Rockchip Audio Algorithm Utils

ID: RK-KF-YF-999

Release Version: V1.0.0

Date: 2023-05-10

Security Level: □Top-Secret □Secret □Internal ■Public

DISCLAIMER

THIS DOCUMENT IS PROVIDED “AS IS”. ROCKCHIP ELECTRONICS CO., LTD.(“ROCKCHIP”)DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS,MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip’s registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

This document primarily introduces skills and methods in detail to RockAA Utils testing.

Product Version

SoC	Kernel Version
All Series	Universal

Target Audience

This document is mainly intended for the following engineers:

Technical Support Engineers

Software Development Engineers

Revision History

Version	Author	Date	Change Description
V1.0.0	Xing Zheng	2023-05-10	Initial version

Table of Contents

1. Overview

1.1 RockAA Utils Function Introduction

RockAA (Rockchip Audio Algorithm) Utils are used to help customers in quickly validating the effectiveness of Rockchip audio algorithms without caring about the media framework.

RockAA Utils draws on the design philosophy of tinycap and tinyplay, mainly consisting of the rockaa_capt and rockaa_play tools, which achieve verification in different usage scenarios through various command parameters. They all depend on the common library librockaa.so. Moreover, because rockaa_capt is primarily used for talk testing and rockaa_play is mainly used for playing audio effects testing, the audio algorithm libraries they depend on also differ, with the specific dependency relationships as follows:

```
rockaa_capt
├── librockaa.so              # Common library for RockAA tools
├── libaec_bf_process.so      # Acoustic Echo Cancellation algorithm library
└── librkaudio_common.so      # Common audio algorithm library

rockaa_play
├── librockaa.so              # Common library for RockAA tools
└── librkaudio_effect.so      # Audio effect algorithm library
```

2. Usage Scenarios Introduction

The main applicable scenarios currently include:

- Talk
- Playback Effect

2.1 Offline File Talking Testing

Before debugging the audio algorithm, most developers will collect audio in advance on a mass-produced model with a certain structure, use this audio as a fixed sound source input, and adjust and optimize the algorithm parameters until the output meets the expected audio data:

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

This log indicates operation under talking test via offline file process:

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

Let's analyze the usage of several key parameters

```
"-I" is the only parameter for switching between offline file mode and real-time
online mode. If this parameter is included, it indicates operation in offline
file mode, with the following "record.pcm" representing the pre-recorded audio
source from the device that is ready for processing.
"-O out_talk.pcm" indicates the name of the PCM output after the audio algorithm
has been applied, which is a single channel with a sample rate and bit width
consistent with the "-b 16 -r 16000" specified later. If not specified, a default
PCM file starting with "TALK" will be generated;
"-c 2" indicates the total number of channels, "-C 1" indicates the number of
channels for the remote reference signal, and the number of channels for the
local MIC is the difference between them, such as in this example: "2-1=1", which
is a single MIC pickup scenario;
"-b 16 -r 16000" indicates the two basic PCM formats in addition to the number of
channels, with a 16-bit depth and a 16kHz sampling rate;
"-Z 1" indicates summary debug mode, which will provide a one-time summary of the
frame count, as well as the minimum, maximum, and average time consumption and
load after the test is completed:
[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" indicates that debug mode statistics are not enabled. "-Z 2" indicates
that time consumption and load statistics are performed for each frame. For
example:
[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" indicates a combination of the effects of "-Z 1" and "-Z 2", that is, load
is printed for each frame, and summary debug information is printed at the end.
...
"-F config_aivqe.json" indicates the JSON parameter configuration file for the
audio algorithm. If not specified, the program will exit directly with the
corresponding log prompt:
./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 Real-time Online Talking Testing

As mentioned above, if the "-I" parameter is not included, it means that the parameters from the previous offline file test have achieved good results, and the testing will enter a real-time online mode for a more user-like experience:

```
./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
```

This log indicates operation under real-time online talk testing:

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

Most parameters are the same as in the aforementioned file offline test, and here are some additional description for the different parameters:

```
"record.wav" is used here as the wav file format to save before audio algorithm
processing;
"-D 0 -d 0" comes from the concepts of TINYALSA and ALSA, indicating audio
capture from card0 and device0;
"-t 10" means recording for 10 seconds;
"-F config_aivqe.json" specifies the json parameter configuration file for the
audio algorithm. If not specified, it will work in bypass capture mode, similar
to tinycap, with corresponding log prompts. For example:
./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 Offline Playback Audio Effect Testing

For testing the playback sound effect scenario, developers can also add the "-I" parameter first, using the existing audio source to be tested for offline testing and parameter tuning:

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

This log indicates operation under the offline playback effect test:

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

Let's analyze the usage of several key parameters:

```
"-I" here still serves as the only parameter for switching between offline file
mode and real-time online mode. If this parameter is included, it indicates
operation in offline file mode, with the following "speech_2ch.wav" being the
pre-collected audio source ready for processing.
"-O out_effect.pcm" indicates the PCM name output after specifying the audio
algorithm, which is a single channel, with the sample rate and bit width
consistent with the later specified "-b 16 -r 16000". If not specified, it will
default to generate an output file starting with "EFFECTPLAY";
"-c -b -r" still indicates the three basic elements of PCM format, that is, the
number of channels, bit depth, and sample rate, which can be forcibly specified
as PCM format with the lowercase "-i raw". This example is a wav input,
automatically parsed and played according to the wav format;
"-p 256" means processing 256 sample points per frame, which at 16kHz is
256/16000=16ms per frame.
"-Z 0 -Z 1 -Z 2 -Z 3" have the same effect as the aforementioned rockaa_capt.
"-F config_playeffect.bin" indicates the specification of the sound effect
configuration file. If not specified, it will exit directly with the
corresponding log prompt:
./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 Real-time Online Audio Effect Playback Testing

If the parameters from the previous offline file test have achieved good results, remove the "-I" parameter to enter the real-time online audio effect playback test:

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

This log indicates operation under the real-time online audio effect playback test:

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

Most parameters function similarly to the aforementioned ones. It is important to note that if the "-F" parameter is not specified at this time, it means bypass playback, which has the same effect as tinyplay and has corresponding log prompts. For example:

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

3. More Detailed Usage in Help Information

The help list of rockaa_capt:

```
# ./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
```

The help list for rockaa_play:

```
# ./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
```