Rockchip Gstreamer用户指南

文件标识：RK-YH-YF-921

发布版本：V1.2.1

日期：2024-01-16

文件密级：□绝密 □秘密 □内部资料 ■公开

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

本文档主要介绍 Gstreamer及相关插件的编译和测试方法。

支持的系统和版本

系统	版本
Buildroot	1.22.x
Debian10 (Buster)	1.14.4
Debian11 (Bullseye)	1.18.5
Yocto	1.20.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
2022-02-24	V1.0.1	Jair Wu	修复错误的命令选项
2022-05-10	V1.1.0	Jair Wu	新增MPP插件和环境变量说明，增加命令示例
2022-07-26	V1.1.1	LGZ	新增Gstreamer简介及AFBC dump解码数据
2022-10-25	V1.1.2	Jair Wu	新增v4l2src插件说明
2023-10-16	V1.2.0	Jair Wu	更新插件说明，更新FAQ
2024-01-16	V1.2.1	Jair Wu	新增管道分析方法

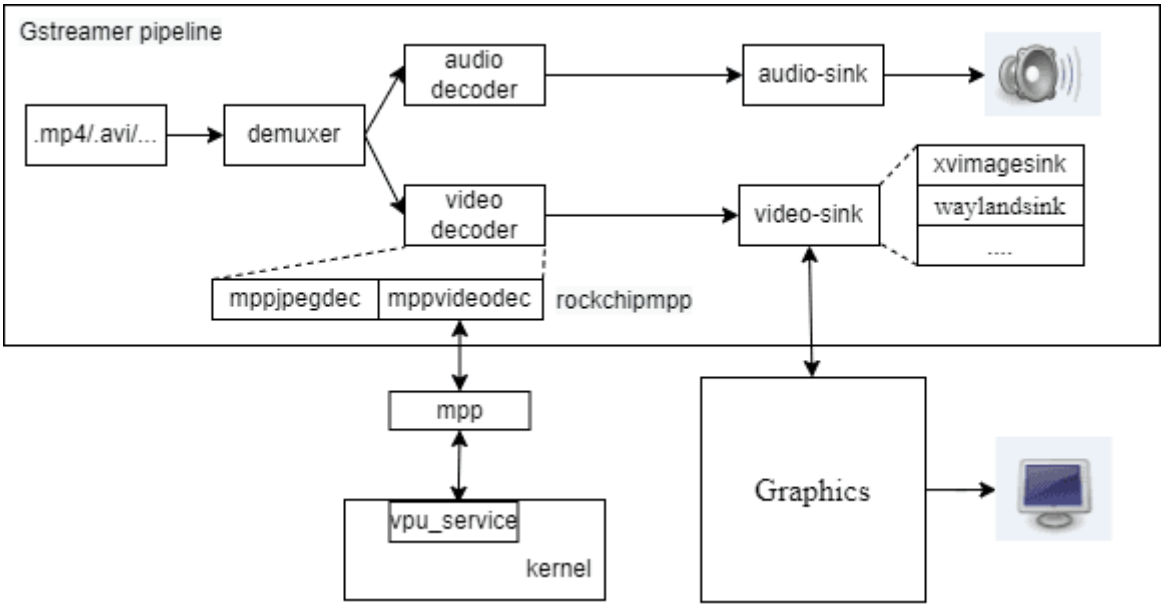
目录

1. GStreamer 简介

GStreamer是一个开源多媒体框架，目前Linux SDK（除了IPC外）的多媒体都主要用GStreamer来对接app 和 编解码组件。利用GStreamer插件的强大特性，通过编写的GStreamer插件适配Rockchip硬件，使得app能够使用硬件编解码进行加速。

1.1 GStreamer 视频编解码适配方案

如下图所示，以视频播放为例，说明下Rockchip平台视频编解码和显示基本流程：



视频文件（如mp4）先经过demuxer解封装为视频(如h264,h265编码)和音频流，视频流经过解码器video decoder（如mppvideodec和mppjpegdec）解码；音频流经过音频解码器audio decoder解码，最后通过显示插件（如xvimagesink，waylandsink等）将解码后的视频数据送显，通过音频播放插件（alsasink等）将解码后的音频数据送入声卡播放声音。

Rockchip平台针对视频编解码实现硬件加速，通过插件rockchipmpp实现，其中包括解码插件：mppvideodec和mppjpegdec；编码插件：mpph264enc，mppvp8enc，mppjpegenc等。GStreamer在视频解码阶段会优先调用rockchipmpp插件，大致流程如下：mppvideodec等插件调用MPP提供的接口，MPP是Rockchip平台的视频编解码中间件会调用vpu驱动（vpu_service）。硬件编解码功能也可直接通过MPP提供测试接口进行测试 (比如mpi_dec_test\mpi_enc_test...)。

解码后的视频数据经过显示插件（如xvimagesink，waylandsink等）送入显示设备进行显示，不同显示插件调用不同显示架构接口以对接不同的显示架构，如xvimagesink会调用X11接口对接X11显示架构，waylandsink调用Wayland接口对接Wayland显示架构等。

显示相关具体参考`SDK文档Rockchip_Developer_Guide_Linux_Graphics_CN.pdf`

MPP源码参考`<SDK>/external/mpp/`，MPP相关说明文档参考`<SDK>/docs/Linux/Multimedia/Rockchip_Developer_Guide_MPP_CN.pdf`

测试demo参考:`<SDK>/external/mpp/test`

2. 源码及编译

2.1 源码路径

Buildroot：

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/`。

Debian：

Debian版本源码可通过Debian仓库查找下载，并在对应的版本打上补丁，补丁可通过Redmine等渠道向开发者获取，目前已有的补丁为Debian10对应的1.14.4和Debian11对应的1.18.5。

Gstreamer-rockchip：

MPP编解码插件及rkximagesink显示插件源码在`<SDK>/external/gstreamer-rockchip`，Buildroot与Debian共用同一仓库。

2.2 编译

Buildroot：

开启相关宏（默认开启），直接在SDK根目录编译即可，相关宏均统一整理至`<SDK>/buildroot/configs/rockchip/multimedia/gst/`，在目标config里直接包含即可。

```
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_SOUPHTTPSRC=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->gstreamer 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 -d -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`libgstreamer-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。部分插件的编译依赖于系统环境，如发现缺少插件，可检查编译脚本和日志，安装依赖库后重新编译，并确保在debian/*.install文件中有包含目标库。

3. 基本命令

- 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
```

- gst-discoverer-1.0

  分析命令，可以针对提供的uri进行格式分析，如果缺少相应插件该命令会打印相应报错，示例如下：

```
# 分析本地文件
gst-discoverer-1.0 test.mp4
# 分析rtsp码流
gst-discoverer-1.0 rtsp://127.0.0.1:8554/
```

- 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。

4. 插件介绍

本章节主要对由Rockchip开发或比较常用的插件进行介绍和说明，完整的插件说明建议查阅官方在线文档。

Gstreamer的插件主要分为三大类：源插件（Source）、过滤器/类过滤器插件（Filter）、接收插件（Sink）。

源插件只产生数据，但不接收数据，比如filesrc插件，用于读取文件，videotestsrc插件，用于生成指定的图像等。

过滤器/类过滤器插件接收数据，对数据进行一些处理，再发送给后级，比如一些解封装器，编解码器等插件。

接收插件只接收数据，但不产生数据，比如filesink，用于保存文件，waylandsink，用于渲染画面等。

Gstreamer将核心库和插件库分开进行管理（在最新的GitLab中已经将核心库和插件库合并到同一仓库中作为subprojects管理），核心库几乎不进行多媒体处理，仅提供一些基础类，基础插件的定义，多媒体处理都是由各个插件实现。

插件库分为gst-plugins-base，gst-plugins-bad，gst-plugins-good，gst-plugins-ugly等仓库。

以下会针对一些常用的插件、插件属性、典型管道示例进行说明，完整的插件列表和插件说明可以直接通过gst-inspect-1.0命令查看。

4.1 gstreamer-rockchip

源码路径为`<SDK>/external/gstreamer-rockchip`。

4.1.1 kmssrc

用于从指定对象获取图像数据。

源码路径`<SDK>/external/gstreamer-rockchip/gst/kmssrc`。

命令示例

```
# 按默认的参数取流显示
gst-launch-1.0 kmssrc ! waylandsink
# 指定帧率取流显示
gst-launch-1.0 kmssrc sync-fb=0 ! 'video/x-raw,framerate=10/1' ! fpsdisplaysink
video-sink=fakesink
# 取流编码实现录屏功能
gst-launch-1.0 kmssrc sync-fb=0 ! mpph264enc ! h264parse ! filesink
location=/tmp/out.h264
```

属性说明

- connector-id

  从指定的connector取流，可通过modetest -c查看ID

```
connector-id        : DRM connector ID (0 = unspecified)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- crtc-id

  从指定的crtc取流，可通过modetest -p查看ID

```
crtc-id             : DRM crtc ID (0 = unspecified)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- encoder-id

  从指定的encoder取流，可通过modetest -e查看ID

```
encoder-id          : DRM encoder ID (0 = unspecified)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- fb-id

  从指定的framebuffer取流，可通过modetest -p查看ID

```
fb-id               : DRM FB ID (0 = unspecified)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- plane-id

  从指定的plane取流，可通过modetest -p查看ID

```
plane-id            : DRM plane ID (0 = unspecified)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- dma-feature

  输出DMA buffer

```
dma-feature         : Enable GST DMA feature
                      flags: readable, writable
                      Boolean. Default: false
```

- framerate-limit

  限制最大帧率

```
framerate-limit     : Limited framerate
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 120
```

- num-buffers

  输出指定数量的帧，输出该数量的帧后生成EOS信号

```
num-buffers         : Number of buffers to output before sending EOS (-1 =
unlimited)
                      flags: readable, writable
                      Integer. Range: -1 - 2147483647 Default: -1
```

- sync-fb

  和FB flip信号同步

```
sync-fb             : Sync with FB flip
                      flags: readable, writable
                      Boolean. Default: true
```

- sync-vblank

  和vblank信号同步

```
sync-vblank         : Sync with vblank
                      flags: readable, writable
                      Boolean. Default: true
```

4.1.2 mppvideodec

调用MPP接口进行解码，可以支持H263，H264，H265，AV1，VP8、VP9、MPEG等，具体解码能力不同平台存在差异，可查看相关Datasheet了解。

源码路径`<SDK>/external/gstreamer-rockchip/gst/rockchipmpp/`。

命令示例

```
# 由Gstreamer自动查找解码器，由于mppvideodec优先级最高，只要是该插件支持的格式，最终就会选择该插件解码
gst-play-1.0 test.mp4
# 手动构建pipeline
gst-launch-1.0 filesrc location=test.mp4 ! parsebin ! mppvideodec ! waylandsink
```

属性说明

- arm-afbc

  开启AFBC功能，输出的数据经过压缩，可减少带宽占用，可由其他支持AFBC解码功能的模块（如VOP）进行处理

```
arm-afbc            : Prefer ARM AFBC compressed format
                      flags: readable, writable
                      Boolean. Default: false
```

- crop-rectangle

  指定裁剪范围

```
crop-rectangle      : The crop rectangle ('<x, y, width, height>')
                      flags: writable
                      Default: "<  >"
                      GstValueArray of GValues of type "gint" Write only
```

- dma-feature

  输入DMA buffer

```
dma-feature         : Enable GST DMA feature
                      flags: readable, writable
                      Boolean. Default: false
```

- fast-mode

  开启MPP的FAST-MODE，部分平台和格式可以有效提升性能

```
fast-mode           : Enable MPP fast decode mode
                      flags: readable, writable
                      Boolean. Default: true
```

- ignore-error

  忽略MPP框架上报的错误

```
ignore-error        : Ignore MPP decode errors
                      flags: readable, writable
                      Boolean. Default: true
```

- format

  指定输出格式，由RGA进行转码，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
format              : Prefered output format
                      flags: readable, writable
                      Enum "GstMppVideoDecFormat" Default: 0, "auto"
                         (0): auto             - Auto
                         (23): NV12             - NV12
                         (24): NV21             - NV21
                         (2): I420             - I420
                         (3): YV12             - YV12
                         (51): NV16             - NV16
                         (60): NV61             - NV61
                         (30): BGR16            - BGR565
                         (15): RGB              - RGB
                         (16): BGR              - BGR
                         (11): RGBA             - RGBA8888
                         (12): BGRA             - BGRA8888
                         (7): RGBx             - RGBX8888
                         (8): BGRx             - BGRX8888
```

- width

  指定输出宽度，由RGA进行缩放，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
width               : Width (0 = original)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- height

指定输出高度，由RGA进行缩放，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
height              : Height (0 = original)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- rotation

  指定输出旋转角度，由RGA进行旋转，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
rotation            : Rotation
                      flags: readable, writable
                      Enum "GstMppDecRotation" Default: 0, "0"
                         (0): 0                - Rotate 0
                         (90): 90              - Rotate 90
                         (180): 180            - Rotate 180
                         (270): 270            - Rotate 270
```

4.1.3 mppjpegdec

调用MPP接口进行JPEG解码，具体解码能力不同平台存在差异，可查看相关Datasheet了解。

源码路径`<SDK>/external/gstreamer-rockchip/gst/rockchipmpp/`。

命令示例

```
gst-launch-1.0 filesrc location=nv12.jpg ! parsebin ! mppjpegdec ! filesink
location=nv12.yuv
```

属性说明

- crop-rectangle

  指定裁剪范围

```
crop-rectangle      : The crop rectangle ('<x, y, width, height>')
                      flags: writable
                      Default: "<  >"
                      GstValueArray of GValues of type "gint" Write only
```

- dma-feature

  输入DMA buffer

```
dma-feature         : Enable GST DMA feature
                      flags: readable, writable
                      Boolean. Default: false
```

- fast-mode

  开启MPP的FAST-MODE，部分平台和格式可以有效提升性能

```
fast-mode           : Enable MPP fast decode mode
                      flags: readable, writable
                      Boolean. Default: true
```

- ignore-error

  忽略MPP框架上报的错误

```
ignore-error        : Ignore MPP decode errors
                      flags: readable, writable
                      Boolean. Default: true
```

- format

  指定输出格式，由RGA进行转码，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
format              : Prefered output format
                      flags: readable, writable
                      Enum "GstMppVideoDecFormat" Default: 0, "auto"
                         (0): auto             - Auto
                         (23): NV12             - NV12
                         (24): NV21             - NV21
                         (2): I420             - I420
                         (3): YV12             - YV12
                         (51): NV16             - NV16
                         (60): NV61             - NV61
                         (30): BGR16            - BGR565
                         (15): RGB              - RGB
                         (16): BGR              - BGR
                         (11): RGBA             - RGBA8888
                         (12): BGRA             - BGRA8888
                         (7): RGBx             - RGBX8888
                         (8): BGRx             - BGRX8888
```

- width

  指定输出宽度，由RGA进行缩放，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
width               : Width (0 = original)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- height

  指定输出高度，由RGA进行缩放，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
height              : Height (0 = original)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- rotation

指定输出旋转角度，由RGA进行旋转，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
rotation            : Rotation
                      flags: readable, writable
                      Enum "GstMppDecRotation" Default: 0, "0"
                         (0): 0                - Rotate 0
                         (90): 90              - Rotate 90
                         (180): 180            - Rotate 180
                         (270): 270            - Rotate 270
```

4.1.4 mpph264enc/mpph265enc/mppvp8enc

调用MPP接口进行H264、H265、VP8编码，具体编码能力不同平台存在差异，可查看相关Datasheet了解。

源码路径`<SDK>/external/gstreamer-rockchip/gst/rockchipmpp/`。

命令示例

```
# 将10s时长的640x320@NV12裸流编码H264并封装为MP4文件
gst-launch-1.0 videotestsrc num-buffers=600 ! video/x-
raw,format=NV12,width=640,height=320,framerate=60/1 ! mpph264enc ! h264parse !
qtmux ! filesink location=h264.mp4
# 将10s时长的640x320@NV12裸流编码H265并封装为MP4文件
gst-launch-1.0 videotestsrc num-buffers=600 ! video/x-
raw,format=NV12,width=640,height=320,framerate=60/1 ! mpph265enc ! h265parse !
qtmux ! filesink location=h265.mp4
# 将10s时长的640x320@NV12裸流编码VP8并封装为MP4文件
gst-launch-1.0 videotestsrc num-buffers=600 ! video/x-
raw,format=NV12,width=640,height=320,framerate=60/1 ! mppvp8enc ! qtmux !
filesink location=vp8.mp4
```

属性说明

- arm-afbc

  如果前级输入是经过AFBC压缩的数据，则需要开启该选项

```
arm-afbc            : Input is ARM AFBC compressed format
                      flags: readable, writable
                      Boolean. Default: false
```

- bps

  编码后码率

```
bps                 : Target BPS (0 = auto calculate)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- bps-max

  编码后码率最大值

```
bps-max             : Max BPS (0 = auto calculate)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- bps-min

  编码后码率最小值

```
bps-min             : Min BPS (0 = auto calculate)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- gop

  Group of pictures，即两个I帧之间的距离

```
gop                 : Group of pictures starting with I frame (-1 = FPS, 1 = all
I frames)
                      flags: readable, writable
                      Integer. Range: -1 - 2147483647 Default: -1
```

- qp-init

  影响编码质量和编码后码率

```
qp-init             : Initial QP (lower value means higher quality)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 51 Default: 26
```

- qp-max

```
qp-max              : Max QP (0 = default)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 51 Default: 0
```

- qp-max-i

```
qp-max-i            : Max Intra QP (0 = default)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 51 Default: 0
```

- qp-min

```
qp-min              : Min QP (0 = default)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 51 Default: 0
```

- qp-min-i

```
qp-min-i            : Min Intra QP (0 = default)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 51 Default: 0
```

- rc-mode

  码率控制模式

```
rc-mode             : RC mode
                      flags: readable, writable
                      Enum "GstMppEncRcMode" Default: 1, "cbr"
                         (0): vbr              - Variable bitrate
                         (1): cbr              - Constant bitrate
                         (2): fixqp            - Fixed QP
```

- width

  输入图像宽度，与实际宽度不符合时会使用RGA缩放，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
width               : Width (0 = original)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- height

  输入图像高度，与实际高度不符合时会使用RGA缩放，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
height              : Height (0 = original)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- rotation

  旋转输入图像，使用RGA旋转，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
rotation            : Rotation
                      flags: readable, writable
                      Enum "GstMppEncRotation" Default: 0, "0"
                         (0): 0                - Rotate 0
                         (90): 90               - Rotate 90
                         (180): 180             - Rotate 180
                         (270): 270             - Rotate 270
```

4.1.5 mppjpegenc

调用MPP接口进行JPEG编码，具体编码能力不同平台存在差异，可查看相关Datasheet了解。

源码路径`<SDK>/external/gstreamer-rockchip/gst/rockchipmpp/`。

命令示例

```
# 将640x320@NV12裸流编码为JPEG文件
gst-launch-1.0 videotestsrc num-buffers=1 ! video/x-
raw,format=NV12,width=640,height=320,framerate=60/1 ! mppjpegenc ! filesink
location=nv12.jpg
```

属性说明

- arm-afbc

  如果前级输入是经过AFBC压缩的数据，则需要开启该选项

```
arm-afbc            : Input is ARM AFBC compressed format
                      flags: readable, writable
                      Boolean. Default: false
```

- bps

  编码后码率

```
bps                 : Target BPS (0 = auto calculate)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- bps-max

  编码后码率最大值

```
bps-max             : Max BPS (0 = auto calculate)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- bps-min

  编码后码率最小值

```
bps-min             : Min BPS (0 = auto calculate)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- q-factor

  品质因素，影响编码质量和编码后码率

```
q-factor            : Quality Factor
                      flags: readable, writable
                      Unsigned Integer. Range: 1 - 99 Default: 80
```

- qf-max

```
qf-max              : Max Quality Factor
                      flags: readable, writable
                      Unsigned Integer. Range: 1 - 99 Default: 99
```

- qf-min

```
qf-min              : Min Quality Factor
                      flags: readable, writable
                      Unsigned Integer. Range: 1 - 99 Default: 1
```

- rc-mode

  码率控制模式

```
rc-mode             : RC mode
                      flags: readable, writable
                      Enum "GstMppEncRcMode" Default: 1, "cbr"
                         (0): vbr              - Variable bitrate
                         (1): cbr              - Constant bitrate
                         (2): fixqp            - Fixed QP
```

- width

  输入图像宽度，与实际宽度不符合时会使用RGA缩放，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
width               : Width (0 = original)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- height

  输入图像高度，与实际高度不符合时会使用RGA缩放，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
height              : Height (0 = original)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 2147483647 Default: 0
```

- rotation

  旋转输入图像，使用RGA旋转，需要开启BR2_PREFER_ROCKCHIP_RGA，并保证运行时GST_MPP_NO_RGA环境变量不存在

```
rotation            : Rotation
                      flags: readable, writable
                      Enum "GstMppEncRotation" Default: 0, "0"
                         (0): 0                - Rotate 0
                         (90): 90               - Rotate 90
                         (180): 180             - Rotate 180
                         (270): 270             - Rotate 270
```

4.1.6 rkximagesink

调用X接口绘制窗体，调用DRM接口直接送显，主要用于零拷贝送显，需独占硬件图层。

源码路径`<SDK>/external/gstreamer-rockchip/gst/rkximage/`。

命令示例

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

属性说明

- connector-id

  指定connector渲染

```
connector-id        : DRM connector id
                      flags: 可读, 可写
                      Integer. Range: -1 - 2147483647 Default: -1
```

- plane-id

  指定plane渲染

```
plane-id            : DRM plane id
                      flags: 可读, 可写
                      Integer. Range: -1 - 2147483647 Default: -1
```

- sync

  是否与时钟同步

```
sync                : Sync on the clock
                      flags: 可读, 可写
                      Boolean. Default: true
```

4.2 core elements

4.2.1 fakesink

将收到的数据全部丢弃，比如关闭音频等操作。

命令示例

```
gst-launch-1.0 filesrc location=/tmp/test ! fakesink
gst-play-1.0 /oem/SampleVideo_1280x720_5mb.mp4 --audiosink=fakesink
```

4.2.2 filesrc

从文件读取数据。

命令示例

```
gst-launch-1.0 filesrc location=/tmp/test ! filesink location=/tmp/test2
gst-launch-1.0 filesrc location=nv12_640x320.yuv blocksize=307200 ! video/x-
raw,format=NV12,width=640,height=320 ! mpph264enc ! filesink location=out.h264
```

属性说明

- blocksize

  帧大小，当读取YUV裸流文件编码时比较有用，默认的4096通常小于一帧的大小，编码插件会报错

```
blocksize           : Size in bytes to read per buffer (-1 = default)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 4294967295 Default: 4096
```

- location

  指定文件路径，支持绝对或相对地址

```
location            : Location of the file to read
                      flags: readable, writable, changeable only in NULL or READY
state
                      String. Default: null
```

- num-buffers

  指定读取固定帧数的数据

```
num-buffers         : Number of buffers to output before sending EOS (-1 =
unlimited)
                      flags: readable, writable
                      Integer. Range: -1 - 2147483647 Default: -1
```

4.2.3 filesink

将收到的数据保存为文件。

命令示例

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

属性说明

- blocksize

  帧大小

```
blocksize           : Size in bytes to pull per buffer (0 = default)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 4294967295 Default: 4096
```

- location

  指定文件保存位置

```
location            : Location of the file to write
                      flags: readable, writable
                      String. Default: null
```

4.3 gst1-plugins-base

4.3.1 appsrc

用于从外部导入数据，比如有私有的加密包，需要外部应用解密后再导入gstreamer处理等类似情况，该插件通常只在编写代码时使用，在命令行中使用意义不大，具体使用可以参考`gst-plugins-base-<version>/tests/examples/app/`中的示例。

4.3.2 appsink

用于导出gstreamer数据，比如获取解码后数据，然后进行算法处理等。该插件通常只在编写代码时使用，在命令行中使用意义不大，具体使用可以参考`gst-plugins-base-<version>/tests/examples/app/`中的示例。该插件通常可以配合appsrc使用，使用appsink从pipeline中取出数据，经过处理后，再使用appsrc送入pipeline进行后续流程。

4.3.3 decodebin/decodebin3

该插件会自动查找合适的解码插件，当前级数据存在多条轨道时，该插件也会同步生成多条解码路径，如音轨，视轨，字幕等

命令示例

```
gst-launch-1.0 filesrc location=/oem/SampleVideo_1280x720_5mb.mp4 ! decodebin !
waylandsink
gst-launch-1.0 filesrc location=/oem/SampleVideo_1280x720_5mb.mp4 ! decodebin
name=d d. ! waylandsink d. ! pulsesink
```

4.3.4 playbin/playbin3

该插件集成了decodebin，并且会自动查找合适的sink插件

命令示例

```
# 自动查找合适的sink插件
gst-launch-1.0 playbin uri=file:///oem/SampleVideo_1280x720_5mb.mp4
# 手动指定sink插件，并且只开启音轨和视轨（flags=3）
gst-launch-1.0 playbin uri=file:///oem/SampleVideo_1280x720_5mb.mp4
videosink=waylandsink audiosink=pulsesink flags=3
```

属性说明

- audio-sink

  指定audio sink，如alsasink，pulsesink等

```
audio-sink          : the audio output element to use (NULL = default sink)
                      flags: readable, writable
                      Object of type &amp;quot;GstElement&amp;quot;
```

- flags

  插件行为控制，如只渲染视频，只渲染视频和音频等

```
flags              : Flags to control behaviour
                     flags: readable, writable
                     Flags "GstPlayFlags" Default: 0x00000617, "soft-
colorbalance+deinterlace+soft-volume+text+audio+video"
                        (0x00000001): video            - Render the video stream
                        (0x00000002): audio            - Render the audio stream
                        (0x00000004): text             - Render subtitles
                        (0x00000008): vis              - Render visualisation
when no video is present
                        (0x00000010): soft-volume      - Use software volume
                        (0x00000020): native-audio     - Only use native audio
formats
                        (0x00000040): native-video     - Only use native video
formats
                        (0x00000080): download         - Attempt progressive
download buffering
                        (0x00000100): buffering        - Buffer demuxed/parsed
data
                        (0x00000200): deinterlace      - Deinterlace video if
necessary
                        (0x00000400): soft-colorbalance - Use software color
balance
                        (0x00000800): force-filters    - Force audio/video
filter(s) to be applied
                        (0x00001000): force-sw-decoders - Force only software-
based decoders (no effect for playbin3)
```

- video-sink

  指定video sink，如waylandsink，kmssink等

```
video-sink         : the video output element to use (NULL = default sink)
                     flags: readable, writable
                     Object of type "GstElement"
```

4.3.5 uridecodebin/uridecodebin3

该插件集成了decodebin，并且可以根据提供的uri自动查找合适的source插件。

命令示例

```
gst-launch-1.0 uridecodebin uri=file:///oem/SampleVideo_1280x720_5mb.mp4 !
waylandsink
gst-launch-1.0 uridecodebin uri=file:///oem/SampleVideo_1280x720_5mb.mp4 name=d
d. ! waylandsink d. ! pulsesink
```

属性说明

- uri

指定源uri，支持`file://`，`rtps://`，`http://`等

```
uri                 : URI to decode
                      flags: readable, writable
                      String. Default: null
```

4.3.6 videoconvert

该插件用于图像格式转换，适配了RGA加速，部分平台不带RGA功能，需查阅Datasheet了解。需要开启BR2_PREFER_ROCKCHIP_RGA，并设置环境变量。

命令示例

```
# 在/etc/profile.d/gst.sh取消以下两句的注释，或每次手动执行
export GST_VIDEO_CONVERT_USE_RGA=1
export GST_VIDEO_FLIP_USE_RGA=1
gst-launch-1.0 videotestsrc ! video/x-raw,format=NV12 ! videoconvert ! video/x-
raw,format=BGRA ! waylandsink
```

4.3.7 videoscale

该插件用于图像大小缩放，适配了RGA加速，部分平台不带RGA功能，需查阅Datasheet了解。需要开启BR2_PREFER_ROCKCHIP_RGA，并设置环境变量。

命令示例

```
# 在/etc/profile.d/gst.sh取消以下两句的注释，或每次手动执行
export GST_VIDEO_CONVERT_USE_RGA=1
export GST_VIDEO_FLIP_USE_RGA=1
gst-launch-1.0 videotestsrc ! video/x-raw,width=640,height=320 ! videoscale !
video/x-raw,width=1280,height=720 ! waylandsink
```

4.3.8 videotestsrc

该插件用于生成各种格式和分辨率的视频流。

命令示例

```
  # 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
```

4.3.9 xvimagesink

该插件用于在XV环境下渲染图像，由GPU进行合成。

命令示例

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

4.4 gst1-plugins-good

4.4.1 v4l2src

从摄像头获取视频数据

命令示例

```
gst-launch-1.0 v4l2src ! video/x-raw,width=1920,height=1080,format=NV12 !
waylandsink
```

属性说明

- device

  指定设备节点

```
device              : Device location
                      flags: readable, writable
                      String. Default: "/dev/video-camera0"
```

- min-buffers

  指定驱动最低缓存大小

```
min-buffers         : Override the driver's min buffers (0 means auto)
                      flags: readable, writable
                      Unsigned Integer. Range: 0 - 64 Default: 0
```

- num-buffers

  指定输出固定帧数的数据

```
num-buffers         : Number of buffers to output before sending EOS (-1 =
unlimited)
                      flags: readable, writable
                      Integer. Range: -1 - 2147483647 Default: -1
```

4.4.2 rtspsrc

从RTSP服务器中获取视频流

命令示例

```
gst-launch-1.0 rtspsrc location=rtsp://192.168.1.105:8554/ ! rtph264depay !
h264parse ! mppvideodec ! waylandsink
```

属性说明

- location

  指定码流地址

```
location            : Location of the RTSP url to read
                      flags: readable, writable
                      String. Default: null
```

4.4.3 videoflip

该插件用于图像翻转，适配了RGA加速，部分平台不带RGA功能，需查阅Datasheet了解。需要开启BR2_PREFER_ROCKCHIP_RGA，并设置环境变量。

命令示例

```
# 在/etc/profile.d/gst.sh取消以下两句的注释，或每次手动执行
export GST_VIDEO_CONVERT_USE_RGA=1
export GST_VIDEO_FLIP_USE_RGA=1
gst-launch-1.0 videotestsrc ! videoflip video-direction=90r ! waylandsink
```

属性说明

- video-direction

  视频方向

```
video-direction     : Video direction: rotation and flipping
                      flags: readable, writable, controllable, changeable in
NULL, READY, PAUSED or PLAYING state
                      Enum "GstVideoOrientationMethod" Default: 0, "identity"
                         (0): identity         - GST_VIDEO_ORIENTATION_IDENTITY
                         (1): 90r              - GST_VIDEO_ORIENTATION_90R
                         (2): 180              - GST_VIDEO_ORIENTATION_180
                         (3): 90l              - GST_VIDEO_ORIENTATION_90L
                         (4): horiz            - GST_VIDEO_ORIENTATION_HORIZ
                         (5): vert             - GST_VIDEO_ORIENTATION_VERT
                         (6): ul-lr            - GST_VIDEO_ORIENTATION_UL_LR
                         (7): ur-ll            - GST_VIDEO_ORIENTATION_UR_LL
                         (8): auto             - GST_VIDEO_ORIENTATION_AUTO
                         (9): custom           - GST_VIDEO_ORIENTATION_CUSTOM
```

4.5 gst1-plugins-bad

4.5.1 kmssink

该插件用于图像渲染，使用kms接口实现，需要独占硬解图层。

命令示例

```
gst-launch-1.0 videotestsrc ! kmssink
```

属性说明

- connector-id

  指定connector渲染

```
connector-id        : DRM connector id
                      flags: readable, writable
                      Integer. Range: -1 - 2147483647 Default: -1
```

- fullscreen

  是否全屏显示

```
fullscreen          : Force showing fullscreen
                      flags: readable, writable
                      Boolean. Default: false
```

- hdr-enable

  开启HDR功能

```
hdr-enable          : Enable HDR
                      flags: readable, writable
                      Boolean. Default: true
```

- plane-id

  指定plane渲染

```
plane-id            : DRM plane id
                      flags: readable, writable
                      Integer. Range: -1 - 2147483647 Default: -1
```

- render-rectangle

  指定渲染范围

```
render-rectangle    : The render rectangle ('<x, y, width, height>')
                      flags: writable
                      Default: "<  >"
                      GstValueArray of GValues of type "gint" Write only
```

- sync

是否与时钟同步

```
sync               : Sync on the clock
                     flags: readable, writable
                     Boolean. Default: true
```

- sync-mode

  帧同步模式设置

```
sync-mode          : Prefered frame syncing mode
                     flags: readable, writable
                     Enum "GstKMSSyncMode" Default: 0, "auto"
                        (0): auto             - Sync with page flip or vblank
event
                        (1): flip             - Sync with page flip event
                        (2): vblank           - Sync with vblank event
                        (3): none             - Ignore syncing
```

4.5.2 waylandsink

该插件用于在wayland境下渲染图像，由GPU进行合成。

命令示例

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

属性说明

- fullscreen

  是否全屏显示

```
fullscreen         : Whether the surface should be made fullscreen
                     flags: readable, writable
                     Boolean. Default: false
```

- layer

  指定显示层级

```
layer              : Wayland window layer
                     flags: readable, writable
                     Enum "GstWlWindowLayer" Default: 1, "normal"
                        (0): top              - Top
                        (1): normal           - Normal
                        (2): bottom           - Bottom
```

- render-rectangle

  指定渲染范围

```
render-rectangle    : The render rectangle ('<x, y, width, height>')
                      flags: writable
                      Default: "<  >"
                      GstValueArray of GValues of type "gint" Write only
```

- rotate-method

  设置图像旋转

```
rotate-method       : rotate method
                      flags: readable, writable
                      Enum "GstVideoOrientationMethod" Default: 0, "identity"
                         (0): identity         - GST_VIDEO_ORIENTATION_IDENTITY
                         (1): 90r              - GST_VIDEO_ORIENTATION_90R
                         (2): 180              - GST_VIDEO_ORIENTATION_180
                         (3): 90l              - GST_VIDEO_ORIENTATION_90L
                         (4): horiz            - GST_VIDEO_ORIENTATION_HORIZ
                         (5): vert             - GST_VIDEO_ORIENTATION_VERT
                         (6): ul-lr            - GST_VIDEO_ORIENTATION_UL_LR
                         (7): ur-ll            - GST_VIDEO_ORIENTATION_UR_LL
                         (8): auto             - GST_VIDEO_ORIENTATION_AUTO
                         (9): custom           - GST_VIDEO_ORIENTATION_CUSTOM
```

- sync

  是否与时钟同步

```
sync                : Sync on the clock
                      flags: readable, writable
                      Boolean. Default: true
```

4.5.3 fpsdisplaysink

统计视频帧率，同时会将图像中转至下一级Sink显示。

命令示例

```
  # 日志等级为TRACE(7)即可查看实时帧率，设置为DEBUG(5)则只显示最大/最小帧率
  GST_DEBUG=fpsdisplaysink:7 gst-play-1.0 --flags=3 --videosink="fpsdisplaysink
video-sink=xvimagesink signal-fps-measurements=true text-overlay=false
sync=false"
```

属性说明

- signal-fps-measurements

  是否生成fps-measurements信号，用于打印实时帧率

```
signal-fps-measurements: If the fps-measurements signal should be emitted.
                      flags: readable, writable
                      Boolean. Default: false
```

- sync

是否与时钟同步

```
sync               : Sync on the clock (if the internally used sink doesn't have
this property it will be ignored
                     flags: readable, writable
                     Boolean. Default: true
```

- text-overlay

  是否将帧率叠加至图像

```
text-overlay       : Whether to use text-overlay
                     flags: readable, writable
                     Boolean. Default: true
```

- video-sink

  指定实际的渲染插件

```
video-sink         : Video sink to use (Must only be called on NULL state)
                     flags: readable, writable
                     Object of type "GstElement"
```

5. Rockchip MPP插件

基于MPP的硬件编解码插件。基于Gstreamer原有GstVideoDecoder类和GstVideoEncoder类开发。源码地址`<SDK>/external/gstreamer-rockchip/gst/rockchipmpp`。

解码支持的格式有JPEG，MPEG，VP8，VP9，H264，H265，AV1[1]。

编码支持的格式有JPEG，H264，H265，VP8。

5.1 gstmppdec

源码地址为gstreamer-rockchip/gst/rockchipmpp/，包含插件mppvideodec，mppjpegdec，以下以mppvideodec为例进行说明。

```
gstreamer-rockchip/gst/rockchipmpp/
├── gstmppdec.c
├── gstmppdec.h
├── gstmppjpegdec.c
├── gstmppjpegdec.h
├── gstmppvideodec.c
├── gstmppvideodec.h
……
```

5.1.1 主要函数说明

gst_mpp_dec_start： 创建MPP实例，内存分配器等。

gst_mpp_dec_set_format： 对MPP实例进行初始化，设置编解码类型和格式，设置Fast Mode，Ignore Error等属性。

gst_mpp_dec_handle_frame： 通过get_mpp_packet获取mpp_packet，填充数据后通过send_mpp_packet发至MPP解码。

gst_mpp_dec_loop： 通过poll_mpp_frame获取解码帧，并推送至下一级插件。

gst_mpp_dec_rga_convert： 如在输出buffer前需要进行格式转换，旋转，缩放，裁剪等操作，则会通过RGA[2]完成，再推送至下一级插件。

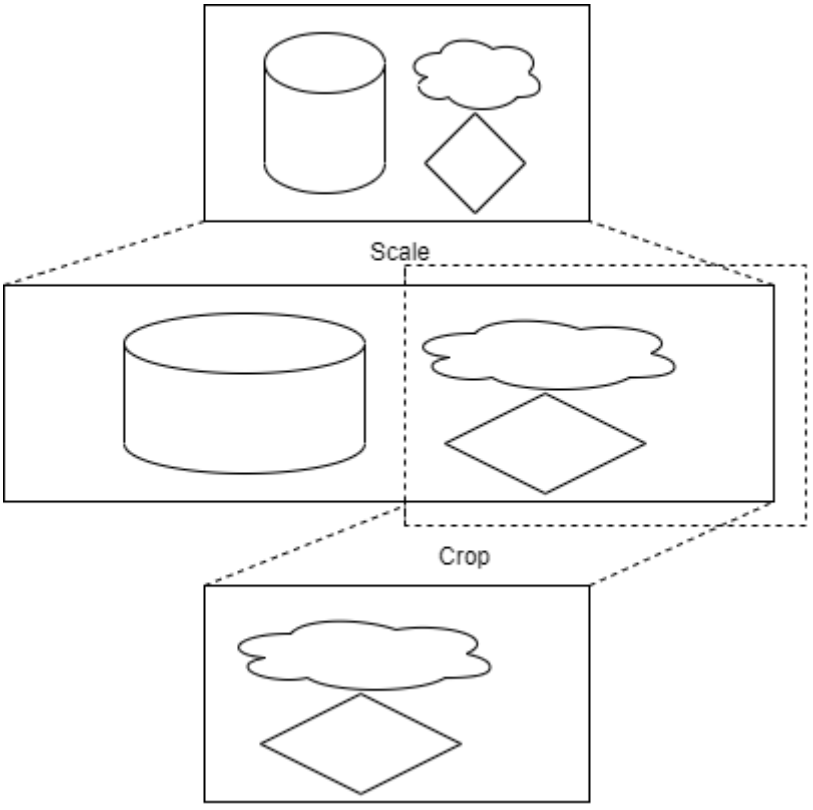
5.1.2 主要属性说明

rotation： 旋转角度，默认为0°，可选0°，90°，180°，270°。

width： 宽度，默认为0，不进行缩放。

height： 高度，默认为0，不进行缩放。

crop-rectangle： 裁剪，使用方式为<x, y, w, h>，即裁剪源<x, y>为起点，宽高为w * h的图像送至下级。需要注意的是，缩放的优先级比裁剪高，因此裁剪参数应以缩放后宽高为标准进行计算，如图所示为指定`crop-rectangle='<1920,0,1920,1080>' width=3840 height=1080`时的处理逻辑：



arm-afbc： AFBC压缩格式，默认不开启，部分平台如RK3399不支持。开启后可以降低DDR带宽占用，部分芯片解码效率会有明显提高。

format： 输出格式，默认为0 "auto"，不进行格式转换。

fast-mode： 开启MPP Fast Mode，如在RK3588平台上可以使部分解码流程并行，提升解码效率。默认开启。

ignore-error： 忽略MPP解码错误，强制输出解码帧。默认开启。

5.2 gstmppenc

源码地址为gstreamer-rockchip/gst/rockchipmpp/，包含插件mpph264enc，mppvp8enc，mppjpegenc等，以下以mpph264enc为例进行说明。

```
gstreamer-rockchip/gst/rockchipmpp/
├── gstmppenc.c
├── gstmppenc.h
├── gstmppjpegenc.c
├── gstmppjpegenc.h
├── gstmpph264enc.c
├── gstmpph264enc.h
……
```

5.2.1 主要函数说明

gst_mpp_enc_start： 创建和初始化MPP实例，设置实例类型和格式。

gst_mpp_enc_apply_properties： 设置编码参数，如gop，bps等。

gst_mpp_enc_handle_frame： 传入上一级插件的输出buffer，并存入编码器缓存中。

gst_mpp_rga_convert： 如需要对输入的buffer进行旋转，缩放等操作，则会先使用RGA[3]完成操作，再存入缓存。

gst_mpp_enc_loop： 按时间顺序取出编码器缓存内的buffer，使用encode_put_frame送至MPP编码，再使用encode_get_packet获取编码后码流，并送至下一级插件。

5.2.2 主要属性说明

width： 宽度，默认为0，不进行缩放。

height： 高度，默认为0，不进行缩放。

rc-mode： 码率控制模式，可选VBR，CBR和Fixed QP。

bps： 目标码率，在Fixed QP模式下忽略。

bps-max： 最高码率，在Fixed QP模式下忽略。

bps-min：最低码率，在Fixed QP模式下忽略。

gop： Group Of Picture，即两I帧的间隔。如0表示仅有一个I帧，其余为P帧，1表示全为I帧，2表示每两帧为I帧，即ＩＰＩＰＩＰ...形式。默认为-1，按帧率设置，即每秒有一个I帧。

level： 表示 SPS 中的 level_idc 参数。

profile： 表示 SPS 中的 profile_idc 参数。

rotation： 旋转输入buffer，可选0°，90°，180°，270°。

6. 环境变量

常用环境变量均整理至/etc/profile.d/gst.sh，相关详细说明可以直接查看脚本内注释。

```
# 关闭NV12 10bit输出，当解码器遇到该格式，会使用RGA转为NV12。
export GST_MPP_DEC_DISABLE_NV12_10=1
# 关闭NV16 10bit输出，当解码器遇到该格式，会使用RGA转为NV12。
export GST_MPP_DEC_DISABLE_NV16_10=1
# 开启格式转换，mppvideodec始终输出NV12格式。
export GST_MPP_VIDEODEC_DEFAULT_FORMAT=NV12
# 开启AFBC压缩格式，效果等同于设置mppvideodec arm-afbc=true，适用于gst-play-1.0等无法直接操作mppvideodec的情况。需要注意不是所有Sink都可以支持
export GST_MPP_VIDEODEC_DEFAULT_ARM_AFBC=1
# 设置v4l2输出格式。
export GST_V4L2_PREFERRED_FOURCC=NV12:YU12:NV16:YUY2
# 设置videoconvert输出格式。
export GST_VIDEO_CONVERT_PREFERRED_FORMAT=NV12:NV16:I420:YUY2
# 限制v4l2src插件最大分辨率。
export GST_V4L2SRC_MAX_RESOLUTION=3840x2160
# 设置v4l2驱动最小缓存大小。
export GST_V4L2_MIN_BUFS=64
# 设置videoconvert videoflip和videoscale使用RGA进行硬件加速。
# 需要注意会存在加速失败的情况，并且行为有可能和官方软件预期的不一样。
export GST_VIDEO_CONVERT_USE_RGA=1
export GST_VIDEO_FLIP_USE_RGA=1
...
```

7. 命令示例

7.1 播放视频

```
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
```

7.2 多路视频播放

```
# 使用waylandsink的render-rectangle指定不同的渲染位置
gst-launch-1.0 filesrc location=/usr/local/test.mp4 ! parsebin ! mppvideodec ! waylandsink render-rectangle='<0,0,400,400>' &
gst-launch-1.0 filesrc location=/usr/local/test.mp4 ! parsebin ! mppvideodec ! waylandsink render-rectangle='<0,500,400,400>' &
gst-launch-1.0 filesrc location=/usr/local/test.mp4 ! parsebin ! mppvideodec ! waylandsink render-rectangle='<0,1000,400,400>' &
```

7.3 编码预览

使用tee插件，将摄像头采集的数据拷贝为两路，其中一路送至mpph264enc进行编码，而后送至filesink保存文件。另一路送至autovideosink显示。注意在tee插件后需要加上queue插件，会对数据进行缓存，防止出现卡死的情况。

```
gst-launch-1.0 v4l2src ! 'video/x-raw,format=NV12' ! tee name=tv ! queue !
mpph264enc ! 'video/x-h264' ! h264parse ! 'video/x-h264' ! filesink
location=/data/out.h264 tv. ! queue ! autovideosink
```

7.4 拆分码流

部分插件如qtdemux，会出现多个Source Pad的情况，如音频流、视频流、字幕流等，则可以将该插件命名，并提取出需要的码流。如将qtdemux命名为qt，则qt.audio_0就是第一个音频流，qt.video_0就是第一个视频流，可提取后分别做处理。同样建议在分流后加上queue插件。不同插件码流命名方式不同，可以通过gst-inspect命令查看命名方式，或直接使用类似`qt. ! queue ! mppvidedec`的形式进行构建，gstreamer会与后级插件协商格式。

```
gst-launch-1.0 filesrc location=test.mp4 ! qtdemux name=qt qt.audio_0 ! queue !
filesink location=audio.bin qt.video_0 ! queue ! filesink location=video.bin
```

8. 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，NV16。开启方法如下：

```
# 开启全局AFBC，适用于使用gst-play-1.0等无法直接操作mppvideodec的情况
export GST_MPP_VIDEODEC_DEFAULT_ARM_AFBC=1
# 单独开启AFBC
gst-launch-1.0 filesrc location=/test.mp4 ! parsebin ! mppvideodec arm-afbc=true
! waylandsink
```

waylandsink和xvimagesink支持AFBC格式合成，或使用kmssink/rkximagesink指定Cluster图层播放，该方式需要独占图层，如：

```
# GST_DEBUG=*mpp*:4开启mpp插件DEBUG开关，可以通过rkmpp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 gst-play-1.0 --flags=3 --videosink=waylandsink test.mp4
GST_DEBUG=*mpp*:4 gst-play-1.0 --flags=3 --videosink="kmssink plane-id=101"
...
0:00:00.256819945 29143   0x7f70008700 INFO                    mppdec
gstmppdec.c:465:gst_mpp_dec_apply_info_change:<mppvideodec0> applying NV12(AFBC)
1920x1080 (1920x1104)
...
```

8.1 AFBC dump解码数据

GStreamer想要查看硬件解码的数据是否正确，可以通过下面的方式dump解码数据（一般为NV12等格式图像）：

1. 打开MPP 日志功能

```
export mpp_debug=0x400
```

然后/data目录下就会有MPP自己dump的解码后数据。这是MPP自带的dump调试功能，开启AFBC时dump不支持；未开启AFBC时dump的数据一般为NV12格式，可以使用rawplayer（或其他裸视频播放器）查看。

2. 使用GStreamer的插件filesink dump 解码数据,这种方式无论开不开启AFBC都支持dump，使用方式如下：

```
gst-launch-1.0 uridecodebin uri=file://xxx ! filesink location=xxx.yuv
```

解码的AFBC数据就在xxx.yuv文件中，因为开启了AFBC还要将dump出的图像再讲过解压缩才能够使用rawplayer（或其他裸视频播放器）查看，解压缩命令（解压缩软件afbcDec要找相关负责人获取）：

```
./afbcDec filename w h format afbcmode
#eg: ./afbcDec 178_Surfa_id-26_1088x1824_z-0.bin 1088 1824 0 1
# 0=RGBA,1=NV12,2=RGB888, afbcmode 0=afbc, 1=afbc|YTR
```

afbcDec输出的图像格式为ARGB。

若想要查看视频的每一帧是否正确，还要将filesink dump的文件分帧：因为上面的解压缩软件只能转一帧数据，而dump出的视频所有帧都在同一个文件。分帧的示例如下：

```
# GST_DEBUG查看每帧大小
GST_DEBUG=filesink:6 gst-launch-1.0 uridecodebin uri=file://xxx ! filesink
location=xxx.yuv

0:00:01.224149631 14266   0x7f7c00ab00 DEBUG  filesink
gstfilesink.c:769:gst_file_sink_flush_buffer:<filesink0> Flushing out buffer of
size 1390080
# 使用split命令分帧
split -b 1390080 -a 5 -d  xxx.yuv dump_frame
```

9. 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。

10. 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
# 也可以直接使用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。若不存在该节点，则可通过modetest -p查看。

11. 分析Gstreamer管道

Gstreamer提供将pipeline状态生成为dot文件的方法，用于pipeline分析。使用方法如下：

```
# 以下命令在板端执行
# 开启dot文件导出
$ export GST_DEBUG_DUMP_DOT_DIR=/tmp
# 执行测试命令
$ gst-play-1.0 test.mp4
# 查看dot文件是否正常导出
$ ls -alF /tmp/*.dot
-rw-rw-r-- 1 root root 49141 1月  16 10:35 /tmp/0.00.00.385340438-gst-play.async-
done.dot
# 以下命令在PC端执行
$ adb pull /tmp/0.00.00.385340438-gst-play.async-done.dot .
$ dot 0.00.00.385340438-gst-play.async-done.dot -Tpng > out.png
```

out.png里面即绘制出了pipeline的完整流程，可用于管道状态分析。有时会有多个dot文件生成，对应pipeline不同时期的不同状态，可以一一转为对应的png图片进行分析。

其中dot命令需要安装graphviz，下载地址为https://www.graphviz.org/download/。

12. FAQ

12.1 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。

12.2 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。

12.3 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`。

12.4 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。

12.5 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gstreamer测试。

12.6 运行解压缩命令afbcDec时遇到缺少库libgraphic_lsf.so报错

找相关负责人获取libgraphic_lsf.so，将缺少的libgraphic_lsf.so库拷贝到/usr/lib/目录即可。

12.7 v4l2src无法满帧获取码流

1. 开启性能模式，`echo performance | tee $(find /sys/ -name *governor)`
2. 使用v4l2-ctl命令测试帧率，命令为`v4l2-ctl -d /dev/video0 --set-fmt-video=width=1920,height=1080,pixelformat=NV12 --stream-mmap=3 --stream-skip=1 --stream-poll`，其中的/dev/video0节点及格式按照需求修改。

3. 如果v4l2-ctl可以满足帧率要求，而使用gstreamer v4l2src插件无法满足，可以尝试指定min-buffers参数，具体用法为`gst-launch-1.0 v4l2src min-buffers=64 ! video/x-raw,width=1920,height=1080,format=NV12 ! waylandsink`。
4. 尝试使用queue插件

```
# min-threshold-time单位为ns，此处为5s
gst-launch-1.0 v4l2src ! queue max-size-buffers=0 max-size-time=0 max-size-bytes=0 min-threshold-time=5000000000 ! autovideosink
```

5. 尝试指定延迟播放

```
# ts-offset单位为ns，此处为0.5s
gst-launch-1.0 v4l2src ! queue max-size-bytes=1000000000 max-size-buffers=0 max-size-time=0 ! autovideosink ts-offset=500000000
```

6. 强制开启v4l2src拷贝

```
gst1-plugins-good $ git diff
diff --git a/sys/v4l2/gstv4l2object.c b/sys/v4l2/gstv4l2object.c
index 2fb9091b..efa94561 100644
@@ -4792,6 +4794,7 @@ gst_v4l2_object_decide_allocation (GstV4l2Object * obj,
GstQuery * query)
  /* We can't share our own pool, if it exceed V4L2 capacity */
  if (min + obj->min_buffers + 1 > VIDEO_MAX_FRAME)
    can_share_own_pool = FALSE;
+  can_share_own_pool = FALSE;

  /* select a pool */
  switch (obj->mode) {
```

12.8 v4l2src格式协商不通过

`v4l2-ctl -d /dev/video0 --list-formats-ext`列出支持的所有格式和分辨率，其中`/dev/video0`按实际节点配置。

如果摄像头支持YUV格式（如NV12，NV16等）、RGB格式（如BGRA，BGRx等）裸流输出，则后续可以直接使用显示插件、编码插件等，如果摄像头支持jpeg格式输出，则后续可以使用mppjpegdec等插件进行解码，再进行后续处理。如果摄像头输出与后级插件支持格式不匹配，则可在两插件间添加videoconvert插件进行格式转换。

12.9 v4l2src如何获取HDMIIN数据

具体用法与摄像头一致。可通过`grep '' /sys/class/video4linux/*/name`命令找到名为hdmirx的节点，将v4l2src参数替换为对应节点即可。

12.10 v4l2src无法获取4K以上分辨率的数据

在脚本内做了限制，编辑/etc/profile.d/gst.sh，将`export GST_V4L2SRC_MAX_RESOLUTION=3840x2160`修改为需要的大小。

buildroot环境使用的脚本为`<SDK>/buildroot/package/gstreamer1/gstremaer1/gst.sh`，debian使用的脚本为`<SDK>/debian/overlay/etc/profile.d/gst.sh`。

1. 该处仅列出插件支持的格式，具体芯片是否支持请查询相关datasheet。

2. 目前部分平台如RK3588 RGA功能异常，不建议使用。

3. 目前部分平台如RK3588 RGA功能异常，不建议使用。