CV180X & CV181X Audio Quality Tuning User Guide

Version: 0.3

Release date: 2022-08-08

Contents

Revision History

Revision	Date	Description
0.0.0.0	2021/10/22	Initial
0.1.0.0	2022/05/31	Request review version
v0.2	2022/08/08	update

1 Disclaimer

Terms and Conditions

The document and all information contained herein remain the CVITEK Co., Ltd's (
"CVITEK") confidential information, and should not disclose to any third party or use it in
any way without CVITEK's prior written consent. User shall be liable for any damage and
loss caused by unauthority use and disclosure.

CVITEK reserves the right to make changes to information contained in this document at any
time and without notice.

All information contained herein is provided in "AS IS" basis, without warranties of any kind,
expressed or implied, including without limitation mercantability, non-infringement and fitness
for a particular purpose. In no event shall CVITEK be liable for any third party's software
provided herein, User shall only seek remedy against such third party. CVITEK especially
claims that CVITEK shall have no liable for CVITEK's work result based on Customer's
specification or published shandard.

Contact Us

> **Address** Building 1, Yard 9, FengHao East Road, Haidian District, Beijing, 100094,
> China
>
> Building T10, UpperCoast Park, Huizhanwan, Zhancheng Community, Fuhai
> Street, Baoan District, Shenzhen, 518100, China

> **Phone** +86-10-57590723 +86-10-57590724

> **Website** https://www.sophgo.com/

> **Forum** https://developer.sophgo.com/forum/index.html

2 Introduction

2.1 Overview

The VQE (Voice Quality Enhancement) module includes three sub-functions: AGC (Auto Gain Control), ANR (Audio Noise Reduction), and AEC (Acoustic Echo Cancelling).

It is mainly used for recording to provide better sound quality for the client in different product forms and usage scenarios with a single microphone.

The audio foundation mainly used by VQE is speech.

Therefore, the sampling rate mainly supports 8kHz and 16kHz speech audio signals.

This guide focuses on the debugging part of AEC.

2.1.1 The Function of AEC
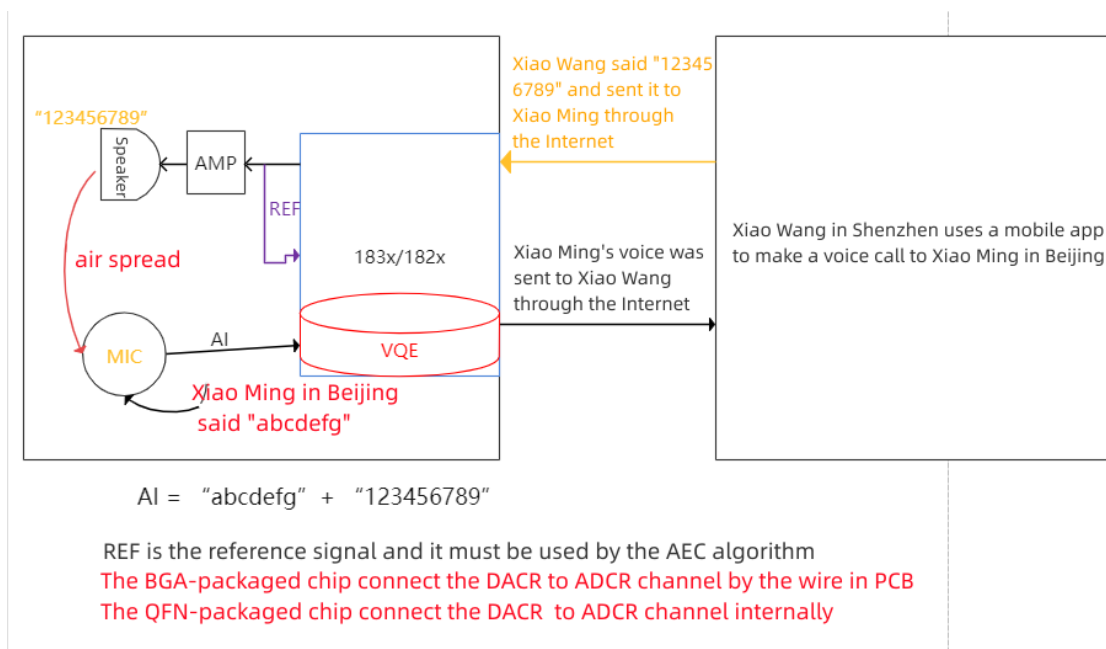
From the above figure, it can be seen that if the Audio Input data is directly sent to Xiao Wang without being processed by the red VQE module, Xiao Wang in Shenzhen will hear two types of sounds— "123456789" and "abcdefg".

"123456789" is Xiao Wang's own voice, which will result in a poor experience for Xiao Wang.

One of the functions of the red VQE, AEC, is used to filter out the sound of "123456789".

After filtering by VQE, Xiao Wang will only hear "abcdefg" said by Xiao Ming.

The sound content of Audio Input is the content of the "ain_record.pcm" file.

## 2.1.2 Basic Requirement for Algorithm

Recording requirement:

- The sampling rate only supports 8kHz or 16kHz, and the playback and recording parameters should be the same.

- AGC/ANR only support mono, not stereo.

- AEC requires dual-channel recording (the left channel is the near-end sound recorded by the mic, and the right channel is the sound sent from the far end).

- The sampling bit depth is 16 bits (enBitwidth = AUDIO_BIT_WIDTH_16).

- The recorded left and right channels cannot be distorted (such as waveform too large, poor quality of mic and speaker, distortion caused by interference in the PCB analog circuit, etc.).

- The amplitude of the near-end sound recorded by the left channel mic should be larger than that of the sound recorded from the speaker (far-end sound), otherwise, it will affect the algorithm processing effect.

- The amplitude of the reference signal in the right channel should be larger than the far-end sound in the sound recorded by the left channel mic,

  otherwise it will affect the algorithm processing effect.

Normal waveform graph (both mic channel and reference signal channel waveforms are moderate, without distortion, no background noise interference, etc.)：
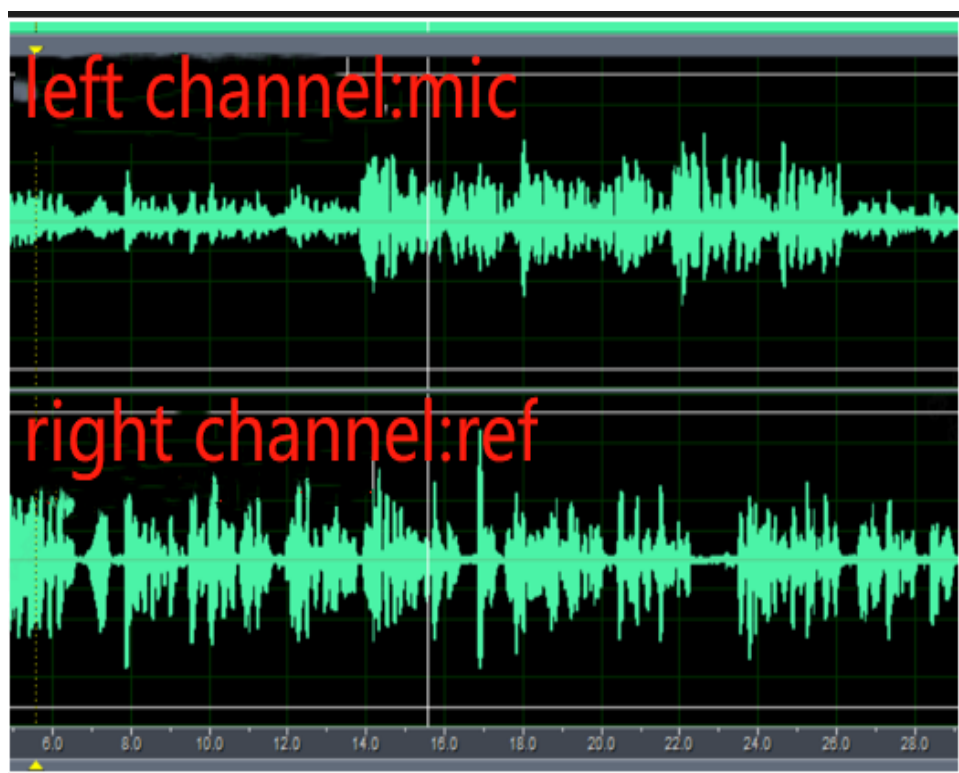
The waveform diagram below is unacceptable:
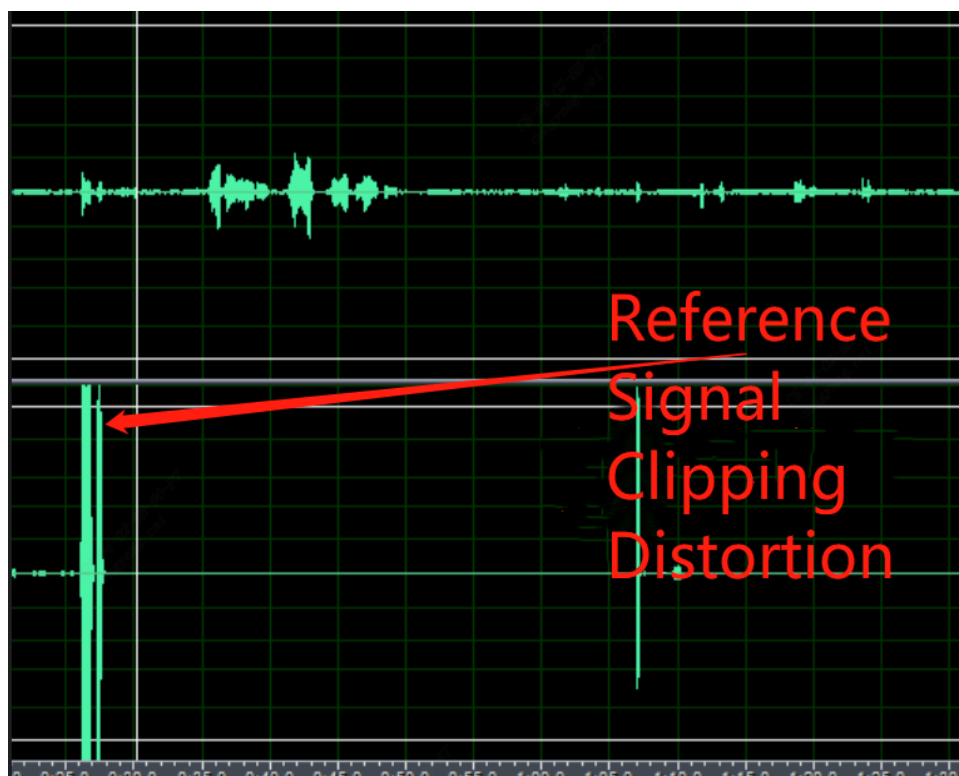
Adjust Method：

1. Reduce the gain of the ADCR channel. For QFN packaging, it is recommended to set it to 1. For BGA packaging, adjust it accordingly.

2. If the distortion still occurs after the first step, reduce the gain of the Audio

Output channel.

Note: The amplitude of the reference signal is affected by the original data amplitude sent by the other party,

the gain of the Audio Output channel, and the gain of the ADCR channel.

Hardware requirements:

- The board hardware should have a mic component.

- The board hardware should have a speaker for audio output.

- The board should have an AEC circuit:

  the sound of the speaker hardware should be captured into the recording's right channel (ADCR) without interference.

More details to see: 《CViTEK Audio Hardware, Structural Design, and Device Selection Guide.docx》

Machine structure requirements：

- The MIC should have a separate sound chamber design and be sealed, and should have an external shockproof rubber sleeve with good shock absorption.

- The MIC pickup should face the opposite direction of the speaker.

- The speaker should have a separate sound chamber design with rubber shock absorption and good shock absorption effect.

- The farther the distance between the MIC and the speaker, the better, and the angle between them should ensure minimal sound coupling.

More details to see: 《CViTEK Audio Hardware, Structural Design, and Device Selection Guide.docx》

2.1.3   Ideal Debug Environment Requirements

Use the customer's complete prototype, and the prototype should be structurally sealed as much as possible.

The mic and spk used must be inside the customer's complete prototype.

Adjust the appropriate ADC/DAC gain level to ensure the stability of the mic in (reception) and ref in (playback circuit).

Confirm that there is no pop noise, circuit noise, or intermittent signal interference before capturing the correct speech pattern.

3 Echo Cancellation Debugging

3.1 Basic Check

There are several requirements for AEC debugging in the environment, which involve recording, playback, volume, and testing environment.

AEC parameter tuning can only be carried out after the basic requirements are met.

Step 1: Set the left and right gain value of ADC, DAC respectively:

During the adjustment process, the user may need to adjust the left and right volume of mic separately, or adjust the left and right volume of the speaker separately.

This can be set through the "sample_audio 9" command, and **the DAC_R and DAC_L should be consistent** as follows:

The user can use:

"sample_audio 9" enter, then input "9" and press enter, and then input "8" (volume value) and press enter to adjust the speaker volume of **the left channel**;

"sample_audio 9" , then input "10" and press enter, and then input "8" (volume value) and press enter to adjust the speaker volume of **the right channel**;

"sample_audio 9" , then input "11" and press enter, and then input "7" (volume value) and press enter to adjust **the mic recording volume**;

"sample_audio 9" , then input "12" and press enter, and then input "1" (volume value) and press enter to adjust **the recording volume of the reference signal (ref)**.

> **Attention:** Dual OS System is currently not support
>
> ```
> ./sample_audio 9
> ```

Step 2: Confirm the correctness of recording and play:

Before using two-way communication, please confirm that the microphone and speaker functions are working properly by using the "sample_audio" program provided in the SDK for verification.

First, check the IP address in the serial terminal, and then **SSH into the board to obtain a terminal** "ssh root@192.168.1.3" .

Then enter the user name "root" and the password "cvitek" to log in to the board.
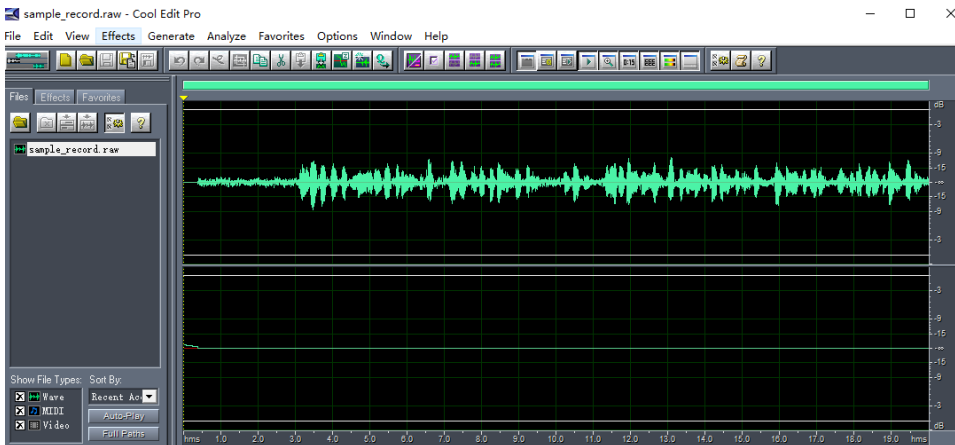
[Recording normal test]:

./sample_audio 4 –list -r 8000 -R 8000 -c 2 -p 320 -C 0 -V 0 -F Cvi_8k_2chn.raw -T 10

is correspondding to parameter corresponding parameter (sample_rate: 8000/ channel : 2/ pre-riod_size:320/ VQEon:0/**preset recording seconds:10**).

Then speak into mic **as if reading a text**.

The Cvi_8k_2chn.raw file will be produced.

[Verification]： Please take out it to a computer and play it, then check if there is any waveform of the audio recording. For example:

The waveform above should be moderate and not exceed the white line above, otherwise it will distort the wave. If it's too large or too small, you can go to the ssh terminal before pressing Enter for "Presets recording seconds: 10" and execute "sample audio 9". Then enter "11" to select the ADCL sub-function, and enter "7" for the gain value which can be set by the customer according to their own actual situation of the waveform.

[Playing Normal Test]：

./sample_audio 5 –list -r 8000 -R 8000 -c 2 -p 320 -C 0 -V 0 -F Cvi_8k_2chn.raw -T 10 speech playing parameter .

(channel:2/ sample rate:8000/period size:320/ VQE on:0)

[Verification]: When playing back, make sure that the speaker is producing sound **at an appropriate volume without any obvious distortion such as popping, noise, or breaking sound.**

Step 3：Confirm that the echo cancellation effect is normal:

Before using the remote intercom program on the client side, we can verify the effect of on-board echo cancellation in the following way.

[AEC Effect Test]:

Execute the commands "touch /tmp/ain_record" and "touch /tmp/dump_ao_output" (If test many times, before each test execute "rm /tmp/*.pcm" to delete files, and do not test for too long as the PCM files will consume DDR memory and cause the system to freeze).

Then execute "**sample_audio 10**

./sample_audio 10 –list -r 8000 -R 8000 -c 2 -p 320 -C 0 -V 1 -F play.wav -T 10"

Audio recording parameter(sample rate:8000/ chn:2/ period size:320)

Set the VQE/AEC to "on" and set the recording seconds and the file name to play.

After 2 seconds of playback (playing the text you read to yourself earlier),

the person speaks into the mic (for example 123456789abcdef),

the customer can speak for a maximum of 20 seconds.

After recording for some time, double-click enter and wait for the end.

Before inputting the "record" time and pressing "enter", set appropriate gain values for ADCL, ADCR, and DACR in the SSH terminal. The setting method have been mentioned above: execute "sample_audio 9" : ADCL:11->set as 10; ADCR:12->set as 1; DACR:10-> set as 8. The values 10, 1, and 8 should be adjusted as needed by the user, and remembered for use in your own intercom program.

[**Verification**]:

Output "sample_record.raw".

Execute "cp /tmp/*.pcm ./ "to copy the file to the current directory.

Copy the" sample_record.raw" and "ain_record.pcm" files to your computer

and use free playback software such as audacity / Cool Edit to view them.

Under normal circumstances, "sample_record.raw" is monaural

and you can only hear "123456789abcdef" without interruption,

which indicates that the effect is good.

In the actual intercom program, the sound of "sample_record.raw" is the audio to sent to the other person for listening.

If you can still hear text or "123456789abcdef" with interruptions or missing words, it means that the effect is not good.

"ain_record.pcm" file is stereo, with the left channel being the sound collected by the mic (including human speech "123456789abcdef" and text sound), and the right channel only containing the sound of the speaker, which is the text sound (**the waveform** of the left and right channels **should be moderate and not distorted**).

3.2 Echo Parameter (under the non-ideal debugging result)

SerialNumber	Command	Corresponding File Name	Annotation
1	touch /tmp/ain_record	/tmp/ain_record.pcm	the most raw "pcm" data from the bottom layer
2	touch /tmp/dump_before_aec	**/tmp/dump_before_aec.pcm**	the pcm(none bind mode) before vqe
3	touch /tmp/dump_after_aec	**/tmp/dump_after_aec.pcm**	the pcm(none bind mode) after vqe
4	touch /tmp/dump_ao_output	/tmp/dump_ao_output.pcm	raw data of the sound played

Data to be provided when the AEC effect is not good (please ensure that the basic requirements of AEC algorithm are met first):

1. Note inside the simple txt file:

    -cv180x⋯ project name

    -sample rate

    -sample_audio 8 printed parameter： ADC L / ADC R （recording volumn）

    DAC L / DAC R   （playback volumn）

    -vqeconfig.txt contains the parameters of the structure bellow in "code" :

```
static CVI_BOOL _update_agc_anr_setting(AI_TALKVQE_CONFIG_S *pstAiVqeTalkAttr)
{
    if (pstAiVqeTalkAttr == NULL)
        return CVI_FALSE;

    pstAiVqeTalkAttr->u32OpenMask |= (NR_ENABLE | AGC_ENABLE | DCREMOVER_ENABLE);

    AUDIO_AGC_CONFIG_S st_AGC_Setting;
    AUDIO_ANR_CONFIG_S st_ANR_Setting;

    st_AGC_Setting.para_agc_max_gain = 0;
    st_AGC_Setting.para_agc_target_high = 2;
    st_AGC_Setting.para_agc_target_low = 72;
    st_AGC_Setting.para_agc_vad_ena = CVI_TRUE;
    st_ANR_Setting.para_nr_snr_coeff = 15;
    st_ANR_Setting.para_nr_init_sile_time = 0;

    pstAiVqeTalkAttr->stAgcCfg = st_AGC_Setting;
    pstAiVqeTalkAttr->stAnrCfg = st_ANR_Setting;

    pstAiVqeTalkAttr->para_notch_freq = 0;
    printf("pstAiVqeTalkAttr:u32OpenMask[0x%x]\n", pstAiVqeTalkAttr->u32OpenMask);
    return CVI_TRUE;
}

static CVI_BOOL _update_aec_setting(AI_TALKVQE_CONFIG_S *pstAiVqeTalkAttr)
{
    if (pstAiVqeTalkAttr == NULL)
        return CVI_FALSE;

    AI_AEC_CONFIG_S default_AEC_Setting;

    memset(&default_AEC_Setting, 0, sizeof(AI_AEC_CONFIG_S));
    default_AEC_Setting.para_aec_filter_len = 13;
    default_AEC_Setting.para_aes_std_thrd = 37;
    default_AEC_Setting.para_aes_supp_coeff = 60;
    pstAiVqeTalkAttr->stAecCfg = default_AEC_Setting;
    pstAiVqeTalkAttr->u32OpenMask = LP_AEC_ENABLE | NLP_AES_ENABLE |
                    NR_ENABLE | AGC_ENABLE;
    printf("pstAiVqeTalkAttr:u32OpenMask[0x%x]\n", pstAiVqeTalkAttr->u32OpenMask);
    return CVI_FALSE;
}
```

2.dump file:

The "*.pcm" file in "/tmp" directory

4 **FAQ**

1. No waveform on the right channel of the "ain_record.pcm" file in BGA packaging platform.

   Answer: Check if there is a reference signal introduced into the ADCR channel on the hardware pcb.

2. There is echo in the intercom (the other person involved can hear their own voice).

   Answer: Check if the recording waveform of the left and right channels in "ain_record.pcm" is too large and distorted. Check if the mic and spk are damaged, if there is interference in the analog circuit. If all of these issues have been ruled out, consider the algorithm and algorithm parameters.

3. The gain value of my ADCR has been set to 1, and the gain set for Audio Output is not large, but the right channel of the "ain_record.pcm" is very distorted despite being attenuated.

   Answer: Check the sound data sent by the other person via the network, save it as a file, and view the waveform on a PC.

4. The sound heard on the machine side is problematic, such as fluctuations in volume and lost words.

   Answer: Assuming the machine is intercomming with a mobile app, the sound played by the machine is sent from the mobile phone. This is generally a problem with the mobile phone. The results of AEC processing **in all intercoms are manifested at the other end**. In other words, our processing results are manifested on the mobile phone side. As long as the person on the mobile phone cannot hear their own voice **and the other person's voice is not lost or stuttered (which may be due to network or processor overload)**, then our AEC processing results are okay.

5. In order to prevent distortion of the right channel waveform of the "ain_record.pcm" on the QFN packaging platform, the gain of the Audio Output volume was reduced, resulting in the speaker not being loud enough. What should I do?

   Answer: The premise for the normal operation of the AEC algorithm is that there is no distortion (including peak attenuation, mic issues, spk issues, and distortion caused by interference). Because the QFN connects the ADCR to the DACR internally and the ADC gain does not have the function of reducing volume, the only solution in this situation is to reduce the Audio Output and increase the amplification coefficient of the power amplifier to solve this contradiction. On the BGA platform, the wiring on the PCB connects the ADCR to the DACR, so resistor dividers can be adjusted to avoid this situation.

6. The following test methods are not effective: using mobile phone playback to replace speaking for testing; machine and mobile phone placed together with one person speaking simul-

taneously.

7. There is serious howling during intercom debugging.

   Answer: Determine whether the distance between the speaker and mic is too close. If there is still howling when they are far apart, the gain value of the mic can be appropriately reduced.

5 VQE Algorithm Function Introduction

5.1 Voice Quality Enhancement (VQE)

For speech signal processing (SSP) algorithms, when the near-end speech signal is interfered with by echo from the far end or stationary noise from the near end, the algorithm functions within SSP can be used to suppress these interferences and improve the quality of the speech signal. The solutions provided in SSP include linear echo cancellation (AEC), nonlinear echo suppression (AES), speech noise reduction (NR), automatic gain control (AGC), and other functions. The SSP algorithm supports speech signals with a sampling rate of 8kHz or 16kHz, mono channel, and 16-bit sampling length. The following pages will introduce each algorithm function and the parameters used.

The parameter "para_fun_config" corresponds to the "u32OpenMask" in the "cvi_comm_aio.h" file, which controls the SSP algorithm functions in the microphone path. The parameter "para_spk_fun_config" controls the SSP algorithm functions in the speaker path. The algorithm functions corresponding to each bit are described in the table below.

Table 5.1: para_fun_config parameter illustration

para_fun_config	Description (Microphone Path)
Bit 0	0: Turn off AEC
	1: Turn on AEC
Bit 1	0: Turn off AES
	1: Turn on AES
Bit 2	0: Turn off NR
	1: Turn on NR
Bit 3	0: Turn off AGC
	1: Turn on AGC
Bit 4	0: Turn off Notch Filter
	1: Turn on Notch Filter
Bit 5	0: Turn off DC Filter
	1: Turn on DC Filter
Bit 6	0: Turn off DG
	1: Turn on DG
Bit 7	0: Turn off Delay
	1: Turn on Delay

Table 5.2: para_spk_fun_config parameter illustration

para_spk_fun_config	Description (Speaker Path)
Bit 0	0: Turn off AGC
	1: Turn on AGC
Bit 1	0: Turn off EQ
	1: Turn on EQ

5.2 AEC/AES (Acoustic Echo Cancellation/Acoustic Echo Suppression)

Any architecture of a duplex communication system has the interference of echo. An echo canceller can eliminate the echo caused by the speaker output coupled back to the microphone through the near-end acoustic path. By using the provided solution, the linear adaptive filter module (AEC) combined with the nonlinear echo suppression module (AES) can effectively suppress the echo and thus improve the quality of speech communication.
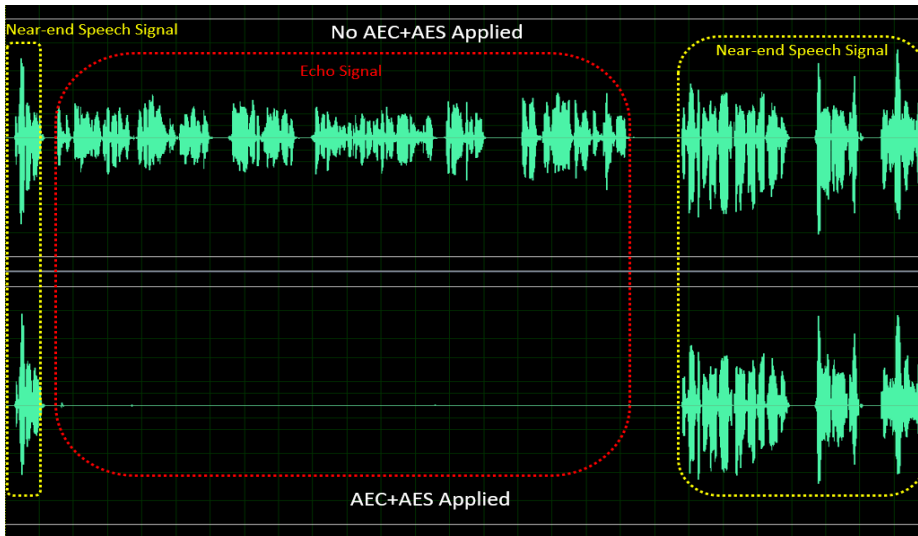

Figure: Performance before and after AEC+AES processing

Four adjustable parameters are provided to tune the performance of AEC/AES, namely:

para_aec_init_filter_len/para_aec_filter_len: the length of the adaptive filter. Adjust the appropriate filter length according to the different echo tail times of the samples. Selecting a longer length will result in higher MIPS and power consumption.

para_aec_init_filter_len is only used for tuning when the echo first appears.

para_aes_std_thrd: the threshold for residual echo detection. Setting a higher value will result in better quality of near-end speech but more residual echo. Conversely, setting a lower value will result in poorer quality of near-end speech but less residual echo.

para_aes_supp_coeff: the strength of residual echo suppression. Setting a larger value will result in stronger suppression of residual echo, but will also lead to more loss/damage of fine details in near-end speech.

Table 5.3: AEC/AES parameter illustration

AEC/AES Parameter	Adjustable Range	Description
para_aec_init_filter_len/para_aec_filter_len	1-13	8kHz sampling rate: [1,13] corresponding to [20ms,260ms] 16kHz sampling rate: [1,13] corresponding to [10ms,130ms]
para_aes_std_thrd	0 - 39	0: the threshold for residual echo detection is the smallest. 39: the threshold for residual echo detection is the biggest.
para_aes_supp_coeff	0 - 100	0: the strength of residual echo suppression is the smallest. 100: the strength of residual echo suppression is the biggest.

5.3   NR (Noise Reduction)

The NR module can suppress stationary noise in the surrounding environment, such as fan noise, air conditioning noise, engine noise, white/pink noise, and so on. With the help of its proprietary speech intelligent "Speech VAD" algorithm, NR can preserve the speech signal while effectively suppressing stationary noise, thereby improving the quality of voice communication.
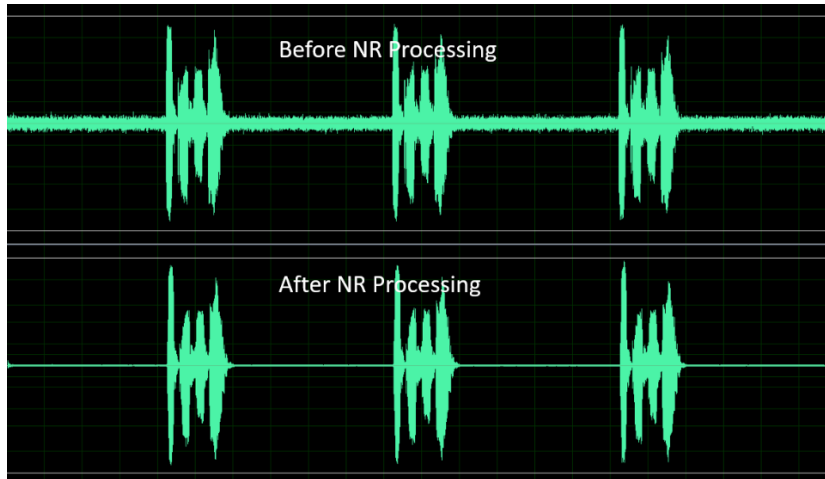

Fig. 5.1: Performance before and after NR processing

There are three adjustable parameters provided for tuning the performance of NR, which are:

para_nr_init_sile_time: the initial length of silence time. When the CODEC is powered on, it will produce random meaningless noise signals, and **para_nr_init_sile_time** can be set to silence this period of signal.

para_nr_snr_coeff: the signal-to-noise ratio (SNR) tracking coefficient. If the parameter value is large, NR will have higher noise reduction ability, but the speech signal may be more prone to distortion. Conversely, if the parameter value is small, NR will suppress less noise signals but will

have better speech quality performance. The following table shows the appropriate adjustment range of this parameter based on different SNR environments. The larger the parameter value is for each SNR condition, the greater the suppression of stationary noise.

Table 5.4: NR parameter illustration

NR Parameter	Adjustable Range	Description
para_nr_init_sile_time	0 - 250	Corresponding to 0s to 5s, each stage is 20ms.

Table 5.5: para_nr_snr_coeff parameter illustration

The ambientenviron-ment ofSNR	Adjustable Range	Description
Low	0 - 3	0: It is the least active in noise reduction 3: It is the most active in noise reduction
Medium	4 - 10	4: It is the least active in noise reduction 10: It is the most active in noise reduction
High	11 - 25	11: It is the least active in noise reduction 25: It is the most active in noise reduction

5.4 AGC (Automatic Gain Control)

The AGC module can automatically adjust the output level to a predetermined range, providing a more comfortable listening experience. If the input signal is below the "Target Low," the AGC will adjust the output level towards "Target Low." On the other hand, if the input signal is above "Target High," the AGC will adjust the output level towards "Target High."
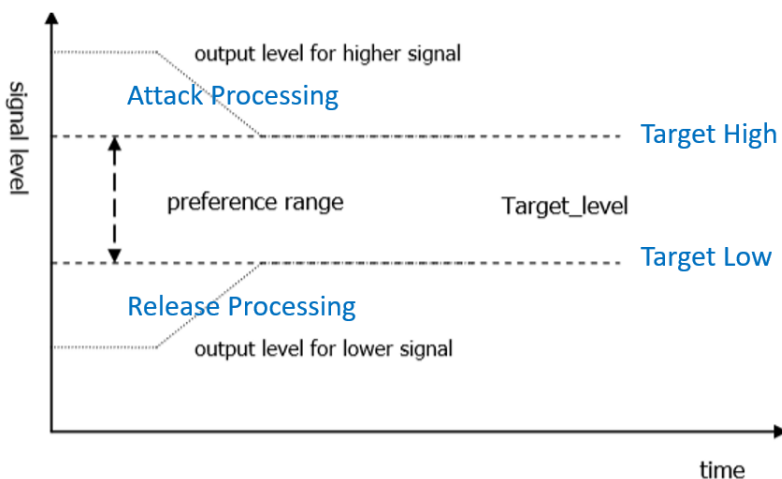

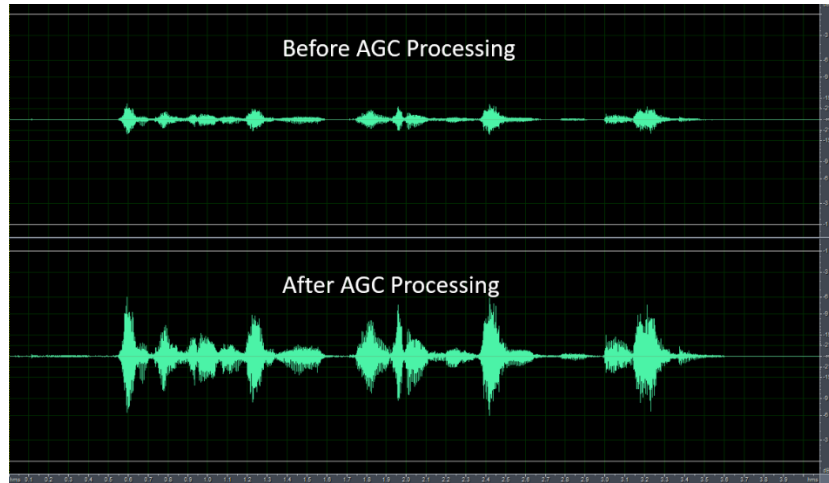Fig. 5.2: AGC adjust the signal level

Fig. 5.3: Performance before and after AGC processing

Four adjustable parameters are provided to adjust the performance of AGC in the microphone path, which are:

para_agc_max_gain: This parameter is the maximum gain that the signal can be amplified.

para_agc_target_high: This parameter is the "Target High" level that the AGC will converge to. For input signals above **para_agc_target_high**, the AGC will converge to **para_agc_target_high**.

para_agc_target_low: This parameter is the "Target Low" level that the AGC will converge to. For input signals below **para_agc_target_low**, the AGC will converge to **para_agc_target_low**. If para_agc_max_gain is reached before **para_agc_target_low**, the AGC will only converge to **para_agc_max_gain**.

para_agc_vad_ena: Speech-activated AGC function. Enabling this function, while also enabling NR and AEC/AES functions, allows the AGC to avoid amplifying residual stationary noise and residual echo to achieve optimal results.

Table 5.6: the AGC parameter illustration of microphone path

AGC Parameter	Adjustable Range	Description
para_agc_max_gain	0 - 6	The maximum increase gain corresponding to [0, 6] is [6dB, 42dB], with 6dB per step.
para_agc_target_high	0 - 36	0 to 36 corresponding to 0dB to -36dB
para_agc_target_low	0 - 72	0 to 72 corresponding to 0dB to -72dB
para_agc_vad_ena	0 - 1	0: Turn off Speech-activated AGC function 1: Turn on Speech-activated AGC function

The three parameters **para_spk_agc_max_gain**, **para_spk_agc_target_high**, and **para_spk_agc_target_low** are used to adjust the performance of the AGC in the speaker

path, and their parameter definitions and adjustment ranges are the same as those of the AGC in the microphone path.

5.5 Notch Filter

Notch FilterParameter	Adjustable Range	Description
para_notch_freq	0 - 1	0: notch frequency is 1kHz 1: notch frequency is 4kHz

5.6 DG (Digital Gain)

This feature helps to reduce residual echo and residual stationary noise. It is not recommended to enable this feature if the gain level of the mic channel is small.

DG Parameter	Adjustable Range	Description
para_dg_target	1 - 12	1: The residual echo/noise suppression is the least aggressive, but provides the best voice quality. 12: The residual echo/noise suppression is the most aggressive, but provides the worst voice quality.

5.7 Delay

This function is used to delay the reference signal and can accelerate the convergence of the echo that appears at the beginning of AEC/AES. If the convergence of the initial echo has already been accelerated by adjusting the parameter para_aec_init_filter_len, it is not recommended to enable this function.

Delay Parameter	Adjustable Range	Description
para_delay_sample	1 - 3000	1 to 3000 corresponding to 1 to 3000 samples

5.8 Equalizer

This function is used for equalizing the speech signal, which can be adjusted by center frequency, gain, and quality factor of the band to achieve the desired frequency response of the speech signal, or to compensate for the imperfect frequency response caused by hardware or speaker units.

Equalizer Parameter	Adjustable Range	Description
para_spk_eq_nband	1 - 5	1 to 5 corresponding to 1 to 5 bands
para_spk_eq_freq[]	8kHz Fs: 0 - 9 16kHz Fs: 0 - 10	Band's center frequency, refer to label para_spk_eq_freq
para_spk_eq_gain[]	0 - 60	Band's gain, refer to label para_spk_eq_gain
para_spk_eq_qfactor[]	0 - 17	Band's quality factor. 0: The frequency response curve centered at para_spk_eq_freq is smoothest, but has the widest range of influence on nearby frequencies. 17: The frequency response curve centered at para_spk_eq_freq is sharpest, but has the narrowest range of influence on nearby frequencies.

[Note]: When para_spk_eq_nband is configured as 1, there is need to simultaneously configure the corresponding:

para_spk_eq_freq[0]、para_spk_eq_gain[0]、para_spk_eq_qfactor[0]. When

para_spk_eq_nband is configured as 2, there is need to simultaneously configure the corresponding: para_spk_eq_freq[0]、

para_spk_eq_gain[0]、para_spk_eq_qfactor[0] and para_spk_eq_freq[1]、

para_spk_eq_gain[1]、para_spk_eq_qfactor[1], the parameters for array index 0 correspond to the first band, and the parameters for array index 1 correspond to the second band, and so on.

para_spk_eq_freq Parameter	The Corresponding Center Frequency (Hz)
0	100
1	200
2	250
3	350
4	500
5	800
6	1200
7	2500
8	3300
9	3990
10 (only suit for sampling rate 16kHz)	7990

| para_spk_eq_gain Parameter	The Corresponding gain (dB)	pa ra_spk_eq_gain Parameter	The Corresponding gain (dB)
0	-40	31	-9
1	-39	32	-8
2	-38	33	-7
3	-37	34	-6
4	-36	35	-5
5	-35	36	-4
6	-34	37	-3
7	-33	38	-2
8	-32	39	-1
9	-31	40	+0
10	-30	41	+1
11	-29	42	+2
12	-28	43	+3
13	-27	44	+4
14	-26	45	+5
15	-25	46	+6
16	-24	47	+7
17	-23	48	+8
18	-22	49	+9
19	-21	50	+10
20	-20	51	+11
21	-19	52	+12
22	-18	53	+13
23	-17	54	+14
24	-16	55	+15
25	-15	56	+16
26	-14	57	+17
27	-13	58	+18
28	-12	59	+19
29	-11	60	+20
30	-10		

VQE processes data for the Audio Input and Audio Output paths separately through two scheduling logics: UpVQE and DnVQE.

UpVQE includes AEC, AES, NR, and AGC.

Each path has different corresponding parameters, which can be found in the header file cvi_comm_aio.h.

Currently, DnVQE is not supported.

6 Block Diagram:
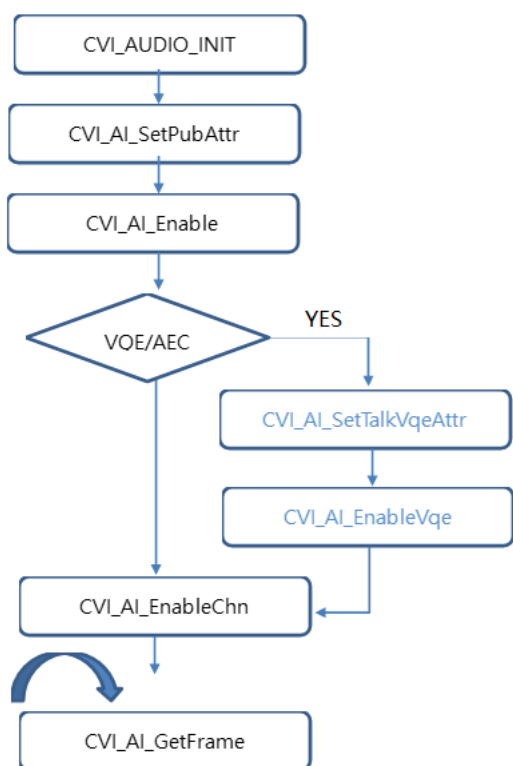
[Illustration]:

Similar to the general audio recording process, after initialization, a thread is used to capture audio frames using CVI_AI_GetFrame.

However, when VQE/AEC is enabled:

1. Two VQE APIs are set:

   CVI_AI_SetTalkVqeAttr

   CVI_AI_EnableVqe

2. When AEC is enabled, the audio input is dual-channel and the audio output is single-channel:

   Please note that the recording attribute CVI_AI_SetPubAttr must be set to dual-channel.

3. When storing with CVI_AI_GetFrame, the channel count for each frame is only for single-channel.

6.1   Detailed Code Explanation

(Please refer to cvi_sample_audio.c, case 10; and cvi_aec_test.c for unit testing process):

The user can enable the VQE function through two APIs, CVI_AI_SetTalkVqeAttr and CVI_AI_EnableVqe.   Only three parameter structures need to be set in CVI_AI_SetTalkVqeAttr:

AUDIO_DEV AiDevId: the audio device ID, which should be consistent with the CVI_AI_Enable ID.

AI_CHN AiCh: the audio device channel, which should be consistent with the CVI_AI_EnableChn ID.

AI_TALKVQE_CONFIG_S *pstVqeConfig,

The AI_TALKVQE_CONFIG_S structure will have the following sub-structures:

```
CVI_U16   para_client_config;
CVI_U32   u32OpenMask;
CVI_S32   s32WorkSampleRate;
/* Sample Rate: 8KHz/16KHz. default: 8KHz*/
//MIC IN VQE setting
AI_AEC_CONFIG_S     stAecCfg;
AUDIO_ANR_CONFIG_S  stAnrCfg;
AUDIO_AGC_CONFIG_S  stAgcCfg;
AUDIO_DELAY_CONFIG_S stAecDelayCfg;
CVI_S32 s32RevMask;//turn this flag to default 0x11
CVI_S32 para_notch_freq;//user can ignore this flag
CVI_CHAR customize[MAX_AUDIO_VQE_CUSTOMIZE_NAME];
```

[**Parameter**]:

```
AI_AEC_CONFIG_S stAecCfg;

AUDIO_ANR_CONFIG_S stAnrCfg;

AUDIO_AGC_CONFIG_S stAgcCfg;
```

The three structures mentioned above can set the corresponding VQE parameters, and decide which switch is needed to use the current VQE via u32OpenMask.

```
s32FrameSample = 160; // The sampling number of each audio frame is a multiple␣
↪of 160.

s32WorkSampleRate //Sampling rate(only support speech sampling rate 8000/16000)

enWorkstate = VQE_WORKSTATE_COMMON;
```

```
para_notch_freq = 0; // Please set the custom parameters to 0.

s32RevMask = 0; // Please set the custom config(custom parameters) to 0.

customize // Please set the custom blocking to "none".
```

[Switch]:

ex: (If you need to fully enable the function.)

```
u32OpenMask = LP_AEC_ENABLE|NLP_AES_ENABLE|NR_ENABLE|AGC_ENABLE|DCREMOVER_
↪ENABLE|DG_ENABLE|DELAY_ENABLE
```

ex: (If you need to only enable the AGC/ANR.)

```
u32OpenMask = (AI_TALKVQE_MASK_AGC)|(AI_TALKVQE_MASK_ANR)
```

ex: (Enable AEC/AGC/ANR)

```
u32OpenMask = (AI_TALKVQE_MASK_AEC)|(AI_TALKVQE_MASK_AGC)|(AI_TALKVQE_MASK_ANR)
```