Rockchip Rockit用户指南

文件标识：RK-YH-YF-960

发布版本：V0.0.1

日期：2022-08-17

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 rockit模块功能及参数介绍

产品版本

芯片名称	版本
RK3588	5.10

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
2022-08-17	V0.0.1	xdj	初始版本

目录

1. 源码路径

- rockit仓库位于/external/rockit。

2. 开发指南文档

- Rockit开发指南：/docs/Socs/RV1126_RV1109/Multimedia/Rockchip_Developer_Guide_Linux_Rockit_CN.pdf。
- Rockit开发指南：/external/rockit/tgi/doc/Rockchip_Developer_Guide_Linux_Rockit_CN.pdf。
- MPI开发指南：/external/rockit/mpi/doc/Rockchip_Developer_Guide_Linux_Rockit_CN.pdf。
- 多媒体处理应用开发常见问题与解决方法：/external/rockit/mpi/doc/Rockchip_FAQ_MPI_CN.pdf。

3. ROCKIT编译

- 执行“source buildroot/build/envsetup.sh”命令，输入对应的序号；
- 在buildroot目录下，输入make menuconfig进入图形化内核配置，搜索“rockit”，打开其配置。由于后续需要，可同时开启RGA和GPU配置；

```
BR2_PACKAGE_ROCKIT=y
```

- 执行“make rockit”命令编译
- 重新编译烧录固件

4. 常用模块

4.1 VENC模块

4.1.1 功能描述

VENC 模块，即视频编码模块，主要支持H264、H265、JPEG、MJPEG。本模块支持多路实时编码，且每路编码独立，编码协议和编码 profile 可以不同。本模块支持视频编码同时，调用 Region 模块对编码图像内容进行叠加和遮挡。

4.1.2 测试命令

```
rk_mpi_venc_test
```

4.1.3 参数介绍

```
-i, --input=<str>              # 输入的文件名
-o, --output=<str>             # 编码输出目录
-n, --loop_count=<int>         # 循环测试的次数，默认值为1
-w, --width=<int>              # 输入图片宽度<必需>
-h, --height=<int>             # 输入图片高度<必需>
--vir_width=<int>              # 虚拟宽度
--vir_height=<int>             # 虚拟高度
-f, --pixel_format=<int>       # 输入图片像素格式，默认值0（0: NV12)
                               # 完整格式详情参看
<SDK>/external/rockit/mpi/sdk/include/rk_comm_video.h
                               # PIXEL_FORMAT_E的定义，下同
-C, --codec=<int>              # venc 编码器(8:h264, 9:mjpeg, 12:h265, 15:jpeg,
...). 默认值为8
--channel_index=<int>          # venc 通道索引，默认值为0
--enc_buf_cnt=<int>            # venc 编码输出缓冲区计数. 默认值为8
--crop=<int>                   # crop 类型(0: none 1, crop_only 2:crop _scale). 默认
值为0
-g,--gop_mode=<int>            # gop 模式(0/1:NORMALP 2:TSVC2 3:TSVC3 4:TSVC4
5:SMARTP) 默认值为0
-s,--snap_pic_cnt=<int>        # 输出码流的帧率
--rotation=<int>               # 图片旋转输出(0: 0 1: 90 2: 180 3: 270). 默认值为0
--compress_mode=<int>          # 设置输入压缩模式(0: MODE_NONE 1.AFBC_16x16). 默认值为0
--rc_mode=<int>                # rc 模式，默认值为1
                               # 0:NULL
                               # 1:H264CBR
                               # 2:H264VBR
                               # 3:H264AVBR
                               # 4:H264FIXQP
                               # 5:MJPEGCBR
                               # 6:MJPEGVBR
                               # 7:MJPEGFIXQP
                               # 8:H265CBR
                               # 9:H265VBR
                               # 10:H265AVBR
                               # 11:H265FIXQP
-b,--bit_rate=<int>            # 比特率 kbps（h264/h265对应的范围是3-200000，jpeg/mjpeg
的范围是5-800000，默认值为
                               # 10*1024kb)
--gop_size=<int>               # gop 的大小（范围>=1，默认值为60)
--full_range=<int>             # 设置颜色范围（0: limit color range，1: full color
range)，默认值为1
--slice_split=<int>            # 切片分割测试（0: 禁用，1: 启用)，默认为0
```

4.2 VPSS模块

4.2.1 功能描述

VPSS（Video Process Sub-System）是视频处理子系统，支持的具体图像处理功能包括Crop、Scale、像素格式转换、固定角度旋转、Cover/Coverex、Mirror/Flip、压缩解压等。

4.2.2 测试命令

```
rk_mpi_vpss_test
```

4.2.3 参数介绍

```
-i,--input=<str>                    # 输入文件名字，默认值为NULL
-o,--output=<str>                   # 输出文件路径，默认值NULL
-n,--loop_count=<int>               # 循环测试的次数，默认值为1
--video_proc_dev_type=<int>         # 视频处理的设备类型(0: GPU,1: RGA)，默认值为0
-g,--group_count=<int>              # vpss group的计数，默认值为1
-c,--channel_count=<int>            # vpss的通道数，默认值为1
--group_crop_en                     # vpss group裁剪已启用，默认值为0
--channel_crop_en                   # vpss信道裁剪已启用，默认值为0
--group_crop_ratio=<int>            # vpss group裁剪比,范围为1-1000，默认值为1000
--channel_crop_ratio=<int>          # vpss信道裁剪比.范围为1-1000，默认值为1000
-w, --src_width=<int>               # 源位图对应的宽度<必需>
-h, --src_height=<int>              # 源位图对应的高度<必需>
--src_vir_width=<int>               # 源位图对应的虚宽，默认为0
--src_vir_height=<int>              # 源位图对应的虚高，默认为0
-m, --src_compress=<int>            # 源位图压缩模式，默认为0
-f, --src_format=<int>              # 源位图像素格式，默认值为0(即NV12)
-W, --dst_width=<int>               # 目标位图对应的宽度<必需>
-H, --dst_height=<int>              # 目标位图对应的高度<必需>
-M, --dst_compress=<int>            # 目标位图压缩模式，默认值为0
-F, --dst_format=<int>              # 目标位图像素格式，默认值为0(即NV12)
-r, --rotation=<int>                # 固定角度旋转（0: 0. 1: 90. 2: 180. 3: 270）
--grp_rotation=<int>                # 固定组旋转角度，默认值为0(0: 0. 1: 90. 2: 180. 3:
270)
-R, --rotation_ex=<int>             # 任意角度旋转，默认值为0
-a, --attach_mb_pool=<int>          # 是否使用公共内存池，默认为0,不启用(0: RK_FALSE, 1:
RK_TRUE)
--chn_mode=<int>                    # 通道模式，默认值为0(0: USER, 1: AUTO, 2: PASS-
THOUGH)
-d, --chn_depth=<int>               # 通道输出深度，默认值为8
--mirror=<int>                      # 图片镜像操作，默认值为0
--flip=<int>                        # 图片上下翻转操作
--src_chn_rate=<int>                # 源位图通道帧率控制，默认为-1
--dst_chn_rate=<int>                # 目标位图通道帧率控制，默认为-1
--src_grp_rate=<int>                # 源位图 vpss group帧率控制，默认为-1
--dst_grp_rate=<int>                # 目标位图 vpss group帧率控制，默认为-1
```

4.3 ADEC模块

4.3.1 功能描述

ADEC 模块,即音频解码模块，音频解码内部提供g711a、g711u、g722、g726等格式的音频解码功能，并支持外部注册解码器。

4.3.2 测试命令

```
rk_mpi_adec_test
```

4.3.3 参数介绍

```
-i, --input=<str>              # 输入文件名字，如(./*.mp3)<必需>
-C, --codec=<str>              # 解码，如(mp2/g711a/g711u/g726)<必需>
--input_ch=<int>               # 输入流的通道数量<必需>
--input_rate=<int>             # 输入流的采样率<必需>
-o, --output=<str>             # 输出的文件名字，如(./*.pcm),无默认值
-n, --loop_count=<int>         # 循环测试的次数，默认值为1
-c, --channel_count=<int>      # adec信道的计数，  默认值为1
--dec_mode=<int>               # 音频流解码格式(0: pack mode, 1: stream mode)，默认值
为0
--query_stat=<int>             # 查询adec统计信息(0：查询，1：不查询)，默认值为0
--clr_buf=<int>                # 清除通道缓冲区，范围(0,1)，默认值为0
```

4.4 AENC模块

4.4.1 功能描述

AENC 模块,即音频编码模块，音频编码内部提供g711a、g711u、g722、g726等格式的音频编码功能，并支持外部注册编码器。

4.4.2 测试命令

```
rk_mpi_aenc_test
```

4.4.3 参数介绍

```
-i, --input=<str>              # 输入文件名字，如(./*.mp3) <必需>
-C, --codec=<str>              # 编码，如(mp3/aac/flac/mp2/g722/g726)<必需>
--input_ch=<int>               # 输入流通道的数量<必需>
--input_rate=<int>             # 输入流的采样率<必需>
--input_format=<int>           # 输入流的输入格式 <必需>
-o, --output=<str>             # 输出文件名，如.(./*.pcm)，无默认值
-n, --loop_count=<int>         # 循环测试的次数，默认值为1
-c, --channel_count=<int>      # adec信道的计数，默认值为1
--frame_size=<int>             # 发送帧的大小，默认值为1024
```

4.5 AO模块

4.5.1 功能描述

音频输出模块，音频输入通过对RK芯片音频接口的控制实现音频输出功能。

4.5.2 测试命令

```
rk_mpi_ao_test
```

4.5.3 参数介绍

```
-i, --input=<str>              # 输入文件名称，如(./*.pcm) <必需>
--input_ch=<int>               # 输入通道的数量 <必需>
--input_rate=<int>             # 输入数据的采样率 <必需>
--device_ch=<int>              # 声卡的通道数量，默认值为2
--device_rate=<int>            # 声卡的采样率，默认值为48000
-o, --output=<str>             # 输出的文件名字，如(./ao)，无默认值
-n, --loop_count=<int>         # 循环测试的次数，默认值为1
-c, --channel_count=<int>      # ao的通道计数，默认值为1
--bit=<int>                    # 声卡的位宽度，范围是（8， 16， 24），默认值为16
--sound_card_name=<str>        # 所打开的声卡名称，默认值为NULL
--set_volume=<int>             # 音量设置，范围是(0, 100)，默认值为100
--set_mute=<int>               # 静音设置，1表示静音，0 表示非静音，默认值为0
--set_track_mode=<int>         # 设置通道模式，默认值为0
                               # 0:normal(左右声道正常),
                               # 1:both_left(左右声道都是左声道数据),
                               # 2:both_right(左右声道都是右声道数据),
                               # 3:exchange(左右声道数据交换),
                               # 4:mix(左右声道数据混音输出),
                               # 5:left_mute(左声道静音),
                               # 6:right_mute(右声道静音),
                               # 7:both_mute(左右声道都静音)
--get_volume=<int>             # 查询音量，0表示不查询，1表示查询，默认值为0
--get_mute=<int>               # 查询设置静音时的静音状态，1表示静音，0表示非静音，默认值为
0
--get_track_mode=<int>         # 查询通道模式，默认值为0
--query_stat=<int>             # 查询ao统计信息
```

```
--pause_resume=<int>            # 暂停恢复功能，1表示暂停播放1秒，0表示不暂停，默认值为0
--save_file=<int>               # 测试ao保存文件，如果启用，则必须将其设置为输出文件，0表示
不启用，1表示启用。 默认值为0
--query_file_stat=<int>         # 查询文件状态，0表示不查询，1表示查询。 默认值为0
--get_attr=<int>                # 获取设备的属性，0表示不获取，1表示获取，默认值为0
```

4.6 AI模块

4.6.1 功能描述

音频输入模块，音频输入通过对RK芯片音频接口的控制实现音频输入功能。

4.6.2 测试命令

```
rk_mpi_ai_test
```

4.6.3 参数介绍

```
--device_rate=<int>             # 声卡的采样率<必需>
--device_ch=<int>               # 声卡通道的数量<必需>
--out_ch=<int>                  # 输出数据的通道<必需>
--out_rate=<int>                # 输出数据的采样率<必需>
-o, --output=<str>              # 输出文件的名字，如(./ai)，默认值为NULL
-n, --loop_count=<int>          # 循环测试的次数，默认值为1
-c, --channel_count=<int>       # adec信道的计数，默认值为1
--bit=<int>                     # 声卡的位宽，可选值为(8, 16, 24)，默认值为16
--sound_card_name=<str>         # 所打开声卡的名字，无默认值
```

4.7 MB模块

4.7.1 功能描述

实现通用化内存接口，内存和内存池管理。

4.7.2 测试命令

```
rk_mpi_mb_test
```

4.7.3 参数介绍

```
-n, --loop=<int>           # 循环次数，默认值为1
-c, --mb_count=<int>       # mb计数，默认值为1
-s, --mb_size=<int>        # mb的大小，默认值为4MB
-p, --pool_count=<int>     # pool的计数，默认值为1
-a, --pre_alloc=<int>      # 是否提前分配，默认值为0（0: no 1: yes）
-r, --remap_mode=<int>     # 重映射模式，默认值为2（0: none, 256: no cache, 512:
cached）
-t, --alloc_type=<int>     # 分配类型，默认值为0（0: DMA, 1: malloc）
```

4.8 VGS模块

4.8.1 功能描述

VGS（ Video Graphics Sub-System）视频图形子系统，主要是对输入的图像进行缩放、旋转、叠加OSD、叠加COVER、画线等操作。

4.8.2 测试命令

```
rk_mpi_vgs_test
```

4.8.3 参数介绍

```
-i, --input=<str>          # 输入文件名，如（/userdata/1080p.nv12）<必需>
-o, --output=<str>         # 输出文件路径，如（/userdata/vgs/），无默认值
-n, --loop_count=<int>     # 测试循环次数，默认值为1
-j, --job_number=<int>     # vgs对应的job编号，默认值为1
-t, --task_number=<int>    # 每一个job对应的task数量，默认值为1
--osd_file=<str>           # osd 文件路径，如（/userdata/vgs/），无默认值
--task_type=<int>          # vgs对应的task类型，范围是（1,6）（1.scale. 2.rotate.
3.draw_line. 4.cover.
                           # 5.osd. 6.mosaic.）
--task_mode=<int>          # task模式，默认值为1（1: 每一个job只有一个task类型 2: 所有的
task类型都在一个job上）
--task_array_size=<int>    # array task的大小，范围为（1,100），默认值为1
--angle=<int>              # 旋转角度，默认值为1（0:0. 1:90. 2:180. 3:270）
--src_width=<int>          # 源位图宽度，如（1920）<必需>
--src_height=<int>         # 源位图高度，如（1080）<必需>
--src_vir_width=<int>      # 源位图虚宽，如（1920）
--src_vir_height=<int>     # 源位图虚高，如（1080）
--src_compress=<int>       # 源位图的压缩模式，默认值为0.
--src_format=<int>         # 源位图像素格式，默认值为0（ 0 is NV12）
--dst_width=<int>          # 源位图宽度，如（1920）<必需>
--dst_height=<int>         # 目标位图的高度，如（1080）<必需>
--dst_compress=<int>       # 目标位图的压缩格式.，默认值为0
--dst_format=<int>         # 目标位图像素格式dst pixel format，默认值为0（ 0 is NV12）
```

```
--osd_width=<int>          # osd宽度，  如（1920）<需要在 OSD 上>
--osd_height=<int>         # osd高度，如（1080）<需要在 OSD 上>
--osd_compress=<int>       # osd压缩模式，默认值为0
--osd_format=<int>         # osd像素格式，默认值为0（0 is NV12）
--crop_x=<int>             # 裁剪比例矩形x坐标，默认值为0
--crop_y=<int>             # 裁剪比例矩形y坐标，默认值为0
--crop_w=<int>             # 裁剪比例矩形宽度，默认值为0
--crop_h=<int>             # 裁剪比例矩形高度，默认值为0
```

4.9 VDEC模块

4.9.1 功能描述

VDEC 模块提供驱动视频解码硬件工作的 MPI 接口，实现视频解码功能。

4.9.2 测试命令

```
rk_mpi_vdec_test
```

4.9.3 参数介绍

```
-i, --input=<str>              # 输入的文件名<必需>
-o, --output=<str>             # 解码输出目录
-C, --codec=<int>              # 输入流解码器(8:h264 9:mjpeg 12:h265,...) <在
StreamMode需要>
-n, --loop_count=<int>         # 循环测试的次数，默认值为1
-w, --width=<int>              # 输入源宽度<StreamMode上需要>
-h, --height=<int>             # 输入源高度<StreamMode上需要>
--channel_index=<int>          # vdec通道索引，默认值为0
-c, --channel_count=<int>      # vdec通道计数，默认值为1
--dec_mode=<int>               # vdec解码模式，可选值(0:StreamMode, 1:FrameMode)，默认
值为0
--dec_buf_cnt=<int>            # vdec解码输出缓冲区计数，默认值为8
--compress_mode=<int>          # vdec压缩模式，默认值为0（ 0: NONE, 1: AFBC_16X16）
--en_mbpool=<int>              # 开启mb pool，默认值为0
--pixfmt=<int>                 # jpeg 输出像素模式，默认值为0（0: YUV420SP）
--en_dei=<int>                 # 开启deinterlace，默认值为0
--en_colmv=<int>               # 开启colmv，默认值为1
```

4.10 TDE模块

4.10.1 功能描述

TDE（Two Dimensional Engine）利用硬件RGA提供快速的图形处理功能，主要有快速位图搬移、快速色彩填充、快速位图旋转、快速位图缩放、位图格式转换、位图alpha 叠加、ColorKey 操作。

4.10.2 测试命令

```
rk_mpi_tde_test
```

4.10.3 参数介绍

```
-i, --input=<str>          # 输入文件名字，如（/userdata/1080p.nv12） <必需>
-w, --src_width=<int>      # 源位图的宽度<必需>
-h, --src_height=<int>     # 源位图的高度<必需>
--src_vir_width=<int>      # 源位图的虚宽
--src_vir_height=<int>     # 源位图的虚高
--src_compress=<int>       # 源位图的压缩模式
-W, --dst_width=<int>      # 目标位图的宽度<必需>
-H, --dst_height=<int>     # 目标位图的高度<必需>
--dst_compress=<int>       # 目标位图的压缩模式
-o, --output=<str>         # 输出文件路径，如（/userdata/tde/），无默认值
--background=<str>         # 背景文件，如（/userdata/tde/xxx.bin），无默认值
-n, --loop_count=<int>     # 循环测试的次数，默认值为1
-p, --operation=<int>      # 操作类型，默认值为0
                           # 0: quick copy（快速拷贝），
                           # 1: quick resize（缩放），
                           # 2: quick fill（填充），
                           # 3: rotation（旋转），
                           # 4: mirror and flip（镜像或翻转），
                           # 5: colorkey
--src_rect_x=<int>         # 源位图操作区域对应的x坐标，默认值为0
--src_rect_y=<int>         # 源位图操作区域对应的y坐标，默认值为0
--src_rect_w=<int>         # 源位图操作区域的宽度，默认值为src_width
--src_rect_h=<int>         # 源位图操作区域的高度，默认值为src_height
--dst_rect_x=<int>         # 目标位图操作区域对应的x坐标，默认值为0
--dst_rect_y=<int>         # 目标位图操作区域对应的y坐标，默认值为0
--dst_rect_w=<int>         # 目标位图对应的宽度，默认值为dst_width
--dst_rect_h=<int>         # 目标位图对应的高度，默认值为dst_height
--performace=<int>         # 测试性能模式，默认值为0
--proc_time=<int>          # 处理时间，默认值为800
--colorkey=<int>           # colorkey的值，默认值为0
-c, --fill color=<int>     # 填充颜色，默认值为0
-r, --rotation=<int>       # 旋转角度，默认值为0（0: 0. 1: 90. 2: 180. 3: 270）
-m, --mirror=<int>         # 镜像或翻转. 默认值为0（0: none. 1: flip. 2: mirror. 3:
both）
```

4.11 VO模块

4.11.1 功能描述

视频输出（VO）模块用于视频输出管理，支持多VOP以及多图层显示。实现启用视频输出设备或通道、发送视频数据或者UI数据到输出通道等功能。

4.11.2 测试命令

```
rk_mpi_vo_test
```

4.11.3 参数介绍

```
-i, --input=<str>          # 输入配置文件<必需>
-d, --device_id=<int>      # vop对应id，如(0/1)，默认值为0
-l, --layer_id=<int>       # 图层id. 如(0/2/4/6)，默认值为0
--wbc_enable=<int>         # 回写启用. 如(0)，默认值为0
--wbc_bind=<int>           # 启用回写绑定，默认值为1(0: disable, 1: enable)
--ui=<int>                 # ui，如(0)，默认值为0
--loopCount=<int>          # 循环次数，如(0)，默认值为10
--ui_alpha=<int>           # ui_alpha，如(0)，默认值为0
-w, --Windows=<int>        # 窗口数量，如 [1-64]，默认值为4,最大值为63
--ConnectorType=<int>      # 连接的类型，如(0: HDMI 1: EDP 2: VGA)<必需>
--layer_mode=<int>         # 图层类型，如(0: CURSOR 1: UI 2: Video)<必需>
--display_mode=<int>       # 播放，如(12/14)，默认值为12 (12 为 1080P60)<必需>
--display0_mode=<int>      # 播放，如(12/14)， 默认值为12 (12 为 1080P60)<必需>
--disp_frmrt=<int>         # 显示帧率，默认值为25
--disp_frmrt_ratio=<int>   # 显示帧率比率，如(32, 16, 8, 4, 2 ,1)，默认值为1
--aspect_mode=<int>        # 屏幕纵横比，如(1: ratio no change 2: ratio manual set),
默认值为1
--border_lpx=<int>         # 边框属性对应的lpx，默认值为2
--border_rpx=<int>         # 边框属性对应的rpx，默认值为2
--border_tpx=<int>         # 边框属性对应的tpx，默认值为2
--border_bpx=<int>         # 边框属性对应的bpx，默认值为2
--disp_x=<int>             # disp_x，默认值为0
--disp_y=<int>             # disp_y，默认值为0
--video_format=<int>       # 视频像素格式(0: ARGB8888 1: ABGR888 2: RGB888 3: BGR888
4: RK_FMT_YUV420SP)，默认值
                           # 为4(4 表示 RGB888)
--disp_width=<int>         # dst 宽度，如(1920)<必需>
--disp_height=<int>        # dst 高度. 如(1080)<必需>
--image0_width=<int>       # image0 宽度，如(1920)<必需>
--image0_height=<int>      # image0 高度，如(1080)<必需>
--image1_width=<int>       # image1 宽度，如(1024)<必需>
--image1_height=<int>      # image1 高度，如(768)<必需>
--disp0_width=<int>        # 目标位图宽度，如(1920)<必需>
--disp0_height=<int>       # 目标位图高度，如(1080)<必需>
--disp1_width=<int>        # 目标位图宽度，如(1024)<必需>
--disp1_height=<int>       # 目标位图高度，如(768)<必需>
```

```
--image_width=<int>        # 目标位图宽度，如(1920)<必需>
--image_height=<int>       # 目标位图高度，如(1080)<必需>
--wbc_width=<int>          # 目标位图宽度，如(1920)<必需>
--wbc_height=<int>         # 目标位图高度，如(1080)<必需>
--wbc_compress=<int>       # 回写压缩模式，默认值为0
--wbc_format=<int>         # 回写像素格式,如(0: ARGB8888 1: ABGR888 2: RGB888 3:
BGR888)，默认值为0
--wbc_type=<int>           # 回写操作类型，如(0: dev 1: video)，默认值为1
--wbc_id=<int>             # 回写操作id，默认值为0
--voplay=<str>             # 播放视频测试，默认值为0 (0: RK_FALSE, 1: RK_TRUE)
--bBorder=<str>            # 边界启用，默认值为0 (0: RK_FALSE, 1: RK_TRUE)
--wbc_auto=<str>           # 回写自动绑定，默认值为1 (0: RK_FALSE, 1: RK_TRUE)
--screen0_chn=<int>        # screen0对应的通道数，默认值为16
--chn_display=<int>        # 通道显示模式，如(0: normol 1: pause 2: step 3: speed),
默认值为0
--screen1_chn=<int>        # screen1对应的通道数，默认值为4
--screen0_rows=<int>       # 显示screen0的行和列，默认值为4 (4x4)
--screen1_rows=<int>       # 显示screen1的行和列，默认值为3 (3x3)
--en_wbc=<int>             # 启用回写，默认值为0
--en_chnPriority=<int>     # 启用通道优先级，默认值为0
--wbc_src=<int>            # 回写的来源，默认值为1
--double_screen=<int>      # 是否双屏，默认值为1( 0: FALSE, 1: TRUE)
--Homologous=<int>         # Homologous 显示，默认值为0
```

4.12 AVS模块

4.12.1 功能描述

AVS（Any View Stitching，全景拼接）实现的功能：对多路图像进行全景拼接，并且按照指定的投影模式输出图像。

4.12.2 测试命令

```
rk_mpi_avs_test
```

4.12.3 参数介绍

```
-m, --test_mode=<int>                # 测试模式，默认值为0
                                     # 0: avs module. 8xEquirectangular,
6xRectilinear.
                                     # 1: avs module. 6xNoBlend_Hor,
6xNoBlend_Ver, 4xNoBlend_Qr.
                                     # 2: vi -> avs -> vo. 6xEquirectangular.)
-n, --loop_count=<int>               # 循环测试的次数，默认值为100
-c, --link_compress_mode=<int>       # 压缩模式，默认值为1( 0: Uncompress, 1: AFBC)
-p, --avs_pipe_sync=<int>            # 是否启用avs管道同步，默认值为0( 0: Disable)
--params_sources=<int>               # 通过 mesh 或者 calib的用于全景拼接所需的参数，默
认值为0(0: mesh, 1: calib)
--connector_type=<int>               # 连接类型，默认值为3（0: HDMI0, 3: MIPI)
```

4.13 VI模块

4.13.1 功能描述

视频输入（VI）模块实现的功能：通过 MIPI Rx(含 MIPI 接口、LVDS 接口)，BT.1120，BT.656，BT.601，DC 等接口接收视频数据。VI 将接收到的数据存入到指定的内存区域，实现视频数据的采集。

4.13.2 测试命令

```
rk_mpi_vi_test
```

4.13.3 参数介绍

```
-w, --width=<int>          # 设置捕获通道宽度，默认值为0<必需>
-h, --height=<int>         # 设置捕获通道高度，默认值为0<必需>
-d, --dev=<int>            # 设置设备id，默认值为0
-p, --pipe=<int>           # 设置管道id，默认值为0
-c, --channel=<int>        # 设置通道id，默认值为1
-l, --loopcount=<int>      # 设置捕获帧计数（默认值为100）
-C, --compressmode=<int>   # 设置捕获压缩模式，默认值为0（0:MODE_NONE 1:AFBC_16x16）
-o, --output=<int>         # 保存输出文件,文件存放
于/data/test_<devid>_<pipeid>_<channelid>.bin，默认值为0（0:no
                           # save 1:save）
-m, --mode=<int>           # 测试模式，默认值为1
                           # 0:vi get&release frame
                           # 1:vi bind one venc（h264）
                           # 2:vi bind two venc（h264）
                           # 3:vi bind vpss bind venc
                           # 4:vi bind vo（only support 356x now）
-t, --memorytype=<int>     # 设置buf内存类型，默认值为4<必需>
                           # 1:mmap（hdmiin/bt1120/sensor input）
                           # 2:userptr（invalid）
                           # 3:overlay（invalid）
                           # 4:dma（sensor）
```

```
-n, --name=<str>           # 设置实体名称，无默认值（rv1126 sensor:rkispp_m_bypass
rkispp_scale0 rkispp_scale1
                           # rkispp_scale2; rv1126
hdmiin/bt1120/sensor:/dev/videox such as /dev/video19
                           # /dev/video20）<必需>;rk356x
hdmiin/bt1120/sensor:/dev/videox such as /dev/video0
                           # /dev/video1）
-D, --depth=<int>          # 通道输出深度，默认{u32BufCount（not bind） or 0（bind
venc/vpss/...）}
-f, --format=<int>         # 设置格式，默认值为0
                           # 0:RK_FMT_YUV420SP
                           # 10:RK_FMT_YUV422_UYVY1310
                           # 80:RK_FMT_RGB_BAYER_SBGGR_12BPP
--src_rate=<int>           # 源位图帧速率，默认值为-1（-1:not control; other:1-
max_fps<isp_out_fps>）
--dst_rate=<int>           # 目标位图帧速率，默认值为 -1（-1:not control; other:1-
src_fps<src_rate>）
-U, --user_pic=<int>       # 启用使用用户指定图片作为vi输入
--rgn_type=<int>           # rgn类型（0:overlay,
1:overlayEx,2:cover,3:coverEx,4:mosaic,5:moscaiEx）
--rgn_cnt=<int>            # rgn计数，默认值为1，最大值为8
--en_rgn=<int>             # 启用 rgn，默认值为0
--get_connect_info=<int>   # 获取连接信息，默认值为0
--get_edid=<int>           # 获取edid，默认值为0
--set_edid=<int>           # 设置edid，默认值为0
```

4.14 RGN模块

4.14.1 功能描述

用户一般都需要在视频中叠加 OSD 用于显示一些特定的信息（如：通道号、时间戳等），必要时还会填充色块。这些叠加在视频上的 OSD 和遮挡在视频上的色块统称为区域。REGION 模块，用于统一管理这些区域资源。

4.14.2 测试命令

```
rk_mpi_rgn_test
```

4.14.3 参数介绍

```
-i, --input_raw_name=<str>    # 输入原始数据文件名，默认值为RK_NULL
--input_bmp_name=<str>        # 输入 bmp 数据文件名<必需>
-o, --output_name=<str>       # 输出流文件名，默认为RK_NULL
-r, --rgn_count=<int>         # rgn的句柄数量，默认值为1
-p, --operation=<int>         # rgn操作， 默认值为0（0: overlay. 1: cover. 2:
mosaic. 3: line）
-x, --rect_x=<int>            # rgn区域起始 x 坐标，默认值为0
-y, --rect_y=<int>            # rgn区域起始 y 坐标，默认值为0
```

```
-w, --bmp_w=<int>               # bmp 的宽度，默认值为0<必需>
-h, --bmp_h=<int>               # bmp 的高度，  默认值为0<必需>
-W, --raw_w=<int>               # 原始宽度，默认值为0 <必需>
-H, --raw_h=<int>               # 原始高度，  默认值为0 <必需>
-F, --raw_fmt=<int>             # 原始像素格式，默认值为0（0: NV12）
-f, --format=<int>              # bmp像素格式，默认值为65557（65546: ARGB1555, 65557:
BGRA5551）
--mod=<int>                     # 附加模块，默认值为4（ 4: VENC, 6: VPSS）
--cover_type=<int>              # 覆盖类型，默认为0（ 0: rect, 1: quad）
```

4.15 SYS模块

4.15.1 功能描述

系统控制模块实现RK MPI通用功能接口，提供系统相关功能、大块物理内存管理等功能。

4.15.2 测试命令

```
rk_mpi_sys_test
```

4.15.3 参数介绍

```
-n, --loop_count=<int>          #循环测试次数，默认值为1
--device_id=<int>               #模块设备对应的id，  默认值为0
--src_channel_id=<int>          #原模块通道id，默认值为0
--dst_channel_count=<int>       #dst模块信道的计数，  默认值为1
```