

# Rockchip RKADK Development Guide

文件标识: RK-KF-YF-904

发布版本: V1.3.2

日期: 2022-06-24

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 © 2021 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](http://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](mailto:fae@rock-chips.com)

## 前言

### 概述

本文主要描述了Rkadk 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) <a href="#">RKADK_RECORD_GetAencChn</a> (2) <a href="#">RKADK_GetThmInMp4</a> (3) <a href="#">RKADK_PHOTO_GetThmInJpg</a> (4) <a href="#">RKADK_PARAM_GetVencChnId</a> 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) <a href="#">RKADK_GetThmInMp4Ex</a> (2) <a href="#">RKADK_ThmBufFree</a> (3) <a href="#">RKADK_PHOTO_GetThmInJpgEx</a> (4) <a href="#">RKADK_PHOTO_ThumbBufFree</a> 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) <a href="#">RKADK_PHOTO_GetData</a> (2) <a href="#">RKADK_PHOTO_FreeData</a> (3) <a href="#">RKADK_PLAYER_GetDuration</a> 2、新增INI配置注意事项
V1.3.12	ZQX	2022-08-24	1、增加API说明： (1) <a href="#">RKADK_OSD_Init</a> (2) <a href="#">RKADK_OSD_Deinit</a> (3) <a href="#">RKADK_OSD_Update</a> (4) <a href="#">RKADK_OSD_AttachToStream</a> (5) <a href="#">RKADK_OSD_DettachFromStream</a>

## 目录

### Rockchip RKADK Development Guide

系统概述

录像

概述

API参考

[RKADK\\_RECORD\\_Create](#)

[RKADK\\_RECORD\\_Destroy](#)

RKADK\_RECORD\_Start  
RKADK\_RECORD\_Stop  
RKADK\_RECORD\_ManualSplit  
RKADK\_RECORD\_RegisterEventCallback  
RKADK\_RECORD\_GetAencChn  
RKADK\_GetThmInMp4  
RKADK\_GetThmInMp4Ex  
RKADK\_ThmBufFree

## 数据类型

### 公共数据类型

RKADK\_MW\_PTR  
RKADK\_MAX\_SENSOR\_CNT  
RECORD\_FILE\_NUM\_MAX  
MUXER\_EVENT\_E  
MUXER\_FILE\_EVENT\_INFO\_S  
RKADK\_REC\_EVENT\_INFO\_S  
RKADK\_REC\_EVENT\_CALLBACK\_FN  
RKADK\_REC\_TYPE\_E  
RKADK\_REC\_REQUEST\_FILE\_NAMES\_FN  
RKADK\_RECORD\_ATTR\_S  
MUXER\_MANUAL\_SPLIT\_TYPE\_E  
MUXER\_PRE\_MANUAL\_SPLIT\_ATTR\_S  
RKADK\_REC\_MANUAL\_SPLIT\_ATTR\_S

## 拍照

### 概述

### API 参考

RKADK\_PHOTO\_Init  
RKADK\_PHOTO\_DeInit  
RKADK\_PHOTO\_TakePhoto  
RKADK\_PHOTO\_GetThmInJpg  
RKADK\_PHOTO\_GetThmInJpgEx  
RKADK\_PHOTO\_ThumbBufFree  
RKADK\_PHOTO\_GetData  
RKADK\_PHOTO\_FreeData

## 数据类型

RKADK\_PHOTO\_TYPE\_E  
RKADK\_PHOTO\_SINGLE\_ATTR\_S  
RKADK\_PHOTO\_MULTIPLE\_ATTR\_S  
RKADK\_PHOTO\_THUMB\_ATTR\_S  
RKADK\_PHOTO\_DATA\_RECV\_FN\_PTR  
RKADK\_PHOTO\_ATTR\_S  
RKADK\_JPG\_THUMB\_TYPE\_E  
RKADK\_THUMB\_TYPE\_E  
RKADK\_THUMB\_ATTR\_S  
RKADK\_PHOTO\_DATA\_ATTR\_S

## 远程预览

### 概述

### API 参考

RKADK\_STREAM\_VideoInit  
RKADK\_STREAM\_VideoDeInit  
RKADK\_STREAM\_VencStart  
RKADK\_STREAM\_VencStop  
RKADK\_STREAM\_GetVideoInfo  
RKADK\_STREAM\_VencRegisterCallback  
RKADK\_STREAM\_VencUnRegisterCallback  
RKADK\_STREAM\_AudioInit  
RKADK\_STREAM\_AudioDeInit  
RKADK\_STREAM\_AencStart

[RKADK\\_STREAM\\_AencStop](#)  
[RKADK\\_STREAM\\_GetAudioInfo](#)  
[RKADK\\_STREAM\\_AencRegisterCallback](#)  
[RKADK\\_STREAM\\_AencUnRegisterCallback](#)

#### 数据类型

[RKADK\\_CODEC\\_TYPE\\_E](#)  
[RKADK\\_VENC\\_DATA\\_PROC\\_FUNC](#)  
[RKADK\\_VIDEO\\_STREAM\\_S](#)  
[RKADK\\_VENC\\_DATA\\_PACK\\_S](#)  
[RKADK\\_VENC\\_DATA\\_TYPE\\_S](#)  
[RKADK\\_VIDEO\\_INFO\\_S](#)  
[RKADK\\_AENC\\_DATA\\_PROC\\_FUNC](#)  
[RKADK\\_AUDIO\\_STREAM\\_S](#)  
[RKADK\\_AUDIO\\_INFO\\_S](#)

### 播放器

#### 概述

#### API 参考

[RKADK\\_PLAYER\\_Create](#)  
[RKADK\\_PLAYER\\_Destroy](#)  
[RKADK\\_PLAYER\\_SetDataSource](#)  
[RKADK\\_PLAYER\\_Prepare](#)  
[RKADK\\_PLAYER\\_SetVideoSink](#)  
[RKADK\\_PLAYER\\_Play](#)  
[RKADK\\_PLAYER\\_Stop](#)  
[RKADK\\_PLAYER\\_Pause](#)  
[RKADK\\_PLAYER\\_Seek](#)  
[RKADK\\_PLAYER\\_GetPlayStatus](#)  
[RKADK\\_PLAYER\\_GetDuration](#)

#### 数据类型

[RKADK\\_PLAYER\\_EVENT\\_E](#)  
[RKADK\\_PLAYER\\_EVENT\\_FN](#)  
[RKADK\\_PLAYER\\_CFG\\_S](#)  
[RKADK\\_PLAYER\\_VO\\_FORMAT\\_E](#)  
[RKADK\\_PLAYER\\_VO\\_DEV\\_E](#)  
[RKADK\\_PLAYER\\_VO\\_INTF\\_TYPE\\_E](#)  
[RKADK\\_PLAYER\\_VO\\_ATTR\\_S](#)  
[RKADK\\_PLAYER\\_FRAMEINFO\\_S](#)  
[RKADK\\_PLAYER\\_STATE\\_E](#)

### 直播

#### 概述

#### API参考

##### RTSP

[RKADK\\_RTSP\\_Init](#)  
[RKADK\\_RTSP\\_DeInit](#)  
[RKADK\\_RTSP\\_Start](#)  
[RKADK\\_RTSP\\_Stop](#)

##### RTMP

[RKADK\\_RTMP\\_Init](#)  
[RKADK\\_RTMP\\_DeInit](#)

### 存储

#### 概述

#### API参考

[RKADK\\_STORAGE\\_Init](#)  
[RKADK\\_STORAGE\\_Deinit](#)  
[RKADK\\_STORAGE\\_GetDevAttr](#)  
[RKADK\\_STORAGE\\_GetMountStatus](#)  
[RKADK\\_STORAGE\\_GetCapacity](#)  
[RKADK\\_STORAGE\\_GetFileList](#)

[RKADK\\_STORAGE\\_FreeFileList](#)  
[RKADK\\_STORAGE\\_GetFileNum](#)  
[RKADK\\_STORAGE\\_GetDevPath](#)  
[RKADK\\_STORAGE\\_Format](#)

#### 数据类型

[RKADK\\_MOUNT\\_STATUS](#)  
[RKADK\\_SORT\\_TYPE](#)  
[RKADK\\_SORT\\_CONDITION](#)  
[RKADK\\_STR\\_FOLDER\\_ATTR](#)  
[RKADK\\_STR\\_DEV\\_ATTR](#)  
[RKADK\\_FILE\\_INFO](#)  
[RKADK\\_FILE\\_LIST](#)  
[RKADK\\_FILE\\_LIST\\_ARRAY](#)

#### 本地预览

##### 概述

##### API参考

[RKADK\\_DISP\\_Init](#)  
[RKADK\\_DISP\\_DeInit](#)

#### VI ISP API

##### 概述

##### API参考

[RKADK\\_VI\\_ISP\\_Start](#)  
[RKADK\\_VI\\_ISP\\_Stop](#)

#### 参数设置

##### 概述

##### API参考

[RKADK\\_PARAM\\_Init](#)  
[RKADK\\_PARAM\\_GetCamParam](#)  
[RKADK\\_PARAM\\_SetCamParam](#)  
[RKADK\\_PARAM\\_GetCommParam](#)  
[RKADK\\_PARAM\\_SetCommParam](#)  
[RKADK\\_PARAM\\_SetDefault](#)  
[RKADK\\_PARAM\\_GetResolution](#)  
[RKADK\\_PARAM\\_GetResType](#)  
[RKADK\\_PARAM\\_GetVencChnId](#)

#### 数据类型

[RKADK\\_DEFPARAM\\_PATH](#)  
[RKADK\\_DEFPARAM\\_PATH\\_SENSOR\\_PREFIX](#)  
[RKADK\\_PARAM\\_PATH](#)  
[RKADK\\_PARAM\\_PATH\\_SENSOR\\_PREFIX](#)  
[RKADK\\_PARAM\\_TYPE\\_E](#)  
[RKADK\\_PARAM\\_RES\\_E](#)  
[RKADK\\_STREAM\\_TYPE\\_E](#)  
[RKADK\\_PARAM\\_CODEC\\_CFG\\_S](#)  
[RKADK\\_PARAM\\_BITRATE\\_S](#)  
[RKADK\\_PARAM\\_REC\\_TIME\\_S](#)  
[RKADK\\_PARAM\\_GOP\\_S](#)  
[RKADK\\_VQE\\_MODE\\_E](#)  
[MUXER\\_TYPE\\_E](#)  
[MUXER\\_PRE\\_RECORD\\_MODE\\_E](#)

#### 水印

##### 概述

##### API 参考

[RKADK\\_OSD\\_Init](#)  
[RKADK\\_OSD\\_Deinit](#)  
[RKADK\\_OSD\\_Update](#)  
[RKADK\\_OSD\\_AttachToStream](#)  
[RKADK\\_OSD\\_DettachFromStream](#)

数据类型

RKADK\_OSD\_ATTR\_S

RKADK\_OSD\_STREAM\_ATTR\_S

INI文件解析

全局INI配置文件

Sensor INI配置文件

INI 配置注意事项

示例

rkadk\_record\_test

rkadk\_photo\_test

rkadk\_stream\_test

rkadk\_player\_test

rkadk\_thumb\_test

rkadk\_rtsp\_test

rkadk\_rtmp\_test

rkadk\_storage\_test

rkadk\_disp\_test

---

## 系统概述

rkadk在rkmedia和rockit的基础上做了进一步封装，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，支持应用软件快速开发。

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。

## 录像

### 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。

## API参考

### 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。

#### 【举例】

[rkadk\\_record\\_test](#)。

#### 【相关主题】

[RKADK\\_RECORD\\_Destroy](#)。

## 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](#)

**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](#)



## 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](#)

## 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

**RKADK\_RECORD\_RegisterEventCallback**

【描述】

注册录像事件回调。

【语法】

RKADK\_S32 RKADK\_RECORD\_RegisterEventCallback([RKADK\\_MW\\_PTR](#) pRecorder, [RKADK\\_REC\\_EVENT\\_CALLBACK\\_FN](#) pfnEventCallback);

【参数】

参数名称	描述	输入/输出
pRecorder	录像任务指针	输入
pfnEventCallback	录像事件回调函数指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk\_record.h

库文件：librkadk.so

【注意】

- 创建录像任务后，才能使用RKADK\_RECORD\_RegisterEventCallback 接口。

【举例】

[rkadk\\_record\\_test](#)。

【相关主题】

无

**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

**RKADK\_GetThmInMp4**

【描述】

从MP4文件中获取缩略图数据。

【语法】

RKADK\_S32 RKADK\_GetThmInMp4(RKADK\_CHAR \*pszFileName, RKADK\_U8 \*pu8Buf,  
RKADK\_U32 \*pu32Size);

【参数】

参数名称	描述	输入/输出
pszFileName	MP4文件路径	输入
pu8Buf	缩略图数据指针	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

## 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_CHAR *pszFileName,
                                RKADK\_THUMB\_ATTR\_S *pstThumbAttr);
```

【参数】

参数名称	描述	输入/输出
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](#)

## 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](#)

# 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

[MUXER\\_EVENT\\_E](#)：录像事件枚举类型

[MUXER\\_FILE\\_EVENT\\_INFO\\_S](#)：文件相关事件信息结构体

[RKADK\\_REC\\_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

[MUXER\\_MANUAL\\_SPLIT\\_TYPE\\_E](#)：手动切分枚举类型

[MUXER\\_PRE\\_MANUAL\\_SPLIT\\_ATTR\\_S](#)：手动切分预录属性结构体

[RKADK\\_REC\\_MANUAL\\_SPLIT\\_ATTR\\_S](#)：手动切分属性结构体

# 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)

```

## RKADK\_MW\_PTR

### 【说明】

定义任务指针

### 【定义】

```
typedef RKADK_VOID *RKADK_MW_PTR;
```

## 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
```

## 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
```

## MUXER\_EVENT\_E

【说明】

定义录像事件枚举类型。

【定义】

```
typedef enum rkMUXER_EVENT_E {
    MUXER_EVENT_STREAM_START = 0,
    MUXER_EVENT_STREAM_STOP,
    MUXER_EVENT_FILE_BEGIN,
    MUXER_EVENT_FILE_END,
    MUXER_EVENT_MANUAL_SPLIT_END,
    MUXER_EVENT_ERR_CREATE_FILE_FAIL,
    MUXER_EVENT_ERR_WRITE_FILE_FAIL,
    MUXER_EVENT_BUTT
} MUXER_EVENT_E;
```

【成员】

成员名称	描述
MUXER_EVENT_STREAM_START	Reserved
MUXER_EVENT_STREAM_STOP	Reserved
MUXER_EVENT_FILE_BEGIN	开始录制一个新文件
MUXER_EVENT_FILE_END	文件录制结束
MUXER_EVENT_MANUAL_SPLIT_END	手动切分文件录制结束
MUXER_EVENT_ERR_CREATE_FILE_FAIL	Reserved
MUXER_EVENT_ERR_WRITE_FILE_FAIL	Reserved

【相关数据类型及接口】

MUXER\_FILE\_EVENT\_INFO\_S

【说明】

定义文件相关事件信息结构体。

【定义】

```
typedef struct rkMUXER_FILE_EVENT_INFO_S {
    RK_CHAR asFileName[MUXER_FILE_NAME_LEN];
    RK_U32 u32Duration; // ms
} 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\_REC\_EVENT\_INFO\_S

【说明】

定义录像事件信息结构体。

【定义】

```
typedef struct rkMUXER_EVENT_INFO_S {
    MUXER_EVENT_E enEvent;
    union {
        MUXER_FILE_EVENT_INFO_S stFileInfo;
        MUXER_ERROR_EVENT_INFO_S stErrorInfo;
    } unEventInfo;
} MUXER_EVENT_INFO_S;

typedef MUXER_EVENT_INFO_S RKADK_REC_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\_REC\_EVENT\_CALLBACK\_FN

【说明】

定义录像事件回调函数指针。

【定义】

```
typedef RKADK_VOID (*RKADK_REC_EVENT_CALLBACK_FN)(RKADK_MW_PTR pRecorder, const
RKADK_REC_EVENT_INFO_S *pstEventInfo);
```

【相关数据类型及接口】

[RKADK\\_MW\\_PTR](#)

[RKADK\\_REC\\_EVENT\\_INFO\\_S](#)

[RKADK\\_RECORD\\_RegisterEventCallback](#)

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】

[RKADK\\_RECORD\\_ATTR\\_S](#)

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(*pszFilename)[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](#)

## 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_REC_TYPE_E enRecType; /* record type */
    RKADK_REC_REQUEST_FILE_NAMES_FN pfnRequestFileNames; /* rec callbak */
} RKADK_RECORD_ATTR_S;
```

【成员】

成员名称	描述
s32CamID	Camera id
enRecType	录像类型
pfnRequestFileNames	请求文件名函数指针

【相关数据类型及接口】

[RKADK\\_REC\\_TYPE\\_E](#)

[RKADK\\_REC\\_REQUEST\\_FILE\\_NAMES\\_FN](#)

[RKADK\\_RECORD\\_Create](#)

## MUXER\_MANUAL\_SPLIT\_TYPE\_E

【说明】

定义手动切分类型。

【定义】

```
typedef enum {
    MUXER_POST_MANUAL_SPLIT = 0, /* post manual split type */
    MUXER_PRE_MANUAL_SPLIT,      /* pre manual split type */
    MUXER_NORMAL_MANUAL_SPLIT,   /* normal manual split type */
    MUXER_MANUAL_SPLIT_BUTT
} MUXER_MANUAL_SPLIT_TYPE_E;
```

【成员】

成员名称	描述
MUXER_POST_MANUAL_SPLIT	Reserved
MUXER_PRE_MANUAL_SPLIT	手动切分预录
MUXER_NORMAL_MANUAL_SPLIT	Reserved

【相关数据类型及接口】

[RKADK\\_REC\\_MANUAL\\_SPLIT\\_ATTR\\_S](#)

MUXER\_PRE\_MANUAL\_SPLIT\_ATTR\_S

【说明】

定义手动切分预录结构体。

【定义】

```
typedef struct {
    RK_U32 u32DurationSec; /* file duration of manual split file */
} MUXER_PRE_MANUAL_SPLIT_ATTR_S;
```

【成员】

成员名称	描述
u32DurationSec	手动切分录像文件时长

【相关数据类型及接口】

[RKADK\\_REC\\_MANUAL\\_SPLIT\\_ATTR\\_S](#)

RKADK\_REC\_MANUAL\_SPLIT\_ATTR\_S

【说明】

定义手动切分属性结构体。

【定义】

```
typedef struct {
    MUXER_MANUAL_SPLIT_TYPE_E enManualType;           /* maual split type */
    union {
        MUXER_POST_MANUAL_SPLIT_ATTR_S stPostSplitAttr; /* post manual split
    attr */
        MUXER_PRE_MANUAL_SPLIT_ATTR_S stPreSplitAttr;   /* pre manual split attr
    */
        MUXER_NORMAL_MANUAL_SPLIT_ATTR_S stNormalSplitAttr; /* normal manual split
    attr */
    };
} MUXER_MANUAL_SPLIT_ATTR_S;

typedef MUXER_MANUAL_SPLIT_ATTR_S RKADK_REC_MANUAL_SPLIT_ATTR_S;
```

【成员】

成员名称	描述
enManualType	手动切分类型
stPostSplitAttr	Reserved
stPreSplitAttr	手动切分预录属性结构体
stNormalSplitAttr	Reserved

【相关数据类型及接口】

- [MUXER\\_MANUAL\\_SPLIT\\_TYPE\\_E](#)
- [MUXER\\_PRE\\_MANUAL\\_SPLIT\\_ATTR\\_S](#)
- [RKADK\\_RECORD\\_ManualSplit](#)

# 拍照

## 概述

提供基本的抓拍功能，提供JPEG封装拍照，支持单拍、多拍模式。

## API 参考

### RKADK\_PHOTO\_Init

【描述】

拍照任务初始化。

【语法】

RKADK\_S32 RKADK\_PHOTO\_Init([RKADK\\_PHOTO\\_ATTR\\_S](#) \*pstPhotoAttr);

【参数】

参数名称	描述	输入/输出
pstPhotoAttr	拍照任务属性指针	输入

#### 【返回值】

返回值	描述
0	成功
非0	失败

#### 【需求】

头文件：rkadk\_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](#)

## RKADK\_PHOTO\_DeInit

#### 【描述】

拍照任务反初始化。

#### 【语法】

```
RKADK_S32 RKADK_PHOTO_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\_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](#)

## RKADK\_PHOTO\_TakePhoto

【描述】

拍照。

【语法】

RKADK\_S32 RKADK\_PHOTO\_TakePhoto([RKADK\\_PHOTO\\_ATTR\\_S](#) \*pstPhotoAttr);

【参数】

参数名称	描述	输入/输出
pstPhotoAttr	拍照任务属性指针	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk\_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

## 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\_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
pszFileName	JPG文件路径	输入
eThmType	缩略图类型	输入
pu8Buf	缩略图数据指针	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

## 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_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
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](#)

## 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](#)



## RKADK\_PHOTO\_GetData

【描述】

获取JPG数据，默认输出JPG原始分辨率。

【语法】

```
RKADK_S32 RKADK_PHOTO_GetData(RKADK_CHAR *pcFileName,  
                                RKADK\_PHOTO\_DATA\_ATTR\_S *pstDataAttr);
```

【参数】

参数名称	描述	输入/输出
pstDataAttr	JPG数据属性结构体指针	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\\_photo\\_test](#)

【相关主题】

[RKADK\\_PHOTO\\_FreeData](#)

## RKADK\_PHOTO\_FreeData

【描述】

释放[RKADK\\_PHOTO\\_GetData](#)申请的内存。必须和[RKADK\\_PHOTO\\_GetData](#)配套使用。

【语法】

```
RKADK_S32 RKADK_PHOTO_FreeData(RKADK\_PHOTO\_DATA\_ATTR\_S *pstDataAttr);
```

【参数】

参数名称	描述	输入/输出
pstDataAttr	JPG数据属性结构体指针	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\\_photo\\_test](#)

【相关主题】

[RKADK\\_PHOTO\\_GetData](#)

## 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\\_DATA\\_RECV\\_FN\\_PTR](#)：拍照数据接收函数指针

[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

[RKADK\\_PHOTO\\_DATA\\_ATTR\\_S](#)：JPG数据属性结构体

## 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\\_PHOTO\\_ATTR\\_S](#)

## 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\\_PHOTO\\_ATTR\\_S](#)

## 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 meas 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\_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_MULTI,         /* 多个MPF缩略图 */
    RKADK_PHOTO_MPF_BUTT
} 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\_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_U8 *pu8DataBuf,
                                              RKADK_U32 u32DataLen);
```

【成员】

成员名称	描述
pu8DataBuf	数据指针
u32DataLen	数据长度

【相关数据类型及接口】

[RKADK\\_PHOTO\\_ATTR\\_S](#)

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_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_PHOTO_THUMB_ATTR_S stThumbAttr;
    RKADK_PHOTO_DATA_RECV_FN_PTR pfnPhotoDataProc;
} RKADK_PHOTO_ATTR_S;
```

【成员】

成员名称	描述
u32CamID	Camera id
RKADK_PHOTO_TYPE_E	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)
stThumbAttr	缩略图参数属性
pfnPhotoDataProc	拍照数据接收回调函数指针

【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\\_THUMB\\_ATTR\\_S](#)

[RKADK\\_PHOTO\\_DATA\\_RECV\\_FN\\_PTR](#)

[RKADK\\_PHOTO\\_Init](#)

[RKADK\\_PHOTO\\_TakePhoto](#)

## 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](#)

## 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_RGB888,
    RKADK_THUMB_TYPE_RGBA8888
} 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_RGB888	输出RGB888格式缩略图
RKADK_THUMB_TYPE_RGBA8888	输出RGB8888格式缩略图

【相关数据类型及接口】

[RKADK\\_THUMB\\_ATTR\\_S](#)

[RKADK\\_GetThmInMp4Ex](#)

[RKADK\\_PHOTO\\_GetThmInJpgEx](#)

## 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](#)

## RKADK\_PHOTO\_DATA\_ATTR\_S

【说明】

定义缩略图属性结构体。

【定义】

```
typedef RKADK_THUMB_ATTR_S RKADK_PHOTO_DATA_ATTR_S;
```

【相关数据类型及接口】

[RKADK\\_THUMB\\_ATTR\\_S](#)

[RKADK\\_PHOTO\\_GetData](#)

[RKADK\\_PHOTO\\_FreeData](#)

# 远程预览

## 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。

## API参考

### RKADK\_STREAM\_VideoInit

【描述】

初始化Video模块：VI、VENC。

【语法】

RKADK\_S32 RKADK\_STREAM\_VideoInit(RKADK\_U32 u32CamID, [RKADK\\_CODEC\\_TYPE\\_E](#) enCodecType);

【参数】

参数名称	描述	输入/输出
u32CamID	Camera id	输入
enCodecType	编码类型	输入

【返回值】



返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk\_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](#)

## RKADK\_STREAM\_VideoDeInit

【描述】

反初始化Video模块：VI、VENC。

【语法】

RKADK\_S32 RKADK\_STREAM\_VideoDeInit(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\_stream.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk\\_stream\\_test](#)

【相关主题】

[RKADK\\_STREAM\\_VideoInit](#)

## RKADK\_STREAM\_VencStart

【描述】

启动VENC。

【语法】

RKADK\_S32 RKADK\_STREAM\_VencStart(RKADK\_U32 u32CamID, [RKADK\\_CODEC\\_TYPE\\_E](#) enCodecType, RKADK\_S32 s32FrameCnt);

【参数】

参数名称	描述	输入/输出
u32CamID	Camera id	输入
enCodecType	编码类型	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](#)

## RKADK\_STREAM\_VencStop

【描述】

停止VENC。

【语法】

RKADK\_S32 RKADK\_STREAM\_VencStop(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\_stream.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk\\_stream\\_test](#)

【相关主题】

[RKADK\\_STREAM\\_VencStart](#)

## 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

**RKADK\_STREAM\_VencRegisterCallback**

【描述】

注册Video数据输出回调。

【语法】

RKADK\_S32 RKADK\_STREAM\_VencRegisterCallback(RKADK\_U32 u32CamID,  
[RKADK\\_VENC\\_DATA\\_PROC\\_FUNC](#) pfnDataCB);

【参数】

参数名称	描述	输入/输出
u32CamID	Camera id	输入
pfnDataCB	数据输出回调函数	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】

- 回调函数不能处理耗时操作，否则数据流将被阻塞。

【举例】

[rkadk\\_stream\\_test](#)

【相关主题】

[RKADK\\_STREAM\\_VencUnRegisterCallback](#)

## RKADK\_STREAM\_VencUnRegisterCallback

【描述】

反注册Video数据输出回调。

【语法】

RKADK\_S32 RKADK\_STREAM\_VencUnRegisterCallback(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\_stream.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk\\_stream\\_test](#)

【相关主题】

[RKADK\\_STREAM\\_VencRegisterCallback](#)

## RKADK\_STREAM\_AudioInit

【描述】

初始化Audio模块：AI、AENC。

【语法】

RKADK\_S32 RKADK\_STREAM\_AudioInit([RKADK\\_CODEC\\_TYPE\\_E](#) enCodecType);

【参数】

参数名称	描述	输入/输出
enCodecType	编码类型	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk\_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](#)

## RKADK\_STREAM\_AudioDeInit

【描述】

反初始化Audio模块：AI、AENC。

【语法】

RKADK\_S32 RKADK\_STREAM\_AudioDeInit([RKADK\\_CODEC\\_TYPE\\_E](#) enCodecType);

【参数】

参数名称	描述	输入/输出
enCodecType	编码类型	输入

【返回值】

返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk\_stream.h

库文件：librkadk.so

【注意】

无

【举例】

[rkadk\\_stream\\_test](#)

【相关主题】

[RKADK\\_STREAM AudioInit](#)

## RKADK\_STREAM\_AencStart

【描述】

启动AENC。

【语法】

RKADK\_S32 RKADK\_STREAM\_AencStart();

【参数】

无

【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](#)

## RKADK\_STREAM\_AencStop

【描述】

停止AENC。

【语法】

RKADK\_S32 RKADK\_STREAM\_AencStop();

【参数】

无

【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](#)

## RKADK\_STREAM\_GetAudioInfo

【描述】

获取Audio信息。

【语法】

RKADK\_S32 RKADK\_STREAM\_GetAudioInfo([RKADK\\_AUDIO\\_INFO\\_S](#) \*pstAudioInfo);

【参数】

参数名称	描述	输入/输出
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

RKADK\_STREAM\_AencRegisterCallback

【描述】

注册Audio数据输出回调。

【语法】

RKADK\_VOID RKADK\_STREAM\_AencRegisterCallback([RKADK\\_CODEC\\_TYPE\\_E](#) enCodecType, [RKADK\\_AENC\\_DATA\\_PROC\\_FUNC](#) pfnDataCB);

【参数】

参数名称	描述	输入/输出
enCodecType	编码类型	输入
pfnDataCB	数据输出回调函数	输入

【返回值】

无

【需求】

头文件：rkadk\_stream.h

库文件：librkadk.so

【注意】

- 回调函数不能处理耗时操作，否则数据流将被阻塞。
- 允许同时注册两个回调，同时获取PCM和AENC音频流，通过enCodecType 区分。

【举例】

[rkadk\\_stream\\_test](#)

【相关主题】

[RKADK\\_STREAM\\_AencUnRegisterCallback](#)

RKADK\_STREAM\_AencUnRegisterCallback

【描述】

反注册Audio数据输出回调。

【语法】

RKADK\_VOID RKADK\_STREAM\_AencUnRegisterCallback([RKADK\\_CODEC\\_TYPE\\_E](#) enCodecType);

【参数】

参数名称	描述	输入/输出
enCodecType	编码格式	输入

【返回值】

无

#### 【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\\_AencRegisterCallback](#)

## 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\\_AENC\\_DATA\\_PROC\\_FUNC](#): AENC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\_CODEC\_TYPE\_E

#### 【说明】

定义编码格式枚举类型, Audio编码默认不支持MP3和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_MP3,  
    RKADK_CODEC_TYPE_G711A,  
    RKADK_CODEC_TYPE_G711U,  
    RKADK_CODEC_TYPE_G726,  
    RKADK_CODEC_TYPE_MP2,  
    RKADK_CODEC_TYPE_PCM,  
}
```

```
RKADK_CODEC_TYPE_BUTT
} RKADK_CODEC_TYPE_E;
```

## 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](#)

[RKADK\\_STREAM\\_VencRegisterCallback](#)

## 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_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

【相关数据类型及接口】

[RKADK\\_VENC\\_DATA\\_PACK\\_S](#)

[RKADK\\_VENC\\_DATA\\_PROC\\_FUNC](#)

## 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](#)

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: JPEG 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](#)

**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](#)

## RKADK\_AENC\_DATA\_PROC\_FUNC

【说明】

定义AENC数据回调函数指针。

【定义】

```
typedef RKADK_S32 (*RKADK_AENC_DATA_PROC_FUNC) (RKADK_AUDIO_STREAM_S
*pAstreamData);
```

【相关数据类型及接口】

[RKADK\\_AUDIO\\_STREAM\\_S](#)

[RKADK\\_STREAM\\_AencRegisterCallback](#)

## 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_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

【相关数据类型及接口】

[RKADK\\_AENC\\_DATA\\_PROC\\_FUNC](#)

## 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
u32ChnCntt	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](#)

---

# 播放器

---

# 概述

提供本地录像文件和音频文件播放功能，支持基本的播控操作：播放、暂停、Seek。

## API 参考

### 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](#)

### 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](#)

## RKADK\_PLAYER\_SetDataSource

【描述】

设置待播放文件路径。

【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

**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

**RKADK\_PLAYER\_SetVideoSink**

【描述】

注册视频播放回调对象，播放音频不需要调用该接口。

【语法】

RKADK\_S32 RKADK\_PLAYER\_SetVideoSink([RKADK\\_MW\\_PTR](#) pPlayer,  
[RKADK\\_PLAYER\\_FRAMEINFO\\_S](#) \*pstFrameInfo);

【参数】

参数名称	描述	输入/输出
pPlayer	播放器指针	输入
pstFrameInfo	图像信息结构体	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

**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](#)

**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](#)

## 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](#)

## 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](#)

## 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

**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

**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\\_PLAYER\\_VO\\_FORMAT\\_E](#)：图像像素格式枚举类型

[RKADK\\_PLAYER\\_VO\\_DEV\\_E](#)：显示输出设备号枚举类型

[RKADK\\_PLAYER\\_VO\\_INTF\\_TYPE\\_E](#): 显示接口枚举类型

[RKADK\\_PLAYER\\_FRAMEINFO\\_S](#): 图像信息结构体

[RKADK\\_PLAYER\\_STATE\\_E](#): 播放状态枚举类型

[RKADK\\_PLAYER\\_VO\\_ATTR\\_S](#): 视频输出属性结构体

## 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_STARTED,  
    RKADK_PLAYER_EVENT_PAUSED,  
    RKADK_PLAYER_EVENT_STOPPED,  
    RKADK_PLAYER_EVENT_EOF,  
    RKADK_PLAYER_EVENT_SOF,  
    RKADK_PLAYER_EVENT_PROGRESS,  
    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_STARTED	开始播放
RKADK_PLAYER_EVENT_PAUSED	暂停播放
RKADK_PLAYER_EVENT_STOPPED	停止播放
RKADK_PLAYER_EVENT_EOF	播放结束
RKADK_PLAYER_EVENT_SOF	Reserved
RKADK_PLAYER_EVENT_PROGRESS	播放进度，附加值为当前播放时间，单位ms
RKADK_PLAYER_EVENT_SEEK_END	Seek 完成
RKADK_PLAYER_EVENT_ERROR	播放错误

### 【相关数据类型及接口】

[RKADK\\_PLAYER\\_EVENT\\_FN](#)



## 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](#)

## 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_PLAYER_EVENT_FN pfnPlayerCallback;
} RKADK_PLAYER_CFG_S;
```

【成员】

成员名称	描述
bEnableVideo	Reserved
bEnableAudio	Reserved
pfnPlayerCallback	播放事件回调函数指针

【相关数据类型及接口】

[RKADK\\_PLAYER\\_EVENT\\_FN](#)

[RKADK\\_PLAYER\\_Create](#)

# RKADK\_PLAYER\_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_NV12,
    VO_FORMAT_NV21
} RKADK_PLAYER_VO_FORMAT_E;
```

【相关数据类型及接口】

[RKADK\\_PLAYER\\_FRAMEINFO\\_S](#)

# RKADK\_PLAYER\_VO\_DEV\_E

【说明】

定义显示输出设备号枚举类型。

【定义】

```
typedef enum {
    VO_DEV_HD0 = 0,
    VO_DEV_HD1
} RKADK_PLAYER_VO_DEV_E;
```

【成员】

成员名称	描述
VO_DEV_HD0	显示输出设备0
VO_DEV_HD1	显示输出设备1

【相关数据类型及接口】

[RKADK\\_PLAYER\\_FRAMEINFO\\_S](#)

# RKADK\_PLAYER\_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_MIPI,
} RKADK_PLAYER_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

【相关数据类型及接口】

[RKADK\\_PLAYER\\_FRAMEINFO\\_S](#)

**RKADK\_PLAYER\_VO\_ATTR\_S**

【说明】

定义视频输出属性结构体。

【定义】

```
typedef struct {
    RKADK_BOOL bMirror;
    RKADK_BOOL bFlip;
    RKADK_U32 u32Rotation;
    RKADK_RECT_S stChnRect;
} RKADK_PLAYER_VO_ATTR_S;
```

【成员】

成员名称	描述
bMirror	视频镜像
bFlip	视频翻转
u32Rotation	视频旋转：0, 90, 180, 270
stChnRect	视频输出显示区域

【相关数据类型及接口】

[RKADK\\_PLAYER\\_FRAMEINFO\\_S](#)

## RKADK\_PLAYER\_FRAMEINFO\_S

### 【说明】

定义图像信息结构体。

### 【定义】

```
typedef struct {  
    RKADK_U32  u32FrmInfos32x;  
    RKADK_U32  u32FrmInfos32y;  
    RKADK_U32  u32Dispwidth;  
    RKADK_U32  u32DispHeight;  
    RKADK_U32  u32Imgwidth;  
    RKADK_U32  u32ImgHeight;  
    RKADK_U32  u32VoLayerMode;  
    RKADK_U32  u32ChnnNum;  
    RKADK_U32  u32BorderCoIor;  
    RKADK_U32  u32BorderTopwidth;  
    RKADK_U32  u32BorderBottomwidth;  
    RKADK_U32  u32BorderLeftwidth;  
    RKADK_U32  u32BorderRightwidth;  
    RKADK_PLAYER_VO_CHNN_MODE_E  u32EnMode;  
    RKADK_PLAYER_VO_FORMAT_E  u32VoFormat;  
    RKADK_PLAYER_VO_DEV_E  u32VoDev;  
    RKADK_PLAYER_VO_INTF_TYPE_E  u32EnIntfType;  
    RKADK_U32  u32DispFrmRt;  
    VO_INTF_SYNC_E  enIntfSync;  
    VO_SYNC_INFO_S  stSyncInfo;  
    RKADK_PLAYER_VO_ATTR_S  stVoAttr;  
} RKADK_PLAYER_FRAMEINFO_S;
```

### 【成员】

成员名称	描述
u32FrmlInfoS32x	Layer显示区域x坐标
u32FrmlInfoS32y	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
u32VoLayerMode	定义图层类型
u32ChnnNum	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）
u32EnMode	画面高宽比模式（Reserved）
u32VoFormat	定义图像像素格式
u32VoDev	定义输出设备
u32EnIntfType	设置显示接口类型
u32DispFrmRt	设置分辨率
enIntfSync	设置屏幕接口同步模式
stSyncInfo	屏幕属性结构体
stVoAttr	视频输出属性结构体

#### 【注意】

- 视频输出相关属性具体可参考Rockit VO文档Rockchip\_Developer\_Guide\_MPI\_VO.pdf。
- 视频镜像、翻转、旋转、显示区域通过stVoAttr设置；不支持同时设置Mirrot/Flip 和 Rotation，设置Mirrot/Flip后，Rotation将失效。
- Layer显示区域建议不设置，使用默认值。

#### 【相关数据类型及接口】

[RKADK\\_PLAYER\\_VO\\_FORMAT\\_E](#)

[RKADK\\_PLAYER\\_VO\\_DEV\\_E](#)

[RKADK\\_PLAYER\\_VO\\_INTF\\_TYPE\\_E](#)

[RKADK\\_PLAYER\\_VO\\_ATTR\\_S](#)

[RKADK\\_PLAYER\\_SetVideoSink](#)

# 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](#)

# 直播

## 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。

## API参考

### RTSP

#### 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](#)

**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](#)

**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](#)

**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](#)

**RTMP**

**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**

**【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](#)

# 存储

## 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

## API参考

**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](#)

**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](#)

## 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
<a href="#">RKADK_STR_DEV_ATTR</a>	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

## 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
<a href="#">RKADK_MOUNT_STATUS</a>	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

## 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

## 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](#)

## 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](#)

## 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

## 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

**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

**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

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 本地预览

# 概述

提供本地预览功能。

## API参考

### 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](#)

### 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](#)

# VI ISP API

## 概述

VI ISP模块是对ISP模块的再次封装，记录ISP上下文环境，方便应用层使用。本章节只介绍[RKADK\\_VI\\_ISP\\_Start](#)和[RKADK\\_VI\\_ISP\\_Stop](#)，其他API具体用法参考docs/RV1126\_RV1109/Camera/Rockchip\_Development\_Guide\_ISP2x\_CN\_v1.6.3.pdf。

## API参考

### RKADK\_VI\_ISP\_Start

【描述】

启动ISP。

【语法】

```
int RKADK_VI_ISP_Start(RKADK_U32 u32CamId, rk_aiq_working_mode_t WDRMode,
    bool bMultiCam, const char *iq_file_dir, int fps);
```

【参数】

参数名称	描述	输入/输出
u32CamId	Camera Id	输入
WDRMode	AIQ pipeline工作模式，详情参考ISP文档	输入
bMultiCam	是否多Sensor	输入
iq_file_dir	AIQ文件路径	输入
fps	帧率	输入

【返回值】



返回值	描述
0	成功
非0	失败

【需求】

头文件：rkadk\_vi\_isp.h

库文件：librkadk.so

【注意】

- 除Player模块外，其他任一模块启动之前，都必须先调用RKADK\_VI\_ISP\_Start启动ISP。
- 支持重复调用RKADK\_VI\_ISP\_Start。
- 必须和[RKADK\\_VI\\_ISP\\_Stop](#)配套使用，每调用一次RKADK\_VI\_ISP\_Sart，必须对应调用一次[RKADK\\_VI\\_ISP\\_Stop](#)。

【举例】

[rkadk\\_record\\_test](#)

[rkadk\\_photo\\_test](#)

[rkadk\\_stream\\_test](#)

[rkadk\\_rtsp\\_test](#)

[rkadk\\_rtmp\\_test](#)

[rkadk\\_disp\\_test](#)

【相关主题】

[RKADK\\_VI\\_ISP\\_Stop](#)

## RKADK\_VI\_ISP\_Stop

【描述】

停止ISP。

【语法】

```
int RKADK_VI_ISP_Stop(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\_vi\_isp.h

库文件：librkadk.so

【注意】

- 支持重复调用RKADK\_VI\_ISP\_Stop。
- 必须和[RKADK\\_VI\\_ISP\\_Start](#)配套使用，每调用一次[RKADK\\_VI\\_ISP\\_Sart](#)，必须对应调用一次RKADK\_VI\_ISP\_Stop。

【举例】

[rkadk\\_record\\_test](#)

[rkadk\\_photo\\_test](#)

[rkadk\\_stream\\_test](#)

[rkadk\\_rtsp\\_test](#)

[rkadk\\_rtmp\\_test](#)

[rkadk\\_disp\\_test](#)

【相关主题】

[RKADK\\_VI\\_ISP\\_Start](#)

# 参数设置

## 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。

## API参考

### 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

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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

## 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](#)

## 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](#)

## 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

## 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](#)：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

[MUXER\\_TYPE\\_E](#)：录像文件类型枚举

[MUXER\\_PRE\\_RECORD\\_MODE\\_E](#)：预录模式枚举

## 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 "/etc/rkadk/rkadk_defsetting.ini"
```

## 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 "/etc/rkadk/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。

## 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"
```

## 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。

## 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_REC,          /* record enable, bool*/  
    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_PHOTO_ENABLE,     /* photo enable, bool*/
    RKADK_PARAM_TYPE_SNAP_NUM,         /* continue snap num */

    /* COMM Dependent Param */
    RKADK_PARAM_TYPE_REC_UNMUTE,        /* record audio mute, bool */
    RKADK_PARAM_TYPE_AUDIO,             /* speaker enable, bool */
    RKADK_PARAM_TYPE_VOLUME,            /* speaker volume, [0,100] */
    RKADK_PARAM_TYPE_MIC_UNMUTE,        /* mic(mute) enable, bool */
    RKADK_PARAM_TYPE_MIC_VOLUME,        /* mic volume, [0,100] */
    RKADK_PARAM_TYPE_OSD,               /* show osd or not, bool */
    RKADK_PARAM_TYPE_OSD_TIME_FORMAT,   /* osd format for time */
    RKADK_PARAM_TYPE_BOOT_SOUND,        /* boot sound enable, bool */
    RKADK_PARAM_TYPE_OSD_SPEED,         /* speed osd enable, bool */
    RKADK_PARAM_TYPE_BUTTON,
} 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, <a href="#">RKADK_PARAM_GOP_S</a>
RKADK_PARAM_TYPE_RES	录像分辨率, <a href="#">RKADK_PARAM_RES_E</a>
RKADK_PARAM_TYPE_PHOTO_RES	拍照分辨率, <a href="#">RKADK_PARAM_RES_E</a>
RKADK_PARAM_TYPE_CODEC_TYPE	录像编码类型, <a href="#">RKADK_PARAM_CODEC_CFG_S</a>
RKADK_PARAM_TYPE_BITRATE	比特率, <a href="#">RKADK_PARAM_BITRATE_S</a>
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_REC	是否开机录像
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, <a href="#">RKADK_PARAM_REC_TIME_S</a>
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, <a href="#">RKADK_PARAM_REC_TIME_S</a>
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, <a href="#">RKADK_PARAM_REC_TIME_S</a>
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_PHOTO_ENABLE	是否开机启动拍照
RKADK_PARAM_TYPE_SNAP_NUM	单次拍照张数
RKADK_PARAM_TYPE_REC_UNMUTE	是否使能录像静音
RKADK_PARAM_TYPE_AUDIO	是否使能Speaker
RKADK_PARAM_TYPE_VOLUME	Speaker音量[0,100]
RKADK_PARAM_TYPE_MIC_UNMUTE	是否使能麦克风

成员名称	描述
RKADK_PARAM_TYPE_MIC_VOLUME	麦克风音量[0,100]
RKADK_PARAM_TYPE_OSD	是否显示水印
RKADK_PARAM_TYPE_OSD_TIME_FORMAT	水印时间格式
RKADK_PARAM_TYPE_OSD_SPEED	是否显示速度水印
RKADK_PARAM_TYPE_BOOTSOUND	是否播放开机音乐

#### 【注意】

- Flip、Mirror、Antifog、WDR、HDR等Camere 硬件相关设置，在调用 RKADK\_PARAM\_SetCamParam 之前需要先调用[VI ISP API](#)对应接口设置。
- 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。
- 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\_UNMUTE：和RKADK\_RECORD\_GetAencChn、RK\_MPI\_AENC\_SetMute搭配使用。
- RKADK\_PARAM\_TYPE\_FPS：和RKADK\_VI\_ISP\_SET\_FrameRate、[RKADK\\_PARAM\\_GetVencChnId](#)、RK\_MPI\_VENC\_SetFps搭配使用。RK\_MPI\_VENC\_SetFps说明参考Rockchip\_Developer\_Guide\_Linux\_RKMedia\_CN.pdf文档。
- RKADK\_PARAM\_TYPE\_GOP：和[RKADK\\_PARAM\\_GetVencChnId](#)、RK\_MPI\_VENC\_SetGop搭配使用。RK\_MPI\_VENC\_SetGop说明参考Rockchip\_Developer\_Guide\_Linux\_RKMedia\_CN.pdf文档。
- RKADK\_PARAM\_TYPE\_BITRATE：和[RKADK\\_PARAM\\_GetVencChnId](#)、RK\_MPI\_VENC\_SetBitrate搭配使用。RK\_MPI\_VENC\_SetBitrate说明参考Rockchip\_Developer\_Guide\_Linux\_RKMedia\_CN.pdf文档。

#### 【相关数据类型及接口】

[RKADK\\_PARAM\\_GetCamParam](#)

[RKADK\\_PARAM\\_SetCamParam](#)

[RKADK\\_PARAM\\_GetCommParam](#)

[RKADK\\_PARAM\\_SetCommParam](#)

## 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,    /* 2340*1296 */
    RKADK_RES_1440P,    /* 2560*1440 */
    RKADK_RES_1520P,    /* 2688*1520 */
    RKADK_RES_1600P,    /* 2560*1600 */
    RKADK_RES_1620P,    /* 2880*1620 */
    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](#)

RKADK\_STREAM\_TYPE\_E

【说明】

定义流枚举类型。

【定义】

```
typedef enum {
    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

【相关数据类型及接口】

[RKADK\\_PARAM\\_GetVencChnId](#)

## 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](#)

## 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](#)

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](#)

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](#)

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_AO,
    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
RKADK_VQE_MODE_AO	Reserved

MUXER\_TYPE\_E

【说明】

定义录像文件类型枚举

【定义】

```
typedef enum rkMUXER_TYPE_E {
    MUXER_TYPE_MP4 = 0,
    MUXER_TYPE_MPEGTS,
    MUXER_TYPE_FLV,
    MUXER_TYPE_BUTT
} MUXER_TYPE_E;
```

【成员】

成员名称	描述
MUXER_TYPE_MP4	MP4
MUXER_TYPE_MPEGTS	Reserved
MUXER_TYPE_FLV	FLV

MUXER\_PRE\_RECORD\_MODE\_E

【说明】

定义预录模式枚举

【定义】

```
typedef enum rkMUXER_PRE_RECORD_TYPE_E {
    MUXER_PRE_RECORD_NONE = 0,
    MUXER_PRE_RECORD_MANUAL_SPLIT,
    MUXER_PRE_RECORD_SINGLE,
    MUXER_PRE_RECORD_NORMAL
} MUXER_PRE_RECORD_MODE_E;
```

【成员】

成员名称	描述
MUXER_PRE_RECORD_NONE	不预录
MUXER_PRE_RECORD_MANUAL_SPLIT	手动切分文件预录
MUXER_PRE_RECORD_SINGLE	第一个文件预录
MUXER_PRE_RECORD_NORMAL	所有文件都预录

# 水印

## 概述

提供基本的水印功能

## API 参考

### RKADK\_OSD\_Init

【描述】

水印任务初始化。

【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。

## RKADK\_OSD\_Deinit

### 【描述】

水印任务反初始化。

### 【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。

## RKADK\_OSD\_Update

### 【描述】

水印内容更新。

### 【语法】

RKADK\_S32 RKADK\_OSD\_Update([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。

**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】

- 同一水印不能在同一流类型上重复叠加。

**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

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

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_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

【相关数据类型及接口】

[RKADK\\_OSD\\_ATTR\\_S](#)

## RKADK\_OSD\_STREAM\_ATTR\_S

【说明】

定义水印位置信息。

【定义】

```
typedef struct {
    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

【相关数据类型及接口】

[RKADK\\_OSD\\_STREAM\\_ATTR\\_S](#)

## INI文件解析

### 全局INI配置文件

```
[version]
version                = 1.3.1          /* version */

/* 普通参数 */
[common]
sensor_count           = 1              /* Camera Sensor 个数 */
rec_unmute             = TRUE           /* 是否使能录像静音 */
enable_speaker         = TRUE           /* 是否使能Speaker */
speaker_volume         = 80             /* Speaker音量, [0,100] */
mic_unmute             = TRUE           /* 是否使能麦克风 */
mic_volume             = 80             /* 麦克风音量, [0,100] */
osd_time_format        = 0             /* 时间水印格式 */
osd                    = TRUE           /* 是否显示水印 */
boot_sound             = TRUE           /* 是否播放开机音乐 */
osd_speed              = FALSE          /* 是否显示速度水印 */

/* Audio 参数 */
[audio]
audio_node             = default        /* Audio 设备节点 */
sample_format          = 1             /* 采样格式, 特指SAMPLE_FORMAT_E, 详见
Rkmedia 开发文档 */
channels               = 1             /* 通道数 */
samplerate             = 48000         /* 采样率 */
samples_per_frame      = 1024          /* 每帧采样个数 */
```

```

bitrate                = 64000          /* 比特率 */
ai_layout              = 0              /* AI layout mode, 详见Rkmedia开发文档
*/
vqe_mode               = 1              /* 配置音频输入声音质量增强, 特指
RKADK_VQE_MODE_E */
codec_type             = 8              /* Record 和 Live Audio 编码类型, 默认适
配MP2, 特指RKADK_CODEC_TYPE_E */

/* 缩略图参数 */
[thumb]
thumb_width            = 320            /* 缩略图宽 */
thumb_height          = 180            /* 缩略图高 */
venc_chn              = 15             /* jpeg 编码通道 */

```

## 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 */
enable_record          = TRUE           /* 是否使能录像 */
enable_photo           = TRUE           /* 是否使能拍照 */
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] */

/* 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             /* VI捕获视频缓冲区计数 */
width                  = 2688           /* video宽 */
height                 = 1520           /* video高 */
pix_fmt                = FBC0           /* VI输出格式, 详见Rkmedia 开发
文档 */
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
width                  = 0
height                 = 0
pix_fmt                = NV12
module                 = RECORD_MAIN|PHOTO

[vi.2]
chn_id                 = 2
device_name            = rkisp_scale1
buf_cnt                = 2
width                  = 0
height                 = 0

```

```

pix_fmt                = NV12
module                 = NONE

[vi.3]
chn_id                 = 3
device_name            = rkispp_scale2
buf_cnt                = 4
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MUXER_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 */

/* 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 */
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 */
rga_chn                = 0          /* RGA通道号 */
rc_mode                = CBR        /* 编码协议类型, 支持CBR、VBR、AVBR */
max_qp                 = 48         /* QP最大值, 取值范围[8, 51] */
min_qp                 = 8          /* QP最小值, 取值范围[0, 48], 不能大于
max_qp */
full_range             = TRUE
scaling_list           = TRUE
hier_qp_en             = TRUE
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care rec.1*/
[record.1]
record_time            = 60
splite_time            = 60
lapse_interval         = 60
width                  = 848
height                 = 480
bitrate                = 407040
gop                    = 30
profile                = 100
codec_type             = 0
venc_chn               = 1
rga_chn                = 0

```



```

rc_mode                = VBR
max_qp                 = 48
min_qp                 = 8
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 */
snap_num               = 1             /* 单次拍照张数 */
venc_chn               = 2            /* Venc通道号 */
rga_chn                = 0            /* RGA通道号 */

/* 远程预览 VENC 参数 */
[preview]
width                  = 848           /* video 宽 */
height                 = 480           /* video 高 */
bitrate                = 407040        /* 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               = 1            /* Venc通道号 */
rga_chn                = 0            /* RGA通道号 */
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[8, 51] */
min_qp                 = 8            /* QP最小值, 取值范围[0, 48], 不能大于
max_qp */
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
bitrate                = 4194304
gop                    = 30
profile                = 100
codec_type             = 0
venc_chn               = 1
rga_chn                = 0
rc_mode                = VBR
max_qp                 = 48
min_qp                 = 8
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]

```

```

width                = 720                /* 显示宽 */
height              = 1280               /* 显示高 */
enable_buf_pool     = TRUE               /* 是否使能buffer pool */
buf_pool_cnt        = 3                 /* 是否使能buffer pool 个数 */
rotation            = 90                 /* 旋转度数, Options:
0/90/180/270 */
rga_chn              = 0                 /* RGA通道号 */
device_node         = /dev/dri/card0    /* 视频输出设备节点 */
img_type            = RGB888            /* 视频输出格式 */
plane_type          = 1                 /* 视频输出图层类型, 详见Rkmedia开发文档 */
z_pos               = 1                 /* 输出图层Z轴高度 */
vo_chn              = 0                 /* VO通道号 */

```

## INI 配置注意事项

- 默认INI 配置文件路径, 通过rkadk\_param.h中的RKADK\_DEFPARAM\_PATH 和 RKADK\_DEFPARAM\_PATH\_SENSOR\_PREFIX 宏修改。
- INI 配置文件路径, 通过rkadk\_param.h中的RKADK\_PARAM\_PATH和 RKADK\_PARAM\_PATH\_SENSOR\_PREFIX 宏修改, 也可以通过RKADK\_PARAM\_Init API 设置。
- 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RGA 通道, rga\_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rga\_chn, 内部使用RGA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RGA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。

## 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。

## rkadk\_record\_test

### 【说明】

Record 测试。

### 【代码路径】

external/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。	/oem/etc/iqfiles
-l	Camera id	0
-t	录像类型	0
-p	ini配置文件目录路径	/data/rkadk

## rkadk\_photo\_test

### 【说明】

Photo测试。

### 【代码路径】

external/rkadk/examples/rkadk\_photo\_test.c

### 【快速使用】

```
./rkadk_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。	/oem/etc/iqfiles
-l	Camera id	0
-p	ini配置文件目录路径	/data/rkadk
-t	获取的JPG图片的数据类型	NV12

## 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】

external/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。	/oem/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

## rkadk\_player\_test

### 【说明】

本地文件播放测试。

### 【代码路径】

external/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	0
-m	Video镜像	disbale
-f	Video翻转	disbale
-v	是否使能Video播放	disbale

### 【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。

# rkadk\_thumb\_test

获取文件缩略图测试。

【代码路径】

external/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

# rkadk\_rtsp\_test

RTSP直播测试。

【代码路径】

external/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。	/oem/etc/iqfiles
-l	Camera id	0
-p	ini配置文件目录路径	/data/rkadk

# rkadk\_rtmp\_test

RTMP直播测试。

【代码路径】

external/rkadk/examples/rkadk\_rtmp\_test.c

【快速使用】

```
1、板端启动wifi
2、板端运行./rkadk_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。	/oem/etc/iqfiles
-l	Camera id	0
-p	ini配置文件目录路径	/data/rkadk

# rkadk\_storage\_test

【说明】

存储模块测试。

【代码路径】

external/rkadk/examples/rkadk\_storage\_test.c

【快速使用】

```
./rkadk_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。

# rkadk\_disp\_test

本地预览测试。

【代码路径】

external/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。	/oem/etc/iqfiles
-l	Camera id	0
-p	ini配置文件目录路径	/data/rkadk