

视频编码

文件标识: RK-SYS1-MPI-VENC

发布版本: V0.2.0

日期: 2021.1

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

VENC 模块, 即视频编码模块。本模块支持多路实时编码, 且每路编码独立, 编码协议和编码 profile 可以不同。本模块支持视频编码同时, 调度 Region 模块对编码图像内容进行叠加和遮挡。

产品版本

芯片名称	内核版本
RK356X	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
V0.1.0	许丽明	2021-01-09	初始版本
V0.2.0	许丽明	2021-01-23	完善数据结构定义

目录

视频编码

目录

功能描述

举例

API 参考

- RK_MPI_VENC_CreateChn
- RK_MPI_VENC_DestroyChn
- RK_MPI_VENC_ResetChn
- RK_MPI_VENC_StartRecvFrame
- RK_MPI_VENC_StopRecvFrame
- RK_MPI_VENC_QueryStatus
- RK_MPI_VENC_SetChnAttr
- RK_MPI_VENC_GetChnAttr
- RK_MPI_VENC_SendFrame
- RK_MPI_VENC_GetStream
- RK_MPI_VENC_ReleaseStream
- RK_MPI_VENC_SetChnAttr
- RK_MPI_VENC_GetChnAttr
- RK_MPI_VENC_SetJpegParam
- RK_MPI_VENC_GetRcParam
- RK_MPI_VENC_SetRcParam
- RK_MPI_VENC_RequestIDR
- RK_MPI_VENC_GetRoiAttr
- RK_MPI_VENC_SetRoiAttr

数据类型

- VENC_MAX_CHN_NUM
- VENC_CHN_ATTR_S
- VENC_ATTR_S
- VENC_RC_ATTR_S
- VENC_H264_CBR_S
- VENC_H264_VBR_S
- VENC_H264_AVBR_S
- VENC_H265_CBR_S
- VENC_H265_VBR_S
- VENC_H265_AVBR_S
- VENC_MJPEG_CBR_S
- VENC_MJPEG_VBR_S
- VENC_GOP_ATTR_S
- VENC_RECV_PIC_PARAM_S
- VENC_CHN_STATUS_S
- VENC_STREAM_S
- VENC_PACK_S

功能描述

实现图像编码，支持H264、H265、MJPEG。

举例

```
VENC_RECV_PIC_PARAM_S pstRecvParam;
VENC_CHN_ATTR_S pstAttr;
MB_POOL vencPool

pstAttr.stVencAttr.enType = PT_H264;
pstAttr.stVencAttr.u32Picwidth = 720;
pstAttr.stVencAttr.u32PicHeight = 576;
pstAttr.stVencAttr.u32MaxPicwidth = 720;
pstAttr.stVencAttr.u32MaxPicHeight = 576;
pstAttr.stVencAttr.u32StreamBufCnt = 10;
pstAttr.stVencAttr.u32BufSize = 720 * 576 * 3 / 2;

RK_MPI_VENC_CreateChn(0, &gVencCtx->pstAttr);
RK_MPI_VENC_StartRecvFrame(0, &pstRecvParam);

MB_POOL_CONFIG_S pstMbPoolCfg;
memset(&pstMbPoolCfg, 0, sizeof(MB_POOL_CONFIG_S));
pstMbPoolCfg.u64MBSize = 720 * 576 * 2;
pstMbPoolCfg.u32MBCnt = 10;
pstMbPoolCfg.enAllocType = MB_ALLOC_TYPE_DMA;

vencPool = RK_MPI_MB_CreatePool(&pstMbPoolCfg);

RK_MPI_VENC_SendFrame(0, pstFrame, -1);
RK_MPI_VENC_GetStream(0, pstStream, -1);
RK_MPI_VENC_ReleaseStream(0, pstStream);

RK_MPI_VENC_StopRecvFrame(0);
RK_MPI_VENC_DestroyChn(0);
RK_MPI_MB_DestroyPool(vencPool);
```

API 参考

视频编码模块主要提供视频编码通道的创建和销毁、视频编码通道的复位、开启和停止接收图像、设置和获取编码通道属性、获取和释放码流等功能。

该功能模块为用户提供以下 API：

- **RK_MPI_VENC_CreateChn**：创建编码通道。
- **RK_MPI_VENC_DestroyChn**：销毁编码通道。
- **RK_MPI_VENC_ResetChn**：复位编码通道。
- **RK_MPI_VENC_StartRecvFrame**：开启编码通道接收输入图像。
- **RK_MPI_VENC_StopRecvFrame**：停止编码通道接收输入图像。

- **RK_MPI_VENC_QueryStatus**: 查询编码通道状态。
- **RK_MPI_VENC_SetChnAttr**: 设置编码通道的编码属性。
- **RK_MPI_VENC_GetChnAttr**: 获取编码通道的编码属性。
- **RK_MPI_VENC_SendFrame**: 用户发送原始图像进行编码。
- **RK_MPI_VENC_GetStream**: 获取编码码流。
- **RK_MPI_VENC_ReleaseStream**: 释放码流缓存。
- **RK_MPI_VENC_SetChnAttr**: 设置编码通道的编码属性。
- **RK_MPI_VENC_GetChnAttr**: 获取编码通道的编码属性。
- **RK_MPI_VENC_SetJpegParam**: 设置 JPEG 编码的参数集合。
- **RK_MPI_VENC_GetRcParam**: 获取通道码率控制高级参数。
- **RK_MPI_VENC_SetRcParam**: 设置通道码率控制高级参数。
- **RK_MPI_VENC_RequestIDR**: 请求 IDR 帧。
- **RK_MPI_VENC_GetRoiAttr**: 获取编码通道的感兴趣区域编码配置。
- **RK_MPI_VENC_SetRoiAttr**: 设置编码通道的感兴趣区域编码配置。

RK_MPI_VENC_CreateChn

【描述】
创建编码通道。

【语法】

```
RK_S32 RK_MPI_VENC_CreateChn(VENC_CHN veChn, const VENC_CHN_ATTR_S *pstAttr);
```

【参数】

参数名	描述	输入/输出
veChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstAttr	编码通道属性指针。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】
无

RK_MPI_VENC_DestroyChn

【描述】
销毁编码通道。

【语法】

```
RK_S32 RK_MPI_VENC_DestroyChn(VENC_CHN veChn);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_ResetChn

【描述】

复位通道。

【语法】

```
RK_S32 RK_MPI_VENC_ResetChn(VENC_CHN VeChn);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_StartRecvFrame

【描述】

开启编码通道接收输入图像。

【语法】

```
RK_S32 RK_MPI_VENC_StartRecvFrame(VENC_CHN VeChn, const VENC_RECV_PIC_PARAM_S *pstRecvParam);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstRecvParam	接收图像参数结构体指针。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_StopRecvFrame

【描述】

停止编码通道接收输入图像。

【语法】

```
RK_S32 RK_MPI_VENC_StopRecvFrame(VENC_CHN VeChn);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_QueryStatus

【描述】

查询编码通道状态。

【语法】

```
RK_S32 RK_MPI_VENC_QueryStatus(VENC_CHN VeChn, VENC_CHN_STATUS_S *pstStatus);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstStatus	编码通道的状态指针。	输出

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_SetChnAttr

【描述】

设置编码通道属性。

【语法】

```
RK_S32 RK_MPI_VENC_SetChnAttr(VENC_CHN VeChn, const VENC_CHN_ATTR_S
*pstChnAttr);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstChnAttr	编码通道属性指针。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_GetChnAttr

【描述】

获取编码通道属性。

【语法】

```
RK_S32 RK_MPI_VENC_GetChnAttr(VENC_CHN VeChn, VENC_CHN_ATTR_S *pstChnAttr);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstChnAttr	编码通道属性指针。	输出

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_SendFrame

【描述】

用户发送原始图像进行编码。

【语法】

```
RK_S32 RK_MPI_VENC_SendFrame(VENC_CHN VeChn, const VIDEO_FRAME_INFO_S *pstFrame,
RK_S32 s32MilliSec);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstFrame	原始图像信息结构指针。	输入
s32MilliSec	超时参数 s32MilliSec 设为-1 时，为阻塞接口；0 时为非阻塞接口；大于 0 时为超时等待时间，超时时间的单位为毫秒（ms）。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

RK_MPI_VENC_GetStream

【描述】

获取编码的码流。

【语法】

```
RK_S32 RK_MPI_VENC_GetStream(VENC_CHN VeChn, VENC_STREAM_S *pstStream, RK_S32 s32MilliSec);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstStream	码流结构体指针。	输出
s32MilliSec	超时参数 s32MilliSec 设为-1 时，为阻塞接口；0 时为非阻塞接口；大于 0 时为超时等待时间，超时时间的单位为毫秒（ms）。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_ReleaseStream

【描述】

释放码流缓存。

【语法】

```
RK_S32 RK_MPI_VENC_ReleaseStream(VENC_CHN VeChn, VENC_STREAM_S *pstStream);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstStream	码流结构体指针。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_SetChnAttr

【描述】

设置编码通道属性。

【语法】

```
RK_S32 RK_MPI_VENC_SetChnAttr(VENC_CHN VeChn, const VENC_CHN_ATTR_S *pstChnAttr);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstChnAttr	编码通道属性指针。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_GetChnAttr

【描述】

获取编码通道属性。

【语法】

```
RK_S32 RK_MPI_VENC_GetChnAttr(VENC_CHN VeChn, VENC_CHN_ATTR_S *pstChnAttr);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstChnAttr	编码通道属性指针。	输出

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_SetJpegParam

【描述】

设置 JPEG 协议编码通道的高级参数。

【语法】

```
RK_S32 RK_MPI_VENC_SetJpegParam(VENC_CHN VeChn, const VENC_JPEG_PARAM_S
*pstJpegParam);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstJpegParam	JPEG 协议编码通道的高级参数集合。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_GetRcParam

【描述】

获取通道码率控制高级参数。

【语法】

```
RK_S32 RK_MPI_VENC_GetRcParam(VENC_CHN VeChn, VENC_RC_PARAM_S* pstRcParam);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstRcParam	通道码率控制参数指针。	输出

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_SetRcParam

【描述】

设置编码通道码率控制器的高级参数。

【语法】

```
RK_S32 RK_MPI_VENC_SetRcParam(VENC_CHN VeChn, const VENC_RC_PARAM_S *pstRcParam);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstRcParam	编码通道码率控制器的高级参数。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_RequestIDR

【描述】

请求 IDR 帧。

【语法】

```
RK_S32 RK_MPI_VENC_RequestIDR(VENC_CHN VeChn, RK_BOOL bInstant);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
bInstant	是否使能立即编码 IDR 帧。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_GetRoiAttr

【描述】

获取 H.264/H.265 通道的 ROI 配置高级属性。

【语法】

```
RK_S32 RK_MPI_VENC_GetRoiAttr(VENC_CHN VeChn, VENC_ROI_ATTR_S *pstRoiAttr,
RK_S32 roi_index);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstRoiAttr	对应 ROI 区域的配置。	输出
u32Index	H.264/H.265 协议编码通道 ROI 区域索引。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

RK_MPI_VENC_SetRoiAttr

【描述】

设置 H.264/H.265 通道的 ROI 配置高级属性。

【语法】

```
RK_S32 RK_MPI_VENC_SetRoiAttr(VENC_CHN VeChn, const VENC_ROI_ATTR_S *pstRoiAttr,
RK_S32 roi_index);
```

【参数】

参数名	描述	输入/输出
VeChn	编码通道号。取值范围：(0, VENC_MAX_CHN_NUM)。	输入
pstRoiAttr	ROI 区域参数。	输入
u32Index	H.264/H.265 协议编码通道 ROI 区域索引。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，请参见错误码。

【注意】

无

数据类型

VENC_MAX_CHN_NUM

【说明】

定义编码通道最大个数。

【定义】

```
#define VENC_MAX_CHN_NUM          16
```

【注意事项】

无

VENC_CHN_ATTR_S

【说明】

定义编码通道属性结构体。

【定义】

```
typedef struct rkVENC_CHN_ATTR_S {  
    VENC_ATTR_S      stVencAttr;  
    VENC_RC_ATTR_S   stRcAttr;  
    VENC_GOP_ATTR_S  stGopAttr;  
} VENC_CHN_ATTR_S;
```

【成员】

成员名称	描述
stVencAttr	编码器属性。
stRcAttr	码率控制器属性。
stGopAttr	Gop Mode 类型的结构体。

【注意事项】

无

VENC_ATTR_S

【说明】

定义编码器属性结构体。

【定义】

```
typedef struct rkVENC_ATTR_S {
    RK_CODEC_ID_E enType;
    RK_U32 u32BufSize;
    RK_U32 u32Profile;
    RK_BOOL bByFrame;
    RK_U32 u32PicWidth;
    RK_U32 u32PicHeight;
    RK_U32 u32VirWidth;
    RK_U32 u32VirHeight;
    RK_U32 u32StreamBufCnt;
    union {
        VENC_ATTR_H264_S stAttrH264e;
        VENC_ATTR_H265_S stAttrH265e;
        VENC_ATTR_MJPEG_S stAttrMjpege;
        VENC_ATTR_JPEG_S stAttrJpege;
    };
} VENC_ATTR_S;
```

【成员】

成员名称	描述
enType	编码协议类型。H264、H265等。
u32BufSize	码流 buffer 大小。 推荐值： 对于 H264/H265： 一幅图像大小的 1/2。 对于 Jpeg/Mjpeg： 一幅图像宽高的乘积。 静态属性。
u32Profile	编码的等级。H.264 取值。 66：Baseline。 77：Main Profile。 100：High Profile。 H.265 取值 0：Main Profile。 1：Main 10 Profile。 Jpeg/Mjpeg 取值 0：Baseline 静态属性。
bByFrame	帧/包模式获取码流。取值范围： {RK_TRUE, RK_FALSE}。 RK_TRUE：按帧获取。 RK_FALSE：按包获取。 静态属性。
u32PicWidth	编码图像宽度, 以像素为单位。
u32PicHeight	编码图像高度, 以像素为单位。
u32VirWidth	硬件编码缓存宽度。以像素单位。
u32VirHeight	硬件编码缓存高度。以像素单位。
u32StreamBufCnt	编码输出的最大缓存个数。
stAttrH264e/stAttrMjpege/stAttrJpege/stAttrH265e	码率控制器属性。

【注意事项】

VENC_RC_ATTR_S

【说明】

定义编码通道码率控制器属性。

【定义】

```

/* RW; the type of rc*/
VENC_RC_MODE_E enRcMode;
union {
    VENC_H264_CBR_S stH264Cbr;
    VENC_H264_VBR_S stH264Vbr;
    VENC_H264_AVBR_S stH264Avbr;
}

```



```
VENC_MJPEG_CBR_S  stMjpegCbr;
VENC_MJPEG_VBR_S  stMjpegVbr;

VENC_H265_CBR_S  stH265Cbr;
VENC_H265_VBR_S  stH265Vbr;
VENC_H265_AVBR_S  stH265Avbr;

};
```

【成员】

成员名称	描述
enRcMode	RC 模式。
stH264Cbr	H.264 协议编码通道 Cbr 模式属性。
stH264Vbr	H.264 协议编码通道 Vbr 模式属性。
stH264Avbr	H.264 协议编码通道 AVbr 模式属性。
stMjpegCbr	Mjpeg 协议编码通道 cbr 模式属性。
stMjpegVbr	Mjpeg 协议编码通道 vbr 模式属性。
stH265Cbr	H.265 协议编码通道 Cbr 模式属性。
stH265Vbr	H.265 协议编码通道 Vbr 模式属性。
stH265Avbr	H.265 协议编码通道 AVbr 模式属性。

【注意事项】

无

VENC_H264_CBR_S

【说明】

定义 H.264 编码通道 CBR 属性结构。

【定义】

```
typedef struct rkVENC_H264_CBR_S {
    RK_U32  u32Gop;
    RK_U32  u32SrcFrameRateNum;
    RK_U32  u32SrcFrameRateDen;
    RK_U32  fr32DstFrameRateNum;
    RK_U32  fr32DstFrameRateDen;
    RK_U32  u32BitRate;
} VENC_H264_CBR_S;
```

【成员】

成员名称	描述
u32Gop	H.264 gop 值。 取值范围：[1, 65536]。
u32SrcFrameRateNum	输入帧率分子
u32SrcFrameRateDen	输出帧率分母
fr32DstFrameRateNum	输出帧率分子
fr32DstFrameRateDen	输出帧率分母
u32BitRate	平均 bitrate，以 kbps 为单位。

【注意事项】

假设输入帧率为30， u32SrcFrameRateNum 应设置为30， u32SrcFrameRateDen设置为1。
假设输出帧率为30， fr32DstFrameRateNum 应设置为30， fr32DstFrameRateDen设置为1。
假设输入帧率25，输出帧率12，则表示将从25帧输入图像中取12帧进行编码，其余13帧将丢掉

VENC_H264_VBR_S

【说明】

定义 H.264 编码通道 VBR 属性结构。

【定义】

```
typedef struct rkVENC_H264_VBR_S {
    RK_U32  u32Gop;
    RK_U32  u32SrcFrameRateNum;
    RK_U32  u32SrcFrameRateDen;
    RK_U32  fr32DstFrameRateNum;
    RK_U32  fr32DstFrameRateDen;
    RK_U32  u32BitRate;
} VENC_H264_VBR_S;
```

【成员】

成员名称	描述
u32Gop	H.264 gop 值。 取值范围：[1, 65536]。
u32SrcFrameRateNum	输入帧率分子
u32SrcFrameRateDen	输出帧率分母
fr32DstFrameRateNum	输出帧率分子
fr32DstFrameRateDen	输出帧率分母
u32BitRate	平均 bitrate，以 kbps 为单位。

【注意事项】

请参见 VENC_H264_CBR_S注意事项

VENC_H264_AVBR_S

【说明】

定义 H.264 编码通道 AVBR 属性结构。

【定义】

```
typedef struct rkVENC_H264_AVBR_S {
    RK_U32  u32Gop;
    RK_U32  u32SrcFrameRateNum;
    RK_U32  u32SrcFrameRateDen;
    RK_U32  fr32DstFrameRateNum;
    RK_U32  fr32DstFrameRateDen;
    RK_U32  u32BitRate;
} VENC_H264_AVBR_S;
```

【成员】

成员名称	描述
u32Gop	H.264 gop 值。 取值范围：[1, 65536]。
u32SrcFrameRateNum	输入帧率分子
u32SrcFrameRateDen	输出帧率分母
fr32DstFrameRateNum	输出帧率分子
fr32DstFrameRateDen	输出帧率分母
u32BitRate	平均 bitrate，以 kbps 为单位。

【注意事项】

请参见 VENC_H264_CBR_S 注意事项

VENC_H265_CBR_S

【说明】

定义 H.265 编码通道 CBR 属性结构。

【定义】

```
typedef struct rkVENC_H265_CBR_S {
    RK_U32  u32Gop;
    RK_U32  u32SrcFrameRateNum;
    RK_U32  u32SrcFrameRateDen;
    RK_U32  fr32DstFrameRateNum;
    RK_U32  fr32DstFrameRateDen;
    RK_U32  u32BitRate;
} VENC_H265_CBR_S;
```

【成员】

成员名称	描述
u32Gop	H.265 gop 值。 取值范围：[1, 65536]。
u32SrcFrameRateNum	输入帧率分子
u32SrcFrameRateDen	输出帧率分母
fr32DstFrameRateNum	输出帧率分子
fr32DstFrameRateDen	输出帧率分母
u32BitRate	平均 bitrate，以 kbps 为单位。

【注意事项】

请参见 VENC_H264_CBR_S 注意事项

VENC_H265_VBR_S

【说明】

定义 H.265 编码通道 VBR 属性结构。

【定义】

```
typedef struct rkVENC_H265_VBR_S {  
    RK_U32  u32Gop;  
    RK_U32  u32SrcFrameRateNum;  
    RK_U32  u32SrcFrameRateDen;  
    RK_U32  fr32DstFrameRateNum;  
    RK_U32  fr32DstFrameRateDen;  
    RK_U32  u32BitRate;  
} VENC_H265_VBR_S;
```

【成员】

成员名称	描述
u32Gop	H.265 gop 值。 取值范围：[1, 65536]。
u32SrcFrameRateNum	输入帧率分子
u32SrcFrameRateDen	输出帧率分母
fr32DstFrameRateNum	输出帧率分子
fr32DstFrameRateDen	输出帧率分母
u32BitRate	平均 bitrate，以 kbps 为单位。

【注意事项】

请参见 VENC_H264_CBR_S 注意事项

VENC_H265_AVBR_S

【说明】

定义 H.265 编码通道 AVBR 属性结构。

【定义】

```
typedef struct rkVENC_H265_AVBR_S {
    RK_U32  u32Gop;
    RK_U32  u32SrcFrameRateNum;
    RK_U32  u32SrcFrameRateDen;
    RK_U32  fr32DstFrameRateNum;
    RK_U32  fr32DstFrameRateDen;
    RK_U32  u32BitRate;
} VENC_H265_AVBR_S;
```

【成员】

成员名称	描述
u32Gop	H.265 gop 值。 取值范围：[1, 65536]。
u32SrcFrameRateNum	输入帧率分子
u32SrcFrameRateDen	输出帧率分母
fr32DstFrameRateNum	输出帧率分子
fr32DstFrameRateDen	输出帧率分母
u32BitRate	平均 bitrate，以 kbps 为单位。

【注意事项】

请参见 VENC_H264_CBR_S注意事项

VENC_MJPEG_CBR_S

【说明】

定义 mjpeg 编码通道 CBR 属性结构。

【定义】

```
typedef struct rkVENC_MJPEG_CBR_S {
    RK_U32  u32SrcFrameRateNum;
    RK_U32  u32SrcFrameRateDen;
    RK_U32  fr32DstFrameRateNum;
    RK_U32  fr32DstFrameRateDen;
    RK_U32  u32BitRate;
} VENC_MJPEG_CBR_S;
```

【成员】

成员名称	描述
u32SrcFrameRateNum	输入帧率分子
u32SrcFrameRateDen	输出帧率分母
fr32DstFrameRateNum	输出帧率分子
fr32DstFrameRateDen	输出帧率分母
u32BitRate	平均 bitrate，以 kbps 为单位。

【注意事项】
请参见 VENC_H264_CBR_S 注意事项

VENC_MJPEG_VBR_S

【说明】
定义 jpeg 编码通道 VBR 属性结构。
【定义】

```
typedef struct rkVENC_MJPEG_VBR_S {
    RK_U32 u32SrcFrameRateNum;
    RK_U32 u32SrcFrameRateDen;
    RK_U32 fr32DstFrameRateNum;
    RK_U32 fr32DstFrameRateDen;
    RK_U32 u32BitRate;
} VENC_MJPEG_VBR_S;
```

【成员】

成员名称	描述
u32SrcFrameRateNum	输入帧率分子
u32SrcFrameRateDen	输出帧率分母
fr32DstFrameRateNum	输出帧率分子
fr32DstFrameRateDen	输出帧率分母
u32BitRate	平均 bitrate，以 kbps 为单位。

【注意事项】
请参见 VENC_H264_CBR_S 注意事项

VENC_GOP_ATTR_S

【说明】
定义编码器 GOP 属性结构体。
【定义】

```
typedef struct rkVENC_GOP_ATTR_S {  
    VENC_GOP_MODE_E enGopMode;  
    RK_U32 u32GopSize;  
    RK_S32 s32IPQpDelta;  
    RK_U32 u32BgInterval;  
    RK_S32 s32ViQpDelta;  
} VENC_GOP_ATTR_S;
```

【成员】

成员名称	描述
enGopMode	编码 GOP 类型。
u32GopSize	GOP长度。
s32IPQpDelta	I 帧相对 P 帧的 QP 差值。
u32BgInterval	长期参考帧的间隔。 取值范围：[u32GopSize, 65536]，且必须是 u32GopSize的整数倍。
s32ViQpDelta	虚拟 I 帧相对于普通 P 帧的 QP 差值。 取值范围：[-10, 30]。

【注意事项】

无

VENC_RECV_PIC_PARAM_S

【说明】

定义编码通道连续接收并编码的帧数结构体。

【定义】

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

【成员】

成员名称	描述
s32RecvPicNum	编码通道连续接收并编码的帧数。范围：[-1,0)∪(0 ∞]

【注意事项】

无

VENC_CHN_STATUS_S

【说明】

定义编码通道的状态结构体。

【定义】

```
typedef struct rkVENC_CHN_STATUS_S {
    RK_U32  u32LeftPics;
    RK_U32  u32LeftStreamBytes;
    RK_U32  u32LeftStreamFrames;
    RK_U32  u32CurPacks;
    RK_U32  u32LeftRecvPics;
    RK_U32  u32LeftEncPics;
    RK_BOOL  bJpegSnapEnd;
    VENC_STREAM_INFO_S  stVencStrmInfo;
} VENC_CHN_STATUS_S;
```

【成员】

成员名称	描述
u32LeftPics	待编码的图像数。
u32LeftStreamBytes	码流 buffer 剩余的 byte 数。
u32LeftStreamFrames	码流 buffer 剩余的帧数。
u32CurPacks	当前帧的码流包个数。
u32LeftRecvPics	剩余待接收的帧数，在用户调用接口RK_MPI_VENC_StartRecvFrame设置接收帧数后有效。
u32LeftEncPics	剩余待编码的帧数，在用户调用接口RK_MPI_VENC_StartRecvFrame设置接收帧数后有效。
bJpegSnapEnd	Jpege 抓拍模式下指示抓拍过程是否结束。
stVencStrmInfo	编码器码流特征信息。

【注意事项】

无

VENC_STREAM_S

【说明】

定义帧码流类型结构体。

【定义】

```
typedef struct rkVENC_STREAM_S {
    VENC_PACK_S  ATTRIBUTE*  pstPack;
    RK_U32      ATTRIBUTE  u32PackCount;
    RK_U32      u32Seq;

    union {
        VENC_STREAM_INFO_H264_S  stH264Info;
        VENC_STREAM_INFO_JPEG_S   stJpegInfo;
        VENC_STREAM_INFO_H265_S   stH265Info;
        VENC_STREAM_INFO_PRORES_S stProresInfo;
    };

    union {
        VENC_STREAM_ADVANCE_INFO_H264_S  stAdvanceH264Info;
```



```

        VENC_STREAM_ADVANCE_INFO_JPEG_S    stAdvanceJpegInfo;
        VENC_STREAM_ADVANCE_INFO_H265_S    stAdvanceH265Info;
        VENC_STREAM_ADVANCE_INFO_PRORES_S  stAdvanceProresInfo;
    };
} VENC_STREAM_S;

```

【成员】

成员名称	描述
pstPack	帧码流包结构。
u32PackCount	一帧码流的所有包的个数。
u32Seq	码流序列号。按帧获取帧序号；按包获取包序号。
stH624Info/stJpegInfo/ stH265Info/stProresInfo	码流特征信息。
stAdvanceH264Info/ stAdvanceJpegInfo/ stAdvanceH265Info/stAdvanceProresInfo	码流高级特征信息。

【注意事项】

无

VENC_PACK_S

【说明】

定义帧码流包结构体。

【定义】

```

typedef struct rkVENC_PACK_S {
    MB_BLK          pMbBlk;
    RK_U32          u32Len;
    RK_U64          u64PTS;
    RK_BOOL         bFrameEnd;
    RK_BOOL         bStreamEnd;
    VENC_DATA_TYPE_U DataType;
    RK_U32          u32Offset;
    RK_U32          u32DataNum;
    VENC_PACK_INFO_S stPackInfo[8];
} VENC_PACK_S;

```

【成员】

成员名称	描述
pMbBlk	码流包缓存块句柄。
u32Len	码流包长度。
u64PTS	时间戳。单位：us。
bFrameEnd	帧结束标识。 取值范围： RK_TRUE：该码流包是该帧的最后一个包。 RK_FALSE：该码流包不是该帧的最后一个包。
bStreamEnd	流结束标识。 取值范围： RK_TRUE：该码流包是数据流的最后一个包，编码器将输出编码图像最后一帧码流。 RK_FALSE：该码流包不是数据流的最后一个包。
DataType	码流类型，支持 H.264/JPEG/ H.265 协议类型的数据包。
u32Offset	码流包中有效数据与码流包缓存块 的偏移。
u32DataNum	当前码流包（当前包的类型由 DataType 指定）数据中包含其他类型码流包的个数。
stPackInfo	当前码流包数据中包含其他类型码流包数据信息。

【注意事项】

无

VENC_JPEG_PARAM_S

【说明】

定义 JPEG 协议编码通道高级参数结构体。

【定义】

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

【成员】

成员名称	描述
u32Qfactor	具体含义请参见 RFC2435 协议，系统默认为 90。取值范围：[1, 99]。
u8YQt	Y 量化表。取值范围：[1, 255]。
u8CbQt	Cb 量化表。取值范围：[1, 255]。
u8CrQt	Cr 量化表。取值范围：[1, 255]。
u32MCUPerECS	每个 ECS 中包含多少个 MCU，系统默认为 0，表示不划分 Ecs。

【注意事项】

无

VENC_ROI_ATTR_S

【说明】

定义编码感兴趣区域信息。

【定义】

```
typedef struct rkVENC_ROI_ATTR_S {
    RK_U32    u32Index;
    RK_BOOL   bEnable;
    RK_BOOL   bAbsQp;
    RK_S32    s32Qp;
    RK_BOOL   bIntra;
    RECT_S    stRect;
} VENC_ROI_ATTR_S;
```

【成员】

成员名称	描述
u32Index	ROI 区域的索引，系统支持的索引范围为[0,7]，不支持超出这个范围的索引。
bEnable	是否使能这个 ROI 区域。
bAbsQp	ROI 区域 QP 模式。RK_FALSE：相对 QP。RK_TURE：绝对 QP
s32Qp	QP 值，当 QP 模式为 RK_FALSE 时，s32Qp 为 QP 偏移，s32Qp 范围[-51,51]，当 QP 模式为 RK_TRUE 时，s32Qp 为宏块 QP 值，s32Qp 范围[0,51]。
bIntra	是否为I帧
RECT_S	ROI 区域。s32X、s32Y、u32Width、u32Height 必须是 16 对齐。

【注意事项】

无

错误码

视频编码 API 错误码如下所示：

错误代码	宏定义	描述
0xA00488002	RK_ERR_VENC_INVALID_CHNID	通道 ID 超出合法范围
0xA00488003	RK_ERR_VENC_ILLEGAL_PARAM	参数超出合法范围
0xA00488004	RK_ERR_VENC_EXIST	试图申请或者创建已经存在的设备、通道或者资源
0xA00488005	RK_ERR_VENC_UNEXIST	试图使用或者销毁不存在的设备、通道或者资源
0xA00488006	RK_ERR_VENC_NULL_PTR	函数参数中有空指针
0xA00488007	RK_ERR_VENC_NOT_CONFIG	使用前未配置
0xA00488008	RK_ERR_VENC_NOT_SUPPORT	不支持的参数或者功能
0xA00488009	RK_ERR_VENC_NOT_PERM	该操作不允许，如试图修改静态配置参数
0xA0048800C	RK_ERR_VENC_NOMEM	分配内存失败，如系统内存不足
0xA0048800D	RK_ERR_VENC_NOBUF	分配缓存失败，如申请的数据缓冲区太大
0xA0048800E	RK_ERR_VENC_BUF_EMPTY	缓冲区中无数据
0xA00488010	RK_ERR_VENC_BUF_FULL	缓冲区中数据满
0xA00488010	RK_ERR_VENC_SYS_NOTREADY	系统没有初始化或没有加载 相应模块
0xA00488012	RK_ERR_VENC_BUSY	VENC 系统忙