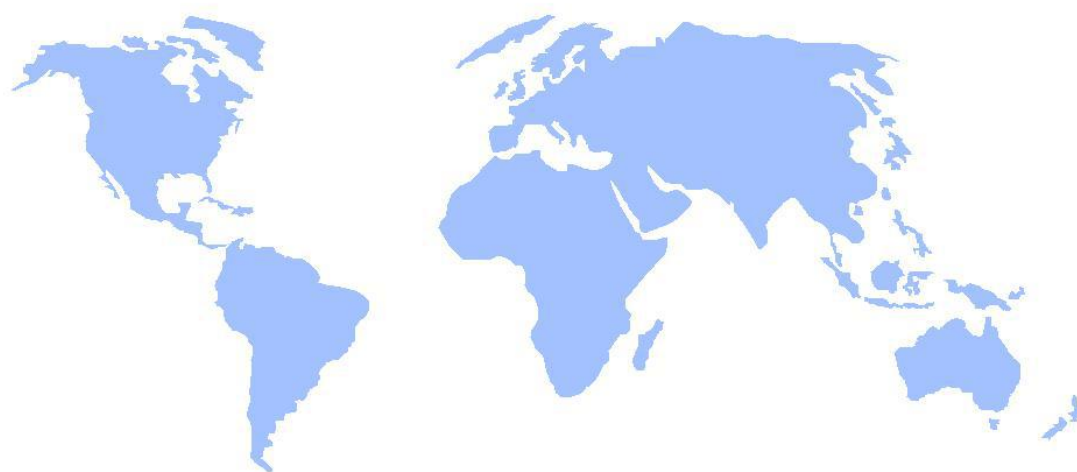




特殊设置 SDK 使用说明书



www.seuic.com

修订记录

版本号	修订日期	修订内容	修订人员
0.0	2014/08/05	建立本文档	钱进
0.1	2014/08/05	加入 keypad、touch 接口的说明	张松
0.2	2014/08/08	更改文档模板，并按模板重写文档	徐良伟
1.0	2014/08/09	加入“系统包的使用”，并按 touch 新接口修改文档	徐良伟
1.1	2014/09/02	加入 touch 新的接口说明 加入 misc 接口	魏启运
1.2	2015/04/29	在 misc 接口中增加启用/禁用状态栏下拉的接口，增加卸载/加载外置 SD 卡的接口	张松

目录

概述.....	3
硬件平台.....	3
软件平台.....	3
适用对象.....	3
系统包的使用.....	3
触摸屏接口.....	5
TouchService 类.....	5
使用方法.....	5
函数接口.....	5
1. 获取当前手套模式开关状态.....	5
2. 设置手套模式开关.....	5
3. 获取当前触摸屏使能状态.....	6
4. 设置触摸屏使能开关.....	6
5. 获取虚拟按键屏蔽状态.....	6
6. 设置虚拟按键屏蔽状态.....	7
键盘接口.....	8
KeypadService 类.....	8
使用方法.....	8
函数接口.....	8
1. 获取键盘模式.....	8
2. 设置键盘模式.....	9
Misc 接口.....	10
Misc 类.....	10
使用方法.....	10
函数接口.....	10
1. 获取设备 SN.....	10

概述

本文档介绍了我司设备上提供的设备相关的包的使用说明，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：

- AUTOID 9
- AUTOID PAD
- AUTOID 6L-P

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 4.3 版本，支持 eclipse 开发工具。

适用对象

除了 Android 的标准功能外，希望用到设备提供的增值功能的开发人员。这部分增值功能包括一维扫描、二维扫描、键盘特殊设置、触摸特殊设置等实用功能。

系统包的使用

如果接口中标明是系统包，表明该包是内置到系统中的，无需将该包加入到 apk 中。下面以键盘接口为例，来说明如何在 eclipse 中使用系统包。

1. 在工程目录下建一个 libsref 目录(目录名称可以任意指定)，将 keypad.jar 拷贝到此目录下。
2. 在 eclipse 中选中工程，在菜单中选择 Project - Properties - Java Build Path，点击 Add External JARS...按钮，选择刚才的 keypad.jar。这时可以在工程视图的 Referenced Libraries 下看到刚才我们引用的 keypad.jar，展开可以看到所有 keypad.jar 提供的包、类、函数、变量。
3. AndroidManifest.xml 中使用 uses-library 用包名来标示 keypad.jar 是引用包。

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

</application>

触摸屏接口

包名	com.seuic.touch
包文件	touch.jar
系统包	是
包含的类	TouchService
作用	提供触摸屏相关的控制接口

TouchService 类

使用方法

```
import com.seuic.touch.TouchService;  
TouchService touch = TouchService.getInstance();
```

函数接口

函数	说明
getGloveMode	获取当前手套模式开关状态
setGloveMode	设置手套模式开关
getTouchEnabled	获取当前触屏使能状态
setTouchEnabled	设置触摸屏使能开关
getVirKeyDisabled	获取虚拟按键屏蔽状态(仅 D500 支持)
setVirKeyDisabled	设置虚拟按键屏蔽状态(仅 D500 支持)

1. 获取当前手套模式开关状态

int getGloveMode()

参数

无

返回值

返回手套模式状态，有以下几种：

状态	值	说明
MODE_GLOVE_OFF	0	手套模式关闭
MODE_GLOVE_ON	1	手套模式打开
	<0	触摸接口不正常，或该设备不支持手套模式设置

2. 设置手套模式开关

boolean setGloveMode(int mode)

参数

mode

手套模式状态，见 `getGloveMode()` 函数返回值的说明。

返回值

返回 `true` 表示成功，`false` 表示失败。

3. 获取当前触摸屏使能状态

`int getTouchEnable()`

参数

无

返回值

返回触摸屏使能状态，有以下几种：

状态	值	说明
TOUCH_ENABLED_OFF	0	触摸屏处于锁屏状态
TOUCH_ENABLED_ON	1	触摸屏处于正常使用状态
	<0	触摸接口不正常，或该设备不支持触摸屏使能设置

4. 设置触摸屏使能开关

`boolean setTouchEnabled(int enable, Context context)`

参数

enable

触摸屏使能状态，见 `getTouchEnabled()` 函数返回值的说明。

context

调用者的上下文，用于向系统广播锁屏通知，以便在通知栏显示当前锁屏状态。

返回值

返回 `true` 表示成功，`false` 表示失败。

5. 获取虚拟按键屏蔽状态

`int getVirKeyDisabled ()`

参数

无

返回值

返回虚拟按键屏蔽状态，有以下几种：

状态	值	说明
VKEY1_DISABLE_MASK	$1 \ll 0$	虚拟键一被屏蔽
VKEY2_DISABLE_MASK	$1 \ll 1$	虚拟键二被屏蔽
VKEY3_DISABLE_MASK	$1 \ll 2$	虚拟键三被屏蔽
VKEY_ALL_DISABLE_MASK	0x7	所有虚拟按键均屏蔽
	0	所有虚拟按键未被屏蔽
	<0	触摸接口不正常，或设备不支持触摸屏设置

6. 设置虚拟按键屏蔽状态

boolean setVirKeyDisabled(int disableMask)

参数

disableMask

虚拟按键屏蔽状态，见 GetVirKeyDisabled ()函数返回值的说明。

返回值

返回 true 表示成功，false 表示失败。

键盘接口

包名	com.seuic.keypad
包文件	keypad.jar
系统包	是
包含的类	KeypadService
作用	提供键盘相关的控制接口

KeypadService 类

使用方法

```
import com.seuic.keypad.KeypadService;
KeypadService keypad = KeypadService.getInstance();
```

函数接口

函数	说明
getMode	获取键盘模式(仅 D500 支持)
setMode	设置键盘模式(仅 D500 支持)

1. 获取键盘模式

`int getMode()`

参数

无

返回值

返回键盘功能键模式，有以下几种：

状态	值	说明
MODE_NORMAL	0	标准模式。根据当前 Num、Fn 的状态灯输出数字或功能键。
MODE_STICK_ONCE	1	Fn 键只粘滞一次模式。当处于 Fn 状态时，按一下按键输出功能键，然后自动切换到 Num 模式。
	<0	键盘接口不正常，或该设备不支持键盘模式设置

2. 设置键盘模式

boolean setMode(int mode)

参数

mode

键盘模式状态，见 `getMode()` 函数返回值的说明。

返回值

返回 `true` 表示成功，`false` 表示失败。

Misc 接口

包名	com.seuic.misc
包文件	misc.jar
系统包	是
包含的类	Misc
作用	提供系统其它接口

Misc 类

使用方法

```
import com.seuic.misc.Misc;
Misc misc = new Misc();
```

函数接口

函数	说明
getSN	获取设备 SN
getStatusBarEnabled	获取系统状态栏是否允许下拉的状态
setStatusBarEnabled	设置系统状态栏是否允许下拉
getExtSDCardMounted	获取外置 SD 卡是否卸载
setExtSDCardMounted	设置外置 SD 卡卸载或加载

1. 获取设备 SN

`String getSN()`

参数

无

返回值

返回键盘功能键模式，有以下几种：

状态	值	说明
null	null	获取 SN 失败
String 类型	String 类型	获取成功

2. 获取系统状态栏是否允许下拉的状态

`int getStatusBarEnabled(Context context)`

参数

context

上下文。

返回值

返回状态栏是否允许下拉的状态，如下：

状态	值	说明
STATUSBAR_DISABLED	0	系统状态栏不允许下拉。
STATUSBAR_ENABLED	1	系统状态栏允许下拉。
	<0	接口不正常，或该设备不支持系统状态栏的设置。

3. 设置系统状态栏是否允许下拉

void setStatusBarEnabled(Context context, int enable)

参数

context

上下文。

enable

状态栏是否允许下拉，取值见 `getStatusBarEnabled()` 返回值。

返回值

无

4. 获取外置 SD 卡是否卸载

int getExtSDCardMounted(Context context)

参数

context

上下文。

返回值

返回外置 SD 卡的状态，如下：

状态	值	说明
EXT_SDCARD_UNMOUNTED	0	外置 SD 卡卸载。
EXT_SDCARD_MOUNTED	1	外置 SD 卡未卸载。
	<0	接口不正常，或该设备不支持外置 SD 卡的卸载。

5. 设置外置 SD 卡卸载或装载

void setExtSDCardMounted(Context context, int enable)

参数

context

上下文。

enable

设置外置 SD 卡的卸载加载，取值见 `getExtSDCardMounted()` 返回值。

返回值

无