

Rockchip RKMedia Development Guide

File identification: RK-KF-YF-382

Release version: V1.0.0

Date: 2020-09-03

File Confidentiality: ☐ Top Secret ☐ Secret ☐ Internal Information ☒ Public

Disclaimer

This document is provided "as is". Rockchip Microelectronics Co., Ltd. ("the company", the same below) does not give any statement, information and content The accuracy, reliability, completeness, merchantability, specific purpose and non-infringement of the content provide any express or implied statement or guarantee. this The document is only used as a reference for instructions.

Due to product version upgrades or other reasons, this document may be updated or modified from time to time without any notice.

Trademark statement

"Rockchip", "Rockchip" and "Rockchip" are all registered trademarks of our company and are owned by our company.

All other registered trademarks or trademarks that may be mentioned in this document are owned by their respective owners.

Copyright © 2020 Rockchip Microelectronics Co., Ltd.

Beyond the scope of fair use, without the written permission of the company, any unit or individual shall not extract or copy part or all of the content of this document without authorization. Ministry, not spread in any form.

Rockchip Microelectronics Co., Ltd.

Rockchip Electronics Co., Ltd.

Address: No. 18, Area A, Software Park, Tongpan Road, Fuzhou City, Fujian Province

URL: www.rock-chips.com

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

Customer Service Fax: +86-591-83951833

Customer Service Email: fae@rock-chips.com

This article mainly describes the RKMedia media development reference.

product version

Chip name	Kernel version
RK1126/RK1109	Linux V4.19

Audience

This document (this guide) is mainly applicable to the following engineers:

Technical Support Engineer

Software Development Engineer

Revision record

version number	Author	Modified date	Modify the description
V0.0.1	Fan Lichuang/Yu Yongzhen	2020-08-31	initial version
V1.0.0	Lin Liu Diming	2020-09-03	Add data type and error code, associated link

table of Contents

[Rockchip RKMedia Development Guide](#)

- [1. System Overview](#)
 - [1.1 Overview](#)
 - [1.2 System architecture](#)
 - [1.3 Number of system resources table](#)
- [2. System Control](#)
 - [2.1 Overview](#)
 - [2.2 Function description](#)
 - [2.2.1 System binding](#)
 - [2.3 API reference](#)
 - [2.3.1 RK_MPI_SYS_Init](#)
 - [2.3.2 RK_MPI_SYS_DumpChn](#)
 - [2.3.3 RK_MPI_SYS_Bind](#)
 - [2.3.4 RK_MPI_SYS_UnBind](#)
 - [2.3.5 RK_MPI_SYS_RegisterEventCb](#)
 - [2.3.6 RK_MPI_SYS_RegisterOutCb](#)

[2.3.7 RK_MPI_SYS_SendMediaBuffer](#)
[2.3.8 RK_MPI_SYS_GetMediaBuffer](#)
[2.3.9 RK_MPI_MB_ReleaseBuffer](#)
[2.3.10 RK_MPI_MB_GetPtr](#)
[2.3.11 RK_MPI_MB_GetFD](#)
[2.3.12 RK_MPI_MB_GetSize](#)
[2.3.13 RK_MPI_MB_GetModelID](#)
[2.3.14 RK_MPI_MB_GetChannelID](#)
[2.3.15 RK_MPI_MB_GetTimestamp](#)

[2.4 Data type](#)

[2.4.1 Basic data types](#)

[2.4.1.1 Common data types](#)
[2.4.1.2 IMAGE_TYPE_E](#)
[2.4.1.3 CODEC_TYPE_E](#)
[2.4.1.4 MOD_ID_E](#)
[2.4.1.5 Sample_Format_E](#)
[2.4.1.6 RECT_S](#)

[2.4.2 System control data type](#)

[2.4.2.1 MPP_CHN_S](#)
[2.4.2.2 EventCbFunc](#)
[2.4.2.3 MEDIA_BUFFER](#)
[2.4.2.4 OutCbFunc](#)
[2.4.2.5 MB_IMAGE_INFO_S](#)

[2.5 Error code](#)

[3. Video input](#)

[3.1 Overview](#)

[3.2 Function description](#)

[3.2.1 VI node name](#)
[3.2.2 VI working mode](#)

[3.3 API reference](#)

[3.3.1 RK_MPI_VI_EnableChn](#)
[3.3.2 RK_MPI_VI_DisableChn](#)
[3.3.3 RK_MPI_VI_SetChnAttr](#)
[3.3.4 RK_MPI_VI_GetChnRegionLuma](#)
[3.3.5 RK_MPI_VI_StartStream](#)

[3.4 Data type](#)

[3.4.1 VI_MAX_DEV_NUM](#)
[3.4.2 VI_MAX_CHN_NUM](#)
[3.4.3 VI_PIPE](#)

[3.4.4 VI_CHN](#)

[3.4.5 VI_CHN_ATTR_S](#)

[3.4.6 VIDEO_REGION_INFO_S](#)

[3.5 Error code](#)

[4. Video encoding](#)

[4.1 Overview](#)

[4.2 Function description](#)

[4.2.1 Data flow chart](#)
[4.2.2 Rate control](#)
[4.2.3 GOP Mode](#)
[4.2.4 Region of Interest \(ROI\)](#)
[4.2.5 Rotation](#)

[4.3 API reference](#)

[4.3.1 RK_MPI_VENC_CreateChn](#)
[4.3.2 RK_MPI_VENC_DestroyChn](#)
[4.3.3 RK_MPI_VENC_SetRcParam](#)
[4.3.4 RK_MPI_VENC_SetRcMode](#)
[4.3.5 RK_MPI_VENC_SetRcQuality](#)
[4.3.6 RK_MPI_VENC_SetBitrate](#)
[4.3.7 RK_MPI_VENC_RequestIDR](#)
[4.3.8 RK_MPI_VENC_SetEps](#)
[4.3.9 RK_MPI_VENC_SetGop](#)
[4.3.10 RK_MPI_VENC_SetAvcProfile](#)
[4.3.11 RK_MPI_VENC_InsertUserData](#)
[4.3.12 RK_MPI_VENC_SetRoiAttr](#)
[4.3.13 RK_MPI_VENC_SetGopMode](#)
[4.3.14 RK_MPI_VENC_RGN_Init](#)
[4.3.15 RK_MPI_VENC_RGN_SetBitMap](#)
[4.3.16 RK_MPI_VENC_RGN_SetCover](#)
[4.3.17 RK_MPI_VENC_SetJpegParam](#)
[4.3.18 RK_MPI_VENC_StartRecvFrame](#)

[4.4 Data Type](#)

[4.4.1 VENC_MAX_CHN_NUM](#)
[4.4.2 VENC_CHN](#)

[4.4.3 VENC_ATTR_JPEG_S](#)
[4.4.4 VENC_ATTR_MJPEG_S](#)
[4.4.5 VENC_ATTR_H264_S](#)
[4.4.6 VENC_ATTR_H265_S](#)
[4.4.7 VENC_ATTR_S](#)
[4.4.8 VENC_MJPEG_CBR_S](#)
[4.4.9 VENC_MJPEG_VBR_S](#)
[4.4.10 VENC_H264_CBR_S](#)
[4.4.11 VENC_H264_VBR_S](#)
[4.4.12 VENC_H265_CBR_S](#)
[4.4.13 VENC_H265_VBR_S](#)
[4.4.14 VENC_RC_MODE_E](#)
[4.4.15 VENC_RC_ATTR_S](#)
[4.4.16 VENC_GOP_MODE_E](#)
[4.4.17 VENC_GOP_ATTR_S](#)
[4.4.18 VENC_CHN_ATTR_S](#)
[4.4.19 VENC_PARAM_MJPEG_S](#)
[4.4.20 VENC_PARAM_H264_S](#)
[4.4.21 VENC_PARAM_H265_S](#)
[4.4.22 VENC_RC_PARAM_S](#)
[4.4.23 VENC_RC_QUALITY_E](#)
[4.4.24 VENC_ROI_ATTR_S](#)
[4.4.25 OSD_REGION_ID_E](#)
[4.4.26 OSD_REGION_INFO_S](#)

Page 5

[4.4.27 OSD_PIXEL_FORMAT_E](#)
[4.4.28 BITMAP_S](#)
[4.4.29 COVER_INFO_S](#)
[4.4.30 VENC_RECV_PIC_PARAM_S](#)
[4.4.31 VENC_JPEG_PARAM_S](#)

[4.5 Error code](#)

[5. Motion detection](#)

[5.1 Overview](#)

[5.2 Function description](#)

[5.3 API reference](#)

[5.3.1 RK_MPI_ALGO_MD_CreateChn](#)
[5.3.2 RK_MPI_ALGO_MD_DestroyChn](#)

[5.4 Data Type](#)

[5.4.1 ALGO_MD_MAX_CHN_NUM](#)
[5.4.2 ALGO_MD_ROI_RET_MAX](#)
[5.4.3 ALGO_MD_CHN](#)
[5.4.4 ALGO_MD_ATTR_S](#)

[5.5 Error code](#)

[6. Occlusion detection](#)

[6.1 Overview](#)

[6.2 Function description](#)

[6.3 API reference](#)

[6.3.1 RK_MPI_ALGO_OD_CreateChn](#)
[6.3.2 RK_MPI_ALGO_OD_DestroyChn](#)

[6.4 Data Type](#)

[6.4.1 ALGO_OD_MAX_CHN_NUM](#)
[6.4.2 ALGO_OD_ROI_RET_MAX](#)
[6.4.3 ALGO_OD_CHN](#)
[6.4.4 ALGO_OD_ATTR_S](#)

[6.5 Error code](#)

[7. Audio](#)

[7.1 Overview](#)

[7.2 Function description](#)

[7.2.1 Audio input and output](#)
[7.2.2 Audio Codec](#)
[7.2.3 Audio Algorithm](#)

[7.3 API reference](#)

[7.3.1 Audio input](#)
[7.3.1.1 RK_MPI_AI_EnableChn](#)
[7.3.1.2 RK_MPI_AI_DisableChn](#)
[7.3.1.3 RK_MPI_AI_SetChnAttr](#)
[7.3.1.4 RK_MPI_AI_SetVolume](#)
[7.3.1.5 RK_MPI_AI_GetVolume](#)
[7.3.1.6 RK_MPI_AI_SetTalkVgeAttr](#)
[7.3.1.7 RK_MPI_AI_GetTalkVgeAttr](#)
[7.3.1.8 RK_MPI_AI_SetRecordVgeAttr](#)
[7.3.1.9 RK_MPI_AI_GetRecordVgeAttr](#)
[7.3.1.10 RK_MPI_AI_EnableVge](#)
[7.3.1.11 RK_MPI_AI_DisableVge](#)

[7.3.2 Audio output](#)
[7.3.2.1 RK_MPI_AO_EnableChn](#)
[7.3.2.2 RK_MPI_AO_DisableChn](#)
[7.3.2.3 RK_MPI_AO_SetChnAttr](#)
[7.3.2.4 RK_MPI_AO_SetVolume](#)
[7.3.2.5 RK_MPI_AO_GetVolume](#)
[7.3.2.6 RK_MPI_AO_SetVgeAttr](#)
[7.3.2.7 RK_MPI_AO_GetVgeAttr](#)
[7.3.2.8 RK_MPI_AO_EnableVge](#)

[7.3.2.9 RK_MPI_AO_DisableVge](#)

[7.3.3 Audio coding](#)

[7.3.3.1 RK_MPI_AENC_CreateChn](#)
[7.3.3.2 RK_MPI_AENC_DestroyChn](#)

[7.3.4 Audio decoding](#)

[7.3.4.1 RK_MPI_ADEC_CreateChn](#)
[7.3.4.2 RK_MPI_ADEC_DestroyChn](#)

[7.4 Data Type](#)

[7.4.1 Audio input](#)

[7.4.1.1 AI_MAX_CHN_NUM](#)
[7.4.1.2 AI_CHN](#)
[7.4.1.3 AI_CHN_ATTR_S](#)
[7.4.1.4 AI_TALKVOE_CONFIG_S](#)
[7.4.1.5 AI_RECORDVOE_CONFIG_S](#)

[7.4.2 Audio output](#)

[7.4.2.1 AO_MAX_CHN_NUM](#)
[7.4.2.2 AO_CHN](#)
[7.4.2.3 AO_CHN_ATTR_S](#)
[7.4.2.4 AO_VOE_CONFIG_S](#)

[7.4.3 Audio coding](#)

[7.4.3.1 AENC_MAX_CHN_NUM](#)
[7.4.3.2 AENC_CHN](#)
[7.4.3.3 AENC_ATTR_AAC_S](#)
[7.4.3.4 AENC_ATTR_MP2_S](#)
[7.4.3.5 AENC_ATTR_G711A_S](#)
[7.4.3.6 AENC_ATTR_G711U_S](#)
[7.4.3.7 AENC_ATTR_G726_S](#)
[7.4.3.8 AENC_CHN_ATTR_S](#)

[7.4.4 Audio decoding](#)

[7.4.4.1 ADEC_MAX_CHN_NUM](#)
[7.4.4.2 ADEC_CHN](#)
[7.4.4.3 ADEC_ATTR_AAC_S](#)
[7.4.4.4 ADEC_ATTR_MP2_S](#)
[7.4.4.5 ADEC_ATTR_G711A_S](#)
[7.4.4.6 ADEC_ATTR_G711U_S](#)
[7.4.4.7 ADEC_ATTR_G726_S](#)
[7.4.4.8 ADEC_CHN_ATTR_S](#)

[7.5 Error code](#)

[7.5.1 Audio input error code](#)
[7.5.2 Audio output error code](#)
[7.5.3 Audio encoding error code](#)
[7.5.4 Audio decoding error code](#)

[8. RGA](#)

[8.1 Overview](#)

[8.2 Function description](#)

[8.3 API reference](#)

[8.3.1 RK_MPI_RGA_CreateChn](#)
[8.3.2 RK_MPI_RGA_DestroyChn](#)

[8.4 Data Type](#)

[8.4.1 RGA_MAX_CHN_NUM](#)
[8.4.2 RGA_CHN](#)
[8.4.3 RGA_INFO_S](#)
[8.4.4 RGA_ATTR_S](#)

[8.5 Error code](#)

[9. Video output](#)

[9.1 Overview](#)

[9.2 Function description](#)

[9.3 API reference](#)

[9.3.1 RK_MPI_VO_CreateChn](#)
[9.3.2 RK_MPI_VO_DestroyChn](#)

[9.4 Data Type](#)[9.4.1 VO_MAX_CHN_NUM](#)[9.4.2 VO_CHN](#)[9.4.3 VO_CHN_ATTR_S](#)[9.5 Error code](#)[10. Restrictions on Use](#)

1. System Overview

1.1 Overview

RKMedia provides a media processing solution that supports the rapid development of application software. RKMedia takes further steps on the basic API of each module. Packaging simplifies the difficulty of application development. The platform supports the following functions: VI (input video capture), VENC (H.265/H.264/JPEG/MJPEG encoding), VDEC (H.265/H.264/JPEG, MJPEG decoding), VO (video output display), RGA video processing (including rotation, scaling, cropping), AI (audio capture), AO (audio output), AENC (audio coding), ADEC (audio decoding), MD (motion detection), OD (occlusion detection).

1.2 System Architecture

1.3 Number of system resources table

Page 9

Module name	Number of channels
VI	4
VENC	16
VDEC	16
AI	1
AO	1
AENC	16
ADEC	16
MD	4
OD	4
RGA	16
VO	2

2. System control

2.1 Overview

Binding relationship, provide current system version, system log management.

2.2 Function description

2.2.1 System binding

RKMedia provides system binding interface ([RK_MPI_SYS_Bind](#)), that is, to establish the relationship between the two by binding the data source to the data receiver Link (only allow the data receiver to bind the data source). After binding, the data generated by the data source will be automatically sent to the receiver. Currently supported bindings The relationship is shown in [Table 2-1](#)Shown.

Table 2-1 Binding relationship supported by RKMedia

data source	Data recipient
VI	VO/RGA/VENC/MD/OD
VDEC	VO/RGA/VENC/MD/OD
RGA	VO/VENC/MD/OD
AI	AO/AENC
ADEC	AO

2.3 API reference

2.3.1 RK_MPI_SYS_Init

【description】

Initialize the system.

【grammar】

RK_S32 RK_MPI_SYS_Init();

【parameter】

no.

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

2.3.2 RK_MPI_SYS_DumpChn

【description】

Print channel information.

【grammar】

RK_VOID RK_MPI_SYS_DumpChn([MOD_ID_E](#) enModId);

【parameter】

parameter name	description	Input / output
enModId	Module number.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

2.3.3 RK_MPI_SYS_Bind

【description】

Data source to data receiver binding interface.

【grammar】

RK_S32 RK_MPI_SYS_Bind(const [MPP_CHN_S](#) *pstSrcChn,const[MPP_CHN_S](#) *pstDestChn);

【parameter】

parameter name	description	Input / output
pstSrcChn	Source channel pointer.	enter
pstDestChn	Destination channel pointer.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

Binding relationship is currently supported systems, see [Table 2-1](#).

If you use this function, you cannot use[RK_MPI_MB_ReleaseBuffer](#)retrieve data.

【Example】

no.

【related topic】

[RK_MPI_SYS_UnBind](#)

2.3.4 RK_MPI_SYS_UnBind

【description】

Data source to data receiver unbinding interface.

【grammar】

RK_MPI_SYS_UnBind(const [MPP_CHN_S](#) *pstSrcChn,const[MPP_CHN_S](#) *pstDestChn);

【parameter】

parameter name	description	Input / output
pstSrcChn	Source channel pointer.	enter
pstDestChn	Destination channel pointer.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_SYS_Bind](#)

2.3.5 RK_MPI_SYS_RegisterEventCb

【description】

Register event callbacks, such as motion detection events.

【grammar】

RK_S32 RK_MPI_SYS_RegisterEventCb(const [MPP_CHN_S](#) *pstChn,[EventCbFunc](#) cb);

【parameter】

parameter name	description	Input / output
pstChn	Specify the channel pointer.	enter
cb	Event callback function.	Output

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

2.3.6 RK_MPI_SYS_RegisterOutCb

【description】

Register data output callback.

【description】

Register data output callback. Note: The callback function cannot handle time-consuming operations, otherwise the data stream of the corresponding channel will be blocked.

【grammar】

RK_S32 RK_MPI_SYS_RegisterOutCb(const [MPP_CHN_S](#) *pstChn,[OutCbFunc](#) cb);

【parameter】

parameter name	description	Input / output
pstChn	Specify the channel pointer.	enter
cb	Data output callback function.	Output

【return value】

return value	description
0	success.
Non-zero	Failure, see the error code for its value .

【demand】
Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

The callback function cannot handle time-consuming operations, otherwise the data flow of the corresponding channel will be blocked.

【Example】

no.

【related topic】

no.

2.3.7 RK_MPI_SYS_SendMediaBuffer

【description】

Input data to the specified channel, such as sending the local yuv file to the encoder for encoding.

【grammar】

RK_S32 RK_MPI_SYS_SendMediaBuffer([MOD_ID_E](#) enModID, RK_S32 s32ChnID,[MEDIA_BUFFER](#) buffer);

【parameter】

Page 15

parameter name	description	Input / output
enModID	Module number.	enter
s32ChnID	Channel number.	enter
buffer	Buffer.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see the error code for its value .

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_SYS_GetMediaBuffer](#)

2.3.8 RK_MPI_SYS_GetMediaBuffer

【description】

Obtain data from the specified channel.

【grammar】

[MEDIA_BUFFER](#) RK_MPI_SYS_GetMediaBuffer(MOD_ID_E enModID, RK_S32 s32ChnID, RK_S32 s32MilliSec);

【parameter】

parameter name	description	Input / output
enModID	Module number.	enter
s32ChnID	Channel number.	enter
s32MilliSec	Block waiting time.	enter
【return value】		

Page 16

Return value type	description
MEDIA_BUFFER	Buffer pointer.
【demand】	
Header file: rkmedia_api.h	
Library file: libeasymedia.so	
【note】	
If used RK_MPI_SYS_Bind , this function cannot get data.	
【Example】	
no.	
【related topic】	
RK_MPI_SYS_SendMediaBuffer	
RK_MPI_MB_ReleaseBuffer	

2.3.9 RK_MPI_MB_ReleaseBuffer

【description】	
Release the buffer.	
【grammar】	
RK_S32 RK_MPI_MB_ReleaseBuffer(MEDIA_BUFFER mb);	
【parameter】	

parameter name	description	Input / output
mb	Buffer.	enter
【return value】		
return value	description	
0	success.	
Non-zero	Failure, see the error code for its value .	

【demand】	
Header file: rkmedia_api.h	
Library file: libeasymedia.so	
【note】	
no.	
【Example】	

no.

【related topic】

[RK_MPI_SYS_GetMediaBuffer](#)

2.3.10 RK_MPI_MB_GetPtr

【description】

Obtain the buffer pointer from the specified [MEDIA_BUFFER](#) .

【grammar】

void *RK_MPI_MB_GetPtr([MEDIA_BUFFER](#) mb);

【parameter】

parameter name	description	Input / output
mb	Buffer.	enter

【return value】

Return value type	description
void *	Buffer pointer.

【demand】

Header file: rkmedia_buffer.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

2.3.11 RK_MPI_MB_GetFD

【description】

Obtain the file descriptor from the specified [MEDIA_BUFFER](#) .

【grammar】

int RK_MPI_MB_GetFD([MEDIA_BUFFER](#) mb);

【parameter】

parameter name	description	Input / output
mb	Buffer.	enter

【return value】

Return value type	description
int	File descriptor.

【demand】

Header file: rkmedia_buffer.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

2.3.12 RK_MPI_MB_GetSize

【description】

Obtain the buffer size from the specified [MEDIA_BUFFER](#) .

【grammar】

size_t RK_MPI_MB_GetSize([MEDIA_BUFFER](#) mb);

【parameter】

parameter name	description	Input / output
mb	Buffer.	enter

【return value】

Return value type	description
size_t	The buffer size.

【demand】

Header file: rkmedia_buffer.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

2.3.13 RK_MPI_MB_GetModeID

【description】

Obtain the module ID from the specified [MEDIA_BUFFER](#) .

【grammar】

[MOD_ID_E](#) RK_MPI_MB_GetModeID([MEDIA_BUFFER](#) mb);

【parameter】

parameter name	description	Input / output
mb	Buffer.	enter

【return value】

Return value type	description
-------------------	-------------

[MOD_ID_E](#)

Module ID.

【demand】

Header file: rkmedia_buffer.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

2.3.14 RK_MPI_MB_GetChannelID

【description】

Obtain the channel ID from the specified [MEDIA_BUFFER](#) .

【grammar】

RK_S16 RK_MPI_MB_GetChannelID([MEDIA_BUFFER](#) mb);

【parameter】

Page 20

parameter name	description	Input / output
mb	Buffer.	enter

【return value】

Return value type	description
RK_S16	Channel ID.

【demand】

Header file: rkmedia_buffer.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

2.3.15 RK_MPI_MB_GetTimestamp

【description】

Get the timestamp from the specified [MEDIA_BUFFER](#) .

【grammar】

RK_U64 RK_MPI_MB_GetTimestamp([MEDIA_BUFFER](#) mb);

【parameter】

parameter name	description	Input / output
mb	Buffer.	enter

【return value】		
Return value type		description
RK_S16		Timestamp.

【demand】

Header file: rkmedia_buffer.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

2.4 data types

2.4.1 Basic data types

The basic data types are defined as follows:

2.4.1.1 Common data types

```
typedef unsigned char RK_U8;
typedef unsigned short RK_U16;
typedef unsigned int RK_U32;

typedef signed char RK_S8;
typedef short RK_S16;
typedef int RK_S32;

typedef unsigned long RK_UL;
typedef signed long RK_SL;

typedef float RK_FLOAT;
typedef double RK_DOUBLE;

#ifdef _M_IX86
typedef unsigned long long RK_U64;
typedef long long RK_S64;
#else
typedef unsigned __int64 RK_U64;
typedef __int64 RK_S64;
#endif

typedef char RK_CHAR;
#define RK_VOID void

typedef unsigned int RK_HANDLE;

/*-----*
 * const definition
 *-----*/

typedef enum {
    RK_FALSE = 0 ,
    RK_TRUE = 1 ,
} RK_BOOL;

#ifdef NULL
#define NULL 0L
#endif
```

```
#define RK_NULL 0L
#define RK_SUCCESS 0
#define RK_FAILURE (-1)

#define MAX_FILE_PATH_LEN 256
```

2.4.1.2 IMAGE_TYPE_E

【Description】

Define the image format enumeration type.

【definition】

```
typedef enum rk_IMAGE_TYPE_E {
    IMAGE_TYPE_UNKNOW = 0 ,
    IMAGE_TYPE_GRAY8,
    IMAGE_TYPE_GRAY16,
    IMAGE_TYPE_YUV420P,
    IMAGE_TYPE_NV12,
    IMAGE_TYPE_NV21,
    IMAGE_TYPE_YV12,
    IMAGE_TYPE_FBC2,
    IMAGE_TYPE_FBC0,
    IMAGE_TYPE_YUV422P,
    IMAGE_TYPE_NV16,
    IMAGE_TYPE_NV61,
    IMAGE_TYPE_YV16,
    IMAGE_TYPE_YUYV422,
    IMAGE_TYPE_UYVY422,
    IMAGE_TYPE_RGB332,
    IMAGE_TYPE_RGB565,
    IMAGE_TYPE_BGR565,
    IMAGE_TYPE_RGB888,
    IMAGE_TYPE_BGR888,
    IMAGE_TYPE_ARGB8888,
    IMAGE_TYPE_ABGR8888,
    IMAGE_TYPE_JPEG,

    IMAGE_TYPE_BUTT
} IMAGE_TYPE_E;
```

2.4.1.3 CODEC_TYPE_E

【Description】

Define the codec format enumeration type.

【definition】

```
typedef enum rk_CODEC_TYPE_E {
    RK_CODEC_TYPE_NONE = -1 ,
    // Audio
    RK_CODEC_TYPE_AAC,
```

```
    RK_CODEC_TYPE_MP2,
    RK_CODEC_TYPE_VORBIS,
    RK_CODEC_TYPE_G711A,
    RK_CODEC_TYPE_G711U,
    RK_CODEC_TYPE_G726,
    // Video
    RK_CODEC_TYPE_H264,
    RK_CODEC_TYPE_H265,
    RK_CODEC_TYPE_JPEG,
    RK_CODEC_TYPE_MJPEG,
    RK_CODEC_TYPE_NB
} CODEC_TYPE_E;
```

2.4.1.4 MOD_ID_E**【Description】**

Define the module ID enumeration type.

【definition】

```
typedef enum rkMOD_ID_E {
    RK_ID_UNKNOW = 0 ,
    RK_ID_VB,
    RK_ID_SYS,
    RK_ID_VDEC,
    RK_ID_VENC,
    RK_ID_H264E,
    RK_ID_JPEG,
    RK_ID_H265E,
    RK_ID_VO,
    RK_ID_VI,
    RK_ID_AIO,
    RK_ID_AI,
    RK_ID_AO,
    RK_ID_AENC,
    RK_ID_ADEC,
    RK_ID_ALGO_MD,
    RK_ID_ALGO_OD,
    RK_ID_RGA,

    RK_ID_BUTT,
} MOD_ID_E;
```

2.4.1.5 Sample_Format_E**【Description】**

Define the sampling format enumeration type.

【definition】

```
typedef enum rkSample_Format_E {
    RK_SAMPLE_FMT_NONE = -1 ,
    RK_SAMPLE_FMT_U8,
```

Page 24

```
    RK_SAMPLE_FMT_S16,
    RK_SAMPLE_FMT_S32,
    RK_SAMPLE_FMT_FLT,
    RK_SAMPLE_FMT_U8P,
    RK_SAMPLE_FMT_S16P,
    RK_SAMPLE_FMT_S32P,
    RK_SAMPLE_FMT_FLTP,
    RK_SAMPLE_FMT_G711A,
    RK_SAMPLE_FMT_G711U,
    RK_SAMPLE_FMT_NB
} Sample_Format_E;
```

2.4.1.6 RECT_S**【Description】**

Define the area attribute structure.

【definition】

```
typedef struct rkRECT_S {
    RK_S32 s32X;
    RK_S32 s32Y;
    RK_U32 u32Width;
```

```
RK_U32 u32Height;  
} RECT_S;
```

【member】

Member name	description
s32X	X-axis coordinate of the area
s32Y	Y-axis coordinate of the area
u32Width	The width of the area
u32Height	The height of the area

【Precautions】

no.

[Related data types and interfaces]

no.

2.4.2 System control data type

The data types related to system control are defined as follows:

[MPP_CHN_S](#): Define the channel structure of the module device.

[EventCbFunc](#): Event callback function pointer.

[MEDIA_BUFFER](#): Data buffer pointer.

[OutCbFunc](#): Data output callback function pointer.

[MB_IMAGE_INFO_S](#): Image information structure.

2.4.2.1 MPP_CHN_S

【Description】

Define the module device channel structure.

【definition】

```
typedef struct rkMPP_CHN_S {  
    MOD_ID_E enModId;  
    RK_S32 s32DevId;  
    RK_S32 s32ChnId;  
} MPP_CHN_S;
```

【member】

Member name	description
enModId	Module number.
s32DevId	Device No.
s32ChnId	Channel number.

2.4.2.2 EventCbFunc

【Description】

Event callback function pointer.

【definition】

```
typedef struct rkMD_EVENT_S {  
    RK_U16 u16Cnt;
```

```

    RK_U32 u32Width;
    RK_U32 u32Height;
    RECT_S stRects[ 4096 ];
} MD_EVENT_S;

typedef struct rkOD_EVENT_S {
    RK_U16 u16Cnt;
    RK_U32 u32Width;
    RK_U32 u32Height;
    RECT_S stRects[ 10 ];
    RK_U16 u16Occlusion[ 10 ];
} OD_EVENT_S;

typedef struct rkEVENT_S {
    EVENT_TYPE_E type;

```

```

    MOD_ID_E mode_id;
    union {
        MD_EVENT_S md_event;
        OD_EVENT_S stOdEvent;
    };
} EVENT_S;

typedef void ( * EventCbFunc)(EVENT_S * event);

```

【member】

Member name	description
type	Event type.
mode_id	Module number.
md_event	Motion detection event.
stOdEvent	Occlusion detection event.

2.4.2.3 MEDIA_BUFFER

【Description】

Data buffer pointer.

【definition】

```
typedef void * MEDIA_BUFFER;
```

[Related data types and interfaces]

[OutCbFunc](#)

2.4.2.4 OutCbFunc

【Description】

Data output callback function pointer.

【definition】

```
typedef void ( * OutCbFunc)(MEDIA_BUFFER mb);
```

[Related data types and interfaces]

[MEDIA_BUFFER](#)

2.4.2.5 MB_IMAGE_INFO_S

【Description】

Image information structure.

【definition】

```
typedef struct rkMB_IMAGE_INFO {
    RK_U32 u32Width;
    RK_U32 u32Height;
    RK_U32 u32VerStride;
    RK_U32 u32HorStride;
    IMAGE_TYPE_E enImgType;
} MB_IMAGE_INFO_S;
```

【member】

Member name	description
u32Width	width.
u32Height	height.
u32VerStride	Imaginary wide.
u32HorStride	False height.
enImgType	Image format type.

[Related data types and interfaces]

[IMAGE_TYPE_E](#)

2.5 Error code

System control error codes such as [Table 2-2](#) shows:

Table 2-2 System control API error codes

error code	Macro definition	description
1	RK_ERR_SYS_NULL_PTR	Null pointer error
2	RK_ERR_SYS_NOTREADY	System control properties are not configured
3	RK_ERR_SYS_NOT_PERM	Operation not allowed
4	RK_ERR_SYS_NOMEM	Failed to allocate memory, such as insufficient system memory
5	RK_ERR_SYS_ILLEGAL_PARAM	Invalid parameter setting
6	RK_ERR_SYS_BUSY	System is busy
7	RK_ERR_SYS_NOT_SUPPORT	Unsupported features

3. Video input

3.1 Overview

The function realized by the video input (VI) module: ISPP driver realizes the standard V4L2 device, through the encapsulation of the V4L2 API, it can be collected ISPP multi-channel video data. VI stores the received data into the designated memory area to realize the collection of video data.

3.2 Functional description

3.2.1 VI node name

The creation of VI needs to specify the name of the video node, such as "/dev/video0". It is special on the RV1126/RV1109 platform, corresponding to the node name As follows.

Table 3-1 ISPP node name (RV1126/RV1109 chip)

ISPP node name	Video node path	Maximum width	Supported output format
rkispp_m_bypass	/dev/video13	Does not support setting resolution, no Support zoom	NV12/NV16/YUYV/ FBC0 / FBC2
rkispp_scale0	/dev/video14	3264, up to 8 times zoom	NV12/NV16/YUYV
rkispp_scale1	/dev/video15	1280, maximum support 8 times zoom	NV12/NV16/YUYV
rkispp_scale2	/dev/video16	1280, maximum support 8 times zoom	NV12/NV16/YUYV

3.2.2 VI working mode

VI has two working modes, as shown in the following table

Pattern name	Macro definition name	Function Description
Normal mode	VI_WORK_MODE_NORMAL	Compared to "brightness mode", In this mode, the camera data is normally read and sent to the subsequent stage.
Brightness mode	VI_WORK_MODE_LUMA_ONLY	In brightness mode, the VI is only used for brightness statistics. At this time, the VI module cannot pass the callback function or RK_MPI_SYS_GetMediaBuffer gets data.

3.3 API reference

3.3.1 RK_MPI_VI_EnableChn

【description】

Enable the VI channel.

【grammar】

RK_S32 RK_MPI_VI_EnableChn([VI_PIPE](#) ViPipe, [VI_CHN](#) ViChn);

【parameter】

parameter name	description	Input / output
ViPipe	VI pipe number.	enter
ViChn	VI channel number. Value range: [0, VI_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_VI_DisableChn](#)

3.3.2 RK_MPI_VI_DisableChn

【description】

Close the VI channel.

【grammar】

RK_S32 RK_MPI_VI_DisableChn([VI_PIPE](#) ViPipe, [VI_CHN](#) ViChn);

【parameter】

parameter name	description	Input / output
ViPipe	VI pipe number.	enter
ViChn	VI channel number. Value range: [0, VI_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_VI_EnableChn](#)

3.3.3 RK_MPI_VI_SetChnAttr

【description】

Set VI channel properties.

【grammar】

RK_MPI_VI_SetChnAttr([VI_PIPE](#) ViPipe, [VI_CHN](#) ViChn, const [VI_CHN_ATTR_S](#) *pstChnAttr);

【parameter】

parameter name	description	Input / output
ViPipe	VI pipe number.	enter
ViChn	VI channel number. Value range: [0, VI_MAX_CHN_NUM).	enter
pstChnAttr	VI channel attribute structure pointer.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

3.3.4 RK_MPI_VI_GetChnRegionLuma

【description】

Obtain the area brightness information.

【grammar】

RK_S32 RK_MPI_VI_GetChnRegionLuma([VI_PIPE](#) ViPipe,[VI_CHN](#) ViChn, const [VIDEO_REGION_INFO_S](#) *pstRegionInfo, RK_U64 *pu64LumaData, RK_S32 s32MilliSec);

【parameter】

parameter name	description	lose In / lose Out
ViPipe	VI pipe number.	lose Enter
ViChn	VI channel number. Value range: [0, VI_MAX_CHN_NUM).	lose Enter
pstRegionInfo	Regional information. Where pstRegionInfo->pstRegion is the regional attribute of the statistical regions which starts immediately. Starting position, width, height; pstRegionInfo->u32RegionNum is the number of statistical regions.	lose Enter
pu64LumaData	The memory pointer for receiving area brightness and statistics information. The memory size should be greater than or equal to sizeof(RK_U64)*pstRegionInfo->u32RegionNum.	lose Out
s32MilliSec	Timeout parameter s32MilliSec: -1 means blocking mode; 0 means non-blocking mode; more than 0 means non-blocking mode, the unit of timeout time is milliseconds (ms).	lose Enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

This interface does not support FBC0/FBC2 compression format.

【Example】

no.

【related topic】

no.

3.3.5 RK_MPI_VI_StartStream

【description】

Start the video stream.

【grammar】

RK_S32 RK_MPI_VI_StartStream([VI_PIPE](#) ViPipe, [VI_CHN](#) ViChn);

【parameter】

parameter name	description	Input / output
ViPipe	VI pipe number.	enter
ViChn	VI channel number. Value range: [0, VI_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

3.4 data types

The data types related to video input are defined as follows:

[VI_MAX_DEV_NUM](#): Define the maximum number of VI devices.

[VI_MAX_CHN_NUM](#): Define the total number of VI physical channels and extended channels.

[VI_PIPE](#): VI pipe number.

[VI_CHN](#): VI channel number.

[VI_CHN_ATTR_S](#): VI channel attribute structure pointer.

[VIDEO_REGION_INFO_S](#): Define the video area information.

3.4.1 VI_MAX_DEV_NUM

【Description】

Define the maximum number of VI devices.

【definition】

```
RV1109 / RV1126:
#define VI_MAX_DEV_NUM 4
```

3.4.2 VI_MAX_CHN_NUM

【Description】

Define the total number of VI physical channels and expansion channels.

【definition】

```
RV1109 / RV1126:
#define VI_MAX_CHN_NUM VI_MAX_DEV_NUM
```

3.4.3 VI_PIPE

【Description】

VI pipe number.

【definition】

```
typedef RK_S32 VI_PIPE;
```

3.4.4 VI_CHN

【Description】

VI channel number.

【definition】

```
typedef RK_S32 VI_CHN;
```

3.4.5 VI_CHN_ATTR_S

【Description】

VI channel attribute structure pointer.

【definition】

```
typedef char RK_CHAR;

typedef enum rkVI_CHN_WORK_MODE {
    VI_WORK_MODE_NORMAL = 0 ,
    // for vi single caculate luma.
    // In this mode, vi has no output,
    // and data cannot be obtained from vi.
    VI_WORK_MODE_LUMA_ONLY,
} VI_CHN_WORK_MODE;

typedef struct rkVI_CHN_ATTR_S {
    const RK_CHAR * pcVideoNode;
    RK_U32 u32Width;
    RK_U32 u32Height;
    IMAGE_TYPE_E enPixFmt;
    RK_U32 u32BufCnt; // VI capture video buffer cnt.
    VI_CHN_WORK_MODE enWorkMode;
} VI_CHN_ATTR_S;
```

【member】

Member name	description
pcVideoNode	The path of the video node.
u32Width	video width.
u32Height	video height.
enPixFmt	video format.
u32BufCnt	VI capture video buffer count
enWorkMode	VI channel working mode

【Precautions】

VI_WORK_MODE_LUMA_ONLY mode is used for VI brightness statistics. In this mode, the VI has no output and cannot be obtained from the VI data.

[Related data types and interfaces]

[IMAGE_TYPE_E](#)

[RK_MPL_VI_SetChnAttr](#)

3.4.6 VIDEO_REGION_INFO_S

【Description】

Define the video area information.

【definition】

```
typedef struct rkVIDEO_REGION_INFO_S {
    RK_U32 u32RegionNum; /* count of the region */
    RECT_S * pstRegion; /* region attribute */
} VIDEO_REGION_INFO_S;
```

【member】

Member name	description
u32RegionNum	The number of video areas.
pstRegion	Pointer of location information of video area.

[Related data types and interfaces]

[RECT_S](#)
[RK_MPI_VI_GetChnRegionLuma](#)

3.5 Error code

Video input API error codes such as[Table 3-2](#)Shown:

Table 3-2 Video input API error codes

error code	Macro definition	description
10	RK_ERR_VI_INVALID_CHNID	Invalid video input channel number
11	RK_ERR_VI_BUSY	Video input system is busy
12	RK_ERR_VI_EXIST	Video input channel already exists
13	RK_ERR_VI_NOT_CONFIG	Video input is not configured
14	RK_ERR_VI_TIMEOUT	Video input timeout
15	RK_ERR_VI_BUF_EMPTY	Video input buffer is empty
16	RK_ERR_VI_ILLEGAL_PARAM	Video input parameter setting is invalid
17	RK_ERR_VI_NOTREADY	The video input system is not initialized

4. Video encoding

4.1 Overview

VENC module, namely video encoding module. This module supports multi-channel real-time encoding, and each channel is independent of encoding. The encoding protocol and encoding profile can To be different. While supporting video encoding, the Region module is scheduled to superimpose and occlude the encoded image content. stand by H264/H1265/MJPEG/JPEG encoding.

4.2 Function description

4.2.1 Data flow chart

Note: The functions described in the dashed box are optional and will only be triggered when the encoder is configured accordingly.

4.2.2 Rate control

Encoder type	Support code control type
H265	CBR / VBR
H264	CBR / VBR
MJPEG	CBR / VBR

4.2.3 GOP Mode

GOP Mode is used to customize the dependency of the reference frame, and currently supports the following modes. Note: It can be customized according to requirements.

name	Macro definition	description
Normal mode	VENC_GOPMODE_NORMALP	The most common scenario, one I frame every GopSize
Smart P frame mode	VENC_GOPMODE_SMARTP	A virtual I frame every GopSize, every BgInterval An I frame
Multi-layer time domain reference module formula	VENC_GOPMODE_TSVC	The coding dependency is divided into multiple layers, which can be RK_MPI_MB_GetTsvcLevel to obtain layer information, So as to customize the code stream. For example, only playing the 0th layer code stream can realize quick preview.

4.2.4 Region of Interest (ROI)

By configuring the region of interest of the encoder, the QP can be customized for the specified region. For example, a camera facing the corridor, what the user is really interested in It is in the middle of the corridor. ROI can be configured to make the coding quality in the center of the corridor higher, the image is clearer, and the border of the corridor (wall, ceiling, etc.) The image quality of the non-interest area will be low. In this way, the user's area of interest is highlighted while keeping the bit rate basically unchanged.

The system provides 8 regions of interest, the priority increases from REGION_ID_0~REGION_ID_7. In areas where multiple ROIs overlap, its QP The strategy will be configured according to the high priority area.

REGION_ID_0
REGION_ID_1
REGION_ID_2
REGION_ID_3
REGION_ID_4
REGION_ID_5
REGION_ID_6
REGION_ID_7

4.2.5 rotation (Rotation)

The encoder supports 4 types of rotation, 0°, 90°, 180°, 270°. Encoder rotation currently does not support FBC format, FBC format The rotation needs to be achieved by the rotation of ISPP.

4.3 API reference

4.3.1 RK_MPI_VENC_CreateChn

【description】

Create an encoding channel.

【grammar】

RK_MPI_VENC_CreateChn([VENC_CHN](#) VeChn,[VENC_CHN_ATTR_S](#) *stVencChnAttr);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter
stVencChnAttr	Encoding channel attribute pointer.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.3.2 RK_MPI_VENC_DestroyChn

【description】

Destroy the encoding channel.

【grammar】

RK_S32 RK_MPI_VENC_DestroyChn([VENC_CHN](#) VeChn);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.

Non-zero Failure, see [error code for](#) its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.3.3 RK_MPI_VENC_SetRcParam

【description】

Set the bit rate control parameters.

【grammar】

RK_MPI_VENC_SetRcParam([VENC_CHN](#) VeChn, const [VENC_RC_PARAM_S](#) *pstRcParam);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter
pstRcParam	The advanced parameters of the code rate controller of the encoding channel.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.3.4 RK_MPI_VENC_SetRcMode

【description】

Set the bit rate control mode.

【grammar】

RK_S32 RK_MPI_VENC_SetRcMode([VENC_CHN](#) VeChn,[VENC_RC_MODE_E](#) RcMode);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter
RcMode	Rate control mode.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.3.5 RK_MPI_VENC_SetRcQuality

【description】

Set the encoding quality. Used for H264/H265 encoder.

【grammar】

RK_MPI_VENC_SetRcQuality([VENC_CHN](#) VeChn, [VENC_RC_QUALITY](#) E RcQuality);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter
RcQuality	Encoding quality.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.3.6 RK_MPI_VENC_SetBitrate

【description】

Set the bit rate.

【grammar】

RK_MPI_VENC_SetBitrate([VENC_CHN](#) VeChn, RK_U32 u32BitRate, RK_U32 u32MinBitRate, RK_U32 u32MaxBitRate);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter
u32BitRate	Target bit rate.	enter
u32MinBitRate	The minimum bit rate.	enter
u32MaxBitRate	Maximum bit rate.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.3.7 RK_MPI_VENC_RequestIDR

【description】

Request IDR frame. After calling this interface, the encoder immediately refreshes the IDR frame.

【grammar】

RK_S32 RK_MPI_VENC_RequestIDR([VENC_CHN](#) VeChn, RK_BOOL bInstant);

【parameter】

parameter name	description	Input / output
----------------	-------------	----------------

VeChn	Encoding channel number. Value range: {0, VENC_MAX_CHN_NUM }.	enter
bInstant	Whether to enable IDR frame encoding immediately.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.3.8 RK_MPI_VENC_SetFps

【description】

Set the encoding frame rate.

【grammar】

RK_S32 RK_MPI_VENC_SetFps([VENC_CHN](#) VeChn, RK_U8 u8OutNum, RK_U8 u8OutDen, RK_U8 u8InNum, RK_U8 u8InDen);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: {0, VENC_MAX_CHN_NUM }.	enter
u8OutNum	The denominator of the encoding output frame rate.	enter
u8OutDen	Encode the output frame rate numerator.	enter
u8InNum	Encode the denominator of the input frame rate.	enter
u8InDen	Encode the input frame rate numerator.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

The output frame rate cannot be greater than the input frame rate.

【Example】

no.

【related topic】

no.

4.3.9 RK_MPI_VENC_SetGop

【description】

Set the GOP. Used for H264/H265 encoder.

【grammar】

RK_S32 RK_MPI_VENC_SetGop([VENC_CHN](#) VeChn, RK_U32 u32Gop);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter
u32Gop	GOP.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.3.10 RK_MPI_VENC_SetAvcProfile

【description】

Set the profile. Used for H264 encoder.

【grammar】

RK_MPI_VENC_SetAvcProfile([VENC_CHN](#) VeChn, RK_U32 u32Profile,RK_U32 u32Level);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter
u32Profile	Profile IDC value.	enter
u32Level	Level IDC value.	enter

【return value】

return value	description
0	success.

Non-zero Failure, see [error code for](#) its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

For the time being, only u32Profile is supported as 66, 77, and 100, corresponding to Baseline, Main Profile, and High Profile respectively.

【Example】

no.

【related topic】

no.

4.3.11 RK_MPI_VENC_InsertUserData

【description】

Insert user data, the inserted data will be reflected in the SEI packet of the code stream. Used for H264/H265 encoder.

【grammar】

RK_MPI_VENC_InsertUserData([VENC_CHN](#) VeChn, RK_U8 *pu8Data, RK_U32 u32Len);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: {0, VENC_MAX_CHN_NUM).	enter
pu8Data	User data pointer.	enter
u32Len	User data length.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

For the time being, only u32Profile is supported as 66, 77, and 100, corresponding to Baseline, Main Profile, and High Profile respectively.

【Example】

no.

【related topic】

no.

4.3.12 RK_MPI_VENC_SetRoiAttr

【description】

Set the ROI coding area of interest. Used for H264/H265 encoder.

```
【grammar】
RK_MPI_VENC_SetRoiAttr( VENC\_CHN VeChn,const VENC\_ROI\_ATTR\_S *pstRoiAttr);

【parameter】
```

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: {0, VENC_MAX_CHN_NUM).	enter
pstRoiAttr	ROI area parameters.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.3.13 RK_MPI_VENC_SetGopMode

【description】

Set GopMode. Used for H264/H265 encoder.

```
【grammar】
RK_S32 RK_MPI_VENC_SetGopMode( VENC\_CHN VeChn,VENC\_GOP\_ATTR\_S GopMode);
```

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: {0, VENC_MAX_CHN_NUM).	enter
GopMode	GOP attribute structure.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.3.14 RK_MPI_VENC_RGN_Init

【description】

Initialize OSD.

【grammar】

RK_S32 RK_MPI_VENC_RGN_Init([VENC_CHN](#) VeChn);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

Calling[RK_MPI_VENC_RGN_SetBitMap](#)Or [RK_MPI_VENC_RGN_SetCover](#)Before, you must call this interface first, and And each encoding channel can only be called once.

【Example】

no.

【related topic】

[RK_MPI_VENC_RGN_SetBitMap](#)

[RK_MPI_VENC_RGN_SetCover](#)

4.3.15 RK_MPI_VENC_RGN_SetBitMap

【description】

Set the OSD bitmap.

【grammar】

RK_S32 RK_MPI_VENC_RGN_SetBitMap([VENC_CHN](#) VeChn, const[OSD_REGION_INFO_S](#) *pstRgnInfo, const [BITMAP_S](#) *pstBitmap);

【parameter】

parameter name	description	Input / output
----------------	-------------	----------------

VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter
pstRgnInfo	OSD area information.	enter
pstBitmap	Bitmap information and data.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

Before calling this interface, you must first call[RK_MPI_VENC_RGN_Init](#).

【Example】

no.

【related topic】

[RK_MPI_VENC_RGN_Init](#)

4.3.16 RK_MPI_VENC_RGN_SetCover

【description】

Set privacy mask.

【grammar】

RK_S32 RK_MPI_VENC_RGN_SetCover([VENC_CHN](#) VeChn, const[OSD_REGION_INFO_S](#) *pstRgnInfo, const [COVER_INFO_S](#) *pstCoverInfo);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter
pstRgnInfo	OSD area information.	enter
pstCoverInfo	Privacy obscures information.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

Before calling this interface, you must first call[RK_MPI_VENC_RGN_Init](#).

【Example】

no.

【related topic】

[RK_MPI_VENC_RGN_Init](#)

4.3.17 RK_MPI_VENC_SetJpegParam

【description】

Set JPEG encoding parameters.

【grammar】

RK_S32 RK_MPI_VENC_SetJpegParam([VENC_CHN](#) VeChn, const[VENC_JPEG_PARAM_S](#) *pstJpegParam);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter
pstJpegParam	The advanced parameters of the JPEG protocol encoding channel.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.3.18 RK_MPI_VENC_StartRecvFrame

【description】

Set the number of frames received by the encoder. The encoder created by default will continue to receive VI data through The RK_MPI_VENC_StartRecvFrame interface can set the number of received frames. After reaching the specified number, the encoder will sleep until the next Call this interface one time to change the number of received frames.

【grammar】

RK_S32 RK_MPI_VENC_StartRecvFrame([VENC_CHN](#) VeChn, const [VENC_RECV_PIC_PARAM_S](#) *pstRecvParam);

【parameter】

parameter name	description	Input / output
VeChn	Encoding channel number. Value range: [0, VENC_MAX_CHN_NUM).	enter
pstRecvParam	Receive image parameter structure pointer, used to specify the number of image frames that need to be received.	

【return value】

return value	description
--------------	-------------

0 success.

Non-zero Failure, see [error code for](#) its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

4.4 Data Type

The video coding related data types are defined as follows:

[VENC_MAX_CHN_NUM](#): Define the total number of VENC physical channels and extended channels.

[VENC_CHN](#): VENC channel number.

[VENC_ATTR_JPEG_S](#): Define the attribute structure of the JPEG capture encoder.

[VENC_ATTR_MJPEG_S](#): Define the attribute structure of the MJPEG encoder.

[VENC_ATTR_H264_S](#): Define the attribute structure of the H.264 encoder.

[VENC_ATTR_H265_S](#): Define the attribute structure of the H.265 encoder.

[VENC_ATTR_S](#): Define the encoder attribute structure.

[VENC_MJPEG_CBR_S](#): Define the CBR attribute structure of the MJPEG encoding channel.

[VENC_MJPEG_VBR_S](#): Define the VBR attribute structure of the MJPEG encoding channel.

[VENC_H264_CBR_S](#): Define the CBR attribute structure of the H.264 encoding channel.

[VENC_H264_VBR_S](#): Define the VBR attribute structure of the H.264 encoding channel.

[VENC_H265_CBR_S](#): Define the CBR attribute structure of the H.265 encoding channel.

[VENC_H265_VBR_S](#): Define the VBR attribute structure of the H.265 encoding channel.

[VENC_RC_MODE_E](#): Define the code rate controller mode of the encoding channel.

[VENC_RC_ATTR_S](#): Define the code rate controller attributes of the encoding channel.

[VENC_GOP_MODE_E](#): Define Gop Mode type.

[VENC_GOP_ATTR_S](#): Define the encoder GOP attribute structure.

[VENC_CHN_ATTR_S](#): VENC channel attribute structure.

[VENC_PARAM_MJPEG_S](#): MJPEG channel parameters.

[VENC_PARAM_H264_S](#): H.264 channel parameters.

[VENC_PARAM_H265_S](#): H.265 channel parameters.

[VENC_RC_PARAM_S](#): Advanced parameters of the code rate controller of the encoding channel.

[VENC_RC_QUALITY_E](#): Encoding quality.

[VENC_ROI_ATTR_S](#): ROI attribute structure.

[OSD_REGION_ID_E](#): OSD region ID enumeration type.

- [OSD_REGION_INFO_S](#): OSD area information.
- [OSD_PIXEL_FORMAT_E](#): OSD pixel format type enumeration.
- [BITMAP_S](#): Bitmap information and data.
- [COVER_INFO_S](#): Privacy masking information.
- [VENC_RECV_PIC_PARAM_S](#): Receive image parameter structure pointer, used to specify the number of image frames that need to be received.
- [VENC_JPEG_PARAM_S](#): The advanced parameters of the JPEG protocol encoding channel.

4.4.1 VENC_MAX_CHN_NUM

【Description】
The total number of VENC physical channels and expansion channels.

【definition】

```
RV1109 / RV1126:  
#define VENC_MAX_CHN_NUM 16
```

4.4.2 VENC_CHN

【Description】
VENC channel number.

【definition】

```
typedef RK_S32 VENC_CHN;
```

4.4.3 VENC_ATTR_JPEG_S

【Description】
Define the attribute structure of the JPEG capture encoder.

【definition】

```
typedef struct rkVENC_ATTR_JPEG_S {  
    RK_U32 u32ZoomWidth; // Zoom to specified width  
    RK_U32 u32ZoomHeight; // Zoom to specified height  
    RK_U32 u32ZoomVirWidth;  
    RK_U32 u32ZoomVirHeight;  
} VENC_ATTR_JPEG_S;
```

【member】

Member name	description
u32ZoomWidth	The specified width of the zoom.
u32ZoomHeight	The specified height of the zoom.
u32ZoomVirWidth	The virtual height of the zoom.
u32ZoomVirHeight	The virtual width of the zoom.

4.4.4 VENC_ATTR_MJPEG_S

【Description】

Define the attribute structure of the MJPEG encoder.

【definition】

```
typedef struct rkVENC_ATTR_MJPEG_S {
    RK_U32 u32ZoomWidth; // Zoom to specified width
    RK_U32 u32ZoomHeight; // Zoom to specified height
    RK_U32 u32ZoomVirWidth;
    RK_U32 u32ZoomVirHeight;
} VENC_ATTR_MJPEG_S;
```

【member】

Member name	description
u32ZoomWidth	The specified width of the zoom.
u32ZoomHeight	The specified height of the zoom.
u32ZoomVirWidth	The virtual width of the zoom.
u32ZoomVirHeight	The virtual height of the zoom.

4.4.5 VENC_ATTR_H264_S

【Description】

Define the attribute structure of the H.264 encoder.

【definition】

```
typedef struct rkVENC_ATTR_H264_S {
    RK_U32 u32Level;
    // reserved
} VENC_ATTR_H264_S;
```

【member】

Member name	description
u32Level	Profile IDC value.

4.4.6 VENC_ATTR_H265_S

【Description】

Define the attribute structure of the H.265 encoder.

【definition】

```
typedef struct rkVENC_ATTR_H265_S {
    // reserved
} VENC_ATTR_H265_S;
```

4.4.7 VENC_ATTR_S

【Description】

Define the encoder attribute structure.

【definition】

```
typedef struct rkVENC_ATTR_S {
    CODEC_TYPE_E enType; // RW; the type of encodec
```

```
IMAGE_TYPE_E imageType; // the type of input image
RK_U32 u32VirWidth; // stride width, same to buffer_width, must greater than
                        // width, often set vir_width=(width+15)&(~15)
RK_U32 u32VirHeight; // stride height, same to buffer_height, must greater
                        // than height, often set vir_height=(height+15)&(~15)
RK_U32 u32Profile; // RW;
                        // H.264: 66: baseline; 77:MP; 100:HP;
                        // H.265: default:Main;
                        // Jpege/MJpege: default:Baseline
RK_BOOL bByFrame; // RW; Range:[0,1];
                        // get stream mode is slice mode or frame mode
RK_U32 u32PicWidth; // RW; width of a picture to be encoded, in pixel
RK_U32 u32PicHeight; // RW; height of a picture to be encoded, in pixel
VENC_ROTATION_E enRotation;
union {
    VENC_ATTR_H264_S stAttrH264e; // attributes of H264e
    VENC_ATTR_H265_S stAttrH265e; // attributes of H265e
    VENC_ATTR_MJPEG_S stAttrMjpege; // attributes of Mjpege
    VENC_ATTR_JPEG_S stAttrJpege; // attributes of jpeg
};
} VENC_ATTR_S;
```

【member】

Page 56

Member name	description
enType	Encoding protocol type.
imageType	Enter the image type.
u32VirWidth	stride width (same as buffer_width), must be greater than width, usually set vir_width=(width+15)&(~15).
u32VirHeight	stride height (same as buffer_height), must be greater than height, usually set vir_height=(height+15)& (~15).
u32Profile	Encoding level. H.264: 66: Baseline; 77: Main Profile; 100: High Profile; H.265: default:Main; Jpege/MJpege: default:Baseline
bByFrame	Whether to obtain the code stream according to the frame mode. Value range: [0, 1]. 1: frame mode. 0: slice mode.
u32PicWidth	Encode the image width. In pixels.
u32PicHeight	The height of the encoded image. In pixels.
stAttrH264e/stAttrH265e/stAttrMjpege/stAttrJpege	Encoder properties of a certain protocol.

[Related data types and interfaces]

[VENC_ATTR_JPEG_S](#)

[VENC_ATTR_MJPEG_S](#)

[VENC_ATTR_H264_S](#)

[VENC_ATTR_H265_S](#)

[VENC_CHN_ATTR_S](#)

4.4.8 VENC_MJPEG_CBR_S

【Description】

Define the CBR attribute structure of the MJPEG encoding channel.

【definition】

```
typedef struct hiVENC_MJPEG_CBR_S {
    RK_U32 u32SrcFrameRateNum;
    RK_U32 u32SrcFrameRateDen;
    RK_FR32 fr32DstFrameRateNum;
    RK_FR32 fr32DstFrameRateDen;
    RK_U32 u32BitRate; // RW; Range:[2000, 98000000]; average bitrate
} VENC_MJPEG_CBR_S;
```

【member】

Member name	description
u32SrcFrameRateNum	The frame rate numerator of the data source.
u32SrcFrameRateDen	The denominator of the frame rate of the data source.
fr32DstFrameRateNum	Target frame rate numerator.
fr32DstFrameRateDen	The denominator of the target frame rate.
u32BitRate	Average bit rate, value range: [2000, 98000000].

4.4.9 VENC_MJPEG_VBR_S

【Description】

Define the VBR attribute structure of the MJPEG encoding channel.

【definition】

```
typedef struct hiVENC_MJPEG_VBR_S {
    RK_U32 u32SrcFrameRateNum;
    RK_U32 u32SrcFrameRateDen;
    RK_FR32 fr32DstFrameRateNum;
    RK_FR32 fr32DstFrameRateDen;
    RK_U32 u32BitRate; // RW; Range:[2000, 98000000]; average bitrate
} VENC_MJPEG_VBR_S;
```

【member】

Member name	description
u32SrcFrameRateNum	The frame rate numerator of the data source.
u32SrcFrameRateDen	The denominator of the frame rate of the data source.
fr32DstFrameRateNum	Target frame rate numerator.
fr32DstFrameRateDen	The denominator of the target frame rate.
u32BitRate	Average bit rate, value range: [2000, 98000000].

4.4.10 VENC_H264_CBR_S

【Description】

Define the CBR attribute structure of the H.264 encoding channel.

【definition】

```
typedef struct rkVENC_H264_CBR_S {  
    RK_U32 u32Gop; // RW; Range:[1, 65536]; the interval of I Frame.  
    RK_U32 u32SrcFrameRateNum;  
    RK_U32 u32SrcFrameRateDen;  
    RK_FR32 fr32DstFrameRateNum;  
    RK_FR32 fr32DstFrameRateDen;  
    RK_U32 u32BitRate; // RW; Range:[2, 614400]; average bitrate  
} VENC_H264_CBR_S;
```

【member】

Member name	description
u32Gop	I frame interval, value range: [1, 65536].
u32SrcFrameRateNum	The frame rate numerator of the data source.
u32SrcFrameRateDen	The denominator of the frame rate of the data source.
fr32DstFrameRateNum	Target frame rate numerator.
fr32DstFrameRateDen	The denominator of the target frame rate.
u32BitRate	Average bit rate, value range: [2, 614400].

4.4.11 VENC_H264_VBR_S

【Description】

Define the VBR attribute structure of the H.264 encoding channel.

【definition】

```
typedef struct rkVENC_H264_VBR_S {  
    RK_U32 u32Gop; // RW; Range:[1, 65536]; the interval of ISLICE.  
    RK_U32 u32SrcFrameRateNum;  
    RK_U32 u32SrcFrameRateDen;  
    RK_FR32 fr32DstFrameRateNum;  
    RK_FR32 fr32DstFrameRateDen;  
    RK_U32 u32MaxBitRate; // RW; Range:[2, 614400];the max bitrate  
} VENC_H264_VBR_S;
```

【member】

Member name	description
u32Gop	ISLICE interval, value range: [1, 65536].
u32SrcFrameRateNum	The frame rate numerator of the data source.
u32SrcFrameRateDen	The denominator of the frame rate of the data source.
fr32DstFrameRateNum	Target frame rate numerator.
fr32DstFrameRateDen	The denominator of the target frame rate.
u32BitRate	Average bit rate, value range: [2, 614400].

4.4.12 VENC_H265_CBR_S

【Description】

Define the CBR attribute structure of the H.265 encoding channel.

【definition】

```
typedef struct rkVENC_H264_CBR_S VENC_H265_CBR_S ;
```

[Related data types and interfaces]

[VENC_H264_CBR_S](#)

4.4.13 VENC_H265_VBR_S

【Description】

Define the VBR attribute structure of the H.265 encoding channel.

【definition】

```
typedef struct rkVENC_H264_VBR_S VENC_H265_VBR_S ;
```

[Related data types and interfaces]

[VENC_H264_VBR_S](#)

4.4.14 VENC_RC_MODE_E

【Description】

Define the code rate controller mode of the encoding channel.

【definition】

```
typedef enum rkVENC_RC_MODE_E {
    // H264
    VENC_RC_MODE_H264CBR = 1 ,
    VENC_RC_MODE_H264VBR,
    // MJPEG
    VENC_RC_MODE_MJPEGCBR,
    VENC_RC_MODE_MJPEGVBR,
    // H265
    VENC_RC_MODE_H265CBR,
    VENC_RC_MODE_H265VBR,
    VENC_RC_MODE_BUTT,
} VENC_RC_MODE_E;
```

4.4.15 VENC_RC_ATTR_S

【Description】

Define the code rate controller properties of the encoding channel.

【definition】

```
typedef struct rkVENC_RC_ATTR_S {
    /* RW; the type of rc*/
    VENC_RC_MODE_E enRcMode;
    union {
        VENC_H264_CBR_S stH264Cbr;
        VENC_H264_VBR_S stH264Vbr;
```



```
VENC_MJPEG_CBR_S stMjpegCbr;
VENC_MJPEG_VBR_S stMjpegVbr;

VENC_H265_CBR_S stH265Cbr;
VENC_H265_VBR_S stH265Vbr;
};
} VENC_RC_ATTR_S;
```

【member】

Member name	description
enRcMode	Encoding protocol type.
stH264Cbr	H.264 protocol encoding channel Cbr mode attribute.
stH264Vbr	H.264 protocol encoding channel Vbr mode attribute.
stMjpegCbr	Cbr mode attribute of MJPEG protocol encoding channel.
stMjpegVbr	MJPEG protocol encoding channel Vbr mode attribute.
stH265Cbr	H.265 protocol encoding channel Cbr mode attribute.
stH265Vbr	H.265 protocol encoding channel Vbr mode attribute.

[Related data types and interfaces]

- [VENC_MJPEG_CBR_S](#)
- [VENC_MJPEG_VBR_S](#)
- [VENC_H264_CBR_S](#)
- [VENC_H264_VBR_S](#)
- [VENC_H265_CBR_S](#)
- [VENC_H265_VBR_S](#)
- [VENC_RC_MODE_E](#)

4.4.16 VENC_GOP_MODE_E

【Description】

Define the Gop Mode type.

【definition】

```
typedef enum rkVENC_GOP_MODE_E {
    VENC_GOPMODE_NORMALP = 0 ,
    VENC_GOPMODE_TSVC,
    VENC_GOPMODE_SMARTP,
    VENC_GOPMODE_BUTT,
} VENC_GOP_MODE_E;
```

【note】

For specific mode description, please refer to [GOP Mode](#) .

4.4.17 VENC_GOP_ATTR_S

【Description】

Define the encoder GOP attribute structure.

【definition】

```
typedef struct rkVENC_GOP_ATTR_S {
    VENC_GOP_MODE_E enGopMode;
```

```
RK_U32 u32GopSize;
RK_S32 s32IPQpDelta;
RK_U32 u32BgInterval;
RK_S32 s32ViQpDelta;
} VENC_GOP_ATTR_S;
```

【member】

Member name	description
enGopMode	Encoding GOP type.
u32GopSize	Encode GOP size.
s32IPQpDelta	The QP difference between the I frame and the P frame.
u32BgInterval	Long-term reference frame interval.
s32ViQpDelta	The QP difference between the virtual I frame and the normal P frame.

[Related data types and interfaces]

[VENC_GOP_MODE_E](#)

4.4.18 VENC_CHN_ATTR_S

【Description】

VENC channel attribute structure.

【definition】

```
typedef struct rkVENC_CHN_ATTR_S {
    VENC_ATTR_S stVencAttr; // the attribute of video encoder
    VENC_RC_ATTR_S stRcAttr; // the attribute of rate ctrl
    VENC_GOP_ATTR_S stGopAttr; // the attribute of gop
} VENC_CHN_ATTR_S;
```

【member】

Member name	description
stVencAttr	Encoder attributes.
stRcAttr	Bit rate controller properties.
stGopAttr	GOP attributes.

[Related data types and interfaces]

[VENC_ATTR_S](#)

[VENC_RC_ATTR_S](#)

[VENC_GOP_ATTR_S](#)

4.4.19 VENC_PARAM_MJPEG_S

【Description】

MJPEG channel parameters.

【definition】

```
typedef struct rkVENC_PARAM_MJPEG_S {  
    // reserved  
} VENC_PARAM_MJPEG_S;
```

4.4.20 VENC_PARAM_H264_S

【Description】

H.264 channel parameters.

【definition】

```
typedef struct rkVENC_PARAM_H264_S {  
    RK_U32 u32StepQp;  
    RK_U32 u32MaxQp; // RW; Range:[8, 51];the max QP value  
    RK_U32 u32MinQp; // RW; Range:[0, 48]; the min QP value,can not be larger than  
        // u32MaxQp  
    RK_U32 u32MaxIQp; // RW; max qp for i frame  
    RK_U32 u32MinIQp; // RW; min qp for i frame,can not be larger  
        // than u32MaxIQp  
    // RK_S32 s32MaxReEncodeTimes; // RW; Range:[0, 3]; Range:max number  
    // of re-encode times.*/  
} VENC_PARAM_H264_S;
```

【member】

Member name	description
u32StepQp	The step value of QP.
u32MaxQp	QP maximum value, value range [8, 51].
u32MinQp	The minimum value of QP, the value range is [0, 48], and cannot be greater than u32MaxQp.
u32MaxIQp	The maximum QP of the I frame.
u32MinIQp	The QP minimum value of the I frame.

4.4.21 VENC_PARAM_H265_S

【Description】

H.265 channel parameters.

【definition】

```
typedef struct rkVENC_PARAM_H265_S {  
    RK_U32 u32StepQp;  
    RK_U32 u32MaxQp; // RW; Range:[8, 51];the max QP value  
    RK_U32 u32MinQp; // RW; Range:[0, 48];the min QP value ,can not be larger than  
        // u32MaxQp  
    RK_U32 u32MaxIQp; // RW; max qp for i frame  
    RK_U32 u32MinIQp; // RW; min qp for i frame,can not be larger than u32MaxIQp  
  
    // RK_S32 s32MaxReEncodeTimes; // RW; Range:[0, 3]; Range:max number  
    // of re-encode times.*/  
    // RK_U32 u32DeltIpQp;
```

```
} VENC_PARAM_H265_S;
```

【member】

Member name	description
u32StepQp	The step value of QP.
u32MaxQp	QP maximum value, value range [8, 51].
u32MinQp	The minimum value of QP, the value range is [0, 48], and cannot be greater than u32MaxQp.
u32MaxIQp	The maximum QP of the I frame.
u32MinIQp	The QP minimum value of the I frame.

4.4.22 VENC_RC_PARAM_S

【Description】

The advanced parameters of the code rate controller of the encoding channel.

【definition】

```
typedef struct rkVENC_RC_PARAM_S {  
    RK_U32 s32FirstFrameStartQp; // RW; Start QP value of the first frame  
    union {  
        VENC_PARAM_H264_S stParamH264;  
        VENC_PARAM_H265_S stParamH265;  
        VENC_PARAM_MJPEG_S stParamMjpeg;  
    };  
} VENC_RC_PARAM_S;
```

【member】

Member name	description
s32FirstFrameStartQp	The QP value of the first frame.
stParamH264	H.264 channel parameters.
stParamH265	H.265 channel parameters.
stParamMjpeg	MJPEG channel parameters.

[Related data types and interfaces]

[VENC_PARAM_H264_S](#)

[VENC_PARAM_H265_S](#)

[VENC_PARAM_MJPEG_S](#)

4.4.23 VENC_RC_QUALITY_E

【Description】

Enumerated type of encoding quality.

【definition】

```
typedef enum rkVENC_RC_QUALITY_E {  
    VENC_RC_QUALITY_HIGHEST,  
    VENC_RC_QUALITY_HIGHER,  
    VENC_RC_QUALITY_HIGH,  
    VENC_RC_QUALITY_MEDIUM,  
    VENC_RC_QUALITY_LOW,  
    VENC_RC_QUALITY_LOWER,  
    VENC_RC_QUALITY_LOWEST,  
    VENC_RC_QUALITY_BUTT,  
} VENC_RC_QUALITY_E;
```

4.4.24 VENC_ROI_ATTR_S

【Description】

ROI area parameters.

【definition】

```
typedef struct hiVENC_ROI_ATTR_S {
    RK_U32 u32Index; // RW; Range:[0, 7]; Index of an ROI. The system supports
                    // indexes ranging from 0 to 7
    RK_BOOL bEnable; // RW; Range:[0, 1]; Whether to enable this ROI
    RK_BOOL bAbsQp; // RW; Range:[0, 1]; QP mode of an ROI.HI_FALSE: relative
                    // QP.HI_TURE: absolute QP.
    RK_S32 s32Qp; // RW; Range:[-51, 51]; QP value,only relative mode can QP value
                    // less than 0.
    RK_BOOL bIntra; // flag of forced intra macroblock
    RECT_S stRect; // RW; Region of an ROI
} VENC_ROI_ATTR_S;
```

【member】

Member name	description
u32Index	ROI index value, the value range is [0, 7].
bEnable	Whether to enable ROI.
bAbsQp	QP mode of ROI, value range: [0, 1]. 1: absolute QP. 0: relative QP.
s32Qp	QP value, value range: [-51, 51]. Only the relative mode can make the QP value less than 0.
bIntra	Mandatory marking of intra-frame macroblocks.
stRect	ROI area.

[Related data types and interfaces]

[RECT_S](#)

4.4.25 OSD_REGION_ID_E

【Description】

OSD area ID enumeration type.

【definition】

```
typedef enum rkOSD_REGION_ID_E {
    REGION_ID_0 = 0 ,
    REGION_ID_1,
    REGION_ID_2,
    REGION_ID_3,
    REGION_ID_4,
    REGION_ID_5,
    REGION_ID_6,
    REGION_ID_7
} OSD_REGION_ID_E;
```

4.4.26 OSD_REGION_INFO_S

【Description】

OSD area information.

【definition】

```
typedef struct rkOSD_REGION_INFO_S {
    OSD_REGION_ID_E enRegionId;
    RK_U32 u32PosX;
    RK_U32 u32PosY;
    RK_U32 u32Width;
    RK_U32 u32Height;
    RK_U8 u8Inverse;
    RK_U8 u8Enable;
} OSD_REGION_INFO_S;
```

【member】

Member name	description
enRegionId	OSD area index value, the value range is [0, 7].
u32PosX	The X-axis coordinate of the OSD area.
u32PosY	The Y-axis coordinate of the OSD area.
u32Width	OSD area width.
u32Height	The height of the OSD area.
u8Inverse	Whether the OSD area is inverted.
u8Enable	Whether the OSD area is enabled.

[Related data types and interfaces]

[OSD_REGION_ID_E](#)

4.4.27 OSD_PIXEL_FORMAT_E

【Description】

OSD pixel format type enumeration.

【definition】

```
typedef enum rkOSD_PIXEL_FORMAT_E {
    PIXEL_FORMAT_ARGB_1555 = 0,
    PIXEL_FORMAT_ARGB_8888,
} OSD_PIXEL_FORMAT_E;
```

4.4.28 BITMAP_S

【Description】

Bitmap information and data.

【definition】

```
typedef struct rkBITMAP_S {
    OSD_PIXEL_FORMAT_E enPixelFormat; /* Bitmap's pixel format */
    RK_U32 u32Width;                  /* Bitmap's width */
    RK_U32 u32Height;                 /* Bitmap's height */
    RK_VOID * pData;                  /* Address of Bitmap's data */
} BITMAP_S;
```

【member】

Member name	description
enPixelFormat	Bitmap pixel format.
u32Width	The width of the bitmap.
u32Height	The height of the bitmap.
pData	The address of the bitmap data.

[Related data types and interfaces]

[OSD_PIXEL_FORMAT_E](#)

4.4.29 COVER_INFO_S

【Description】

Privacy obscures information.

【definition】

```
typedef struct rkCOVER_INFO_S {
    OSD_PIXEL_FORMAT_E enPixelFormat; /* Bitmap's pixel format */
    RK_U32 u32Color;                  /* Covered area color */
} COVER_INFO_S;
```

【member】

Member name	description
enPixelFormat	Bitmap pixel format.
u32Color	The color of the occluded area.

[Related data types and interfaces]

[OSD_PIXEL_FORMAT_E](#)

4.4.30 VENC_RECV_PIC_PARAM_S

【Description】

Receive image parameter structure pointer, used to specify the number of image frames that need to be received.

【definition】

```
typedef struct rkVENC_RECV_PIC_PARAM_S {
    RK_S32 s32RecvPicNum;
} VENC_RECV_PIC_PARAM_S;
```

【member】

Member name	description
-------------	-------------

s32RecvPicNum

The number of image frames that need to be received.

4.4.31 VENC_JPEG_PARAM_S

【Description】

The advanced parameters of the JPEG protocol encoding channel.

【definition】

```
typedef struct rkVENC_JPEG_PARAM_S {
    RK_U32 u32Qfactor; // 1-10
    RK_U8 u8YQt[ 64 ];
    RK_U8 u8CbQt[ 64 ];
    RK_U8 u8CrQt[ 64 ];
    RK_U32 u32MCUPerECS;
} VENC_JPEG_PARAM_S;
```

【member】

Member name	description
u32Qfactor	For specific meaning, please refer to the RFC2435 protocol, value range: [1, 10].
u8YQt	Y quantization table.
u8CbQt	Cb quantization table.
u8CrQt	Cr quantization table.
u32MCUPerECS	How many MCUs are included in each ECS.

4.5 Error code

Video encoding API error codes such as[Table 4-1](#)Shown:

Table 4-1 Video encoding API error codes

Wrong generation code	Macro definition	description
20	RK_ERR_VENC_INVALID_CHNID	Channel ID is out of legal range
twenty one	RK_ERR_VENC_ILLEGAL_PARAM	Parameter out of legal range
twenty two	RK_ERR_VENC_EXIST	Try to apply for or create an existing device, channel or Resource
twenty three	RK_ERR_VENC_UNEXIST	Attempts to use or destroy non-existent equipment, channels, or Resources
twenty four	RK_ERR_VENC_NULL_PTR	Null pointer in function parameter
25	RK_ERR_VENC_NOT_CONFIG	Not configured before use
26	RK_ERR_VENC_NOT_SUPPORT	Unsupported parameter or function
27	RK_ERR_VENC_NOT_PERM	The operation is not allowed, such as trying to modify static configuration parameters
28	RK_ERR_VENC_NOBUF	Failed to allocate cache, such as the requested data buffer is too large
29	RK_ERR_VENC_BUF_EMPTY	No data in the buffer
30	RK_ERR_VENC_BUF_FULL	Data in the buffer is full

31	RK_ERR_VENC_NOTREADY	The system is not initialized or the corresponding module is not loaded
32	RK_ERR_VENC_BUSY	VENC system is busy

5. Motion detection

5.1 Overview

The motion detection (MD) module realizes motion area detection, and supports a maximum of 4096 areas.

5.2 Function description

The MD algorithm is implemented by software, and the input resolution should not be too large. The typical resolution is 640x480. The larger the resolution, the higher the CPU load.

5.3 API reference

5.3.1 RK_MPI_ALGO_MD_CreateChn

【description】

Create MD channel.

【grammar】

RK_MPI_ALGO_MD_CreateChn([ALGO_MD_CHN](#) MdChn, const [ALGO_MD_ATTR_S](#) *pstChnAttr);

【parameter】

parameter name	description	Input / output
MdChn	Motion detection channel number. Value range: [0, ALGO_MD_MAX_CHN_NUM).	enter
pstChnAttr	Motion detection channel properties.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_ALGO_MD_DestroyChn](#)

5.3.2 RK_MPI_ALGO_MD_DestroyChn

【description】

Destroy the MD channel.

【grammar】

RK_S32 RK_MPI_ALGO_MD_DestroyChn([ALGO_MD_CHN](#) MdChn);

【parameter】

parameter name	description	Input / output
MdChn	Motion detection channel number. Value range: [0, ALGO_MD_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_ALGO_MD_CreateChn](#)

5.4 Data Type

The data types related to motion detection are defined as follows:

[ALGO_MD_MAX_CHN_NUM](#): Define the maximum number of motion detection channels.

[ALGO_MD_ROI_RET_MAX](#): Define the maximum number of ROI areas for each channel of motion detection.

[ALGO_MD_CHN](#): Motion detection channel number.

[ALGO_MD_ATTR_S](#): Define the attribute structure of motion detection channel.

5.4.1 ALGO_MD_MAX_CHN_NUM

【Description】

Define the maximum number of motion detection channels.

【definition】

RV1109 / RV1126:
#define ALGO_MD_MAX_CHN_NUM VI_MAX_CHN_NUM

[Related data types and interfaces]

[VI_MAX_CHN_NUM](#)

5.4.2 ALGO_MD_ROI_RET_MAX

【Description】
Define the maximum number of ROI areas for each channel of motion detection.

【definition】

```
RV1109 / RV1126:  
#define ALGO_MD_ROI_RET_MAX 4096
```

5.4.3 ALGO_MD_CHN

【Description】
Motion detection channel number.

【definition】

```
typedef RK_S32 ALGO_MD_CHN;
```

5.4.4 ALGO_MD_ATTR_S

【Description】
Define the attribute structure of the motion detection channel.

【definition】

```
typedef struct rkALGO_MD_ATTR_S {  
    IMAGE_TYPE_E imageType; // the type of input image  
    RK_U32 u32Width;  
    RK_U32 u32Height;  
    RK_U16 u16RoiCnt; // RW; Range:[0, ALGO_MD_ROI_RET_MAX].  
    RECT_S stRoiRects[ALGO_MD_ROI_RET_MAX];  
    RK_U16 u16Sensitivity; // value 0(sys default) or [1-100].  
} ALGO_MD_ATTR_S;
```

【member】

Member name	description
imageType	Enter the image type.
u32Width	The width of the motion detection area.
u32Height	The height of the motion detection area.
u16RoiCnt	Number of ROI regions, value range: [0, ALGO_MD_ROI_RET_MAX].
stRoiRects	The structure array of the ROI area attribute.
u16Sensitivity	Motion detection sensitivity, value range: [1, 100].

[Related data types and interfaces]

[RECT_S](#)

[IMAGE_TYPE_E](#)

[ALGO_MD_ROI_RET_MAX](#)

5.5 Error code

Video encoding API error codes such as [Table 5-1](#) Shown:

Table 5-1 Video encoding API error codes

Wrong generation code	Macro definition	description
70	RK_ERR_ALGO_MD_INVALID_CHNID	Channel ID is out of legal range
71	RK_ERR_ALGO_MD_BUSY	Motion detection system is busy
72	RK_ERR_ALGO_MD_EXIST	Try to apply for or create an existing device, communication Tao or resource
73	RK_ERR_ALGO_MD_NOT_CONFIG	Not configured before use
74	RK_ERR_ALGO_MD_ILLEGAL_PARAM	Parameter out of legal range

6. occlusion detection

6.1 Overview

The Occlusion Detection module implements occlusion alarms and supports a maximum of 10 areas.

6.2 Function description

The OD algorithm is implemented by software, and the input resolution should not be too large. The typical resolution is 640x480. The larger the resolution, the higher the CPU load.

6.3 API reference

6.3.1 RK_MPI_ALGO_OD_CreateChn

【description】

Create an OD channel.

【grammar】

RK_S32 RK_MPI_ALGO_OD_CreateChn([ALGO_OD_CHN](#) OdChn, const [ALGO_OD_ATTR_S](#) *pstChnAttr);

【parameter】

parameter name	description	Input / output
OdChn	Block detection channel number. Value range: [0, ALGO_OD_MAX_CHN_NUM).	enter
pstChnAttr	The properties of the occlusion detection channel.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_ALGO_OD_DestroyChn](#)

6.3.2 RK_MPI_ALGO_OD_DestroyChn

【description】

Destroy the OD channel.

【grammar】

RK_S32 RK_MPI_ALGO_OD_DestroyChn([ALGO_OD_CHN](#) OdChn);

【parameter】

parameter name	description	Input / output
OdChn	Block detection channel number. Value range: [0, ALGO_OD_MAX_CHN_NUM).	enter

【return value】

Page 76

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_ALGO_OD_CreateChn](#)

6.4 Data Type

The data types related to occlusion detection are defined as follows:

[ALGO_OD_MAX_CHN_NUM](#): Define the maximum number of occlusion detection channels.

[ALGO_OD_ROI_RET_MAX](#): Define the maximum number of ROI regions for each channel of occlusion detection.

[ALGO_OD_CHN](#): Block detection channel number.

[ALGO_OD_ATTR_S](#): Define the attribute structure of the occlusion detection channel.

6.4.1 ALGO_OD_MAX_CHN_NUM

【Description】

Define the maximum number of occlusion detection channels.

【definition】

```
RV1109 / RV1126:
#define ALGO_OD_MAX_CHN_NUM VI_MAX_CHN_NUM
```

[Related data types and interfaces]

[VI_MAX_CHN_NUM](#)

6.4.2 ALGO_OD_ROI_RET_MAX

【Description】

Define the maximum number of ROI regions for each channel of occlusion detection.

【definition】

```
RV1109 / RV1126:
#define ALGO_OD_ROI_RET_MAX 10
```

6.4.3 ALGO_OD_CHN

【Description】

Block detection channel number.

【definition】

```
typedef RK_S32 ALGO_OD_CHN;
```

6.4.4 ALGO_OD_ATTR_S

【Description】

Define the occlusion detection channel attribute structure.

【definition】

```
typedef struct rkALGO_OD_ATTR_S {
    IMAGE_TYPE_E enImageType; // the type of input image
    RK_U32 u32Width;
    RK_U32 u32Height;
    RK_U16 u16RoiCnt; // RW; Range:[0, ALGO_OD_ROI_RET_MAX].
    RECT_S stRoiRects[ALGO_OD_ROI_RET_MAX];
    RK_U16 u16Sensitivity; // value 0(sys default) or [1-100].
} ALGO_OD_ATTR_S;
```

【member】

Member name	description
enImageType	Enter the image type.
u32Width	Block the width of the detection area.
u32Height	Block the height of the detection area.
u16RoiCnt	Number of ROI regions, value range: [0, ALGO_OD_ROI_RET_MAX].

stRoiRects	The structure array of the ROI area attribute.
u16Sensitivity	The sensitivity of occlusion detection, the value range: [1, 100].

[Related data types and interfaces]

- [RECT_S](#)
- [IMAGE_TYPE_E](#)
- [ALGO_OD_ROI_RET_MAX](#)

6.5 Error code

Video encoding API error codes such as [Table 6-1](#) Shown:

Table 6-1 Video encoding API error codes

Wrong generation code	Macro definition	description
80	RK_ERR_ALGO_OD_INVALID_CHNID	Channel ID is out of legal range
81	RK_ERR_ALGO_OD_BUSY	Motion detection system is busy
82	RK_ERR_ALGO_OD_EXIST	Try to apply for or create an existing device, communication Tao or resource
83	RK_ERR_ALGO_OD_NOT_CONFIG	Not configured before use
84	RK_ERR_ALGO_OD_ILLEGAL_PARAM	Parameter out of legal range

7. Audio

7.1 Overview

The AUDIO module includes four sub-modules: audio input, audio output, audio encoding, and audio decoding.

The audio input and output module realizes audio input and output functions by encapsulating the Linux ALSA audio interface.

The audio encoding and decoding module is realized by encapsulating the ffmpeg audio encoder. Support G711A/G711U/G726 /MP2.

7.2 Function description

7.2.1 Audio input and output

Audio input AI output AO, used for docking with Audio Codec to complete sound recording and playback. RKMedia AI/AO depends on Linux ALSA devices, different sound cards, as long as they support the ALSA driver, can use the AI/AO interface. Audio algorithms are integrated in AI, which can be passed The configuration is turned on. After the algorithm is turned on, AI outputs the PCM data processed by the algorithm.

7.2.2 Audio Codec

The audio codec is realized through the encapsulation of ffmpeg, and currently supports G711A/G711U/G726/MP2.

7.2.3 Audio Algorithm

Currently supports AEC algorithm for intercom scenes and ANR algorithm for recording scenes.

7.3 API reference

7.3.1 Audio input

7.3.1.1 RK_MPI_AI_EnableChn

【description】

Open the AI channel.

【grammar】

RK_S32 RK_MPI_AI_EnableChn([AI_CHN](#) AiChn);

【parameter】

parameter name	description	Input / output
AiChn	Audio input channel number. Value range: [0, AI_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_AI_DisableChn](#)

7.3.1.2 RK_MPI_AI_DisableChn

【description】

Close the AI channel.

【grammar】

RK_S32 RK_MPI_AI_DisableChn([AI_CHN](#) AiChn);

【parameter】

parameter name	description	Input / output
AiChn	Audio input channel number. Value range: [0, AI_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.

Non-zero Failure, see [error code for](#) its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_AI_EnableChn](#)

7.3.1.3 RK_MPI_AI_SetChnAttr

【description】

Set AO channel properties.

【grammar】

RK_S32 RK_MPI_AI_SetChnAttr([AI_CHN](#) AiChn, const [AI_CHN_ATTR_S](#) *pstAttr);

【parameter】

parameter name	description	Input / output
AiChn	Audio input channel number. Value range: [0, AI_MAX_CHN_NUM).	enter
pstAttr	AI channel attribute pointer.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

7.3.1.4 RK_MPI_AI_SetVolume

【description】

Set the volume.

【grammar】

RK_S32 RK_MPI_AI_SetVolume([AI_CHN](#) AiChn, RK_S32 s32Volume);

【parameter】

parameter name	description	Input / output
AiChn	Audio input channel number. Value range: [0, AI_MAX_CHN_NUM).	enter
s32Volume	The volume of the audio input channel. Value range: [0, 100].	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

7.3.1.5 RK_MPI_AI_GetVolume

【description】

Get the volume.

【grammar】

RK_S32 RK_MPI_AI_GetVolume([AI_CHN](#) AiChn, RK_S32 *ps32Volume);

【parameter】

parameter name	description	Input / output
AiChn	Audio input channel number. Value range: [0, AI_MAX_CHN_NUM).	enter
ps32Volume	The volume of the audio input channel.	Output

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

7.3.1.6 RK_MPI_AI_SetTalkVqeAttr

【description】

Set AI's sound quality enhancement function (Talk) related attributes.

【grammar】

RK_S32 RK_MPI_AI_SetTalkVqeAttr([AI_CHN](#) AiChn, [AI_TALKVQE_CONFIG_S](#) *pstVqeConfig);

【parameter】

parameter name	description	Input / output
AiChn	Audio input channel number. Value range: [0, AI_MAX_CHN_NUM).	enter
pstVqeConfig	Audio input sound quality enhancement configuration structure pointer.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

7.3.1.7 RK_MPI_AI_GetTalkVqeAttr

【description】

Get AI's sound quality enhancement function (Talk) related attributes.

【grammar】

RK_S32 RK_MPI_AI_GetTalkVqeAttr([AI_CHN](#) AiChn, [AI_TALKVQE_CONFIG_S](#) *pstVqeConfig);

【parameter】

parameter name	description	Input / output
AiChn	Audio input channel number. Value range: [0, AI_MAX_CHN_NUM).	enter

pstVqeConfig	Audio input sound quality enhancement configuration structure pointer.	Output
【return value】		
return value	description	
0	success.	
Non-zero	Failure, see error code for its value.	
【demand】		
Header file: rkmedia_api.h		
Library file: libeasymedia.so		
【note】		
no.		
【Example】		
no.		
【related topic】		
no.		

7.3.1.8 RK_MPI_AI_SetRecordVqeAttr

【description】		
Set AI's sound quality enhancement function (Record) related attributes.		
【grammar】		
RK_S32 RK_MPI_AI_SetRecordVqeAttr(AI_CHN AiChn, AI_RECORDVQE_CONFIG_S *pstVqeConfig);		
【parameter】		
parameter name	description	Input / output
AiChn	Audio input channel number. Value range: [0, AI_MAX_CHN_NUM).	enter
pstVqeConfig	Audio input sound quality enhancement configuration structure pointer.	enter
【return value】		
return value	description	
0	success.	
Non-zero	Failure, see error code for its value.	

【demand】		
Header file: rkmedia_api.h		
Library file: libeasymedia.so		
【note】		
no.		
【Example】		
no.		
【related topic】		
no.		

7.3.1.9 RK_MPI_AI_GetRecordVqeAttr

【description】

Get the attributes related to AI's sound quality enhancement function (Record).

【grammar】

RK_S32 RK_MPI_AI_GetRecordVqeAttr([AI_CHN](#) AiChn, [AI_RECORDVQE_CONFIG_S](#) *pstVqeConfig);

【parameter】

parameter name	description	Input / output
AiChn	Audio input channel number. Value range: [0, AI_MAX_CHN_NUM).	enter
pstVqeConfig	Audio input sound quality enhancement configuration structure pointer.	Output

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

7.3.1.10 RK_MPI_AI_EnableVqe

【description】

Enable AI's sound quality enhancement function.

【grammar】

RK_S32 RK_MPI_AI_EnableVqe([AI_CHN](#) AiChn);

【parameter】

parameter name	description	Input / output
AiChn	Audio input channel number. Value range: [0, AI_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

[related topic]

no.

7.3.1.11 RK_MPI_AI_DisableVqe

[description]

Disable AI's sound quality enhancement function.

[grammar]

RK_S32 RK_MPI_AI_DisableVqe([AI_CHN](#) AiChn);

[parameter]

parameter name	description	Input / output
AiChn	Audio input channel number. Value range: [0, AI_MAX_CHN_NUM).	enter

[return value]

return value	description
0	success.
Non-zero	Failure, see error code for its value.

[demand]

Header file: rkmedia_api.h

Library file: libeasymedia.so

[note]

no.

[Example]

no.

[related topic]

no.

7.3.2 Audio output

7.3.2.1 RK_MPI_AO_EnableChn

[description]

Open the AO channel.

[grammar]

RK_S32 RK_MPI_AO_EnableChn([AO_CHN](#) AoChn);

[parameter]

parameter name	description	Input / output
AoChn	Audio output channel number. Value range: [0, AO_MAX_CHN_NUM).	enter

[return value]

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】
Header file: rkmedia_api.h
Library file: libeasymedia.so

【note】
no.

【Example】
no.

【related topic】
[RK_MPI_AO_DisableChn](#)

7.3.2.2 RK_MPI_AO_DisableChn

【description】
Close the AO channel.

【grammar】
RK_S32 RK_MPI_AO_DisableChn([AO_CHN](#) AoChn);

【parameter】

parameter name	description	Input / output
AoChn	Audio output channel number. Value range: [0, AO_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】
Header file: rkmedia_api.h
Library file: libeasymedia.so

【note】
no.

【Example】
no.

【related topic】
[RK_MPI_AO_EnableChn](#)

7.3.2.3 RK_MPI_AO_SetChnAttr

【description】
Set AO channel properties.

【grammar】
RK_S32 RK_MPI_AO_SetChnAttr([AO_CHN](#) AoChn, const [AO_CHN_ATTR_S](#) *pstAttr);

【parameter】

parameter name	description	Input / output
AoChn	Audio output channel number. Value range: [0, AO_MAX_CHN_NUM).	enter
pstAttr	Audio output channel attribute pointer.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

7.3.2.4 RK_MPI_AO_SetVolume

【description】

Set the volume.

【grammar】

RK_S32 RK_MPI_AO_SetVolume([AO_CHN](#) AoChn, RK_S32 s32Volume);

【parameter】

parameter name	description	Input / output
AoChn	Audio output channel number. Value range: [0, AO_MAX_CHN_NUM).	enter
s32Volume	The volume of the audio output channel. Value range: [0, 100].	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

7.3.2.5 RK_MPI_AO_GetVolume

【description】

Get the volume.

【grammar】

RK_S32 RK_MPI_AO_GetVolume([AO_CHN](#) AoChn, RK_S32 *ps32Volume);

【parameter】

parameter name	description	Input / output
AoChn	Audio output channel number. Value range: [0, AO_MAX_CHN_NUM).	enter
ps32Volume	The volume of the audio output channel.	Output

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

7.3.2.6 RK_MPI_AO_SetVqeAttr

【description】

Set the properties of AO's sound quality enhancement function.

【grammar】

RK_S32 RK_MPI_AO_SetVqeAttr([AO_CHN](#) AoChn, [AO_VQE_CONFIG_S](#) *pstVqeConfig);

【parameter】

parameter name	description	Input / output
AoChn	Audio output channel number. Value range: [0, AO_MAX_CHN_NUM).	enter
pstVqeConfig	Audio output sound quality enhancement configuration structure pointer.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

7.3.2.7 RK_MPI_AO_GetVqeAttr

【description】

Get the properties related to the sound quality enhancement function of AO.

【grammar】

RK_S32 RK_MPI_AO_GetVqeAttr([AO_CHN](#) AoChn, [AO_VQE_CONFIG_S](#) *pstVqeConfig);

【parameter】

parameter name	description	Input / output
AoChn	Audio output channel number. Value range: [0, AO_MAX_CHN_NUM).	enter
pstVqeConfig	Audio output sound quality enhancement configuration structure pointer.	Output

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

no.

7.3.2.8 RK_MPI_AO_EnableVqe

【description】

Enable the sound quality enhancement function of AO.

【grammar】

RK_S32 RK_MPI_AO_EnableVqe([AO_CHN](#) AoChn);

【parameter】

parameter name	description	Input / output
----------------	-------------	----------------

AoChn	Audio output channel number. Value range: [0, AO_MAX_CHN_NUM).	enter
[return value]		

return value	description
0	success.
Non-zero	Failure, see error code for its value.

[demand]

Header file: rkmedia_api.h

Library file: libeasymedia.so

[note]

no.

[Example]

no.

[related topic]

no.

7.3.2.9 RK_MPI_AO_DisableVqe

[description]

Disable the sound quality enhancement function of AO.

[grammar]

RK_S32 RK_MPI_AO_DisableVqe([AO_CHN](#) AoChn);

【parameter】		
parameter name	description	Input / output
AoChn	Audio output channel number. Value range: [0, AO_MAX_CHN_NUM).	enter

[return value]	
return value	description
0	success.
Non-zero	Failure, see error code for its value.

[demand]

Header file: rkmedia_api.h

Library file: libeasymedia.so

[note]

no.

[Example]

no.
[related topic]
no.

7.3.3 Audio coding

7.3.3.1 RK_MPI_AENC_CreateChn

[description]

Create audio encoding channels.

[grammar]

RK_MPI_AENC_CreateChn([AENC_CHN](#) AencChn,const [AENC_CHN_ATTR_S](#) *pstAttr);

[parameter]

parameter name	description	Input / output
AencChn	Audio encoding channel number. Value range: [0, AENC_MAX_CHN_NUM).	enter
pstAttr	Audio encoding channel attribute pointer.	enter

[return value]

return value	description
0	success.
Non-zero	Failure, see error code for its value.

[demand]

Header file: rkmedia_api.h

Library file: libeasymedia.so

[note]

Currently supports protocols such as [audio codec](#)Shown.

[Example]

no.

[related topic]

[RK_MPI_AENC_DestroyChn](#)

7.3.3.2 RK_MPI_AENC_DestroyChn

[description]

Destroy the audio encoding channel.

[grammar]

RK_S32 RK_MPI_AENC_DestroyChn([AENC_CHN](#) AencChn);

[parameter]

parameter name	description	Input / output
AencChn	Audio encoding channel number. Value range: [0, AENC_MAX_CHN_NUM).	enter

[return value]

return value	description
0	success.

Non-zero Failure, see [error code for](#) its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_AENC_CreateChn](#)

7.3.4 Audio decoding

7.3.4.1 RK_MPI_ADEC_CreateChn

【description】

Create audio decoding channel.

【grammar】

RK_S32 RK_MPI_ADEC_CreateChn([ADEC_CHN](#) AdecChn, const [ADEC_CHN_ATTR_S](#) *pstAttr);

【parameter】

parameter name	description	Input / output
AdecChn	Audio decoding channel number. Value range: [0, ADEC_MAX_CHN_NUM).	enter
pstAttr	Audio decoding channel attribute pointer.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

Currently supports protocols such as [audio codec](#)Shown.

【Example】

no.

【related topic】

[RK_MPI_ADEC_DestroyChn](#)

7.3.4.2 RK_MPI_ADEC_DestroyChn

【description】

Destroy the audio decoding channel.

【grammar】

RK_S32 RK_MPI_ADEC_DestroyChn([ADEC_CHN](#) AdecChn);

【parameter】

parameter name	description	Input / output
AdecChn	Audio decoding channel number. Value range: [0, ADEC_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_ADEC_CreateChn](#)

7.4 Data Type

7.4.1 Audio input

The data types related to audio input are defined as follows:

[AI_MAX_CHN_NUM](#): The maximum number of audio input channels.

[AI_CHN](#): Audio input channel number.

[AI_CHN_ATTR_S](#): Audio input attribute structure.

[AI_TALKVOE_CONFIG_S](#): Audio input sound quality enhancement (Talk) configuration information structure.

[AI_RECORDVOE_CONFIG_S](#): Audio input sound quality enhancement (Record) configuration information structure.

7.4.1.1 AI_MAX_CHN_NUM

【Description】

The maximum number of audio input channels.

【definition】

```
RV1109 / RV1126:  
#define AI_MAX_CHN_NUM 1
```

7.4.1.2 AI_CHN

【Description】

Audio input channel number.

【definition】

```
typedef RK_S32 AI_CHN;
```

7.4.1.3 AI_CHN_ATTR_S

【Description】

Audio input attribute structure.

【definition】

```
typedef struct rkAI_CHN_ATTR_S {
    RK_CHAR * pcAudioNode;
    Sample_Format_E enSampleFormat;
    RK_U32 u32Channels;
    RK_U32 u32SampleRate;
    RK_U32 u32NbSamples;
} AI_CHN_ATTR_S;
```

【member】

Member name	description
pcAudioNode	Audio device node path.
enSampleFormat	Sampling format.
u32Channels	Number of channels.
u32SampleRate	Sampling Rate.
u32NbSamples	The number of sampling points per frame.

[Related data types and interfaces]

[Sample_Format_E](#)

7.4.1.4 AI_TALKVQE_CONFIG_S

【Description】

Audio input sound quality enhancement (Talk) configuration information structure.

【definition】

```
#define AI_TALKVQE_MASK_AEC 0x1
#define AI_TALKVQE_MASK_ANR 0x2
#define AI_TALKVQE_MASK_AGC 0x4

typedef struct rkAI_TALKVQE_CONFIG_S {
    RK_U32 u32OpenMask;
    RK_S32 s32WorkSampleRate;
    RK_S32 s32FrameSample;
    RK_CHAR aParamFilePath[MAX_FILE_PATH_LEN];
} AI_TALKVQE_CONFIG_S;
```

【member】

Member name	description
	Mask value enabled by each function of Talk Vqe.
u32OpenMask	Currently supports AI_TALKVQE_MASK_AEC, AI_TALKVQE_MASK_ANR, AI_TALKVQE_MASK_AGC.
s32WorkSampleRate	Working sampling frequency.
s32FrameSample	Number of sampling points.
aParamFilePath	Parameter file path.

[Related data types and interfaces]

[MAX_FILE_PATH_LEN](#)

7.4.1.5 AI_RECORDVQE_CONFIG_S

【Description】

Audio input sound quality enhancement (Record) configuration information structure.

【definition】

```
#define AI_RECORDVQE_MASK_ANR 0x1

typedef struct rkAI_RECORDVQE_CONFIG_S {
    RK_U32 u32OpenMask;
    RK_S32 s32WorkSampleRate;
    RK_S32 s32FrameSample;
    struct {
        RK_FLOAT fPostAddGain; /* post-gain 0*/
        RK_FLOAT fGmin; /* spectral gain floor,unit:(dB),default:-30dB */
        RK_FLOAT fNoiseFactor; /* noise suppression factor,default:0.98 */
    } stAnrConfig;
} AI_RECORDVQE_CONFIG_S;
```

【member】

Member name	description
	Mask value enabled by each function of Record Vqe. AI_RECORDVQE_MASK_ANR is currently supported.
u32OpenMask	
s32WorkSampleRate	Working sampling frequency.
s32FrameSample	Number of sampling points.
stAnrConfig.fPostAddGain	ANR's post-gain.
stAnrConfig.fGmin	The lower limit of the ANR spectrum gain, in dB, the default value is -30dB.
stAnrConfig.fNoiseFactor	ANR noise suppression coefficient, the default value is 0.98.

7.4.2 Audio output

The audio output related data types are defined as follows:

- [AO_MAX_CHN_NUM](#): The maximum number of audio output channels.
- [AO_CHN](#): Audio output channel number.
- [AO_CHN_ATTR_S](#): Audio output attribute structure.
- [AO_VQE_CONFIG_S](#): Audio output sound quality enhancement configuration information structure.

7.4.2.1 AO_MAX_CHN_NUM

【Description】

The maximum number of audio output channels.

【definition】

```
RV1109 / RV1126:
#define AO_MAX_CHN_NUM 1
```

7.4.2.2 AO_CHN

【Description】

Audio output channel number.

【definition】

```
typedef RK_S32 AO_CHN;
```

7.4.2.3 AO_CHN_ATTR_S

【Description】

Audio output attribute structure.

【definition】

```
typedef struct rkAO_CHN_ATTR_S {
    RK_CHAR * pcAudioNode;
    Sample_Format_E enSampleFormat;
    RK_U32 u32Channels;
    RK_U32 u32SampleRate;
    RK_U32 u32NbSamples;
} AO_CHN_ATTR_S;
```

【member】

Page 101

Member name	description
pcAudioNode	Audio device node path.
enSampleFormat	Sampling format.
u32Channels	Number of channels.
u32SampleRate	Sampling Rate.
u32NbSamples	The number of sampling points per frame.

[Related data types and interfaces]

[Sample_Format_E](#)

7.4.2.4 AO_VQE_CONFIG_S

【Description】

Audio output sound quality enhancement configuration information structure.

【definition】

```
#define AO_VQE_MASK_ANR 0x1
#define AO_VQE_MASK_AGC 0x2

typedef struct rkAO_VQE_CONFIG_S
```

```
{
    RK_U32 u32OpenMask;
    RK_S32 s32WorkSampleRate;
    RK_S32 s32FrameSample;
    RK_CHAR aParamFilePath[MAX_FILE_PATH_LEN];
} AO_VQE_CONFIG_S;
```

【member】

Member name	description
u32OpenMask	Mask value enabled by each function of AO Vqe. Currently supports AO_VQE_MASK_ANR, AO_VQE_MASK_AGC.
s32WorkSampleRate	Working sampling frequency.
s32FrameSample	Number of sampling points.
aParamFilePath	Parameter file path.

[Related data types and interfaces]

[MAX_FILE_PATH_LEN](#)

7.4.3 Audio coding

The audio coding related data types are defined as follows:

[AENC_MAX_CHN_NUM](#): The maximum number of audio coding channels.

[AENC_CHN](#): Audio encoding channel number.

[AENC_ATTR_AAC_S](#): AAC encoding protocol attribute structure.

[AENC_ATTR_MP2_S](#): MP2 encoding protocol attribute structure.

[AENC_ATTR_G711A_S](#): G.711A encoding protocol attribute structure.

[AENC_ATTR_G711U_S](#): G.711U encoding protocol attribute structure.

[AENC_ATTR_G726_S](#): G.726 encoding protocol attribute structure.

[AENC_CHN_ATTR_S](#): Audio coding attribute structure.

7.4.3.1 AENC_MAX_CHN_NUM

【Description】

The maximum number of audio encoding channels.

【definition】

```
RV1109 / RV1126:
#define AENC_MAX_CHN_NUM 16
```

7.4.3.2 AENC_CHN

【Description】

Audio encoding channel number.

【definition】

```
typedef RK_S32 AENC_CHN;
```

7.4.3.3 AENC_ATTR_AAC_S

【Description】

AAC encoding protocol attribute structure.

【definition】

```
typedef struct rkAENC_ATTR_AAC_S {
    RK_U32 u32Channels;
    RK_U32 u32SampleRate; // 96000, 88200, 64000, 48000, 44100, 32000,
                        // 24000, 22050, 16000, 12000, 11025, 8000, 7350
} AENC_ATTR_AAC_S;
```

【member】

Member name	description
u32Channels	Number of channels.
u32SampleRate	Sampling Rate. The value range is: 96000, 88200, 64000, 48000, 44100, 32000, 24000, 22050, 16000, 12000, 11025, 8000, 7350.

7.4.3.4 AENC_ATTR_MP2_S

【Description】

MP2 encoding protocol attribute structure.

【definition】

```
typedef struct rkAENC_ATTR_MP2_S {
    RK_U32 u32Channels;
    RK_U32 u32SampleRate; // 44100, 48000, 32000, 22050, 24000, 16000, 0
} AENC_ATTR_MP2_S;
```

【member】

Member name	description
u32Channels	Number of channels.
u32SampleRate	Sampling Rate. The value range is: 44100, 48000, 32000, 22050, 24000, 16000, 0.

7.4.3.5 AENC_ATTR_G711A_S

【Description】

G.711A encoding protocol attribute structure.

【definition】

```
typedef struct rkAENC_ATTR_G711A_S {
    RK_U32 u32Channels;
    RK_U32 u32SampleRate;
    RK_U32 u32NbSample;
} AENC_ATTR_G711A_S;
```

【member】

Member name	description
u32Channels	Number of channels.
u32SampleRate	Sampling Rate.
u32NbSample	The number of sampling points per frame.

7.4.3.6 AENC_ATTR_G711U_S

【Description】
G.711U encoding protocol attribute structure.

【definition】

```
typedef struct rkAENC_ATTR_G711U_S {  
    RK_U32 u32Channels;  
    RK_U32 u32SampleRate;  
    RK_U32 u32NbSample;  
} AENC_ATTR_G711U_S;
```

【member】

Member name	description
u32Channels	Number of channels.
u32SampleRate	Sampling Rate.
u32NbSample	The number of sampling points per frame.

7.4.3.7 AENC_ATTR_G726_S

【Description】
G.726 encoding protocol attribute structure.

【definition】

```
typedef struct rkAENC_ATTR_G726_S {  
    RK_U32 u32Channels;  
    RK_U32 u32SampleRate;  
} AENC_ATTR_G726_S;
```

【member】

Member name	description
u32Channels	Number of channels.
u32SampleRate	Sampling Rate.

7.4.3.8 AENC_CHN_ATTR_S

【Description】
Audio encoding attribute structure.

【definition】

```
typedef struct rkAENC_CHN_ATTR_S {  
    CODEC_TYPE_E enCodecType; /*payload type 0*/  
    RK_U32 u32Bitrate;  
    RK_U32 u32Quality;  
    union {  
        AENC_ATTR_AAC_S stAencAAC;  
        AENC_ATTR_MP2_S stAencMP2;  
        AENC_ATTR_G711A_S stAencG711A;  
        AENC_ATTR_G711U_S stAencG711U;  
        AENC_ATTR_G726_S stAencG726;  
    };  
};
```

} AENC_CHN_ATTR_S;	
【member】	
Member name	description
enCodecType	Encoding protocol type.
u32Bitrate	Bit rate.
u32Quality	Encoding quality.
stAencAAC/stAencMP2/stAencG711A/stAencG711U/stAencG726	Related coding protocol attribute structure.

[Related data types and interfaces]

[CODEC_TYPE_E](#)

7.4.4 Audio decoding

The data types related to audio decoding are defined as follows:

[ADEC_MAX_CHN_NUM](#): The maximum number of audio decoding channels.

[ADEC_CHN](#): Audio decoding channel number.

[ADEC_ATTR_AAC_S](#): AAC decoding protocol attribute structure.

[ADEC_ATTR_MP2_S](#): MP2 decoding protocol attribute structure.

[ADEC_ATTR_G711A_S](#): G.711A decoding protocol attribute structure.

[ADEC_ATTR_G711U_S](#): G.711U decoding protocol attribute structure.

[ADEC_ATTR_G726_S](#): G.726 decoding protocol attribute structure.

[ADEC_CHN_ATTR_S](#): Audio decoding attribute structure.

7.4.4.1 ADEC_MAX_CHN_NUM

【Description】

The maximum number of audio decoding channels.

【definition】

```
RV1109 / RV1126:
#define ADEC_MAX_CHN_NUM 16
```

7.4.4.2 ADEC_CHN

【Description】

Audio decoding channel number.

【definition】

```
typedef RK_S32 ADEC_CHN;
```

7.4.4.3 ADEC_ATTR_AAC_S

【Description】

AAC decoding protocol attribute structure.

【definition】

```
typedef struct rkADEC_ATTR_AAC_S {
    // reserved
```

```
} ADEC_ATTR_AAC_S;
```

7.4.4.4 ADEC_ATTR_MP2_S

【Description】

MP2 decoding protocol attribute structure.

【definition】

```
typedef struct rkADEC_ATTR_MP2_S {  
    // reserved  
} ADEC_ATTR_MP2_S;
```

7.4.4.5 ADEC_ATTR_G711A_S

【Description】

G.711A decoding protocol attribute structure.

【definition】

```
typedef struct rkADEC_ATTR_G711A_S {  
    RK_U32 u32Channels;  
    RK_U32 u32SampleRate;  
} ADEC_ATTR_G711A_S;
```

【member】

Page 107

Member name	description
u32Channels	Number of channels.
u32SampleRate	Sampling Rate.

7.4.4.6 ADEC_ATTR_G711U_S

【Description】

G.711U decoding protocol attribute structure.

【definition】

```
typedef struct rkADEC_ATTR_G711U_S {  
    RK_U32 u32Channels;  
    RK_U32 u32SampleRate;  
} ADEC_ATTR_G711U_S;
```

【member】

Member name	description
u32Channels	Number of channels.
u32SampleRate	Sampling Rate.

7.4.4.7 ADEC_ATTR_G726_S

【Description】

G.726 decoding protocol attribute structure.

【definition】

```
typedef struct rkADEC_ATTR_G726_S {  
    // reserved  
} ADEC_ATTR_G726_S;
```

7.4.4.8 ADEC_CHN_ATTR_S

【Description】

Audio decoding attribute structure.

【definition】

```
typedef struct rkADEC_CHN_ATTR_S {
    CODEC_TYPE_E enCodecType;
    union {
        ADEC_ATTR_AAC_S stAdecAAC;
        ADEC_ATTR_MP2_S stAdecMP2;
        ADEC_ATTR_G711A_S stAdecG711A;
        ADEC_ATTR_G711U_S stAdecG711U;
        ADEC_ATTR_G726_S stAdecG726;
    };
} ADEC_CHN_ATTR_S;
```

【member】

Member name	description
enCodecType	Encoding protocol type.
stAdecAAC/stAdecMP2/stAdecG711A/stAdecG711U/stAdecG726	Related decoding protocol attribute structure.

[Related data types and interfaces]

[CODEC_TYPE_E](#)

7.5 Error code

7.5.1 Audio input error code

Audio input API error codes such as [Table 7-1](#) Shown:

Table 7-1 Audio input API error codes

Wrong generation code	Macro definition	description
40	RK_ERR_AI_INVALID_DEVID	Invalid audio input device number
41	RK_ERR_AI_BUSY	Audio input system is busy
42	RK_ERR_AI_EXIST	Attempt to apply for or create an existing device, channel or capital source
43	RK_ERR_AI_NOTOPEN	The system is not turned on, has not been initialized or enabled
44	RK_ERR_AI_NOT_CONFIG	Not configured before use

7.5.2 Audio output error code

Audio output API error codes such as [Table 7-2](#) Shown:

Table 7-2 Audio output API error codes

Wrong generation code	Macro definition	description
50	RK_ERR_AO_INVALID_DEVID	Invalid audio output device number
51	RK_ERR_AO_BUSY	The audio output system is not initialized
52	RK_ERR_AO_NOTREADY	Attempt to apply for or create an existing device, channel or capital source
53	RK_ERR_AO_NOTOPEN	The system is not turned on, has not been initialized or enabled

7.5.3 Audio encoding error codes

Audio encoding API error codes such as [Table 7-3](#)Shown:

Table 7-3 Audio coding API error codes

error code	Macro definition	description
60	RK_ERR_AENC_INVALID_DEVID	Invalid audio encoding device number
61	RK_ERR_AENC_BUSY	Audio coding system is busy
62	RK_ERR_AENC_CODEC_NOT_SUPPORT	Audio encoding is not supported

7.5.4 Audio decoding error code

Audio decoding API error codes such as [Table 7-4](#)Shown:

Table 7-3 Audio decoding API error codes

error code	Macro definition	description
100	RK_ERR_ADEC_INVALID_DEVID	Invalid audio decoding device number
101	RK_ERR_ADEC_BUSY	Audio decoding system is busy
102	RK_ERR_ADEC_CODEC_NOT_SUPPORT	Audio decoding is not supported

8. RGA

8.1 Overview

The RGA module is used for 2D image cropping, format conversion, scaling, rotation, image overlay, etc.

8.2 Function description

The RGA channel in rkmedia only supports format conversion, scaling, cropping, and rotation functions. For image overlay, you need to call librga.so library separately.
See docs/Linux/Multimedia/《Rockchip_Developer_Guide_Linux_RGA_CN.pdf》

8.3 API reference

8.3.1 RK_MPI_RGA_CreateChn

【description】

Create RGA channel.

【grammar】

RK_S32 RK_MPI_RGA_CreateChn([RGA_CHN](#) RgaChn,[RGA_ATTR_S](#) *pstRgaAttr);

【parameter】

parameter name	description	Input / output
RgaChn	RGA channel number. Value range: [0, RGA_MAX_CHN_NUM).	enter
pstAttr	RGA channel attribute pointer.	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_RGA_DestroyChn](#)

8.3.2 RK_MPI_RGA_DestroyChn

【description】

Destroy the RGA channel.

【grammar】

RK_S32 RK_MPI_RGA_DestroyChn([RGA_CHN](#) RgaChn);

【parameter】

parameter name	description	Input / output
RgaChn	RGA channel number. Value range: [0, RGA_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】
Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPL_RGA_CreateChn](#)

8.4 Data Type

RGA related data types are defined as follows:

[RGA_MAX_CHN_NUM](#): The maximum number of RGA channels.

[RGA_CHN](#): RGA channel number.

[RGA_INFO_S](#): RGA area attribute structure.

[RGA_ATTR_S](#): RGA attribute structure.

8.4.1 RGA_MAX_CHN_NUM

【Description】
The maximum number of RGA channels.

【definition】

RV1109 / RV1126:
#define RGA_MAX_CHN_NUM 16

8.4.2 RGA_CHN

【Description】
RGA channel number.

【definition】

typedef RK_S32 RGA_CHN;

8.4.3 RGA_INFO_S

【Description】
RGA area attribute structure.

【definition】

typedef struct rkRGA_INFO_S {
 IMAGE_TYPE_E imgType;
 RK_U32 u32X;
 RK_U32 u32Y;
 RK_U32 u32Width;
 RK_U32 u32Height;
 RK_U32 u32HorStride; // horizontal stride

```
    RK_U32 u32VirStride; // virtual stride
} RGA_INFO_S;
```

【member】

Member name	description
imgType	Image format type.
u32X	The X-axis coordinate of RGA.
u32Y	The Y-axis coordinate of RGA.
u32Width	The width of the RGA.
u32Height	The height of the RGA.
u32HorStride	Imaginary wide.
u32VirStride	False height.

[Related data types and interfaces]

[IMAGE_TYPE_E](#)

8.4.4 RGA_ATTR_S

【Description】

RGA attribute structure.

【definition】

```
typedef struct rkRGA_ATTR_S {
    RGA_INFO_S stImgIn; // input image info
    RGA_INFO_S stImgOut; // output image info
    RK_U16 u16Rotaion; // support 0/90/180/270.
    RK_BOOL bEnBufPool;
    RK_U16 u16BufPoolCnt;
} RGA_ATTR_S;
```

【member】

Member name	description
stImgIn	Enter image information.
stImgOut	Output image information.
u16Rotaion	Rotation angle. Value range: 0, 90, 180, 270.
bEnBufPool	Enable the buffer pool.
u16BufPoolCnt	Buffer pool count.

[Related data types and interfaces]

[RGA_INFO_S](#)

8.5 Error code

RGA API error codes are shown in [Table 8-1](#) :

Table 8-1 RGA API error codes

Wrong generation code	Macro definition	description
90	RK_ERR_RGA_INVALID_CHNID	RGA input device number is invalid
91	RK_ERR_RGA_BUSY	RGA system is busy
92	RK_ERR_RGA_EXIST	Try to apply for or create an existing device, channel or Resources
93	RK_ERR_RGA_NOT_CONFIG	Not configured before use
94	RK_ERR_RGA_ILLEGAL_PARAM	Illegal parameter

9. Video output

9.1 Overview

The VO module is used for video output management.

9.2 Function description

The VO module is an encapsulation of DRM/KMS and supports multi-VOP and multi-layer display.

9.3 API reference

9.3.1 RK_MPI_VO_CreateChn

【description】
Create a VO channel.

【grammar】
RK_S32 RK_MPI_VO_CreateChn([VO_CHN](#) VoChn, const[VO_CHN_ATTR_S](#) *pstAttr);

【parameter】

parameter name	description	Input / output
VoChn	VO channel number. Value range: [0, VO_MAX_CHN_NUM).	enter
pstAttr	VO channel attribute pointer.	enter
【return value】		

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_VO_DestroyChn](#)

9.3.2 RK_MPI_VO_DestroyChn

【description】

Destroy the VO channel.

【grammar】

RK_S32 RK_MPI_VO_DestroyChn([VO_CHN](#) VoChn);

【parameter】

parameter name	description	Input / output
VoChn	VO channel number. Value range: [0, VO_MAX_CHN_NUM).	enter

【return value】

return value	description
0	success.
Non-zero	Failure, see error code for its value.

【demand】

Header file: rkmedia_api.h

Library file: libeasymedia.so

【note】

no.

【Example】

no.

【related topic】

[RK_MPI_VO_CreateChn](#)

9.4 Data Type

The video output related data types are defined as follows:

[VO_MAX_CHN_NUM](#): The maximum number of video output channels.

[VO_CHN](#): Video output channel number.

[VO_CHN_ATTR_S](#): Video output attribute structure.

9.4.1 VO_MAX_CHN_NUM

【Description】
The maximum number of video output channels.

【definition】

RV1109 / RV1126:
#define VO_MAX_CHN_NUM 2

9.4.2 VO_CHN

【Description】
Video output channel number.

【definition】

typedef RK_S32 VO_CHN;

9.4.3 VO_CHN_ATTR_S

【Description】
Video output attribute structure.

【definition】

```
typedef struct rkVO_CHN_ATTR_S {  
    RK_U32 u32Width;  
    RK_U32 u32Height;  
    RK_U32 u32VerStride;  
    RK_U32 u32HorStride;  
    IMAGE_TYPE_E enImgType;  
    RK_U16 u16Fps;  
    RK_U16 u16Zpos;  
} VO_CHN_ATTR_S;
```

Member name	description
u32Width	Video output width.
u32Height	Video output height.
u32VerStride	Imaginary wide.
u32HorStride	False height.
enImgType	Image format type.
u16Fps	Frame rate.
u16Zpos	Layer selection.

[Related data types and interfaces]

[IMAGE_TYPE_E](#)

9.5 Error code

Video output API error codes such as [Table 9-1](#)Shown:

Table 9-1 RGA API error codes

Wrong generation code	Macro definition	description
110	RK_ERR_VO_INVALID_DEVID	Device ID is out of legal range
111	RK_ERR_VO_EXIST	Try to apply for or create an existing device, channel or Resources
112	RK_ERR_VO_NOT_CONFIG	Not configured before use
113	RK_ERR_VO_TIMEOUT	Video output timeout
114	RK_ERR_VO_BUF_EMPTY	Video output buffer is empty
115	RK_ERR_VO_ILLEGAL_PARAM	Illegal parameter
116	RK_ERR_VO_NOTREADY	The system is not initialized

10. Restrictions on use

It is important to note that rkmedia has special requirements for the destruction order of modules: the subsequent modules in the data flow pipeline must be pinned before the previous modules. destroy. such as:
VI --> RGA --> VENC
The recommended order of destruction is as follows:
destroy VENC
destroy RGA
destroy VI
Take VI as an example. VI is the data generator. The buffer produced by the data pipeline may be occupied by the subsequent stage when the data pipeline is destroyed, resulting in the resources managed by VI also occupied. When you open it again, you will encounter Device Busy error. This problem may occur when frequently creating and destroying data channels.