RK3399 VR Android 参数配置和调试说明

V1.0_20170106

文件状态:	当前版本:	V1. 0
[]正在修改	作 者:	邓敬威/张文平
[√] 正式发布	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

版本	修改人	修改日期	修改说明	备注
V1.0	邓敬威/张文平	2017.01.06	初版	

目 录

1	宏配置	4
	系统属性说明	
	2.1 主屏方向	
	2.3 头盔 LCD 的刷新方向	
	2.5 视频是否支持 ATW	
	2.6 双屏 LCD 扫描方向	
	2.7 VOP 取反	
	2.8 VR 光学参数(FOV)调节	
2	VR 系统调试	
J	VIV 2N >/L 例 以	/

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	True:支持 AFBC 功能
	提升显示效率,降低系统负载。	False:不支持
	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	

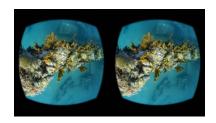
2 系统属性说明

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(LCDCO/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。



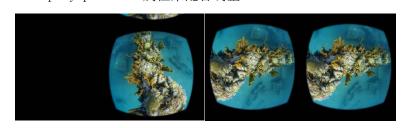
对于分体机,这里还需要单独说明下,分体机分为主板 (rk3399 硬件板) + 头盔板,目前我们默认的产品形态为,主板不带 lcd 屏,通过 typec 线或者 hdmi 线连接头盔板,所以我们有上述主显和次显的默认定义,如果需要主板上带屏,则需要另外配置,我们系统默认不支持这种形态。

2.1 主屏方向

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

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。

用于支持**次屏**的旋转,true 表示次屏可以随着系统 gsensor 或者 app 触发旋转,false 则不旋转。在主屏和次屏都是横屏的情况下,这个属性不需要配置成 true,其他情况下则不定,具体看需求,举两个列子:

- 1) 平板接 HDMI, HDMI 为次屏, true 表示 HDMI 显示可以随着系统的 gsensor 旋转, false 则固定方向显示。
- 2) 平板接 VR 头盔,头盔为次屏,如果头盔为竖屏,则需要把该属性设置为 true。这两个属性的作用和第一节主屏的两个属性类似。先调节 ro. orientation. einit 属性,当这个属性无法调节到正确的显示效果的时候,就需要 ro. rotation. external 属性值 true和 false 切换来配合调整。

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(LCDCO/VopB)。

属性值默认为 0, 表示从 fb0 (默认对应的是 1cdc0, 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 1cdc1, 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

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。

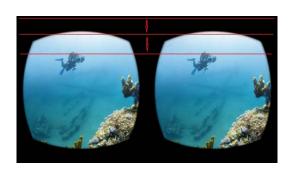


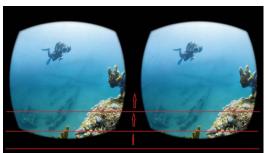
由于双屏机器在硬件设计上,屏幕的物理方向有上下相反的两种接法,对应的屏幕扫描方式也有从下往上扫和从上往下扫两种方式。软件上通过 sys. vr. scan 配置,有 0 和 1 两种。

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
D VRJni
           : BindThreadsToCpu45 : Current platform = rk3399
           : BindThreadsToCpu45 -> bind process sched_setaffinity success!
: BindThreadsToCpu45 -> bind thread 12788 sched_setaffinity success!
D VRJni
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!
 VRJni
D VRJni
            : BindThreadsToCpu45 -> bind thread 12797 sched_setaffinity success!
D VRJni
             BindThreadsToCpu45 -> bind thread 12798 sched_setaffinity success
           : BindThreadsToCpu45 -> bind thread 12799 sched_setaffinity success!
: BindThreadsToCpu45 -> bind thread 12800 sched_setaffinity success!
D VRJni
D VRJni
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!
                                                                                                 3
  VRJni
             current platform screen flush rate = 60
              --> successfully open /sys/class/graphics/fb0/vsync
D VRJni
```

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

```
atw=16.5, wait=(2.2+4.4), draw=(4.0+4.2 avg=4.0+4.0=8.0)

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,

app=16.4,

atw=16.3, 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)

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)
```

```
ath=17.5, wait=(1.712.3), draw=(7.3+6.4 avg=7.1+7.0=14.0) #right=7.3 #left=6.4 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=24.5, 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 #right=7.7 #left=6.2 #right=7.7 #left=6.2 #right=7.7 #left=6.2 #right=7.1 #left=6.8 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 #right=7.1 #left=6.3 atw=16.6, wait=(1.0+1.3), draw=(7.1+6.7 avg=7.1+7.0=14.0) #right=7.1 #left=6.3 #right=7.1 #left=6.7 atw=16.4, wait=(1.0+1.3), draw=(7.1+6.7 avg=7.1+7.0=14.0) #right=7.1 #left=6.7 #right=6.1 #left=7.5 app=25.1,
```

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) frequence.
 */
&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";
       };
};
```