下电成功； false 上、下电失败。** | | |
| **参考代码** | **无** | | |

**3.47 connect【连接RFID模块】**

|  |  |  |  |
| --- | --- | --- | --- |
| **定义** | **public boolean connect();** | | |
| **说明** | **连接RFID模块** | | |
| **参数** | **名称** | **类型** | **备注** |
|  |  |  |
| **返回**  **(boolean)** | **true:连接成功**  **false:连接失败** | | |
| **参考代码** | **无** | | |

附录1：**常见错误码**

|  |  |
| --- | --- |
| **错误代码** | **描述** |
| 0 | **执行成功。** |
| 1 | **未询查到电子标签。** |
| 2 | **电源不足，标签电源不足，无法执行存储写入操作。** |
| 3 | **存储器超限或不被支持的PC值，规定存储位置不存在或标**  **签不支持PC值。** |
| 4 | **存储器锁定，存储位置锁定永久锁定，且不可写入。** |
| 5 | **访问密码错误。** |
| 9 | **销毁密码错误。** |
| 10 | **销毁密码不能为全0。** |
| 11 | **电子标签不支持该命令。** |
| 12 | **对该命令，访问密码不能为0。** |
| 13 | **电子标签已经被设置了读保护，不能再次设置。** |
| 14 | **电子标签没有被设置读保护，不需要解锁。** |
| 15 | **非特定错误，标签不支持特定错误代码。** |
| 16 | **有字节空间被锁定，写入失败。** |
| 17 | **不能锁定。** |
| 18 | **已经锁定，不能再次锁定。** |
| 19 | **参数保存失败,但设置的值在读写模块断电前有效。** |
| 20 | **无法调整。** |
| 248 | **天线检测错误** |
| 249 | **命令执行出错。** |
| 250 | **有电子标签，但通信不畅，无法操作。** |
| 251 | **无电子标签可操作。** |
| 252 | **电子标签返回错误代码。** |
| 253 | **命令长度错误。** |
| 254 | **不合法的命令。** |
| 255 | **参数错误。** |
| 48 | **通讯错误。** |

**附录2：频段说明**

欧标频段（865~868MHz）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 频段名称 | 频点个数 | 频道间隔  kHz | 频点公式  MHz | 区域代码（十六进制） | 范围（MHZ） |
| Ukraine | 7 | 100 | 868.0 + N\*0.1(MHz) 其中N∈[0, 6] | 6 | 868.0-868.6 |
| Chinese 1 | 20 | 250 | 840.125 + N \* 0.25 (MHz) 其中N∈[0, 19] | 8 | 840.125-844.875 |
| EU | 4 | 600 | 865.7 + N \* 0.6(MHz) 其中N∈[0, 3] | 9 | 865.7-867.5 |

美标频段（902~928MHz）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 频段名称 | 频点个数 | 频道间隔  kHz | 频点公式  MHz | 区域代码（十六进制） | 范围（MHZ） |
| Chinese 2 | 20 | 250 | 920.125 + N \* 0.25 (MHz) 其中N∈[0, 19] | 1 | 920.125-924.875 |
| US | 50 | 500 | 902.75 + N \* 0.5 (MHz) 其中N∈[0,49] | 2 | 902.75-927.25 |
| Korean | 32 | 200 | 917.1 + N \* 0.2 (MHz) 其中N∈[0, 31] | 3 | 917.1-923.7 |
| Peru | 12 | 900 | 916.2 + N\*0.9(MHz) 其中N∈[0, 11] | 7 | 916.2-926.1 |
| US 3 | 53 | 500 | 902 + N \* 0.5 (MHz) 其中N∈[0,52] | C | 902-928 |
| Taiwan | 12 | 500 | 922.25 + N \* 0.5 (MHz) 其中N∈[0,11] | A | 922.25-927.75 |

设置这些特定区域频段的示例:

如欧标：（从上到下依次对应参数0-14）

865.1 0

865.3 1

865.5 2

865.7 3

865.9 4

866.1 5

866.3 6

866.5 7

866.7 8

866.9 9

867.1 10

867.3 11

867.5 12

867.7 13

867.9 14

如果需要设置欧标 865.3-867.5

mRfidManager.setFrequencyRegion(4, 2, 12);

获取通过上述方法设置的频段信息。

RfidDate frequencyRegion = mRfidManager.getFrequencyRegion();

frequencyRegion.btRegion:区域（4）

frequencyRegion.btFrequencyStart:起始频点（2）

frequencyRegion.btFrequencyEnd:截止频点（12）

**附录3: RSSI参数对照表**

|  |  |  |  |
| --- | --- | --- | --- |
| **RSSI参数** | **信号强度** | **RSSI参数** | **信号强度** |
| 98(0x62) | -31dBm | 64(0x40) | -66dBm |
| 97(0x61) | -32dBm | 63(0x3F) | -67dBm |
| 96(0x60) | -33dBm | 62(0x3E) | -68dBm |
| 95(0x5F) | -34dBm | 61(0x3D) | -69dBm |
| 94(0x5E) | -35dBm | 60(0x3C) | -70dBm |
| 93(0x5D) | -36dBm | 59(0x3B) | -71dBm |
| 92(0x5C) | -37dBm | 58(0x3A) | -72dBm |
| 91(0x5B) | -38dBm | 57(0x39) | -73dBm |
| 90(0x5A) | -39dBm | 56(0x38) | -74dBm |
| 89(0x59) | -41dBm | 55(0x37) | -75dBm |
| 88(0x58) | -42dBm | 54(0x36) | -76dBm |
| 87(0x57) | -43dBm | 53(0x35) | -77dBm |
| 86(0x56) | -44dBm | 52(0x34) | -78dBm |
| 85(0x55) | -45dBm | 51(0x33) | -79dBm |
| 84(0x54) | -46dBm | 50(0x32) | -80dBm |
| 83(0x53) | -47dBm | 49(0x31) | -81dBm |
| 82(0x52) | -48dBm | 48(0x30) | -82dBm |
| 81(0x51) | -49dBm | 47(0x2F) | -83dBm |
| 80(0x50) | -50dBm | 46(0x2E) | -84dBm |
| 79(0x4F) | -51dBm | 45(0x2D) | -85dBm |
| 78(0x4E) | -52dBm | 44(0x2C) | -86dBm |
| 77(0x4D) | -53dBm | 43(0x2B) | -87dBm |
| 76(0x4C) | -54dBm | 42(0x2A) | -88dBm |
| 75(0x4B) | -55dBm | 41(0x29) | -89dBm |
| 74(0x4A) | -56dBm | 40(0x28) | -90dBm |
| 73(0x49) | -57dBm | 39(0x27) | -91dBm |
| 72(0x48) | -58dBm | 38(0x26) | -92dBm |
| 71(0x47) | -59dBm | 37(0x25) | -93dBm |
| 70(0x46) | -60dBm | 36(0x24) | -94dBm |
| 69(0x45) | -61dBm | 35(0x23) | -95dBm |
| 68(0x44) | -62dBm | 34(0x22) | -96dBm |
| 67(0x43) | -63dBm | 33(0x21) | -97dBm |
| 66(0x42) | -64dBm | 32(0x20) | -98dBm |
| 65(0x41) | -55dBm | 31(0x1F) | -99dBm |