

音频输出

文件标识：RK-SYS1-MPI-AO

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

本文档主要介绍AO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	2021-1-4	初始版本
v0.2.0	周弟东	2021-1-21	完善数据类型释义

目录

音频输出

目录

基本概念

举例

API 参考

- RK_MPI_AO_SetPubAttr
- RK_MPI_AO_GetPubAttr
- RK_MPI_AO_Enable
- RK_MPI_AO_Disable
- RK_MPI_AO_EnableChn
- RK_MPI_AO_DisableChn
- RK_MPI_AO_SendFrame
- RK_MPI_AO_EnableReSmp
- RK_MPI_AO_DisableReSmp
- RK_MPI_AO_PauseChn
- RK_MPI_AO_ResumeChn
- RK_MPI_AO_ClearChnBuf
- RK_MPI_AO_QueryChnStat
- RK_MPI_AO_SetTrackMode
- RK_MPI_AO_GetTrackMode
- RK_MPI_AO_SetVolume
- RK_MPI_AO_GetVolume
- RK_MPI_AO_SetMute
- RK_MPI_AO_GetMute
- RK_MPI_AO_SaveFile
- RK_MPI_AO_QueryFileStatus
- RK_MPI_AO_ClrPubAttr
- RK_MPI_AO_WaitEos

数据类型

- AUDIO_DEV
- AO_CHN
- AO_MAX_CHN_NUM
- AO_DEV_MAX_NUM
- MAX_AUDIO_FILE_PATH_LEN
- MAX_AUDIO_FILE_NAME_LEN
- AIO_ATTR_S
- AUDIO_FRAME_S
- AUDIO_SOUND_MODE_E
- AUDIO_BIT_WIDTH_E
- AUDIO_STREAM_S
- AUDIO_SAMPLE_RATE_E
- AO_CHN_STATE_S

AUDIO_TRACK_MODE_E
AUDIO_FADE_RATE_E
AUDIO_FADE_S
AUDIO_SAVE_FILE_INFO_S
AUDIO_FILE_STATUS_S
错误码

基本概念

音频输出（AO）主要实现启用音频输出设备、发送音频帧到输出通道等功能。

举例

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

aoAttr.soundCard.channels = 2;
aoAttr.soundCard.sampleRate = 48000;
aoAttr.soundCard.bitwidth = AUDIO_BIT_WIDTH_16;

aoAttr.enBitwidth = AUDIO_BIT_WIDTH_16;
aoAttr.enSamplerate = AUDIO_SAMPLE_RATE_44100;
aoAttr.enSoundmode = AUDIO_SOUND_MODE_STEREO;
aoAttr.u32FrmNum = 4;
aoAttr.u32PtNumPerFrm = 1024;

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

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

result = RK_MPI_AO_EnableChn(aoDevId, aoChn);
if (result != 0) {
    RK_LOGE("ao enable channel fail, reason = %d", result);
    return RK_FAILURE;
}

result = RK_MPI_AO_EnableReSmp(aoDevId, aoChn, AUDIO_SAMPLE_RATE_44100);
if (result != 0) {
    RK_LOGE("ao enable channel fail, reason = %d", result);
    return RK_FAILURE;
}

MB_BLK blk = RK_MPI_MB_GetMB(pool_id, 2048, RK_TRUE);
```

```

frame.pMbBlk = blk;
frame.u32Len = size;
frame.u64TimeStamp = timeStamp;
frame.enBitwidth = AUDIO_BIT_WIDTH_16;
frame.enSoundMode = AUDIO_SOUND_MODE_STEREO;

result = RK_MPI_AO_SendFrame(aoDevId, aoChn, &frame, s32MilliSec);
if (result < 0) {
    RK_MPI_MB_ReleaseMB(blk);
}

RK_MPI_AO_WaitEos(aoDevId, chn, -1);

RK_S32 result = RK_MPI_AO_DisableChn(aoDevId, aoDevId);
if (result != 0) {
    RK_LOGE("ao disable channel fail, reason = %d", result);
    return RK_FAILURE;
}
result = RK_MPI_AO_Disable(aoDevId);
if (result != 0) {
    RK_LOGE("ao disable fail, reason = %d", result);
    return RK_FAILURE;
}

```

详细测试DEMO，请参考发布文件：test_mpi_ao.cpp。

API 参考

该功能模块为用户提供以下API:

- [RK MPI AO SetPubAttr](#): 设置 AO 设备属性。
- [RK MPI AO GetPubAttr](#): 获取 AO 设备属性。
- [RK MPI AO Enable](#): 启用 AO 设备。
- [RK MPI AO Disable](#): 禁用 AO 设备。
- [RK MPI AO EnableChn](#): 启用 AO 通道。
- [RK MPI AO DisableChn](#): 禁用 AO 通道。
- [RK MPI AO SendFrame](#): 发送 AO 音频帧。
- [RK MPI AO EnableReSmp](#): 启用 AO 重采样。
- [RK MPI AO DisableReSmp](#): 禁用 AO 重采样。
- [RK MPI AO PauseChn](#): 暂停 AO 通道。
- [RK MPI AO ResumeChn](#): 恢复 AO 通道。
- [RK MPI AO ClearChnBuf](#): 清除 AO 通道中当前的音频数据缓存。
- [RK MPI AO QueryChnStat](#): 查询 AO 通道中当前的音频数据缓存状态。
- [RK MPI AO SetTrackMode](#): 置 AO 设备声道模式。
- [RK MPI AO GetTrackMode](#): 获取 AO 设备声道模式。
- [RK MPI AO SetVolume](#): 设置 AO 设备音量大小。
- [RK MPI AO GetVolume](#): 获取 AO 设备音量大小。
- [RK MPI AO SetMute](#): 设置 AO 设备静音状态。
- [RK MPI AO GetMute](#): 获取 AO 设备静音状态。
- [RK MPI AO SaveFile](#): 开启音频输出保存文件功能。
- [RK MPI AO QueryFileStatus](#): 查询音频输出通道是否处于存文件的状态。
- [RK MPI AO ClrPubAttr](#): 清除 AO 设备属性。
- [RK MPI AO WaitEos](#): 等待指定设备和通道播放完成。

RK_MPI_AO_SetPubAttr

【描述】

设置AO设备（声卡）参数。

【语法】

RK_S32 RK_MPI_AO_SetPubAttr([AUDIO_DEV](#) AoDevId, const [AIO_ATTR_S](#) *pstAttr);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
pstAttr	音频输出设备属性。	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】

- 在设置声卡设备属性之前需要保证AO处于禁用状态，如果处于启用状态则需要首先禁用AO声卡设备。
- 必须设置声卡设备的支持的采样率，声道数。其他参数可选配置，已设默认值，参见[AIO_ATTR_S](#)结构体。
- LINUX系统声卡驱动配置参见/etc/asound.conf文件，LINUX rk356x声卡默认配置使用pcm.card0。

RK_MPI_AO_GetPubAttr

【描述】

获取AO设备属性。

【语法】

RK_S32 RK_MPI_AO_GetPubAttr([AUDIO_DEV](#) AoDevId, [AIO_ATTR_S](#) *pstAttr);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
pstAttr	音频输出设备属性。	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_AO_Enable

【描述】

启用AO设备。

【语法】

RK_S32 RK_MPI_AO_Enable([AUDIO_DEV](#) AoDevId);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_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 AO设备属性，否则会返回属性未配置的错误。
- 如果 AO设备已经启用，重复启用，则返回正在使用中的错误。

RK_MPI_AO_Disable

【描述】

禁用AO设备。

【语法】

```
RK_S32 RK_MPI_AO_Disable(AUDIO\_DEV AoDevId);
```

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_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 AO 设备未启用，则返回未启用的错误码。
- 禁用 AO 设备前必须先禁用设备下所有 AO 通道。
- 声卡设备使用完成后，必须关闭声卡设备。

RK_MPI_AO_EnableChn

【描述】

启用AO通道。

【语法】

RK_S32 RK_MPI_AO_EnableChn([AUDIO_DEV](#) AoDevId, [AO_CHN](#) AoChn);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_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AO通道前，必须先启用其所属的AO设备，否则返回设备未启动的错误码。

RK_MPI_AO_DisableChn

【描述】

禁用 AO 通道。

【语法】

RK_S32 RK_MPI_AO_DisableChn([AUDIO_DEV](#) AoDevId, [AO_CHN](#) AoChn);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_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AO通道。

RK_MPI_AO_SendFrame

【描述】

发送AO音频帧。

【语法】

RK_S32 RK_MPI_AO_SendFrame([AUDIO_DEV](#) AoDevId, [AO_CHN](#) AoChn, const [AUDIO_FRAME_S](#) *pstData, RK_S32 s32MilliSec);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_MAX_CHN_NUM)。	输入
pstData	音频帧结构体指针。	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，其值为错误码。

【注意】

- 该接口用于用户主动发送音频帧至 AO 输出，如果 AO 通道已经通过系统绑定 (RK_MPI_SYS_Bind) 接口与 AI 或 ADEC 绑定，不需要也不建议调此接口。

- 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。
- 调用该接口发送音频帧到 AO 输出时，必须先开启对应的 AO 通道和声卡设备。
- 音频申请内存的方式建议尽量使用malloc方式申请。

RK_MPI_AO_EnableReSmp

【描述】

启用AO重采样。

【语法】

RK_S32 RK_MPI_AO_EnableReSmp([AUDIO_DEV](#) AoDevId, [AO_CHN](#) AoChn, [AUDIO_SAMPLE_RATE_E](#) enInSampleRate);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_MAX_CHN_NUM)。	输入
enInSampleRate	音频重采样的输入采样率。	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】

- 应该在启用AO通道之后，绑定AO通道之前，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AO通道后，如果重新启用AO通道，并使用重采样功能，需调用此接口重新启用重采样。
- AO重采样的输入采样率必须与AO设备属性配置的采样率不相同。

【举例】

以 ADEC 到 AO 的解码回放 8K 到 32K 重采样为例，配置如下：

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

aoAttr.enBitwidth = AUDIO_BIT_WIDTH_16;
aoAttr.enSamplerate = AUDIO_SAMPLE_RATE_8000;
aoAttr.enSoundmode = AUDIO_SOUND_MODE_MONO;

RK_MPI_AO_SetPubAttr(aoDevId, &aoAttr);
RK_MPI_AO_EnableReSmp(AoDev, AoChn, AUDIO_SAMPLE_RATE_8000);
```

RK_MPI_AO_DisableReSmp

【描述】

禁用 AO 重采样。

【语法】

RK_S32 RK_MPI_AO_DisableReSmp([AUDIO_DEV](#) AoDevId, [AO_CHN](#) AoChn);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_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 AO 重采样功能的话，应该调用此接口将其禁用。

RK_MPI_AO_PauseChn

【描述】

暂停 AO 通道。

【语法】

RK_S32 RK_MPI_AO_PauseChn([AUDIO_DEV](#) AoDevId, [AO_CHN](#) AoChn);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_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】

- AO 通道暂停后，如果绑定的 ADEC 通道继续向此通道发送音频帧数据，发送的音频帧数据将会被阻塞。而如果绑定的 AI 通道继续向此通道发送音频帧数据，在通道缓冲未满的情况下则将音频帧放入缓冲区，在满的情况下则将音频帧丢弃。
- AO 通道为禁用状态时，不允许调用此接口暂停 AO 通道。

RK_MPI_AO_ResumeChn

【描述】

启用 AO 重采样。

【语法】

RK_S32 RK_MPI_AO_ResumeChn([AUDIO_DEV](#) AoDevId, [AO_CHN](#) AoChn);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_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】

- AO 通道暂停后可以通过调用此接口重新恢复。
- AO 通道为暂停状态或使能状态下，调用此接口返回成功；否则调用将返回错误。

RK_MPI_AO_ClearChnBuf

【描述】

清除 AO 通道中当前的音频数据缓存。

【语法】

RK_S32 RK_MPI_AO_ClearChnBuf([AUDIO_DEV](#) AoDevId, [AO_CHN](#) AoChn);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_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】

- 在 AO 通道成功启用后再调用此接口。
- 为完全清除解码回放通路上所有缓存数据，此接口还应该与[RK_MPI_ADEC_ClearChnBuf](#) 接口配合使用。

RK_MPI_AO_QueryChnStat

【描述】

查询 AO 通道中当前的音频数据缓存状态。

【语法】

RK_S32 RK_MPI_AO_QueryChnStat([AUDIO_DEV](#) AoDevId, [AO_CHN](#) AoChn, [AO_CHN_STATE_S](#) *pstStatus);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_MAX_CHN_NUM)。	输入
pstStatus	缓存状态结构体指针。	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】

- 在 AO 通道成功启用后再调用此接口。

RK_MPI_AO_SetTrackMode

【描述】

设置 AO 设备声道模式。

【语法】

RK_S32 RK_MPI_AO_SetTrackMode([AUDIO_DEV](#) AoDevId, [AUDIO_TRACK_MODE_E](#) enTrackMode);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
enTrackMode	音频设备声道模式。	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】

- [AUDIO_TRACK_MODE_E](#)定义参见rk_comm_aio.h。
- 在 AO 通道成功启用后再调用此接口。

RK_MPI_AO_GetTrackMode

【描述】

获取 AO 设备声道模式。

【语法】

RK_S32 RK_MPI_AO_GetTrackMode([AUDIO_DEV](#) AoDevId, [AUDIO_TRACK_MODE_E](#)* penTrackMode);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
penTrackMode	音频设备声道模式指针。	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 AO 通道成功启用后再调用此接口。

RK_MPI_AO_SetVolume

【描述】

设置 AO 设备音量大小。

【语法】

RK_S32 RK_MPI_AO_SetVolume([AUDIO_DEV](#) AoDevId, RK_S32 s32VolumeDb);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
s32VolumeDb	音频设备音量大小。	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 AO 通道成功启用后再调用此接口。

RK_MPI_AO_GetVolume

【描述】

获取 AO 设备音量大小。

【语法】

```
RK_S32 RK_MPI_AO_GetVolume([AUDIO_DEV] (###AUDIO_DEV) AoDevId, RK_S32 *ps32VolumeDb);
```

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM]。	输入
ps32VolumeDb	音频设备音量大小指针。	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 AO 通道成功启用后再调用此接口。

RK_MPI_AO_SetMute

【描述】

设置 AO 设备静音状态。

【语法】

RK_S32 RK_MPI_AO_SetMute([AUDIO_DEV](#) AoDevId, RK_BOOL bEnable, const [AUDIO_FADE_S](#) *pstFade);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM]。	输入
bEnable	音频设备是否启用静音。RK_TRUE：启用静音功能；RK_FALSE：关闭静音功能。	输入
pstFade	淡入淡出结构体指针。	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】

- 在 AO 通道成功启用后再调用此接口。
- 调用此接口时，用户可以选择是否使用淡入淡出功能，如果不使用淡入淡出则将结构体指针赋为空即可。

RK_MPI_AO_GetMute

【描述】

获取 AO 设备音量大小。

【语法】

RK_S32 RK_MPI_AO_GetVolume([AUDIO_DEV](#) AoDevId, RK_S32 *ps32VolumeDb);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
pbEnable	音频设备静音状态指针。	输出
pstFade	淡入淡出结构体指针。	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】

- 在 AO 通道成功启用后再调用此接口。

RK_MPI_AO_SaveFile

【描述】

开启音频输出保存文件功能。

【语法】

RK_S32 RK_MPI_AO_SaveFile([AUDIO_DEV](#) AoDevId, [AO_CHN](#) AoChn, [AUDIO_SAVE_FILE_INFO_S](#)* pstSaveFileInfo);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_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】

- 在 AO 通道成功启用后再调用此接口。

RK_MPI_AO_QueryFileStatus

【描述】

查询音频输出通道是否处于存文件的状态。

【语法】

```
RK_S32 RK_MPI_AO_QueryFileStatus(AUDIO\_DEV AoDevId, AO\_CHN AoChn,
AUDIO\_FILE\_STATUS\_S* pstFileStatus);
```

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_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出通道是否处于存文件的状态。

RK_MPI_AO_ClrPubAttr

【描述】

清除 AO 设备属性。

【语法】

```
RK_S32 RK_MPI_AO_ClrPubAttr(AUDIO\_DEV AoDevId);
```

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_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_AO_WaitEos

【描述】

等待指定设备和通道播放完成。

【语法】

RK_S32 RK_MPI_AO_WaitEos([AUDIO_DEV](#) AoDevId, [AO_CHN](#) AoChn, RK_S32 s32MilliSec);

【参数】

参数名	描述	输入/输出
AoDevId	音频设备号。 取值范围：[0, AO_DEV_MAX_NUM)。	输入
AoChn	音频输出通道号。 取值范围：[0, AO_MAX_CHN_NUM)。	输入
s32MilliSec	等待的超时时间, -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】

- 在 AO 通道成功启用后再调用此接口。
- 当在ADEC绑定AO之后，建议调用此接口使用阻塞模式，等待播放完成。
- s32MilliSec 的值必须大于等于-1，等于-1 时表示阻塞模式等待播放完成，等于 0 时采用非阻塞模式，大于 0 时，阻塞 s32MilliSec 毫秒后，则返回超时并报错。

数据类型

AUDIO_DEV

【说明】

定义 AO 设备句柄。

【定义】

```
typedef RK_S32 AUDIO_DEV;
```

AO_CHN

【说明】

定义 AO 通道。

【定义】

```
typedef RK_S32 AO_CHN;
```

AO_MAX_CHN_NUM

【说明】

定义音频输出通道的最大个数。

【定义】

```
#define AO_MAX_CHN_NUM 3
```

AO_DEV_MAX_NUM

【说明】

定义音频输出设备的最大个数。

【定义】

```
#define AO_DEV_MAX_NUM 2
```

MAX_AUDIO_FILE_PATH_LEN

【说明】

音频保存文件的路径的最大长度限制。

【定义】

```
#define MAX_AUDIO_FILE_PATH_LEN 256
```

MAX_AUDIO_FILE_NAME_LEN

【说明】

音频保存文件的名称的最大长度限制。

【定义】

```
#define MAX_AUDIO_FILE_NAME_LEN 256
```

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 {  
    // 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
enSamplerate	音频采样率。
enBitwidth	音频采样精度。
enSoundmode	音频声道模式。
u32EXFlag	设置默认为0。
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	支持的声道数目。
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。
- 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。

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_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;
```

AO_CHN_STATE_S

【说明】

音频输出通道的数据缓存状态结构体。

【定义】

```
typedef struct rkAO_CHN_STATE_S {
    RK_U32    u32ChnTotalNum;    /* total number of channel buffer */
    RK_U32    u32ChnFreeNum;     /* free number of channel buffer */
    RK_U32    u32ChnBusyNum;     /* busy number of channel buffer */
} AO_CHN_STATE_S;
```

【成员】

成员名称	描述
u32ChnTotalNum	输出通道总的缓存块数。
u32ChnFreeNum	可用的空闲缓存块数。
u32ChnBusyNum	被占用缓存块数。

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_FADE_RATE_E

【说明】
定义音频输出设备淡入淡出速度类型。

【定义】


```
typedef enum rkAUDIO_FADE_RATE_E {
    AUDIO_FADE_RATE_1    = 0,
    AUDIO_FADE_RATE_2    = 1,
    AUDIO_FADE_RATE_4    = 2,
    AUDIO_FADE_RATE_8    = 3,
    AUDIO_FADE_RATE_16   = 4,
    AUDIO_FADE_RATE_32   = 5,
    AUDIO_FADE_RATE_64   = 6,
    AUDIO_FADE_RATE_128  = 7,

    AUDIO_FADE_RATE_BUTT

} AUDIO_FADE_RATE_E;
```

【成员】

成员名称	描述
AUDIO_FADE_RATE_1	1个采样点改变一次。
AUDIO_FADE_RATE_2	2个采样点改变一次。
AUDIO_FADE_RATE_4	4个采样点改变一次。
AUDIO_FADE_RATE_8	8个采样点改变一次。
AUDIO_FADE_RATE_16	16个采样点改变一次。
AUDIO_FADE_RATE_32	32个采样点改变一次。
AUDIO_FADE_RATE_64	64个采样点改变一次。
AUDIO_FADE_RATE_128	128个采样点改变一次。

AUDIO_FADE_S

【说明】

音频输出设备淡入淡出配置结构体。

【定义】

```
typedef struct rkAUDIO_FADE_S {
    RK_BOOL          bFade;
    AUDIO_FADE_RATE_E enFadeInRate;
    AUDIO_FADE_RATE_E enFadeOutRate;
} AUDIO_FADE_S;
```

【成员】

成员名称	描述
bFade	是否开启淡入淡出功能。RK_TRUE：开启淡入淡出功能。RK_FALSE：关闭淡入淡出功能。
enFadeInRate	音频输出设备音量淡入速度。
enFadeOutRate	音频输出设备音量淡出速度。

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。
filePath	音频文件保存路径。
fileName	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

音频输出API错误码如下所示。

错误代码	宏定义	描述
0xA00B8001	RK_ERR_AO_INVALID_DEVID	音频输出设备号无效
0xA00B8002	RK_ERR_AO_INVALID_CHNID	音频输出通道号无效
0xA00B8003	RK_ERR_AO_ILLEGAL_PARAM	音频输出参数设置无效
0xA00B8005	RK_ERR_AO_NOT_ENABLED	音频输出设备或通道没使能
0xA00B8006	RK_ERR_AO_NULL_PTR	输出空指针错误
0xA00B8007	RK_ERR_AO_NOT_CONFIG	音频输出设备属性未设置
0xA00B8008	RK_ERR_AO_NOT_SUPPORT	操作不被支持
0xA00B8009	RK_ERR_AO_NOT_PERM	操作不允许
0xA00B800C	RK_ERR_AO_NOMEM	系统内存不足
0xA00B800D	RK_ERR_AO_NOBUF	音频输出缓存不足
0xA00B800E	RK_ERR_AO_BUF_EMPTY	音频输出缓存为空
0xA00B800F	RK_ERR_AO_BUF_FULL	音频输出缓存为满
0xA00B8010	RK_ERR_AO_SYS_NOTREADY	音频输出系统未初始化
0xA00B8012	RK_ERR_AO_BUSY	音频输出系统忙
0xA00B8041	RK_ERR_AO_VQE_ERR	AO VQE 处理错误