

密级状态： 绝密( ) 秘密( ) 内部资料( ) 公开(  )

## 显示效果调节 APK 使用指南

(第二系统产品部)

<b>文件状态:</b> <input type="checkbox"/> 草稿 <input type="checkbox"/> 正在修改 <input checked="" type="checkbox"/> 正式发布	<b>文件标识:</b>	RK-SM-YF-293
	<b>当前版本:</b>	1.0.0
	<b>作 者:</b>	林垚
	<b>完成日期:</b>	2021-06-30
	<b>审 核:</b>	黄祖芳
	<b>审核日期:</b>	2021-06-30



---

## 免责声明

本文档按“现状”提供，瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

## 商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自所有者所有。

## 版权所有 © 2020 瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园 A 区 18 号

网址：[www.rock-chips.com](http://www.rock-chips.com)

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：[fae@rock-chips.com](mailto:fae@rock-chips.com)

# 前言

## 概述

本文档将介绍显示效果调节 APK 使用说明。

## 修订记录

日期	版本	作者	修改说明
2021/06/30	V1.0.0	林垚	正式发布

# 目录

<b>1</b>	<b>显示效果调节 APK 使用说明.....</b>	<b>1</b>
1.1	显示效果调节 APK 使用说明.....	1
1.2	PC 上 3D LUT 调整方法.....	2
1.3	传输 3D LUT 配置到文件.....	3
1.4	屏幕显示调节 API.....	4

# 1 显示效果调节 APK 使用说明

## 1.1 显示效果调节 APK 使用说明



图 1 主界面

显示效果调节 APK 源码位于：`packages/apps/DisplayAdjust` 路径下，需要手动编译安装才能使用。如果使用 `install apk` 的方法安装调试，需要把工程中 `system/lib` 和 `system/lib64` 的 `libsava_baseparameter_util.so` push 到板上同名目录下，并在 `system/etc/public.libraries.txt` 里面加上 `libsava_baseparameter_util.so`。也可以在 SDK 工程中手动编译 APK，烧录完整镜像进行调试。若要默认编译显示调节 APK，则需要在产品 `device.mk` 里面增加 `PRODUCT_PACKAGES+=DisplayAdjust`。

如上图所示，主界面主要包括两个部分：左侧的控制按钮和右侧的展示图片。控制按钮从上至下依次为：屏幕选择按钮，右侧展示图片选择按钮，亮度控制按钮，对比度控制按钮，饱和度控制按钮，色调控制按钮，色温控制按钮，重置按钮，3D LUT 设置按钮（3D LUT 设置功能只有 RK356X 平台上支持），导出 `baseparameter` 按钮。

**屏幕选择按钮：**可以选择需要调节屏幕效果的屏幕；

**右侧展示图片选择按钮：**可以根据需求从本地图片中选择右侧用于展示的图片；

**亮度控制按钮：**可以通过“+”、“-”按钮或者 SeekBar 调节亮度，调节范围为 0-100；

**对比度控制按钮：**可以通过“+”、“-”按钮或者 SeekBar 调节对比度，调节范围为 0-100；

**饱和度控制按钮：**可以通过“+”、“-”按钮或者 SeekBar 调节饱和度，调节范围为 0-100；

**色调控制按钮：**可以通过“+”、“-”按钮或者 SeekBar 调节色调，调节范围为 0-100；

**色温控制按钮：**可以通过“+”、“-”按钮或者 SeekBar 调节色温，默认可以调节的值为 3500，5500，6500，7500。色温范围可以根据需求自行修改，修改 `MainActivity` 中的

COLOR\_TEMPERATURE\_VALUE 数组即可。色温的取值范围从 1000 至 25100，必须能被 100 整除。6500 为默认标准值所以 COLOR\_TEMPERATURE\_VALUE 数组中必须包含 6500；

**重置按钮：**重置亮度、对比度、饱和度、色调和色温数值。其中亮度、对比度、饱和度、色调默认值为 50，色温默认值为 6500。亮度、对比度、饱和度、色调和色温的默认值可以通过 MainActivity 中的 DEFAULT\_VALUE 修改，色温的默认值可以通过 MainActivity 中的 DEFAULT\_COLOR\_TEMPERATURE 修改；

**3D LUT 设置按钮：**点击 3D LUT 设置按钮，从配置文件中读取 3D LUT 并设置到屏幕上。配置文件默认位置为：/sdcard/3d.lut，可以通过 MainActivity 中的 LUT\_3D\_CONFIG\_PATH 修改。PC 上如何调节 3D LUT 效果并把配置文件传到 Android，请参考下面两节内容；

**导出 baseparameter 按钮：**将机器上现有的 baseparameter 分区数据导出，用于显示效果固化，保存的默认路径为：/sdcard/baseparameter.img，可以通过 MainActivity 中的 OUTPUT\_IMAGE\_PATH 修改。注意导出之前需要暂时将 Selinux 关闭，才能成功保存 baseparameter.img。baseparameter.img 保存成功后替换工程中的/device/rockchip/common/baseparameter.img，再重新打包 update.img，烧录该 update.img 即可实现指定的屏幕效果。

## 1.2 PC 上 3D LUT 调整方法

PC 上要进行 3D LUT 调节，需要使用 3D LUT Creator 这款软件，3D LUT Creator 的下载和详细使用说明请参考软件官网：<https://www.3dlutcreator.com/>。

3D LUT Creator 使用流程：

- (1) 设置软件:点击左上角: File -> Proferences,参考下图进行设置;

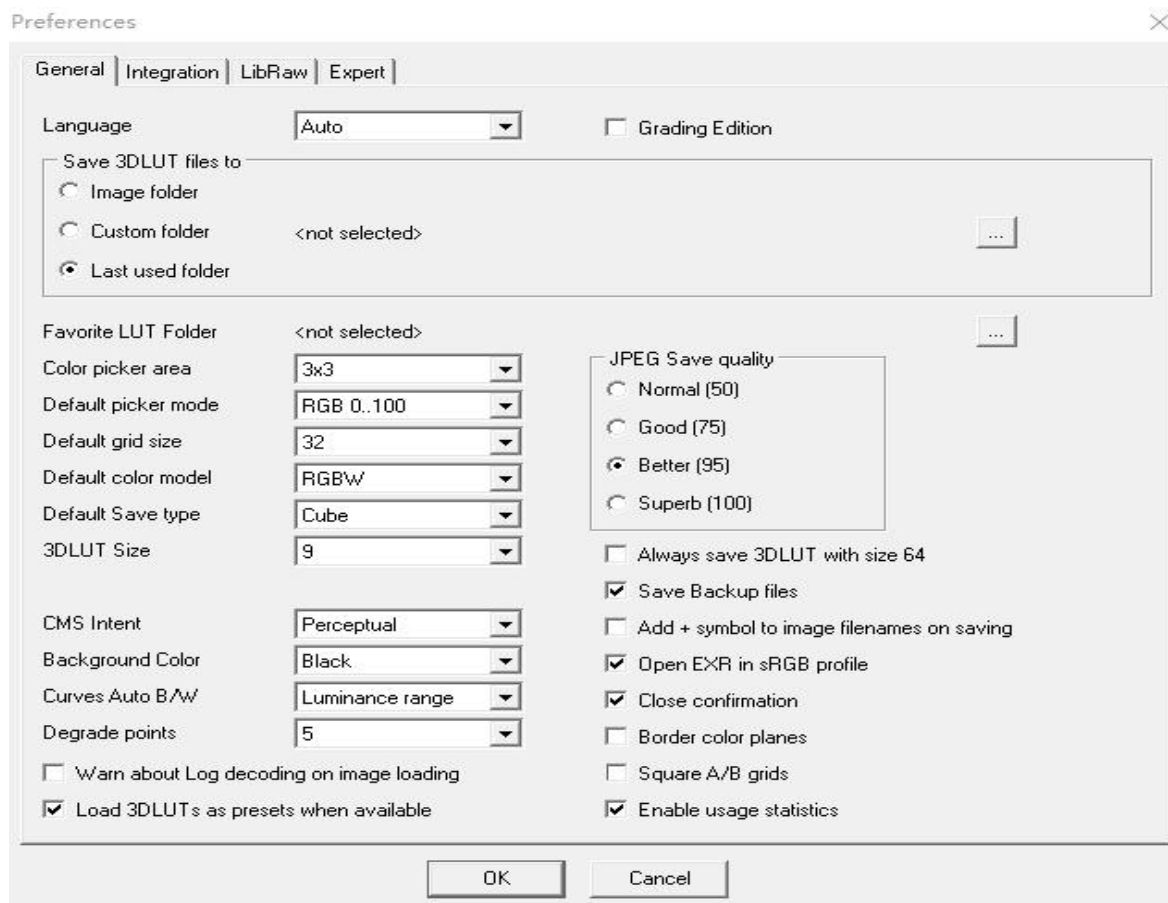


图 2 设置软件



(2) 调节效果，详细调节方法见官网：

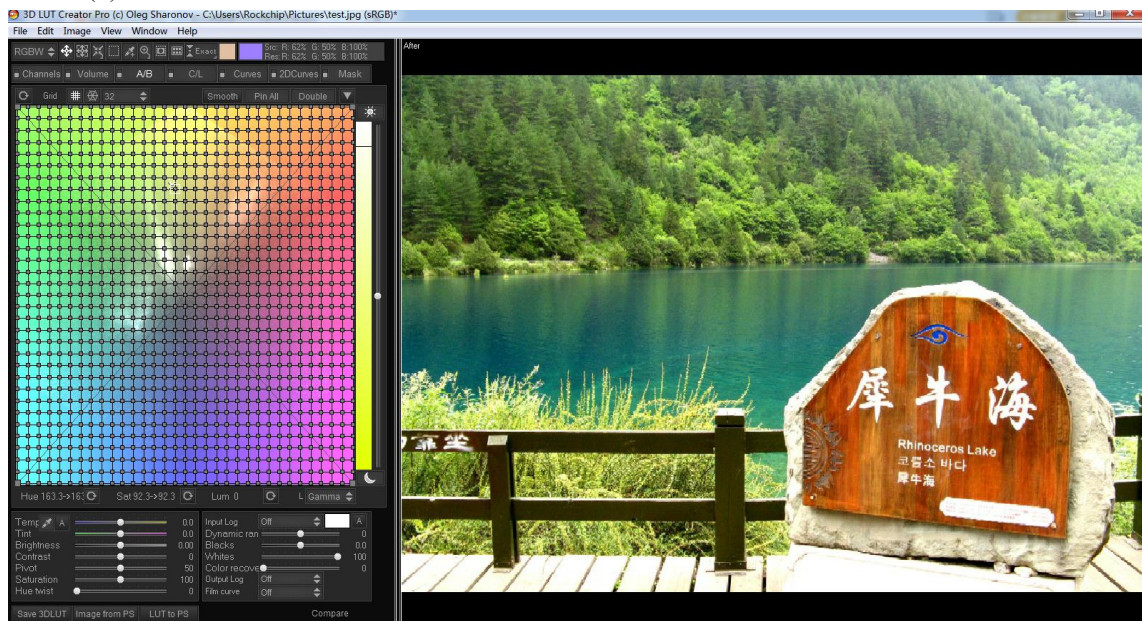


图 3 调节效果

(3) 保存效果：点击左下角 Save 3D LUT，弹出菜单后按 .lut 格式保存 lut 表；

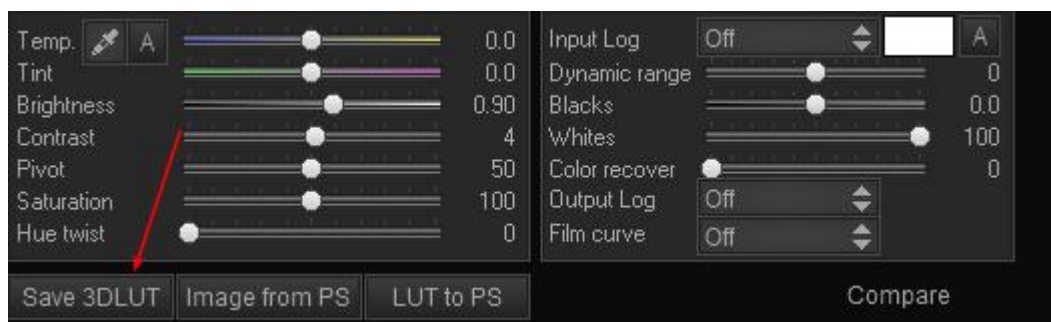


图 4 保存效果

### 1.3 传输 3D LUT 配置到文件

3D LUT 配置文件生成后需要传输到 Android 机器上，这一步需要使用 RKADBDTool 这个工具，工具使用界面如下：

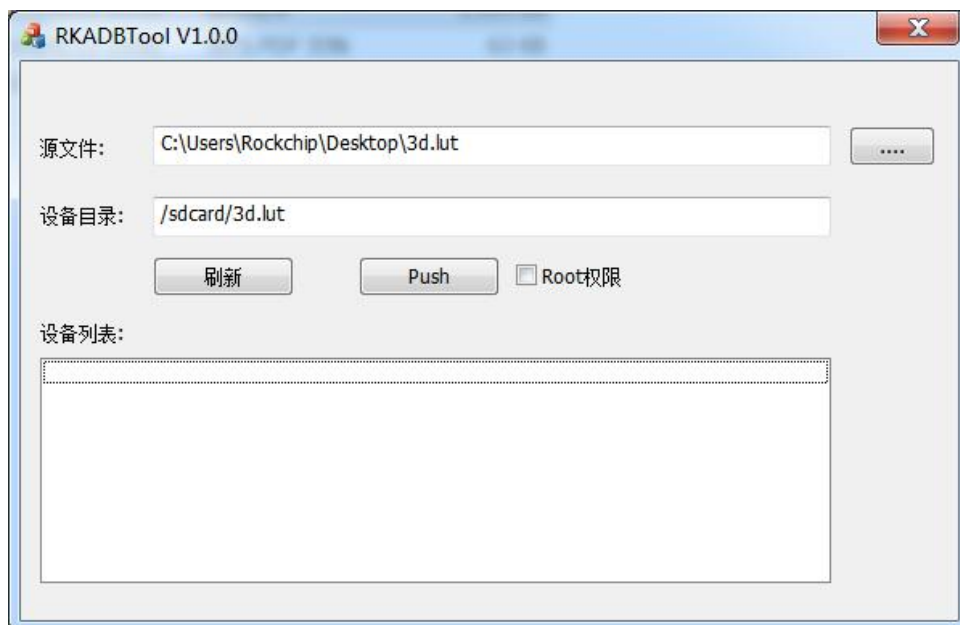


图 5 RKADBTool 工具使用界面

其中源文件选择 PC 上 3D LUT 配置文件的位置,设备目录填 Android 上保存配置文件的路径,默认为/sdcard/3d.lut, 点击 Push 即可将 3D LUT 配置文件从 PC 传到 Android 上, 注意使用该工具前必须保证 Android 上 ADB 能够使用。

## 1.4 屏幕显示调节 API

屏幕显示调节 API 位于 android.os.RkDisplayOutputManager 下, 以下为具体 api 介绍:

### **String[] getConnectorInfo()**

[功能] 获取当前屏幕信息;

[参数] void;

[返回值] 屏幕信息以字符串数组形式返回;

[补充说明] 屏幕信息示例为: ["type:14,id:0,state:1", "type:10,id:0,state:2", "type:11,id:0,state:2"], 其中 type 代表 connector type, id 代表 connector id, state 代表连接状态, 1 代表已连接, 2 代表未连接。

### **int setBrightness(int display, int bright\_percent)**

[功能] 设置亮度;

[参数]

**display**: 屏幕序号, 可用 getConnectorInfo 接口获取, 例如 getConnectorInfo 的结果为 ["type:14,id:0,state:1", "type:10,id:0,state:2", "type:11,id:0,state:2"], 其中 type:14,id:0 的屏幕序号为 0;

**bright\_percent**: 亮度;

[返回值] 返回亮度设置结果;

[补充说明] 亮度范围: 0-100, 默认为 50;

### **int getBrightness(int display)**

[功能] 获取亮度;

[参数]

**display**: 屏幕序号, 可用 getConnectorInfo 接口获取, 例如 getConnectorInfo 的结果为



[“type:14,id:0,state:1”, “type:10,id:0,state:2”, “type:11,id:0,state:2”], 其中 type:14,id:0 的屏幕序号为 0;

[返回值] 返回亮度;

[补充说明] 亮度范围: 0-100, 默认为 50。

#### **int setContrast(int display, int con\_percent)**

[功能] 设置对比度;

[参数]

**display**: 屏幕序号, 可用 getConnectorInfo 接口获取, 例如 getConnectorInfo 的结果为 [“type:14,id:0,state:1”, “type:10,id:0,state:2”, “type:11,id:0,state:2”], 其中 type:14,id:0 的屏幕序号为 0;

**con\_percent**: 对比度;

[返回值] 返回对比度设置结果;

[补充说明] 对比度范围: 0-100, 默认为 50。

#### **int getContrast(int display)**

[功能] 获取对比度;

[参数]

**display**: 屏幕序号, 可用 getConnectorInfo 接口获取, 例如 getConnectorInfo 的结果为 [“type:14,id:0,state:1”, “type:10,id:0,state:2”, “type:11,id:0,state:2”], 其中 type:14,id:0 的屏幕序号为 0;

[返回值] 返回对比度;

[补充说明] 对比度范围: 0-100, 默认为 50。

#### **int setSaturation(int display, int sat\_percent)**

[功能] 设置饱和度;

[参数]

**display**: 屏幕序号, 可用 getConnectorInfo 接口获取, 例如 getConnectorInfo 的结果为 [“type:14,id:0,state:1”, “type:10,id:0,state:2”, “type:11,id:0,state:2”], 其中 type:14,id:0 的屏幕序号为 0;

**sat\_percent**: 饱和度;

[返回值] 返回饱和度设置结果;

[补充说明] 饱和度范围: 0-100, 默认为 50。

#### **int getSaturation(int display)**

[功能] 获取饱和度;

[参数]

**display**: 屏幕序号, 可用 getConnectorInfo 接口获取, 例如 getConnectorInfo 的结果为 [“type:14,id:0,state:1”, “type:10,id:0,state:2”, “type:11,id:0,state:2”], 其中 type:14,id:0 的屏幕序号为 0;

[返回值] 返回饱和度;

[补充说明] 饱和度范围: 0-100, 默认为 50。

#### **int setHue(int display, int degree\_percent)**

[功能] 设置色调;

[参数]

**display**: 屏幕序号, 可用 getConnectorInfo 接口获取, 例如 getConnectorInfo 的结果为 [“type:14,id:0,state:1”, “type:10,id:0,state:2”, “type:11,id:0,state:2”], 其中 type:14,id:0 的屏幕序号为 0;

**degree\_percent**: 色调;

**[返回值]** 返回色调设置结果;

**[补充说明]** 色调范围: 0-100, 默认为 50。

### **int getHue(int display)**

**[功能]** 获取色调;

**[参数]**

**display:** 屏幕序号, 可用 `getConnectorInfo` 接口获取, 例如 `getConnectorInfo` 的结果为 `[“type:14,id:0,state:1”, “type:10,id:0,state:2”, “type:11,id:0,state:2”]`, 其中 `type:14,id:0` 的屏幕序号为 0;

**[返回值]** 返回色调;

**[补充说明]** 色调范围: 0-100, 默认为 50。

### **int setGamma(int display, int size, int[] red, int[] green, int[] blue)**

**[功能]** 设置色温;

**[参数]**

**display:** 屏幕序号, 可用 `getConnectorInfo` 接口获取, 例如 `getConnectorInfo` 的结果为 `[“type:14,id:0,state:1”, “type:10,id:0,state:2”, “type:11,id:0,state:2”]`, 其中 `type:14,id:0` 的屏幕序号为 0;

**size:** rgb 数组长度;

**red, green, blue:** rgb 数组, 数组长度应与 `size` 一致;

**[返回值]** 返回色温设置结果;

**[补充说明]** 3326, 3368 平台 `lutsize` 为 256,3288,3399,356x 平台 `lut size` 为 1024。

### **set3DLut(int display, int size, int[] red, int[] green, int[] blue)**

**[功能]** 设置 3D LUT;

**[参数]**

**display:** 屏幕序号, 可用 `getConnectorInfo` 接口获取, 例如 `getConnectorInfo` 的结果为 `[“type:14,id:0,state:1”, “type:10,id:0,state:2”, “type:11,id:0,state:2”]`, 其中 `type:14,id:0` 的屏幕序号为 0;

**size:** rgb 数组长度;

**red, green, blue:** rgb 数组, 数组长度应与 `size` 一致, 这组数据应由 PC 工具生成转换而来;

**[返回值]** 返回 3D LUT 设置结果;

**[补充说明]** 现在仅 356X 平台上支持 3D LUT 设置, `size` 为 729。

### **int updateDispHeader()**

**[功能]** 更新 `baseparameter` 分区中 `disp` 的 `header` 信息;

**[参数]** void;

**[返回值]** 返回更新结果;

**[补充说明]** 应在第一次使用 `baseparameter` 分区时更新当前屏幕信息到 `baseparameter` 中。