



HiMPP

FAQs

Issue 04

Date 2015-02-10

Copyright © HiSilicon Technologies Co., Ltd. 2012-2015. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of HiSilicon Technologies Co., Ltd.

Trademarks and Permissions



HISILICON, and other HiSilicon icons are trademarks of HiSilicon Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <http://www.hisilicon.com>

Email: support@hisilicon.com



About This Document

Purpose

This document describes the solutions to the problems that may occur when you use the HiSilicon media processing platform (HiMPP).

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3531	V100
Hi3532	V100
Hi3521	V100
Hi3520A	V100
Hi3520D	V100/V200
Hi3515A	V100
Hi3515C	V100

Intended Audience

This document is intended for:






- Technical support personnel
- Software development engineers



Conventions

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
 DANGER	Indicates a hazard with a high level of risk which, if not avoided, will result in death or serious injury.
 WARNING	Indicates a hazard with a medium or low level of risk that, if not avoided, could result in minor or moderate injury.
 CAUTION	Indicates a potentially hazardous situation, which if not avoided, could result in equipment damage, data loss, performance degradation, or unexpected results.
 TIP	Indicates a tip that may help you solve a problem or save time.
 NOTE	Provides additional information to emphasize or supplement important points of the main text.

General Conventions

The general conventions that may be found in this document are defined as follows.

Convention	Description
Times New Roman	Normal paragraphs are in Times New Roman.
Boldface	Names of files, directories, folders, and users are in boldface . For example, log in as user root .
<i>Italic</i>	Book titles are in <i>italics</i> .
Courier New	Examples of information displayed on the screen are in Courier New.

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.



Issue 04 (2015-02-10)

This issue is the fourth official release, which incorporates the following changes:

Chapter 1 FAQs

The description in section 1.2.1 is updated.

Sections 1.3.1, 1.4, 3.2 and 3.3 are added.

Issue 03 (2013-07-03)

This issue is the third official release, which incorporates the following changes:

Chapter 1 FAQs

The description in section 1.1 is updated.

Issue 02 (2013-06-21)

This issue is the second official release, which incorporates the following changes:

Chapter 1 FAQs

The description in section 1.3 is updated.

Section 1.4 is added.

Issue 01 (2012-09-20)

This issue is the first official release, which incorporates the following changes:

Chapter 1 FAQs

In section 1.2, the descriptions of video output and examples are updated.

Issue 00B02 (2012-08-09)

This issue is the second draft release, which incorporates the following changes:

Chapter 1 FAQs

Section 1.2 is added.

Issue 00B01 (2012-06-30)

This issue is the first draft release.



Contents

About This Document.....	i
1 System Control	1
1.1 Log Information	1
1.1.1 How Do I View HiMPP Logs?	1
1.2 Memory Usage	2
1.2.1 How Do I Adjust the Memories Occupied by Media Services?	2
1.3 Performance	7
1.3.1 How Do I Reduce CPU Usage During memcpy?	7
1.4 Tailoring	8
1.4.1 How Do I Reduce the Application Size When the Static Libraries Are Used?	8
2 Video	9
2.1 VO	9
2.1.1 How Do I Improve the VGA Display Effect?	9
2.2 VPSS	12
2.2.1 How Do I Adjust the Denoising Effect of the VPSS?	12
3 Audio	13
3.1 What Do I Do If the Crackle Occurs When the AO Channel Is Enabled?	13



Figures

Figure 2-1 Video processing workflow	10
Figure 2-2 VGA test standard defined by Tektronix.....	10
Figure 2-3 R423 and R638 in the VGA circuit.....	11
Figure 2-4 VGA output circuit.....	12



1 System Control

1.1 Log Information

1.1.1 How Do I View HiMPP Logs?

[Symptom]

How do I view HiMPP logs and change the log level?

[Cause Analysis]

The logs record the error causes, error locations, and system running status during the running of the software development kit (SDK). Logs can help you locate errors.

Currently, the logs are classified into seven levels, and the default level is level 3. A higher log level indicates that more information is recorded. When the level is set to level 7, the information about the running status of the entire system is recorded in logs in real time. The mass information, however, significantly reduces the overall performance of the system. Typically, you are advised to set the log level to level 3. In this case, information is recorded in logs only when errors occur and most errors can be located.

[Solution]

You can run the following commands to obtain logs, and view or change the log level:

- To view the log level of each module, run the **cat /proc/umap/logmpp** command. Then, the log levels of all the modules are listed.
- To change the log level of a module, run the **echo "venc=4" > /proc/umap/logmpp** command. In this command, **venc** is a module name. This name must be the same as that displayed after the **cat** command is executed.
- To change the log levels of all the modules, run the **echo "all=4" > /proc/umap/logmpp** command.
- To obtain logs, run the **cat /dev/logmpp** command. Then, all the log information is displayed. If all the log information is read, the command is blocked until new log information is recorded. Press **Ctrl+C** to exit. To use the device node in **/dev/logmpp**, run **open** and **read** commands.



1.2 Memory Usage

1.2.1 How Do I Adjust the Memories Occupied by Media Services?

[Symptom]

Media services require memories for normal running. The memories mainly indicate the media memory zone (MMZ). The HiMPP allocates memories based on services. When the memories are insufficient, you can adjust the allocated memories.

[Cause Analysis]

The HiSilicon SDK allows you to adjust the allocated memories when the memories are insufficient. This section briefly describes the measures to minimize memory usage. For details, see related documents.

[Solution]

- Check the operating system (OS) memory and MMZ memory.
For details, see chapter 6 "Allocating and Using the Address Space" in the *Description of the Installation and Upgrade of the Hi35xx SDK* under **SDK\01.software\board\documents_cn**.
- Adjust the memories occupied by SDK services.

- Entire system

Ensure that the size of a D1-series (D1, 2CIF, and CIF) picture is an integral multiple of the sizes of the other D1-series pictures. For example, the size of a D1 picture is 704 x 576, the size of a 2CIF picture is 352 x 576, and the size of a CIF picture is 352 x 288. If the size of a 2CIF picture is 352 x 576 and the size of a CIF picture is 360 x 288, the picture sizes do not meet requirements. In addition, if the size of the picture captured by a VI channel is 720 x 576 and the size of the picture encoded by a VENC channel is 704 x 576, the picture sizes do not meet requirements.

- Minimum buffer size for each module

For details, see the *HiMPP Media Processing Software Development Reference*.

- Just enough public video buffers (VBs)

See HI_MPI_VB_SetConf.

For details, see chapter 2 "System Control" in the *HiMPP Media Processing Software Development Reference*.

In the proc information about the VB, if **IsComm** is **1**, the VB pool is a public VB pool. If **MinFree** of the public VB pool is **0** and there is no displayed information indicating that a module cannot obtain the VB in the logmpp, the public VBs are just enough.

- Video processing subsystem (VPSS)

For details, see chapter 5 "VPSS" in the *HiMPP Media Processing Software Development Reference*.

Measure	MPI	Benefit	Impact	Note
Disable backup frames.	HI_MPI_VPSS_EnableBackupFrame, HI_MPI_VPSS_DisableBackupFrame	Each VPSS group spares one frame buffer.	If the picture is switched when the video output unit (VOU)	



Measure	MPI	Benefit	Impact	Note
			pauses, the background color of the device is displayed.	
Disable the noise reduction (NR), hist, and deinterlacing (DEI) functions simultaneously.	HI_MPI_VPSS_CreateGroup	Each VPSS group spares the buffers for two frames (reference frame and reconstruction frame).	The picture quality is affected.	If any of the NR, hist, and DEI functions is enabled, the buffer for the reference frame and reconstruction frame will be allocated.

– Video encoding (VENC)

For details, see chapter 6 "VENC" in the *HiMPP Media Processing Software Development Reference*.

Measure	MPI/Parameter	Benefit	Impact	Note
Dynamically switch the encoding resolution.	HI_MPI_VENC_GetChnAttr, HI_MPI_VENC_SetChnAttr	The VENC channel is not destroyed when the encoding resolution is switched, which reduces memory fragments.	None	After the encoding resolution is switched, all parameters are restored to default values.
Use the user VB pool mode during multi-channel encoding.	H264eVBSrc HI_MPI_VENC_AttachVbPool, HI_MPI_VENC_DetachVbPool	$(n - 2 \times \text{VedNum})$ frame buffers are spared during n -channel encoding.	None	
The reference frame shares the memory for storing the luminance with the reconstruction frame.	H264eRcnEqualRef	One memory for storing the luminance is spared for each channel.	<ul style="list-style-type: none">Only I frames can be inserted when the jumbo frame appears, bit rate overshoot	



Measure	MPI/Parameter	Benefit	Impact	Note
			occurs, or the stream buffer is full. <ul style="list-style-type: none">• One more memory for storing the luminance is required in 2x frame skipping reference mode.	

– Video decoding (VDEC)

For details, see chapter 10 "VDEC" in the *HiMPP Media Processing Software Development Reference*.

Measure	MPI/Parameter	Benefit	Impact	Note
Set the number of display frames to 0.	HI_MPI_VDEC_GetDisplayFrameNum, HI_MPI_VDEC_SetDisplayFrameNum, HI_MPI_VDEC_GetPrtclParam, HI_MPI_VDEC_SetPrtclParam	The buffer for storing display frames does not need to be allocated for the VDEC channel.	The VO display smoothness is affected.	<ul style="list-style-type: none">• The VPSS backup frames must be disabled.• The number of VDEC display frames cannot be set to 0 if the decoded picture is transmitted to a standard definition (SD) VO device.• You are advised to set the number of display frames to 0 by calling HI_MPI_VDEC_SetDisplayFrameNum before the VDEC channel is created.
Set the	HI_MPI_VDEC_Crea	The buffer for	None	Se the channel



Measure	MPI/Parameter	Benefit	Impact	Note
number of reference frames in the VDEC channel that decodes only I frames to 0 .	teChn	storing reference frames does not need to be allocated.		decoding mode to I mode (only I frames are decoded); otherwise, an error occurs in the logmpp.
Disable the B frame decoding function in the VDEC channels that do not need to decode B frames.	HI_MPI_VDEC_CreateChn	The buffer for outputting the Pmv information does not need to be allocated.	None	
Allocate the stream buffers in reduction mode.	MiniBufMode	The stream buffer size can be reduced.	You need to discard the jumbo frame whose size exceeds the stream buffer size.	This measure is valid only when streams are decoded by frame.

– VO

For details, see chapter 4 "VO" in the *HiMPP Media Processing Software Development Reference*.

Measure	MPI	Benefit	Impact	Note
Set the display queue length to the minimum value 3 .	HI_MPI_VO_GetDispBufLen, HI_MPI_VO_SetDispBufLen	The Hi3531/Hi3532 high definition (HD) device and cascade device spare three frame buffers. The Hi3521/Hi3520A/Hi3520D HD device and cascade device spare one frame buffer.	The VO display smoothness is affected.	
Disable the frame rate doubling function.	HI_MPI_VO_GetPubAttr, HI_MPI_VO_SetPubAttr	The number of required extra buffers in frame rate doubling mode is equal to	When n fields are read, only $n/2$ frames are displayed. The picture	



Measure	MPI	Benefit	Impact	Note
		the display queue length. When the frame rate doubling function is disabled, the required extra buffers do not need to be allocated for the HD device.	smoothness is affected in the violent movement scenario (the value of n depends on the VI timing).	
Set the size of the canvas at the PIP layer for the HD device to the size of the displayed picture.	HI_MPI_VO_GetPipLayerAttr, HI_MPI_VO_SetPipLayerAttr, HI_MPI_VO_GetChnDispPos, HI_MPI_VO_SetChnDispPos	The memory performance is improved by setting stImageSize to the size of the displayed picture.		The automatic zoom-in function of the VOU is not supported at the PIP layer.
Use the automatic zoom-in function of the VOU.	HI_MPI_VO_GetVideoLayerAttr, HI_MPI_VO_SetVideoLayerAttr	If stImageSize is less than stDispRect , the HD VO device at the video layer can use the automatic zoom-in function, which reduces memory usage and bandwidth.		The SD device does not support scaling.

– HiFB

For details, see the *HiFB Development Guide*.

Measure	MPI/Parameter	Benefit	Impact	Note
Set an appropriate size for the physical display buffer at the graphics layer.	video	Setting an appropriate size for the physical display buffer at the graphics layer according to the actual resolution avoids memory waste.	None	If a VO device is not used, the physical display buffer at the corresponding graphics layer does not need to be allocated.



Measure	MPI/Parameter	Benefit	Impact	Note
Set the number of video compress (VCMP) buffers at the graphics layer to 0 .	u32VcmpBufNum	One VCMP buffer is spared in compression mode at the graphics layer.	Artifacts may occur at the bottom of the displayed picture at the graphics layer.	

- Two-Dimensional engine (TDE)

For details, see the *TDE API Reference*.

Measure	Parameter	Benefit	Impact	Note
Allocate an appropriate temporary buffer for the macroblock bitmap.	g_u32TdeTmpBufW, g_u32TdeTmpBufH	Allocating an appropriate temporary buffer avoids memory waste.	None.	The temporary buffer is used only when HI_TDE2_Osd2 Mb or HI_TDE2_MbBl it is being called. You can set g_u32TdeTmpBufW and g_u32TdeTmpBufH to 0 if the two interfaces are not used.

1.3 Performance

1.3.1 How Do I Reduce CPU Usage During memcpy?

[Symptom]

The CPU usage is quite high if the application needs to frequently use functions such as memcpy, memset, memcmp, strcpy, strcmp, and strlen.

[Solution]

The functions memcpy, memset, memcmp, strcpy, strcmp, and strlen are optimized to reduce the CPU usage.

[See Also]

mpp/include/hi_mem.h

mpp/lib/libmem.a

mpp/lib/libmem.so



[Instruction]

Add the **hi_mem.h** file to the application, and link the **libmem** library during compilation.

1.4 Tailoring

1.4.1 How Do I Reduce the Application Size When the Static Libraries Are Used?

[Symptom]

The application uses only a small part of the functions in the **libmpi.a** library. However, the application needs to link to the associated library files such as **vqev2**. As a result, the application size is too large.

[Cause Analysis]

The application needs to link to all the functions defined in the MPI libraries by default when it links to the MPI libraries. Therefore, the application needs to use other libraries associated with the MPI libraries.

[Solution]

When the libraries of the HiMPP are generated, add the **-ffunction-sections** compilation option to **Makefile.param**. When the application links to the MPI libraries during compilation, add the **-Wl,-gc-sections** compilation option to **Makefile**. This deletes the functions that are not used and reduces the application size significantly.



2 Video

2.1 VO

2.1.1 How Do I Improve the VGA Display Effect?

[Symptom]

The VGA or HDMI display effect is not satisfied. For example, the luminance, contrast, hue, and saturation of the displayed video do not reach the expected values. In this case, what do I do to optimize the display effect?

[Cause Analysis]

The quality of VGA pictures is related to the VGA circuits. The SDK configurations are tuned based on reference VGA circuits of HiSilicon. If the VGA picture quality is poor, check the VGA circuits of the board.

- If your VGA circuits are the same as the reference VGA circuits delivered by HiSilicon and the VGA configurations for the SDK are default ones, the VGA color bars indicator meets the VGA test standard defined by Tektronix. See [Figure 1-2](#).

If the VGA display effect is not satisfied (for example, pictures are too bright or the contrast is too large), do as follows:

- Check the monitor settings.
- Separately adjust the VGA effect by calling `HI_MPI_VO_SetVgaParam` for the Hi3531, Hi3532, Hi3521, or Hi3520A or by calling `HI_MPI_VO_SetDevCSC()` for the Hi3520D, Hi3515A, or Hi3515C. Note that `HI_MPI_VO_SetVgaParam` does not apply to the Hi3515A or Hi3520D. The related parameters are as follows:

u32Gain: current gain. If pictures are too dark or too bright, you are advised to set this parameter first.

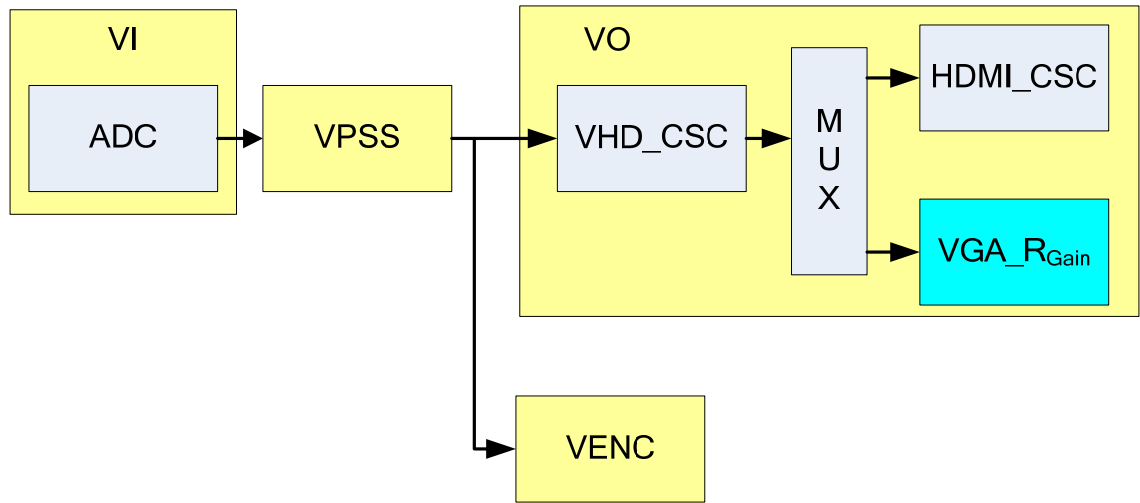
u32luminance: luminance

u32Contrast: contrast

u32Hue: hue

u32Saturation: saturation

Note that the Hi3520D, Hi3515A, or Hi3515C does not support the **u32Gain** parameter. [Figure 1-1](#) shows the video processing workflow. The **VHD_CSC** parameter is adjusted when `HI_MPI_VO_SetDevCSC()` is called; therefore, the HDMI channel is not affected, and you need to adjust the HDMI effect by calling `HI_MPI_HDMI_SetCsc()`.

Figure 2-1 Video processing workflow


If you want to change the default parameter values of the analog-to-digital converter (ADC) of the video input unit (VIU), you are advised to change the values based on the VENC picture effect because the VENC picture effect is closely related to ADC parameters. See [Figure 1-2](#).

Figure 2-2 VGA test standard defined by Tektronix

Color Bars

Line = 145

Average = 1

Color Bars	G	B	R
White	705.904	705.864	696.541
Yellow	701.935	-3.77	693.01
Cyan	701.532	705.356	-3.405
Green	701.561	-3.488	-0.261
Magenta	-4.5	705.698	697.161
Red	0.473	-3.303	693.509
Blue	0.773	705.767	-3.369
Black	0.555	-2.975	-0.034

- If your VGA circuits are different from the reference VGA circuits delivered by HiSilicon, you can separately adjust the VGA effect by calling `HI_MPI_VO_SetVgaParam` for the Hi3531, Hi3532, Hi3521, or Hi3520A or by calling `HI_MPI_VO_SetDevCSC()` for the Hi3515A or Hi3520D. Note that `HI_MPI_VO_SetVgaParam` does not apply to the Hi3515A or Hi3520D.



You can also change your VGA circuits based on reference VGA circuits delivered by HiSilicon. This section uses the Hi3521 circuits as an example. For the circuits of other chips, see the board design schematic diagrams. Note the following when you change VGA circuits:

- Set R423 to 51 k Ω and R638 to 1.1 k Ω , and ensure that the resistor precision is 1%. See [Figure 1-3](#).
- If no video buffer (Z1 in [Figure 1-4](#)), keep the 75 Ω resistors R366, R363, and R367 not connected. See [Figure 1-4](#).

Figure 2-3 R423 and R638 in the VGA circuit

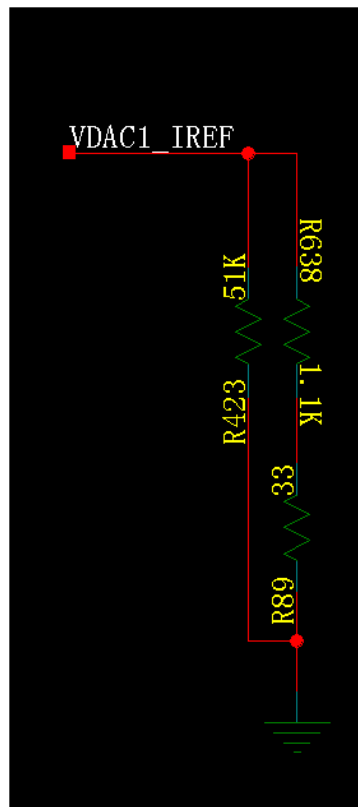
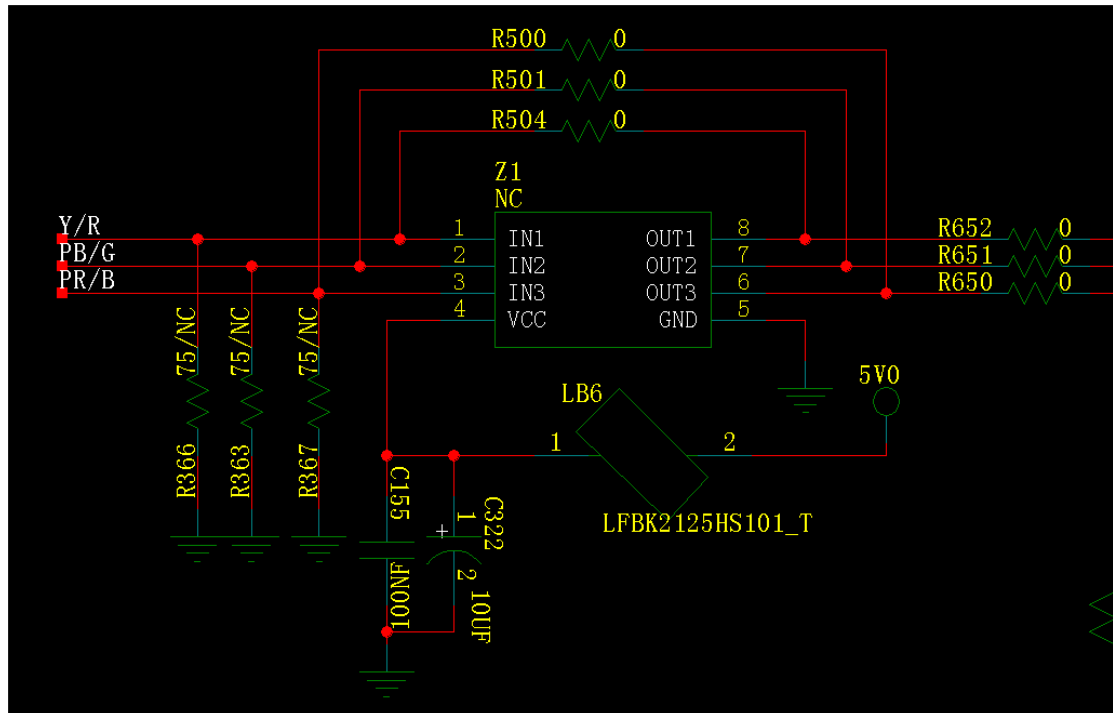


Figure 2-4 VGA output circuit



2.2 VPSS

2.2.1 How Do I Adjust the Denoising Effect of the VPSS?

[Symptom]

How do I adjust the denoising effect of the VPSS for the Hi3520D, Hi3515A, or Hi3515C?

[Cause Analysis]

The **TfStr** and **SfStr** parameters impose the same impacts on the four back-end physical channels of the VPSS, but the **Motion** parameter is valid only for the encoding channel.

[Solution]

- You are advised to set the **SfStr** and **TfStr** parameters at a ratio ranging from 2:1 to 4:1 for previewing channels. When the denoising effect of previewing channels is optimal, adjust the denoising effect of encoding channels.
- Adjust the **Motion** parameter for encoding channels. A larger **Motion** parameter value indicates higher denoising strength. However, if the **Motion** parameter value is too large, smearing occurs.



3 Audio

3.1 What Do I Do If the Crackle Occurs When the AO Channel Is Enabled?

[Symptom]

The crackle occurs when the audio output (AO) channel is enabled.

[Cause Analysis]

According to the voice generation principle, the crackle occurs when the voice suddenly changes. When the AO channel is enabled, the sudden change in voice results in the crackle.

[Solution]

Disable the CODEC output of the digital-to-analog converter (DAC) module before enabling the AO channel. Then enable the CODEC output after data is output from the AO channel.

3.2 How Do I Play the Audio Streams Encoded by HiSilicon on the PC?

3.2.1 How Do I Play G711/G726/ADPCM Audio Streams Encoded by HiSilicon on the PC?

[Symptom]

The G711/G726/ADPCM audio streams encoded by HiSilicon cannot be played directly by using software on the PC.

[Cause Analysis]

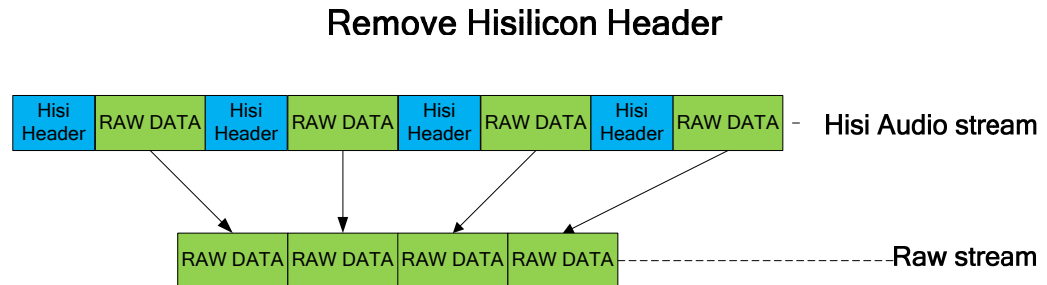
A HiSilicon voice frame header is added at the beginning of each frame in the audio streams encoded by HiSilicon. For details, see section 9.2.2.3 in the *HiMPP V2.0 Media Processing Software Development Reference*. The HiSilicon voice frame header cannot be identified by software on the PC.

[Solution]



Remove the HiSilicon voice frame header at the beginning of each frame, add the WAV header to the frames in the raw stream, and play the streams using software on the PC. [Figure 2-1](#) shows how to remove the HiSilicon voice frame header.

Figure 3-1 Remove the HiSilicon voice frame header



The reference code for removing the HiSilicon voice frame header is as follows:

```
int HisiVoiceGetRawStream(short *Hisivoicedata, short *outdata, int
hisisamplelen)
{
    int len = 0, outlen = 0;
    short *copyHisidata, *copyoutdata;
    int copysamplelen = 0;
    copysamplelen = hisisamplelen;
    copyHisidata = Hisivoicedata;
    copyoutdata = outdata;
    while(copysamplelen > 2)
    {
        len = copyHisidata[1]&0x00ff;
        copysamplelen -= 2;
        copyHisidata += 2;
        if(copysamplelen < len)
        {
            break;
        }
        memcpy(copyoutdata, copyHisidata, len * sizeof(short));
        copyoutdata += len;
        copyHisidata += len;
        copysamplelen -= len;
        outlen += len;
    }
    return outlen;
}
```



NOTE

- The audio streams in ADPCM_DVI4 or ADPCM_ORG_DVI4 format are used for network transfer over the Real-time Transport Protocol (RTP) and cannot be played by the client programs on the PC. For details, see the RFC3551 standard.
- The method of adding the WAV header is not provided in this document. You can add the WAV header by following the WAV header standard. For details, see the reference links [https://msdn.microsoft.com/en-us/library/dd390970\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/dd390970(v=vs.85).aspx) and <http://www.moon-soft.com/program/FORMAT/windows/wavec.htm>.

3.3 How Do I Play Standard Audio Streams on HiSilicon Chips?

3.3.1 How Do I Play Standard G711/G726/ADPCM Audio Streams on HiSilicon Chips?

[Symptom]

The standard G711/G726/ADPCM audio streams cannot be played directly on HiSilicon chips.

[Cause Analysis]

To ensure that the previous-generation chips are compatible, the audio streams can be played on HiSilicon chips only after the HiSilicon voice frame header is added at the beginning of each frame in the raw audio streams.

[Solution]

To play G711/G726/ADPCM audio streams on HiSilicon chips, obtain the raw stream data, add the HiSilicon voice frame header at the beginning of each frame based on the frame data length **PersampleLen**.

- Step 1** Obtain the raw stream data. Remove the WAV header if the WAV header is added to the frame.
- Step 2** Obtain the data length of each frame (**PersampleLen**, a short number).

Table 3-1 Data length of each frame

Encoding Format	Data length of each frame	Remarks
G711	$N \times 40$	N is a positive integer ranging from 1 to 5.
G726 (16 kbits/s)	$N \times 10$	N is a positive integer ranging from 1 to 5.
G726 (24 kbits/s)	$N \times 15$	N is a positive integer ranging from 1 to 5.
G726 (32 kbits/s)	$N \times 20$	N is a positive integer ranging from 1 to 5.
G726 (40 kbits/s)	$N \times 25$	N is a positive integer ranging from 1 to 5.
IMA ADPCM	Number of bytes in each block/2	The number of bytes in each block indicates the number of bytes in the encoded IMA ADPCM data of each block, corresponding to nblockalign (0x20–0x21, 2-byte) of the IMA



Encoding Format	Data length of each frame	Remarks
		ADPCM WAV header.

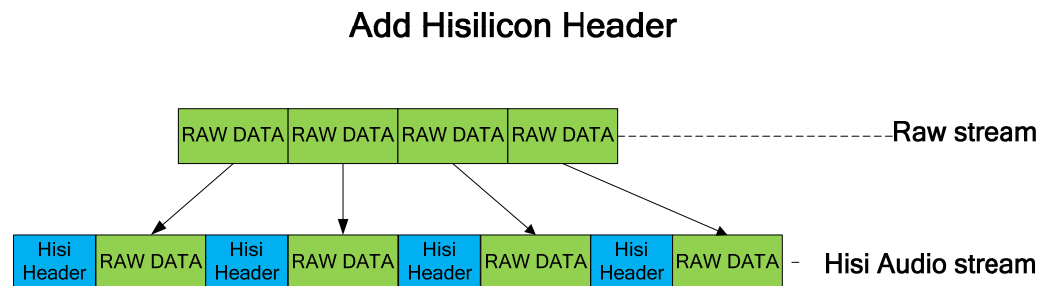


NOTE

- Of all the ADPCM formats, only the IMA ADPCM format is supported. The number of bytes in each sampling point (**whitspersample**) must be 4.
- If the WAV header is added to the frames in ADPCM streams, the number of bytes in each block can be obtained from the WAV header. For the raw ADPCM streams, the number of bytes in each block must be obtained from the provider of the streams.
- Only the mono-channel encoding format is supported.

Step 3 Add the HiSilicon voice frame header, as shown in [Figure 2-2](#).

Figure 3-2 Add the HiSilicon voice frame header



The reference code for adding the HiSilicon voice frame header is as follows:

```
int HisiVoiceAddHisiHeader(short *inputdata, short *Hisivoicedata, int
PersampleLen,int inputsamplelen)
{
    int len = 0, outlen = 0;
    short HisiHeader[2];
    short *copyHisiidata, *copyinputdata;
    int copysamplelen = 0;

    HisiHeader[0] = (short)(0x001<<8) & (0x0300);
    HisiHeader[1] = PersampleLen & 0x00ff;

    copysamplelen = inputsamplelen;
    copyHisiidata = Hisivoicedata;
    copyinputdata = inputdata;

    while(copysamplelen >= PersampleLen)
```



```
{  
    memcpy(copyHisidata, HisiHeader, 2 * sizeof(short));  
    outlen += 2;  
    copyHisidata += 2;  
  
    memcpy(copyHisidata, copyinputdata, PersampleLen * sizeof(short));  
    copyinputdata += PersampleLen;  
    copyHisidata += PersampleLen;  
    copysamplelen -= PersampleLen;  
    outlen += PersampleLen;  
}  
  
return outlen;  
}  
  
----End
```