

密级状态：绝密() 秘密() 内部() 公开(√)

RK 单麦克语音通话 3A 算法集成说明及参数 调试说明文档

文件状态： [] 正在修改 [√] 正式发布	当前版本：	V3.0
	作 者：	金剑、顾艳梅、李东强
	完成日期：	2018-4-18
	审 核：	
	完成日期：	

福州瑞芯微电子股份有限公司

Fuzhou Rockchips Electronics Co., Ltd

(版本所有，翻版必究)

版本历史

版本号	作者	修改日期	修改说明	备注
V1.0	金剑	2017-12-1	发布初版	
V2.0	金剑 顾艳梅	2018-1-18	调试参数简化	
V3.0	李东强	2018-4-18	Android 集成调试	

目 录

1. 版本功能说明.....	1
1.1 版本名称.....	1
1.2 算法库类别.....	1
1.3 算法库功能.....	1
1.4 包含文件.....	1
2. 接口函数.....	1
2.1 设置参数.....	2
2.2 初始化.....	2
2.3 帧处理.....	3
2.4 系统释放.....	4
3. 测试数据勾取方法.....	4
4. 参数调试说明.....	5
4.1 总模块参数设置.....	5
4.2 回声消除参数设置.....	5
4.3 噪声抑制参数设置.....	6
4.4 自动增益控制参数设置.....	7
4.5 频响补偿参数设置.....	10
4.6 舒适噪声参数设置.....	11
5. Android 集成调试说明.....	11
5.1 算法库与算法配置参数文件集成.....	11
5.2 Audio Hal 修改集成 audio 3a 算法.....	12
5.3 Android audio 3a 算法调试.....	12

1. 版本功能说明

1.1 版本名称

RK_1MIC_3A_FOR_CALL_V2.0。

1.2 算法库类别

Linux1108 和 NDK。

该算法库仅限于 RK 平台使用。

1.3 算法库功能

单麦克风语音通话 3A 算法，包括回声消除 AEC、语音降噪 ANR 和自动增益控制 AGC 等。

回声消除 AEC 用于消除麦克采集到的扬声器播放的声音；

语音降噪 ANR 用于消除麦克采集到的环境噪音；

自动增益控制 AGC 则用于压缩信号幅度的动态范围，避免信号过小导致响度较小，也可避免信号过大，导致削顶失真，即可以起到提高小信号、压缩大信号的作用。

1.4 包含文件



2. 接口函数

该库的调用过程如下：

第一步，设置参数：

该步骤用于对 3A 算法开放的音频参数进行设置，该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 VOICE_参数设置文件夹中 RK_VoiceSetPara.c 文件单独在 PC 机上建立工程，配置参数，配置好的参数保存为 RK_VoicePara.bin 文件，将该文件放置平台系统目录下，将该文件路径名作为初始化的输入参数。

第二步，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 RK_VOICE_Init，其输入参数为第一步在 PC 机上生成的参数文件路径名，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 0，否则返回其他错误码值，如果初始化失败，将导致第三步逐帧处理异常。

第三步，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，例如 16k 采样时，每帧 256 个样点。帧处理包括两个接口函数，即录音处理 RK_VOICE_ProcessTx 和播放处理 RK_VOICE_ProcessRx，录音处理用于处理麦克风采集的数据，处理后结果经网络发向对端；播放处理用于处理网络接收语音，处理后送往扬声器进行播放。

第四步，系统释放：

该步骤用于通话结束后，接口函数为 RK_VOICE_Destroy，释放 3A 算法内存变量。

2.1 设置参数

参数设置本应该通过调试工具实现，由于客户使用较急，以及当前公司内部开发资源调配，当前 2.0 版本采用提供配置参数 c 代码，客户在 PC 机上通过建立 VC 工程，可实现对参数的配置，参数配置后保存为 RK_VoicePara.bin 文件；后续将会提供对应的可视化调试工具。

参数的配置方法在 RK_VoiceSetPara.c 以及本文档最后一章节有详细说明。

2.2 初始化

函数名称：int RK_VOICE_Init(char *pchParaFile)

函数名称：初始化接口函数

函数功能：完成3A处理初始化，包括参数配置，内存申请等

函数输入：

pchParaFile ----- 参数bin文件输入路径，如"/para/RK_VoicePara.bin"

函数输出：

3A初始化完成

函数返回：

初始化成功状态：

0 表示初始化成功

其他 表示初始化失败

2.3 帧处理

（1）麦克录音信号即 TX 处理

函数名称：void RK_VOICE_ProcessTx(short int *pshwIn,
short int *pshwRef,
short int *pshwOut,
int swFrmLen);

函数功能：完成上行信号逐帧处理

函数输入：pshwIn ----- 麦克输入信号指针

pshwRef ----- 用于回声消除参考信号指针

swFrmLen ----- 输入该帧信号采样点数，注意：8k 采样，固定值 128；

16k 采样，固定值 256

函数输出：pshwOut ----- 上行处理后信号指针

（2）网侧播放信号即 RX 处理

函数名称：void RK_VOICE_ProcessRx(short int *pshwIn,
short int *pshwOut,
int swFrmLen)

函数功能：完成下行信号逐帧处理，处理后送往扬声器

函数输入：pshwIn ----- 远端（网络侧）输入信号指针

swFrmLen ----- 输入该帧信号采样点数，注意：8k 采样，固定值 128；

16k 采样，固定值 256

函数输出：pshwOut ----- 下行处理后信号指针

2.4 系统释放

函数名称：void RK_VOICE_Destroy ()

函数功能：用于通话结束后，释放算法系统内存

参数说明：无

使用方法：RK_VOICE_Destroy ()

注意事项：在通话结束后调用，与 RK_VOICE_Init()对应。

3. 测试数据勾取方法

算法是否集成正确、算法是否满足要求以及如何调整算法参数，这需要能够将通话中实际运行的输入输出数据勾取出来，然后进行观察，如观察时延，观察回声大小以及非线性程度，根据观察结果调试参数，或者将勾取数据提给 RK 相关人员进行算法仿真，也可以给出结论和调整后的参数。

➤ 数据勾取位置

勾取数据时，需要勾取 4 个地方的数据，录音处理函数 RK_VOICE_ProcessTx 的输入和输出（输入为麦克录音数据、回声消除参考数据；输出为 TX 处理后信号）；播放处理函数 RK_VOICE_ProcessRx 的输入和输出（输入为网络侧解码语音数据；输出为 RX 处理后信号）。

➤ 数据勾取注意

- 1) 通话两方一个叫近端，一个叫远端，近端我们通常指的是测试这一端。
- 2) 通话两端不要互相干扰，要离的足够远，或者完全隔音，否则声音会互相串扰，自激，而且也不符合通话场景；
- 3) 尽量保证远端处理是好的，如果远端回声处理有问题，这样也会互相串扰，这个远端可以是固定电话，也可以扬声器改成很小的音量，或者扬声器静音；
- 4) 一次通话尽量快速录三种场景数据，即 PASS 态，单讲态，还有双讲态，pass 指的是仅有近端说话，单讲指的是仅有远端说话，双讲指的是远近端同时都说话；
- 5) 一次录音数据不要过长，几十秒到几分钟相对比较合适；

6) 勾取的各个位置的语音数据时间要是同步的。

4. 参数调试说明

参数文件需要通过参数工具配置生成，本版本已在 VOICE_参数文件目录下生成了一些常用的参数文件，该目录下包括默认参数和备选参数，初步调试时可使用默认参数，然后根据测试效果来选用备选参数，若均没有合适参数，则需要利用 VOICE_参数工具目录下的.c 文件生成需要的参数文件。

参数的具体调试方法如下：

参数工具.c 文件中先会申请 500 个 short int 数据，并清 0。

4.1 总模块参数设置

总模块参数配置用于配置算法总体参数，可配置参数为 0 号参数，表示当前的采样率，1 号到 9 号为预留参数。

```
/* ----- 总模块参数设置 ----- 参数序号: pshwPara[0]至 pshwPara[9] */  
pshwPara[0] = 16000; /* INT16 Q0 采样率设置；当前仅支持 8000,16000 */
```

4.2 回声消除参数设置

此模块用于配置回声消除模块，可配置参数为 10 号到 13 号参数，14 号参数到 99 号参数为预留参数。

```
/* ----- AEC 参数设置 ----- 参数序号: pshwPara[10]至 pshwPara[99] */  
pshwPara[10] = 1; /* 使能标志：延时估计使能 */
```

该参数用于配置延时估计使能开关，当参考 ref 信号与麦克采集回声的相对时延不确定时，可开启该功能，即参数置 1，需要说明的是，延时发生抖动和跳变时，延时估计需要一定的时间才可以修正，该时间内可能会漏回声，因此，需要避免时延抖动；在集成时，建议将时延做成固定延时，然后将该功能关闭。固定延时的一个很好的方法便是采用硬件回采的方式，即使用 ADC 将扬声器即将输出的模拟信号，与麦克采集信号同步采集送进算法，回采的信号即为回声消除参考信号，这样回声参考信号与麦克采集的回声相对时延则基本保持

不变且延时较小。

```
pshwPara[11] = 1; /* INT16 Q0 默认的最大 delay 值 关闭时延估计模块时 */
```

该参数用于时延配置，在 10 号参数置 0，即时延估计功能关闭时，需要配置一个固定延时，该参数即用于配置该固定延时，其单位为采样点，如 16k 采样，延时 5ms，则配置为 $16 \times 5 = 80$ 。注意：该参数最大有效值为 4096，即 16k 采样时，最大有效时延为 256 毫秒。

```
pshwPara[12] = 5; /* INT16 Q0 回声消除抑制等级 0: 不消除; 1-7 抑制程度逐渐加强 */
```

该参数用于配置回声消除的抑制程度，置 0 时表示回声消除功能不使能，置 1 到 7 时表示回声消除的抑制程度逐步加大，但对近端有用语音的损伤也会逐渐变大，因此，要想避免漏回声且要较好的近端语音质量，如连续的双讲效果，需要防止回声的非线性产生，喇叭破音，腔体共振，麦克风隔音不好都会导致大的回声非线性失真，从而只能通过提高抑制等级来加强回声抑制，从而损伤近端语音，降低效果。

回声消除参考信号越接近扬声器中播放的模拟信号则回声消除效果越好，因此，条件允许的话建议采用硬件回采的方式，使用 ADC 将扬声器即将输出的模拟信号，与麦克采集信号同步采集送进算法，回采的信号即为回声消除参考信号；若条件不允许进行硬件回采，则回声参考信号要尽量获取送往扬声器的数字信号，即在软件系统层面，数据越接近扬声器播放的信号其效果越好。

```
pshwPara[13] = 200; /* INT16 Q0 参考信号电路噪声幅度 */
```

该参数用于配置参考信号电路噪声幅度，即在参考输入信号小于该值时，则认为对端没有讲话，此时麦克信号透传，该值可选用默认值 200，在电路噪声很大时可适当提高。

4.3 噪声抑制参数设置

此模块用于配置噪声抑制模块，包括录音数据 TX 的噪声抑制控制和播放数据 RX 的噪声抑制控制。

TX 的 ANR 可配置参数为 100 号参数，第 101 到第 109 号参数为预留参数。

```
/* ----- TX ANR 参数设置 ----- 参数序号: pshwPara[100]至 pshwPara[109] */
```

```
pshwPara[100] = 5; /* INT16 Q0 噪声消除抑制等级 0: 不消除;  
1-5 抑制程度逐渐加强 */
```

该参数用于配置 TX 噪声抑制的程度，配置为 0 表示不进行降噪；配置 1 到 5 表示噪声抑制程度逐步加强，在这里需要说明的是噪声抑制程度越强，噪声抑制的越干净，但语音的损伤也相对会提高，具体等级设置需要根据用户的需求来设定。

RX 的 ANR 可配置参数为 110 号参数，第 111 到第 119 号参数为预留参数。

```
/* ----- RX ANR 参数设置 ----- 参数序号: pshwPara[110]至 pshwPara[119]  
*/  
pshwPara[110] = 5; /* INT16 Q0 噪声消除抑制等级 0: 不消除; 1-5 抑  
制程度逐渐加强 */
```

该参数用于配置 RX 噪声抑制的程度，配置为 0 表示不进行降噪；配置 1 到 5 表示噪声抑制程度逐步加强，在这里需要说明的是噪声抑制程度越强，噪声抑制的越干净，但语音的损伤也相对会提高，具体等级设置需要根据用户的需求来设定。

4.4 自动增益控制参数设置

该模块用于配置自动增益控制参数，分录音数据 TX 的自动增益控制和播放数据 RX 的自动增益控制。自动增益控制可以用于在数字域放大信号或缩小信号，同时可以做到提高小信号，压缩大信号，防止溢出削顶导致的失真。

AGC 将信号分为压缩段、线性段和扩张段，压缩段指的是该段内信号为能量大信号，对该段内数据进行幅度压缩，其压缩程度由压缩段的斜率来控制，压缩段的斜率通常配置小于 1，如 0.4，起到压缩作用；线性段指的是该段内信号为能量适中信号，对该段内信号进行线性增益处理，其线性程度同样可以用线性段的斜率来控制，线性段的斜率通常配置为 1；扩张段指的是该段内信号为能量小信号，对该段内信号进行幅度扩大，其扩大程度由扩张段的斜率进行控制，扩张段的斜率通常配置大于 1，如 1.4。

AGC 在对信号的幅度进行控制时，从大变小或从小变大时，会进行渐变处理，渐变的程度则由各段的平滑点数来决定，平滑点数设置越大，则表明 AGC 的幅度控制响应越慢，否则响应越快；响应越慢，则 AGC 的控制速度越慢，语音自然度较好，但幅度动态范围缺乏有效控制，响应越快效果则反之。通常对于 16k 采样配置 40 到 200 点相对比较合适。另外，AGC 的参数配置涉及定点化问题，因此实际配置时需关注浮点部分，即只需配置浮

点部分，定点部分已经固化，如(short int)(3.0f*(1<<5))表示该值为 16 比特 Q5 定标，配置参数时只需去修改 3.0f 这个值即可。

TX 的 AGC 可配置参数为第 120 号到第 130 号参数，第 131 到第 139 号参数为预留参数。

```
/* ----- TX AGC 参数设置 ----- 参数序号: pshwPara[120]至 pshwPara[139]
*/
```

```
pshwPara[120] = 1; /* 使能标志: 上行 AGC 消除使能 */
```

该参数用于配置 TX 的 AGC 功能是否打开，打开置 1，关闭置 0。

```
pshwPara[121] = (short int)(3.0f*(1<<5)); /* INT16 Q5 线性段提升 dB 数 */
```

该参数用于配置线性增益值，即对信号进行整体放大或缩小，如上面配置为 3.0f，即表示当前信号整体提升 3dB。

```
pshwPara[122] = (short int)(-55.0f*(1<<5)); /* INT16 Q5 扩张段起始能量 dB 阈值 */
```

该参数用于配置扩张段起始能量阈值，可配范围-90 到 0，且应小于第 143 号参数。

```
pshwPara[123] = (short int)(-40.0f*(1<<5)); /* INT16 Q5 线性段起始能量 dB 阈值 */
```

该参数用于配置线性段起始能量阈值，可配范围-90 到 0，且应小于第 144 号参数。

```
pshwPara[124] = (short int)(-16.0f*(1<<5)); /* INT16 Q5 压缩段起始能量 dB 阈值 */
```

该参数用于配置压缩段起始能量阈值，可配范围-90 到 0，建议-20dB 到-10dB 之间。

```
pshwPara[125] = (short int)(1.5f*(1<<12)); /* INT16 Q12 扩张段斜率 */
```

该参数用于配置扩张段斜率。

```
pshwPara[126] = (short int)(0.8f*(1<<12)); /* INT16 Q12 线性段斜率 */
```

该参数用于配置线性段斜率。

```
pshwPara[127] = (short int)(0.4f*(1<<12)); /* INT16 Q12 压缩段斜率 */
```

该参数用于配置压缩段斜率。

```
pshwPara[128] = 40; /* INT16 Q0 扩张段时域平滑点数 */
```

该参数用于配置扩张段时域平滑点数。

```
pshwPara[129] = 80; /* INT16 Q0 线性段时域平滑点数 */
```

该参数用于配置线性段时域平滑点数。

```
pshwPara[130] = 80; /* INT16 Q0 压缩段时域平滑点数 */
```

该参数用于配置压缩段时域平滑点数。

RX 的 AGC 可配置参数为第 140 号到第 150 号参数, 第 151 到第 159 号参数为预留参数。

```
/* ----- RX AGC 参数设置 ----- 参数序号: pshwPara[140]至 pshwPara[159] */
```

```
pshwPara[140] = 1; /* 使能标志: 下行 AGC 消除使能 */
```

该参数用于配置 RX 的 AGC 功能是否打开, 打开置 1, 关闭置 0。

```
pshwPara[141] = (short int)(6.0f*(1<<5)); /* INT16 Q5 线性段提升 dB 数 */
```

该参数用于配置线性增益值, 即对信号进行整体放大或缩小, 如上面配置为 6.0f, 即表示当前信号整体提升 6dB。

```
pshwPara[142] = (short int)(-55.0f*(1<<5)); /* INT16 Q5 扩张段起始能量 dB 阈值 */
```

该参数用于配置扩张段起始能量阈值, 可配范围-90 到 0, 且应小于第 143 号参数。

```
pshwPara[143] = (short int)(-40.0f*(1<<5)); /* INT16 Q5 线性段起始能量 dB 阈值 */
```

该参数用于配置线性段起始能量阈值, 可配范围-90 到 0, 且应小于第 144 号参数。

```
pshwPara[144] = (short int)(-16.0f*(1<<5)); /* INT16 Q5 压缩段起始能量 dB 阈值 */
```

该参数用于配置压缩段起始能量阈值, 可配范围-90 到 0, 建议-20dB 到-10dB 之间。

```
pshwPara[145] = (short int)(1.2f*(1<<12)); /* INT16 Q12 扩张段斜
```

率 */

该参数用于配置扩张段斜率。

```
pshwPara[146] = (short int)(0.8f*(1<<12)); /* INT16 Q12 线性段斜
```

率 */

该参数用于配置线性段斜率。

```
pshwPara[147] = (short int)(0.4f*(1<<12)); /* INT16 Q12 压缩段斜
```

率 */

该参数用于配置压缩段斜率。

```
pshwPara[148] = 40; /* INT16 Q0 扩张段时域平滑
```

点数 */

该参数用于配置扩张段时域平滑点数。

```
pshwPara[149] = 80; /* INT16 Q0 线性段时域平滑
```

点数 */

该参数用于配置线性段时域平滑点数。

```
pshwPara[150] = 80; /* INT16 Q0 压缩段时域平滑
```

点数 */

该参数用于配置压缩段时域平滑点数。

4.5 频响补偿参数设置

频响补偿用于改变信号的频响，主要用于匹配录音和播放系统的频响特性，如播放声音时匹配扬声器的频响。通常录音处理即 TX 处理 EQ 关闭，播放处理即 RX 处理 EQ 根据需求决定是否打开，打开后需要设定 EQ 的频响曲线，然后根据频响曲线设计 FIR 滤波器，将设计好的 FIR 滤波器系数填入参数表中。该功能默认关闭，打开时可联系 RK，由开发人员根据客户需求进行滤波器设定。

```
/* ----- TX EQ 参数设置 ----- 参数序号:pshwPara[160]至 pshwPara[299] */
```

```
pshwPara[160] = 0; /* 使能标志：上行 EQ 使能标
```

志 */

```
pshwPara[161] = 1; /* INT16 Q0 EQ 滤波器长度 */
```

```
pshwPara[162] = (short int)(1.0f*(1<<15)); /* INT16 Q15 EQ 滤波器系数数组
```

```
*/
```

```
/* ----- RX EQ 参数设置 ----- 参数序号: pshwPara[300]至 pshwPara[439]
```

```
*/
```

```
pshwPara[300] = 0; /* 使能标志: 下行 EQ 使能标
```

```
志 */
```

```
pshwPara[301] = 1; /* INT16 Q0 EQ 滤波器长度 */
```

```
pshwPara[302] = (short int)(1.0f*(1<15)); /* INT16 Q15 EQ 滤波器系数数组
```

```
*/
```

4.6 舒适噪声参数设置

舒适噪声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。

该模块可配置参数为第 440 号参数，第 441 到第 449 号参数为预留参数。

```
/* ----- TX CNG 参数设置 ----- 参数序号: pshwPara[440]至 pshwPara[449]
```

```
*/
```

```
pshwPara[440] = 0; /* INT16 Q0 舒适噪声施加等级 0: 不施加; 1-5
```

```
舒适噪声逐渐加大 */
```

该参数用于配置舒适噪声增加程度，通常置 0。

5. Android 集成调试说明

5.1 算法库与算法配置参数文件集成

将算法库与算法配置参数文件拷贝到 sdk 的 device/rockchip/rk3399 目录（集成到 3399 平台），修改 BoardConfig.mk 和 device.mk。将 BOARD_USE_AUDIO_3A 设置为 true，编译时会将算法库与参数拷贝到相应路径。

```
# for audio 3A algorithm
BOARD_USE_AUDIO_3A := false
```

```
#for audio 3a
ifeq ($(BOARD_USE_AUDIO_3A),true)
PRODUCT_COPY_FILES += \
    $(LOCAL_PATH)/libvoiceprocess.so:system/lib/libvoiceprocess.so \
    $(LOCAL_PATH)/RK_VoicePara.bin:system/etc/RK_VoicePara.bin
endif
```

5.2 Audio Hal 修改集成 audio 3a 算法

在 android 平台上，我们将 audio 3a 的算法处理加在 audio hal 上，默认 3a 算法处理是关闭的，如需使用 3A 算法需要将 device/rockchip/rk339/BoardConfig.mk 中的 BOARD_USE_AUDIO_3A 修改为 true，audio HAL 编译会去判断这个值，判断是否使能 3A 算法。

```
# for audio 3A algorithm
BOARD_USE_AUDIO_3A := false

endif

ifeq ($(strip $(BOARD_USE_DRM)),true)
LOCAL_CFLAGS += -DUSE_DRM
endif
ifeq ($(strip $(BOARD_USE_AUDIO_3A)),true)
LOCAL_CFLAGS += -DAUDIO_3A
endif
ifeq ($(strip $(TARGET_BOARD_PLATFORM)),rk3368)
```

注：3a 的算法接口只支持采样率 16k(8k)，单声道的音频数据，如果 audio Hal 实际取到的音频数据格式不是 16k(8k)，单声道的音频数据，需对音频数据进行重采样

5.3 Android audio 3a 算法调试

audio 3a 算法的处理主要是在 hardware/rockchip/audio/tinyalsa_hal/voice_preprocess.c，调试时可以将 LOG_NDEBUG 的宏定义注释，看详细的 log。

```
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the license for the specific language governing permissions and
* limitations under the License.
*/

/**
 * @file    voice_preprocess.c
 * @author  Sun Mingjun <smj@rock-chips.com>
 * @date    2017-05-08
 */

// #define LOG_NDEBUG 0

#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>
#include <errno.h>
#include <pthread.h>
"voice_preprocess.c" [dos] 685L, 27281C
```

需抓取算法前后的音频数据，可以将 ALSA_3A_DEBUG 宏定义的注释取消，在 data

目录创建 3a_capture_in.pcm， 3a_capture_out.pcm， 3a_playback_in.pcm， 3a_playback_out.pcm 并修改文件的读写权限。测试后将文件 pull 出来，确认算法的处理效果。

3a_capture_in.pcm: 算法 TX 处理前音频数据

3a_capture_out.pcm: 算法 TX 处理后音频数据

3a_playback_in.pcm: 算法 RX 处理前音频数据

3a_playback_out.pcm: 算法 RX 处理后音频数据

```
//#define ALSA_3A_DEBUG
#ifdef ALSA_3A_DEBUG
FILE *in_capture_debug;
FILE *out_capture_debug;
FILE *in_playback_debug;
FILE *out_playback_debug;
#endif

char tmp_capture_buffer[capture_min_buffersize];
char tmp_outplayback_buffer[playback_min_buffersize];
char tmp_outcapture_buffer[capture_min_buffersize];
#ifdef ALSA_3A_DEBUG
in_capture_debug = fopen("/data/3a_capture_in.pcm","wb");//please touch /data/3a_in.pcm first
out_capture_debug = fopen("/data/3a_capture_out.pcm","wb");//please touch /data/3a_out.pcm first
in_playback_debug = fopen("/data/3a_playback_in.pcm","wb");//please touch /data/3a_ref.pcm first
out_playback_debug = fopen("/data/3a_playback_out.pcm","wb");//please touch /data/3a_rx.pcm first
#endif

while (handle->voice_thread.running) {
    bool isGetBuffer = false;
```