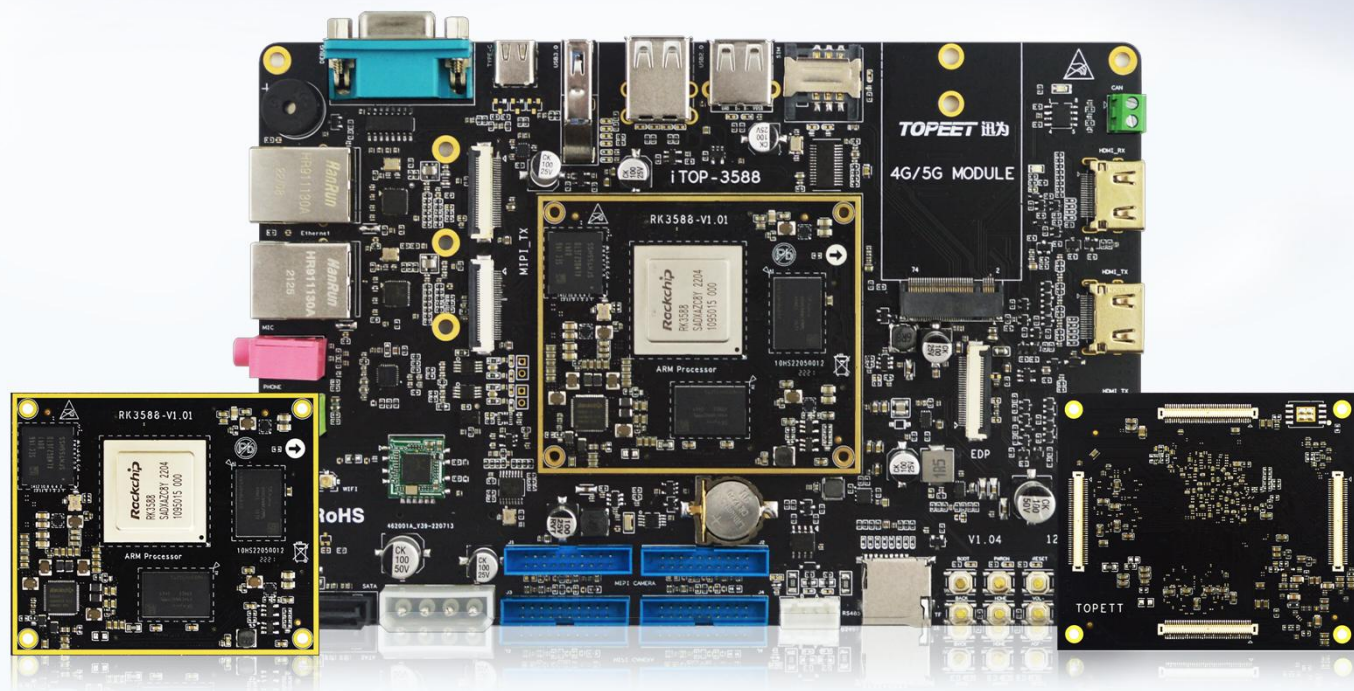


强大的 AI 能力 更快更强

超长供货周期 | 7X24 小时稳定运行 | 8K 视频编解码



iTOP-RK3588 开发板使用手册

八核 64 位 CPU | 主频 2.4GHz | NPU 算力 6T | 4800 安防级别 ISP

更新记录

更新版本	修改内容
V1.0	初版
V1.1	修改了屏幕选择部分，区分新旧版本的 LVDS 10.1 寸 1280*800 屏幕
V1.2	新增 1.2.10 副屏显示配置章节

目录

更新记录	2
目录	3
版权声明	4
更多帮助	5
第 1 章 Android 多屏显示	6
1.1 Android 多屏同显	6
1.2 Android 多屏异显	8
1.2.1 指定主副屏	9
1.2.2 异显方案介绍	9
1.2.3 Android Presentation	10
1.2.4 Android Activity 指定屏幕启动	16
1.2.5 支持输入法在副屏显示	19
1.2.6 副屏 DPI 设置	19
1.2.7 主屏幕可以触摸，副屏不触摸	19
1.2.8 多屏异触	20
1.2.9 鼠标异屏切换	21
1.2.10 副屏显示配置	22
第 2 章 Linux 多屏显示	24
2.1 设置屏幕	24
2.2 buildroot 多屏同显/异显	26
2.3 debian/ubuntu 多屏同显/异显/异触	27

版权声明

本文档版权归北京迅为电子有限公司所有。未经本公司书面许可，任何单位和个人无权以任何形式复制、传播、转载本文档的任何内容，违者将被追究法律责任。

更多帮助

注意事项与维护

- ❖ 请注意和遵循标注在产品上的所有警示和指引信息；
- ❖ 请勿带电插拔核心板及外围模块；
- ❖ 使用产品之前，请仔细阅读本手册，并妥善保管，以备将来参考；
- ❖ 请使用配套电源适配器，以保证电压、电流的稳定；
- ❖ 请勿在冷热交替环境中使用本产品，避免结露损坏元器件；
- ❖ 请保持产品干燥，如果不慎被任何液体泼溅或浸润，请立刻断电并充分晾干；
- ❖ 请勿使用有机溶剂或腐蚀性液体清洗本产品；
- ❖ 请勿在无尘、脏乱的环境中使用本产品，如果长期不使用，请包装好本产品；
- ❖ 如果在震动场景使用，请做好核心板与底板的固定，避免核心板跌落损坏；
- ❖ 请勿在通电情况下，插拔核心板及外围模块(特别是串口模块)；
- ❖ 请勿自行维修、拆解本产品，如产品出现故障应及时联系本公司进行维修；
- ❖ 请勿自行修改或使用未经授权的配件，由此造成的损坏将不予保修；

资料的更新

为了确保您的资料是最新状态，请密切关注我们的动态，我们将会通过微信公众号和 QQ 群推送。

关注“迅为电子”微信公众号，不定期分享教程、资料 and 行业干货及产品一线资料。

迅为新媒体账号

官网：<https://www.topeetboard.com>

知乎：<https://www.zhihu.com/people/topeetabc123>

CSDN：<https://blog.csdn.net/BeiJingXunWei>



售后服务政策

1. 如产品使用过程中出现硬件故障可根据售后服务政策进行维修
2. 服务政策：参见官方网售后服务说明
<https://www.topeetboard.com/sydyml/Service/bx.html>

送修地址：

1. 地址：北京市海淀区永翔北路 9 号中国航发大厦三层
2. 联系人：迅为开发板售后服务部
3. 电话：010-85270716
4. 邮编：100094
5. 邮寄须知：建议使用顺丰、圆通或韵达，且不接受任何到付

技术支持范围

1. 了解产品的软、硬件资源提供情况咨询
2. 产品的软、硬件手册使用过程中遇到的问题
3. 下载和烧写更新系统过程中遇到的问题
4. 产品用户的资料丢失、更新后重新获取
5. 产品的故障判断及售后维修服务。

PS：（由于嵌入式系统知识范围广泛，我们无法保证对各种问题都能一一解答，部分内容无法供技术支持，只能提供建议。）

技术支持

1. 周一至周五：（法定节假日除外）
上午 9:00 ~ 11:30 / 下午 13:30 ~ 17:30
2. QQ 技术交流群：
824412014
822183461
95631883
861311530

第 1 章 Android 多屏显示

RK3588 最多可以支持 7 个屏幕显示，这主要取决于 RK3588 芯片内置了 VOP 控制器，支持四路视频同显或异显，可有效提高行业定制的拓展性。本章节我们将介绍在 RK3588 的 Android12 系统上实现多屏显示。

1.1 Android 多屏同显

iTOP-RK3588 开发板支持以下屏幕

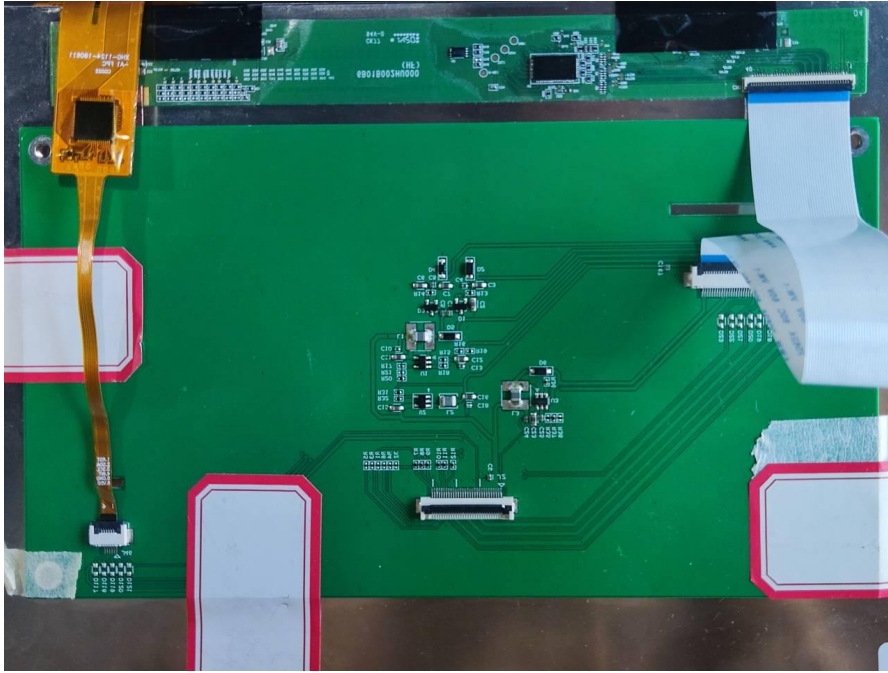
迅为 MIPI 7 寸屏幕
标准 HDMI 屏幕（通过 HDMI 线连接）
迅为 LVDS 7 寸屏幕
迅为 LVDS 10.1 寸 1024*600 屏幕
迅为 LVDS 10.1 寸 1280*800 屏幕
迅为 LVDS 10.1 寸 1280*800 新屏幕

然后修改 Android12 源码 kernel-5.10/arch/arm64/boot/dts/rockchip/topeet_screen_lcds.dts 中的设备树文件，如下图所示。LVDS 屏幕是 MIPI 通过转接板连接显示的，RK3588 不支持 LVDS 显示接口。

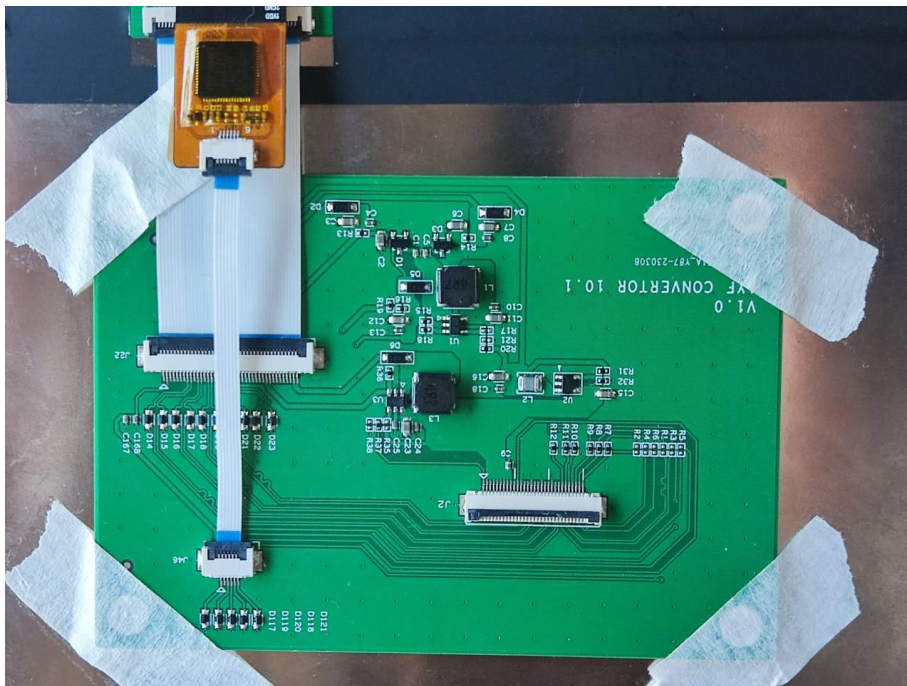
```
//#define LCD_TYPE_MIPI0 //VP2
#define LCD_TYPE_LVDS_10_1_1280x800_GT911 //VP2
//#define LCD_TYPE_LVDS_10_1_1280x800_GT9271 //VP2
//#define LCD_TYPE_LVDS_10_1_1024x600_GT911 //VP2
//#define LCD_TYPE_LVDS_7_0 //VP2
//#define LCD_TYPE_HDMI1 //VP1
//#define LCD_TYPE_HDMI0 //VP0
//#define LCD_TYPE_TYPEC_DP //VP1
//#define LCD_TYPE_MIPI1 //VP3
```

注：

其中#define LCD_TYPE_LVDS_10_1_1280X800_GT911 和#define
LCD_TYPE_LVDS_10_1_1280X800_GT9271 不同之处在于屏幕背板，可通过如下进行区分：
#define LCD_TYPE_LVDS_10_1_1280X800_GT911



```
#define LCD_TYPE_LVDS_10_1_1280X800_GT9271
```



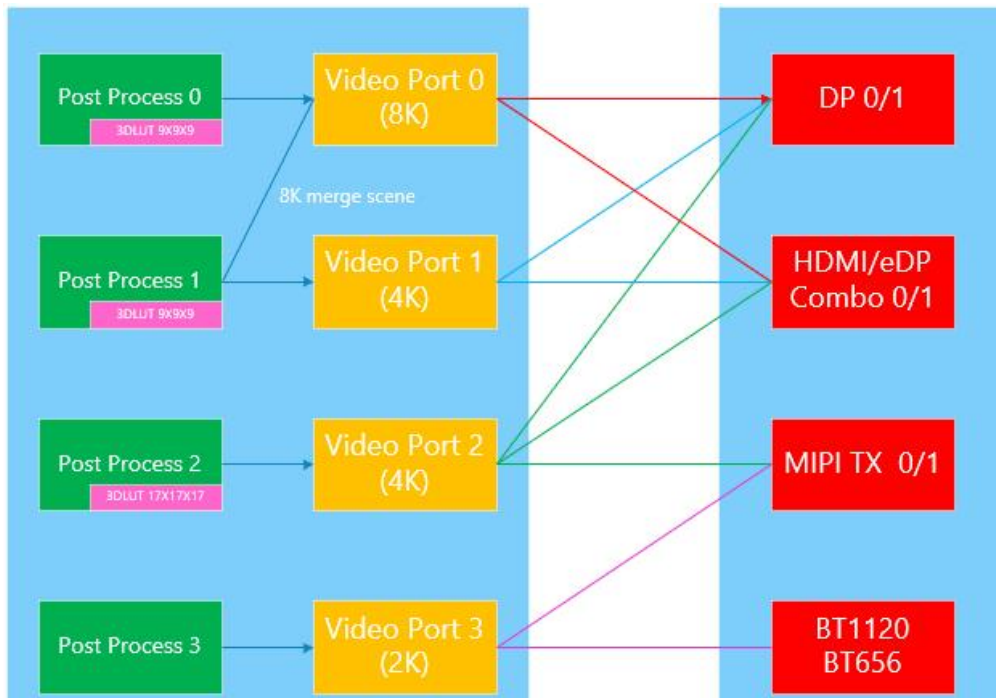
我们可以根据自己的需求来修改 `topeet_screen_choose.dtsi` 文件。总之，修改 `topeet_screen_choose.dtsi` 的规律是：打开所连接的屏幕对应的宏定义，rk3588 有四个显示通道 `vp0 vp1 vp2 vp3`，所打开的宏定义对应的 `vp` 通道不能重复。默认打开对应的宏定义为多屏同显。

如果打开的宏定义是 `#define LCD_TYPE_LVDS_10_1_1280x800_GT9271`，需要将 `kernel-5.10/arch/arm64/configs/rockchip_defconfig` 文件中的

CONFIG_TOUCHSCREEN_GT9271=y 宏定义使能，然后再编译源码

如果打开的宏定义是#define LCD_TYPE_LVDS_10_1_1280x800_GT911 和#define LCD_TYPE_LVDS_10_1_1024x600_GT911,需要将 kernel-5.10/arch/arm64/configs/rockchip_defconfig 文件中的 CONFIG_TOUCHSCREEN_GT9XX 宏定义使能，然后再编译源码。

RK3588 VP 和各显示接口的连接关系如下图所示。



1.2 Android 多屏异显

如果要支持多屏异显，首先要调试好双屏或多屏都可以正常显示，并且分别挂载在不同的vp上,可以参考上一个小节根据自己的需求打开 topeet_screen_choose.dtsi 设备树中的宏定义。默认开启后，多屏显示的是同一个画面(如果要支持双屏或者多屏异显，需要应用支持异显，才会显示不同的内容)。查看当前的屏幕显示状态输入以下命令

```
cat /sys/kernel/debug/dri/0/summary
```

如果双屏或者多屏的显示器分辨率比例不同，比如说主屏是 16:9，副屏是 4:3，那么副屏显示的畫面会拉伸或者缩小，原因是副屏的畫面是主屏映射过去的，所以副屏上显示的是主屏的分辨率。如果同样的屏幕情况下，应用支持异显，那么副屏会显示实际正常的分辨率。所以如果要支持双屏或者多屏同显并要求显示正常，双屏或者多屏的屏幕的分辨率比要一致。

1.2.1 指定主副屏

多屏异显的实际应用中，需要指定哪个是主屏，哪个是副屏。修改 Android12 源码目录下 3588-android12/device/rockchip/rk3588/rk3588_s/rk3588_s.mk 文件，添加如下内容：

如果要指定 MIPI 屏幕为主屏，HDMI 屏幕为副屏，如下设置：

```
PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.primary=DSI
PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.extend=HDMI-A
```

如果要指定 MIPI 屏幕为主屏，EDP 屏幕为副屏，如下设置：

```
PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.primary=DSI
PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.extend=eDP
```

如果需要定义多个主副屏时，需要用逗号隔开，比如指定 MIPI eDP 为主屏，HDMI-A 为副屏，设置如下：

```
PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.primary=DSI,eDP
PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.extend=HDMI-A
```

1.2.2 异显方案介绍

目前有两种异显方案分别是 [Android Presentation](#) 和 [Andorid Activity 指定屏幕启动](#)。

Android Presentation，需要 APP 应用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。

接下来我们详细讲述这两种方法。

1.2.3 Android Presentation

本小节讲解 Android Presentation 异显方案，瑞芯微和迅为提供了使用 Android Presentation 编写好的 APP 例程，接下来进行测试。

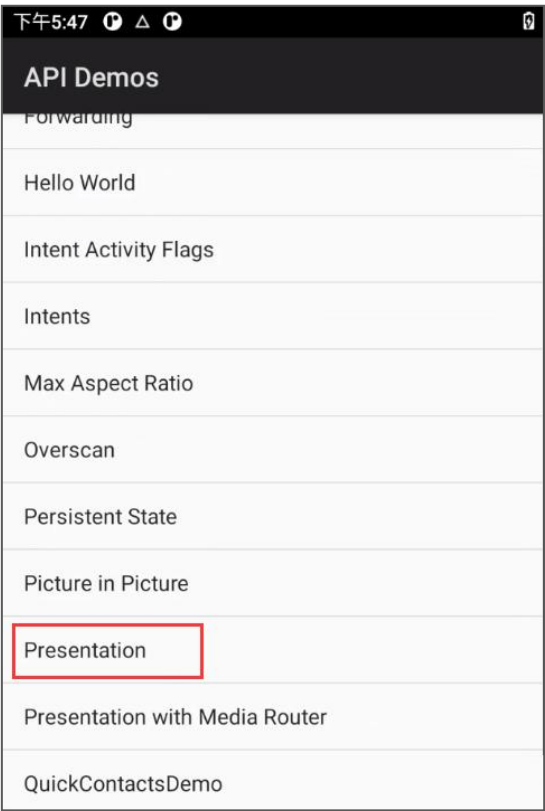
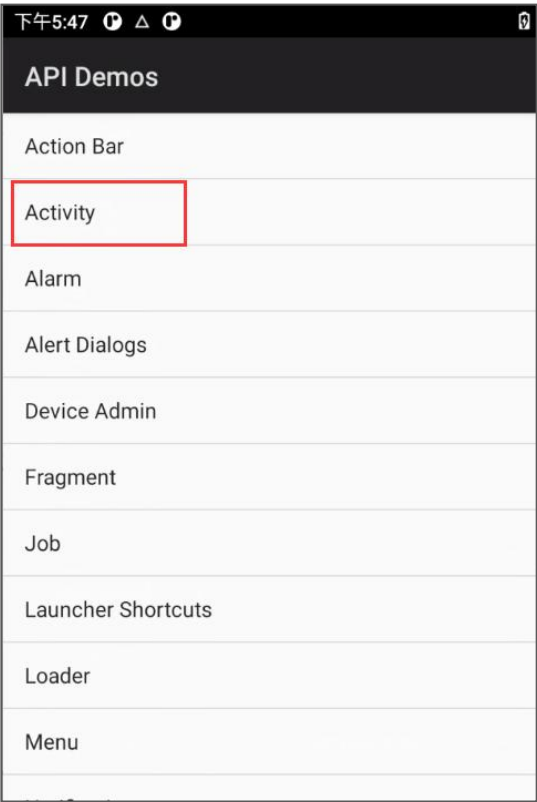
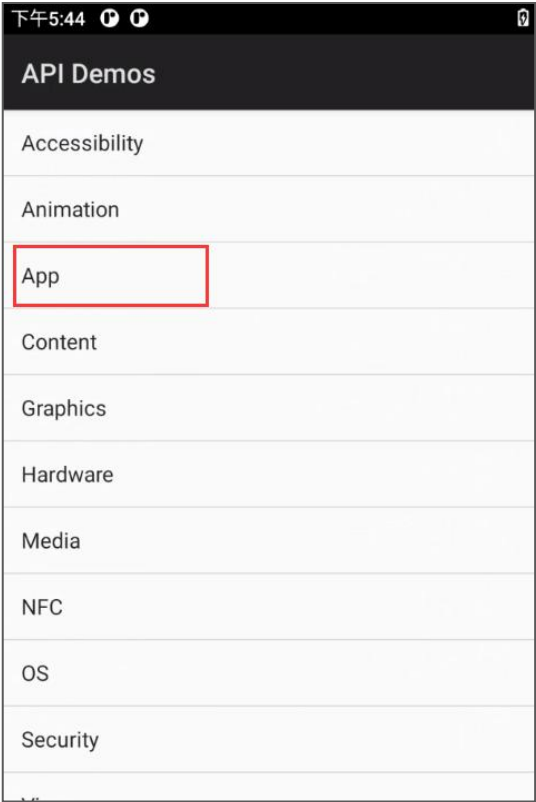
1.2.3.1 ApiDemos APP

在 Android12 源码中瑞芯微已经提供了使用 Presentation 接口的测试 APP，在源码 development/samples/ApiDemos 目录下，需要使用 mm 命令编译 APP。迅为提供了编译好的测试 APP 在网盘资料“[iTOP-RK3588 开发板/02_【iTOP-RK3588 开发板】/12_多屏显示配套资料/Android/ApiDemosAPK](#)”目录下。

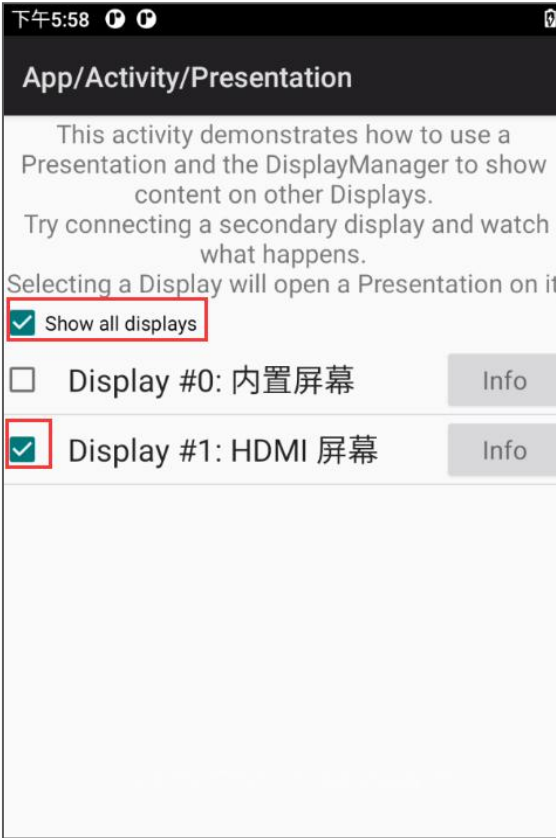
此 APP 源码在 Android 源码 3588-android12/development/samples/ApiDemos 目录下，进入此目录，输入“mm”命令进行编译，如下图所示：

```
topeet@ubuntu:~/Android/3588-android12$ cd development/samples/ApiDemos
topeet@ubuntu:~/Android/3588-android12/development/samples/ApiDemos$ mm
22:06:09 *****
22:06:09 You are building on a machine with 14.6GB of RAM
22:06:09
22:06:09 The minimum required amount of free memory is around 16GB,
22:06:09 and even with that, some configurations may not work.
22:06:09
22:06:09 If you run into segfaults or other errors, try reducing your
22:06:09 -j value.
22:06:09 *****
build/make/core/soong_config.mk:195: warning: BOARD_PLAT_PUBLIC_SEPOLICY_DIR has been deprecated. Use SYSTEM_EXT_PUBLIC_SEPOLICY_DIRS instead.
build/make/core/soong_config.mk:196: warning: BOARD_PLAT_PRIVATE_SEPOLICY_DIR has been deprecated. Use SYSTEM_EXT_PRIVATE_SEPOLICY_DIRS instead.
=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=12
TARGET_PRODUCT=rk3588_s
TARGET_BUILD_VARIANT=userdebug
TARGET_BUILD_TYPE=release
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=generic
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv8-a
TARGET_2ND_CPU_VARIANT=generic
```

然后使用 adb 命令安装 APP，依次点击 App->Activity->Presentation 选项，即可进入 Presentation 调用界面。如下图所示：



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。如下所示:



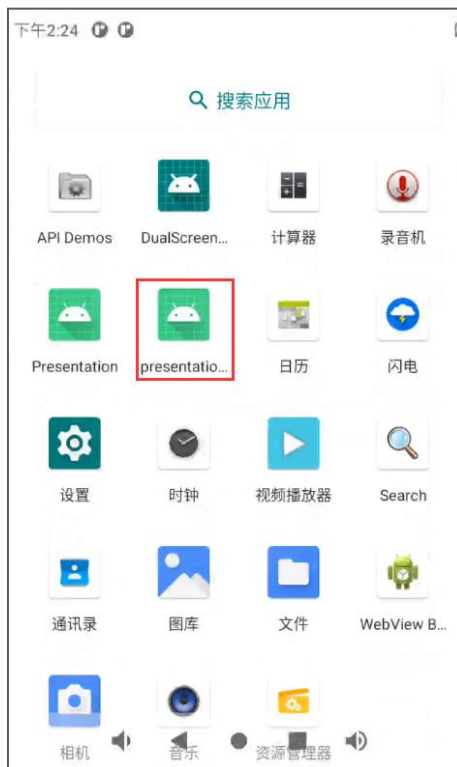
副屏 HDMi 屏幕显示图片，如下所示：



1.2.3.2 Presentation sampleAPP

此 APP 在网盘资料 “iTOP-RK3588 开发板/02_【iTOP-RK3588 开发板】/12_多屏显示配

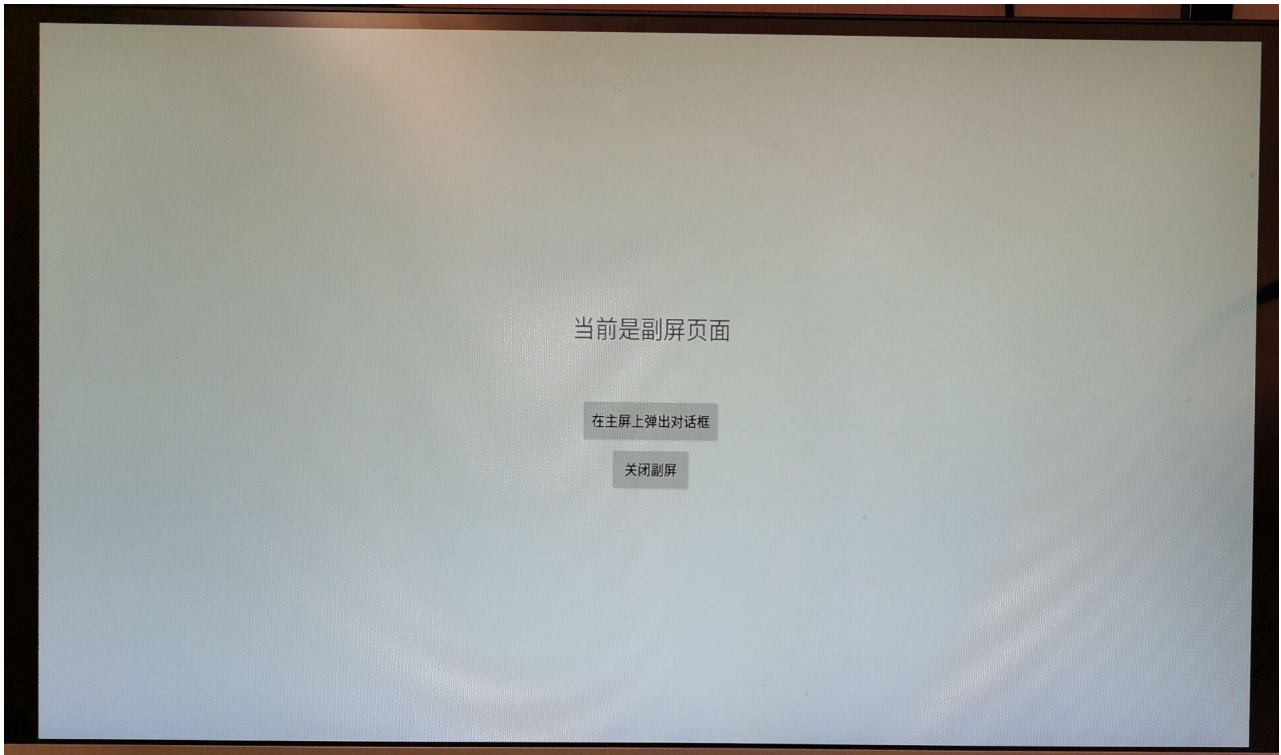
套资料/Android/Presentation sample”目录下，下载此 APP 工程，然后用 Androidstudio 打开，并编译 apk。编译成功之后，安装 APP，APP 安装成功之后如下图所示：



打开 APP 如下图所示：



点击“显示副屏”按钮，会在副屏显示，如下图所示：



APP 重要知识点讲解:

随着用户的需求增多，特别是对于一些 Android 平板电脑以及其他的一些 Android 智能设备可能有多个屏幕，用户不仅想要在主屏幕上显示内容，同样在第二屏幕上也要显示想要的内容，这样可以达到更好的体验效果。Google 也是不负众望在 Android4.2 版本以后提供了 Presentation 类，可以轻松实现在两块屏幕上同时显示不同的内容。Presentation 是一个特殊的 dialog，它的目的是显示内容到第二屏幕。在 Presentation 创建的时候关联一个目标设备，确定 Presentation 要显示在那个设备上，根据这个设备的信息来配置 Presentation 的 context 和 resources 信息。目前系统提供了两种方式与目标显示屏进行绑定。

1 通过 MediaRouter 接口获取并绑定:

选择显示 presentation 的设备最简单的方法就是使用 Media Router，media router 服务持续追踪在系统中哪个音频、视频线路是可用的，当 routes 被选择或者不被选择，或者更好的显示 presentation display 的线路发生改变后 media router 就会发送消息。所以一些应用程序可以自动监视这些消息来在首选设备上显示或取消 presentation。

首选显示 presentation 的设备是 Media Router 进行推荐的，如果应用想要显示内容在第二屏幕上就该使用该设备。下面将给出如何使用 media router 在主屏的设备创建和显示 presentation。

```
MediaRouter mediaRouter =(MediaRouter)
```

```
context.getSystemService(Context.MEDIA_ROUTER_SERVICE); MediaRouter.RouteInfo route =
mediaRouter.getSelectedRoute();

    if(route !=null){

        Display presentationDisplay = route.getPresentationDisplay();

        if(presentationDisplay !=null){

            Presentation presentation =new MyPresentation(context, presentationDisplay);

            presentation.show();

        }

    }

}
```

根据上面代码可以看出通过系统服务 Media Router 来选择一个合适的 route, 从 route 中来获取首选的 display, 如果 display 不为空就会创建该 presentation, 创建完成后调用 presentation 的 show 方法将该 presentation 显示出来。

2 通过 DisplayManager 接口获取并绑定

另外一个获取首选 display 的方法是直接使用 display Manager 来获取。display manager 服务提供方法枚举和描述系统中所有的设备, 包括可以显示 presentation 的设备。display manager 持续追踪系统中所有的设备, 然而, 并不是所有的设备都适合显示 presentation, 例如, 如果一个 Activity 试图在主屏幕上显示一个 presentation, 它可能会掩盖自己的内容, 这就像在 Activity 上打开一个 dialog 一样。

```
DisplayManager displayManager =(DisplayManager)
context.getSystemService(Context.DISPLAY_SERVICE); Display[] presentationDisplays
=displayManager.getDisplays(DisplayManager.DISPLAY_CATEGORY_PRESENTATION);

    if(presentationDisplays.length >0){

        // If there is more than one suitable presentation display, then we could consider
        // giving the user a choice. For this example, we simply choose the first display
        // which is the one the system recommends as the preferred presentation display.

        Display display = presentationDisplays[1]; // displays[1]是副屏

        Presentation presentation =new MyPresentation(context, presentationDisplay);

        presentation.show();

    }
```

```
}
```

根据上面代码可以看出通过系统服务 Display Manager 来获取所有适合显示 presentation 的 display 列表，如果获取到了多个适合 display 对象，我们选择副屏的 display。最后显示 presentation。

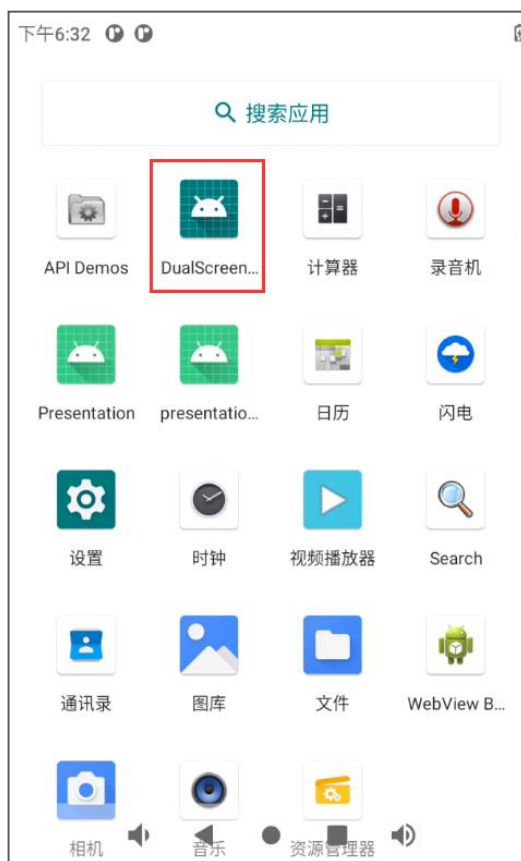
1.2.4 Android Activity 指定屏幕启动

本小节讲解 Android Activity 指定屏幕启动异显方案，瑞芯微提供了使用 Android Activity 指定屏幕启动编写好的 APP 例程，接下来进行测试。

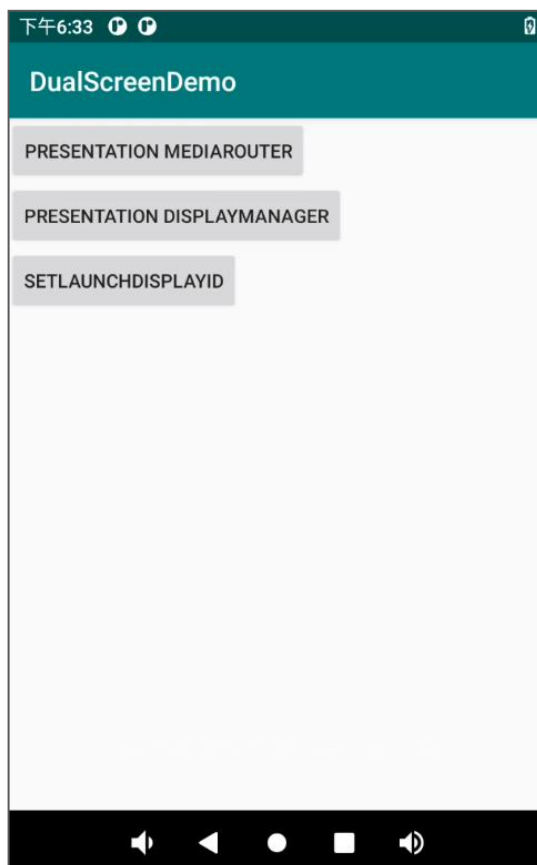
1.2.4.1 dualscreendemo APP

RK DualScreen 主要区别与 android presentation，在于它实现了应用的派发，允许厂商快速根据现有的 app 功能，进行模块的集成，减少开发周期和研发成本。示例代码为 DualScreenDemo.zip 工程，在网盘资料“[iTOP-RK3588 开发板/02_【iTOP-RK3588 开发板】/12_多屏显示配套资料/Android/DualScreenDemo](#)”目录下下载。

使用 Androidstudio 打开此工程代码，编译安装 APP，安装完成后如下所示：



打开此 APP，如下所示，有三个按钮。



第三个按钮使用了 Android Activity 指定屏幕启动，关键代码如下所示，第一个和第二个按钮使用了 Presentation。

```
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());
        Intent intent = new Intent("android.intent.action.MUSIC_PLAYER");
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent, options.toBundle());
    }
}
```

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，分别是 setLaunchDisplayId()和 getLaunchDisplayId()。

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。

在上述代码中使用 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。

1.2.5 支持输入法在副屏显示

修改 Android12 源码下的 device/rockchip/common/display_settings.xml 文件，如下所示：

```
1 <?xml version='1.0' encoding='utf-8' standalone='yes' ?>
2 <display-settings>
3 <config identifier="0" />
4 <display
5   name="local:1"
6   shouldShowSystemDecors="false"
7   shouldShowIme="true"
8   forcedDensity="240" />
9 </display-settings>
```

将设置相应屏幕 shouldShowIme 为 true,如上配置支持输入法在 local:1 的副屏上显示。

1.2.6 副屏 DPI 设置

修改 Android12 源码下的 device/rockchip/common/display_settings.xml 文件，如下所示：

```
1 <?xml version='1.0' encoding='utf-8' standalone='yes' ?>
2 <display-settings>
3 <config identifier="0" />
4 <display
5   name="local:1"
6   shouldShowSystemDecors="false"
7   shouldShowIme="true"
8   forcedDensity="240" />
9 </display-settings>
```

修改 forcedDensity 的值可以修改副屏的屏幕密度。

1.2.7 主屏幕可以触摸，副屏不触摸

如果想要实现主屏可以触摸，副屏不用触摸的功能。有两种解决方式：第一种是在 App 应用代码中实现。第二种需要修改 framework 层的源码。

第一种方式：

```
getWindow().setFlags(WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL,
    WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL);
```

第二种方式：

修改 Android12 源码 frameworks/native/services/inputflinger/reader/EventHub.cpp 文件，屏

蔽掉 `device->classes |= INPUT_DEVICE_CLASS_EXTERNAL;`属性，如下图所示：

```
1467 // Determine whether the device is external or internal.
1468 if (isExternalDeviceLocked(device)) {
1469     // device->classes |= INPUT_DEVICE_CLASS_EXTERNAL;
1470 }
```

1.2.8 多屏异触

多屏异触需要指定主副屏。这里使用修改 Android 源码的方式实现绑定副屏触摸芯片。这样设置以后，系统开机会自动将 ft5x06 触摸绑定到副屏上面去。修改文件：
`frameworks/native/services/inputflinger/reader/EventHub.cpp`，添加如下加红的代码。通过修改 `EventHub.cpp` 文件，在 `isExternalDeviceLocked` 函数中判断设备名称为指定的副屏设备（比如副屏的设备名称是 `generic ft5x06 (79)`），则返回 `true`。

```
bool EventHub::isExternalDeviceLocked(Device* device) {
    const char *I2C_DEVICE_NAME = "generic ft5x06 (79)";
    ALOGE("input device is '%s'", device->identifier.name.c_str());
    if (strcmp(device->identifier.name.c_str(), I2C_DEVICE_NAME) == 0) {
        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;
}
```

这里的 `I2C_DEVICE_NAME` 需要根据实际情况添加，用户可根据 `ALOGE` 打印信息在 `logcat` 中寻找识别到的芯片实际的名字。下面是添加了 `ALOGE` 打印信息后在 `logcat` 中得到的设备名字的截图：

```

06-02 10:15:34.242 D/EventHub( 446): No input device configuration file found for device 'goodix-ts'.
06-02 10:15:34.242 E/EventHub( 446): input device is 'goodix-ts'. aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
06-02 10:15:34.242 I/EventHub( 446): usingClocklock=true
06-02 10:15:34.242 I/HiidlServiceManagement( 446): Registered android.frameworks.schedulerservice@1.0::ISchedulingPolicyService/default
06-02 10:15:34.243 I/EventHub( 446): New device: id=1, fd=156, path='/dev/input/event3', name='goodix-ts', classes=0x14, configuration="", keyLayout="",
keyCharacterMap="", builtinKeyboard=false,
06-02 10:15:34.243 D/EventHub( 446): No input device configuration file found for device 'generic ft5x06 (79)'.
06-02 10:15:34.243 E/EventHub( 446): input device is 'generic ft5x06 (79)'. aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
06-02 10:15:34.244 I/EventHub( 446): usingClocklock=true
06-02 10:15:34.244 I/EventHub( 446): New device: id=2, fd=157, path='/dev/input/event2', name='generic ft5x06 (79)', classes=0x14, configuration="",
keyLayout="", keyCharacterMap="", builtinKeyboard=false,
06-02 10:15:34.244 D/EventHub( 446): No input device configuration file found for device 'adc-keys'.

```

上图中的 goodix-ts 和 generic ft5x06 (79)即为这里使用的触摸 IC 芯片的名字，将这个复制到代码中即可。修改完毕，重新编译 Android 镜像，用户测试的时候可以直接使用 ApiDemo APP 进行测试。

1.2.9 鼠标异屏切换

在多屏异显情况下，修改以下代码使鼠标箭头可以移动到需要的副屏上去进行操作。

我们需要修改 Android12 源码中

frameworks/native/services/inputflinger/reader/mapper/CursorInputMapper.cpp 文件

搜索"sys.mouse.presentation"，然后将下面的代码中 0 改为 1。

```

char mMousePresentation[PROPERTY_VALUE_MAX] = {1};

property_get("sys.mouse.presentation", mMousePresentation, "1");

if (strcmp(mMousePresentation, "1") == 0) {

    displayId = mDisplayId;

    float minX, minY, maxX, maxY;

    if (mPointerController->getBounds(&minX, &minY, &maxX, &maxY)) {

if(xCursorPosition==minX||xCursorPosition==maxX||yCursorPosition==minY||yCursorPosition==
maxY){

        displayId=getPolicy()->notifyDisplayIdChanged();

        mDisplayId=displayId;

    }

}

} else {

    displayId = mPointerController->getDisplayId();

```

```
}
```

然后重新编译 Android 源码，然后烧写镜像。

1.2.10 副屏显示配置

对于主屏的显示配置可参考迅为“[05_Android 系统开发/13【北京迅为】itop-3588 开发板 android 系统和应用开发手册](#)”手册 2.7 Android 旋转屏幕小节进行配置。

1.2.10.1 副屏全屏显示

设置副屏（副屏 1、副屏 2 等）是否全屏显示可通过修改 `persist.sys.rotation.efull-n`（ $n=1、2、3、\dots$ ）参数的取值进行配置，若配置为 `true` 则会进行全屏显示。

现以 MIPI 主屏、HDMI 副屏为例，修改如下：

在 `device/rockchip/rk3588/topeet_rk3588/topeet_rk3588.mk` 文件中添加如下内容，使得 HDMI 副屏可以全屏显示（由于副屏图像参考主屏宽高比进行缩放，若宽高比不一致，则出现黑边，所以默认 HDMI 副屏显示会有黑边）：

```
PRODUCT_PROPERTY_OVERRIDES += persist.sys.rotation.efull-1=true
```

```
38 #
39 ## add Rockchip properties
40 #
41 PRODUCT_PROPERTY_OVERRIDES += ro.sf.lcd_density=320
42 PRODUCT_PROPERTY_OVERRIDES += ro.wifi.sleep.power.down=true
43 PRODUCT_PROPERTY_OVERRIDES += persist.wifi.sleep.delay.ms=0
44 PRODUCT_PROPERTY_OVERRIDES += persist.bt.power.down=true
45 PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.primary=DSI
46 PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.extend=HDMI-A,eDP
47 PRODUCT_PROPERTY_OVERRIDES += persist.sys.rotation.efull-1=true
48 PRODUCT_PROPERTY_OVERRIDES += sys.mouse.presentation=1
49
```

修改完成后，保存退出，重新编译烧写镜像即可。

注：由于 mipi 竖屏显示与 HDMI 宽高比相差较大，HDMI 设置全屏显示后拉伸较明显！

1.2.10.2 副屏显示方向

副屏的旋转角度可通过 `persist.sys.rotation.cinit-n`（ $n=1、2、3、\dots$ ）属性控制，该属性用于设置副屏（副屏 1、副屏 2 等）的显示方向，可取值为 0/1/2/3，分别表示将副屏显示方向旋转（顺时针）0 度、90 度、180 度、270 度。

现以 MIPI 主屏、HDMI 副屏为例，修改如下：

在 `device/rockchip/rk3588/topeet_rk3588/topeet_rk3588.mk` 文件中添加如下内容，使得

HDMI 副屏的显示方向旋转 90 度：

```
PRODUCT_PROPERTY_OVERRIDES += persist.sys.rotation.einit-1=1
```

```
PRODUCT_PROPERTY_OVERRIDES += persist.sys.rotation.efull-1=false
```

```
38 #
39 ## add Rockchip properties
40 #
41 PRODUCT_PROPERTY_OVERRIDES += ro.sf.lcd_density=320
42 PRODUCT_PROPERTY_OVERRIDES += ro.wifi.sleep.power.down=true
43 PRODUCT_PROPERTY_OVERRIDES += persist.wifi.sleep.delay.ms=0
44 PRODUCT_PROPERTY_OVERRIDES += persist.bt.power.down=true
45 PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.primary=DSI
46 PRODUCT_PROPERTY_OVERRIDES += vendor.hwc.device.extend=HDMI-A,eDP
47 PRODUCT_PROPERTY_OVERRIDES += persist.sys.rotation.einit-1=1
48 PRODUCT_PROPERTY_OVERRIDES += persist.sys.rotation.efull-1=false
49 PRODUCT_PROPERTY_OVERRIDES += sys.mouse.presentation=1
50
```

修改完成后，保存退出，重新编译烧写镜像即可，HDMI 会旋转 90° 竖屏显示。

第 2 章 Linux 多屏显示

2.1 设置屏幕

iTOP-3568 开发板支持以下种类屏幕：

迅为 MIPI 7 寸屏幕
标准 HDMI 屏幕（通过 HDMI 线连接）
迅为 LVDS 7 寸屏幕
迅为 LVDS 10.1 寸 1024*600 屏幕
迅为 LVDS 10.1 寸 1280*800 屏幕
迅为 LVDS 10.1 寸 1280*800 新屏幕

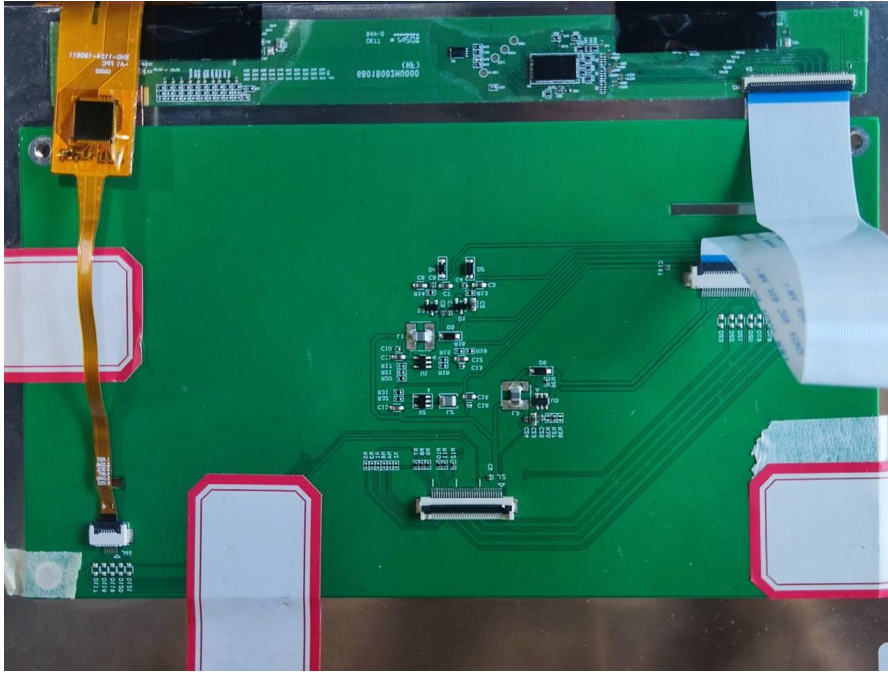
然后修改 Linux 源码 kernel/arch/arm64/boot/dts/rockchip/topeet_screen_lcds.dts 中的设备树文件，如下图所示。LVDS 屏幕是 MIPI 通过转接板连接显示的，RK3588 不支持 LVDS 显示接口。

```
//#define LCD_TYPE_MIPI0 //VP2
#define LCD_TYPE_LVDS_10_1_1280x800_GT911 //VP2
//#define LCD_TYPE_LVDS_10_1_1280x800_GT9271 //VP2
//#define LCD_TYPE_LVDS_10_1_1024x600_GT911 //VP2
//#define LCD_TYPE_LVDS_7_0 //VP2
//#define LCD_TYPE_HDMI1 //VP1
//#define LCD_TYPE_HDMI0 //VP0
//#define LCD_TYPE_TYPEC_DP //VP1
//#define LCD_TYPE_MIPI1 //VP3
```

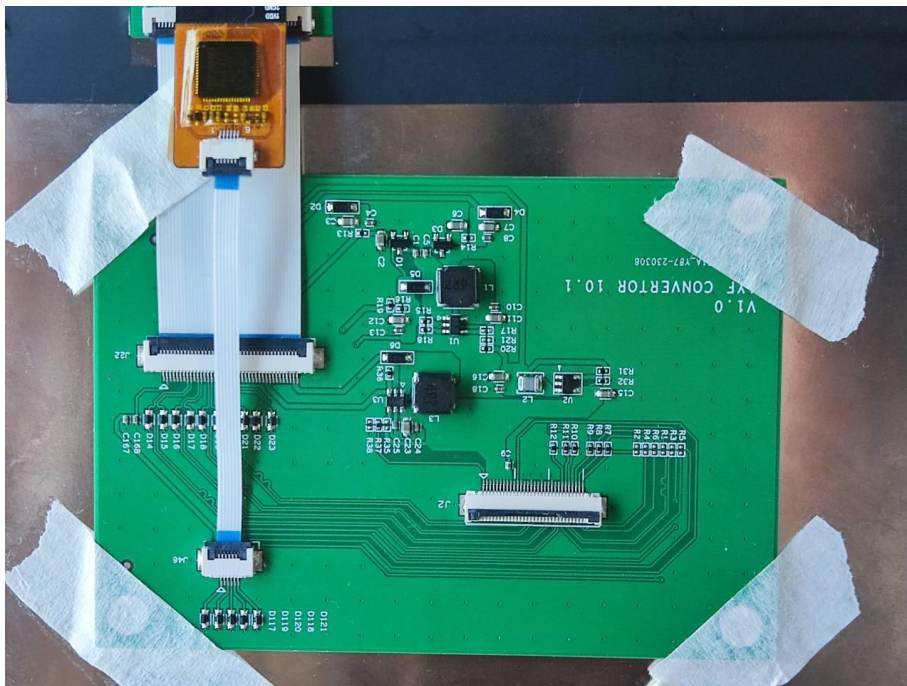
注：其中#define LCD_TYPE_LVDS_10_1_1280X800_GT911 和#define

LCD_TYPE_LVDS_10_1_1280X800_GT9271 不同之处在于屏幕背板，可通过如下进行区分：

#define LCD_TYPE_LVDS_10_1_1280X800_GT911

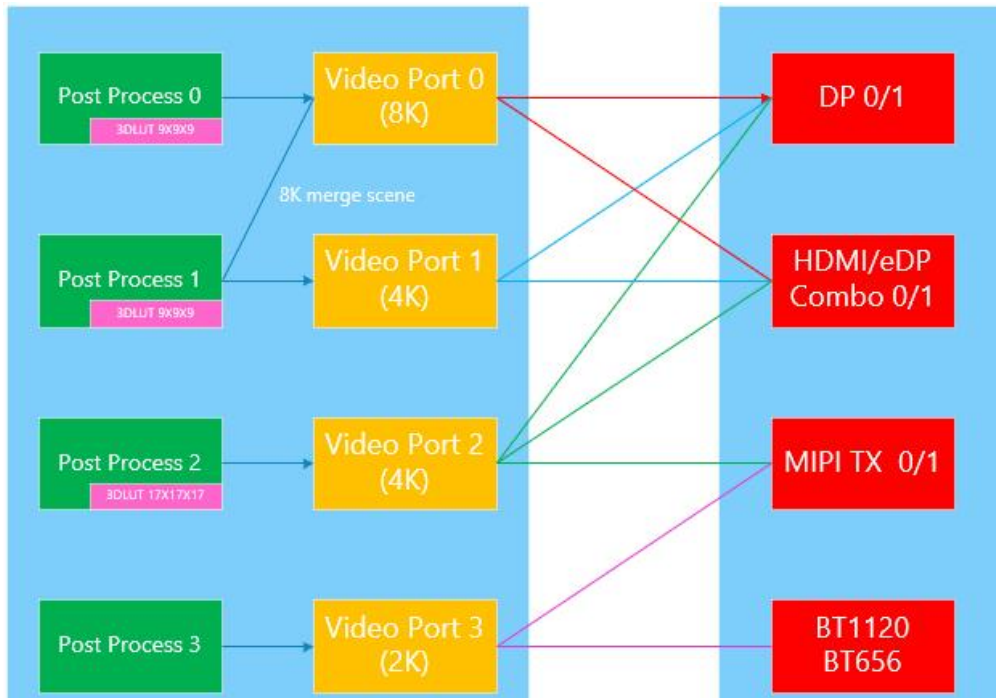


```
#define LCD_TYPE_LVDS_10_1_1280X800_GT9271
```



我们可以根据自己的需求来修改 `topeet_screen_choose.dtsi` 文件。总之，修改 `topeet_screen_choose.dtsi` 的规律是：打开所连接的屏幕对应的宏定义，rk3588 有三个显示通道 `vp0 vp1 vp2 vp3`，所打开的宏定义对应的 `vp` 通道不能重复。默认打开对应的宏定义为多屏同显。

RK3588 VP 和各显示接口的连接关系如下图所示。



2.2 buildroot 多屏同显/异显

Buildroot SDK 的 Weston 支持多屏同异显及热拔插等功能，不同显示器屏幕的区分根据 drm 的 name (通过 `/sys/class/drm/card0-name` 获取)，相关配置通过环境变量设置，如下所示：

```
# /etc/init.d/S50launcher
start)
...
export WESTON_DRM_PRIMARY=HDMI-A-1 # 指定主显为 HDMI-A-1
export WESTON_DRM_MIRROR=1 # 使用镜像模式(多屏同显)，不设置此环境变量即为
异显
export WESTON_DRM_KEEP_RATIO=1 # 镜像模式下缩放保持纵横比,不设置此变量即
为强制
全屏
export WESTON_DRM_PREFER_EXTERNAL=1 # 外置显示器连接时自动关闭内置显示
器
export WESTON_DRM_PREFER_EXTERNAL_DUAL=1 # 外置显示器连接时默认以第一
个外显为
主显
```

```
weston --tty=2 -B=drm-backend.so --idle-time=0&
```

镜像模式缩放显示内容时需要依赖 RGA 加速。同时也支持在开发板的 /etc/xdg/weston/weston.ini 文件的 output 段单独禁用指定屏幕:

```
# /etc/xdg/weston/weston.ini
[output]
name=DSI-1
mode=off
# off|current|preferred|<WIDTHxHEIGHT@RATE>
```

2.3 debian/ubuntu 多屏同显/异显/异触

“xrandr”是一款官方的 RandR 扩展配置工具，它可以设置屏幕显示的大小、方向、镜像等。对于多屏幕显示的情况，可以使用 xrandr 进行管理。输入命令 “xrandr --help” 查看帮助信息。

RK3588 Debian11 系统支持 xrandr 命令管理多屏扩展显示。同时接俩个或者多个屏幕，系统会默认非热插拔显示设备作为主屏，默认会设置两个屏幕为扩展显示。输入 “xrandr” 命令可以查看屏幕信息。

可以使用 xrandr 在两个屏幕中选择哪个做主屏，哪个做副屏。输入以下命令，设置 HDMI 为主屏，DSI-0 屏幕为副屏。异显模式下支持 On right、Above、On left、Below 四种模式。其中--right-of 可以替换成 left-of, below,same-as, preferred, off 等等，从而实现不同的屏幕显示功能。

```
xrandr --output DSI-0 --right-of HDMI-1 --auto
```

输入以下命令将 HDMI-1 屏幕和 LVDS-1 屏幕设置成同显。

```
xrandr --output DSI-0 --same-as LVDS-1 --auto
```

xinput 命令用于管理系统的输入设备，首先必须安装 xinput

```
apt-get update
```

```
apt-get install xinput
```

输入 “xinput” 命令可以查看输入设备，如下图所示：

```
root@linaro-alip:~# xinput
[ Virtual core pointer                id=2    [master pointer (3)]
  ↳ Virtual core XTEST pointer        id=4    [slave pointer (2)]
  ↳ generic ft5x06 (30)                id=9    [slave pointer (2)]
[ Virtual core keyboard               id=3    [master keyboard (2)]
  ↳ Virtual core XTEST keyboard        id=5    [slave keyboard (3)]
  ↳ adc-keys                           id=6    [slave keyboard (3)]
  ↳ rk805 pwrkey                       id=7    [slave keyboard (3)]
  ↳ fdd70030.pwm                      id=8    [slave keyboard (3)]
root@linaro-alip:~#
```

使用 xinput 对触摸设备和显示设备进行绑定。Ft5x06 触摸设备的 id 是 9，将其与 DSI-0 绑定。同理，我们将其他屏幕的触摸 id 和对应的屏幕也进行绑定。

```
xinput map-to-output 9 DSI-0
```