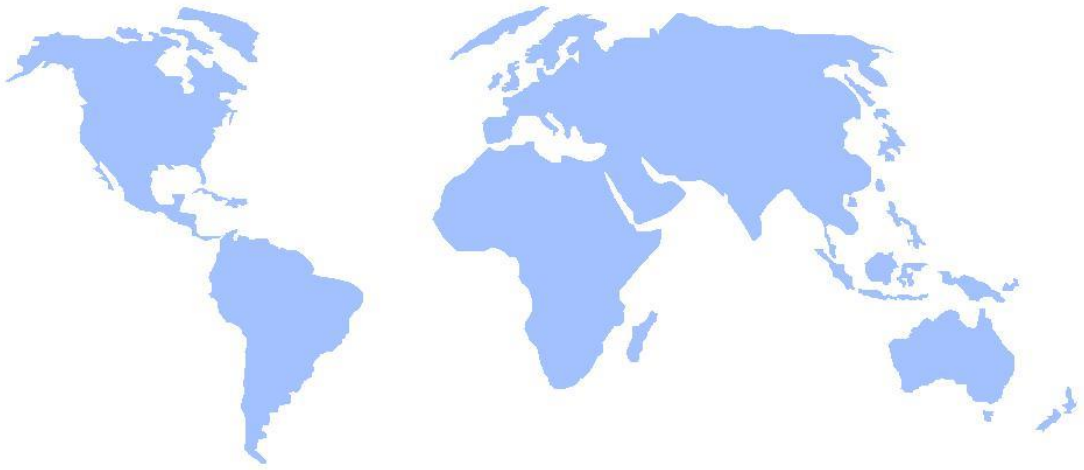
Android UHF setParameters Interface User Manual



Contents

UHF interface

Instructions

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

Function interface

1. Set function parameters

boolean setParameters(int id, int value)

parameter

id

Function ID

value

Parameter value

The following table:

id (function parameter name)	value	Remarks
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)	Find card speed
PARAMETER_EXTENSIONS_FASTID (5)	0,1(DISABLE,ENABLE)	Fastid
PARAMETER_CLEAR_EPCLIST_WHEN_START_INVENTORY (6)	0,1(DISABLE,ENABLE)	EPC is cleared before

		starting to find card
PARAMETER_HIDE_PC (7)	0,1(DISABLE,ENABLE)	Hide PC number
PARAMETER_CLEAR_EPCLIST (8)	0,1(DISABLE,ENABLE)	Clear EPC
PARAMETER_ALGORITHM_STARTQVALUE (9)	0~15(4)	Initial Q value
PARAMETER_ALGORITHM_MINQVALUE (10)	0~15(0)	Minimum Q value
PARAMETER_ALGORITHM_MAXQVALUE (11)	0~15(15)	Maximum Q value
PARAMETER_ALGORITHM_TOGGLETARGET (13)	0,1(DISABLE,ENABLE)	Toggletarget
PARAMETER_EXTENSIONS_TAGFOCUS (15)	0,1(DISABLE,ENABLE)	TagFocus

return value

boolean; true: success, false: failure

2. Set parameter bytes

Boolean setParamBytes(int id, byte[] value)

parameter

id

Function ID

value

Parameter value (when no setting is needed, byte[] is set to null.)

The following table:

id (function parameter name)	value	Remarks
PARAMETER_TAG_EMBEDEDDATA (30)	EmbededData	Additional data
PARAMETER_TAG_FILTER (31)	TagFilter	Filter condition

return value

boolean; true: success, false: failure

byte array definition

EmbededData {

public int bank; //One byte additional data area 2, 3, respectively TID, User

public int startaddr; //Start address of one byte additional data

public int bytecnt; //One byte additional data length

public byte[] accesspwd; //Four bytes access password, default no password 00000000

}

TagFilter {
public int bank; //One byte filter data area 1, 2, 3, respectively EPC, TID, User
public int startaddr; //Start address of one byte filter data
public int flen; //One byte filter data length
public int isInvert; //Does one byte filter data match?
public byte[] fdata; //Filter data
}

Set additional data list:

```
byte[] embd = new byte[255];
embd[0] = (byte)mBankIndex;
embd[1] =
(byte)Integer.parseInt(et_embeded_address.getText().toString());
embd[2] =
(byte)Integer.parseInt(et_embeded_length.getText().toString());
System.arraycopy(getHexByteArray(et_embeded_acpwd.getText().toString()),
0, embd, 3, 4);
mService.setParamBytes(UHFService.PARAMETER_TAG_EMBEDEDDATA, embd);
```

Cancel the additional data list:

```
mService.setParamBytes(UHFService.PARAMETER_TAG_EMBEDEDDATA, null)
```

Set the filter data list:

```
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);
```

Unfiltered data list:

```
mService.setParamBytes(UHFService.PARAMETER_TAG_FILTER, null);
```

getHexByteArray method:

```
/** *
String to hexadecimal form array*/
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;
}
```