

Rockchip Audio Algorithm Utils

前言

概述

本文主要介绍RockAA Utils测试用途和使用方法。

产品版本

芯片名称	内核版本
全系列	通用

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

目录

Rockchip Audio Algorithm Utils

1. 概述
 - 1.1 RockAA Utils功能描述
2. 使用场景和用法介绍
 - 2.1 文件离线通话对讲测试
 - 2.2 实时在线通话对讲测试
 - 2.3 文件离线播放音效测试
 - 2.4 实时在线播放音效测试
3. 更多细节用法的help信息

1. 概述

1.1 RockAA Utils功能描述

RockAA (Rockchip Audio Algorithm) Utils可以帮助客户在不依赖媒体框架的前提下，快捷地初步验证Rockchip音频相关算法效果。

RockAA Utils借鉴了tinycap和tinypplay的设计思路，主要由rockaa_capt和rockaa_play工具组成，通过不同的命令参数达到不同使用场景下的验证。它们均依赖公共库librockaa.so。并且，因为rockaa_capt主要用于通话测试，rockaa_play主要用于播放音效测试，依赖的音频算法库也有所不同，具体依赖关系如下：

```
rockaa_capt
├─ librockaa.so          # rockaa工具公共库
├─ libaec_bf_process.so  # 回声消除算法库
└─ librkaudio_common.so  # 音频算法公共库

rockaa_play
├─ librockaa.so          # rockaa工具公共库
└─ librkaudio_effect.so  # 音效算法库
```

2. 使用场景和用法介绍

目前适用的场景主要有：

- 通话对讲（Talk）
- 播放音效（Playback Effect）

2.1 文件离线通话对讲测试

大多数开发者调试音频算法前会涉及，即在结构确定好的量产机型上预先已采集好音频，用此音频作为固定音源输入，调整优化算法参数，直到输出符合预期的音频数据：

```
./rockaa_capt -I record.pcm -O out_talk.pcm -c 2 -C 1 -b 16 -r 16000 -Z 1 -F
config_aivqe.json
```

此log表示工作在文件离线通话对讲测试下：

```
[tagI] Capture via offline file process, in_file: record.wav
```

分解一下几个参数的主要用法：

"-I"作为离线文件模式或实时在线模式切换的唯一参数。如果包含该参数，表示工作在离线文件模式下，后面跟的"record.pcm"作为事先设备采集好的待处理音源。

```

"-O out_talk.pcm"表示指定音频算法后输出的pcm名称，为单通道，采样率和位宽与后面指定的"-b 16 -r 16000"保持一致。如果不指定，会默认生成以"TALK"开头的out_pcm;
"-c 2"表示总channel数，"-C 1"表示远端回采参考信号的channel数，近端MIC的channel数为它们的相差的通道数，如本例为："2-1=1"，为单MIC拾音场景；
"-b 16 -r 16000"表示音频除声道数外的另外两个基本PCM格式，16bit位深，16kHz采样率；
"-Z 1"表示summary debug mode，在结束测试后会一次性统计帧数，以及最小、最大、平均耗时和负载情况：
[tagI] == frames:63, period_time:16000us, cost_time min:657us max:1375us
avg:793us, load min:4.11% max:8.59% avg:4.96% ==
"-Z 0"表示不开启debug mode统计。"-Z 2"表示对每帧都进行耗时和负载统计。比如：
[tagI] frame[1] period_time: 16000 us, cost_time: 942 us, load: 5.89%
[tagI] frame[2] period_time: 16000 us, cost_time: 808 us, load: 5.05%
[tagI] frame[3] period_time: 16000 us, cost_time: 858 us, load: 5.36%
"-Z 3"表示"-Z 1"和"-Z 2"效果的结合，即每帧都打印负载，且结束时打印summary debug信息。
...
"-F config_aivqe.json"表示指定音频算法的json参数配置文件。如果没有指定，会直接退出，并有相应的log提示：
./rockaa_capt -I record.wav -O out_talk.pcm -c 2 -C 1 -b 16 -r 16000 -Z 1
input file: record.wav
[tagE] Can't support in_file: record.wav without conf file

```

2.2 实时在线通话对讲测试

如上所述，如果不包含"-I"参数，意味着先前离线文件测试的参数已达到较好的效果，将进入实时在线模式进行更接近用户效果的测试体验：

```

./rockaa_capt record.wav -O out_talk.pcm -D 0 -d 0 -c 2 -C 1 -b 16 -r 16000 -Z 1
-F config_aivqe.json

```

此log表示工作在实时在线通话对讲测试下：

```

[tagI] Capture via run-time process

```

大多数参数与上述的文件离线测试相同，这里再补充一下有些区别的参数用法：

```

"record.wav"此时作为音频算法处理前的wav文件格式保存；
"-D 0 -d 0"来自于TINYALSA和ALSA的概念，表示从card0和device0设备采集音频；
"-t 10"表示录制10s；
"-F config_aivqe.json"表示指定音频算法的json参数配置文件。如果没有指定，会工作在bypass capture模式下，效果和tinycap相同，并有相应的log提示。如：
./rockaa_capt record.wav -D 0 -d 0 -c 2 -C 1 -b 16 -r 16000 -Z 1
[tagI] Not specify conf_path and capture raw pcm directly

```

2.3 文件离线播放音效测试

测试播放音效场景下，开发者也可以先加上"-I"参数，使用已有的待测音源，先进行离线测试调参：

```

./rockaa_play -I speech_2ch.wav -O out_effect.pcm -p 256 -Z 1 -F
config_playeffect.bin

```

此log表示工作在文件离线播放音效测试下：

```
[tagI] Playback via offline file process, in_file: speech_2ch.wav
```

分解一下几个参数的主要用法：

"-I"这里依然作为离线文件模式或实时在线模式切换的唯一参数。如果包含该参数，表示工作在离线文件模式下，后面跟的"speech_2ch.wav"作为事先设备采集好的待处理音源。

"-O out_effect.pcm"表示指定音频算法后输出的pcm名称，为单通道，采样率和位宽与后面指定的"-b 16 -r 16000"保持一致。如果不指定，会默认生成以"EFFECTPLAY"开头的输出文件；

"-c -b -r"依旧表示基本PCM格式三要素，即声道数，位深，采样率，可以通过小写的"-i raw"强行指定PCM格式。本例为wav输入，自动解析并按wav格式播放；

"-p 256"表示每帧处理256个采样点，16kHz下就是 $256/16000=16\text{ms}$ 每帧。

"-Z 0 -Z 1 -Z 2 -Z 3"的效果和上述rockaa_capt的相同。

"-F config_playeffect.bin"表示指定音效配置文件。如果没有指定，会直接退出，并有相应的log提示：

```
./rockaa_play -I speech_2ch.wav -O out_effect.pcm -p 256 -Z 1  
[tagE] Can't support in_file: speech_2ch.wav without conf file
```

2.4 实时在线播放音效测试

如果先前离线文件测试的参数已达到较好的效果，去除"-I"参数，将进入实时在线音效播放测试：

```
./rockaa_play speech_2ch.wav -D 0 -d 0 -p 256 -Z 1 -F config_playeffect.bin
```

此log表示工作在实时在线播放音效测试下：

```
[tagI] Playback via run-time process
```

大部分参数功能与上述的相同，需要注意的是，如果这时候不指定"-F"参数，意味着bypass playback，效果和tinyplay相同并有相应的log提示。如：

```
./rockaa_play speech_2ch.wav -D 0 -d 0 -p 256 -Z 1  
[tagI] Not specify conf_path and playback raw pcm directly
```

3. 更多细节用法的help信息

rockaa_capt的help列表：

```
# ./rockaa_capt  
Usage: ./rockaa_capt [OPTION]...  
  
-h, help  
-v print current version  
-b, bits of samples  
-r, the sample rate for stream (default: 16000)  
-c, the number of channels (default: 2)  
-C, the number of far-end channels (default: 1)  
-p, the size of per-period (default: 256)  
-n, the number of periods (default: 4)
```

```
-t, capture time (default: UINT_MAX)
-a, enable DOA and return result (default: disabled)
-F, the path of json config file, without it capture raw pcm directly
-I, use offline input file mode, without capture via sound card
-Z, use debug mode, 0:disabled 1:summary 2:per-frame 3:summary+per-frame
(default: 0)
For example:
    ./rockaa_capt cap.wav -c 2 -C 1 -b 16 -r 16000 -t 10 -Z 1 -F
config_aivqe.json
```

rockaa_play的帮助列表:

```
# ./rockaa_play
usage: ./rockaa_play file.wav [options]
options:
-D | --card <card number>      The card to receive the audio
-d | --device <device number>  The device to receive the audio
-p | --period-size <size>      The size of the PCM's period
-n | --period-count <count>    The number of PCM periods
-i | --file-type <file-type>   The type of file to read (raw or wav)
-c | --channels <count>        The amount of channels per frame
-r | --rate <rate>             The amount of frames per second
-b | --bits <bit-count>        The number of bits in one sample
-f | --float                    The frames are in floating-point PCM
-M | --mmap                    Use memory mapped IO to play audio
-F | --conf-name                The parameters of configuration File, (Mandatory)
-I | --in-file                  Use offline input file mode, without playback via
sound card
-O | --out-file                 The output file name
-Z | --debug-mode               Use debug mode, 0:disabled 1:summary 2:per-frame
3:summary+per-frame (default: 0)
For example:
    ./rockaa_play -I speech_2ch.wav -O out_effect.pcm -p 256 -Z 1 -F
config_playeffect.bin
```