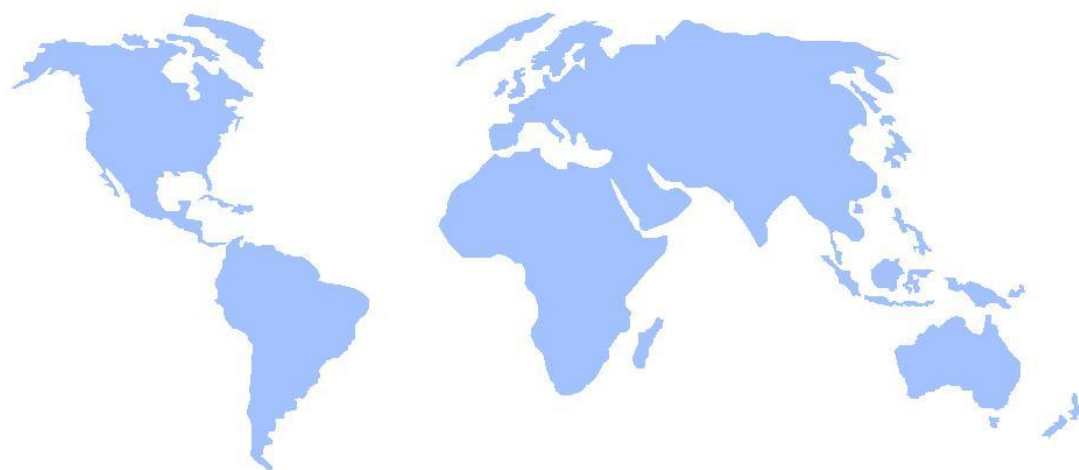




Android UHF 接口 使用说明书



www.seuic.com

修订记录

版本号	修订日期	修订内容	修订人员
1.0	2016/1/9	建立本文档。	冯晓东
1.1	2019/02/01	增加 blockwrite、blockerase 接口。	冯晓东
1.2	2019/08/05	增加 isopen 接口、getInstance(Context)。	冯晓东
1.3	2020/07/01	增加 kill 接口、setParameters 接口、setParamBytes 接口。	冯晓东
1.4	2020/11/25	更改 isopen 为 isOpen，增加系统休眠 uhf 模块会断电说明。	冯晓东
1.5	2021/04/26	增加温度、LED 标签接口	冯晓东
1.6	2021/05/24	增加回调方式实时返回 EPC	冯晓东

目录

概述.....	4
硬件平台.....	4
软件平台.....	4
适用对象.....	4
引用方法.....	4
UHF 接口.....	5
UHFSERVICE 类.....	5
使用方法.....	5
函数接口.....	5
1. 打开 UHF.....	6
2. 关闭 UHF.....	6
3. 是否打开 UHF.....	6
4. 获取固件版本号.....	6
5. 获取温度.....	7
6. 获取功率.....	7
7. 设置功率.....	7
8. 获取区域.....	7
9. 设置区域.....	7
10. 单次寻卡.....	8
11. 开始连续寻卡.....	8
12. 停止开始寻卡.....	8
13. 获取连续寻卡标签数量.....	8
14. 获取连续寻卡标签 ID.....	9
15. 读标签.....	9
16. 写标签.....	10
17. 块写标签.....	11
18. 块擦除标签.....	12
19. 锁标签.....	12
20. 销毁标签.....	13
21. 设置参数.....	13
22. 设置参数 bytes (新模块支持).....	13

23. 读 LED 标签.....	13
24. 读温度标签.....	14
25. 回调函数.....	14
26. 注册回调函数.....	15
27. 卸载回调函数.....	15
EPC 类.....	15
类成员变量.....	15
使用方法.....	15
函数接口.....	16
1. 获取 EPC 对象中的 id 数据.....	16
2. 比较 id 号是否相同.....	16
3. 获取附加数据（新模块支持）.....	16

概述

本文档介绍了 android 上提供的 UHF 设备相关的接口的使用说明, 以便帮助用户更好的使用终端产品提供的特殊功能。

硬件平台

SDK 适用于以下终端设备:

- D500、D501H、D502H、R109 等

具体使用范围在函数接口说明中指明, 若无指明表示都支持。

软件平台

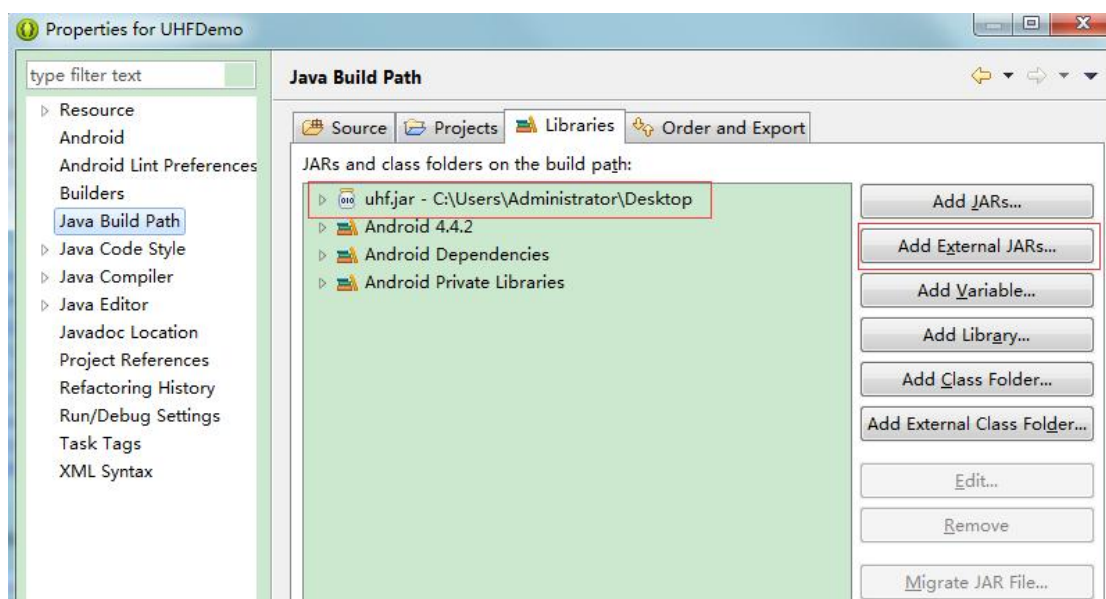
SDK 基于 Android 4.4 以上版本, 支持 eclipse、Android Studio 开发工具。

适用对象

除了 Android 的标准功能外, 希望用到 UHF 模块的开发人员。

引用方法

1. 以下以 Eclipse 为例, 右击项目 - Properties - Java Build Path - Libraries - Add External JARs, 选择 uhf.jar 文件



2. AndroidManifest.xml 中使用 uses-library 用包名来标示 uhf.jar 是引用包。

```
<application
.....
    <uses-library
        android:name="com.seuic.uhf" />
    .....
</application>
```

UHF 接口

包名	com.seuic.uhf
包文件	uhf.jar
系统包	是
包含的类	UHFSERVICE
作用	提供 UHF 相关的控制接口

UHFSERVICE 类

使用方法

```
import com.seuic.uhf.UHFSERVICE;
UHFSERVICE mDevice = UHFSERVICE.getInstance();

//如果希望系统 UHFDemo 服务退出, 则使用带参数的 getInstance 接口。
UHFSERVICE mDevice = UHFSERVICE.getInstance(Context context);
```

注意: 系统休眠时 UHF 模块会断电, 因此系统休眠前需要调用 close 接口, 当系统唤醒后需要调用 open 接口重新打开模块。

函数接口

函数	说明
open	打开 UHF
close	关闭 UHF
isOpen	是否打开 UHF
getFirmwareVersion	获取固件版本号
getTemperature	获取温度
getPower	获取功率
setPower	设置功率
getRegion	获取区域
setRegion	设置区域
inventoryOnce	单次寻卡

inventoryStart	开始连续寻卡
inventoryStop	停止连续寻卡
getTagIDCount	获取连续寻卡标签数量
getTagIDs	获取连续寻卡标签 ID
readTagData	读标签
writeTagData	写标签
blockWriteTagData	块写标签
blockEraseTagData	块擦除标签
killTag	销毁标签
setParameters	设置参数
setParamBytes	设置参数 bytes
readTagLED	读 LED 标签
readTagTemperature	读温度标签
registerReadTags	注册回调函数
unregisterReadTags	卸载回调函数

1. 打开 UHF

`boolean open ()`

参数

无

返回值

boolean; true:成功、false:失败

2. 关闭 UHF

`void close ()`

参数

无

返回值

无

3. 是否打开 UHF

`boolean isOpen ()`

参数

无

返回值

boolean; true:打开、false:关闭

4. 获取固件版本号

`String getFirmwareVersion ()`

参数

无

返回值

String; 非 null:成功、null:失败

5. 获取温度

`String getTemperature ()`

参数

无

返回值

String; 非 null:成功、null:失败 (°C)

6. 获取功率

`int getPower ()`

参数

无

返回值

int; 非 0:成功、0:失败 (5~33 dBm)

7. 设置功率

`boolean setPower (int power)`

参数

power

设置的功率值 (5~33 dBm)

返回值

boolean; true:成功、false:失败

8. 获取区域

`String getRegion ()`

参数

无

返回值

String; 非空:成功、null:失败

区域值包括: "FCC" , "ETSI" , "China1" , "China2"

9. 设置区域

`boolean setRegion (String region)`

参数

region

区域值包括: "FCC" , "ETSI"

返回值

boolean; true:成功、false:失败

10. 单次寻卡

boolean inventoryOnce(EPC epc, int timeout)

参数

epc

EPC 类对象

timeout

命令超时时间 (0~500 ms)

返回值

boolean; true:成功、false:失败

示例:

```
EPC epc = new EPC();  
boolean result = mService.inventoryOnce(epc, 100);
```

11. 开始连续寻卡

boolean inventoryStart ()

参数

无

返回值

boolean; true:成功、false:失败

注: 通过 `getTagIDs` 获取寻卡数据, 与 `inventoryStop()`接口配对使用。

示例:

```
boolean result = mService.inventoryStart();
```

12. 停止开始寻卡

boolean inventoryStop ()

参数

无

返回值

boolean; true:成功、false:失败

示例:

```
Boolean result = mService.inventoryStop();
```

13. 获取连续寻卡标签数量

int getTagIDCount ()

参数

无

返回值

int; 非 0:获取的标签数量、0:未获取到标签

14. 获取连续寻卡标签 ID

`List<EPC> getTagIDs ()`

参数

无

返回值

`List<EPC>`; EPC 对象的 List 集合

注: 返回的 List 集合是已经排重处理的。

示例:

```
List<EPC> mAllEpcs = new ArrayList<EPC>();
class InventoryThread extends Thread {
    @Override
    public void run() {
        while (mInventoryStart) {
            try {
                sleep(100);
                mAllEpcs = mService.getTagIDs();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

15. 读标签

`boolean readTagData (byte[] Epc, byte[] PassWord, int Bank, int Offset ,int Len, byte[] Data)`

参数

Epc

标签的 ID (PC+EPC) //请使用 EPC 的 `getId()`方法获取 `epc`

PassWord

访问密码 (长度为 4 个字节)

Bank

标签的存储区 (0: 密码区 1: EPC 区 2: TI 区 3: 用户区)

Offset

起始地址 (单位为字节)

Len

要读取的数据长度, 单位为字节

Data

存放读到的数据

返回值

`boolean`; `true`:成功、`false`:失败

示例:

```

String epclD = "1122334455667788";
String psw = "12345678";
byte[] data = new byte[256];
boolean result = mService.readTagData(getHexByteArray(epclD),
    getHexByteArray(psw),
    Integer.parseInt(bank),
    Integer.parseInt(offset),
    Integer.parseInt(len), data);

/**
 * 字符串 转 十六进制形式的数组
 */
public static byte[] getHexByteArray(String hexString) {
    byte[] buffer = new byte[hexString.length() / 2];
    if (hexString == null || hexString.equals("")) {
        return null;
    }
    hexString = hexString.toUpperCase();
    int length = hexString.length() / 2;
    char[] hexChars = hexString.toCharArray();
    for (int i = 0; i < length; i++) {
        int pos = i * 2;
        buffer[i] = (byte) (charToByte(hexChars[pos]) << 4 | charToByte(hexChars[pos +
1]));
    }
    return buffer;
}
}

```

16. 写标签

`boolean writeTagData (byte[] Epc, byte[] PassWord, int Bank, int Offset ,int Len, byte[] Data)`

参数

Epc

标签的 ID (PC+EPC) //请使用 EPC 的 `getId()`方法获取 `epc`

PassWord

访问密码 (长度为 4 个字节)

Bank

标签的存储区 (0: 密码区 1: EPC 区 2: TI 区 3: 用户区)

Offset

起始地址 (单位为字节)

Len

要写入的数据长度, 单位为字节

Data

存放准备要写入标签的数据

返回值

boolean; true:成功、false:失败

示例:

```
String epcID = "1122334455667788";
String psw = "12345678";
String data = "11223344";
boolean result = mService.writeTagData(getHexByteArray(epcID),
    getHexByteArray(psw),
    Integer.parseInt(bank),
    Integer.parseInt(offset),
    Integer.parseInt(len),
    getHexByteArray(data))
```

17. 块写标签

boolean blockWriteTagData (byte[] Epc, byte[] PassWord, int Bank, int Offset ,int Len, byte[] Data)

参数

Epc

标签的 ID (PC+EPC) //请使用 EPC 的 getId()方法获取 epc

PassWord

访问密码 (长度为 4 个字节)

Bank

标签的存储区 (0: 密码区 1: EPC 区 2: TI 区 3: 用户区)

Offset

起始地址 (单位为字节)

Len

要写入的数据长度, 单位为字节

Data

存放准备要块写入标签的数据

返回值

boolean; true:成功、false:失败

示例:

```
String epcID = "1122334455667788";
String psw = "12345678";
String data = "11223344";
boolean result = mService.blockWriteTagData(getHexByteArray(epcID),
    getHexByteArray(psw),
    Integer.parseInt(bank),
    Integer.parseInt(offset),
    Integer.parseInt(len),
    getHexByteArray(data))
```

18. 块擦除标签

`boolean blockEraseTagData (byte[] Epc, byte[] PassWord, int Bank, int Offset ,int Len)`

参数

Epc

标签的 ID (PC+EPC) //请使用 EPC 的 `getId()`方法获取 `epc`

PassWord

访问密码 (长度为 4 个字节)

Bank

标签的存储区 (0: 密码区 1: EPC 区 2: TI 区 3: 用户区)

Offset

起始地址 (单位为字节)

Len

要擦除的数据长度, 单位为字节

返回值

`boolean`; `true`:成功、`false`:失败

示例:

```
String epcID = "1122334455667788";
String psw = "12345678";
boolean result = mService.blockEraseTagData(getHexByteArray(epcID),
    getHexByteArray(psw),
    Integer.parseInt(bank),
    Integer.parseInt(offset),
    Integer.parseInt(len))
```

19. 锁标签

`boolean lockTag(byte[] Epc, byte[] PassWord, int Flag)`

参数

Epc

标签的 ID (PC+EPC) //请使用 EPC 的 `getId()`方法获取 `epc`

PassWord

访问密码 (长度为 4 个字节)

Flag

0,1,2,3 解锁标签 (0: 密码区 1: EPC 区 2: TI 区 3: 用户区)

10,11,12,13 锁标签 (10: 密码区 11: EPC 区 12: TI 区 13: 用户区)

20,21,22,23 永久锁标签 (20: 密码区 21: EPC 区 22: TI 区 23: 用户区)

返回值

`boolean`; `true`:成功、`false`:失败

示例:

```
String epcID = "1122334455667788";
String psw = "12345678";
mLockFlag = 13;//锁 user 区
```

```
boolean result = mService.lockTag(getHexByteArray(epcID),
    getHexByteArray(psw),
    mLockFlag)
```

20. 销毁标签

boolean killTag(byte[] Epc, byte[] KillPwd)

参数

Epc

标签的 ID (PC+EPC) //请使用 EPC 的 `getId()`方法获取 `epc`

KillPwd

销毁密码 (长度为 4 个字节) (销毁密码必须非 0)

返回值

boolean; true:成功、false:失败

示例:

```
String epcID = "1122334455667788";
String psw = "12345678";
boolean result = mService.killTag(getHexByteArray(epcID),
    getHexByteArray(psw))
```

21. 设置参数

boolean setParameters(int id, int value)

参数

参数说明参考文档: [Android UHF setParameters 接口使用说明书.doc](#)

返回值

boolean; true:成功、false:失败

示例:

```
int setsession = 1;
mService.setParameters(UHFService.PARAMETER_INVENTORY_SESSION,
    setsession);
```

22. 设置参数 bytes (新模块支持)

boolean setParamBytes(int id, byte[] value)

参数

参数说明参考文档: [Android UHF setParameters 接口使用说明书.doc](#)

返回值

boolean; true:成功、false:失败

23. 读 LED 标签

boolean readTagLED(byte[] Epc, byte[] PassWord, int Manufacturer)

参数

Epc

标签的 ID (PC+EPC) (如果不指定则设置 null) //请使用 EPC 的 getId()方法
获取 epc

PassWord

访问密码 (长度为 4 个字节, 若无密码, 默认 null)

Manufacturer

标签的制造商 (1: 宜链)

返回值

boolean; true:成功、false:失败

示例:

```
String epcID = "1122334455667788";  
String psw = "12345678";  
mService.readTagLED(BaseUtil.getHexByteArray(epcID), psw, 1);
```

24. 读温度标签

`double readTagTemperature(byte[] Epc, byte[] PassWord, int Manufacturer)`

参数

Epc

标签的 ID (PC+EPC) (如果不指定则设置 null) //请使用 EPC 的 getId()方法
获取 epc

PassWord

访问密码 (长度为 4 个字节, 若无密码, 默认 null)

Manufacturer

标签的制造商 (1: 宜链 , 2: 悦和)

返回值

double; -100:失败(°C)

示例:

```
String epcID = "1122334455667788";  
String psw = "12345678";  
mService.readTagTemperature(BaseUtil.getHexByteArray(epcID), psw, 1);
```

25. 回调函数

```
interface IReadTagsListener{  
    void tagsRead(List<EPC> epclist);  
}
```

参数

epclist

EPC 集合的 list, 返回最新盘存到的标签 EPC。

返回值

无

示例:

```
private IReadTagsListener mIReadTagsListener = new IReadTagsListener(){  
    @Override
```

```
public void tagsRead(List<EPC> epcList){
    mEPCList.addAll(epcList);
}
};
```

26. 注册回调函数

`void registerReadTags (IReadTagsListener listener)`

参数

listener

实现 IReadTagsListener 接口的对象，需要实现 tagsRead 接口。

返回值

无

示例:

```
mService.registerReadTags(mIReadTagsListener);
```

27. 卸载回调函数

`void unregisterReadTags (IReadTagsListener listener)`

参数

listener

调用 registerReadTags 时的对象。

返回值

无

示例:

```
mService.unregisterReadTags(mIReadTagsListener);
```

EPC 类

类成员变量

```
byte[] id;
int len;
int rssi;
int count;
```

使用方法

```
import com.seuic.uhf.EPC;
EPC epc = new EPC();
```

函数接口

1. 获取 EPC 对象中的 id 数据

`String getId()`

参数

无

返回值

String; 十六进制的表示的字符串

2. 比较 id 号是否相同

`boolean equals(Object obj)`

参数

EPC 对象

返回值

boolean; true:成功、false:失败

3. 获取附加数据 (新模块支持)

`String getEmbedded()`

参数

无

返回值

String; 十六进制的表示的字符串