

## RK3399 VR Android 参数配置和调试说明

**V1.0\_20170106**

文件状态： [ ] 正在修改 [√] 正式发布	当前版本：	V1.0
	作 者：	邓敬威/张文平
	完成日期：	2017-01-06
	审 核：	
	完成日期：	

福州瑞芯微电子股份有限公司

Fuzhou Rockchips Semiconductor Co., Ltd

(版本所有,翻版必究)

## 更新记录

[illegible]

# 目 录

1 宏配置 .....	4
2 系统属性说明.....	4
2.1 主屏方向.....	5
2.2 次屏方向.....	5
2.3 头盔 LCD 的刷新方向 .....	6
2.5 视频是否支持 ATW.....	6
2.6 双屏 LCD 扫描方向 .....	6
2.7 VOP 取反.....	7
2.8 VR 光学参数（FOV）调节.....	7
3 VR 系统调试 .....	7

# 1 宏配置

宏参数名	功能	备注
DUAL_SCREEN	表示 VR 头盔的屏为单屏还是双屏	True: 双屏 False: 单屏
BOARD_USE_AFBC_LAYER	是否使能 AFBC 功能，该功能能够提升显示效率，降低系统负载。 但是需要满足两个条件： 1. VR 头盔的屏为单屏 2. 不需要两个显示屏，例如 hdmi 和头盔同时显示或者头盔和 rk3399 上外接的 lcd 同时显示	True: 支持 AFBC 功能 False: 不支持

# 2 系统属性说明

以下提到的屏幕或显示默认都是指主显示，涉及次显的会具体指出。区分主显和副显（次显）的方法是查看 kernel dts 配置中 vopb\_rk\_fb 和 vopl\_rk\_fb 节点的 rockchip,prop 值为 PRMRY 还是 EXTEND，PRMRY 表示主显，EXTEND 表示次显。对于 VR 来说，目前内核以下三个参考 dts：

✧ 分体 VR 头盔

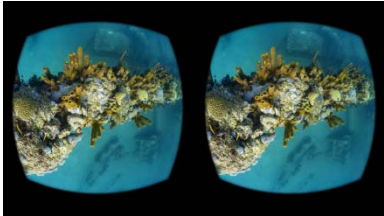
arch/arm64/boot/dts/rockchip/rk3399-disvr-android.dts  
arch/arm64/boot/dts/rockchip/rk3399-box-rev1-disvr.dts

✧ 一体 VR 头盔

arch/arm64/boot/dts/rockchip/rk3399-vr-android.dts

上述 dts 默认配置都是 vopb（vop big/lcdc0）为主显，vopl（vop little/lcdc1）为次显，所以这边我们默认定义主显对应 Vop Big（LCDC0 或者 VopB），次显或者副显对应的是 Vop little（LCDC1 或者 VopL）。

为了确保 VR 的显示效果，我们要求头盔接的是主显（LCDC0/VopB），具体的配置方法请参考《RK3399\_VR 分体机\_软件开发指南.pdf》。当头盔显示效果如下图所示，说明头盔显示效果已经正确，可以忽略下述第一节。



对于分体机，这里还需要单独说明下，分体机分为主板（rk3399 硬件板）+ 头盔板，目前我们默认的产品形态为，主板不带 lcd 屏，通过 typec 线或者 hdmi 线连接头盔板，所以我们有上述主显和次显的默认定义，如果需要主板上带屏，则需要另外配置，我们系统默认不支持这种形态。

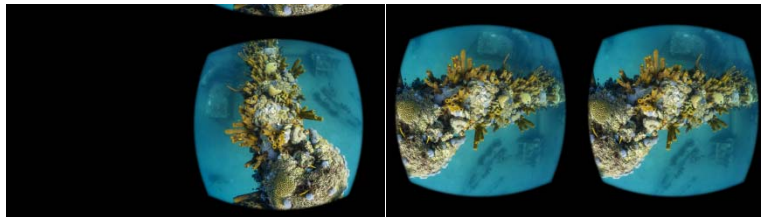
## 2.1 主屏方向

### ✧ `ro.sf.hwrotation` 属性

0, 90, 180, 270 初始化主屏旋转角度, 与 `persist.display.portrait` 配合使用, 因为硬件上屏幕正装反装的情况都会存在, 所以有的时候需要把这个属性设置为 180。如果在集成的时候发现屏幕方向不对, 可以调节这个属性来控制。

### ✧ `persist.display.portrait` 属性

true 表示主屏是按照竖屏显示, false 表示横屏横屏显示;如下左图为属性 false 的情况, 右图属性为 true 的情况。当使用 `ro.sf.hwrotation` 属性设置 0, 90, 180, 360 都无法调整到正确的效果的时候, 就需要切换 `persist.display.portrait` 属性来配合调整。



## 2.2 次屏方向

### ✧ `ro.orientation.einit` 属性

0, 90, 180, 270 初始化次屏 (或者叫次显, 如 HDMI, DP 等扩展显示, 下同) 的旋转方向。作用与第 2 点类似。

### ✧ `ro.rotation.external` 属性

上面说到的两个属性 `ro.sf.hwrotation` 和 `ro.orientation.einit` 表示的是初始化的方向, 如果某些 app 可以设定方向或者需要根据 `gsensor` 方向进行实时调整, 则需要考虑加上这里的 `ro.rotation.external` 属性, 具体的原则如下:

android 系统默认只有主显会根据框架上报的 `gsensor` 或者 app 设置的旋转方向进行实时的旋转, 如果要求次显也根据上述方向进行旋转必须配置该属性, true 表示次屏可以随着系统 `gsensor` 或者 app 触发旋转, false 则不旋转。

举个例子:

根据之前我们的描述, 我们默认设定头盔为主显, 所以头盔能够根据 app 设置的旋转方向进行旋转, 但是如果我外接一个 hdmi 电视, 并且安装了一个 app, 这个 app 在启动的时候会将方向颠倒 180 度, 则此时 hdmi 电视因为是次显将无法根据 app 设置的旋转方向进行旋转, 我们看到的现象将是头盔的方向是正常的, 但是 hdmi 电视的方向是不正常的。

系统默认将这个值定义为 true, 一般情况下, VR 的 apk 不支持 `gsensor` 方向的旋转 (这里的旋转指的是从横屏变成竖屏, 也就是说旋转 0/90/180/270), 所以我们不需要太关心这个属性。

## 2.3 头盔 LCD 的刷新方向

VR 头盔我们默认为主显，也就是说接的是 VOP Big (LCDC0/VopB)。

### ✧ sys.vr.vsync 属性

属性值默认为 0, 表示从 fb0 (默认对应的是 lcdc0, 也就是主显) 获取 vsync 信号;  
属性值为 5 则表示从 fb5 (默认对应的是 lcdc1, 也就是次显) 获取 vsync 信号;  
RK VR 应用会根据该属性值, 在应用打开初始化的时候去获取 fb0 或者 fb5 的  
dsp\_mode 和 fps 刷新率的值, 如配错, 会有撕裂现象。

## 2.5 视频是否支持 ATW

### ✧ vr.video.direct:

true 表示视频播放时关闭 ATW。

false 则表示视频播放打开 ATW; 其他场景默认是有 ATW 的, 无法关闭。

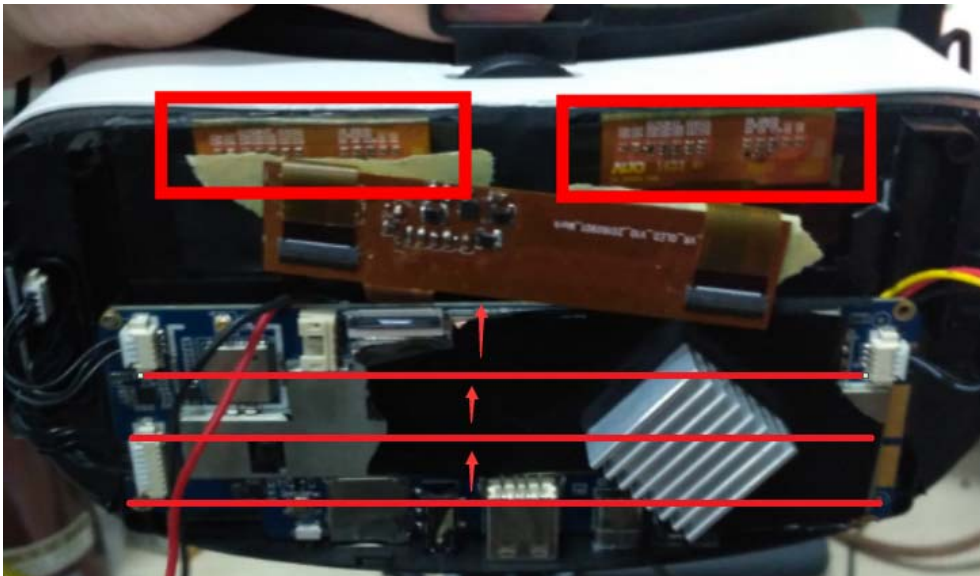
默认为 true。

推荐保持默认值 true, 在视频场景关闭 ATW, 可以有效降低功耗, 但是可能会影响视频效果。

## 2.6 双屏 LCD 扫描方向

### ✧ sys.vr.scan:

该属性**仅针对双屏机器**, 单屏机器的屏幕扫描方式是左右方向的, 不需要考虑此属性。

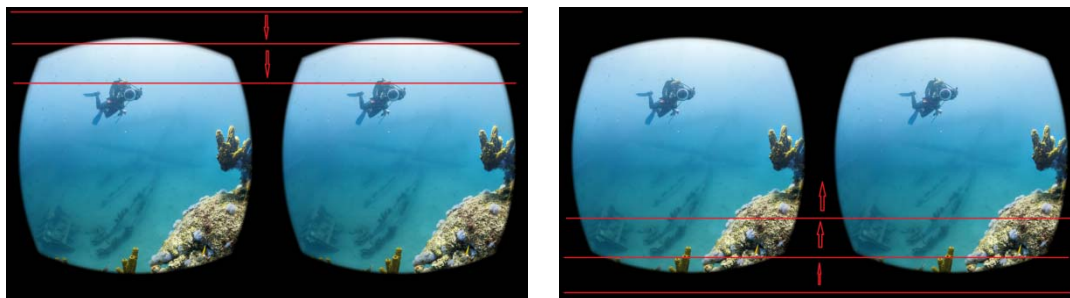


由于双屏机器在硬件设计上, 屏幕的物理方向有上下相反的两种接法, 对应的屏幕扫描方式也有从下往上扫和从上往下扫两种方式。软件上通过 sys.vr.scan 配置, 有 0 和 1 两种。

如上图所示, 该双屏工程机的屏幕排线接在上方, 因此屏幕中的扫描方向就是从下往上扫描, sys.vr.scan 属性应该设置为 0。如果硬件上屏幕排线接在下方 (暂时没有样例图), 那么屏幕的扫描方向就和示例图的工程机相反, sys.vr.scan 属性应该设置为 1。

sys.vr.scan 属性如配置错误，会有撕裂现象。该属性的配置还要配合第 2.7 节 vop 取反的配置，上述讨论的配置都是在 vop 取反关闭的情况下讨论的，如打开 vop 取反，则上述 sys.vr.scan 属性配置也要取反。双屏机器可切换该属性的值进行调试。

## 2.7 VOP 取反



在 VR 系统设计中我们加入了一些优化设计，降低撕裂的概率，例如这边所讲的 vop 取反。上面两张图的红色辅助线表示扫描线。正常情况下，不开启 vop 取反，屏幕的扫描方向是从上往下，如左图所示；开启 vop 取反，屏幕的扫描方向则是从下往上，如右图所示。双屏机器**建议**开启 vop 取反，开启 vop 取反则必须关闭 AFBC，可以降低撕裂的概率。单屏机器**不建议**开启 vop 取反，建议开启 AFBC。

开启方法：

sys.vr.pmirror 为 1 则主屏开启镜像，0 则关闭

sys.vr.emirror 为 1 则次屏开启镜像，0 则关闭

## 2.8 VR 光学参数（FOV）调节

FOV 参数调节请参考《RKVR 光学参数调节》文档

# 3 VR 系统调试

✧ VR Log 使能和 log 具体含义：

```
setprop sys.vr.log 1
```

```
logcat -c;logcat | grep VRJni
```

(VRJ 大写 ni 小写)

然后重新启动调试的 VR 应用，可以看到如下打印（以 rk3399 为例）：

```

D VRJni : current platform panle type = 0
D VRJni : panelType = 0 1
D VRJni : BindThreadsToCpu45 : Current platform = rk3399
D VRJni : BindThreadsToCpu45 -> bind process sched_setaffinity success!
D VRJni : BindThreadsToCpu45 -> bind thread 12788 sched_setaffinity success!
D VRJni : BindThreadsToCpu45 -> bind thread 12794 sched_setaffinity success!
D VRJni : BindThreadsToCpu45 -> bind thread 12795 sched_setaffinity success!
D VRJni : BindThreadsToCpu45 -> bind thread 12796 sched_setaffinity success!
D VRJni : BindThreadsToCpu45 -> bind thread 12797 sched_setaffinity success! 2
D VRJni : BindThreadsToCpu45 -> bind thread 12798 sched_setaffinity success!
D VRJni : BindThreadsToCpu45 -> bind thread 12799 sched_setaffinity success!
D VRJni : BindThreadsToCpu45 -> bind thread 12800 sched_setaffinity success!
D VRJni : BindThreadsToCpu45 -> bind thread 12801 sched_setaffinity success!
D VRJni : BindThreadsToCpu45 -> bind thread 12803 sched_setaffinity success!
D VRJni : BindThreadsToCpu45 -> bind thread 12804 sched_setaffinity success!
D VRJni : current platform screen flush rate = 60 3
D VRJni : --> successfully open /sys/class/graphics/fb0/vsync 4

```

上述图中的 log 分成 3 个部分，每个部分的含义分别说明如下：

1 --> 当前系统配置的单双屏状态值。0 代表单屏，1 代表双屏，配置错误会有撕裂。参考之前 dsp\_mode 配置小点的内容。

2 --> 线程初始化操作，全部 success 为正常，如有 error 出现，会导致撕裂。

如果出现 error，请执行 `cat /dev/cpuset/foreground/cpus`，结果如下：

```

126|shell@rk3399_64_vr:/ $ cat /dev/cpuset/foreground/cpus
0-5

```

如果不是这个结果请检查代码：device/rockchip/rk3399/init.vr.rc 是否有如下配置，如果没有请确认代码是否更新成功，如果是新加的 init 文件请手动添加。

```

on boot

# update cpusets feature nodes for rk3399 vr
write /dev/cpuset/foreground/cpus 0-5
write /dev/cpuset/foreground/boost/cpus 0-5
write /dev/cpuset/background/cpus 0
write /dev/cpuset/system-background/cpus 0-3

```

3 --> 当前系统配置的 fb 的刷新率值。对应机器的 /sys/class/graphics/fb0/fps 或 /sys/class/graphics/fb5/fps。参考之前 sys.vr.vsync 属性配置的内容。如果 sys.vr.vsync 配置正确，而刷新率不是目标刷新率，请验证是不是 dts 中屏幕的 clock 值是否配置有误。

4 --> 当前系统配置的 vsync 节点打开是否成功。fb0 还是 fb5 取决于 sys.vr.vsync 属性。

✧ 查看应用实时渲染时间：

```
setprop sys.vr.log 1
```

```
logcat -c;logcat | grep VRJni
```



```

app=16.5, atw=16.5, wait=(2.2+4.4), draw=(4.0+4.2 avg=4.0+4.0=8.0)
app=16.3, atw=17.0, wait=(1.9+4.4), draw=(4.0+4.0 avg=4.0+4.0=8.0)
app=16.3, atw=16.5, wait=(1.9+4.5), draw=(3.9+3.8 avg=4.0+4.0=8.0)
app=17.2, atw=16.4, wait=(2.2+4.4), draw=(4.1+3.9 avg=4.0+4.0=8.0)
app=16.4, atw=16.8, wait=(2.2+4.1), draw=(4.1+4.0 avg=4.0+4.0=8.0)
app=16.6, atw=16.6, wait=(2.2+4.4), draw=(3.9+3.9 avg=4.0+4.0=8.0)
app=16.4, atw=16.7, wait=(1.9+4.4), draw=(4.1+4.0 avg=4.0+4.0=8.0)
app=16.4, atw=16.9, wait=(1.6+4.4), draw=(4.1+4.3 avg=4.0+4.0=8.0)
app=25.3, atw=17.5, wait=(0.7+0.7), draw=(7.3+6.4 avg=7.1+7.0=14.0) #right=7.3 #left=6.4
app=24.5, atw=15.4, wait=(1.6+1.8), draw=(6.8+7.1 avg=7.1+7.0=14.0) #right=6.8 #left=7.1
app=25.6, atw=17.5, wait=(0.8+1.1), draw=(7.1+7.1 avg=7.1+7.0=14.0) #right=7.1 #left=7.1
app=25.6, atw=16.5, wait=(1.3+0.9), draw=(7.1+6.2 avg=7.1+7.0=14.0) #right=7.1 #left=6.2
app=25.3, atw=15.7, wait=(1.9+1.0), draw=(7.7+7.2 avg=7.1+7.0=14.0) #right=7.7 #left=7.2
app=25.3, atw=18.0, wait=(0.6+1.3), draw=(7.1+6.8 avg=7.1+7.0=14.0) #right=7.1 #left=6.8
app=25.7, atw=16.0, wait=(1.6+0.9), draw=(7.1+6.3 avg=7.1+7.0=14.0) #right=7.1 #left=6.3
app=25.1, atw=16.6, wait=(1.0+1.3), draw=(7.1+6.7 avg=7.1+7.0=14.0) #right=7.1 #left=6.7
app=25.1, atw=16.4, wait=(1.6+1.9), draw=(6.1+7.5 avg=7.1+7.0=14.0) #right=6.1 #left=7.5

```

超时打印

如上两图，是在屏幕 60fps 刷新率情况下的打印。上图渲染时间充足，而下图的 log 有带#符号字样的超时打印。这个时候就要去看下当前的 CPU GPU DDR 的频率是否过低。

✓ CPU

查看小核频率

```
cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
```

查看大核频率

```
cat /sys/devices/system/cpu/cpu4/cpufreq/cpuinfo_cur_freq
```

✓ GPU

```
cat /sys/devices/platform/ff9a0000.gpu/devfreq/ff9a0000.gpu/cur_freq
```

✓ DDR

开机串口打印的最前有显示 ddr 频率。

CPU 频率低于 408M，GPU 频率低于 300M，DDR 频率低于 666M，都是可能造成 VR 应用性能不足，渲染超时的原因，视频播放等场景的 CPU 和 GPU 频率要求会更高，屏幕刷新率高于 60fps 也需要更高的频率支持。所以要确认下编译 kernel 时 dts 里面这三个频点的最低配置是否有问题，建议参考下述两个 dts 进行配置：

arch/arm64/boot/dts/rockchip/rk3399-box-rev1-disvr.dts

arch/arm64/boot/dts/rockchip/rk3399-disvr-android.dts

在上述两个 dts 中，有如下代码，如果下述代码的最低频率仍然无法满足要求，则可以自行添加，将低频 disabled 掉：

```

/*
 * if the screen of vr helmet has a high screen resolution or
 * high refresh rate, please increase the lowest gpu(gpu_opp_table)
 * and cpu(cluster1_opp) frequency.
 */
&gpu_opp_table {
    opp@200000000 {
        status = "disabled";
    };
}

```

```
        opp@297000000 {  
            status = "disabled";  
        };  
};  
  
&cluster1_opp {  
    opp@408000000 {  
        status = "disabled";  
    };  
    opp@600000000 {  
        status = "disabled";  
    };  
    opp@816000000 {  
        status = "disabled";  
    };  
};
```