HMS

Debugging Guide

Issue	**04**
Date	**2015-12-22**

Trademarks and Permissions

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/cn/

Email: support@hisilicon.com

About This Document

Purpose

This document describes the Proc debugging information about the high-definition (HD) software development kit (SDK).

Related Versions

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

Product	Version
Hi3798C	V1XX
Hi3798M	V1XX
Hi3796C	V1XX
Hi3796M	V1XX
Hi3798C	V2XX

Intended Audience

This document is intended for:

- Technical support personnel
- Software development engineers

Change History

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

Issue 04 (2015-12-22)

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

The debugging information of the VDEC, ADEC, AO, HDMI, Media_mem, PMOC, and HiPlayer modules is modified.

Issue 03 (2015-10-28)

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

Section 2.33 is modified.

Issue 02 (2015-05-20)

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

Hi3798C V200 is supported.

Issue 01 (2014-11-05)

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

Hi3796M V100 is supported.

Issue 00B01 (2014-06-13)

This issue is the first draft release.

Contents

Figures

1 Overview

The debugging information is obtained from the Proc file system of Linux. The information reflects the current system status and can be used to locate and analyze problems. You can also use the Proc file system to complete driver settings for debugging.

The Proc files of the HiSilicon SDK are under **/proc/msp**. Other directories under **/proc** are the Proc file system directories inherent in Linux.

You can view the Proc information by using either of the following ways:

- Run the **cat** command in the console. For example, run **cat /proc/msp/vdec00**.
- To obtain help information, run the **help** command on the module, for example, **echo help > /proc/msp/avplay00**.
- The debugging information varies according to the version or running status of the HD chip. This document takes HiSTBLinux V100R003 as an example.

CAUTION

Note the following when reading the parameter descriptions in this document:

- For the parameter that indicates whether, On/Off, or enable, if mappings between the values and definitions are not specified, the value **1** indicates affirmative and the value **0** indicates negative by default.
- For the parameter whose value is **0** or **1**, if mappings between the values and definitions are not specified, the value **1** indicates affirmative and the value **0** indicates negative.
- For the parameter whose value is **aaa**, **bbb**, or **ccc**, if the mappings between the values and the definitions are not specified, you can identify the parameter definitions based on **aaa**, **bbb**, or **ccc**.

2 Proc Information

2.1 SYS

[Debugging Information]

```
# cat /proc/msp/sys
SDK_VERSION:[HiSTBLinuxV100R003] Build Time:[Jun5 2014, 12:00:40]
CHIP_VERSION: HI3798C(0x21)_v200
DOLBY: YES
DTS: YES
ADVCA: NO
ROVI(Macrovision): NO
```

[Analysis]

The basic information about the system is recorded.

[Parameter Description]

Parameter	Description
SDK_VERSION	Version and compiled time of the SDK
CHIP_VERSION	Chip version
DOLBY	Whether Dolby is supported Yes: supported No: not supported
DTS	Whether DTS is supported Yes: supported No: not supported
ROVI	Whether Macrovision is supported Yes: supported No: not supported

After the .ko file is replaced, check the SDK version and compiled time to ensure that the correct .ko file is used. An incorrect .ko file may result in version upgrade failure.

The chip version indicates the version of the chip that is running programs. If the flag on the chip is blocked by the heat sink, you can obtain the chip version here.

[Tip]

To view the version of the SDK corresponding to the released library files and .ko files, run the following command:

```
strings library files or .ko files | grep SDK_VERSION
```

After the command is executed, the SDK version information is displayed as follows:

```
$ strings ./pub/lib/share/libhi_common.so | grep SDK_VERSION
SDK_VERSION:[HiSTBLinuxV100R003C00SPC010] Build Time:[Jun 5 2014, 10:31:53]
$ strings ./pub /kmod/hi_common.ko | grep SDK_VERSION
SDK_VERSION:[HiSTBLinuxV100R003C00SPC010] Build Time:[Jun 5 2014, 10:31:12]
```

2.2 Demux

2.2.1 demux_main

[Debugging Information]

```
# cat /proc/msp/demux_main
DmxIdPortId
0--
1--
2--
3--
4--
5--
6--
7--
8--
9--
10--
11--
12--
13--
```

type "echo help > /proc/msp/demux_main" to get help information [Analysis]

Information about the binding relationship between Demuxs and ports is recorded.

[Parameter Description]

Parameter	Description
DmxId	Demux ID
PortId	Port ID. The IF ports are numbered from 0, the TSI ports are numbered from 32, and the RAM ports are numbered from 128. The port ID indicates the ID of the port to which the Demux is bound. "--" indicates that the Demux is not bound to any port, and it cannot receive any data.

[Debugging Commands]

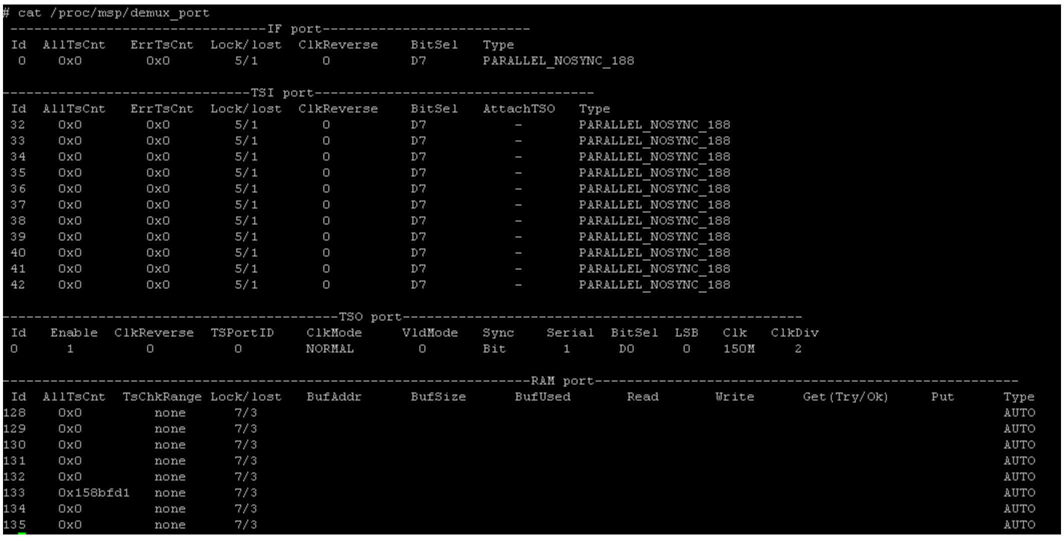
- **echo help > /proc/msp/demux_main**: Obtains help.
- Recording port streams
  - **echo save allts start x[portid] > /proc/msp/demux_main**: Starts recording.
  - **echo save allts stop > /proc/msp/demux_main**: Stops recording.

  For example, to record streams from TSI1 based on the proc information of the preceding instance, run **echo save allts start 32 > /proc/msp/demux_main**.
- Recording streams sent to the IP port
  - **echo save ipts start x[ram portid]> /proc/msp/demux_main**: Starts recording.
  - **echo save ipts stop > /proc/msp/demux_main**: Stops recording.

  For example, to record streams from RAM0 based on the proc information of the preceding instance, run **echo save allts start 128 > /proc/msp/demux_main**.
- Recording TSs from the Demux
  - **echo save dmxts start x[dmxid] > /proc/msp/demux_main**: Starts recording.
  - **echo save dmxts stop > /proc/msp/demux_main**: Stops recording.
- Recording ESs from the Demux
  - **echo save es start > /proc/msp/demux_main**: Starts recording.
  - **echo save es stop > /proc/msp/demux_main**: Stops recording.

2.2.2 demux_port

[Debugging Information]



```
# cat /proc/msp/demux_port
 --------------------------------IF port--------------------------
 Id  AllTsCnt   ErrTsCnt  Lock/lost  ClkReverse    BitSel   Type
  0    0x0        0x0        5/1        0          D7       PARALLEL_NOSYNC_188

-------------------------------TSI port-----------------------------------
 Id  AllTsCnt   ErrTsCnt  Lock/lost  ClkReverse    BitSel   AttachTSO   Type
 32    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 33    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 34    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 35    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 36    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 37    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 38    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 39    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 40    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 41    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 42    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188

----------------------------------------TSO port-------------------------------------------------
 Id   Enable  ClkReverse  TSPortID    ClkMode     VldMode   Sync    Serial  BitSel  LSB   Clk   ClkDiv
 0      1        0           0       NORMAL        0       Bit       1      D0      0    150M     2

----------------------------------------------------------------RAM port-------------------------------------------------------------
 Id  AllTsCnt  TsChkRange Lock/lost   BufAddr      BufSize      BufUsed       Read       Write      Get(Try/Ok)      Put      Type
128    0x0         none      7/3                                                                                             AUTO
129    0x0         none      7/3                                                                                             AUTO
130    0x0         none      7/3                                                                                             AUTO
131    0x0         none      7/3                                                                                             AUTO
132    0x0         none      7/3                                                                                             AUTO
133    0x158bfd1   none      7/3                                                                                             AUTO
134    0x0         none      7/3                                                                                             AUTO
135    0x0         none      7/3                                                                                             AUTO
```

[Analysis]

Information about the Demux ports is recorded.

The IF port is the input port for tuner signals. It uses the embedded QAM and is numbered from 0. There is typically no more than one IF port.

The TSI port is also the input port for tuner signals. It is different from the IF port because it uses the external QAM.

The RAM port is a DDR port.

The TSO port is a TS output port.

[Parameter Description]

Parameter	Description
Id	Port ID. The tuner ports are numbered from 0, and the RAM ports are numbered from 128.
AllTsCnt	Count of total TS packets, 32 bits#
ErrTsCnt	Count of error TS packets, 16 bits
Lock/lost	Sync lock threshold and sync loss threshold
ClkReverse	Whether the TS input clock is inversed (valid only for tuner ports) 0: not inversed 1: inversed
BitSel	Port line sequence select (valid only for tuner ports) D7: If the port type is parallel, cdata[7] is the most significant bit. If the port type is serial, cdata[7] is the data line. D0: If the port type is parallel, cdata[0] is the most significant bit. If the port type is serial, cdata[0] is the data line.
Type	Port type PARALLEL_BURST: parallel burst mode PARALLEL_VALID: parallel valid mode PARALLEL_NOSYNC_188: auto-sync 188 mode PARALLEL_NOSYNC_204: auto-sync 204 mode PARALLEL_NOSYNC_188_204: 188/204 automatic identification mode AUTO: auto-sync mode USER_DEFINED: user-defined mode SERIAL: serial mode SER_SERIAL2BIT_NOSYNC: 2-bit serial

Parameter	Description
	mode SER_NOSYNC: serial auto-sync mode SER_2BIT_NOSYNC: 2-bit serial auto-sync mode ERR: incorrect setting
Enable	Whether to enable the TSO port 0: disabled 1: enabled
TSPortID	ID of the IF/TSI/RAM port from which the TSO port streams come
ClkMode	TSO port clock mode NORMAL: even clock JITTER: jitter clock
VldMode	Working mode of the TSO port valid signal 0: The valid signal is high only when data is output (corresponding to HI_UNF_DMX_TSO_VALID_ACTIVE_OUTPUT). 1: The valid signal is always high (corresponding to HI_UNF_DMX_TSO_VALID_ACTIVE_HIGH).
Sync	Duration for outputting the sync signal from the TSO port Bit: The sync signal is transmitted only when the first bit of the sync byte is output. Byte: The sync signal is transmitted when the sync byte is output.
Serial	Output mode of the TSO port 0: parallel 1: serial
LSB	Endian mode of the TSO port 0: The most significant bit is output first (big-endian). 1: The least significant bit is output first (little-endian).
Clk	TSO module clock frequency
ClkDiv	TSO port divider TSO port output frequency = TSO module clock frequency (Clk)/TSO port divider (ClkDiv)

Parameter	Description
TsChkRange	TS packet length check range (valid for only the RAM port) none: The RAM port type is AUTO, and the hardware automatically detects the packet length (ranging from 188 to 255). x–y: Packet length range. (When the RAM port type is **USER_DEFINED**, the length range of the detected packets can be defined by the user.)
BufAddr	Start address of a TS buffer
BufSize	Size of a TS buffer
BufUsed	Size and percentage of the used TS buffer
Read	Offset of the read pointer of the TS buffer
Write	Offset of the write pointer of the TS buffer
Get(Try/Ok)	**Try** indicates the times that HI_UNF_DMX_GetTSBuffer is called. **Ok** indicates the times that HI_UNF_DMX_GetTSBuffer is successfully called.
Put	Times that HI_UNF_DMX_PutTSBuffer is successfully called

In normal cases, **AllTsCnt** gradually increases. Otherwise, no data is input from the port. Check the following items:

- IF/TSI port
  - Whether the tuner is locked
  - Whether the port type is correctly set
- RAM port
  - Whether the port type is correctly set
  - Whether external data is transmitted to the port, that is, whether HI_UNF_DMX_GetTSBuffer and HI_UNF_DMX_PutTSBuffer are called to inject TS data

If **ErrTsCnt** continuously increases, the Demux continuously receives error TS packets. In this case, mosaics may be displayed and audio intermittent playback may occur.

2.2.3 demux_chan

[Debugging Information]

```
cat /proc/msp/demux_chan
DmxIdChnIdPIDType Mod StatKeyIdAcquire(Try/Ok)Release
800x1022VIDPLY OPEN--1936/1481475
```

```
810x1023AUDPLY OPEN--593/569568
```

[Analysis]

Information about the Demux channels is recorded.

[Parameter Description]

Parameter	Description
DmxId	ID of the Demux to which a channel belongs
ChnId	Channel ID
PID	Channel PID
Type	Type of a channel. The value can be **SEC**, **PES**, **AUD**, **VID**, **ECM,** or **PST**. Specify the channel type when creating a channel.
Mod	Output mode of a channel. The value can be **PLY**, **REC**, or **P&R**. Specify the channel output mode when creating a channel.
Stat	Channel status. The value can be **OPEN** or **CLOSE**.
KeyId	ID of the key area to which a channel is bound. "**--**" indicates that the channel is not bound to a key area.
Acquire(Try/Ok)	**Try**: number of attempts to obtain channel buffer **Ok**: times of successfully obtaining channel buffer If the output mode of a channel is **REC**, **Try** and **Ok** are always **0**.
Release	Times that HI_UNF_DMX_ReleaseBuf is successfully called

2.2.4 demux_chanbuf

[Debugging Information]

```
cat /proc/msp/demux_chanbuf
DmxId  ChnId  FqId  OqId  Size  BlkCnt  BlkSize  Read  Write  Used  Overflow
   8     0      1     0 16383K 1008    16644    475    473    99%    0
   8     1      2     1 255K  204    1280    160    158    99%    0
```

[Analysis]

Information about the Demux channel buffers is recorded.

[Parameter Description]

Parameter	Description
DmxId	ID of the Demux to which a channel belongs
ChnId	Channel ID

Parameter	Description
FqId	ID of the FQ to which the channel is bound
OqId	ID of the OQ to which the channel is bound
Size	Size of a channel buffer. The Demux divides the channel buffer into several blocks.
BlkCnt	Count of channel buffer blocks
BlkSize	Size of a channel buffer block
Read	Read pointer to a channel buffer block. Its value constantly changes when data is read from the channel.
Write	Write pointer to a channel buffer block. Its value constantly changes when data is written to the channel.
Used	Percentage of the used channel buffer
Overflow	Times of channel buffer overflows. Data is lost when overflow occurs.

2.2.5 demux_filter

[Debugging Information]

```
cat /proc/msp/demux_filter
DmxId ChnId FltIdDepth Param
0001Match : 00
Mask: 00
Negate: 00
1--11Match : 02
Mask: 00
Negate: 00
```

[Analysis]

Information about the Demux filters is recorded.

[Parameter Description]

Parameter	Description
DmxId	ID of the Demux to which a channel belongs
ChnId	ID of the channel to which a filter is bound. "--" indicates that the filter is not bound to any channel.
FltId	Filter ID

Parameter	Description
Depth	Filter depth
Param	**Match**: matched bytes of a filter, in hexadecimal **Mask**: masked bytes of a filter, in hexadecimal **Negate**: negated bytes of a filter, in hexadecimal

2.2.6 demux_key

[Debugging Information]

```
cat /proc/msp/demux_key
DmxId KeyId ChnCnt AttachCnt DetachCnt SetEvenKeyCnt SetOddKeyCnt
EvenKeyOddKey
0011012
13bebd9d64 cbcff00c 00000000 00000000 4df5fd5b 90a60cde 00000000 00000000
```

[Analysis]

Information about the Demux key areas is recorded.

[Parameter Description]

Parameter	Description
DmxId	ID of the Demux to which a channel belongs
keyId	Key area ID
AttachCnt	Number of times that the key is bound
DetachCnt	Number of times that the key is unbound
SetEvenKeyCnt	Number of times that the even key is configured
SetOddKeyCnt	Number of times that the odd key is configured
ChanCnt	Number of channels to which a key area is bound. The value **0** indicates that the key area is not bound to any channel.
EvenKey	Even key, in hexadecimal. The first value corresponds to CW1 to CW4, the second value corresponds to CW5 to CW8, the third value corresponds to CW9 to CW12, and the fourth value corresponds to CW13 to CW16.
OddKey	Odd key, in hexadecimal. The first value corresponds to CW1 to CW4, the second value corresponds to CW5 to CW8, the third value corresponds to CW9 to CW12, and the fourth value corresponds to CW13 to CW16.

2.2.7 demux_pcr

[Debugging Information]

```
cat /proc/msp/demux_pcr
DmxId PcrId PID   CurrPcr    CurrScr
0     0   0x1fff 0xffffffff  0xffffffff
```

[Analysis]

Information about the Demux program clock reference (PCR) channels is recorded.

[Parameter Description]

Parameter	Description
DmxId	ID of the Demux to which a PCR channel belongs
PcrId	PCR channel ID
PID	PID of a PCR channel
CurrPcr	Current program clock reference (PCR) value, in ms
CurrScr	Current local clock value, in ms

2.2.8 demux_rec

[Debugging Information]

```
cat /proc/msp/demux_rec
DmxId RecId FqId OqId Type Descramed Status  Size  BlkCnt BlkSize Read Write
Overflow
  1     0     4     3 pid     1      start 1880K   40    48128 36 36   0
```

[Analysis]

Information about Demux recording is recorded.

[Parameter Description]

Parameter	Description
DmxId	Demux ID
RecId	Recording instance ID
FqId	ID of the FQ to which the recording is bound
OqId	ID of the OQ to which the recording is bound
Type	Recording type

Parameter	Description
	pid: Data with the specified PID is recorded all: Data with all PIDs is recorded.
Descramed	Whether to record descrambled data 0: no 1: yes
Status	Recording status Start: recording Stop: stopped
Size	Size of the recording buffer. The Demux divides the recording buffer into several blocks.
BlkCnt	Number of recording buffer blocks
BlkSize	Size of a recording buffer block
Read	Read pointer to a recording buffer block
Write	Write pointer to a recording buffer block
Overflow	Times of recording buffer overflows

2.2.9 demux_rec_index

[Debugging Information]

```
cat /proc/msp/demux_rec_index
DmxId RecId FqId OqId ScdId Type  Pid    Size BlkCnt BlkSize Read Write Overflow
  1  0     5    4     0 video 0x200 56K   64    896    18 18   0
```

[Analysis]

Information about the Demux recording index is recorded.

[Parameter Description]

Parameter	Description
DmxId	Demux ID
RecId	Recording instance ID
FqId	ID of the FQ to which the recording index is bound
OqId	ID of the OQ to which the recording index is bound
ScdId	ID of the SCD to which the recording index is bound

Parameter	Description
Type	Index type Audio: audio index Video: video index None: no index
Pid	Index PID
Size	Size of the index buffer. The Demux divides the index buffer into several blocks.
BlkCnt	Number of index buffer blocks
BlkSize	Size of an index buffer block
Read	Read pointer to an index buffer block
Write	Write pointer to an index buffer block
Overflow	Count of index buffer overflows

[Debugging Commands]

- Obtain help information.
  echo help > /proc/msp/demux_main
- Record port streams.
  - Start recording: **echo save allts start x[portid] > /proc/msp/demux_main**
  - Stop recording: **echo save allts stop > /proc/msp/demux_main**
- Record streams received by IP ports.
  - Start recording: **echo save ipts start x[ram portid]> /proc/msp/demux_main**
  - Stop recording: **echo save ipts stop > /proc/msp/demux_main**
- Record TSs output by the Demux.
  - Start recording: **echo save dmxts start x[dmxid] > /proc/msp/demux_main**
  - Stop recording: **echo save dmxts stop > /proc/msp/demux_main**
- Record ESs output by the Demux.
  - Start recording: **echo save es start > /proc/msp/demux_main**
  - Stop recording: **echo save es stop > /proc/msp/demux_main**

2.3 AVPLAY

[Debugging Information]

```
#cat /proc/msp/avplay00
----------------------Hisilicon AVPLAY0 Out Info-------------------
Stream Type :TS   |DmxId :0
CurStatus    :PLAY |OverflowProc  :RESET
Sync ID      :sync00      |ThreadID     :1340
```

```
ThreadScheTimeOutCnt  :0 |ThreadExeTimeOutCnt :0
------------------------------VID CHANNEL--------------------------
Vid Enable   :TRUE |Vdec Type   :MPEG2
VidOverflowNum:0   |Vdec Mode    :NORMAL
VidPid:0x200 |FrcEnable    :TRUE
FrcInRate    :50.0 |FrcOutRate   :50.0
TplaySpeed   :1.0  |LowDelayEnable:FALSE
Vdec ID      :vdec00
FrameChanID :port0000->win0100(master)
FrameChanID :port0001->win0000(slave00)
AcquireFrame(Try/OK)  :1840/461
SendFrame(Try/OK)    :484/481(master)
SendFrame(Try/OK)    :481/481(slave00)
------------------------------AUD CHANNEL--------------------------
Aud Enable   :TRUE |Adec Type    :mp3
AudOverflowNum:0   |AdecDelayMs :0
DmxAudChnNum  :1
DmxAudPid    :0x28a
Adec ID      :adec00
Track ID     :track00(master)
AcquireStream(Try/OK) :1864/279
SendStream(Try/OK)   :279/279
AcquireFrame(Try/OK)  :1864/426
SendFrame(Try/OK)    :426/426(master)
```

[Analysis]

The status of an AVPLAY is recorded. The HD chip supports 16 players and simultaneous playback of at most 16 programs (which is restricted by the decoding capability of the chip). Each player generates a node in **/proc/msp**. You need to select the correct AVPLAY during debugging.

[Parameter Description]

Parameter	Description
Stream Type	Type of an input stream. The HD chip supports TS and ES.
DmxId	Demux ID
CurStatus	Status of an AVPLAY, such as **STOP**, **PLAY**, **TPLAY**, **PAUSE**, and **EOS**
OverflowProc	Mode of processing the buffer overflow **RESET**: to reset the buffer when it overflows **DISCARD**: to discard the buffer when it overflows

Parameter	Description
Sync ID	SYNC instances corresponding to an AVPLAY
Thread ID	ID of an AVPLAY data processing thread
ThreadScheTimeOutCnt	Timeout count for calling an AVPLAY data processing thread (the default timeout threshold is 30 ms)#
ThreadExeTimeOutCnt	Timeout count for executing an AVPLAY data processing thread (the default timeout threshold is 30 ms)#
Vid Enable	Working status of a video decoder (VDEC) HI_TRUE: enabled HI_FALSE: disabled
Vdec Type	Video decoding type: MPEG2, MPEG4 AVS, H263, H264, VC1, and UNKNOW
VidOverflowNum	Count of video ES buffer overflows
Vdec Mode	Video decoding mode **NORMAL**: to decode all the frames **IP**: to decode I frames and P frames **I**: to decode only I frames
VidPid	Video PID
FrcEnable	Whether to enable frame rate conversion
FrcInRate	Conversion to the input frame rate
FrcOutRate	Conversion to the output frame rate
TplaySpeed	Play speed
LowDelayEnable	Whether to enable low delay
Vdec ID	ID of the VDEC instance corresponding to an AVPLAY
FrameChanID	ID of the video frame channel corresponding to an AVPLAY, for example, **port0000->win0100(master)**, indicating that video frames are transmitted from port 0000 of the VPSS to WIN0100 over the AVPLAY. **master**, **slave**, and **virtual** indicate the master channel, slave channel, and virtual channel respectively.
AcquireFrame(Try/OK)	Number of attempts to obtain video frames from the VDEC to an AVPLAY and success times
SendFrame(Try/OK)	Number of attempts to send video frames

Parameter	Description
	from an AVPLAY to the VOU and success times. **master**, **slave**, and **virtual** indicate the master channel, slave channel, and virtual channel respectively.
Aud Enable	Working status of an ADEC **HI_TRUE**: enabled **HI_FALSE**: disabled
Adec Type	ADEC type
AudOverflowNum	Count of audio ES buffer overflows
AdecDelayMs	ADEC buffer data amount (ms)
DmxAudChnNum	Number of audio Demux channels
AudPid	Audio PID
Track ID	ID of the audio track corresponding to an AVPLAY. **master**, **slave**, and **virtual** indicate the master channel, slave channel, and virtual channel respectively.
AcquireStream(Try/OK)	Number of attempts and successes that the AVPLAY obtains ES buffers from the Demux
SendStream(Try/OK)	Number of attempts and successes that the AVPLAY releases ES buffers to the Demux
AcquireFrame(Try/OK)	Number of attempts and successes that the AVPLAY obtains decoded audio frames from the ADEC and transmits the frames to the AOU for playing
SendFrame(Try/OK)	Number of attempts and successes that the AVPLAY sends audio frames to the AOU

[Debugging Commands]

- Obtain help information.

  echo help > /proc/msp/avplayxx
- Enable/disable frame rate conversion.

  echo FrcEnable=true|false > /proc/msp/avplayxx

If the audio or video cannot be properly played, locate the fault by checking the following information:

- Whether the tuner is locked and the Demux of the AVPLAY has data. (Locate the TS port of the Demux based on the Demux ID and view the debugging information about demux_port.)
- Whether data is transmitted to the audio and video channel of the AVPLAY. (Check that, in the debugging information about demux_chan, the channel type is an audio and video

one. The channel that has the same PID as the AVPLAY is the channel used by the AVPLAY.)

- Whether the ADEC and VDEC are enabled. (Check the values of Vid Enable and Aud Enable.)
- Whether the working status is Play. (Check the value of CurrStatus.)
- Whether the audio and video PIDs are correctly set. (Check the values of VidPid and AudPid.)
- Whether the audio and video decoding types are correctly set. (Check the values of Vdec Type and Aud Type.)
- If the preceding information is correct, check the debugging information about the ADEC and VDEC of the AVPLAY.

If the audio and video are not synchronized, view the synchronization debugging information using the hSync handle.

2.4 VDEC

[Debugging Information]

```
# cat /proc/msp/vdec00
============================== VDEC0 ================================
Work State: RUN
VpssID: vpss00
VfmwID: vfmw00
Codec ID: MPEG2(0x0)
Mode: NORMAL
Priority: 3
ErrCover: 100
OrderOutput: 0
CtrlOption: 0x0
Capability: NORMAL/FULLHD/H264
--------------------------Stream Information--------------------------
Source: User0
StreamSize(Total/Current): 0x4f2d3dc/0x4fc000
BitRate(bps): 14145256
StreamBuffer(Total/Used/Percent): 0x500000/0x4fec24/99%
--------------------------Picture Information-------------------------
Width*Height: 1280*720
Stride(Y/C): 0x500/0x500
FrameRate(fps): Real(25.19) FrameInfo(23999)
PlayFormat: OTHER(43)
FrmPackingType: Normal
Aspect(User/Decode): 0:0/16:9
FieldMode: Frame
Type: Progressive
```

```
VideoFormat: UNKNOWN
TopFirst: 1
ErrFrame: 2
TypeNum(I/P): 0/2403
DMX/USER->VDEC
GetStreamBuffer(Try/OK) : 9665/5068
PutStreamBuffer(Try/OK) : 5068/5068
VDEC->VFMW
AcquireStream(Try/OK): 52793/5068
ReleaseStream(Try/OK): 4749/4749
VFMW->VPSS
AcquireFrame(Try/OK): 1186/1184
ReleaseFrame(Try/OK): 1183/1183
VPSS->AVPLAY
AcquireFrame(Try/OK): 1186/1184
ReleaseFrame(Try/OK): 1183/1183
# cat /proc/vfmw00
============================= vfmw00 info ==============================
VersionNum:2013091102
VfmwID :0
----------------------------- scd stream info ---------------------------
RawStream(Size/Num):5230258/320
SCDSegStream(Size/Num):1753506/65
SCDSegBuffer(Total/Use/Percent):1965056/1753506/89%
----------------------------- vdh frame info ----------------------------
VDH 0 load:15.9%
Decode(Width*Height):2560*544
Display(Width*Height):2560*544
FrameRate:24.2 fps
VDHFrameBuffer(Total/Use/Percent) :18/17/94%
=========================================================================
==
```

[Analysis]

The working status of the VDEC is recorded. If multiple VDECs work simultaneously, the debugging information about all these VDECs is listed. **VDEC00** and **VFMW00** indicate VDEC 0.

[Parameter Description]

Parameter	Description
Work State	Working status: **RUN**, **STOP**
Codec ID	Decoding type MPEG2, MPEG4, AVS, H263, H264, VC1, VP6, VP6F,

Parameter	Description
	VP6A, SORENSON, RAW, JPEG, VP8, VP9, and OTHER
Mode	Decoding mode NORMAL: All frames are decoded. IP: Only I and P frames are decoded. I: Only I frames are decoded. DROP_INVALID_B: All frames except the B frames that follow the I frames are decoded.
Priority	Decoding priority
ErrCover	Error concealment threshold of the output frames of a VDEC (in the unit of percentage) 0: No frames are output if errors occur. 100: All frames are output regardless of the error percentage. Other values: Frames are output or not based on the specified error percentage.
OrderOutput	Output sequence 0: output in the display sequence 1: output in the decoding sequence
CtrlOption	Special control option of decoding See the definition of **s32CtrlOptions** in HI_UNF_VCODEC_ATTR_S.
Capability(MPEG/H264)	Decoding capability of the VDEC. There are three parameters: Decoder type NORMAL: normal decoder IFRAME: decoder for decoding only I frames Supported resolution QCIF: The resolution of the picture to be decoded is less than or equal to 176 x 144. CIF: The resolution of the picture to be decoded is less than or equal to 352 x 288. D1: The resolution of the picture to be decoded is less than or equal to 720 x 576. 720P: The resolution of the picture to be decoded is less than or equal to 1280 x 720. FULLHD: The resolution of the picture to be decoded is less than or equal to 1920 x 1080. UNKNOWN: unknown Supported protocol NOT_H264: The protocols (supported by the chip) except H.264 are supported. H264: All the protocols (supported by the chip) including H.264 are supported.

Parameter	Description
Stream	Stream transfer status of the decoder Source: The value can be **User** or **Demux**, indicating whether the data is from the user or the Demux. The value followed indicates the handle of the ES buffer or the Demux channel. StreamSize(Total/Current): **Total** is the number of total input bytes of the current decoder. **Current** is the number of bytes of the streams in the VDEC_Firmware buffer area. BitRate(bps): average bit rate of stream input StreamBuffer(Total/Used/Percent): **Total** is the size of the decoded stream buffer. **Used** is the size of the used code buffer. **Percent** is the proportion of the used buffer.
Picture	Parameters of decoded pictures Width*Height: aspect ratio of the original picture Stride(Y/C): stride of the Y and C component data FrameRate(fps): The first value is the counted average frame rate, and the second value is the default frame rate or the frame rate described in the streams. PlayFormat: video standard. It is defined in HI_UNF_ENC_FMT_E. Aspect: aspect ratio of the picture. **User** is the user-defined aspect ratio. **0:0** indicates that the aspect ratio is not set. **Decode** is the aspect ratio in the decoded information. FieldMode: Encoding mode of the frame or field. The value can be **Frame, Top, Bottom**, or **UNKNOWN**, indicating the frame mode, top field mode, bottom field mode, and unknown respectively. Type: video sampling type. **Progressive** indicates progressive sampling, and **Interlace** indicates interlaced sampling. VideoFormat: video format. It is defined in HI_UNF_VIDEO_FORMAT_E. TopFirst: top field first label ErrFrame: number of error video frames TypeNum(P/I): Statistics of picture progressive or interlacing information It can be ignored.
DMX/USER->VDEC	Times that the AVPLAY writes data to the VDEC in ES mode Get(Try/OK): number of attempts/successes to obtain the buffer Put(Try/OK): number of attempts/successes to write data to the buffer It is used only when the user injects data.
VFMW->VPSS	Information about the output of decoded video frames to the VPSS AcquireFrame(Try/OK): number of attempts/successes that the

Parameter	Description
	VPSS obtains decoded video frames ReleaseFrame(Try/OK): number of attempts/successes that the VPSS releases the buffer for storing video frames
VDEC->VFMW	AcquireStream(Try/OK): number of attempts/successes that the firmware obtains video ES data ReleaseStream(Try/OK): number of attempts/successes that the firmware releases the video ES buffer
VPSS->AVPLAY	AcquireFrame(Try/OK): number of attempts/successes that the AVPLAY obtains decoded frames data from the VPSS ReleaseFrame(Try/OK): number of attempts/successes that the VO releases the buffer for storing decoded frames
VersionNum	Version of the VFMW
VfmwID	ID of the current decoding channel
RawStream(Size/Num)	Size and number of raw streams received but not segmented
SCDSegStream	Size and number of streams segmented but not decoded
SCDSegBuffer(Total/Use/Percent)	Total size of the stream buffer for the SCD, and size and proportion of used buffer
VDH 0 load	VDH usage
Decode(Width*Height)	Decoding size
Display(Width*Height)	Display size
FrameRate	Decoding frame rate
VDHFrameBuffer(Total/Use/Percent)	Total number, used number, and used percentage of decoding buffers

To check whether the VDEC works properly, you can check whether the previous buffers are switched successfully and whether data loss occurs. If the times of obtain and release operations are inconsistent, the decoded data and the data sent to the VOU are inconsistent, and the picture may be displayed intermittently.

The video decoding process is as follows:

1. The VDEC obtains ES data from the Demux or ES buffer and sends it to the firmware for decoding (Firmware: Stream Input(VDEC->Firmware)).
2. The firmware saves the decoded 1D or 2D frames in the frame buffer.
3. The VDEC obtains the video frames from the frame buffer and puts the frames in the frame queue (Firmware: Frame Output(Firmware->VDEC)).
4. The VPSS obtains the video frames from the frame queue of the VDEC in a thread and decodes the video frames (Frame Output(VDEC->VPSS)).
5. The AVPLAY attempts to obtain the video frames from the VDEC. The VDEC obtains the video frames from the VPSS and returns them to the AVPLAY.

----End

ErrCover is defined by **u32ErrCover** (one of VDEC attributes).

- 0: No frames are output if errors occur.
- 100: All frames are output.

Continuous increase in the value of **ErrFrame** may result in mosaic or intermittent display.

- If ErrCover is 100, mosaics are displayed.
- If ErrCover is 0, intermittent display occurs.
- The increase of the value of ErrFrame is caused by stream errors. If the value increases, check the correctness of the input streams.

Normally, the values of **Try** and **Ok** increase continuously. If the values remain unchanged, the processing of the item is faulty.

2.5 SYNC

[Debugging Information]

```
#cat /proc/msp/sync00
---------Hisilicon SYNC0 Out Info---------
__________Hisilicon SYNC ATTR____________
SyncPrint :1
SyncRef    :AUDIO
SyncStart.VidPlusTime      :60
SyncStart.VidNegativeTime :-20
SyncStart.bSmoothPlay      :1
SyncNovel.VidPlusTime      :3000
SyncNovel.VidNegativeTime :-3000
SyncNovel.bSmoothPlay      :0
VidPtsAdjust:0
AudPtsAdjust:0
PreSyncTimeoutMs   :1000
bQuickOutput:0
__________Hisilicon PCR____________
CrtStatus    :PLAY
PreSyncStartSysTime :2621862
PreSyncEndSysTime    :2622493
PreSyncFinish :1
BufFundEndSysTime    :2622712
BufFundFinish :1
PreSyncTarget :VID
PreSyncTargetTime    :87017132
PcrFirstCome  :1
PcrFirstSysTime      :2621870
```

```
PcrFirst     :87016494
PcrLast      :87037404
PcrLocalTime  :87037417
PcrAdjustMode :0
PcrAudSyncOK  :1
PcrVidSyncOK  :1
__________Hisilicon VID____________
VidFirstCome:1
VidFirstSysTime    :2622232
VidFirstPts :87017132
VidLastPts  :87036412
VidPreSyncTargetInit:1
VidPreSyncTargetTime:87017132
VidFirstPlay:1
VidFirstPlayTime   :2622712
VidBlockFlag:0
VidBufPercent      :0
VidLocalTime:87036394
VidPcrDiff  :-1022
VidAudDiff  :14
VidDiscard  :0
VidSyndAdjust      :0
VidDiscardCnt      :0
VidRepeatCnt:68
__________Hisilicon AUD____________
AudFirstCome  :1
AudFirstSysTime      :2621955
AudFirstPts :-1
AudLastPts   :87037352
AudPreSyncTargetInit  :1
AudPreSyncTargetTime  :87016640
AudFirstPlay  :1
AudFirstPlayTime     :2623481
AudBlockFlag  :0
AudBufPercent :0
AudLocalTime  :87036380
AudPcrDiff   :-1036
AudBufTime   :1047
AudLastBufTime:1047
AudDiscardCnt :0
AudRepeatCnt  :0
```

[Analysis]

The synchronization information is recorded. The HD chip supports 16 AVPLAYs and simultaneous playback of 16 programs. Each AVPLAY uses one independent sync thread. Each sync thread generates a node in **/proc/msp**. You need to select the correct sync thread during debugging.

[Parameter Description]

Parameter	Description
Hisilicon SYNC ATTR	Synchronization attributes SyncPrint: printing status. **1** indicates printing is enabled and **0** indicates printing is disabled. SyncRef: synchronization control mode. It is defined in HI_UNF_SYNC_REF_E. SyncStart.VidPlusTime: plus time range in the start region during video synchronization SyncStart.VidNegativeTime: negative time range in the start region during video synchronization SyncStart.bSmoothPlay: slow playing enable flag in the start region. **1** indicates that it is enabled and **0** indicates that it is disabled. SyncNovel.VidPlusTime: plus time range in the error region SyncNovel.VidNegativeTime: negative time range in the error region SyncNovel.bSmoothPlay: slow playing enable flag in the error region. **1** indicates that it is enabled and **0** indicates that it is disabled. VidPtsAdjust: adjustment value of the video presentation time stamp (PTS) AudPtsAdjust: adjustment value of the audio PTS PreSyncTimeoutMs: pre-synchronization timeout period bQuickOutput: enable flag of the function of fast outputting the first frame. **1** indicates that it is enabled and **0** indicates that it is disabled.
Hisilicon PCR	PCR information CrtStatus: current working status of the PCR. The value can be **STOP**, **PLAY**, **TPLAY**, or **PAUSE**. PreSyncStartSysTime: pre-synchronization start time PreSyncEndSysTime: pre-synchronization

Parameter	Description
	end time PreSyncFinish: pre-synchronization completion flag. **1** indicates that the pre-synchronization is complete and **0** indicates that the pre-synchronization is not complete. BufFundEndSysTime: audio and video data accumulation end time BufFundFinish: audio and video data accumulation completion label. **1** indicates that it is complete and **0** indicates that it is not complete. PreSyncTarget: pre-synchronization target. The value can be **AUD**, **VID**, or **PCR.** PreSyncTargetTime: pre-synchronization target time PcrFirstCome: whether the PCR comes. **1** indicates that it comes and **0** indicates that it does not come. PcrFirstSysTime: system time when the PCR comes for the first time PcrFirst: time value of the first PCR PcrLast: time value of the last PCR PcrLocalTime: local system time set based on the PCR PcrAdjustMode: PCR synchronization mode. The synchronization mode is valid only when **SyncRef** is PCR. **0** indicates that the local clock is modified based on the PCR. **1** indicates that the local clock is modified based on the APTS. PcrAudSyncOK: whether the PCR and audio synchronization is complete 1: complete 0: not complete PcrVidSyncOK: whether the PCR and video synchronization is complete 1: complete 0: not complete
Hisilicon VID	Video information VidFirstCome: whether the first video frame comes. **1** indicates that it comes and **0** indicates that it does not come. VidFirstSysTime: system time when the first video frame comes VidFirstPts: PTS of the first video frame

Parameter	Description
	VidLastPts: PTS of the previous video frame VidPreSyncTargetInit: whether the video comes during pre-synchronization. **1** indicates yes, and **0** indicates no. VidPreSyncTargetTime: time that the video comes during pre-synchronization VidFirstPlay: whether the first video frame is played. **1** indicates that it is played and **0** indicates that it is not played. VidFirstPlayTime: system time when the first video frame is played VidBlockFlag: whether the video buffer is blocked. **1** indicates that it is blocked and **0** indicates that it is not blocked. VidBufPercent: Video buffer percentage VidDiscard: whether the video is in the frame discarding mode. **1** indicates that it is in the frame discarding mode and **0** indicates that it is not in the frame discarding mode. VidSyndAdjust: whether the video is synchronized. **1** indicates yes, and **0** indicates no. VidLocalTime: local time of the video PTS VidPcrDiff: time difference between the video and PCR during synchronization VidAudDiff: time difference between the video and the audio during synchronization VidDiscardCnt: number of discarded video frames during synchronization VidRepeatCnt: number of repeated video frames during synchronization
Hisilicon AUD	Audio information AudFirstCome: whether the first audio frame comes. **1** indicates that it comes and **0** indicates that it does not come. AudFirstSysTime: system time when the first audio frame comes AudFirstPts: PTS of the first audio frame AudLastPts: PTS of the previous audio frame AudPreSyncTargetInit: whether the audio comes during pre-synchronization. **1** indicates yes, and **0** indicates no. AudPreSyncTargetTime: time that the audio

Parameter	Description
	comes during pre-synchronization AudFirstPlay: whether the first audio frame is played. **1** indicates that it is played and **0** indicates that it is not played. AudFirstPlayTime: system time when the first audio frame is played AudBlockFlag: whether the audio buffer is blocked. **1** indicates that it is blocked and **0** indicates that it is not blocked. AudBufPercent: audio buffer percentage AudLocalTime: local time of the audio PTS AudPcrDiff: time difference between the audio and PCR during synchronization AudBufTime: play time of the AO buffer data AudLastBufTime: previous play time of the AO buffer data AudDiscardCnt: number of discarded audio frames during synchronization AudRepeatCnt: number of repeated audio frames during synchronization

[Debugging Commands]

- Obtain help information.

  echo help > /proc/msp/sync00
- Modify synchronous reference information.

  echo SyncRef = audio|pcr|scr|none > proc/msp/syncxx
- Modify synchronous start intervals.

  echo SyncStart.VidPlusTime = xxx > /proc/msp/syncxx

  echo SyncStart.VidNegativeTime = xxx > /proc/msp/syncxx

  echo SyncStart.bSmoothPlay = true|false > /proc/msp/syncxx
- Modify synchronous stop intervals.

  echo SyncNovel.VidPlusTime = xxx > /proc/msp/syncxx

  echo SyncNovel.VidNegativeTime = xxx > /proc/msp/syncxx

  echo SyncNovel.bSmoothPlay = true|false > /proc/msp/syncxx
- Modify the pre-synchronous timeout duration.

  echo PreSyncTimeoutMs = xxx > /proc/msp/syncxx
- Modify the quick output setting for the first frame.

  echo bQuickOutput = true|false > /proc/msp/syncxx

[Debugging Description]

VidAudDiff indicates the difference between audio PTS and video PTS. If the difference is beyond the start interval [SyncStart.VidNegativeTime, SyncStart.VidPlusTime], videos are not played in synchronous mode. Perform synchronous adjustment as required.

VidDiscardCnt and **VidRepeatCnt** indicate the current synchronous video adjustment process. If the value of **VidDiscardCnt** increases, frame loss occurs. If the value of **VidRepeatCnt** increases, video frames are displayed repeatedly. Either of the preceding conditions causes video play errors.

2.6 WINDOW

[Debugging Information]

```
cat /proc/msp/windowXXYY
------------------------------Win0100[Z=0]-------------------------------
------------Win Info-------------------|-------------Frame Info----------
Enable           :True             |Type/PixFmt      :2D       /NV21
State            :Run              |Rotation         :Rotation_00
Type             :Main             |W/H(Aspect W:H) :1920/1080(  16: 9)
*LayerID         :0                |Disp(X/Y/W/H)     : 0/ 0/1920/1080
AspectRatioConvert :Full             |FrameRate        :500.50000
CustAspectRatio    :0 :0           |ColorSpace       :BT601_YUV_LIMITED
Crop             :False            |Fieldmode(Origin) :Frame(Top)
Crop(L/T/R/B)     : 0/ 0/ 0/ 0 |OriRect(X/Y/W/H)  :0/0/720/576
In  (X/Y/W/H)     : 0/ 0/ 0/ 0 |FrameIndex       :0x63
Out(X/Y/W/H)      : 0/ 0/ 0/ 0 |SrcPTS/PTS       :0xffffffff/0x4a41d52
DispMode/RightFirst:2D       /False    |PlayTime         :1
*Masked          :False            |FieldMode        :All
AttachSource      :False            |Fidelity         :0
*CallBack(Acquire) :N                |YAddr/YStride    :0x24e41000/0x900
*CallBack(Release) :Y                |CAddr/CStride    :0x250a0800/0x900
*CallBack(SetAttr) :Y                |
SlaveWinID       :0000             |
bQuickoutMode     :0               |
bVirtualCoordinate :1                |
bDcmp            :0                |
bMute            :0                |
-----------------------Buffer State---------------------------------
Queue(Try/OK)      :213/213
Release(Try/OK)     :194/194
Config           :155
Underload         :81631
Discard          :58
UndispFrame(Q/DQ) :4/4
*FieldUnmatchCnt    :40880
--------------------------------------------------------------------
```

```
BufferQueue[state, FrameID]
(State: 1,Empty[15]; 2,Write[0]; 3,ToDisp[0]; 4,Disp[1]; 5,Disped[0])
[1,0x62] [1,0x5c] [1,0x61] [1,0x62]
[1,0x61] [1,0x5c] [1,0x5e] [4,0x63]
[1,0x5d] [1,0x5b] [1,0x5d] [1,0x5f]
[1,0x60] [1,0x5f] [1,0x5e] [1,0x60]
```

[Analysis]

The status of a window is recorded. Each window generates one node in **/proc/msp**. You need to select the correct window during debugging.

window*XXYY* is available in the **msp** directory. *XX* is the ID of a display channel, and *YY* is the window ID, for example, **window0100**. **01** (bit[15:8]) indicates that the window is created based on DISPLAY1, and **00** (bit[7:0]) indicates that the window ID is 0.

[Parameter Description]

Window attribute configurations

Parameter	Description
Enable	**TRUE**: enabled **FALSE**: disabled
State	Current working status **Run** **Pause** **FreezeLast** **FreezeBlack**
Type	Window type **Display**: independent display window **Virtual**: virtual window Main: main window in same-source display Slave: slave window in same-source display
LayerID	ID of the video layer used by a window
AspectRatioConvert	Aspect ratio conversion mode **Full**: to fully fill the screen **LetterBox**: to add black borders **PanAndScan**: to crop the screen **Combined**: to combine pictures **FullHori**: to fully fill the screen in the horizontal direction **FullVert**: to fully fill the screen in the vertical direction **Customer**: user-defined setting (the aspect ratio specified by **CustAspectRatio** takes effect forcibly)
CustAspectRatio	User-defined aspect ratio for video display

Parameter	Description
Crop	**TRUE**: enabled **FALSE**: disabled
In (X/Y/W/H)	Coordinates of the video image display area (available only when **Crop** is **FALSE**)#
Crop(L/T/R/B)	Width of the cropped area at the top, bottom, left, and right of video images (available only when **Crop** is **TRUE**)
Out (X/Y/W/H)	Coordinates of the video image display area on the virtual screen
DispMode	Display mode 2D FPK SBS_HALF TAB FILED_ALTE LINE_ALTE SBS_FULL L_DEPTH LDEP_GDEP
RightFirst	**TRUE**: right eye first, with a display sequence from the right eye to the left eye **FALSE**: left eye first, with a display sequence from the left eye to the right eye
Masked	**TRUE**: enabled **FALSE**: disabled
AttachSource	TRUE: The window is bound to a data source. FALSE: The window is not bound to a data source.
CallBack(Acquire)	**TRUE**: video frame request function bound **FALSE**: video frame request function not bound#
CallBack(Release)	**Y**: video frame release function bound **N**: video frame release function not bound
CallBack(SetAttr)	**Y**: video frame attribute setting function bound **N**: video frame attribute setting function not bound
SlaveWinID	**FFFF**: no slave window Other value: ID of a slave window
bQuickoutMode	TRUE: The window is in quick output mode. FALSE: The window is in normal mode.
bVirtualCoordinate	TRUE: The window uses the virtual screen coordinate system

Parameter	Description
	configured in DISPLAY. FALSE: The window uses the physical coordinate system, which is related to the current DISPLAY output resolution.
bDcmp	TRUE: The window decompression function is enabled. FALSE: The window decompression function is disabled.
bMute	TRUE: The window black frame function is enabled. FALSE: The window black frame function is disabled.

Current video frame information

Parameter	Description
Type	Frame type **NotStereo** **SideBySide** **TopAndBottom** **MVC**
PixFmt	Pixel format
Rotation	Rotation degree Rotation_00: rotate by 0 degree Rotation_90: rotate by 90 degrees Rotation_180: rotate by 180 degrees Rotation_270: rotate by 270 degrees
W/H(WvsH)	**W**: horizontal pixels of a picture **H**: vertical pixels of a picture **WvsH**: aspect ratio of a picture
Disp(X/Y/W/H)	Display area of a picture
FrameRate	Video frame rate
ColorSpace	Video color space
Fieldmode(Origin)	Original field flag and current field flag of a picture **Top** **Bottom** **All**
OriRect(X/Y/W/H)	Original display area of a picture (resolution)
FrameIndex	Frame sequence number of a picture
SrcPTS/PTS	Original timestamp and current timestamp of a picture
PlayTime	Times of displaying a picture

Parameter	Description
FieldMode	Data carried by the current video frame: **Top**, **Bottom**, and **All**
Fidelity	Fidelity flag
Y/CAddr	Physical addresses for the luminance and chrominance components of a picture
Y/CStride	Stride for the luminance and chrominance components of a picture

Window buffer status information

Parameter	Description
Queue(Try/OK)	Count of video frames pushed by an application and count of video frames successfully pushed
Release(Try/OK)	Count of video frames released by a window and count of video frames successfully released#
Config	Count of displayed video frames
Underload	Count of underloads
Discard	Count of ignored video frames#
UndispFrame(Q/DQ)	I/O count of video frames that do not need to be displayed
FieldUnmatchCnt	Count of mismatch between video field sequence and display interface field sequence
Empty	Number of empty buffers
Write	Number of buffers for executing the write operation
ToDisp	Number of buffers to be displayed
Disp	Number of buffers that are being displayed
Disped	Number of buffers to be released after displaying
Empty(Read/Write)	Read/write pointer of an idle queue
Full(Read/Write)	Read/write pointer of a video frame queue
BQState [state, FrameID]	Buffer queue status **State**: **1, Empty**; **2, Write**; **3, ToDisp**; **4, Disp**; **5, Disped** **FrameID**: ID of the current video frame held

If the video flickers violently or rollback occurs when the video is being played, check whether the number of underloads increases. If the number of underloads increases, the stream underruns and images are repeatedly displayed. Typically, the number of underloads should not increase and may increase only when the video playback starts.

[Debugging Commands]

- Obtain help information.

echo help> /proc/msp/winXXXX

- Pause or resume a window.
  echo pause on/off   > /proc/msp/winXXXX
- Reset a window by using black frames or still frames.
  echo reset black/still   > /proc/msp/winXXXX
- Freeze a window by using black frames or still frames.
  echo freeze   black/still/off   > /proc/msp/winXXXX
- Move a window up or down.
  echo order up/down   > /proc/msp/winXXXX
- Enable or disable the quick output mode.
  echo quick on/off   > /proc/msp/winXXXX
- Set the depth of field for a window. (available only in 3D mode)
  echo depth X > /proc/msp/winXXXX
- Capture the current video image and save it to the specified absolute path.
  echo capture path   > /proc/msp/winXXXX
- Set window rotation.
  echo rota   0/90/180/270   > /proc/msp/winXXXX
- Set window flipping.
  echo flip   hori/vert   > /proc/msp/winXXXX

2.7 DISPLAY

[Debugging Information]

```
# cat /proc/msp/disp1
---------Hisilicon DISP 1 State---------
State             :Open
Formt/DispMode     :1080i50/2D
RightEyeFirst      :Disable
VirtualScreen      :1280/720
Offset(L/T/R/B)    :0/0/0/0
AspectRatioMode    :Customer Setting
AspectRatio       :16:9
ColorSpace        :BT709_YUV_LIMITED->BT709_YUV_LIMITED
Bright            :50
Contrast          :50
Saturation        :50
Hue              :50
Background (R/G/B)  :0x0/0x0/0x0
Zorder(Bot->Top)    :VIDEO->GFX
```

```
AttachRole        :source
AttachDisp        :display0
Interface         :YPbPr0 (  0/  1/  3) HDMI0
InitCount         :2
OpenCnt[User/Kernel]:1/1
LowbandCount       :0
------------------CAST Info---------------------------------
State            :Enable
Crop             :False
CropRect(L/T/R/B) :0/0/0/0
Resolution        :1280/720
PixelFormat        :NV21
FrameRate         :2500
LowDelay          :Enable
MemoryType        :DispAllocate
BufferNumber       :5
BufferWidth/Height  :1280/720
BufferSize        :0x0
BufferStride       :0
bAttached         :0
------------------Buffer---------------------------------
Acquire(Try/OK)    :0/0
Release(Try/OK)    :0/0
---------------------------------------------------------
BufferQueue: [state, FrameID]
(State: 1,Empty[0]; 2,Write[0]; 3,Full[5]; 4,Use[0]
[3,0x1] [3,0x2] [3,0x3] [3,0x4]
[3,0x5]
```

[Analysis]

Configurations and running status of the display module are recorded. An HD chip supports simultaneous running of two display channels. **disp0** is the Proc of DISPLAY0, and **disp1** is the Proc of DISPLAY1.

[Parameter Description]

Display debugging information

Parameter	Description
Open	Channel status Open: enabled Close: disabled
Formt	Display channel output format 1080P60 1080P50

Parameter	Description
	1080P30 1080P25 1080P24 1080i60 1080i50 720P60 720P50 576P50 480P60 PAL NTSC …
DispMode	Display mode 2D FPK SBS_HALF TAB FILED_ALTE LINE_ALTE SBS_FULL L_DEPTH LDEP_GDEP
RightEyeFirst	Whether to enable right eye first **TRUE**: right eye first **FALSE**: left eye first
VirtualScreen	Virtual screen resolution
Offset (L/T/R/B)	Offset of the virtual screen on the physical screen in the left, top, right, and bottom directions
AspectRatioMode	Aspect ratio mode of a display device **Auto**: output resolution **User**: user-defined
AspectRatio	Aspect ratio of a display device
ColorSpace	Overlay color space and output color space BT601_YUV_LIMITED BT601_YUV_FULL BT601_RGB_LIMITED BT601_RGB_FULL NTSC1953

Parameter	Description
	BT470_M BT470_BG BT709_YUV_LIMITED BT709_YUV_FULL BT709_RGB_LIMITED BT709_RGB_FULL REC709 SMPT170M SMPT240M BT878 XVYCC JPEG
Bright	Brightness ranging from 0 to 100
Contrast	Contrast ranging from 0 to 100
Saturation	Saturation ranging from 0 to 100
Hue	Hue ranging from 0 to 100
Background (R/G/B)	R/G/B component settings of the background, ranging from 0 to 255
Zorder(Bot->Top)	Video layer and graphics layer sequence
AttachRole	Binding relationship **source**: source display channel **destination**: destination display channel#
AttachDisp	ID of the bound display channel
Interface	VO interfaces **CVBS0**: VDAC number. The interface is unused when this parameter is set to **255**. **YPbPr0**: VDAC numbers of Y, Pb, and Pr signals **HDMI0**: HDMI number
InitCount	Count of initialization
OpenCnt[User/Kerne]	Count of times when a display channel is enabled (in the user or kernel state)
LowbandCount	Count of low-bandwidth interrupts

CAST debugging information

Parameter	Description
State	CAST status **Enable**: enabled **Disable**: disabled
Crop	Whether to crop the screen **TRUE**: yes **FALSE**: no
CropRect(L/T/R/B)	Pixel settings of the cropped area in the left, top, right, and bottom directions (available only when **Crop** is **TRUE**)
Resolution	Picture resolution
PixelFormat	Picture pixel format
FrameRate	Picture frame rate
LowDelay	Low delay **TRUE**: enabled **FALSE**: disabled
MemoryType	Frame buffer type **DispAlloc**: determined by the CAST **UserAlloc**: determined by users
BufferNumber	Frame buffer size
BufferWidth/Height	Width and height of the frame buffer, with a unit of pixel. The value of **Resolution** shall not be larger than the value of **BufferWidth/Height**.
BufferSize	Size of each picture frame buffer
BufferStride	Start positions of adjacent rows in a picture
bAttached	Whether to use the binding mode TRUE: The cast is bound to the consumer. Frame reception and transmission are implemented by using callback functions within the driver. FALSE: separated mode. Frame reception and transmission are implemented by calling APIs at the UNF layer.
Acquire(Try/OK)	Times of requesting pictures by a user and success times
Release(Try/OK)	Times of releasing pictures by a user and success times
Empty	Number of empty buffers
Full	Number of full buffers
Use	Number of buffers in use
[state, FrameID]	Buffer queue status **state**: **1, Empty**; **2, Write**; **3, Full**; **4, Use**

Parameter	Description
	FrameID: picture ID

[Debugging Commands]

- Obtain help information.

  echo help > /proc/msp/dispX
- Switch the output format.

  echo fmt 1080i50/720p50/pal/ntsc/... > /proc/msp/dispX
- Switch the output mode.

  echo 3d fp/sbs_hf/tab> /proc/msp/dispX
- Set right eye first in 3D mode.

echo rf on/off > /proc/msp/dispX

- Set the brightness.

  echo bright X > /proc/msp/dispX
- Set the contrast.

  echo contrast X > /proc/msp/dispX
- Set the hue.

  echo hue X > /proc/msp/dispX
- Set the saturation.

  echo satu X > /proc/msp/dispX
- Set the pixels for the left offset on the screen.

  echo left X > /proc/msp/dispX
- Set the pixels for the top offset on the screen.

  echo top X > /proc/msp/dispX
- Set the pixels for the right offset on the screen.

  echo right X > /proc/msp/dispX
- Set the pixels for the bottom offset on the screen.

  echo bottom X > /proc/msp/dispX
- Set the overlay sequence for moving video layers up or down.

  echo video up/down > /proc/msp/dispX
- Retain the default value of the reset parameter.

  echo reset > /proc/msp/dispX
- Capture the screen and save it to the specified absolute path.

2.8 ADEC

[Debugging Information]

```
# cat /proc/msp/adec00
--------------------------- ADEC[00] State --------------------------
```

```
WorkState:start
CodecID:0x202f1011
DecoderName:Dolby TrueHD Decoder
Description:hisi_truehd
DecodeThreadID :2537
Volume :100
SampleRate:96000
BitWidth:24
Channels:2
PcmSamplesPerFrame:1840
BitsBytePerFrame:0x0
StreamFormat:non-packet
TryDecodetimes :1088
FrameNum(Total/Error):767/0
StreamBuf(Total/Use/Percent)(Bytes):262144/205883/78%
StreamBuf(readPos/writePos):0xdbcc/0x9000
OutFrameBuf(Total/Use/Percent) :8/7/87%
GetBuffer(Try/OK):4967/3529
PutBuffer(Try/OK):3529/3529
SendStream(Try/OK):0/0
ReceiveFrame(Try/OK):767/760
PtsLostNum:0
DecodeThreadExecTimeOutCnt:1
DecodeThreadScheTimeOutCnt:0
```

[Analysis]

The ADEC status is recorded.

[Parameter Description]

Parameter	Description
WorkState	Working status of the ADEC **start**: working **stop**: not working
Codec ID	Audio decoding type The value in the brackets is the value of **HA_AUDIO_ID**.
DecoderName	ADEC name
Description	Decoder description
DecodeThreadID	ID of a decoding thread. It can be ignored.
Volume	Volume of the decoded audio. It is not the volume of the output audio. The value is read-only and cannot be configured by using any interface.

Parameter	Description
SampleRate	Sampling rate of the PCM data decoded from streams. It is determined by the stream attributes.
BitWidth	Bit width of the PCM data decoded from streams. It is determined by the stream attributes.
Channels	Number of audio channels for the PCM data decoded from the stream. It is determined by the stream attributes.
PcmSamplesPerFrame	Number of sampling points for each PCM data frame decoded from the stream. It can be ignored.
BitsBytePerFrame	Size of each transparent data frame decoded from the stream. It can be ignored.
StreamFormat	Whether to enable the packet mode
TryDecodetimes	Number of attempts to decode the undecoded data in the decoding buffer. It can be ignored.
FrameNum(Total/Error)	Number of total and error frames that the ADEC decodes
StreamBuf(Total/Use/Percent)(Bytes)	Total size, used size, and used percentage of the input frame buffer of the ADEC
StreamBuf(readPos/writePos)	Position of the read/write pointer of the ADEC input frame buffer
OutFrameBuf(Total/Use/Percent)	Total size, used size, and used percentage of the output frame buffer of the ADEC
GetBuffer(Try/OK)	**Try**: number of attempts to obtain the ES buffer from the ADEC **OK**: number of successes to obtain the ES buffer from the ADEC It is valid in ES mode.
PutBuffer(Try/OK)	**Try**: number of attempts to release the ES buffer to the ADEC **OK**: number of successes to release the ES buffer to the ADEC It is valid in ES mode.
SendStream(Try/OK)	**Try**: number of attempts to send audio ES data from the AVPLAY to the ADEC **OK**: number of successes to send audio ES data from the AVPLAY to the ADEC The AVPLAY obtains ES data from the Demux and sends it to the ADEC for decoding. It is valid in TS mode.
ReceiveFrame(Try/OK)	Try: **Try**: number of attempts to obtain decoded audio frames from the ADEC **OK**: number of successes to obtain decoded audio frames from the ADEC The AVPLAY obtains audio frames from the ADEC and

Parameter	Description
	sends them to the AOU for playing.
PtsLostNum	Number of PTS losses
DecodeThreadExecTimeOutCnt	Timeout (30 ms) count for executing an ADEC decoding thread
DecodeThreadScheTimeOutCnt	Timeout (30 ms) count for calling an ADEC decoding thread

In ES mode, HI_UNF_AVPLAY_GetBuf is called to send audio ES data to the ADEC for decoding and playing.

- If the value of Try of GetBuffer remains unchanged, HI_UNF_AVPLAY_GetBuf is not called to inject data.
- If the value of OK of PutBuffer remains unchanged, data injection fails possibly because the buffer of the ADEC is full.

[Debugging Information]

```
#echo save_es start > /proc/msp/adec00
#echo save_es stop > /proc/msp/adec00
#echo save_pcm start > /proc/msp/adec00
#echo save_pcm stop > /proc/msp/adec00
#echo help > /proc/msp/adec00
```

[Analysis]

The ES data before decoding is recorded by running the following command:

echo save_es start|stop > /proc/msp/adec*XX* (*XX* indicates the ADEC channel ID.)

- Start recording.

  echo save_es start > /proc/msp/adec00

  Send the command for saving ES data to ADEC00. (00 is the ADEC channel ID.)
- Stop recording.

  echo save_es stop > /proc/msp/adec00
- View the following files in the specified directory after recording is completed:

  adec0_00.es, **adec0_01.es**
- **adec0_00.es**: first ES data saved for ADEC00
- **adec0_01.pcm**: second ES data saved for ADEC00

Record the PCM data after decoding by running the following command:

echo save_pcm start|stop > /proc/msp/adec*XX*

- Start recording.

  echo save_pcm start > /proc/msp/adec00

  Send the command for saving decoded PCM data to ADEC00.
- Stop recording.

echo save_pcm stop > /proc/msp/adec00

- View the following files in the specified directory after recording is completed: **adec0_00.pcm**, **adec0_01.pcm**
- **adec0_00.pcm**: first PCM data saved for ADEC00
- **adec0_01.pcm**: second PCM data saved for ADEC00

Obtain help information by running the following command:

echo help > /proc/msp/adecXX

2.9 AO

[Debugging Information]

```
# cat /proc/msp/sound0
--------------------------Sound[0]Status--------------------------
SampleRate:48000
SPDIF Status:UserSetMode(RAW) DataFormat(DD)
HDMI Status:UserSetMode(RAW) DataFormat(DDP)
--------------------------OutPort Status-----------------------------
ADAC0: Status(start), Mute(off), Volume(00dB), TrackMode(STEREO)
SampleRate(048000), Channel(02), BitWidth(16), Engine(PCM), AOP(0x0),
PortID(0x12)
DmaCnt(000850), BufEmptyCnt(000002), FiFoEmptyCnt(000000)
SPDIF0: Status(start), Mute(off), Volume(00dB), TrackMode(STEREO)
SampleRate(048000), Channel(02), BitWidth(16), Engine(SPDIF RAW), AOP(0x1),
PortID(0x21)
DmaCnt(000828), BufEmptyCnt(000000), FiFoEmptyCnt(000001)
HDMI0: Status(start), Mute(off), Volume(00dB), TrackMode(STEREO)
SampleRate(192000), Channel(02), BitWidth(16), Engine(SPDIF RAW), AOP(0x3),
PortID(0x20)
DmaCnt(000756), BufEmptyCnt(000000), FiFoEmptyCnt(000000)
I2S0: Status(start), Mute(off), Volume(00dB), TrackMode(STEREO)
SampleRate(048000), Channel(02), BitWidth(16), Engine(PCM), AOP(0x4),
PortID(0x10)
DmaCnt(000850), BufEmptyCnt(000000), FiFoEmptyCnt(000000)
--------------------- Cast Status----------------------
Cast(0): *Aop(0x5), Status(start), UserEnable(On), Weight(100/100),
Mute(off)
SampleRate(048000), Channel(02), BitWidth(16)
MaxFrameNum(08), SamplePerFrame(01024), AcquireFrame(On)
-------------------- Track Status----------------------
Track(0): Type(master), Status(start), Weight(100/100), ChannelMode(STEREO),
Mute(off) StartThresHold(050ms)SpeedRate(00), AddMuteFrames(0000),
SendCnt(Try/OK)(001112/000322)
```

```
AIP(0): Engine(PCM), SampleRate(048000), Channel(02), BitWidth(16),
DataFormat(PCM) , Priority(low), FifoBypass(off)
EmptyCnt(000000), EmptyWarningCnt(000000), Latency/Threshold(378ms/400ms)
AIP(1): Engine(SPDIF RAW), SampleRate(048000), Channel(02), BitWidth(16),
DataFormat(DD) , Priority(low), FifoBypass(off)
EmptyCnt(000000), EmptyWarningCnt(000000), Latency/Threshold(388ms/400ms)
AIP(2): Engine(HDMI RAW), SampleRate(192000), Channel(02), BitWidth(16),
DataFormat(DDP) , Priority(low), FifoBypass(off)
EmptyCnt(000000), EmptyWarningCnt(000000), Latency/Threshold(348ms/400ms)
```

[Analysis]

Tcat /proc/msp/sound*N* indicates the status of audio output device *N*.

[Parameter Description]

Parameter	Description
SampleRate	Sampling rate of an output device
SPDIF Status	SPDIF output status UserSetMode(RAW): user-defined SPDIF output mode DataFormat(DD): SPDIF output data format, such as PCM, DD, and DTS
HDMI Status	HDMI output status UserSetMode(RAW): user-defined HDMI output mode DataFormat(DDP): HDMI output data format, such as PCM, DD, DDP, DTS, DTSHD, and TRUEHD
ADAC0 (SPDIF0, HDMI0)	Status of ports bound to the sound device: Status(start): start/stop status of a port Mute(off/0x0): mute state of a port. It can be set to **On** or **Off**. **Off** indicates that the mute state is disabled. Volume(00dB): volume setting for a port TrackMode(STEREO): audio channel mode for a port SampleRate(048000): output sampling rate for a port Channel(02): number of output audio channels for a port BitWidth(16): output sampling bit width for a port Engine(PCM): AOE bound to a port. It can be ignored. AOP(0x0): AOP bound to a port. It can be ignored. PortID(0x12): Output channel bound to a port. It can be ignored. DmaCnt(002245): number of times that the DMA transfers data. If the value of this parameter increases, port outputs are available. BufEmptyCnt(000000): number of buffer underflow interrupts FiFoEmptyCnt(000000): number of times that no data is interrupted in FIFO
Cast(n)	Cast status (*n* indicates the Cast ID)

Parameter	Description
	Aop(n): ID of the Aop used by Cast, which can be ignored Status(start): Cast status UserEnable(On): whether Cast is enabled Weight(100/100): Cast weight (that is, volume of the Cast audio-left and audio-right channels) Mute(off): port mute status. The value is **On** or **Off**. **Off** indicates that the port mute function is disabled. SampleRate(048000): output sampling rate of the port Channel(02): number of output channels of the port BitWidth(16): output sampling bit width of the port MaxFrameNum(08): maximum number of frames SamplePerFrame(01024): number of PCM sampling points read each time AcquireFrame(On): current read status. The value is **On** or **Off**.
Track(n)	Track status (**n** indicates the track ID) Type(master): track type Status(start): track start, stop, or pause status Weight(100/100): track weight of the audio-left/audio-right channel, that is, track volume ChannelMode(STEREO): track channel mode Mute(off): track mute status. **off** indicates that the track mute function is disabled, that is, audio is output. The value can be **on** or **off**. SpeedRate(00): track speed rate AddMuteFrames(0000): number of transparent SID frames added SendCnt(Try/OK): number of attempts and successes to send data to the track AIP(*n*): status of the AIP used by the track (*n* indicates the AIP ID). It can be ignored. Engine(PCM): AOE bound to the AIP SampleRate(048000): input sampling rate of the AIP Channel(02): AIP input channel ID BitWidth(16): input sample bit width for the AIP DataFormat(PCM): AIP data format Priority(low): AIP priority FifoBypass(off): whether the AIP data passes the FIFO EmptyCnt(000000): number of times that the AIP buffer is empty EmptyWarningCnt(000000): number of times that no data is available in AIP FIFO when data is available in the AIP buffer Latency/Threshold(328ms/400ms): data amount in the AIP buffer and threshold of data amount (ms) StartThresHold: track start threshold. Only master track has this

Parameter	Description
	attribute (unit: ms).

[Debugging Information]

```
#echo save_track 0 start > /proc/msp/sound0
#echo save_track 0 stop > /proc/msp/sound0
#echo save_sound start > /proc/msp/sound0
#echo save_sound stop > /proc/msp/sound0
#echo help > /proc/msp/sound0
```

[Analysis]

Record the PCM data of a track by running the following command:

echo save_track track_id start|stop > /proc/msp/sound*N* (**track_id** indicates the track ID, and *N* indicates the sound channel ID.)

- Start recording.

  echo save_track 0 start > /proc/msp/sound0

  Send the command for saving track PCM data to Sound0.
- Stop recording.

  echo save_track 0 stop > /proc/msp/sound0
- View the file with the name in the following format in the specified directory after recording is completed:

  track0_*hh_mm_ss*.pcm: Stores the PCM data of track 0 when the system time is *hh* (hour):*mm* (minute):*ss* (second).

Record the PCM data of a sound channel by running the following command:

echo save_sound start|stop > /proc/msp/sound*N* (*N* indicates the sound channel ID.)

- Start recording.

  echo save_sound start > /proc/msp/sound0

  Send the command for saving sound PCM data to Sound0.
- Stop recording.

  echo save_sound stop > /proc/msp/sound0
- View the file with the name in the following format in the specified directory after recording is completed:

  sound0_*hh_mm_ss*.pcm: Stores the PCM data of sound 0 when the system time is *hh* (hour):*mm* (minute):*ss* (second).

Obtain help information by running the following command:

echo help > /proc/msp/soundN

[HDMI/S/PDIF output mode configuration]

Set the HDMI/S/PDIF output mode forcibly as follows:

- Set the HDMI output of sound 0 to decoding output by running the following command:

echo hdmi=pcm > /proc/msp/sound0

- Set the HDMI output of sound 0 to automatic (automatically identifying peripheral capabilities) by running the following command:

  echo hdmi=auto > /proc/msp/sound0

- Set the HDMI output of sound 0 to blu-ray downgrading output by running the following command:

  echo hdmi=hbr2lbr > /proc/msp/sound0

- Set the HDMI output of sound 0 to passthrough output by running the following command:

  echo hdmi=raw > /proc/msp/sound0

- Set the S/PDIF output of sound 0 to decoding output by running the following command:

  echo spdif=pcm > /proc/msp/sound0

- Set the S/PDIF output of sound 0 to passthrough output by running the following command:

  echo spdif=pcm > /proc/msp/sound0

2.10 HiGo

[Debugging Information]

```
# cat /proc/msp/higo/sys
Time: Oct 25 2011 09:17:57
Version: 3.0.0.1 a12
SurfaceUsed: 1
LayerUsed: 1
MmzMemUsed: 50148B
SysMemUsed: 732B
# cat /proc/msp/higo/layer
----------------------- HIGO_LAYER_HD_0 --------------
Visible: TURE
Format: 8888
FlushType: TRIPPLE
Position: (0,0)
Deflicker: AUTO
CanvasSurface: type(Default) resolution(1280,720)
DisplaySurface: resolution (1280, 720)
Alpha:255
BGColor: 0x0
Colorkey: 0x0
StereoDepth: 0x0
# cat /proc/msp/higo/surface
number width height pitch phyaddrsize(KB) formatalpha colorkey
0000128072051200x87823000 3600HIGO_PF_8888 255false
```

[Analysis]

All the debugging information about the HiGo is recorded.

[Parameter Description]

Parameter	Description
sys info	System information Time: version compilation time Version: version number SurfaceUsed: number of used surfaces LayerUsed: number of used layers MmzMemUsed: size of the used media memory zone (MMZ) memory SysMemUsed: size of the used OS memory
layer Info	Visible: display status Format: pixel format of the layer FlushType: refresh type Position: start position Deflicker: anti-flicker level CanvasSurface: type and resolution of a canvas surface DisplaySurface: resolution of a display surface Alpha: alpha information BGColor: background color Colorkey: colorkey value StereoDepth: depth of field in 3D mode
surface information	number: surface index ID width: surface width height: surface height pitch: surface pitch phyaddr: physical address of the surface size(KB): surface memory size format: surface pixel format alpha: alpha information colorkey: colorkey information

2.11 HiFB

[Debugging Information]

Information about the HiFB is recorded. hifb0, hifb1, and hifb2 indicate the HD Proc information, and hifb4 indicates the SD Proc information.

```
# cat /proc/msp/hifb0
LayerId:layer_hd_0
ShowState:ON
ColorFormat::ARGB8888
Stride:5120
Offset:(0,0)
Resolution(real/virtual/max):(1280,720)/(1280,720)/(1920,1080)
MemSize:3600 KB
StartPosition:(114, 69)
BufferMode:single
PixelAlpha:enable(TRUE) alpha0(0x0) alpha1(0xff)
GlobalAlpha:0xff
Colorkey :enable(FALSE) value(0)
Deflicker:enable(FALSE) mode(NONE) level(NONE)
3DMode:input(Mono) output(Mono)
DisplayResolution:(1280, 720)
CanavasAddr:0x0
CanavasUpdateRect:(0,0,1280,720)
CanvasResolution :(0,0)
CanvasPitch:0
CanvasFormat:RGB565
```

[Analysis]

The Proc information about the HiFB consists of two parts:

- hifb0
  Corresponds to the HD graphics layer 0.
- hifb4
  Corresponds to the SD graphics layer 0.

[Parameter Description]

Parameter	Description
HIFB Info	LayerId: ID of a picture layer ShowState: layer status. **on** indicates that the layer is enabled. ColorFormat: pixel format Stride: line stride Offset: virtual resolution offset Resolution(real/virtual/max): actual, virtual, or maximum resolution MemSize: memory size StartPosition: start position for displaying BufferMode: refresh mode PixelAlpha: pixel alpha value (enable/alpha0/alpha1) GlobalAlpha: global alpha value

Parameter	Description
	Colorkey: colorkey value (enable/colorkey value) Deflicker: anti-flicker (enable/mode/level) 3DMode: 3D mode (source 3D mode/display 3D mode) CanavasAddr: canvas buffer address CanavasUpdateRect: canvas refresh area CanvasResolution: canvas resolution CanvasPitch: canvas stride CanvasFormat: canvas pixel format DisplayResolution: display resolution

[Others]

- Obtain help information.
  echo help > /proc/msp/hifb0
- Display a graphics layer.
  echo show > /proc/msp/hifb0
- Hide a graphics layer.
  echo hide > /proc/msp/hifb0
- Set alpha for a graphics layer.
  echo alpha=255 > /proc/msp/hifb0
- Capture pictures in the display buffer.
  echo capture > /proc/msp/hifb0

2.12 TDE

[Debugging Information]

```
# cat /proc/msp/tde
--------- Hisilicon TDE Node params Info ---------
--- HWNode update: 0x373189 ---
(INS):0xbc4f0049
(S1_XY):0x1e6
(S2_XY):0xf3
(S2_SIZE):0x384004c
(TAR_XY):0xf002a1
(TS_SIZE):0x40f0002f
(2D_RSZ):0x3040011
(HF_COEF_ADDR):0x860ad140
(VF_COEF_ADDR):0x860ad520
(RSZ_Y_OFST):0x0
(RSZ_X_OFST):0x3239
```

[Analysis]

TDE operations are submitted to the hardware by nodes. Each node includes the register configuration information corresponding to the operation. The TDE debugging information records the configuration information about the latest 8 nodes (each HWNode update indicates a node).

To display the TDE debugging information, run the following command:

```
/proc/msp $ echo proc on > tde
```

To hide the TDE debugging information, run the following command:

```
/proc/msp $ echo proc off > tde
```

The Proc information is displayed by default.

[Parameter Description]

Parameter	Description
HWNode update	Object register of the TDE operation. It is represented by two 32-bit numbers. Each bit represents one register. **1** indicates the register is configured and **0** indicates the register is not configured. Bits 0 to 31 represent the following registers respectively: INS, S1_ADDR, S1_TYPE, S1_XY, S1_FILL, S2_ADDR, S2_TYPE, S2_XY, S2_SIZE, S2_FILL, TAR_ADDR, TAR_TYPE, TAR_XY, TS_SIZE, COLOR_CONV, CLUT_ADDR, 2D_RSZ, HF_COEF_ADDR, VF_COEF_ADDR, RSZ_STEP, RSZ_Y_OFST, RSZ_X_OFST, DFE_COEF0, DFE_COEF1, DFE_COEF2, DFE_COEF3, ALU, CK_MIN, CK_MAX, CLIP_START, CLIP_STOP, Y1_ADDR, Y1_PITCH, Y2_ADDR, Y2_PITCH, RSZ_VSTEP, ARGB_ORDER, CK_MASK, COLORIZE, ALPHA_BLEND, ICSC_ADDR, and OCSC_ADDR
INS	Value of the register INS configured by the operation. It is listed only when the register INS needs to be configured. The principle applies to all the following parameters.
S1_ADDR	Value of the register S1_ADDR configured by the operation
S1_TYPE	Value of the register S1_TYPE configured by the operation
S1_XY	Value of the register S1_XY configured by the operation
S1_FILL	Value of the register S1_FILL configured by the operation
S2_ADDR	Value of the register S2_ADDR configured by the operation
S2_TYPE	Value of the register S2_TYPE configured by the operation
S2_XY	Value of the register S2_XY configured by the operation
S2_SIZE	Value of the register S2_SIZE configured by the operation
S2_FILL	Value of the register S2_FILL configured by the operation
TAR_ADDR	Value of the register TAR_ADDR configured by the operation

Parameter	Description
TAR_TYPE	Value of the register TAR_TYPE configured by the operation
TAR_XY	Value of the register TAR_XY configured by the operation
TS_SIZE	Value of the register TS_SIZE configured by the operation
COLOR_CONV	Value of the register COLOR_CONV configured by the operation
CLUT_ADDR	Value of the register CLUT_ADDR configured by the operation
2D_RSZ	Value of the register 2D_RSZ configured by the operation
HF_COEF_ADDR	Value of the register HF_COEF_ADDR configured by the operation
VF_COEF_ADDR	Value of the register VF_COEF_ADDR configured by the operation
RSZ_STEP	Value of the register RSZ_STEP configured by the operation
RSZ_Y_OFST	Value of the register RSZ_Y_OFST configured by the operation
RSZ_X_OFST	Value of the register RSZ_X_OFST configured by the operation
DFE_COEF0	Value of the register DFE_COEF0 configured by the operation
DFE_COEF1	Value of the register DFE_COEF2 configured by the operation
DFE_COEF2	Value of the register S1_FILL configured by the operation
DFE_COEF3	Value of the register DFE_COEF3 configured by the operation
ALU	Value of the register ALU configured by the operation
CK_MIN	Value of the register CK_MIN configured by the operation
CK_MAX	Value of the register CK_MAX configured by the operation
CLIP_START	Value of the register CLIP_START configured by the operation
CLIP_STOP	Value of the register CLIP_STOP configured by the operation
Y1_ADDR	Value of the register Y1_ADDR configured by the operation
Y1_PITCH	Value of the register Y1_PITCH configured by the operation
Y2_ADDR	Value of the register Y2_ADDR configured by the operation
Y2_PITCH	Value of the register Y2_PITCH configured by the operation
RSZ_VSTEP	Value of the register RSZ_VSTEP configured by the operation
ARGB_ORDER	Value of the register ARGB_ORDER configured by the operation
CK_MASK	Value of the register CK_MASK configured by the operation
COLORIZE	Value of the register COLORIZE configured by the operation
ALPHA_BLEND	Value of the register ALPHA_BLEND configured by the operation
ICSC_ADDR	Value of the register ICSC_ADDR configured by the operation

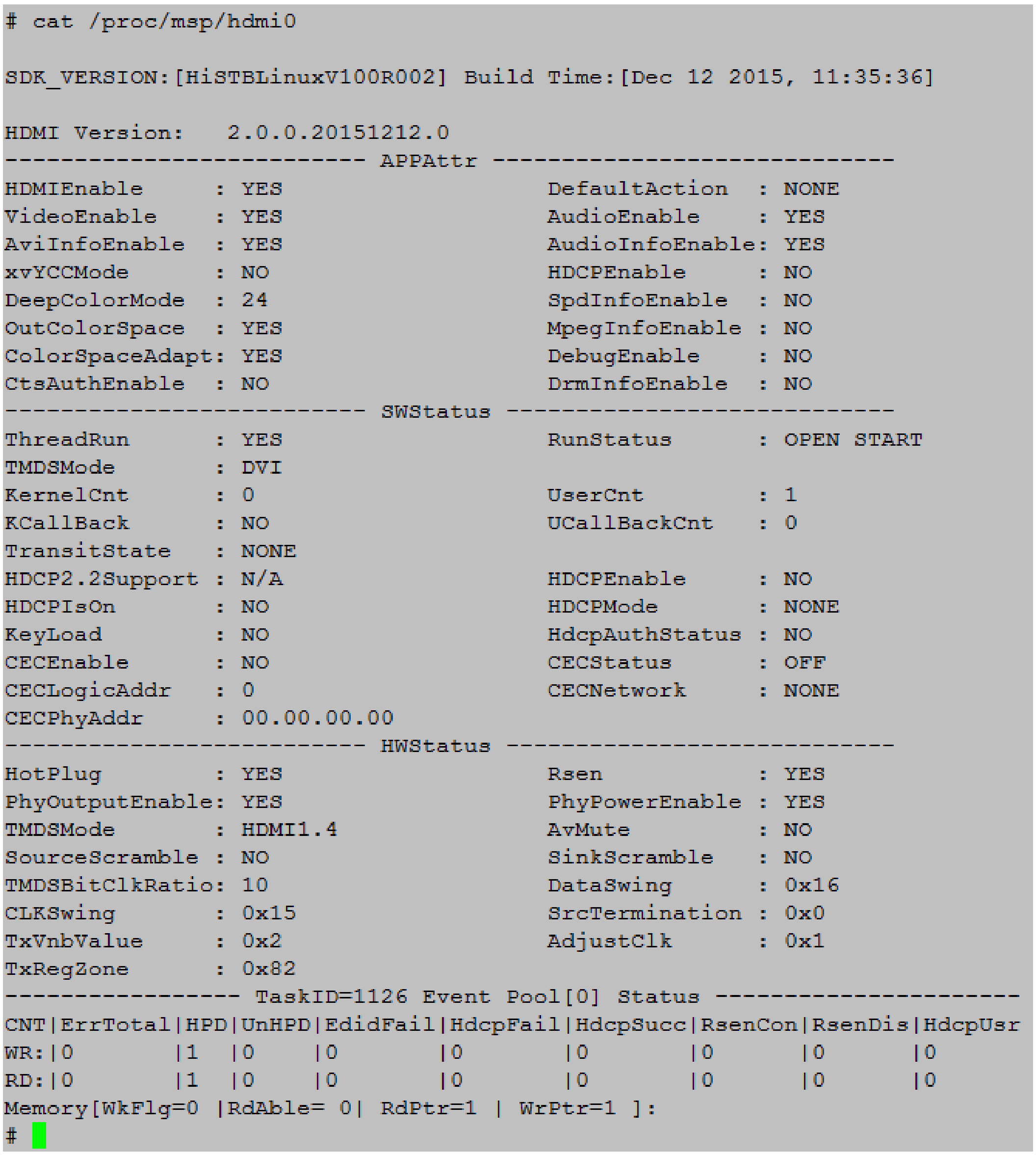
Parameter	Description
OCSC_ADDR	Value of the register OCSC_ADDR configured by the operation

2.13 HDMI

[Debugging Information]

When the HDMI outputs signals properly, information similar to the following is displayed (see Figure 2-1, Figure 2-2, Figure 2-3 and Figure 2-4):

Figure 2-1 Displayed information 1 for HDMI normal outputs

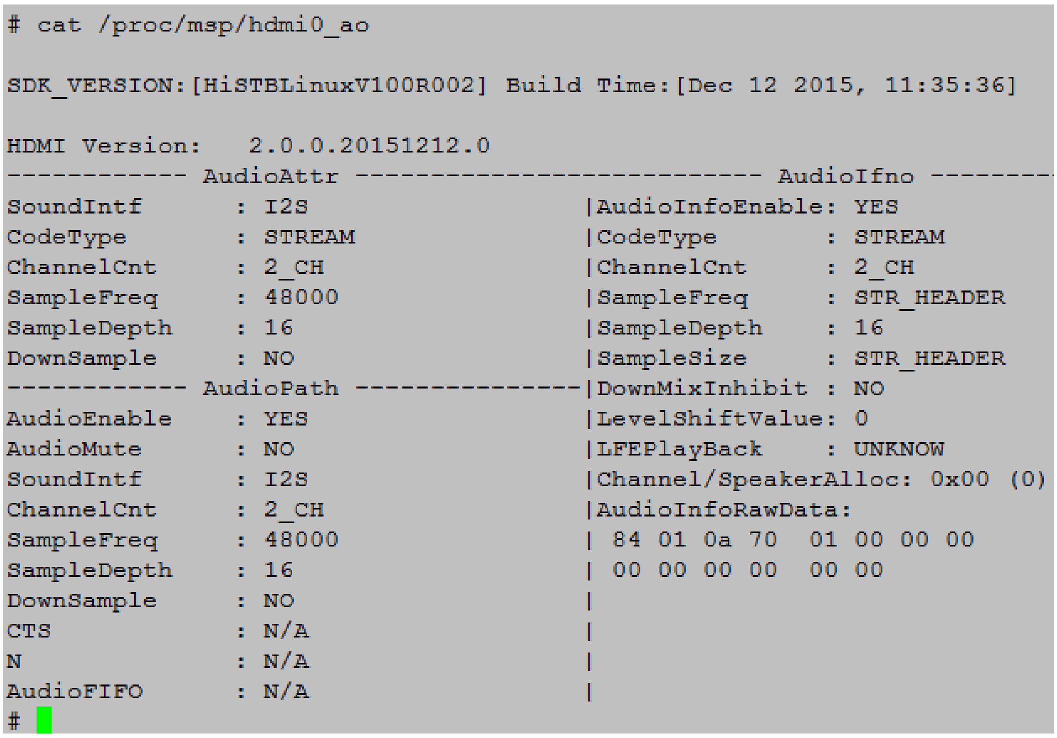


```
# cat /proc/msp/hdmi0

SDK_VERSION:[HiSTBLinuxV100R002] Build Time:[Dec 12 2015, 11:35:36]

HDMI Version:   2.0.0.20151212.0
-------------------------- APPAttr -----------------------------
HDMIEnable     : YES                    DefaultAction  : NONE
VideoEnable    : YES                    AudioEnable    : YES
AviInfoEnable  : YES                    AudioInfoEnable: YES
xvYCCMode      : NO                     HDCPEnable     : NO
DeepColorMode  : 24                     SpdInfoEnable  : NO
OutColorSpace  : YES                    MpegInfoEnable : NO
ColorSpaceAdapt: YES                    DebugEnable    : NO
CtsAuthEnable  : NO                     DrmInfoEnable  : NO
-------------------------- SWStatus ----------------------------
ThreadRun      : YES                    RunStatus      : OPEN START
TMDSMode       : DVI
KernelCnt      : 0                      UserCnt        : 1
KCallBack      : NO                     UCallBackCnt   : 0
TransitState   : NONE
HDCP2.2Support : N/A                    HDCPEnable     : NO
HDCPIsOn       : NO                     HDCPMode       : NONE
KeyLoad        : NO                     HdcpAuthStatus : NO
CECEnable      : NO                     CECStatus      : OFF
CECLogicAddr   : 0                      CECNetwork     : NONE
CECPhyAddr     : 00.00.00.00
-------------------------- HWStatus ----------------------------
HotPlug        : YES                    Rsen           : YES
PhyOutputEnable: YES                    PhyPowerEnable : YES
TMDSMode       : HDMI1.4                AvMute         : NO
SourceScramble : NO                     SinkScramble   : NO
TMDSBitClkRatio: 10                     DataSwing      : 0x16
CLKSwing       : 0x15                   SrcTermination : 0x0
TxVnbValue     : 0x2                    AdjustClk      : 0x1
TxRegZone      : 0x82
----------------- TaskID=1126 Event Pool[0] Status ----------------------
CNT|ErrTotal|HPD|UnHPD|EdidFail|HdcpFail|HdcpSucc|RsenCon|RsenDis|HdcpUsr
WR:|0       |1  |0    |0       |0       |0       |0      |0      |0
RD:|0       |1  |0    |0       |0       |0       |0      |0      |0
Memory[WkFlg=0 |RdAble= 0| RdPtr=1 | WrPtr=1 ]:
# 
```

Figure 2-2 Displayed information 2 for HDMI normal outputs

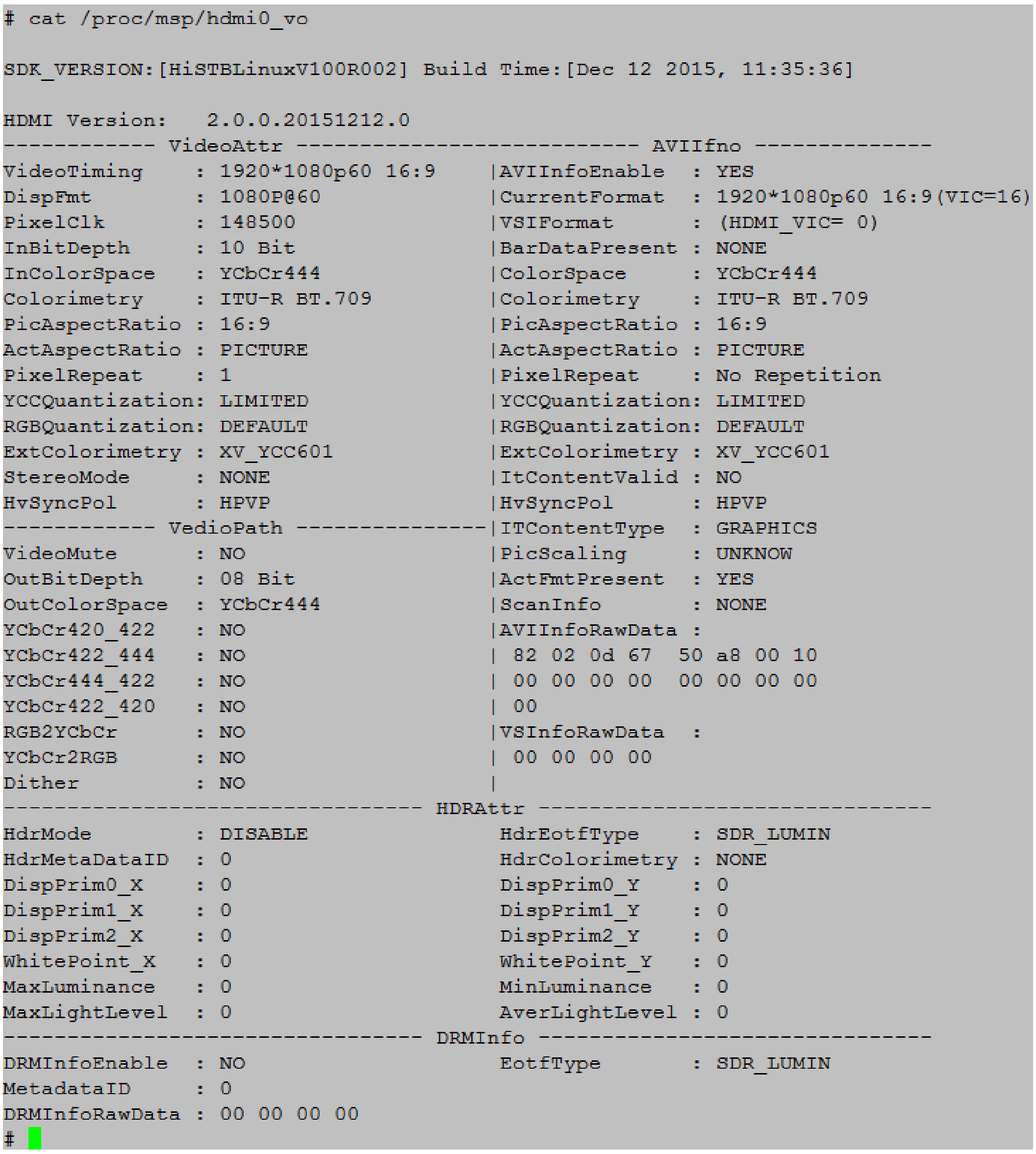


```
# cat /proc/msp/hdmi0_ao

SDK_VERSION:[HiSTBLinuxV100R002] Build Time:[Dec 12 2015, 11:35:36]

HDMI Version:   2.0.0.20151212.0
------------ AudioAttr --------------------------- AudioIfno --------
SoundIntf      : I2S                  |AudioInfoEnable: YES
CodeType       : STREAM               |CodeType       : STREAM
ChannelCnt     : 2_CH                 |ChannelCnt     : 2_CH
SampleFreq     : 48000                |SampleFreq     : STR_HEADER
SampleDepth    : 16                   |SampleDepth    : 16
DownSample     : NO                   |SampleSize     : STR_HEADER
------------ AudioPath ---------------|DownMixInhibit : NO
AudioEnable    : YES                  |LevelShiftValue: 0
AudioMute      : NO                   |LFEPlayBack    : UNKNOW
SoundIntf      : I2S                  |Channel/SpeakerAlloc: 0x00 (0)
ChannelCnt     : 2_CH                 |AudioInfoRawData:
SampleFreq     : 48000                | 84 01 0a 70  01 00 00 00
SampleDepth    : 16                   | 00 00 00 00  00 00
DownSample     : NO                   |
CTS            : N/A                  |
N              : N/A                  |
AudioFIFO      : N/A                  |
# 
```

Figure 2-3 Displayed information 3 for HDMI normal outputs

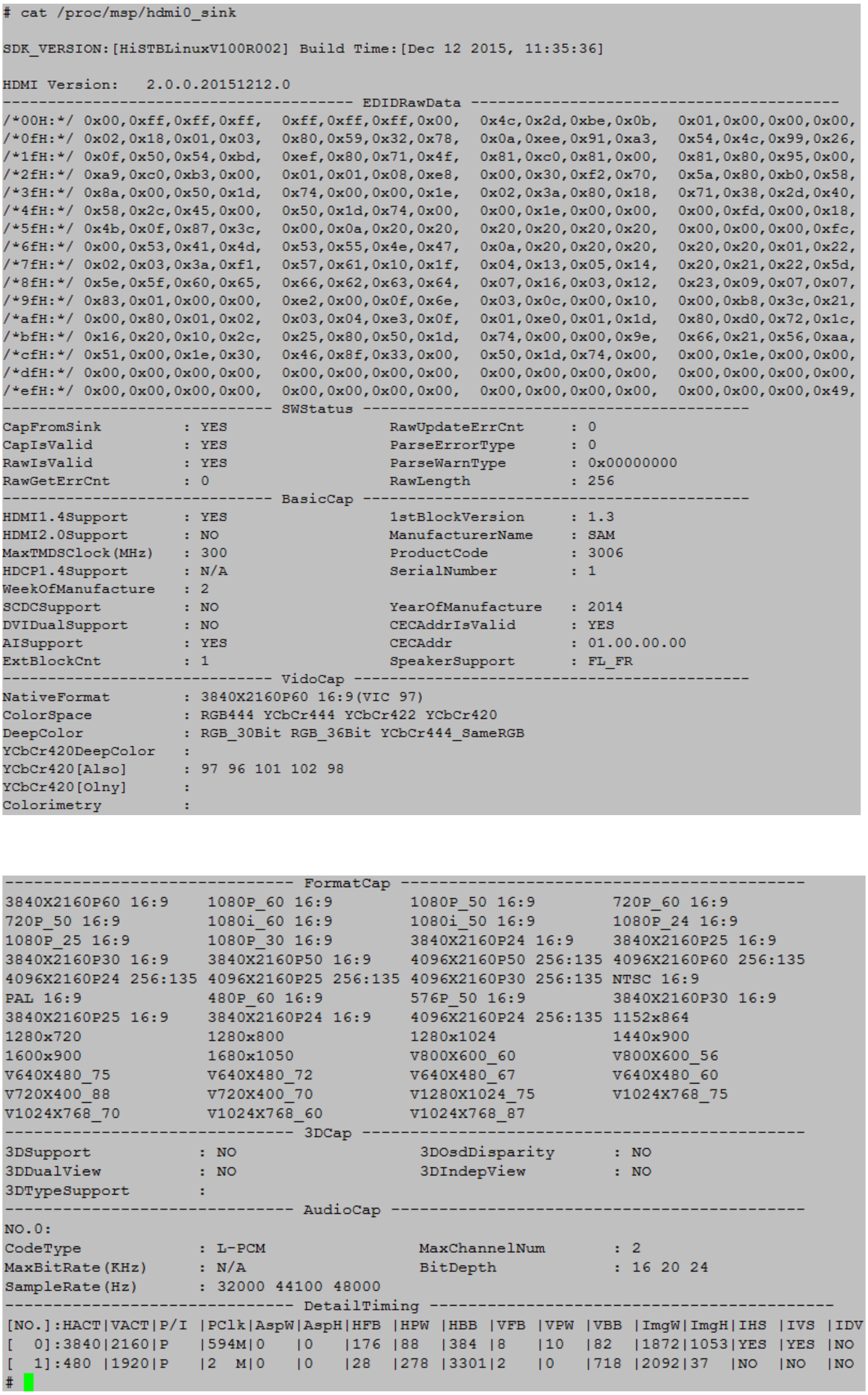


```
# cat /proc/msp/hdmi0_vo

SDK_VERSION:[HiSTBLinuxV100R002] Build Time:[Dec 12 2015, 11:35:36]

HDMI Version:   2.0.0.20151212.0
------------ VideoAttr --------------------------- AVIIfno --------------
VideoTiming    : 1920*1080p60 16:9    |AVIInfoEnable  : YES
DispFmt        : 1080P@60             |CurrentFormat  : 1920*1080p60 16:9(VIC=16)
PixelClk       : 148500               |VSIFormat      : (HDMI_VIC= 0)
InBitDepth     : 10 Bit               |BarDataPresent : NONE
InColorSpace   : YCbCr444             |ColorSpace     : YCbCr444
Colorimetry    : ITU-R BT.709         |Colorimetry    : ITU-R BT.709
PicAspectRatio : 16:9                 |PicAspectRatio : 16:9
ActAspectRatio : PICTURE              |ActAspectRatio : PICTURE
PixelRepeat    : 1                    |PixelRepeat    : No Repetition
YCCQuantization: LIMITED              |YCCQuantization: LIMITED
RGBQuantization: DEFAULT              |RGBQuantization: DEFAULT
ExtColorimetry : XV_YCC601            |ExtColorimetry : XV_YCC601
StereoMode     : NONE                 |ItContentValid : NO
HvSyncPol      : HPVP                 |HvSyncPol      : HPVP
------------ VedioPath ---------------|ITContentType  : GRAPHICS
VideoMute      : NO                   |PicScaling     : UNKNOW
OutBitDepth    : 08 Bit               |ActFmtPresent  : YES
OutColorSpace  : YCbCr444             |ScanInfo       : NONE
YCbCr420_422   : NO                   |AVIInfoRawData :
YCbCr422_444   : NO                   | 82 02 0d 67  50 a8 00 10
YCbCr444_422   : NO                   | 00 00 00 00  00 00 00 00
YCbCr422_420   : NO                   | 00
RGB2YCbCr      : NO                   |VSInfoRawData  :
YCbCr2RGB      : NO                   | 00 00 00 00
Dither         : NO                   |
--------------------------------- HDRAttr -------------------------------
HdrMode        : DISABLE               HdrEotfType    : SDR_LUMIN
HdrMetaDataID  : 0                     HdrColorimetry : NONE
DispPrim0_X    : 0                     DispPrim0_Y    : 0
DispPrim1_X    : 0                     DispPrim1_Y    : 0
DispPrim2_X    : 0                     DispPrim2_Y    : 0
WhitePoint_X   : 0                     WhitePoint_Y   : 0
MaxLuminance   : 0                     MinLuminance   : 0
MaxLightLevel  : 0                     AverLightLevel : 0
--------------------------------- DRMInfo -------------------------------
DRMInfoEnable  : NO                    EotfType       : SDR_LUMIN
MetadataID     : 0
DRMInfoRawData : 00 00 00 00
# 
```

Figure 2-4 Displayed information 4 for HDMI normal outputsHDMI



```
# cat /proc/msp/hdmi0_sink

SDK_VERSION:[HiSTBLinuxV100R002] Build Time:[Dec 12 2015, 11:35:36]

HDMI Version:   2.0.0.20151212.0
--------------------------------------- EDIDRawData -----------------------------------------
/*00H:*/ 0x00,0xff,0xff,0xff,  0xff,0xff,0xff,0x00,  0x4c,0x2d,0xbe,0x0b,  0x01,0x00,0x00,0x00,
/*0fH:*/ 0x02,0x18,0x01,0x03,  0x80,0x59,0x32,0x78,  0x0a,0xee,0x91,0xa3,  0x54,0x4c,0x99,0x26,
/*1fH:*/ 0x0f,0x50,0x54,0xbd,  0xef,0x80,0x71,0x4f,  0x81,0xc0,0x81,0x00,  0x81,0x80,0x95,0x00,
/*2fH:*/ 0xa9,0xc0,0xb3,0x00,  0x01,0x01,0x08,0xe8,  0x00,0x30,0xf2,0x70,  0x5a,0x80,0xb0,0x58,
/*3fH:*/ 0x8a,0x00,0x50,0x1d,  0x74,0x00,0x00,0x1e,  0x02,0x3a,0x80,0x18,  0x71,0x38,0x2d,0x40,
/*4fH:*/ 0x58,0x2c,0x45,0x00,  0x50,0x1d,0x74,0x00,  0x00,0x1e,0x00,0x00,  0x00,0xfd,0x00,0x18,
/*5fH:*/ 0x4b,0x0f,0x87,0x3c,  0x00,0x0a,0x20,0x20,  0x20,0x20,0x20,0x20,  0x00,0x00,0x00,0xfc,
/*6fH:*/ 0x00,0x53,0x41,0x4d,  0x53,0x55,0x4e,0x47,  0x0a,0x20,0x20,0x20,  0x20,0x20,0x01,0x22,
/*7fH:*/ 0x02,0x03,0x3a,0xf1,  0x57,0x61,0x10,0x1f,  0x04,0x13,0x05,0x14,  0x20,0x21,0x22,0x5d,
/*8fH:*/ 0x5e,0x5f,0x60,0x65,  0x66,0x62,0x63,0x64,  0x07,0x16,0x03,0x12,  0x23,0x09,0x07,0x07,
/*9fH:*/ 0x83,0x01,0x00,0x00,  0xe2,0x00,0x0f,0x6e,  0x03,0x0c,0x00,0x10,  0x00,0xb8,0x3c,0x21,
/*afH:*/ 0x00,0x80,0x01,0x02,  0x03,0x04,0xe3,0x0f,  0x01,0xe0,0x01,0x1d,  0x80,0xd0,0x72,0x1c,
/*bfH:*/ 0x16,0x20,0x10,0x2c,  0x25,0x80,0x50,0x1d,  0x74,0x00,0x00,0x9e,  0x66,0x21,0x56,0xaa,
/*cfH:*/ 0x51,0x00,0x1e,0x30,  0x46,0x8f,0x33,0x00,  0x50,0x1d,0x74,0x00,  0x00,0x1e,0x00,0x00,
/*dfH:*/ 0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,
/*efH:*/ 0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x49,
------------------------------ SWStatus -------------------------------------------
CapFromSink          : YES                    RawUpdateErrCnt      : 0
CapIsValid           : YES                    ParseErrorType       : 0
RawIsValid           : YES                    ParseWarnType        : 0x00000000
RawGetErrCnt         : 0                      RawLength            : 256
------------------------------ BasicCap -------------------------------------------
HDMI1.4Support       : YES                    1stBlockVersion      : 1.3
HDMI2.0Support       : NO                     ManufacturerName     : SAM
MaxTMDSClock(MHz)    : 300                    ProductCode          : 3006
HDCP1.4Support       : N/A                    SerialNumber         : 1
WeekOfManufacture    : 2
SCDCSupport          : NO                     YearOfManufacture    : 2014
DVIDualSupport       : NO                     CECAddrIsValid       : YES
AISupport            : YES                    CECAddr              : 01.00.00.00
ExtBlockCnt          : 1                      SpeakerSupport       : FL_FR
------------------------------ VidoCap --------------------------------------------
NativeFormat         : 3840X2160P60 16:9(VIC 97)
ColorSpace           : RGB444 YCbCr444 YCbCr422 YCbCr420
DeepColor            : RGB_30Bit RGB_36Bit YCbCr444_SameRGB
YCbCr420DeepColor    :
YCbCr420[Also]       : 97 96 101 102 98
YCbCr420[Olny]       :
Colorimetry          :
```

```
------------------------------ FormatCap ------------------------------------------
3840X2160P60 16:9    1080P_60 16:9        1080P_50 16:9        720P_60 16:9
720P_50 16:9         1080i_60 16:9        1080i_50 16:9        1080P_24 16:9
1080P_25 16:9        1080P_30 16:9        3840X2160P24 16:9    3840X2160P25 16:9
3840X2160P30 16:9    3840X2160P50 16:9    4096X2160P50 256:135 4096X2160P60 256:135
4096X2160P24 256:135 4096X2160P25 256:135 4096X2160P30 256:135 NTSC 16:9
PAL 16:9             480P_60 16:9         576P_50 16:9         3840X2160P30 16:9
3840X2160P25 16:9    3840X2160P24 16:9    4096X2160P24 256:135 1152x864
1280x720             1280x800             1280x1024            1440x900
1600x900             1680x1050            V800X600_60          V800X600_56
V640X480_75          V640X480_72          V640X480_67          V640X480_60
V720X400_88          V720X400_70          V1280X1024_75        V1024X768_75
V1024X768_70         V1024X768_60         V1024X768_87
------------------------------ 3DCap ----------------------------------------------
3DSupport            : NO                     3DOsdDisparity       : NO
3DDualView           : NO                     3DIndepView          : NO
3DTypeSupport        :
------------------------------ AudioCap -------------------------------------------
NO.0:
CodeType             : L-PCM                  MaxChannelNum        : 2
MaxBitRate(KHz)      : N/A                    BitDepth             : 16 20 24
SampleRate(Hz)       : 32000 44100 48000
------------------------------ DetailTiming -----------------------------------------
[NO.]:HACT|VACT|P/I |PClk|AspW|AspH|HFB |HPW |HBB |VFB |VPW |VBB |ImgW|ImgH|IHS |IVS |IDV
[  0]:3840|2160|P   |594M|0   |0   |176 |88  |384 |8   |10  |82  |1872|1053|YES |YES |NO
[  1]:480 |1920|P   |2  M|0   |0   |28  |278 |3301|2   |0   |718 |2092|37  |NO  |NO  |NO
#
```

[Analysis]

Information about the HDMI output management module is recorded.

[Parameter Description]

Parameter		Description
APPAttr	HDMIEnable	HDMI enable Value: {YES, NO}
	DefaultAction	Default working mode Value: {NONE, HDMI, DVI, UNKNOW}
	VideoEnable	Whether to enable video output for the current user Value: {YES, NO}
	AudioEnable	Whether to enable audio for the current user Value: {YES, NO}
	AviInfoEnable	Whether to enable the AVI information frame for the current user Value: {YES, NO}
	AudioInfoEnable	Whether to enable the audio information frame for the current user Value: {YES, NO}
	xvYCCMode	Whether to enable the xvYCC output for the current user Value: {YES, NO}
	HDCPEnable	Whether to enable HDCP for the current user Value: {YES, NO}
	DeepColorMode	Picture color depth configured by the current user Value: {24, 30, 36, 48, OFF, UNKNOW}

Parameter		Description
	SpdInfoEnable	Whether to enable the SPD information frame for the current user Value: {YES, NO}
	OutColorSpace	Color space output configured by the current user Value: {RGB, YCbCr422, YCbCr444, YCbCr420, BUTT}
	MpegInfoEnable	Whether to enable output of the MPEG information frame for the current user Value: {YES, NO}
	DebugEnable	Whether to enable the debugging mode for the current user Value: {YES, NO}
	CtsAuthEnable	Whether to enable the certification mode for the current user Value: {YES, NO}
	DrmInfoEnable	Whether to enable the DRM information frame for the current user Value: {YES, NO}
SWStatus	ThreadRun	Whether to run the current HDMI thread Value: {YES, NO}
	RunStatus	HDMI running status Value: {NONE, OPEN, START, STOP, CLOSE}
	TMDSMode	Current TMDS operating mode Value: {NONE, DVI, HDMI1.4,

Parameter		Description
		HDMI2.0, AUTO, UNKNOW}
	KernelCnt	Count of HDMI devices opened by the kernel
	UserCnt	Count of HDMI devices opened by the user
	KCallBack	Whether HDMI callback is registered Value: {YES, NO}
	UCallBackCnt	Times of user callback
	TransitState	Recording of the process status Value: {BOOT->MCE, MCE->APP, BOOT->APP}
	HDCP2.2Support	Whether the current sink supports HDCP 2.2 Value: {YES, NO}
	HDCPEnable	Whether to enable the current HDCP Value: {YES, NO}
	HDCPIsOn	Whether to open the current HDCP Value: {YES, NO}
	HDCPMode	Current HDCP mode Value: {NONE, HDCP1.4, HDCP2.2, AUTO, UNKNOW}
	KeyLoad	Whether the key is loaded Value: {YES, NO}
	HdcpAuthStatus	HDCP certification status Value: {YES, NO}

Parameter		Description
		Note: **YES** indicates that the certification is successful, and **NO** indicates that the certification fails.
	CECEnable	Whether to enable the current CEC Value: {YES, NO}
	CECStatus	CEC status Value: {YES, NO}
	CECLogicAddr	Logical address of the current CEC device
	CECNetwork	Current CEC network Value: {TV, RECORDDEV_1, RECORDDEV_2, TUNER_1, PLAYDEV_1, AUDIOSYSTEM, TUNER_2, TUNER_3, PLAYDEV_2, RECORDDEV_3, TUNER_4, PLAYDEV_3, RESERVED_1, RESERVED_2, SPECIALUSE, BROADCAST}
	CECPhyAddr	Physical address of the current CEC device
HWStatus	HotPlug	Whether current hardware detects hot plug Value: {YES, NO}
	Rsen	Whether current hardware detects Rsen Value: {YES, NO}
	PhyOutputEnable	Whether to enable current PHY output

Parameter		Description
		Value: {YES, NO}
	PhyPowerEnable	Whether to enable the current PHY power Value: {YES, NO}
	SourceScramble	Whether scramble has been configured successfully for the current source end Value: {YES, NO}
	SinkScramble	Whether scramble has been configured successfully for the current sink end Value: {YES, NO}
	TMDSBitClkRatio	Ratio of the current TMDS clock frequency to the TMDS bit clock frequency Value: {10, 40}
	DataSwing	PHY indicator, which is used to adjust the data swing and affects the size of the eye pattern
	CLKSwing	PHY indicator, which is used to adjust the clock swing and affects the size of the eye pattern
	SrcTermination	PHY indicator. The data/clock swing of each channel is adjusted separately.
	TxVnbValue	PHY indicator. The clock swing and data swing are both adjusted. The central point of the single-ended waveform is affected. The wave crest remains unchanged, and the

Parameter		Description
		central point fluctuates.
	AdjustClk	PHY indicator. The swing is fine-tuned and the clock swing is affected.
	TxRegZone	PHY indicator, adaptive to the HDMI-TX PHY PLL
TaskID Event Pool Status	ErrTotal	Total times of failure events
	HPD	Times of plug-in for the hot-plug event
	UnHPD	Times of plug-out for the hot-plug event
	EdidFail	Times of EDID reading failures
	HdcpFail	Times of HDCP certification failures
	HdcpSucc	Times of HDCP certification success
	RsenCon	Times of Rsen connection
	RsenDis	Times of Rsen disconnection
	HdcpUsr	Times that the user sets the HDCP
	WkFlg	Wakeup flag Value: {HI_TRUE, HI_FALSE}
	RdAble	Number of readable events in the event pool
	RdPtr	Event read pointer in the event pool
	WrPtr	Event write pointer in the event pool
AudioAttr	SoundIntf	Type of the HDMI audio interface

Parameter		Description
		Value: {I2S, SPDIF, HBRA, UNKNOW}
	CodeType	Audio encoding type Value: {STREAM, L-PCM, AC3, MPEG1, MP3, MPEG2, AAC_LC, DTS, ATRAC, OneBitAudio, EAC3, DTS-HD, MAT, DST, WMA_PRO, Reserved, UNKNOW} Note: For details, see the *EIA-CEA-861-D (F)*.
	ChannelCnt	Number of audio channels Value: {STR_HEADER, 2_CH, 3_CH, 4_CH, 5_CH, 6_CH, 7_CH, 8_CH} Note: For details, see the *EIA-CEA-861-D (F)*.
	SampleFreq	Audio sampling frequency Value: {STR_HEADER, 32 kHz, 44.1 kHz, 48 kHz , 88.2 kHz , 96 kHz, 176.4 kHz, 192 kHz} Note: The Hi3536 does not support sampling rates lower than 32 kHz.
	SampleDepth	Sampling depth (bit width) Value: {STR_HEADER, 16 bits, 20 bits, 24 bits}
	DownSample	Whether there is

Parameter		Description
		down sampling Value: {YES, NO}
AudioIfno	AudioInfoEnable	Whether to enable the audio information frame Value: {YES, NO}
	CodeType	Audio encoding type Value: {STREAM, L-PCM, AC3, MPEG1, MP3, MPEG2, AAC_LC, DTS, ATRAC, OneBitAudio, EAC3, DTS-HD, MAT, DST, WMA_PRO, Reserved, UNKNOW} Note: For details, see the *EIA-CEA-861-D (F)*.
	ChannelCnt	Number of audio channels Value: {STR_HEADER, 2_CH, 3_CH, 4_CH, 5_CH, 6_CH, 7_CH, 8_CH} Note: For details, see the *EIA-CEA-861-D (F)*.
	SampleFreq	Audio sampling frequency Value: {STR_HEADER, 32 kHz, 44.1 kHz, 48 kHz , 88.2 kHz , 96 kHz, 176.4 kHz, 192 kHz} Note: The Hi3536 does not support sampling rates lower than 32 kHz.
	SampleDepth	Sampling depth Value: {UNKNOW,

Parameter		Description
		8 bits, 16 bits, 18 bits, 20 bits, 24 bits, 32 bits, BUTT}
	SampleSize	Sampling size Value: {STR_HEADER, 16 bits, 20 bits, 24 bits}
	DownMixInhibit	Down mixing inhibit flag Value: {YES, NO} Note: For details, see the *EIA-CEA-861-D (F)*.
	LevelShiftValue	Level shift value Value: {0−15 dB} Note: For details, see the *EIA-CEA-861-D (F)*.
	LFEPlayBack	LFE playback level information Value: {UNKNOW, 0 dB, +10 dB, Reserved} Note: For details, see the *EIA-CEA-861-D (F)*.
	Channel/SpeakerAlloc	Channel/Speaker allocation Value: {0x00−0xff} Note: This parameter is expressed in hexadecimal or decimal. For details, see the *EIA-CEA-861-D (F)*.
	AudioInfoRawData	AudioInfoframe raw data
AudioPath	AudioEnable	Whether to enable the audio Value: {YES, NO}

Parameter		Description
	AudioMute	Whether to mute the audio Value: {YES, NO}
	SoundIntf	Audio interface type Value: {I2S, SPDIF, HBRA, UNKNOW} Note: The Hi3536 supports only the I^{2}S audio interface currently.
	ChannelCnt	Number of current audio output channels Value: {STR_HEADER, 2_CH, 3_CH, 4_CH, 5_CH, 6_CH, 7_CH, 8_CH} Note: For details, see the *EIA-CEA-861-D (F)*.
	SampleFreq	Current audio sampling frequency Value: {STR_HEADER, 32 kHz, 44.1 kHz, 48 kHz , 88.2 kHz , 96 kHz, 176.4 kHz, 192 kHz} Note: The Hi3536 does not support sampling rates lower than 32 kHz.
	SampleDepth	Current audio sampling depth Value: {UNKNOW, 8 bits, 16 bits, 18 bits, 20 bits, 24 bits, 32 bits, BUTT}
	DownSample	Whether to down sample the current audio output Value: {YES, NO}
	CTS	CTS value

Parameter		Description
		Note: For details, see the *EIA-CEA-861-D (F)*.
	N	N value Note: For details, see the *EIA-CEA-861-D (F)*.
	AudioFIFO	Audio FIFO
VideoAttr	VideoTiming	Current video timing Note: For details, see the *EIA-CEA-861-D (F)* and *VESA Display Monitor Timing Standard*.
	DispFmt	Current video format Note: For details, see the *EIA-CEA-861-D (F)* and *VESA Display Monitor Timing Standard*.
	PixelClk	Pixel clock Note: For details, see the *EIA-CEA-861-D (F)* and *VESA Display Monitor Timing Standard*.
	InBitDepth	Color depth of the input video Value: {8 bits, 10 bits, 12 bits, 16 bits, UNKNOW} Note: UNKNOW indicates unknown.
	InColorSpace	Color space of the input video Value: {RGB, YCbCr422, YCbCr444, YCbCr420, UNKNOW}

Parameter		Description
		Note: UNKNOW indicates unknown.
	Colorimetry	Color gamut of the input video Value: {No Data, SMPTE 170M, ITU-R BT.709, Extended}
	PicAspectRatio	Aspect ratio of the input video Value: {NONE, 4:3, 16:9, FUTURE, UNKNOW} Note: UNKNOW indicates unknown.
	ActAspectRatio	Aspect ratio of the valid picture in the input video Value: {Reserved, 16:9_TOP, 14:9_TOP, 16:9_CENTER, PICTURE, 4:3, 16:9, 14:9, 4:3_SP_14_9, 16:9_SP_14_9, 16:9_SP_4_3, UNKNOW} Note: UNKNOW indicates unknown.
	PixelRepeat	Times of pixel repetition
	YCCQuantization	YCC quantization range Value: {LIMITED, FULL, UNKNOW} Note: UNKNOW indicates unknown.
	RGBQuantization	RGB quantization range Value: {DEFAULT, LIMITED, FULL, UNKNOW} Note: UNKNOW indicates unknown.

Parameter		Description
	ExtColorimetry	Extended color gamut for the input picture Value: {XV_YCC601, XV_YCC709, S_YCC601, ADOBE_YCC601, ADOBE_RGB, BT2020_cYCC, BT2020_RGB/YCC, UNKNOW}
	StereoMode	Stereo mode Value: {FRAME_PACK, FIELD_ALTER, LINE_ALTERN, SBS_FULL, L_DEPTH, L_DEPTH_GGD, TAndB, Reserved, SByS_HALF, NONE} Note: NONE refers to unknown.
	HvSyncPol	Horizontal/Vertical sync polarity Value: {HPVP, HPVN, HNVP, HNVN, UNKNOW} Note: H indicates horizontal; V indicates vertical; P indicates positive; N indicates negative; UNKNOW indicates unknown.
AVIIfno	AVIInfoEnable	Whether to enable an AVI information frame. Value: {YES, NO}
	CurrentFormat	Current video format/VIC Note: For details, see the *EIA-CEA-861-D (F)*.

Parameter		Description
	VSIFormat	4K non-3D format/VIC Note: For details, see the *HDMI Specification 2.0*.
	BarDataPresent	Bar information Value: {NONE, HnVp, HpVn, HpVp} Note: H indicates horizontal; V indicates vertical; P indicates positive; N indicates negative. For details, see the *EIA-CEA-861-D (F)*.
	ColorSpace	Video color space Value: {RGB, YCbCr422, YCbCr444, YCbCr420, UNKNOW} Note: UNKNOW indicates unknown.
	Colorimetry	Color gamut Value: {No Data, SMPTE 170M, ITU-R BT.709, Extended}
	PicAspectRatio	Video aspect ratio Value: {NONE, 4:3, 16:9, FUTURE, UNKNOW} Note: UNKNOW indicates unknown.
	ActAspectRatio	Aspect ratio of the valid picture of the video Value: {Reserved, 16:9_TOP, 14:9_TOP, 16:9_CENTER, PICTURE, 4:3, 16:9, 14:9, 4:3_SP_14_9,

Parameter		Description
		16:9_SP_14_9, 16:9_SP_4_3, UNKNOW} Note: UNKNOW indicates unknown.
	PixelRepeat	Times of pixel repetition
	YCCQuantization	YCC quantization range Value: {LIMITED, FULL, UNKNOW} Note: UNKNOW indicates unknown.
	RGBQuantization	RGB quantization range Value: {DEFAULT, LIMITED, FULL, UNKNOW} Note: UNKNOW indicates unknown.
	ExtColorimetry	Picture extended color gamut Value: {XV_YCC601, XV_YCC709, S_YCC601, ADOBE_YCC601, ADOBE_RGB, BT2020_YCC, BT2020_RGB/YCC, UNKNOW} Note: UNKNOW indicates unknown.
	ItContentValid	Whether the IT content is valid Value: {YES, NO} Note: For details, see the *EIA-CEA-861-D (F)*.
	HvSyncPol	Horizontal/Vertical sync polarity Value: {HPVP, HPVN, HNVP, HNVN, UNKNOW}

Parameter		Description
		Note: H indicates horizontal; V indicates vertical; P indicates positive; N indicates negative. Note: UNKNOW indicates unknown.
	ITContentType	IT content type Value: {GRAPHICS, PHOTO, CINEMA, GAME}
	PicScaling	Picture equalization Value: {UNKNOW, HpVn, HnVp, HpVp} Note: H indicates horizontal; V indicates vertical; P indicates positive; N indicates negative. For details, see the *EIA-CEA-861-D (F)*.
	ActFmtPresent	Whether the valid information is displayed Value: {YES, NO} Note: For details, see the *EIA-CEA-861-D (F)*.
	ScanInfo	Scan flag bit Value: {NONE, OVER_SCAN, UNDERS_SCAN, Reserved}
	AVIInfoRawData	AVI information frame raw data
	VSInfoRawData	VSIF information frame raw data
VedioPath	VideoMute	Video mute flag Value: {YES, NO}
	OutBitDepth	Color depth of the

Parameter		Description
		output video Value: {8 bits, 10 bits, 12 bits, 16 bits, UNKNOW}
	OutColorSpace	Color space of the output picture Value: {RGB, YCbCr422, YCbCr444, YCbCr420, UNKNOW}
	YCbCr420_422	Whether the color space is converted from YCbCr420 to YCbCr422 Value: {YES, NO}
	YCbCr422_444	Whether the color space is converted from YCbCr422 to YCbCr444 Value: {YES, NO}
	YCbCr444_422	Whether the color space is converted from YCbCr444 to YCbCr422 Value: {YES, NO}
	YCbCr422_420	Whether the color space is converted from YCbCr422 to YCbCr420 Value: {YES, NO}
	RGB2YCbCr	Whether the color space is converted from RGB to YCbCr Value: {YES, NO}
	YCbCr2RGB	Whether the color space is converted from YCbCr to RGB
	Dither	Whether to enable the dither Value: {YES, NO}
HDRAttr	HdrMode	Current HDR mode Value: {DISABLE, DOLBY_NORMAL

Parameter		Description
		, DOLBY_TUNNELING, HDR, UNKNOW}
	HdrEotfType	EOTF type in HDR mode Value: {SDR_LUMIN, HDR_LUMIN, SMPTE_ST_2048, FUTURE, UNKNOW}
	HdrMetaDataID	MetaData type in HDR mode Value: 0: Static Metadata Type 1 Other types are not supported currently.
	HdrColorimetry	Colorimetry in HDR mode Value: {NONE, ITU_601, ITU_709, EXTENDED, XV_YCC_601, XV_YCC_709, S_YCC_601, ADOBE_YCC_601, ADOBE_RGB, 2020_nconst_luminous, 2020_const_luminous}
	DispPrim0_X	Value of display_primaries_x[0] See the *CEA-861.3*.
	DispPrim0_Y	Value of display_primaries_y[0] See the *CEA-861.3*.
	DispPrim1_X	Value of display_primaries_x[1] See the *CEA-861.3*.

Parameter		Description
	DispPrim1_Y	Value of display_primaries_y[1] See the *CEA-861.3*.
	DispPrim2_X	Value of display_primaries_x[2] See the *CEA-861.3*.
	DispPrim2_Y	Value of display_primaries_y[2] See the *CEA-861.3*.
	WhitePoint_X	Value of white_point_x See the *CEA-861.3*.
	WhitePoint_Y	Value of white_point_y See the *CEA-861.3*.
	MaxLuminance	Value of max_display_mastering_luminance See the *CEA-861.3*.
	MinLuminance	Value of min_display_mastering_luminance See the *CEA-861.3*.
	MaxLightLevel	Maximum content light level See the *CEA-861.3*.
	AverLightLevel	Maximum frame-average light level See the *CEA-861.3*.
DRMInfo	DRMInfoEnable	Whether to enable the current DRM information frame Value: {YES, NO}
	EotfType	Current EOTF type Value: {SDR_LUMIN, HDR_LUMIN, SMPTE_ST_2048,

Parameter		Description
		FUTURE, UNKNOW}
	MetadataID	Current MetaData type
	DRMInfoRawData	DRM information frame raw data
EDIDRawData		Raw EDID data, 256 bytes
SWStatus	CapFromSink	Whether the capability set is from the monitor (sink) Value: {YES, NO} Note: The capability set may be from data used for tests. In this case, the parameter value is **NO**.
	RawUpdateErrCnt	Number of raw data update errors
	CapIsValid	Whether the capability set is valid Value: {YES, NO}
	ParseErrorType	EDID parsing error flag 0: No error occurs. 1: A check error occurs. 2: A data header error occurs. 3: The basic data block is not the 1.3 version. 4: The tag value of the extended data block is unknown. 5: The CEA value is invalid. The value 6 or values greater than 6 are reserved.
	RawIsValid	Whether the raw data is valid

Parameter		Description
		Value: {YES, NO}
	ParseWarnType	EDID parsing alarm flag Value: {0x00000000−0xffffffff} Each bit corresponds to an alarm. Bit 0: No alarm is raised. Bit 1: The vendor block is invalid. Bit 2: The number of DTD blocks exceeds the threshold. Bit 3: The DTD block is invalid. Bit 4: There is no extended block. Bit 5: The number of extended blocks is greater than 4. Bit 6: The number of audio blocks exceeds the threshold. Bit 7: The number of VICs exceeds the threshold. Bit 8: The VIC is invalid. Bit 9: The VSDB is invalid. Bit 10: The HFVSDB is invalid. Bit 11: The SPEAKER DB is invalid. Bit 12: The number of YCBCR420 VICs exceeds the threshold. Bit 13: The data block length is invalid. Bits 14−31: reserved

Parameter		Description
	RawGetErrCnt	Number of times that the raw data fails to be obtained
	RawLength	Raw data length
BasicCap	HDMI1.4Support	Whether the sink end supports HDMI 1.4 Value: {YES, NO}
	1stBlockVersion	Version number of the first EDID block
	HDMI2.0Support	Whether HDMI 2.0 is supported Value: {YES, NO}
	ManufacturerName	Sink vendor name
	MaxTMDSClock(MHz)	Maximum TMDS clock Note: For details, see the *EIA-CEA-861-D (F)*.
	ProductCode	Sink-end product number
	HDCP1.4Support	Whether the sink end supports only HDCP 1.4 Value: {YES, NO}
	SerialNumber	Sink product serial number
	HDCP2.2Support	Whether the sink end supports HDCP 2.2 Value: {YES, NO}
	WeekOfManufacture	Week of manufacture for the sink-end product Note: For details, see the *EIA-CEA-861-D (F)*.
	SCDCSupport	Whether the sink end supports SCDC Value: {YES, NO}

Parameter		Description
	YearOfManufacture	Year of manufacture for the sink-end product
	DVIDualSupport	Whether the sink end supports DVI
	CECAddrIsValid	Whether the CEC address at the sind end is valid Value: {YES, NO}
	AISupport	Whether the sink end supports AI Value: {YES, NO}
	CECAddr	CEC address
	ExtBlockCnt	Number of EDID extended blocks
	SpeakerSupport	Supported speaker at the sink end Value: {FL_FR, LFE, FC, RL_RR, RC, FLC_FRC, RLC_RRC, FLW_FRW, FLH_FRH, TC, FCH, UNKNOW} Note: For details, see the *EIA-CEA-861-D (F)*.
VidoCap	NativeFormat	Standard that requires no conversion (optimal standard) Note: For details, see the *EIA-CEA-861-D (F)*.
	ColorSpace	Supported color space Value: {RGB444, YCbCr444, YCbCr422, YCbCr420} Note: For details, see the *EIA-CEA-861-D*

Parameter		Description
		(F).
	DeepColor	Supported number of deep color bits Value: {RGB_30Bit, RGB_36Bit, RGB_48Bit, YCbCr444_SameRGB} Note: For details, see the *EIA-CEA-861-D (F)*.
	YCbCr420DeepColor	Supported number of deep color bits for YCbCr420 Value: {30 bits, 36 bits, 48 bits} Note: For details, see the *EIA-CEA-861-D (F)*.
	YCbCr420[Also]	The VIC codes of RGB, YCbCr444, YCbCr422, and YCbCr420 are supported. Note: For details, see the *EIA-CEA-861-D (F)*.
	YCbCr420[Olny]	Only the VIC code of YCbCr420 is supported Note: For details, see the *EIA-CEA-861-D (F)*.
	Colorimetry	Colorimetry color gamut Value: {xvYCC601, xvYCC709, sYCC601 ,AdobleYCC601, AdobleRGB, BT2020cYCC, BT2020YCC,

Parameter		Description
		BT2020RGB} Note: For details, see the *EIA-CEA-861-D (F)*.
FormatCap		Standards supported by the sink end Note: For details, see the *EIA-CEA-861-D (F)* and *VESA Display Monitor Timing Standard*.
3DCap	3DSupport	Whether the sink end supports 3D Value: {YES, NO}
	3DDualView	Whether the sink end supports 3D dual view Value: {YES, NO}
	3DTypeSupport	3D display type supported by the sink end Value: {FRAME_PACK, FIELD_ALTER, LINE_ALTERN, SBS_FULL, L_DEPTH, L_DEPTH_GGD, TAndB, Reserved, SByS_HALF, NONE}
	3DOsdDisparity	Whether the sink end supports 3D OSD disparity Value: {YES, NO}
	3DIndepView	Whether the sink end supports 3D independent view Value: {YES, NO}
AudioCap	NO.x:	No. x
	CodeType	Encoding type Value: {STREAM, L-PCM, AC3,

Parameter		Description
		MPEG1, MP3, MPEG2, AAC_LC, DTS, ATRAC, OneBitAudio, EAC3, DTS-HD, MAT, DST, WMA_PRO, Reserved, UNKNOW} Note: For details, see the *EIA-CEA-861-D (F)*.
	MaxChannelNum	Maximum number of channels
	MaxBitRate(KHz)	Maximum bit rate
	BitDepth	Bit depth Value: {0, 8, 16, 18, 20, 24, 32, BUTT}
	SampleRate(Hz)	Sampling rate Value: {0, 8 kHz, 11025 Hz, 12000 Hz, 16000 Hz, 22050 Hz, 24000 Hz, 32000 Hz, 44100 Hz, 48000 Hz, 88200 Hz, 96000 Hz, 176400 Hz, 192000 Hz, 768000 Hz, BUTT}
DetailTiming	NO.	ID
	HACT	Valid horizontal pixel
	VACT	Valid vertical pixel
	P/I	Progressive/Interlaced
	PClk	Pixel clock
	AspW	Width of the aspect ratio
	AspH	Height of the aspect ratio
	HFB	Horizontal front blank

Parameter		Description
	HPW	Horizontal pulse width
	HBB	Horizontal back blank
	VFB	Vertical front blank
	VPW	Vertical pulse width
	VBB	Vertical back blank
	ImgW	Picture width
	ImgH	Picture height
	IHS	Whether the horizontal sync pulse is inverted
	IVS	Whether the vertical sync pulse is inverted
	IDV	Whether the valid signal is inverted

2.14 TUNER

2.14.1 Proc

[Debugging Information]

```
# cat /proc/msp/tuner
---------Hisilicon TUNER Info---------
Port: LockStat    I2CChannel    Frequency(KHz)    SymbRate    QamMode
0:     locked  4    610000    6875000     QAM_64
DemodType      Demodaddr      TunerType      Tuneraddr
3130_inside    0xa0           CD1616        0xc0
BER:0*(E-7),  SNR:43,  SignalStrength:1102
all_rs_package: 6587,  corrected_rs_package: 6587,  error_rs_package:0
```

[Analysis]

Information about all tuners is recorded, including whether the turners are locked and the locked frequency. The port ID is the turner ID.

[Parameter Description]

Parameter	Description
Port	Tuner ID
LockStat	Lock flag. The value is **Locked** or **Unlocked**.
I2CChannel	ID of the Inter-integrated circuit (I^2C) channel used by the tuner
Frequency(kHz)	Frequency, in kHz
SymbRate	Symbol rate, in symbol/s
QamMode	Modulation mode QAM_16/QAM_32/QAM_64/QAM_128/QAM_256/BPSK/QPSK/8PSK/16APSK/32APSK/QAM_INVALID
DemodType	Demod type For example, 3130_inside/3130_outside/hi3136i/hi3136e/avl6211/mxl101
Demodaddr	I^2C address for the Demod
TunerType	Tuner type
Tuneraddr	I^2C address for the tuner
BER	Bit error rate *X**(E–7) indicates *X**10 to the power of –7 (*X* is a variable).
SNR	Signal-to-noise ratio (SNR)
SignalStrength	Signal strength. The signal strength in the Proc information may not be the actual signal strength. It may be the automatic gain control (AGC) gain value. The actual signal strength is obtained by calculation, which varies according to components.
all_rs_package	Number of output TSs generated within the preset time
corrected_rs_package	Number of TSs corrected by the RS within the preset time
error_rs_package	Number of TSs that cannot be corrected by the RS within the preset time. Pay attention to this parameter. When the value of this parameter is not 0, error packages exist. Check the operating environment and signal channels.

The video and audio clips can be played and data can be received in DVB mode only when the tuner is locked. If the audio and video cannot be played or the Demux has no data, check whether the tuner is locked.

2.14.2 Tuner_Reg

[Debugging Information]

```
cat /proc/msp/tuner_reg
---------Hisilicon TUNER Info---------
{AGC_CTRL_1_ADDR  ,0xf0 },
{AGC_CTRL_4_ADDR  ,0x02 },
{      BAGC_CTRL_3_ADDR  ,0x00 },
{     BAGC_CTRL_17_ADDR  ,0x34 },
```

[Analysis]

Information about all quadrature amplitude modulation (QAM) registers is recorded (only supported by the Hi3130 series chips or embedded QAM). Identify the corresponding register descriptions in the *QAM User Guide* based on the register name. For example, remove the register name extension _ADDR from the register name AGC_CTRL_1_ADDR contained in the debugging information and convert all upper-case letters into lower-case letters, that is, agc_ctrl_1. Search for agc_ctrl_1 in the *QAM User Guide* and view the corresponding register descriptions.

2.15 IR

[Debugging Information]

```
$ cat /proc/msp/ir
---------Hisilicon IR Info---------
IR Enable     :Enable
IR Code :IR_S2
IR WorkMode    :Chip Report Symbols
IR FetchMode :Drive Report Parsed Symbols
IR KeyUpEnable      :Enable
IR UpEventDelay:300(ms)
IR RepeatkeyEnable     :Enable
IR RepkeyDelayTime    :200(ms)
IR ReportKeyBlockTime :200(ms)
IR ModuleFrequence :24(MHz)
Registered Protocols info:
No.0: Status:    Enabled,   Name: nec simple 2headers gd
No.1: Status:    Enabled,   Name: extended rc5 14bit
No.2: Status:    Enabled,   Name: rc5 14bit data
No.3: Status:    Enabled,   Name: nec simple 2headers changshu
No.4: Status:    Enabled,   Name: nec full 2headers
No.5: Status:   Disabled,   Name: tc9012
```

```
Key getting info:
Get(Try/OK)     :20/2
Key buffer info:
Buffer size:   100 keys
Reader at:     2
Writer at:     2
Symbol buffer info:
Buffer size:   100 symbols
Reader at:     36
Writer at:     36
```

[Analysis]

Information about the remote control is recorded. Some attributes become valid or invalid when the value of **IR Code** changes. Therefore, the debugging information may be different from that displayed in the preceding example.

[Parameter Description]

Parameter	Description
Enable	Enable flag Enable: enabled Disable: disabled
Code	IR driver code flag R_STD: standard IR (obtained by logical parsing and supporting one protocol at a time) IR_S2: common IR IR_LIRC: universal IR
WorkMode	IR working mode Chip Report Symbols: The driver parses key values. Chip Report Keys: The chip parses key values.
FetchMode	Type of obtained remote control code Drive Report Raw Symbols: raw level Drive Report Parsed Symbols: parsed level, that is, key value
KeyUpEnable	Whether to report the key-up status Enable: yes Disable: no
UpEventDelay	Key-up report delay, in ms
RepeatkeyEnable	Whether to report the repeated use of the same key

Parameter	Description
	Enable: report Disable: not report
RepkeyDelayTime	Threshold for determining whether a key is repeatedly pressed (in ms)
ReportKeyBlockTime	Maximum waiting time for obtaining a key value, in ms
ModuleFrequence	IR module reference clock divider (MHz)
Registered Protocols info	Supported protocols and their enable/disable status
Key getting info	Try: number of attempts to obtain key values OK: number of successes to obtain key values
Key buffer info	Size of the key buffer area and positions of the read and write pointers
Symbol buffer info	Size of the level buffer area and positions of the read and write pointers

When the remote control cannot receive key values, perform the following operations:

1. Verify that the IR module is enabled and the attribute of **IR Enable** is **1**.
2. Check whether data is available in the symbol buffer. If yes, go to step 5. If no, go to step 3.
3. Verify that the IR interrupt mask register is properly set.
4. Verify that the IR receiver is properly connected.
5. Verify that the remote control is compatible with the protocols and the protocols are enabled in **Registered Protocols info**. (When the IR_STD driver is enabled, check whether the HI_UNF_IR_SetCodeType function is called to set the remote control type.)
6. Verify that the value of **ModuleFrequence** is consistent with the reference clock of the current IR module.

----End

2.16 I2C

[Debugging Information]

```
$  cat /proc/msp/i2c
---------Hisilicon Standard I2C Info---------
No.   Rate
0    400000
1    400000
```

```
2    400000
3    400000
4    400000
5    400000
---------Hisilicon GPIO simulate I2C Info---------
No.   SCL_IO     SDA_IO
610 11
```

[Analysis]

The standard I^{2}C communication rate and information about the General Purpose Input/Output (GPIO) port for simulating I^{2}C are checked.

[Parameter Description]

Parameter	Description
No.	ID of the standard I2C or GPIO-simulated I2C
Rate	Communication rate of the standard I2C (the GPIO-simulated I2Cdoes not support rate configuration)
SCL_IO	GPIO port for simulating the I2C_SCL
SDA_IO	GPIO port for simulating the I2C SDA

[Debugging Commands]

Read or write data by using an I^{2}C channel.

$ echo id address offset [value] >/proc/msp/i2c

For example:

$echo 4 a0 5d 80 >/proc/msp/i2c

The preceding information indicates that I^{2}C channel 4 is used to write data 0x80 to the I^{2}C device address a0 with the offset address 5d.

$ echo 4 a0 5d >/proc/msp/i2c

0x80

The preceding information indicates that I^{2}C channel 4 is used to read data from the I^{2}C device address a0 with the device offset address 5d.

[Command Description]

This proc command can be used to debug only the slave component with the 8-bit register address, and the length of read/written data is 1 byte. If a slave component whose register address is more than eight bits needs to be debugged, or the length of read/written data is greater than 1 byte, use the i2c_read and i2c_write functions in the I^{2}C sample.

[Parameter Description]

Parameter	Description
id	I^2C channel used by the device#
address	Address of a device on the I^2C bus
offset	On-chip offset address of a device
value	Value to be written. This parameter is optional. If you specify the parameter, a write operation is performed; if you do not specify the parameter, a read operation is performed.

The Proc system automatically determines the operation type (read or write) based on the command length. If the **value** parameter is not specified, the read operation is performed; otherwise, the write operation is performed. Note that if the **value** parameter is specified, enter a space after the value.

- If the used I^2C bus connection fails, "wait write data timeout!" is displayed. Check whether the I^2C channel and the address are correct, and whether the connection of the I^2C hardware is normal.
- If a GPIO I^2C bus is used, and the GPIO I^2C channel that does not apply for resources is used to perform an operation (for example, echo 8 a0 5d >/proc/msp/i2c), "gpio_i2c_read failed(I2cNum=8 not valid)!" is displayed.
- If the value of offset is 0 to f (in hexadecimal), add 0 before the value, that is, 00 to 0f.

  The values of **Address** and **offset** do not start with **0x**.

Modify the standard I^2C communication rate.

$ echo SetRate id rate > /proc/msp/i2c

For example:

$ echo SetRate 1 100000 > /proc/msp/i2c

2.17 SCI

[Debugging Information]

```
# cat /proc/msp/sci0
---------Hisilicon SCI0 Info---------
Sci State:RX
SetFrequency:3570
Protocol:T=0
VccEnLevel:1
DetectLevel:1
ClockMode:OD
ResetMode:CMOS
VccMode:OD
ATR :0x3b 0x9f 0x11 0x40 0x60 0x49 0x52 0x44 0x45 0x54 0x4f 0x20 0x41 0x43
0x53 0x20 0x56 0x35 0x2e 0x33
```

```
TS:0x3b
T0:0x9f
TA1 :0x11 (FI = 372,DI = 1)
TD1 :0x40
TC2 :0x60 (only for T0 show IC card max char timeout)
HistoryByte :0x49 0x52 0x44 0x45 0x54 0x4f 0x20 0x41 0x43 0x53 0x20 0x56 0x35
0x2e 0x33
ActualSciClk :3600
ExpectBaudRate:9596
CalcBaudFlag :0
bSetExtBaudFlag:0
ClkRate(F):372
BitRate(D):1
BaudRate:9677
AddCharGuard:0 etu
BlockGuard:22 etu
Value:5
Baud:2231
CharTimeout :92160
```

[Analysis]

Information about the smart card interface (SCI) running status is recorded.

[Parameter Description]

Parameter	Description
Sci State	Current status of the SCI card, including NOCARD, INACTIVECARD, and RX
SetFrequency	Card frequency (in kHz). For the T0 and T1 cards, the frequency ranges from 1 MHz to 5 MHz; for the T14 card, the frequency must be 6 MHz.
Protocol	When the protocol is T=0, T=1, or T=14, the corresponding card type is T0, T1, or T14.
VccEnLevel	Power supply voltage enable signal 0: active low 1: active high
DetectLevel	Detect card level 0: active low 1: active high
ClockMode	Clock pin output mode 0: CMOS 1: OD
ResetMode	Output mode of the reset pin

Parameter	Description
	0: CMOS 1: OD
VccMode	Output mode of the Vcc pin 0: CMOS 1: OD
ATR	Reset response. It consists of a series of bytes, which define the features of the communication between the card and the terminal.
TS	The start byte is the first byte received by the terminal, which is used to define the logic direction of the subsequent bytes. 0x3B: direct convention 0x3F: inverse convention
T0	T0 characters Upper half bytes (b5–b8): Whether the subsequent characters TA1 to TD1 exist Lower half byte (b4–b1): The number of selectable historical characters ranges from 0 to 15.
TA1	TA1 is used to transmit the values of FI and DI. FI (b5–b8) is used to define the value of F (clock rate conversion factor). DI (b4–b1) is used to define the value of D (bit rate regulator factor).
TD1	The upper half bytes (b5–b8) indicate whether the subsequent bytes TA2 to TD2 exist. When the value is **1**, the corresponding bytes (TA2 to TD2) exist. The lower half bytes (b4–b1) indicate the protocol type used for subsequent information exchange. **0** indicates the T0 protocol, **1** indicates the T1 protocol, and **e** indicates the T14 protocol.
TC2	WI value. It is used to determine the maximum interval between the rising edge of the start bit of any character sent by the IC card and the rising edge of the start bit of the previous character sent by the IC card or terminal.
HistoryByte	Historical bytes
ActualSciClk	Clock that the chip outputs to the card after the SCI clock configures the register by calculation. A deviation is allowed between **ActualSciClk** and the configured clock, but the deviation must fall within the clock range supported by the card.
ExpectBaudRate	Expected baud rate ExpectBaudRate = SetFrequency x BitRate x 1000/ClkRate
CalcBaudFlag	If the ATR receives TA2 and bit 5 is 0, the specified mode is used and this flag bit is **1**. The ETU is set based on the received TA1 parameter. If TA2 is not received, the interactive mode is used by default and this flag bit is **0**.

Parameter	Description
bSetExtBaudFlag	When it is **1**, applications set the F and D factor flag bits of the ETU externally by calling HI_UNF_SCI_SetEtuFactor(). When it is **0**, those flag bits are not externally configured.
ClkRate(F)	Configured clock rate factor of the ETU
BitRate(D)	Configured rate adjustment factor of the ETU
BaudRate	Calculated baud rate BaudRate = ActualSciClk x BitRate x 1000/ClkRate
AddCharGuard	Extra character guard time (in ETU)
BlockGuard	T1 block guard time (in ETU)
Value	Cycle for configuring the ETU baud rate
Baud	Clock frequency divider for configuring the ETU baud rate
CharTimeout	Character timeout period (in ETU)

2.18 LOG

[Debugging Information]

```
# cat /proc/msp/log
---------------- Log Path -----------------------
log path:
---------------- Store Path ---------------------
store path:/mnt
---------------- Module Log Level ----------------
Log moduleLevel
--------------------------
HI_SYS1(ERROR)
HI_MODULE 1(ERROR)
HI_LOG1(ERROR)
HI_PROC1(ERROR)
HI_MEM1(ERROR)
HI_STAT1(ERROR)
HI_PDM1(ERROR)
HI_MEMDEV 1(ERROR)
HI_DEMUX1(ERROR)
HI_ADEC1(ERROR)
HI_AO1(ERROR)
HI_AI1(ERROR)
HI_ADSP1(ERROR)
```

```
HI_VFMW1(ERROR)
HI_SVDEC1(ERROR)
HI_DISP1(ERROR)
HI_HDMI1(ERROR)
HI_VO1(ERROR)
HI_VPSS1(ERROR)
HI_VDEC1(ERROR)
HI_VI1(ERROR)
HI_VENC1(ERROR)
HI_TDE1(ERROR)
jpeg1(ERROR)
HI_JPGE1(ERROR)
HI_FB1(ERROR)
HI_PNG1(ERROR)
HI_PVR1(ERROR)
HI_AVPLAY 1(ERROR)
HI_SYNC1(ERROR)
VSYNC1(ERROR)
ASYNC1(ERROR)
HI_MCE1(ERROR)
HI_IR1(ERROR)
HI_I2C1(ERROR)
HI_SCI1(ERROR)
HI_WDG1(ERROR)
HI_GPIO1(ERROR)
HI_GPIO_I2C1(ERROR)
hi_tuner1(ERROR)
HI_KEYLED 1(ERROR)
HI_CIPHER 1(ERROR)
HI_OTP1(ERROR)
HI_PM1(ERROR)
HI_SIL92931(ERROR)
HI_NDPT1(ERROR)
echo hi_avplay=2 > /proc/msp/log
echo log=/mnt > /proc/msp/log
echo storepath=/mnt > /proc/msp/log
# echohelp>/proc/msp/log
To modify the level, use command line in shell:
echo module_name = level_number > /proc/msp/log
level_number: 0-fatal, 1-error, 2-warning, 3-info
example: 'echo HI_DEMUX=3 > /proc/msp/log'
will change log level of module "HI_DEMUX" to 3, then,
all messages with level higher than "info" will be printed.
Use 'echo "all = x" > /proc/msp/log' to change all modules.
```

```
To modify the log path, use command line in shell:
Use 'echo "log = x" > /proc/msp/log' to set log path.
Use 'echo "log = /dev/null" > /proc/msp/log' to close log udisk output.
example: 'echo log=/home > /proc/msp/log'
To modify the debug file store path, use command line in shell:
Use 'echo "storepath = x" > /proc/msp/log' to set debug file path.
example: 'echo storepath=/tmp > /proc/msp/log'
```

[Analysis]

The debugging level of each module is recorded.

[Parameter Description]

Parameter	Description
Log Module	Module name
Level	Level of the displayed log for each module 0: fatal 1: error 2: warning 3: info 4: debug

- Information of the error level for all modules is output by default.
- To modify the print level of a specific module, run the following command:

  #echo *module name*=*print level* > /proc/msp/log

The module name is not case sensitive.

- You can dynamically set a storage path for the debugging file (for example, Demux recording file) by running the following command:

  echo storepath=*storage path* > /proc/msp/log
- You can dynamically modify the log output position to the USB flash drive or serial port by running **echo log = *output position* > /proc/msp/log**. For example, you can run**echo log = /home > /proc/msp/log** to output the log file to a specific directory. You do not need to specify the file name because the file name is fixed at stb.log.
- To disable file output of logs, run **echo log = /dev/null > /proc/msp/log**. Then logs are output over the serial port.

  To modify the output position of logs, choose **UDisk Log Support** from the compilation options and access the corresponding menu by running the **menuconfig** command.

```
make menuconfig
Common--->
[*] Log Support
[*]UDisk Log Support
```

- To set the compilation level of logs, change the value of Compile Log Level and access the corresponding menu by running the menuconfig command.

```
make menuconfig
Common--->
[*] Log Support
(1)Compile Log Level (0,4)
```

Re-compile the SDK after the modification.

- To support network logs, choose Network Log Support from the compilation options and access the corresponding menu by running the menuconfig command.

```
make menuconfig
Common --->
[*] Log Support
[*]Network Log Support
```

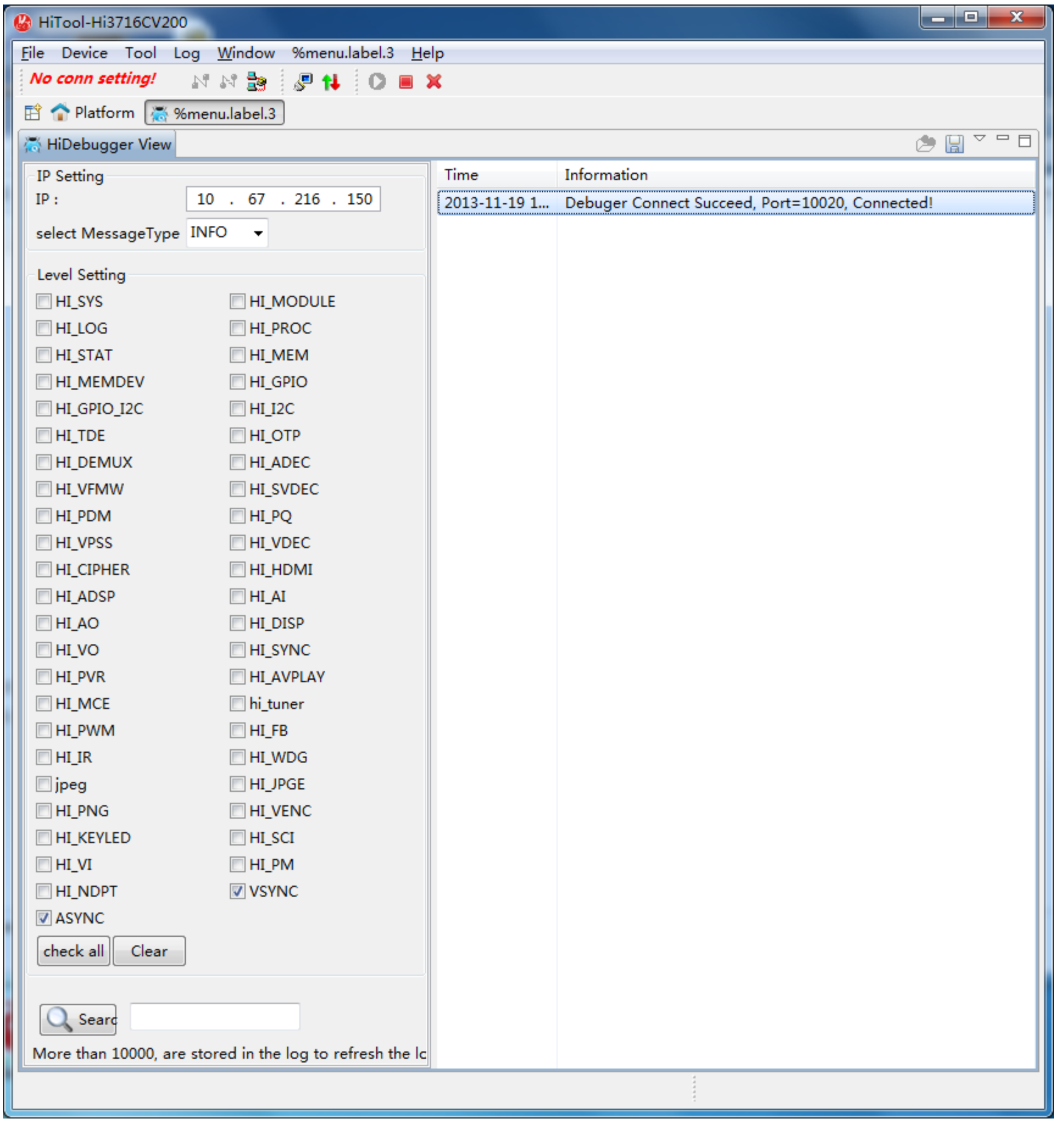
Then select **Msp Debug Tool Support** and access the corresponding menu by running the **menuconfig** command.

```
make menuconfig
Rootfs--->
Board Tools Config--->
[*] Msp Debug Tool Support
```

After re-compilation of the SDK, msp_debug is available in the file system. Run msp_debug on the board (at the background), and then run the required sample.

Run the HiTool on the PC, and choose the HiDebugger View, as shown in Figure 2-5.

Figure 2-5 Running HiDebuger



IP Setting
IP : 10 . 67 . 216 . 150
select MessageType INFO

Level Setting

☐ HI_SYS ☐ HI_MODULE
☐ HI_LOG ☐ HI_PROC
☐ HI_STAT ☐ HI_MEM
☐ HI_MEMDEV ☐ HI_GPIO
☐ HI_GPIO_I2C ☐ HI_I2C
☐ HI_TDE ☐ HI_OTP
☐ HI_DEMUX ☐ HI_ADEC
☐ HI_VFMW ☐ HI_SVDEC
☐ HI_PDM ☐ HI_PQ
☐ HI_VPSS ☐ HI_VDEC
☐ HI_CIPHER ☐ HI_HDMI
☐ HI_ADSP ☐ HI_AI
☐ HI_AO ☐ HI_DISP
☐ HI_VO ☐ HI_SYNC
☐ HI_PVR ☐ HI_AVPLAY
☐ HI_MCE ☐ hi_tuner
☐ HI_PWM ☐ HI_FB
☐ HI_IR ☐ HI_WDG
☐ jpeg ☐ HI_JPGE
☐ HI_PNG ☐ HI_VENC
☐ HI_KEYLED ☐ HI_SCI
☐ HI_VI ☐ HI_PM
☐ HI_NDPT ☑ VSYNC
☑ ASYNC

check all Clear

More than 10000, are stored in the log to refresh the lc

Time	Information
2013-11-19 1...	Debuger Connect Succeed, Port=10020, Connected!

Set the IP address, click the green button in the tool bar to connect to the board, and select the debugging level and required modules to start obtaining the debugging information.

2.19 Interrupts

[Debugging Information]

```
$ cat /proc/interrupts
   CPU0       CPU1
 57: 57  0       GIC  timer
 58:    1591315  0       GIC  timer4
 59:  0 51281991       GIC  timer6
 66:  0  0       GIC  hi_mci
 67:  0  0      GIC  hi_mci
 79:  0  0      GIC  hi_ir
 81:      1157  0      GIC  uart-pl011
 98:  0  0      GIC  ehci_hcd:usb1
 99:  1  0      GIC  ohci_hcd:usb2
102:  0  0      GIC  ahci
103:  0  0      GIC  gmac0
104:  0  0      GIC  gmac1
107:  0  0      GIC  CI_IRQ
114:  0  0      GIC  Demux
122: 25506901  0      GIC  DISP_IRQ
123:  0  0      GIC  tde_osr_isr
125:  0  0      GIC  VPSS_IRQ
126:  0  0      GIC  mali_mmu_irq_handlers, mali_gp_irq_handlers,
mali_mmu_irq_handlers, mali_pp_irq_handlers, mali_mmu_irq_handlers,
mali_pp_irq_handlers
128:  0  0      GIC  png_isr
129:  0  0    GIC  x5_jpeg
134:  0  0      GIC  jpge
140:  0  0      GIC  gpio00
141:  0  0      GIC  gpio01
142:  0  0      GIC  gpio02
143:  0  0      GIC  gpio03
144:  0  0      GIC  gpio04
145:  0  0      GIC  gpio05
146:  0  0      GIC  gpio06
147:  0  0      GIC  gpio07
148:  0  0      GIC  gpio08
149:  0  0      GIC  gpio09
150:  0  0      GIC  gpio10
151:  0  0      GIC  gpio11
152:  0  0      GIC  gpio12
153:  0  0      GIC  gpio13
154:  0  0      GIC  gpio14
155:  0  0      GIC  gpio15
156:  0  0      GIC  gpio16
157:  0  0      GIC  gpio17
IPI0:  0  0  Timer broadcast interrupts
IPI1:931     21205  Rescheduling interrupts
IPI2:  0  0  Function call interrupts
IPI3: 11 28  Single function call interrupts
IPI4:  0  0  CPU stop interrupts
IPI5:  0  0  CPU backtrace
Err:  0
```

You can check whether the module is running properly by viewing the interrupt response counts.

For example, for the Demux module, you can check whether the Demux receives TS stream through the transport stream interface (TSI).

2.20 Media-mem

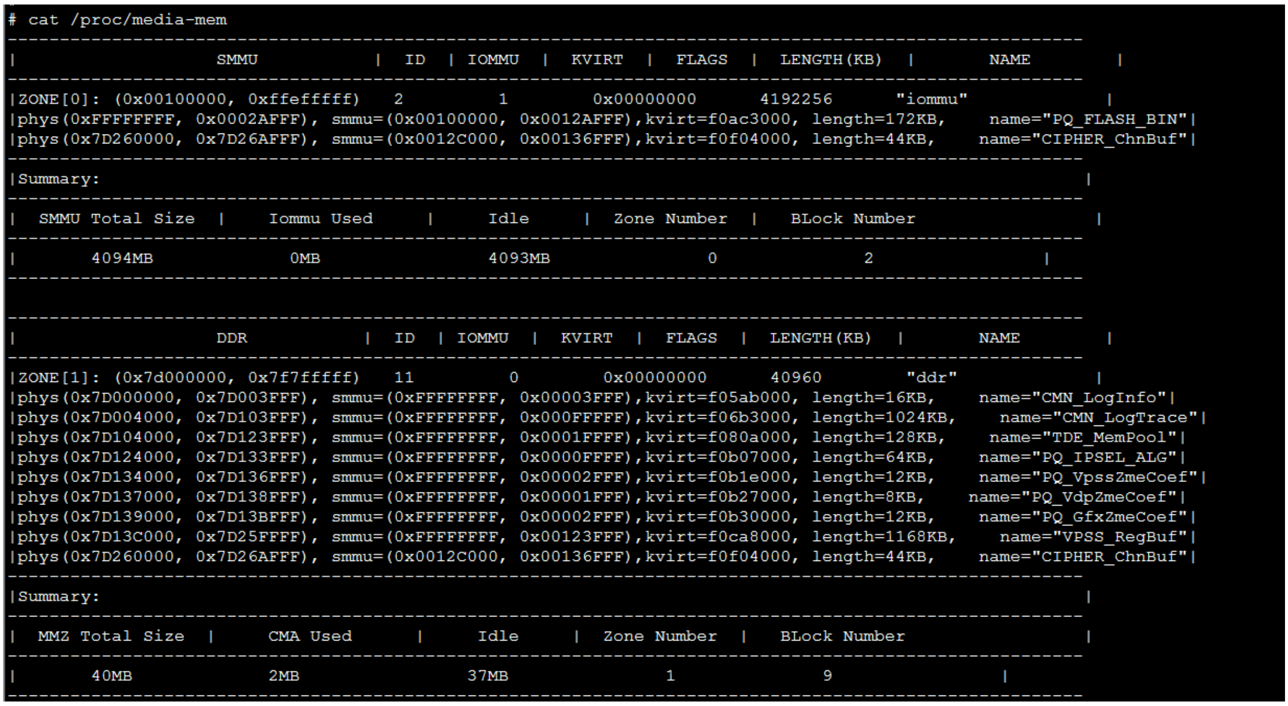
[Debugging Information]

```
# cat /proc/media-mem
```

NOTE

The SMMU information shown in Figure 2-6 applies only to Hi3798C V200.

Figure 2-6 PMOC debugging information



```
# cat /proc/media-mem
-----------------------------------------------------------------------------------------------------------
|                    SMMU             |  ID   | IOMMU  |   KVIRT   |  FLAGS   |  LENGTH(KB)  |       NAME        |
-----------------------------------------------------------------------------------------------------------
|ZONE[0]: (0x00100000, 0xffefffff)   2          1         0x00000000      4192256      "iommu"               |
|phys(0xFFFFFFFF, 0x0002AFFF), smmu=(0x00100000, 0x0012AFFF),kvirt=f0ac3000, length=172KB,    name="PQ_FLASH_BIN"|
|phys(0x7D260000, 0x7D26AFFF), smmu=(0x0012C000, 0x00136FFF),kvirt=f0f04000, length=44KB,    name="CIPHER_ChnBuf"|
-----------------------------------------------------------------------------------------------------------
|Summary:                                                                                                  |
-----------------------------------------------------------------------------------------------------------
|  SMMU Total Size  |     Iommu Used     |      Idle      |  Zone Number  |   BLock Number                  |
-----------------------------------------------------------------------------------------------------------
|       4094MB             0MB                4093MB              0              2                |
-----------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------
|                    DDR             |  ID  | IOMMU  |   KVIRT   |  FLAGS   |  LENGTH(KB)  |       NAME        |
-----------------------------------------------------------------------------------------------------------
|ZONE[1]: (0x7d000000, 0x7f7fffff)   11          0          0x00000000      40960        "ddr"               |
|phys(0x7D000000, 0x7D003FFF), smmu=(0xFFFFFFFF, 0x00003FFF),kvirt=f05ab000, length=16KB,    name="CMN_LogInfo"|
|phys(0x7D004000, 0x7D103FFF), smmu=(0xFFFFFFFF, 0x000FFFFF),kvirt=f06b3000, length=1024KB,    name="CMN_LogTrace"|
|phys(0x7D104000, 0x7D123FFF), smmu=(0xFFFFFFFF, 0x0001FFFF),kvirt=f080a000, length=128KB,    name="TDE_MemPool"|
|phys(0x7D124000, 0x7D133FFF), smmu=(0xFFFFFFFF, 0x0000FFFF),kvirt=f0b07000, length=64KB,    name="PQ_IPSEL_ALG"|
|phys(0x7D134000, 0x7D136FFF), smmu=(0xFFFFFFFF, 0x00002FFF),kvirt=f0b1e000, length=12KB,    name="PQ_VpssZmeCoef"|
|phys(0x7D137000, 0x7D138FFF), smmu=(0xFFFFFFFF, 0x00001FFF),kvirt=f0b27000, length=8KB,    name="PQ_VdpZmeCoef"|
|phys(0x7D139000, 0x7D13BFFF), smmu=(0xFFFFFFFF, 0x00002FFF),kvirt=f0b30000, length=12KB,    name="PQ_GfxZmeCoef"|
|phys(0x7D13C000, 0x7D25FFFF), smmu=(0xFFFFFFFF, 0x00123FFF),kvirt=f0ca8000, length=1168KB,    name="VPSS_RegBuf"|
|phys(0x7D260000, 0x7D26AFFF), smmu=(0x0012C000, 0x00136FFF),kvirt=f0f04000, length=44KB,    name="CIPHER_ChnBuf"|
-----------------------------------------------------------------------------------------------------------
|Summary:                                                                                                  |
-----------------------------------------------------------------------------------------------------------
|  MMZ Total Size  |      CMA Used      |      Idle      |  Zone Number  |   BLock Number                  |
-----------------------------------------------------------------------------------------------------------
|       40MB             2MB                37MB              1              9                |
-----------------------------------------------------------------------------------------------------------
```

[Analysis]

Media-mem describes the utilization of the MMZ/SMMU memory in services.

[Parameter Description]

Parameter	Description
SMMU	The current memory zone is the SMMU zone and the physical addresses are not necessarily continuous.
ID	ID of the memory block in the current memory zone

Parameter	Description
IOMMU	Type of the current memory zone. 1 indicates the SMMU zone and 0 indicates the DDR zone.
KVIRT	Start virtual address of a memory block
FLAGS	Attribute flag of a memory block, reserved
LENGTH(KB)	Length of a memory block, with a unit of KB
NAME	Name of a zone
ZONE[0]	SMMU start address and end address of the SMMU zone
phys	Physical address space of the memory block. When the start address is all Fs, the address space is invalid and meaningless.
smmu	SMMU virtual address space of the memory block. When the start address is all Fs, the address space is invalid and meaningless.
kvirt	Start address of the kernel-mode virtual address of the memory block
length	Length of the memory block
name	Name of the memory block
SMMU Total Size	Total size of the SMMU zone
Iommu Used	Memory used by the SMMU zone, in MB
Idle	Total size of the idle memory
Zone Number	Current zone No.
Block Number	Number of memory blocks allocated in the current zone
ZONE[1]	Physical start address and end address of the DDR zone
DDR	The current memory zone is the DDR zone and the physical memory is continuous.
MMZ Total Size	Total size of memory blocks in MMZ
CMA Used	Used memory size in MMZs

- To check the MMZ memory usage, run **cat /proc/media-mem**.
- When the MMZ memory fails to be obtained due to insufficiency, information about the usage of all MMZ memories are displayed.

2.21 STAT

[Debugging Information]

```
$ cat /proc/msp/stat
=========== host isr stat ===========
isr stat is disabled!
=========== host thread stat ===========
=========== host event stat ===========
KEYIN  = 0  (keyvalue 0x0)
KEYOUT = 0  (keyvalue 0x0)
ASTOP  = 76639
VSTOP  = 76639
CONNECT= 39806859
LOCKED = 39807068
ASTART = 39807641
VSTART = 39807639
CWSET  = 0
STREAMIN      = 39807655
ISTREAMGET     = 39807805 (size 5)
FRAMEDECED     = 39807819
VPSSGET= 39807819
VPSSOUT= 39807833
AVPLAYGET     = 39807837
PRESYNC= 0
BUFREADY      = 0  (type 0)
FRAMESYNCOK = 39807837
VOGET  = 39807837
IFRAMEINTER    = 480
TOTAL  = 39807837
```

[Analysis]

Information about the program switching time statistics and time point of each event during program switching is recorded.

[Parameter Description]

Parameter	Description
KeyIn	Time point when a key is successfully parsed
KeyOut	Time point when a key is fetched
AudStop	Time point when audio playback is stopped
VidStop	Time point when video playback is stopped
Connect	Time point when the turner starts to lock the

Parameter	Description
	frequency
Locked	Time point when the turner successfully locks the frequency
AudStart	Time point when audio playback starts
VidStart	Time point when video playback starts
CWSET	Time point when the key area CW is set
STREAMIN	Time point when the decoder receives streams
ISTREAMGET	Time point when the decoder receives the first I frame
FRAMEDECED	Time point when the decoder successfully decodes the first I frame
VPSSGET	Time point when the VPSS receives the first frame
VPSSOUT	Time point when the VPSS outputs the first frame
AVPLAYGET	Time point when the AVPLAY receives the first frame
PRESYNC	Time point when pre-synchronization is complete
BUFREADY	Time point when frames are buffered
FRAMESYNCOK	Time point when synchronization of the first frame is complete
VOGET	Time point when the VOU receives the first frame
IFRAMEINTER	Interval between I frames in streams
TOTAL	Total time for program switching

⚠ **CAUTION**

If you want to check the time period for switching a program, press the program switching key, release it, and then check the stat information directly without pressing any other key. Otherwise, the values of **KeyIn** and **KeyOut** may be incorrect, which results in incorrect program switching statistics.

2.22 Startup Screen and Fastplay

[Debugging Information]

```
# cat /proc/msp/mce
---------------------------PlayParam------------------------------
enPlayType:      DVB, bPlayEnable:  1
u32VideoPid:   257, u32AudioPid:      258
enVideoType: MPEG2, enAudioType: MP2
u32Volume:50, enTrackMode:1
enSigType:1
u32Freq:   3750000, u32SymbolRate:10490000
enPolar:  0
---------------------------PlayStatus-----------------------------
hAvplay:  0, hWindow:      2359552
BeginTime:    1187, EndTime:    0
```

[Analysis]

Information about fastplay parameters and current fastplay status is recorded.

[Parameter Description]

Parameter	Description
enPlayType	Fastplay type 0: DVB fastplay 1: TS fastplay 2: animation fastplay
bPlayEnable	Fastplay enable 0: disabled 1: enabled
u32VideoPid	Video PID
u32AudioPid	Audio PID
enVideoType	Video protocol type 0: MPEG2 1: MPEG4 2: H264
enAudioType	Audio protocol type 0: MP2 1: MP3
u32Volume	Volume
enTrackMode	Audio channel mode, corresponding to the enumeration HI_UNF_TRACK_MODE_E

Parameter	Description
enSigType	DVB signal type, corresponding to the enumeration HI_UNF_TUNER_SIG_TYPE_E
u32Freq	DVB frequency
u32SymbolRate	DVB symbol rate
enPolar	Polarization mode
hAvplay	Fastplay AVPLAY handle
hWindow	Fastplay window handle
BeginTime	Fastplay start time
EndTime	Fastplay end time

[Debugging Information]

```
# echo stop > /proc/msp/mce
```

[Analysis]

Fastplay stops.

2.23 AENC

[Debugging Information]

```
# cat /proc/msp/aenc00
----------------------- AENC[00] State -------------------------
WorkStatus:start
Codec ID:0x80020001
Description:aac
Sample Rate:48000
Channels:2
BitWidth:16
AttachSource :Track08
TryEncodeTimes:6599
EncodeFrameNum(Total/Error):3768/0
FrameBuf(Total/Use/Percent)(Bytes):524288/672/0%
StreamBuf(Total/Use/Percent) :32/0/0%
SendFrame(Try/OK):3462/3462
ReceiveStream(Try/OK):11471/3768
ReleaseStream(Try/OK):3768/3768
```

[Analysis]

The AENC status is recorded.

[Parameter Description]

Parameter	Description
WorkStatus	Working status of the AENC start: working stop: not working
Codec ID	Audio encoding type. The value of **HA_AUDIO_ID** is displayed in the brackets.
Description	Encoder description
Sample Rate	Sampling rate of streams after encoding
Channels	Number of channels for streams after encoding
BitWidth	Sampling bit width of output data after encoding
AttachSource	Audio source input channel bound during encoding
TryEncodeTimes	Number of attempts for encoding. It can be ignored.
EncodeFrameNum(Total/Error)	Number of frames that the AENC encodes and number of error frames
FrameBuf(Total/Use/Percent)(Bytes)	Total size (unit: byte), used size (unit: byte), and used percentage of the input frame buffer of the AENC
StreamBuf(Total/Use/Percent)	Total size, used size, and used percentage of the output frame buffer of the AENC
SendFrame(Try/OK)	Try: number of attempts to input PCM audio frames into the AENC OK: number of successes to input PCM audio frames into the AENC
ReceiveStream(Try/OK)	Try: number of attempts to obtain encoded audio frames from the AENC OK: number of successes to obtain encoded audio frames from the AENC
ReleaseStream(Try/OK)	Try: number of attempts to release audio frames output by the AENC OK: number of successes to release audio frames output by the AENC

[Debugging Information]

```
#echo save_pcm start > /proc/msp/aenc00
#echo save_pcm stop > /proc/msp/aenc00
#echo save_es start > /proc/msp/aenc00
#echo save_es stop > /proc/msp/aenc00
```

```
#echo help > /proc/msp/aenc00
```

[Analysis]

The PCM data before encoding is recorded by running the following command:

echo save_pcm start|stop > /proc/msp/aenc*XX* (*XX* indicates the number of AENC channels.)

- Start recording.

  echo save_pcm start > /proc/msp/aenc00

  Send the command for saving PCM data before encoding to AENC00. (00 is the AENC channel ID.)
- Stop recording.

  echo save_pcm stop > /proc/msp/aenc00
- View the following files in the specified directory after recording is completed:

  aenc0_00.pcm, **aenc0_01.pcm**
- **aenc0_00.pcm**: first PCM data saved for AENC00
- **aenc0_00.pcm**: second PCM data saved for AENC00

The ES data after encoding is recorded by running the following command:

echo save_es start|stop > /proc/msp/aencXX

- Start recording.

  echo save_es start > /proc/msp/aenc00

  Send the command for saving encoded ES data to AENC00.
- Stop recording.

  echo save_es stop > /proc/msp/aenc00
- View the following files in the specified directory after recording is completed:

  aenc0_00.aac, **aenc0_01.aac**
- **aenc0_00.aac**: first ES data saved for AENC00
- **aenc0_01.aac**: second ES data saved for AENC00

Obtain help information by running the following command:

echo help > /proc/msp/aencXX

2.24 PVR

2.24.1 pvr_play

[Debugging Information]

Proc information in normal playing:

```
# cat /proc/hisi/pvr_xxxx/pvr_play
---------Hisilicon PVR Playing channel Info---------
chan 0 information
Play filename:/hdd/rec_v33_a36.ts
```

```
Stram type:H264DemuxID :0
TsBuffer handle :0xa0401
Avplay handle:0x410000
Cipher handle:0
Play State:PLAYPlay Speed:1024
Stream Read Pos :0x2cef4f4
Index Start:0
Index End:6258
Index Last:6258
IDR flag:0
B frame ref flag:1
Continuous flag :0
DispOptimize flag:0
Start Frm&GOP num:1024 25
Total GOP num:148
Max GOP size:53
Average GOP size:42
Index GOP distr :13 9 5 12 12 96 0 0 0 0 0 0 0
Index Read Now:882
```

Proc information in fast forwarding:

```
# cat /proc/hisi/pvr_xxxx/pvr_play
---------Hisilicon PVR Playing channel Info---------
chan 0 information
Play filename:/hdd/rec_v33_a36.ts
Stram type:H264
DemuxID :0
TsBuffer handle :0xa0401
Avplay handle:0x410000
Cipher handle:0
Play State:FF
Play Speed:4096
Stream Read Pos :0x3ccf94a
Index Start:0
Index End:6258
Index Last:6258
Start Frm&GOP num:1024 25
Total GOP num:148
Max GOP size:53
Average GOP size:42
Index GOP distr :13 9 5 12 12 96 0 0 0 0 0 0 0
Index Read Now:1235
-------- FF control ------------
Status:FF Till End
```

```
Decodec ablity:0
Width:720
Heigth:576
Disp Frame buffer:6
VO Frame buffer:3
Ori frame rate:25
VO In FrmRate(Int:dec):0:781
Max Dec Frm Num 1s:0
IDR flag:0
B frame ref flag:0
Continuous flag:0
DispOptimize flag:0
Field flag:0
---PVR Extract Frm info---
Extract or not:1
Entry frame num:2316
Try frame num:121
Start frame num:2196
End frame num:2316
frame cnt to send:0
Total frame num:0
Total GOP num:0
Total I frame num:0
Total P frame num:0
Total B frame num:0
-----------------------------
```

Proc information when rewinding optimization is enabled during rewinding:

```
# cat /proc/hisi/pvr_xxxx/pvr_play
---------Hisilicon PVR Playing channel Info---------
chan 0 information
Play filename:/hdd/rec_v33_a36.ts
Stram type:H264
DemuxID :0
TsBuffer handle :0xa0401
Avplay handle:0x410000
Cipher handle:0
Play State:FB
Play Speed:-4096
Stream Read Pos :0x11a22a40
Index Start:0
Index End:6258
Index Last:6258
IDR flag:0
```

```
B frame ref flag:1
Continuous flag :0
DispOptimize flag:0
Start Frm&GOP num:1024 25
Total GOP num:148
Max GOP size:53
Average GOP size:42
Index GOP distr :13 9 5 12 12 96 0 0 0 0 0 0 0
Index Read Now:5538
-------- FB control -----------
Status:FB Till Start
Decodec ablity:0
Width:720
Heigth:576
Disp Frame buffer:6
VO Frame buffer:3
Ori frame rate:25
VO In FrmRate(Int:dec):6:250
Max Dec Frm Num 1s:0
IDR flag:0
B frame ref flag:0
Continuous flag:0
DispOptimize flag:0
Field flag:0
Supported max gop size:3
----FB 2X Optimize info--
FBOptimize flag:0
---PVR Extract Frm info---
Extract or not:1
Entry frame num:0
Try frame num:0
Start frame num:0
End frame num:35
frame cnt to send:18
Total frame num:36
Total GOP num:3
Total I frame num:3
Total P frame num:9
Total B frame num:24
Need drop I Frm:0
-----------------------------
```

Proc information when frame extraction is enabled during rewinding:

```
---------Hisilicon PVR Playing channel Info---------
```

```
chan 0 information
Play filename:/tmp//rec_v518_a710.ts
Stram type:MPEG2
DemuxID:0
TsBuffer handle:0xa0401
Avplay handle:0x410000
Cipher handle:0
Play State:FB
Play Speed:-2048
Stream Read Pos:0x71af5f
Index Start:0
Index End:1020
Index Last:1020
Start Frm&GOP num:0 0
Total GOP num:84
Max GOP size:12
Average GOP size:12
Index GOP distr:0 83 0 0 0 0 0 0 0 0 0 0 0 0
Index Read Now:214
-------- FB control -----------
Status:Success
Decodec ablity:65
Width:720
Heigth:576
Disp Frame buffer:6
VO Frame buffer:3
Ori frame rate:26
VO In FrmRate(Int:dec):3:250
Max Dec Frm Num 1s:326
IDR flag:0
B frame ref flag:0
Continuous flag:1
DispOptimize flag:0
Field flag:0
Supported max gop size:3
----FB 2X Optimize info--
FBOptimize flag:1
Display distance:7
FrmCnt not disp From Last :5
Display enable(this gop):1
Vdec FBOptimizeFlag(Should be 1):1
---PVR Extract Frm info---
Extract or not:0
-----------------------------
```

[Analysis]

Status parameters for all playing channels of the current PVR are displayed. In the Proc path **pvr_***xxxx*, *xxxx* indicates the PID of the current PVR playing.

[Parameter Description]

Parameter	Description
Play filename	Path and name of the file that is being played
Stram type	Video format
DemuxID	ID of the Demux used
TsBuffer handle	TS buffer handle
Avplay handle	AVPLAY handle
Cipher handle	CIPHER handle
Play State	Playing status
Play Speed	Playing speed
Stream Read Pos	Position where the stream is read
Index Start	Start index number
Index End	End index number
Index Last	Last index number
Start Frm&GOP num	Number of group of pictures (GOPs) contained in the first *n* frames in a stream
Total GOP num	Total number of GOPs in a stream
Max GOP size	Size of the largest GOP in a stream
Average GOP size	Average size of GOPs in a stream
Index GOP distr	GOP distribution in a stream
Index Read Now	Current index position being read
Status	Smooth control status
Decodec ablity	Chip decoding capability
Width	Current stream width
Heigth	Current stream height
Disp Frame buffer	Total number of frame buffers for video decoding
VO Frame buffer	Number of frame buffers for display
Ori frame rate	Stream frame rate
VO In FrmRate(Int:dec)	Output frame rate (**Int** indicates the integral part and **dec** indicates the decimal part.)

Parameter	Description
Max Dec Frm Num 1s	Maximum number of frames that can be decoded by the decoding module within one second in the current stream
IDR flag	Whether the stream contains the IDR frame
B frame ref flag	Whether the B frame in the stream is referenced
Continuous flag	Stream continuity flag
DispOptimize flag	Display optimization flag
Field flag	Field mode flag
Extract or not	Whether to extract frames
Entry frame num	ID of the frame for entering the next smooth control (displayed only when frames are extracted)
Try frame num	Frame extraction range (displayed only when frames are extracted)
Start frame num	Start frame ID for the frame extraction (displayed only when frames are extracted)
End frame num	End frame ID for the frame extraction (displayed only when frames are extracted)
frame cnt to send	Number of extracted frames (displayed only when frames are extracted)
Total frame num	Total number of frames in the stream (displayed only when frames are extracted)
Total GOP num	Total number of GOPs in the stream (displayed only when frames are extracted)
Total I frame num	Total number of I frames (displayed only when frames are extracted)
Total B frame num	Total number of P frames (displayed only when frames are extracted)
Need drop I Frm	Number of I frames to be discarded (displayed only when frames are extracted during rewinding)
FBOptimize flag	Fast rewind optimization flag (displayed only during rewind optimization)
Display distance	Fast rewind display interval (displayed only during rewind optimization)
FrmCnt not disp From Last	Number of frames that are not displayed (displayed only during rewind optimization)
Display enable(this gop)	Whether to display the GOP (displayed only during rewind optimization)
Vdec FBOptimizeFlag(Should be	Fast rewind optimization enable (displayed only during

Parameter	Description
1)	rewind optimization)

2.24.2 pvr_rec

[Debugging Information]

```
# cat /proc/hisi/pvr_xxxx/pvr_rec
---------Hisilicon PVR Recording channel Info---------
chan 0 information:
L2Cache support:No
Rec filename:/hdd/rec_v513_a660.ts
Stream type:MPEG2
DemuxID :1
Record State:2
Rewind:1
Rewind Type:size
Rewind size:0x27fff1000
Rewind times:0
Max size:0x27fff1000
Max time:0
UserData size:4052
ClearStream:1
IndexType:1
IndexPid:0x201/513
Global offset:0x2a03ba
File offset:0x2a03ba
Index Write:83
CurrentTime(ms) :3204
IndexPos S/E/L:0/236/236
```

Event history[Analysis]

Status parameters for all recording channels of the current PVR are displayed. In the Proc path **pvr_***xxxx*, *xxxx* indicates the PID of the current PVR recording.

[Parameter Description]

Parameter	Description
Rec filename	Path and name of a recording file
L2Cache support	Whether to use the L2 cache
Stream type	Video format
DemuxID	ID of the Demux used

Parameter	Description
Record State	Recording status
Rewind	Rewinding flag for recording
Rewind Type	Rewinding type
Rewind size	Rewinding size
Rewind times	Rewinding time
Max size	Maximum size of a recording file
Max time	Maximum recording duration of a recording file
UserData size	Size of the user-defined data area
ClearStream	Clear stream flag
IndexType	Recorded index type
IndexPid	Recorded index PID
Global offset	Total recorded size, including the overwritten data
File offset	Offset of the currently recorded frame in the recording file
Index Write	ID of the currently recorded frame
CurrentTime(ms)	Recording time of the current frame
IndexPos	Position of the start frame/end frame/last frame of the current index
Event history	Recent five callback events

2.25 VI

[Debugging Information]

```
# cat /proc/msp/vi0000
-------------------------- VI Info -------------------------
Type             :Virtual
InputRect         :0/0/640/480
BufferNum         :6
State             :Start
DstID             :venc(port100)
                 :win(port101)
                 :win(port102)
------------------------- Statistics -------------------------
CAM/USER->VI
CapFrame(Try/OK/Freq):500/500/19
```

```
VI->VPSS
Acquire(Try/OK)    :3160/500
Release(Try/OK)    :500/500
VPSS->VI
Acquire(Try/OK)    :3390/1500
Release(Try/OK)    :1498/1498
VI->WIN
SendFrame(Try/OK) :1000/1000
VI->VENC
SendFrame(Try/OK) :500/500
VI->USER
Acquire(Try/OK)    :0/0
Release(Try/OK)    :0/0
VIBuffer(Total/Used):6/0
VIBufferDetail     :[0,0,0,0,0,0]
```

[Analysis]

Information about the VI is recorded, including VI attributes, status of frames being processed, and buffer queue status.

[Parameter Description]

Parameter	
Type	VI type Virtual: virtual VI
InputRect	The four parameters indicate the X coordinate, Y coordinate, width, and height respectively.
BufferNum	Number of frame buffers
State	State flag Start: started Stop: stopped
LowDelay	Low delay mode flag True: enabled False: disabled
DstID	Information about the bound lower-level modules The lower-level WIN and VENC modules can be bound. Provided that the value of this parameter is **win(port0)**, **win** indicates that the WIN module is bound, and **port0** indicates that the VPSS port handle is 0.

Parameter	
CAM/USER->VI	Information about frames sent by a camera or user to the VI CapFrame(Try/OK/Freq): **Try** indicates the number of attempts to transmit frames. **OK** indicates the number of frames successfully transmitted. **Freq** indicates the frequency for sending frames.
VI->VPSS	Information about frames received by the VPSS from the VI Acquire(Try/OK): number of frames received by the VPSS. **Try** indicates the number of attempts to receive frames. **OK** indicates the number of frames successfully received. Acquire(Try/OK): number of frames released by the VPSS. **Try** indicates the number of attempts to release frames. **OK** indicates the number of frames successfully released.
VPSS->VI	Information about frames received by the VI from the VPSS Acquire(Try/OK): number of frames received by the VI. **Try** indicates the number of attempts to receive frames. **OK** indicates the number of frames successfully received. Acquire(Try/OK): number of frames released by the VI. **Try** indicates the number of attempts to release frames. **OK** indicates the number of frames successfully released.
VI->WIN	Information about frames sent from the VI to the WIN SendFrame(Try/OK): number of sent frames. **Try** indicates the number of attempts to send frames. **OK** indicates the number of frames successfully sent.
VI->VENC	Information about frames sent from the VI to the VENC SendFrame(Try/OK): number of sent frames. **Try** indicates the number of attempts to send frames. **OK** indicates the number of frames successfully sent.
VI->USER	Information about frames obtained from the VI by a user Acquire(Try/OK): number of frames obtained from the VI by a user. **Try**

Parameter	
	indicates the number of attempts to obtain frames. **OK** indicates the number of frames successfully obtained. Release(Try/OK): number of frames released to the VI by a user. **Try** indicates the number of attempts to release frames. **OK** indicates the number of frames successfully released.
VIBuffer	Internal buffer queue status of the VI Total: total number of buffers allocated Used: number of used buffers
VIBufferDetail	Node status of the VI internal buffer queue 0: The current node is not used. 1: The current node is being used by the VPSS. 2: The current node is used and released.

[Debugging Method]

You can process internal behavior or data of the VI during dynamic operation of the VI to identify problems. You can debug the VI by using the following method:

- Save internal luminance-bandwidth-chrominance (YUV) data frames.

```
echo save_yuv > /proc/msp/vi0000
```

Save YUV data frames to the **/mnt/** directory, open the file by using the YUV picture software, and identify problems.

- Start delay statistics.

```
echo low_delay_stat start > /proc/msp/vi0000 (Starts delay statistics.)
```

After delay statistics start, run **cat /proc/msp/low_delay_statistics** to view the time required for each module to process frames.

echo low_delay_stat stop > /proc/msp/vi0000 (Stops delay statistics.)

2.26 VP

[Debugging Information]

```
# cat /proc/hisi/vp/vp_info
---------Hisilicon VP Out Info---------
chan 0 information:
---- VP Source Info ----
Attached:YES
VP SrcMode:USBCAM
Cap Type:MJPEG
```

```
Src handle:0x10
Buf count:4
---- VP Preview Info ----
Work State:Run
Window_ID:win0100
Avplay_ID:avplay01
---- VP Sender Info ----
Work State:Run
Venc Type:H264
Venc_ID:venc00
Avplay_ID:avplay01
---- VP EmptyPack Info ----
EmptyPackage:Stop
---- VP Receiver Info ----
Work State:Run
Vdec Type:H264
Avplay_ID:avplay00
Window_ID:win0101
----- VP Network Info -----
Ipv4:YES
Rtcp Enable:YES
SendInterval:5000(ms)
---Sender side Statics:
Send(byte):1315350354
Send(packet):924580
LostRate:0%
Lost Packets:28
Jitter:3(ms)
Delay:0(ms)
---Receiver side Statics:
Recv(byte):1317308610
Recv(packet):925951
LostRate:0%
Lost Packets:28
Jitter:3(ms)
Delay:0(ms)
```

[Analysis]

Information about the VP is displayed.

[Parameter Description]

Parameter	Description
Source Info	Information about the input source

Parameter	Description
	Attached: whether the input source is bound Yes: bound No: not bound VP SrcMode: input source type (USBCAM or VI) Cap Type: type of data captured by the USB camera (MJPEG or YUYV) Src handl: source handle Buf count: number of allocated buffers (ranging from 4 to 16)
Preview Info	Local preview information Work State: run/stop state of preview Window_ID: ID of the window instance used for preview Avplay_ID: ID of the AVPLAY instance used for preview
Sender Info	Information about the transmit end Work State: run/stop state of the transmit end Venc Type: encoding type Venc_ID: ID of the VENC instance for transmitting the local video Avplay_ID: ID of the AVPLAY instance for transmitting the local video
EmptyPack Info	Empty packet information EmptyPackage: whether the transmit end is transmitting empty packets. Run indicates yes, and Stop indicates no.
Receiver Info	Information about the receive end Work State: run/stop state of the receive end Vdec Type: decoding type Avplay_ID: ID of the AVPLAY instance for playing the video from the peer end Window_ID: ID of the window instance for playing the video from the peer end
Network Info	Network information Ipv4: whether IPv4 is used. **Yes** indicates IPv4, and **No** indicates IPv6. Rtcp Enable: whether to enable RTCP. **Yes** indicates enabled, and **No** indicates disabled. SendInterval: RTCP transmit interval (in the unit of ms) Transmit end statistics Send(byte): number of transmitted bytes Send(packet): number of transmitted packets LostRate: packet loss rate Lost Packets: number of lost packets Jitter: jitter, in the unit of ms

Parameter	Description
	Delay: delay, in the unit of ms Receive end statistics Recv(byte): number of received bytes Recv(packet): number of received packets LostRate: packet loss rate Lost Packets: number of lost packets Jitter: jitter, in the unit of ms Delay: delay, in the unit of ms

2.27 PDM

[Debugging Information]

```
# cat /proc/msp/pdm
---------------------------Dispaly0------------------------------
format:              11,
source display:       1
background color:      0x000000
HuePlus/Brightness/Contrast/Saturation: 50/50/50/50
virtual screen(Width/Height):         1280/720
offset(Left/Top/Right/Bottom):        0/0/0/0
bGammaEnable:         0
pixelformat:          9
aspectRatio:          4to3
CVBS:                3
---------------------------Dispaly1------------------------------
format:              6,
source display:       1
background color:      0x000000
HuePlus/Brightness/Contrast/Saturation: 50/50/50/50
virtual screen(Width/Height):         1280/720
offset(Left/Top/Right/Bottom):        0/0/0/0
bGammaEnable:         0
pixelformat:          9
aspectRatio:          16to9
HDMI:                HDMI_0
YPbPr(Y/Pb/Pr):       1/2/0
---------------------------Sound0------------------------------
DAC:                       DAC0
SPDIF:                      SPDIF0
```

```
HDMI:                       HDMI0
I2S0 attr:
       Master:              TRUE
       PcmSampleRiseEdge:     TRUE
       Master colock:        128*fs
       Bit colock:           1 DIV
       Channel:             1
       Mode:                standard
       Bit Depth:            16
       PCM Delay Cycle:       0
```

[Analysis]

Parameter information of the baseparam partition is displayed.

[Parameter Description]

Parameter	Description
format	Display format, corresponding to **HI_UNF_ENC_FMT_E**
source display	Same-source display
background color	Background color. For example, **FF00FF** indicates that the red, green, and blue colors are 0xFF, 0x0, and 0xFF respectively.
HuePlus/Brightness/ Contrast/Saturation	Hue, brightness, contrast, and saturation
virtual screen(Width/Height)	Width and height of the virtual screen
offset(Left/Top/Right/ Bottom)	Offset on the virtual screen
bGammaEnable	Whether to enable gamma
pixelformat	Pixel format, corresponding to **HIGO_PF_E**
aspectRatio	Aspect ratio of a device, corresponding to **HI_UNF_DISP_ASPECT_RATIO_E**
CVBS	CVBS DAC
HDMI	HDMI ID
YPbPr(Y/Pb/Pr)	YPbPr DAC
DAC	DAC interface
SPDIF	SPDIF interface
HDMI	HDMI interface
Master	Master/Slave mode.

Parameter	Description
	TRUE: master mode FALSE: slave mode
PcmSampleRiseEdge	PCM sampling edge TRUE: rising edge FALSE: falling edge
Master colock	Master working clock
Bit colock	Bit working clock
Channel	I^{2}S audio channel
Mode	I^{2}S interface mode Standard: I^{2}S standard mode PCM mode
Bit Depth	Bit width
PCM Delay Cycle	PCM timeslot

2.28 VENC

[Debugging Information]

```
# cat /proc/msp/venc00
--------------------- VENC[00] -----------------------
-------------------- User  Config ----------------------
Version                   :001.002.2015042101
CodecID                   :H.264(0x4)
Capability                 :720P
Profile(Level)             :High(31)
Resolution                 :1280X720
TargetBitRate             :4194(kbps)
Gop                       :100
FrmRate(Input/OutPut)     :50/25(fps)
priority                  :0
QuickEncode               :FALSE
DriftRateThr              :NA
Split                     :Enable(FALSE)   Size(0)
StreamBufSize             :1843(KB)
MaxQP/MinQP                :48/16
QLevel                     :0
Rotation                  :0
AutoRequestIFrm            :Enable(FALSE)
```

```
------------------ Real-time  Statistics ----------------------
WorkStatus                 :Start
SourceID                   :Win00
FrameInfo                  :SP420_UV
InputFrmRate(Use/Real)      :60/60(fps)
TargetFrmRate(Use/Real)     :25/25(fps)
BitRate                     :3556(kbps)
EncodeNum                   :268
SkipNum                     :Total(370) FrmRateCtrl(370) SamePTS(0)
QuickEncode(0) TooFewBuf(0) TooManyBits(0),ErrCfg(0)
FrameBuffer:
   VpssQueueBuf(Total/Used) :NA/NA
   VpssImgBuf  (Total/Used) :NA/NA
   VencQueueBuf(Total/Used) :6/0
StreamBuffer:
   Total/Used/Percent(Bytes):1843200/0/0%
Statistics(Total):
   AcquireFrame(Try/OK)      :950/638
   ReleaseFrame(Try/OK)     :638/638
   AcquireStream(Try/OK)    :1659/274
   ReleaseStream(Try/OK)    :274/274
Statistics(PerSecond):
  AcquireFrame(Try/OK)    :83/60
ReleaseFrame(Try/OK)    :59/59
```

[Analysis]

The VENC debugging information describes the encoder types and parameters. The VENC encodes the input frames and transfers the raw streams to the next unit. The VENC debugging information also describes the frame data and stream data.

[Parameter Description]

Parameter	Description
CodecID	VENC type 0x1: MPEG-4 0x4: H.264
Capability	Encoding capability level QCIF, CIF, D1, 720P, 1080P, and UNKNOWN#
Profile(Level)	Encoding standard (level) Supported standard: baseline, main profile, and high profile Supported level: 1.0–4.1
Resolution	Encoding output resolution

Parameter	Description
TargetBitRate	Target bit rate specified by a user, with a unit of kbit/s
Gop	A GOP contains at least one I frame, and Gop indicates the length of a picture group.
FrmRate	Frame rate Input: actual input frame rate, with a unit of FPS Encoded: actual frame rate for encoding, with a unit of FPS
Priority	Channel priority, ranging from 0 to 7 in ascending order
QuickEncode	Quick encoding mode FALSE: disabled TRUE: enabled
DriftRateThr	Encoder bit rate drift threshold. If it is **20**, the drift threshold is 20%.
Split	Slice encoding Enable: whether to enable slice encoding. **FALSE** indicates the disabled state, and **TRUE** indicates the enabled state. Size: slice size, with a unit of byte (available when slice encoding is enabled)
StreamBufSize	Stream buffer size specified by a user, with a unit of KB
MaxQP/MinQP	Quantization parameter, ranging from 0 to 51. MaxQP: maximum quantization parameter specified by a user MinQP: minimum quantization parameter specified by a user
QLevel	JPEG encoding quantification parameter, ranging from 1 to 99 The larger the quantization parameter, the better the JPEG image quality.
Rotation	Rotational angle
WorkStatus	Working status of an encoding instance Start: starting Stop: stopped
SourceID	Encoding source The string indicates the source type, and the

Parameter	Description
	following digits indicate the handle ID. For example, **VI01** and **Win02**
FrameInfo	Input frame format The string indicates the storage format, the following digits indicate the sampling format, and the information behind the underscore (_) indicates the pixel order. For example, SP420_UV, Planer420, and Package422_YUYV
InputFrmRate	Real-time input frame rate, with a unit of FPS Use: actual input frame rate of the VENC Real: real-time input frame rate of the VENC
OutputFrmRate	Real-time output frame rate, with a unit of FPS Use: actual output frame rate of the VENC Real: real-time output frame rate of the VENC Remarks: The actual input or output frame rate may be inconsistent with the value set by a user. In bound mode, the internal input frame rate shall prevail. When the output frame rate set by a user is larger than that contained in the frame information, the output frame rate contained in the frame information shall prevail. This ensures that the output frame rate is not larger than the input frame rate.
BitRate	Actual bit rate for encoding, with a unit of kbit/s
EncodedNum	Number of encoded frames
SkipNum	Number of frames that are not encoded Total: total number of frames that are not encoded FrmRcCtrl: number of frames that are not encoded because of frame rate control SamePTS: number of frames that are not encoded because of the same PTS QuickEncode: number of frames that are not encoded because of quick encoding TooFewBuffer: number of frames that are not encoded because of insufficient internal stream buffer

Parameter	Description
	ErrCfg: number of frames that are not encoded because of frame configuration errors
FrameBuffer	Frame buffer usage VpssQueueBuf(Total/Used): total number of VPSS input frame buffer queues created by the VENC/number of used frames VpssImgBuf(Total/Used): total number of VPSS output frame buffer queues created by the VENC/number of used frames VencQueueBuf(Total/Used):total number of VENC frame buffers/number of used frames
StreamBuffer	Internal stream buffer utilization, with a unit of byte Total/Used/Percent: total size, used size, and used percentage (%) of stream buffer#
Statistics(Total)	General statistics AcquireFrame(Try/OK): total number of attempts or successes to obtain frames ReleaseFrame(Try/OK): total number of attempts or successes to release frames AcquireStream(Try/OK): total number of attempts or successes to obtain streams ReleaseStream(Try/OK): total number of attempts or successes to release streams#
Statistics (PerSecond)	Information about output frames AcquireFrame(Try/OK): number of attempts or successes to obtain frames per second ReleaseFrame(Try/OK): number of attempts or successes to release frames per second

[Debugging Method]

- You can read the stream data being processed by the VENC and save it as a file for debugging by writing the Proc file when the VENC is running. The file is saved in the **/mnt** directory by default.

```
echo save_stream P1 P2 > /proc/msp/vencXX
```

  - When **P1** is **second**, **P2** serves as a time parameter with a unit of second. You can save the streams generated by the VENC as a file.

    For example, you can read the stream data generated in 5 seconds and save the stream data as a file by running the **echo save_stream second 5> /proc/msp/venc** command.

  - When **P1** is **frame**, **P2** serves as a frame parameter with a unit of frame. You can save the streams generated by the VENC as a file.
  - For example, you can read stream data from 60 frames and save it as a file by running the **echo save_stream frame 60>/proc/msp/venc** command. Read the stream file from the board for debugging.
- You can read the YUV data before encoding by the VENC and save it as a file for debugging by writing the Proc file when the VENC is running.

  echo save_yuv start > /proc/msp/venc

  - Start saving the YUV file.

echo save_yuv stop > /proc/msp/venc

- Stop saving the YUV file.
- You can control the VENC hardware low-power related configurations by writing the Proc file when the VENC is running.

  echo ClkGateEn P1 > /proc/msp/vencXX

  - When **P1** is **0**, disable the VEDU clock gating.
- When **P1** is **1**, enable only the VEDU frame clock gating.
- When **P1** is **2**, enable both the VEDU frame and macroblock clock gating.

  echo LowPowEn P1 > /proc/msp/vencXX
- When **P1** is **0**, disable the encoding algorithm low power.
- When **P1** is **1**, enable the encoding algorithm low power.

2.29 AI

[Debugging Information]

```
# cat /proc/msp/ai0
--------------------- AI[0] Status ---------------------
Status      :start
AiPort      :I2S0
SampleRate  :48000
PcmFrameMaxNum     :6
PcmSamplesPerFrame :30720
DmaCnt      :1960
BufFullCnt :0
FiFoFullCnt  :0
FrameBuf(Total/Use/Percent)(Bytes) :184320/82176/44%
AcquireFrame(Try/OK) :98/32
ReleaseFrame(Try/OK) :97/97
```

[Analysis]

The AI status is recorded.

[Parameter Description]

Parameter	Description
Status	Working status start: working stop: not working
AiPort	Audio input port
SampleRate	Sampling frequency for audio input
PcmFrameMaxNum	Maximum number of frames in the buffer
PcmSamplesPerFrame	Number of data sampling points per frame
DmaCnt	Times of transferring data by the DMA
BufFullCnt	Times of buffer overload
FiFoFullCnt	Times of FiFo overload
FrameBuf(Total/Use/Percent)(Bytes)	Total size (unit: byte), used size (unit: byte), and used percentage of the frame buffer
AcquireFrame(Try/OK)	Try: number of attempts to obtain frame buffer OK: number of successes to obtain frame buffer
ReleaseFrame(Try/OK)	Try: number of attempts to release frame buffer OK: number of successes to release frame buffer

2.30 VPSS

[Debugging Information]

```
# cat /proc/msp/vpss00
```

The proc information contains the following two parts:

```
|--------VPSS0000----------------|
|--------PortInfo----------------|
```

That is:

- Instance information
- Port information

2.30.1 Instance Information

The instance information is as follows:

```
--------VPSS0000------------- |
ID:0x0|
State:working|
Priority :0|
QuickOutPut:off|
SourceID :Vdec(00)|
Version:2013082419|
-------- Algorithm------------|
P/I Setting:auto|
Deinterlace:5 field|
Sharpness:auto|
*ProgRevise:on
--------Detect Info------------|
TopFirst(Src):NA(Top)|
InRate(Src):0(25000 )|
*Trans:OFF|
Progressive/Interlace(Src):P(P)|
-----SourceFrameList Info-----|
(source to vpss) |
Mutual Mode:vpss active|
GetSrcImg(Try/OK): 25/25|
GetOutBuf(Try/OK):118/25|
ProcessHZ(Try/OK):118/25|
Acquire(Try/OK):|
2742/2740|
Release(Try/OK):|
2739/2739|
```

[Analysis]

Instance information of the VPSS is recorded, including the instance status, algorithm configuration, detection information, and upper-level buffer interaction information.

[Parameter Description]

Parameter	Description
ID	ID of an instance
State	Working status of an instance Working/Stop: whether the VPSS instance is working
Priority	Instance priority, ranging from **0** to **32** in ascending order The default value is **16**.
QuickOutPut	Quick output mode On/off: When the quick output mode is enabled, the VPSS processes the latest frame of the upper level module.

Parameter	Description
Source Module ID	Name and ID of the upper-level module Module name: **Vi**, **Vdec**, or **Venc** Module ID: **00**, **01**, or **02**
P/I Setting	Progressive or interlaced detection mode **I**, **P**, and **auto** indicate forced interlaced input source, forced progressive input source, and automatic interlaced/progressive correction respectively.
Deinterlace	De-interlacing algorithm mode **off**, **auto**, **3 field**, **4 field**, **5 field** indicate disabled, adaptation based on picture information, 3 field mode, 4 field mode, and 5 field mode respectively.
Sharpness	Sharpening algorithm mode **off**, **on**, and **auto** indicate disabled, enabled, and adaptation based on picture information respectively.
TopFirst(Src)	Detected field sequence information (field sequence information in a stream)
InRate(Src)	Detected frame rate information (frame rate information in a stream)
Progressive/Interlace(Src)	Corrected progressive/interlaced information (progressive/interfaced information in a stream)
Mutual Mode	Interaction mode with the upper-level module **src active** and **vpss active** indicate the upper-level pushing mode and VPSS proactive acquisition mode respectively.
GetSrcImg(Try/OK)	Real-time statistics on the buffer interaction with the upper-level module **Try** and **OK** indicate the number of attempts to request frames from the upper-level module within 1 second and the success times respectively.
GetOutBuf(Try/OK)	Real-time statistics on the output buffer **Try** and **OK** indicate the times of querying the output buffer within 1 second and the times that the output buffer can be written respectively.
ProcessHZ(Try/OK)	Real-time statistics on the instance processing times **Try** and **OK** indicate the times of querying an instance by the service threads within 1 second and the times of processing the current instance within 1 second respectively.
Acquire(Try/OK)	Statistics on the buffer interaction with the upper-level module **Try** and **OK** indicate the number of attempts to obtain frames from the upper-level module and the success times respectively.

Parameter	Description
Release(Try/OK)	Statistics on the buffer interaction with the upper-level module **Try** and **OK** indicate the number of attempts to release frames to the upper-level module and the success times respectively.

2.30.2 Port Information

The port information is as follows:

```
------------PortInfo-----------
ID :0x0
State      :on
PixelFormat:YCrCb420
Resolution :1280*720
ColorSpace : BT709_YUV
DisplayPixelAR(W/H):16/9
Aspect Mode:Full
Support3DStream   :off
MaxFrameRate      :2500
LowDelay    :off
--------OutFrameList Info-------
 (vpss to sink)
BufManager :vpss
BufNumber   :6+0
BufFul      :5
BufEmpty    :1
AcquireHZ   :26
Acquire(Try/OK):
      2719/2719
Release(Try/OK):
      2714/2714
```

[Parameter Description]

Parameter	Description
ID	Port ID. **0xffffffff** indicates that the port has not been created.#
State	Port status Off: disabled On: enabled
PixelFormat	Output pixel order format

Parameter	Description
Resolution	Output resolution
ColorSpace	Output color space#
DisplayPixelAR(W/H)	Aspect ratio of a display device#
Aspect Mode	Aspect ratio conversion mode **Full**, **LBOX**, **PANSCAN**, **COMBINED**, **FULL_H, FULL_V**, and **CUSTOMER**
Support3DStream	Output of 3D frame packing data When this mode is enabled, 3D streams are processed to generate frame packing data.
MaxFrameRate	Maximum output frame rate When the maximum output frame rate is smaller than the input frame rate, the VPSS discards frames.
LowDelay	Low delay mode
HorizonFlip	Horizontal flip enable
VerticalFlip	Vertical flip enable
Rotation	Rotation mode Off/90/180/270: rotation disabled or rotated by 90/180/270 degrees
BufManager	Output frame buffer management module **vpss** and **usr** indicate VPSS-based management and user-based management respectively.#
BufNumber	Number of frame buffers at the output port **M+N**: **M** indicates the number of left-eye frame buffers, and **N** indicates the number of right-eye frame buffers.
BufFul	Number of frame buffers that have been written
BufEmpty	Number of idle frame buffers
Acquire HZ	Count of frames obtained by the lower-level module per second
Acquire(Try/OK)	Statistics on the buffer interaction with the lower-level module **Try** and **OK** indicate the number of attempts to obtain frames by the lower-level module and the success times respectively.
Release(Try/OK)	Statistics on the buffer interaction with the lower-level module

Parameter	Description
	Try and **OK** indicate the number of attempts to release frames by the lower-level module and the success times respectively.

2.31 KEYLED

[Debugging Information]

```
# cat /proc/msp/keyled
---------Hisilicon KeyLed Info---------
KeyStatus                    :Open
LedStatus                    :Open
Select KeyLed Type             :CT1642
Timeout of Reading Key          :500(ms)
Key Up Report                 :Enable
Repeat Key Report              :Enable
Repeat Key Report time          :200(ms)
Led Display Code              :0xbeb666f2
Led Display Time              :12:25(hour:minute)
Led Flash Pin                 :Fourth
Led Flash Pin Level            :1
Key Buffer Length               :8
Key Buffer Head               :6
Key Buffer Tail               :6
Key Come Number               :6
Key Read Number               :6
```

[Analysis]

The KEYLED module status is recorded.

[Parameter Description]

Parameter	Description
KeyStatus	Key status Open: enabled Close: disabled
LedStatus	LED status Open: enabled Close: disabled
Select KeyLed Type	Current front panel driver type

Parameter	Description
	PT6961: front panel with the PT6961 chip CT1642: front panel with the CT1642 chip PT6964: front panel with the PT6964 chip FD650: front panel with the FD650 chip
Timeout of Reading Key	Timeout period for obtaining the key, in ms
Key Up Report	Whether to report the key-up status Enable: report Disable: not report
Repeat Key Report	Whether to report the repeated use of the same key Enable: report Disable: not report
Repeat Key Report time	Time interval for reporting repeated use of the same key, in ms
Led Display Code	Codes corresponding to the digits or characters displayed on the LED
Led Display Time	Time displayed on the LED (hour:minute)
Led Flash Pin	LED blink position (valid when the driver supports blinking) Not Set: Blinking is not set. First: The first LED tube blinks. Second: The second LED segment blinks. Third: The third LED segment blinks. Fourth: The fourth LED segment blinks. All: The four LED segments all blink. NONE: None of the LED segments blinks.
Led Flash Pin Level	LED blinking level, ranging from 1 to 5. The greater the value, the higher the blinking frequency.
Key Buffer Length	Length of the key buffer
Key Buffer Head	Position of the read pointer of the key buffer
Key Buffer Tail	Position of the write pointer of the key buffer
Key Come Number	Total number of key values received by the driver
Key Read Number	Total number of key values read from the driver

2.32 PMOC

[Debugging Information]

```
# cat /proc/msp/pm
---------Hisilicon PM Info---------
Scene                         :Standard
Ethernet WakeUp                :Disable
USB WakeUp                    :Disable
Display Mode                   :Display Time
Display Value                  :0x00093a00
WakeUp TimeOut                 :4294967295(s)
KeyLed Type                   : CT1642
KeyLed WakeUp Key               :0
IR Type                       :NEC Simple
IR WakeUp Key Number            :1
IR WakeUp Key                 :High 32-bit(0x0), Low 32-bit(0x639cff00)
  Wake Up Reset              :Enable
Debug Mode                    :Save debug info to ram
Gpio wake up port               :255
Power off GPIO Number           :0
Power off GPIO Direction         :0
```

[Analysis]

The PMOC module status is recorded.

[Parameter Description]

Parameter	Description
Scene	Standby scenario Standard: standard: scenario
Ethernet WakeUp	Network wakeup Enable: enabled Disable: disabled
USB WakeUp	Wakeup by using the USB mouse or keyboard Enable: enabled Disable: disabled
Display Mode	Front panel display mode in standby mode No Display: No information is displayed. Display Number: Digits are displayed. Display Time: Time is displayed.
Display Value	Digits or time displayed on the LED in

Parameter	Description
	standby mode When **Display Mode** is set to **Display Number**, bits 24 to 31 indicate the digits (0 to 9) displayed on the first LED segment, bits 16 to 23 indicate the digits displayed on the second LED segment, bits 8 to 15 indicate the digits displayed on the third LED segment, and bits 0 to 7 indicate the digits displayed on the fourth LED segment. When **Display Mode** is set to **Display Time**, bits 24 to 31 are meaningless, bits 16 to 23 indicate the hour, bits 8 to 15 indicate the minute, and bits 0 to 7 indicate the second.
WakeUp TimeOut	Time interval for scheduled standby wakeup, in s
KeyLed Type	Front panel driver type used during standby wakeup PT6961: front panel with the PT6961 chip CT1642: front panel with the CT1642 chip PT6964: front panel with the PT6964 chip FD650: front panel with the FD650 chip
KeyLed WakeUp Key	Wakeup key on the front panel used for standby wakeup
IR Type	Type of the remote control used for standby wakeup NEC Simple: remote control complying with the NEC Simple protocol TC9012: remote control complying with the TC9012 protocol NEC Full: remote control complying with the NEC Full protocol SONY: remote control complying with the SONY protocol RAW: simultaneous parsing of remote controls complying with various protocol types
IR WakeUp Key Number	Number of used standby wakeup keys of the remote control, dependent on **IR Type**. When **IR Type** is set to **RAW**, a maximum of six groups of keys can be configured; otherwise, only one group can be configured.
IR WakeUp Key	Remote control key values used during standby wakeup (divided into upper 32 bits

Parameter	Description
	and lower 32 bits)
Wake Up Reset	Whether to restart the board during standby wakeup Enable: restart Disable: not restart
Debug Mode	Debugging mode No debug info: no debugging information Save debug info to ram: The debugging information is saved in the memory. Show debug info to LED: The debugging information is displayed on the LED. Save to ram Show to LED: The debugging information is saved to the memory while being displayed on the LED. Show debug info in serial port: The debugging information is displayed over the serial port.
Gpio wake up port	GPIO wakeup port. **255** indicates disabled. Currently only the GPIO that is not powered off during standby is valid. For details, see the chip data sheet.
Power off GPIO Number	ID of the GPIO pin that independently controls the power supply during standby
Power off GPIO Direction	Whether the GPIO pin that independently controls the power supply during standby outputs high level or low level

The default standby wakeup mode is raw wakeup, that is, the system resumes the state before entering the standby mode. If you want to set the standby mode to the mode of system restart after wakeup, run the following command:

```
#echo reset=1 > /proc/msp/pm
```

Run the following command to switch back to the raw wakeup mode:

```
#echo reset=0 > /proc/msp/pm
```

The debugging mode is disabled in standby mode by default. Run the following command to check the wakeup parameters when the system is entering the standby mode.

```
#echo debug=4 > /proc/msp/pm
```

2.33 HiPlayer

2.33.1 Viewing HiPlayer Proc Information

Run **echo help >/proc/hisi/hiplayer00/contro**l over the serial port.

00 is the ID of the HiPlayer instance. If two HiPlayer instances are created, they will be hiplayer00 and hiplayer01.

[Debugging Information]

```
# echo help >/proc/hisi/hiplayer00/control
************Hisilicon HiPlayer00 Help Info Begin**********************
echo SendAud = 1|0 >/proc/hisi/hiplayer00/control
echo SendVid = 1|0 >/proc/hisi/hiplayer00/control
echo SendSubt = 1|0 >/proc/hisi/hiplayer00/control
echo DumpAVEs = 1|0 >/proc/hisi/hiplayer00/control
echo help >/proc/hisi/hiplayer00/control
*************Hisilicon HiPlayer00 Help Info End*********************
```

[Command Description]

Command	Function
echo SendAud = 1|0 >/proc/hisi/hiplayer00/control	Enables/Disables the audio data output. 0: The HiPlayer does not transmit audio data to the AVPLAY. The AVPLAY has no audio output after playing data in the buffer. 1: The HiPlayer transmits audio data to the AVPLAY. The default value is **1**.
echo SendVid = 1|0 >/proc/hisi/hiplayer00/control	Enables/Disables the video data output. 0: The HiPlayer does not transmit video data to the AVPLAY. A static frame is displayed in the video window after data in the buffer is played. 1: The HiPlayer transmits video data to the AVPLAY. The default value is **1**.
echo SendSubt = 1|0 >/proc/hisi/hiplayer00/control	Enables/Disables the subtitle data output. 0: The HiPlayer does not transmit subtitle data to the subtitle output (SO) module. No subtitle is output after the SO module outputs subtitle data in the buffer. 1: The HiPlayer transmits subtitle data to the SO module. The default value is **1**.
echo DumpAVEs = 1|0 >/proc/hisi/hiplayer00/control	Enables/Disables the function of storing audio and video ES data. Ensure that the **/sdcard/hiplayer** directory has been created and the write permission has been obtained before this command is executed. The audio and video

Command	Function
	ES data files are stored in this directory by default. 1: enable 0: disable The generated files are as follows: # ls /sdcard/hiplayer/ -l hiplayer_aud_stream_00.dat hiplayer_vid_stream_00.dat If this function is enabled and disabled frequently during the playback of a stream, multiple ES files are generated, which are differentiated by using the IDs at the end of the file names.

After running the preceding commands, you can view the configured values by **running cat /proc/hisi/hiplayer00/control**.

2.33.2 Viewing File Information

Run **cat /proc/hisi/hiplayer00/fileinfo** over the serial port.

[Debugging Information]

```
# cat /proc/hisi/hiplayer00/fileinfo
*************Hisilicon HiPlayer00 Media File Info Begin**************
Stream type: ES
Source type: LOCAL
File size: 5273124381 bytes
Start time:  0:0:0
Duration:   0:41:29
bps:       16945523 bits/s
Is Divx File:NO
Program 0:
 video 0 info:
     stream idx: 1
     format:      H264
     w * h:       1920 * 1080
     fps:          23.976
     bps:          15406664 bits/s
     duration:    0:41:29
 audio 0 info:
     stream idx: 0
     format:       PCM
     samplerate: 48000 Hz
     bitpersample: 16
     channels:     2
```

```
    bps:          16 bits/s
    lang:         eng
    subID:        -1
    duration:     0:41:29
******************Hisilicon HiPlayer00 Media File Info
End*********************
```

[Analysis]

Running this command displays the audio and video information about the file being played. If the audio or video fails to be played, you can run this command to check whether the corresponding format is supported.

[Parameter Description]

Parameter	Description
Stream type	Stream type, TS or ES
Source type	Media type, local or network media
File size	File length, in byte
Start time	Playback start time
Duration	Total playback duration
fps	File frame rate, in frame/s
bps	File bit rate, in bit/s
Is Divx File	Whether the file is a Divx file (Yes/No)
Program n	A program in the file, which can contain multiple videos, audios, and subtitle streams
Video n info	Video information of channel *n*
Audio n info	Audio information of channel *n*
stream idx	ID of the stream in the program
format	Encoding format
w * h	Width and height
samplerate	Sampling rate
bitpersample	Sampling bit width
channels	Number of audio channels
lang	Audio language
subID	ID of the sub-audio stream. For the TrueHD audio, this field may contain one AC3 sub-stream. The processing strategy is to separate the two streams and record their relationship by using this field. If there is no sub-stream, **subID** is **−1**.

2.33.3 Viewing the HiPlayer Control Information

Run **cat /proc/hisi/hiplayer00/control** over the serial port.

[Debugging Information]

```
cat /proc/hisi/hiplayer00/control
*****Hisilicon HiPlayer00 Out Info Begin*********
________HiPlayer Control Info Begin_______________
CurStatus                          :PLAY
GotoStatus                         :INVALID
PlaySpeed                          :(x)
CurProgramId                        :0
CurVideoId                         :0
CurAudioId                         :0
CurSubtitleId                       :0
EndOfFile                          :0
StopCurOpe                       :0
EmptyCount                         :0
PlayProgress                        :71(0-100)
DownloadProgress                   :0(0-100)
LastPlayDuration                   :1773294(ms)
FirstPts                            :0(ms)
LastPts                             :1784242(ms)
LastIFramePts                       :1783783(ms)
LastSeekPts                        :1773294(ms)
CallSeekPts                        :-1(ms)
FirstPtsAfterSeek                   :0(ms)
CallSeekPos                        :-1(byte)
LastVidPos                         :3763485855(byte)
SubtitleCodecType                   :0
LogLevel                            :1 (0|1|2|3|4 =
QUITE|FATAL|ERROR|WARNING|INFO)
_________HiPlayer Control Info End________________
__________HiPlayer Tplay Control Info Begin________
TotalSendIFrame                    :1783
SystimeOfSendLastIFrame             :166310397(ms)
TotalIFrameScale                   :0(ms)
BaseSystimeOfTPlay                  :0(ms)
BaseIFramePtsOfTPlay                 :0(ms)
NumOfDiscardIFrame                  :0
NumOfSendPBFrame                     :0
_________HiPlayer Tplay Control End________________
________HiPlayer NetBuffer Info Begin_____________
StatusOfBuffer                       :INVALID
```

```
DurationOfFormatBuffer              :0(ms)
SizeOfFormatBuffer                  :0(bytes)
DurationOfPlayerBuffer              :0(ms)
SizeOfPlayerBuffer                  :0(bytes)
DownloadFinish                       :0
ResetPlayer                          :0
ReportFirstFrameTime                 :1
SetBufferUnderrun                    :0
UnderrunCount                        :0
________HiPlayer NetBuffer Control End____________
________HiPlayer Proc Control Info Begin__________
CurStreamIdx                         :0
CurFrameType                         :AUD
CurFrameSize                         :4096(bytes)
CurFramePts                          :1784746(ms)
SendAudio                            :0
SendVideo                            :0
SendSubtitle                          :0
DumpAVEsData                          :1
PrintFrameInfo                       :0
ReadFrameFromDemux(Try/OK)            :128478/128478
SendVidToAVPlay(Try/OK)               :65183/42781
SendAudToAVPlay(Try/OK)               :206776/85696
SendSubToSo(Try/OK)                   :0/0
_______HiPlayer Proc Control End_________________
*****Hisilicon HiPlayer00 Out Info End*********
```

[Analysis]

The media playback control information of the HiPlayer can be viewed.

[Parameter Description]

Parameter	Description
CurStatus:PLAY	Current status of the player
GotoStatus:INVALID	Status that the player is to be switched to. **INVALID** indicates that the status is not switched. If you perform the stop operation, **GotoStatus** is **STOP**.
PlaySpeed:(x)	Playback speed, for example, 4x, 8x, or 16x speed#
CurProgramId:0	ID of the program being played. A stream may contain multiple programs, similar to TSs.
CurVideoId:0	ID of the video being played. If the program

Parameter	Description
	being played contains three videos, the video ID ranges from 0 to 2.
CurAudioId:0	ID of the audio being played. If the program being played contains five audio clips, the audio ID ranges from 0 to 4.
CurSubtitleId:0	ID of the subtitle being played. If the program being played contains nine subtitles, the subtitle ID ranges from 0 to 8.
EndOfFile:0	Whether the end of file is reached 1: The end of file is reached, but the player continues playing because data in the AVPLAY buffer may not be played completely. 0: The end of file is not reached.#
StopCurOpe:0	Whether to stop the current operation forcibly. The current operation can be forcibly stopped by calling HI_SVR_PLAYER_Invoke(HI_FORMAT_INVOKE_PRE_CLOSE_FILE). 1: The current operation is forcibly stopped. 0: The current operation is not forcibly stopped.#
EmptyCount:0	Count of times that the player checks whether data in the buffer is played completely after the end of file is reached#
PlayProgress:71(0-100)	Current playback progress, in percentage
DownloadProgress:0(0-100)	Current downloading progress, in percentage
LastPlayDuration:1773294(ms)	This parameter can be ignored.
FirstPts:0(ms)	PTS of the first frame that is transmitted to the AVPLAY
LastPts:1784242(ms)	PTS of the last frame that is transmitted to the AVPLAY
LastIFramePts:1783783(ms)	PTS of the last I frame that is transmitted to the AVPLAY
LastSeekPts:1773294(ms)	This field is reserved.
CallSeekPts:-1(ms)	Whether to perform the seek operation. If this parameter is not **–1**, the HiPlayer performs the seek operation. Values greater than or equal to 0 are valid.#
FirstPtsAfterSeek:0(ms)	This field is reserved.

Parameter	Description
CallSeekPos:-1(byte)	This field is reserved.
LastVidPos:3763485855(byte)	Byte offset of the last frame transmitted to the AVPLAY
SubtitleCodecType:0	This field is reserved.
LogLevel	Log level. Application configures the log level by calling the HiPlayer interface HI_FORMAT_INVOKE_SET_LOG_LEVEL. The log level configured by application is valid only when **echo SetLogLevel = -1|0|1|2|3|4|5|6 >/proc/hisi/hiplayer/common** is set to **−1**.
TotalSendIFrame:1783	Total number of transmitted I frames. This parameter is set to **0** when the fast-forward operation is performed after the rewind operation or the rewind operation is performed after the fast-forward operation.
SystimeOfSendLastIFrame:166310397(ms)	Time when the last I frame is transmitted. It is obtained by calling gettimeofday.
TotalIFrameScale:0(ms)	Total interval between transmitted I frame data, that is, total differences between adjacent PTSs
BaseSystimeOfTPlay:0(ms)	Time point for starting the Tplay operation. It is obtained by calling gettimeofday.
BaseIFramePtsOfTPlay:0(ms)	PTS of the first I frame transmitted by Tplay
NumOfDiscardIFrame:0	Number of I frames discarded by Tplay. Frames are discarded when the display time is reached (when the system time minus the Tplay start time is greater than the frame PTS).
NumOfSendPBFrame:0	Total number of transmitted PB frames
StatusOfBuffer:INVALID	Status of the buffer for playback over the network
DurationOfFormatBuffer:0(ms)	Duration of data left in the parser buffer#
SizeOfFormatBuffer:0(bytes)	Number of data bytes left in the parser buffer
DurationOfPlayerBuffer:0(ms)	Duration of data left in the player buffer
SizeOfPlayerBuffer:0(bytes)	Number of data bytes left in the player buffer
DownloadFinish:0	0: File downloading is incomplete. 1: File downloading is complete.

Parameter	Description
ResetPlayer:0	0: Buffer data is not deleted (this may occur in a live play scenario). 1: Buffer data is deleted.
ReportFirstFrameTime:1	This parameter can be ignored.
SetBufferUnderrun:0	Whether the player is paused automatically when buffer data is insufficient 1: The player is paused automatically when buffer data is insufficient, and is resumed when data is sufficient. 0: The player is not paused automatically.
UnderrunCount	Number of unloads for the player. The value is cleared after the player starts to replay.
CurStreamIdx:0	ID of the current stream (AUDIO/VIDEO/SUBTITLE)
CurFrameType:AUD	Type of the current frame
CurFrameSize:4096(bytes)	Number of bytes of the current frame
CurFramePts:1784746(ms)	PTS of the current frame
SendAudio:0	0: Audio data is not transmitted. 1: Audio data is transmitted.
SendVideo:0	0: Video data is not transmitted. 1: Video data is transmitted.
SendSubtitle:0	0: Subtitle data is not transmitted. 1: Subtitle data is transmitted.
DumpAVEsData:1	1: Audio and video ES data is stored. 0: Audio and video ES data is not stored.
PrintFrameInfo:0	Whether to display frame information 1: The audio, video, and subtitle frame information is displayed. 0: The audio, video, and subtitle frame information is not displayed.
ReadFrameFromDemux(Try/OK) :128478/128478	Number of attempts to read data from the file parser/Number of times that data is successfully read from the file parser If the **Try** value increases continuously but the **OK** value does not increase, parsing is abnormal. You can check whether data is read properly if intermittence occurs.
SendVidToAVPlay(Try/OK):65183/42781	Number of times that the HiPlayer transmits video data to the AVPLAY/Number of

Parameter	Description
	times that video data is successfully transmitted to the AVPLAY If video intermittence occurs, check whether the **Try** value increases continuously but the **OK** value does not increase. If yes, video data fails to be transmitted.
SendAudToAVPlay(Try/OK):206776/85696	Number of times that the HiPlayer transmits audio data to the AVPLAY/Number of times that audio data is successfully transmitted to the AVPLAY If audio intermittence occurs, check whether the **Try** value increases continuously but the **OK** value does not increase. If yes, audio data fails to be transmitted.
SendSubToSo(Try/OK):0/0	Number of times that the HiPlayer transmits subtitle data to the AVPLAY/Number of times that subtitle data is successfully transmitted to the AVPLAY If audio or video intermittence occurs, check whether the **Try** value increases continuously but the **OK** value does not increase. If yes, subtitle data fails to be transmitted, which results in failures in transmitting audio and video data in time.

2.33.4 Checking Parser Information

Run **cat /proc/hisi/hiplayer00/format** over the serial port.

[Debugging Information]

```
cat /proc/hisi/hiplayer00/format
********Hisilicon HiPlayer00 Format Info Begin*************
_________HiPlayer Format Demux Info Begin_____________
FileUrl                 :/mnt/[SD]_AVC_478Kbps_848x480.mov
DemuxName                : libffmpegformat.so
SupportFormatName       :
SupportProtocolName      :
Merit                    :0
PreCloseFile            :0
GetMsgFin                :1
_________HiPlayer Format Demux Info End_______________
_________HiPlayer Format Parameter Info Begin_________
BufferConfigType         :SIZE
MaxBufferSize             :10485760(bytes)
TimeOutOfNetwork         :120000(ms)
```

```
BufferStart              :524288(size-bytes/time-ms)
BufferEnough             :4194304(size-bytes/time-ms)
BufferFull               :5242880(size-bytes/time-ms)
HlsStartMode             :NORMAL
DstSubCodeType            :UTF-8
Cookie                   :NotSet
UserAgent                 :NotSet
Referer                  :NotSet
________HiPlayer Format Parameter Info End___________
*******Hisilicon HiPlayer00 Format Info End*************
```

[Analysis]

Information about the current parser can be viewed.

[Parameter Description]

Parameter	Description
FileUrl:/mnt/[SD]_AVC_478Kbps_848x480.mov	URL of the played file
DemuxName:libffmpeg.so	Name of the used file parser
SupportFormatName:	File formats supported by the parser
SupportProtocolName:	Protocols supported by the parser
Merit:0	Parser priority. The default value is 0. The value ranges from 0x0 to 0xFFFFFFFF.#
PreCloseFile:0	This field is reserved.
GetMsgFin:1	This field is reserved.
BufferConfigType:SIZE	Buffer configuration type SIZE: by byte TIME: by time
MaxBufferSize:10485760(bytes)	Maximum buffer size, in byte
TimeOutOfNetwork:120000(ms)	Network timeout period
BufferStart:524288(size-bytes/time-ms)	Threshold indicating the buffer is insufficient. The BufferStart event is reported if data amount in the buffer is less than this value.#
BufferEnough:4194304(size-bytes/time-ms)	Threshold indicating the buffer is sufficient. The BufferEnough event is reported if data amount in the buffer is greater than this value.#
BufferFull:5242880(size-bytes/time-ms)	Threshold indicating the buffer is full. The BufferFull event is reported if data amount

Parameter	Description
	in the buffer reaches this value.#
HlsStartMode:NORMAL	HTTP live streaming (HLS) boot mode NORMAL: The playback starts from the highest bit rate by default. FAST: The playback starts from the lowest bit rate.
DstSubCodeType:UTF-8	Destination subtitle encoding type
Cookie:NotSet	Cookie information
UserAgent: NotSet	UserAgent information
Referer: NotSet	Referrer information

2.33.5 Viewing SDK Adaptation Information

Run **cat /proc/hisi/hiplayer00/adapter** over the serial port.

[Debugging Information]

```
cat /proc/hisi/hiplayer00/adapter
*****Hisilicon HiPlayer00 Player Adapter Info Begin******
AdapterStatus           :PLAY
bExtraAVPlay            :1
AVPlayHandle            :0x410000
AudTrackHandle          :0xffffffff
WindowHandle            :0x0
SOHandle                :0x5edfd020
AudMixHeight            :100
SoundPort               :0
Display                 :0
Window-X                :0(pix)
Window-Y                :0(pix)
Window-w                :0(pix)
Window-H                :0(pix)
VDecErrCover             :0
VidPtsOffset             :0(ms)
AudPtsOffset             :0(ms)
SubPtsOffset             :0(ms)
UseFFmpegAudCodec        :1
PlaySpeed               :0(x)
bVidStart               :1
bAudStart               :1
CurProgramId             :0
```

```
CurVideoId              :0
CurAudioId              :0
CurSubtitleId           :0
AudFrameSendLen         :0(bytes)
LastPgsSubPts           :0(ms)
EndOfStream             :0
*****Hisilicon HiPlayer00 Player Adapter Info Begin*****
```

[Analysis]

Information about the underlying playback module used by the HiPlayer can be viewed.

[Parameter Description]

Parameter	Description
AdapterStatus:PLAY	Current AVPLAY state
bExtraAVPlay:1	Whether the AVPLAY is externally created 1: The AVPLAY is created externally and transmitted to the HiPlayer for use. 0: The AVPLAY is created by the HiPlayer.
AVPlayHandle:0x410000	AVPLAY handle#
AudTrackHandle:0xffffffff	Audio track handle#
WindowHandle:0x0	Video window handle
SOHandle:0x5edfd020	SO handle
AudMixHeight:100	MixHeight value
SoundPort:0	Audio port
Display:0	Display ID
Window-X:0(pix)	X coordinate of the position of the video window
Window-Y:0(pix)	Y coordinate of the position of the video window
Window-W:0(pix)	Width of the video window
Window-H:0(pix)	Height of the video window Note that if the X, Y, W, and H of a window are all 0s, the full-screen mode is used.
VdecErrCover:0	Video output error threshold
VidPtsOffset:0(ms)	Video PTS offset
AudPtsOffset:0(ms)	Audio PTS offset
SubPtsOffset:0(ms)	Subtitle PTS offset
bUseFFmpegAudCodec:1	Whether the FFMPEG software decoding is

Parameter	Description
	used
PlaySpeed:0(x)	Playback speed (2/4/8/16/32)#
bVidStart:1	1: The video channel is started properly. 0: The video channel fails to be started, and videos cannot be played.
bAudStart:1	1: The audio channel is started properly. 0: The audio channel fails to be started, and audio cannot be played.
CurProgramId:0	ID of the current program
CurVideoId:0	ID of the current video
CurAudioId:0	ID of the current audio
CurSubtitleId:0	ID of the current subtitle
AudFrameSendLen:0(bytes)	Number of bytes of the current audio and video frames that have been transmitted to the AVPLAY
LastPgsSubPts:0(ms)	This parameter can be ignored.
EndOfStream:0	Whether the AVPLAY has reported the EndOfStream event 1: yes 0: no

2.33.6 Viewing the HiPlayer Static Proc Information

The proc information is stored under the **/proc/hisi/hiplayer/common** directory. This directory exists only in the following situations:

- The HiPlayer is in the playback process.
- The HiPlayer stops playing but the player process is not exited. When the player process is exited, the **common** directory will also be destroyed.

The **common** directory is created only after the HiPlayer is started for at least once.

Check the proc information by running the following commands:

```
cat /proc/hisi/hiplayer/common
Current debug log level: -1
echo SetLogLevel = -1|0|1|2|3|4|5|6 >/proc/hisi/hiplayer/common
Log level (-1|0|1|2|3|4|5|6 = Disable
this|QUIET|FATAL|ERROR|WARNING|INFO|DEBUG|VERBOSE)
echo hiplayerstreampath = dir >/proc/hisi/hiplayer/common
echo enabledumpstream = 1|0 >/proc/hisi/hiplayer/common
```

[Parameter Description]

Parameter	Description
Current debug log level : -1	Current log level
echo SetLogLevel = -1|0|1|2|3|4|5|6 >/proc/hisi/hiplayer/common	When **SetLogLevel** is set to **−1**, the log level set by application by calling the HiPlayer interface HI_FORMAT_INVOKE_SET_LOG_LEVEL is valid. The values 0 to 5 correspond to QUIET, FATAL, ERROR, WARNING, INFO, DEBUG, and VERBOSE respectively. If **SetLogLevel** is set to a value ranging from 0 to 5, then the log level set by application by calling HI_FORMAT_INVOKE_SET_LOG_LEVEL is invalid.
echo hiplayerstreampath = /data/mediadump >/proc/hisi/hiplayer/common	Sets the stream recording path. **dir** is the directory. The default directory is **/data/mediadump**. This directory is also valid for the **echo DumpAVEs = 1|0 >/proc/hisi/hiplayer00/control** command.
echo enabledumpstream = 1|0 >/proc/hisi/hiplayer/common	Starts/Stops recording the played file streams. The recorded streams are stored under the path configured by running the **hiplayerstreampath** command.

2.33.7 Viewing the Program Switching Performance Statistics Information

Run **cat/proc/hisi/hiplayer00/switch-pg** over the serial port after the program is played.

[Debugging Information]

```
cat /proc/hisi/hiplayer00/switch-pg
************************Hisilicon HiPlayer00 Out Info
Begin**************************
_____________________SWITCH-PG Info Begin Time(ms)_______________
Demux Name: mov,mp4,m4a,3gp
Protocol Name: md
Set DataSource: 4249079355
UNF_AVPLAY_Create: 4249079442
Do Prepare: 4249079371
SVR_Format_FindStream: 4249079449
Do avformat_open_input : 4249079449
Do init_input: 4249079449
Do avio_open_h: 4249079449
avio_open_h Done: 4249079451
av_probe_input_buffer: 4249079451
probe_input_buffer Done: 4249079593
```

```
open_input done: 4249080161
estimate duration start: 4249080162
estimate duration done : 4249080162
FindStream Done: 4249080165
SVR_FORMAT_GetFileInfo : 4249080161
GetFileInfo Done: 4249080163
Prepare async Complete : 4249080167
Do Start Enter: 4249080289
PLAYER_STATE change: 4249080316
First VFrame Sent: 4249080319
First VFrame Decoded: 4249080399
FIRST_FRAME_TIME: 4249080399
---------------In Total---------------------
ExitPlay Consume:0
Enter Play Consume:1044
---------------Details-----------------------
Prepare Schedule:16
init_input:144
iformat->read_header:568
Find Stream Info:2
StartSchedule:122
Enter Play UNF Consume :27
Send1st VFrame:3
Decode1st VFrame:80
---------------Supplement--------------------
Exit Play UNF Consume:0
Bytes of 1st VFrame:3081
Send1st AFrame:8
FF Read 1st VFrame:25
FF Read 1st VKFrame:26
FF Read 1st AFrame:27
1st VFramePts:33
1st VKeyFrame Pts:33
Download Bandwidth:89034417
_____________________________SWITCH-PG Info End_________________
```

[Analysis]

After the HiPlayer enters the playback status and before any operation is performed, run the commands to check the program switching performance statistics information. If one operation is not performed during the playback status, the corresponding parameters will not be displayed.

[Parameter Description]

Parameter	Description
Demux Name: mov, mp4, m4a, 3gp	Name of the parser (more specific compared with the parser concept described in section 2.33.4 Checking Parser Information"2.33.4 Checking Parser Information"). The file type is visible.
Protocol Name: md	Name for reading the data protocol, such as http, file, and so on
Set DataSource: 4249079355	Absolute time when HiMediaPlayer::setdatasource() is called, in ms, valid only in the Android version
UNF_AVPLAY_Create: 4249079442	Absolute time when the AVPLAY starts to be created, in ms, valid only in the Android version
Do Prepare: 4249079371	Absolute time when HiMediaPlayer::prepare() is called, in ms, valid only in the Android version
SVR_Format_FindStream: 4249079449	Absolute time when the HiPlayer internal interface SVR_Format_FindStream() is called, in ms
Do avformat_open_input: 4249079449	Absolute time when the internal interface avformat_open_input() is called, in ms
Do init_input: 4249079449	Absolute time when the internal interface init_input() is called, in ms
Do avio_open_h: 4249079449	Absolute time when the internal interface avio_open_h() is called, in ms
avio_open_h Done: 4249079451	Absolute time when the internal interface avio_open_h() is returned, in ms
av_probe_input_buffer: 4249079451	Absolute time when the internal interface av_probe_input_buffer () is called, in ms
probe_input_buffer Done: 4249079593	Absolute time when the internal interface av_probe_input_buffer () is returned, in ms
open_input done: 4249080161	Absolute time when the internal interface avformat_open_input() is returned, in ms
estimate duration start: 4249080162	Absolute time when the counting of the file duration starts, in ms
estimate duration done: 4249080162	Absolute time when the counting of the file duration completes, in ms
FindStream Done: 4249080165	Absolute time when the file information parsing interface is returned, in ms
SVR_FORMAT_GetFileInfo: 4249080161	Absolute time when the obtaining of file information starts, in ms

Parameter	Description
GetFileInfo Done: 4249080163	Absolute time when the obtaining of file information completes, in ms
Prepare async Complete: 4249080167	Absolute time when the HiPlayer completes file parsing and is ready to enter the playback status, in ms
Do Start Enter: 4249080289	Android version: absolute time when the upper-layer application calls the start() interface, in ms Linux version: absolute time when the upper-layer application calls HiPlayer HI_SVR_PLAYER_Play(), in ms
PLAYER_STATE change: 4249080316	Absolute time when the HiPlayer reports a playback status event, in ms, valid only in the Android version
First VFrame Sent: 4249080319	Absolute time when the first video frame is transmitted to the AVPLAY after the playback starts, in ms
First VFrame Decoded: 4249080399	Absolute time when the decoding of the first video frame completes after the playback starts, in ms
FIRST_FRAME_TIME: 4249080399	Absolute time when the HiMediaPlayer receives the first video frame decoded event reported by the HiPlayer, in ms, valid only in the Android version
ExitPlay Consume: 0	Total time consumed in exiting the last playback, in ms. This parameter is not supported.
Enter Play Consume: 1044	Total time consumed in entering the playback, in ms
Prepare Schedule: 16	Time consumed from the time of calling HiMediaPlayer::Setdatasource() to the time of calling HiMediaPlayer::prepare(), in ms, valid only in the Android version
init_input: 144	Time consumed in calling the internal interface init_input(), in ms
iformat->read_header: 568	Time consumed in calling the internal interface read_header(), in ms
Find Stream Info: 2	Time consumed in calling the internal interface findstream(), in ms
StartSchedule: 122	Time consumed from the time the HiPlayer completes file parsing to the time the upper-layer application calls HiPlayer HI_SVR_PLAYER_Play(), in ms

Parameter	Description
Enter Play UNF Consume: 27	Total time consumed by the AVPLAY to enter the playback status (excluding the decoding time), in ms
Send1st VFrame: 3	Time consumed from the time the HiPlayer enters the playback status to the time it completes the transmission of the first video frame to the AVPLAY, in ms
Decode1st VFrame: 80	Time consumed in completing the decoding of the first video frame, in ms
Exit Play UNF Consume: 0	Total time consumed by the AVPLAY in exiting the last playback, in ms. This parameter is not supported currently.
Bytes of 1st VFrame: 3081	Size of the first video key frame, in byte
Send1st AFrame: 8	Time consumed from the time the HiPlayer enters the playback status to the time it completes the transmission of the first audio frame to the AVPLAY, in ms
FF Read 1st VFrame: 25	Time consumed from the time the internal interface findstream() is returned to the time the first video frame is read, in ms
FF Read 1st VKFrame: 26	Time consumed from the time the internal interface findstream() is returned to the time the first video key frame is read, in ms
FF Read 1st AFrame: 27	Time consumed from the time the internal interface findstream() is returned to the time the first audio frame is read, in ms
1st VFramePts: 33	PTS of the first video frame, in ms
1st VKeyFrame Pts: 33	PTS of the first video key frame, in ms. When the first video frame is the key frame, then the values of **1st VFrame Pts** and **1st VKeyFrame Pts** are the same.
Download Bandwidth: 89034417	Download speed, in bit/s, valid only in network playback

2.33.8 Viewing the Seek Performance Statistics Information

Run **cat/proc/hisi/hiplayer00/seek** over the serial port after the seek operation.

[Debugging Information]

```
cat /proc/hisi/hiplayer00/seek
***********************Hisilicon HiPlayer00 Out Info
Begin*************************
```

```
________________________Seek Info Begin Time(ms)_______________
SeekTo: 4251885281
Do async seek: 4251885281
Do seek pts: 4251885284
PAdpt Seek Complete: 4251885351
Do AVPLAY Reset: 4251885284
Do AVPLAY ResetDone: 4251885329
Do SeekVidBefore: 4251885351
HI_SVR_FORMAT_SeekPts: 4251885351
Do CMD_SEEK: 4251885356
Do Read_Seek: 4251885566
Read_Seek Done: 4251885567
Thread_SeekPts 1 Done: 4251885394
Thread_AVSyncDone: 4251885566
Thread_SeekPts 2 Done: 4251885568
Do seek complete: 4251885580
First VFrame Sent: 4251885591
First VFrame Decoded: 4251885642
---------------In Total---------------------
Seek Consume: 361
---------------Details-----------------------
SeekSchedule 1: 3
PAdptSeek: 67
SeekSchedule 2: 0
FF Seek Schedule: 5
Thread_SeekPts 1: 38
Thread_AVSync: 172
Thread_SeekPts 2: 2
Complete Seek: 12
Send1st VFrame: 11
Decode1st VFrame: 51
---------------Supplement--------------------
Seek UNF Consume: 45
Bytes of 1st VFrame: 71745
Send1st AFrame: 13
FF Read 1st VFrame: 1
FF Read 1st VKFrame: 1
FF Read 1st AFrame: 2
1st VFramePts: 408642
1st VKeyFrame Pts: 408642
FF Read_Seek: 1
ff_seek_frame_binary: 0
seek_frame_generic: 0
Download Bandwidth: 94131622
```

```
_____________________________Seek Info End_________________
```

[Analysis]

After the seek operation is performed and before any other operation is performed, run the commands to check the seek performance statistics information. If one operation is not performed during the seek process, the corresponding parameters will not be displayed.

[Parameter Description]

Parameter	Description
SeekTo: 4251885281	Absolute time when HiMediaPlayer::seekto() is called, in ms, valid only for the Android version
Do async seek: 4251885281	Absolute time when HiPlayer HI_SVR_PLAYER_Seek() is called, in ms
Do seek pts: 4251885284	Absolute time when the HiPlayer starts to execute the seek operation, in ms
PAdpt Seek Complete: 4251885351	Absolute time when the AVPLAY completes the seek operation, in ms
Do AVPLAY Reset: 4251885284	Absolute time when the AVPLAY reset operation starts, in ms
Do AVPLAY ResetDone: 4251885329	Absolute time when the AVPLAY reset operation completes, in ms
Do SeekVidBefore: 4251885351	Absolute time when the video stream seek operation starts, in ms
HI_SVR_FORMAT_SeekPts: 4251885351	Absolute time when the internal interface HI_SVR_FORMAT_SeekPts() is called, in ms
Do CMD_SEEK: 4251885356	Absolute time when the FFMPEG receives the seek command, in ms, valid only in network playback
Do Read_Seek: 4251885566	Absolute time when the calling of the parser seek interface starts, in ms
Read_Seek Done: 4251885567	Absolute time when the calling of the parser seek interface completes, in ms
ff_seek_frame_binary: 4251885567	Absolute time when the calling of the internal interface ff_seek_frame_binary() starts, in ms
seek_frame_binary Done: 4251885567	Absolute time when the calling of the internal interface ff_seek_frame_binary() completes, in ms
Do seek_frame_generic: 4251885567	Absolute time when the calling of the internal interface seek_frame_generic starts, in ms

Parameter	Description
seek_frame_generic Done: 4251885567	Absolute time when the calling of the internal interface seek_frame_generic completes, in ms
Thread_SeekPts 1 Done: 4251885394	Absolute time when the internal interface Thread_SeekPts is returned after the first calling, in ms, valid only in network playback
Thread_AVSyncDone: 4251885566	Absolute time when the internal interface Thread_AVSync() is returned, in ms, valid only in network playback
Thread_SeekPts 2 Done: 4251885568	Absolute time when the internal interface Thread_SeekPts is returned after the second calling, in ms, valid only in network playback
Do seek complete: 4251885580	Absolute time when the HiPlayer completes the seek operation, in ms (the first video frame is not displayed at this time)
First VFrame Sent: 4251885591	Absolute time when the transmission of the first video frame to the AVPLAY completes, in ms
First VFrame Decoded: 4251885642	Absolute time when the decoding of the first video frame completes, in ms
Seek Consume: 361	Total time consumed by the HiPlayer during the seek operation, in ms
SeekSchedule 1: 3	Android version: time consumed from the time HiMediaPlayer::seekto() is called to the time the HiPlayer starts to execute the seek operation, in ms Linux version: time consumed from the time HiPlayer HI_SVR_PLAYER_Seek() is called to the time the HiPlayer starts to execute the seek operation, in ms
PAdptSeek: 67	Time consumed by calling the HiPlayer internal interface to execute the AVPLAY seek operation, in ms
SeekSchedule 2: 0	Time consumed from the time the AVPLAY seek operation completes to the time the calling of the FFMPEG SEEK interface starts, in ms
FF Seek Schedule: 5	Time consumed by the FFMPEG to transmit and receive the seek message, in ms, valid only in network playback
Thread_SeekPts 1: 38	Time consumed in calling the internal interface Thread_SeekPts to execute the

Parameter	Description
	seek operation for the first time, in ms, valid only in network playback
Thread_AVSync : 172	Time consumed in calling the internal interface Thread_AVSync() to execute the audio/video synchronization operation, in ms, valid only for some parsers in network playback
Thread_SeekPts 2: 2	Time consumed in calling the internal interface Thread_SeekPts to execute the seek operation for the second time, in ms, valid only in network playback
Thread_SeekPts: 2	Time consumed in calling the internal interface Thread_SeekPts, in ms, valid only in local playback
Complete Seek: 12	Time consumed from the time the internal interface Thread_SeekPts is called for the second time to the time the HiPlayer seek operation completes, in ms, valid only in network playback
Send1st VFrame: 11	Time consumed from the time the HiPlayer completes the seek operation to the time it completes the transmission of the first video frame to the AVPLAY, in ms
Decode1st VFrame: 51	Time consumed in completing the decoding of the first video frame, in ms
Seek UNF Consume: 45	Total time consumed by the AVPLAY during the seek operation, in ms
Bytes of 1st VFrame: 71745	Size of the first video frame after the seek operation, in byte
Send1st AFrame: 13	Time consumed from the time the HiPlayer completes the seek operation to the time it completes the transmission of the first audio frame to the AVPLAY, in ms
FF Read 1st VFrame: 1	Time consumed from the time the HiPlayer completes the seek operation to the time the FFMPEG reads the first video frame, in ms
FF Read 1st VKFrame: 1	Time consumed from the time the HiPlayer completes the seek operation to the time the FFMPEG reads the first video key frame, in ms
FF Read 1st AFrame: 2	Time consumed from the time the HiPlayer completes the seek operation to the time the FFMPEG reads the first audio frame, in ms

Parameter	Description
1st VFramePts: 408642	PTS of the first video frame after the seek operation, in ms
1st VKeyFrame Pts: 408642	PTS of the first video key frame after the seek operation, in ms. When the first video frame is the key frame, then the values of **1st VFrame Pts** and **1st VKeyFrame Pts** are the same.
FF Read_Seek: 1	Time consumed by the FFMPEG parser to execute the seek operation, in ms
ff_seek_frame_binary: 0	Time consumed in calling the internal interface ff_seek_frame_binary(), in ms
seek_frame_generic: 0	Time consumed in calling the internal interface seek_frame_generic(), in ms
Download Bandwidth: 94131622	Download speed, in bit/s, valid only in network playback

3 Fault Identification and Debugging

3.1 Demux

3.1.1 Debugging Commands

The Demux module provides the following debugging command to modify the default directory for storing files:

```
echo storepath=path > /proc/msp/log
For example:
echo storepath =/home > /proc/msp/log
```

Saving ESs

To start saving ES data, run the following command:

echo save es start   > /proc/msp/demux_main

- Audio data is saved in dmx_aud_x.es (*x* starts from 0 and is incremented by 1 each time audio data is saved).
- Video data is saved in dmx_vid_x.es (*x* starts from 0 and is incremented by 1 each time video data is saved).

To stop saving ES data, run the following command:

echo save es stop   > /proc/msp/demux_main

You can check whether the playback errors are caused by the Demux or front-end devices by saving ES files.

For example, when mosaics appear in the video or intermittence occurs in the audio, you can save the ESs using the preceding commands and play the saved file using **sample/sample_esplay** or the ES play tool (such as elecard) in the PC.

- If mosaic or intermittence still occurs when the saved ES files are played, the problem is not caused by the AVPLAY, VOU, and audio modules.
- If the saved ESs are played properly or different issues occur, the AVPLAY, VOU, and audio module processing is faulty. In this case, you need to locate the problem further.

Recording All TSs

To start recording all TSs, run the following command:

```
echo save allts start PortId > /proc/msp/demux_main
```

PortId indicates the port ID. The IF ports are numbered from 0, the TSI ports are numbered from 32, and the RAM ports are numbered from 128.

TSs are saved in **dmx_allts_x.ts**, and x indicates PortId.

To stop recording all TSs, run the following command:

```
echo save allts stop   > /proc/msp/demux_main
```

In this scenario, the Demux records all TS packets from a specific port. You can check whether the signal source is abnormal by recording all TSs. For example, if section packet loss occurs, you can record all TSs and then check whether packet loss still occurs by analyzing the streams or reading the saved files in the local. If yes, packet loss occurs before the TSs are received by the Demux, and you need to identify whether the problem is caused by the QAM/tuner or the route.

CAUTION

- Confirm the port that is used before calling the command to record all TSs. Port 0 is used if the signals are transmitted over the embedded QAM.
- Data amount is large when all TSs are recorded. If the network file system is used to record all TSs, ensure that the network bandwidth meets the requirement (higher than 5 MB/s). If the network speed is slow, you are advised to save the recorded files in removable hard disks.

Saving Data Input from the RAM Port

To start saving data input from the RAM port, run the following command:

```
echo save ipts start   > /proc/msp/demux_main
```

Data input from the RAM port is saved in **dmx_ram_*x*.ts** (*x* starts from 0 and is incremented by 1 each time the data is saved).

To stop saving data input from the RAM port, run the following command:

```
echo save ipts stop   > /proc/msp/demux_main
```

When streams are recorded over the network port, data saved by running the preceding command is the data sent by users and is not processed. You can check whether the problem is caused by the network environment or network RX devices (for example, a socket) using this method.

Recording TSs from All Demux Channels

To start recording TSs from all DEXMUX channels, run the following command:

```
echo save dmxts start   DmxId > /proc/msp/demux_main
```

TSs are saved in **dmx_rects_*x*.ts** (*x* starts from 0 and is incremented by 1 each time TSs are saved).

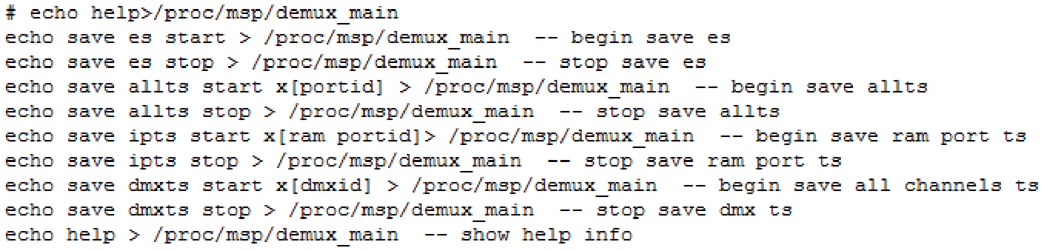
To stop recording TSs from all DEXMUX channels, run the following command:

echo save dmxts stop > /proc/msp/demux_main

This command is used to locate problems related to encrypted streams. If the streams are encrypted, running this command records data in the channels mounted to the Demux as unscrambled streams.

Viewing the Help Information

To view the help information, run the following command:



```
# echo help>/proc/msp/demux_main
echo save es start > /proc/msp/demux_main  -- begin save es
echo save es stop > /proc/msp/demux_main  -- stop save es
echo save allts start x[portid] > /proc/msp/demux_main  -- begin save allts
echo save allts stop > /proc/msp/demux_main  -- stop save allts
echo save ipts start x[ram portid]> /proc/msp/demux_main  -- begin save ram port ts
echo save ipts stop > /proc/msp/demux_main  -- stop save ram port ts
echo save dmxts start x[dmxid] > /proc/msp/demux_main  -- begin save all channels ts
echo save dmxts stop > /proc/msp/demux_main  -- stop save dmx ts
echo help > /proc/msp/demux_main  -- show help info
```

3.1.2 What Do I Do If Data Cannot Be Received?

If data cannot be received, check the following items:

- Whether there is data transmitted over the port
- Whether the Demux is correctly configured
- Whether data exists in the channel (If there is data in the channel, check whether the applications obtain the data.)
- Whether channel buffers are properly released
- Whether corresponding data exists in the data source

Checking Port Data

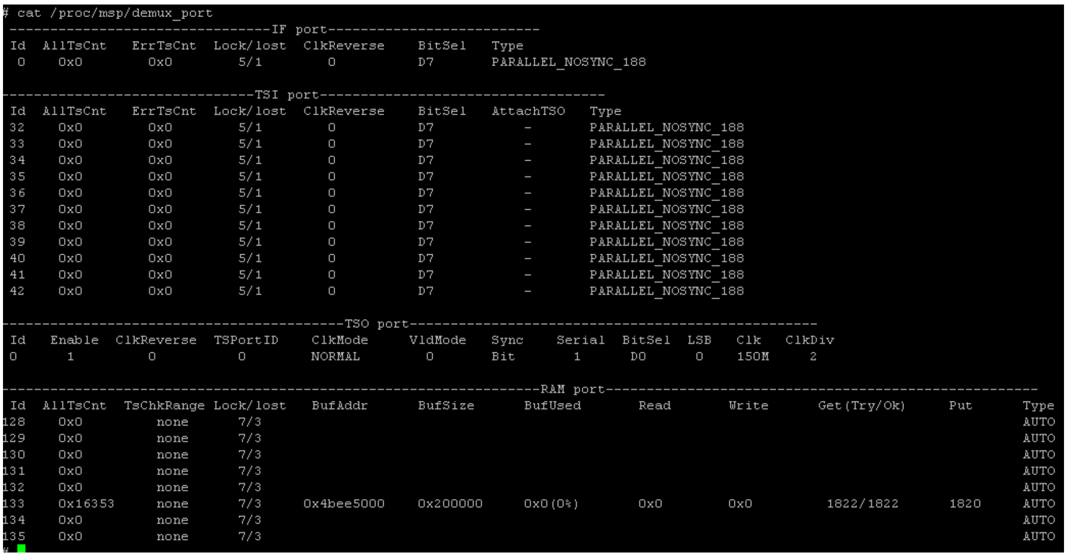
Find the port corresponding to the channel first. Typically, if the embedded QAM is used, port 0 is used. In other cases, you can check the ID of the DEUMX corresponding to the channel by running **cat /proc/msp/demux_chan** and check the ID of the port corresponding to the Demux by running **cat /proc/msp/demux_main**. After obtaining the port ID, check whether the TS packets transferred over the port increase by running **cat /proc/msp/demux_port** repeatedly to determine whether data is transmitted over the port. For example:

```
# cat /proc/msp/demux_chan
DmxId  ChnId  PID      Type Mod Stat  KeyId   Acquire(Try/Ok)   Release
  8      0    0x1022   VID  PLY OPEN --      1977/1461         1461
  8      1    0x1023   AUD  PLY OPEN --       199/195          195
```

The preceding Proc information shows that channel 0 and channel 1 correspond to Demux 8.

```
# cat /proc/msp/demux_main
DmxId PortId
  0    --
  1    --
  2    --
  3    --
  4    --
  5    --
  6    --
  7    --
  8    133
  9    --
  10   --
  11   --
  12   --
  13   --
type "echo help > /proc/msp/demux_main" to get help information
```

The preceding Proc information shows that Demux 8 corresponds to port 13.



```
# cat /proc/msp/demux_port
 --------------------------------IF port--------------------------
 Id  AllTsCnt   ErrTsCnt  Lock/lost  ClkReverse    BitSel   Type
  0    0x0        0x0        5/1        0          D7       PARALLEL_NOSYNC_188

-------------------------------TSI port-----------------------------------
 Id  AllTsCnt   ErrTsCnt  Lock/lost  ClkReverse    BitSel   AttachTSO   Type
 32    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 33    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 34    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 35    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 36    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 37    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 38    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 39    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 40    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 41    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188
 42    0x0        0x0        5/1        0          D7           -       PARALLEL_NOSYNC_188

-----------------------------------------TSO port---------------------------------------------------
 Id    Enable  ClkReverse  TSPortID     ClkMode     VldMode    Sync    Serial  BitSel  LSB   Clk   ClkDiv
 0       1         0          0         NORMAL         0       Bit       1       D0     0    150M     2

-----------------------------------------------------------------RAM port------------------------------------------------------
 Id  AllTsCnt  TsChkRange Lock/lost   BufAddr      BufSize      BufUsed       Read       Write      Get(Try/Ok)     Put      Type
128    0x0        none      7/3                                                                                             AUTO
129    0x0        none      7/3                                                                                             AUTO
130    0x0        none      7/3                                                                                             AUTO
131    0x0        none      7/3                                                                                             AUTO
132    0x0        none      7/3                                                                                             AUTO
133    0x16353    none      7/3     0x4bee5000   0x200000     0x0(0%)       0x0        0x0         1822/1822      1820     AUTO
134    0x0        none      7/3                                                                                             AUTO
135    0x0        none      7/3                                                                                             AUTO
#
```

Check the demux_port information repeatedly and compare the obtained information to see whether the value of **AllTsCnt** increases.

- If the value of AllTsCnt does not increase, there is no data transferred over the port.

  In this case, find the reason why no data is transferred over the port. For example, for the IF/TSI port, check whether the tuner is locked. For the RAM port, check whether applications send data to the corresponding port.

- If the value of AllTsCnt increases, there is data transferred over the port.

  In this case, following the instructions in the next section.

Checking the Demux Configuration

Check the Demux channel configuration information by running **cat /proc/msp/demux_chan**. For example:

```
# cat /proc/msp/demux_chan
```

```
DmxId  ChnId  PID      Type Mod Stat  KeyId   Acquire(Try/Ok)   Release
  8     0   0x1022   VID  PLY OPEN --      1977/1461        1461
  8     1   0x1023   AUD  PLY OPEN --       199/195         195
```

The preceding information indicates that channel 0 is a video channel, its PID is 0x1022, its current status is OPEN, and it is not bound to any key area. If the channel type, PID, or status is not as expected, find the cause of errors.

Checking the Channel Buffer

Check whether there is data in the Demux channel by running **cat /proc/msp/demux_chanbuf**. If **buffer used** is **0** and the read and write pointers obtained by repeatedly running **cat /proc/msp/demux_chanbuf** do not change, there is no data in the channel. Otherwise, there is data in the channel.

```
# cat /proc/msp/demux_chanbuf
DmxId  ChnId  FqId  OqId  Size  BlkCnt  BlkSize  Read  Write  Used  Overflow
  8    0     1    0 16383K 1008   16644    455    453    99%    0
  8    1     2    1 255K 204   1280    195    193    99%    0
```

The preceding information shows that channel 0 and channel 1 both have data. In this case, check whether the applications obtain data by checking the Proc information about the channel repeatedly. If the **Acquire Try** value does not increase, the applications do not obtain data. In this case, check the applications. See the following Proc information:

```
# cat /proc/msp/demux_chan
DmxId  ChnId  PID      Type Mod Stat  KeyId   Acquire(Try/Ok)   Release
  8     0   0x1022   VID  PLY OPEN --      1977/1461        1461
  8     1   0x1023   AUD  PLY OPEN --       199/195         195
# cat /proc/msp/demux_chan
DmxId  ChnId  PID      Type Mod Stat  KeyId   Acquire(Try/Ok)   Release
  8     0   0x1022   VID  PLY OPEN --     33844/1557        1557
  8     1   0x1023   AUD  PLY OPEN --       633/615         614
```

The preceding Proc information shows that the **Acquire Try** value of channel 0 increases from 1977 to 33844, indicating that applications are obtaining data.

Checking Whether Channel Buffers Are Properly Released

HI_UNF_DMX_AcquireBuf and HI_UNF_DMX_ReleaseBuf must be called in pairs. If you find that the channel buffer usage is higher than 80% by running **cat /proc/msp/demux_chanbuf,** but the applications cannot obtain data, channel buffer overflow occurs because buffers are not properly released as required.

You can check whether buffers are released in time by comparing the times that buffers are successfully obtained and released in the demux_chan information. If the difference between the two values is small, such as 1 or 2 (the difference may be several dozens for the video channel), buffers are obtained and released properly. However, if the difference between the two values is great, for example, the difference is several hundred, the applications obtain the buffers without releasing the buffers in time. In this case, check and modify the applications.

```
# cat /proc/msp/demux_chan
DmxId  ChnId  PID      Type Mod Stat  KeyId   Acquire(Try/Ok)   Release
```

```
  8      0    0x1022    VID  PLY OPEN --       1977/1461        1461
  8      1    0x1023    AUD  PLY OPEN --        199/195         195
```

Checking Whether Corresponding Data Exists in the Data Source

Checking whether the corresponding data exists in the data source only reflects whether the data processed by the QAM has problems. Problems related to the channel cannot be excluded. To check whether the cause is a channel issue, you can record all Demux TSs by referring to the preceding section, and analyze the recorded streams by using the stream tool to see whether data with the specific PID that meets the filtering conditions exists.

NOTE

When the audio or video clips cannot be played, you can also locate the problems by using this method.

3.1.3 Locating the Problem of Mosaics or Intermittent Playback (Including Scrambled Streams)

Mosaics or intermittence during video playback may be caused by errors during back-end decoding or errors in the front-end data source. You can save the AV ES streams by running the command for saving ESs (which is described in section 3.1.1 Debugging Commands"3.1.1 Debugging Commands") and play the ESs in a PC or esplay. If the ESs can be properly played, an error may occur in the decoding module or the VOU. You need to check these modules to locate the problem. If an error also occurs during ES playback, you can locate the problem using the method described in the preceding section.

If the streams are scrambled, you can record the scrambled streams as unscrambled streams by using the method of recording TSs from all Demux channels (which is described in section 3.1.1 Debugging Commands"3.1.1 Debugging Commands") and then further locate the problem by checking whether the Demux obtains data incorrectly (the probability is low). The fault location for scrambled streams is complicated because CA descrambling is involved. To check whether the problem is caused by errors in the front-end data source, you are advised to play AV files using STBs from HiSilicon and other vendors and then compare the playback effects.

3.1.4 RAM Port

Failing to Obtain TS Buffers

When audio and video channels are created in the Demux corresponding to the RAM port, back pressure occurs in the RAM port if the audio and video channels do not read data. The RAM port then cannot obtain the TS buffer.

Check whether there is data transmitted over the Demux channels by running **cat /proc/msp/demux_chanbuf**. If the buffer usage of a specific audio or video channel is higher than 80%, back pressure occurs. You need to analyze the applications to find out why the buffer is full. You can find out the information such as whether the channel is a video or audio channel and whether the software obtains the TS buffer by checking demux_chan continuously.

```
# cat /proc/msp/demux_chanbuf
DmxId  ChnId  FqId  OqId  Size  BlkCnt  BlkSize  Read  Write  Used  Overflow
  8     0      1     0 16383K 1008    16644     455     453     99%    0
  8     1      2     1 255K  204    1280     195     193     99%    0
```

You can also find the information such as the sizes of the RAM port TS buffer and used buffer, and whether the upper-layer applications obtain the TS buffer.

RAM Port Data Error

To locate a problem related to the RAM port, you need to check whether an error occurs in the data before being transferred to the RAM port. You can save the data transmitted to the RAM port by using the method described in section 3.1.1 Debugging Commands"3.1.1 Debugging Commands", and then check whether an error occurs in the data before the data is transmitted to the RAM port by playing or analyzing the recorded data streams. You can locate the problems related to the IPTV and video on demand (VOD) by using this method.

3.2 VDEC

3.2.1 VDEC Overview

The VDEC implements encapsulation of the video firmware (VFMW). The VFMW is the core software module for operating the VDEC. The decoder connects to the Demux and VPSS or VO, and video information must be transmitted over it.

The VFMW includes two hardware modules: SCD and VDH.

The functions of these modules are described as follows:

- The SCD module pre-processes TSs that are transferred to the VFMW and transfers the data to the VDH.
- The VDH decodes video data. The decoded data is one-dimensional and cannot be played.

The decoder cannot work properly if an error occurs in any of these modules. You can locate most of the problems related to the decoder by viewing **/proc/VFMW**, **/proc/VFMW_dec**, **/proc/VFMW_scd**, and **/proc/vfmw_dbg**.

To view the general status of the VDEC, run the **cat/proc/vdec00** command. **00** indicates the channel ID.

cat /proc/msp/vdec00

The following information is displayed:

```
============================== VDEC0 ================================
Work State: RUN
VpssID: vpss00
VfmwID: vfmw00
Codec ID: H264(0x4)
Mode: NORMAL
Priority: 3
ErrCover: 100
OrderOutput: 0
CtrlOption*: 0x0
Capability: NORMAL/2160P(2)/H264
Dynamic Frame Store: Enable
```

```
-------------------Dynamic Frame Store Information--------------------
Dynamic Frame Store Mode: Self
DFS use MMZ: 1
DFS config Frame Number: 9
DFS ExtraFrame Number: 6
DFS DelayTime(ms): 0
DFS Max Mem Use(byte): 0xfa00000
DFS Memory PhyAddress: 0x0
DFS Memory Length(byte): 0
--------------------------Stream Information--------------------------
Source: User0
StreamSize(Total/Current): 0x758959b0/0xe96ac4
BitRate(bps): 7388984
LumaBitdepth: 8
ChromaBitdepth: 8
StreamBufferSize(Total/Used/Persent): 0x1000000/0xe96ac4/91%
--------------------------Picture Information-------------------------
Width*Height: 1920*1080
Stride(Y/C): 0x800/0x800
FrameRate(fps): Real(24.99) FrameInfo(25000)
PlayFormat: OTHER(273)
FrmPackingType: 2D
Aspect(User/Decode): 0:0/0:2
FieldMode: Frame
Type: Interlace
YUVFormat: SP420
TopFirst: TRUE
ErrFrame: 0
TypeNum(I/P): 53362/0
VP6 Picture Reversed: NO
DMX/USER->VDEC
GetStreamBuffer(Try/OK): 331891/120372
PutStreamBuffer(Try/OK): 120372/120372
VDEC->VFMW
AcquireStream(Try/OK): 121607/120372
ReleaseStream(Try/OK): 119438/119438
VFMW->VPSS
AcquireFrame(Try/OK): 53387/53362
ReleaseFrame(Try/OK): 53359/53359
VPSS->AVPLAY
AcquireFrame(Try/OK): 317331/106672
ReleaseFrame(Try/OK): 213342/213342
```

- **Work State** indicates the current running status of the decoding channel. **VpssID** indicates the ID of the corresponding VPSS channel. **VfmwID** indicates the ID of the corresponding VFMW channel. **Codec ID** indicates the decoding video protocol type of the current channel. **ErrCover** indicates the error threshold for decoding outputs. The value **100** indicates that all streams are output, and the value **0** indicates that the stream is not output if any error occurs. **Capability** indicates the channel capability.

- **Stream Information** indicates the status information of streams. **StreamSize(Total/Current)** indicates the size of all streams received by the VDEC and the size of streams that are not decoded in the current buffer. **LumaBitdepth** and **ChromaBitdepth** indicate the bit depths of the stream luminance and chrominance respectively. The value **8** indicates 8 bits, and the value **10** indicates 10 bits. **StreamBufferSize(Total/Used/Persent)** indicates the usage of the stream buffer. **Total** indicates the size of the VDEC stream buffer, **Used** indicates the size of the occupied stream buffer, and **Persent** indicates the utilization of the stream buffer.
- **Picture Information** indicates the decoding information of the current stream. **Width*Height** indicates the resolution. **FrameRate(fps)** indicates the frame rate, **Real(24.99)** indicates the actual statistics, and **FrameInfo(25000)** indicates the frame rate obtained from the stream information. **TopFirst** indicates the field priority. **True** indicates top field first, and **False** indicates bottom field first. **TypeNum(I/P)** indicates the number of interlaced frames/progressive frames.
- **DMX/USER->VDEC** indicates the information about obtaining and releasing stream buffers from the VDEC by the upstream module.
- **VDEC->VFMW** indicates the information about obtaining and releasing raw stream data from the VDEC by the VFMW.
- **VFMW->VPSS** indicates the information about obtaining and releasing decoded frame data from the VDEC by the VPSS.
- **VPSS->AVPLAY** indicates the information about obtaining and releasing frame data from the VPSS by the AVPLAY through the VDEC.

3.2.2 Dynamically Adjusting the VFMW Module by Running the Echo Command

You can view the global information and debugging options of the VFMW module by running the following command:

cat /proc/VFMW_dbg

Information similar to the following is displayed:

```
------------------------debug options--------------------------
tracer address     :0x835ee000
path to save debug data    :/mnt
print enable word :0x3
VFMW_state enable word     :0xffffffff
bitstream control period   :200
frame control period:0
rcv/rls img control period  :500
---------------------------------------------------------------
you can perform VFMW debug with such command:
echo [arg1] [arg2] > /proc/VFMW
debug action   arg1      arg2
------------------------------  --------  ---------------------
```

```
set print enable0x0      print_enable_word
set err_thr    0x2      (chan_id<<24)|err_thr
set dec order output   0x4      (chan_id<<24)|dec_order_output_enable
set dec_mode(0/1/2=IPB/IP/I)   0x5    (chan_id<<24)|dec_mode
set discard_before_dec_thr     0x7      (chan_id<<24)|stream_size_thr
set postprocess options 0xa      (dc<<8)|(db<<4)|dr,
0000=auto,0001=on,0010=off
set frame/adaptive storage     0xb      0:frame only, 1:adaptive
pay attention to the channel   0xd      channel number
channel vcmp config    0xe      chanId: arg2>>27,
  mirror_en: (arg2>>26)&1,
  vcmp_en: (arg2>>25)&1,
  wm_en: (arg2>>24)&1,
  wm_start: (arg2>>12)&0xfff,
  wm_end: (arg2)&0xfff
print tracer   0x100    tracer address. do not care if VFMW still running
start/stop raw stream saving   0x200    chan_id
start/stop stream seg saving   0x201    chan_id
start/stop 2D yuv saving0x202    chan_id
save a single 2D frame  0x203    frame phy addr
save a single 1D frame  0x204    frame phy addr width
height=(height+PicStructure)
set dec_task_schedule_delay    0x400    schedule_delay_time(ms)
set dnr_active_interval 0x401    dnr_active_interval(ms)
stop/start syntax dec 0x402    do not care
set trace controller   0x500    VFMW_state_word in /proc/VFMW_prn
set bitstream control period   0x501    period (ms)
set frame control period0x502    period (ms)
set rcv/rls img control period  0x503    period (ms)
where:
debug action   arg1      arg2
------------------------------  --------  -
```

- debug action indicates the debug action to be performed.
- arg1 indicates the command code of the debug operation.
- arg2 indicates the parameter of the debug operation.

```
cat /proc/VFMW_prn
'print_enable_word' definition, from bit31 to bit0:
---------------------------------------------------------------
<not used>  DEC_MODE   PTS DNR
FOD SCD_INFO   SCD_STREAM  SCD_REGMSG
BLOCK      DBG SE  SEI
SLICE      PIC SEQ MARK_MMCO
POC DPB REF QUEUE
```

```
IMAGE      STR_BODY    STR_TAIL    STR_HEAD
STREAM     UPMSG      RPMSG      DNMSG
VDMREG     CTRLERROR      FATAL
---------------------------------------------------------------
'VFMW_state' definition, from bit31 to bit0:
---------------------------------------------------------------
<not used>      <not used>      <not used>    <not used>
<not used>      <not used>      <not used>    <not used>
<not used>      <not used>      <VO_REL_IMG>  <VO_RCV_IMG>
<2D_TO_QUEUE>    <DNR_INTERRUPT>  <DNR_START> <1D_TO_QUEUE>
<VDH_REPAIR>     <VDH_INTERRUPT>  <VDH_START> <GENERATE_DECPARAM>
<DECSYNTAX_SEG>  <SCD_INTERRUPT>  <SCD_START> <RCV_RAW>
```

Displaying the PTS Information

The command word for enabling print is **0**. The print contents are defined by a 32-bit number. The definitions are as follows:

```
'print_enable_word' definition, from bit31 to bit0:
 <not used>  DEC_MODE   PTS DNR
 FOD SCD_INFO   SCD_STREAM  SCD_REGMSG
 BLOCK      DBG SE  SEI
 SLICE      PIC SEQ MARK_MMCO
 POC DPB REF QUEUE
 IMAGE      STR_BODY   STR_TAIL   STR_HEAD
 STREAM     RPMSG      UPMSG      DNMSG
 VDMREG     CTRLERROR      FATAL
```

For example, the command word for enabling PTS print is **0x20000000**. You can run the following command to display the PTS.

```
echo 0 0x20000000 > /proc/VFMW
```

Saving Streams and YUV Before Decoding

If you cannot determine whether erratic display is caused by the errors in the decoder or modules involved in the communication between the network and Demux, save the streams transmitted to the decoder for analyzing. If the saved streams are normal, the problem is caused by the decoder; otherwise, it is caused by the modules before the decoder. At present, the VDEC and VFMW are available for decoding. Both modules provide the stream recording function. When you suspect that a decoding error occurs, record streams by using the VDEC or VFMW.

The command code for saving streams by using the VDEC is **savestream**. The corresponding parameter **arg2** indicates the channel ID.

echo savestream 0 > /proc/msp/vdec_ctrl

The command code for the stream saving command **start/stop raw stream saving** before decoding is 0x200. The corresponding parameter **arg2** indicates the channel ID. For example, you can save the streams of channel 0 by running the following command:

echo 0x200 0 > /proc/VFMW

Streams are saved in **/mnt/** by default with the name **VFMW_raw_save_NUM.dat**. **NUM** increases from 0 based on the number of stored files. You can also modify the directory for storing the file. For example, you can store the file in **/home** by running the following command:

echo 0x200 0 /home > /proc/VFMW

You can test the stored streams by using sample/esplay or third-party tools such as VLC.

You can save data output from the VDEC by directly saving the YUV data. The commands are **saveyuv** and **saveoneyuv**. The **saveyuv** command stores decoded data continuously. The **saveoneyuv** command stores only one frame of decoded data. The data is stored in the **/mnt/** directory by default. You can also change the directory. The following commands save the decoded data from channel 0 to the **/home** directory:

echo saveyuv 0 /home > /proc/msp/vdec_ctrl

echo saveoneyuv 0 /home > /proc/msp/vdec_ctrl

Playing One Frame Per Second

When you need to play the video slowly to analyze the display problem, adjust the scheduling interval of the VDH in the decoder. For example, you can adjust the VDH to schedule once per second. In this case, the decoder outputs one frame per second.

echo 0x401 1000 > /proc/VFMW

You can cancel the scheduling interval and return to the normal playback mode by running the following command:

echo 0x401 0 > /proc/VFMW

3.2.3 Locating the Playback Pause Issue

You can locate the problems by viewing the internal information about the VFMW. Typically, the reasons why the playback pauses are as follows:

- Insufficient streams
- Insufficient decoder performance (or scheduling errors)
- Display (including synchronization) errors

Checking VFMW Streams

If the streams are insufficient or lost, the playback pauses. You can confirm whether the problem is caused by insufficient streams by running the following command:

cat /proc/VFMW_scd

Information similar to the following is displayed:

```
============== scd info ==============================IsScdDrvOpen:1
SCDState:0 ThisInstID:0 LastProcessTime:169427729 HwMemAddr :0x8729f000
HwMemSize :131072 DownMsgMemAddr:0x8729f000
UpMsgMemAddr:0x872a2000-------------- inst[0] -------------Mode:1 cfg
VidStd:0 is wait seg ext:0 cfg BufPhyAddr:0x89d5c7ff cfg BufSize:2129920 raw
```

```
Total size:5224476 raw Total num:319 seg Total size:1840411 seg Total num:143
actual bitrate:51349 Kbps
```

Pay attention to the number of raw packets and seg packets.

- raw Total size: number of bytes in streams that are not processed by the SCD module
- raw Total num: number of packets in streams that are not processed by the SCD module
- seg Total size: number of bytes in streams that are processed by the SCD module
- seg Total num: number of packets in streams that are processed by the SCD module

The numbers of raw packets and seg packets must be analyzed based on the scenario and packet size.

During local playback, the numbers of raw packets and seg packets are both large. The number of bytes can be more than 100 KB when the streams are sufficient.

During live playback, the number of raw packets is relatively small. However, if the number of bytes in seg packets is more than 10 KB, the playback is normal.

If the total numbers of packets and bytes are both small, for example, if the number of packets is less than ten and the number of bytes is less than 10 KB, you need to analyze the stream supply problem.

- If the numbers of total packets and bytes are small, and the actual bit rate is also small, the stream supply is insufficient. In this case, check whether the Demux or channel has errors.
- If the number of total packets and bytes is small but the actual bit rate is large, frame loss occurs in the VDH, or the streams are forcibly released during parsing. Those video frames are processed by the SCD, and therefore the actual bit rate is large. However, the VDH discards frames continuously, or the streams are continuously released before entering the VDH, which results in small data amount in the frame buffer of the SCD.

  The VDH discards frames due to the following reasons:
- The decoding format is incorrect.

  For example, if the stream format is H.264 but the decoding format is MPEG2, a lot of frames are discarded.
- The VDH is faulty.

Checking the Output Frame Buffers of the Decoder

If the playback pause is caused because the VOU or VPSS obtains buffers too slowly, back pressure occurs in the decoder. The output frame buffer of the decoder is full.

If the DNR module is used, check the DNR information by running the following command:

```
cat /proc/vfmw_dec
============= DEC INFO Chan [0] ==============
Dec by VDH:-1(NA)
IsOpen:1
IsRun:1
Priority:3
eVidStd:H264
ChanCapLevel:18
LastFrameState:BLANK
```

```
IsOmxPath:0
DecMode:0
DecOrderOutput:0
ErrorThred:100
StreamThred:0
ChanMem VDH Mem Phy:0x0
ChanMem VDH Mem Size:0
ChanMem SCD Mem Phy:0x4dc91000
ChanMem SCD Mem Size:31539200
ChanMem CTX Mem Phy:0x4faa5000
ChanMem CTX Mem Size:6026584
Ref,Read,New:4,3,7
VoQue Detail:(47,50,57)
Actual DecMode:0
Actual DecOrderOutput:0
Image Format:0x15fa2
Frame Packing Type:0
------------------- FSP INFO ------------------
FSP DecFsWidth:1920
FSP DecFsHeight:1088
FSP DispFsWidth:1920
FSP DispFsHeight:1088
FSP ExpectedDecFsNum:5
FSP TotalValidFSNum:15
FSP ActualPmvNum:15
FSP ActualDecFsNum:15
03 03 01 03 03 03 03 03 12 02 03 03 03 12 12
FSP LogicFsNum:64
03 03 01 03 03 03 03 03 12 02 03 03 03 12 12 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
---------------Dynamic Frame Store---------------
Frame Size(byte): 3891744
Max Time(ms): 30
Reference Frame Number: 5
Pre-alloced Frame Number/Time(ms): 0/0
Dynamic alloced Frame Number/Time(ms): 15/23
All Frame Number/Time(ms): 15/30
------------------- VDH INFO ------------------
VDH 0:17.2%
Frame Rate:24.9 fps(Chan 0)
Cycle State:234/mb
VDH 1:0.0%
```

```
Frame Rate:0.0 fps(Chan 1)
Cycle State:0/mb
```

You can check the usage of the decoding buffer by observing the information similar to "03 03 01 03 03 03 03 03 13 02 03 03 03 13 13".

- If the first bit is **0**, the current frame is not referenced; if it is **1**, the current frame is referenced.
- If the second bit is **3**, the buffer is occupied; if it is **1**, the buffer is being used by the decoder; if it is **0**, the buffer is idle.

  When intermittence occurs during playback, check whether most of the second bits are **3**. If yes, the downstream module does not fetch frames in time. If many of the second bits are **0**, the decoder does not decode data in the buffer in time.

3.3 Display Channel

3.3.1 Proc Information Analysis

You can view the status of the display channels by running the **cat /proc/msp/disp*X*** command.

```
# cat /proc/msp/disp1
-------------------Hisilicon DISP 1 State-------------------
State:Open
Formt/DispMode      :1080i50/2D
RightEyeFirst:Enable
VirtualScreen:1280/720
Offset      :0/0/0/0
AspectRatioMode     :Auto
AspectRatio  :16:9
ColorSpace :BT709_YUV_LIMITED->BT709_YUV_LIMITED
Bright      :50
Contrast     :50
Saturation :50
Hue  :50
Background (R/G/B) :0x0/0x0/0x0
Zorder(Bot->Top)    :Video->Gfx
AttachRole :Source
AttachDisp :Disp0
Interface    :YPbPr0(1/2/0)  HDMI0
InitCount    :2
OpenCnt[User/Kerne l]  :1/1
LowbandCount :0
------------------CAST Info---------------------------------
State:Enable
Crop :FALSE
```

```
CropRect    :0/0/0/0
Resolution   :1280/720
PixelFormat  :NV21
FrameRate    :30
LowDelay    :Disable
MemoryType :DispAlloc
BufferNumber :6
BufferWidth/Height :1280/720
BufferSize :1382400
BufferStride :1280
------------------Buffer-----------------------
Acquire(Try/OK)     :1280/1024
Release(Try/OK)     :1022/1022
-----------------------------------------------
BufferQueue[state, FrameID]
(State: 1,Empty[0]; 2,Write[1]; 3,Full[3]; 4,Use[2])
[3,0x03fe] [3,0x03ff] [3,0x0400] [2,0x0000] [4,0x03fc] [4,0x03fd]
```

For details about the parameters, ses section 2.7 DISPLAY"2.7 DISPLAY". **disp0** indicates DISPLAY0, and **disp1** indicates DISPLAY1.

Bright, **Contrast**, and **Saturation** are important parameters. If **Contrast** is **0**, the output of pictures cannot be seen.

3.3.2 Common Issues

What Do I Do If No Picture Is Displayed When the TV is Connected by Using Component Outputs?

Perform the following steps:

1. Check the display information of the TV first to confirm whether the TV detects input (the output format information such as "1080i50" is displayed) but displays the black screen, or detects no input ("No signal input" is displayed).

   If no input is detected, check whether the cables are properly connected. If input is detected, check whether the status of the current display channel is normal by using the Proc information.

2. Check whether the TV supports the HD output format. For example, early TVs do not support the 1080p50/p60 format.

3. Check whether the component signals are correctly connected.

   The TV displays normally only when the Y, Pb, and Pr signals at the TX end are correctly connected to the Y, Pb, and Pr connectors at the RX end. The Y signal contains the sync signal. If the Y signal is not correctly connected, the TV cannot work because it detects no sync signal. You are advised to reconnect the Y signal at the TX end to the connectors at the RX end one by one. If the pictures can be properly output when the signal connects to a particular connector, this connector is the right connector for the Y component. Then you can analyze why the component signals are set incorrectly.

For the VGA output, check whether the INTF is bound to the RGB port and whether the VDAC sequence is correct.

----End

What Do I Do If Pictures Can Be Displayed but the Colors Are Abnormal When the TV Is Connected by Using the Component Outputs?

Exchange the Pb and Pr connectors at the TX end. If the colors are normal, the two chrominance component signals (Pb and Pr) were incorrectly connected. Check whether the configured DAC sequence is consistent with the actual DAC sequence.

What Do I Do If the RX End Does Not Display When VGA Output Is Used?

You are advised to set the resolution to 800 x 600, which is supported by most TVs and monitors. If you have correctly set the resolution but the display device does not respond, check whether the setting of pin multiplexing is correct.

What Do I Do If Transient Black Screen Occurs During Video or Graphics Output?

Check whether the output formats are the same before and after the black screen occurs. For example, check whether the output format switches from 1080i50 to 1080p50. If the output format changes, it is normal that black screen occurs.

What Do I Do If the Edge of the SD Screen Is Incomplete Compared with the HD Screen in the Same-Source Mode?

It is caused by the SD TV processing and is not a problem.

3.4 HDMI

3.4.1 Proc Information Analysis

[Debugging Information]

When the HDMI outputs signals properly, information similar to the following is displayed:

```
# cat /proc/msp/hdmi0
SDK_VERSION:[HiSTBLinuxV100R002] Build Time:[Dec 12 2015, 11:35:36]
HDMI Version: 2.0.0.20151212.0
-------------------------- APPAttr ------------------------------
HDMIEnable    : YES              DefaultAction  : NONE
VideoEnable   : YES              AudioEnable    : YES
AviInfoEnable  : YES              AudioInfoEnable: YES
xvYCCMode     : NO               HDCPEnable    : NO
DeepColorMode  : 24               SpdInfoEnable  : NO
OutColorSpace  : YES              MpegInfoEnable : NO
ColorSpaceAdapt: YES              DebugEnable    : NO
```

```
CtsAuthEnable  : NO                 DrmInfoEnable  : NO
-------------------------- SWStatus ----------------------------
ThreadRun      : YES                RunStatus      : OPEN START
TMDSMode       : DVI
KernelCnt      : 0                  UserCnt        : 1
KCallBack      : NO                 UCallBackCnt : 0
TransitState : NONE
HDCP2.2Support : N/A                 HDCPEnable     : NO
HDCPIsOn       : NO                 HDCPMode       : NONE
KeyLoad        : NO                 HdcpAuthStatus : NO
CECEnable      : NO                 CECStatus      : OFF
CECLogicAddr : 0                   CECNetwork     : NONE
CECPhyAddr     : 00.00.00.00
-------------------------- HWStatus ----------------------------
HotPlug        : YES                Rsen           : YES
PhyOutputEnable: YES                 PhyPowerEnable : YES
TMDSMode       : HDMI1.4            AvMute         : NO
SourceScramble : NO                  SinkScramble : NO
TMDSBitClkRatio: 10                  DataSwing      : 0x16
CLKSwing       : 0x15               SrcTermination : 0x0
TxVnbValue     : 0x2                AdjustClk      : 0x1
TxRegZone      : 0x82
----------------- TaskID=1126 Event Pool[0] Status ----------------------
CNT|ErrTotal|HPD|UnHPD|EdidFail|HdcpFail|HdcpSucc|RsenCon|RsenDis|HdcpUsr
WR:|0      |1  |0    |0      |0      |0      |0      |0      |0
RD:|0      |1  |0    |0      |0      |0      |0      |0      |0
Memory[WkFlg=0 |RdAble= 0| RdPtr=1 | WrPtr=1 ]:
# cat /proc/msp/hdmi0_vo
SDK_VERSION:[HiSTBLinuxV100R002] Build Time:[Dec 12 2015, 11:35:36]
HDMI Version: 2.0.0.20151212.0
------------ VideoAttr --------------------------- AVIIfno --------------
VideoTiming    : 1920*1080p60 16:9    |AVIInfoEnable  : YES
DispFmt        : 1080P@60           |CurrentFormat  : 1920*1080p60
16:9(VIC=16)
PixelClk       : 148500             |VSIFormat      : (HDMI_VIC= 0)
InBitDepth     : 10 Bit             |BarDataPresent : NONE
InColorSpace : YCbCr444            |ColorSpace     : YCbCr444
Colorimetry    : ITU-R BT.709        |Colorimetry    : ITU-R BT.709
PicAspectRatio : 16:9                |PicAspectRatio : 16:9
ActAspectRatio : PICTURE             |ActAspectRatio : PICTURE
PixelRepeat    : 1                  |PixelRepeat    : No Repetition
YCCQuantization: LIMITED             |YCCQuantization: LIMITED
RGBQuantization: DEFAULT             |RGBQuantization: DEFAULT
ExtColorimetry : XV_YCC601           |ExtColorimetry : XV_YCC601
```

```
StereoMode    : NONE              |ItContentValid : NO
HvSyncPol     : HPVP              |HvSyncPol     : HPVP
------------ VedioPath ---------------|ITContentType  : GRAPHICS
VideoMute     : NO               |PicScaling    : UNKNOW
OutBitDepth    : 08 Bit            |ActFmtPresent  : YES
OutColorSpace  : YCbCr444          |ScanInfo      : NONE
YCbCr420_422 : NO               |AVIInfoRawData :
YCbCr422_444 : NO               | 82 02 0d 67  50 a8 00 10
YCbCr444_422 : NO               | 00 00 00 00  00 00 00 00
YCbCr422_420 : NO               | 00
RGB2YCbCr     : NO               |VSInfoRawData  :
YCbCr2RGB     : NO               | 00 00 00 00
Dither       : NO               |
--------------------------------- HDRAttr ------------------------------
HdrMode       : DISABLE            HdrEotfType    : SDR_LUMIN
HdrMetaDataID  : 0                 HdrColorimetry : NONE
DispPrim0_X    : 0                 DispPrim0_Y    : 0
DispPrim1_X    : 0                 DispPrim1_Y    : 0
DispPrim2_X    : 0                 DispPrim2_Y    : 0
WhitePoint_X : 0                  WhitePoint_Y : 0
MaxLuminance : 0                  MinLuminance : 0
MaxLightLevel  : 0                 AverLightLevel : 0
--------------------------------- DRMInfo ------------------------------
DRMInfoEnable  : NO                 EotfType      : SDR_LUMIN
MetadataID    : 0
DRMInfoRawData : 00 00 00 00
# cat /proc/msp/hdmi0_ao
SDK_VERSION:[HiSTBLinuxV100R002] Build Time:[Dec 12 2015, 11:35:36]
HDMI Version: 2.0.0.20151212.0
------------ AudioAttr --------------------------- AudioIfno ------------
SoundIntf     : I2S               |AudioInfoEnable: YES
CodeType      : STREAM            |CodeType    : STREAM
ChannelCnt    : 2_CH              |ChannelCnt    : 2_CH
SampleFreq    : 48000             |SampleFreq    : STR_HEADER
SampleDepth    : 16               |SampleDepth    : 16
DownSample    : NO                |SampleSize    : STR_HEADER
------------ AudioPath ---------------|DownMixInhibit : NO
AudioEnable    : YES              |LevelShiftValue: 0
AudioMute     : NO               |LFEPlayBack    : UNKNOW
SoundIntf     : I2S               |Channel/SpeakerAlloc: 0x00 (0)
ChannelCnt    : 2_CH              |AudioInfoRawData:
SampleFreq    : 48000             | 84 01 0a 70  01 00 00 00
SampleDepth    : 16               | 00 00 00 00  00 00
DownSample    : NO                |
CTS          : N/A                 |
N            : N/A                 |
AudioFIFO     : N/A                 |
# cat /proc/msp/hdmi0_sink
SDK_VERSION:[HiSTBLinuxV100R002] Build Time:[Dec 12 2015, 11:35:36]
HDMI Version: 2.0.0.20151212.0
--------------------------------------- EDIDRawData
-----------------------------------------
/*00H:*/ 0x00,0xff,0xff,0xff,  0xff,0xff,0xff,0x00,  0x4c,0x2d,0xbe,0x0b,
0x01,0x00,0x00,0x00,
/*0fH:*/ 0x02,0x18,0x01,0x03,  0x80,0x59,0x32,0x78,  0x0a,0xee,0x91,0xa3,
0x54,0x4c,0x99,0x26,
/*1fH:*/ 0x0f,0x50,0x54,0xbd,  0xef,0x80,0x71,0x4f,  0x81,0xc0,0x81,0x00,
0x81,0x80,0x95,0x00,
/*2fH:*/ 0xa9,0xc0,0xb3,0x00,  0x01,0x01,0x08,0xe8,  0x00,0x30,0xf2,0x70,
0x5a,0x80,0xb0,0x58,
/*3fH:*/ 0x8a,0x00,0x50,0x1d,  0x74,0x00,0x00,0x1e,  0x02,0x3a,0x80,0x18,
0x71,0x38,0x2d,0x40,
/*4fH:*/ 0x58,0x2c,0x45,0x00,  0x50,0x1d,0x74,0x00,  0x00,0x1e,0x00,0x00,
0x00,0xfd,0x00,0x18,
/*5fH:*/ 0x4b,0x0f,0x87,0x3c,  0x00,0x0a,0x20,0x20,  0x20,0x20,0x20,0x20,
0x00,0x00,0x00,0xfc,
/*6fH:*/ 0x00,0x53,0x41,0x4d,  0x53,0x55,0x4e,0x47,  0x0a,0x20,0x20,0x20,
0x20,0x20,0x01,0x22,
/*7fH:*/ 0x02,0x03,0x3a,0xf1,  0x57,0x61,0x10,0x1f,  0x04,0x13,0x05,0x14,
0x20,0x21,0x22,0x5d,
/*8fH:*/ 0x5e,0x5f,0x60,0x65,  0x66,0x62,0x63,0x64,  0x07,0x16,0x03,0x12,
0x23,0x09,0x07,0x07,
/*9fH:*/ 0x83,0x01,0x00,0x00,  0xe2,0x00,0x0f,0x6e,  0x03,0x0c,0x00,0x10,
0x00,0xb8,0x3c,0x21,
/*afH:*/ 0x00,0x80,0x01,0x02,  0x03,0x04,0xe3,0x0f,  0x01,0xe0,0x01,0x1d,
0x80,0xd0,0x72,0x1c,
/*bfH:*/ 0x16,0x20,0x10,0x2c,  0x25,0x80,0x50,0x1d,  0x74,0x00,0x00,0x9e,
0x66,0x21,0x56,0xaa,
/*cfH:*/ 0x51,0x00,0x1e,0x30,  0x46,0x8f,0x33,0x00,  0x50,0x1d,0x74,0x00,
0x00,0x1e,0x00,0x00,
/*dfH:*/ 0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,
/*efH:*/ 0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x49,
------------------------------ SWStatus
--------------------------------------------
CapFromSink        : YES                RawUpdateErrCnt    : 0
CapIsValid         : YES                ParseErrorType     : 0
```

```
RawIsValid         : YES               ParseWarnType      : 0x00000000
RawGetErrCnt       : 0                 RawLength         : 256
------------------------------ BasicCap
-------------------------------------------
HDMI1.4Support     : YES               1stBlockVersion    : 1.3
HDMI2.0Support     : NO                ManufacturerName   : SAM
MaxTMDSClock(MHz) : 300               ProductCode        : 3006
HDCP1.4Support     : N/A               SerialNumber       : 1
WeekOfManufacture : 2
SCDCSupport        : NO                YearOfManufacture : 2014
DVIDualSupport     : NO                CECAddrIsValid     : YES
AISupport          : YES               CECAddr           : 01.00.00.00
ExtBlockCnt        : 1                 SpeakerSupport     : FL_FR
------------------------------ VidoCap
--------------------------------------------
NativeFormat       : 3840X2160P60 16:9(VIC 97)
ColorSpace         : RGB444 YCbCr444 YCbCr422 YCbCr420
DeepColor          : RGB_30Bit RGB_36Bit YCbCr444_SameRGB
YCbCr420DeepColor :
YCbCr420[Also]     : 97 96 101 102 98
YCbCr420[Olny]     :
Colorimetry        :
------------------------------ FormatCap
------------------------------------------
3840X2160P60 16:9   1080P_60 16:9      1080P_50 16:9      720P_60 16:9
720P_50 16:9       1080i_60 16:9      1080i_50 16:9      1080P_24 16:9
1080P_25 16:9      1080P_30 16:9      3840X2160P24 16:9   3840X2160P25
16:9
3840X2160P30 16:9   3840X2160P50 16:9   4096X2160P50 256:135 4096X2160P60
256:135
4096X2160P24 256:135 4096X2160P25 256:135 4096X2160P30 256:135 NTSC 16:9
PAL 16:9           480P_60 16:9       576P_50 16:9       3840X2160P30 16:9
3840X2160P25 16:9   3840X2160P24 16:9   4096X2160P24 256:135 1152x864
1280x720           1280x800           1280x1024          1440x900
1600x900           1680x1050          V800X600_60        V800X600_56
V640X480_75         V640X480_72         V640X480_67         V640X480_60
V720X400_88         V720X400_70         V1280X1024_75       V1024X768_75
V1024X768_70        V1024X768_60        V1024X768_87
------------------------------ 3DCap
----------------------------------------------
3DSupport          : NO                3DOsdDisparity     : NO
3DDualView         : NO                3DIndepView        : NO
3DTypeSupport      :
------------------------------ AudioCap
```

```
-------------------------------------------
NO.0:
CodeType          : L-PCM             MaxChannelNum      : 2
MaxBitRate(KHz)    : N/A               BitDepth           : 16 20 24
SampleRate(Hz)     : 32000 44100 48000
------------------------------ DetailTiming
------------------------------------------
[NO.]:HACT|VACT|P/I |PClk|AspW|AspH|HFB |HPW |HBB |VFB |VPW |VBB
|ImgW|ImgH|IHS |IVS |IDV
[  0]:3840|2160|P |594M|0 |0 |176 |88  |384 |8 |10  |82  |1872|1053|YES |YES
|NO
[  1]:480 |1920|P |2  M|0 |0 |28  |278 |3301|2 |0 |718 |2092|37  |NO  |NO  |NO
```

For details about the preceding Proc information, see the analysis of the HDMI Proc information in section 2.13 HDMI"2.13 HDMI."

3.4.2 Common Issues

What Do I Do If No Audio Is Output After the STB Is Started?

Do as follows:

- Check whether the issue occurs because the HDMI is in DVI mode.

  Check whether the HDMI or DVI mode is used by analyzing the proc information (**/proc/msp/hdmi0**).
- Ensure that the configured AO sampling rate is consistent with the HDMI sampling rate displayed by running **cat /proc/msp/hdmi0**.
- Analyze the proc information **/proc/msp/hdmi0** to check whether transparent transmission is configured. If yes, run **cat /proc/msp/hdmi0_sink** to check whether the sink device supports transparent transmission.

Why Does Crackles Occur After Fastboot Is Started?

This is a compatibility issue. It is caused because the TV still recovers clocks for HDMI inputs and outputs audio when the STB is in HDMI mode and the AO module has no clock and audio output. Most TVs do not recover clocks for HDMI inputs and output audio in the same situation.

Why Does the Screen Go Red?

Do as follows:

- Check whether the HDMI Proc information contains information about the information frame. If not, the information frame is not transmitted, and the screen goes red.
- Check whether HI_UNF_HDMI_Start is called in the hot-plugging function. If not, errors (such as red screen) occur during hot plugging. Check whether HI_UNF_HDMI_Start is called in applications. If yes, red screen sometimes occurs.

What Do I Do If the EDID Cannot Be Obtained?

Do as follows:

- Ensure that the pull-up and pull-down resistors of the HDMI I^2C channels are properly connected. For details, see the reference design of the hardware schematic diagram.
- Measure the I^2C waveform when inserting or removing the HDMI cable. It is most likely that the I^2C is faulty.

What Do I Do If the Screen Sometimes Goes Blank?

This issue is typically caused by the hardware.

- Ensure that the specifications of the HDMI eye pattern fall within the reference range.
- Ensure that the power design of the HDMI is based on the hardware reference design. Otherwise, the clock jitter is abnormal and errors occur.

What Do I Do If the TV Fails to Be Authenticated?

This issue is analyzed as follows:

- The HDCP authentication fails.

  The HDCP is sensitive to hot plugging. For details about the hot plugging invocation process, see **sample/common/hi_adp_hdmi.c** of HiSilicon.
- The authentication fails because the DVI device is connected and the information frames are transmitted.

  Set the output mode of the HDMI to **HI_UNF_HDMI_FORCE_DVI** forcibly by calling HI_UNF_HDMI_Init().
- The aspect ratio fails the authentication.

  Ensure that the aspect ratio is properly set by calling HI_UNF_HDMI_SetInfoFrame and HI_UNF_HDMI_SetAttr.
- The pixel repetition authentication fails.

  Ensure that the ZME and IFIR settings of the VDP and the Dither settings of the HDMI are correct.

3.5 Video Display

3.5.1 Common Video Display Issues

What Do I Do If the Video Screen Goes Blank or Static?

Check whether the window is in the pause or freeze state by viewing the window Proc information consecutively for multiple times. If the window is in the pause or freeze state, static screen or blank screen is normal. If the window is not in the pause or freeze state, check whether the value of **QFrame(Try/OK)** increases. If no, the application (generally, an AVPLAY) pushes no new video images, contact professional personnel to identify the problem.

What Do I Do If the Screen Does not Change After Setting the Output Aspect Ratio (16:9 or 4:3) and Conversion Mode (Adding Black Borders and Clipping)?

Check whether the settings of **ARConvert** are as expected by viewing the window Proc information.

If the settings are correct, check whether the aspect ratio (WvsH) of the video source is consistent with that of the output window. If yes, the conversion mode is invalid.

What Do I Do If Display Discontinuity, Frame Skipping, or Intermittent Display (Especially the Horizontal Scrolling Subtitles) Occurs?

Check whether the value of **Underload** increases by viewing the window Proc information:

- If yes, underload occurs. The problem may be caused by system performance errors. Analyze the problem based on the specific scenarios.
- If no, check whether the value of State is QuickMode. If the value of State is QuickMode, the video is in quick output mode, and the display discontinuity is normal.
- In other cases, analyze the AVPLAY and synchronization module.

What Do I Do If the Third Window and Subsequent Windows Fail to Be Created or Enabled in the Multi-Window Scenario?

When there are three or more windows, the window layout must meet the following requirements:

- Only one window can overlap with other windows, and other windows cannot overlap each other.
- The Z order of the window that overlaps with other windows must be greater than or less than the Z orders of all windows that it overlaps with.

Therefore, you may need to change the window layout or disable windows that need to be mounted to the background to meet the preceding requirements.

NOTE

- The preceding requirements apply to only enabled windows but not disabled ones.
- The window layout indicates the relative positions and overlapping sequence of windows when multiple windows are created.

What Do I Do If the Content of a Window Is Switched to the Content of Another Window When Streams Are Played in the Multi-Window Scenario?

This kind of issue also occurred during development. To locate the problem, capture streams of the window in which the issue occurs for multiple times, and compare the window contents based on time sequence to check whether the content is switched. The commands are as follows:

- **echo help > /proc/msp/winxxx**
- **echo capture path > /proc/msp/winxxx**

If the content is switched, streams transmitted by the upstream module are faulty. Otherwise, VO processing is faulty.

3.6 Audio

3.6.1 Proc Information Analysis

The audio data is derived from the ES and TSs. Various audio formats (such as mp3, pcm, aac, ac3, dts, dtshd, dd, and ddp) are supported. The output interfaces include Adac, SPDIF, and HDMI. Therefore, the problems related to audio may be caused by various reasons.

The Proc information of the ADEC and VOU can be used to analyze audio issues.

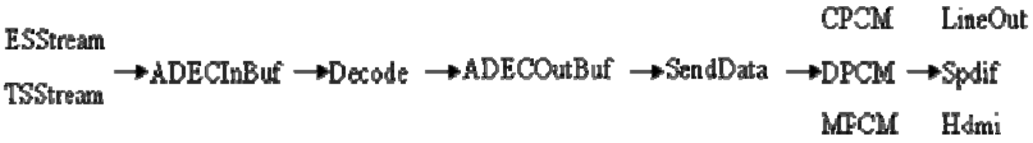
The Proc information of the ADEC is as follows:

```
#cat /proc/msp/adec00
--------------------------- ADEC[00] State --------------------------
WorkState                         :start
CodecID                           :0x202f1011
DecoderName                       :Dolby TrueHD Decoder
Description                       :hisi_truehd
DecodeThreadID                     :2537
Volume                           :100
SampleRate                        :96000
BitWidth                          :24
Channels                          :2
PcmSamplesPerFrame                  :1840
BitsBytePerFrame                   :0x0
StreamFormat                       :non-packet
TryDecodetimes                     :1088
FrameNum(Total/Error)               :767/0
StreamBuf(Total/Use/Percent)(Bytes)    :262144/205883/78%
StreamBuf(readPos/writePos)          :0xdbcc/0x9000
OutFrameBuf(Total/Use/Percent)        :8/7/87%
GetBuffer(Try/OK)                  :4967/3529
PutBuffer(Try/OK)                  :3529/3529
SendStream(Try/OK)                  :0/0
ReceiveFrame(Try/OK)                :767/760
PtsLostNum                        :0
DecodeThreadExecTimeOutCnt           :1
DecodeThreadScheTimeOutCnt           :0
AO PROC information
# cat /proc/msp/sound0
--------------------------  Sound[0]  Status  --------------------------
SampleRate    :48000
SPDIF Status :UserSetMode(RAW) DataFormat(DD)
HDMI Status   :UserSetMode(RAW) DataFormat(DDP)
--------------------------  OutPort Status  -----------------------------
ADAC0: Status(start), Mute(off), Volume(00dB), TrackMode(STEREO)
```

```
     SampleRate(048000), Channel(02), BitWidth(16), Engine(PCM), AOP(0x0),
PortID(0x12)
     DmaCnt(000850), BufEmptyCnt(000002), FiFoEmptyCnt(000000)
SPDIF0: Status(start), Mute(off), Volume(00dB), TrackMode(STEREO)
     SampleRate(048000), Channel(02), BitWidth(16), Engine(SPDIF RAW),
AOP(0x1), PortID(0x21)
     DmaCnt(000828), BufEmptyCnt(000000), FiFoEmptyCnt(000001)
HDMI0: Status(start), Mute(off), Volume(00dB), TrackMode(STEREO)
     SampleRate(192000), Channel(02), BitWidth(16), Engine(SPDIF RAW),
AOP(0x3), PortID(0x20)
     DmaCnt(000756), BufEmptyCnt(000000), FiFoEmptyCnt(000000)
I2S0: Status(start), Mute(off), Volume(00dB), TrackMode(STEREO)
     SampleRate(048000), Channel(02), BitWidth(16), Engine(PCM), AOP(0x4),
PortID(0x10)
     DmaCnt(000850), BufEmptyCnt(000000), FiFoEmptyCnt(000000)
--------------------- Cast Status  ----------------------
Cast(0): *Aop(0x5), Status(start), UserEnable(On), Weight(100/100),
Mute(off)
        SampleRate(048000), Channel(02), BitWidth(16)
        MaxFrameNum(08), SamplePerFrame(01024), AcquireFrame(On)
-------------------- Track Status  ----------------------
Track(0): Type(master), Status(start), Weight (L/R)(100/100),
ChannelMode(STEREO), Mute(off)
SpeedRate(00), AddMuteFrames(0000), SendCnt(Try/OK)(001112/000322)
AIP(0): Engine(PCM), SampleRate(048000), Channel(02), BitWidth(16),
DataFormat(PCM) , Priority(low), FifoBypass(off)
       EmptyCnt(000000), EmptyWarningCnt(000000),
Latency/Threshold(378ms/400ms)
AIP(1): Engine(SPDIF RAW), SampleRate(048000), Channel(02), BitWidth(16),
DataFormat(DD) , Priority(low), FifoBypass(off)
       EmptyCnt(000000), EmptyWarningCnt(000000),
Latency/Threshold(388ms/400ms)
AIP(2): Engine(HDMI RAW), SampleRate(192000), Channel(02), BitWidth(16),
DataFormat(DDP) , Priority(low), FifoBypass(off)
       EmptyCnt(000000), EmptyWarningCnt(000000),
Latency/Threshold(348ms/400ms)
```

Figure 3-1 shows the audio data streams.

Figure 3-1 Audio data streams



Perform the following steps when an error occurs in the AOU:

1. Check whether audio data is properly transmitted to ADECInBuf.

   ESs:

   ```
   GetBuffer(Try/OK)     :a1/a2
   PutBuffer(Try/OK)     :a3/a4
   ```

   If the value of **a1** does not increase, an error occurs in the thread for transmitting ES data.

   TSs:

   ```
   SendStream(Try/OK)    :a5/a6
   ```

   If the value of **a5** does not increase, an error occurs when the Demux transmits audio ES data.

2. Check whether audio decoding is normal.

   ```
   CodecID                   :b1
   TryDecodetimes           :b2
   FrameNum(Total/Error)          :b3/b4
   ```

   If the values of **a4** and **a6** increase, but the value of **b3** does not increase, audio decoding is not implemented.

   If the value of **b2** increases but that of **b3** does not increase, decoding fails. In this case, check whether the b1 decoder is appropriate.

3. Check whether the AVPLAY properly obtains the decoded audio data from ADECOutBuf.

   ```
   ReceiveFrame(Try/OK) :c1/c2
   ```

   - If the value of c1 does not increase, the interface for obtaining decoded audio data is not called.
   - If the value of c1 increases but that of c2 does not increase, the AVPLAY fails to obtain the decoded audio data.

4. Check whether audio data is properly transmitted to a track. Check the track status by running the following command:

   ```
   # cat /proc/msp/sound0
   SendCnt(Try/OK)  (d1/d2)
   ```

   If the value of **d1** does not increase, the interface for transmitting data to the track is not called.

   If the value of **d1** increases but that of **d2** does not increase, audio data fails to be transmitted to the track.

5. Check whether data output at the audio port is normal. Check the AO status by running the following command:

   ```
   # cat /proc/msp/sound0
   DmaCnt(e1)
   ```

   If the value of **e1** does not increase, no data output occurs at the audio port.

----End

3.6.2 What Do I Do If No Audio Is Output?

If the ES streams cannot be played, check whether the corresponding decoding library exists or whether the audio format is incorrectly identified. For the TSs, the incorrect audio PID also leads to abnormal playback.

Check whether the SDK supports the audio stream format. Currently, the SDK only supports the following sampling rates (Hz): 8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100, 48000, 88200, 96000, 176400, and 192000.

If the ADEC and AO Proc information is normal but no audio is output, note the following:

- ADAC

  Check whether the mute circuit is configured. If yes, verify the GPIO pin configurations.

- HDMI

  Verify the HDMI connection and check whether transparent transmission is enabled for the HDMI but the device connected to the HDMI does not support transparent output.

3.6.3 How Do I Set Transparent Audio Transmission?

When the audio (such as ac3, dd, ddp, dts, and dtshd) containing transparently transmitted data is played, the decoding mode HD_DEC_MODE_THRU (only for transparent transmission) or HD_DEC_MODE_SIMUL (PCM + transparent transmission) is used, and the output of transparently transmitted data is enabled, the AO Proc information similar to the following is displayed:

```
SPDIF Status :UserSetMode(RAW) DataFormat(DD)
HDMI Status   :UserSetMode(RAW) DataFormat(DDP)
```

No audio output is available when a TV connects to the HDMI. In this case, the output data of the HDMI is transparent data (which can be decoded by power amplifiers but cannot be decoded by most TVs). To enable audio output for the TV, set the HDMI to the PCM or AUTO mode.

When the audio format is TrueHD, the PCM data cannot be decoded and needs to be transparently transmitted over the HDMI interface.

NOTE

When the DDP and DTSHD are transparently transmitted to the amplifier (not all amplifiers are supported), the sampling rate is 192 kHz.

3.6.4 What Do I Do If the Audio Output Is Abnormal?

Checking the Configurations of Parameters Related to the PCM Audio Playback

When playing the PCM audio, ensure that the following parameters are correctly set:

- Channels (number of channels)
- SamplesPerSec (sampling rate)
- BitsPerSample (bit width)

Locating the Problem that Noises Occur During Audio Output

Check whether AOU underloads occur by observing the AO Proc information. If yes, data source errors occur, or stream pushing is insufficient. You are advised to check the original streams on a PC by using a player. If the original streams are normal, stream pushing may be insufficient. Perform further operations to locate the problem.

Locating the Problem That the Playback of Audio in Local Composite Streams Is Abnormal

Check whether the AO and ADEC are abnormal by observing the AO and ADEC Proc information respectively. If the ADEC is abnormal, analyze whether the audio ESs are correctly demultiplexed. Do as follows:

- Check the audio information in the composite streams using the mediainfo tool. Check whether the decoder type in the audio information is consistent with that in the ADEC Proc information.
- Extract audio ESs using the Demux and play the audio ESs by using Sample/Esplay. If the playback is abnormal, check the ADEC. If the playback is normal, an error may occur during demultiplexing.

3.7 Synchronization

AV sync adjustment aims to synchronize the audio and video while ensuring proper AV playback. Many factors may lead to async. Sync adjustment cannot avoid async and the adjustment takes some time for fast or slow sync. User experience may be affected because the normal AV playback speed must be changed during sync adjustment. You can locate sync problems according to the following sections.

3.7.1 Disabling Synchronization

Disable synchronization by running the following command:

```
echo SyncRef = none >/proc/msp/sync00
```

- If the problem is solved when synchronization is disabled, the problem is probably caused by synchronization. In this case, go to section 3.7.2 Checking the Proc Information About the Sync Module"3.7.2 Checking the Proc Information About the Sync Module" to further locate the problem.
- If the problem still exists after synchronization is disabled, the problem is not caused by synchronization. In this case, you need to check other modules.

3.7.2 Checking the Proc Information About the Sync Module

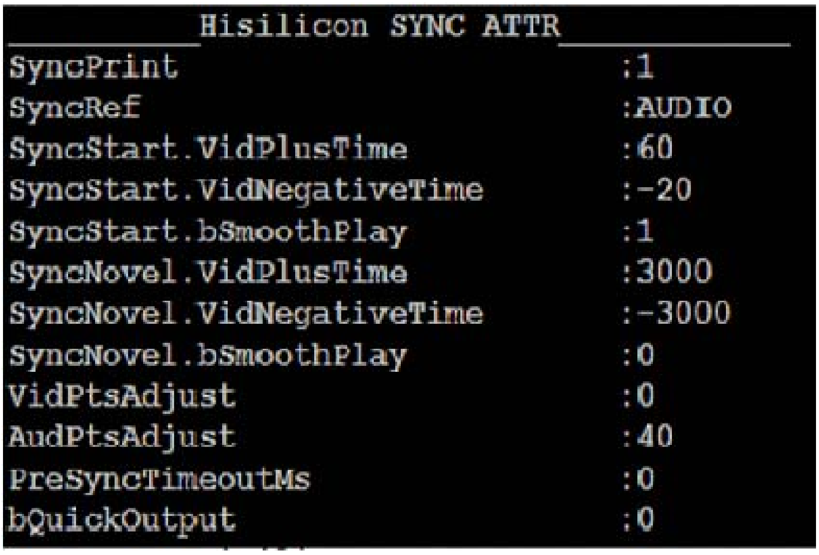
Check the Proc information about the sync module by running the following command:

```
$cat /proc/msp/sync00
```

Check the synchronization attributes.

Figure 3-2 shows the configuration of synchronization attributes.

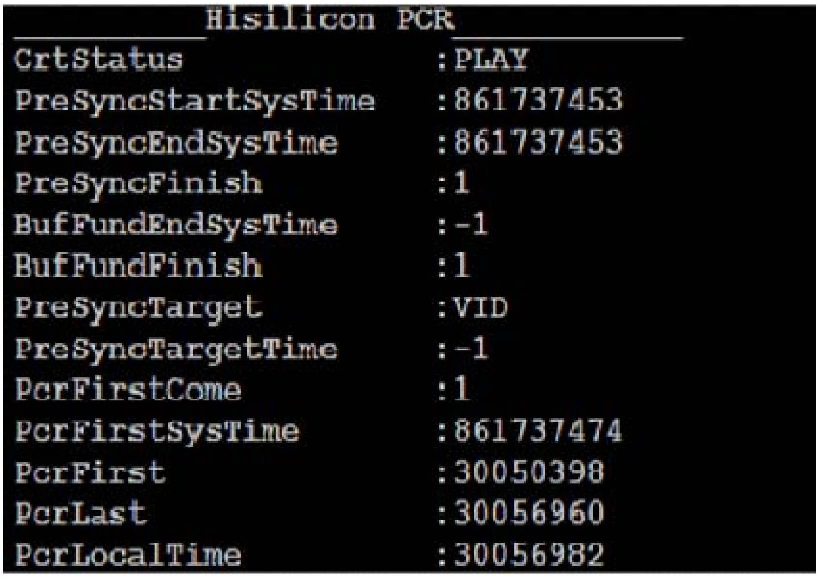
Figure 3-2 Configuration of synchronization attributes



```
__________Hisilicon SYNC ATTR____________
SyncPrint                       :1
SyncRef                         :AUDIO
SyncStart.VidPlusTime           :60
SyncStart.VidNegativeTime       :-20
SyncStart.bSmoothPlay           :1
SyncNovel.VidPlusTime           :3000
SyncNovel.VidNegativeTime       :-3000
SyncNovel.bSmoothPlay           :0
VidPtsAdjust                    :0
AudPtsAdjust                    :40
PreSyncTimeoutMs                :0
bQuickOutput                    :0
```

- Check whether the synchronization reference SyncRef is properly configured. **AUDIO** indicates that the synchronization is based on audio; **PCR** indicates that the synchronization is based on PCR; and **None** indicates free playing.
- Check the sync adjustment range.
  - The interval between **SyncStart.VidPlusTime** and **SyncStart.VidNegativeTime** indicates the sync start region, for example, [–20, +60]. **SyncStart.bSmoothPlay** indicates whether slow sync is set for the sync start region.
- The interval between **SyncNovel.VidPlusTime** and **SyncNovel.VidNegativeTime** indicates the sync exception region, for example, [–3000, +3000]. **SyncNovel.bSmoothPlay** indicates whether slow sync is set for the sync exception region.
- Check the pre-synchronization status and PCR status.

Figure 3-3 Pre-sync status and PCR status



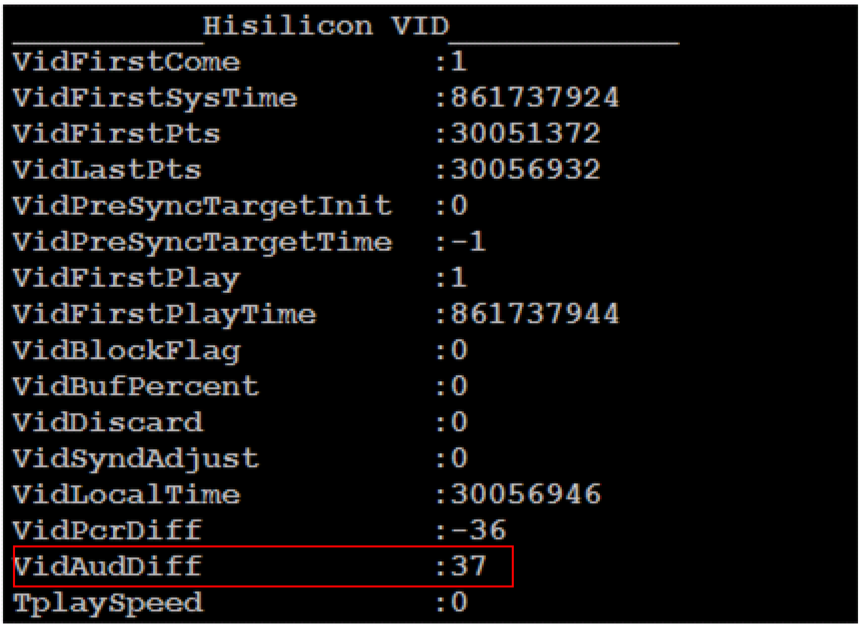
```
__________Hisilicon PCR____________
CrtStatus              :PLAY
PreSyncStartSysTime    :861737453
PreSyncEndSysTime      :861737453
PreSyncFinish          :1
BufFundEndSysTime      :-1
BufFundFinish          :1
PreSyncTarget          :VID
PreSyncTargetTime      :-1
PcrFirstCome           :1
PcrFirstSysTime        :861737474
PcrFirst               :30050398
PcrLast                :30056960
PcrLocalTime           :30056982
```

If **PreSyncEndSysTime** minus **PreSyncStartSysTime** is greater than or equal to **PreSyncTimeoutMs**, the pre-sync times out. When the AV is being played, the adjustment continues but intermittence may occur. Otherwise, the AV is adjusted to the sync state in the pre-sync phase.

If the target for pre-sync is VID, the video leads the audio. The video needs to wait for the audio. If the target for pre-sync is AUD, the audio leads the video. The audio needs to wait for the video. Before the pre-sync times out, the pre-sync is complete when the audio PTS and video PTS are close to the pre-sync target time.

- Check the sync adjustment status of the video.

Figure 3-4 Video sync adjustment status



```
__________Hisilicon VID____________
VidFirstCome              :1
VidFirstSysTime           :861737924
VidFirstPts               :30051372
VidLastPts                :30056932
VidPreSyncTargetInit      :0
VidPreSyncTargetTime      :-1
VidFirstPlay              :1
VidFirstPlayTime          :861737944
VidBlockFlag              :0
VidBufPercent             :0
VidDiscard                :0
VidSyndAdjust             :0
VidLocalTime              :30056946
VidPcrDiff                :-36
VidAudDiff                :37
TplaySpeed                :0
```

VidAudDiff indicates the difference between the current video PTS and the audio PTS. If the difference falls within the sync start region, the sync adjustment is complete. Otherwise, the sync is being implemented.

VidDiscardCnt indicates the number of video frames discarded during sync adjustment. **VidRepeatCnt** indicates the number of frames repeated during sync adjustment.

3.7.3 Locating Sync Problems by Using msp_debug

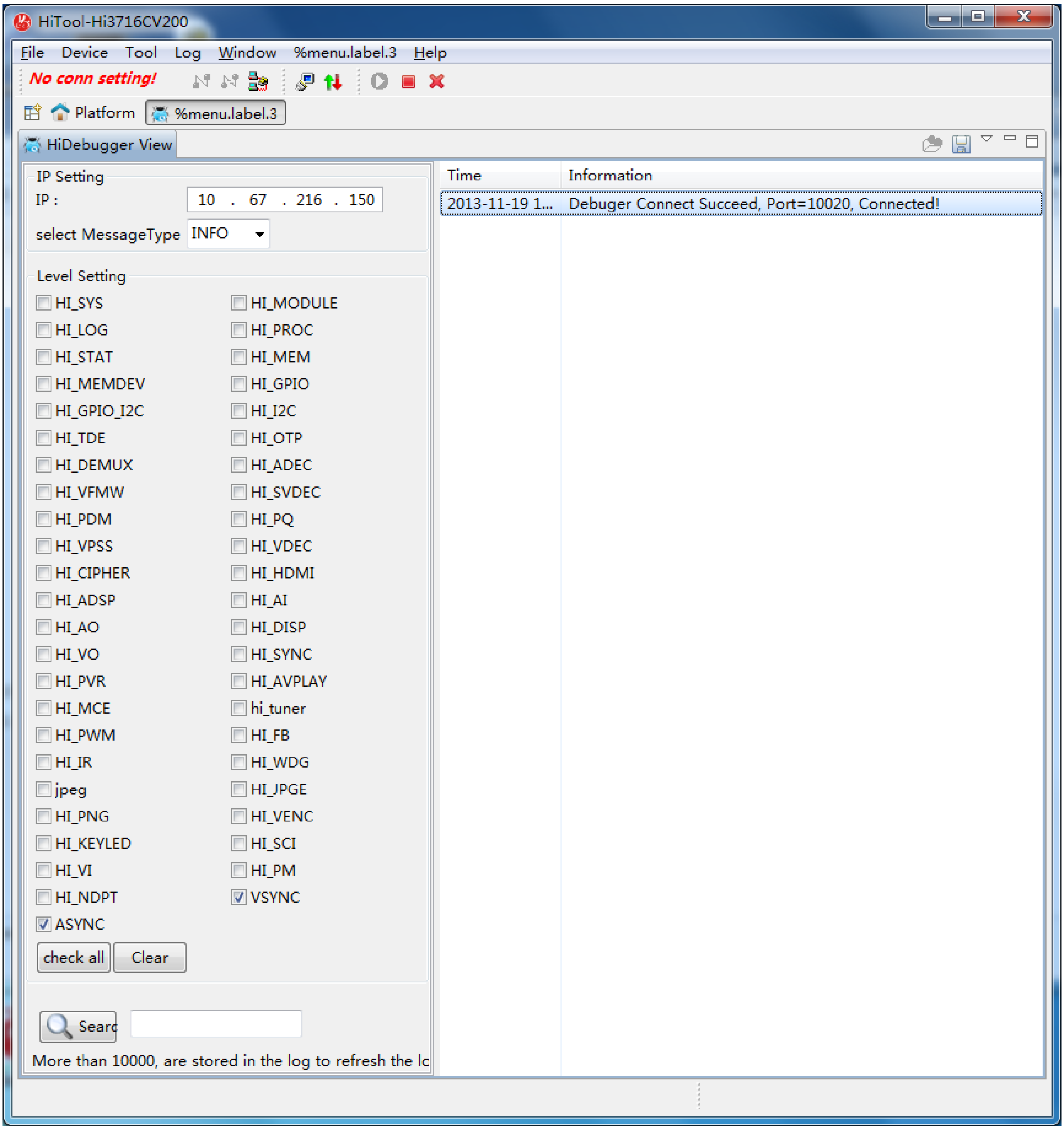
To obtain the sync log information by using msp_debug, perform the following steps:

1. Run the following commands on the board:

   msp_debug &

2. Run the HiTool (**tools/windows/HiTool**) in the SDK on a PC, and select the HiDebugger on the HiTool platform.
3. Set the HiDebugger, as shown in Figure 3-5.

Figure 3-5 Setting the HiDebuger



HiTool-Hi3716CV200

IP Setting

IP : 10 . 67 . 216 . 150

select MessageType INFO

Level Setting

□ HI_SYS □ HI_MODULE
□ HI_LOG □ HI_PROC
□ HI_STAT □ HI_MEM
□ HI_MEMDEV □ HI_GPIO
□ HI_GPIO_I2C □ HI_I2C
□ HI_TDE □ HI_OTP
□ HI_DEMUX □ HI_ADEC
□ HI_VFMW □ HI_SVDEC
□ HI_PDM □ HI_PQ
□ HI_VPSS □ HI_VDEC
□ HI_CIPHER □ HI_HDMI
□ HI_ADSP □ HI_AI
□ HI_AO □ HI_DISP
□ HI_VO □ HI_SYNC
□ HI_PVR □ HI_AVPLAY
□ HI_MCE □ hi_tuner
□ HI_PWM □ HI_FB
□ HI_IR □ HI_WDG
□ jpeg □ HI_JPGE
□ HI_PNG □ HI_VENC
□ HI_KEYLED □ HI_SCI
□ HI_VI □ HI_PM
□ HI_NDPT ☑ VSYNC
☑ ASYNC

More than 10000, are stored in the log to refresh the lc

Time	Information
2013-11-19 1...	Debuger Connect Succeed, Port=10020, Connected!

- Enter the board IP address such as 10.67.212.150.
- Click ▶ to connect the board.

  If the board is successfully connected, the message "Debugger Connect Succeed" is displayed.
- Set MessageType to INFO.
- Select VSYNC and ASYNC.
- Click 💾 to save logs to a specific file.

4. Play the async programs or streams. The HiDebuger records sync logs.

   Send the logs to R&D engineers for analysis.

   ----End

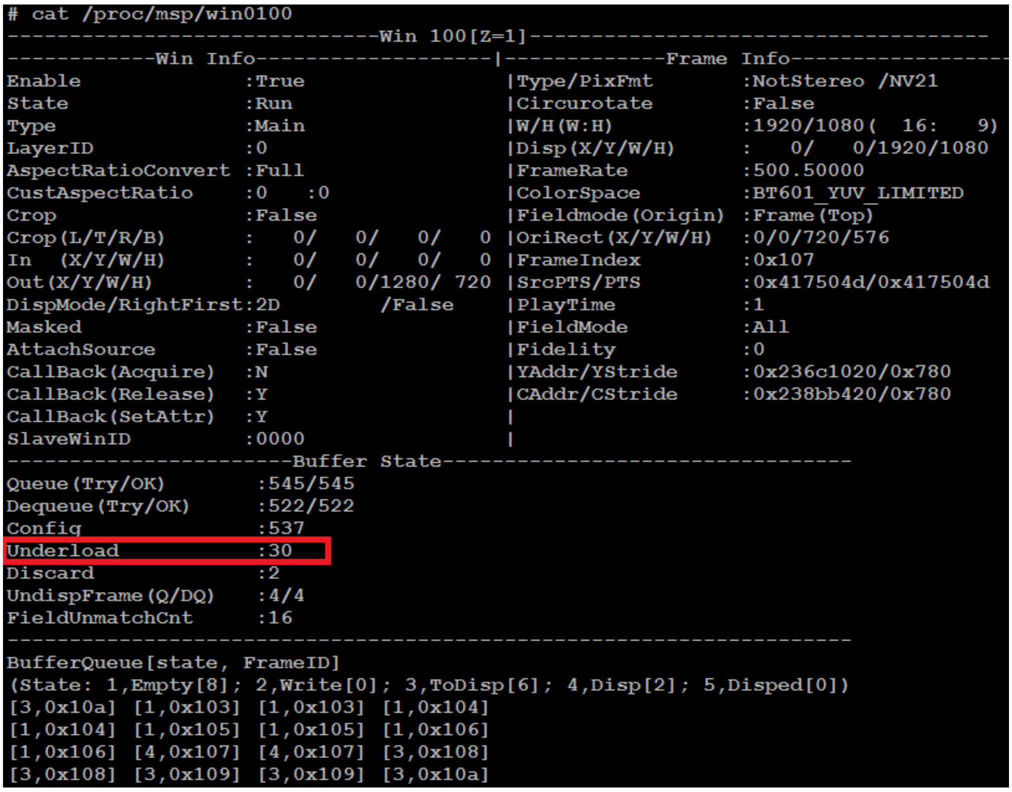
3.7.4 Checking the Cause for Audio and Video Async

Check for AV underloads by viewing the VO and AO Proc information as follows:

- Check whether the value of Unload increases by running the **cat /proc/msp/window0100** command consecutively for multiple times.
- Check whether the value of FifoUnderflowCnt increases by running the **cat /proc/msp/sound0** command consecutively for multiple times.
- AV underloads result in AV async. Identify the cause of AV underloads.

1. Run the **cat /proc/msp/window0100** command consecutively for multiple times and check whether the value of **Unload** increases.

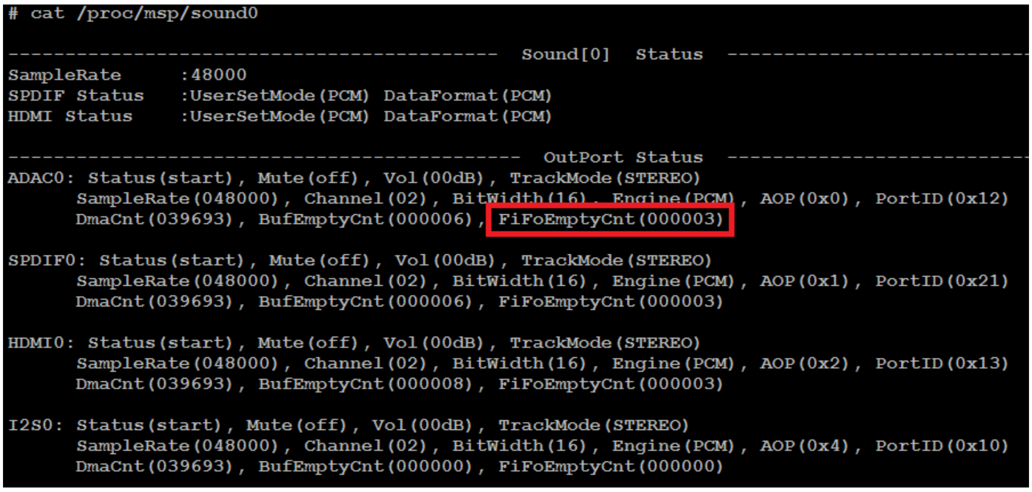
Figure 3-6 cat /proc/msp/win0100 command outputs



```
# cat /proc/msp/win0100
------------------------------Win 100[Z=1]-------------------------------------
------------Win Info-------------------|-------------Frame Info-----------------
Enable              :True               |Type/PixFmt        :NotStereo /NV21
State               :Run                |Circurotate        :False
Type                :Main               |W/H(W:H)           :1920/1080(  16:   9)
LayerID             :0                  |Disp(X/Y/W/H)      :    0/    0/1920/1080
AspectRatioConvert  :Full               |FrameRate          :500.50000
CustAspectRatio     :0    :0            |ColorSpace         :BT601_YUV_LIMITED
Crop                :False              |Fieldmode(Origin)  :Frame(Top)
Crop(L/T/R/B)       :    0/    0/    0/    0 |OriRect(X/Y/W/H)   :0/0/720/576
In  (X/Y/W/H)       :    0/    0/    0/    0 |FrameIndex         :0x107
Out(X/Y/W/H)        :    0/    0/1280/ 720 |SrcPTS/PTS         :0x417504d/0x417504d
DispMode/RightFirst:2D        /False    |PlayTime           :1
Masked              :False              |FieldMode          :All
AttachSource        :False              |Fidelity           :0
CallBack(Acquire)   :N                  |YAddr/YStride      :0x236c1020/0x780
CallBack(Release)   :Y                  |CAddr/CStride      :0x238bb420/0x780
CallBack(SetAttr)   :Y                  |
SlaveWinID          :0000               |
-----------------------Buffer State---------------------------------
Queue(Try/OK)        :545/545
Dequeue(Try/OK)      :522/522
Config               :537
Underload            :30
Discard              :2
UndispFrame(Q/DQ)    :4/4
FieldUnmatchCnt      :16
--------------------------------------------------------------------
BufferQueue[state, FrameID]
(State: 1,Empty[8]; 2,Write[0]; 3,ToDisp[6]; 4,Disp[2]; 5,Disped[0])
[3,0x10a] [1,0x103] [1,0x103] [1,0x104]
[1,0x104] [1,0x105] [1,0x105] [1,0x106]
[1,0x106] [4,0x107] [4,0x107] [3,0x108]
[3,0x108] [3,0x109] [3,0x109] [3,0x10a]
```

2. Run the **cat /proc/msp/sound0** command consecutively for multiple times and check whether the value of **IfFiFoEmptyCnt** increases.

Figure 3-7 cat /proc/msp/sound0 command outputs



```
# cat /proc/msp/sound0

-------------------------------------------  Sound[0]  Status  --------------------------
SampleRate      :48000
SPDIF Status    :UserSetMode(PCM) DataFormat(PCM)
HDMI Status     :UserSetMode(PCM) DataFormat(PCM)

---------------------------------------------  OutPort Status  --------------------------
ADAC0: Status(start), Mute(off), Vol(00dB), TrackMode(STEREO)
      SampleRate(048000), Channel(02), BitWidth(16), Engine(PCM), AOP(0x0), PortID(0x12)
      DmaCnt(039693), BufEmptyCnt(000006), FiFoEmptyCnt(000003)

SPDIF0: Status(start), Mute(off), Vol(00dB), TrackMode(STEREO)
      SampleRate(048000), Channel(02), BitWidth(16), Engine(PCM), AOP(0x1), PortID(0x21)
      DmaCnt(039693), BufEmptyCnt(000006), FiFoEmptyCnt(000003)

HDMI0: Status(start), Mute(off), Vol(00dB), TrackMode(STEREO)
      SampleRate(048000), Channel(02), BitWidth(16), Engine(PCM), AOP(0x2), PortID(0x13)
      DmaCnt(039693), BufEmptyCnt(000008), FiFoEmptyCnt(000003)

I2S0: Status(start), Mute(off), Vol(00dB), TrackMode(STEREO)
      SampleRate(048000), Channel(02), BitWidth(16), Engine(PCM), AOP(0x4), PortID(0x10)
      DmaCnt(039693), BufEmptyCnt(000000), FiFoEmptyCnt(000000)
```

----End

AV underloads result in AV async. Identify the cause of AV underloads.

3.8 Tuner

3.8.1 Debugging Method

The debugging information about the tuner is provided in **/proc/msp**.

You can obtain the information about the tuner type, Demod type, and frequency lock state by running **cat /proc/msp/tuner**.

```
# cat /proc/msp/tuner
---------Hisilicon TUNER Info---------
Port: LockStat I2CChannel  Frequency(KHz)  SymbRate  QamMode  DemodType
 0: locked       3       610000       6875000 QAM_64 3130_outside
Demodaddr       TunerType      Tuneraddr
0xa0            TDA18250B      0xc0
BER:0.0*(E-7),  SNR:48,  SignalStrength:5699
all_rs_package:5136,  corrected_rs_package:5136,  error_rs_package:0
```

You can check the lock state by running **cat /proc/msp/tuner**. If **LockStat** is **unlocked**, the frequency is not locked. If **LockStat** is **locked**, the frequency is locked. You can also directly read the Demod lock indicator register to view the lock phase, for example, AGC lock, CBS lock, TR lock, frame sync lock, or FEC lock, so that you can know about the STB operating status more accurately. For example, if the embedded Demod of Hi379X V100 is used, the address of its lock indicator register is 0x2c, run the **echo** command to directly read the register value. The read register value is 0x37. According to the register description in the Hi379X V100 data sheet, the Demod is locked. Note that the address for the Demod lock indicator register and the meaning of each bit of the register vary according to the Demod type. For details, see the related Demod document.

```
# echo 5 a0 4a >/proc/msp/i2c
0xf8
```

3.8.2 What Do I Do If the Tuner Cannot Lock the Frequency?

Failing to lock the frequency may be caused by many reasons, for example, the signal is weak, the I^2C bus is not properly connected, or the component is not correctly configured. Therefore, you need to find the cause first. A big difference between satellite and cable STBs is that the satellite STB has a low-noise block downconverter (LNB) at the front end. You need to confirm whether the issue is caused because the LNB is incorrectly configured or the receiver cannot receive signals properly.

Finding the Cause of the Problem

Check whether the problem is caused because the signal is weak or the LNB, tuner, or Demod is incorrectly configured. Check whether the signal is normal by using a normal STB. The LNB can output the 13/18 V and 22 kHz signals. You can directly measure the signals by using an oscilloscope, or test the LNB by using the loopout signals of a normal STB. Some tuners provide indication that shows whether the phase-locked loop (PLL) is locked. You can check whether the tuner is locked by reading the corresponding register over the I^2C, or check whether the tuner IF is normal by observing the IF output signals of the tuner using a spectrum analyzer.

Solving the Problem Based on the Cause

- The LNB output signals are abnormal.

  The LNB can be configured to output the 13/18 V and 22 kHz signals. Observe the signals by using an oscilloscope. If the signals comply with the standards, there is no problem.

- The I^2C bus connection is faulty.

  The tuner and Demod are both controlled over the I^2C bus and cannot work properly if the connection of the I^2C bus is faulty. Check the following:

  - Whether the I^2C channel is correctly set
  - Whether the address of the slave device is correct
  - Whether the slave device is always in the reset state
  - Whether the hardware pin multiplexing is correctly set

    You can run the **echo** command for debugging. The format is as follows:

  - Read:

```
echo <I2C channel ID> <Address of the slave device> <Address of the register> >/proc/msp/i2c
```

  - Write:

```
echo <I2C channel ID> <Address of the slave device> <Address of the register> <Value to be written> >/proc/msp/i2c
```

- Other causes

  The problem may be caused for other reasons, for example, the AGC voltage is abnormal (possibly because the AGC pin multiplexing is incorrect) or the AD is set incorrectly and you need to check the configuration of corresponding registers in the data sheet.

3.9 Startup Screen and Fastplay

When the problems related to the startup screen and fastplay occur, for example, the startup screen or fastplay cannot be displayed, artifacts occur, or the switchover is not smooth, you can locate the problems by using the methods described in this section.

3.9.1 Checking the Parameter Configuration

Perform the following steps:

1. Verify the basic parameters by running the **cat /proc/msp/pdm** command on the board.
2. Verify the fastplay parameters by running the **cat /proc/msp/mce** command on the board.
3. If the parameters in the Proc information are consistent with those configured when the image is created, check whether the parameters, such as the DAC configuration, fastplay frequency, and tuner parameters, can ensure the proper running of the startup screen and fastplay.
4. If the parameters in the Proc information are inconsistent with those configured when the image is created, send the image to the R&D engineers for analysis.

----End

3.9.2 Checking the Startup Screen and Fastplay Logs

Run **setenv loglevel 4** under the boot command line, and change the print level of the startup screen and fastplay to Info, and display the logs and send the logs to R&D engineers for analysis.

3.9.3 Accelerating the Boot of Fastplay

To accelerate the fastplay boot, perform the following steps:

1. Set the environment variable **bootdelay** to **0** by running **setenv bootdelay 0;saveenv** in the boot command line.

   bootdelay is set to **1** by default to enable you to enter the boot command line easily. Setting **bootdelay** to **0** can accelerate the fastplay boot by 1s (the default value of **bootdelay** is changed to **0** in the boot code).

2. Disable kernel verification during booting by running **setenv verify n;saveenv** in the boot command line.

   If an error occurs in the kernel, the system is suspended no matter whether the kernel is checked during booting. Therefore, the system is not affected if the kernel is not checked. This accelerates the boot by 1s (the kernel is not checked by default in the boot code).

3. Cancel the BogoMIPS calculation at the kernel stage by adding **lpj=5996544** to **bootargs**.

   BogoMIPS is used to measure the CPU running speed. Canceling the BogoMIPS calculation reduces the startup time by 0.2s.

4. Set the bootcmd.

   Method: Enter **setenv bootcmd 'nand read 0x1FFFFC0 0x0 0x600000;bootm 0x1FFFFC0 '** in the boot command line.

Description: After the bootcmd is set, the boot reads the kernel image from the flash to 0x1FFFFC0 and starts it from 0x1FFFFC0.

If you retain the default bootcmd setting (**nand read 0x82000000 0x0 0x600000;bootm 0x82000000**), the boot reads the kernel image from the flash to 0x82000000, copies it from 0x82000000 to 0x1FFFFC0, and then starts it from 0x1FFFFC0.

NOTE

The 0x1FFFFC0 address mentioned above is an example. The address may vary based on SDK versions. To identify the address, view ***Entry Point:*** xxxxxx in the kernel boot log. xxxxxxx-0x40 is the address. When the address is 0x1FFFFC0, the following information is contained in the kernel boot log: ***Entry Point:02000000***.

----End

3.10 VP

For details about the Proc debugging, see the debugging description of the VI, VPSS, VENC, VDEC, AVPLAY, and VOU.

3.10.1 What Do I Do If the Local Video Cannot Be Played?

Check whether the capturing device (USB camera) is properly connected. If a USB camera is used as the video source, ensure that **/dev/video0** can be detected after the camera is inserted. Then check the Proc information about the VENC module to see whether the frame rate and bit rate are normal.

If the USB camera captures MJPEG data in the VP configuration, ensure that the camera supports the MJPEG format.

3.10.2 What Do I Do If the Remote Video Cannot Be Played?

Check whether the network connection is normal by using the **ping** command and whether local videos are properly displayed. If yes, view the Proc information of the VDEC module and check whether packets are received over the network and transmitted to the VDEC module properly. If packet reception is abnormal, find out why the video RTP packets are not received by using the network packet capture tools. If packet reception is normal, check whether the AVPLAY and VOU work properly by viewing the Proc information.

In addition, you can check whether the video can be properly played after local encoding and decoding in the non-network environment by using sample_vivenc in the SDK.

3.10.3 What Do I Do If Intermittence or Delay Occurs?

Check the system resources, including the CPU usage and memory usage, and disable the services not required for the video communication scenario. Then view the VENC and VDEC Proc information and check whether the encoding and decoding frame rates are close. If the encoding frame rate is much greater than the decoding frame rate, check whether packet loss, delay, and jitter occurs. If the encoding and decoding frame rates are normal, view the AVPLAY and VO Proc information and check whether underload occurs.

3.11 SCI

3.11.1 What Do I Do If the System Has No Response When a Smart Card Is Inserted?

Do as follows:

- Run **cat /proc/msp/sci0** to view the card status information. If "Sci State :NOCARD" is displayed, the card is not inserted properly or the card detection level is incorrectly set.
- Run **cat /proc/msp/sci0** to view the DetectLevel of the card to check whether the card detection level is active low or active high. For details, see the circuit schematic design diagram. The enSciDetect parameter must be consistent with that in the circuit schematic diagram when HI_UNF_SCI_ConfigDetect(enSciPort, enSciDetect) is called.
- Run **cat /proc/msp/sci0** and check whether "SCI0 is not open" is displayed. If yes, the sci0 device is not opened.

3.11.2 What Do I Do If the ATR Cannot Be Received Due to the Reset Failure of the Smart Card?

Do as follows:

- The time for sending all answer to reset (ATR) after the reset signal is detected varies according to the smart card. Check the code to see whether there is enough time to wait for the ATR after HI_UNF_SCI_ResetCard() is called, or obtain the ATR by calling HI_UNF_SCI_GetATR() after the queried card status is HI_UNF_SCI_STATUS_READY. This is especially important in the CA certification test.
- Run **cat /proc/msp/sci0**. If "Sci State :INACTIVECARD" is displayed, the setting of the voltage enable pin may be incorrect. For details, see the circuit schematic diagram. The parameter of enSciVcc must be consistent with the parameter in the circuit schematic diagram when HI_UNF_SCI_ConfigVccEn() is called.
- Run **cat /proc/msp/sci0** and check whether the displayed ProtocolType is consistent with the actual protocol type of the smart card. If not, modify enSciProtocol in the HI_UNF_SCI_Open() function to achieve consistency.
- Run **cat /proc/msp/sci0**. If "Card State: WAITATR" is displayed, a card is inserted and activated, but fails to be reset. Repeatedly insert and remove the card to see whether it can be reset successfully.
- Check whether the power pin of the SCI has common connector voltage (VCC) signals by using an oscilloscope. If there is no VCC signal, check whether the SCI hardware connection is normal.
- Check whether the reset pin of the SCI interface has low level reset voltage signals and whether the level changes to high level after reset by using an oscilloscope.
- If the detected signals are normal, check whether the data I/O lines have data response signals by using an oscilloscope. If there is no data response signal, the card may be abnormal.

3.11.3 What Do I Do If the ATR Can Be Received But Data Cannot Be Received and a Message Indicating Reception Timeouts Is Displayed After the Commands Are Transmitted?

Do as follows:

- Run c**at /proc/msp/sci0** and check whether the displayed ProtocolType is consistent with the actual protocol type of the smart card.
- Run **cat /proc/msp/sci0** and check whether the clock mode is the OD mode or CMOS mode and whether it is consistent with that of the hardware circuit. Configure enClkMode correctly when HI_UNF_SCI_ConfigClkMode() is called.
- After ensuring that the parameters are set correctly, run **cat /proc/msp/sci0** to view the SCI ATR information. Analyze whether the parameters are correct. For example:
  - ActualSciClk :4000

    Actual SCI output clock obtained by configuring registers. A deviation is allowed between **ActualSciClk** and the configured clock, but the deviation must fall within the clock range supported by the card.
- ExpectBaudRate :10215

  ETU value calculated by using the configured F and D factors and the configured clock. The formula is as follows: ExpectBaudRate = SetFrequency x BitRate x 1000/ClkRate.
- CalcBaudFlag :0

  If the ATR receives TA2 and bit 5 is **0**, the specified mode is used and this flag bit is **1**. The ETU is set by using the received TA1 parameter. If TA2 is not received, the interactive mode is used by default and this flag bit is **0**.
- bSetExtBaudFlag :1

  When this parameter is **1**, the applications set the F and D factors of the ETU externally by calling HI_UNF_SCI_SetEtuFactor ().
- ClkRate(F) :372

  Actually configured clock rate conversion factor of the ETU
- BitRate(D) :1

  Actually configured rate adjustment factor of the ETU
- BaudRate :10752

  BaudRrate = ActualSciClk x BitRate x 1000/ClkRate
- AddCharGuard :0 etu

  Add extra character guard time by configuring **TC2** in the received ATR or by calling HI_UNF_SCI_SetGuardTime (). **AddCharGuard** is set to **0** by default.

After analyzing the ATR and Proc information, check whether the configured values correctly correspond to the cards. The default ETU is set based on the default values (**F** of T0 and T1 cards is set to **372**; and **F** of the T14 card is set to **620**) without the negotiation of the ATR values. If the card has special requirements on the ETU, it must be configured based on the TA1 value returned by the ATR or based on other fixed baud rate.

- If the card supports the negotiation mode, the negotiation of the required baud rate can be implemented by calling HI_UNF_SCI_NegotiatePPS().
- If the card does not support the negotiation mode, you can configure the card by calling HI_UNF_SCI_SetFiDi(). (Use this function with cautions because errors may occur when this function is called to configure some cards.)

Note the following:

HI_UNF_SCI_NegotiatePPS() and HI_UNF_SCI_SetFiDi() can be called only when the card status is ready. The **bSetExtBaudFlag** is set when HI_UNF_SCI_SetFiDi() is called.

- If the Proc information and card settings are correct, check whether the actual clock falls within the required range of the card. Reduce the clock frequency based on the range specified in the card manufacturer instruction and test again. For example, set T0 to 3570 kHz. In OD mode, you must also check whether the quality of the rising edge of the clock signal is good.
- When the smart card is tested, note that different smart cards require different command words. For example, the SCI driver transmits instructions and expects to receive 0x19, but the actually used card returns 0x09. In this case, the driver assumes that the data is still to be received and waits until data reception times out.
- If the Proc information and card settings are correct, measure the data signal of the SCI I/O pin by using an oscilloscope to check whether the ETU (calculated by 1/BaudRate) is consistent with the minimum time width of the data transmitted by the smart card. If they are inconsistent, check whether the ETU parameter value meets the card requirement. If necessary, change the ETU and conduct tests again.

3.11.4 How Do I Increase the Wait Time for Receiving the ATR After Cold Reset?

After a smart card is reset and activated, **SCI_ATRSTIME** determines the wait time for receiving the ATR. If the ATR is not received, the release timing is automatically started, and an ATR wait timeout interrupt is reported (SCI_RIS [atrstoutim] is **1**). The value of **SCI_ATRSTIME** is 65535 SCI clocks at most and is 40000 SCI clocks by default. If you want to increase the wait time for receiving the ATR, change the value of the

SCI_DFT_ATRS_TIME macro in **drv_sci.h** to a larger value:

```
#define SCI_DFT_ATRS_TIME       40000 //Change the value of ATR clk to a larger
value.
```

3.12 KEYLED

3.12.1 What Do I Do If the KEYLED Panel Has No Display?

Perform the following steps:

- Check whether the selected KEYLED driver type is consistent with that of the KEYLED chip on the board. At present, the HiSilicon SDK supports the following KEYLED drivers: PT6961 driver, PT6964 driver, CT1642 driver, and FD650 driver. Ensure that the HI_UNF_KEYLED_SelectType function for selecting the KEYLED driver returns a success code, that is, the corresponding KEYLED driver is compiled into the SDK. The current SDK supports one or multiple front panels, which needs to be configured in KeyledConfig of the Msp menu of menuconfig.
- For the front panel that uses the chip logic driver (CT1642), check whether the multiplexing of the KEYLED signal pins is set to the LED and KEY modes.
- For the front panel that uses the GPIO driver, check whether the GPIO pins connected to the front panel are consistent with the pins defined by the driver signals. If no, modify the pin IDs defined by the driver signals in KeyledConfig of menuconfig.

- For the front panel that uses the GPIO driver, check whether the multiplexing of the KEYLED signal pins is set to the GPIO mode.
- Check whether the configured LED segment code in the running applications is correct and whether the LED is co-cathode or co-anode.
- If the problem still exists after the preceding checks, test whether the signal pin of the KEYLED interface has correct signal output by using an oscilloscope.

3.12.2 How Do I Enable CT1642 to Display During System Booting?

⚠ **CAUTION**

The SDKs in the version earlier than SPC091 do not allow CT1642 to display during system booting and only allow it to display for a while during booting. Do not perform the following steps for a panel other than CT1642. When the Hi3798 series chip is used, CT1642 is driven by the chip. You need only to call the corresponding driver interface in the boot code to display specified characters or digits, and the LED keeps displaying the contents during system booting. In this case, the following steps are also unnecessary.

Only the SDKs in the version earlier than SPC091 support this function. The configuration is as follows:

1. Run **make menuconfig**, choose **Msp** > **KeyledConfig** > **Ct1642 Keyled Support,** and set **CFG_KEYLED_CT1642_KERNEL_SUPPORT** to **YES** (you must select **Build MSP in Kernel** in the Kernel menu first).
2. Recompile the SDK by running **make rebuild**.
3. Re-burn the kernel and rootfs after compilation.
4. Enter the characters to be displayed. For example, to display **boot**, add **LedDisplay=boot** in **bootargs**.

----End

3.12.3 How Do I Modify the GPIO Pins Connected to the Front Panel?

When you use the GPIO pins instead of HiSilicon chip logic drivers to scan keys and output data for LED display, and the GPIO pins are different from those on the HiSilicon demo board, perform the following steps:

1. Multiplex the corresponding pins as GPIO by modifying the register configurations in the fastboot table.

   The items to be modified are determined by the IDs and number of used GPIO ports.

   For details about pin multiplexing, see the *Hixxxx Hardware User Guide*.

   Modify the corresponding KEYLED code and set the GPIO required by the front panel.

2. Run **make menuconfig** in the SDK root directory to enter the configuration GUI. Choose **Msp** > **KeyledConfig**, and then configure the corresponding GPIO values based on the used front panel.

   The following is an example.

   The PT6964 front panel is used and its line sequence is as follows: The Data signal connects to the LED_DATA pin, the CLK signal connects to the LED_CLK pin, and the STB signal connects to the LED_STB pin.

   Configure the GPIO pin multiplexing in the fastboot table as follows:

   - Change the value of led_gpio_jtag on the pin_mux page to 2 (corresponding to gpio5_2 and gpio5_3).
   - Change the value of csn0_gpio_jtag on the pin_mux page to 2 (corresponding to gpio5_4).

   Choose **KeyledConfig** > **Pt6964KeyledSupport**, and change the values as follows:

```
ClockGpioNumber   0x2b
STBGpioNumber     0x2c
DINOUTGpioNumber 0x2a
```

   Then recompile the SDK.

CAUTION

The preceding GPIO settings do not affect the KEYLED drivers in standby mode.

----End

3.13 Loader

3.13.1 How Can I Use the BootLoader to Output Debugging Logs?

In the BootLoader scheme, change the value of **loglevel** (ranging from 0 to 4) in the fastboot mode to control log output. When **loglevel** is **0**, no log is output. When **loglevel** is **4**, all logs are output.

```
fastboot# setenv loglevel 4
fastboot# saveenv
```

3.13.2 How Can I Use the AppLoader to Output Debugging Logs?

In the AppLoader scheme, change the value of the environment variable **loglevel** (ranging from 0 to 4) to control log output. When **loglevel** is **0**, no log is output. When **loglevel** is **4**, all logs are output.

```
# loglevel=4 ./loader
```

3.13.3 What Do I Do If the Upgrade Fails After the 1080p Startup Logo Is Burnt?

The creation of the loader graphics layer depends on the **baseparam** parameter so that the TV standard does not need to be changed during upgrade. As the HiFB memory is allocated based on the 720p resolution by default, the HiFB MMZ needs to be reallocated if the **baseparam** parameter specifies that the pictures with the resolution of 1080p are used. Otherwise, the upgrade fails due to the insufficient memory.

To solve this problem, you can modify the loader loading script, and transfer a parameter to specify the size of the memory to be allocated by the HiFB when running **insmod hi_fb.ko**.

Note that this problem only occurs for the AppLoader because the HiFB MMZ does not need to be specified for the BootLoader.

3.13.4 What Do I Do If the Kernel Fails the CRC Check During Booting?

When the value of the last 12 bits of the kernel size (in hexadecimal) is between 0x000 and 0x03F, the cyclic redundancy check (CRC) fails. As a result, the kernel cannot be started. The affected versions range from SPC060 to SPC060F and SPC070 to SPC091.

Solution:

Set the kernel 8 KB-aligned by using **uimage_align.sh** as follows:

```
Usage: ./uimage_align.sh  uImage hi_kernel.img
```

- uImage: unaligned kernel
- hi_kernel.img: 8 KB-aligned kernel

The aligned kernel (**hi_kernel.img**) can be started properly.

3.14 PMOC

The following describes how to debug the MCU part in the standby process.

3.14.1 Displaying the Debugging Information Over the Serial Port

When the serial port module is working properly, you can print a single character by using printf_char in standby code, or print digits by using printf_val. To ensure proper running of the serial port module, make sure that the clock of the serial port is enabled and the power supply is normal. For details about the control registers, see the data sheet of the related chip.

On the earlier platforms, the DEBUG_PRINT_ON macro is added to the standby code to control the functions of serial port modules. When the DEBUG_PRINT_ON macro is defined, you can print debugging information over a serial port by using a printing function.

On the current series platforms, you can set the following debug parameters to enable the serial port display function of the standby module:

Run **echo debug=4 > /proc/msp/pm** to enter the standby mode. The wakeup parameters during the standby entering process are displayed by default, which can be used to check whether the parameters to be configured can be configured correctly. The format of the parameters is as follows:

```
======== [MCU PARAM] ========
Chip Type:0x00000021
Chip Version:0x00000200
Suspend Type:0x00000000
Reboot Enable:0x00000000
Debug Mask:0x00000004
Power Gpio No:0x00000000
Power Gpio Value:0x00000000
Display Mode:0x00000002
Display Time Hour:0x00000009
Display Time Minute:0x0000003A
Display Time Second:0x00000000
WakeUp Time:0xFFFFFFFF
WakeUp Key:0x00000000
IR Type:0x00000000
IR No:0x00000001
IR KeyLow:0x639CFF00
IR KeyHigh:0x00000000
WakeUpGpioNo:0x000000FF
======== [MCU Init Ready] ========
```

For further debugging requirements, use printf_char, printf_val, printf_str, and printf_hex in the standby code as required. Note that the preceding serial port print functions modify the global variables **regAddr** and **regData** internally. Therefore, you need to reconfigure or obtain the values of the two variables after the debugging information is displayed, preventing the original process of the code from being affected.

⚠ **CAUTION**

If the serial port display function is enabled, the standby process cannot proceed after the MCU exits. Therefore, the serial port display function can be used for debugging only the standby code in the MCU.

3.14.2 Displaying Debugging Information on the LEDs

Perform the following steps:

1. View the debugging parameters by running the following command:

   ```
   cat /proc/msp/pm
   ```

2. Set the debugging parameters by running the following command:

   ```
   echo debug=2 > /proc/msp/pm
   ```

   0: The debugging function is disabled.

   1: The debugging information is stored in the MCU memory.

2: The debugging information is displayed on the LED of the front panel.

3: The debugging information is stored in the MCU memory and displayed on the LED of the front panel.

4: The debugging information is output to the serial port.

3. Call dbg_val() in the PMOC code to display the data on the LED.

CAUTION

Debugging information is displayed on the LED after you enable the KEYLED module in standby mode and before you exit from the standby mode while disabling the KEYLED module.

----End

3.14.3 Storing the Debugging Information in the Memory

During debugging in the PMOC code, write the debugging information to the memory and view the debugging information after wakeup. This debugging method requires at least one wakeup mode such as the timer or IR wakeup mode.

Perform the following steps:

1. View the debugging parameters by running the following command:

   ```
   cat  /proc/msp/pm
   ```

2. Set the debugging parameters.

   ```
   echo debug=1 > /proc/msp/pm
   ```

3. Call dbg_val() in the PMOC code to store the debugging information in the memory.
4. In wakeup mode, run the **himd.l** command to view debugging information in the memory.

CAUTION

You can dynamically change the address of the memory for storing the debugging information in dbg_val(). The MCU RAM or the SRAM of the master chip can be used based on the actual situation. However, do not use the memory for storing the standby code and parameters.

----End

3.14.4 Debugging the Kernel Driver in Standby Mode

When the system does not enter the MCU during the standby process or errors occurs on the program after exiting the MCU, you can check the printed standby information about the kernel to locate the cause of the issue.

Make the following modifications:

- Change the variable in **source/kernel/linux-3.18.y/kernel/printk/printk.c** to **false**.

```
bool console_suspend_enabled = false;
```

- Change the macro definition in **source/kernel/linux-3.18.y/include/linux/suspend.h** to **true**.

```
#define pm_print_times_enabled  (true)
```

Recompile the kernel.

3.15 gprof

The gprof is a program analyzing tool on Linux. It records the statistics when the program is running as logs. The information includes the time consumed by each function, the function invocation relationship, and the number of times that each function is called. The gprof helps analyze the most time-consuming function and the program running process. To use the gprof, perform the following steps:

1. Use the **–pg** option (to display the function invocation relationship) when compiling the program to generate an executable program.
2. Run the executable program to generate the **gmon.out** file, which contains the profiling data.
3. Analyze the **gmon.out** file by using the gprof to debug the program. Take **arm-hisiv200-linux-gprof** as an example:

```
arm-hisiv200-linux-gprof prog gmon.out > profile.txt  //prog is the
executable program compiled by using the –pg option.
```

This command outputs the profiling result to **profile.txt**. You can view the profiling result by reading **profile.txt.**

For details about parameters of the gprof, run **arm-hisiv200-linux-gprof --help**.

----End