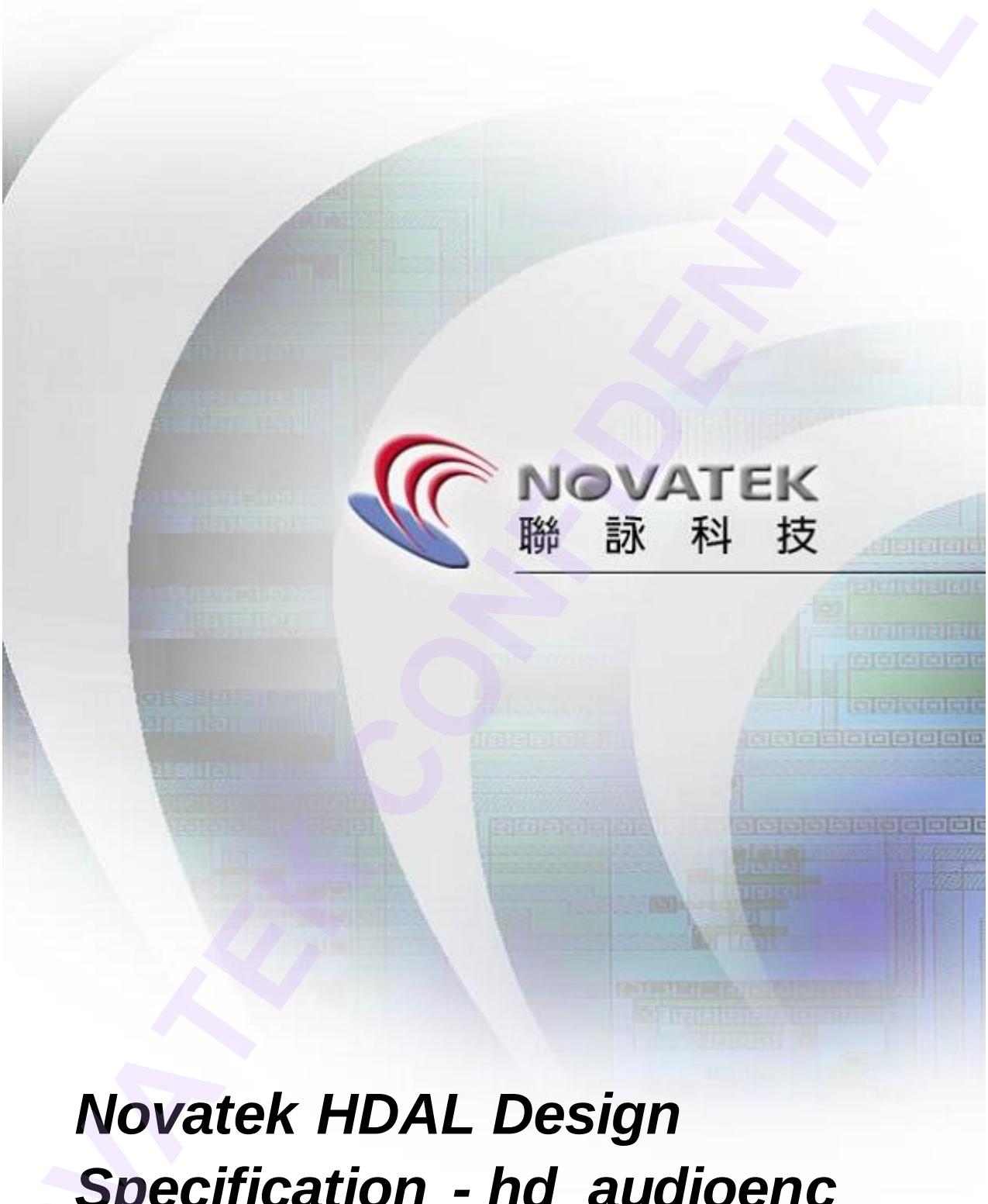


Novatek HDAL Design Specification - hd_audioenc

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

Table of Content

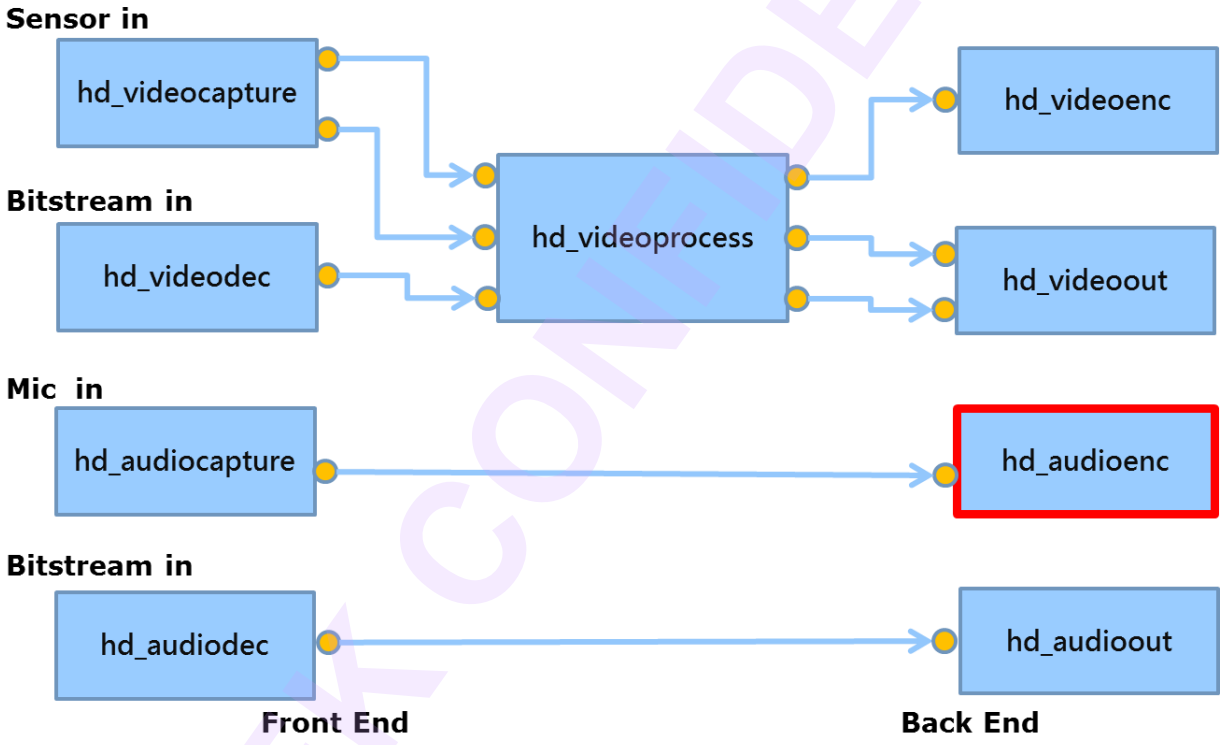
Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

1 Introduction

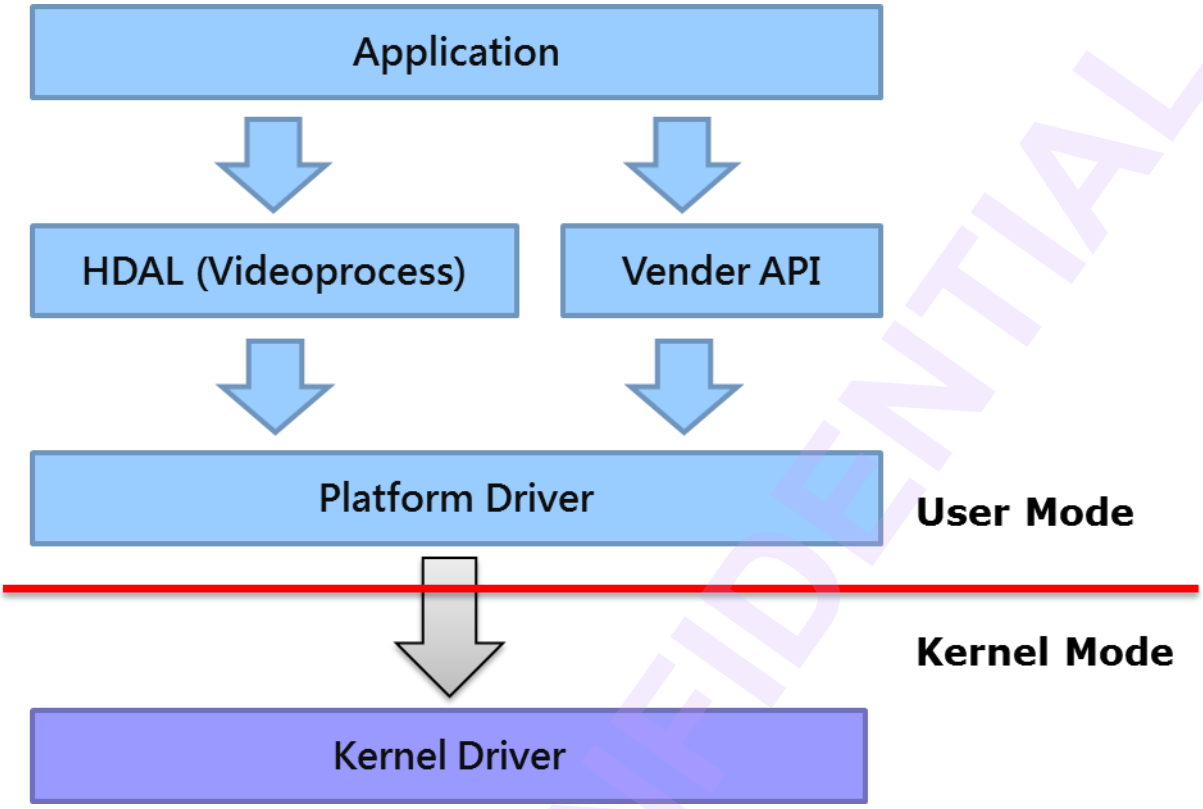
The major purpose of hd_audioenc is to get raw data from upper module, and controls the audio encoder to encode the raw data then return the encoded bitstream data which can be used for reducing the storage space and the bandwidth. This document will talk about the red block in the following diagram. The device driver is not the main point in this document.



Module diagram is shown as below:

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

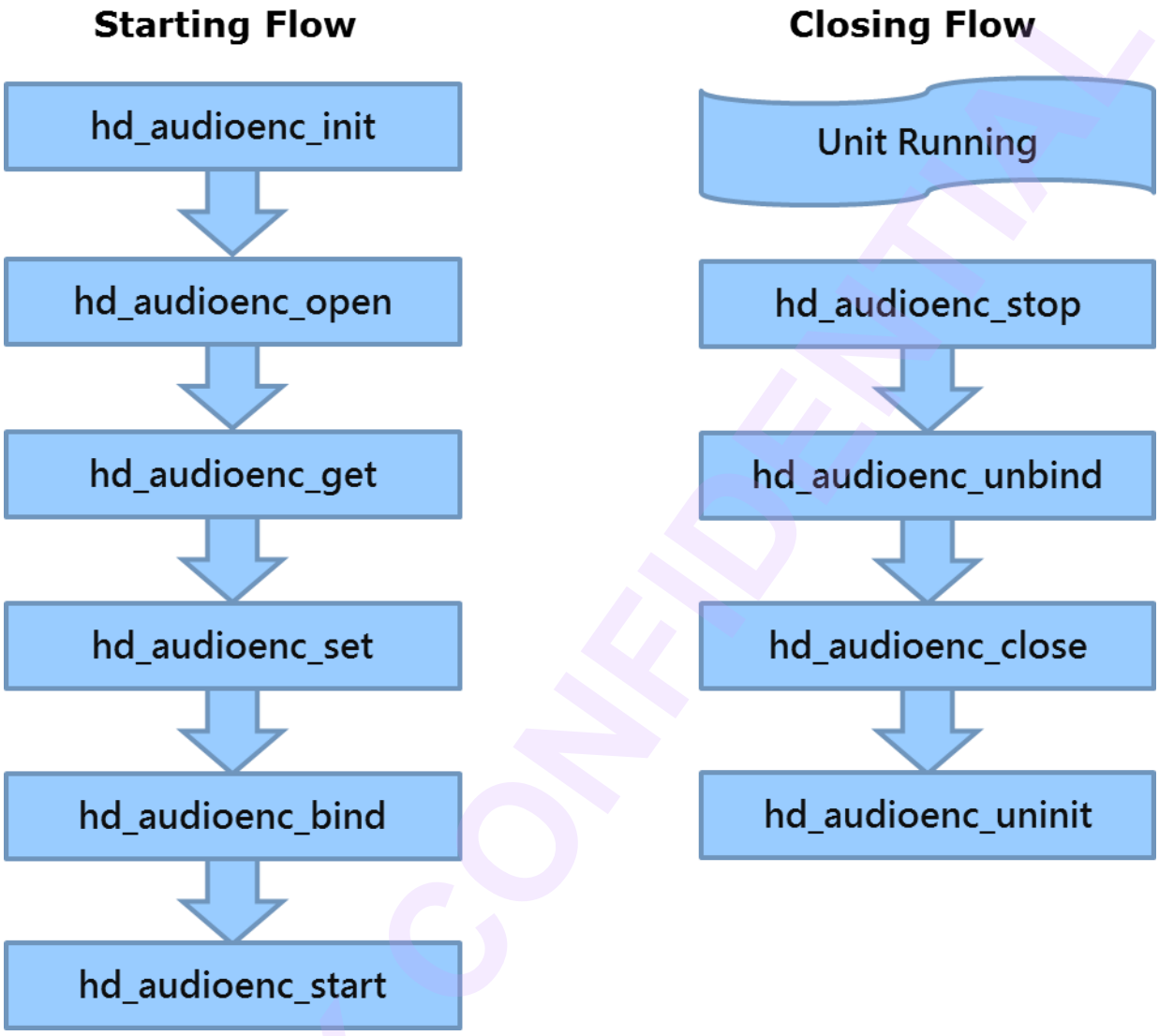


1.1 Basic Flow

The call sequence is needed to be done correctly for the unit. The standard starting flows of most modules are init, open, get, set and start. The standard closing flows of most modules are stop, unbind, close and uninit. The basic flow is shown as below.

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.



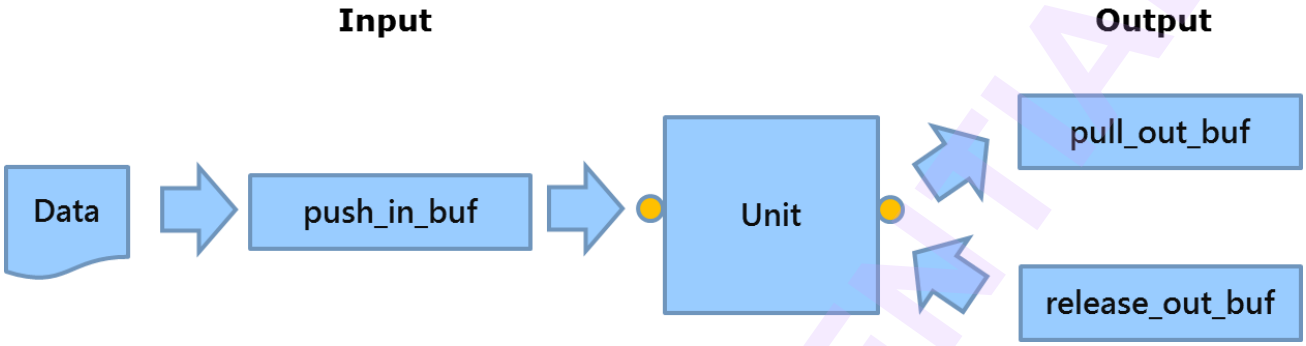
Now, below section in this chapter is mainly about what things to do in those functions above.

1.2 Single Trigger Operation

Single trigger operation is used to trigger the unit to do one job, such as to grab one PCM frame from audio capture; or encode one audio frame to bitstream by using audio encoder. There are some APIs for the input port and output port. The sequence for input port is push; the sequence for output port is pull and release. The flow is

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

shown as below.



NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

2 Function and data structure definition

2.1 General function

2.1.1 hd_audioenc_init

[Description]
Initialize the unit

[Syntax]
HD_RESULT hd_audioenc_init(VOID);

[Parameter]

Value	Description
VOID	Not available

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

2.1.2 hd_audioenc_open

[Description]
Open the unit

[Syntax]
HD_RESULT hd_audioenc_open(HD_IN_ID in_id, HD_OUT_ID out_id,
HD_PATH_ID* p_path_id)

[Parameter]

Value	Description

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

in_id	id of input port.
out_id	id of output port.
p_path_id	pointer of the path id

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

2.1.3 hd_audioenc_get

[Description]
Get parameters from unit by path id

[Syntax]
HD_RESULT hd_audioenc_get(HD_PATH_ID path_id, HD_AUDIOENC_PARAM_ID id, VOID* p_param)

[Parameter]

Value	Description
path_id	the path id
id	id of parameters
p_param	pointer of parameters

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure
HD_ERR_NOT_SUPPORT	Not support this parameter

2.1.4 hd_audioenc_set

[Description]
Set parameters to unit by path id

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

[Syntax]
HD_RESULT hd_audioenc_set(HD_PATH_ID path_id, HD_AUDIOENC_PARAM_ID id, VOID* p_param)

[Parameter]

Value	Description
path_id	the path id
id	id of parameters
p_param	pointer of parameters

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure
HD_ERR_NOT_SUPPORT	Not support this parameter

2.1.5 hd_audioenc_bind

[Description]
Bind this unit with destination unit

[Syntax]
HD_RESULT hd_audioenc_bind(HD_OUT_ID out_id, HD_IN_ID dest_in_id)

[Parameter]

Value	Description
out_id	id of output port.
dest_in_id	id of input port.

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

2.1.6 hd_audioenc_start

[Description]
Start the unit

[Syntax]
HD_RESULT hd_audioenc_start(HD_PATH_ID path_id)

[Parameter]

Value	Description
path_id	pointer of the path id

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

2.1.7 hd_audioenc_stop

[Description]
Stop the unit

[Syntax]
HD_RESULT hd_audioenc_stop(HD_PATH_ID path_id)

[Parameter]

Value	Description
path_id	pointer of the path id

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

2.1.8 hd_audioenc_unbind

[Description]
Unbind the unit

[Syntax]
HD_RESULT hd_audioenc_open(HD_IN_ID in_id, HD_OUT_ID out_id, HD_PATH_ID* p_path_id)

[Parameter]

Value	Description
in_id	id of input port.
out_id	id of output port.
p_path_id	pointer of the path id

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

2.1.9 hd_audioenc_close

[Description]
Close the unit

[Syntax]
HD_RESULT hd_audioenc_close(HD_PATH_ID path_id)

[Parameter]

Value	Description
path_id	pointer of the path id

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

2.1.10 hd_audioenc_uninit

[Description]
Uninitialize the unit

[Syntax]
HD_RESULT hd_audioenc_uninit(VOID);

[Parameter]

Value	Description
VOID	Not available

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

2.1.11 hd_audioenc_push_in_buf

[Description]
Push the audio frame buffer to unit

[Syntax]
HD_RESULT hd_audioenc_push_in_buf(HD_PATH_ID path_id, HD_AUDIO_FRAME* p_in_audio_frame, HD_AUDIO_BS* p_user_out_audio_bs, INT32 wait_ms);

[Parameter]

Value	Description
path_id	the path id
p_in_audio_frame	pointer of the input audio frame
p_user_out_audio_bs	pointer of the output audio bitstream
wait_ms	timeout value in ms

[Return Value]

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

[Difference]

Chip	Description
IPC	All functions are supported.
NVR	All functions are NOT supported.

2.1.12 hd_audioenc_pull_out_buf

[Description]
Pull the audio bitstream buffer from unit

[Syntax]
HD_RESULT hd_audioenc_pull_out_buf(HD_PATH_ID path_id, HD_AUDIO_BS* p_audio_bs, INT32 wait_ms);

[Parameter]

Value	Description
path_id	the path id
p_audio_bs	pointer of the output audio bitstream
wait_ms	timeout value in ms

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

[Difference]

Chip	Description
IPC	All functions are supported.
NVR	All functions are NOT supported.

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

2.1.13 hd_audioenc_release_out_buf

[Description]
Release the audio bitstream buffer which is get from unit

[Syntax]
HD_RESULT hd_audioenc_release_out_buf(HD_PATH_ID path_id, HD_AUDIO_BS* p_audio_bs)

[Parameter]

Value	Description
path_id	the path id
p_audio_bs	pointer of the output audio bitstream

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

[Difference]

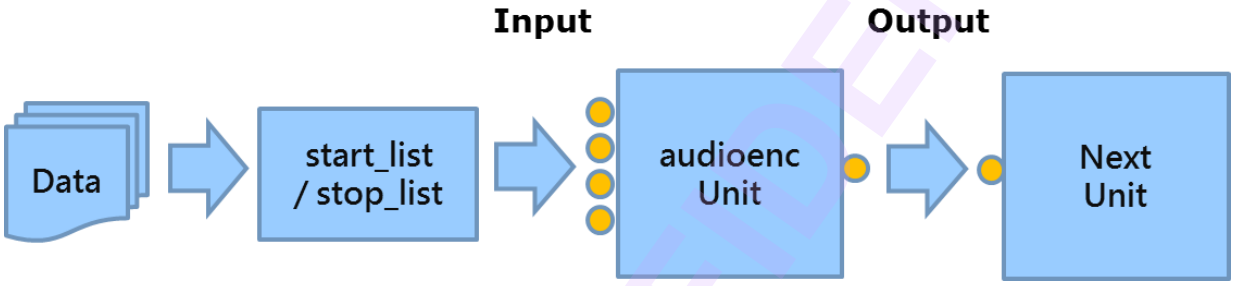
Chip	Description
IPC	All functions are supported.
NVR	All functions are NOT supported.

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

2.2 Multi List Operation

Multi list operation is used to send mulit bitstream simultaneously, it is very efficiency in the multi channels case. The flow is shown as below:



2.2.1 hd_audioenc_start_list

[Description]
Execute multi-start in one command

[Syntax]
HD_RESULT hd_audioenc_start_list(HD_PATH_ID *path_id, UINT num);

[Parameter]

Value	Description
path_id	the path id
num	path number

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

[Difference]

Chip	Description

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and doesnot assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

IPC	Not supported.
NVR	All functions are supported.

2.2.2 hd_audioenc_stop_list

[Description]
Execute multi-stop in one command

[Syntax]
HD_RESULT hd_audioenc_stop_list(HD_PATH_ID *path_id, UINT num);

[Parameter]

Value	Description
path_id	the path id
num	path number

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

[Difference]

Chip	Description
IPC	Not supported.
NVR	All functions are supported.

2.2.3 hd_audioenc_poll_list

[Description]
Query the bitstream status of all given channels

[Syntax]
HD_RESULT hd_audioenc_poll_list(HD_AUDIOENC_POLL_LIST *p_poll, UINT32 num, INT32 wait_ms);

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

NOVATEK CONFIDENTIAL

[Parameter]

Value	Description
p_poll	The path information of multi channels
num	Number of channels
wait_ms	The timeout value in millisecond while polling

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

[Difference]

Chip	Description
IPC	Not supported.
NVR	All functions are supported.

2.2.4 hd_audioenc_recv_list

[Description]

Receive bitstream data for all channels

[Syntax]

HD_RESULT hd_audioenc_recv_list(HD_AUDIOENC_RECV_LIST *p_audio_bs, UINT32 num);

[Parameter]

Value	Description
p_videoenc_list	An array of bitstream structure to be filled for multi channels
num	The number of channels to retrieve bitstream

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

[Return Value]

Value	Description
HD_OK	Success
HD_ERR_NG	Failure

[Difference]

Chip	Description
IPC	Not supported.
NVR	All functions are supported.

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

2.3 Data structure definition

The audioenc provides the following parameter IDs:

- HD_AUDIOENC_PARAM_DEVCOUNT
  - NVR/IPC. support get with ctrl path
  - using HD_DEVCOUNT struct (device id max count)
- HD_AUDIOENC_PARAM_SYSCAPS
  - NVR/IPC. support get with ctrl path
  - using HD_AUDIOENC_SYSCAPS
- HD_AUDIOENC_PARAM_BUFINFO
  - IPC only. support get with i/o path
  - using HD_AUDIOENC_BUFINFO struct
- HD_AUDIOENC_PARAM_PATH_CONFIG
  - IPC only. support get/set with i/o path
  - using HD_AUDIOENC_PATH_CONFIG
- HD_AUDIOENC_PARAM_IN
  - NVR/IPC. support get/set with i/o path
  - using HD_AUDIOENC_IN struct (input frame paramter)
- HD_AUDIOENC_PARAM_OUT
  - NVR/IPC. support get/set with i/o path
  - using HD_AUDIOENC_OUT struct (output bitstream paramter)

2.3.1 HD_AUDIOENC_SYSCAPS

[Description]
System capability

[Parameter]

Value	Description
dev_id	device id

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

chip_id	chip id of this device
max_in_count	max count of input of this device
max_out_count	max count of output of this device
dev_caps	capability of device, combine caps of HD_DEVICE_CAPS and HD_AUDIOENC_DEVCAPS
in_caps	capability of input, combine caps of HD_AUDIO_CAPS and HD_AUDIOENC_INCAPS
out_caps	capability of output, combine caps of HD_AUDIO_CAPS and HD_AUDIOENC_OUTCAPS

2.3.2 HD_AUDIOENC_BUFINFO

[Description]

Buffer information

[Parameter]

Value	Description
buf_info	physical addr/size of bitstream buffer, for user space to mmap see HD_BUFINFO

[Difference]

Chip	Description
IPC	Supported.
NVR	Not supported.

2.3.3 HD_AUDIOENC_PATH_CONFIG

[Description]

Path configuration

[Parameter]

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

Value	Description
max_mem	(IPC only) maximum memory information see HD_AUDIOENC_MAXMEM
data_pool	(NVR only) pool memory information

2.3.4 HD_AUDIOENC_MAXMEM

[Description]

Maximum memory information

[Parameter]

Value	Description
sample_rate	sample rate
sample_bit	sample bit
mode	sound mode

[Difference]

Chip	Description
IPC	Supported.
NVR	Not supported.

2.3.5 HD_AUDIOENC_IN

[Description]

Input frame parameter

[Parameter]

Value	Description
sample_rate	sample rate
sample_bit	sample bit
mode	sound mode

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

2.3.6 HD_AUDIOENC_OUT

[Description]

Output bitstream parameter

[Parameter]

Value	Description
codec_type	NVR/IPC. codec type
aac_adts	IPC only. enable/disable AAC ADTS header (1: enable, 0: disable)

2.3.7 HD_AUDIOENC_RET_EVENT

[Description]

The polling result of audio bitstream

[Parameter]

Value	Description
Event	poll status
bs_size	bitstream size

[Difference]

Chip	Description
IPC	Not supported.
NVR	Supported.

[Difference]

Chip	Description
IPC	Not supported.
NVR	Supported.

2.3.8 HD_AUDIOENC_POLL_LIST

[Description]

The polling item including path information

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

Use this type to form an array in hd_audioenc_poll_list() to get the results for all paths.

[Parameter]

Value	Description
path_id	path ID
revent	The returned event value

[Difference]

Chip	Description
IPC	Not supported.
NVR	Supported.

2.3.9 HD_AUDIOENC_USER_BS

[Description]

Audio bitstream data and relative information

[Parameter]

Value	Description
sign	signature = MAKEFOURCC('A','S','T','M')
p_next	pointer to next meta
acodec_format	encoded format of audio frame
timestamp	encoded timestamp
size	size of encoded data
newbs_flag	Flag notification of new seting, such as GM_FLAG_NEW_BITRATE
p_user_buf	bitstream buffer pointer
user_buf_size	AP provide bs_buf max size

[Difference]

Chip	Description
IPC	Not supported.
NVR	Supported.

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

2.3.10 HD_AUDIOENC_RECV_LIST

[Description]

The audio bitstream item including path information

Use this type to form an array in hd_audioenc_recv_list() to get the audio bitstreams for all paths.

[Parameter]

Value	Description
path_id	path ID
user_bs	audio encode user bitstream
retval	less than 0: recv bistream fail.

[Difference]

Chip	Description
IPC	Not supported.
NVR	Supported.

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

3 Trouble shooting

The audioenc module supports two kinds of debug mechanism for user. User can use proc command or debug menu to debug.

3.1 Debug menu for IPC

In application, call hd_debug_run_menu() to open the debug menu.

```
==============================
 HDAL
------------------------------
 01 : AUDIOCAPTURE
 02 : AUDIOOUT
 03 : AUDIOENC
 04 : AUDIODEC
 05 : VIDEOCAPTURE
 06 : VIDEOOUT
 07 : VIDEOPROCESS
 08 : VIDEOENC
 09 : VIDEODEC
 10 : OSG
 11 : COMMON
 12 : UTIL
 13 : DEBUG
------------------------------
 254 : Quit
 255 : Return
------------------------------
```

Enter “3” to open AUDIOENC debug menu

```
==============================
 AUDIOENC
------------------------------
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
01 : dump status
------------------------------
254 : Quit
255 : Return
------------------------------
```

Note: The items in the menu may vary for IPC or NVR/DVR.

3.1.1 dump status

Enter “1” to show the status of audioenc

```
Run: 01 : dump status
HDAL_VERSION: 00010001:00010001


-------------------------AUDIOENC 0  PATH & BIND -------------------------------
in    out   state   bind_src          bind_dest
0     0     START   AUDIOCAP_0_OUT_0     (null)
-------------------------AUDIOENC 0  PATH CONFIG -------------------------------
in    out   sr    ch    bit   codec
0     0     48000 2     16    AAC
-------------------------AUDIOENC 0  IN FRAME ----------------------------------
in    sr    ch    bit
0     48000 2     16
-------------------------AUDIOENC 0  OUT BS ------------------------------------
out   codec adts
0     AAC   1
------------------------- AUDIOENC 0  IN WORK STATUS -------------------------
in   PUSH drop wrn  err  PROC drop wrn  err  REL
0    47   0    0    0    47   0    0    0    47
------------------------- AUDIOENC 0  OUT WORK STATUS ------------------------
out  NEW  drop wrn  err  PROC drop wrn  err  PUSH drop wrn  err
0    47   0    0    0    47   0    0    0    47   0    0    0
------------------------- AUDIOENC 0  USER WORK STATUS -----------------------
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and doesnot assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
out   PULL   drop  wrn   err   REL
0     0      0     0     0     0
```

As above, the debug menu shows the path & bind information, path_config , input frame / output bitstream information, more detail can see the table as below.

[PATH & BIND]

Status	Description	Value
in	input id of path	0 ~ [max_in_count]
out	output id of path	0 ~ [max_out_count]
state	state of path	OFF/OPEN/START (default OFF)
bind_src	current binding source of input	bind: [module]_[device_id]_OUT_[output_id] not-bind: (null)
bind_dest	current binding source of output	bind: [module]_[device_id]_IN_[input_id] not-bind: (null)

[PATH CONFIG]

Value	Description	Value
in	input id of path	0 ~ [max_in_count]
out	output id of path	0 ~ [max_out_count]
sr	maximum sample rate	enum: user assign sample rate see HD_AUDIO_SR default 0 (n/a)
ch	maximum channel count (sound mode)	enum: user assign channel count see HD_AUDIO_SOUND_MODE default 0 (n/a)
bit	maximum bit width	enum: user assign bit width see HD_AUDIO_BIT_WIDTH default 0 (n/a)

[IN FRAME]

Value	Description	Value
in	input id of path	0 ~ [max_in_count]
sr	current input sample rate	enum: user assign sample rate see HD_AUDIO_SR default 0 (n/a)
ch	current input channel count (sound	enum: user assign channel count,

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

	mode)	see HD_AUDIO_SOUND_MODE default 0 (n/a)
bit	current input bit width	enum: user assign bit width see HD_AUDIO_BIT_WIDTH default 0 (n/a)

[OUT BS]

Value	Description	Value
out	output id of path	0 ~ [max_out_count]
codec	current input audio codec type	enum: user assign codec type see HD_AUDIO_CODEC default 0 (n/a)

3.2 proc command for IPC

User can obtained debugging information from the proc file system of Linux.

3.2.1 dump status

```
[dump info]
cat /proc/hdal/aenc/info
```

the result is exactly the same as 3.1.1 Dump status

```
root@NVTEVM:~$ cat /proc/hdal/aenc/info
HDAL_VERSION: 00010001:00010001


------------------------AUDIOENC 0  PATH & BIND -------------------------------
in    out    state   bind_src            bind_dest
0     0      START   AUDIOCAP_0_OUT_0     (null)
------------------------AUDIOENC 0  PATH CONFIG -------------------------------
in    out    sr     ch     bit    codec
0     0      48000  2      16     AAC
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
-------------------------AUDIOENC 0  IN FRAME ----------------------------------
in    sr    ch    bit
0     48000 2     16
-------------------------AUDIOENC 0  OUT BS ------------------------------------
out   codec  adts
0     AAC    1
------------------------- AUDIOENC 0  IN WORK STATUS -------------------------
in    PUSH  drop  wrn   err   PROC  drop  wrn   err   REL
0     47    0     0     0     47    0     0     0     47
------------------------- AUDIOENC 0  OUT WORK STATUS ------------------------
out   NEW   drop  wrn   err   PROC  drop  wrn   err   PUSH  drop  wrn   err
0     47    0     0     0     47    0     0     0     47    0     0     0
------------------------- AUDIOENC 0  USER WORK STATUS -----------------------
out   PULL   drop  wrn   err   REL
0     0     0     0     0     0
```

3.2.2 debug command

```
[debug port]
echo debug [dev] [i/o] [mask] > /proc/hdal/aenc/cmd
where [dev] = d0 , [i/o] = i0, i1, i2, ..., o0, o1, o2, ... , [mask] = show info mask

[ Sample ]
echo debug d0 o0 mfff > /proc/hdal/aenc/cmd
```

this debug command can show more debug log on console

```
root@NVTEVM:~$ hd_audio_record
[  903.490764] hd_reset - begin
[  903.495618] hd_reset - end
HDAL_VERSION: 00010001:00010001
[  903.500121]
[  903.500121]  "audenc".out[0]: open begin, state=0
[  903.507520]  "audenc".out[0]: cmd OPEN

[  903.513580]  "audenc".out[0]: open end, state=1
[  903.519170]  "audenc".out[0]: set param(0000f000)=2
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and doesnot assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
[  903.524957]  "audenc".out[0]: set param(0000f008)=2
[  903.530719]  "audenc".out[0]: set param(0000f003)=48000
[  903.536815]  "audenc".out[0]: set param(0000f002)=2
[  903.542562]  "audenc".out[0]: set param(0000f004)=16
[  903.548397]  "audenc".out[0]: set param(0000f000)=2
[  903.554143]  "audenc".out[0]: set param(0000f005)=1
[  903.559888]  "audenc".out[0]: set param(0000f007)=7
[  903.565632]  "audenc".out[0]: set param(0000ff00)=2
[  903.772517]
[  903.772517]  "audenc".out[0]: start begin, state=1
[  903.779755]  "audenc".out[0]: cmd RDYSYNC
[  903.784646]  "audenc".out[0]: cmd START
[  903.789765]  "audenc".out[0]: start end, state=2
[  903.795388]  "audenc".out[0]: get param(0000f00b)=310525952
[  903.801874]  "audenc".out[0]: get param(0000f00c)=40000
Enter q to exit, Enter d to debug


dump main bitstream to file (/mnt/sd/dump_bs_16_2_48000_aac.dat) ....


if you want to stop, enter "q" to exit !!


q
[  910.664775]
[  910.664775]  "audenc".out[0]: stop begin, state=2
[  910.671929]  "audenc".out[0]: cmd STOP
[  910.676550]  "audenc".out[0]: stop end, state=1
[  910.682095]  "audenc".out[0]: set param(0000f008)=0
[  910.687850]
[  910.687850]  "audenc".out[0]: close begin, state=1
[  910.695068]  "audenc".out[0]: cmd CLOSE
[  910.699822]  "audenc".out[0]: close end, state=0
```

3.2.3 trace command

```
[trace port]
echo trace [dev] [i/o] [mask] > /proc/hdal/aenc/cmd
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
where [dev] = d0 , [i/o] = i0, i1, i2, ..., o0, o1, o2, ... , [mask] = show info mask


[ Sample ]
echo trace d0 o0 mfff > /proc/hdal/aenc/cmd
```

this trace command could enable module internal debug message to know what's going on for the AUDIOENC module.

3.2.4 probe command

```
[probe port]
echo probe [dev] [i/o] [mask] > /proc/hdal/aenc/cmd
where [dev] = d0 , [i/o] = i0, i1, i2, ..., o0, o1, o2, ... , [mask] = show info mask


[ Sample ]
echo probe d0 o0 mffff > /proc/hdal/aenc/cmd
```

this probe command could print per-data status

```
[ 1308.422250] "audenc".out[0] - NEW - new -- h=00000002 size=00000000 addr=00000002 OK
[ 1308.435837] "audenc".out[0] - PUSH - rel -- h=00000005 (result=0) OK
[ 1308.443181] "audenc".out[0] - PUSH - rel -- h=00000006 (result=0) OK
[ 1308.443557] "audenc".out[0] - NEW - new -- h=00000003 size=00000000 addr=00000003 OK
[ 1308.459200] "audenc".out[0] - PUSH - rel -- h=00000007 (result=0) OK
[ 1308.464886] "audenc".out[0] - NEW - new -- h=00000004 size=00000000 addr=00000004 OK
[ 1308.475229] "audenc".out[0] - PUSH - rel -- h=00000008 (result=0) OK
[ 1308.482570] "audenc".out[0] - PUSH - rel -- h=00000009 (result=0) OK
[ 1308.486256] "audenc".out[0] - NEW - new -- h=00000005 size=00000000 addr=00000005 OK
[ 1308.498647] "audenc".out[0] - PUSH - rel -- h=0000000a (result=0) OK
[ 1308.505994] "audenc".out[0] - PUSH - rel -- h=0000000b (result=0) OK
[ 1308.507567] "audenc".out[0] - NEW - new -- h=00000006 size=00000000 addr=00000006 OK
[ 1308.522031] "audenc".out[0] - PUSH - rel -- h=0000000c (result=0) OK
[ 1308.528908] "audenc".out[0] - NEW - new -- h=00000007 size=00000000 addr=00000007 OK
[ 1308.538064] "audenc".out[0] - PUSH - rel -- h=0000000d (result=0) OK
[ 1308.545404] "audenc".out[0] - PUSH - rel -- h=0000000e (result=0) OK
[ 1308.550229] "audenc".out[0] - NEW - new -- h=00000008 size=00000000 addr=00000008 OK
[ 1308.561440] "audenc".out[0] - PUSH - rel -- h=0000000f (result=0) OK
[ 1308.568779] "audenc".out[0] - PUSH - rel -- h=00000010 (result=0) OK
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and doesnot assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

3.2.5 perf command

```
[perf port]
echo perf [dev] [i/o] > /proc/hdal/aenc/cmd

[ Sample ]
echo perf d0 i0 > /proc/hdal/aenc/cmd
```

this perf command could print data count per second

```
[  160.142626] "audenc".in[0] Perf! -- (Audio) 48 KSample/sec
[  161.145301] "audenc".in[0] Perf! -- (Audio) 48 KSample/sec
[  162.147967] "audenc".in[0] Perf! -- (Audio) 48 KSample/sec
```

3.2.6 save command

```
[save port]
echo save [dev] [i/o] [count] > /proc/hdal/aenc/cmd
where [count] means how many i/o datas to save

[ Sample ]
echo save d0 i0 > /proc/hdal/aenc/cmd
```

this save command could save i/o data to SDCard for debug purpose.

```
[  425.124721] save i/o begin: "audenc".in[0] count=1
[  429.599569] "audenc".in[0] Save -- h=00000001 t=000000001af2266d (ARAW: 16.2.48000
94006368 00000000 4096)
[  429.614003] "audenc".in[0] Save -- //mnt//sd//isf_  audenc_in[0]_16_2_48000_c0.aud ok

[  429.622737] save port end
```

3.3 proc command for NVR

3.3.1 Dump info

```
[dump info]
cat /proc/videograph/hdal_setting
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

The result will show the audiocenc information.

```
root@NVTEVM:/$ cat /proc/videograph/hdal_setting
------------------------- AUDIOENC 0  OUT ----------------------------
in      codec
0       PCM
```

3.4 Debug menu for NVR

Calling hd_debug_run_menu() from app will pop out debug_menu.

The currently supported audioenc module debug menu is as below.

```
==============================
 AUDIOENC
------------------------------
 01 : dump status
------------------------------
 254 : Quit
 255 : Return
------------------------------
1
Run: 01 : dump status
------------------------- AUDIOENC 0  OUT ----------------------------
in      codec
0       PCM
```

User can choose the number to dump the status what you want. The dump result is just like the example shown on 3.3.

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

4 Sample Codes

4.1 audio_record (IPC)

This sample code shows that audiocap and audioenc on binding mode, audiocap capture PCM data from microphone and push data to audioenc, it will encode the PCM data to a bitstream. This sample also dump the bitstream to sdcard.

```
/* Set cap configuration */
ret = hd_audiocap_open(0, HD_AUDIOCAP_0_CTRL, &audio_cap_ctrl); //open this for device control
audio_dev_cfg.in_max.sample_rate = HD_AUDIO_SR_48000;
audio_dev_cfg.in_max.sample_bit = HD_AUDIO_BIT_WIDTH_16;
audio_dev_cfg.in_max.mode = HD_AUDIO_SOUND_MODE_STEREO;
audio_dev_cfg.in_max.frame_sample = 1024;
audio_dev_cfg.frame_num_max = 10;
ret = hd_audiocap_set(audio_cap_ctrl, HD_AUDIOCAP_PARAM_DEV_CONFIG, &audio_dev_cfg);
if (ret != HD_OK) { return ret; }
audio_drv_cfg.mono = HD_AUDIO_MONO_RIGHT;
ret = hd_audiocap_set(audio_cap_ctrl, HD_AUDIOCAP_PARAM_DRV_CONFIG, &audio_drv_cfg);
if (ret != HD_OK) { return ret; }

/* Set cap parameter */
// set audiocap input parameter
audio_cap_param.sample_rate = HD_AUDIO_SR_48000;
audio_cap_param.sample_bit = HD_AUDIO_BIT_WIDTH_16;
audio_cap_param.mode = HD_AUDIO_SOUND_MODE_STEREO;
audio_cap_param.frame_sample = 1024;
ret = hd_audiocap_set(audio_cap_path, HD_AUDIOCAP_PARAM_IN, &audio_cap_param);
if (ret != HD_OK) { return ret; }

/* set enc path configuration */
audio_path_cfg.max_mem.codec_type = enc_type;
audio_path_cfg.max_mem.sample_rate = HD_AUDIO_SR_48000;
audio_path_cfg.max_mem.sample_bit = HD_AUDIO_BIT_WIDTH_16;
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
audio_path_cfg.max_mem.mode = HD_AUDIO_SOUND_MODE_STEREO;
ret = hd_audioenc_set(audio_enc_path, HD_AUDIOENC_PARAM_PATH_CONFIG, &audio_path_cfg);
if (ret != HD_OK) { return ret; }

/* set enc parameter */
audio_in_param.sample_rate = HD_AUDIO_SR_48000;
audio_in_param.sample_bit = HD_AUDIO_BIT_WIDTH_16;
audio_in_param.mode = HD_AUDIO_SOUND_MODE_STEREO;
ret = hd_audioenc_set(audio_enc_path, HD_AUDIOENC_PARAM_IN, &audio_in_param);
if (ret != HD_OK) { return ret; }
audio_out_param.codec_type = enc_type;
audio_out_param.aac_adts = (enc_type == HD_AUDIO_CODEC_AAC) ? TRUE : FALSE;
ret = hd_audioenc_set(audio_enc_path, HD_AUDIOENC_PARAM_OUT, &audio_out_param);
if (ret != HD_OK) { return ret; }

/* Bind audio_record module */
hd_audiocap_bind(HD_AUDIOCAP_0_OUT_0, HD_AUDIOENC_0_IN_0);

/* Pull out buffer */
ret = hd_audioenc_pull_out_buf(p_stream0->enc_path, &data_pull, -1);
if (ret == HD_OK) {
     hd_audioenc_get(p_stream0->enc_path, HD_AUDIOENC_PARAM_BUFINFO, &phy_buf_main);
     vir_addr_main = (UINT32)hd_common_mem_mmap(HD_COMMON_MEM_MEM_TYPE_CACHE,
                                                phy_buf_main.buf_info.phy_addr,
                                                phy_buf_main.buf_info.buf_size);
     #define PHY2VIRT_MAIN(pa) (vir_addr_main + (pa - phy_buf_main.buf_info.phy_addr))
     va = PHY2VIRT_MAIN(data_pull.phy_addr);
     size = data_pull.size;
     sprintf(filename, "dump_bs_main.dat");
     save_output(filename, va, size);
}

/* Release out buffer */
hd_common_mem_munmap(vir_addr_main, phy_buf_main.buf_info.buf_size);
hd_audioenc_release_out_buf(p_stream0->enc_path, &data_pull);
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

4.2 audio_encode_only (IPC)

This sample code demonstrates how to use the single trigger operation to encode PCM data to bitstream.

```
/* Allocate common buffer */
mem_cfg.pool_info[0].type     = HD_COMMON_MEM_COMMON_POOL;
mem_cfg.pool_info[0].blk_size = MAX_FRM_BLK_SIZE;
mem_cfg.pool_info[0].blk_cnt  = MAX_FRM_BLK_CNT;
mem_cfg.pool_info[0].ddr_id   = DDR_ID0;
ret = hd_common_mem_init(&mem_cfg);

/* set enc path configuration */
audio_path_cfg.max_mem.codec_type = enc_type;
audio_path_cfg.max_mem.sample_rate = HD_AUDIO_SR_48000;
audio_path_cfg.max_mem.sample_bit = HD_AUDIO_BIT_WIDTH_16;
audio_path_cfg.max_mem.mode = HD_AUDIO_SOUND_MODE_STEREO;
ret = hd_audioenc_set(audio_enc_path, HD_AUDIOENC_PARAM_PATH_CONFIG, &audio_path_cfg);
if (ret != HD_OK) { return ret; }

/* set enc parameter */
audio_in_param.sample_rate = HD_AUDIO_SR_48000;
audio_in_param.sample_bit = HD_AUDIO_BIT_WIDTH_16;
audio_in_param.mode = HD_AUDIO_SOUND_MODE_STEREO;
ret = hd_audioenc_set(audio_enc_path, HD_AUDIOENC_PARAM_IN, &audio_in_param);
if (ret != HD_OK) { return ret; }
audio_out_param.codec_type = enc_type;
audio_out_param.aac_adts = (enc_type == HD_AUDIO_CODEC_AAC) ? TRUE : FALSE;
ret = hd_audioenc_set(audio_enc_path, HD_AUDIOENC_PARAM_OUT, &audio_out_param);
if (ret != HD_OK) { return ret; }

/* Push in buffer */
blk = hd_common_mem_get_block(HD_COMMON_MEM_COMMON_POOL, blk_size, ddr_id);
pa = hd_common_mem_blk2pa(blk);
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and doesnot assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
va= hd_common_mem_mmap(HD_COMMON_MEM_MEM_TYPE_CACHE, pa, blk_size);
fread((void *)va, 1, length, bs_fd);
audio_frame.phy_addr[0] = pa;
audio_frame.size = length;
ret = hd_audioenc_push_in_buf(p_stream0->enc_path, &audio_frame, NULL, 0);
if (ret != HD_OK) { return ret; }

/* Release in buffer */
hd_common_mem_munmap((void *)va, blk_size)
hd_common_mem_release_block(blk);

/* Pull out buffer */
ret = hd_audioenc_pull_out_buf(p_stream0->enc_path, &data_pull, -1);
if (ret == HD_OK) {
     hd_audioenc_get(p_stream0->enc_path, HD_AUDIOENC_PARAM_BUFINFO, &phy_buf_main);
     vir_addr_main = (UINT32)hd_common_mem_mmap(HD_COMMON_MEM_MEM_TYPE_CACHE,
                                                phy_buf_main.buf_info.phy_addr,
                                                phy_buf_main.buf_info.buf_size);
     #define PHY2VIRT_MAIN(pa) (vir_addr_main + (pa - phy_buf_main.buf_info.phy_addr))
     va = PHY2VIRT_MAIN(data_pull.phy_addr);
     size = data_pull.size;
     sprintf(filename, "dump_bs_main.dat");
     save_output(filename, va, size);
}

/* Release out buffer */
hd_common_mem_munmap(vir_addr_main, phy_buf_main.buf_info.buf_size);
hd_audioenc_release_out_buf(p_stream0->enc_path, &data_pull);
```

4.3 audio_record(NVR)

audioenc doesn't support push/pull operation. User must bind it with audiocap to get audio data. The following demonstrates how to get PCM data by using binding method.

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and doesnot assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include <pthread.h>
#include "hdal.h"
#include "hd_debug.h"

#define MAX_BITSTREAM_NUM   1
#define BITSTREAM_LEN      12800

typedef struct _AUDIO_RECORD {
     UINT32 enc_exit;
     HD_PATH_ID audcap_path_id[MAX_BITSTREAM_NUM];
     HD_PATH_ID audenc_path_id[MAX_BITSTREAM_NUM];
     INT dev_idx;
} AUDIO_RECORD;

HD_RESULT init_module(void)
{
    HD_RESULT ret;
     if((ret = hd_audiocap_init()) != HD_OK)
       return ret;
    if((ret = hd_audioenc_init()) != HD_OK)
       return ret;
    return HD_OK;
}

HD_RESULT open_module(AUDIO_RECORD *p_rec_info)
{
     INT i;
    HD_RESULT ret;
    for (i = 0; i < MAX_BITSTREAM_NUM; i++) {
          if((ret = hd_audiocap_open(HD_AUDIOCAP_IN(p_rec_info->dev_idx, 0),
HD_AUDIOCAP_OUT(p_rec_info->dev_idx, i), &p_rec_info->audcap_path_id[i])) != HD_OK)
               return ret;
          if((ret = hd_audioenc_open(HD_AUDIOENC_IN(0, i), HD_AUDIOENC_OUT(0, i),
&p_rec_info->audenc_path_id[i])) != HD_OK)
               return ret;
    }
    return HD_OK;
}

HD_RESULT close_module(AUDIO_RECORD *p_rec_info)
{
     INT i;
    HD_RESULT ret;
    for (i = 0; i < MAX_BITSTREAM_NUM; i++) {
          if((ret = hd_audiocap_close(p_rec_info->audcap_path_id[i])) != HD_OK)
               return ret;
          if((ret = hd_audioenc_close(p_rec_info->audenc_path_id[i])) != HD_OK)
               return ret;
    }
    return HD_OK;
}
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
HD_RESULT exit_module(void)
{
    HD_RESULT ret;
      if((ret = hd_audiocap_uninit()) != HD_OK)
        return ret;
    if((ret = hd_audioenc_uninit()) != HD_OK)
        return ret;
    return HD_OK;
}

static void *encode_thread(void *arg)
{
      INT i, ret;
      CHAR filename[50];
      CHAR *pcm_data[MAX_BITSTREAM_NUM];
      CHAR *bitstream_data[MAX_BITSTREAM_NUM];
      FILE *pcm_fd[MAX_BITSTREAM_NUM], *pcmlen_fd[MAX_BITSTREAM_NUM];
      HD_AUDIOENC_POLL_LIST poll_list[MAX_BITSTREAM_NUM];
      HD_AUDIOENC_RECV_LIST recv_list[MAX_BITSTREAM_NUM];
      AUDIO_RECORD  *p_rec_info = (AUDIO_RECORD *) arg;

      for (i = 0; i < MAX_BITSTREAM_NUM; i++) {
            //open files for writing
            sprintf(filename, "Dev%dCH%d_audio.pcm", p_rec_info->dev_idx, i);
            pcm_fd[i] = fopen(filename, "wb");
            if (pcm_fd[i] == NULL) {
                  printf("open file error(%s)! \n", filename);
                  exit(1);
            }
            printf("Record file: [%s]\n", filename);
            sprintf(filename, "CH%d_audio.pcmlen", i);
            pcmlen_fd[i] = fopen(filename, "wb");
            if (pcmlen_fd[i] == NULL) {
                  printf("open file error(%s)! \n", filename);
                  exit(1);
            }

            //prepare buffer for receiving data
            pcm_data[i] = (char *)malloc(BITSTREAM_LEN);
            if (pcm_data[i] == 0) {
                  return 0;
            }
            memset(pcm_data[i], 0, BITSTREAM_LEN);
            bitstream_data[i] = (char *)malloc(BITSTREAM_LEN);
            if (bitstream_data[i] == 0) {
                  return 0;
            }
            memset(bitstream_data[i], 0, BITSTREAM_LEN);
      }

      memset(poll_list, 0, sizeof(poll_list));
      for (i = 0; i < MAX_BITSTREAM_NUM; i++) {
            poll_list[i].path_id = p_rec_info->audenc_path_id[i];
      }

      while (p_rec_info->enc_exit == 0) {
            //call poll to check the stream availability
```

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
            ret = hd_audioenc_poll_list(poll_list, MAX_BITSTREAM_NUM, 1000);
            if (ret == HD_ERR_TIMEDOUT) {
                printf("Poll timeout!!\n");
                continue;
            }

            //fill the structure and receive it from audioenc
            memset(recv_list, 0, sizeof(recv_list));
            for (i = 0; i < MAX_BITSTREAM_NUM; i++) {
                if (poll_list[i].revent.event != TRUE) {
                    continue;
                }
                if (poll_list[i].revent.bs_size > BITSTREAM_LEN) {
                    printf("buffer size is not enough! %lu, %d\n",
                            poll_list[i].revent.bs_size, BITSTREAM_LEN);
                    continue;
                }
                recv_list[i].path_id = p_rec_info->audenc_path_id[i];
                recv_list[i].user_bs.p_user_buf = bitstream_data[i];
                recv_list[i].user_bs.user_buf_size = BITSTREAM_LEN;
            }
            if ((ret = hd_audioenc_recv_list(recv_list, MAX_BITSTREAM_NUM)) < 0) {
                printf("Error return value %d\n", ret);
            } else {
                for (i = 0; i < MAX_BITSTREAM_NUM; i++) {
                    if (!recv_list[i].path_id) {
                        continue;
                    }
                    if (recv_list[i].retval < 0) {
                        printf("get bitstreame error! ret = %d\n", ret);
                    } else if (recv_list[i].retval >= 0) {
                        //write audio data
                        if (recv_list[i].user_bs.size > 0) {
                            fwrite(recv_list[i].user_bs.p_user_buf, 1,
recv_list[i].user_bs.size, pcm_fd[i]);
                            fprintf(pcmlen_fd[i], "%lu\n",
recv_list[i].user_bs.size);
                        }
                        fflush(pcm_fd[i]);
                        fflush(pcmlen_fd[i]);
                    }
                }
            }
        }

        //close files and free buffer
        for (i = 0; i < MAX_BITSTREAM_NUM; i++) {
            fclose(pcm_fd[i]);
            fclose(pcmlen_fd[i]);
        }
        return 0;
}

HD_RESULT set_param(AUDIO_RECORD *p_rec_info, INT ch)
{
    HD_RESULT ret;
      HD_AUDIOCAP_IN audiocap_param;
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
    HD_AUDIOENC_OUT audioenc_param;

    //set audiocap parameters
    ret = hd_audiocap_get(p_rec_info->audcap_path_id[ch], HD_AUDIOCAP_PARAM_IN,
&audiocap_param);
    if (ret != HD_OK) {
        printf("hd_audiocap_get(HD_AUDIOCAP_PARAM_IN) fail\n");
        goto exit;
    }
    audiocap_param.sample_rate = HD_AUDIO_SR_8000;
    audiocap_param.sample_bit = HD_AUDIO_BIT_WIDTH_16;
    audiocap_param.mode = HD_AUDIO_SOUND_MODE_MONO;
    audiocap_param.frame_sample = 320;  // for 25fps: 8000/25=320
    ret = hd_audiocap_set(p_rec_info->audcap_path_id[ch], HD_AUDIOCAP_PARAM_IN,
&audiocap_param);
    if (ret != HD_OK) {
        printf("hd_audiocap_set(HD_AUDIOCAP_PARAM_IN) fail\n");
        goto exit;
    }

    //set audioenc parameters
    ret = hd_audioenc_get(p_rec_info->audenc_path_id[ch], HD_AUDIOENC_PARAM_OUT,
&audioenc_param);
    if (ret != HD_OK) {
        printf("hd_audioenc_get(HD_AUDIOENC_PARAM_OUT) fail\n");
        goto exit;
    }
    audioenc_param.codec_type = HD_AUDIO_CODEC_PCM;
    ret = hd_audioenc_set(p_rec_info->audenc_path_id[ch], HD_AUDIOENC_PARAM_OUT,
&audioenc_param);
    if (ret != HD_OK) {
        printf("hd_audioenc_set(HD_AUDIOENC_PARAM_OUT) fail\n");
        goto exit;
    }

exit:
    return ret;
}

int main(int argc, char** argv)
{
   HD_RESULT ret;
   INT key, i, value;
    pthread_t enc_thread_id;
    AUDIO_RECORD  rec_info = {0};

   if (argc == 2) {
    value = atoi(argv[1]);
        if (value >= 0 && value <= 1) {
            printf(" Input devid(%d)\n", value);
        } else {
            printf("Wrong argument, in_dev range(0~2)\n");
            exit(0);
        }
   } else {
    value = 0;
   }
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
    //set default in-dev port
    rec_info.dev_idx = value;

    // init hdal
      ret = hd_common_init(1);
    if(ret != HD_OK) {
        printf("common init fail\n");
        goto exit;
    }

    //init audiocap and audioenc modules
    ret = init_module();
    if(ret != HD_OK) {
        printf("init fail\n");
        goto exit;
    }

    //open audiocap and audioenc modules
    ret = open_module(&rec_info);
    if(ret != HD_OK) {
        printf("open fail\n");
        goto exit;
    }

    //setup runtime parameters
    for (i = 0; i < MAX_BITSTREAM_NUM; i++) {
           ret = set_param(&rec_info, i);
           if(ret != HD_OK) {
                printf("set param fail\n");
                goto exit;
           }
    }

    //bind audio record: audiocap -> audioenc
    for (i = 0; i < MAX_BITSTREAM_NUM; i++) {
           ret = hd_audiocap_bind(HD_AUDIOCAP_OUT(rec_info.dev_idx, i), HD_AUDIOENC_IN(0,
i));
           if(ret != HD_OK) {
                printf("bind fail\n");
                goto exit;
           }
    }

      //start to run
    for (i = 0; i < MAX_BITSTREAM_NUM; i++) {
           ret = hd_audiocap_start(rec_info.audcap_path_id[i]);
          if(ret != HD_OK) {
              printf("start audiocap fail\n");
              goto exit;
          }
           ret = hd_audioenc_start(rec_info.audenc_path_id[i]);
          if(ret != HD_OK) {
              printf("start audioenc fail\n");
              goto exit;
          }
    }
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

```
    //create a thread to put audio data
      ret = pthread_create(&enc_thread_id, NULL, encode_thread, (void *)&rec_info);
      if (ret < 0) {
           printf("create encode thread failed");
           return -1;
      }

      //main waiting loop
      printf("Enter q to exit\n");
      while (1) {
           key = getchar();
           if (key == 'q') {
                rec_info.enc_exit = 1;
                break;
           } else if (key == 'd') {
                hd_debug_run_menu();

           }
      }

      //wait thread finish
      pthread_join(enc_thread_id, NULL);

      //stop modules and unbind the connection
    for (i = 0; i < MAX_BITSTREAM_NUM; i++) {
      ret = hd_audiocap_stop(rec_info.audcap_path_id[i]);
           if(ret != HD_OK) {
                printf("stop audiocap fail\n");
           }
           ret = hd_audioenc_stop(rec_info.audenc_path_id[i]);
           if(ret != HD_OK) {
                printf("stop audioenc fail\n");
           }
           ret = hd_audiocap_unbind(HD_AUDIOCAP_OUT(rec_info.dev_idx, i));
           if(ret != HD_OK) {
                printf("unbind fail\n");
           }
    }

exit:
      //close and uninit modules
      ret = close_module(&rec_info);
      if(ret != HD_OK) {
           printf("close fail\n");
      }
      ret = exit_module();
      if(ret != HD_OK) {
           printf("exit fail\n");
      }
      ret = hd_common_uninit();
    if(ret != HD_OK) {
       printf("uninit fail\n");
    }
      return 0
```

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and doesnot assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.

NOVATEK CONFIDENTIAL

Copyright © 2018 Novatek Microelectronics Corp. All Rights Reserved.
With respect to the information represented in this document, Novatek makes no warranty, expressed or implied, including the warranties of merchantability, fitness for a particular purpose and non-infringement, and does not assume any legal liability or responsibility for the accuracy, completeness or usefulness of any such information.