



# ***Novatek HDAL Design Specification - hd\_logger***

---

**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

NT9668x/NT98313 Design Specification - hd_logger.....		1
1	Introduction .....	3
2	Function and data structure definition .....	4
2.1	Data structure definition for hd_logger .....	4
2.2	Function definition for hd_logger .....	4
2.2.1	Global function.....	4
2.2.2	Macros .....	4
2.3	Control flow .....	7
3	Use cases .....	8
3.1	Example.....	8

# 1 Introduction

The hd\_logger provides printing debug message for all HDAL modules. Here provides several types message for modules, list as below:

- **\_ERR**: error message including prompt names of module, function and line number. (default: on)
- **\_WRN**: warning message including prompt names of module, function and line number. (default: on)
- **\_IND**: verbose message including prompt names of module, function and line number. (default: off)
- **\_MSG**: hints message such as a menu list not including prompt names of module, function and line number. (default: on)
- **\_FUNC**: only use to trace at the begging and the end of function. (default: off)

The hd\_logger allows user hook their print function by using the definition of **PRINT\_FUNC\_USER**.

## 2 Function and data structure definition

### 2.1 Data structure definition for hd\_logger

N/A

### 2.2 Function definition for hd\_logger

#### 2.2.1 Global function

```
void hd_printf(const char *msg_with_format, ...);
```

**Description:**

The raw debug message output of HDAL

**Param:**

msg\_with\_format: formatted string

**Return value:**

None

#### 2.2.2 Macros

module	macros
hd_audiocapture	HD_AUDIOCAPTURE_ERR(fmtstr, args...) HD_AUDIOCAPTURE_WRN(fmtstr, args...) HD_AUDIOCAPTURE_IND(fmtstr, args...) HD_AUDIOCAPTURE_MSG(fmtstr, args...) HD_AUDIOCAPTURE_FUNC(fmtstr, args...)
hd_audioout	HD_AUDIOOUT_ERR(fmtstr, args...) HD_AUDIOOUT_WRN(fmtstr, args...) HD_AUDIOOUT_IND(fmtstr, args...) HD_AUDIOOUT_MSG(fmtstr, args...) HD_AUDIOOUT_FUNC(fmtstr, args...)
hd_audioenc	HD_AUDIOENC_ERR(fmtstr, args...) HD_AUDIOENC_WRN(fmtstr, args...) HD_AUDIOENC_IND(fmtstr, args...)

	HD_AUDIOENC_MSG(fmtstr, args...) HD_AUDIOENC_FUNC(fmtstr, args...)
hd_audiocapture	HD_AUDIOCAPTURE_ERR(fmtstr, args...) HD_AUDIOCAPTURE_WRN(fmtstr, args...) HD_AUDIOCAPTURE_IND(fmtstr, args...) HD_AUDIOCAPTURE_MSG(fmtstr, args...) HD_AUDIOCAPTURE_FUNC(fmtstr, args...)
hd_audiodec	HD_AUDIODEC_ERR(fmtstr, args...) HD_AUDIODEC_WRN(fmtstr, args...) HD_AUDIODEC_IND(fmtstr, args...) HD_AUDIODEC_MSG(fmtstr, args...) HD_AUDIODEC_FUNC(fmtstr, args...)
hd_videocapture	HD_VIDEOCAPTURE_ERR(fmtstr, args...) HD_VIDEOCAPTURE_WRN(fmtstr, args...) HD_VIDEOCAPTURE_IND(fmtstr, args...) HD_VIDEOCAPTURE_MSG(fmtstr, args...) HD_VIDEOCAPTURE_FUNC(fmtstr, args...)
hd_videoout	HD_VIDEOOUT_ERR(fmtstr, args...) HD_VIDEOOUT_WRN(fmtstr, args...) HD_VIDEOOUT_IND(fmtstr, args...) HD_VIDEOOUT_MSG(fmtstr, args...) HD_VIDEOOUT_FUNC(fmtstr, args...)
hd_videoprocess	HD_VIDEOPROCESS_ERR(fmtstr, args...) HD_VIDEOPROCESS_WRN(fmtstr, args...) HD_VIDEOPROCESS_IND(fmtstr, args...) HD_VIDEOPROCESS_MSG(fmtstr, args...) HD_VIDEOPROCESS_FUNC(fmtstr, args...)
hd_videoenc	HD_VIDEOENC_ERR(fmtstr, args...) HD_VIDEOENC_WRN(fmtstr, args...) HD_VIDEOENC_IND(fmtstr, args...) HD_VIDEOENC_MSG(fmtstr, args...) HD_VIDEOENC_FUNC(fmtstr, args...)
hd_videodec	HD_VIDEODEC_ERR(fmtstr, args...) HD_VIDEODEC_WRN(fmtstr, args...) HD_VIDEODEC_IND(fmtstr, args...) HD_VIDEODEC_MSG(fmtstr, args...) HD_VIDEODEC_FUNC(fmtstr, args...)

hd_gfx	HD_GFX_ERR(fmtstr, args...) HD_GFX_WRN(fmtstr, args...) HD_GFX_IND(fmtstr, args...) HD_GFX_MSG(fmtstr, args...) HD_GFX_FUNC(fmtstr, args...)
hd_common	HD_COMMON_ERR(fmtstr, args...) HD_COMMON_WRN(fmtstr, args...) HD_COMMON_IND(fmtstr, args...) HD_COMMON_MSG(fmtstr, args...) HD_COMMON_FUNC(fmtstr, args...)
hd_util	HD_UTIL_ERR(fmtstr, args...) HD_UTIL_WRN(fmtstr, args...) HD_UTIL_IND(fmtstr, args...) HD_UTIL_MSG(fmtstr, args...) HD_UTIL_FUNC(fmtstr, args...)
hd_debug	HD_DEBUG_ERR(fmtstr, args...) HD_DEBUG_WRN(fmtstr, args...) HD_DEBUG_IND(fmtstr, args...) HD_DEBUG_MSG(fmtstr, args...) HD_DEBUG_FUNC(fmtstr, args...)

**Description:**

Each level debug message for each module

**Param:**

formatted string

**Return value:**

None

## 2.3 Control flow

All of HDAL use the macro provided by hd\_logger to print debug message.

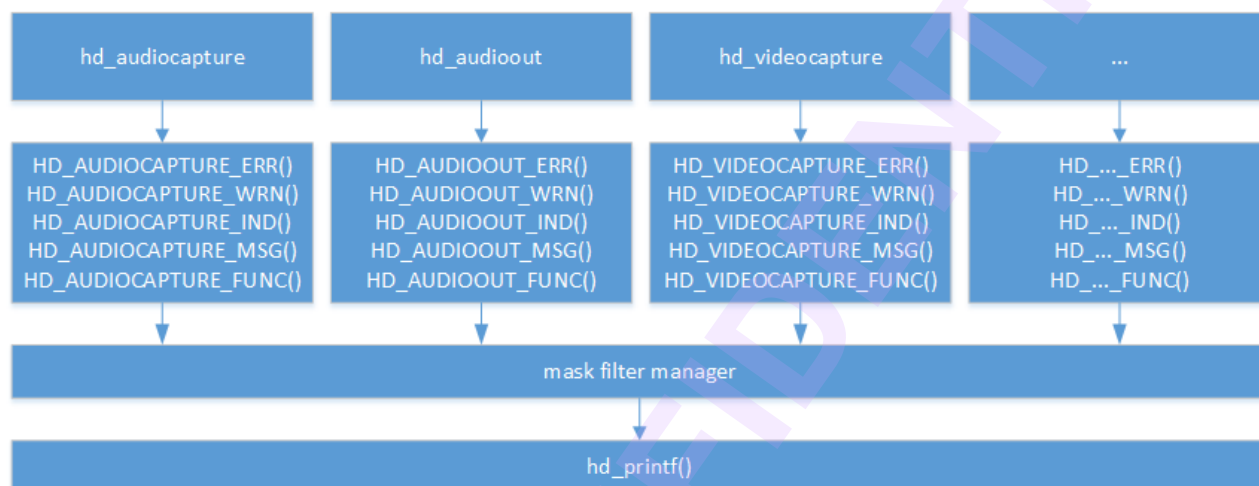


Figure 2-1 control flow of debug macro

## 3 Use cases

To simplify the programming, we suggest using an internal header for a module and make a copy of definition as following.

```
#define HD_MODULE_NAME HD_AUDIOCAPTURE

#define DBG_ERR(fmtstr, args...) HD_LOG_BIND(HD_MODULE_NAME, _ERR)("\033[1;31m" fmtstr "\033[0m", ##args)

#define DBG_WRN(fmtstr, args...) HD_LOG_BIND(HD_MODULE_NAME, _WRN)("\033[1;33m" fmtstr "\033[0m", ##args)

#define DBG_IND(fmtstr, args...) HD_LOG_BIND(HD_MODULE_NAME, _IND)(fmtstr, ##args)

#define DBG_DUMP(fmtstr, args...) HD_LOG_BIND(HD_MODULE_NAME, _MSG)(fmtstr, ##args)

#define DBG_FUNC_BEGIN(fmtstr, args...) HD_LOG_BIND(HD_MODULE_NAME, _FUNC)("BEGIN: " fmtstr, ##args)

#define DBG_FUNC_END(fmtstr, args...) HD_LOG_BIND(HD_MODULE_NAME, _FUNC)("END: " fmtstr, ##args)
```

After making a copy, just replace HD\_AUDIOCAPTURE with another module name defined in HD\_LOG\_MASK\_ (e.g. HD\_AUDIOOUT).

### 3.1 Example

Following is an example to explain how to use the macros as stated above.

```
int foo(void)
{
    DBG_FUNC_BEGIN("with init val=%d\n", 0);
    DBG_ERR("this is error message, er=%d\n", -1);
    DBG_WRN("this is warning message, wr=%d\n", -2);
    DBG_IND("this is debug message, got val=%d\n", 3);
    DBG_DUMP("this is normal message without function name, module name and line number.\n");
    DBG_FUNC_END("with end val=%d\n", 99);
    return 0;
}
```

The result shows as below.

```
--- [AUDIOCAPTURE][foo()] BEGIN: with init val=0
ERR: [AUDIOCAPTURE][foo():156]: this is error message, er=-1 (with red color)
```

**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.



WRN: [AUDIOCAPTURE][foo():157]: this is warning message, wr=-2 (with yellow color)

IND: [AUDIOCAPTURE][foo():158]: this is debug message, got val=3

This is normal message without function name, module name and line number.

--- [AUDIOCAPTURE][foo()] END: with end val=99