音频编码

文件标识：RK-SYS1-MPI-AENC

发布版本：V0.1.0

日　　期：2021.3

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。

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

本文档主要介绍AENC的API和数据类型。

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	周弟东	2021-3-11	初始版本

目录

基本概念

音频编码主要实现创建编码通道、发送音频帧编码及获取编码码流等功能。

举例

```
RK_S32 s32ret;
AENC_CHN_ATTR_S stAencAttr;

// init aenc params
AENC_CHN AdChn = 0;
RK_BOOL bBlock = RK_TRUE;
stAencAttr.enType = RK_AUDIO_ID_ADPCM_G722;
stAencAttr.stAencCodec.u32Channels = 2;
stAencAttr.stAencCodec.u32SampleRate = 44100;
stAencAttr.stAencCodec.enBitwidth = AUDIO_BIT_WIDTH_16;
stAencAttr.u32BufCount = 4;
```

```
stAencAttr.extraDataSize = 0;
stAencAttr.extraData = RK_NULL;

s32ret = RK_MPI_AENC_CreateChn(AdChn, &stAencAttr);
if (s32ret) {
    RT_LOGE("create aenc chn %d err:0x%x\n", AdChn, s32ret);
    return s32ret;
}

MB_EXT_CONFIG_S extConfig = {0};
extConfig.pFreeCB = aenc_data_free;
extConfig.pOpaque = srcData;
extConfig.pu8VirAddr = srcData;
extConfig.u64Size    = srcSize;
RK_MPI_SYS_CreateMB(&(stAudioFrm.pMbBlk), &extConfig);
s32ret = RK_MPI_AENC_SendFrame(AdChn, &stAudioFrm, RK_NULL, -1);
if (s32ret != RK_SUCCESS) {
     RK_LOGE("fail to send aenc stream.");
     RK_MPI_SYS_Free(stAudioFrm.pMbBlk);
}

s32ret = RK_MPI_AENC_GetStream(AdChn, &pstStream, -1);
if (s32ret == RK_SUCCESS) {
    MB_BLK bBlk = pstStream.pMbBlk;
    RK_VOID *pstFrame = RK_MPI_MB_Handle2VirAddr(bBlk);
    if (pstFrame) {
        RK_MPI_AENC_ReleaseStream(AdChn, &pstStream);
    }
}
// destroy a aenc chn
s32ret = RK_MPI_AENC_DestroyChn(AdChn);
if (RK_SUCCESS != s32ret) {
    return s32ret;
}
```

详细测试DEMO，请参考发布文件：test_mpi_aenc.cpp。

API 参考

该功能模块为用户提供以下API:

- RK_MPI_AENC_CreateChn：创建音频编码通道。
- RK_MPI_AENC_DestroyChn：销毁音频编码通道。
- RK_MPI_AENC_SendFrame：发送音频编码音频帧。
- RK_MPI_AENC_GetStream：获取音频编码码流。
- RK_MPI_AENC_ReleaseStream：释放音频编码码流。
- RK_MPI_AENC_SaveFile：开启音频编码之前通道存文件功能。
- RK_MPI_AENC_QueryFileStatus：查询音频编码通道是否处于存文件的状态。

RK_MPI_AENC_CreateChn

【描述】
创建音频解码通道。
【语法】
RK_S32 RK_MPI_AENC_CreateChn(AENC_CHN AeChn, const AENC_CHN_ATTR_S *pstAttr);

【参数】

参数名	描述	输入/输出
AeChn	通道号。 取值范围：[0,AENC_MAX_CHN_NUM)。	输入
pstAttr	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】

- 音频编码支持的解码协议：RK_AUDIO_ID_ADPCM_G722、RK_AUDIO_ID_ADPCM_G726、RK_AUDIO_ID_PCM_MULAW、RK_AUDIO_ID_PCM_ALAW等，参见rk_common.h中RK_CODEC_ID_E枚举音频定义。
- 音频编码的初始化属性必须设置码流的采样率（u32SampleRate）、声道数（u32Channels）、采样精度（enBitwidth）、codec id（enType)参数。
- 在通道闲置时才能使用此接口，如果通道已经被创建，则返回通道已经创建的错误。

RK_MPI_AENC_DestroyChn

【描述】
销毁音频编码通道。
【语法】

RK_S32 RK_MPI_AENC_DestroyChn(AENC_CHN AeChn);

【参数】

参数名	描述	输入/输出
AeChn	通道号。 取值范围：[0,AENC_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】

- 通道未创建的情况下调用此接口会返回RK_ERR_AENC_UNEXIST。
- 建议通道使用完成后调用此接口。

RK_MPI_AENC_SendFrame

【描述】
发送音频编码音频帧。
【语法】

RK_S32 RK_MPI_AENC_SendFrame(AENC_CHN AeChn, const AUDIO_FRAME_S *pstFrm, const AEC_FRAME_S *pstAecFrm, RK_S32 s32MilliSec);

【参数】

参数名	描述	输入/输出
AeChn	通道号。 取值范围：[0,AENC_MAX_CHN_NUM)。	输入
pstFrm	音频帧结构体指针。	输入
pstAecFrm	回声抵消参考帧结构体指针。	输入
s32MilliSec	发送数据的超时时间: -1表示阻塞模式； 0表示非阻塞模式； >0表示阻塞s32MilliSec毫秒，超时则报错返回。	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】

- 目前回声抵消未实现，pstAecFrm可置为NULL。
- s32MilliSec的值必须大于等于-1，等于-1时采用阻塞模式发送数据，等于0 时采用非阻塞模式发送数据，大于 0 时，阻塞 s32MilliSec 毫秒后，则返回超时并报错。
- 该接口用于用户主动发送音频帧进行编码，如果 AENC 通道已经通过系统绑定（HI_MPI_SYS_Bind）接口与 AI 绑定，不需要也不建议调此接口。
- 调用该接口发送音频编码音频帧时，必须先创建对应的编码通道。

RK_MPI_AENC_GetStream

【描述】
获取编码后码流。
【语法】

RK_S32 RK_MPI_AENC_GetStream(AENC_CHN AeChn, AUDIO_STREAM_S *pstStream, RK_S32 s32MilliSec);

【参数】

参数名	描述	输入/输出
AeChn	通道号。 取值范围：[0,AENC_MAX_CHN_NUM)。	输入
pstStream	音频帧数据结构体。	输出
s32MilliSec	获取数据的超时时间: -1表示阻塞模式； 0表示非阻塞模式； >0表示阻塞s32MilliSec毫秒，超时则报错返回。	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】

- 必须创建通道后才可能获取码流。
- s32MilliSec的值必须大于等于-1，等于-1 时采用阻塞模式获取数据，等于 0 时采用非阻塞模式获取数据，大于0时，阻塞s32MilliSec 毫秒后，没有数据则返回超时并报错。

RK_MPI_AENC_ReleaseStream

【描述】
释放从音频编码通道获取的码流。
【语法】

RK_S32 RK_MPI_AENC_ReleaseStream(AENC_CHN AeChn, const AUDIO_STREAM_S *pstStream);

【参数】

参数名	描述	输入/输出
AeChn	通道号。 取值范围：[0,AENC_MAX_CHN_NUM)。	输入
pstStream	获取的码流指针。	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】

- 码流最好能够在使用完之后立即释放，如果不及时释放，会导致编码过程阻塞。
- 释放的码流必须是从该通道获取的码流，不得对码流信息结构体进行任何修改，否则会导致码流不能释放，使此码流 buffer 丢失，甚至导致程序异常。

- 释放码流时必须保证通道已经被创建，否则直接返回失败，如果在释放码流过程中销毁通道则会立刻返回失败。

RK_MPI_AENC_SaveFile

【描述】
开启音频编码之前通道存文件功能。
【语法】

RK_S32 RK_MPI_AENC_SaveFile(AENC_CHN AeChn, const AUDIO_SAVE_FILE_INFO_S *pstSaveFileInfo);

【参数】

参数名	描述	输入/输出
AeChn	通道号。 取值范围：[0,AENC_MAX_CHN_NUM)。	输入
pstSaveFileInfo	音频保存文件属性结构体指针。	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】

- 在 AENC通道成功启用后再调用此接口。

RK_MPI_AENC_QueryFileStatus

【描述】
查询频编码通道是否处于存文件的状态。
【语法】

RK_S32 RK_MPI_AENC_QueryFileStatus(AENC_CHN AeChn, AUDIO_FILE_STATUS_S* pstFileStatus);

【参数】

参数名	描述	输入/输出
AeChn	通道号。 取值范围：[0,AENC_MAX_CHN_NUM)。	输入
pstFileStatus	状态属性结构体指针。	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】

- 此接口用于查询音频输出通道是否处于存文件的状态。

数据类型

AENC_CHN

【说明】
定义 AENC 通道。
【定义】

```
typedef RK_S32 AENC_CHN;
```

AENC_MAX_CHN_NUM

【说明】
定义音频解码通道的最大个数。
【定义】

```
#define  AENC_MAX_CHN_NUM  32
```

AENC_ATTR_CODEC_S

【说明】
定义音频编码器属性结构体
【定义】

```
typedef struct rkAENC_ATTR_CODEC_S {
    RK_U32 u32Channels;
    RK_U32 u32SampleRate;
    AUDIO_BIT_WIDTH_E enBitwidth;
} AENC_ATTR_CODEC_S;
```

【成员】

成员名称	描述
u32Channels	码流声道。
u32SampleRate	码流采样率。
enBitwidth	采样精度。

AENC_CHN_ATTR_S

【说明】
定义编码通道属性结构体。
【定义】

```
typedef struct rkAENC_CHN_ATTR_S {
    RK_CODEC_ID_E     enType;         /* audio codec id */
    AENC_ATTR_CODEC_S stAencCodec;    /* channel count & samplerate */
    RK_U32            u32BufCount;     /* encode buffer count */
    MB_BLK            extraData;      /* encode key parameters */
    RK_U32            extraDataSize;  /* key parameters size */
} AENC_CHN_ATTR_S;
```

【成员】

成员名称	描述
enType	解码器codec id。
stAencCodec	解码器属性参数。
u32BufCount	音频编码缓存个数。
extraData	解码器外部数据。
extraDataSize	解码器外部数据大小。

AUDIO_STREAM_S

【说明】
定义音频码流结构体。
【定义】

```
typedef struct rkAUDIO_STREAM_S {
    MB_BLK pMbBlk;
    RK_U32 u32Len; /* stream lenth, by bytes */
    RK_U64 u64TimeStamp; /* frame time stamp*/
    RK_U32 u32Seq; /* frame seq,if stream is not a valid frame, u32Seq is 0*/
    RK_BOOL bBypassMbBlk; /* FALSE: copy, TRUE: MbBlock owned by internal */
} AUDIO_STREAM_S;
```

【成员】

成员名称	描述
pMbBlk	缓存块句柄。
u32Len	音频码流长度。
u64TimeStamp	音频码流时间戳。
bBypassMbBlk	是否需要拷贝外部定义的MB_BLK到内部。FALSE: 需要拷贝，TRUE：不需要拷贝

AUDIO_FRAME_S

【说明】
定义音频帧数据结构体。
【定义】

```
typedef struct rkAUDIO_FRAME_S {
    MB_BLK              pMbBlk;
    AUDIO_BIT_WIDTH_E   enBitWidth;     /*audio frame bitwidth*/
    AUDIO_SOUND_MODE_E  enSoundMode;    /*audio frame momo or stereo mode*/
    RK_U64              u64TimeStamp;   /*audio frame timestamp*/
    RK_U32              u32Seq;         /*audio frame seq*/
    RK_U32              u32Len;         /*data lenth per channel in frame,
u32Len <= 0 mean eos*/
} AUDIO_FRAME_S;
```

【成员】

成员名称	描述
pMbBlk	缓存块句柄。
enBitWidth	音频采样精度。
enSoundMode	音频声道模式。
u64TimeStamp	音频帧时间戳。以 μs 为单位。
u32Seq	音频帧序号。
u32Len	音频帧长度。以 byte 为单位。

【注意事项】

- AUDIO_FRAME_S的数据存储在pMbBlk中，输入输出均需要外部申请合理合法的缓存块。

AUDIO_BIT_WIDTH_E

【说明】
定义音频采样精度。
【定义】

```
typedef enum rkAUDIO_BIT_WIDTH_E {
    AUDIO_BIT_WIDTH_8   = 0,   /* 8bit width */
    AUDIO_BIT_WIDTH_16  = 1,   /* 16bit width*/
    AUDIO_BIT_WIDTH_24  = 2,   /* 24bit width*/
    AUDIO_BIT_WIDTH_BUTT,
} AUDIO_BIT_WIDTH_E;
```

【成员】

成员名称	描述
AUDIO_BIT_WIDTH_8	采样精度为8bit位宽。
AUDIO_BIT_WIDTH_16	采样精度为16bit位宽。
AUDIO_BIT_WIDTH_24	采样精度为24bit位宽。

错误码

音频解码API错误码如下所示。

错误代码	宏定义	描述
0xA00C8001	RK_ERR_AENC_INVALID_DEVID	音频设备号无效
0xA00C8002	RK_ERR_AENC_INVALID_CHNID	音频编码通道号无效
0xA00C8003	RK_ERR_AENC_ILLEGAL_PARAM	音频编码参数设置无效
0xA00C8004	RK_ERR_AENC_EXIST	音频编码通道已经创建
0xA00C8005	RK_ERR_AENC_UNEXIST	音频编码通道未创建
0xA00C8006	RK_ERR_AENC_NULL_PTR	输入参数空指针错误
0xA00C8007	RK_ERR_AENC_NOT_CONFIG	编码通道未配置
0xA00C8008	RK_ERR_AENC_NOT_SUPPORT	操作不被支持
0xA00C8009	RK_ERR_AENC_NOT_PERM	操作不允许
0xA00C800C	RK_ERR_AENC_NOMEM	系统内存不足
0xA00C800D	RK_ERR_AENC_NOBUF	编码通道缓存分配失败
0xA00C800E	RK_ERR_AENC_BUF_EMPTY	编码通道缓存空
0xA00C8010	RK_ERR_AENC_BUF_FULL	音频输出缓存为满
0xA00C8010	RK_ERR_AENC_SYS_NOTREADY	系统没有初始化
0xA00C8040	RK_ERR_AENC_DECODER_ERR	音频编码数据错误