

Rockchip Android12 SDK 多屏异显开发说明

文件标识: RK-KF-YF-746

发布版本: V1.0.0

日期: 2022-12-05

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供, 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, 由其各自拥有者所有。

版权所有 © 2022 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。

瑞芯微电子股份有限公司

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

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

前言

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Android12.0, 适用芯片RK3288/RK339/RK3568/RK3588。

产品版本

芯片名称	安卓版本
RK3288	Android12
RK3399	Android12
RK3568	Android12
RK3588	Android12

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修订日期	修改说明
V1.0.0	董正勇	2022-12-05	初始版本

目录

Rockchip Android12 SDK 多屏异显开发说明

- 异显方案介绍
 - [Android Presentation](#)
 - [Android Activity 指定屏幕启动](#)
- 多屏异触配置
 - [原理说明](#)
 - [双屏异触配置](#)
 - [代码修改方式](#)
 - [配置触摸屏的idc文件](#)
 - [查看配置是否成功](#)
 - [三屏异触配置](#)
- 副屏初始方向配置
 - [RK3399/RK3288配置](#)
 - [RK3588/RK3568配置](#)
- 其他开发配置
 - [支持输入法在副屏显示](#)
 - [副屏DPI配置](#)
 - [异显状态鼠标主副屏切换](#)

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 activity 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 activity 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。

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。在该界面需点击对应副屏的checkbox选项，即可在副屏显示相应图片。

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 多显示器支持需要设备支持

<featurename="android.software.activities_on_secondary_displays" />。同时application或者activity需要支持分屏属性即或者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 根据指定应用名称以及display id，把应用启动到display id对应的屏幕上。

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 startApp(int displayId,String packageName,String className){
    ActivityOptions options = ActivityOptions.makeBasic();
    options.setLaunchDisplayId(displayId);
    Intent intent=new Intent();
    ComponentName componentName=null;
    componentName = new ComponentName(packageName, className);
    Log.d(TAG,"componentName="+componentName);
    intent.setComponent(componentName);
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    try {
        startActivity(intent, options.toBundle());
    }catch(ActivityNotFoundException e){
        e.printStackTrace();
    }
}
```

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）

多屏异触配置

原理说明

从如下的MontionEvent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。

双屏异触配置

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。

代码修改方式

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;
}
```

配置触摸屏的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。
- 如果未指定该值，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>

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

三屏异触配置

三屏场景中，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，TP 主副屏根据 INTERNAL/EXTERNAL 进行区分配置。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]`

三屏场景副屏 TP 配置步骤：

1 把 TP 配置成副屏（请参考双屏异触的配置方式，配置副屏 TP）。

2 区分两个副屏，把 TP 配置到指定屏幕，可以根据 `port` 值进行区分，配置 TP 到指定屏幕（例如名称为 `goodix-ts` 的 TP 配置到 `port 1` 的副屏上）。

```
--- a/services/inputflinger/reader/mapper/TouchInputMapper.cpp
```

```
+++ b/services/inputflinger/reader/mapper/TouchInputMapper.cpp
@@ -584,6 +584,11 @@ std::optional<DisplayViewport>
TouchInputMapper::findViewport() {
    ViewportType viewportTypeToUse;
    if (mParameters.associatedDisplayIsExternal) {
        viewportTypeToUse = ViewportType::VIEWPORT_EXTERNAL;
+       std::string name=getDeviceContext().getName();
+       const char *USB_DEVICE_NAME="goodix-ts";
+       if(strcmp(USB_DEVICE_NAME,name.c_str())==0){
+           return mConfig.getDisplayViewportByPort(1);
+       }
    } else {
        viewportTypeToUse = ViewportType::VIEWPORT_INTERNAL;
    }
}
```

副屏初始方向配置

RK3399/RK3288配置

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

设置属性 persist.sys.rotation.efull true 使副屏同显时满屏显示。

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或者热插拔后生效。

RK3588/RK3568配置

RK3588/RK3568存在多个副屏的使用场景，需要对多个副屏进行区分：

以三屏场景为例，该场景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）来控制对应屏幕的方向（这两个属性后缀分别对应port1,port2 两个副屏）。

两个副屏需要分别设置 persist.sys.rotation.efull-1，persist.sys.rotation.efull-2 (属性值为true)来控制对应屏幕同显时满屏显示（这两个属性后缀分别对应port1,port2 两个副屏）。

其他开发配置

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>
```

副屏DPI配置

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>
```

异显状态鼠标主副屏切换

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。