

OLED屏幕因其像素自发光特性，相比LCD屏色彩更鲜艳、低温时也不会产生拖影，拥有更好的观感，从而逐渐在中高端市场中流行并取代LCD的位置。

不过，OLED屏幕的流行也带来了新的问题。为了保证色彩准确度，目前使用OLED屏幕的主流手机亮度降低到一定程度之后，会切换为低频PWM调光，各家厂商的阈值不一，用另一台手机打开相机对着屏幕即可查看到一道道黑色条纹的交替闪烁。

相对于高亮度下的DC调光，以及手机中罕见的高频PWM调光，一些对于闪烁比较敏感的用户面对低频PWM调光的屏幕可能会感到不适。因此，各家厂商纷纷推出让敏感用户也可能舒适使用OLED屏幕手机的全程DC调光方案，具体方法有所不同，低亮度下的屏幕观感也有所区别。

Redmi K20 Pro搭载的纯硬件DC调光，是市面上所有DC调光中唯一不损失显示数据，观感最佳的方案。如果要详细了解这种方案的优点，有必要先介绍DC和PWM两种调光方法本身，以及市面上DC调光方案应用的手段区别。

## DC与PWM：各自优劣均十分明显

目前主流的显示设备，包括手机、笔记本电脑、显示器在内，无论是传统LCD面板还是OLED面板，调整显示内容亮度的方式都可以归为DC和PWM两种。

其中，DC调光的手段比较传统，即通过调整发光元件电压或电流，也就是改变功率的方式来控制显示亮度。主流LCD屏幕由于发光来自于面板背后的背光LED灯，工艺已经非常成熟，发光均匀性不难控制，因此全程DC调光并不影响LCD屏幕在各个亮度下的表现，显示观感更多取决于背光LED灯之前的液晶面板本身。

但DC调光遇到OLED面板则产生了问题。由于主流OLED面板的原理是通过几百万个RGB子像素自发光形成图像，在现有的工艺条件下，几乎不可能保证这几百万个像素在所有亮度下的表现完全一致，从而导致色彩准确度降低，这种情况在亮度低的时候更加明显。因此，为了保证OLED面板在一定亮度以下的观感，才引入了PWM调光。

相对而言，PWM调光从原理上看更先进一些。它不会改变功率，而是通过控制发光元件在一定的频率上交替闪烁，利用人眼的视觉残留效应达到连续显示的效果，需要低亮度则交替闪烁中熄屏时间更久，高亮度则亮屏

时间更长。如果用另一台手机打开相机对PWM调光屏幕录像，可以很明显看到一道道黑色条纹的交替闪烁。由于PWM调光并不改变亮屏时的功率，因此可以有效提升低亮度下屏幕的显示参数，如色彩及灰阶准确度、色域覆盖等。

PWM调光也有自己的劣势。有研究发现，长期观看低频交替闪烁可能会给敏感人群造成不适，尤其是频率在1000Hz以下的闪烁。目前手机的PWM调光多为240Hz，只有部分笔记本电脑和显示器中可见数千甚至数万Hz的PWM频率。没有闪烁的DC调光屏幕则不存在这个问题。

### DC调光（无闪烁条纹）和PWM调光（有闪烁条纹）的对比效果

因此，大多数厂商的OLED屏幕手机采用的是DC和PWM混合调光的方式。在中高亮度下采用DC调光防止闪烁，低亮度下采用PWM调光以获得更好的观感。但随着人们每天使用手机的时间越来越长，低频PWM给敏感人群造成的不适也时有报告，因此不少厂商都探索并推出了全程DC调光方案，在防止低亮度屏幕闪烁的同时，通过不同的手段来抑制屏幕显示参数下降。

### 目前OLED屏幕手机普遍采用的调光方式

#### 市面类DC调光方案回顾 色彩数据均有丢失

在各厂商推出全程DC调光模式之前，就有一些防闪烁类的App提供类似的功能，原理是强制屏幕保持在最高亮度，显示图像的亮度则通过GPU后期渲染来调整。缺点是屏幕亮度维持在最高和GPU后期渲染都需要消耗很多电量，同时由于屏幕本身亮度维持最高，系统内置的自动亮度功能也无法起作用，显示的实时渲染图像也非常粗糙。

包括小米首发的纯硬件DC调光方案在内，目前市面上全程DC调光实现的方法均是让屏幕亮度降低到DC和PWM调光切换节点后维持不变，通过不同的手段调整显示图像，使其符合设置的亮度。

### 实现全程DC调光的方法

要深入了解这些方法的不同点，有必要简单介绍Android手机显示图像的过程及简单的系统架构。

以上是Google官方Android开发者社区提供的平台架构图，我们只需要知道其中的四个模块：从最接近用户的顶层往下查看，蓝色的是App即应用程序，绿色的是Android Framework框架，然后跳过紫色和黄色是青色的HAL硬件抽象层，最后是最底层的Linux Kernel即核心。一个App显示图像的过程如下图所示：

- ① Android应用把要显示的图形分解成一个个图层，然后进行各个图层的测量和绘制，将测量出的共24位红（R）、绿（G）、蓝（B），以及透明度（Alpha）四个分量信号传输给Framework框架。
- ② Framework框架接收App传输的RGB和Alpha信号，并进行各个图层的渲染。
- ③ 待框架层渲染完毕后，再由HAL硬件抽象层根据RGB信号和Alpha透明度，叠加图层合成完整的图像，然后将合成后图像的RGB数值转交给Kernel核心。
- ④ 由Kernel核心调用SoC中的显示处理模块进行色域映射、色彩增强等优化，处理后的数据传输给屏幕面板驱动芯片。
- ⑤ 由屏幕的驱动芯片进行处理开始在屏幕上绘制。

除小米首发的纯硬件方案外，其他DC调光方案分别作用于显示图像过程的不同阶段，相比防闪烁类的App有着巨大进步，不仅兼容自动亮度，图像质量也有大幅提升。不过它们的共同点都是最后传输给OLED屏幕驱动芯片的数据相比App绘制的数据更少，压缩了灰阶值。

## 1、蒙版+dither抖动，作用于显示流程第②部分

这种方法相当于在人眼与屏幕之间叠加了一个半透明的蒙版，Framework在渲染图

层时会根据蒙版修改Alpha通道的透明度以压缩灰阶值。随后从Framework传输至HAL层进行图像合成的色彩位数已经小于24位，低于原有色彩数据。然后在Kernel进行色域映射、色彩增强优化后加入dither抖动算法，对小于24位的色彩进行补偿。

## 2、 PCC+dither抖动，作用于显示流程第④部分

小米9开发版最早加入的DC调光方案就是采用PCC+dither的方式。在Kernel进行色域映射、色彩增强后调用高通PCC ( Picture Color Calibration，图像色彩校正) 硬件模块，压缩输出给屏幕数据的色彩位数，计算应该显示的亮度，随后同样通过dither抖动算法进行色彩补偿。

以上两种算法在低亮度下计算的色彩准确度比较依赖于DC/PWM切换临界点时屏幕的Gamma值。如果屏幕原有的Gamma未经过矫正，色彩不准确的问题经过DC调光算法方大后会更加明显。

## 3、 Pixelworks第三方显示芯片+dither抖动，同样作用于显示流程第④部分

黑鲨2采用的这种方案流程和PCC+dither抖动类似，不同点在于灰阶压缩由第三方Pixelworks显示芯片和高通PCC硬件模块共同实现。相比单纯的高通PCC模块处理，第三方Pixelworks芯片并不依赖屏幕在临界点时的Gamma值，因此色彩准确度更佳，但引入第三方芯片也带来了额外的成本和功耗。

三种方案都采用了dither抖动算法，它可以改善由于色彩位数减少导致的图像过渡不均匀问题，但模拟的色彩深度会导致分辨率上的损失，屏幕上容易出现一个个"小麻点"或者"小方块"，毕竟经过色彩有损压缩再模拟放大，计算得到的数据和原始色彩数据还是存在不小差异。未经色彩压缩、色彩压缩后和经dither处理后的对比效果示例如下，从左到右分别为色彩未压缩、色彩压缩后、色彩压缩+dither抖动补偿的效果：

## 灰阶无损 首发纯硬件DC调光方案

相比前面介绍的三种数据有损方案，Redmi K20 Pro的纯硬件DC调光传输给OLED面板驱动芯片的数据并未减少，灰阶、色彩位数都保持原样。同样的，这种纯硬件DC调光在屏幕亮度降低至DC/PWM切换临界点后使其维持不变，通过集成在面板驱动芯片中的自研算法，来调整显示图像让它符合设置的亮度。

由于传输给驱动芯片的数据没有减少，纯硬件DC调光方案并未引入dither抖动来补偿色彩位数，也避免了dither抖动带来的"小方块"现象。通过一系列实拍对比，可以看出采用纯硬件DC调光的Redmi K20 Pro色彩过渡效果更好，观感更佳，罕见普通DC调光出现的过渡不均匀、分辨率降低等现象。