**密级状态：** 绝密（　　） 秘密（　　） 内部资料（　　） 公开（ √ ）

# Rockchip_Introduction_Android_Box_Display_Framework_Configuration_CN

（技术部，第一系统产品部）

| | | |
|---|---|---|
| **文件状态：**<br>[ ] 草稿<br>[ ] 正在修改<br>[√] 正式发布 | **文件标识：** | RK-SM-YF-504 |
| | **当前版本：** | 1.0.1 |
| | **作　　者：** | YHC, ASX, WGH |
| | **完成日期：** | 2018-02-10 |
| | **审　　核：** | CW, ZXZ |
| | **审核日期：** | 2019-11-27 |

福州瑞芯微电子股份有限公司
Fuzhou Rockchip Electronics Co.,Ltd.

## 免责声明

本文档按“现状”提供，福州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，由其各自拥有者所有。




福州瑞芯微电子股份有限公司
Fuzhou 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
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

# 前言

## 概述

本文档是基于 Rockchip Android 8.1 及以上版本的 BOX 平台开发显示框架的配置帮助文档。

介绍了 BOX 平台显示框架配置，包括用户显示设置：分辨率、缩放、颜色、BCSH 以及显示框架所需配置：Framebuffer 分辨与和主副显示设备配置。

## 产品版本

| 芯片名称 | 内核版本 | Android 版本 |
| --- | --- | --- |
| RK3328/RK3368/RK3229 | Linux 4.4/Linux 4.19 | >=Android 8.1 |
| | | |

## 读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

## 修订记录

| 日期 | 版本 | 作者 | 修改说明 |
| --- | --- | --- | --- |
| 2018-2-10 | V1.0.0 | YHC, ASX | 创建初始版本 |
| 2019-11-26 | V1.0.1 | WGH | 修改部分字段 |
| | | | |
| | | | |
| | | | |

# 目录

**ROCKCHIP_INTRODUCTION_ANDROID_BOX_DISPLAY_FRAMEWORK_CONFIGURATION_CN ..... I**
**前言 ..... 1**
**1. 显示设置流程 ..... 3**
1.1 显示设置流程 ..... 3
1.2 分辨率过滤配置 ..... 3
**2. BASEPARAMETER 分区 ..... 5**
2.1 BASEPARAMETER.IMG ..... 5
2.2 SAVEBASEPARAMETER 工具 ..... 5
2.3 分区数据结构 ..... 6
2.3.1 分区数据结构整体概览 ..... 6
2.3.2 disp_info 说明 ..... 7
2.3.3 Backup 区域 ..... 9
2.4 其他调试信息 ..... 10
2.5 FAQ ..... 10

 

# 1. 显示设置流程

## 1.1 显示设置流程

当前 RK 平台 Android8.1 及以上 android 版本采用的是 DRM（Direct Rendering Manager）显示框架。上层接口配置 DRM 相关参数，最终由 HWC 通过 libdrm 配置下去，整理流程如下图所示：

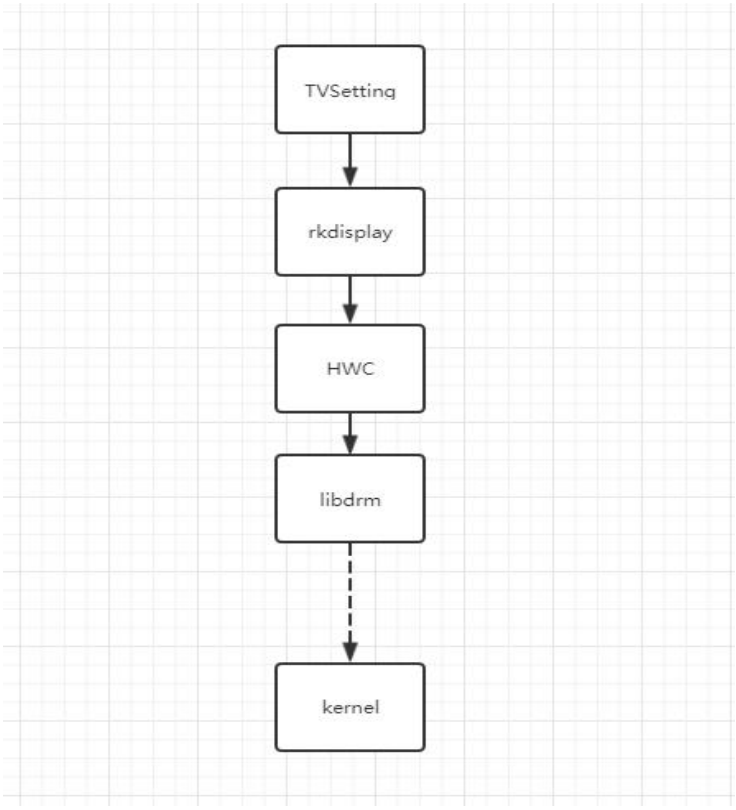


图 1 显示设置流程

## 1.2 分辨率过滤配置

因为初始获取到的全部分辨率过多，有些分辨率对用户来说并不需要，因此在 SDK 的 HWC 模块中对 HDMI 和 DP 等拔插设备的分辨率进行了过滤。

位于 device/rockchip/common/resolution_white.xml 路径的配置文件定义了能够通过过滤的白名单，HWC 中会根据该配置文件对初始的分辨率进行过滤筛选后再传递给上层，该 XML 文件的每一个<resolution>块定义了一个能够通过过滤的分辨率，其中详细项的定义如下：

表 1 分辨率过滤项定义说明

| 项定义 | 说明 |
| --- | --- |
| clock | 时钟 |
| hdisplay | 见图 2 的标示 |

| 项定义 | 说明 |
| --- | --- |
| hsync_start | 见图 2 的标示 |
| hsync_end | 见图 2 的标示 |
| htotal | 见图 2 的标示 |
| hskew | 见图 2 的标示 |
| vdisplay | 见图 2 的标示 |
| vsync_start | 见图 2 的标示 |
| vsync_end | 见图 2 的标示 |
| vtotal | 见图 2 的标示 |
| vscan | 见图 2 的标示 |
| vrefresh | 显示设备帧率 |
| flags | flags 的定义如下：<br>`DRM_MODE_FLAG_PHSYNC (1<<0)`<br>`DRM_MODE_FLAG_NHSYNC (1<<1)`<br>`DRM_MODE_FLAG_PVSYNC (1<<2)`<br>`DRM_MODE_FLAG_NVSYNC (1<<3)`<br>`DRM_MODE_FLAG_INTERLACE (1<<4)` |
| vic | HDMI 标准对应定义的 VIC 值，如 HDMI 标准中未定义置 0 |

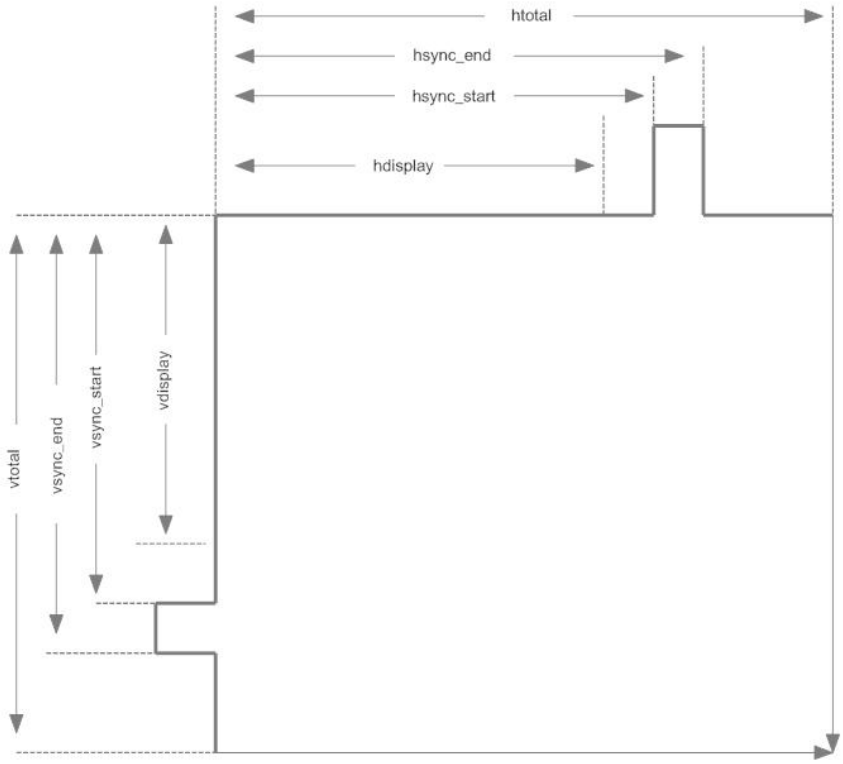


图 2 分辨率项定义示意图

# 2. Baseparameter 分区

显示设置数据都存放在了 Baseparameter 分区中，该分区大小为 1MB，通过烧写 baseparameter.img 来设置初始值。

## 2.1 baseparameter.img

SDK 默认使用的 baseparameter.img 的路径为 device/rockchip/common/baseparameter/baseparameter_fb1080.img。可以使用 saveBaseParameter 工具制作该分区镜像，并在 BoardConfig.mk 中将 TARGET_BASE_PARAMETER_IMAGE 设置为新镜像的路径：

```
TARGET_BASE_PARAMETER_IMAGE := device/rockchip/rk3328/baseparameter.img
```

## 2.2 saveBaseParameter 工具

SDK 自带了 saveBaseParameter 工具能够查看、设置和导出 baseparameter 分区，该工具的源码位于 device/rockchip/common/baseparameter/saveBaseParameter 目录，默认 eng 和 userdebug 编译模式下会编译该工具，可以直接在设备中执行 saveBaseParameter 命令。命令的常用参数如下：

**1) 帮助信息打印**

使用-h 参数可以打印出命令的帮助，如下所示：

```
rk3328_box:/ $ saveBaseParameter -h

saveParameter: read and write baseparameter partition tool

Usage:
        -h      Help info
        -p      Print Baseparamter
        -t      output to target file (e: "/sdcard/baseparameter.img)"
        -d      Choose Display to Setting (e: 0 or 1)
        -f      Framebuffer Resolution (e: 1920x1080@60)
        -D      Display Attach Devices (e: HDMI-A,TV)
        -c      Color (e: RGB-8bit or YCBCR444-10bit)
        -u      Is Enable Auto Resolution (2:auto resolution; 1:set one fixed resolution)
        -o      Overscan (e: overscan "100,100,100,100")
        -b      BCSH (e: "50,50,50,50")
        -R      Reset Baseparameter (1:only reset user setting baseparameter partition; 2:reset baseparameter paratition include backup)

Example: saveBaseParameter -d 0 -f 1920x1080@60 -D "HDMI-A,TV" -c Auto -u 2 -o "100,100,100,100" -b "50,50,50,50"

===== Rockchip All Rights Reserved =====

rk3328_box:/ $
```

图 3 saveBaseParameter 帮助

**2) 分区数据打印**

```
rk3328_box:/ # saveBaseParameter -p
print baseparameter
========== base parameter ==========
-main:
        resolution: 3840x2160@p-4016-4104-4400-2168-2178-2250-5
        corlor: format 0 depth 8
        fbinfo: 1920x1080@60.000000 device:HDMI-A,TV
        bcsh: 50 50 50 50
        overscan: 95 100 95 100
        feature:  0x0
-aux:
        resolution: 0x0@p-0-0-0-0-2178-0-0
        corlor: format 0 depth 0
        fbinfo: 0x0@0.000000 device:
        bcsh: 0 0 0 0
        overscan: 0 0 0 0
        feature:  0x0

========= backup parameter ==========
-main:
        resolution: 0x0@p-0-0-0-0-0-0-0
        corlor: format 4 depth 0
        fbinfo: 1920x1080@60.000000 device:HDMI-A,TV
        bcsh: 50 50 50 50
        overscan: 100 100 100 100
        feature:  0x3
-aux:
        resolution: 0x0@p-0-0-0-0-0-0-0
        corlor: format 0 depth 0
        fbinfo: 0x0@0.000000 device:
        bcsh: 0 0 0 0
        overscan: 0 0 0 0
        feature:  0x0
====================================
```

图 4 saveBaseParameter 打印分区数据

其中 baseparameter 分区主要分为两区域：基本区域和备份区域。基本区域在设置时候会被写入用户设置，当恢复出厂设置时，会将基本区域数据擦除，然后将备份区域数据写入基本区域。

每个区域又分为主显示和副显示，其中每个都有如下的值：

- resolution：分辨率
- color：颜色
- fbinfo：framebuffer 分辨率
- device：显示设备
- bcsh：颜色、对比度、饱和度、灰度（Brightness, Contrast, Saturation , Hue）
- overscan：调整缩放（顺序依次为：left, top, right, bottom）
- feature：特殊功能字段

**3) 导出文件**

使用-t <path>参数能将 baseparameter 分区导出至指定文件路径，而后将该 img 文件 pull 出来，就可以直接使用该 img 进行烧写。

## 2.3 分区数据结构

### 2.3.1 分区数据结构整体概览

```
struct disp_info{
    struct drm_display_mode resolution;
    struct overscan scan;
    enum output_format  format;
```

```
    enum output_depth depthc;
    unsigned int feature;
struct hwc_inital_info hwc_info;
struct bcsh_info bcsh;
unsigned int reserve[128];
struct lut_data mlutdata;
};
struct disp_info main;
struct disp_info aux;
```

main、aux 分别保存主副屏信息，main 的信息存储在 baseparameter 前 8kb 的空间内，aux 偏移为 8kb。

disp_info 的大小不为 8kb，因此 main 和 aux 读取和写入的时候需要分开，lseek 8kb 之后再读写 aux。

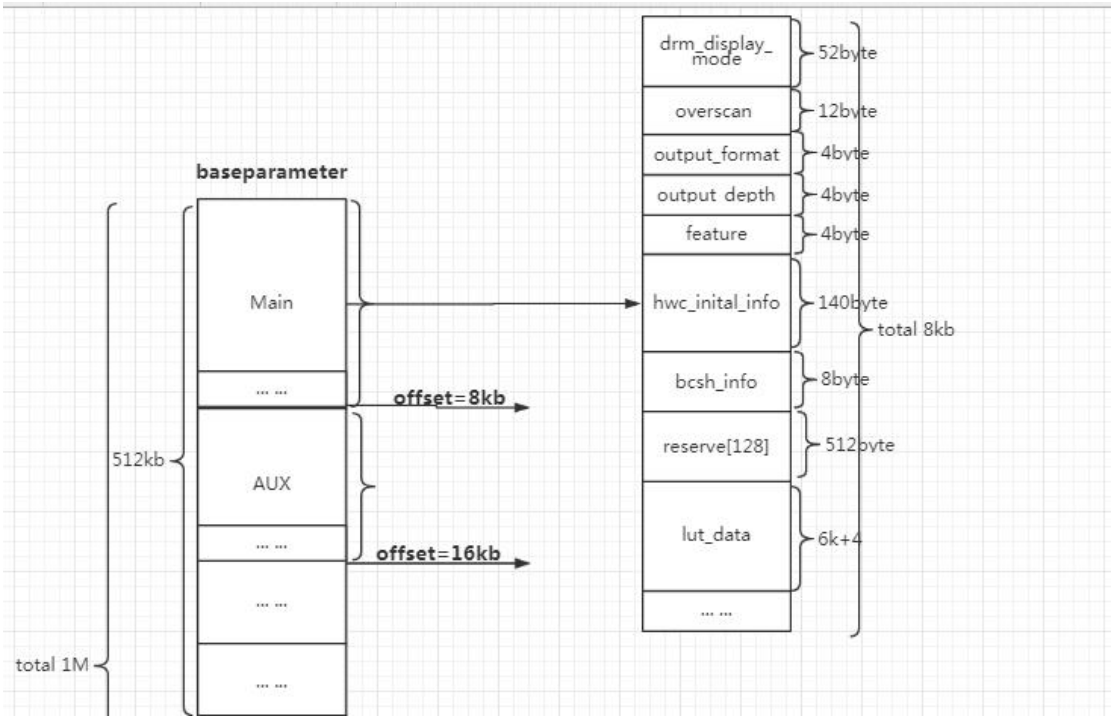


图 5 分区结构概览

### 2.3.2 disp_info 说明

1）drm_display_mode

存储分辨率的时序信息

```
struct drm_display_mode {
    /* Proposed mode values */
    int clock;       /* in kHz */
    int hdisplay;
    int hsync_start;
    int hsync_end;
    int htotal;
    int vdisplay;
    int vsync_start;
    int vsync_end;
    int vtotal;
    int vrefresh;
```

```
    int vscan;
    unsigned int flags;
    int picture_aspect_ratio;
};
```

2）overscan

存放缩放相关：

```
struct overscan {
    unsigned int maxvalue;
    unsigned short leftscale;
    unsigned short rightscale;
    unsigned short topscale;
    unsigned short bottomscale;
};
```

3）output_format

颜色格式，属性字符串为 Auto 时，选择 output_ycbcr_high_subsampling

```
enum output_format {
    output_rgb=0,
    output_ycbcr444=1,
    output_ycbcr422=2,
    output_ycbcr420=3,
    output_ycbcr_high_subsampling=4,  // (YCbCr444 > YCbCr422 > YCbCr420 >
RGB)
    output_ycbcr_low_subsampling=5    , // (RGB > YCbCr420 > YCbCr422 >
YCbCr444)
    invalid_output=6,
};
```

4）output_depth

色深，属性字符串为 Auto 时，选择 Automatic

```
enum  output_depth{
    Automatic=0,
    depth_24bit=8,
    depth_30bit=10,
};
```

5）feature

feature 目前有如下 flag，配置分辨率以及颜色的 AUTO 模式，是否开启 hdcp1x，是否过滤分辨率列表

```
#define RESOLUTION_AUTO        (1<<0)
#define COLOR_AUTO             (1<<1)
#define HDCP1X_EN              (1<<2)
#define RESOLUTION_WHITE_EN       (1<<3)
```

6）hwc_inital_info

配置 FrameBuffer 大小，fps，以及主副屏的挂载设备

```
struct hwc_inital_info{
    char device[128];
    unsigned int framebuffer_width;
    unsigned int framebuffer_height;
    float fps;
};
```

device 例子：HDMI-A,TV

Fps 可以用来限制帧率

7）bcsh_info

保存 bcsh 信息，取值范围 0~100，默认值都是 50

```
struct bcsh_info {
    unsigned short brightness;
    unsigned short contrast;
    unsigned short saturation;
    unsigned short hue;
};
```

8）Reserve

unsigned int reserve[128];

预留信息，供以后使用。

9）lut_data

保存 lut 表信息，size 表示每个 rgb lut 表中有多少个数据，最大为 1024。

```
struct lut_data{
    uint16_t size;
    uint16_t lred[1024];
    uint16_t lgreen[1024];
    uint16_t lblue[1024];
};
```

### 2.3.3 Backup 区域

由于有恢复出厂设置的需求，baseparameter 的后半部分用来保存初始数据，起始偏移 512kb。

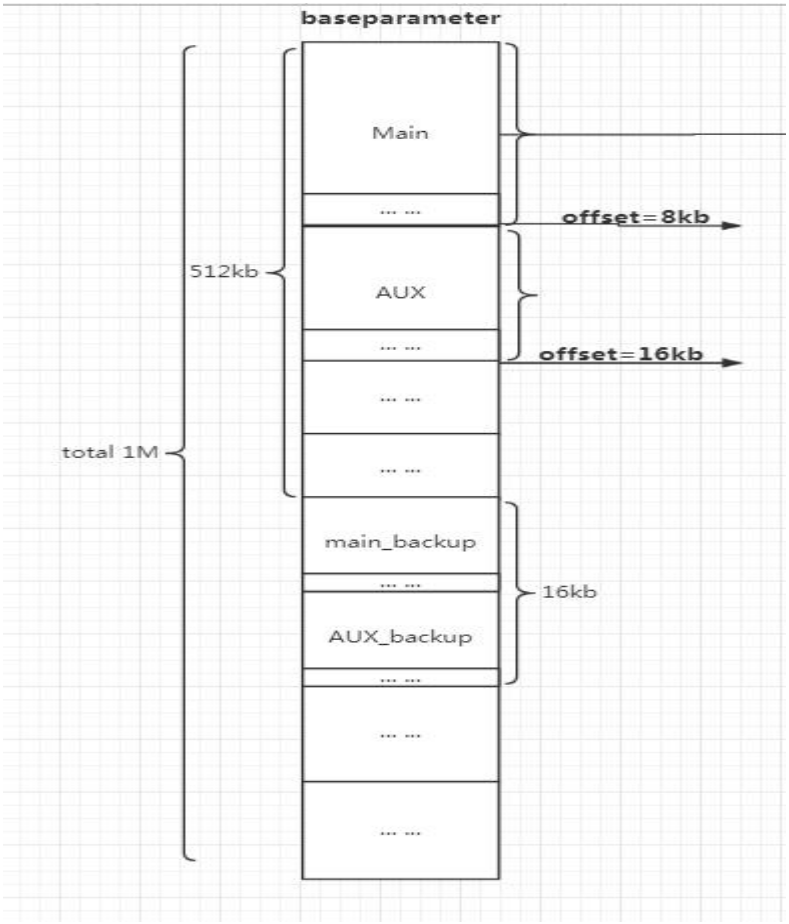


图 6 backup 区域分区结构

## 2.4 其他调试信息

1) 属性值

可以通过以下两个只读属性来分别查询主副显示器的输出接口的名称。

表 2 主副显示器查询

| 属性 | 功能说明 |
| --- | --- |
| vendor.hwc.device.main | 查询当前主显的输出接口 |
| vendor.hwc.device.aux | 查询当前副显的输出接口 |

2) HWC 日志

通过抓取 logcat |grep BP 命令能够看到 HWC 开机时候读取 baseparameter 分区是否正确。

## 2.5 FAQ

1) 将 framebuffer 分辨率更改为 1280x720 要如何操作？

a) 获取 root 权限

```
su
```

b) 清除分区

```
saveBaseParameter -R 2
```

c) 写入设置

```
saveBaseParameter -d 0 -f 1280x720@60 -D "HDMI-A,TV" -c Auto -u 2 -o
"100,100,100,100" -b "50,50,50,50"
```

d) 打印确认

```
saveBaseParameter -p
```

e) 导出 image

```
saveBaseParameter -t /sdcard/baseparameter.img
adb pull /sdcard/baseparameter.img
```

f) 烧写确认

烧写生成的 baseparameter.img，而后重启开机后执行 logcat |grep BP 确认 HWC 所读取的数据是否正确。

g) 放置到 SDK 中

例如放到 device/rockchip/rk3328/baseparameter.img，而后按如下补丁修改

device/rockchip/rk3328/rk3328_box/BoardConfig.mk，重新编译后即可。

```
diff --git a/rk3328_box/BoardConfig.mk b/rk3328_box/BoardConfig.mk
index 3831ce8..f151fe7 100755
--- a/rk3328_box/BoardConfig.mk
+++ b/rk3328_box/BoardConfig.mk
@@ -16,6 +16,8 @@

 TARGET_BOARD_PLATFORM_PRODUCT := box

+TARGET_BASE_PARAMETER_IMAGE :=
device/rockchip/rk3328/baseparameter.img
+
 # Use the non-open-source parts, if they're present
 -include vendor/rockchip/rk3328/BoardConfigVendor.mk
 -include device/rockchip/common/BoardConfig.mk
```