

Rockchip Linux Graphics开发指南

文档标识: RK-SM-YF-345

发布版本: V1.2.0

日期: 2023-12-22

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

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, 由其各自拥有者所有。

版权所有© 2023 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

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

文档主要介绍 Rockchip Linux Graphics使用说明，旨在帮助工程师更快上手Graphics开发及相关调试方法。

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

各芯片系统支持状态

芯片名称	Buildroot	Debian	Yocto
RK180X	Y	N	N
RK3036	Y	N	N
RK312X/PX3SE	Y	N	N
RK322X	Y	N	N
RK3288	Y	Y	Y
RK3308	Y	N	N
RK3326/PX30	Y	Y	Y
RK3358	Y	N	N
RK3399/RK3399Pro	Y	Y	Y
RK356X	Y	Y	Y
RK3588	Y	Y	Y

修订记录

日期	版本	作者	修改说明
2020-03-18	V1.0.0	Caesar Wang	初始版本
2020-07-03	V1.1.0	Caesar Wang	增加屏幕参数调整说明
2020-07-28	V1.1.1	Ruby Wang	一些语句描述修正
2023-12-21	V1.2.0	ZhiZhan Chen	适配最新SDK

目录

Rockchip Linux Graphics开发指南

1. Rockchip Linux Graphics介绍
 - 1.1 概述
 - 1.2 芯片硬件模块介绍
 - 1.2.1 VOP (Video Output Processor)
 - 1.2.2 GPU (Graphics Process Unit)
 - 1.2.3 RGA (Raster Graphic Acceleration)
 - 1.3 图像软件模块介绍
 - 1.3.1 LIBDRM
 - 1.3.2 LIBMALI
 - 1.3.3 ZERO-COPY
 - 1.3.4 X11
 - 1.3.5 Wayland
 - 1.3.6 None
 - 1.3.7 显示架构的选择
 - 1.4 多屏显示功能的介绍
 - 1.4.1 Debian双屏显示功能介绍
 - 1.4.1.1 系统的显示设备及设备名
 - 1.4.1.2 双屏显示模式设置
 - 1.4.2 Buildroot双屏显示功能介绍
 - 1.5 屏幕显示效果调整

1. Rockchip Linux Graphics介绍

1.1 概述

Rockchip Linux平台的图形处理模块，基于DRM和DMA-BUF技术，提供了符合ARM Linux标准的高效图形处理环境。其主要优势在于通用架构，简化了定制化开发流程，并且允许广泛利用现有组件，成为众多开源项目在ARM端的首选适配平台。尽管如此，由于技术和架构复杂性，理解和应用这一平台可能需要一定的学习过程。开发者可通过查阅 `docs/cn/Common/DISPLAY/`、`docs/cn/Linux/Graphics/` 文档和[Rockchip wiki](#)来获取深入的信息和指南，这些资料是掌握平台核心技术的重要资源。

1.2 芯片硬件模块介绍

1.2.1 VOP (Video Output Processor)

VOP是从存储器帧缓冲区到显示设备的显示接口，是显示单元用来显示图像（比如输入NV12,RGB的Buffer，显示到屏幕）。

各个平台VOP的基础特性不尽相同，比如各个显示接口支持的最大分辨率，支不支持4K分辨率，AFBC压缩格式等等，具体细节可参考

`docs/cn/Common/DISPLAY/Rockchip_Developer_Guide_DRM_Display_Driver_CN.pdf`。

1.2.2 GPU (Graphics Process Unit)

GPU是一个基于开放标准的图形加速平台。它支持 2D 图形、3D 图形以及 GPU 上的通用计算（GPGPU）。

基本上Rockchip Linux的GPU有提供OpenGL ES, EGL, OpenCL的 API，不支持OPENGL。支持类型如下（各个芯片平台支持的类型不尽相同，具体可以参考对应的芯片手册）：

- OpenGL ES 3.0
- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenCL 1.2
- OpenCL 1.1
- OpenCL 1.0

1.2.3 RGA (Raster Graphic Acceleration)

Rockchip RGA是一个独立的二维光栅图形加速单元。它加速了二维图形操作，例如点/线绘制、图像缩放、旋转、位图、图像合成等。不同芯片平台支持的图像格式不尽相同，具体可以参考对应的芯片手册。代码位于“SDK/external/linux-rga”目录，更多信息可以参考

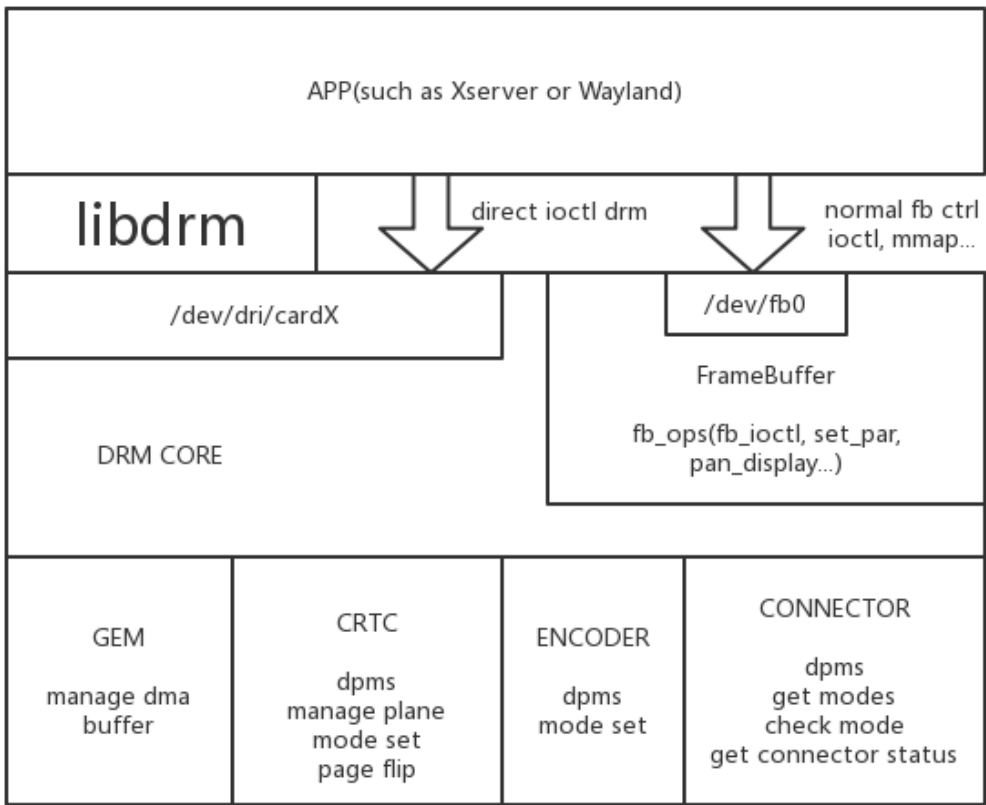
`docs/cn/Common/RGA/Rockchip_Developer_Guide_RGA_CN.pdf`。

1.3 图像软件模块介绍

主要涉及理解 libdrm、wayland 和 x11（合成器），以及 mesa、libmali 和 gtk（应用程序）之间的关系。相关信息可参考 [Linux Graphics Stack](#)。

1.3.1 LIBDRM

LIBDRM是一个跨驱动的中间件，它允许用户空间应用（例如作为Mesa和2D驱动程序）通过DRI与内核通信协议。参考如下DRM结构图：



LIBDRM是DRM下沟通驱动和用户层的库。过去APP可能是直接使用open(fb)这样的方式来和图形驱动沟通，但是在现在的硬件演化下，已经不合适了。有以下一些理由：

- 有多个图层怎么办？
- 有多个屏幕怎么办？
- 怎么处理vsync的问题，怎么同步不撕裂？
- 怎么利用上dmabuf，做到memory zero-copy？

LIBDRM的存在就是用来方便用户层和驱动这些问题，提供API给X11, Wayland这样的display backend使用。如果程序比较简单，比如一个广告机循环播放视频，那是可以直接调用LIBDRM，但是不建议直接使用LIBDRM的API。原因很多：

- 首先，LIBDRM本身是一个比较新的API，而且接触者局限在相关的DRM驱动开发和Wayland/Xserver开发上。
- 另外，LIBDRM的演进比较快，比如说API还分atomic和legacy, 很多效果依赖于厂商的实现，Rockchip有修改一些core API的表现效果来辅助产品项目，所以同一个API在不同平台表现有可能完全不同。这里不是建议使用X11和Wayland，而是用我们封装好的东西。比如，广告机循环播放视频，最好是使用gststreamer，然后选kmssink显示，而不是直接调用LIBDRM的API。

DRM里有crtc, plane, connector这三个概念, 可以这么理解:

- connector就是屏幕, 比如一个HDMI一个connector num, 一个DSI一个connector num。
- crtc表示vop, 一个屏幕一般对应一个crtc。
- plane就是图层, 比如视频层在plane2, UI在plane1, 视频在UI上面。

DRM里API分两套, legacy和atomic。

legacy看名字就是早期的API, 我们现在大部分程序也都是用legacy API。里面有几个关键的功能接口需要注意, drmModeSetCrtc包括了drmModeSetPlane和drmModePageFlip。

drmModeSetCrtc一般是用来设置UI层, 同时用来设置分辨率。

drmModeSetPlane用来设置不同图层的显示, 比如视频。参数分别是要显示buffer fd, 要操作的图层, 要显示的大小, buffer的大小。它会缩放buffer显示到屏幕上。在Rockchip平台上这个API是async的, 连续调用两次, 前面的就被覆盖了, 此时可能需要drmWaitVBlank一下。

drmModeSetPlane为什么会被这么多部分调用? 因为想在legacy的API上也完成多图层的显示。假设目前有图层1和图层2, 图层1调用一次drmModeSetPlane, 图层2也调用一次drmModeSetPlane, 然后它们都等一个vsync单位的时间, 如果屏幕刷新率60hz, 那么最大帧数是不是只有30fps? 为了解决这个问题, 上游的人又开发了atomic的API。

其中atomic API的实质可以理解为一个提交包括了所有图层的更新信息。这样就不用调用两次drmModeSetPlane了, 而是一次的drmModeAtomicCommit, 跟上所有的参数。

```
mainline source code:  
git clone https://gitlab.freedesktop.org/mesa/drm.git
```

详情请参考以下资料:

- Rockchip wiki官网[wiki Libdrm](#)。
- [官方文档](#)。

legacy的实例

- [libdrm里面的modetest](#)
- [gststreamer里的kmssink](#)

atomic的实例

- [modeset-atomic](#)

1.3.2 LIBMALI

综上所述, GPU是提供opengles, egl, opengl API, 若想要这几个工作, 就需要把LIBMALI加进rootfs里。

默认的binary在"SDK/external/libmali"目录下。

命名规则: GPU型号-软件版本-硬件版本(如果有的话, 比如说r1p0区分RK3288和RK3288w)-编译选项。

要注意编译选项:

不带后缀。是x11-gbm, 注意GBM是配置DRM使用的memory机制, 如果不是3.10的kernel, 不要用fbdev。GBM是给QT EGLFS程序用的, 不依赖X11,Wayland。Wayland/ Wayland-gbm给Wayland使用。

1.3.3 ZERO-COPY

用mali显示dmabuf的数据，比如说摄像头，视频，其实是可以利用dmabuf ZERO-COPY机制优化的。不然载入texture还需要CPU去拷贝。X11/Wayland有ZERO-COPY的配置，可以搜下相关的Wayland dmabuf。

1.3.4 X11

就和一般桌面平台差不多，不过X11有个GPU性能缺陷的问题。源码位于：“SDK/external/xserver”目录下。

更多关于X11的细节，可以参考以下链接：

```
https://en.wikipedia.org/wiki/X.Org_Server
https://www.comptechdoc.org/os/linux/howlinuxworks/linux_hlxwindows.html
https://dri.freedesktop.org/wiki/DDX/
https://www.freedesktop.org/wiki/Software/Glamor/
https://en.wikipedia.org/wiki/X.Org_Server
```

1.3.5 Wayland

建议使用Yocto/Buildroot SDK做Wayland的开发。效率上Wayland要比X11好点，主要是兼容性问题。如果不需要桌面，又要多窗口，可以尝试使用Wayland。

参考资料：

```
https://en.wikipedia.org/wiki/Wayland
```

1.3.6 None

除了X11和Wayland之外，还有None，这也是嵌入式上接触比较多的。比如MiniGUI，SDL皆是如此。

1.3.7 显示架构的选择

- EGL program + X11
- X11
- Wayland
- None

多窗口的功能需求,选择:

- X11
- Wayland

桌面的功能需求,选择:

- X11

4K视频播放+全屏:

- X11

- Wayland

4K视频播放+多窗口:

- X11
- Wayland

1.4 多屏显示功能的介绍

Rockchip Linux 在 Debian/Buildroot 平台上对DP/HDMI/MIPI/eDP/LVDS等显示接口可以任意组合，支持多屏同显或异显的功能。当双屏异显时，一个显示接口当主屏，另一个当副屏。下面主要介绍双屏异显功能。

1.4.1 Debian双屏显示功能介绍

在 Debian 使用X11系统，可以使用 xrandr 去设置双屏同显和异显功能。

"xrandr" 是一款官方的 RandR (Resize and Rotate)Wikipedia:X Window System 扩展配置工具。它可以设置屏幕显示的大小、方向、镜像等。对多显示器的情况请参考 [Multihead](#) 页面。

1.4.1.1 系统的显示设备及设备名

支持命令行和界面下对双屏显示模式进行设置。

菜单界面:

使用鼠标点击右键，应用程序->设置->显示:



命令行:


```
su linaro -c "DISPLAY=:0 xrandr"
```

输出:

```
root@linaro-alip:/# su linaro -c "DISPLAY=:0 xrandr"
Screen 0: minimum 320 x 200, current 3000 x 1920, maximum 16384 x 16384
HDMI-1 connected 1920x1080+0+0 (normal left inverted right x axis y axis) 0mm x
0mm
    1920x1080    60.00*+  60.00
    1600x900     60.00
    1280x1024    60.02
    1280x720     60.00    60.00
    720x480      59.94    59.94
HDMI-2 disconnected (normal left inverted right x axis y axis)
DSI-1 connected 1080x1920+1920+0 (normal left inverted right x axis y axis) 0mm x
0mm
    1080x1920    59.90*+
DP-1 disconnected (normal left inverted right x axis y axis)
DP-2 disconnected (normal left inverted right x axis y axis)
```

可以看到当前系统有两个显示设备，设备名分别为 HDMI-1 和 DSI-1。

1.4.1.2 双屏显示模式设置

双屏支持双屏同显，双屏异显模式。异显模式下支持 On right、Above、On left、Below 四种模式。

菜单界面:

在显示设置中使用鼠标拖动屏幕的位置,则可以切换双屏的显示模式。

命令行:

```
su linaro -c "DISPLAY=:0 xrandr --output HDMI-1 --above DSI-1"
```

其中--above 参数可以设置为 right-of, left-of, below, same-as 切换双屏的显示模式。

Default/same-as 模式下为双屏同显。

1.4.2 Buildroot 双屏显示功能介绍

Buildroot SDK 的 Weston 支持多屏同异显及热拔插等功能，不同显示器屏幕的区分根据 drm 的 name（通过 Weston 启动 log 或者 /sys/class/drm/card0-<name> 获取），相关配置通过环境变量设置，如：

```
# /etc/profile.d/weston.sh
export WESTON_DRM_PRIMARY=HDMI-A-1 # 指定主显为HDMI-A-1
export WESTON_DRM_SINGLE_HEAD=1 # 强制单显
export WESTON_DRM_MIRROR=1 # 使用镜像模式（多屏同显），不设置此环境变量即为异显
export WESTON_DRM_KEEP_RATIO=1 # 镜像模式下缩放保持纵横比，不设置此变量即为强制全屏
export WESTON_DRM_HEAD_MODE=primary # 只使能主显
export WESTON_DRM_HEAD_MODE=internal # 只使能内置显示器
export WESTON_DRM_HEAD_MODE=external # 只使能外置显示器
export WESTON_DRM_HEAD_MODE=external-dual # 使能所有显示器，优先外置显示器
export WESTON_DRM_HEAD_FALLBACK=1 # 未匹配到显示器时，使能任意一个有效显示器
export WESTON_DRM_MASTER=1 # 允许关闭未使用的屏幕
```

```
export WESTON_OUTPUT_FLOW=horizontal # 默认水平排列
export WESTON_OUTPUT_FLOW=vertical # 默认垂直排列
export WESTON_OUTPUT_FLOW=same-as # 所有显示器默认位置(0,0)
```

镜像模式缩放显示内容时需要依赖RGA加速。

同时也支持在weston.ini的output段单独禁用指定屏幕：

```
# /etc/xdg/weston/weston.ini

[output]
name=LVDS-1

mode=off
# off|current|preferred|<WIDTHxHEIGHT@RATE>
```

更多细节请参考 [docs/cn/Linux/Graphics/Rockchip_Developer_Guide_Buildroot_Weston_CN.pdf](#)

1.5 屏幕显示效果调整

在 Linux 平台上可以通过 modetest 对屏幕的色调、饱和度、对比度和亮度（hue, saturation, contrast and brightness）等参数进行调整。

使用modetest和modetest -w选项设置 -w <obj_id>:<prop_name>: 相关属性。

比如设置EDP屏幕的色调：

modetest -M rockchip 可以获取eDP屏幕Connectors id是92。

```
Connectors:
id      encoder status      name      size (mm)      modes      encoders
92      91      connected  eDP-1      129x171      1          91
modes:
      name refresh (Hz) hdisp hss hse htot vdisp vss vse vtot)
1536x2048 60 1536 1548 1564 1612 2048 2056 2060 2068 200000 flags: nhsync,
nvsync; type: preferred
...
```

设置EDP屏幕的hue值为60，默认hue值是50，可以调整范围是0~100。

```
modetest -M rockchip -w 92:hue:60
```

同理可以设置lvds/hdmi/mipi等其他屏幕的色调、饱和度、对比度和亮度等参数。

另外，X11上，可以通过xrandr，调节gamma属性。

首先查看板子“etc/X11/xorg.conf.d/20-modesetting.conf”内是否包含配置：

```
Option      "UseGammaLUT"      "true"
```

若不包含，可更新xserver deb安装包以获得支持。

修改命令（指定屏幕，修改RGB分量的gamma权重）：

```
xrandr --output HDMI-1 --gamma 1:1:0.9
```