

Rockchip RKADK Development Guide

文件标识: RK-KF-YF-904

发布版本: V1.1.0

日期: 2021-08-23

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

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Rkadk 组件开发参考。

产品版本

芯片名称	内核版本
RV1126, RV1109	Linux 4.19

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

目录

Rockchip RKADK Development Guide

1. 系统概述

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_ManualSplit

2.2.6 RKADK_RECORD_RegisterEventCallback

2.2.7 RKADK_RECORD_GetAencChn

2.2.8 RKADK_GetThmInMp4

2.3 数据类型

2.3.1 公共数据类型

2.3.2 RKADK_MW_PTR

2.3.3 RECORD_FILE_NUM_MAX

2.3.4 MUXER_EVENT_E

2.3.5 MUXER_FILE_EVENT_INFO_S

2.3.6 RKADK_REC_EVENT_INFO_S

2.3.7 RKADK_REC_EVENT_CALLBACK_FN

2.3.8 RKADK_REC_TYPE_E

2.3.9 RKADK_REC_REQUEST_FILE_NAMES_FN

2.3.10 RKADK_RECORD_ATTR_S

2.3.11 MUXER_MANUAL_SPLIT_TYPE_E

2.3.12 MUXER_PRE_MANUAL_SPLIT_ATTR_S

2.3.13 RKADK_REC_MANUAL_SPLIT_ATTR_S

3. 拍照

3.1 概述

3.2 API 参考

3.2.1 RKADK_PHOTO_Init

3.2.2 RKADK_PHOTO_DeInit

3.2.3 RKADK_PHOTO_TakePhoto

3.2.4 RKADK_PHOTO_GetThmInJpg

3.3 数据类型

3.3.1 RKADK_PHOTO_TYPE_E

3.3.2 RKADK_PHOTO_SINGLE_ATTR_S

3.3.3 RKADK_PHOTO_MULTIPLE_ATTR_S

3.3.4 RKADK_PHOTO_THUMB_ATTR_S

3.3.5 RKADK_PHOTO_DATA_RECV_FN_PTR

3.3.6 RKADK_PHOTO_ATTR_S

3.3.7 RKADK_JPG_THUMB_TYPE_E

4. 远程预览

4.1 概述

4.2 API参考

4.2.1 RKADK_STREAM_VideoInit

4.2.2 RKADK_STREAM_VideoDeInit

4.2.3 RKADK_STREAM_VencStart

4.2.4 RKADK_STREAM_VencStop

4.2.5 RKADK_STREAM_GetVideoInfo

4.2.6 RKADK_STREAM_VencRegisterCallback

- 4.2.7 RKADK_STREAM_VencUnRegisterCallback
- 4.2.8 RKADK_STREAM_AudioInit
- 4.2.9 RKADK_STREAM_AudioDeInit
- 4.2.10 RKADK_STREAM_AencStart
- 4.2.11 RKADK_STREAM_AencStop
- 4.2.12 RKADK_STREAM_GetAudioInfo
- 4.2.13 RKADK_STREAM_AencRegisterCallback
- 4.2.14 RKADK_STREAM_AencUnRegisterCallback

4.3 数据类型

- 4.3.1 RKADK_CODEC_TYPE_E
- 4.3.2 RKADK_VENC_DATA_PROC_FUNC
- 4.3.3 RKADK_VIDEO_STREAM_S
- 4.3.4 RKADK_VENC_DATA_PACK_S
- 4.3.5 RKADK_VENC_DATA_TYPE_S
- 4.3.6 RKADK_VIDEO_INFO_S
- 4.3.7 RKADK_AENC_DATA_PROC_FUNC
- 4.3.8 RKADK_AUDIO_STREAM_S
- 4.3.9 RKADK_AUDIO_INFO_S

5. 播放器

5.1 概述

5.2 API 参考

- 5.2.1 RKADK_PLAYER_Create
- 5.2.2 RKADK_PLAYER_Destroy
- 5.2.3 RKADK_PLAYER_SetDataSource
- 5.2.4 RKADK_PLAYER_Prepare
- 5.2.5 RKADK_PLAYER_SetVideoSink
- 5.2.6 RKADK_PLAYER_Play
- 5.2.7 RKADK_PLAYER_Stop
- 5.2.8 RKADK_PLAYER_Pause
- 5.2.9 RKADK_PLAYER_Seek
- 5.2.10 RKADK_PLAYER_GetPlayStatus

5.3 数据类型

- 5.3.1 RKADK_PLAYER_EVENT_E
- 5.3.2 RKADK_PLAYER_EVENT_FN
- 5.3.3 RKADK_PLAYER_CFG_S
- 5.3.4 RKADK_PLAYER_VO_FORMAT_E
- 5.3.5 RKADK_PLAYER_VO_DEV_E
- 5.3.6 RKADK_PLAYER_VO_INTF_TYPE_E
- 5.3.7 RKADK_PLAYER_FRAMEINFO_S
- 5.3.8 RKADK_PLAYER_STATE_E

6. 直播

6.1 概述

6.2 API参考

6.2.1 RTSP

- 6.2.1.1 RKADK_RTSP_Init
- 6.2.1.2 RKADK_RTSP_DeInit
- 6.2.1.3 RKADK_RTSP_Start
- 6.2.1.4 RKADK_RTSP_Stop

6.2.2 RTMP

- 6.2.2.1 RKADK_RTMP_Init
- 6.2.2.2 RKADK_RTMP_DeInit

7. 本地预览

7.1 概述

7.2 API参考

7.2.1 RKADK_DISP_Init

7.2.2 RKADK_DISP_DeInit

8. 参数设置

8.1 概述

8.2 API参考

8.2.1 RKADK_PARAM_Init

8.2.2 RKADK_PARAM_GetCamParam

8.2.3 RKADK_PARAM_SetCamParam

8.2.4 RKADK_PARAM_GetCommParam

8.2.5 RKADK_PARAM_SetCommParam

8.2.6 RKADK_PARAM_SetDefault

8.2.7 RKADK_PARAM_GetResolution

8.2.8 RKADK_PARAM_GetResType

8.2.9 RKADK_PARAM_GetVencChnId

8.3 数据类型

8.3.1 RKADK_PARAM_TYPE_E

8.3.2 RKADK_PARAM_RES_E

8.3.3 RKADK_STREAM_TYPE_E

8.3.4 RKADK_PARAM_CODEC_CFG_S

8.3.5 RKADK_PARAM_BITRATE_S

8.3.6 RKADK_PARAM_REC_TIME_S

8.4 INI文件解析

9. 示例

9.1 rkadk_record_test

9.2 rkadk_photo_test

9.3 rkadk_stream_test

9.4 rkadk_player_test

9.5 rkadk_thumb_test

9.6 rkadk_rtsp_test

9.7 rkadk_rtmp_test

9.8 rkadk_disp_test

1. 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。

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。

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。

【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_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

2.2.6 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

2.2.7 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

2.2.8 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

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

[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

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;

#ifdef _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】

定义录像任务指针

【定义】

```

typedef RKADK_VOID *RKADK_MW_PTR;

```

2.3.3 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

```

2.3.4 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】

[RKADK_REC_EVENT_INFO_S](#)

2.3.5 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](#)

2.3.6 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】

[MUXER_EVENT_E](#)

2.3.7 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](#)

2.3.8 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](#)

2.3.9 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](#)

2.3.10 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](#)

2.3.11 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 maunal 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](#)

2.3.12 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](#)

2.3.13 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](#)

3. 拍照

3.1 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。

3.2 API 参考

3.2.1 RKADK_PHOTO_Init

【描述】

拍照任务初始化。

【语法】

RKADK_S32 RKADK_PHOTO_Init([RKADK_PHOTO_ATTR_S](#) *pstPhotoAttr);

【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](#)

3.2.2 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](#)

3.2.3 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

3.2.4 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

3.3 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_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](#)：缩略图类型枚举

3.3.1 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_PHOTO_ATTR_S](#)

3.3.2 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_PHOTO_ATTR_S](#)

3.3.3 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 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_ATTR_S](#)

3.3.4 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_MULTII,            /* 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_ATTR_S](#)

3.3.5 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](#)

3.3.6 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](#)

3.3.7 RKADK_JPG_THUMB_TYPE_E

【说明】

定义缩略图类型枚举。

【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](#)

4. 远程预览

4.1 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。

4.2 API参考

4.2.1 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](#)

4.2.2 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](#)

4.2.3 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](#)

4.2.4 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](#)

4.2.5 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

4.2.6 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](#)

4.2.7 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](#)

4.2.8 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](#)

4.2.9 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](#)

4.2.10 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](#)

4.2.11 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](#)

4.2.12 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

4.2.13 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](#)

4.2.14 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](#)

4.3 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_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

4.3.1 RKADK_CODEC_TYPE_E

【说明】

定义编码格式枚举类型。

【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;
```

4.3.2 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](#)

[RKADK_STREAM_VencRegisterCallback](#)

4.3.3 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_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](#)

4.3.4 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](#)

4.3.5 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: 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
enJPEGType	Reserved

【相关数据类型及接口】

[RKADK_CODEC_TYPE_E](#)

[RKADK_VENC_DATA_PACK_S](#)

4.3.6 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](#)

4.3.7 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](#)

4.3.8 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](#)

4.3.9 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](#)

5. 播放器

5.1 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。

5.2 API 参考

5.2.1 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](#)

5.2.2 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](#)

5.2.3 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

5.2.4 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

5.2.5 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

5.2.6 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](#)

5.2.7 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】

5.2.8 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](#)

5.2.9 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](#)

5.2.10 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

5.3 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_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

5.3.1 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_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](#)

5.3.2 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](#)

5.3.3 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_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](#)

5.3.4 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](#)

5.3.5 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](#)

5.3.6 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](#)

5.3.7 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 u32BorderColor;
    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_FRAMEINFO_S;
```

【成员】

成员名称	描述
u32FrmInfoS32x	图像起始位置x坐标
u32FrmInfoS32y	图像起始位置y坐标
u32DispWidth	图像分辨率宽度
u32DispHeight	图像分辨率高度
u32ImgWidth	图像画布宽度
u32ImgHeight	图像画布高度
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

【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。

【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_SetVideoSink](#)

5.3.8 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. 直播

6.1 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。

6.2 API参考

6.2.1 RTSP

6.2.1.1 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](#)

6.2.1.2 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](#)

6.2.1.3 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](#)

6.2.1.4 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](#)

6.2.2 RTMP

6.2.2.1 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](#)

6.2.2.2 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](#)

7. 本地预览

7.1 概述

提供本地预览功能。

7.2 API参考

7.2.1 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](#)

7.2.2 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](#)

8. 参数设置

8.1 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，其在工程中的位置为：external/rkadk/rkadk_defsetting.ini。编译时会将rkadk_defsetting.ini拷贝到etc目录，运行时如果未检测到data/rkadk_setting.ini，会自动将rkadk_defsetting.ini拷贝为data/rkadk_setting.ini。

8.2 API参考

8.2.1 RKADK_PARAM_Init

【描述】

初始化参数模块

【语法】

RKADK_S32 RKADK_PARAM_Init(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

【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。

【举例】

[rkadk_record_test](#)

【相关主题】

无

8.2.2 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](#)

8.2.3 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](#)

8.2.4 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](#)

8.2.5 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](#)

8.2.6 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

8.2.7 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](#)

8.2.8 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](#)

8.2.9 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

8.3 数据类型

参数模块主要提供以下数据类型：

[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

8.3.1 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_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_BOOTSOUND,      /* boot sound enable, bool */  
    RKADK_PARAM_TYPE_OSD_SPEED,      /* speed osd enable, bool */  
    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_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_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, 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_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
RKADK_PARAM_TYPE_MIC_VOLUME	麦克风音量[0,100]

成员名称	描述
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 之前需要手动调用RKAIQ 对应接口设置，RKAIQ使用示例位于：
external/rkadk/examples/common/isp/sample_common_isp.c，也可参加
docs/RV1126_RV1109/Camera/Rockchip_Development_Guide_ISP2x_CN_v1.6.3.pdf
- 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。

【相关数据类型及接口】

[RKADK_PARAM_GetCamParam](#)

[RKADK_PARAM_SetCamParam](#)

[RKADK_PARAM_GetCommParam](#)

[RKADK_PARAM_SetCommParam](#)

8.3.2 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](#)

8.3.3 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](#)

8.3.4 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](#)

8.3.5 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](#)

8.3.6 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](#)

8.4 INI文件解析

```
/* 普通参数 */
[common]
sensor_count           = 1           /* Camera Sensor 个数，目前只调试过单Camera */
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 */
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，目前只支持ANR */
```

```

/* 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 0 参数, 对应实际Sensor个数 */
[sensor.0]
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] */

/* Sensor 0 VI通道配置参数 */
[sensor.0.vi.0]
chn_id                 = 0            /* 通道号 */
device_name            = rkispp_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 */

[sensor.0.vi.1]
chn_id                 = 1
device_name            = rkispp_scale0
buf_cnt                = 4
width                  = 0
height                 = 0
pix_fmt                = NV12
module                 = RECORD_MAIN|PHOTO

[sensor.0.vi.2]
chn_id                 = 2
device_name            = rkispp_scale1
buf_cnt                = 2
width                  = 0
height                 = 0
pix_fmt                = NV12
module                 = NONE

[sensor.0.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

/* Sensor 0 Record 参数 */
[sensor.0.rec]
record_type          = 0          /* 录像类型，特指RKADK_REC_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 */

/* Sensor 0 Record 0 VENC 参数 */
[sensor.0.rec.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 */
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

/* Sensor 0 Record 1 VENC 参数，当 file_num = 1 时，不需要care rec.1*/
[sensor.0.rec.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
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

/* Sensor 0 Photo VENC 参数 */
[sensor.0.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 */

/* Sensor 0 远程预览 VENC 参数 */
[sensor.0.stream]
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 */
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

/* Sensor 0 直播 VENC 参数 */
[sensor.0.live]
width                        = 1280
height                      = 720
bitrate                     = 4194304
gop                          = 30
profile                     = 100
codec_type                   = 0
venc_chn                     = 1
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

/* Sensor 0 本地预览参数 */
[sensor.0.disp]
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 */

rotaion                      = 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       = 0          /* 视频输出图层类型, 详见Rkmedia 开发文档 */
z_pos            = 0          /* 输出图层z轴高度 */
vo_chn           = 0          /* VO通道号 */
```


9. 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。

9.1 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。	/etc/iqfiles
-I	Camera id	0
-t	录像类型	0

9.2 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。	//etc/iqfiles
-I	Camera id	0

9.3 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】

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。	/etc/iqfiles
-I	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

9.4 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
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。

9.5 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

9.6 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。	/etc/iqfiles
-I	Camera id	0

9.7 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。	/etc/iqfiles
-I	Camera id	0

9.8 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。	/etc/iqfiles
-I	Camera id	0