

三星 S Pen 悬空操作适配指导

从三星 Galaxy Note 系列诞生起，S Pen 就是每一代 Note 系列的标配与亮点。S Pen 始终坚持在硬件和软件方面优化创新，从 Galaxy Note9 开始 S Pen 增加了蓝牙功能，通过单击或双击 S Pen 上的按键实现远程控制，有效遥控距离高达 10 米。比如可以使用 S Pen 翻看相册中的照片，相册中的视频也可以使用 S Pen 播放、暂停，还可以使用 S Pen 远程控制相机进行拍照，轻松捕捉精彩瞬间。



图 1 | 控制拍照

在新一代 Galaxy Note10 上这支小小的笔有了新的设计和功能，它增加了陀螺仪和加速传感器，可以感知手部动作，通过简单的手势即可操控手机。比如顺时针或逆时针移动 S Pen 可调整摄像头焦距，以便拍摄近景或全景。在相册中通过左滑右滑可切换到下一张照片，或返回到上一张照片。通过手势向上调高音量，手势向下降低音量，无需再触碰音量键。

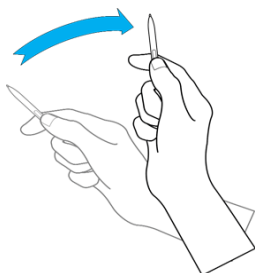


图 2 | 手势

本文将介绍 S Pen 包含按键和手势动作的悬空操作功能，以便开发者把 S Pen 适配到更多的第三方 APP 中，实现支持更多的操控与应用。

一. 悬空操作实现原理

S Pen 通过 BLE 连接到手机，由 S Pen Framework 管理连接，并会将 BLE 事件转换为 KeyEvent，应用只需要回调现有的 KeyEvent-Callback 来处理 S Pen 事件即可。

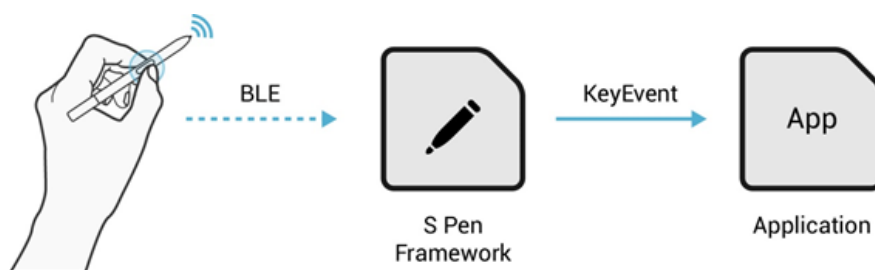


图 3 | 流程图

为了实现应用功能，应用必须定义 RemoteActions 并遵循下面的实现过程。

1. 在 Manifest 文件中指定要映射到 S Pen 远程事件的 KeyEvent。
2. 实现 KeyEvent-Callback (如果需要)。

二. 悬空操作实现方法

1. 在需要响应的 activity 中添加 `<intent-filter>` 和 `<meta-data>`

必须在 `<meta-data>` 中指定定义 RemoteActions 的资源文件.xml

```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2   package="com.samsung.android.blesample ">
3
4   <application ... >
5     <activity android:name="SampleActivity" >
6       <intent-filter>
7         <action android:name="com.samsung.android.support.REMOTE_ACTION" />
8       </intent-filter>
9
10      <meta-data android:name="com.samsung.android.support.REMOTE_ACTION"
11        android:resource="@xml/remote_action_sample"/>
12    </activity>
13    ...
14  </application>
15 </manifest>
```

图 4 | Manifest 代码举例

2. 在 res / xml /下创建一个.xml 文件，将文件名命名为步骤 2 中指定的资源文件名称。

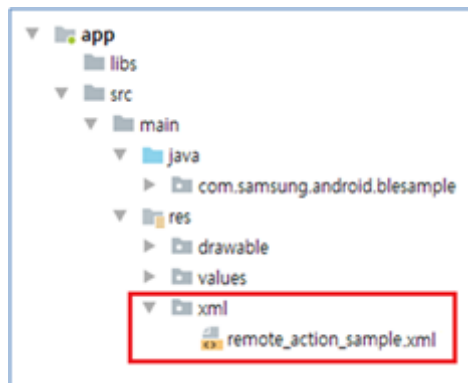


图 5 | 目录举例

在 xml 文件中具有<remote-actions>的根元素，并且可以包括多个<action>元素。另外，每个<action>包含 id, label, priority, trigger_key 等信息。

```

1 <?xml version="1.0" encoding="utf-9">
2 <remote-actions version="float"
3 enable_key="string">
4   <action
5     id="string"
6     label="@string/resource_name"
7     priority="int"
8     trigger_key="string"
9     repeatable=["true" | "false"]
10    repeatable_interval=["short" | "medium" | "long" | "int"]>
11     <preference name="gesture" value=["click", "double_click", "swipe_left",
12         "swipe_right", "swipe_up", "swipe_down", "circle_ccw", "circle_cw"]/>
13     <preference name="button_only" value=["true" | "false"]/>
14     <preference name="motion_only" value=["true" | "false"]/>
15   </action>
16 </remote-actions>
17

```

图 6 | xml 文件举例

下面对 xml 文件中的元素进行介绍:

1) <action>

名称	必要性	设置值	说明
id	强制	只允许使用字母，数字或下划线。	针对指定动作的唯一id
label	强制	只允许应用程序包中的字符串资源ID	在S Pen悬空操作设定页面上显示的动作标签
priority	强制	Version 1.0: 基于优先级，1被映射到单击手势，2被映射到双击手势，3及之后是App动作列表中注册的候选动作。 Version 1.2: 默认手势由preference Tag确定。	不允许出现相同优先级数字越小，优先级越高 1代表最高优先级
trigger_key	强制	1. 使用Android KeyEvent.java中定义的按键名称，但不包括“KEYCODE_”。 示例：KEYCODE_PAGE_UP-> trigger_key = "PAGE_UP" 2. 可使用“+”作为分隔符来定义KeyShortcut 示例1：KEYCODE_CTRL_LEFT和KEYCODE_N的组合： trigger_key = "CTRL_LEFT + N" 示例2：KEYCODE_CTRL_LEFT, KEYCODE_SHIFT_LEFT和KEYCODE_Z的组合： trigger_key = "CTRL_LEFT + SHIFT_LEFT + Z"	应用程序定义要接收的KeyEvent
repeatable	可选	此操作可重复时设置为true	Version 1.2
repeatable_interval	可选	支持short, medium, long或int类型 (short: 300ms, medium: 500ms, long: 1000ms, int: 50~3000ms)	Version 1.2

图 7 | action 表格

2) <preference>

```
1 <preference name="gesture" value=["click", "double_click", "swipe_left", "swipe_right",  
2     "swipe_up", "swipe_down", "circle_ccw", "circle_cw"]/>  
3 <preference name="button_only" value=["true" | "false"]/>  
4 <preference name="motion_only" value=["true" | "false"]/>
```

图 8 | preference 代码示例

描述: 确定要映射的默认手势，检查此操作是按下按钮还是手势动作

属性:

- name: 可设置为 "gesture" , "button_only" , "motion_only"
 - "gesture" : 设置您要设置为默认的手势类型
 - "button_only" : 仅为与按钮相关的手势提供操作
 - "motion_only" : 仅为与动作相关的手势提供动作
- value
 - 如果 name = "gesture" , 则下表中的 value 都可以使用。
 - 多个值由 "|" 分隔, 例如: "click | swipe_down | swipe_down" 。

值	手势类型	说明
"click"	Button	单击
"double_click"	Button	双击
"swipe_left"	Motion	左滑 ←
"swipe_right"	Motion	右滑 →
"swipe_up"	Motion	上滑 ↑
"swipe_down"	Motion	下滑 ↓
"circle_ccw"	Motion	逆时针画圈
"circle_cw"	Motion	顺时针画圈

图 9 | value 说明

- 如果 name = "button_only" 或者 "motion_only" ，则 value 允许设置为"true"或"false"。
- 如果这个配置与 "gesture" 的值相矛盾，该设置将被忽略。例如，如果将 "swipe_left" 设置为 "gesture" 的值，而 "button_only" 为 true，则配置是矛盾的，将忽略 "button_only" 设置。

3. KeyEvent-Callback 的实现

与处理 Android KeyEvent 相同，建议在 onKeyDown 处理发送的 KeyEvent。

示例：

- Page Down 键：在 Y 方向上滚动 ScrollView +500
- Page Up 键：在 Y 方向上滚动 ScrollView -500

```
1 public class SampleActivity extends Activity {
2     private ScrollView mScrollView;
3     @Override
4     protected void onCreate(Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         setContentView(R.layout.sample_activity1_layout);
7         mScrollView = (ScrollView) findViewById(R.id.scroll_view);
8     }
9
10    @Override
11    public boolean onKeyDown(int keyCode, KeyEvent event) {
12        if (keyCode == KeyEvent.KEYCODE_PAGE_DOWN) {
13            mScrollView.smoothScrollBy(0, 500);
14        } else if (keyCode == KeyEvent.KEYCODE_PAGE_UP) {
15            mScrollView.smoothScrollBy(0, -500);
16        }
17        return super.onKeyDown(keyCode, event);
18    }
19 }
```

图 10 | KeyEvent 响应代码示例

可参考 Android 开发者官网。

<https://developer.android.google.cn/reference/android/view/KeyEvent.Callback.html>

三. S Pen 一般控制

如果应用没有按照上面的悬空操作添加功能，对于相机拍照和注册了 Media session 的媒体应用将

遵循一般控制原则，S Pen Framework 会将按键和手势动作按下表转化为 KeyEvent 传送给应用，应用只需要对下面的 KeyEvent 进行处理就可以响应 S Pen。

动作	功能	对应事件
按一下	播放/暂停	KeyEvent.KEYCODE_MEDIA_PLAY_PAUSE
按两下	播放下一首	KeyEvent.KEYCODE_MEDIA_NEXT
向上滑	增大音量	KeyEvent.KEYCODE_VOLUME_UP
向下滑	减小音量	KeyEvent.KEYCODE_VOLUME_DOWN
向左滑	播放上一首	KeyEvent.KEYCODE_MEDIA_PREVIOUS
向右滑	播放下一首	KeyEvent.KEYCODE_MEDIA_NEXT

四. S Pen Remote SDK

S Pen 的新功能可识别手势动作，通过 S Pen Framework 可以轻松识别向上，向下，向左，向右，顺时针和逆时针响应方向动作，可以根据需要将这些按钮和动作功能添加到应用程序功能中。但是，如果应用需要获得 S Pen 运动的原始数据，实现更强大的应用功能，必须使用 S Pen Remote SDK。S Pen Remote SDK 提供识别运动坐标和按键功能，应用可以利用原始动作数据，设计自定义的手势轨迹，或者利用相对位置来完成用户交互，增强游戏和应用的用户体验。关于 S Pen Remote SDK 后续将会详细介绍，请大家持续关注。

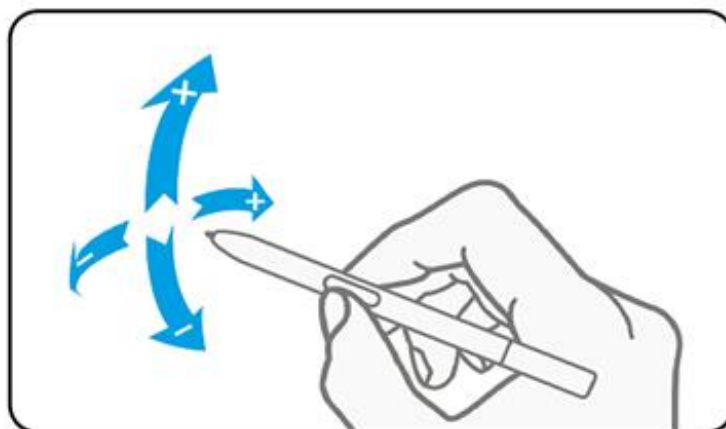


图 11 | 原始坐标手势图

五. 联系我们

如果您对 S Pen 悬空操作有任何问题，可以发邮件到下面的邮箱与我们联系。

邮箱地址: rdtpservice@samsung.com

邮件主题: 三星 S Pen+APP 名