



# Linux 高刷屏 开发指南

版本号: 1.0

发布日期: 2024.11.28

## 版本历史

版本号	日期	制/修订人	内容描述
1.0	2024.11.28	AWA2081	添加初版



# 目 录

<b>1 概述</b>	<b>1</b>
1.1 编写目的	1
1.2 相关人员	1
1.3 适用范围	1
<b>2 模块描述</b>	<b>2</b>
2.1 高刷	2
2.2 VRR	2
<b>3 屏幕筛选</b>	<b>3</b>
<b>4 VRR 切换方案</b>	<b>4</b>
4.1 VFP 切换方案	4
4.2 VBP 切换方案	5
4.2.1 VBP 切换周期	5
<b>5 软件架构</b>	<b>6</b>
5.1 软件架构	6
5.2 模块说明	6
<b>6 配置说明</b>	<b>8</b>
6.1 像素时钟频点计算	8
6.2 Kernel 配置说明	8
6.2.1 VFP 切换方案配置	8
6.2.2 VBP 切换方案配置	9
6.3 上层策略配置说明	10
6.3.1 高刷基本属性配置说明	10
6.3.2 原生 Auto 策略相关属性配置说明	10
6.3.3 白名单策略配置说明	11
6.3.4 高刷策略说明	12
<b>7 常见问题分析</b>	<b>13</b>
7.1 显示异常	13
7.2 DSC 2k 高刷闪屏	13
7.3 刷新率切换问题	13
<b>8 参考文献</b>	<b>14</b>

# 1 概述

## 1.1 编写目的

本文档主要介绍了全志 Android 15 平台高刷功能方案、配置说明以及调试方法等，本文档仅适用于单 DSI 屏项目。

## 1.2 相关人员

适用于需要基于全志平台实现 VRR 相关功能的工程师。

## 1.3 适用范围

表 1-1: 适用产品列表

产品名称	内核版本
A733	Linux-6.6

## 2 模块描述

### 2.1 高刷

屏幕刷新率指屏幕刷新或更新的频率，即每秒屏幕画面被刷新的次数。

大多数设备的屏幕典型刷新率是 60Hz，表示屏幕每秒刷新 60 次，即屏幕上的图像每 16.67ms 更新一次。一个画面或图像占据屏幕的这段时间称为刷新时间，刷新时间与刷新率成反比。

屏幕展示的是静态图片，前后两张图片的时间间隔很短且图片差别跟小，一个动作通过多张图片展示，利用人眼的视觉停留效应，可以呈现动态效果。

通常，90Hz 刷新率称为高刷，在显示同样的画面时，90Hz 刷新率的帧数是 60Hz 刷新率帧数的 1.5 倍，帧数越多，画面更流畅。即刷新率越高，在单位时间内展现的图片越多，画面衔接更流畅更细腻，用户体验更好。

### 2.2 VRR

高刷新率提高画面流畅度的同时，也会带来额外的功耗开销。屏幕刷新的频率越高，会导致 CPU、GPU、DDR 等模块以及屏幕本身的功耗增加，从而导致设备的耗电量增加，对设备续航能力挑战较大。

针对高刷新率带来的功耗问题，引入 VRR（Variable Refresh Rate，可变刷新率）技术。

设置中有多档刷新率供用户选择和切换，或设置为自动刷新率。在自动刷新率模式下，系统根据应用、亮度、电量等因素调整屏幕刷新率。例如，在支持 90Hz 刷新率的屏幕上，系统会自动在 60Hz/90Hz 之间进行切换。此方式可以在保证用户体验的同时兼顾功耗问题。

VRR 可以节省功耗是相对于固定高刷而言的。VRR 可使刷新率在 60Hz/90Hz 之间动态切换，意味着设备的刷新率一直大于或等于 60Hz，故其功耗一定大于固定 60Hz 下的功耗，同理，其功耗也一定小于固定 90Hz 下的功耗。

### 3 屏幕筛选

- 1、屏幕的每 lane 速率  $\text{pixel\_clk} * \text{bpp} / \text{lane} > 1.5 \text{ Gbps}$  时，该屏幕必须带有 DSC，且需要询问屏厂是否支持  $\text{vbp} > 1400$ 、 $\text{vsync} = 30$ ；
- 2、dsc 屏的 vbp 支持动态调整，dsc 屏的分辨率：x、y  $\leq 2000$ 。
- 3、屏幕是单 dsi 屏。

#### ⚠ 注意

在使用 dsc 屏做高刷时，由于 vbp 切换不是主流的切换方法，且不是很成熟，总会遇到一些奇奇怪怪的屏端问题，需要找屏厂大力支持，建议选择技术支持给力的屏厂屏。

## 4 VRR 切换方案

高刷功能需要依赖屏幕能支持目标刷新率（例如 60Hz 和 90Hz）。芯片根据帧同步信号 VSYNC 定时发送数据到屏幕，屏幕获取图像数据并更新显示内容。

LCD 显示帧率 FPS 的计算公式如下：

$$\text{FPS} = \text{pixel\_clk} / [(\text{HFP} + \text{HBP} + \text{HSYNC} + \text{HACTIVE}) * (\text{VFP} + \text{VBP} + \text{VSYNC} + \text{VACTIVE})]$$

### 4.1 VFP 切换方案

使用该方案需要满足两个条件：

- dphy 每 lane 速率  $\text{pixel\_clk} * \text{bpp} / \text{lane} < 1.5 \text{ Gbps}$ 。
- 不使用 DSC。

当外接屏幕满足以上条件时，可采用 VFP（Vertical Front Porch，垂直前肩）切换方案：保持 pixel\_clk 不变，通过修改显示的 VFP 值实现帧率改变。在屏幕支持的情况下，通过配置不同的 VFP 以支持不同的刷新率。

VFP 方案只需软件在一个 VSYNC 周期的 VACTIVE 的区间内，将 VFP 参数配置到显示模块寄存器中，芯片显示硬件模块会在当前 VSYNC 信号内自动切换到新的刷新率，同时 LCD 模组根据新接收到的 VFP 参数自动切换到新的刷新率。如下图所示：

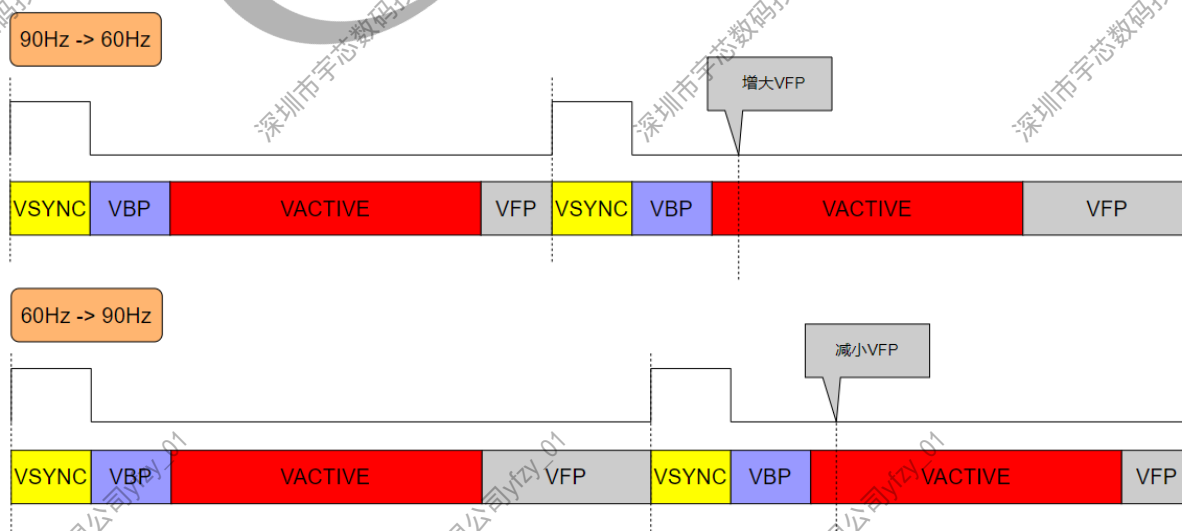


图 4-1: VFP 动态调帧示意图

## 4.2 VBP 切换方案

使用该方案需要满足两个条件：

- LCD 屏幕 vbp > 1400、vsync = 30 显示不会异常。
- 使用 dsc。

当外接屏幕满足以上条件时，可采用 VBP（Vertical Back Porch，垂直后肩）切换方案：保持 pixel\_clk 不变，通过修改显示的 VBP 值实现帧率改变。在屏幕支持的情况下，通过配置不同的 VBP 以支持不同的刷新率。

VBP 方案需要软件在一个或多个 VSYNC 周期的 VSYNC 的区间内，将 VBP 参数配置到显示模块寄存器中，芯片显示硬件模块会在当前 VSYNC 信号内自动切换到新的刷新率，同时 LCD 模组根据新接收到的 VBP 参数自动切换到新的刷新率。如下图所示：

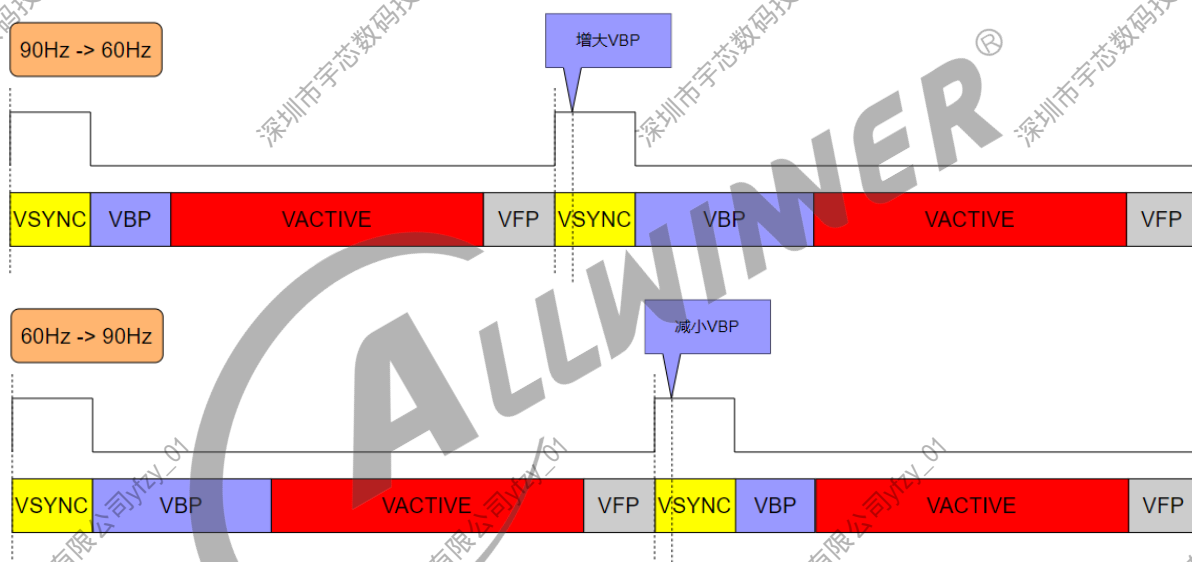


图 4-2: VBP 动态调帧示意图

### 4.2.1 VBP 切换周期

如何判断是一个 VSYNC 周期还是多个 VSYNC 周期切换 VBP：

- **多个 VSYNC**：有些屏幕的 vbp 不能一次性变化过大，否则切帧时会出现花屏，所以需要在多个 VSYNC 周期内按照屏幕 vbp 变化容忍值递增/递减变化，dts 中 **dsc,vrr-setp** 等于 vbp 变化容忍值。
- **一个 VSYNC**：若屏幕的 vbp 变化容忍值可以满足 90Hz -> 60Hz vbp 所需要增加的值，则驱动会在一个 VSYNC 周期内完成切帧，dts 中 **dsc,vrr-setp** 等于 90Hz -> 60Hz vbp 所需要增加的值。

dsc,vrr-setp 具体可以设置成多少可以向屏厂咨询，或自行 try;

## 5 软件架构

### 5.1 软件架构

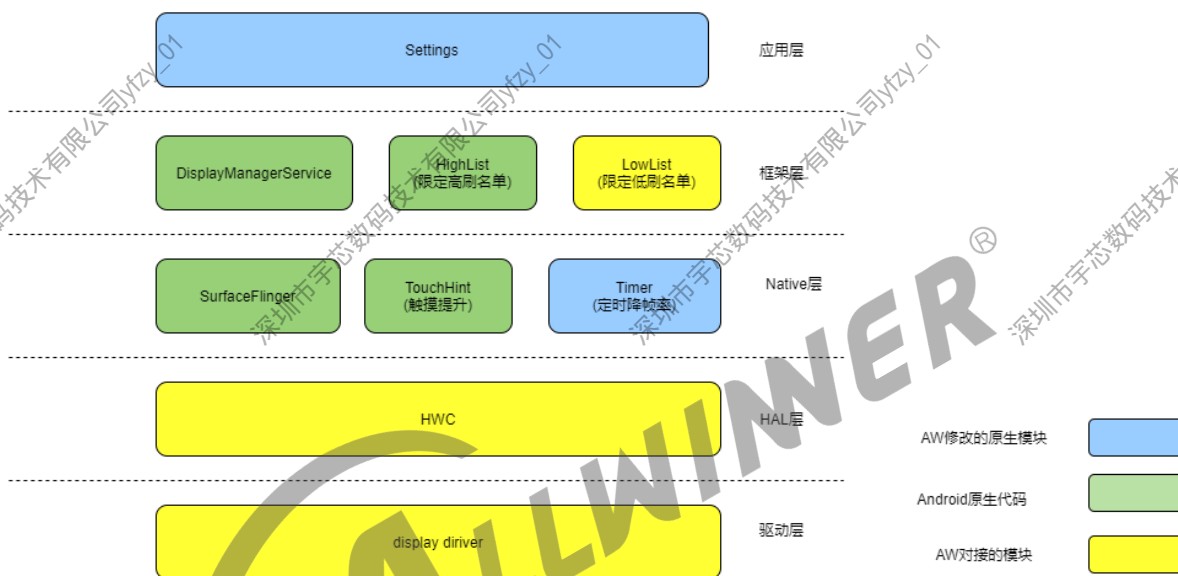


图 5-1

### 5.2 模块说明

高刷 VRR 的调用软件层主要包含 5 个部分，各部分主要功能说明如下：

- **应用层：**向用户展示支持帧率设置界面，并接受用户的切换请求，通过写 Settings 数据库通知到 Framework 层。
- **Framework 层：**对应用层上报支持的 Mode，并监听应用层切换帧率行为，进行帧率切换决策。决策条件包括原生 DMS（Display Manager Service，显示管理器服务）策略以及新增的白名单和 3s 降帧策略，最终调用 Native 接口将 DMS 的 VRR 策略设置到 SurfaceFlinger。
- **Native 层：**SurfaceFlinger 对 Framework 上报支持的 Mode，保存 DMS 设置的 VRR 策略，基于当前 DMS 设置的 VRR 策略并结合内部的 Auto 策略计算得到最佳的刷新率，然后调用 HAL 层接口下发切换到刷新率对应的 Display Mode。
- **HAL 层**

- 接收 Kernel 上报的 Display Timing，根据实际的 Timing 配置将实际支持的 Display Mode 上报给 SurfaceFlinger。
- 接收 SurfaceFlinger 下发的 Mode ID，调用 Lib 库将 Display Mode 下发到 Kernel。
- Kernel 层
- DTS 配置 Display Timing，上报给 HWC。
- 接收 HWC 下发的 Display Mode，配置相关的硬件寄存器实现帧率切换。



## 6 配置说明

### 6.1 像素时钟频点计算

芯片支持的像素时钟频点计算公式如下：

- 像素时钟： $\text{pixel\_clk} = \text{vt} * \text{ht} * \text{fps}$
- dphy 每 lane 速率： $\text{hs\_clk} = \text{pixel\_clk} * \text{bpp} / \text{lane}$
- $\text{hs\_pclk} * m > \text{vco} \Rightarrow m = \text{vco} / \text{hs\_pclk}$  (向上取整)
- $n = \text{pclk} * m / \text{dcxo}$  (四舍五入)
- 芯片支持的像素时钟频点： $\text{dcxo} * n / (m * \text{bpp} / \text{lane})$

vco 是 pll 稳定工作频点；dcxo 是晶振频点；bpp 是输出给屏端的位深，带 dsc 的是压缩之后的位深。

在 A733 平台， $\text{vco} = 1260\text{MHz}$ ， $\text{dcxo} = 26\text{MHz}$ 。

### 6.2 Kernel 配置说明

对于 Kernel 的配置，主要是计算 60Hz 和 90Hz 的 timing。涉及两种切换方案：VFP 切换和 VBP 切换。

#### 6.2.1 VFP 切换方案配置

若要实现 60Hz/90Hz 刷新率，需配置相应的 timing。以全志平台搭载的 JT109WXL001 屏幕为例，该屏幕默认的初始刷新率为 60Hz，通过动态升帧实现 90Hz，从而实现 VRR。方法如下：

**步骤 1** 按照屏厂给的初始化代码点亮刷新率 90Hz，且保证 TP 也能正常使用。

**步骤 2** 调整 Porch 值使 pixel\_clk 值接近芯片支持的频点。以 A733 为例，芯片支持的档位之一为 119MHz，因此需要调整 Porch 值使 pixel\_clk 值接近 119MHz。

**步骤 3** 根据实际档位对 VFP 值进行微调。FPS 计算公式： $\text{FPS} = \text{pixel\_clk} / [(\text{HFP} + \text{HBP} + \text{HSYNC} + \text{HACTIVE}) * (\text{VFP} + \text{VBP} + \text{VSYNC} + \text{VACTIVE})]$

**步骤 4** 以 90Hz 的 timing 为基础，以 VFP 为唯一变量，增大 VFP，代入 FPS 计算公式，可得到 60Hz 的 timing 配置如下

```
JT109WXL001_timing0: timing0 { /* 60Hz */
    clock-frequency = <119180880>;
    hactive = <800>;
    vactive = <1280>;
    hback-porch = <40>;
    hfront-porch = <40>;
    hsync-len = <4>;
    vback-porch = <40>;
    vfront-porch = <923>;
    vsync-len = <4>;
};
JT109WXL001_timing1: timing1 { /* 90Hz */
    clock-frequency = <119180880>;
    hactive = <800>;
    vactive = <1280>;
    hback-porch = <40>;
    hfront-porch = <40>;
    hsync-len = <4>;
    vback-porch = <40>;
    vfront-porch = <174>;
    vsync-len = <4>;
};
```

**步骤 5** 设置系统默认按照 60Hz 启动：native-mode = <&JT109WXL001\_timing0>;

底层配置好后，驱动加载完成，底层会上报已配置的几个不同刷新率的 timing，上层集成后，会在设置界面提供可切换的屏幕刷新率选项。

## 6.2.2 VBP 切换方案配置

若要实现 60Hz/90Hz 刷新率，需配置相应的 timing。以全志平台搭载的 C00ZJ120H216001 屏幕为例，该屏幕默认的初始刷新率为 60Hz，通过动态升帧实现 90Hz，从而实现 VRR。方法如下：

**步骤 1** 按照屏厂给的初始化代码点亮刷新率 90Hz，且保证 TP 也能正常使用。

**步骤 2** 调整 Porch 值使 pixel\_clk 值接近芯片支持的频点。以 A733 为例，芯片支持的档位之一为 351MHz，因此需要调整 Porch 值使 pixel\_clk 值接近 351MHz。

**步骤 3** 根据实际档位对 VBP 值进行微调。FPS 计算公式： $FPS = pixel\_clk / [(HFP+HBP+HSYNC+HACTIVE) \times (VFP+VBP+VSYNC+VACTIVE)]$

**步骤 4** 以 90Hz 的 timing 为基础，以 VBP 为唯一变量，增大 VBP，代入 FPS 计算公式，可得到 60Hz 的 timing 配置如下

```
display-timings {
    native-mode = <&C00ZJ120H216001_timing0>;
    C00ZJ120H216001_timing0: timing0 { /* 60Hz */
        clock-frequency = <351000000>;
        hback-porch = <120>;
        hactive = <1200>;
```

```
hfront-porch = <180>;
hsync-len = <72>;
vback-porch = <1440>;
vactive = <2000>;
vfront-porch = <250>;
vsync-len = <30>;
};

C00ZJ120H216001_timing1: timing1 { /* 90Hz */
clock-frequency = <351000000>;
hback-porch = <120>;
hactive = <1200>;
hfront-porch = <180>;
hsync-len = <72>;
vback-porch = <200>;
vactive = <2000>;
vfront-porch = <250>;
vsync-len = <30>;
};
};
```

**步骤 5** 设置的vbp 变化容忍值。

**步骤 6** 设置系统默认按照 60Hz 启动：native-mode = <&C00ZJ120H216001\_timing0>;

底层配置好后，驱动加载完成，底层会上报已配置的几个不同刷新率的 timing，上层集成后，会在设置界面提供可切换的屏幕刷新率选项。

## 6.3 上层策略配置说明

### 6.3.1 高刷基本属性配置说明

配置默认帧率和峰值帧率

device/softwinner/jupiter//overlay/overlay/frameworks/base/core/res/res/values/config.xml

```
<integer name="config_defaultRefreshRate">0</integer>
<integer name="config_defaultPeakRefreshRate">90</integer>
```

如上所示，config\_defaultRefreshRate 配置成 0，使用默认帧率，config\_defaultPeakRefreshRate 配置成 90，为最高帧率

### 6.3.2 原生 Auto 策略相关属性配置说明

参考 Google 的文档：<https://source.android.com/docs/core/graphics/multiple-refresh-rate?hl=zh-cn>

Android 中使用两个 SettingsProvider 的值来表征当前的帧率，一个是 Set-

tings.System.PEAK\_REFRESH\_RATE，另外一个 Settings.System.MIN\_REFRESH\_RATE，这两个值开机第一次使用上面的 config\_defaultRefreshRate 和 config\_defaultPeakRefreshRate 进行初始化，在设置中可以修改这两个值



图 6-1

**60Hz 模式：**MIN\_REFRESH\_RATE 和 PEAK\_REFRESH\_RATE 都会被设置为 60，系统的最高帧率为 60Hz

**90Hz 模式：**MIN\_REFRESH\_RATE 设置为 60，PEAK\_REFRESH\_RATE 被设置为 90，系统按照 Android 策略决定是否以 90Hz 帧率运行

**Auto 模式：**MIN\_REFRESH\_RATE 设置为 0，PEAK\_REFRESH\_RATE 被设置为 90，帧率在此范围内调整

### 6.3.3 白名单策略配置说明

对于想将其限制为使用 60Hz 的 app，可以将包名加入下面的名单中，加入的应用启动后，帧率固定为 60Hz，不会受点击事件提升

```
device/softwinner/jupiter/<方案目录>/overlay/overlay/frameworks/base/core/res/res/values/config.xml
```

```
<string-array name="config_highRefreshRateBlacklist">
  <item>com.softwinner.camera2</item>
  <item>com.google.android.apps.photos</item>
</string-array>
```

对于想限制一直使用 90Hz 的 app，可以将报名加入下面的名单，加入的应用启动后，帧率固定为 90Hz，不会下降

```
<!-- this package will use 90Hz. -->
<string-array name="config_lowRefreshRateBlacklist">
  <item>com.mxplayer.mxplayer</item>
  <item>com.softwinner.videoplayer</item>
</string-array>
```

### 6.3.4 高刷策略说明

Android 的高刷策略主要是以下因素影响：

- **默认刷新率设置：**默认刷新率值在 `R.integer.config_defaultRefreshRate` 配置叠加层中设置。该值用于确定动画和轻触互动的标准设备刷新率。
- **峰值刷新率设置：**峰值刷新率值是从 `Settings.System.PEAK_REFRESH_RATE` 读取的。该值在运行时更改，以反映当前设备设置（例如，从菜单选项更改）。默认值在 `R.integer.config_defaultPeakRefreshRate` 配置叠加层中设置。
- **最小刷新频率设置：**最小刷新频率值是从 `Settings.System.MIN_REFRESH_RATE` 读取的。该值可以在运行时更改，以反映当前设备设置（例如，从菜单选项更改）。默认值为 0，因此不存在默认的最小值。
- **应用要求的 ModelId：**应用可以设置 `WindowManager.LayoutParams.preferredDisplayModelId` 以反映屏幕运行的首选配置。在大多数情况下，`DisplayManager` 会相应地设置默认配置 ID，并设置最小和最大刷新频率以匹配配置的刷新频率。
- **省电模式：**当设备处于低功耗模式（通过 `Settings.Global.LOW_POWER_MODE` 指示）时，刷新率上限为 60Hz。
- **触摸操作：**Auto 和 90Hz 模式下，触摸屏幕或旋转，会短时提升帧率到 90Hz，三秒钟后恢复
- **禁止高刷名单：**处于该名单的 app，触摸和旋转事件不会提升帧率
- **禁止低刷名单：**处于该名单的 app，90Hz 和 Auto 模式下，除省电模式外不会降低到 60Hz

## 7 常见问题分析

配置高刷功能前，需先确保设备的基本显示功能正常，排除硬件问题以及软件流程问题。添加高刷配置后，若仍遇到显示问题，可参照以下方法进行检查。

### 7.1 显示异常

- 检查 CLK 配置是否正确，CLK 配置使用以下命令查看：
- pixel\_clk: `cat /sys/kernel/debug/clk/clk_summary | grep disp*_dsi*_ls`
- 检查屏幕初始化代码和刷新率是否匹配，具体细节请联系屏幕厂商确认。
- 检查开发板的线序是否有问题，与屏端是否能一一对应。

### 7.2 DSC 2k 高刷闪屏

检查 DDR 频率是否过低：要求最低频率 800MHz。

```
cat /sys/class/devfreq/xxxx.dmcfreq/min_freq
```

### 7.3 刷新率切换问题

使用 VBP 方案时：

- 帧率切换时间长：检查 VSYNC 是否  $\geq 30$ 。
- 切换闪屏：检查设置的 vbp 变化容忍值是否过大，或不合适。

## 8 参考文档

《AW\_DRM 屏幕适配\_ 调试指南》






## 著作权声明

版权所有 ©2024 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

## 商标声明

、、**全志科技**、（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

## 免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。