

# Rockchip Gstreamer用户指南

---

文件标识: RK-YH-YF-921

发布版本: V1.0.0

日期: 2022-01-06

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 © 2022 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

本文档主要介绍 Gstreamer及相关插件的编译和测试方法。

## 产品版本

芯片名称	版本
RK356X	1.14.x
RK3588	1.18.x

## 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
2022-01-06	V1.0.0	Jair Wu	初始版本

## 目录

### Rockchip Gstreamer用户指南

1. 源码及编译
  - 1.1 源码路径
  - 1.2 编译
2. 常用命令
3. 常用插件
  - 3.1 Source
  - 3.2 Sink
4. 命令示例
5. AFBC
6. 字幕
7. 图层指定
8. FAQ

# 1. 源码及编译

## 1.1 源码路径

Gstreamer及相关插件的源码均通过网络下载，再打上我们提供的补丁的方式生成，具体可以查看

`<SDK>/buildroot/package/gstreamer1/`。

rkmpv插件源码在 `<SDK>/external/gstreamer-rockchip`。

## 1.2 编译

**buildroot:**

开启相关宏（默认开启），直接在SDK根目录编译即可。支持选择编译版本，如

`BR2_PACKAGE_GSTREAMER1_14` 和 `BR2_PACKAGE_GSTREAMER1_18`。

```
BR2_PACKAGE_MPP=y
BR2_PACKAGE_MPP_ALLOCATOR_DRM=y
BR2_PACKAGE_GSTREAMER1_ROCKCHIP=y
BR2_PACKAGE_LINUX_RGA=y
BR2_PACKAGE_CA_CERTIFICATES=y
BR2_PACKAGE_LIBSOUP_SSL=y
BR2_PACKAGE_GSTREAMER1=y
BR2_PACKAGE_GST1_PLUGINS_BASE=y
BR2_PACKAGE_GST1_PLUGINS_BASE_PLUGIN_ALSA=y
BR2_PACKAGE_GST1_PLUGINS_BASE_PLUGIN_VIDEOCONVERT=y
BR2_PACKAGE_GST1_PLUGINS_BASE_PLUGIN_VIDEOTESTSRC=y
BR2_PACKAGE_GST1_PLUGINS_GOOD=y
BR2_PACKAGE_GST1_PLUGINS_GOOD_PLUGIN_AUDIOPARSERS=y
BR2_PACKAGE_GST1_PLUGINS_GOOD_PLUGIN_AUTODETECT=y
BR2_PACKAGE_GST1_PLUGINS_GOOD_PLUGIN_DEINTERLACE=y
BR2_PACKAGE_GST1_PLUGINS_GOOD_PLUGIN_FLV=y
BR2_PACKAGE_GST1_PLUGINS_GOOD_PLUGIN_GDKPIXBUF=y
BR2_PACKAGE_GST1_PLUGINS_GOOD_PLUGIN_MATROSKA=y
BR2_PACKAGE_GST1_PLUGINS_GOOD_PLUGIN_MPG123=y
BR2_PACKAGE_GST1_PLUGINS_GOOD_PLUGIN_SOUPHTTSPSRC=y
BR2_PACKAGE_GST1_PLUGINS_BAD=y
BR2_PACKAGE_GST1_PLUGINS_BAD_PLUGIN_DVBSUBOVERLAY=y
BR2_PACKAGE_GST1_PLUGINS_BAD_PLUGIN_DVDSPU=y
BR2_PACKAGE_GST1_PLUGINS_BAD_PLUGIN_JPEGFORMAT=y
BR2_PACKAGE_GST1_PLUGINS_BAD_PLUGIN_KMS=y
BR2_PACKAGE_GST1_PLUGINS_BAD_PLUGIN_MPEGDEMUX=y
BR2_PACKAGE_GST1_PLUGINS_BAD_PLUGIN_MPEG2ENC=y
BR2_PACKAGE_GST1_PLUGINS_BAD_PLUGIN_VIDEOPARSERS=y
BR2_PACKAGE_GST1_PLUGINS_BAD_PLUGIN_ADPCMDEC=y
BR2_PACKAGE_GST1_PLUGINS_BAD_PLUGIN_ADPCMENC=y
BR2_PACKAGE_GST1_PLUGINS_BAD_PLUGIN_FAAD=y
BR2_PACKAGE_GST1_PLUGINS_UGLY=y
BR2_PACKAGE_GST1_PLUGINS_UGLY_PLUGIN_ASFDEMUX=y
BR2_PACKAGE_GST1_PLUGINS_UGLY_PLUGIN_DVDLPCMDEC=y
```

```
BR2_PACKAGE_GST1_PLUGINS_UGLY_PLUGIN_DVDSUB=y
BR2_PACKAGE_GST1_PLUGINS_UGLY_PLUGIN_MPEG2DEC=y
...
```

完整插件列表可进入menuconfig->Target packages->Audio and video applications->gststreamer 1.x查看。

### debian:

需要将源码放至板端，并确认源码根目录下存在 `debian` 目录。进入源码根目录，执行：

```
# 1 更新软件源
apt update
# 2 安装依赖库
apt build-dep .
# 3 可选：开始编译deb安装包
dpkg-buildpackage -b -s -uc -us
# 编译完成后会在上一级目录生成deb安装包，使用dpkg -i xxx.deb即可安装。
# 3 可选：编译并安装
meson build && ninja -C build install
```

通常建议使用第一种即编译deb安装包的方式，可以保证编译，安装等选项统一。

注意：某些编译选项依赖于 `video-format.h` 等头文件内的宏定义，因此需要先安装 `libgststreamer-plugins-base1.0-dev` 包，保证 `video-format.h` 等头文件最新，从而保证某些功能开启。

## 2. 常用命令

- `gst-launch-1.0`

Gstreamer启动器，用于快速构建pipeline，示例如下：

```
# 使用videotestsrc生成一段视频，并使用xvimagesink显示
gst-launch-1.0 videotestsrc ! xvimagesink
```

- `gst-play-1.0`

Gstreamer播放器，用于播放各种流媒体，示例如下：

```
# 播放test.mp4，并通过xvimagesink显示
gst-play-1.0 test.mp4 --videosink=xvimagesink
# 常用命令选项
--flags          # bit0:视频, bit1:音频, bit2:字幕, 如--flags=1表示只播放视频
--videosink      # 指定videosink
--audiosink      # 指定audiosink
--use-playbin3   # 使用playbin3, 否则使用playbin2
```

- `gst-inspect-1.0`

查找器，用于列出所有插件或某一插件的具体信息，示例如下：

```
# 不带任何参数，列出所有插件
gst-inspect-1.0
# 列出xvimagesink插件的所有信息
gst-inspect-1.0 xvimagesink
```

- 开启日志功能

```
设置环境变量
export GST_DEBUG=2
或在命令前指定，命令结束即失效
GST_DEBUG=2 gst-play-1.0 ...

指定不同模块不同日志等级，支持通配符，fpsdisplaysink指定为DEBUG（5），xvimage*指定为
FIXME（3），其他指定为WARNING（2）
GST_DEBUG=2,fpsdisplaysink:5,xvimage*:3
```

日志等级分为ERROR(1)，WARNING(2)，FIXME(3)，INFO(4)，DEBUG(5)，LOG(6)，TRACE(7)等。

## 3. 常用插件

### 3.1 Source

指可以产生数据但不能接收数据的插件。

- filesrc

从文件读取数据，示例如下：

```
gst-launch-1.0 filesrc location=/tmp/test ! filesink location=/tmp/test2
```

- videotestsrc

生成视频数据，示例如下：

```
# 使用默认格式输出视频
gst-launch-1.0 videotestsrc ! xvimagesink
# 使用指定格式输出视频
gst-launch-1.0 videotestsrc ! "video/x-raw,width=1920,height=1080,format=
(string)NV12" ! xvimagesink
```

### 3.2 Sink

指可以接受数据但不会发送数据的插件。

- filesink

将收到的数据保存为文件，示例如下：

```
gst-launch-1.0 filesrc location=/tmp/test ! filesink location=/tmp/test2
```

- fakesink

将收到的数据全部丢弃，示例如下：

```
gst-launch-1.0 filesrc location=/tmp/test ! fakesink
```

- xvimagesink

视频Sink，接收视频并显示，使用X11接口实现，示例如下：

```
gst-launch-1.0 videotestsrc ! xvimagesink
```

- kmssink

视频Sink，接收视频并显示，使用kms接口实现，需要独占硬解图层，示例如下：

```
gst-launch-1.0 videotestsrc ! kmssink
# 常用命令
connector-id      指定屏幕
plane-id          指定硬件图层
render-rectangle  指定渲染范围
```

- waylandsink

视频Sink，接收视频并显示，使用wayland接口实现，示例如下：

```
gst-launch-1.0 videotestsrc ! waylandsink
```

- rkximagesink

视频Sink，接收视频并显示，使用drm接口实现零拷贝等功能，性能较好，但需要独占硬解图层。示例如下：

```
gst-launch-1.0 videotestsrc ! rkximagesink
```

- fpsdisplaysink

视频Sink，接收视频并统计帧率，同时会将视频中转至下一级Sink显示，示例如下：

```
gst-launch-1.0 videotestsrc ! fpsdisplaysink video-sink=xvimagesink
```

## 4. 命令示例

---

```
GST_DEBUG=fpsdisplaysink:6 gst-play-1.0 --flags=3 --videosink="fpsdisplaysink
video-sink=xvimagesink signal-fps-measurements=true text-overlay=false
sync=false" --audiosink="alsasink device=hw:0,0" test.mp4
```

`GST_DEBUG=fpsdisplaysink:6` 设置Gstreamer日志等级，将fpsdisplaysink指定为6，其余模块关闭日志

<code>--flags=3</code>	关闭字幕
<code>--videosink="fpsdisplaysink ..."</code>	指定fpsdisplaysink用于帧率统计
<code>video-sink=xvimagesink</code>	指定xvimagesink作为最终显示Sink
<code>signal-fps-measurements=true</code>	开启FPS统计
<code>text-overlay=false</code>	关闭帧率显示，true则会在画面上叠加帧率信息
<code>sync=false</code>	关闭时钟同步
<code>--audiosink="alsasink ..."</code>	指定alsasink为音频Sink
<code>device=hw:0,0</code>	指定声卡hw:0,0

## 5. AFBC

AFBC全称ARM Frame Buffer Compression，是一种压缩格式，用于节省带宽。目前mppvideodec插件支持AFBC的编码格式有：H264，H265，VP9，支持的色彩格式有NV12，NV12 10bit。开启方法如下：

```
export GST_MPP_VIDEODEC_DEFAULT_ARM_AFBC=1
```

# AFBC需要使用Cluster图层播放

# 或使用waylandsink，可以使用GPU合成到Esmart/Smart图层

# GST\_DEBUG=\*mpp\*:4开启mpp插件DEBUG开关，可以通过rkmppt打印的日志判断是否成功开启AFBC，如未打印AFBC可能是未成功开启或格式不支持压缩

```
GST_DEBUG=*mpp*:4,fpsdisplaysink:6 gst-play-1.0 --flags=3 --
videosink="fpsdisplaysink video-sink=waylandsink text-overlay=false signal-fps-
measurements=true" test.mp4
GST_DEBUG=*mpp*:4,fpsdisplaysink:6 gst-play-1.0 --flags=3 --
videosink="fpsdisplaysink video-sink=\"kmssink plane-id=101\" text-overlay=false
signal-fps-measurements=true" test.mp4
```

## 6. 字幕

开启字幕会出现卡顿，通常字幕合成需要从视频中截取部分图像并转为RGB，再合成字幕后再转回源格式，才能进行送显，即解码的耗时还需考虑字幕合成的耗时，导致整体帧率下降。使用gst-play-1.0命令测试可以通过 `--flags=3` 关闭字幕。字幕需要自行使用QT等框架独立于视频层实现。

## 7. 图层指定

使用rkximagesink或kmssink时，需要独占一个硬件图层，并且插件会自动寻找图层播放，但自动寻找的图层可能无法满足需求，因此需要手动指定图层，方法如下：



```
gst-play-1.0 --flags=3 test.mp4 --videosink="kmssink plane-id=117"
```

其中117即目标图层的ID，可通过 `/sys/kernel/debug/dri/0/state` 节点确认，可以使用如下命令列出所有图层：

```
root@linaro-alip:/# cat /sys/kernel/debug/dri/0/state | grep "plane\[\"
plane[57]: Smart1-win0
plane[71]: Cluster1-win0
plane[87]: Smart0-win0
plane[101]: Cluster0-win0
plane[117]: Esmart1-win0
plane[131]: Esmart0-win0
# 也可以使用cat /sys/kernel/debug/dri/0/state列出完整信息
```

其中plane[xx]即为plane-id。通常不同图层支持的格式不同，如Cluster支持AFBC，但Esmart不支持AFBC，具体可查阅datasheet或TRM了解。

## 8. FAQ

### 1. 播放4K 30FPS不会卡顿，播放4K 60FPS出现卡顿

由于系统负载、DDR带宽等问题，有可能导致无法达到4K 60FPS，可以尝试开启AFBC，参考[AFBC](#)章节。另外可以关闭字幕和sink的同步功能，如 `gst-play-1.0 test.mp4 --flags=3 --videosink="waylandsink sync=false"`，在帧率无法达到60FPS时，开启sync会由于视频帧时间戳无法对齐时钟从而出现明显丢帧。

### 2. 播放某些片源比较卡顿，CPU占用率很高

目前硬解支持H264，H265，VP8，VP9，MPEG。可以通过 `echo 0x100 > /sys/module/rk_vcodec/parameters/mpp_dev_debug` 开启DEBUG，看串口或dmesg有没有出现解码打印。如果没有可能是硬解不支持的格式。

### 3. 某些片源无法播放，LOG卡住未打印进度或进度始终为0

可以尝试使用playbin3，如 `gst-play-1.0 --flags=3 --use-playbin3 test.mp4`。

### 4. 开启AFBC后播放4K视频时出现闪烁

首先确认开启性能模式，`echo performance | tee $(find /sys/ -name *governor)`。另外确认在纵向上是否有明显缩放，如使用竖屏播放横屏画面，在这种情况下AFBC性能没有非AFBC性能好。

### 5. 播放有画面但没有声音

可以手动指定下audiosink，如 `gst-play-1.0 --flags=3 test.mp4 --audiosink="alsasink device=hw:0,0"`。建议先使用aplay等基础测试工具测试可用再使用gststreamer测试。