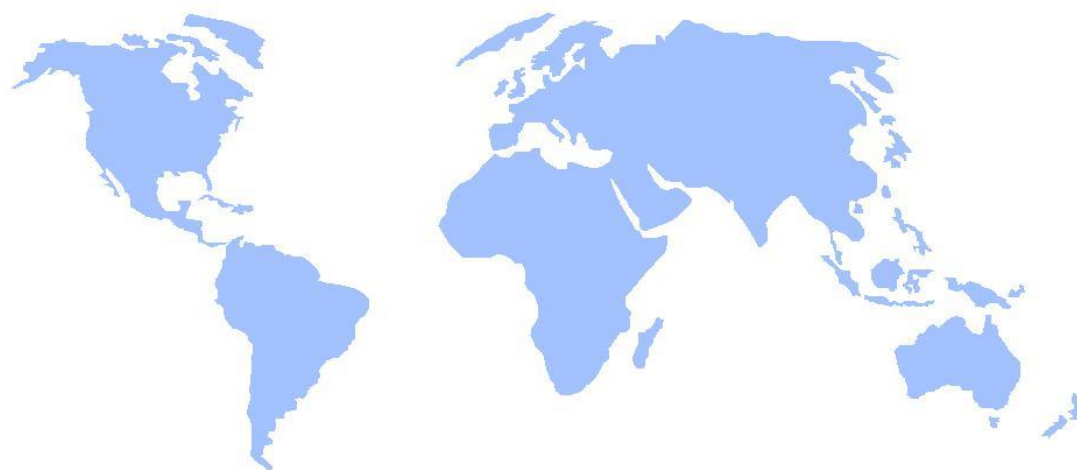




Android UHF setParameters 接口 使用说明书



www.seuic.com

修订记录

版本号	修订日期	修订内容	修订人员
1.0	2019/12/30	建立本文档	冯晓东
2.0	2020/07/01	增加 setParamBytes 接口	冯晓东

目录

UHF 接口	2
使用方法	2
函数接口	2
1. 设置功能参数.....	2
2. 设置参数 bytes	3

UHF 接口

使用方法

```
import com.seuic.uhf.UHFService;  
UHFService mDevice = UHFService.getInstance();  
列如:  
mDevice.setParameters(UHFService.PARAMETER_CLEAR_EPCLIST, 1);
```

函数接口

1. 设置功能参数

boolean setParameters(int id, int value)

参数

id

功能 ID

value

参数值

如下表:

id (功能参数名)	value 值	备注
PARAMETER_LINK_PROFILE (0)	0,1,2,3(P0,P1,P2,P3)	Profile

PARAMETER_INVENTORY_SESSION (1)	0,1,2,3(S0,S1,S2,S3)	Session
PARAMETER_INVENTORY_SESSION_TARGET (2)	0,1(A,B)	Target
PARAMETER_INVENTORY_SPEED (3)	0,1(fast,slow)	寻卡速度
PARAMETER_EXTENSIONS_FASTID (5)	0,1(DISABLE,ENABLE)	Fastid
PARAMETER_CLEAR_EPCLIST_WHEN_START_INVENTORY (6)	0,1(DISABLE,ENABLE)	开始寻卡是否清空之前 EPC
PARAMETER_HIDE_PC (7)	0,1(DISABLE,ENABLE)	隐藏 PC 号
PARAMETER_CLEAR_EPCLIST (8)	0,1(DISABLE,ENABLE)	清空 EPC
PARAMETER_ALGORITHM_STARTQVALUE (9)	0~15(4)	初始 Q 值
PARAMETER_ALGORITHM_MINQVALUE (10)	0~15(0)	最小 Q 值
PARAMETER_ALGORITHM_MAXQVALUE (11)	0~15(15)	最大 Q 值
PARAMETER_ALGORITHM_TOGGLETARGET (13)	0,1(DISABLE,ENABLE)	Toggletarget
PARAMETER_EXTENSIONS_TAGFOCUS (15)	0,1(DISABLE,ENABLE)	TagFocus

返回值

boolean; true:成功、false:失败

2. 设置参数 bytes

Boolean setParamBytes(int id, byte[] value)

参数

id

功能 ID

value

参数值(当不需要设置时，byte[]置为 null。)

如下表：

id (功能参数名)	value 值	备注
PARAMETER_TAG_EMBEDDEDATA (30)	EmbeddedData	附加数据
PARAMETER_TAG_FILTER (31)	TagFilter	过滤条件

返回值

boolean; true:成功、false:失败

byte 数组定义

```
EmbeddedData {
    public int bank; //一个字节附加数据区域 2,3，分别是 TID、User
    public int startaddr; //一个字节附加数据起始地址
    public int bytecnt; //一个字节附加数据长度
    public byte[] accesspwd; //四个字节访问密码，默认无密码 00000000
}
```

```

TagFilter {
    public int bank; //一个字节过滤数据区域 1,2,3, 分别是 EPC、TID、User
    public int startaddr; //一个字节过滤数据起始地址
    public int flen; //一个字节过滤数据长度
    public int isInvert; //一个字节过滤数据是否匹配
    public byte[] fdata; //过滤数据
}

```

设置附加数据 示例:

```

byte[] embd = new byte[255];
embd[0] = (byte)mBankIndex;
embd[1] = (byte)Integer.parseInt(et_embedded_address.getText().toString());
embd[2] = (byte)Integer.parseInt(et_embedded_length.getText().toString());
System.arraycopy(getHexByteArray(et_embedded_acpwd.getText().toString()),
0, embd, 3, 4);
mService.setParamBytes(UHFService.PARAMETER_TAG_EMBEDDEDDATA, embd);

```

取消附加数据 示例:

```

mService.setParamBytes(UHFService.PARAMETER_TAG_EMBEDDEDDATA, null)

```

设置过滤数据 示例:

```

byte[] val = new byte[255];
val[0] = (byte)mBankIndex;
val[1] = (byte)Integer.parseInt(et_filter_offset.getText().toString());
val[2] = (byte)Integer.parseInt(et_filter_len.getText().toString());
val[3] = (byte)mIsInvert;
byte[] data = getHexByteArray(et_filter_data.getText().toString());
if(val[2] != data.length) {
    return false;
}
System.arraycopy(data, 0, val, 4, val[2]);
mService.setParamBytes(UHFService.PARAMETER_TAG_FILTER, val);

```

取消过滤数据 示例:

```

mService.setParamBytes(UHFService.PARAMETER_TAG_FILTER, null);

```

getHexByteArray 方法:

```

/**
 * 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;
}
```