

# Rockchip Rockit User Guide

---

ID: RK-YH-YF-960

Release Version: V0.0.1

Release Date: 2022-08-17

Security Level: ☐Top-Secret ☐Secret ☐Internal ☒Public

## DISCLAIMER

THIS DOCUMENT IS PROVIDED "AS IS". ROCKCHIP ELECTRONICS CO., LTD. ("ROCKCHIP") DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

## Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip's registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

**All rights reserved. ©2022. Rockchip Electronics Co., Ltd.**

Beyond the scope of fair use, neither any entity nor individual shall extract, copy, or distribute this document in any form in whole or in part without the written approval of Rockchip.

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian, PRC

Website: [www.rock-chips.com](http://www.rock-chips.com)

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: [fae@rock-chips.com](mailto:fae@rock-chips.com)

## Preface

## Overview

This document presents a brief introduction to functions and parameters of Rockit Module.

## Product Version

Chipset	Kernel Version
RK3588	5.10

## Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

## Revision History

Date	version	Author	Change Description
2022-08-17	V0.0.1	xdj	Initial version

## Contents

### Rockchip Rockit User Guide

1. Path of Source Code
2. Developer Documents
3. Rockit Compilation
4. Common Modules
  - 4.1 VENC Module
    - 4.1.1 Introduction
    - 4.1.2 Testing Commands
    - 4.1.3 Parameters
  - 4.2 VPSS Module
    - 4.2.1 Introduction
    - 4.2.2 Testing Command
    - 4.2.3 Parameters
  - 4.3 ADEC Module
    - 4.3.1 Introduction
    - 4.3.2 Testing Command
    - 4.3.3 Parameters
  - 4.4 AENC Module
    - 4.4.1 Introduction
    - 4.4.2 Testing Command
    - 4.4.3 Parameters
  - 4.5 AO Module
    - 4.5.1 Introduction
    - 4.5.2 Testing Command
    - 4.5.3 Parameters
  - 4.6 AI Module
    - 4.6.1 Introduction
    - 4.6.2 Test Commands
    - 4.6.3 Parameters
  - 4.7 MB Module
    - 4.7.1 Introduction
    - 4.7.2 Test Commands
    - 4.7.3 Parameters
  - 4.8 VGS Module
    - 4.8.1 Introduction
    - 4.8.2 Test Commands
    - 4.8.3 Parameters
  - 4.9 VDEC Module
    - 4.9.1 Introduction
    - 4.9.2 Test Commands
    - 4.9.3 Parameters
  - 4.10 TDE Module
    - 4.10.1 Introduction
    - 4.10.2 Test Commands
    - 4.10.3 Parameters
  - 4.11 VO Module
    - 4.11.1 Introduction
    - 4.11.2 Test Commands
    - 4.11.3 Parameters
  - 4.12 AVS Module
    - 4.12.1 Introduction
    - 4.12.2 Test Commands
    - 4.12.3 Parameters
  - 4.13 VI Module
    - 4.13.1 Introduction

- 4.13.2 Test Commands
  - 4.13.3 Parameters
- 4.14 RGN Module
  - 4.14.1 Introduction
  - 4.14.2 Test Command
  - 4.14.3 Parameters
- 4.15 SYS Module
  - 4.15.1 Introduction
  - 4.15.2 Test Command
  - 4.15.3 Parameter

## 1. Path of Source Code

---

- The rockit repository is located at `<SDK>/external/rockit`.

## 2. Developer Documents

---

- Rockit Developer Guide:  
`<SDK>/docs/Socs/RV1126_RV1109/Multimedia/Rockchip_Developer_Guide_Linux_Rockit_CN.pdf`
- Rockit Developer Guide:  
`<SDK>/external/rockit/tgi/doc/Rockchip_Developer_Guide_Linux_Rockit_CN.pdf`
- MPI Developer Guide:  
`<SDK>/external/rockit/mpi/doc/Rockchip_Developer_Guide_Linux_Rockit_CN.pdf`
- FAQ about Multimedia Application Development: :  
`<SDK>/external/rockit/mpi/doc/Rockchip_FAQ_MPI_CN.pdf`

## 3. Rockit Compilation

---

- Execute the command `source buildroot/build/envsetup.sh` and enter the corresponding number.
- In the buildroot directory, input `make menuconfig` to enter the graphical kernel configuration. Search for `rockit` and enable its configuration. Additionally, enable RGA and GPU configurations if needed.

```
BR2_PACKAGE_ROCKIT=y
```

- Execute the command `make rockit` to build.
- Rebuild and burn the firmware.

## 4. Common Modules

---

### 4.1 VENC Module

#### 4.1.1 Introduction

The video encoding module(VENC) mainly supports H264, H265, JPEG, and MJPEG encoding. This module supports real-time encoding of multiple streams, with each stream being independently encoded and supporting different encoding protocols and profiles. This module also supports overlay and occlusion of encoded image content by calling the Region module.

## 4.1.2 Testing Commands

```
rk_mpi_venc_test
```

### 4.1.3 Parameters

```
-i, --input=<str>           # Input file name
-o, --output=<str>          # Encoding output directory
-n, --loop_count=<int>      # Number of loop test, default value is 1
-w, --width=<int>           # The width of input image <required>
-h, --height=<int>          # The height of input image <required>
--vir_width=<int>           # Virtual width
--vir_height=<int>          # Virtual height
-f, --pixel_format=<int>    # The pixel format of input image, default value is
0 (0: NV12)

                                # For detailed format introduction, please refer to
<SDK>/external/rockit/mpi/sdk/include/rk_comm_video.h
                                # Definition of PIXEL_FORMAT_E, the same as the
following parameters
-C, --codec=<int>           # VENC encoder (8: h264, 9: mjpeg, 12: h265, 15:
jpeg, ...). Default value is 8
--channel_index=<int>       # VENC channel index, default value is 0
--enc_buf_cnt=<int>         # VENC encoding output buffer count, default value
is 8
--crop=<int>                # Crop type (0: none, 1: crop_only, 2: crop_scale).
Default value is 0
-g, --gop_mode=<int>        # GOP mode (0/1: NORMALP 2: TSVC2 3: TSVC3 4: TSVC4
5: SMARTP). Default value is 0
-s, --snap_pic_cnt=<int>    # Frame rate of the output bitstream
--rotation=<int>            # Image rotation output (0: 0, 1: 90, 2: 180, 3:
270). Default value is 0
--compress_mode=<int>       # Set input compression mode (0: MODE_NONE, 1:
AFBC_16x16). Default value is 0
--rc_mode=<int>             # RC mode, default value is 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>        # Bit rate in kbps (range for h264/h265: 3-200000,
range for jpeg/mjpeg: 5-800000, default value is 10*1024kb)
--gop_size=<int>            # Size of the GOP (range >= 1, default value: 60)
--full_range=<int>          # Set color range (0: limit color range, 1: full
color range), default value is 1
--slice_split=<int>         # Slice split testing (0: disabled, 1: enabled),
default value is 0
```

## 4.2 VPSS Module

### 4.2.1 Introduction

The Video Process Sub-System (VPSS) supports various image processing functions, including Crop, Scale, Pixel format conversion, Fixed-angle Rotation, Cover/Coverex, Mirror/Flip, Compression, Decompression, etc.

### 4.2.2 Testing Command

```
rk_mpi_vpss_test
```

### 4.2.3 Parameters

<code>-i, --input=&lt;str&gt;</code>	# Input file name, default value is NULL
<code>-o, --output=&lt;str&gt;</code>	# Output file path, default value is NULL
<code>-n, --loop_count=&lt;int&gt;</code>	# Number of loop test, default value: 1
<code>--video_proc_dev_type=&lt;int&gt;</code>	# Device type for video processing (0: GPU, 1: RGA), default value: 0
<code>-g, --group_count=&lt;int&gt;</code>	# Count of vpss groups, default value is 1
<code>-c, --channel_count=&lt;int&gt;</code>	# Number of vpss channels, default value is 1
<code>--group_crop_en</code>	# vpss group cropping is enabled, default value is 0
<code>--channel_crop_en</code>	# vpss channel cropping is enabled, default value is 0
<code>--group_crop_ratio=&lt;int&gt;</code>	# vpss group cropping ratio, range: 1-1000, default value is 1000
<code>--channel_crop_ratio=&lt;int&gt;</code>	# vpss channel cropping ratio, range: 1-1000, default value is 1000
<code>-w, --src_width=&lt;int&gt;</code>	# Width of the source bitmap <required>
<code>-h, --src_height=&lt;int&gt;</code>	# Height of the source bitmap <required>
<code>--src_vir_width=&lt;int&gt;</code>	# Virtual width of the source bitmap, default value is 0
<code>--src_vir_height=&lt;int&gt;</code>	# Virtual height of the source bitmap, default value is 0
<code>-m, --src_compress=&lt;int&gt;</code>	# Compression mode of the source bitmap, default value is 0
<code>-f, --src_format=&lt;int&gt;</code>	# Pixel format of the source bitmap, default value is 0 (NV12)
<code>-W, --dst_width=&lt;int&gt;</code>	# Width of the destination bitmap <required>
<code>-H, --dst_height=&lt;int&gt;</code>	# Height of the destination bitmap <required>
<code>-M, --dst_compress=&lt;int&gt;</code>	# Compression mode of the destination bitmap, default value is 0
<code>-F, --dst_format=&lt;int&gt;</code>	# Pixel format of the destination bitmap, default value is 0 (NV12)
<code>-r, --rotation=&lt;int&gt;</code>	# Fixed-angle rotation (0: 0, 1: 90, 2: 180, 3: 270)
<code>--grp_rotation=&lt;int&gt;</code>	# Fixed rotation angle for the vpss group, default value is 0 (0: 0, 1: 90, 2: 180, 3: 270)
<code>-R, --rotation_ex=&lt;int&gt;</code>	# Arbitrary rotation angle, default value is 0

```

-a, --attach_mb_pool=<int>          # Whether to use a shared memory pool, default
value is 0 (0: RK_FALSE, 1: RK_TRUE)
--chn_mode=<int>                    # Channel mode, default value is 0 (0: USER, 1:
AUTO, 2: PASS-THOUGH)
-d, --chn_depth=<int>              # Depth of channel output, default value: 8
--mirror=<int>                      # Image mirror operation, default value: 0
--flip=<int>                        # Image flip operation (0: disabled, 1:
enabled)
--src_chn_rate=<int>                # Source bitmap channel frame rate control,
default value is -1
--dst_chn_rate=<int>                # Destination bitmap channel frame rate
control, default value is -1
--src_grp_rate=<int>                # Source bitmap vpss group frame rate control,
default value is -1
--dst_grp_rate=<int>                # Destination bitmap vpss group frame rate
control, default value is -1

```

## 4.3 ADEC Module

### 4.3.1 Introduction

The audio decoding module (ADEC) provides audio decoding functionality for formats such as g711a, g711u, g722, g726, etc. It also supports external registered decoders.

### 4.3.2 Testing Command

```
rk_mpi_adec_test
```

### 4.3.3 Parameters

```

-i, --input=<str>                  # Input file name, e.g., (./*.mp3) <required>
-C, --codec=<str>                  # Codec, e.g., (mp2/g711a/g711u/g726) <required>
--input_ch=<int>                   # Number of input stream channels <required>
--input_rate=<int>                 # Sample rate of the input stream <required>
-o, --output=<str>                 # Output file name, e.g., (./*.pcm), no default
value
-n, --loop_count=<int>             # Number of loop test, default value is 1
-c, --channel_count=<int>          # Number of adec channels, default value is 1
--dec_mode=<int>                   # Audio stream decoding mode (0: pack mode, 1:
stream mode), default value is 0
--query_stat=<int>                 # Query adec statistics information (0: query, 1:
do not query), default value is 0
--clr_buf=<int>                    # Clear channel buffer (0: disabled, 1: enabled),
default value is 0

```



## 4.4 AENC Module

### 4.4.1 Introduction

The audio encoding module (AENC ) provides audio encoding functionality for formats such as g711a, g711u, g722, g726, etc. It also supports external registered encoders.

### 4.4.2 Testing Command

```
rk_mpi_aenc_test
```

### 4.4.3 Parameters

```
-i, --input=<str>           # Input file name, e.g., (./*.mp3) <required>
-C, --codec=<str>          # Codec, e.g., (mp3/aac/flac/mp2/g722/g726)
                             <required>
--input_ch=<int>            # Number of input stream channels <required>
--input_rate=<int>          # Sample rate of the input stream <required>
--input_format=<int>        # Input format of the input stream <required>
-o, --output=<str>          # Output file name, e.g., (./*.pcm), no default
                             value
-n, --loop_count=<int>      # Number of loop test, default value is 1
-c, --channel_count=<int>   # Number of aenc channels, default value is 1
--frame_size=<int>          # Size of the transmitted frame, default value is
                             1024
```

## 4.5 AO Module

### 4.5.1 Introduction

The audio output (AO) module implements audio output functionality through control of the audio interface of RK SoC.

### 4.5.2 Testing Command

```
rk_mpi_ao_test
```

### 4.5.3 Parameters

```
-i, --input=<str>           # Input file name, e.g., (./*.pcm) <required>
--input_ch=<int>            # Number of input channels <required>
--input_rate=<int>          # Sample rate of the input data <required>
```

```

--device_ch=<int>           # Number of channels of the sound card, default
value is 2
--device_rate=<int>         # Sample rate of the sound card, default value is
48000
-o, --output=<str>          # Output file name, e.g., (

./ao), no default value
-n, --loop_count=<int>      # Number of loop test, default value is 1
-c, --channel_count=<int>   # Number of ao channels, default value is 1
--bit=<int>                 # Bit width of the sound card, valid values is (8,
16, 24), default value: 16
--sound_card_name=<str>     # Name of the opened sound card, default value is
NULL
--set_volume=<int>          # Volume setting, valid range is (0, 100), default
value is 100
--set_mute=<int>            # Mute setting, 1 represents mute, 0 represents
unmute, default value is 0
--set_track_mode=<int>      # Set channel mode, default value is 0
                           # 0: normal (left and right channels are normal)
                           # 1: both_left (Data of left and right channels are
both from left channel)
                           # 2: both_right (Data of left and right channels
are both from right channel)
                           # 3: exchange (left and right channel data
exchange)
                           # 4: mix (left and right channel data mixed output)
                           # 5: left_mute (left channel is muted)
                           # 6: right_mute (right channel is muted)
                           # 7: both_mute (both left and right channels are
muted)
--get_volume=<int>          # Query volume, 0 represents no query, 1 represents
query, default value is 0
--get_mute=<int>            # Query mute status when setting mute, 1 represents
mute, 0 represents unmute, default value is 0
--get_track_mode=<int>      # Query channel mode, default value is 0
--query_stat=<int>          # Query ao statistics information, default value is
0
--pause_resume=<int>        # Pause and resume functionality, 1 represents
pause playback for 1 second, 0 represents no pause, default value is 0
--save_file=<int>           # Test ao and save file, if enabled, it must be set
as an output file, 0 represents disabled, 1 represents enabled, default value is
0
--query_file_stat=<int>     # Query file status, 0 represents no query, 1
represents query, default value is 0
--get_attr=<int>            # Get device attributes, 0 represents no getting, 1
represents getting, default value is 0

```

## 4.6 AI Module

### 4.6.1 Introduction

The audio input(AI) module enables audio input functionality by controlling the audio interface of Rockchip SoC.

## 4.6.2 Test Commands

```
rk_mpi_ai_test
```

## 4.6.3 Parameters

```
--device_rate=<int>           # Sample rate of the sound card <required>
--device_ch=<int>             # Number of channels of the sound card <required>
--out_ch=<int>                 # Number of channels for output data <required>
--out_rate=<int>               # Sample rate for output data <required>
-o, --output=<str>            # Name of the output file, e.g., (./ai), default
value is NULL
-n, --loop_count=<int>        # Number of loop test, default value is 1
-c, --channel_count=<int>     # Count of adec channels, default value is 1
--bit=<int>                   # Bit width of the sound card, 8, 16, 24 are
optional, default value is 16
--sound_card_name=<str>       # Name of the opened sound card, no default value
```

## 4.7 MB Module

### 4.7.1 Introduction

It used to implement a generalized memory interface for memory and memory pool management.

### 4.7.2 Test Commands

```
rk_mpi_mb_test
```

### 4.7.3 Parameters

```
-n, --loop=<int>              # Number of loops, default value is 1
-c, --mb_count=<int>          # Count of MBs, default value is 1
-s, --mb_size=<int>           # Size of MBs, default value is 4MB
-p, --pool_count=<int>        # Count of pools, default value is 1
-a, --pre_alloc=<int>         # Whether to pre-allocate, default value is 0 (0: no,
1: yes)
-r, --remap_mode=<int>        # Remap mode, default value is 2 (0: none, 256: no
cache, 512: cached)
-t, --alloc_type=<int>        # Allocation type, default value is 0 (0: DMA, 1:
malloc)
```

## 4.8 VGS Module

### 4.8.1 Introduction

The Video Graphics Sub-System (VGS ) primarily performs operations such as scaling, rotating, overlaying OSD, overlaying COVER, drawing lines, etc., on the input image.

### 4.8.2 Test Commands

```
rk_mpi_vgs_test
```

### 4.8.3 Parameters

```
-i, --input=<str>           # Input file name, e.g., (/userdata/1080p.nv12)
<required>
-o, --output=<str>          # Output file path, e.g., (/userdata/vgs/), no default
value
-n, --loop_count=<int>      # Number of test loops, default value is 1
-j, --job_number=<int>      # Job ID corresponding to vgs, default value is 1
-t, --task_number=<int>     # Number of tasks per job, default value is 1
--osd_file=<str>           # Path of OSD file, e.g., (/userdata/vgs/), no default
value
--task_type=<int>           # Task type corresponding to vgs, range is (1,6) (1:
scale, 2: rotate, 3: draw_line, 4: cover, 5: osd, 6: mosaic)
--task_mode=<int>           # Task mode, default value is 1 (1: each job has only
one task type, 2: all task types are on one job)
--task_array_size=<int>     # Size of array task, range is (1,100), default value
is 1
--angle=<int>               # Rotation angle, default value is 1 (0: 0, 1: 90, 2:
180, 3: 270)
--src_width=<int>           # Width of source bitmap, e.g., (1920) <required>
--src_height=<int>          # Height of source bitmap, e.g., (1080) <required>
--src_vir_width=<int>       # Virtual width of source bitmap, e.g., (1920)
--src_vir_height=<int>      # Virtual height of source bitmap, e.g., (1080)
--src_compress=<int>        # Compression mode of the source bitmap, default value
is 0
--src_format=<int>          # Pixel format of the source bitmap, default value is 0
(0 is NV12)
--dst_width=<int>           # Destination bitmap width, e.g., (1920) <required>
--dst_height=<int>          # Destination bitmap height, e.g., (1080) <required>
--dst_compress=<int>        # Compression format of the destination bitmap, default
value is 0
--dst_format=<int>          # Pixel format of the destination bitmap, default value
is 0 (0 is NV12)
--osd_width=<int>           # Width of OSD, e.g., (1920) <required on OSD>
--osd_height=<int>          # Height of OSD, e.g., (1080) <required on OSD>
--osd_compress=<int>        # Compression mode of OSD, default value is 0
--osd_format=<int>          # Pixel format of OSD, default value is 0 (0 is NV12)
--crop_x=<int>              # X-coordinate of the cropped ratio rectangle, default
value is 0
```

```
--crop_y=<int>           # Y-coordinate of the cropped ratio rectangle, default
value is 0
--crop_w=<int>           # Width of the cropped ratio rectangle, default value
is 0
--crop_h=<int>           # Height of the cropped ratio rectangle, default value
is 0
```

## 4.9 VDEC Module

### 4.9.1 Introduction

The VDEC module provides an MPI interface for hardware operation of driving video decoding and implements video decoding functionality.

### 4.9.2 Test Commands

```
rk_mpi_vdec_test
```

### 4.9.3 Parameters

```
-i, --input=<str>         # Input file name <required>
-o, --output=<str>        # Decoding output directory
-C, --codec=<int>         # Input stream decoder (8: h264, 9: mjpeg, 12:
h265, ...) <required in StreamMode>
-n, --loop_count=<int>    # Number of loop test, default value is 1
-w, --width=<int>         # Width of input source <required in StreamMode>
-h, --height=<int>        # Height of input source <required in StreamMode>
--channel_index=<int>     # VDEC channel index, default value is 0
-c, --channel_count=<int> # Count of VDEC channels, default value is 1
--dec_mode=<int>          # VDEC decoding mode, options values are (0:
StreamMode, 1: FrameMode), default value is 0
--dec_buf_cnt=<int>       # Count of VDEC decoding output buffer, default
value is 8
--compress_mode=<int>     # VDEC compression mode, default value is 0 (0:
NONE, 1: AFBC_16X16)
--en_mbpool=<int>         # Enable mb pool, default value is 0
--pixfmt=<int>            # JPEG output pixel mode, default value is 0 (0:
YUV420SP)
--en_dei=<int>            # Enable deinterlace, default value is 0
--en_colmv=<int>          # Enable colmv, default value is 1
```

## 4.10 TDE Module

### 4.10.1 Introduction

The TDE (Two Dimensional Engine) provide fast graphics processing functions by hardware RGA, including fast bitmap movement, fast color filling, fast bitmap rotation, fast bitmap scaling, bitmap format conversion, bitmap alpha overlay, and ColorKey operations.

### 4.10.2 Test Commands

```
rk_mpi_tde_test
```

### 4.10.3 Parameters

```
-i, --input=<str>           # Input file name, e.g., (/userdata/1080p.nv12)
<required>
-w, --src_width=<int>       # Width of source bitmap <required>
-h, --src_height=<int>      # Height of source bitmap <required>
--src_vir_width=<int>       # Virtual width of source bitmap
--src_vir_height=<int>      # Virtual height of source bitmap
--src_compress=<int>        # Compression mode of the source bitmap
-W, --dst_width=<int>       # Width of destination bitmap <required>
-H, --dst_height=<int>      # Height of destination bitmap <required>
--dst_compress=<int>        # Compression mode of the destination bitmap
-o, --output=<str>          # Path of output file, e.g., (/userdata/tde/), no
default value
--background=<str>         # Background file, e.g., (/userdata/tde/xxx.bin), no
default value
-n, --loop_count=<int>      # Number of loop test, default value is 1
-p, --operation=<int>       # Operation type, default value is 0
                             # 0: quick copy,
                             # 1: quick resize,
                             # 2: quick fill,
                             # 3: rotation,
                             # 4: mirror and flip,
                             # 5: colorkey
--src_rect_x=<int>          # X-coordinate of the source bitmap operation area,
default value is 0
--src_rect_y=<int>          # Y-coordinate of the source bitmap operation area,
default value is 0
--src_rect_w=<int>          # Width of the source bitmap operation area, default
value is src_width
--src_rect_h=<int>          # Height of the source bitmap operation area, default
value is src_height
--dst_rect_x=<int>          # X-coordinate of the destination bitmap operation
area, default value is 0
--dst_rect_y=<int>          # Y-coordinate of the destination bitmap operation
area, default value is 0
--dst_rect_w=<int>          # Width of the destination bitmap operation area,
default value is dst_width
--dst_rect_h=<int>          # Height of the destination bitmap operation area,
default value is dst_height
--performace=<int>         # Performance test mode, default value is 0
--proc_time=<int>          # Processing time, default value is 800
```

```

--colorkey=<int>          # Colorkey value, default value is 0
-c, --fill color=<int>    # Fill color, default value is 0
-r, --rotation=<int>      # Rotation angle, default value is 0 (0: 0, 1: 90, 2:
180, 3: 270)
-m, --mirror=<int>        # Mirror or flip, default value is 0 (0: none, 1: flip,
2: mirror, 3: both)

```

## 4.11 VO Module

### 4.11.1 Introduction

The Video Output (VO) module is used for video output management, supporting multiple VOPs and multi-layer display. It enables functions such as enabling video output devices or channels, sending video data or UI data to output channels, etc.

### 4.11.2 Test Commands

```
rk_mpi_vo_test
```

### 4.11.3 Parameters

```

-i, --input=<str>          # Input configuration file <required>
-d, --device_id=<int>      # VOP ID, e.g., (0/1), default value is 0
-l, --layer_id=<int>       # Layer ID, e.g., (0/2/4/6), default value is 0
--wbc_enable=<int>         # Enable WBC, e.g., (0), default value is 0
--wbc_bind=<int>           # Enable WBC binding, default value is 1 (0: disable,
1: enable)
--ui=<int>                 # UI, e.g., (0), default value is 0
--loopCount=<int>          # Number of loops, e.g., (0), default value is 10
--ui_alpha=<int>           # ui_alpha, e.g., (0), default value is 0
-w, --Windows=<int>        # Number of windows, e.g., [1-64], default value is 4,
maximum value is 63
--ConnectorType=<int>      # Connected type, e.g., (0: HDMI 1: EDP 2: VGA)
<required>
--layer_mode=<int>         # Layer type, e.g., (0: CURSOR 1: UI 2: Video)
<required>
--display_mode=<int>       # Display mode, e.g., (12/14), default value is 12 (12
represents 1080P60) <required>
--display0_mode=<int>      # Display mode, e.g., (12/14), default value is 12 (12
represents 1080P60) <required>
--disp_frmrt=<int>         # Display frame rate, default value is 25
--disp_frmrt_ratio=<int>   # Display frame rate ratio, e.g., (32, 16, 8, 4, 2, 1),
default value is 1
--aspect_mode=<int>        # Screen aspect ratio, e.g., (1: ratio is no change 2:
set ratio manually), default value is 1
--border_lpx=<int>         # Border attribute lpx, default value is 2
--border_rpx=<int>         # Border attribute rpx, default value is 2
--border_tpx=<int>         # Border attribute tpx, default value is 2
--border_bpx=<int>         # Border attribute bpx, default value is 2
--disp_x=<int>             # disp_x, default value is 0

```

```

--disp_y=<int>           # disp_y, default value is 0
--video_format=<int>     # Pixel format of video (0: ARGB8888 1: ABGR888 2:
RGB888 3: BGR888 4: RK_FMT_YUV420SP), default value is 4 (4 for RGB888)
--disp_width=<int>       # Destination width, e.g., (1920) <required>
--disp_height=<int>      # Destination height, e.g., (1080) <required>
--image0_width=<int>     # Width of image0, e.g., (1920) <required>
--image0_height=<int>    # Height of image0, e.g., (1080) <required>
--image1_width=<int>     # Width of image1, e.g., (1024) <required>
--image1_height=<int>    # Height of image1, e.g., (768) <required>

>
--disp0_width=<int>      # Width of destination bitmap, e.g., (1920) <required>
--disp0_height=<int>     # Height of destination bitmap, e.g., (1080) <required>
--disp1_width=<int>      # Width of destination bitmap, e.g., (1024) <required>
--disp1_height=<int>     # Height of destination bitmap, e.g., (768) <required>
--image_width=<int>      # Width of destination bitmap, e.g., (1920) <required>
--image_height=<int>     # Height of destination bitmap, e.g., (1080) <required>
--wbc_width=<int>        # Width of destination bitmap, e.g., (1920) <required>
--wbc_height=<int>       # Height of destination bitmap, e.g., (1080) <required>
--wbc_compress=<int>    # WBC compression mode, default value is 0
--wbc_format=<int>       # WBC pixel format, e.g., (0: ARGB8888 1: ABGR888 2:
RGB888 3: BGR888), default value is 0
--wbc_type=<int>         # WBC operation type, e.g., (0: dev 1: video), default
value is 1
--wbc_id=<int>           # WBC operation ID, default value is 0
--voplay=<str>           # Video playback test, default value is 0 (0: RK_FALSE,
1: RK_TRUE)
--bBorder=<str>          # Enable Border, default value is 0 (0: RK_FALSE, 1:
RK_TRUE)
--wbc_auto=<str>         # WBC auto bind, default value is 1 (0: RK_FALSE, 1:
RK_TRUE)
--screen0_chn=<int>      # Number of channels for screen0, default value is 16
--chn_display=<int>      # Channel display mode, e.g., (0: normal 1: pause 2:
step 3: speed), default value is 0
--screen1_chn=<int>      # Number of channels for screen1, default value is 4
--screen0_rows=<int>     # Rows and columns for displaying screen0, default
value is 4 (4x4)
--screen1_rows=<int>     # Rows and columns for displaying screen1, default
value is 3 (3x3)
--en_wbc=<int>           # Enable WBC, default value is 0
--en_chnPriority=<int>    # Enable channel priority, default value is 0
--wbc_src=<int>          # WBC source, default value is 1
--double_screen=<int>    # Enable double screen, default value is 1 (0: FALSE,
1: TRUE)
--Homologous=<int>       # Homologous display, default value is 0

```

## 4.12 AVS Module

### 4.12.1 Introduction

The Any View Stitching (AVS) module is responsible for panoramic image stitching from multiple video sources and outputting the stitched image in the specified projection mode.



## 4.12.2 Test Commands

```
rk_mpi_avs_test
```

## 4.12.3 Parameters

```
-m, --test_mode=<int>          # Test mode, default value is 0
                                # 0: avs module. 8xEquirectangular,
                                6xRectilinear.
                                # 1: avs module. 6xNoBlend_Hor,
                                6xNoBlend_Ver, 4xNoBlend_Qr.
                                # 2: vi -> avs -> vo. 6xEquirectangular.)
-n, --loop_count=<int>         # Number of test loops, default value is 100
-c, --link_compress_mode=<int> # Compression mode, default value is 1 (0:
Uncompress, 1: AFBC)
-p, --avs_pipe_sync=<int>      # Enable AVS pipe sync, default value is 0
(0: Disable)
--params_sources=<int>         # Parameters used for panoramic stitching
through mesh or calib, default value is 0 (0: mesh, 1: calib)
--connector_type=<int>         # Connector type, default value is 3 (0:
HDMI0, 3: MIPI)
```

## 4.13 VI Module

### 4.13.1 Introduction

The Video Input (VI) module is responsible for capturing video data through various interfaces such as MIPI Rx (including MIPI interface, LVDS interface), BT.1120, BT.656, BT.601, DC, etc. The VI module stores the captured data in the specified memory area for video data collection.

### 4.13.2 Test Commands

```
rk_mpi_vi_test
```

### 4.13.3 Parameters

```
-w, --width=<int>              # Set the width of capture channel, default value is 0
<required>
-h, --height=<int>            # Set the height of capture channel, default value is 0
<required>
-d, --dev=<int>               # Set device ID, default value is 0
-p, --pipe=<int>              # Set pipe ID, default value is 0
-c, --channel=<int>           # Set channel ID, default value is 1
-l, --loopcount=<int>         # Set capture frame count (default value is 100)
-C, --compressmode=<int>      # Set capture compression mode, default value is 0 (0:
MODE_NONE, 1: AFBC_16x16)
```

```

-o, --output=<int>          # Save output file, which is saved as
                             /data/test_<devid>_<pipeid>_<channelid>.bin, default value is 0 (0: no save, 1:
                             save)
-m, --mode=<int>            # Test mode, default value is 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>      # Set buffer memory type, default value is 4 <required>
                             # 1: mmap (hdmiin/bt1120/sensor input)
                             # 2: userptr (invalid)
                             # 3: overlay (invalid)
                             # 4: dma (sensor)
-n, --name=<str>            # Set entity name, no default value (rv1126 sensor:
                             rkispp_m_bypass rkispp_scale0 rkispp_scale1 rkispp_scale2; rv1126
                             hdmiin/bt1120/sensor: /dev/videox such as /dev/video19 /dev/video20) <required>;
                             rk356x hdmiin/bt1120/sensor: /dev/videox such as /dev/video0 /dev/video1)
-D, --depth=<int>          # Channel output depth, default value is {u32BufCount
                             (not bind) or 0 (bind venc/vpss/...)}
-f, --format=<int>          # Set format, default value is 0
                             # 0: RK_FMT_YUV420SP
                             # 10: RK_FMT_YUV422_UYVY1310
                             # 80: RK_FMT_RGB_BAYER_SBGGR_12BPP
--src_rate=<int>            # Source bitmap frame rate, default value is -1 (-1:
                             not control; other: 1-max_fps<isp_out_fps>)
--dst_rate=<int>            # Destination bitmap frame rate, default value is -1
                             (-1: not control; other: 1-src_fps<src_rate>)
-U, --user_pic=<int>        # Enable using user-specified picture as vi input
--rgn_type=<int>            # rgn type (0: overlay, 1: overlayEx, 2: cover, 3:
                             coverEx, 4: mosaic, 5: mosaicEx)
--rgn_cnt=<int>             # rgn count, default value is 1, maximum value is 8
--en_rgn=<int>              # Enable region, default value is 0
--get_connect_info=<int>    # Get connection information, default value is 0
--get_edid=<int>            # Get edid, default value is 0
--set_edid=<int>            # Set edid default value is 0

```

## 4.14 RGN Module

### 4.14.1 Introduction

Users usually require overlay OSD on videos to display specific information such as channel number, timestamp, etc. When necessary, they may also fill color blocks. These OSDs overlaid on the video and color blocks obstructing the video are collectively referred to as regions. The REGION module is used to manage these region resources.

### 4.14.2 Test Command

```
rk_mpi_rgn_test
```

### 4.14.3 Parameters

```
-i, --input_raw_name=<str>      # The file name of input raw data, default value is
RK_NULL
--input_bmp_name=<str>          # The file name of input bmp data <required>
-o, --output_name=<str>         # The file name of output stream, default value is
RK_NULL
-r, --rgn_count=<int>           # Number of rgn handles, default value is 1
-p, --operation=<int>           # Rgn operation, default value is 0 (0: overlay, 1:
cover, 2: mosaic, 3: line)
-x, --rect_x=<int>              # Starting x-coordinate of rgn area, default value
is 0
-y, --rect_y=<int>              # Starting y-coordinate of rgn area, default value
is 0
-w, --bmp_w=<int>               # Width of bmp, default value is 0 <required>
-h, --bmp_h=<int>               # Height of bmp, default value is 0 <required>
-W, --raw_w=<int>               # Raw width, default value is 0 <required>
-H, --raw_h=<int>               # Raw height, default value is 0 <required>
-F, --raw_fmt=<int>             # Raw pixel format, default value is 0 (0: NV12)
-f, --format=<int>              # Bmp pixel format, default value is 65557 (65546:
ARGB1555, 65557: BGRA5551)
--mod=<int>                     # Additional module, default value is 4 (4: VENC,
6: VPSS)
--cover_type=<int>              # Cover type, default value is 0 (0: rect, 1: quad)
```

## 4.15 SYS Module

### 4.15.1 Introduction

The System Control module implements the RK MPI common functional interfaces, providing system-related functions and large block physical memory management.

### 4.15.2 Test Command

```
rk_mpi_sys_test
```

### 4.15.3 Parameter

```
-n, --loop_count=<int>          # Number of loop test, default value is 1
--device_id=<int>               # ID of the module device, default value is 0
--src_channel_id=<int>          # Source module channel ID, default value is 0
--dst_channel_count=<int>       # Count of destination module channels, default
value is 1
```

