

Rockchip RKADK Development Guide

文件标识：RK-KF-YF-904

发布版本：V2.2.3

日期：2024-09-11

文件密级：☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供，瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有 © 2024 瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园A区18号

网址：www.rock-chips.com

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：fae@rock-chips.com

前言

概述

本文主要描述了Rkadm 组件开发参考。

产品版本

芯片名称	内核版本
RV1126/RV1109	Linux 4.19
RV1106/RV1103	Linux 5.10

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V1.0.0	CTF	2021-05-02	初始版本
V1.1.0	CTF	2021-08-23	1、增加API说明： (1) RKADK_RECORD_GetAencChn (2) RKADK_GetThmInMp4 (3) RKADK_PHOTO_GetThmInJpg (4) RKADK_PARAM_GetVencChnId 2、新增模块说明： (1)、本地预览模块Display (2)、直播模块Live：包含Rtsp和Rtmp子模块
V1.2.0	CTF	2021-09-26	1、新增获取缩略图扩展API： (1) RKADK_GetThmInMp4Ex (2) RKADK_ThmBufFree (3) RKADK_PHOTO_GetThmInJpgEx (4) RKADK_PHOTO_ThumbBufFree 2、支持多路Sensor 3、支持设置INI文件路径 4、新增VI ISP 模块 5、支持设置VENC GOP
V1.3.0	GZC	2021-11-30	新增存储模块
V1.3.1	CTF	2021-12-20	1、增加API说明： (1) RKADK_PHOTO_GetData (2) RKADK_PHOTO_FreeData (3) RKADK_PLAYER_GetDuration 2、新增INI配置注意事项
V2.0.0	CTF	2023-01-29	1、适配通用Linux SDK 2、增加API说明： (1) RKADK_OSD_Init (2) RKADK_OSD_Deinit (3) RKADK_OSD_UpdateOsdSize (4) RKADK_OSD_AttachToStream (5) RKADK_OSD_DettachFromStream (6) RKADK_OSD_UpdateDisplayAttr
V2.1.0	CTF	2023-04-27	兼容RV1109/RV1126
V2.2.0	CTF	2023-11-07	1、增加UI叠加模块 2、增加JPEG Slice功能 3、Player 新增如下功能： (1) 支持使用第三方demuxer库 (2) 支持Rtsp网络流播放 (3) 支持屏幕快照功能 (4) 兼容RV1106/RV1103、RK3308、RK3506平台 4、增加Record File Cache 说明 5、增加Record/Photo rotation、flip、mirror API说明 6、更新ini说明 7、更新测试用例

版本号	作者	修改日期	修改说明
V2.2.1	CTF	2024-01-04	1、增加Post AI ISP功能
V2.2.2	CTF	2024-04-19	1、增加 AOV(Always On Video) 缩时录影功能
V2.2.3	CTF	2024-09-11	1、增加画中画(Picture-In-Picture) 录像功能 2、增加JPEG FBC0/NV16 编码支持 3、增加API说明： (1) RKADK_RECORD_Single_Start (2) RKADK_RECORD_Single_Stop (3) RKADK_RECORD_SetPipAttr (4) RKADK_RECORD_FileCacheSetMode (5) RKADK_PLAYER_GetSendFrameNum (6) RKADK_PLAYER_SetVdecWaterline (7) RKADK_PLAYER_SetAoVolume

目录

Rockchip RKADK Development Guide

1. 系统概述
 - 1.1 版本说明
2. 录像
 - 2.1 概述
 - 2.2 API参考
 - 2.2.1 RKADK_RECORD_Create
 - 2.2.2 RKADK_RECORD_Destroy
 - 2.2.3 RKADK_RECORD_Start
 - 2.2.4 RKADK_RECORD_Stop
 - 2.2.5 RKADK_RECORD_Single_Start
 - 2.2.6 RKADK_RECORD_Single_Stop
 - 2.2.7 RKADK_RECORD_Reset
 - 2.2.8 RKADK_RECORD_ManualSplit
 - 2.2.9 RKADK_RECORD_GetAencChn
 - 2.2.10 RKADK_GetThmInMp4
 - 2.2.11 RKADK_GetThmInMp4Ex
 - 2.2.12 RKADK_ThmBufFree
 - 2.2.13 RKADK_RECORD_SetRotation
 - 2.2.14 RKADK_RECORD_ToggleMirror
 - 2.2.15 RKADK_RECORD_ToggleFlip
 - 2.2.16 RKADK_RECORD_FileCacheInit
 - 2.2.17 RKADK_RECORD_FileCacheDeInit
 - 2.2.18 RKADK_RECORD_FileCacheSetMode
 - 2.2.19 RKADK_MEDIA_EnablePostIsp
 - 2.2.20 RKADK_MEDIA_SetPostIspAttr
 - 2.2.21 RKADK_RECORD_SetPipAttr
 - 2.3 数据类型
 - 2.3.1 公共数据类型
 - 2.3.2 RKADK_MW_PTR
 - 2.3.3 RKADK_MAX_SENSOR_CNT
 - 2.3.4 RECORD_FILE_NUM_MAX
 - 2.3.5 RKADK_MUXER_EVENT_E
 - 2.3.6 RKADK_MUXER_FILE_EVENT_INFO_S
 - 2.3.7 RKADK_MUXER_EVENT_INFO_S
 - 2.3.8 RKADK_REC_EVENT_CALLBACK_FN
 - 2.3.9 RKADK_REC_TYPE_E
 - 2.3.10 RKADK_REC_REQUEST_FILE_NAMES_FN
 - 2.3.11 RKADK_MOUMNT_SDCARD_FN
 - 2.3.12 RKADK_RECORD_ATTR_S
 - 2.3.13 RKADK_MUXER_MANUAL_SPLIT_TYPE_E
 - 2.3.14 RKADK_REC_MANUAL_SPLIT_ATTR_S
 - 2.3.15 FILE_CACHE_ARG
 - 2.3.16 FILE_WRITE_THREAD_ARG
 - 2.3.17 FILE_SDCARD_ARG
 - 2.3.18 RKADK_POST_ISP_ATTR_S
 - 2.3.19 RKADK_AOV_ATTR_S
 - 2.3.20 RKADK_PIP_ATTR_S
3. AOV(Always On Video)
 - 3.1 概述
 - 3.2 API 参考
 - 3.2.1 RKADK_AOV_Init
 - 3.2.2 RKADK_AOV_DeInit
 - 3.2.3 RKADK_AOV_SetSuspendTime
 - 3.2.4 RKADK_AOV_EnterSleep

- 3.2.5 RKADK_AOV_WakeupLock
 - 3.2.6 RKADK_AOV_WakeupUnlock
- 3.3 数据类型
 - 3.3.1 RKADK_AOV_ARG_S
 - 3.3.2 RKADK_AOV_NOTIFY_CALLBACK
 - 3.3.3 RKADK_AOV_EVENT_E
- 4. 拍照
 - 4.1 概述
 - 4.2 API 参考
 - 4.2.1 RKADK_PHOTO_Init
 - 4.2.2 RKADK_PHOTO_DeInit
 - 4.2.3 RKADK_PHOTO_TakePhoto
 - 4.2.4 RKADK_PHOTO_Reset
 - 4.2.5 RKADK_PHOTO_GetThmInJpg
 - 4.2.6 RKADK_PHOTO_GetThmInJpgEx
 - 4.2.7 RKADK_PHOTO_ThumbBufFree
 - 4.2.8 RKADK_MEDIA_SetVencRotation
 - 4.2.9 RKADK_MEDIA_ToggleVencMirror
 - 4.2.10 RKADK_MEDIA_ToggleVencFlip
 - 4.3 数据类型
 - 4.3.1 RKADK_PHOTO_TYPE_E
 - 4.3.2 RKADK_PHOTO_SINGLE_ATTR_S
 - 4.3.3 RKADK_PHOTO_MULTIPLE_ATTR_S
 - 4.3.4 RKADK_PHOTO_THUMB_ATTR_S
 - 4.3.5 RKADK_PHOTO_RECV_DATA_S
 - 4.3.6 RKADK_PHOTO_DATA_RECV_FN_PTR
 - 4.3.7 RKADK_TAKE_PHOTO_ATTR_S
 - 4.3.8 RKADK_PHOTO_FMT_CHANGE_S
 - 4.3.9 RKADK_PHOTO_ATTR_S
 - 4.3.10 RKADK_JPG_THUMB_TYPE_E
 - 4.3.11 RKADK_THUMB_TYPE_E
 - 4.3.12 RKADK_THUMB_ATTR_S
 - 4.3.13 ROTATION_E
- 5. 远程预览
 - 5.1 概述
 - 5.2 API参考
 - 5.2.1 RKADK_STREAM_VideoInit
 - 5.2.2 RKADK_STREAM_VideoDeInit
 - 5.2.3 RKADK_STREAM_VencStart
 - 5.2.4 RKADK_STREAM_VencStop
 - 5.2.5 RKADK_STREAM_GetVideoInfo
 - 5.2.6 RKADK_STREAM_AudioInit
 - 5.2.7 RKADK_STREAM_AudioDeInit
 - 5.2.8 RKADK_STREAM_AencStart
 - 5.2.9 RKADK_STREAM_AencStop
 - 5.2.10 RKADK_STREAM_GetAudioInfo
 - 5.3 数据类型
 - 5.3.1 RKADK_CODEC_TYPE_E
 - 5.3.2 RKADK_VENC_DATA_PROC_FUNC
 - 5.3.3 RKADK_VIDEO_STREAM_S
 - 5.3.4 RKADK_VENC_DATA_PACK_S
 - 5.3.5 RKADK_VENC_DATA_TYPE_S
 - 5.3.6 RKADK_VIDEO_INFO_S
 - 5.3.7 RKADK_STREAM_VIDEO_ATTR_S
 - 5.3.8 RKADK_AUDIO_DATA_PROC_FUNC
 - 5.3.9 RKADK_AUDIO_STREAM_S
 - 5.3.10 RKADK_AUDIO_INFO_S
 - 5.3.11 RKADK_STREAM_AUDIO_ATTR_S

6. 播放器

6.1 概述

6.2 API 参考

- 6.2.1 RKADK_PLAYER_Create
- 6.2.2 RKADK_PLAYER_Destroy
- 6.2.3 RKADK_PLAYER_SetDataSource
- 6.2.4 RKADK_PLAYER_SetDataParam
- 6.2.5 RKADK_PLAYER_Prepere
- 6.2.6 RKADK_PLAYER_GetCurrentPosition
- 6.2.7 RKADK_PLAYER_Play
- 6.2.8 RKADK_PLAYER_Stop
- 6.2.9 RKADK_PLAYER_Pause
- 6.2.10 RKADK_PLAYER_Seek
- 6.2.11 RKADK_PLAYER_GetPlayStatus
- 6.2.12 RKADK_PLAYER_GetDuration
- 6.2.13 RKADK_PLAYER_Snapshot
- 6.2.14 RKADK_PLAYER_SendAudioPacket
- 6.2.15 RKADK_PLAYER_SendVideoPacket
- 6.2.16 RKADK_PLAYER_GetSendFrameNum
- 6.2.17 RKADK_PLAYER_SetVdecWaterline
- 6.2.18 RKADK_PLAYER_SetAoVolume

6.3 数据类型

- 6.3.1 RKADK_PLAYER_EVENT_E
- 6.3.2 RKADK_PLAYER_EVENT_FN
- 6.3.3 RKADK_PLAYER_CFG_S
- 6.3.4 RKADK_VO_FORMAT_E
- 6.3.5 RKADK_VO_INTF_TYPE_E
- 6.3.6 RKADK_VO_SPLICE_MODE_E
- 6.3.7 RKADK_PLAYER_FRAME_INFO_S
- 6.3.8 RKADK_PLAYER_STATE_E
- 6.3.9 RKADK_PLAYER_SNAPSHOT_S
- 6.3.10 RKADK_PPLAYER_SNAPSHOT_RECV_FN
- 6.3.11 RKADK_PLAYER_SNAPSHOT_CFG_S
- 6.3.12 RKADK_PLAYER_VDEC_CFG_S
- 6.3.13 RKADK_PLAYER_RTSP_CFG_S
- 6.3.14 RKADK_PLAYER_PACKET
- 6.3.15 RKADK_PLAYER_DATA_PARAM_S

7. 直播

7.1 概述

7.2 API参考

7.2.1 RTSP

- 7.2.1.1 RKADK_RTSP_Init
- 7.2.1.2 RKADK_RTSP_DeInit
- 7.2.1.3 RKADK_RTSP_Start
- 7.2.1.4 RKADK_RTSP_Stop

7.2.2 RTMP

- 7.2.2.1 RKADK_RTMP_Init
- 7.2.2.2 RKADK_RTMP_DeInit

8. 存储

8.1 概述

8.2 API参考

- 8.2.1 RKADK_STORAGE_Init
- 8.2.2 RKADK_STORAGE_Deinit
- 8.2.3 RKADK_STORAGE_GetDevAttr
- 8.2.4 RKADK_STORAGE_GetMountStatus
- 8.2.5 RKADK_STORAGE_GetCapacity
- 8.2.6 RKADK_STORAGE_GetFileList
- 8.2.7 RKADK_STORAGE_FreeFileList

- 8.2.8 RKADK_STORAGE_GetFileNum
 - 8.2.9 RKADK_STORAGE_GetDevPath
 - 8.2.10 RKADK_STORAGE_Format
- 8.3 数据类型
 - 8.3.1 RKADK_MOUNT_STATUS
 - 8.3.2 RKADK_SORT_TYPE
 - 8.3.3 RKADK_SORT_CONDITION
 - 8.3.4 RKADK_STR_FOLDER_ATTR
 - 8.3.5 RKADK_STR_DEV_ATTR
 - 8.3.6 RKADK_FILE_INFO
 - 8.3.7 RKADK_FILE_LIST
 - 8.3.8 RKADK_FILE_LIST_ARRAY
- 9. 本地预览
 - 9.1 概述
 - 9.2 API参考
 - 9.2.1 RKADK_DISP_Init
 - 9.2.2 RKADK_DISP_DeInit
 - 9.2.3 RKADK_DISP_SetAttr
 - 9.3 数据类型
 - 9.3.1 RKADK_DISP_ATTR_S
- 10. 水印
 - 10.1 概述
 - 10.2 API 参考
 - 10.2.1 RKADK_OSD_Init
 - 10.2.2 RKADK_OSD_Deinit
 - 10.2.3 RKADK_OSD_UpdateBitMap
 - 10.2.4 RKADK_OSD_AttachToStream
 - 10.2.5 RKADK_OSD_DettachFromStream
 - 10.2.6 RKADK_OSD_UpdateOsdSize
 - 10.2.7 RKADK_OSD_UpdateDisplayAttr
 - 10.3 数据类型
 - 10.3.1 RKADK_OSD_ATTR_S
 - 10.3.2 RKADK_OSD_STREAM_ATTR_S
 - 10.4 RKADK_OSD_TYPE_E
- 11. UI叠加
 - 11.1 概述
 - 11.2 API参考
 - 11.2.1 RKADK_UI_Create
 - 11.2.2 RKADK_UI_Destroy
 - 11.2.3 RKADK_UI_Update
 - 11.3 数据类型
 - 11.3.1 RKADK_UI_ATTR_S
 - 11.3.2 RKADK_UI_FRAME_INFO
 - 11.3.3 RKADK_FORMAT_E
- 12. 参数设置
 - 12.1 概述
 - 12.2 API参考
 - 12.2.1 RKADK_PARAM_Init
 - 12.2.2 RKADK_PARAM_GetCamParam
 - 12.2.3 RKADK_PARAM_SetCamParam
 - 12.2.4 RKADK_PARAM_GetCommParam
 - 12.2.5 RKADK_PARAM_SetCommParam
 - 12.2.6 RKADK_PARAM_SetDefault
 - 12.2.7 RKADK_PARAM_GetResolution
 - 12.2.8 RKADK_PARAM_GetResType
 - 12.2.9 RKADK_PARAM_GetVencChnId
 - 12.3 数据类型
 - 12.3.1 RKADK_DEFPARAM_PATH

- 12.3.2 RKADK_DEFPARAM_PATH_SENSOR_PREFIX
- 12.3.3 RKADK_PARAM_PATH
- 12.3.4 RKADK_PARAM_PATH_SENSOR_PREFIX
- 12.3.5 RKADK_PARAM_TYPE_E
- 12.3.6 RKADK_PARAM_RES_E
- 12.3.7 RKADK_STREAM_TYPE_E
- 12.3.8 RKADK_PARAM_CODEC_CFG_S
- 12.3.9 RKADK_PARAM_BITRATE_S
- 12.3.10 RKADK_PARAM_REC_TIME_S
- 12.3.11 RKADK_PARAM_GOP_S
- 12.3.12 RKADK_VQE_MODE_E
- 12.3.13 RKADK_MUXER_FILE_TYPE_E
- 12.3.14 RKADK_MUXER_PRE_RECORD_MODE_E
- 12.3.15 RKADK_MIC_TYPE_E
- 12.4 INI文件解析
 - 12.4.1 全局INI配置文件
 - 12.4.2 Sensor INI配置文件
 - 12.4.3 INI 配置注意事项
- 13. 示例
 - 13.1 rkadk_record_test
 - 13.2 rkadk_photo_test
 - 13.3 rkadk_stream_test
 - 13.4 rkadk_player_test
 - 13.5 rkadk_thumb_test
 - 13.6 rkadk_rtsp_test
 - 13.7 rkadk_rtmp_test
 - 13.8 rkadk_storage_test
 - 13.9 rkadk_disp_test
 - 13.10 rkadk_ui_test

1. 系统概述

rkadk提供了基础通用组件，如录像、拍照、播放、预览等，简化了应用开发难度，支持快速开发音视频录制相关应用软件。

该组件仅支持单进程的功能实现，如无特殊说明，不支持多进程同时使用。

1.1 版本说明

- 1.x.x版本：在rkmedia和rockit的基础上做的进一步封装，对应master仓库。
- 2.x.x版本：在rockit的基础上做的进一步封装，对应develop仓库。

2. 录像

2.1 概述

提供基本的录像功能，向产品层提供如下功能：

- 录像任务的创建与销毁
- 录像任务的启动与停止
- 手动切分录像文件
- 缩时录像
- 预录像
- 画中画录像

录像任务通过参数模块获取Video和Audio信息，启停VENC，启停AENC；调用封装模块创建录像文件，写帧到文件。

每个录像任务对应一个或多个录像文件，每个文件必须对应一路视频编码通道，如果需要录制音频，需要加上一路音频编码通道。

同一录像任务下的多个录像文件，具有相同的录像类型，相同的切分条件，缩时录像模式下还具有相同的缩时间隔；可以配置不同的录像时间。

Post AI ISP：RV1106/RV1126/RV1109芯片支持通过AI ISP对VI输出图片进行微光降噪及智能增强处理，可在暗光、弱光环境下，依然呈现出无拖影、低噪点、更清晰的画面。

2.2 API参考

2.2.1 RKADK_RECORD_Create

【描述】

创建录像任务。

【语法】

RKADK_S32 RKADK_RECORD_Create([RKADK_RECORD_ATTR_S](#) *pstRecAttr, [RKADK_MW_PTR](#) *ppRecorder);

【参数】

参数名称	描述	输入/输出
pstRecAttr	录像任务属性	输入
ppRecorder	创建的录像任务指针	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- 每个录像任务最大支持同时录制2个录像文件。
- 每个录像文件最少有1路视频流，最大支持同时封装1路视频流和1路音频流。
- 不支持重复创建同一任务。
- 创建录像任务后，需要调用RKADK_RECORD_Start才会开始录像。
- 如果需要使能画中画，必须在RKADK_RECORD_Create时，就通过stPipAttr先使能，才会创建画中画通路。

【举例】

[rkadk_record_test](#)。

【相关主题】

[RKADK_RECORD_Destroy](#)。

2.2.2 RKADK_RECORD_Destroy

【描述】

销毁录像任务。

【语法】

RKADK_S32 RKADK_RECORD_Destroy([RKADK_MW_PTR](#) pRecorder);

【参数】

参数名称	描述	输入/输出
pRecorder	录像任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- 创建录像任务后，才能使用RKADK_RECORD_Destroy接口。
- 只能销毁已创建的录像任务，不支持重复销毁同一录像任务。

【举例】

[rkadk_record_test](#)。

【相关主题】

[RKADK_RECORD_Create](#)

2.2.3 RKADK_RECORD_Start

【描述】

启动录像任务。

【语法】

RKADK_S32 RKADK_RECORD_Start([RKADK_MW_PTR](#) pRecorder);

【参数】

参数名称	描述	输入/输出
pRecorder	录像任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- 创建录像任务后，才能使用RKADK_RECORD_Start接口。
- 支持在停止录像任务后重新启动录像任务。

【举例】

[rkadk_record_test](#)。

【相关主题】

[RKADK_RECORD_Stop](#)

2.2.4 RKADK_RECORD_Stop

【描述】

停止录像任务。

【语法】

RKADK_S32 RKADK_RECORD_Stop([RKADK_MW_PTR](#) pRecorder);

【参数】

参数名称	描述	输入/输出
pRecorder	录像任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- 创建录像任务后，才能使用RKADK_RECORD_Stop接口。
- 不支持重复停止同一录像任务。

【举例】

[rkadk_record_test](#)。

【相关主题】

[RKADK_RECORD_Start](#)

2.2.5 RKADK_RECORD_Single_Start

【描述】

启动指定码流的录像任务。

【语法】

RKADK_S32 RKADK_RECORD_Single_Start([RKADK_MW_PTR](#) pRecorder, [RKADK_STREAM_TYPE_E](#) enStrmType);

【参数】

参数名称	描述	输入/输出
pRecorder	录像任务指针	输入
enStrmType	码流类型	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- 创建录像任务后，才能使用RKADK_RECORD_Single_Start接口。
- 和RKADK_RECORD_Single_Stop配套使用。
- 支持在停止录像任务后重新启动录像任务。

【举例】

[rkadk_record_test](#)。

【相关主题】

[RKADK_RECORD_Single_Stop](#)

2.2.6 RKADK_RECORD_Single_Stop

【描述】

停止指定码流的录像任务。

【语法】

RKADK_S32 RKADK_RECORD_Single_Stop([RKADK_MW_PTR](#) pRecorder, [RKADK_STREAM_TYPE_E](#) enStrmType);

【参数】

参数名称	描述	输入/输出
pRecorder	录像任务指针	输入
enStrmType	码流类型	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- 创建录像任务后，才能使用RKADK_RECORD_Stop接口。
- 和RKADK_RECORD_Single_Start配套使用。
- 不支持重复停止同一录像任务。

【举例】

[rkadk_record_test](#)。

【相关主题】

[RKADK_RECORD_Single_Start](#)

2.2.7 RKADK_RECORD_Reset

【描述】

重新配置录像任务。

【语法】

RKADK_S32 RKADK_RECORD_Reset([RKADK_MW_PTR](#) *ppRecorder);

【参数】

参数名称	描述	输入/输出
ppRecorder	录像任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- 创建录像任务后，才能使用RKADK_RECORD_Reset 接口。
- 切换分辨率、帧率、比特率、编码类型、录像类型时，结合RKADK_PARAM_SetCamParam API重置ini文件中Record模块的参数。
- RV1126/RV1109 不支持动态切换分辨率和编码类型，,需要先RKADK_RECORD_Destroy，然后配置新的分辨率或者编码类型，再重新RKADK_RECORD_Create。
- 如果Photo和Record复用VI 通道，需要同时切换Record和Photo的分辨率，否则会导致VI和VENC分辨率不匹配。

【举例】

[rkadk_record_test](#)。

【相关主题】

无

2.2.8 RKADK_RECORD_ManualSplit

【描述】

手动切分录像文件。

【语法】

RKADK_S32 RKADK_RECORD_ManualSplit([RKADK_MW_PTR](#) pRecorder,
[RKADK_REC_MANUAL_SPLIT_ATTR_S](#) *pstSplitAttr);

【参数】

参数名称	描述	输入/输出
pRecorder	录像任务指针	输入
pstSplitAttr	手动切分属性参数	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- 创建录像任务后，才能使用RKADK_RECORD_ManualSplit接口。
- 支持在手动切分录像文件未结束时，重复手动切分录像文件。

【举例】

[rkadk_record_test](#)。

【相关主题】

无

2.2.9 RKADK_RECORD_GetAencChn

【描述】

获取录像AENC通道号。

【语法】

```
RKADK_S32 RKADK_RECORD_GetAencChn();
```

【返回值】

录像所使用的AENC通道号。

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

2.2.10 RKADK_GetThmInMp4

【描述】

从MP4文件中获取缩略图数据。

【语法】

```
RKADK_S32 RKADK_GetThmInMp4(RKADK_U32 u32CamId, RKADK_CHAR *pszFileName,  
RKADK_U8 *pu8Buf,  
RKADK_U32 *pu32Size);
```

【参数】

参数名称	描述	输入/输出
u32CamId	Camera ID	输入
pszFileName	MP4文件路径	输入
pu8Buf	输入：缩略图数据存储指针，输出：实际缩略图数据	输入/输出
pu32Size	输入：pu8Buf长度，输出：实际缩略图数据长度	输入/输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_thumb.h

库文件：librkadk.so

【举例】

[rkadk_thumb_test](#)

【相关主题】

无

2.2.11 RKADK_GetThmInMp4Ex

【描述】

MP4获取缩略图扩展接口，从MP4文件中获取缩略图数据，支持指定输出缩略图的类型和分辨率。必须和[RKADK_ThmBufFree](#)配套使用。

【语法】

RKADK_S32 RKADK_GetThmInMp4Ex(RKADK_U32 u32CamId, RKADK_CHAR *pszFileName, [RKADK_THUMB_ATTR_S](#) *pstThumbAttr);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera ID	输入
pszFileName	MP4文件路径	输入
pstThumbAttr	缩略图属性结构体指针	输入/输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_thumb.h

库文件：librkadk.so

【举例】

[rkadk_thumb_test](#)

【相关主题】

[RKADK_ThmBufFree](#)

2.2.12 RKADK_ThmBufFree

【描述】

释放[RKADK_GetThmInMp4Ex](#)申请的内存。必须和[RKADK_GetThmInMp4Ex](#)配套使用。

【语法】

RKADK_S32 RKADK_ThmBufFree([RKADK_THUMB_ATTR_S](#) *pstThumbAttr);

【参数】

参数名称	描述	输入/输出
pstThumbAttr	缩略图属性结构体指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_thumb.h

库文件：librkadk.so

【举例】

[rkadk_thumb_test](#)

【相关主题】

[RKADK_GetThmInMp4Ex](#)

2.2.13 RKADK_RECORD_SetRotation

【描述】

设置Record旋转。

【语法】

RKADK_S32 RKADK_RECORD_SetRotation([RKADK_MW_PTR](#) pRecorder, [ROTATION_E](#) enRotation, [RKADK_STREAM_TYPE_E](#) enStreamType);

【参数】

参数名称	描述	输入/输出
pRecorder	录像任务指针	输入
enRotation	旋转角度	输入
enStreamType	码流类型	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【举例】

[rkadk_record_test](#)

2.2.14 RKADK_RECORD_ToggleMirror

【描述】

设置Record Mirror。

【语法】

RKADK_S32 RKADK_RECORD_ToggleMirror([RKADK_MW_PTR](#) pRecorder, [RKADK_STREAM_TYPE_E](#) enStrmType, bool mirror);

【参数】

参数名称	描述	输入/输出
pRecorder	录像任务指针	输入
enStreamType	码流类型	输入
mirror	是否使能mirror	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

2.2.15 RKADK_RECORD_ToggleFlip

【描述】

设置Record Flip。

【语法】

RKADK_S32 RKADK_RECORD_ToggleFlip([RKADK_MW_PTR](#) pRecorder, [RKADK_STREAM_TYPE_E](#) enStrmType, bool flip);

【参数】

参数名称	描述	输入/输出
pRecorder	录像任务指针	输入
enStreamType	码流类型	输入
flip	是否使能flip	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

2.2.16 RKADK_RECORD_FileCacheInit

【描述】

File Cache初始化，旨在使写文件更平滑，使能File Cache之后，文件通过DirectIO方式写入存储设备，内核中的Cache机制不生效。

【语法】

RKADK_S32 RKADK_RECORD_FileCacheInit([FILE_CACHE_ARG](#) *pstFileCacheAttr)

【参数】

参数名称	描述	输入/输出
pstFileCacheAttr	File Cache属性指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- File Cache默认不开启。
- 使能File Cache，必须设置file_cache_env环境变量

```
export file_cache_env=1
```

- 如果使能File Cache，则须在RKADK_RECORD_Create之前调用该接口，且只需调用一次，否则无需调用。
- 使能File Cache，会有额外的内存消耗，内存大小由u32TotalCache决定。
- 该接口必须和[RKADK_RECORD_FileCacheDeInit](#)配套使用

【举例】

[rkadk_record_test](#)

2.2.17 RKADK_RECORD_FileCacheDeInit

【描述】

File Cache反初始化。

【语法】

RKADK_S32 RKADK_RECORD_FileCacheDeInit();

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- 该接口必须和[RKADK_RECORD_FileCacheInit](#)配套使用。

- 如果使能File Cache，则需在RKADK_RECORD_Destroy之后调用该接口，否则无需调用。

[rkadk_record_test](#)

2.2.18 RKADK_RECORD_FileCacheSetMode

【描述】

设置File Cache录像类型。

【语法】

void RKADK_RECORD_FileCacheSetMode([RKADK_REC_TYPE_E](#) enRecType);

【参数】

参数名称	描述	输入/输出
enRecType	录像类型	输入

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- 使能File Cache 后，切换录像类型时，必须调用该API像File Cache 设置新的录像类型。

[rkadk_aov_record_test](#)

2.2.19 RKADK_MEDIA_EnablePostIsp

【描述】

使能Post AI ISP，除接口外还可以通过配置sensor ini中的post_aiisp使能Post AI ISP。

【语法】

RKADK_S32 RKADK_MEDIA_EnablePostIsp(RKADK_U32 u32CamId, [RKADK_STREAM_TYPE_E](#) enStrmType, [RKADK_POST_ISP_ATTR_S](#) *pstPostIspAttr);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera Id	输入
enStreamType	码流类型	输入
pstPostIspAttr	Post AI ISP属性	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_media_comm.h

库文件：librkadk.so

2.2.20 RKADK_MEDIA_SetPostIspAttr

【描述】

动态设置Post AI ISP属性。

【语法】

RKADK_S32 RKADK_MEDIA_SetPostIspAttr(RKADK_U32 u32CamId, [RKADK_STREAM_TYPE_E](#) enStrmType, bool bEnable, [RKADK_POST_ISP_ATTR_S](#) *pstPostIspAttr);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera Id	输入
enStreamType	码流类型	输入
bEnable	Post AI ISP功能使能开关，动态属性	输入
pstPostIspAttr	Post AI ISP属性	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_media_comm.h

库文件：librkadk.so

【注意】

- 必须先使能Post AI ISP之后，才能调用RKADK_MEDIA_SetPostIspAttr，否则会返回失败。

2.2.21 RKADK_RECORD_SetPipAttr

【描述】

动态设置画中画属性。

【语法】

RKADK_S32 RKADK_RECORD_SetPipAttr([RKADK_MW_PTR](#) pRecorder, [RKADK_PIP_ATTR_S](#) *pstPipAttr);

【参数】

参数名称	描述	输入/输出
pRecorder	录像任务指针	输入
pstPipAttr	画中画属性	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_record.h

库文件：librkadk.so

【注意】

- 如果需要使能画中画，必须在RKADK_RECORD_Create时，就通过stPipAttr先使能，才会创建画中画通路。

2.3 数据类型

录像模块主要提供以下数据类型：

[RKADK_MW_PTR](#)：录像任务指针

[RECORD_FILE_NUM_MAX](#)：单个录像任务同时录制最大文件个数

[RKADK_MUXER_EVENT_E](#)：录像事件枚举类型

[RKADK_MUXER_FILE_EVENT_INFO_S](#)：文件相关事件信息结构体

[RKADK_MUXER_EVENT_INFO_S](#)：录像事件信息结构体

[RKADK_REC_EVENT_CALLBACK_FN](#)：事件回调函数指针

[RKADK_REC_TYPE_E](#)：录像类型枚举

[RKADK_REC_REQUEST_FILE_NAMES_FN](#)：请求录像文件名函数指针

[RKADK_RECORD_ATTR_S](#)：录像任务属性结构体

[RKADK_MUXER_MANUAL_SPLIT_TYPE_E](#): 手动切分枚举类型

[RKADK_REC_MANUAL_SPLIT_ATTR_S](#): 手动切分属性结构体

[FILE_CACHE_ARG](#): File Cache属性结构体

[RKADK_POST_ISP_ATTR_S](#): Post AI ISP 属性结构体

[RKADK_MOUMNT_SDCARD_FN](#): Sdcard挂载函数指针

[RKADK_AOV_ATTR_S](#): AOV属性结构体

[RKADK_PIP_ATTR_S](#): 画中画属性结构体

2.3.1 公共数据类型

【说明】

基本数据类型定义。

【定义】

```
typedef unsigned char RKADK_U8;
typedef unsigned short RKADK_U16;
typedef unsigned int RKADK_U32;

typedef signed char RKADK_S8;
typedef short RKADK_S16;
typedef int RKADK_S32;

typedef unsigned long RKADK_UL;
typedef signed long RKADK_SL;

typedef float RKADK_FLOAT;
typedef double RKADK_DOUBLE;

#ifndef _M_IX86
typedef unsigned long long RKADK_U64;
typedef long long RKADK_S64;
#else
typedef unsigned __int64 RKADK_U64;
typedef __int64 RKADK_S64;
#endif

typedef char RKADK_CHAR;
#define RKADK_VOID void

typedef unsigned int RKADK_HANDLE;

typedef RKADK_VOID *RKADK_MW_PTR;

typedef char (*ARRAY_FILE_NAME)[RKADK_MAX_FILE_PATH_LEN];

typedef enum {
    RKADK_FALSE = 0,
    RKADK_TRUE = 1,
} RKADK_BOOL;
```

```
#ifndef NULL
#define NULL 0L
#endif

#define RKADK_NULL 0L
#define RKADK_SUCCESS 0
#define RKADK_FAILURE (-1)
```

2.3.2 RKADK_MW_PTR

【说明】

定义任务指针

【定义】

```
typedef RKADK_VOID *RKADK_MW_PTR;
```

2.3.3 RKADK_MAX_SENSOR_CNT

【说明】

定义支持的最大Sensor个数，可根据实际情况调整

【定义】

```
#define RKADK_MAX_SENSOR_CNT 3
```

2.3.4 RECORD_FILE_NUM_MAX

【说明】

定义单个录像任务同时录制最大文件个数

【定义】

```
#define RECORD_FILE_NUM_MAX 2
```

2.3.5 RKADK_MUXER_EVENT_E

【说明】

定义录像事件枚举类型。

【定义】

```
typedef enum rkMUXER_EVENT_E {
    RKADK_MUXER_EVENT_STREAM_START = 0,
    RKADK_MUXER_EVENT_STREAM_STOP,
    RKADK_MUXER_EVENT_FILE_BEGIN,
    RKADK_MUXER_EVENT_FILE_END,
    RKADK_MUXER_EVENT_MANUAL_SPLIT_END,
    RKADK_MUXER_EVENT_ERR_CREATE_FILE_FAIL,
    RKADK_MUXER_EVENT_ERR_WRITE_FILE_FAIL,
    RKADK_MUXER_EVENT_FILE_WRITING_SLOW,
    RKADK_MUXER_EVENT_ERR_CARD_NONEXIST,
    RKADK_MUXER_EVENT_BUTT
} RKADK_MUXER_EVENT_E;
```

【成员】

成员名称	描述
RKADK_MUXER_EVENT_STREAM_START	开始录像
RKADK_MUXER_EVENT_STREAM_STOP	停止录像
RKADK_MUXER_EVENT_FILE_BEGIN	开始录制一个新文件
RKADK_MUXER_EVENT_FILE_END	文件录制结束
RKADK_MUXER_EVENT_MANUAL_SPLIT_END	手动切分文件录制结束
RKADK_MUXER_EVENT_ERR_CREATE_FILE_FAIL	Reserved
RKADK_MUXER_EVENT_ERR_WRITE_FILE_FAIL	写文件失败
RKADK_MUXER_EVENT_FILE_WRITING_SLOW	写文件慢
RKADK_MUXER_EVENT_ERR_CARD_NONEXIST	sdcard不存在

【相关数据类型及接口】

[RKADK_MUXER_EVENT_INFO_S](#)

2.3.6 RKADK_MUXER_FILE_EVENT_INFO_S

【说明】

定义文件相关事件信息结构体。

【定义】

```
typedef struct {
    RK_CHAR asFileName[RKADK_MUXER_FILE_NAME_LEN];
    RK_U32 u32Duration; // ms
} RKADK_MUXER_FILE_EVENT_INFO_S;
```

【成员】

成员名称	描述
asFileName	文件名
u32Duration	实际录制的文件时长

【相关数据类型及接口】

[RKADK_MUXER_EVENT_INFO_S](#)

2.3.7 RKADK_MUXER_EVENT_INFO_S

【说明】

定义录像事件信息结构体。

【定义】

```
typedef struct {
    RKADK_MUXER_EVENT_E enEvent;
    union {
        RKADK_MUXER_FILE_EVENT_INFO_S stFileInfo;
        RKADK_MUXER_ERROR_EVENT_INFO_S stErrorInfo;
    } unEventInfo;
} RKADK_MUXER_EVENT_INFO_S;
```

【成员】

成员名称	描述
enEvent	录像事件类型
stFileInfo	文件事件信息
stErrorInfo	错误事件信息 (Reserved)

【相关数据类型及接口】

[RKADK_MUXER_EVENT_E](#)

[RKADK_MUXER_FILE_EVENT_INFO_S](#)

2.3.8 RKADK_REC_EVENT_CALLBACK_FN

【说明】

定义录像事件回调函数指针。

【定义】

```
typedef RKADK_MUXER_EVENT_CALLBACK_FN RKADK_REC_EVENT_CALLBACK_FN;

typedef RKADK_VOID (*RKADK_MUXER_EVENT_CALLBACK_FN)(RKADK_MW_PTR pRecorder, const
RKADK_MUXER_EVENT_INFO_S *pstEventInfo);
```

【相关数据类型及接口】

[RKADK_MW_PTR](#)

[RKADK_MUXER_EVENT_INFO_S](#)

2.3.9 RKADK_REC_TYPE_E

【说明】

定义录像类型枚举。

【定义】

```
typedef enum {
    RKADK_REC_TYPE_NORMAL = 0, /* normal record */
    RKADK_REC_TYPE_LAPSE,    /* time lapse record */
    RKADK_REC_TYPE_BUTT
} RKADK_REC_TYPE_E;
```

【成员】

成员名称	描述
RKADK_REC_TYPE_NORMAL	普通录像
RKADK_REC_TYPE_LAPSE	缩时录像

【相关数据类型及接口】

无

2.3.10 RKADK_REC_REQUEST_FILE_NAMES_FN

【说明】

定义请求录像文件名回调函数指针。

【定义】

```
typedef RKADK_S32 (*RKADK_REC_REQUEST_FILE_NAMES_FN)(RKADK_MW_PTR pRecorder,
    RKADK_U32 u32FileCnt, RKADK_CHAR(*paszFilename)[RKADK_MAX_FILE_PATH_LEN]);
```

【成员】

成员名称	描述
pRecorder	录像任务指针
u32FileCnt	请求文件名个数
paszFilename	存储文件名buffer

【相关数据类型及接口】

[RKADK_MW_PTR](#)

[RKADK_RECORD_ATTR_S](#)

2.3.11 RKADK_MOUMNT_SDCARD_FN

【说明】

定义Sdcard挂载函数指针。

【定义】

```
typedef int (*RKADK_MOUMNT_SDCARD_FN)(void);
```

【相关数据类型及接口】

[RKADK_RECORD_ATTR_S](#)

2.3.12 RKADK_RECORD_ATTR_S

【说明】

定义录像任务属性结构体。

【定义】

```
typedef struct {  
    RKADK_S32 s32CamID;           /* camera id */  
    RKADK_U32 u32FragKeyFrame;  
    RKADK_REC_REQUEST_FILE_NAMES_FN pfnRequestFileNames; /* rec callbak */  
    RKADK_REC_EVENT_CALLBACK_FN pfnEventCallback;      /* event callbak */  
    RKADK_AOV_ATTR_S stAovAttr;  
    RKADK_POST_ISP_ATTR_S *pstPostIspAttr;  
    RKADK_MOUMNT_SDCARD_FN pfnMountSdcard;  
    RKADK_PIP_ATTR_S stPipAttr[RECORD_FILE_NUM_MAX];  
} RKADK_RECORD_ATTR_S;
```

【成员】

成员名称	描述
s32CamID	Camera id
u32FragKeyFrame	录像文件是否I帧切片
pfnRequestFileNames	请求文件名函数指针
pfnEventCallback	录像事件回调函数指针
stAovAttr	AOV(Always On Video)属性
pstPostIspAttr	Post AI ISP属性
pfnMountSdcard	Sdcard挂载函数指针
stPipAttr	画中画属性

【相关数据类型及接口】

[RKADK_REC_EVENT_CALLBACK_FN](#)

[RKADK_REC_REQUEST_FILE_NAMES_FN](#)

[RKADK_MOUMNT_SDCARD_FN](#)

[RKADK_AOV_ATTR_S](#)

[RKADK_POST_ISP_ATTR_S](#)

[RKADK_PIP_ATTR_S](#)

[RKADK_RECORD_Create](#)

2.3.13 RKADK_MUXER_MANUAL_SPLIT_TYPE_E

【说明】

定义手动切分类型。

【定义】

```
typedef enum {
    MUXER_PRE_MANUAL_SPLIT,    /* pre manual split type */
    MUXER_NORMAL_MANUAL_SPLIT, /* normal manual split type */
} RKADK_MUXER_MANUAL_SPLIT_ATTR_S;
```

【成员】

成员名称	描述
MUXER_PRE_MANUAL_SPLIT	手动切分录像文件并预录
MUXER_NORMAL_MANUAL_SPLIT	手动切分录像文件

【相关数据类型及接口】

[RKADK_REC_MANUAL_SPLIT_ATTR_S](#)

2.3.14 RKADK_REC_MANUAL_SPLIT_ATTR_S

【说明】

定义手动切分属性结构体。

【定义】

```
typedef MUXER_MANUAL_SPLIT_ATTR_S RKADK_REC_MANUAL_SPLIT_ATTR_S;

typedef struct {
    RKADK_MUXER_MANUAL_SPLIT_TYPE_E enManualType;          /* maual split type */
    RKADK_U32 u32DurationSec; /* file duration of manual split file */
} MUXER_MANUAL_SPLIT_ATTR_S;
```

【成员】

成员名称	描述
enManualType	手动切分类型
u32DurationSec	手动切分录像文件时长

【相关数据类型及接口】

[RKADK_MUXER_MANUAL_SPLIT_TYPE_E](#)

[RKADK_RECORD_ManualSplit](#)

2.3.15 FILE_CACHE_ARG

【说明】

定义File Cache属性结构体。

【定义】

```
typedef struct _FILE_CACHE_ARG {
    const char *sdcard_path;
    int write_cache; /* write cache size(byte), default 1M */
    int total_cache; /* total cache size(byte), default 10M */
    FILE_WRITE_THREAD_ARG write_thread_arg;
    FILE_SDCARD_ARG sdcard_arg;
} FILE_CACHE_ARG;
```

【成员】

成员名称	描述
sdcard_path	sdcard挂载路径
write_cache	每次写文件的Cache大小
total_cache	总Cache大小
write_thread_arg	写文件线程属性
sdcard_arg	sdcard属性

【相关数据类型及接口】

[RKADK_RECORD_FileCacheInit](#)

[FILE_WRITE_THREAD_ARG](#)

[FILE_SDCARD_ARG](#)

2.3.16 FILE_WRITE_THREAD_ARG

【说明】

定义写文件线程属性结构体。

【定义】

```
typedef enum _FILE_SCHED_POLICY{
    FILE_SCHED_OTHER = 0,
    FILE_SCHED_BATCH,
    FILE_SCHED_IDLE,
    FILE_SCHED_FIFO,    /* sched_priority[1, 99] */
    FILE_SCHED_RR,     /* sched_priority[1, 99] */
} FILE_SCHED_POLICY;

typedef struct _FILE_WRITE_THREAD_ARG {
    FILE_SCHED_POLICY sched_policy;
    int priority; /* SCHED_OTHER/SCHED_IDLE/SCHED_BATCH inoperative */
} FILE_WRITE_THREAD_ARG;
```

【成员】

成员名称	描述
sched_policy	线程调度策略
priority	线程优先级，调度策略为SCHED_OTHER/SCHED_IDLE/SCHED_BATCH时，该值无效

【相关数据类型及接口】

[FILE_CACHE_ARG](#)

2.3.17 FILE_SDCARD_ARG

【说明】

定义sdcard属性结构体。

【定义】

```
typedef struct _FILE_SDCARD_ARG {
    FILE_CACHE_SDCARD_LOCK lock;
    FILE_CACHE_SDCARD_UNLOCK unlock;
    FILE_CACHE_MOUMNT_SDCARD mount_sdcard;
    FILE_CACHE_UMOUNMT_SDCARD umount_sdcard;
} FILE_SDCARD_ARG;
```

【成员】

成员名称	描述
lock	sdcard操作加锁函数
unlock	sdcard操作解锁函数
mount_sdcard	sdcard挂载函数
umount_sdcard	sdcard卸载函数

【相关数据类型及接口】

[FILE_CACHE_ARG](#)

2.3.18 RKADK_POST_ISP_ATTR_S

【说明】

定义Post AI ISP属性结构体。

【定义】

```
typedef struct {
    AIISP_CALLBACK_FUNC_S stAilspCallback; /* post isp callback function */
    const RK_CHAR *pModelFilePath; /* post isp model file path */
    RK_U32 u32FrameBufCnt; /* RW; frame buffer cnt */
} RKADK_POST_ISP_ATTR_S;
```

【成员】

成员名称	描述
stAilspCallback	AI NR参数更新回调函数结构体
pModelFilePath	AI ISP模型路径，AI ISP根据输入图像尺寸自动加载适合的模型
u32FrameBufCnt	最大目标图像缓冲个数，默认为1个

【相关数据类型及接口】

[RKADK_MEDIA_EnablePostIsp](#)

[RKADK_MEDIA_SetPostIspAttr](#)

2.3.19 RKADK_AOV_ATTR_S

【说明】

定义AOV属性结构体。

【定义】

```
typedef struct {
    RKADK_ISP_WAKE_UP_PAUSE_FN pfnSingleFrame;
    RKADK_ISP_WAKE_UP_RESUME_FN pfnMultiFrame;
} RKADK_AOV_ATTR_S;
```

【成员】

成员名称	描述
pfnSingleFrame	进入单帧模式函数指针
pfnMultiFrame	进入多帧模式函数指针

【相关数据类型及接口】

[RKADK_RECORD_Create](#)

2.3.20 RKADK_PIP_ATTR_S

【说明】

定义画中画属性结构体。

【定义】

```
typedef struct {
    RKADK_BOOL bEnablePip;           /* enable picture-in-picture */
    RKADK_U32 u32AvsGrpId;           /* avs group id [0, AVS_MAX_GRP_NUM]] */
    RKADK_U32 u32AvsBufCnt;          /* default 2, min value 2 */
    RKADK_U32 u32SubCamId;           /* subwindow camera id */
    RKADK_STREAM_TYPE_E enSubStreamType; /* subwindow stream type */
    RKADK_RECT_S stSubRect;          /* The subwindow is based on the display area of the main window */
} RKADK_PIP_ATTR_S;
```

【成员】

成员名称	描述
bEnablePip	使能画中画
u32AvsGrpId	使能画中画所使用的AVS Group Id
u32AvsBufCnt	AVS最大目标图像缓冲个数
u32SubCamId	子画面Camera Id
enSubStreamType	子画面码流类型
stSubRect	子画面在主画面中的显示区域

【相关数据类型及接口】

[RKADK_RECORD_Create](#)

[RKADK_RECORD_SetPipAttr](#)

3. AOV(Always On Video)

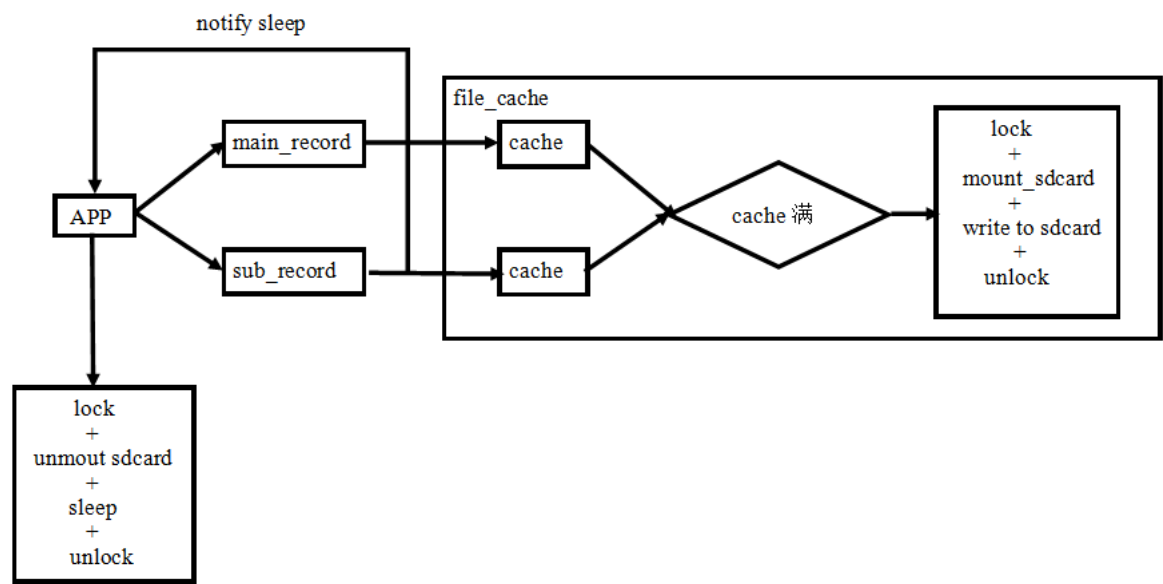
3.1 概述

AOV(Always On Video)基于休眠唤醒实现的低帧率持续录像，有事件时可以切回正常帧率模式，如AI检测、PIR触发、WIFI唤醒等。

使用AOV建议开启file cache功能，file cache使用请参考本文档录像章节的[RKADK_RECORD_FileCacheInit](#)和[RKADK_RECORD_FileCacheDeInit](#) API。

AOV缩时录影模式下为了降低功耗，休眠后sdcard要掉电，应用休眠前要去卸载sdcard，当使能file cache时，file cache写文件时会去尝试挂载sdcard，并进行写文件保护，确保写文件过程中sdcard不会被卸载。

AOV流程如下图所示：



3.2 API 参考

3.2.1 RKADK_AOV_Init

【描述】

AOV初始化。

【语法】

int RKADK_AOV_Init([RKADK_AOV_ARG_S](#) *pstAovAttr);

【参数】

参数名称	描述	输入/输出
pstAovAttr	AOV属性指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_aov.h

库文件：librkadk.so

【注意】

- 必须和RKADK_AOV_DeInit配套使用。
- 不支持重复初始化。

【举例】

[rkadk_aov_record_test](#)

【相关主题】

[RKADK_AOV_DeInit](#)

3.2.2 RKADK_AOV_DeInit

【描述】

AOV反初始化。

【语法】

```
int RKADK_AOV_DeInit();
```

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_aov.h

库文件：librkadk.so

【注意】

- 必须和RKADK_AOV_Init配套使用。
- 不支持重复反初始化。

【举例】

[rkadk_aov_record_test](#)

【相关主题】

3.2.3 RKADK_AOV_SetSuspendTime

【描述】

设置AOV定时休眠时间，单位ms。

【语法】

```
int RKADK_AOV_SetSuspendTime(int u32WakeupSuspendTime);
```

【参数】

参数名称	描述	输入/输出
u32WakeupSuspendTime	休眠时间	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_aov.h

库文件：librkadk.so

【举例】

[rkadk_aov_record_test](#)

3.2.4 RKADK_AOV_EnterSleep

【描述】

进入AOV。

【语法】

```
int RKADK_AOV_EnterSleep();
```

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_aov.h

库文件：librkadk.so

【举例】

[rkadk aov record test](#)

3.2.5 RKADK_AOV_WakeupLock

【描述】

AOV相关操作加锁。

【语法】

```
void RKADK_AOV_WakeupLock();
```

【需求】

头文件：rkadk_aov.h

库文件：librkadk.so

【注意】

- 必须和RKADK_AOV_WakeupUnlock配套使用。

【举例】

[rkadk aov record test](#)

【相关主题】

[RKADK AOV WakeupUnlock](#)

3.2.6 RKADK_AOV_WakeupUnlock

【描述】

AOV相关操作解锁。

【语法】

```
void RKADK_AOV_WakeupUnlock();
```

【需求】

头文件：rkadk_aov.h

库文件：librkadk.so

【注意】

- 必须和RKADK_AOV_WakeupLock配套使用。

【举例】

[rkadk aov record test](#)

【相关主题】

[RKADK AOV WakeupLock](#)

3.3 数据类型

AOV模块主要提供以下数据类型：

[RKADK_AOV_ARG_S](#)：AOV属性结构体

[RKADK_AOV_NOTIFY_CALLBACK](#)：AOV通知回调函数指针

3.3.1 RKADK_AOV_ARG_S

【说明】

定义AOV属性结构体。

【定义】

```
typedef struct {
    RKADK_AOV_NOTIFY_CALLBACK pfnNotifyCallback;
} RKADK_AOV_ARG_S;
```

【成员】

成员名称	描述
pfnNotifyCallback	AOV通知回调函数指针

【相关数据类型及接口】

[RKADK_AOV_Init](#)

[RKADK_AOV_NOTIFY_CALLBACK](#)

3.3.2 RKADK_AOV_NOTIFY_CALLBACK

【说明】

定义AOV通知回调函数指针。

【定义】

```
typedef void (*RKADK_AOV_NOTIFY_CALLBACK)(RKADK_AOV_EVENT_E enEvent, void *msg);
```

【成员】

成员名称	描述
enEvent	AOV通知事件枚举
msg	reserve

【相关数据类型及接口】

[RKADK_AOV_ARG_S](#)

3.3.3 RKADK_AOV_EVENT_E

【说明】

定义AOV通知事件枚举。

【定义】

```
typedef enum {  
    RKADK_AOV_ENTER_SLEEP = 0,  
    RKADK_AOV_EVENT_BUTT  
} RKADK_AOV_EVENT_E;
```

【成员】

成员名称	描述
RKADK_AOV_ENTER_SLEEP	进入AOV休眠

【相关数据类型及接口】

4. 拍照

4.1 概述

提供基本的抓拍功能，提供JPEG封装拍照，包含如下功能：

- 单拍
- 多拍
- JPEG Slice：将VI数据分段编码生成JPEG，可以节省内存用量。当JPEG放大时，使用插值算法，将VI数据分段放大，再分段编码成JPEG。

JPEG Slice 注意事项：

- 分段高要求16像素对齐。
- 考虑最终图像的拼接效果，建议分段后的图片比例与原始图片比例尽量保持一致。
- 分段数越多，所需内存越少，但出图耗时增加。反之当分段数减少时，所需内存增加，但出图速度加快；实际分段数需结合实际情况考量。

4.2 API 参考

4.2.1 RKADK_PHOTO_Init

【描述】

拍照任务初始化。

【语法】

RKADK_S32 RKADK_PHOTO_Init([RKADK_PHOTO_ATTR_S](#) *pstPhotoAttr, [RKADK_MW_PTR](#) *ppHandle);

【参数】

参数名称	描述	输入/输出
pstPhotoAttr	拍照任务属性指针	输入
ppHandle	创建的拍照任务指针	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_photo.h

库文件：librkadk.so

【注意】

- 不支持重复初始化。

【举例】

[rkadk_photo_test](#)

【相关主题】

[RKADK_PHOTO_DeInit](#)

4.2.2 RKADK_PHOTO_DeInit

【描述】

拍照任务反初始化。

【语法】

RKADK_S32 RKADK_PHOTO_DeInit([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	拍照任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_photo.h

库文件：librkadk.so

【注意】

- 不支持重复反初始化。

【举例】

[rkadk_photo_test](#)

【相关主题】

[RKADK_PHOTO_DeInit](#)

4.2.3 RKADK_PHOTO_TakePhoto

【描述】

拍照。

【语法】

RKADK_S32 RKADK_PHOTO_TakePhoto([RKADK_MW_PTR](#) pHandle, [RKADK_TAKE_PHOTO_ATTR_S](#) *pstAttr);

【参数】

参数名称	描述	输入/输出
pHandle	拍照任务指针	输入
pstAttr	拍照属性	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_photo.h

库文件：librkadk.so

【注意】

- 录像任务初始化后，才能使用RKADK_PHOTO_TakePhoto接口。

【举例】

[rkadk_photo_test](#)

【相关主题】

无

4.2.4 RKADK_PHOTO_Reset

【描述】

重新配置拍照任务。

【语法】

RKADK_S32 RKADK_PHOTO_Reset([RKADK_MW_PTR](#) *ppHandle);

【参数】

参数名称	描述	输入/输出
ppHandle	拍照任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_photo.h

库文件：librkadk.so

【注意】

- 录像任务初始化后，才能使用RKADK_PHOTO_Reset接口。
- 切换分辨率时，结合RKADK_PARAM_SetCamParam API重置ini文件中Photo模块的分辨率。
- RV1126/RV1109 不支持动态切换分辨率，,需要先RKADK_PHOTO_DeInit，然后配置新的分辨率，再重新RKADK_PHOTO_Init。
- 如果Photo和Record复用VI通道，需要同时切换Record和Photo的分辨率，否则会导致VI和VENC分辨率不匹配。

【举例】

[rkadk_photo_test](#)

【相关主题】

4.2.5 RKADK_PHOTO_GetThmInJpg

【描述】

从JPG文件中获取缩略图数据。

【语法】

```
RKADK_S32 RKADK_PHOTO_GetThmInJpg(RKADK_U32 u32CamId, RKADK_CHAR *pszFileName,
RKADK\_JPG\_THUMB\_TYPE\_E eThmType,
RKADK_U8 *pu8Buf, RKADK_U32 *pu32Size);
```

【参数】

参数名称	描述	输入/输出
u32CamId	Camera id	输入
pszFileName	JPG文件路径	输入
eThmType	缩略图类型	输入
pu8Buf	输入：缩略图数据存储指针，输出：实际缩略图数据	输入/输出
pu32Size	输入： pu8Buf长度，输出：实际缩略图数据长度	输入/输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_photo.h

库文件：librkadk.so

【举例】

[rkadk_thumb_test](#)

【相关主题】

无

4.2.6 RKADK_PHOTO_GetThmInJpgEx

【描述】

JPG获取缩略图扩展接口，从JPG文件中获取缩略图数据，支持指定输出缩略图的类型和分辨率。必须和[RKADK_PHOTO_ThumbBufFree](#)配套使用。

【语法】

```
RKADK_S32 RKADK_PHOTO_GetThmInJpgEx(RKADK_U32 u32CamId, RKADK_CHAR *pszFileName,  
    RKADK\_JPG\_THUMB\_TYPE\_E eThmType,  
    RKADK\_THUMB\_ATTR\_S *pstThumbAttr);
```

【参数】

参数名称	描述	输入/输出
u32CamId	Camera id	输入
pszFileName	JPG文件路径	输入
eThmType	缩略图类型	输入
pstThumbAttr	缩略图属性结构体指针	输入/输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_photo.h

库文件：librkadk.so

【举例】

[rkadk_thumb_test](#)

【相关主题】

[RKADK_PHOTO_ThumbBufFree](#)

4.2.7 RKADK_PHOTO_ThumbBufFree

【描述】

释放[RKADK_PHOTO_GetThmInJpgEx](#)申请的内存。必须和[RKADK_PHOTO_GetThmInJpgEx](#)配套使用。

【语法】

RKADK_S32 RKADK_PHOTO_ThumbBufFree([RKADK_THUMB_ATTR_S](#) *pstThumbAttr);

【参数】

参数名称	描述	输入/输出
pstThumbAttr	缩略图属性结构体指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_photo.h

库文件：librkadk.so

【举例】

[rkadk_thumb_test](#)

【相关主题】

[RKADK_PHOTO_GetThmInJpgEx](#)

4.2.8 RKADK_MEDIA_SetVencRotation

【描述】

设置VENC通道旋转。

【语法】

RKADK_S32 RKADK_MEDIA_SetVencRotation(RKADK_U32 u32CamId, [ROTATION_E](#) enRotation, [RKADK_STREAM_TYPE_E](#) enStreamType);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera ID	输入
enRotation	旋转角度	输入
enStreamType	码流类型	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_media_comm.h

库文件：librkadk.so

4.2.9 RKADK_MEDIA_ToggleVencMirror

【描述】

设置VENC通道Mirror。

【语法】

RKADK_S32 RKADK_MEDIA_ToggleVencMirror(RKADK_U32 u32CamId, [RKADK_STREAM_TYPE_E](#) enStrmType, bool mirror);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera Id	输入
enStreamType	码流类型	输入
mirror	是否使能mirror	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_media_comm.h

库文件：librkadk.so

4.2.10 RKADK_MEDIA_ToggleVencFlip

【描述】

设置VENC通道Flip。

【语法】

RKADK_S32 RKADK_MEDIA_ToggleVencFlip(RKADK_U32 u32CamId, [RKADK_STREAM_TYPE_E](#) enStrmType, bool flip);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera Id	输入
enStreamType	码流类型	输入
flip	是否使能flip	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_media_comm.h

库文件：librkadk.so

4.3 数据类型

拍照模块主要提供以下数据类型：

[RKADK_PHOTO_TYPE_E](#)：拍照类型枚举

[RKADK_PHOTO_SINGLE_ATTR_S](#)：单拍属性结构体

[RKADK_PHOTO_MULTIPLE_ATTR_S](#)：多拍属性结构体

[RKADK_PHOTO_RECV_DATA_S](#)：拍照数据结构体

[RKADK_PHOTO_DATA_RECV_FN_PTR](#)：拍照数据接收函数指针

[RKADK_TAKE_PHOTO_ATTR_S](#)：拍照属性结构体

[RKADK_PHOTO_ATTR_S](#)：拍照任务属性结构体

[RKADK_JPG_THUMB_TYPE_E](#)：JPG缩略图类型枚举

[RKADK_THUMB_TYPE_E](#)：输出缩略图类型枚举

[RKADK_THUMB_ATTR_S](#)：缩略图属性结构体

[ROTATION_E](#)：旋转类型枚举

[RKADK_PHOTO_FMT_CHANGE_S](#)：格式转换参数结构体，用于FBC0/NV16格式转换

4.3.1 RKADK_PHOTO_TYPE_E

【说明】

定义拍照类型枚举。

【定义】

```
typedef enum {
    RKADK_PHOTO_TYPE_SINGLE = 0,
    RKADK_PHOTO_TYPE_MULTIPLE,
    RKADK_PHOTO_TYPE_LAPSE, // TODO
    RKADK_PHOTO_TYPE_BUTT
} RKADK_PHOTO_TYPE_E;
```

【成员】

成员名称	描述
RKADK_PHOTO_TYPE_SINGLE	单拍模式
RKADK_PHOTO_TYPE_MULTIPLE	多拍模式
RKADK_PHOTO_TYPE_LAPSE	缩时拍照(Reserved)

【相关数据类型及接口】

[RKADK_TAKE_PHOTO_ATTR_S](#)

4.3.2 RKADK_PHOTO_SINGLE_ATTR_S

【说明】

定义单拍属性结构体。

【定义】

```
typedef struct {
    // TODO
    RKADK_S32 s32Time_sec;
} RKADK_PHOTO_SINGLE_ATTR_S;
```

【成员】

成员名称	描述
s32Time_sec	Reserved

【相关数据类型及接口】

[RKADK_TAKE_PHOTO_ATTR_S](#)

4.3.3 RKADK_PHOTO_MULTIPLE_ATTR_S

【说明】

定义多拍属性结构体。

【定义】

```
typedef struct {
    /* s32Count is -1 that means continuous photo, larger than 0 that means photo
    * number */
    RKADK_S32 s32Count;
} RKADK_PHOTO_MULTIPLE_ATTR_S;
```

【成员】

成员名称	描述
s32Count	连拍数量，-1 代表连续拍照直到调用RKADK_PHOTO_DeInit停止

【相关数据类型及接口】

[RKADK_TAKE_PHOTO_ATTR_SS](#)

4.3.4 RKADK_PHOTO_THUMB_ATTR_S

【说明】

定义缩略图参数属性结构体。

【定义】

```
#define RKADK_MPF_LARGE_THUMB_NUM_MAX 2 /* 支持同时生成的最大MPF缩略图个数 */

typedef struct rkSIZE_S {
    RK_U32 u32Width;      /* 缩略图宽 */
    RK_U32 u32Height;     /* 缩略图高 */
} SIZE_S;

typedef enum {
    RKADK_PHOTO_MPF_SINGLE = 0, /* 单个MPF缩略图 */
    RKADK_PHOTO_MPF_MULTIPLE, /* 多个MPF缩略图 */
    RKADK_PHOTO_MPF_BUTTON,
} RKADK_PHOTO_MPF_MODE_E;

typedef struct {
    RKADK_U8 u8LargeThumbNum; /* 生成的MPF缩略图个数 */
    SIZE_S astLargeThumbSize[RKADK_MPF_LARGE_THUMB_NUM_MAX]; /* 各缩略图分辨率 */
} RKADK_PHOTO_MPF_CFG_S;
```

```
typedef struct {
    RKADK_PHOTO_MPF_MODE_E eMode; /* MPF缩略图模式 */
    RKADK_PHOTO_MPF_CFG_S sCfg; /* MPF缩略图配置 */
} RKADK_PHOTO_MPF_ATTR_S;

typedef struct {
    RKADK_BOOL bSupportDCF; /* Fixed resolution: 160 * 120 jpg */
    RKADK_PHOTO_MPF_ATTR_S stMPFAttr;
} RKADK_PHOTO_THUMB_ATTR_S;
```

【成员】

成员名称	描述
bSupportDCF	是否生成DCF缩略图(固定分辨率160*120)
stMPFAttr	MPF缩略图属性参数

【相关数据类型及接口】

[RKADK_PHOTO_ATTR_S](#)

4.3.5 RKADK_PHOTO_RECV_DATA_S

【说明】

定义拍照数据结构体。

【定义】

```
typedef struct {
    RKADK_U8 *pu8DataBuf;
    RKADK_U32 u32DataLen;
    RKADK_U32 u32CamId;
    bool bStreamEnd;
    void *userdata;
} RKADK_PHOTO_RECV_DATA_S;
```

【成员】

成员名称	描述
pu8DataBuf	数据指针
u32DataLen	数据长度
u32CamId	Camera Id
bStreamEnd	数据流是否结束，主要用于JPEG Slice
userdata	用户数据指针

【相关数据类型及接口】

[RKADK_PHOTO_DATA_RECV_FN_PTR](#)

4.3.6 RKADK_PHOTO_DATA_RECV_FN_PTR

【说明】

定义拍照数据接收函数指针。

【定义】

```
typedef void (*RKADK_PHOTO_DATA_RECV_FN_PTR)(RKADK_PHOTO_RECV_DATA_S *pstData);
```

【成员】

成员名称	描述
pstData	数据指针

【相关数据类型及接口】

[RKADK_PHOTO_RECV_DATA_S](#)

[RKADK_PHOTO_ATTR_S](#)

4.3.7 RKADK_TAKE_PHOTO_ATTR_S

【说明】

定义拍照属性结构体。

【定义】

```
typedef struct {
    RKADK_PHOTO_TYPE_E enPhotoType;
    union tagPhotoTypeAttr {
        RKADK_PHOTO_SINGLE_ATTR_S stSingleAttr;
        RKADK_PHOTO_LAPSE_ATTR_S stLapseAttr; // TODO
        RKADK_PHOTO_MULTIPLE_ATTR_S stMultipleAttr;
    } unPhotoTypeAttr;
} RKADK_TAKE_PHOTO_ATTR_S;
```

【成员】

成员名称	描述
enPhotoType	拍照类型
stSingleAttr	单拍参数属性
stMultipleAttr	多拍参数属性
stLapseAttr	缩时拍照参数属性(Reserved)

【相关数据类型及接口】

[RKADK_PHOTO_TYPE_E](#)

[RKADK_PHOTO_SINGLE_ATTR_S](#)

[RKADK_PHOTO_MULTIPLE_ATTR_S](#)

[RKADK_PHOTO_TakePhoto](#)

4.3.8 RKADK_PHOTO_FMT_CHANGE_S

【说明】

定义格式转换参数结构体，用于FBC0/NV16格式转换

【定义】

```
typedef struct {
    RKADK_U32 u32VencChn;
    RKADK_U32 u32VdecChn;

    //format change venc parameter
    RKADK_U32 u32BitRate; //default 4M
    RKADK_U32 u32Profile; //default 77
} RKADK_PHOTO_FMT_CHANGE_S;
```

【成员】

成员名称	描述
u32VencChn	格式转换所使用的VENC通道Id
u32VdecChn	格式转换所使用的VDEC通道Id
u32BitRate	编码码率，默认4M
u32Profile	编码等级，默认77

【相关数据类型及接口】

[RKADK_PHOTO_ATTR_S](#)

[RKADK_PHOTO_Init](#)

4.3.9 RKADK_PHOTO_ATTR_S

【说明】

定义拍照任务属性结构体。

【定义】


```
typedef struct {
    RKADK_U32 u32CamId;
    RKADK_PHOTO_THUMB_ATTR_S stThumbAttr;
    RKADK_PHOTO_DATA_RECV_FN_PTR pfnPhotoDataProc;
    void *userdata;
    RKADK_POST_ISP_ATTR_S *pstPostIsppAttr;
    RKADK_PHOTO_FMT_CHANGE_S stFmtChange;
} RKADK_PHOTO_ATTR_S;
```

【成员】

成员名称	描述
u32CamId	Camera id
stThumbAttr	缩略图参数属性
pfnPhotoDataProc	拍照数据接收回调函数指针
userdata	用户数据指针
pstPostIsppAttr	Post AI ISP属性
stFmtChange	格式转换参数，用于FBC0/NV16格式转换

【相关数据类型及接口】

- [RKADK_PHOTO_THUMB_ATTR_S](#)
- [RKADK_PHOTO_DATA_RECV_FN_PTR](#)
- [RKADK_POST_ISP_ATTR_S](#)
- [RKADK_PHOTO_FMT_CHANGE_S](#)
- [RKADK_PHOTO_Init](#)

4.3.10 RKADK_JPG_THUMB_TYPE_E

【说明】

定义JPG缩略图类型枚举。

【定义】

```
typedef enum {
    RKADK_JPG_THUMB_TYPE_DCF,
    RKADK_JPG_THUMB_TYPE_MFP1,
    RKADK_JPG_THUMB_TYPE_MFP2,
    RKADK_JPG_THUMB_TYPE_BUTT
} RKADK_JPG_THUMB_TYPE_E;
```

【成员】

成员名称	描述
RKADK_JPG_THUMB_TYPE_DCF	DCF缩略图
RKADK_JPG_THUMB_TYPE_MFP1	MPF1缩略图
RKADK_JPG_THUMB_TYPE_MFP2	MPF2缩略图

【相关数据类型及接口】

[RKADK_PHOTO_GetThmInJpg](#)

[RKADK_PHOTO_GetThmInJpgEx](#)

4.3.11 RKADK_THUMB_TYPE_E

【说明】

定义输出缩略图类型枚举。

【定义】

```
typedef enum {  
    RKADK_THUMB_TYPE_NV12 = 0,  
    RKADK_THUMB_TYPE_JPEG,  
    RKADK_THUMB_TYPE_RGB565,  
    RKADK_THUMB_TYPE_RGBA8888,  
    RKADK_THUMB_TYPE_BGRA8888  
} RKADK_THUMB_TYPE_E;
```

【成员】

成员名称	描述
RKADK_THUMB_TYPE_NV12	输出NV12格式缩略图
RKADK_THUMB_TYPE_JPEG	输出JPG格式缩略图
RKADK_THUMB_TYPE_RGB565	输出RGB565格式缩略图
RKADK_THUMB_TYPE_RGBA8888	输出RGBA9888格式缩略图
RKADK_THUMB_TYPE_BGRA8888	输出BGRA8888格式缩略图

【相关数据类型及接口】

[RKADK_THUMB_ATTR_S](#)

[RKADK_GetThmInMp4Ex](#)

[RKADK_PHOTO_GetThmInJpgEx](#)

4.3.12 RKADK_THUMB_ATTR_S

【说明】

定义缩略图属性结构体。

【定义】

```
typedef struct {
    RKADK_THUMB_TYPE_E enType;
    // 4 alignment
    RKADK_U32 u32Width;
    // 2 alignment
    RKADK_U32 u32Height;
    // 4 alignment
    RKADK_U32 u32VirWidth;
    // 2 alignment
    RKADK_U32 u32VirHeight;
    RKADK_U8 *pu8Buf;
    RKADK_U32 u32BufSize;
} RKADK_THUMB_ATTR_S;
```

【成员】

成员名称	描述
enType	希望获取的缩略图类型
u32Width	输入希望获取的缩略图宽，输出实际缩略图宽
u32Height	输入希望获取的缩略图高，输出实际缩略图高
u32VirWidth	输入希望获取的缩略图虚宽，输出实际缩略图虚宽
u32VirHeight	输入希望获取的缩略图虚高，输出实际缩略图虚高
pu8Buf	缩略图数据指针
u32BufSize	缩略图数据长度

【相关数据类型及接口】

- [RKADK_THUMB_TYPE_E](#)
- [RKADK_GetThmInMp4Ex](#)
- [RKADK_PHOTO_GetThmInJpgEx](#)

4.3.13 ROTATION_E

【说明】

定义旋转类型枚举。

【定义】

```
typedef enum rkROTATION_E {  
    ROTATION_0    = 0,  
    ROTATION_90   = 1,  
    ROTATION_180  = 2,  
    ROTATION_270  = 3,  
    ROTATION_BUTT  
} ROTATION_E;
```

【相关数据类型及接口】

[RKADK_MEDIA_SetVencRotation](#)

5. 远程预览

5.1 概述

为预览提供获取Video和Audio信息，启停VENC，启停AENC，注册处理音视频帧数据函数的回调接口。

5.2 API参考

5.2.1 RKADK_STREAM_VideoInit

【描述】

初始化Video模块：VI、VENC。

【语法】

RKADK_S32 RKADK_STREAM_VideoInit([RKADK_STREAM_VIDEO_ATTR_S](#) *pstVideoAttr, [RKADK_MW_PTR](#) *ppHandle);

【参数】

参数名称	描述	输入/输出
pstVideoAttr	Video属性指针	输入
ppHandle	创建的Video任务指针	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_stream.h

库文件：librkadk.so

【注意】

- 不支持重复初始化Video模块。

【举例】

[rkadk_stream_test](#)

【相关主题】

[RKADK_STREAM_VideoDeInit](#)

5.2.2 RKADK_STREAM_VideoDeInit

【描述】

反初始化Video模块：VI、VENC。

【语法】

RKADK_S32 RKADK_STREAM_VideoDeInit([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	Video任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_stream.h

库文件：librkadk.so

【注意】

- 不支持重复反初始化。

【举例】

[rkadk_stream_test](#)

【相关主题】

[RKADK_STREAM_VideoInit](#)

5.2.3 RKADK_STREAM_VencStart

【描述】

启动VENC。

【语法】

RKADK_S32 RKADK_STREAM_VencStart([RKADK_MW_PTR](#) pHandle, RKADK_S32 s32FrameCnt);

【参数】

参数名称	描述	输入/输出
pHandle	Video任务指针	输入
s32FrameCnt	指定需要接收的图像帧数，-1 代表无限接收，直到调用VencStop为止	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_stream.h

库文件：librkadk.so

【注意】

- 初始化Video模块后，才能调用RKADK_STREAM_VencStart接口。
- 调用RKADK_STREAM_VencStart后，触发VENC数据回调函数开始接收数据。

【举例】

[rkadk_stream_test](#)

【相关主题】

[RKADK_STREAM_VencStop](#)

5.2.4 RKADK_STREAM_VencStop

【描述】

停止VENC。

【语法】

RKADK_S32 RKADK_STREAM_VencStop([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	Video任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_stream.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk_stream_test](#)

【相关主题】

[RKADK_STREAM_VencStart](#)

5.2.5 RKADK_STREAM_GetVideoInfo

【描述】

获取Video信息。

【语法】

RKADK_S32 RKADK_STREAM_GetVideoInfo(RKADK_U32 u32CamId, [RKADK_VIDEO_INFO_S](#) *pstVideoInfo);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera id	输入
pstVideoInfo	Video 信息结构体指针	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_stream.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk_stream_test](#)

【相关主题】

无

5.2.6 RKADK_STREAM_AudioInit

【描述】

初始化Audio模块：AI、AENC。

【语法】

RKADK_S32 RKADK_STREAM_AudioInit([RKADK_STREAM_AUDIO_ATTR_S](#) *pstAudioAttr,
[RKADK_MW_PTR](#) *ppHandle);

【参数】

参数名称	描述	输入/输出
pstAudioAttr	Audio属性指针	输入
ppHandle	创建的Audio任务指针	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_stream.h

库文件：librkadk.so

【注意】

- 不支持重复初始化Audio模块。

【举例】

[rkadk_stream_test](#)

【相关主题】

[RKADK_STREAM_AudioDeInit](#)

5.2.7 RKADK_STREAM_AudioDeInit

【描述】

反初始化Audio模块：AI、AENC。

【语法】

RKADK_S32 RKADK_STREAM_AudioDeInit([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	Audio任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_stream.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk_stream_test](#)

【相关主题】

[RKADK_STREAM_AudioInit](#)

5.2.8 RKADK_STREAM_AencStart

【描述】

启动AENC。

【语法】

RKADK_S32 RKADK_STREAM_AencStart([RKADK_MW_PTR](#) *pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	Audio任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_stream.h

库文件：librkadk.so

【注意】

- 初始化Audio模块后，才能调用RKADK_STREAM_AencStart接口。
- 调用RKADK_STREAM_AencStart后，触发AENC数据回调函数开始接收数据。

【举例】

[rkadk_stream_test](#)

【相关主题】

[RKADK_STREAM_AencStop](#)

5.2.9 RKADK_STREAM_AencStop

【描述】

停止AENC。

【语法】

RKADK_S32 RKADK_STREAM_AencStop([RKADK_MW_PTR](#) *pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	Audio任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_stream.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk_stream_test](#)

【相关主题】

[RKADK_STREAM_AencStart](#)

5.2.10 RKADK_STREAM_GetAudioInfo

【描述】

获取Audio信息。

【语法】

RKADK_S32 RKADK_STREAM_GetAudioInfo([RKADK_MW_PTR](#) *pHandle, [RKADK_AUDIO_INFO_S](#) *pstAudioInfo);

【参数】

参数名称	描述	输入/输出
pHandle	Audio任务指针	输入
pstAudioInfo	Audio 信息结构体指针	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_stream.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk_stream_test](#)

【相关主题】

无

5.3 数据类型

播放模块主要提供以下数据类型：

[RKADK_CODEC_TYPE_E](#)：编码格式枚举

[RKADK_VENC_DATA_PROC_FUNC](#)：VENC数据回调函数指针

[RKADK_VIDEO_STREAM_S](#)：Video数据流结构体

[RKADK_VENC_DATA_PACK_S](#)：VENC数据包结构体

[RKADK_VENC_DATA_TYPE_S](#)：VENC数据包类型

[RKADK_VIDEO_INFO_S](#)：Video信息结构体

[RKADK_STREAM_VIDEO_ATTR_S](#)：Video任务属性结构体

[RKADK_AUDIO_DATA_PROC_FUNC](#)：Audio数据回调函数指针

[RKADK_AUDIO_STREAM_S](#)：Audio数据结构体

[RKADK_AUDIO_INFO_S](#)：Audio信息结构体

[RKADK_STREAM_AUDIO_ATTR_S](#)：Audio任务属性结构体

5.3.1 RKADK_CODEC_TYPE_E

【说明】

定义编码格式枚举类型，Audio编码默认不支持AAC。

【定义】

```
typedef enum {  
    //Video  
    RKADK_CODEC_TYPE_H264 = 0,  
    RKADK_CODEC_TYPE_H265,  
    RKADK_CODEC_TYPE_MJPEG,  
    RKADK_CODEC_TYPE_JPEG,  
  
    //Audio  
    RKADK_CODEC_TYPE_G711A,  
    RKADK_CODEC_TYPE_G711U,  
    RKADK_CODEC_TYPE_G726,  
    RKADK_CODEC_TYPE_MP2,  
    RKADK_CODEC_TYPE_MP3,  
    RKADK_CODEC_TYPE_ACC,  
    RKADK_CODEC_TYPE_PCM,  
    RKADK_CODEC_TYPE_BUTT  
} RKADK_CODEC_TYPE_E;
```

5.3.2 RKADK_VENC_DATA_PROC_FUNC

【说明】

定义VENC数据回调函数指针。

【定义】

```
typedef RKADK_S32 (*RKADK_VENC_DATA_PROC_FUNC)(RKADK_VIDEO_STREAM_S *pVStreamData);
```

【相关数据类型及接口】

[RKADK_VIDEO_STREAM_S](#)

5.3.3 RKADK_VIDEO_STREAM_S

【说明】

定义Video 数据流结构体。

【定义】

```
typedef struct {
    RKADK_VENC_DATA_PACK_S astPack; /* stream pack attribute */
    RKADK_U32 u32Seq; /* the list number of stream */
    RKADK_BOOL bEndOfStream; /* frame end flag */
    RKADK_U32 u32CamId;
} RKADK_VIDEO_STREAM_S;
```

【成员】

成员名称	描述
astPack	数据包结构体
u32Seq	数据包序列号
bEndOfStream	Reserved
u32CamId	Camera Id

【相关数据类型及接口】

[RKADK_VENC_DATA_PACK_S](#)

[RKADK_VENC_DATA_PROC_FUNC](#)

5.3.4 RKADK_VENC_DATA_PACK_S

【说明】

定义VENC数据包结构体。

【定义】

```
typedef struct {
    RKADK_U8 *apu8Addr;          /* the virtual address of stream */
    RKADK_U32 au32Len;           /* the length of stream */
    RKADK_U64 u64PTS;           /* time stamp */
    RKADK_VENC_DATA_TYPE_S stDataType; /* the type of stream */
} RKADK_VENC_DATA_PACK_S;
```

【成员】

成员名称	描述
apu8Addr	数据指针
au32Len	数据长度
u64PTS	时间戳
stDataType	数据类型

【相关数据类型及接口】

[RKADK_VENC_DATA_TYPE_S](#)

[RKADK_VIDEO_STREAM_S](#)

5.3.5 RKADK_VENC_DATA_TYPE_S

【说明】

定义VENC数据包类型。

【定义】

```
/* the nalu type of H264 */
typedef enum {
    RKADK_H264E_NALU_BSLICE = 0, /* B SLICE types */
    RKADK_H264E_NALU_PSLICE = 1, /* P SLICE types */
    RKADK_H264E_NALU_ISLICE = 2, /* I SLICE types */
    RKADK_H264E_NALU_IDRSLICE = 5, /* IDR SLICE types */
    RKADK_H264E_NALU_SEI = 6, /* SEI types */
    RKADK_H264E_NALU_SPS = 7, /* SPS types */
    RKADK_H264E_NALU_PPS = 8, /* PPS types */
    RKADK_H264E_NALU_BUTT
} RKADK_H264E_NALU_TYPE_E;

/* the nalu type of H265 */
typedef enum {
    RKADK_H265E_NALU_BSLICE = 0, /* B SLICE types */
    RKADK_H265E_NALU_PSLICE = 1, /* P SLICE types */
    RKADK_H265E_NALU_ISLICE = 2, /* I SLICE types */
    RKADK_H265E_NALU_IDRSLICE = 19, /* IDR SLICE types */
    RKADK_H265E_NALU_VPS = 32, /* VPS types */
    RKADK_H265E_NALU_SPS = 33, /* SPS types */
    RKADK_H265E_NALU_PPS = 34, /* PPS types */
    RKADK_H265E_NALU_SEI = 39, /* SEI types */
    RKADK_H265E_NALU_BUTT
```

```
} RKADK_H265E_NALU_TYPE_E;

typedef struct {
    RKADK_CODEC_TYPE_E enPayloadType; /* H.264/H.265/JPEG/MJPEG */
    union {
        RKADK_H264E_NALU_TYPE_E enH264EType; /* H264E NALU types */
        RKADK_H265E_NALU_TYPE_E enH265EType; /* H265E NALU types */
        RKADK_JPEGE_PACK_TYPE_E enJPEGEType; /* TODO: JPEGE PACK types */
    };
} RKADK_VENC_DATA_TYPE_S;
```

【成员】

成员名称	描述
enPayloadType	编码类型
enH264EType	H264 编码数据包类型
enH265EType	H265 编码数据包类型
enJPEGEType	Reserved

【相关数据类型及接口】

[RKADK_CODEC_TYPE_E](#)

[RKADK_VENC_DATA_PACK_S](#)

5.3.6 RKADK_VIDEO_INFO_S

【说明】

定义Video信息结构体。

【定义】

```
typedef struct {
    RKADK_CODEC_TYPE_E enCodecType;
    RKADK_U32 u32Width;
    RKADK_U32 u32Height;
    RKADK_U32 u32BitRate;
    RKADK_U32 u32FrameRate;
    RKADK_U32 u32Gop;
} RKADK_VIDEO_INFO_S;
```

【成员】

成员名称	描述
enPayloadType	编码类型
u32Width	分辨率宽度
u32Height	分辨率高度
u32BitRate	比特率
u32FrameRate	帧率
u32Gop	I 帧间隔

【相关数据类型及接口】

[RKADK_CODEC_TYPE_E](#)

[RKADK_STREAM_GetVideoInfo](#)

5.3.7 RKADK_STREAM_VIDEO_ATTR_S

【说明】

定义Video任务属性结构体。

【定义】

```
typedef struct {  
    RKADK_U32 u32CamId;  
    RKADK_VENC_DATA_PROC_FUNC pfnDataCB;  
} RKADK_STREAM_VIDEO_ATTR_S;
```

【成员】

成员名称	描述
u32CamId	Camera Id
pfnDataCB	Video数据输出回调函数

【相关数据类型及接口】

[RKADK_VENC_DATA_PROC_FUNC](#)

[RKADK_STREAM_VideoInit](#)

5.3.8 RKADK_AUDIO_DATA_PROC_FUNC

【说明】

定义AENC数据回调函数指针。

【定义】

```
typedef RKADK_S32 (*RKADK_AUDIO_DATA_PROC_FUNC)(RKADK_AUDIO_STREAM_S *pStreamData);
```

【相关数据类型及接口】

[RKADK_AUDIO_STREAM_S](#)

5.3.9 RKADK_AUDIO_STREAM_S

【说明】

定义Audio数据流结构体。

【定义】

```
typedef struct {
    RKADK_U8 *pStream;    /* the virtual address of stream */
    RKADK_U32 u32Len;     /* stream lenth, by bytes */
    RKADK_U64 u64TimeStamp; /* frame time stamp */
    RKADK_U32 u32Seq;     /* frame seq, if stream is not a valid frame,u32Seq is 0 */
    RKADK_CODEC_TYPE_E enType;
} RKADK_AUDIO_STREAM_S;
```

【成员】

成员名称	描述
pStream	数据指针
u32Len	数据长度
u64TimeStamp	时间戳
u32Seq	序列号
enType	Audio数据编码类型

【相关数据类型及接口】

[RKADK_AUDIO_DATA_PROC_FUNC](#)

5.3.10 RKADK_AUDIO_INFO_S

【说明】

定义Audio信息结构体。

【定义】

```
typedef struct {
    RKADK_CODEC_TYPE_E enCodecType;
    RKADK_U32 u32ChnCnt;
    RKADK_U32 u32SampleRate;
    RKADK_U32 u32AvgBytesPerSec;
    RKADK_U32 u32SamplesPerFrame;
    RKADK_U16 u16SampleBitWidth;
} RKADK_AUDIO_INFO_S;
```

【成员】

成员名称	描述
enPayloadType	编码类型
u32ChnCnt	通道数
u32SampleRate	采样率
u32AvgBytesPerSec	字节率
u32SamplesPerFrame	每一帧采样数
u16SampleBitWidth	每个样本比特数

【相关数据类型及接口】

[RKADK_CODEC_TYPE_E](#)

[RKADK_STREAM_GetAudioInfo](#)

5.3.11 RKADK_STREAM_AUDIO_ATTR_S

【说明】

定义Audio任务属性结构体。

【定义】

```
typedef struct {
    RKADK_U32 u32CamId;
    RKADK_CODEC_TYPE_E enCodecType;
    RKADK_AUDIO_DATA_PROC_FUNC pfnPcmDataCB;
    RKADK_AUDIO_DATA_PROC_FUNC pfnAencDataCB;
} RKADK_STREAM_AUDIO_ATTR_S;
```

【成员】

成员名称	描述
u32CamId	Camera Id
enCodecType	编码类型
pfnPcmDataCB	PCM数据输出回调函数
pfnAencDataCB	Audio编码数据输出回调函数

【相关数据类型及接口】

[RKADK_AUDIO_DATA_PROC_FUNC](#)

[RKADK_STREAM_AudioInit](#)

6. 播放器

6.1 概述

提供本地音视频文件、RTSP网络流播放功能，支持基本的播控操作：播放、暂停、Seek、屏幕快照。

Player功能兼容RV1109/RV1126、RV1103/RV1103、RK3308、RK3506平台，其中RV1109/RV1126使用硬件解码，其他平台使用软件解码。

6.2 API 参考

6.2.1 RKADK_PLAYER_Create

【描述】

创建播放器。

【语法】

RKADK_S32 RKADK_PLAYER_Create([RKADK_MW_PTR](#) *ppPlayer, [RKADK_PLAYER_CFG_S](#) *pstPlayCfg);

【参数】

参数名称	描述	输入/输出
ppPlayer	创建的播放器指针	输出
pstPlayCfg	播放器属性	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

- 不支持重复创建同一播放器。

【举例】

[rkadk_player_test](#)

【相关主题】

[RKADK_PLAYER_Destroy](#)

6.2.2 RKADK_PLAYER_Destroy

【描述】

销毁播放器。

【语法】

RKADK_S32 RKADK_PLAYER_Destroy([RKADK_MW_PTR](#) pPlayer);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

- 不支持重复销毁同一播放器。

【举例】

[rkadk_player_test](#)

【相关主题】

[RKADK_PLAYER_Create](#)

6.2.3 RKADK_PLAYER_SetDataSource

【描述】

设置待播放文件路径，Player使能第三方demuxer库时，需使用[RKADK_PLAYER_SetDataParam](#)。

【语法】

RKADK_S32 RKADK_PLAYER_SetDataSource([RKADK_MW_PTR](#) pPlayer, const RKADK_CHAR *pszfilePath);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入
pszfilePath	待播放文件路径	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

- 创建播放器后，才能调用RKADK_PLAYER_SetDataSource接口。

【举例】

[rkadk_player_test](#)

【相关主题】

无

6.2.4 RKADK_PLAYER_SetDataParam

【描述】

设置待播放文件的音视频参数，Player使能第三方demuxer库时，使用该接口，否则使用[RKADK_PLAYER_SetDataSource](#)。

【语法】

```
RKADK_S32 RKADK_PLAYER_SetDataParam(RKADK\_MW\_PTR pPlayer,
RKADK\_PLAYER\_DATA\_PARAM\_S *pstDataParam);
```

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入
pstDataParam	待播放文件的音视频参数	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

- 创建播放器后，才能调用RKADK_PLAYER_SetDataParam接口。

【举例】

[rkadk_player_test](#)

【相关主题】

[RKADK_PLAYER_SendAudioPacket](#)

[RKADK_PLAYER_SendVideoPacket](#)

6.2.5 RKADK_PLAYER_Prepare

【描述】

播放准备。

【语法】

RKADK_S32 RKADK_PLAYER_Prepare([RKADK_MW_PTR](#) pPlayer);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

- 创建播放器和设置播放路径后，才能调用RKADK_PLAYER_Prepare接口。

【举例】

[rkadk_player_test](#)

【相关主题】

无

6.2.6 RKADK_PLAYER_GetCurrentPosition

【描述】

获取当前播放进度。

【语法】

RKADK_S64 RKADK_PLAYER_GetCurrentPosition(RKADK_MW_PTR pPlayer);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入

【返回值】

返回值	描述
播放进度，单位ms	成功
-1	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk_player_test](#)

【相关主题】

无

6.2.7 RKADK_PLAYER_Play

【描述】

开始播放。

【语法】

RKADK_S32 RKADK_PLAYER_Play([RKADK_MW_PTR](#) pPlayer);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

- 调用RKADK_PLAYER_Prepare 后，才能调用RKADK_PLAYER_Play接口。

【举例】

[rkadk_player_test](#)

【相关主题】

[RKADK_PLAYER_Stop](#)

6.2.8 RKADK_PLAYER_Stop

【描述】

停止播放，并释放资源。

【语法】

RKADK_S32 RKADK_PLAYER_Stop([RKADK_MW_PTR](#) pPlayer);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk_player_test](#)

【相关主题】

[RKADK_PLAYER_Play](#)

6.2.9 RKADK_PLAYER_Pause

【描述】

暂停播放。

【语法】

RKADK_S32 RKADK_PLAYER_Pause([RKADK_MW_PTR](#) pPlayer);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk_player_test](#)

【相关主题】

[RKADK_PLAYER_Play](#)

6.2.10 RKADK_PLAYER_Seek

【描述】

Seek。

【语法】

RKADK_S32 RKADK_PLAYER_Seek([RKADK_MW_PTR](#) pPlayer, RKADK_S64 s64TimeInMs);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入
s64TimeInMs	Seek 时长	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

- 调用RKADK_PLAYER_Play后，才能调用RKADK_PLAYER_Seek接口。

【举例】

[rkadk_player_test](#)

【相关主题】

[RKADK_PLAYER_Play](#)

6.2.11 RKADK_PLAYER_GetPlayStatus

【描述】

获取当前播放状态。

【语法】

RKADK_S32 RKADK_PLAYER_GetPlayStatus([RKADK_MW_PTR](#) pPlayer, [RKADK_PLAYER_STATE_E](#) *penState);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入
penState	当前播放状态	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk_player_test](#)

【相关主题】

无

6.2.12 RKADK_PLAYER_GetDuration

【描述】

获取当前播放文件时长。

【语法】

RKADK_S32 RKADK_PLAYER_GetDuration([RKADK_MW_PTR](#) pPlayer, RKADK_U32 *pDuration);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入
pDuration	当前播放文件时长，单位ms	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk_player_test](#)

【相关主题】

无

6.2.13 RKADK_PLAYER_Snapshot

【描述】

Player屏幕快照，调用该接口，会将当前显示的画面，编码成JPEG数据，应用可通过注册[RKADK_PPLAYER_SNAPSHOT_RECV_FN](#)回调获取生成的JPEG数据。

【语法】

RKADK_S32 RKADK_PLAYER_Snapshot([RKADK_MW_PTR](#) pPlayer);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【举例】

[rkadk_player_test](#)

【相关主题】

无

6.2.14 RKADK_PLAYER_SendAudioPacket

【描述】

Player使能第三方demuxer库时，使用该接口发送解封装后的音频数据给Player解码播放。

【语法】

RKADK_S32 RKADK_PLAYER_SendAudioPacket([RKADK_MW_PTR](#) pPlayer, [RKADK_PLAYER_PACKET](#) *pstPacket);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入
pstPacket	发送的数据包	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

- 使用该接口前，需要先调用RKADK_PLAYER_SetDataParam设置音视频相关参数。

【举例】

[rkadk_player_test](#)

【相关主题】

[RKADK_PLAYER_SendVideoPacket](#)

[RKADK_PLAYER_SetDataParam](#)

6.2.15 RKADK_PLAYER_SendVideoPacket

【描述】

Player使能第三方demuxer库时，使用该接口发送解封装后的视频数据给Player解码播放。

【语法】

RKADK_S32 RKADK_PLAYER_SendVideoPacket([RKADK_MW_PTR](#) pPlayer, [RKADK_PLAYER_PACKET](#) *pstPacket);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入
pstPacket	发送的数据包	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

- 使用该接口前，需要先调用RKADK_PLAYER_SetDataParam设置音视频相关参数。

【举例】

[rkadk_player_test](#)

【相关主题】

[RKADK_PLAYER_SendAudioPacket](#)

[RKADK_PLAYER_SetDataParam](#)

6.2.16 RKADK_PLAYER_GetSendFrameNum

【描述】

获取当前解码成功的帧数。

【语法】

RKADK_S32 RKADK_PLAYER_GetSendFrameNum([RKADK_MW_PTR](#) pPlayer);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入

【返回值】

返回值	描述
帧数	成功
-1	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

- Pause/Stop后将重新计数。

【举例】

[rkadk_player_test](#)

【相关主题】

无

6.2.17 RKADK_PLAYER_SetVdecWaterline

【描述】

设置VDEC水线。

【语法】

RKADK_S32 RKADK_PLAYER_SetVdecWaterline([RKADK_MW_PTR](#) pPlayer, RKADK_U32 u32VdecWaterline);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入
u32VdecWaterline	水线值	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

- 设置VDEC水线后，只有当VDEC缓存buffer到达水线值才会开始播放

【举例】

[rkadk_player_test](#)

【相关主题】

无

6.2.18 RKADK_PLAYER_SetAoVolume

【描述】

设置播放音量。

【语法】

RKADK_S32 RKADK_PLAYER_SetAoVolume([RKADK_MW_PTR](#) pPlayer, RKADK_S32 s32Volume);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入
s32Volume	音量值	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_player.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk_player_test](#)

【相关主题】

无

6.3 数据类型

播放模块主要提供以下数据类型：

[RKADK_PLAYER_EVENT_E](#)：播放事件枚举类型

[RKADK_PLAYER_EVENT_FN](#)：播放事件回调函数指针

[RKADK_PLAYER_CFG_S](#)：播放器属性结构体

[RKADK_VO_FORMAT_E](#)：图像像素格式枚举类型

[RKADK_VO_INTF_TYPE_E](#): 显示接口枚举类型

[RKADK_PLAYER_FRAME_INFO_S](#): 图像信息结构体

[RKADK_PLAYER_STATE_E](#): 播放状态枚举类型

[RKADK_PLAYER_DATA_PARAM_S](#): 音视频参数结构体

[RKADK_PLAYER_PACKET](#): 播放器数据包结构体

[RKADK_PLAYER_RTSP_CFG_S](#): Rtsp属性结构体

[RKADK_PLAYER_VDEC_CFG_S](#): VDEC属性结构体

[RKADK_PLAYER_SNAPSHOT_CFG_S](#): 屏幕快照属性结构体

[RKADK_PLAYER_SNAPSHOT_S](#): 屏幕快照数据结构体

[RKADK_PPLAYER_SNAPSHOT_RECV_FN](#): 屏幕快照数据回调函数指针

6.3.1 RKADK_PLAYER_EVENT_E

【说明】

定义播放事件枚举类型。

【定义】

```
typedef enum {  
    RKADK_PLAYER_EVENT_STATE_CHANGED = 0x0,  
    RKADK_PLAYER_EVENT_PREPARED,  
    RKADK_PLAYER_EVENT_PLAY,  
    RKADK_PLAYER_EVENT_PAUSED,  
    RKADK_PLAYER_EVENT_STOPPED,  
    RKADK_PLAYER_EVENT_EOF,  
    RKADK_PLAYER_EVENT_SOF,  
    RKADK_PLAYER_EVENT_SEEK_END,  
    RKADK_PLAYER_EVENT_ERROR,  
    RKADK_PLAYER_EVENT_BUTT  
} RKADK_PLAYER_EVENT_E;
```

【成员】

成员名称	描述
RKADK_PLAYER_EVENT_STATE_CHANGED	状态改变 (Reserved)
RKADK_PLAYER_EVENT_PREPARED	Prepared 完成
RKADK_PLAYER_EVENT_PLAY	开始播放
RKADK_PLAYER_EVENT_PAUSED	暂停播放 (Reserved)
RKADK_PLAYER_EVENT_STOPPED	停止播放
RKADK_PLAYER_EVENT_EOF	播放结束
RKADK_PLAYER_EVENT_SOF	Reserved
RKADK_PLAYER_EVENT_SEEK_END	Seek 完成 (Reserved)

【相关数据类型及接口】

[RKADK_PLAYER_EVENT_FN](#)

6.3.2 RKADK_PLAYER_EVENT_FN

【说明】

定义播放事件回调函数指针。

【定义】

```
typedef RKADK_VOID (*RKADK_PLAYER_EVENT_FN)(RKADK_MW_PTR pPlayer, RKADK_PLAYER_EVENT_E
enEvent, RKADK_VOID *pData);
```

【成员】

成员名称	描述
pPlayer	播放器指针
enEvent	事件类型
pData	事件相关参数

【相关数据类型及接口】

[RKADK_PLAYER_EVENT_E](#)

[RKADK_PLAYER_CFG_S](#)

6.3.3 RKADK_PLAYER_CFG_S

【说明】

定义播放器属性结构体。

【定义】

```
typedef struct {
    RKADK_BOOL bEnableVideo;
    RKADK_BOOL bEnableAudio;
    RKADK_BOOL bEnableThirdDemuxer;
    RKADK_PLAYER_FRAME_INFO_S stFrmInfo;
    RKADK_PLAYER_RTSP_CFG_S stRtspCfg;
    RKADK_PLAYER_VDEC_CFG_S stVdecCfg;
    RKADK_PLAYER_SNAPSHOT_CFG_S stSnapshotCfg;
    RKADK_BOOL bEnableBlackBackground;
    RKADK_PLAYER_EVENT_FN pfnPlayerCallback;
} RKADK_PLAYER_CFG_S;
```

【成员】

成员名称	描述
bEnableVideo	使能视频播放
bEnableAudio	使能音频播放
bEnableThirdDemuxer	使能第三方demuxer库
pfnPlayerCallback	播放事件回调函数指针
stFrmInfo	定义图像信息
stRtspCfg	Rtsp属性
stVdecCfg	VDEC属性
stSnapshotCfg	屏幕快照属性
bEnableBlackBackground	播放完是否黑屏

【相关数据类型及接口】

[RKADK_PLAYER_EVENT_FN](#)

[RKADK_PLAYER_FRAME_INFO_S](#)

[RKADK_PLAYER_RTSP_CFG_S](#)

[RKADK_PLAYER_VDEC_CFG_S](#)

[RKADK_PLAYER_SNAPSHOT_CFG_S](#)

[RKADK_PLAYER_Create](#)

6.3.4 RKADK_VO_FORMAT_E

【说明】

定义图像像素格式枚举类型。

【定义】

```
typedef enum {
    VO_FORMAT_ARGB8888 = 0,
    VO_FORMAT_ABGR8888,
    VO_FORMAT_RGB888,
    VO_FORMAT_BGR888,
    VO_FORMAT_ARGB1555,
    VO_FORMAT_ABGR1555,
    VO_FORMAT_RGB565,
    VO_FORMAT_RGB444,
    VO_FORMAT_NV12,
    VO_FORMAT_NV21
} RKADK_VO_FORMAT_E;
```

【相关数据类型及接口】

[RKADK_PLAYER_FRAME_INFO_S](#)

6.3.5 RKADK_VO_INTF_TYPE_E

【说明】

定义显示接口枚举类型。

【定义】

```
typedef enum {
    DISPLAY_TYPE_HDMI = 0,
    DISPLAY_TYPE_EDP,
    DISPLAY_TYPE_VGA,
    DISPLAY_TYPE_DP,
    DISPLAY_TYPE_HDMI_EDP,
    DISPLAY_TYPE_MIPI,
    DISPLAY_TYPE_DEFAULT,
} RKADK_VO_INTF_TYPE_E;
```

【成员】

成员名称	描述
DISPLAY_TYPE_HDMI	显示接口为HDMI
DISPLAY_TYPE_EDP	显示接口为EDP
DISPLAY_TYPE_VGA	显示接口为VGA
DISPLAY_TYPE_MIPI	显示接口为MIPI
DISPLAY_TYPE_DP	显示接口为DP
DISPLAY_TYPE_HDMI_EDP	显示接口为HDMI EDP
DISPLAY_TYPE_DEFAULT	内部检测，显示接口为实际接入的硬件

【相关数据类型及接口】

[RKADK_PLAYER_FRAME_INFO_S](#)

6.3.6 RKADK_VO_SPLICE_MODE_E

【说明】

定义图层合成方式枚举类型。

【定义】

```
typedef enum {
    SPLICE_MODE_RGA = 0,
    SPLICE_MODE_GPU,
    SPLICE_MODE_BYPASS
} RKADK_VO_INTF_TYPE_E;
```

【成员】

成员名称	描述
SPLICE_MODE_RGA	RGA合成
SPLICE_MODE_GPU	GPU合成
SPLICE_MODE_BYPASS	直通，不合成

【相关数据类型及接口】

[RKADK_PLAYER_FRAME_INFO_S](#)

6.3.7 RKADK_PLAYER_FRAME_INFO_S

【说明】

定义图像信息结构体。

【定义】

```
typedef struct {
    RKADK_U32 u32FrmInfoX;
    RKADK_U32 u32FrmInfoY;
    RKADK_U32 u32DispWidth;
    RKADK_U32 u32DispHeight;
    RKADK_U32 u32ImgWidth;
    RKADK_U32 u32ImgHeight;
    RKADK_U32 u32VoLay;
    RKADK_U32 u32VoDev;
    RKADK_U32 u32VoChn;
    RKADK_U32 u32BorderColor;
    RKADK_U32 u32BorderTopWidth;
    RKADK_U32 u32BorderBottomWidth;
    RKADK_U32 u32BorderLeftWidth;
    RKADK_U32 u32BorderRightWidth;
    RKADK_BOOL bMirror;
    RKADK_BOOL bFlip;
    RKADK_U32 u32Rotation; //0: 0, 1: 90, 2: 180, 3: 270
    RKADK_VO_FORMAT_E u32VoFormat;
```

```

RKADK_VO_INTF_TYPE_E u32EnIntfType;
RKADK_VO_INTF_SYNC_E enIntfSync;
RKADK_VO_SYNC_INFO_S stSyncInfo;
RKADK_VO_SPLICE_MODE_E enVoSpliceMode;
} RKADK_PLAYER_FRAMEINFO_S;

```

【成员】

成员名称	描述
u32FrmlInfoX	Layer显示区域x坐标
u32FrmlInfoY	Layer显示区域y坐标
u32DispWidth	Layer显示区域宽度
u32DispHeight	Layer显示区域高度
u32ImgWidth	Layer画布宽度
u32ImgHeight	Layer画布高度
u32VoLay	视频输出视频层号
u32VoDev	显示输出设备号
u32VoChn	视频输出通道号，取值范围：[0, VO_MAX_CHN_NUM(128)]
u32BorderColor	视频输出通道边框属性：颜色（Reserved）
u32BorderTopWidth	视频输出通道边框属性：上边框宽（Reserved）
u32BorderBottomWidth	视频输出通道边框属性：下边框宽（Reserved）
u32BorderLeftWidth	视频输出通道边框属性：左边框宽（Reserved）
u32BorderRightWidth	视频输出通道边框属性：右边框宽（Reserved）
bMirror	使能mirror
bFlip	使能flip
u32Rotation	旋转，取值：[0: 0, 1: 90, 2: 180, 3: 270]
u32VoFormat	定义图像像素格式
u32EnIntfType	显示接口类型
enIntfSync	屏幕接口同步模式
stSyncInfo	屏幕属性结构体
enVoSpliceMode	图层合成方式

【注意】

- 视频输出相关属性具体可参考Rockit 文档Rockchip_Developer_Guide_MPI.pdf VO章节。

【相关数据类型及接口】

[RKADK_VO_FORMAT_E](#)

[RKADK_VO_INTF_TYPE_E](#)

[RKADK_VO_SPLICE_MODE_E](#)

[RKADK_PLAYER_Create](#)

6.3.8 RKADK_PLAYER_STATE_E

【说明】

定义播放状态枚举类型。

【定义】

```
typedef enum {  
    RKADK_PLAYER_STATE_IDLE = 0, /* The player state before init */  
    RKADK_PLAYER_STATE_INIT, /* The player is in the initial state. It changes  
        to the initial state after being SetDataSource */  
    RKADK_PLAYER_STATE_PREPARED, /* The player is in the prepared state */  
    RKADK_PLAYER_STATE_PLAY, /* The player is in the playing state */  
    RKADK_PLAYER_STATE_TPLAY, /* The player is in the trick playing state, Reserved */  
    RKADK_PLAYER_STATE_PAUSE, /* The player is in the pause state */  
    RKADK_PLAYER_STATE_ERR, /* The player is in the err state */  
    RKADK_PLAYER_STATE_BUTT  
} RKADK_PLAYER_STATE_E;
```

【相关数据类型及接口】

[RKADK_PLAYER_GetPlayStatus](#)

6.3.9 RKADK_PLAYER_SNAPSHOT_S

【说明】

定义屏幕快照数据结构体。

【定义】

```
typedef struct {  
    RKADK_U32 u32Width;  
    RKADK_U32 u32Height;  
    RKADK_U32 u32DataLen;  
    RKADK_U8 *pu8DataBuf;  
} RKADK_PLAYER_SNAPSHOT_S;
```

【成员】

成员名称	描述
u32Width	屏幕快照宽
u32Height	屏幕快照高
u32DataLen	屏幕快照数据长度
pu8DataBuf	屏幕快照数据指针

【相关数据类型及接口】

[RKADK_PPLAYER_SNAPSHOT_RECV_FN](#)

6.3.10 RKADK_PPLAYER_SNAPSHOT_RECV_FN

【说明】

定义屏幕快照数据回调函数指针。

【定义】

```
typedef void (*RKADK_PPLAYER_SNAPSHOT_RECV_FN)(RKADK_PLAYER_SNAPSHOT_S *pstData);
```

【成员】

成员名称	描述
pstData	屏幕快照数据指针

【相关数据类型及接口】

[RKADK_PLAYER_SNAPSHOT_S](#)

[RKADK_PLAYER_SNAPSHOT_CFG_S](#)

6.3.11 RKADK_PLAYER_SNAPSHOT_CFG_S

【说明】

定义屏幕快照属性结构体。

【定义】

```
typedef struct {
    RKADK_U32 u32VencChn;
    RKADK_U32 u32MaxWidth; //Support snapshot max width, default 4096
    RKADK_U32 u32MaxHeight; //Support snapshot max height, default 4096
    RKADK_PPLAYER_SNAPSHOT_RECV_FN pfnDataCallback;
} RKADK_PLAYER_SNAPSHOT_CFG_S;
```

【成员】

成员名称	描述
u32VencChn	JPEG编码通道
u32MaxWidth	屏幕快照最大宽，默认4096
u32MaxHeight	屏幕快照最大高，默认4096
pfnDataCallback	屏幕快照数据回调指针

【相关数据类型及接口】

[RKADK_PPLAYER_SNAPSHOT_RECV_FN](#)

[RKADK_PLAYER_CFG_S](#)

6.3.12 RKADK_PLAYER_VDEC_CFG_S

【说明】

定义VDEC属性结构体。

【定义】

```
typedef struct {
    RKADK_U32 u32FrameBufCnt; //frame buffer cnt(output), default: 3
    RKADK_U32 u32StreamBufCnt; //stream buffer cnt(input), default: 3
} RKADK_PLAYER_VDEC_CFG_S;
```

【成员】

成员名称	描述
u32FrameBufCnt	输出buffer个数，默认3个
u32StreamBufCnt	输入buffer个数，默认3个

【相关数据类型及接口】

[RKADK_PLAYER_CFG_S](#)

6.3.13 RKADK_PLAYER_RTSP_CFG_S

【说明】

定义Rtsp属性结构体。

【定义】

```
typedef struct {
    const char *transport; //udp or tcp, default: udp
    RKADK_U32 u32IoTimeout; //timeout (in microseconds) of socket I/O operations
} RKADK_PLAYER_RTSP_CFG_S;
```

【成员】

成员名称	描述
transport	传输协议，默认UDP
u32IoTimeout	Socket I/O 操作超时时长，单位ms，默认不超时

【相关数据类型及接口】

[RKADK_PLAYER_CFG_S](#)

6.3.14 RKADK_PLAYER_PACKET

【说明】

定义数据包结构体，使能第三方demuxer库是使用。

【定义】

```
typedef struct {
    bool bEofFlag;
    RKADK_S8 *s8PacketData;
    RKADK_S32 s32PacketSize;
    RKADK_U32 u32Seq;
    RKADK_S64 s64Pts;

    //if bypass, must set pFreeCB;
    bool bBypass;
    RKADK_MPI_MB_FREE_CB pFreeCB;
} RKADK_PLAYER_PACKET;
```

【成员】

成员名称	描述
bEofFlag	是否最后一帧数据
s8PacketData	数据指针
s32PacketSize	数据长度
u32Seq	序列号
s64Pts	时间戳
bBypass	数据传输方式，true：直通，false：二次拷贝
pFreeCB	s8PacketData释放函数指针，bBypass 为true时必须设置

【相关数据类型及接口】

[RKADK_PLAYER_SendAudioPacket](#)

[RKADK_PLAYER_SendVideoPacket](#)

6.3.15 RKADK_PLAYER_DATA_PARAM_S

【说明】

定义音视频参数结构体，使能第三方demuxer库时使用。

【定义】

```
typedef struct {
    const RKADK_CHAR *pFilePath;
    RKADK_BOOL blsRtsp;
    RKADK_BOOL bVideoExist;
    RKADK_BOOL bAudioExist;

    //video param
    RKADK_CODEC_TYPE_E enVCodecType;
    RKADK_U32 u32Width;
    RKADK_U32 u32Height;
    RKADK_FORMAT_E enPixFmt; //output pixel format
    RKADK_U32 u32FrameRate;

    //audio param
    RKADK_CODEC_TYPE_E enACodecType;
    RKADK_S32 s32BitWidth;
    RKADK_S32 s32SampleRate;
    RKADK_S32 s32Channel;
} RKADK_PLAYER_DATA_PARAM_S;
```

【成员】

成员名称	描述
pFilePath	待播放文件路径
blsRtsp	是否Rtsp网络流
bVideoExist	待播放文件是否存在视频流
bAudioExist	待播放文件是否存在音频流
enVCodecType	视频流解码格式
u32Width	视频流宽
u32Height	视频流高
u32FrameRate	视频流帧率
enACodecType	音频流解码格式
u32BitWidth	音频流位宽
u32SampleRate	音频流采样率
u32Channel	音频流通道数

【相关数据类型及接口】

7. 直播

7.1 概述

提供标准RTSP直播流的基本服务；提供RTMP直播服务。RTSP和RTMP不支持同时启动。

7.2 API参考

7.2.1 RTSP

7.2.1.1 RKADK_RTSP_Init

【描述】

初始化RTSP模块。

【语法】

RKADK_S32 RKADK_RTSP_Init([RKADK_U32](#) u32CamId, [RKADK_U32](#) port, const char *path, [RKADK_MW_PTR](#) *ppHandle);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera id	输入
port	端口号	输入
path	RTSP地址	输入
ppHandle	创建的RTSP Handle	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_rtsp.h

库文件：librkadk.so

【注意】

- RKADK_RTSP_Init 后调用RKADK_RTSP_Start 启动RTSP直播推流。

【举例】

[rkadk_rtsp_test](#)

【相关主题】

[RKADK_RTSP_DeInit](#)

7.2.1.2 RKADK_RTSP_DeInit

【描述】

反初始化RTSP模块。

【语法】

RKADK_S32 RKADK_RTSP_DeInit([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	创建的RTSP Handle	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_rtsp.h

库文件：librkadk.so

【举例】

[rkadk_rtsp_test](#)

【相关主题】

[RKADK_RTSP_Init](#)

7.2.1.3 RKADK_RTSP_Start

【描述】

启动RTSP直播。

【语法】

RKADK_S32 RKADK_RTSP_Start([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	创建的RTSP Handle	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_rtsp.h

库文件：librkadk.so

【注意】

- RKADK_RTSP_Init 后调用该接口。

【举例】

[rkadk_rtsp_test](#)

【相关主题】

[RKADK_RTSP_Stop](#)

7.2.1.4 RKADK_RTSP_Stop

【描述】

停止RTSP直播。

【语法】

RKADK_S32 RKADK_RTSP_Stop([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	创建的RTSP Handle	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_rtsp.h

库文件：librkadk.so

【举例】

[rkadk_rtsp_test](#)

【相关主题】

[RKADK_RTSP_Start](#)

7.2.2 RTMP

7.2.2.1 RKADK_RTMP_Init

【描述】

初始化RTMP模块。

【语法】

RKADK_S32 RKADK_RTMP_Init([RKADK_U32](#) u32CamId, const char *path, [RKADK_MW_PTR](#) *ppHandle);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera id	输入
path	RTMP地址	输入
ppHandle	创建的RTMP Handle	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_rtmp.h

库文件：librkadk.so

【举例】

[rkadk_rtmp_test](#)

【相关主题】

[RKADK_RTMP_DeInit](#)

7.2.2.2 RKADK_RTMP_DeInit

【描述】

反初始化RTMP模块。

【语法】

RKADK_S32 RKADK_RTMP_DeInit([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	创建的RTMP Handle	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_rtmp.h

库文件：librkadk.so

【举例】

[rkadk_rtmp_test](#)

【相关主题】

[RKADK_RTMP_Init](#)

8. 存储

8.1 概述

提供基本的存储功能，目前包含如下功能：

- 文件检测、存储、获取、管理
- 设备容量和状态查询
- 自动删除文件
- 格式化

8.2 API参考

8.2.1 RKADK_STORAGE_Init

【描述】

存储模块初始化。

【语法】

```
RKADK_S32 RKADK_STORAGE_Init(RKADK\_MW\_PTR *ppHandle, RKADK\_STR\_DEV\_ATTR
*pstDevAttr);
```

【参数】

参数名称	描述	输入/输出
ppHandle	创建的存储模块句柄	输出
pstDevAttr	挂载设备属性指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_storage.h

库文件：librkadk.so

【注意】

- 不支持重复初始化。
- [RKADK_STR_DEV_ATTR](#)传入NULL使用默认属性，默认属性为：2个文件夹、命名为video_front和video_back、以文件名排序、自动删除阈值500~1000M、不使用文件个数限制、限制比例为50%。

【举例】

[rkadk_storage_test](#)

【相关主题】

[RKADK_STORAGE_Deinit](#)

8.2.2 RKADK_STORAGE_Deinit

【描述】

存储模块反初始化。

【语法】

RKADK_S32 RKADK_STORAGE_Deinit([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	存储模块句柄指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_storage.h

库文件：librkadk.so

【注意】

- 不支持重复反初始化。

【举例】

[rkadk_storage_test](#)

【相关主题】

[RKADK_STORAGE_Init](#)

8.2.3 RKADK_STORAGE_GetDevAttr

【描述】

获取挂载设备属性。

【语法】

[RKADK_STR_DEV_ATTR](#) RKADK_STORAGE_GetDevAttr([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	存储模块句柄指针	输入

【返回值】

返回值	描述
RKADK_STR_DEV_ATTR	挂载设备属性结构体

【需求】

头文件：rkadk_storage.h

库文件：librkadk.so

【注意】

- 存储模块初始化后，才能使用[RKADK_STORAGE_GetDevAttr](#)接口。

【举例】

无

【相关主题】

无

8.2.4 RKADK_STORAGE_GetMountStatus

【描述】

获取设备挂载状态。

【语法】

[RKADK_MOUNT_STATUS](#) RKADK_STORAGE_GetMountStatus([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	存储模块句柄指针	输入

【返回值】

返回值	描述
RKADK_MOUNT_STATUS	挂载状态枚举类型

【需求】

头文件：rkadk_storage.h

库文件：librkadk.so

【举例】

[rkadk_storage_test](#)

【相关主题】

无

8.2.5 RKADK_STORAGE_GetCapacity

【描述】

获取设备容量。

【语法】

RKADK_S32 RKADK_STORAGE_GetCapacity([RKADK_MW_PTR](#) *ppHandle, RKADK_S32 *totalSize, RKADK_S32 *freeSize);

【参数】

参数名称	描述	输入/输出
ppHandle	创建的存储模块句柄	输入/输出
totalSize	设备总容量指针	输出
freeSize	设备剩余容量指针	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_storage.h

库文件：librkadk.so

【举例】

[rkadk_storage_test](#)

【相关主题】

无

8.2.6 RKADK_STORAGE_GetFileList

【描述】

获取文件列表。

【语法】

RKADK_S32 RKADK_STORAGE_GetFileList([RKADK_FILE_LIST](#) *list, [RKADK_MW_PTR](#) pHandle, [RKADK_SORT_TYPE](#) sort);

【参数】

参数名称	描述	输入/输出
list	文件列表结构体指针	输入/输出
pHandle	存储模块句柄指针	输入
sort	排序类型	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_storage.h

库文件：librkadk.so

【注意】

- 必须和[RKADK_STORAGE_FreeFileList](#)配套使用。

【举例】

[rkadk_storage_test](#)

【相关主题】

[RKADK_STORAGE_FreeFileList](#)

8.2.7 RKADK_STORAGE_FreeFileList

【描述】

释放文件列表。

【语法】

RKADK_S32 RKADK_STORAGE_FreeFileList([RKADK_FILE_LIST](#) *list);

【参数】

参数名称	描述	输入/输出
list	文件列表结构体指针	输入/输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_storage.h

库文件：librkadk.so

【注意】

- 必须和[RKADK_STORAGE_GetFileList](#)配套使用。

【举例】

[rkadk_storage_test](#)

【相关主题】

[RKADK_STORAGE_GetFileList](#)

8.2.8 RKADK_STORAGE_GetFileNum

【描述】

获取文件数量。

【语法】

RKADK_S32 RKADK_STORAGE_GetFileNum(RKADK_CHAR *fileListPath, [RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
fileListPath	文件列表路径指针	输入
pHandle	存储模块句柄指针	输入

【返回值】

返回值	描述
非负	文件数量
-1	失败

【需求】

头文件：rkadk_storage.h

库文件：librkadk.so

【举例】

无

【相关主题】

无

8.2.9 RKADK_STORAGE_GetDevPath

【描述】

获取挂载设备路径。

【语法】

RKADK_CHAR *RKADK_STORAGE_GetDevPath([RKADK_MW_PTR](#) pHandle);

【参数】

参数名称	描述	输入/输出
pHandle	存储模块句柄指针	输入

【返回值】

返回值	描述
RKADK_CHAR *	挂载设备路径指针

【需求】

头文件：rkadk_storage.h

库文件：librkadk.so

【注意】

- 存储模块初始化后，才能使用[RKADK_STORAGE_GetDevPath](#)接口。

【举例】

[rkadk_storage_test](#)

【相关主题】

无

8.2.10 RKADK_STORAGE_Format

【描述】

设备格式化。

【语法】

RKADK_S32 RKADK_STORAGE_Format([RKADK_MW_PTR](#) pHandle, RKADK_CHAR *cFormat);

【参数】

参数名称	描述	输入/输出
pHandle	存储模块句柄指针	输入
cFormat	文件系统类型	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_storage.h

库文件：librkadk.so

【举例】

无

【相关主题】

无

8.3 数据类型

存储模块主要提供以下数据类型：

[RKADK_MOUNT_STATUS](#)：挂载状态枚举类型

[RKADK_SORT_TYPE](#)：排序类型枚举

[RKADK_SORT_CONDITION](#)：排序条件枚举类型

[RKADK_STR_FOLDER_ATTR](#)：文件夹属性结构体

[RKADK_STR_DEV_ATTR](#)：设备属性结构体

[RKADK_FILE_INFO](#)：文件信息结构体

[RKADK_FILE_LIST](#)：文件列表结构体

[RKADK_FILE_LIST_ARRAY](#)：文件列表组结构体

8.3.1 RKADK_MOUNT_STATUS

【说明】

定义挂载状态枚举类型。

【定义】

```
typedef enum {  
    DISK_UNMOUNTED = 0,  
    DISK_NOT_FORMATTED,  
    DISK_FORMAT_ERR,  
    DISK_SCANNING,  
    DISK_MOUNTED,  
    DISK_MOUNT_BUTT,  
} RKADK_MOUNT_STATUS;
```

【成员】

成员名称	描述
DISK_UNMOUNTED	磁盘未挂载
DISK_NOT_FORMATTED	磁盘未格式化
DISK_FORMAT_ERR	磁盘格式化错误
DISK_SCANNING	磁盘正在扫描中
DISK_MOUNTED	磁盘已挂载

【相关数据类型及接口】

[RKADK_STORAGE_GetMountStatus](#)

8.3.2 RKADK_SORT_TYPE

【说明】

定义排序类型枚举。

【定义】

```
typedef enum {  
    LIST_ASCENDING = 0,  
    LIST_DESCENDING,  
    LIST_BUTT,  
} RKADK_SORT_TYPE;
```

【成员】

成员名称	描述
LIST_ASCENDING	列表以升序排序
LIST_DESCENDING	列表以降序排序

【相关数据类型及接口】

[RKADK_STORAGE_GetFileList](#)

8.3.3 RKADK_SORT_CONDITION

【说明】

定义排序条件枚举类型。

【定义】

```
typedef enum {  
    SORT_MODIFY_TIME = 0,  
    SORT_FILE_NAME,  
    SORT_BUTT,  
} RKADK_SORT_CONDITION;
```

【成员】

成员名称	描述
SORT_MODIFY_TIME	列表以文件修改时间排序
SORT_FILE_NAME	列表以文件名排序

【相关数据类型及接口】

[RKADK_STR_FOLDER_ATTR](#)

8.3.4 RKADK_STR_FOLDER_ATTR

【说明】

定义文件夹属性结构体。

【定义】

```
typedef struct {  
    RKADK_CHAR cFolderPath[RKADK_MAX_FILE_PATH_LEN];  
    RKADK_SORT_CONDITION s32SortCond;  
    RKADK_BOOL bNumLimit;  
    RKADK_S32 s32Limit;  
} RKADK_STR_FOLDER_ATTR;
```

【成员】

成员名称	描述
cFolderPath	文件夹路径
s32SortCond	排序条件
bNumLimit	选项：是否以文件个数设置上限
s32Limit	文件夹容量上限（比例/个数）

【相关数据类型及接口】

[RKADK_SORT_CONDITION](#)

[RKADK_STR_DEV_ATTR](#)

8.3.5 RKADK_STR_DEV_ATTR

【说明】

定义设备属性结构体。

【定义】

```
typedef struct {
    RKADK_CHAR cDevPath[RKADK_MAX_FILE_PATH_LEN];
    RKADK_CHAR cMountPath[RKADK_MAX_FILE_PATH_LEN];
    RKADK_S32 s32FreeSizeDelMin;
    RKADK_S32 s32FreeSizeDelMax;
    RKADK_S32 s32AutoDel;
    RKADK_S32 s32FolderNum;
    RKADK_CHAR cFormatId[RKADK_MAX_FORMAT_ID_LEN];
    RKADK_CHAR cVolume[RKADK_MAX_VOLUME_LEN];
    RKADK_S32 s32CheckFormatId;
    RKADK_STR_FOLDER_ATTR *pstFolderAttr;
} RKADK_STR_DEV_ATTR;
```

【成员】

成员名称	描述
cDevPath	设备名（设备路径）
cMountPath	设备挂载路径
s32FreeSizeDelMin	自动删除阈值下限
s32FreeSizeDelMax	自动删除阈值上限
s32AutoDel	自动删除选项
s32FolderNum	文件夹个数
cFormatId	格式化ID
cVolume	卷标
s32CheckFormatId	检测格式化ID是否匹配
pstFolderAttr	文件夹属性结构体指针

【相关数据类型及接口】

[RKADK_STR_FOLDER_ATTR](#)

[RKADK_STORAGE_Init](#)

[RKADK_STORAGE_GetDevAttr](#)

8.3.6 RKADK_FILE_INFO

【说明】

定义文件信息结构体。

【定义】

```
typedef struct {
    RKADK_CHAR filename[RKADK_MAX_FILE_PATH_LEN];
    off_t stSize;
    time_t stTime;
    void *thumb;
} RKADK_FILE_INFO;
```

【成员】

成员名称	描述
filename	文件名
stSize	文件大小
stTime	文件修改时间
thumb	缩略图指针

【相关数据类型及接口】

[RKADK_FILE_LIST](#)

8.3.7 RKADK_FILE_LIST

【说明】

定义文件列表结构体。

【定义】

```
typedef struct {
    RKADK_CHAR path[RKADK_MAX_FILE_PATH_LEN];
    RKADK_S32 s32FileNum;
    RKADK_FILE_INFO *file;
} RKADK_FILE_LIST;
```

【成员】

成员名称	描述
path	文件列表（文件夹）路径
s32FileNum	文件个数
file	文件信息结构体指针

【相关数据类型及接口】

[RKADK_FILE_INFO](#)

[RKADK_FILE_LIST_ARRAY](#)

8.3.8 RKADK_FILE_LIST_ARRAY

【说明】

定义文件列表组结构体。

【定义】

```
typedef struct {
    RKADK_S32 s32ListNum;
    RKADK_FILE_LIST *list;
} RKADK_FILE_LIST_ARRAY;
```

【成员】

成员名称	描述
s32ListNum	文件列表（文件夹）个数
list	文件列表结构体指针

【相关数据类型及接口】

[RKADK_FILE_LIST](#)

9. 本地预览

9.1 概述

提供本地预览功能。

9.2 API参考

9.2.1 RKADK_DISP_Init

【描述】

初始化本地预览模块。

【语法】

RKADK_S32 RKADK_DISP_Init([RKADK_U32](#) u32CamId);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera id	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_disp.h

库文件：librkadk.so

【举例】

[rkadk_disp_test](#)

【相关主题】

[RKADK_DISP_DeInit](#)

9.2.2 RKADK_DISP_DeInit

【描述】

反初始化本地预览模块。

【语法】

RKADK_S32 RKADK_DISP_DeInit([RKADK_U32](#) u32CamId);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera id	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_disp.h

库文件：librkadk.so

【举例】

[rkadk_disp_test](#)

【相关主题】

[RKADK_DISP_Init](#)

9.2.3 RKADK_DISP_SetAttr

【描述】

设置预览属性。

【语法】

RKADK_S32 RKADK_DISP_SetAttr(RKADK_U32 u32CamId, [RKADK_DISP_ATTR_S](#) *pstAttr);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera id	输入
pstAttr	预览属性	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_disp.h

库文件：librkadk.so

9.3 数据类型

9.3.1 RKADK_DISP_ATTR_S

【说明】

定义预览属性结构体。

【定义】

```
typedef struct {
    RKADK_RECT_S stVpssCropRect;
    RKADK_RECT_S stVoRect;
} RKADK_DISP_ATTR_S;
```

【成员】

成员名称	描述
stVpssCropRect	输入显示区域
stVoRect	输出显示区域

10. 水印

10.1 概述

提供基本的水印功能

10.2 API 参考

10.2.1 RKADK_OSD_Init

【描述】

初始化水印任务。

【语法】

```
RKADK_S32 RKADK_OSD_Init(RKADK\_U32 u32OsdId, RKADK\_OSD\_ATTR\_S *pstOsdAttr);
```

【参数】

参数名称	描述	输入/输出
u32OsdId	水印id	输入
pstOsdAttr	水印任务属性指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_osd.h

库文件：librkadk.so

【注意】

- 不支持重复初始化。

10.2.2 RKADK_OSD_Deinit

【描述】

反初始化水印任务。

【语法】

RKADK_S32 RKADK_OSD_Deinit([RKADK_U32](#) u32OsdId);

【参数】

参数名称	描述	输入/输出
u32OsdId	水印id	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_osd.h

库文件：librkadk.so

【注意】

- 不支持重复反初始化。

10.2.3 RKADK_OSD_UpdateBitMap

【描述】

水印内容更新。

【语法】

RKADK_S32 RKADK_OSD_UpdateBitMap([RKADK_U32](#) u32OsdId, [RKADK_OSD_ATTR_S](#) *pstOsdAttr);

【参数】

参数名称	描述	输入/输出
u32OsdId	水印id	输入
pstOsdAttr	水印任务属性指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_osd.h

库文件：librkadk.so

【注意】

- 必须反初始化任务完成后使用。

10.2.4 RKADK_OSD_AttachToStream

【描述】

水印叠加到目标流。

【语法】

RKADK_S32 RKADK_OSD_AttachToStream([RKADK_U32](#) u32OsdId, [RKADK_U32](#) u32CamId, [RKADK_STREAM_TYPE_E](#) enStrmType, [RKADK_OSD_STREAM_ATTR_S](#) *pstOsdStreamAttr);

【参数】

参数名称	描述	输入/输出
u32OsdId	水印id	输入
u32CamId	Camera id	输入
enStrmType	目标流类型	输入
pstOsdStreamAttr	水印位置信息指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_osd.h

库文件：librkadk.so

【注意】

- 同一水印不能在一流类型上重复叠加。

10.2.5 RKADK_OSD_DettachFromStream

【描述】

水印脱离目标流。

【语法】

RKADK_S32 RKADK_OSD_DettachFromStream([RKADK_U32](#) u32OsdId, [RKADK_U32](#) u32CamId, [RKADK_STREAM_TYPE_E](#) enStrmType);

【参数】

参数名称	描述	输入/输出
u32OsdId	水印id	输入
u32CamId	Camera id	输入
enStrmType	目标流类型	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_osd.h

库文件：librkadk.so

10.2.6 RKADK_OSD_UpdateOsdSize

【描述】

更新水印大小。

【语法】

RKADK_S32 RKADK_OSD_UpdateOsdSize(RKADK_U32 u32OsdId, [RKADK_OSD_ATTR_S](#) *pstOsdAttr);

【参数】

参数名称	描述	输入/输出
u32OsdId	水印id	输入
pstOsdAttr	水印任务属性指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_osd.h

库文件：librkadk.so

10.2.7 RKADK_OSD_UpdateDisplayAttr

【描述】

更新水印显示区域。

【语法】

RKADK_S32 RKADK_OSD_UpdateDisplayAttr(RKADK_U32 u32OsdId, RKADK_U32 u32CamId, [RKADK_STREAM_TYPE_E](#) enStrmType, [RKADK_OSD_STREAM_ATTR_S](#) *pstOsdStreamAttr);

【参数】

参数名称	描述	输入/输出
u32OsdId	水印id	输入
u32CamId	Camera Id	输入
enStrmType	流类型	输入
pstOsdStreamAttr	水印位置信息	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_osd.h

库文件：librkadk.so

10.3 数据类型

水印模块主要提供以下数据类型：

[RKADK_OSD_ATTR_S](#)：水印属性结构体

[RKADK_OSD_STREAM_ATTR_S](#)：水印位置信息结构体

10.3.1 RKADK_OSD_ATTR_S

【说明】

定义水印属性。

【定义】

```
typedef struct {
    RKADK_U32 Width;
    RKADK_U32 Height;
    RKADK_VOID *pData;
    RKADK_FORMAT_E Format;
    RKADK_OSD_TYPE_E enOsdType;
} RKADK_OSD_ATTR_S;
```

【成员】

成员名称	描述
Width	水印宽度
Height	水印高度
pData	水印内容
Format	水印格式
enOsdType	水印叠加类型

【相关数据类型及接口】

[RKADK_OSD_ATTR_S](#)

[RKADK_OSD_TYPE_E](#)

10.3.2 RKADK_OSD_STREAM_ATTR_S

【说明】

定义水印位置信息。

【定义】

```
typedef struct {
    RKADK_BOOL bEnableShow;
    RKADK_U32 Origin_X;
    RKADK_U32 Origin_Y;
} RKADK_OSD_STREAM_ATTR_S;
```

【成员】

成员名称	描述
Origin_X	水印起始位置X偏移量
Origin_Y	水印起始位置Y偏移量
bEnableShow	是否显示水印

【相关数据类型及接口】

[RKADK_OSD_STREAM_ATTR_S](#)

10.4 RKADK_OSD_TYPE_E

【说明】

定义水印叠加类型。

【定义】

```
typedef enum {  
    RKADK_OSD_TYPE_NORMAL = 0, //use encoder do osd  
    RKADK_OSD_TYPE_EXTRA,    //use rga do osd  
    RKADK_OSD_TYPE_BUTT  
} RKADK_OSD_TYPE_E;
```

【成员】

成员名称	描述
RKADK_OSD_TYPE_NORMAL	编码器OSD叠加
RKADK_OSD_TYPE_EXTRA	RGa叠加

【注意】

- RV1109/RV1126 JPEG编码不支持编码器OSD叠加水印，需配置成RKADK_OSD_TYPE_EXTRA。

11. UI叠加

11.1 概述

提供UI叠加功能，用于RV1103/RV1106等单层VOP平台的UI和视频画面的叠加。

11.2 API参考

11.2.1 RKADK_UI_Create

【描述】
初始化UI叠加模块

【语法】
RKADK_S32 RKADK_UI_Create([RKADK_UI_ATTR_S](#) *pstUiAttr, [RKADK_MW_PTR](#) *ppUi);

【参数】

参数名称	描述	输入/输出
pstUiAttr	UI属性	输入
ppUi	创建的UI任务指针	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】
头文件：rkadk_ui.h
库文件：librkadk.so

【举例】
[rkadk_ui_test](#)

【相关主题】
[RKADK_UI_Destroy](#)

11.2.2 RKADK_UI_Destroy

【描述】
反初始化UI叠加模块

【语法】
RKADK_S32 RKADK_UI_Destroy([RKADK_MW_PTR](#) pUi);

【参数】

参数名称	描述	输入/输出
pUi	UI任务指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_ui.h

库文件：librkadk.so

【举例】

[rkadk ui test](#)

【相关主题】

[RKADK UI Create](#)

11.2.3 RKADK_UI_Update

【描述】

刷新UI数据。

【语法】

RKADK_S32 RKADK_UI_Update([RKADK_MW_PTR](#) pUi, [RKADK_UI_FRAME_INFO](#) *pstUiFrameInfo);

【参数】

参数名称	描述	输入/输出
pUi	UI任务指针	输入
pstUiFrameInfo	UI数据指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_ui.h

库文件：librkadk.so

【举例】

[rkadk ui test](#)

11.3 数据类型

[RKADK_UI_ATTR_S](#): UI属性结构体

[RKADK_UI_FRAME_INFO](#): UI数据结构体

11.3.1 RKADK_UI_ATTR_S

【说明】

定义UI属性结构体。

【定义】

```
typedef struct {  
    RKADK_U32 u32DispX;  
    RKADK_U32 u32DispY;  
    RKADK_U32 u32DispWidth;  
    RKADK_U32 u32DispHeight;  
    RKADK_U32 u32DispFrmRt;  
    RKADK_U32 u32ImgWidth;  
    RKADK_U32 u32ImgHeight;  
    RKADK_U32 u32VoLay;  
    RKADK_U32 u32VoDev;  
    RKADK_U32 u32VoChn;  
    RKADK_U32 u32Rotation; //0: 0, 1: 90, 2: 180, 3: 270  
    RKADK_BOOL bMirror;  
    RKADK_BOOL bFlip;  
    RKADK_VO_FORMAT_E enUiVoFormat;  
    RKADK_VO_INTF_TYPE_E enUiVoIntfTye;  
    RKADK_VO_SPLICE_MODE_E enVoSpliceMode;  
} RKADK_UI_ATTR_S;
```

【成员】

成员名称	描述
u32DispX	Layer显示区域x坐标
u32DispY	Layer显示区域y坐标
u32DispWidth	Layer显示区域宽度
u32DispHeight	Layer显示区域高度
u32ImgWidth	Layer画布宽度
u32ImgHeight	Layer画布高度
u32DispFrmRt	刷新帧率
u32VoLay	视频输出视频层号
u32VoDev	显示输出设备号
u32VoChn	视频输出通道号，取值范围：[0, VO_MAX_CHN_NUM(128)]
u32Rotation	旋转，取值：[0: 0, 1: 90, 2: 180, 3: 270]
bMirror	使能mirror
bFlip	使能flip
enUiVoFormat	图像像素格式
enUiVoIntfTye	显示接口类型
enVoSpliceMode	图层合成方式

【相关数据类型及接口】

[RKADK_VO_FORMAT_E](#)

[RKADK_VO_INTF_TYPE_E](#)

[RKADK_VO_SPLICE_MODE_E](#)

[RKADK_UI_Create](#)

11.3.2 RKADK_UI_FRAME_INFO

【说明】

定义UI数据信息。

【定义】

```
typedef struct {  
    RKADK_U32 u32Width;  
    RKADK_U32 u32Height;  
    RKADK_FORMAT_E Format;  
    RKADK_VOID *pMblk;  
} RKADK_UI_FRAME_INFO;
```

【成员】

成员名称	描述
u32Width	UI数据宽度
u32Height	UI数据高度
Format	UI数据格式
pMblk	UI数据指针

【相关数据类型及接口】

[RKADK_FORMAT_E](#)

[RKADK_UI_Update](#)

11.3.3 RKADK_FORMAT_E

【说明】

定义像素格式。

【定义】

```
typedef enum {
    RKADK_FMT_ARGB1555,           /* 16-bit RGB */
    RKADK_FMT_ABGR1555,           /* 16-bit RGB */
    RKADK_FMT_RGBA5551,           /* 16-bit RGB */
    RKADK_FMT_BGRA5551,           /* 16-bit RGB */
    RKADK_FMT_ARGB4444,           /* 16-bit RGB */
    RKADK_FMT_ABGR4444,           /* 16-bit RGB */
    RKADK_FMT_RGBA4444,           /* 16-bit RGB */
    RKADK_FMT_BGRA4444,           /* 16-bit RGB */
    RKADK_FMT_ARGB8888,           /* 32-bit RGB */
    RKADK_FMT_ABGR8888,           /* 32-bit RGB */
    RKADK_FMT_RGBA8888,           /* 32-bit RGB */
    RKADK_FMT_BGRA8888,           /* 32-bit RGB */
    RKADK_FMT_2BPP,
    RKADK_FMT_YUV420SP,
    RKADK_FMT_YUV420SP_10BIT,
    RKADK_FMT_YUV422SP,
    RKADK_FMT_BUTT,
} RKADK_FORMAT_E;
```

【相关数据类型及接口】

[RKADK_UI_FRAME_INFO](#)

12. 参数设置

12.1 概述

参数设置模块与产品形态强相关，通过组合使用通用组件数据结构，定义出适合产品形态的数据结构。

该模块支持获取指定参数，支持保存指定参数，支持参数恢复默认。

为方便编辑，参数以ini文件形式存放。

可通过设置环境变量rkadk_default_ini_path，指定默认ini路径，默认路径为/oem/usr/etc。

```
export rkadk_default_ini_path=/oem/usr/etc
```

12.2 API参考

12.2.1 RKADK_PARAM_Init

【描述】

初始化参数模块

【语法】

RKADK_S32 RKADK_PARAM_Init(char *globalSetting, char **sesnorSettingArray);

【参数】

参数名称	描述	输入/输出
globalSetting	全局ini配置文件路径	输入
sesnorSettingArray	Sensor ini配置文件路径	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_param.h

库文件：librkadk.so

【注意】

- 启动任一模块之前，都必须先调用RKADK_PARAM_Init初始化参数模块。
- 如果没有设置globalSetting，则使用默认路径[RKADK_PARAM_PATH](#)。
- 如果没有设置sesnorSettingArray，则使用默认路径[RKADK_PARAM_PATH_SENSOR_PREFIX](#)，Sensor 配置文件前缀默认rkadk_setting_sensor_n.ini，_n 为Sensor Camera Id，序号从0开始。

【举例】

[rkadk_record_test](#)

【相关主题】

无

12.2.2 RKADK_PARAM_GetCamParam

【描述】

获取Camera相关的参数。

【语法】

RKADK_S32 RKADK_PARAM_GetCamParam(RKADK_S32 s32CamID, [RKADK_PARAM_TYPE_E](#) enParamType, RKADK_VOID *pvParam);

【参数】

参数名称	描述	输入/输出
s32CamID	Camera id	输入
enParamType	参数类型	输入
pvParam	获取到的参数指针	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_param.h

库文件：librkadk.so

【注意】

- 调用RKADK_PARAM_Init初始化参数模块后，才能调用该接口。

【举例】

[rkadk_record_test](#)

【相关主题】

[RKADK_PARAM_SetCamParam](#)

12.2.3 RKADK_PARAM_SetCamParam

【描述】

设置Camera相关的参数。

【语法】

RKADK_S32 RKADK_PARAM_SetCamParam(RKADK_S32 s32CamID, [RKADK_PARAM_TYPE_E](#) enParamType, const RKADK_VOID *pvParam);

【参数】

参数名称	描述	输入/输出
s32CamID	Camera id	输入
enParamType	参数类型	输入
pvParam	设置的参数指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_param.h

库文件：librkadk.so

【注意】

- 调用RKADK_PARAM_Init初始化参数模块后，才能调用该接口。

【举例】

[rkadk_record_test](#)

【相关主题】

[RKADK_PARAM_GetCamParam](#)

12.2.4 RKADK_PARAM_GetCommParam

【描述】

获取普通参数。

【语法】

RKADK_S32 RKADK_PARAM_GetCommParam([RKADK_PARAM_TYPE_E](#) enParamType, RKADK_VOID *pvParam);

【参数】

参数名称	描述	输入/输出
enParamType	参数类型	输入
pvParam	获取到的参数指针	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_param.h

库文件：librkadk.so

【注意】

- 调用RKADK_PARAM_Init初始化参数模块后，才能调用该接口。

【举例】

[rkadk_record_test](#)

【相关主题】

[RKADK_PARAM_SetCommParam](#)

12.2.5 RKADK_PARAM_SetCommParam

【描述】

设置普通参数。

【语法】

RKADK_S32 RKADK_PARAM_SetCommParam([RKADK_PARAM_TYPE_E](#) enParamType, const RKADK_VOID *pvParam);

【参数】

参数名称	描述	输入/输出
enParamType	参数类型	输入
pvParam	设置的参数指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_param.h

库文件：librkadk.so

【注意】

- 调用RKADK_PARAM_Init初始化参数模块后，才能调用该接口。

【举例】

[rkadk_record_test](#)

【相关主题】

[RKADK_PARAM_GetCommParam](#)

12.2.6 RKADK_PARAM_SetDefault

【描述】

恢复默认配置。

【语法】

RKADK_S32 RKADK_PARAM_SetDefault(RKADK_VOID);

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_param.h

库文件：librkadk.so

12.2.7 RKADK_PARAM_GetResolution

【描述】

RKADK_PARAM_RES_E 转换为具体分辨率。

【语法】

RKADK_S32 RKADK_PARAM_GetResolution([RKADK_PARAM_RES_E](#) type, RKADK_U32 *width, RKADK_U32 *height);

【参数】

参数名称	描述	输入/输出
type	分辨率类型	输入
width	转换的分辨率宽	输出
height	转换的分辨率高	输出

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk_param.h

库文件：librkadk.so

【注意】

- 调用RKADK_PARAM_Init初始化参数模块后，才能调用该接口。

【举例】

无

【相关主题】

[RKADK_PARAM_GetResType](#)

12.2.8 RKADK_PARAM_GetResType

【描述】

分辨率转换为RKADK_PARAM_RES_E。

【语法】

[RKADK_PARAM_RES_E](#) RKADK_PARAM_GetResType(RKADK_U32 width, RKADK_U32 height);

【参数】

参数名称	描述	输入/输出
width	分辨率宽	输入
height	分辨率高	输入

【返回值】

返回值	描述
对应的RKADK_PARAM_RES_E	成功
RKADK_RES_BUTT	失败

【需求】

头文件：rkadk_param.h

库文件：librkadk.so

【注意】

- 调用RKADK_PARAM_Init初始化参数模块后，才能调用该接口。

【举例】

无

【相关主题】

[RKADK_PARAM_GetResolution](#)

12.2.9 RKADK_PARAM_GetVencChnId

【描述】

获取Record、Photo、Stream 对应的VENC通道号。

【语法】

RKADK_S32 RKADK_PARAM_GetVencChnId(RKADK_U32 u32CamId, [RKADK_STREAM_TYPE_E](#) enStrmType);

【参数】

参数名称	描述	输入/输出
u32CamId	Camera id	输入
enStrmType	流类型	输入

【返回值】

返回值	描述
对应的VENC通道号	成功
-1	失败

【需求】

头文件：rkadk_param.h

库文件：librkadk.so

【注意】

- 调用RKADK_PARAM_Init初始化参数模块后，才能调用该接口。

【举例】

无

【相关主题】

无

12.3 数据类型

参数模块主要提供以下数据类型：

[RKADK_DEFPARAM_PATH](#)：默认全局ini配置文件路径

[RKADK_DEFPARAM_PATH_SENSOR_PREFIX](#)：默认Sensor ini配置文件路径

[RKADK_PARAM_PATH](#)：全局ini配置文件路径

[RKADK_PARAM_PATH_SENSOR_PREFIX](#)：Sensor ini配置文件路径

[RKADK_PARAM_TYPE_E](#)：参数类型枚举

[RKADK_PARAM_RES_E](#)：分辨率类型枚举

[RKADK_STREAM_TYPE_E](#)：数据流类型枚举

[RKADK_PARAM_CODEC_CFG_S](#)：编码类型配置结构体

[RKADK_PARAM_BITRATE_S](#)：比特率配置结构体

[RKADK_PARAM_REC_TIME_S](#)：录像时长配置结构体

[RKADK_PARAM_GOP_S](#)：VENC GOP配置结构体

[RKADK_VQE_MODE_E](#)：音频输入声音质量增强枚举

[RKADK_MUXER_FILE_TYPE_E](#)：录像文件类型枚举

[RKADK_MUXER_PRE_RECORD_MODE_E](#)：预录模式枚举

[RKADK_MIC_TYPE_E](#)：音频设备声道模式类型枚举

12.3.1 RKADK_DEFPARAM_PATH

【说明】

默认全局ini配置文件路径，存放各Sensor共用的配置，用于恢复默认配置。

【定义】

```
#define RKADK_DEFPARAM_PATH "/oem/usr/etc/rkadk_defsetting.ini"
```

12.3.2 RKADK_DEFPARAM_PATH_SENSOR_PREFIX

【说明】

默认Sensor ini配置文件路径，存放各Sensor独有的配置，用于恢复默认配置。

【定义】

```
#define RKADK_DEFPARAM_PATH_SENSOR_PREFIX "/oem/usr/etc/rkadk_defsetting_sensor"
```

【注意】

- Sensor配置文件前缀默认rkadk_defsetting_sensor_n.ini，_n 为Sensor Camera Id，序号从0开始。

12.3.3 RKADK_PARAM_PATH

【说明】

全局ini配置文件路径，存放各Sensor共用的配置。

【定义】

```
#define RKADK_PARAM_PATH "/data/rkadk/rkadk_setting.ini"
```

12.3.4 RKADK_PARAM_PATH_SENSOR_PREFIX

【说明】

Sensor ini配置文件路径，存放各Sensor独有的配置。用于保存切换录像分辨率、Codec类型等操作时的新配置。

【定义】

```
#define RKADK_PARAM_PATH_SENSOR_PREFIX "/data/rkadk/rkadk_setting_sensor"
```

【注意】

- Sensor 配置文件前缀默认rkadk_setting_sensor_n.ini，_n 为Sensor Camera Id，序号从0开始。

12.3.5 RKADK_PARAM_TYPE_E

【说明】

定义参数类型枚举类型。

【定义】

```
typedef enum {  
    /* Cam Dependent Param */  
    RKADK_PARAM_TYPE_FPS,      /* framerate */  
    RKADK_PARAM_TYPE_GOP,      /* gop */  
    RKADK_PARAM_TYPE_RES,      /* specify RKADK_PARAM_RES_E(record) */  
    RKADK_PARAM_TYPE_PHOTO_RES, /* specify RKADK_PARAM_RES_E(photo) */  
    RKADK_PARAM_TYPE_CODEC_TYPE, /* specify RKADK_PARAM_CODEC_CFG_S(record) */  
    RKADK_PARAM_TYPE_BITRATE,   /* encode bitrate,specify RKADK_PARAM_BITRATE_S */  
    RKADK_PARAM_TYPE_FLIP,      /* bool */  
    RKADK_PARAM_TYPE_MIRROR,    /* bool */  
    RKADK_PARAM_TYPE_LDC,       /* ldc level [0,255] */  
    RKADK_PARAM_TYPE_ANTIFOG,    /* antifog value, [0,10] */  
    RKADK_PARAM_TYPE_WDR,       /* wdr level, [0,10] */  
    RKADK_PARAM_TYPE_HDR,       /* 0: normal, 1: HDR2, 2: HDR3, [0,2] */  
    RKADK_PARAM_TYPE_RECORD_TYPE, /* specify RKADK_REC_TYPE_E */  
    RKADK_PARAM_TYPE_RECORD_TIME, /* specify RKADK_PARAM_REC_TIME_S, record time(s) */  
    RKADK_PARAM_TYPE_PRE_RECORD_TIME, /* pre record time, unit in second(s) */  
    RKADK_PARAM_TYPE_PRE_RECORD_MODE, /* pre record mode, specify MUXER_PRE_RECORD_MODE_E */  
    RKADK_PARAM_TYPE_SPLITTIME, /* specify RKADK_PARAM_REC_TIME_S, manual splite time(s) */  
    RKADK_PARAM_TYPE_FILE_CNT, /* record file count, maximum RECORD_FILE_NUM_MAX */  
}
```



```
RKADK_PARAM_TYPE_LAPSE_INTERVAL, /* specify RKADK_PARAM_REC_TIME_S, lapse interval(s) */
RKADK_PARAM_TYPE_LAPSE_MULTIPLE, /* lapse multiple */
RKADK_PARAM_TYPE_JPEG_SLICE, /* enable/disable JPEG slice */
RKADK_PARAM_TYPE_SLICE_HEIGHT, /* set JPEG slice height */

/* COMM Dependent Param */
RKADK_PARAM_TYPE_REC_MUTE, /* record audio mute, bool */
RKADK_PARAM_TYPE_VOLUME, /* speaker volume, [0,100] */
RKADK_PARAM_TYPE_MIC_VOLUME, /* mic volume, [0,100] */
RKADK_PARAM_TYPE_BUTT
} RKADK_PARAM_TYPE_E;
```

【成员】

成员名称	描述
RKADK_PARAM_TYPE_FPS	帧率
RKADK_PARAM_TYPE_GOP	I帧间隔, RKADK_PARAM_GOP_S
RKADK_PARAM_TYPE_RES	录像分辨率, RKADK_PARAM_RES_E
RKADK_PARAM_TYPE_PHOTO_RES	拍照分辨率, RKADK_PARAM_RES_E
RKADK_PARAM_TYPE_CODEC_TYPE	录像编码类型, RKADK_PARAM_CODEC_CFG_S
RKADK_PARAM_TYPE_BITRATE	比特率, RKADK_PARAM_BITRATE_S
RKADK_PARAM_TYPE_FLIP	上下翻转
RKADK_PARAM_TYPE_MIRROR	左右镜像
RKADK_PARAM_TYPE_LDC	畸变校正[0,255]
RKADK_PARAM_TYPE_ANTIFOG	去雾[0,10]
RKADK_PARAM_TYPE_WDR	宽动态[0,10]
RKADK_PARAM_TYPE_HDR	高动态范围成像[0,10]
RKADK_PARAM_TYPE_RECORD_TYPE	录像类型, RKADK_REC_TYPE_E
RKADK_PARAM_TYPE_RECORD_TIME	录像时长, 录像主码流、子码流支持设置不同时长, RKADK_PARAM_REC_TIME_S
RKADK_PARAM_TYPE_PRE_RECORD_TIME	预录时长
RKADK_PARAM_TYPE_PRE_RECORD_MODE	预录模式, 0: 不预录, 1: 手动切分预录, 2: 首文件预录, 3: 所有文件预录
RKADK_PARAM_TYPE_SPLITTIME	手动切分录像时长, 录像主码流、子码流支持设置不同时长, RKADK_PARAM_REC_TIME_S
RKADK_PARAM_TYPE_FILE_CNT	同时录制文件个数, 最大2
RKADK_PARAM_TYPE_LAPSE_INTERVAL	缩时录像时长, 录像主码流、子码流支持设置不同时长, RKADK_PARAM_REC_TIME_S
RKADK_PARAM_TYPE_LAPSE_MULTIPLE	缩时录像文件播放时长与实际画面内容时间的倍数关系
RKADK_PARAM_TYPE_JPEG_SLICE	是否使能JPEG Slice
RKADK_PARAM_TYPE_SLICE_HEIGHT	JPEG Slice高
RKADK_PARAM_TYPE_REC_MUTE	是否使能录像静音
RKADK_PARAM_TYPE_VOLUME	Speaker音量[0,100]
RKADK_PARAM_TYPE_MIC_VOLUME	麦克风音量[0,100]

【注意】

- Antifog、WDR、HDR等Camere 硬件相关设置，除了调用RKADK_PARAM_SetCamParam设置ini之外，还需调用ISP对应接口使之实际生效。
- RKADK_PARAM_TYPE_LAPSE_MULTIPLE：缩时录像文件播放时长与实际画面内容时间的倍数关系，跟帧率有关，比如普通录像帧率是30fps，缩时录影是1fps，则倍数是30。
- RV1126/RV1109切换分辨率时，当Photo分辨率未设置为Sensor最大支持分辨率时，需和Record主码流分辨率保持一致。
- RKADK_PARAM_TYPE_REC_MUTE：和RKADK_RECORD_GetAencChn、RK_MPI_AENC_SetMute搭配使用。

【相关数据类型及接口】

[RKADK_PARAM_GetCamParam](#)

[RKADK_PARAM_SetCamParam](#)

[RKADK_PARAM_GetCommParam](#)

[RKADK_PARAM_SetCommParam](#)

12.3.6 RKADK_PARAM_RES_E

【说明】

定义播放事件枚举类型。

【定义】

```
typedef enum {
    RKADK_RES_720P = 0, /* 1280*720 */
    RKADK_RES_1080P, /* 1920*1080 */
    RKADK_RES_1296P, /* 2304*1296 */
    RKADK_RES_1440P, /* 2560*1440 */
    RKADK_RES_1520P, /* 2688*1520 */
    RKADK_RES_1600P, /* 2560*1600 */
    RKADK_RES_1620P, /* 2880*1616, height 8 alignment */
    RKADK_RES_1944P, /* 2592*1944 */
    RKADK_RES_2160P, /* 3840*2160 */
    RKADK_RES_BUTT,
} RKADK_PARAM_RES_E;
```

【相关数据类型及接口】

[RKADK_PARAM_GetResolution](#)

[RKADK_PARAM_GetResType](#)

12.3.7 RKADK_STREAM_TYPE_E

【说明】

定义流枚举类型。

【定义】

```
typedef enum {
    RKADK_STREAM_TYPE_SENSOR,
    RKADK_STREAM_TYPE_VIDEO_MAIN,
    RKADK_STREAM_TYPE_VIDEO_SUB,
    RKADK_STREAM_TYPE_SNAP,
    RKADK_STREAM_TYPE_PREVIEW,
    RKADK_STREAM_TYPE_LIVE,
    RKADK_STREAM_TYPE_DISP,
    RKADK_STREAM_TYPE_BUTT
} RKADK_STREAM_TYPE_E;
```

【成员】

成员名称	描述
RKADK_STREAM_TYPE_VIDEO_MAIN	录像主码流
RKADK_STREAM_TYPE_VIDEO_SUB	录像子码流
RKADK_STREAM_TYPE_SNAP	拍照
RKADK_STREAM_TYPE_PREVIEW	远程预览
RKADK_STREAM_TYPE_LIVE	直播
RKADK_STREAM_TYPE_DISP	本地预览
RKADK_STREAM_TYPE_SENSOR	Sensor

【相关数据类型及接口】

[RKADK_PARAM_GetVencChnId](#)

12.3.8 RKADK_PARAM_CODEC_CFG_S

【说明】

定义编码类型配置结构体。

【定义】

```
typedef struct {
    RKADK_STREAM_TYPE_E enStreamType;
    RKADK_CODEC_TYPE_E enCodecType;
} RKADK_PARAM_CODEC_CFG_S;
```

【成员】

成员名称	描述
enStreamType	流类型
enCodecType	编码类型

【相关数据类型及接口】

[RKADK_CODEC_TYPE_E](#)

[RKADK_STREAM_TYPE_E](#)

12.3.9 RKADK_PARAM_BITRATE_S

【说明】

定义比特率配置结构体。

【定义】

```
typedef struct {  
    RKADK_STREAM_TYPE_E enStreamType;  
    RKADK_U32 u32Bitrate;  
} RKADK_PARAM_BITRATE_S;
```

【成员】

成员名称	描述
enStreamType	流类型
u32Bitrate	比特率

【相关数据类型及接口】

[RKADK_STREAM_TYPE_E](#)

12.3.10 RKADK_PARAM_REC_TIME_S

【说明】

定义录像时长配置结构体。

【定义】

```
typedef struct {  
    RKADK_STREAM_TYPE_E enStreamType;  
    RKADK_U32 time;  
} RKADK_PARAM_REC_TIME_S;
```

【成员】

成员名称	描述
enStreamType	流类型
time	录像时长

【相关数据类型及接口】

[RKADK_STREAM_TYPE_E](#)

12.3.11 RKADK_PARAM_GOP_S

【说明】

定义VENC I帧间隔配置结构体。

【定义】

```
typedef struct {
    RKADK_STREAM_TYPE_E enStreamType;
    RKADK_U32 u32Gop;
} RKADK_PARAM_GOP_S;
```

【成员】

成员名称	描述
enStreamType	流类型
u32Gop	I帧间隔

【相关数据类型及接口】

[RKADK_STREAM_TYPE_E](#)

12.3.12 RKADK_VQE_MODE_E

【说明】

定义音频输入声音质量增强枚举类型

【定义】

```
typedef enum {
    RKADK_VQE_MODE_AI_TALK = 0,
    RKADK_VQE_MODE_AI_RECORD,
    RKADK_VQE_MODE_BUTT
} RKADK_VQE_MODE_E;
```

【成员】

成员名称	描述
RKADK_VQE_MODE_AI_TALK	使能AEC、ANR、AGC
RKADK_VQE_MODE_AI_RECORD	使能ANR

12.3.13 RKADK_MUXER_FILE_TYPE_E

【说明】

定义录像文件类型枚举

【定义】

```
typedef enum rkMUXER_TYPE_E {  
    RKADK_MUXER_TYPE_MP4 = 0,  
    RKADK_MUXER_TYPE_MPEGTS,  
    RKADK_MUXER_TYPE_FLV,  
    RKADK_MUXER_TYPE_BUTT  
} RKADK_MUXER_FILE_TYPE_E;
```

【成员】

成员名称	描述
RKADK_MUXER_TYPE_MP4	MP4
RKADK_MUXER_TYPE_MPEGTS	Reserved
RKADK_MUXER_TYPE_FLV	FLV

12.3.14 RKADK_MUXER_PRE_RECORD_MODE_E

【说明】

定义预录模式枚举

【定义】

```
typedef enum {  
    RKADK_MUXER_PRE_RECORD_NONE = 0,  
    RKADK_MUXER_PRE_RECORD_MANUAL_SPLIT, /* manual split file prerecord */  
    RKADK_MUXER_PRE_RECORD_SINGLE /* first file prerecord */  
} RKADK_MUXER_PRE_RECORD_MODE_E;
```

【成员】

成员名称	描述
RKADK_MUXER_PRE_RECORD_NONE	不预录
RKADK_MUXER_PRE_RECORD_MANUAL_SPLIT	手动切分文件预录
RKADK_MUXER_PRE_RECORD_SINGLE	第一个文件预录

12.3.15 RKADK_MIC_TYPE_E

【说明】

定义音频设备声道模式类型

【定义】

```
typedef enum {
    RKADK_MIC_TYPE_LEFT = 0,
    RKADK_MIC_TYPE_RIGHT,
    RKADK_MIC_TYPE_BOTH,
    RKADK_MIC_TYPE_BUTT
} RKADK_MIC_TYPE_E;
```

【成员】

成员名称	描述
RKADK_MIC_TYPE_LEFT	左声道声音
RKADK_MIC_TYPE_RIGHT	右声道声音
RKADK_MIC_TYPE_BOTH	双声道

12.4 INI文件解析

12.4.1 全局INI配置文件

```
[version]
version          = 2.2.0    /* version */

/* 普通参数 */
[common]
sensor_count     = 2        /* Sensor 个数 */
rec_mute         = FALSE    /* 是否使能录像静音 */
speaker_volume   = 80       /* Speaker音量, [0,100] */
mic_volume       = 80       /* 麦克风音量, [0,100] */
vpss_devcie      = 1        /* VPSS硬件设备类型, 0:GPU, 1:RGA */

/* Audio 参数 */
[audio]
ai_audio_node    = hw:0,0   /* Ai 设备节点 */
ao_audio_node    = hw:0,0   /* Ao 设备节点 */
ai_depth         = 1        /* Ai depth 深度 */
bit_width        = 1        /* 采样精度 */
channels         = 1        /* 通道数 */
mic_type         = 0        /* 音频设备声道模式, 特指RKADK_MIC_TYPE_E, 0:左声道, 1:右声道, 2:双声道 */
samplerate       = 16000    /* 采样率 */
samples_per_frame = 576     /* 每帧采样个数 */
bitrate          = 64000    /* 比特率 */
vqe_mode         = 1        /* 配置音频输入声音质量增强, 特指RKADK_VQE_MODE_E */
vqe_config_path  = /oem/usr/share/vqefiles/config_aivqe.json /* vqe 配置文件路径 */
codec_type       = 8        /* Record和Live Audio编码类型, 默认适配MP3, 特指RKADK_CODEC_TYPE_E */
```


12.4.2 Sensor INI配置文件

```
[sensor]
used_isp          = TRUE    /* Sensor是否有经过ISP */
max_width         = 2688    /* 最大分辨率宽 */
max_height        = 1520    /* 最大分辨率高 */
framerate         = 30      /* 帧率 */
flip              = FALSE   /* 上下翻转 */
mirror            = FALSE   /* 左右镜像 */
ldc               = 0       /* 畸变校正, [0,255] */
wdr               = 0       /* 宽动态, [0,10] */
hdr               = 0       /* 高动态范围成像, [0,10] */
antifog           = 0       /* 去雾, [0,10] */
enable_wrap       = FALSE   /* VI是否使能卷绕 */
wrap_buf_line     = 1620    /* 卷绕buffer行高 */

/* VI通道配置参数 */
[vi.0]
chn_id            = 0        /* 通道号 */
device_name       = rkisp_m_bypass /* Video 节点路径 */
buf_cnt           = 4        /* 输出通道总的缓存块数 */
depth             = 0        /* Vi depth 深度 */
width             = 2688     /* Video宽 */
height            = 1520     /* Video高 */
pix_fmt           = FBC0     /* VI输出格式 */
module            = RECORD_MAIN|PHOTO /* 该VI的使用模块, Options:
NONE/RECORD_MAIN/RECORD_SUB/PREVIEW/PHOTO/LIVE/DISP */

[vi.1]
chn_id            = 1
device_name       = rkisp_scale0
buf_cnt           = 4
depth             = 0
width             = 0
height            = 0
pix_fmt           = NV12
module            = RECORD_MAIN|PHOTO

[vi.2]
chn_id            = 2
device_name       = rkisp_scale1
buf_cnt           = 2
depth             = 0
width             = 0
height            = 0
pix_fmt           = NV12
module            = NONE

[vi.3]
chn_id            = 3
device_name       = rkisp_scale2
buf_cnt           = 4
depth             = 0
width             = 848
height            = 480
pix_fmt           = NV12
```

```

module                = RECORD_SUB|PREVIEW|LIVE|DISP

/* Record 参数 */
[record]
record_type          = 0      /* 录像类型, 特指RKADK_REC_TYPE_E */
file_type            = 0      /* 录像文件类型, 特指RKADK_MUXER_FILE_TYPE_E */
pre_record_time      = 0      /* 预录时长 */
pre_record_mode      = 0      /* 预录模式 */
lapse_multiple       = 30     /* 缩时录像文件播放时长与实际画面内容时间的倍数关系 */
file_num             = 1      /* 同时录制文件个数, 最大2 */
switch_res           = TRUE   /* 是否切换分辨率 */
enable_audio         = TRUE   /* Record是否开启录音 */

/* 主码流 Record 0 VENC 参数 */
[record.0]
record_time          = 60     /* 录像时长 */
splite_time          = 60     /* 手动切分录像时长 */
lapse_interval       = 60     /* 缩时录像时长 */
width                = 2688   /* Video 宽 */
height               = 1520   /* Video 高 */
bufsize              = 10379776 /* 码流buffer大小 */
framerate            = 30     /* Venc 帧率 */
bitrate              = 8294400 /* 比特率 */
gop                  = 30     /* I 帧间隔 */
profile              = 100    /* 编码器profile */
codec_type           = 0      /* 编码类型, 特指RKADK_CODEC_TYPE_E */
venc_chn             = 0      /* Venc通道号 */
vpss_grp             = 0      /* VPSS GROUP号 */
vpss_chn             = 0      /* VPSS 通道号 */
post_aiisp           = FALSE  /* 是否使能Post AI ISP */
rc_mode              = CBR    /* 编码协议类型, 支持CBR、VBR、AVBR */
max_qp               = -1     /* QP最大值, 取值范围[1, 51], -1:使用默认值 */
min_qp               = -1     /* QP最小值, 取值范围[1, max_qp], -1:使用默认值 */
i_min_qp             = -1
i_frame_min_qp       = -1
full_range            = TRUE
scaling_list          = FALSE
hier_qp_en           = FALSE
hier_qp_delta        = -3,0,0,0
hier_frame_num       = 3,0,0,0

/* 子码流 Record 1 VENC 参数, 当 file_num = 1 时, 不需要配置rec.1 */
[record.1]
record_time          = 60
splite_time          = 60
lapse_interval       = 60
width                = 848
height               = 480
bufsize              = 2367488
bitrate              = 407040
framerate            = 30
gop                  = 30
profile              = 100
codec_type           = 0
venc_chn             = 1
vpss_grp             = 0      /* VPSS GROUP号 */
vpss_chn             = 0      /* VPSS 通道号 */
post_aiisp           = FALSE  /* 是否使能Post AI ISP */

```

```
rc_mode          = VBR
max_qp           = 48
min_qp           = 8
i_min_qp         = -1
i_frame_min_qp   = -1
full_range       = TRUE
scaling_list     = TRUE
hier_qp_en       = TRUE
hier_qp_delta    = -3,0,0,0
hier_frame_num   = 3,0,0,0
```

/ Photo VENC 参数 */*

```
[photo]
image_width      = 3840 /* 照片宽度 */
image_height     = 2160 /* 照片高度 */
venc_chn         = 2    /* Venc通道号 */
vpss_grp         = 0    /* VPSS GROUP号 */
vpss_chn         = 0    /* VPSS 通道号 */
post_aiisp       = FALSE /* 是否使能Post AI ISP */
enable_combo     = FALSE /* 使能编码通道Combo属性 */
combo_venc_chn   = 0    /* Combo的数据源通道 */
qfactor          = 50    /* 具体含义请参见RFC2435协议，系统默认为70,取值范围[1, 99] */
switch_res       = TRUE  /* 是否切换分辨率 */
jpeg_slice       = FALSE /* 是否使能JPEG Slice */
slice_height     = 0     /* JPEG Slice高，不得大于max_slice_height */
max_slice_width  = 0     /* JPEG Slice最大宽 */
max_slice_height = 0     /* JPEG Slice最大高 */
```

/ 远程预览 VENC 参数 */*

```
[preview]
width            = 848   /* Video 宽 */
height           = 480   /* Video 高 */
bufsize         = 2367488
bitrate          = 407040 /* 比特率 */
framerate        = 30    /* Venc 帧率 */
gop              = 30    /* I 帧间隔 */
profile          = 100   /* 编码器profile */
codec_type       = 0     /* 编码类型，特指RKADK_CODEC_TYPE_E */
venc_chn         = 1     /* Venc通道号 */
vpss_grp         = 0     /* VPSS GROUP号 */
vpss_chn         = 0     /* VPSS 通道号 */
post_aiisp       = FALSE /* 是否使能Post AI ISP */
rc_mode          = VBR   /* 编码协议类型, 支持CBR、VBR、AVBR */
max_qp           = 48    /* QP最大值，取值范围[1, 51] */
min_qp           = 8     /* QP最小值，取值范围[1, min_qp] */
i_min_qp         = -1
i_frame_min_qp   = -1
full_range       = TRUE
scaling_list     = TRUE
hier_qp_en       = TRUE
hier_qp_delta    = -3,0,0,0
hier_frame_num   = 3,0,0,0
```

/ 直播 VENC 参数 */*

```
[live]
width            = 1280
height           = 720
bufsize         = 2367488
```

```
bitrate          = 4194304
framerate        = 30
gop              = 30
profile          = 100
codec_type       = 0
venc_chn         = 1
vpss_grp         = 0
vpss_chn         = 0
post_aiisp       = FALSE
rc_mode          = VBR
max_qp           = 48
min_qp           = 8
i_min_qp         = -1
i_frame_min_qp   = -1
full_range       = TRUE
scaling_list     = TRUE
hier_qp_en       = TRUE
hier_qp_delta    = -3,0,0,0
hier_frame_num   = 3,0,0,0
```

```
/* 本地预览参数 */
```

```
[display]
```

```
x              = 0          /* 显示X坐标 */
y              = 0          /* 显示Y坐标 */
width          = 720        /* 显示宽 */
height         = 1280       /* 显示高 */
rotation       = 90         /* 旋转度数, Options: 0:0, 1:90, 2:180, 3:270 */
vpss_grp       = 0          /* VPSS GROUP号 */
vpss_chn       = 0          /* VPSS 通道号 */
img_type       = RGB888     /* 视频输出格式 */
vo_device      = 0          /* 显示输出设备号 */
vo_layer       = 0          /* 视频输出视频层号 */
vo_chn         = 0          /* VO通道号 */
frame_rate     = 30         /* VO刷新帧率 */
intf_type      = default    /* 显示接口类型, Options: MIPI, HDMI, EDP, VGA, DP, HDMI_EDP, LCD,
default*/
splice_mode    = RGA        /* 图层合成方式, Options: RGA, GPU, BYPASS */
```

12.4.3 INI 配置注意事项

- 通过环境变量配置默认INI 文件路径，例如：export rkadk_default_ini_path=/oem/usr/etc。
- 通过RKADK_PARAM_Init API设置INI文件路径。
- rkadk_defsetting.ini 和 rkadk_setting.ini 中的version 必须保持一致，否则会版本检测失败，使用默认INI 配置。SDK更新时，ini 参数可能会有增减，此时需要注意。
- sensor_count 代表实际使用的Sensor个数，根据实际使用设置，不能大于RKADK_MAX_SENSOR_CNT，目前实际调试过3 Sensor。
- 如果Sensor经过ISP，used_isp必须配置成TRUE，直接通过配置ISP节点分辨率实现缩放，不需要配置VPSS通道，vpss_grp和vpss_chn统一配置成0；如果Sensor不经过ISP，used_isp必须配置成FALSE，并配置对应的vpss_grp和vpss_chn，内部使用VPSS进行缩放。
- 当录像、远程预览、直播等分辨率一样时，建议复用VENC（VENC 参数配置成一样），提高带宽和内存利用率。
- 当VENC复用或者VENC分辨率相同时，建议复用VPSS通道，提高带宽利用率。
- gop建议和framerate配置成一样，确保每秒都有一个I帧。

13. 示例

以下提供功能示例，使用注意事项如下：

- 运行示例前需保证无其他应用占用示例所用节点，如mediaserver、ispserver。
- 示例默认参数适配我司EVB，硬件不同时，示例可能需要显式指定参数或调整代码。

13.1 rkadk_record_test

【说明】

Record 测试。

【代码路径】

rkadk/examples/rkadk_record_test.c

【快速使用】

```
./rkadk_record_test
```

【选项】

选项	描述	默认值
-a	内置ISP功能启用，输入该选项启用内置ISP功能，无参数则使用默认值，参数为aiq文件所在文件夹路径。	/etc/iqfiles
-l	Camera id	0
-m	使能双Sensor测试，options: 0(isp sensor), 1(all isp sensors), 2(isp+ahd sensors)	0
-p	ini配置文件目录路径	/data/rkadm
-k	录像文件是否I帧切片	不切片

13.2 rkadm_photo_test

【说明】

Photo测试。

【代码路径】

/rkadm/examples/rkadm_photo_test.c

【快速使用】

```
./rkadm_photo_test
```

【选项】

选项	描述	默认值
-a	内置ISP功能启用，输入该选项启用内置ISP功能，无参数则使用默认值，参数为aiq文件所在文件夹路径。	/etc/iqfiles
-l	Camera id	0
-p	ini配置文件目录路径	/data/rkadm
-t	获取的JPG图片的数据类型	NV12
-m	使能双Sensor测试，options: 0(isp sensor), 1(all isp sensors), 2(isp+ahd sensors)	0
-o	osd文件路径	NULL
-W	osd宽	0
-H	osd高	0

13.3 rkadk_stream_test

【说明】

获取音频流并编码，输出到文件；获取视频流并编码，输出到文件。

【代码路径】

/rkadk/examples/rkadk_stream_test.c

【快速使用】

```
./rkadk_stream_test
```

【选项】

选项	描述	默认值
-a	内置ISP功能启用，输入该选项启用内置ISP功能，无参数则使用默认值，参数为aiq文件所在文件夹路径。	/etc/iqfiles
-l	Camera id	0
-M	测试模式：audio、video	audio
-e	编码类型	pcm
-o	输出文件路径	/tmp/ai.pcm
-p	ini配置文件目录路径	/data/rkadk
-m	使能双Sensor测试，options: 0(isp sensor), 1(all isp sensors), 2(isp+ahd sensors)	0

13.4 rkadk_player_test

【说明】

本地文件播放测试。

【代码路径】

/rkadk/examples/rkadk_player_test.c

【快速使用】

```
./rkadk_player_test
```

【选项】

选项	描述	默认值
-i	播放文件路径	/etc/bsa_file/8k8bpsMono.wav
-x	Video 显示起始 x 坐标	0
-y	Video 显示起始 y 坐标	0
-W	Video 显示宽度	屏幕物理宽度
-H	Video 显示高度	屏幕物理高度
-r	Video旋转角度，option: 0, 90, 180, 270	0
-p	ini配置文件目录路径	/data/rkadk
-m	Video镜像	disbale
-f	Video翻转	disbale
-a	是否使能audio播放，option: 0(disable), 1(enable)	1
-v	是否使能Video播放	disbale
-s	设置图层合成方式，option: 0(RGA), 1(GPU), 2(ByPass)	0
-P	屏显示像素格式，option: 0(RGB888), 1(NV12), 2(RGB565)	0
-l	显示接口类型，option: 0(DEFAULTT), 1(MIPI), 2(LCD)	1106: 0，其他平台：1
-F	刷新帧率	30
-t	rtsp传输协议，option: 0(udp), 1(tcp)	0
-b	使能播放完黑屏	disable
-T	rtsp socket I/O 操作超时退出时长，单位ms	阻塞，不超时
-l	Vo layer id	0
-O	Vdec 输出buffer个数	3 [1, 8]
-D	使能第三方demuxer库	disable

【注意】

- 播放视频文件时，需要-v 使能Video播放。
- 镜像/翻转不能和旋转同时设置。

13.5 rkadk_thumb_test

获取文件缩略图测试。

【代码路径】

/rkadk/examples/rkadk_thumb_test.c

【快速使用】

获取MP4缩略图：./rkadk_thumb_test -i test_file.mp4
获取JPG缩略图：./rkadk_thumb_test -i test_file.jpg -f jpg -t MPF1

【选项】

选项	描述	默认值
-i	测试文件路径	无
-f	文件格式：mp4、jpg	mp4
-t	JPG缩略图类型：DCF, MPF1, MPF2	DCF
-T	输出缩略图类型：JPG, NV12, RGB565, RGB888	JPG
-W	缩略图宽	从ini获取
-H	缩略图高	从ini获取

13.6 rkadk_rtsp_test

RTSP直播测试。

【代码路径】

/rkadk/examples/rkadk_rtsp_test.c

【快速使用】

- 1、板端启动wifi
- 2、板端运行./rkadk_rtsp_test
- 3、PC端打开VLC -> 媒体 -> 打开网络串流 -> 输入网络URL：rtsp://板端ip地址/live/main_stream

【选项】

选项	描述	默认值
-a	内置ISP功能启用，输入该选项启用内置ISP功能，无参数则使用默认值，参数为aiq文件所在文件夹路径。	/etc/iqfiles
-l	Camera id	0
-p	ini配置文件目录路径	/data/rkadc
-o	osd文件路径	/userdata/rkadc_ARGB8888

13.7 rkadc_rtmp_test

RTMP直播测试。

【代码路径】

/rkadc/examples/rkadc_rtmp_test.c

【快速使用】

- 1、板端启动wifi
- 2、板端运行./rkadc_rtmp_test
- 3、PC端打开VLC -> 媒体 -> 打开网络串流 -> 输入网络URL：rtmp://板端ip地址::1935/live/substream

【选项】

选项	描述	默认值
-a	内置ISP功能启用，输入该选项启用内置ISP功能，无参数则使用默认值，参数为aiq文件所在文件夹路径。	/etc/iqfiles
-l	Camera id	0
-p	ini配置文件目录路径	/data/rkadc

13.8 rkadc_storage_test

【说明】

存储模块测试。

【代码路径】

/rkadc/examples/rkadc_storage_test.c

【快速使用】

./rkadc_storage_test

【选项】

无

【注意】

- 此test生成了全写入0的mp4文件，对自动删除、获取文件列表等功能和接口进行测试，该mp4文件没有实际数据，无法播放。

13.9 rkadk_disp_test

本地预览测试。

【代码路径】

/rkadk/examples/rkadk_disp_test.c

【快速使用】

```
./rkadk_disp_test
```

【选项】

选项	描述	默认值
-a	内置ISP功能启用，输入该选项启用内置ISP功能，无参数则使用默认值，参数为aiq文件所在文件夹路径。	/etc/iqfiles
-l	Camera id	0
-p	ini配置文件目录路径	/data/rkadk

13.10 rkadk_ui_test

UI叠加测试。

【代码路径】

/rkadk/examples/rkadk_ui_test.c

【快速使用】

```
./rkadk_ui_test
```

【选项】

选项	描述	默认值
-a	内置ISP功能启用，输入该选项启用内置ISP功能，无参数则使用默认值，参数为aiq文件所在文件夹路径。	/etc/iqfiles
-l	Camera id	0
-p	ini配置文件目录路径	/data/rkadk
-W	显示区域宽度	720
-H	显示区域高度	1280
-f	屏显示像素格式，option: 0(RGB888), 1(NV12), 2(RGB565), 3(RGB444)	0