

密级状态： 绝密(     )     秘密(     )     内部资料(     )     公开(    √  )

# Rockchip Android11 异显开发指南

(第二系统产品部)

<b>文件状态：</b> [   ] 草稿 [   ] 正在修改 [ √ ] 正式发布	<b>当前版本：</b>	V1.0
	<b>作   者：</b>	董正勇
	<b>完成日期：</b>	2021-06-08
	<b>审   核：</b>	吕金华
	<b>审核日期：</b>	2021-06-10

## 免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

## 商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自所有者所有。

## 版权所有 © 2021 福州瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园 A 区 18 号

网址：[www.rock-chips.com](http://www.rock-chips.com)

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：[fae@rock-chips.com](mailto:fae@rock-chips.com)

## 修改记录

版本号	作者	修改日期	修改说明	备注
V1.0	董正勇	2021.06.08	初版发布	

# 目 录

1 概述.....	2
2 异显方案介绍.....	2
2.1 ANDROID PRESENTATION.....	2
2.2 ANDROID ACTIVITY 指定屏幕启动.....	3
3 双屏异触配置.....	4
3.1 原理说明.....	4
3.2 如何配置副屏 TP.....	5
3.2.1 修改代码.....	5
3.2.2 配置指定触摸屏设备的 IDC 文件.....	5
3.3 查看配置是否成功.....	6
4 副屏方向设置.....	6
4.1 RK3568 副屏方向配置.....	6
4.2 RK3288,RK3399,PX30,RK3326 副屏方向配置.....	7
5 其他开发配置.....	7
5.1 支持输入法在副屏显示.....	7
5.2 副屏 DPI 设置.....	8
5.3 鼠标主副屏切换显示.....	8

## 1 概述

本文档主要介绍 Rockchip SDK 平台上支持的异显方案，以方便客户开发，适用于 Android 版本 Android11.0，适用芯片 RK3399/RK3288/RK3326/PX30/RK3568。

## 2 异显方案介绍

目前有两种异显方案：Android Presentation 和 Andorid Activity 指定屏幕启动。

Android Presentation, 需要 APP 开发中调用相应接口使指定的 view(Presentation view 是一个特殊的 dialog 类型 view)在副屏中显示。

Android Activity 指定屏幕启动,APP 在启动 activity 时可以使用 display id 参数在对应的屏幕上直接显示。

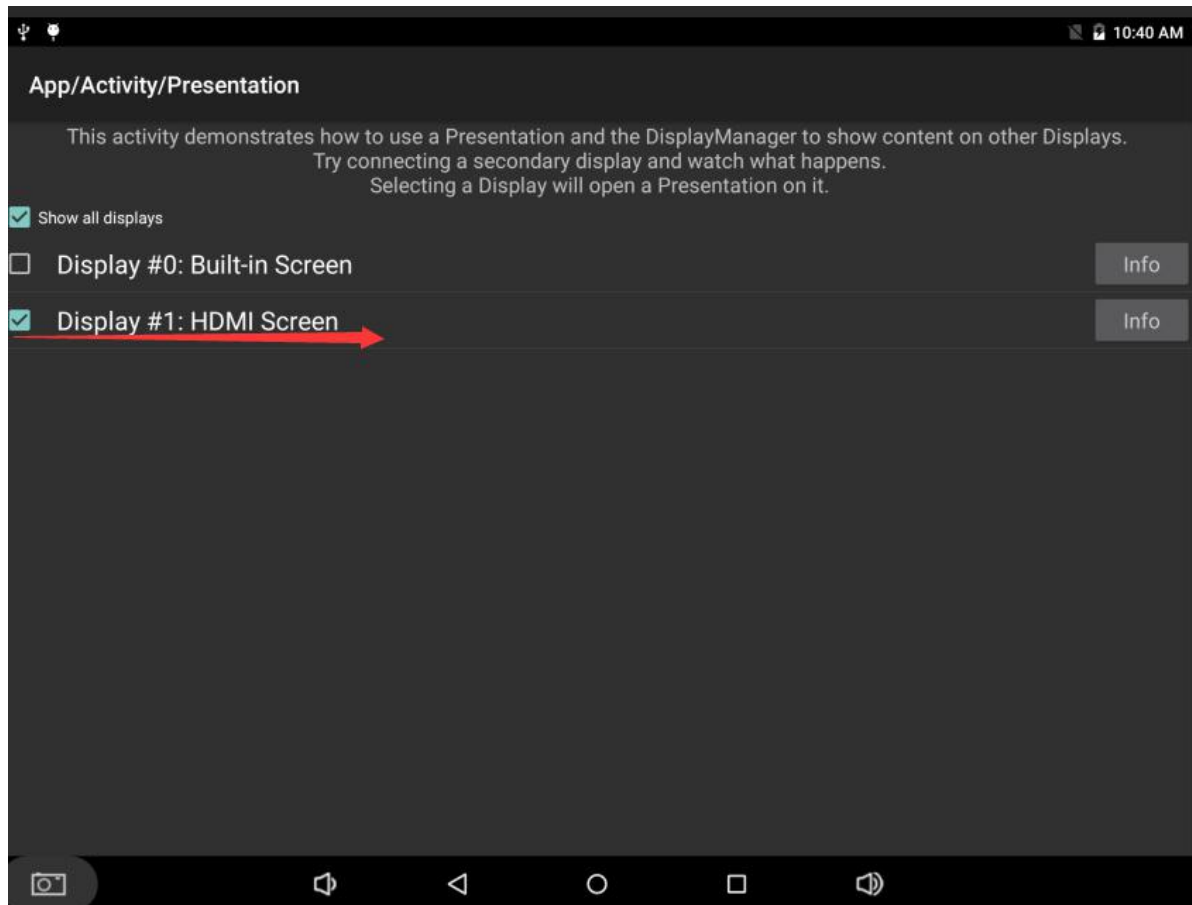
两者主要区别在：

1 前者的 acitivity 需要独立开发，把需要显示的内容投到副屏，后者可以不需要源码，调用命令行或者系统接口把第三方 app 的 acitivity 投送到副屏；

2.前者只有一个 activity 在顶层，通过特殊的 dialog 将指定内容显示在副屏，后者是两个 activity 分别显示在主副屏。

### 2.1 Android Presentation

SDK 中已有该接口使用的相关 demo，请进入 development/samples/ApiDemos 目录，mm 编译生成相应的 apk，安装 apk 后点击 App->Activity->Presentation 选项，即可进入 Presentation 调用界面。在该界面需点击如下副屏的 checkbox 选项，即可在副屏显示相应图片。



具体代码位于如下路径：

development/samples/ApiDemos/src/com/example/android/apis/app/PresentationActivity.java

## 2.2 Android Activity 指定屏幕启动

在 `startActivity` 接口参数中设置指定屏幕的 `display id`，Activity 将在指定屏幕上直接启动显示。

Activity 的多显示器支持需要设备支持 `<feature name="android.software.activities_on_secondary_displays" />`。同时 application 或者 activity 需要支持分屏属性即 `<application>` 或者 `<activity>` 标签下设置新的属性 `android:resizeableActivity="true"`，这个属性在你 target 到 Android N 后 `android:resizeableActivity` 的默认值就是 `true`。

`ActivityOptions` 提供两个新函数以支持多个显示器：

`setLaunchDisplayId()` 指定 Activity 在启动后应显示在哪个显示器上。

`getLaunchDisplayId()` 返回操作组件的当前启动显示器。

`setLaunchDisplayId()` 使用示例：

示例中使用 `MediaRouter` 接口获取副屏的 `display id`，同理也可以使用 `DisplayManager` 接口获取响应的 `display id`。

注意此处只能打开其他包中的 Activity，否则会提示：“App does not support launch on secondary displays”。

```
private void showSecondByActivity(Context context){
    ActivityOptions options = ActivityOptions.makeBasic();
    MediaRouter mediaRouter = (MediaRouter)
        context.getSystemService(Context.MEDIA_ROUTER_SERVICE);
    MediaRouter.RouteInfo route =
        mediaRouter.getSelectedRoute(MediaRouter.ROUTE_TYPE_LIVE_VIDEO);
    if (route != null) {
        Display presentationDisplay = route.getPresentationDisplay();
        options.setLaunchDisplayId(presentationDisplay.getDisplayId());
        //options.
        Intent intent = new Intent("android.intent.action.MUSIC_PLAYER");
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent, options.toBundle());
    }
}
```

同时也对 adb shell 进行了扩展，以支持多个显示器。shell start 命令现在可用于启动操作组件，并指定操作组件的目标显示器：

adb shell am start --display <display\_id> <activity\_name>

例如 adb shell am start --display 1 com.android.settings/.Settings（把 settings 界面启动到指定的副屏）

## 3 双屏异触配置

### 3.1 原理说明

从如下的 MotionEvent 中可以看到系统上报的 event 中包含一个 displayId 的变量，系统会根据这个 displayId 的值将事件分发给对应的主屏或者副屏。

```
MotionEvent(deviceId=2,source=0x00001002,action=1,actionButton=0x00000000,flags=0x00000000,metaState=0x00000000,buttonState=0x00000000,edgeFlags=0x00000000,xPrecision=8.5,yPrecision=8.9, displayId=0, pointers=[0: (496.1, 1029.6)]), policyFlags=0x62000000, age=13734.7ms
```

触摸事件的 displayId 值在 frameworks/native/services/inputflinger/EventHub.cpp 的 openDeviceLocked 中，系统判断如果是副屏的 TP，该触摸设备就会设置

INPUT\_DEVICE\_CLASS\_EXTERNAL 属性，相应的触摸事件就会传给副屏 window。无论是 I2C，USB 或者蓝牙接口，只需要把副屏的触摸设备设成该属性就可以实现副屏的触摸。

## 3.2 如何配置副屏 TP

目前有如下两种方式可以配置副屏 TP：

- 1 修改 EventHub.cpp 代码。
- 2 配置触摸屏的 IDC 文件。

### 3.2.1 修改代码

通过修改 EventHub.cpp 代码如下，在 isExternalDeviceLocked 函数中判断设备名称为指定的副屏设备（例如副屏 TP 名称为 gsl3673），则返回 true。

```
bool EventHub::isExternalDeviceLocked(Device* device) {
+   //example for set external device.
+   const char *USB_DEVICE_NAME ="gsl3673";
+   if(strcmp(device->identifier.name.string(),USB_DEVICE_NAME)==0){
+       ALOGD(" %d   name:  \"%s\\\"\\n\",__LINE__, device->identifier.name.string());
+       return true;
+   }
    if (device->configuration) {
        bool value;
        if (device->configuration->tryGetProperty(String8("device.internal"), value)) {
            return !value;
        }
    }
    return device->identifier.bus == BUS_USB || device->identifier.bus == BUS_BLUETOOTH;
}
```

### 3.2.2 配置指定触摸屏设备的 IDC 文件

触摸屏设备对应的 IDC 文件中设置 device.internal=0



`device.internal`

定义：`device.internal = 0 | 1`

指定输入设备属于内置组件，还是外部连接（很可能可拆卸）的外围设备。

- 如果值为 0，则该设备为外部设备。
- 如果值为 1，则该设备为内部设备。
- 如果该值未指定，则 USB (BUS\_USB) 或蓝牙 (BUS\_BLUETOOTH) 总线上的所有设备的默认值均为 0，否则，值为 1。

具体信息可参考谷歌官方网址：

<https://source.android.com/devices/input/input-device-configuration-files>

### 3.3 查看配置是否成功

可以通过 `dumpsys input` 命令查看 `INPUT_DEVICE_CLASS_EXTERNAL` 是否设置成功，如下 `dumpsys` 信息中根据 `IsExternal` 判断触摸设备 Device 3: Nuvoton HID Transfer 是否已经被设置成副屏。

Device 2: ilitek\_ts

Generation: 14

**IsExternal: false**

HasMic: false

Sources: 0x00001002

KeyboardType: 0

Device 3: Nuvoton HID Transfer

Generation: 17

**IsExternal: true**

HasMic: false

Sources: 0x00001002

KeyboardType: 0

## 4 副屏方向设置

### 4.1 Rk3568 副屏方向配置

RK3568 存在双屏和三屏的使用场景：

在双屏场景下，副屏通过 `persist.sys.rotation.einit-1`（属性值为 0,1,2,3）属性设置不同的方向例如 `setprop persist.sys.rotation.einit-1 1`，设置副屏旋转 90 度，设置该属性后需重启机器验证。

在三屏场景下, 存在一个屏幕做为主屏, 另外两个屏幕做为副屏情况。以如下 `dumpsys input` 信息为例, 系统中三个显示屏对应的 viewport 如下, 其中 Viewport INTERNAL 为主屏, 两个 Viewport EXTERNAL 为副屏, 每个 Viewport 都有独立的 `uniqueId`, `port` 值, 两个 EXTERNAL 副屏可以根据 `uniqueId` 或者 `port` 的值来进行区分。

Viewport INTERNAL: `displayId=0, uniqueId=local:0, port=0, orientation=0, logicalFrame=[0, 0, 1080, 1920], physicalFrame=[0, 0, 1080, 1920], deviceSize=[1080, 1920], isActive=[1]`

Viewport EXTERNAL: `displayId=0, uniqueId=local:1, port=1, orientation=1, logicalFrame=[0, 0, 1080, 1920], physicalFrame=[0, 0, 720, 1280], deviceSize=[720, 1280], isActive=[1]`

Viewport EXTERNAL: `displayId=0, uniqueId=local:2, port=2, orientation=0, logicalFrame=[0, 0, 1080, 1920], physicalFrame=[0, 0, 1440, 900], deviceSize=[1440, 900], isActive=[1]`

两个副屏需要分别设置 `persist.sys.rotation.einit-1` , `persist.sys.rotation.einit-2` (属性值为 0,1,2,3) 这两个属性来控制对应屏幕的方向。(这两个属性分别对应 `port1`, `port2` 两个副屏)。

## 4.2 RK3288,RK3399,PX30,RK3326 副屏方向配置

设置属性 `persist.sys.rotation.einit` (0,1,2,3) 控制副屏方向。

例如 `setprop persist.sys.rotation.einit 1`, 设置副屏旋转 90 度, 设置该属性后需重启机器验证。

## 5 其他开发配置

### 5.1 支持输入法在副屏显示

在 `device/rockchip/common/display_settings.xml` 中设置相应屏幕 `shouldShowIme` 为 `true`, 如下配置支持输入法在 `local:1` 的副屏上显示。

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<display-settings>
<config identifier="0" />
<display
    name="local:1"
    shouldShowIme="true"
    forcedDensity="240"/>
</display-settings>
```

## 5.2 副屏 DPI 设置

在 device/rockchip/common/display\_settings.xml 中设置 forcedDensity，例如：

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<display-settings>
<config identifier="0" />
<display
    name="local:1"
    shouldShowIme="true"
    forcedDensity="240"/>
</display-settings>
```

## 5.3 鼠标主副屏切换显示

设置 sys.mouse.presentation 为 1,打开该功能。异显状态时，开机鼠标默认在主屏显示，当鼠标移动到屏幕的边缘时，会自动切换到副屏的中心位置显示。