

Rockchip麦克风阵列音频算法调试说明文档

文件标识：RK-KF-SF-958

发布版本：V1.4.0

日期：2022-09-02

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 © 2022 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

本文提供一个麦克风阵列结构设计参考文档，工程师可以参照相关内容进行麦克风阵列设计。

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

修订记录

| 版本号 | 作者 | 修改日期 | 修改说明 |
|--------|--------|------------|-------------|
| V1.0.0 | 李茂发 | 2020-11-06 | 初始版本 |
| V1.1.0 | 李茂发 | 2020-11-20 | 增加辅助模块参数 |
| V1.2.0 | 李茂发 | 2021-05-28 | 算法修改参数更新 |
| V1.3.0 | 李茂发 | 2021-08-12 | 算法修改参数更新 |
| V1.3.1 | 廖华平、郑兴 | 2022-08-15 | 整理文档格式 |
| V1.3.2 | 李茂发、郑兴 | 2022-08-30 | 整理文档格式 |
| V1.4.0 | 李茂发、郑兴 | 2022-09-02 | 补充部分参数的适用范围 |

目录

Rockchip麦克风阵列音频算法调试说明文档

1. 概述
 - 1.1 音频算法特性
2. 音频算法说明
 - 2.1 音频算法模块说明
 - 2.1.1 辅助模块
 - 2.1.2 AEC模块
 - 2.1.3 Beamforming模块
 - 2.2 音频算法流程
 - 2.2.1 唤醒流程
 - 2.2.2 通话流程
 - 2.3 音频算法运行效果展示
 - 2.3.1 AEC模块效果
 - 2.3.2 BF模块效果
3. 音频算法接口调用说明
4. 音频算法参数调试说明
 - 4.1 全局模块参数设置
 - 4.2 AEC模块参数设置
 - 4.3 BF模块参数设置
 - 4.4 去混响模块参数设置
 - 4.5 AES模块参数设置
 - 4.6 ANR模块参数设置
 - 4.7 AGC模块参数设置
 - 4.8 舒适噪声参数设置
 - 4.9 DTD模块参数设置
 - 4.10 推荐参数示例

1. 概述

1.1 音频算法特性

多麦克风阵列算法，包括回声消除AEC、波束形成BF、语音降噪ANR、自动增益AGC等。主要包括以下应用场景：

- 回声消除AEC用于消除麦克采集到的扬声器播放的声音
- 波束形成BF用于增强语音，抑制相干噪声和环境噪声
- 语音降噪ANR用于消除麦克采集到的环境噪音
- 自动增益AGC用于增强语音电平信号

2. 音频算法说明

多麦克风声学处理主要完成回声消除和语音增强任务。

2.1 音频算法模块说明

完整的算法包括的流程模块如图1所示，包括如下部分：

2.1.1 辅助模块

这些模块包括高通滤波（High pass-Filter）、预加重（pre-emphasis）、去加重（de-emphasis）等模块，其中，高通滤波模块用于滤除低频电路噪音，预加重用于对语音高频能量提升，增强传输过程高频信噪比，在输出端再通过去加重进行还原。

2.1.2 AEC模块

AEC模块包括MDF模块和Delay模块。

- MDF模块属于线性回声消除算法，用于消除回声
- Delay模块为延迟估计模块，用于估计回采信号与麦克风信号之间的延迟

2.1.3 Beamforming模块

Beamforming（以下简称BF）模块包括FAST_AEC模块、WAKEUP模块、Dereverb模块、NLP模块、AES模块、ANR模块、AGC模块、CNG模块以及DTD模块。

- FAST_AEC模块属于线性回声消除算法，用于抑制残留回声
- WAKEUP模块为RK定制唤醒，主要完成特定唤醒词唤醒机器
- Dereverb模块为去混响模块，用于消除环境混响

- NLP模块属于非线性回声消除算法，用于抑制残留回声
- ANR模块为单通道降噪模块，用于抑制环境噪声
- AGC模块属于自动增益模块，用于控制语音电平信号
- CNG模块为舒适度噪声模块，用于添加舒适度噪声
- DTD模块为双讲检测模块，用于检测语音通话状态

2.2 音频算法流程

多麦克风阵列算法提供两种不同的语音通路。

2.2.1 唤醒流程

该流程通过唤醒进行机器应答以及定位，输出唤醒人所在位置的音频，具体流程如图1所示。

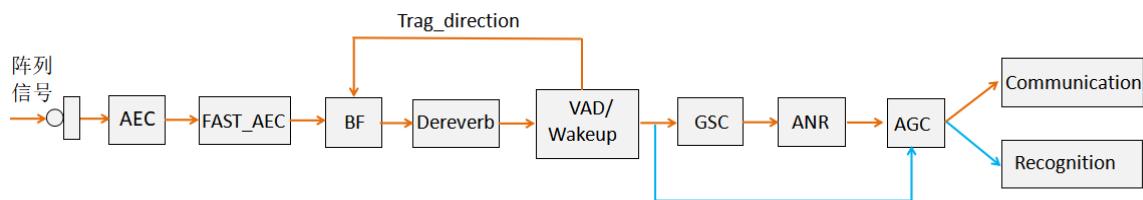


图 1 唤醒流程系统框图

1. 阵列采集阵列信息后，先通过AEC模块消除线性回声，BF内置FAST_AEC的进行进一步的回声抑制；
2. 经过固定波束输出增强后的语音；
3. 接入Dereverb模块去除残留混响；
4. 通过Wakeup模块定位出声源位置；
5. 将定位的语音语音送入GSC后处理模块进行后滤波，（注：蓝色箭头云端识别流程，直接将定位语音送到AGC输出）；
6. 再将处理后的语音送到ANR模块进一步进行降噪；
7. 最后送到AGC模块进行语音增益，送到远端进行通话。

2.2.2 通话流程

该流程不进行唤醒，除了Wakeup模块外其余模块均与唤醒流程相似，具体流程如图2所示。

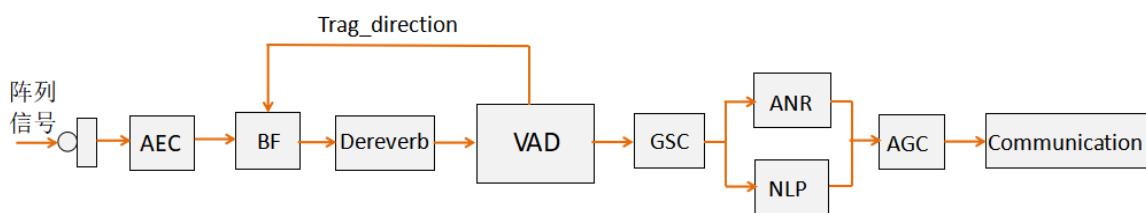


图 2 通话流程系统框图

1. 阵列采集阵列信息后，先通过AEC模块消除线性回声，AES或NLP消除非线性回声；
2. 经过固定波束输出增强后的语音；
3. 接入Dereverb模块去除残留混响；
4. 通过VAD唤醒模块定位出声源位置；
5. 将定位的语音语音送入GSC后处理模块进行后滤波；
6. 再将处理后的语音同时送到NLP和ANR模块进一步进行残留回声抑制以及降噪；

7. 最后送到AGC模块进行语音增益，送到远端进行通话。

2.3 音频算法运行效果展示

2.3.1 AEC模块效果

多麦克风通话算法提供两种不同的语音通路，以下为2中不同通路产生的效果，注该测试音频采用WV2215音箱，麦克风采用MSM261D4030H1CPM，测试环境为瑞芯微演播室，音箱的频响和失真曲线如下所示：

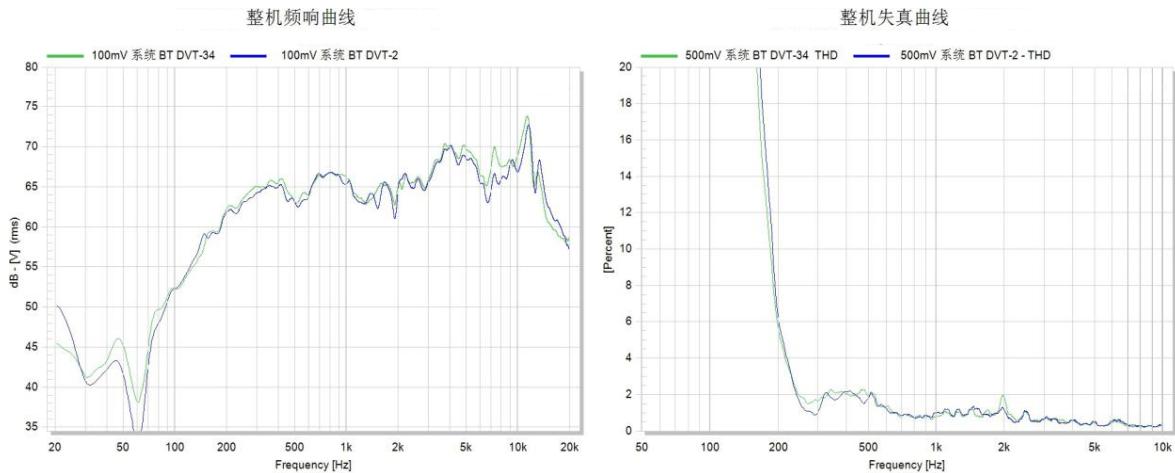


图 3 WV2215频响和失真曲线

采用唤醒加噪声方式进行算法效果统计，左通道为原始输入，右通道为算法处理后输出。其中，左图均为频域图，右图为纯噪声区间内的时域统计结果。

- 线性回声消除效果

以下为线性回声处理结果，需要注意的是该音频并非在专业声学环境测量，存在环境噪声，结果仅做参考。

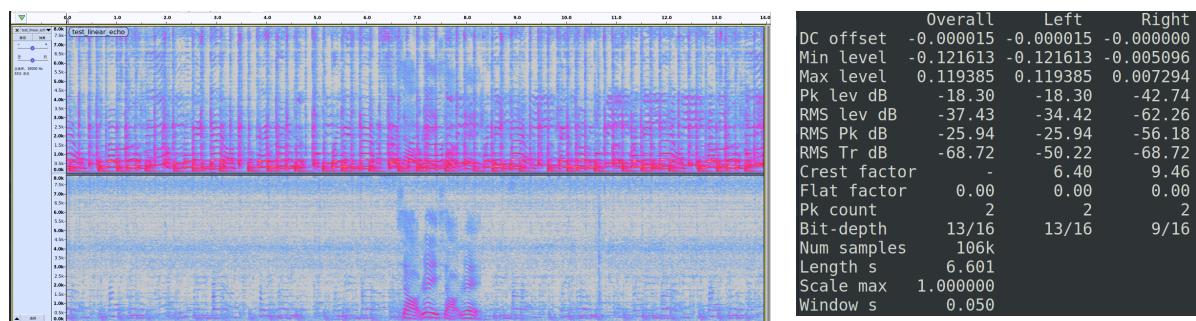


图 4 线性回声消除系统框图

该场景下，回声消除平均在30dB左右。

- 非线性回声消除效果

以下为非线性回声处理结果，需要注意的是该音频并非在专业声学环境测量，结果仅做参考。

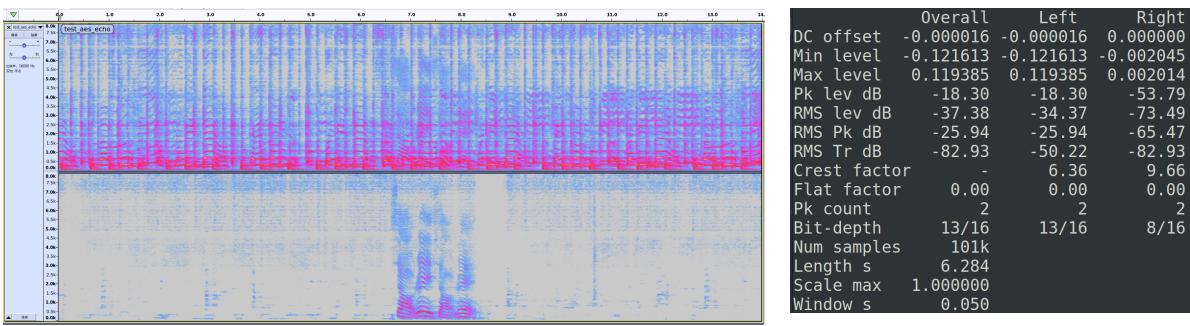


图 5 非线性回声消除系统框图

该场景下，回声消除平均在40dB左右。

2.3.2 BF模块效果

以下为干扰测试，测试场地为瑞芯微演播室，测试声源1m处分贝值为70dB，测试声源距离麦阵5m，干扰声源1m处分贝值为64dB，干扰声源距离麦阵1m：

- 唤醒流程效果

以下为通用环境下唤醒测试，结果仅做参考：

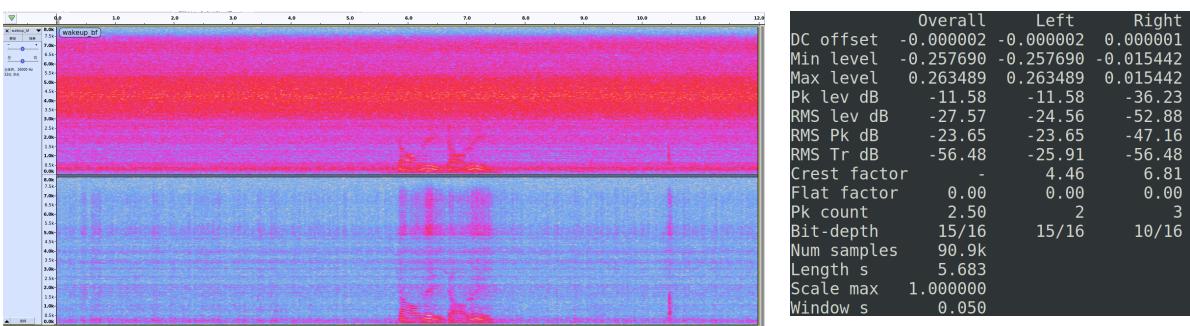


图 6 唤醒流程效果图

该场景下，背景噪声消除平均在28dB左右。

注：语音唤醒只用了固定波束以免造成语音的损伤。

- 通话流程效果

以下为通用环境下通话测试，结果仅做参考：

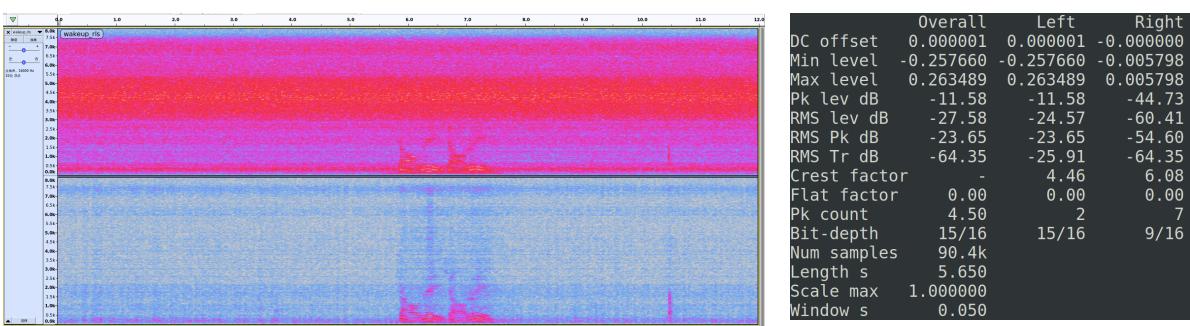


图 7 通话流程效果图

该场景下，背景噪声消除平均在36dB左右。

3. 音频算法接口调用说明

该库的调用过程如下：

1. 设置参数：

该步骤用于对麦克风算法开放的音频参数进行设置，其中，AEC相关参数的设置需要根据产品音频腔体、器件特点以及使用需求等进行调整，使用时需要根据每个产品的特点设置参数，初步集成时可使用默认参数，然后根据效果调整参数。参数设置需要通过rkaudio_preprocess.h文件调整，将该文件路径名作为初始化的输入参数。

2. 初始化：

该步骤用于通话开启后，逐帧处理声音信号前。初始化接口函数名为rkaudio_preprocess_init，具体调用如下所示：

```
st_ptr = rkaudio_preprocess_init(mSampleRate, mBitPerSample, num_src_channel,  
num_ref_channel, &param);
```

其中，mSampleRate采样率为16000Hz，mBitPerSample音频位深为16bit，num_src_channel为mic通道数，num_ref_channel为回采通道数，param为输入参数，该参数通过rkaudio_preprocess.h配置将在后面进行详细说明。

初始化完成算法内部参数初始化和相关内存申请，若初始化成功则返回st_ptr结构体，如果初始化失败，结构体将为NULL。

3. 帧处理：

该步骤即是实时处理通话语音信号，其单位为帧，固定每帧16ms的样点，16k采样时，每帧256个样点。帧处理包括接口函数为rkaudio_preprocess_short，具体调用如下所示：

```
out_size = rkaudio_preprocess_short(st_ptr, (short*)in, out, in_size / 2,  
&is_wakeup);
```

其中，st_ptr为初始化后生成的结构体，in为输入音频单位为byte，out为输出音频单位为short，in_size为一帧读入byte数据量为`256 * num_channel * 2`。

4. 销毁释放：

该步骤用于通话结束后，接口函数为rkaudio_preprocess_destroy(st_ptr)和rkaudio_param_deinit(param)，释放算法内存变量。

4. 音频算法参数调试说明

多麦克风阵列算法通过rkaudio_preprocess.h文件进行参数配置。

4.1 全局模块参数设置

参数设置主要分为AEC模块和BF模块，其中AEC模块通过rkaudio_aec_param_init()函数配置，BF模块通过rkaudio_preprocess_param_init()函数配置。

```
#define NUM_SRC_CHANNEL      8 /* 设置麦克风通道数 */
#define NUM_REF_CHANNEL       2 /* 设置参考通道数，该处设置2路参考通道 */
#define NUM_DROP_CHANNEL      0 /* 设置不处理参考通道数，主要针对2路同样参考通道做丢弃一路
处理 */
#define REF_POSITION          0 /* 参考通道位置，0为参考通道位于mic数据最前面，1则在最后面
*/
static short int Array[NUM_SRC_CHANNEL] = { 9, 7, 5, 3, 2, 4, 6, 8 }; /* 设置麦克风顺序
*/
```

4.2 AEC模块参数设置

AEC各模块开关开关配置如下：

AEC模块通过rkaudio_aec_param_init()函数配置，主要参数如下：

```
param->pos    /* 置0默认回采放输入音频最前面通道，置1回采放输入音频最后面通道 */
param->drop_ref_channel /* 置0不做处理，置1丢弃回采通路的最后一路 */
param->model_aec_en /* 置0不做延迟估计，置EN_DELAY进行延迟估计，软回采必须置1，置
EN_ARRAY_RESET开启麦克风顺序重排功能 */
param->delay_len /* 延迟估计不开时为默认硬回采延迟点，延迟估计打开时为起始延迟估计点 */
param->look_ahead /* 缓存麦克风段数据长度，除非已知采集麦克风信号晚于回采信号可配置，其余情
况默认置0 */
param->Array_list /* 配置麦克风顺序，用于不规则麦克风调整 */
```

4.3 BF模块参数设置

BF各模块开关开关配置如下：

```
typedef enum RKPreprocessEnable_
{
    EN_Fastaec = 1 << 0, /* 打开线性残留回声消除模块，该功能用于唤醒通路进一步消除回声 */
    EN_Wakeup = 1 << 1, /* 打开唤醒模块 */
    EN_Dereverberation = 1 << 2, /* 打开去混响模块 */
    EN_Nlp = 1 << 3, /* 打开非线性残留回声消除模块，该功能用于通话通路进一步消除回声 */
    EN_AES = 1 << 4, /* 同上，同为打开非线性残留回声消除模块 */
    EN_Agc = 1 << 5, /* 打开自动增益控制模块 */
    EN_AnR = 1 << 6, /* 打开噪声抑制模块 */
    EN_GSC = 1 << 7, /* 打开波束后滤波模块，该功能用于进一步消除噪声，视实际效果决定是否打
开 */
    GSC_Method = 1 << 8, /* 关闭则GSC采用LMS方法，打开GSC采用RLS方法，RLS算力高于LMS算
法 */
    EN_Fix = 1 << 9, /* 启用固定方向波束 */
    EN_STDT = 1 << 10, /* 打开双讲检测模块 */
    EN_CNG = 1 << 11, /* 打开舒适度噪声模块 */
    EN_EQ = 1 << 12, /* 打开EQ调节模块 */
    EN_CHN_SELECT = 1 << 13, /* 用于定制化扩展MIC需求场景 */
```

```
} RKPreprocessEnable;
```

BF模块通过rkaudio_preprocess_param_init()函数配置，主要参数如下：

```
param->model_bf_en /* BF模块各模块开关 */
param->Targ /* 当EN_Fix打开时，设置拾音方向 */
param->pos /* 置0默认回采放输入音频最前面通道，置1回采放输入音频最后面通道 */
param->num_ref_channel /* 回采信号数量 */
param->drop_ref_channel /* 丢弃回采信号数量 */
param->dereverb_para = rkaudio_dereverb_param_init() /* 去混响模块参数配置 */
param->aes_para = rkaudio_aes_param_init() /* AES回声抑制参数配置 */
param->nlp_para = rkaudio_nlp_param_init() /* NLP模块参数配置 */
param->anr_para = rkaudio_anr_param_init() /* ANR模块参数配置 */
param->agc_para = rkaudio_agc_param_init() /* AGC模块参数配置 */
param->cng_para = rkaudio_cng_param_init() /* CNG模块参数配置 */
param->dtd_para = rkaudio_dtd_param_init() /* DTD模块参数配置 */
param->eq_para = rkaudio_eq_param_init() /* EQ模块参数配置 */
```

4.4 去混响模块参数设置

去混响模块用于混响消除，该模块通过rkaudio_dereverb_param_init()配置：

```
param->rlsLg /* RLS滤波器阶数，用于唤醒模块去混响，一定范围内，阶数越大，去混响效果越强，算力消耗越大，建议取值：4 */
param->curveLg /* 分布曲线阶数，用于通话模块去混响，阶数越大，去混响效果越强，越容易过消，建议取值：10 */
param->delay /* RLS滤波器延时，决定早期混响保留程度，建议取值：2 */
param->forgetting /* RLS滤波器遗忘因子，因子越小，RLS去混响能力越强，越容易过消，建议范围：0.98-0.999 */
param->T60 /* 混响时间估计值（单位：s），越大，去混响能力越强，但是越容易过消除，低混响场景建议取值0.68，中高混响场景建议取值1.5 */
param->coCoeff /* 互相干性调整系数，防止过消除，越小能力越强，低混响场景建议取值2，中高混响场景建议取值1 */
```

4.5 AES模块参数设置

AES模块用于非线性处理过程，该模块通过rkaudio_aes_param_init()配置。

```
param->Beta_Up = 0.001f; /* 上升速度，该值越大则非线性抑制越强，建议0.001-0.01之间 */
param->Beta_Down = 0.005f; /* 下降速度，该值越大则非线性抑制越弱，建议0.005-0.01之间 */
*/
```

4.6 ANR模块参数设置

降噪模块用于语音降噪，消除环境噪声，通过rkaudio_anr_param_init()函数配置：

```
param->enable_anr /* ANR降噪开关 */
param->noiseFactor /* 降噪等级从0到0.99 */
param->sWU /* 取值范围: 0~10, ANR噪声估计帧长时间片, 无单位, 设置越大噪声估计越准确, 开头收敛越慢 */
param->PsiMin /* 取值范围: 0.0f~1.0f, 语音存在概率判断阈值, 设置越大, 噪音消除效果越强 */
param->PsiMax /* 同上, 取值范围: 0.0f~1.0f, 语音存在概率判断阈值, 设置越大, 噪音消除效果越强 */
*/
param->fGmin /* 取值范围: 0.0f~1.0f, 设置越小, 噪音消除效果越强 */
```

4.7 AGC模块参数设置

AGC模块用于增强语音, 通过rkaudio_agc_param_init()函数配置:

```
param->attack_time = 400.0; /* 建议范围: 20~600ms。触发时间, 即AGC增益上升所需要的时间 */
param->release_time = 400.0; /* 建议范围: 20~600ms。施放时间, 即AGC增益下降所需要的时间 */
param->max_gain = 30.0; /* 建议范围: 0~50dB。最大增益, 同时也是线性段增益 */
param->max_peak = -3.0; /* 取值范围: 0~-87dB。经AGC处理后, 输出语音的最大能量 */
param->fRth0 = -65.0f; /* 取值范围: 0~-87dB。噪声门阈值, 低于该值的语音不被增益 */
param->fRth1 = -60.0f; /* 取值范围: 0~-87dB。扩张段结束能量dB阈值, 高于该值的语音以
max_gain增益 */
param->fRth2 = -35.0f; /* 取值范围: 0~-87dB。压缩段起始能量dB阈值, 高于该值的语音增益逐渐
下降至0dB */
```

注: 压缩段必须满足以下条件: fRth2 + max_gain 必须小于0dB且小于max_peak, 否则会造成截幅。

4.8 舒适噪声参数设置

舒适噪声CNG参数设置用于在通话过程中添加舒适白噪声, 仅在录音处理中存在, 且该模块通常关闭。

```
param->fSmoothAlpha /* 取值范围: 0.0f~1.0f, 施加舒适噪声平滑度, 默认0.92 */
param->fSpeechGain /* 取值范围: 0.0f~1.0f, 施加舒适噪声语音纹理模拟程度, 默认0.3 */
param->fGain /* 建议范围: 0~50dB, 施加舒适噪声幅度比例, 默认2 */
param->fMpy /* 建议范围: 0~50dB, 白噪随机数生成幅度, 默认5 */
```

4.9 DTD模块参数设置

DTD参数设置用于判断语音处于单讲还是双讲态。

```
param->ksiThd_high = 0.70f /* 单双讲判决阈值, ceil阈值, 高于该阈值判断为单讲 */
param->ksiThd_low = 0.50f; /* floor阈值, 低于该阈值判断为单讲 */
```

注: 该值为起始阈值, 目前算法采用自适应更新。

4.10 推荐参数示例

用于唤醒推荐参数如下：

```
param->model_bf_en = EN_Wakeup | EN_Fastaec | EN_STDT | EN_Agc;
```

用于语音通话推荐参数如下：

```
param->model_bf_en = EN_Fastaec | EN_STDT | EN_AES | EN_Anr | EN_Dereverberation  
| EN_Agc;
```