

音频输入

文件标识：RK-SYS1-MPI-AI

发布版本：V0.1.0

日期：2021.2

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

本文档主要介绍AI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	2021-2-1	初始版本

目录

音频输入

目录

基本概念

举例

API 参考

- RK_MPI_AI_SetPubAttr
- RK_MPI_AI_GetPubAttr
- RK_MPI_AI_Enable
- RK_MPI_AI_Disable
- RK_MPI_AI_EnableChn
- RK_MPI_AI_DisableChn
- RK_MPI_AI_GetFrame
- RK_MPI_AI_ReleaseFrame
- RK_MPI_AI_SetChnParam
- RK_MPI_AI_GetChnParam
- RK_MPI_AI_EnableReSmp
- RK_MPI_AI_DisableReSmp
- RK_MPI_AI_SetTrackMode
- RK_MPI_AI_GetTrackMode
- RK_MPI_AI_ClrPubAttr
- RK_MPI_AI_SaveFile
- RK_MPI_AI_QueryFileStatus

数据类型

- AUDIO_DEV
- AI_CHN
- AI_MAX_CHN_NUM
- AI_DEV_MAX_NUM
- AIO_ATTR_S
- AUDIO_FRAME_S
- AUDIO_SOUND_MODE_E
- AUDIO_BIT_WIDTH_E
- AIO_SOUND_CARD
- AUDIO_SAMPLE_RATE_E
- AUDIO_TRACK_MODE_E
- AUDIO_SAVE_FILE_INFO_S
- AUDIO_FILE_STATUS_S

错误码

基本概念

音频输入（AI） 主要实现配置及启用音频输入设备、获取音频帧数据等功能。

举例

```
AIO_ATTR_S aiAttr;
memset(&aiAttr, 0, sizeof(AIO_ATTR_S));

aiAttr.soundCard.channels = 2;
aiAttr.soundCard.sampleRate = 16000;
aiAttr.soundCard.bitwidth = AUDIO_BIT_WIDTH_16;

aiAttr.enBitwidth = AUDIO_BIT_WIDTH_16;
aiAttr.enSamplerate = AUDIO_SAMPLE_RATE_16000;
aiAttr.enSoundmode = AUDIO_SOUND_MODE_STEREO;
aiAttr.u32FrmNum = 4;
aiAttr.u32PtNumPerFrm = 1024;

aiAttr.u32EXFlag = 0;
aiAttr.u32ChnCnt = 2;
RK_S32 result = RK_MPI_AI_SetPubAttr(aiDevId, &aiAttr);
if (result != 0) {
    RK_LOGE("ao set attr fail, reason = %d", result);
    return RK_FAILURE;
}

result = RK_MPI_AI_Enable(aiDevId);
if (result != 0) {
    RK_LOGE("ao enable fail, reason = %d", result);
    return RK_FAILURE;
}

result = RK_MPI_AI_EnableChn(aiDevId, aiChn);
if (result != 0) {
    RK_LOGE("ao enable channel fail, reason = %d", result);
    return RK_FAILURE;
}

result = RK_MPI_AI_EnableReSmp(aiDevId, aiChn, AUDIO_SAMPLE_RATE_16000);
if (result != 0) {
    RK_LOGE("ao enable channel fail, reason = %d", result);
    return RK_FAILURE;
}

AUDIO_FRAME_S frame;

result = RK_MPI_AI_GetFrame(aiDevId, aiChn, &frame, RK_NULL, s32MilliSec);
if (result == 0) {
    RK_MPI_AI_ReleaseFrame(aiDevId, aiChn, &frame, RK_NULL);
}

RK_MPI_AI_DisableReSmp(aiDevId, aiChn);
RK_S32 result = RK_MPI_AO_DisableChn(aiDevId, aiDevId);
if (result != 0) {
    RK_LOGE("ai disable channel fail, reason = %d", result);
    return RK_FAILURE;
}
result = RK_MPI_AI_Disable(aiDevId);
```

```
if (result != 0) {
    RK_LOGE("ai disable fail, reason = %d", result);
    return RK_FAILURE;
}
```

详细测试DEMO，请参考发布文件：test_mpi_ai.cpp。

API 参考

该功能模块为用户提供以下API:

- [RK MPI AI SetPubAttr](#): 设置 AI 设备属性。
- [RK MPI AI GetPubAttr](#): 获取 AI 设备属性。
- [RK MPI AI Enable](#): 启用 AI 设备。
- [RK MPI AI Disable](#): 禁用 AI 设备。
- [RK MPI AI EnableChn](#): 启用 AI 通道。
- [RK MPI AI DisableChn](#): 禁用 AI 通道。
- [RK MPI AI GetFrame](#): 获取音频帧。
- [RK MPI AI EnableReSmp](#): 启用 AI 重采样。
- [RK MPI AI DisableReSmp](#): 禁用 AI 重采样。
- [RK MPI AI ReleaseFrame](#): 释放音频帧。
- [RK MPI AI SetChnParamt](#): 获取 AI 通道参数。
- [RK MPI AI GetChnParam](#): 获取 AI 通道参数。
- [RK MPI AI SetTrackMode](#): 设置声道模式。
- [RK MPI AI GetTrackMode](#): 获取声道模式。
- [RK MPI AI QueryFileStatus](#): 查询音频输出通道是否处于存文件的状态。
- [RK MPI AI SaveFile](#): 开启音频输出保存文件功能。
- [RK MPI AI ClrPubAttr](#): 清除 AI 设备属性。

RK_MPI_AI_SetPubAttr

【描述】

设置 AI 设备（声卡）属性。

【语法】

RK_S32 RK_MPI_AI_SetPubAttr([AUDIO_DEV](#) AiDevId, const [AIO_ATTR_S](#) *pstAttr);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入
pstAttr	AI设备属性指针。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

音频输入的属性包括输入数据的格式，和输入设备（声卡）属性：采样率、声道数、采样精度、buffer大小、每帧的采样点数等，这些属性与codec配置一致，即可成功打开声卡。

- 在设置声卡设备属性之前需要保证AI处于禁用状态，如果处于启用状态则需要首先禁用AI声卡设备。
- 采样率
采样率是指一秒钟的采样率点数。设置声卡设备的采样率，要确认声卡设备是否支持所需要设定的采样率。
- 通道数目
指当前输入设备的 AI 功能的通道数目。需要与对接的Audio Codec的配置保持一致。rk809-codec仅支持2路录制。

RK_MPI_AI_GetPubAttr

【描述】
获取 AI 设备属性。

【语法】

RK_S32 RK_MPI_AI_GetPubAttr([AUDIO_DEV](#) AiDevId, [AIO_ATTR_S](#) *pstAttr);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入
pstAttr	AI设备属性指针。	输出

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 获取的属性为前一次配置的属性。
- 如果从未配置过属性，则返回属性未配置的错误。

RK_MPI_AI_Enable

【描述】
启用AI设备。

【语法】

RK_S32 RK_MPI_AI_Enable([AUDIO_DEV](#) AiDevId);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 要求在启用前配置 AI设备属性，否则会返回属性未配置的错误。
- 如果 AI设备已经启用，重复启用，则返回正在使用中的错误。

RK_MPI_AI_Disable

【描述】

禁用AI设备。

【语法】

RK_S32 RK_MPI_AI_Disable([AUDIO_DEV](#) AiDevId);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 如果 AI 设备未启用，则返回未启用的错误码。
- 禁用 AI 设备前必须先禁用设备下所有 AI 通道。
- 声卡设备使用完成后，必须关闭声卡设备。

RK_MPI_AI_EnableChn

【描述】

启用AI通道。

【语法】

RK_S32 RK_MPI_AI_EnableChn([AUDIO_DEV](#) AiDevId, [AI_CHN](#) AiChn);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入
AiChn	音频输入通道号。 取值范围：[0, AI_MAX_CHN_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 启用AI通道前，必须先启用其所属的AI设备，否则返回设备未启动的错误码。

RK_MPI_AI_DisableChn

【描述】

禁用 AI 通道。

【语法】

RK_S32 RK_MPI_AI_DisableChn([AUDIO_DEV](#) AiDevId, [AI_CHN](#) AiChn);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入
AiChn	音频输入通道号。 取值范围：[0, AI_MAX_CHN_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 通道使用完成后，必须调用关闭AI通道。

RK_MPI_AI_GetFrame

【描述】

获取音频帧。

【语法】

RK_S32 RK_MPI_AI_GetFrame([AUDIO_DEV](#) AiDevId, [AI_CHN](#) AiChn, [AUDIO_FRAME_S](#) *pstFrm,

[AEC_FRAME_S](#) *pstAecFrm, RK_S32 s32MilliSec);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入
AiChn	音频输入通道号。 取值范围：[0, AI_MAX_CHN_NUM)。	输入
pstFrm	音频帧结构体指针。	输出
pstAecFrm	回声抵消参考帧结构体指针。	输出
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，其值为 错误码 。

【注意】

- 目前AI的回声抵消功能还未实现，pstAecFrm置为空。
- 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。
- 获取音频帧数据前，必须先开启对应的AI 通道和声卡设备。

RK_MPI_AI_ReleaseFrame

【描述】

释放音频帧。

【语法】

RK_S32 RK_MPI_AI_ReleaseFrame([AUDIO_DEV](#) AiDevId, [AI_CHN](#) AiChn, const [AUDIO_FRAME_S](#) *pstFrm, const [AEC_FRAME_S](#) *pstAecFrm);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入
AiChn	音频输入通道号。 取值范围：[0, AI_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

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 如果不需要释放回声抵消参考帧，pstAecFrm置为NULL即可。

RK_MPI_AI_SetChnParam

【描述】

设置 AI 通道参数。

【语法】

RK_S32 RK_MPI_AI_SetChnParam([AUDIO_DEV](#) AiDevId, [AI_CHN](#) AiChn, const [AI_CHN_PARAM_S](#) *pstChnParam);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM]。	输入
AiChn	音频输入通道号。 取值范围：[0, AI_MAX_CHN_NUM]。	输入
pstChnParam	音频通道参数。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 通道参数目前只有一个成员变量，用于设置用户获取音频帧的缓存个数。默认为4个。

RK_MPI_AI_GetChnParam

【描述】

获取 AI 通道参数。

【语法】

RK_S32 RK_MPI_AI_GetChnParam([AUDIO_DEV](#) AiDevId, [AI_CHN](#) AiChn, [AI_CHN_PARAM_S](#) *pstChnParam);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入
AiChn	音频输入通道号。 取值范围：[0, AI_MAX_CHN_NUM)。	输入
pstChnParam	音频通道参数。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 无。

RK_MPI_AI_EnableReSmp

【描述】

启用AI重采样。

【语法】

RK_S32 RK_MPI_AI_EnableReSmp([AUDIO_DEV](#) AiDevId, [AI_CHN](#) AiChn, [AUDIO_SAMPLE_RATE_E](#) enOutSampleRate);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入
AiChn	音频输入通道号。 取值范围：[0, AI_MAX_CHN_NUM)。	输入
enOutSampleRate	音频重采样的输出采样率。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 在启用AI通道之后，调用此接口启用重采样功能。
- 允许重复启用重采样功能，但必须保证后配置的重采样输入采样率与之前配置的重采样输入采样率不一样。

- 在禁用AI通道后，如果重新启用AI通道，并使用重采样功能，需调用此接口重新启用重采样。
- AI 重采样的输入采样率必须与AI声卡设备属性配置的采样率不相同。

【举例】

以 AI 从录制 32K 到 8K 的重采样为例，配置如下：

```
/* dev attr of ai */
aiAttr.soundCard.channels = 2;
aiAttr.soundCard.sampleRate = 32000;
aiAttr.soundCard.bitwidth = AUDIO_BIT_WIDTH_16;

aiAttr.enBitwidth = AUDIO_BIT_WIDTH_16;
aiAttr.enSamplerate = AUDIO_SAMPLE_RATE_8000;
aiAttr.enSoundmode = AUDIO_SOUND_MODE_STEREO;
aiAttr.u32FrmNum = 4;
aiAttr.u32PtNumPerFrm = 1024;
RK_MPI_AI_EnableReSmp(AiDev, AiChn, AUDIO_SAMPLE_RATE_8000);
```

RK_MPI_AI_DisableReSmp

【描述】

禁用 AI 重采样。

【语法】

RK_S32 RK_MPI_AI_DisableReSmp([AUDIO_DEV](#) AiDevId, [AI_CHN](#) AiChn);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM]。	输入
AiChn	音频输出通道号。 取值范围：[0, AI_MAX_CHN_NUM]。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 不再使用 AI 重采样功能的话，应该调用此接口将其禁用。

RK_MPI_AI_SetTrackMode

【描述】

设置 AI 通道模式。

【语法】

RK_S32 RK_MPI_AI_SetTrackMode([AUDIO_DEV](#) AiDevId, [AUDIO_TRACK_MODE_E](#) enTrackMode);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入
enTrackMode	音频声道模式。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 在 AI 设备成功启用后再调用此接口。

RK_MPI_AI_GetTrackMode

【描述】

获取 AI 声道模式。

【语法】

```
RK_S32 RK_MPI_AI_GetTrackMode(AUDIO\_DEV AiDevId, AUDIO\_TRACK\_MODE\_E *penTrackMode);
```

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
penTrackMode	音频输入声道模式指针。	输出

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 在 AI 设备成功启用后再调用此接口。

RK_MPI_AI_ClrPubAttr

【描述】

清空Pub属性。

【语法】

```
RK_S32 RK_MPI_AI_ClrPubAttr(AUDIO\_DEV AiDevId);
```

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入

【返回值】

返回值	描述
0	成功。
非0	失败，其值为 错误码 。

【注意】

- 清除设备属性前，需要先停止设备。

RK_MPI_AI_SaveFile

【描述】

开启音频输入保存文件功能。

【语法】

```
RK_S32 RK_MPI_AI_SaveFile(AUDIO\_DEV AiDevId, AI\_CHN AiChn, const AUDIO\_SAVE\_FILE\_INFO\_S *pstSaveFileInfo);
```

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入
AiChn	音频输出通道号。 取值范围：[0, AI_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，其值为 错误码 。

【注意】

- 此接口仅用于 AI 非绑定模式下dump AI 从声卡录制的PCM文件。

RK_MPI_AI_QueryFileStatus

【描述】

查询音频输入通道是否处于存文件的状态。

【语法】

RK_S32 RK_MPI_AI_QueryFileStatus([AUDIO_DEV](#) AiDevId, [AI_CHN](#) AiChn, [AUDIO_FILE_STATUS_S](#)* pstFileStatus);

【参数】

参数名	描述	输入/输出
AiDevId	音频设备号。 取值范围：[0, AI_DEV_MAX_NUM)。	输入
AiChn	音频输入通道号。 取值范围：[0, AI_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，其值为 错误码 。

【注意】

- 此接口用于查询音频输入通道是否处于存文件的状态，当用户调用[RK_MPI_AI_SaveFile](#) 存储文件后，可调用此接口查询存储的文件是否达到了指定的大小，如果 pstFileStatus 的 bSaving 为 RK_TRUE，说明还没有达到指定大小，为RK_FALSE 则已经达到指定大小。

数据类型

AUDIO_DEV

【说明】

定义 AI 设备句柄。

【定义】

```
typedef RK_S32 AUDIO_DEV;
```

AI_CHN

【说明】

定义 AI 通道。

【定义】

```
typedef RK_S32 AI_CHN;
```

AI_MAX_CHN_NUM

【说明】

定义音频输出通道的最大个数。

【定义】

```
#define AI_MAX_CHN_NUM 1
```

AI_DEV_MAX_NUM

【说明】

定义音频输出设备的最大个数。

【定义】

```
#define AI_DEV_MAX_NUM 1
```

AIO_ATTR_S

【说明】

定义音频输入输出设备属性结构体。

【定义】

```
typedef struct rkAIO_ATTR_S {
    // params of sound card
    AIO_SOUND_CARD      soundCard;
    // sample rate to open sound card
    AUDIO_SAMPLE_RATE_E enSamplerate;
    // bitwidth
    AUDIO_BIT_WIDTH_E   enBitwidth;
    // momo or steror
    AUDIO_SOUND_MODE_E  enSoundmode;
    /* expand 8bit to 16bit,use AI_EXPAND(only valid for AI 8bit),
     * use AI_CUT(only valid for extern Codec for 24bit)
     */
    RK_U32               u32EXFlag;
    /* frame num in buf[2,MAX_AUDIO_FRAME_NUM] */
    RK_U32               u32FrmNum;
    /*
     * point num per frame (80/160/240/320/480/1024/2048)
     * (ADPCM IMA should add 1 point, AMR only support 160)
     */
    RK_U32               u32PtNumPerFrm;
    RK_U32               u32ChnCnt;      /* channle number on FS, valid
value:1/2/4/8 */
    /*
     * name of sound card, if it is setted, we will
     * using it to open sound card, otherwise, use
     * the index of device to open sound card
     */
    RK_U8               u8CardName[64];
} AIO_ATTR_S;
```

【成员】

成员名称	描述
soundCard	声卡属性参数。
enSamplerate	音频采样率。对于AI，此采样率是用户获取数据的采样率，对于AO，是用户发送数据的采样率。
enBitwidth	采样精度。对于AI，此采样率是用户获取数据的采样率，对于AO，是用户发送数据的采样率。
enSoundmode	音频声道模式。对于AI，此采样率是用户获取的采样率，对于AO，是用户发送数据的采样率。
u32EXFlag	设置默认为0。暂不支持。
u32FrmNum	处理完一个buffer数据所需的硬件中断次数。范围[2, 300]，默认为4。
u32PtNumPerFrm	每次硬件中断处理音频数据的帧数。范围[128, 256, 512, 1024, 1536, 2048]，默认为1024。
u32ChnCnt	支持的通道数目。
u8CardName[64]	声卡名字。

【注意事项】

- soundCard：配置打开声卡的参数。需要设置声卡驱动支持的采样率，声道数和采样精度。必须配置。
- enSamplerate：对于AI，表示用户获取数据的采样率，比如打开声卡16K，需要读取8K的数据，此时配置enSamplerate为8K，需要调用[RK MPI AI EnableReSmp](#)使能重采样，把16k数据转换成8k读取。对于AO，表示发送数据的采样率，比如用户发送16k采样率的数据，设置enSamplerate为16k，此时声卡只支持打开48k，需要调用[RK MPI AI EnableReSmp](#)使能重采样，把16k数据转换成48k送去播放。必须配置。
- enBitwidth：原理同enSamplerate，设置数据的采样精度。必须配置。
- enSoundmode：原理同enSamplerate，设置数据的声道数。必须配置。
- u32FrmNum：每次DMA运输处理音频数据的帧数。如果周期大小设定得较大，则单次处理的数据较多，这意味着单位时间内硬件中断的次数较少，CPU也就有更多时间处理其他任务，功耗也更低，但这样也带来一个显著的弊端——数据处理的时延会增大。可按默认配置。
- u8CardName：外部可配需要打开的声卡，代码内部有默认配置。

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_SOUND_MODE_E

【说明】

定义音频声道模式。

【定义】

```
typedef enum rkAIO_SOUND_MODE_E {  
    AUDIO_SOUND_MODE_MONO    = 0, /*mono*/  
    AUDIO_SOUND_MODE_STEREO  = 1, /*stereo*/  
    AUDIO_SOUND_MODE_BUTT  
} AUDIO_SOUND_MODE_E;
```

【成员】

成员名称	描述
AUDIO_SOUND_MODE_MONO	单声道。
AUDIO_SOUND_MODE_STEREO	双声道。

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。

AIO_SOUND_CARD

【说明】
定义音频码流结构体。

【定义】

```
typedef struct rkAIO_SOUND_CARD {
    RK_U32  channels;
    RK_U32  sampleRate;
    AUDIO_BIT_WIDTH_E bitwidth;
} AIO_SOUND_CARD;
```

【成员】

成员名称	描述
channels	声道数。
sampleRate	采样率。
bitWidth	采样精度。

【注意事项】

- 设置打开声卡的参数，需要是声卡驱动支持的参数。不然会打开声卡失败，报错。

AUDIO_SAMPLE_RATE_E

【说明】
定义音频采样率。

【定义】

```
typedef enum rkAUDIO_SAMPLE_RATE_E {
    AUDIO_SAMPLE_RATE_DISABLE = 0,
    AUDIO_SAMPLE_RATE_8000    = 8000,    /* 8K samplerate*/
    AUDIO_SAMPLE_RATE_12000   = 12000,   /* 12K samplerate*/
    AUDIO_SAMPLE_RATE_11025   = 11025,   /* 11.025K samplerate*/
    AUDIO_SAMPLE_RATE_16000   = 16000,   /* 16K samplerate*/
    AUDIO_SAMPLE_RATE_22050   = 22050,   /* 22.050K samplerate*/
    AUDIO_SAMPLE_RATE_24000   = 24000,   /* 24K samplerate*/
    AUDIO_SAMPLE_RATE_32000   = 32000,   /* 32K samplerate*/
    AUDIO_SAMPLE_RATE_44100   = 44100,   /* 44.1K samplerate*/
    AUDIO_SAMPLE_RATE_48000   = 48000,   /* 48K samplerate*/
    AUDIO_SAMPLE_RATE_64000   = 64000,   /* 64K samplerate*/
    AUDIO_SAMPLE_RATE_96000   = 96000,   /* 96K samplerate*/
    AUDIO_SAMPLE_RATE_BUTT,
} AUDIO_SAMPLE_RATE_E;
```

AUDIO_TRACK_MODE_E

【说明】
音频设备声道模式类型。

【定义】

```
typedef enum rkAUDIO_TRACK_MODE_E {  
    AUDIO_TRACK_NORMAL          = 0,  
    AUDIO_TRACK_BOTH_LEFT      = 1,  
    AUDIO_TRACK_BOTH_RIGHT     = 2,  
    AUDIO_TRACK_EXCHANGE       = 3,  
    AUDIO_TRACK_MIX             = 4,  
    AUDIO_TRACK_LEFT_MUTE      = 5,  
    AUDIO_TRACK_RIGHT_MUTE     = 6,  
    AUDIO_TRACK_BOTH_MUTE      = 7,  
  
    AUDIO_TRACK_BUTT  
} AUDIO_TRACK_MODE_E;
```

【成员】

成员名称	描述
AUDIO_TRACK_NORMAL	正常模式，不做处理。
AUDIO_TRACK_BOTH_LEFT	两个声道全部为左声道声音。
AUDIO_TRACK_BOTH_RIGHT	两个声道全部为右声道声音。
AUDIO_TRACK_EXCHANGE	左右声道数据互换，左声道为右声道声音，右声道为左声道声音。
AUDIO_TRACK_MIX	左右两个声道输出为左右声道相加（混音）。
AUDIO_TRACK_LEFT_MUTE	左声道静音，右声道播放原右声道声音。
AUDIO_TRACK_RIGHT_MUTE	右声道静音，左声道播放原左声道声音。
AUDIO_TRACK_BOTH_MUTE	左右声道均静音。

AUDIO_SAVE_FILE_INFO_S

【说明】
定义音频保存文件功能配置信息结构体。

【定义】

```
typedef struct rkAUDIO_SAVE_FILE_INFO_S {  
    RK_BOOL      bCfg;  
    RK_CHAR      filePath[MAX_AUDIO_FILE_PATH_LEN];  
    RK_CHAR      fileName[MAX_AUDIO_FILE_NAME_LEN];  
    RK_U32       u32FileSize; /*in KB*/  
} AUDIO_SAVE_FILE_INFO_S;
```

【成员】

成员名称	描述
bCfg	配置使能开关。
aFilePath	音频文件保存路径。
aFileName	音频文件保存名称。
u32FileSize	文件大小，单位byte。

AUDIO_FILE_STATUS_S

【说明】
定义音频文件保存状态结构体。

【定义】

```
typedef struct rkAUDIO_FILE_STATUS_S {  
    RK_BOOL      bSaving;  
} AUDIO_FILE_STATUS_S;
```

【成员】

成员名称	描述
bSaving	文件是否处于保存状态。

错误码

音频输入API错误码如下所示。

错误代码	宏定义	描述
0xA00A8001	RK_ERR_AI_INVALID_DEVID	音频输入设备号无效
0xA00A8002	RK_ERR_AI_INVALID_CHNID	音频输入通道号无效
0xA00A8003	RK_ERR_AI_ILLEGAL_PARAM	音频输入参数设置无效
0xA00A8005	RK_ERR_AI_NOT_ENABLED	音频输入设备或通道没使能
0xA00A8006	RK_ERR_AI_NULL_PTR	输入空指针错误
0xA00A8007	RK_ERR_AI_NOT_CONFIG	音频输入设备属性未设置
0xA00A8008	RK_ERR_AI_NOT_SUPPORT	操作不被支持
0xA00A8009	RK_ERR_AI_NOT_PERM	操作不允许
0xA00A800C	RK_ERR_AI_NOMEM	分配内存失败
0xA00A800D	RK_ERR_AI_NOBUF	音频输入缓存不足
0xA00A800E	RK_ERR_AI_BUF_EMPTY	音频输入缓存为空
0xA00A800F	RK_ERR_AI_BUF_FULL	音频输入缓存为满
0xA00A8010	RK_ERR_AI_SYS_NOTREADY	音频输入系统未初始化
0xA00A8012	RK_ERR_AI_BUSY	音频输入系统忙
0xA00A8041	RK_ERR_AI_VQE_ERR	AO VQE 处理错误