

GM8136

# CAPTURE

---

User Guide

Rev.: 1.0

Issue Date: August 2014





# REVISION HISTORY

## GM8136 Capture User Guide

Date	Rev.	From	To
Aug. 2014	1.0	-	Original

Copyright © 2014 Grain Media, Inc.

All Rights Reserved.

Printed in Taiwan 2014

Grain Media and the Grain Media Logo are trademarks of Grain Media, Inc. in Taiwan and/or other countries. Other company, product and service names may be trademarks or service marks of others.

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in implantation or other life support application where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Grain Media's product specification or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Grain Media or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will Grain Media be liable for damages arising directly or indirectly from any use of the information contained in this document.

Grain Media, Inc.  
5F, No. 5, Li-Hsin Road III, Hsinchu Science Park, Hsinchu City, Taiwan 300, R.O.C.

Grain Media's home page can be found at:  
<http://www.grain-media.com>



# TABLE OF CONTENTS

Chapter 1	Introduction.....	1
	1.1 Overview.....	2
	1.2 Features .....	2
	1.3 Block Diagram .....	3
	1.4 ITU-R BT.656, BT.1120, and BT.601 Formats.....	4
Chapter 2	Crop.....	5
Chapter 3	Scaler .....	7
Chapter 4	Mask.....	9
Chapter 5	OSD Windows .....	11
	5.1 Features .....	12
	5.2 OSD Font .....	13
Chapter 6	Mark .....	17
	6.1 Features .....	18
Chapter 7	Motion Detection .....	19
Chapter 8	Capture Driver Module .....	21
Chapter 9	Capture Module Parameters .....	23
	9.1 vcap300_common Module .....	24
	9.2 vcap0 Module .....	24
	9.3 Input Modules.....	29
Chapter 10	Capture Proc Nodes.....	61
	10.1 /proc/vcap300/version Proc Node .....	64
	10.2 /proc/vcap300/vcap0 Proc Nodes .....	64
	10.3 /proc/vcap300/vcap0/fcs Proc Nodes.....	74
	10.4 /proc/vcap300/vcap0/denoise Proc Nodes.....	76
	10.5 /proc/vcap300/vcap0/sharpness Proc Nodes .....	78
	10.6 /proc/vcap300/vcap0/md Proc Nodes .....	80
	10.7 /proc/vcap300/vcap0/cfg Proc Nodes .....	84
	10.8 /proc/vcap300/vcap0/presmooth Proc Nodes .....	95
	10.9 /proc/vcap300/input_module Proc Nodes .....	96

	10.10 /proc/vcap300/vcap0/diagnostic Proc Nodes .....	98
	10.11 /proc/vcap300/vcap0/dbg_mode Proc Node .....	105
	10.12 /proc/vcap300/vcap0/crop_rule Proc Node .....	106
	10.13 /proc/vcap300/vcap0/grab_filter Proc Node .....	106
	10.14 /proc/vcap300/vcap0/vi_probe Proc Node .....	107
Chapter 11	Front-end Driver Module .....	109
	11.1 TW2968 Driver Module .....	111
	11.2 NVP1118 Driver Module.....	123
	11.3 NVP1918 Driver Module.....	133
	11.4 CX26848 Driver Module .....	143
	11.5 TW9900 Driver Module .....	152
	11.6 TW2868 Driver Module .....	158
	11.7 TW2964 Driver Module .....	169
	11.8 NVP1918C Driver Module.....	180
	11.9 NVP1914 Driver Module.....	191
	11.10 NVP1914C Driver Module.....	201
	11.11 RN6318 Driver Module.....	211
	11.12 RN6314 Driver Module.....	222
	11.13 CX25930 Driver Module.....	233
	11.14 MV4101 Driver Module.....	239
	11.15 DH9901 Driver Module.....	245
	11.16 NVP6114 Driver Module.....	254
	11.17 TP2802 Driver Module .....	265
	11.18 MT9M131 Driver Module.....	269
	11.19 MT9D131 Driver Module .....	271
	11.20 HM1375 Driver Module .....	273
	11.21 OV7725 Driver Module.....	275
Chapter 12	Front-end and VCAP300 Input_Module Driver Porting.....	277
	12.1 Video Channel Mapping .....	280
	12.2 Watch Dog Thread .....	281
	12.3 Summary of Driver Porting .....	282

# LIST OF TABLES

Table 2-1.	Cropping Capabilities .....	5
Table 4-1.	Mask Capabilities .....	9
Table 5-1.	OSD Capabilities .....	12
Table 5-2.	Default Font Characters in OSD Font RAM .....	14
Table 6-1.	Mark Capabilities .....	18
Table 9-1.	Parameters of vcap0 Module .....	24
Table 9-2.	Parameters of vcap300_generic_mX Module .....	29
Table 9-3.	Parameters of vcap300_nvp1118 Module .....	32
Table 9-4.	Parameters of vcap300_nvp1918 Module .....	33
Table 9-5.	Parameters of vcap300_tw2968 Module .....	34
Table 9-6.	Parameters of vcap300_tw9900 Module .....	36
Table 9-7.	Parameters of vcap300_cx26848 Module .....	37
Table 9-8.	Parameters of vcap300_isp Module .....	38
Table 9-9.	Parameters of vcap300_hm1375 Module .....	39
Table 9-10.	Parameters of vcap300_mt9m131 Module .....	40
Table 9-11.	Parameters of vcap300_mt9d131 Module .....	41
Table 9-12.	Parameters of vcap300_ov7725 Module .....	43
Table 9-13.	Parameters of vcap300_nvp1918c Module .....	44
Table 9-14.	Parameters of vcap300_nvp1914 Module .....	45
Table 9-15.	Parameters of vcap300_nvp1914C Module .....	46
Table 9-16.	Parameters of vcap300_tw2964 Module .....	48
Table 9-17.	Parameters of vcap300_tw2868 Module .....	49
Table 9-18.	Parameters of vcap300_cx26848 Module .....	50
Table 9-19.	Parameters of vcap300_rn6318 Module .....	52
Table 9-20.	Parameters of vcap300_rn6314 Module .....	53
Table 9-21.	Parameters of vcap300_cx25930 Module .....	54
Table 9-22.	Parameters of vcap300_mv4101 Module .....	56
Table 9-23.	Parameters of vcap300_dh9901 Module .....	57
Table 9-24.	Parameters of vcap300_nvp6114 Module .....	58

Table 9-25.	Parameters of vcap300_tp2802 Module .....	59
Table 11-1.	Parameters of TW2968 Module .....	111
Table 11-2.	Parameters of NVP1118 Module .....	123
Table 11-3.	Parameters of NVP1918 Module .....	133
Table 11-4.	Parameters of CX26848 Module .....	143
Table 11-5.	Parameters of TW9900 Module .....	152
Table 11-6.	Parameters of TW2868 Module .....	158
Table 11-7.	Parameters of TW2964 Module .....	169
Table 11-8.	Parameters of NVP1918C Module .....	180
Table 11-9.	Parameters of NVP1914 Module .....	191
Table 11-10.	Parameters of NVP1914C Module .....	201
Table 11-11.	Parameters of RN6318 Module .....	211
Table 11-12.	Parameters of RN6314 Module .....	222
Table 11-13.	Parameters of CX25930 Module .....	233
Table 11-14.	Parameters of MV4101 Module .....	239
Table 11-15.	Parameters of DH9901 Module .....	245
Table 11-16.	Parameters of NVP6114 Module .....	254
Table 11-17.	Parameters of TP2802 Module .....	265
Table 11-18.	Parameters of MT9M131 Module .....	269
Table 11-19.	Parameters of MT9D131 Module .....	271
Table 11-20.	Parameters of HM1375 Module .....	273
Table 11-21.	Parameters of HM1375 Module .....	275



# LIST OF FIGURES

Figure 1-1.	Block Diagram of FTVCAP300.....	3
Figure 1-2.	Data and Clock in ITU-R BT.656, BT.1120, and BT.601 Formats.....	4
Figure 5-1.	12x18 OSD Font Bitmap .....	13
Figure 5-2.	OSD Window and Font Color.....	14
Figure 7-1.	128x128 Motion Blocks Mapping on One Image .....	20
Figure 9-1.	VI Bypass Mode .....	26
Figure 9-2.	VI 2CH Byte Interleave Mode.....	27
Figure 9-3.	VI 4CH Byte/Frame Interleave Mode .....	28
Figure 10-1.	Proc Nodes of Capture Module.....	62
Figure 11-1.	Proc Nodes of TW2968 Module .....	113
Figure 11-2.	Proc Nodes of NVP1118 Module .....	125
Figure 11-3.	Proc Nodes of NVP1918 Module .....	135
Figure 11-4.	Proc Nodes of CX26848 Module.....	145
Figure 11-5.	Proc Nodes of TW9900 Module .....	153
Figure 11-6.	Proc Nodes of TW2868 Module .....	160
Figure 11-7.	Proc Nodes of TW2964 Module .....	171
Figure 11-8.	Proc Nodes of NVP1918C Module.....	183
Figure 11-9.	Proc Nodes of NVP1914 Module .....	193
Figure 11-10.	Proc Nodes of NVP1914C Module.....	203
Figure 11-11.	Proc Nodes of RN6318 Module .....	213
Figure 11-12.	Proc Nodes of RN6314 Module .....	224
Figure 11-13.	Proc Nodes of CX25930 Module.....	235
Figure 11-14.	Proc Nodes of MV4101 Module .....	241
Figure 11-15.	Proc Nodes of DH9901 Module .....	247
Figure 11-16.	Proc Nodes of NVP6114 Module .....	257
Figure 11-17.	Proc Nodes of TP2802 Module .....	266
Figure 11-18.	Proc Nodes of MT9M131 Module .....	270
Figure 11-19.	Proc Nodes of MT9D131 Module.....	272
Figure 11-20.	Proc Nodes of HM1375 Module .....	274

Figure 11-21.	Proc Nodes of OV7725 Module .....	276
Figure 12-1.	Driver Architecture of NVP1914 and VCAP300_NVP1914 .....	279
Figure 12-2.	Mapping Table for NVP1914 Video Channel.....	280
Figure 12-3.	Mapping Table for OV7725 + MT9M136 Video Channel .....	281

# Chapter 1

## Introduction

---

This chapter contains the following sections:

- 1.1 Overview
- 1.2 Features
- 1.3 Block Diagram
- 1.4 ITU-R BT.656, BT.1120, and BT.601 Formats

## 1.1 Overview

FTVCAP300 in GM8136 is used to capture the video data from various video interfaces and outputs data through AMBA AXI to a memory. It provides two video input interfaces. The first video interface supports the ITU-R BT.656, BT.1120, and BT.601 format. The second video input interface only supports ISP data format. Each channel can output four resolutions through the linear scaling-down/up function to output various image sizes. The OSD function can help users paste any character on the output image. The mark function can help users paste watermark or logo on the output image. The motion detection function can be applied to various surveillance systems.

## 1.2 Features

GM8136 capture contains the following features:

- Supports maximum capture resolution of 4096 x 4096
- Supports ITU-R BT.656 8-bit and BT.1120 16-bit formats in first video input interface
- Supports ITU-R BT.601 8-bit and BT.601 16-bit formats in first video input interface
- Supports ITU-R BT.656 8-bit and ISP format in second video input interface
- Supports byte/frame interleave mode in first input interface
- Supports embedded de-multiplexer function in first video input interface
- Supports linear scaling-down/up function and four output resolutions in each channel
  - Scaler#0/1/2 have scaling capability
  - Scaler#3 only has bypass capability
- Supports front-end image source cropping function
- Supports back-end image cropping function for four output resolutions per channel
- Supports eight mask windows in each channel with transparency control
- Supports eight font-based OSD windows in each channel with transparency control
- Supports four image mark windows in each channel with transparency control
- Supports OSD border and marquee functions for four output resolutions per channel
- Supports 1-D de-noise
- Supports 1-D false color suppression
- Supports line sharpness function for four output resolutions per channel
- Supports motion detection function
- Supports VBI detection and extraction
- Uses YCbCr 4:2:2 data output format

## 1.3 Block Diagram

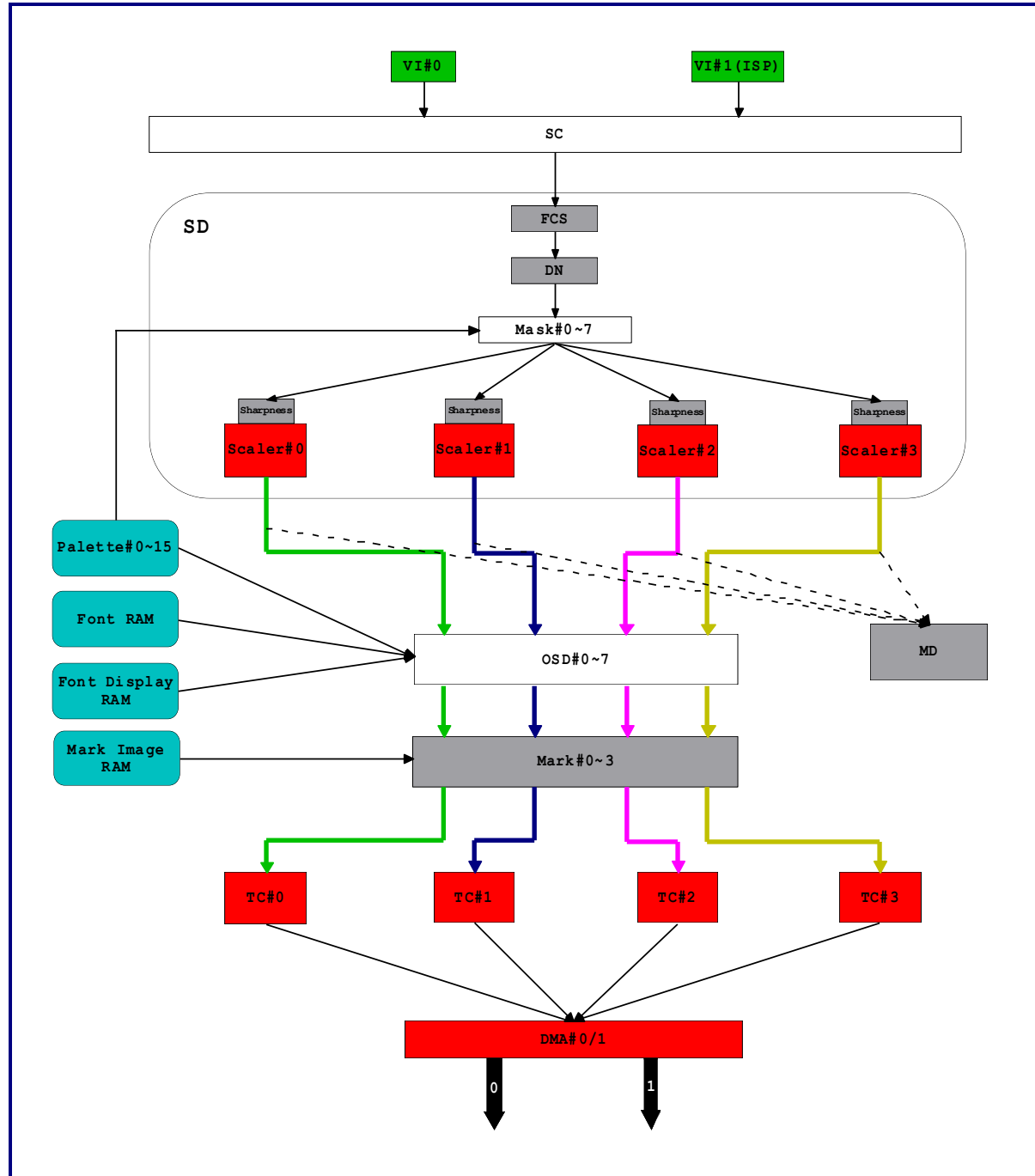


Figure 1-1. Block Diagram of FTVCAP300

## 1.4 ITU-R BT.656, BT.1120, and BT.601 Formats

In GM8136, the FTVCAP300 video interfaces support the standard ITU-R BT.656, BT.1120, and BT.601 formats. FTVCAP300 supports reference H/V from the external signal. Figure 1-2 shows the data and clock in the ITU-R BT.656, BT.1120, and BT.601 formats.

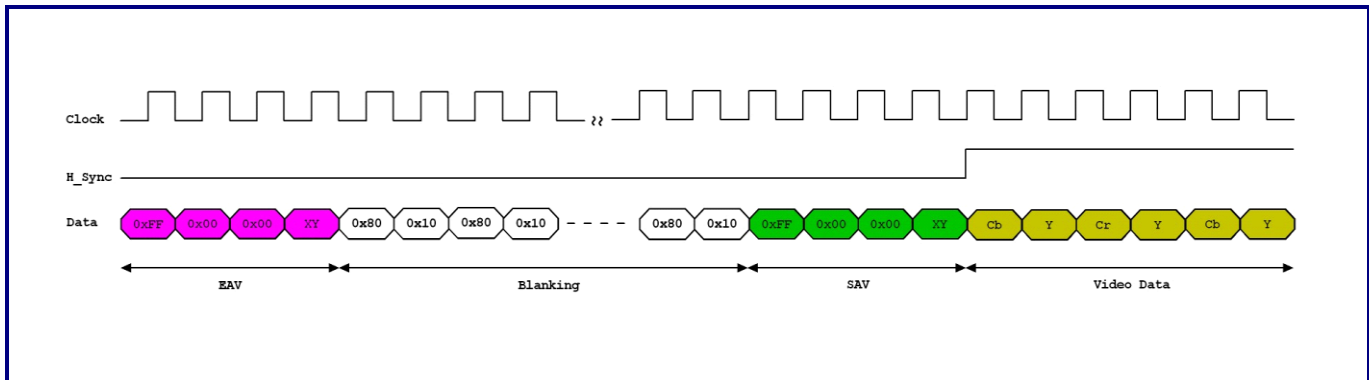


Figure 1-2. Data and Clock in ITU-R BT.656, BT.1120, and BT.601 Formats

# Chapter 2

## Crop

---

The capture supports the crop function for cropping an interesting region of an image. The Source Cropping (SC) engine is the first layer of each channel. The Target Cropping (TC) engine is the second layer for four output resolution paths of each channel. Users can specify a crop region through the middleware control properties of each channel. Table 2-1 shows the cropping capabilities.

**Table 2-1. Cropping Capabilities**

FTVCAP300		Source Cropping	Resolution	Target Cropping
VI0 ~ 1	Channel 0	Yes	Path 0	Yes
			Path 1	Yes
			Path 2	Yes
			Path 3	Yes
	Channel 1	Yes	Path 0	Yes
			Path 1	Yes
			Path 2	Yes
			Path 3	Yes
	Channel 2	Yes	Path 0	Yes

FTVCAP300	Source Cropping	Resolution	Target Cropping
Channel 3	Yes	Path 1	Yes
		Path 2	Yes
		Path 3	Yes
		Path 0	Yes
		Path 1	Yes
		Path 2	Yes
		Path 3	Yes



# Chapter 3

## Scaler

---

In GM8136, FTVCAP300 supports the linear scaling function for each channel to output four resolutions. One image frame can input and output four resolutions at the same time. The scaler engine supports linear scaling-down and scaling-up. The total output widths of four resolution paths must be less than 4096 pixels for the capability of the scaler hardware. The fourth scaler engine does not support the scaling capability.

The scaler engine contains the following features:

- Supports three scaling and one bypass output paths
- Supports sharpness
- Supports linear scaling-down and scaling-up
- Supports frame-to-field output mode
- Supports progressive to interlace (60P to 60I) output mode



# Chapter 4

## Mask

---

In FTVCAP300, each channel supports eight mask windows. Users can apply these mask windows to mask any image region in each channel. Table 4-1 shows the mask capabilities. Because the mask window applies on any channel, all output paths in a channel will display same region of the same mask window.

**Table 4-1. Mask Capabilities**

FTVCAP300		Window Number	Window color
VI0 ~ VI1	Channel 0	8	16 palette colors shared by all channels
	Channel 1	8	
	Channel 2	8	
	Channel 3	8	

In GM8136, FTVCAP300 supports 16 programmable palette colors, which are used for the font/border/background colors and mask window color. Users can customize the palette colors through middleware API. The format is YCbCr422.

The mask contains the following features:

- Supports eight mask windows in a frame for each channel
- Supports hollow function with maximum border size of 32 pixels
- Supports eight window transparencies:  
0%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5%, and 100%

# Chapter 5

## OSD Windows

---

This chapter contains the following sections:

- 5.1 Features
- 5.2 OSD Font

In FTVCAP300, each channel supports eight OSD windows. Users can specify these OSD windows to display the specified output path of each channel. Four output paths in each channel share these OSD windows. Table 5-1 shows the OSD capabilities.

**Table 5-1. OSD Capabilities**

FTVCAP300		Font RAM	Display RAM	Font Number	Window Number
VI0 ~ VI1	Channel 0	8192 x 12 (Bit)	1024 x 9 (Bit)	455	8
	Channel 1	All channels shared	All channels shared	All channels shared	8
	Channel 2				8
	Channel 3				8

## 5.1 Features

OSD contains the following features:

- Supports 455 user-programmable fonts
- Supports font size of 12x18
- Supports 16 programmable palette colors
- Supports eight OSD windows in a frame for each channel
- Maximum 64 fonts in horizontal or vertical direction, and maximum 256 fonts in a font window
- Supports zoom-in with locked aspect ratio: x2, x3, and x4
- Supports zoom-in without locked aspect ratio (Horizontal/Vertical):  
x2/x1, x4/x1, x4/x2, x1/x2, x1/x4, and x2/x4
- Supports eight font/background transparencies:  
0%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5%, and 100%
- Supports programmable 16-type background color
- Supports border function with 16-type border color
- Supports three marquee modes:
  - One horizontal-line marquee effect
  - One vertical-line marquee effect
  - One horizontal-line flip effect
- Supports programmable border size (Maximum of 32 pixels)

5.2 OSD Font

The unit of fonts stored in the OSD Font RAM is 12x18. The OSD window adjusts the row and column spaces to control the space between fonts. Figure 5-1 shows an example of the font bitmap for the character of "A". Users can customize the font bitmap and add/replace the new font bitmap to capture the internal OSD Font RAM through the middleware API. Figure 5-2 shows an example of font and window color relationship.

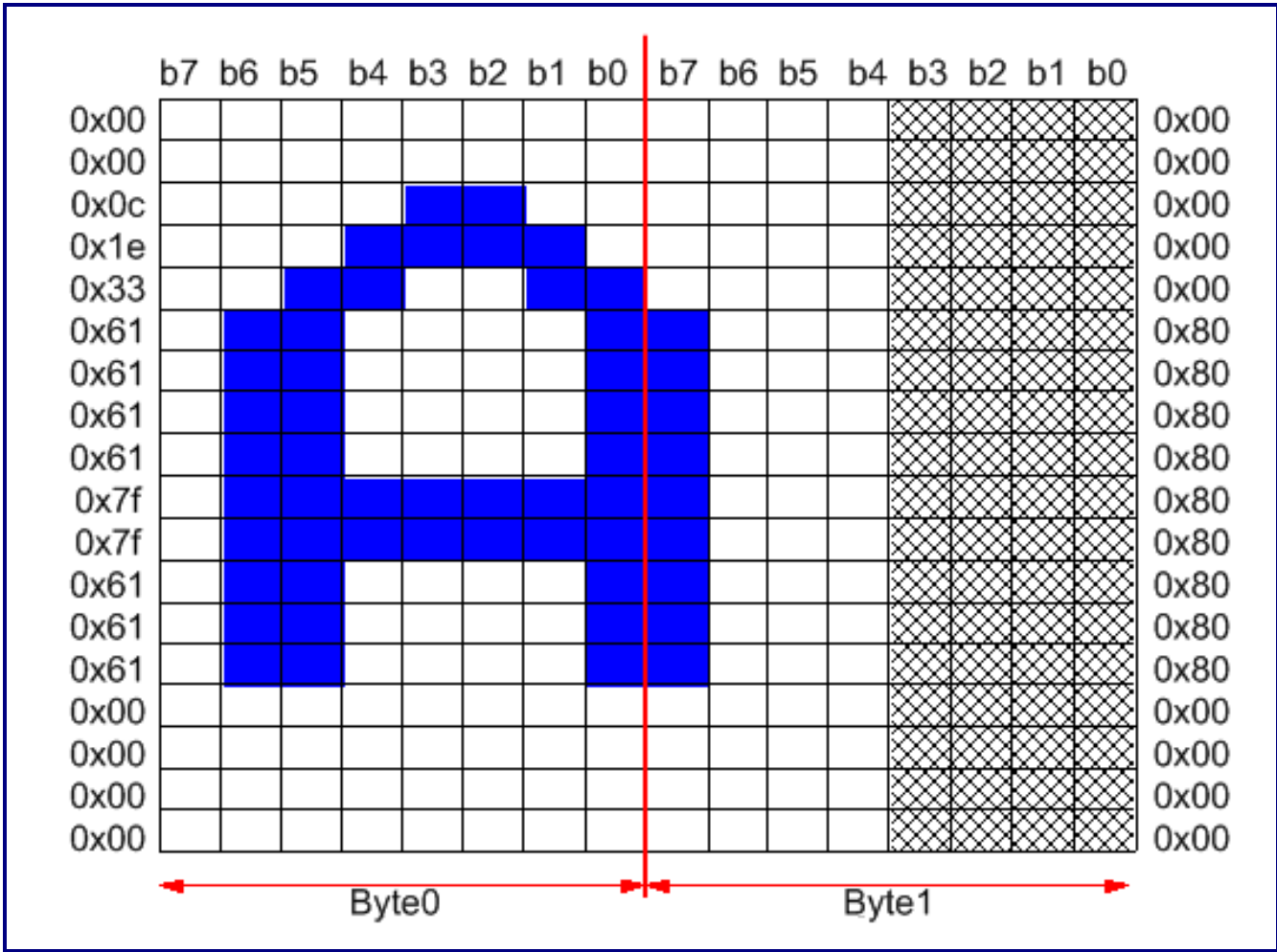


Figure 5-1. 12x18 OSD Font Bitmap

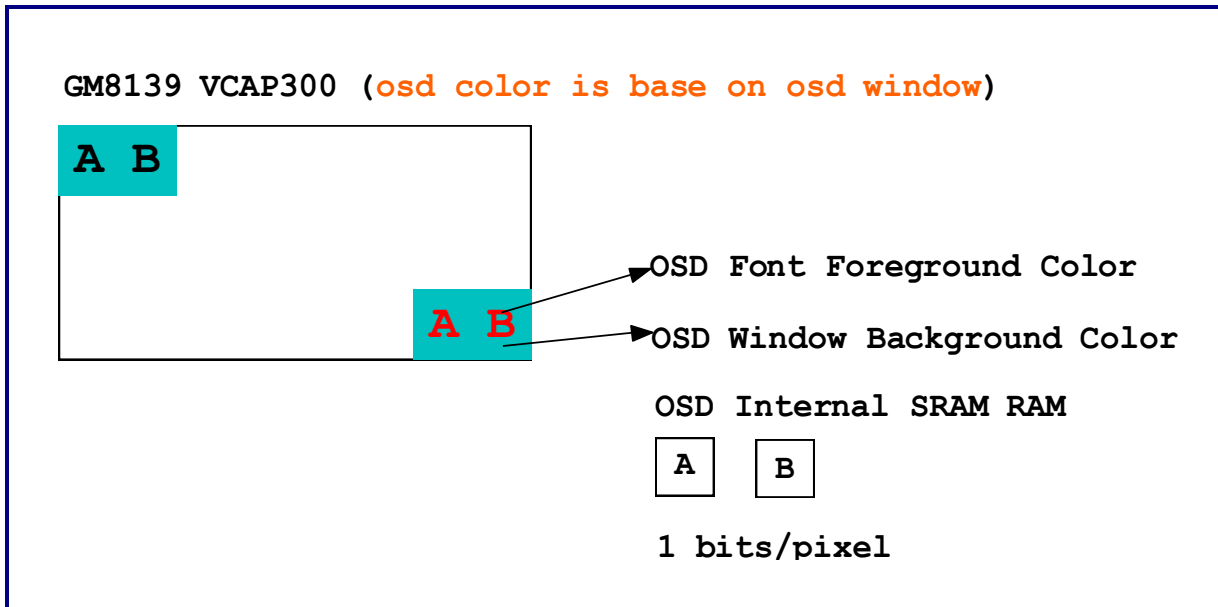


Figure 5-2. OSD Window and Font Color

Table 5-2 lists the default font characters in capturing OSD Font RAM. All font indexes are the same as the standard ASCII table.

Table 5-2. Default Font Characters in OSD Font RAM

Font Character	Font Index	Font Character	Font Index	Font Character	Font Index
space	0x20	@	0x40	`	0x60
!	0x21	A	0x41	a	0x61
"	0x22	B	0x42	b	0x62
#	0x23	C	0x43	c	0x63
\$	0x24	D	0x44	d	0x64
%	0x25	E	0x45	e	0x65
&	0x26	F	0x46	f	0x66
'	0x27	G	0x47	g	0x67
(	0x28	H	0x48	h	0x68
)	0x29	I	0x49	i	0x69
*	0x2a	J	0x4a	j	0x6a
+	0x2b	K	0x4b	k	0x6b
,	0x2c	L	0x4c	l	0x6c



Font Character	Font Index	Font Character	Font Index	Font Character	Font Index
-	0x2d	M	0x4d	m	0x6d
.	0x2e	N	0x4e	n	0x6e
/	0x2f	O	0x4f	o	0x6f
0	0x30	P	0x50	p	0x70
1	0x31	Q	0x51	q	0x71
2	0x32	R	0x52	r	0x72
3	0x33	S	0x53	s	0x73
4	0x34	T	0x54	t	0x74
5	0x35	U	0x55	u	0x75
6	0x36	V	0x56	v	0x76
7	0x37	W	0x57	w	0x77
8	0x38	X	0x58	x	0x78
9	0x39	Y	0x59	y	0x79
:	0x3a	Z	0x5a	z	0x7a
;	0x3b	[	0x5b	{	0x7b
<	0x3c	\	0x5c		0x7c
=	0x3d	]	0x5d	}	0x7d
>	0x3e	^	0x5e	~	0x7e
?	0x3f	_	0x5f	-	-



# Chapter 6

## Mark

---

This chapter contains the following section:

- 6.1 Features

In FTVCAP300, each channel supports four mark windows. Users can specify these mark windows to display specific output path of each channel. The output paths in each channel share these mark windows. Users can use the mark windows to display the company logo or watermark. Table 6-1 shows the mark capabilities.

**Table 6-1. Mark Capabilities**

FTVCAP300		Mark RAM	Window Number
VI0 ~ VI1	Channel 0	1024 x 64 (Bit)	4
	Channel 1	All channels shared	4
	Channel 2		4
	Channel 3		4

## 6.1 Features

Mark contains the following features:

- Supports four mark windows in a frame for each channel
- Supports data format: YCbCr422 (16-bit per pixel)
- Flexible allocation on 4096-pixel RAM size
- Supports flexible dimension on horizontal and vertical directions:  
8-pixel, 16-pixel, 32-pixel, 64-pixel, 128-pixel, and 256-pixel
- Supports zoom-in with locked aspect ratios: x2 and x4
- Selectable overlapping conditions between font and mark:
  - Mark above font
  - Mark below font
- Eight mark window transparencies:  
0%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5%, and 100%

# Chapter 7

## Motion Detection

---

FTVCAP300 supports the motion detection function for various surveillance applications. Users can split the 128 (Row)x128 (Col.) motion block and get the motion event to determine which block has motion. The event value, [0]foreground/[1]background, reflects the status of each motion block. Figure 7-1 shows the motion blocks mapping on an image. Users can refer to the below suggested parameter to adjust the efficiency of the motion detection.

Parameter	Description
alpha	Control the updated speed of the MD model, if the luminance variance of background is significant, users should select bigger value for alpha.
tb	Decide if this MB belongs to the background or foreground, small value of tb with sensitive MD
sigma	Control the noise tolerance ability of the MD model, large value with strong noise tolerance ability
alpha accuracy	Control the update speed of the MD weight, alpha accuracy with large value increases the update speed of the MD weight which can increase the model update speed when the background is changed. The value of the alpha accuracy is formulated as $A * 8191$ , users can choose the suitable value for variable A.

Parameter	Description
tg	Decide if this MB belongs to the background or foreground, the value of tg must be the same as tb.

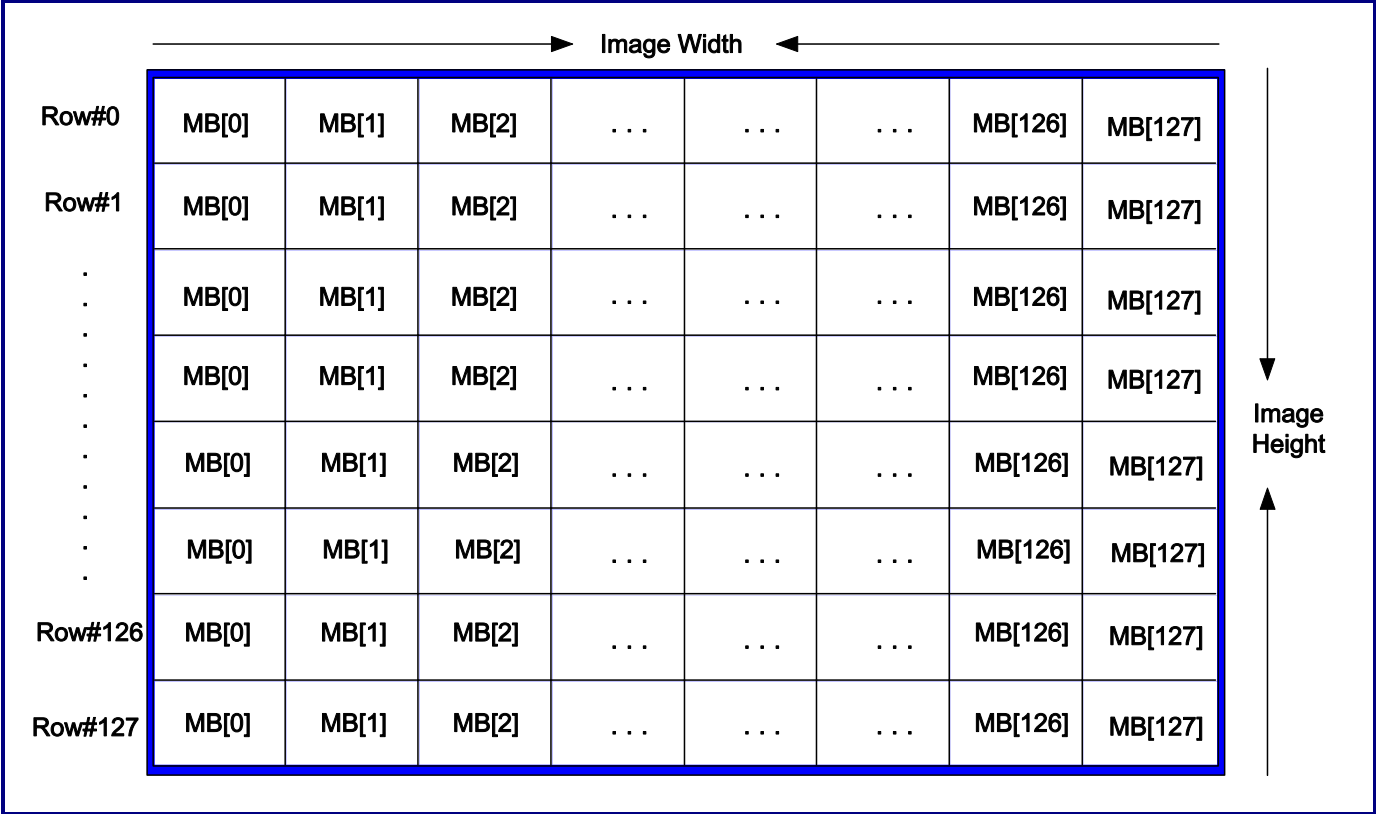


Figure 7-1. 128x128 Motion Blocks Mapping on One Image

# Chapter 8

## Capture Driver Module

---

In the GM8136 SDK release package, users can find the source code and kernel module of the capture from "/module/vcap300". The source code and kernel module of the decoder/sensor are placed in "/module/vcap300/input\_module".

The capture driver module contains the following parts:

- vcap300\_common.ko  
This is the capture core. It includes the capture hardware control layer, middleware communication layer, and export library layer.
- vcap0.ko  
This contains the FTVCAP300 IP information. It includes the IP base address and IRQ number.
- vcap300\_xxx.ko [input module]  
This is the front-end input module driver. It includes the decoder/sensor information. Based on this information, the capture driver will setup the video interface to capture the video data from the front-end device.





# Chapter 9

## Capture Module Parameters

---

This chapter contains the following sections:

- 9.1 vcap300\_common Module
- 9.2 vcap0 Module
- 9.3 Input Modules

## 9.1 vcap300\_common Module

This module parameter is not needed.

## 9.2 vcap0 Module

Table 9-1 lists and describes the parameters when inserting this module.

**Table 9-1. Parameters of vcap0 Module**

Name	Default Value	Description
cap_mode	1	1: Link List Mode  vcap300 supports the link-list capture mode. The link list mode uses the hardware link list table to instruct hardware to capture an image from the front-end device.
vi_mode	{ 0, 0}	vi_mode[x] => x from 0 ~ 1  0: Disable VI is not used.  1: Bypass VI is configured in the 1CH bypass mode.  2: 2CH VI is configured in the 2CH byte interleave mode.  3: 4CH VI is configured in the 4CH byte/frame interleave mode.
cap_md	1	Capture motion detection function control  0: Motion detection function is Off (Can reduce the system memory utilization).  1: Motion detection function is On.
sync_time_div	60	Capture sync. timer divide value for controlling the capture interrupt trigger second  Default is 1/60 (s).
hcrop_rule	{{0,0}, {0,0}, {0,0}}	Capture horizontal source cropping rule {source_width, crop_width}  Users can specify three different cropping rules.

Name	Default Value	Description
vi_max_w	{0, 0, 0, 0, 0, 0, 0, 0, 0}	Capture video interface input signal maximum width 0 ~ 4096 0: Use default SD line buffer calculation mechanism without user specified
ext_irq_src	0	Capture Extra Interrupt Source BIT0: LL_Done, use each channel link-list done as the interrupt source
grab_filter	0x000000040	Capture video frame grab filter for driver to drop fail frame BIT0: SD_JOB_OVF BIT1: SD_SC_MEM_OVF BIT2: SD_PARAM_ERR BIT3: SD_PREFIX_ERR BIT4: SD_TIMEOUT BIT5: SD_PIXEL_LACK BIT6: SD_LINE_LACK Driver will drop the video frame when detecting the frame error, and enable the related bit to disable the frame drop when error is detected.

## 9.2.1 Video Interface Mode

This section explains the function of the vi\_mode module parameter.

### 9.2.1.1 VI Bypass Mode

The video interface should be configured as the bypass mode if the output format of the front-end device is BT656/1120/BT601/ISP 1CH interlace or progressive. Figure 9-1 is the block diagram of VI and channel mapping in the VI bypass mode. Each VI has four channels and each channel has four output resolution paths. In the bypass mode, only CH#0 can be operational. Figure 9-1 shows one channel clock and data.

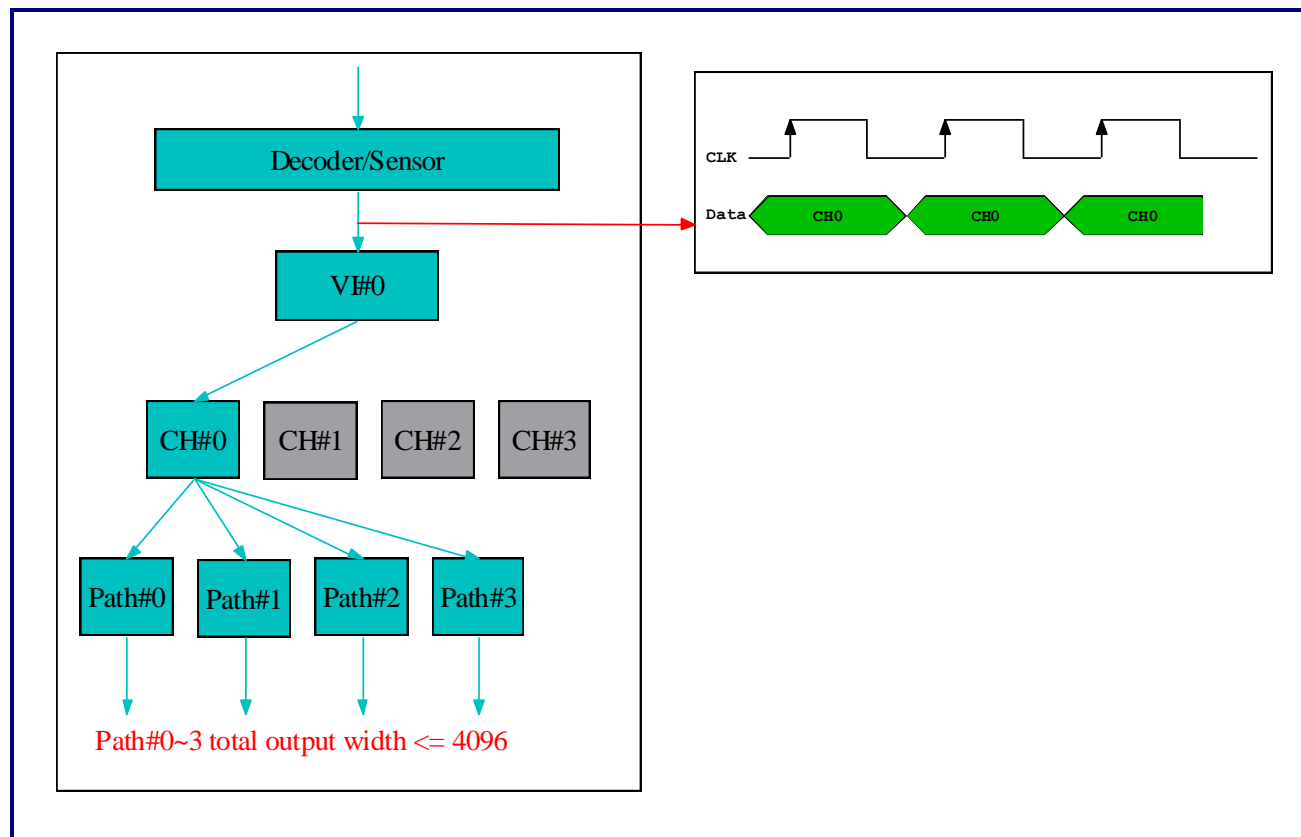


Figure 9-1. VI Bypass Mode

### 9.2.1.2 VI 2CH Byte Interleave Mode (2CH Dual Edge Mode)

The video interface should be configured as the VI 2CH byte interleave mode if the output format of the front-end device is BT656/1120 2CH byte interleave. The capture hardware latches data at the rising edge and falling edge. Figure 9-2 shows the block diagram of VI and channel mapping in the VI 2CH byte interleave mode. In this mode, only CH#0 and CH#2 can be operational. VI#1 does not support this mode. Figure 9-2 shows 2-channel clock and data.

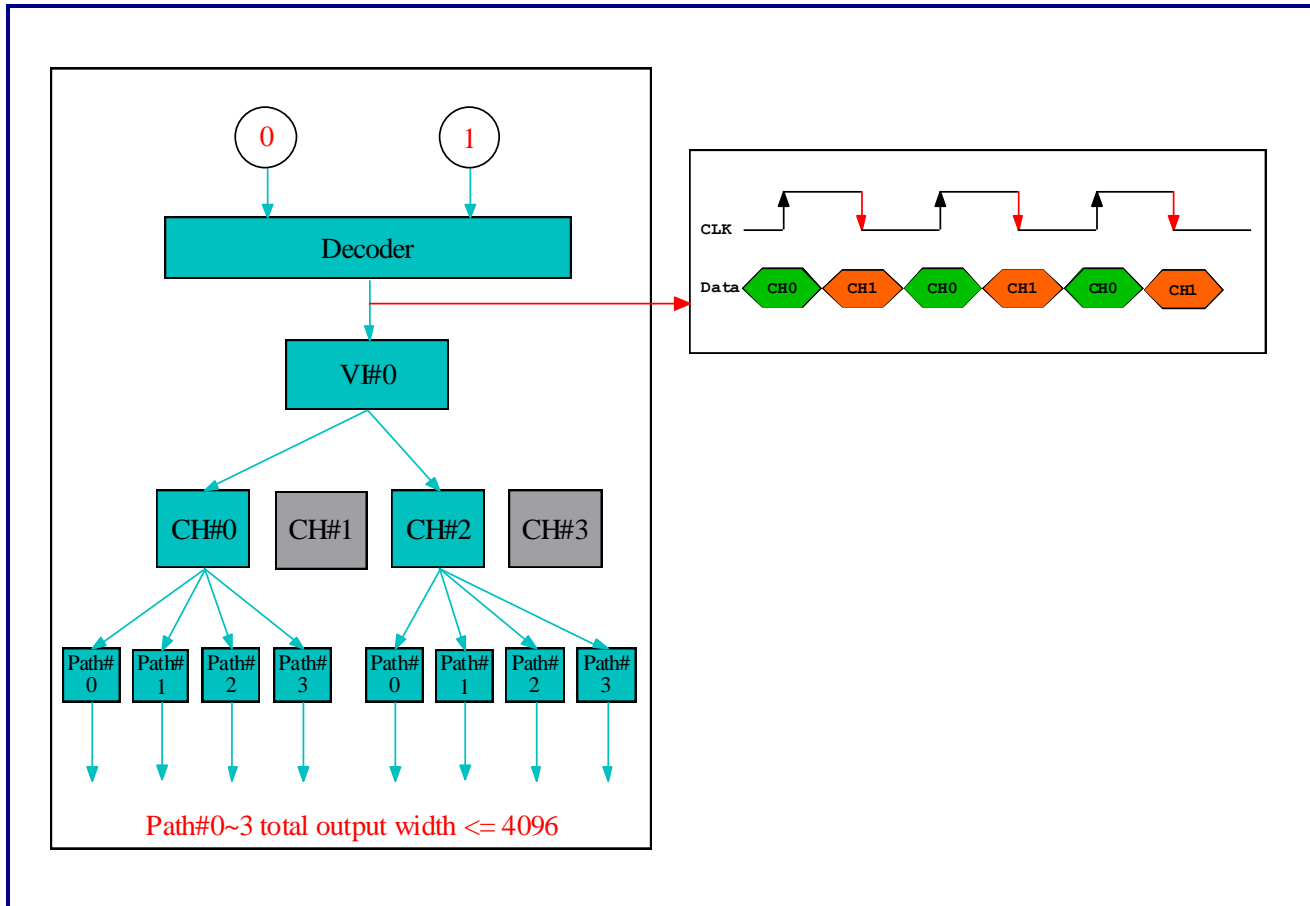


Figure 9-2. VI 2CH Byte Interleave Mode

### 9.2.1.3 VI 4CH Byte/Frame Interleave Mode

The video interface should be configured as the VI 4CH byte/frame interleave mode if the output format of the front-end device is BT656/1120 4CH byte/frame interleave. The captured de-multiplexed 4-channel data use CH#0 ~ 3. Figure 9-3 shows the block diagram of VI and channel mapping in the VI 4CH byte/frame interleave mode. VI#1 does not support this mode. Figure 9-3 shows 4-channel clock and data.

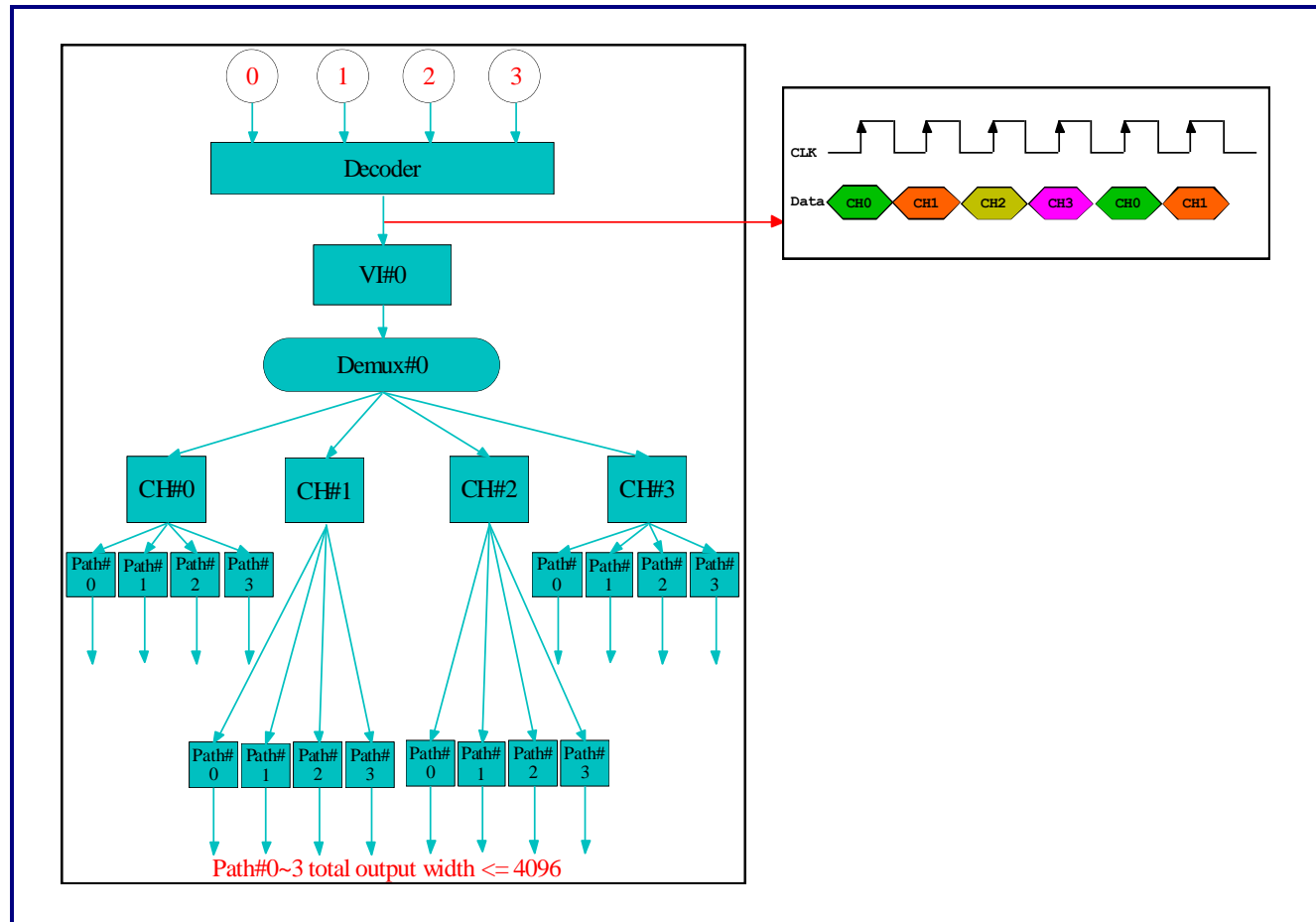


Figure 9-3. VI 4CH Byte/Frame Interleave Mode

## 9.3 Input Modules

### 9.3.1 vcap300\_generic\_mX Module

This generic module is used to configure the video input interface; X is ranging from 0 to 1. Table 9-2 lists and describes the parameters when inserting this module.

Table 9-2. Parameters of vcap300\_generic\_mX Module

Name	Default Value	Description
vi	X	Index of the video input interface is ranging from 0 to 1
vi_src	-1	VI input source selection -1: X_CAP#Xi 0: X_CAP#0 1: X_CAP#1 12: ISP
interface	1	VI input interface format selection 0: BT656 interlace 1: BT656 progressive 2: BT1120 interlace 3: BT1120 progressive 4: RGB888 (Not supported) 5: SDI8BIT interlace 6: SDI8BIT progressive 7: BT601_8BIT interlace 8: BT601_8BIT progressive 9: BT601_16BIT interlace 10: BT601_16BIT progressive 11: ISP
mode	0	VI input mode selection 0: Bypass (1CH mode) 1: 4CH frame Interleave 2: 2CH byte Interleave 3: 4CH byte Interleave
width	640	Image width of the capture source, Max is 4096
height	480	Image height of the capture source, Max is 4096

Name	Default Value	Description
order	0	VI field order selection 0: Any field for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
range	0	Data output range selection 0: 256 levels 1: 240 levels
inv_clk	0	Input pixel clock invert control 0: Disable 1: Enable
yc_swap	0	Output data YC swap control 0: None 1: YC swap 2: CbCr swap 3: YC + CbCr swap
data_swap	0	Input data swap control 0: None 1: Lo8Bit swap 2: Byte swap 3: Lo8Bit + Byte swap 4: Hi8Bit swap 5: LoHi8Bit swap 6: Hi8Bit + Byte swap 7: LoHi8Bit + Byte swap
frame_rate	30	VI source frame rate
speed	1	VI source speed type 0: 50/60I (Interlace) 1: 25/30P (Progressive) 2: 50/60P (Progressive)



Name	Default Value	Description
rgb_param	{ 0, 0, 0, 0, 0 }	RGB888 signal parameter control rgb_param[0] => Vertical polarity 0: Active low 1: Active high rgb_param[1] => Horizontal polarity 0: Active low 1: Active high rgb_param[2] => Data enable polarity 0: Active low 1: Active high rgb_param[3] => Watch data enable 0: Disable 1: Enable rgb_param[4] => Horizontal scaling-down ratio Ratio: 0 ~ 15
timeout	1000 ms	Signal Timeout Threshold (ms)
bt601_param	{0,0,0,0,0}	BT601 signal parameter control bt601_param[0] => Vertical polarity 0: Active low 1: Active high bt601_param[1] => Horizontal polarity 0: Active low 1: Active high bt601_param[2] => Sync Mode 0: None sync. mode 1: Sync. mode bt601_param[3] => Valid data x offset Offset : 0 ~ 255 bt601_param[4] => Valid data y offset Offset : 0 ~ 255
ch_id	{-1, -1, -1, -1}	Video channel index -1: Base on VI number

### 9.3.2 vcap300\_nvp1118 Module

This module is used to configure the video input interface for the NextChip NVP1118 8-CH 720H video decoders and audio codecs. Table 9-3 lists and describes the parameters when inserting this module.

**Table 9-3. Parameters of vcap300\_nvp1118 Module**

Name	Default Value	Description
vport	{ 0x0012, 0x0034, 0x0056, 0x0078 }	NVP1118 VPort link selection vport[x] => NVP1118#x BIT[0:3] => VPORTA BIT[4:7] => VPORTB BIT[8:11] => VPORTC BIT[12:15] => VPORTD 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
mode	{ 2, 2, 2, 2 }	NVP1118 video port output mode selection mode[x] => NVP1118#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 1, 1, 1, 1 }	NVP1118 video standard norm norm[x] => NVP1118#x 0: PAL 1: NTSC

Name	Default Value	Description
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => NVP1118#x 0: Any field for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control Inv_clk[x] = > NVP1118#x 0: Disable 1: Enable

### 9.3.3 vcap300\_nvp1918 Module

This module is used to configure the video input interface for the NextChip NVP1918 8-CH 720H/960H video decoders and audio codecs. Table 9-4 lists and describes the parameters when inserting this module.

**Table 9-4. Parameters of vcap300\_nvp1918 Module**

Name	Default Value	Description
vport	{ 0x0012, 0x0034, 0x0056, 0x0078 }	NVP1918 VPort link selection vport[x] => NVP1918#x BIT[0:3] => VPORTA BIT[4:7] => VPORTB BIT[8:11] => VPORTC BIT[12:15] => VPORTD 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)

Name	Default Value	Description
mode	{ 2, 2, 2, 2 }	NVP1918 video port output mode selection mode[x] => NVP1918#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 1, 1, 1, 1 }	NVP1918 video standard norm norm[x] => NVP1918#x 0: PAL 1: NTSC 2: 960H_PAL 3: 960H_NTSC
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => NVP1918#x 0: Any field for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control Inv_clk[x] = > NVP1918#x 0: Disable 1: Enable

### 9.3.4 vcap300\_tw2968 Module

This module is used to configure the video input interface for the Intersil TW2968 8-CH 720H/960H video decoders and audio codecs. Table 9-5 lists and describes the module parameters when inserting this driver.

**Table 9-5. Parameters of vcap300\_tw2968 Module**

Name	Default Value	Description
vport	{ 0x0012, 0x0034, 0x0056, 0x0078 }	TW2968 VPort link selection vport[x] => TW2968#x BIT[0:3] => VD1 BIT[4:7] => VD2

Name	Default Value	Description
		BIT[8:11] => VD3 BIT[12:15] => VD4 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
mode	{ 2, 2, 2, 2 }	TW2968 video port output mode selection mode[x] => TW2968#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 1, 1, 1, 1 }	TW2968 video standard norm norm[x] => TW2968#x 0: PAL 1: NTSC 2: 960H_PAL 3: 960H_NTSC
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => TW2968#x 0: Any field for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control Inv_clk[x] = > TW2968#x 0: Disable 1: Enable

### 9.3.5 vcap300\_tw9900 Module

This module is used to configure the video input interface for the Intersil TW9900 1-CH 720H video decoders and audio codecs. Table 9-6 lists and describes the module parameters when inserting this driver.

**Table 9-6. Parameters of vcap300\_tw9900 Module**

Name	Default Value	Description
vport	{ 0x0001, 0x0002, 0x0003, 0x0004 }	TW9900 VPort link selection vport[x] => TW9900#x BIT[0:3] => VD1 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
mode	{ 0, 0, 0, 0 }	TW9900 video port output mode selection mode[x] => TW9900#x 0: 1CH
norm	{ 1, 1, 1, 1 }	TW9900 video standard norm norm[x] => TW9900#x 0: PAL 1: NTSC
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => TW9900#x 0: Any field for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface

Name	Default Value	Description
inv_clk	{ 1, 1, 1, 1 }	Input pixel clock invert control Inv_clk[x] = > TW9900#x 0: Disable 1: Enable

### 9.3.6 vcap300\_cx26848 Module

This module is used to configure the video input interface for the Conexant CX26848 8-CH 720H/960H video decoders and audio codecs. Table 9-7 lists and describes the parameters when inserting this module.

**Table 9-7. Parameters of vcap300\_cx26848 Module**

Name	Default Value	Description
vport	{ 0x0012, 0x0034, 0x0056, 0x0078 }	CX26848 VPort link selection vport[x] => CX26848#x BIT[0:3] => VPORTA BIT[4:7] => VPORTB BIT[8:11] => VPORTC BIT[12:15] => VPORTD 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
mode	{ 2, 2, 2, 2 }	CX26848 video port output mode selection mode[x] => CX26848#x 0: 1CH 1: 2CH 2: 4CH

Name	Default Value	Description
norm	{ 1, 1, 1, 1 }	CX26848 video standard norm norm[x] => CX26848#x 0: PAL 1: NTSC 2: 960H_PAL 3: 960H_NTSC
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => CX26848#x 0: Any field for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control Inv_clk[x] = > CX26848#x 0: Disable 1: Enable
data_swap	(0, 0, 0, 0)	Data pin swap control 0: None 1: Lo8Bit swap 2: Byte swap 3: Lo8Bit + Byte swap

### 9.3.7 vcap300\_isp Module

This module is used to configure the video input interface for the ISP interface. Table 9-8 lists and describes the parameters when inserting this module.

**Table 9-8. Parameters of vcap300\_isp Module**

Name	Default Value	Description
vi	1	Index of the video input interface is ranging from 0 to 1.
width	-1	ISP output width -1: Based on the ISP output resolution
height	-1	ISP output height -1: Based on the ISP output resolution



Name	Default Value	Description
range	0	Data output range selection 0: 256 levels 1: 240 levels
inv_clk	0	Input pixel clock invert control 0: Disable 1: Enable
yc_swap	0	Output data YC swap control 0: None 1: YC swap 2: CbCr swap 3: YC + CbCr swap
timeout	-1	Signal Timeout Threshold (ms) -1: Based on the ISP output frame rate
ch_id	-1	Video Channel Index -1: Based on the VI number

### 9.3.8 vcap300\_hm1375 Module

This module is used to configure the video input interface for the HiMax HM1375 sensor. Table 9-9 lists and describes the parameters when inserting this module.

**Table 9-9. Parameters of vcap300\_hm1375 Module**

Name	Default Value	Description
vport	{1}	HM1375 VPort link selection vport[x] => HM1375#x 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported)

Name	Default Value	Description
		9: X_CAPCAS (Not supported)
norm	{1}	HM1375 video standard norm norm[x] => HM1375#x 0: VGA 640x480 1: HD720 1280x720 2: WXGA_960 1280x960 3: SXGA 1280x1024
inv_clk	{0}	Input pixel clock invert control Inv_clk[x] = > HM1375#x 0: Disable 1: Enable

### 9.3.9 vcap300\_mt9m131 Module

This module is used to configure the video input interface for the Aptina MT9M131 sensor. Table 9-10 lists and describes the parameters when inserting this module.

**Table 9-10. Parameters of vcap300\_mt9m131 Module**

Name	Default Value	Description
vport	{1, 2}	MT9M131 VPort link selection vport[x] => MT9M131#x 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)

Name	Default Value	Description
norm	{0, 0}	MT9M131 video standard norm norm[x] => MT9M131#x 0: VGA 640x480 1: QSXGA 640x512 2: SXGA 1280x1024
inv_clk	{0, 0}	Input pixel clock invert control Inv_clk[x] = > MT9M131#x 0: Disable 1: Enable

### 9.3.10 vcap300\_mt9d131 Module

This module is used to configure the video input interface for the Aptina MT9D131 sensor. Table 9-11 lists and describes the parameters when inserting this module.

**Table 9-11. Parameters of vcap300\_mt9d131 Module**

Name	Default Value	Description
vport	{1}	MT9D131 VPort link selection vport[x] => MT9D131#x 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)

Name	Default Value	Description
norm	{1}	MT9D131 video standard norm norm[x] => MT9D131#x 0: SVGA 800x600 1: HD720 1280x720 2: WXGA 1280x800 3: UXGA 1600x1200
inv_clk	{0}	Input pixel clock invert control Inv_clk[x] => MT9D131#x 0: Disable 1: Enable
vs_pol	{1}	BT601 VSync Polarity vs_pol[x] => MT9D131#x 0: Active Low 1: Active High
hs_pol	{1}	BT601 HSync Polarity hs_pol[x] => MT9D131#x 0: Active Low 1: Active High
x_offset	{0}	BT601 Valid data X offset x_offset[x] => MT9D131#x 0 ~ 255
y_offset	{0}	BT601 Valid data Y offset y_offset[x] => MT9D131#x 0 ~ 255
sync_mode	{0}	BT601 sync mode sync_mode[x] => MT9D131#x 0: Disable 1: Enable

### 9.3.11 OV7725 Module

This module is used to configure the video input interface for the Omni\_Vision OV7725 sensor. Table 9-12 lists and describes the parameters when inserting this module.

**Table 9-12. Parameters of vcap300\_ov7725 Module**

Name	Default Value	Description
vport	{1}	OV7725 VPort link selection vport[x] => OV7725#x 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
norm	{0}	OV7725 video standard norm norm[x] => OV7725#x 0: VGA 640x480
inv_clk	{0}	Input pixel clock invert control Inv_clk[x] = > OV7725#x 0: Disable 1: Enable

### 9.3.12 vcap300\_nvp1918c Module

This module is used to configure the video input interface for the NextChip NVP1918C 8-CH 720H/960H video decoders and audio codecs. Table 9-13 lists and describes the parameters when inserting this module.

**Table 9-13. Parameters of vcap300\_nvp1918c Module**

Name	Default Value	Description
vport	{ 0x0012, 0x0034, 0x0056, 0x0078 }	NVP1918C VPort link selection vport[x] => NVP1918C#x BIT[0:3] => VPORTA BIT[4:7] => VPORTB 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
mode	{ 2, 2, 2, 2 }	NVP1918C video port output mode selection mode[x] => NVP1918C#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 1, 1, 1, 1 }	NVP1918C video standard norm norm[x] => NVP1918C#x 0: PAL 1: NTSC 2: 960H_PAL 3: 960H_NTSC

Name	Default Value	Description
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => NVP1918C#x 0: Anyone for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control inv_clk[x] = > NVP1918C#x 0: Disable 1: Enable

### 9.3.13 vcap300\_nvp1914 Module

This module is used to configure the video input interface for the NextChip NVP1914 4-CH 720H/960H video decoders and audio codecs. Table 9-14 lists and describes the parameters when inserting this module.

**Table 9-14. Parameters of vcap300\_nvp1914 Module**

Name	Default Value	Description
vport	{ 0x0012, 0x0034, 0x0056, 0x0078 }	NVP1914 VPort link selection vport[x] => NVP1914#x BIT[0:3] => VPORTA BIT[4:7] => VPORTB 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)

Name	Default Value	Description
mode	{ 2, 2, 2, 2 }	NVP1914 video port output mode selection mode[x] => NVP1914#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 1, 1, 1, 1 }	NVP1914 video standard norm norm[x] => NVP1914#x 0: PAL 1: NTSC 2: 960H_PAL 3: 960H_NTSC
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => NVP1914#x 0: Anyone for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control inv_clk[x] = > NVP1914#x 0: Disable 1: Enable

### 9.3.14 vcap300\_nvp1914C Module

This module is used to configure the video input interface for the NextChip NVP1914C 4-CH 720H/960H video decoders and audio codecs. Table 9-15 lists and describes the parameters when inserting this module.

**Table 9-15. Parameters of vcap300\_nvp1914C Module**

Name	Default Value	Description
vport	{ 0x0012, 0x0034, 0x0056, 0x0078 }	NVP1914C VPort link selection vport[x] => NVP1914C#x BIT[0:3] => VPORTA BIT[4:7] => VPORTB



Name	Default Value	Description
		0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
mode	{ 2, 2, 2, 2 }	NVP1914C video port output mode selection mode[x] => NVP1914C#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 1, 1, 1, 1 }	NVP1914C video standard norm norm[x] => NVP1914C#x 0: PAL 1: NTSC 2: 960H_PAL 3: 960H_NTSC
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => NVP1914C#x 0: Anyone for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control inv_clk[x] = > NVP1914C#x 0: Disable 1: Enable

### 9.3.15 vcap300\_tw2964 Module

This module is used to configure the video input interface for the Intersil TW2964 4-CH 720H/960H video decoders and audio codecs. Table 9-16 lists and describes the parameters when inserting this module.

**Table 9-16. Parameters of vcap300\_tw2964 Module**

Name	Default Value	Description
vport	{ 0x0010, 0x0030, 0x0050, 0x0070 }	TW2964 VPort link selection vport[x] => TW2964#x BIT[0:3] => VD1 BIT[4:7] => VD2 BIT[8:11] => VD3 BIT[12:15] => VD4 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
mode	{ 2, 2, 2, 2 }	TW2964 video port output mode selection mode[x] => TW2964#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 1, 1, 1, 1 }	TW2964 video standard norm norm[x] => TW2964#x 0: PAL 1: NTSC 2: 960H_PAL 3: 960H_NTSC

Name	Default Value	Description
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => TW2964#x 0: Anyone for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control inv_clk[x] = > TW2964#x 0: Disable 1: Enable

### 9.3.16 vcap300\_tw2868 Module

This module is used to configure the video input interface for the Intersil TW2868 8-CH 720H video decoders and audio codecs. Table 9-17 lists and describes the parameters when inserting this module.

**Table 9-17. Parameters of vcap300\_tw2868 Module**

Name	Default Value	Description
vport	{ 0x0012, 0x0034, 0x0056, 0x0078 }	TW2868 VPort link selection vport[x] => TW2868#x BIT[0:3] => VD1 BIT[4:7] => VD2 BIT[8:11] => VD3 BIT[12:15] => VD4 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)

Name	Default Value	Description
mode	{ 2, 2, 2, 2 }	TW2868 video port output mode selection mode[x] => TW2868#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 1, 1, 1, 1 }	TW2868 video standard norm norm[x] => TW2868#x 0: PAL 1: NTSC
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => TW2868#x 0: Anyone for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control inv_clk[x] = > TW2868#x 0: Disable 1: Enable

### 9.3.17 vcap300\_cx26848 Module

This module is used to configure the video input interface for the Conexant CX26848 8-CH 720H/960H video decoders and audio codecs. Table 9-18 lists and describes the parameters when inserting this module.

**Table 9-18. Parameters of vcap300\_cx26848 Module**

Name	Default Value	Description
vport	{ 0x0012, 0x0034, 0x0056, 0x0078 }	CX26848 VPort link selection vport[x] => CX26848#x BIT[0:3] => VD1 BIT[4:7] => VD2 BIT[8:11] => VD3 BIT[12:15] => VD4 0: None 1: X_CAP#0

Name	Default Value	Description
		2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
mode	{ 2, 2, 2, 2 }	CX26848 video port output mode selection mode[x] => CX26848#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 1, 1, 1, 1 }	CX26848 video standard norm norm[x] => CX26848#x 0: PAL 1: NTSC 2: 960H_PAL 3: 960H_NTSC
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => CX26848#x 0: Anyone for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control inv_clk[x] = > CX26848#x 0: Disable 1: Enable
data_swap	0	Data pin swap control of video port 0: None 1: Lo8Bit swap 2: Byte swap 3: Lo8Bit + Byte swap

### 9.3.18 vcap300\_rn6318 Module

This module is used to configure the video input interface for the Richnex RN6318 8-CH 720H/960H video decoders and audio codecs. Table 9-19 lists and describes the parameters when inserting this module.

**Table 9-19. Parameters of vcap300\_rn6318 Module**

Name	Default Value	Description
vport	{ 0x0012, 0x0034 }	RN6318 VPort link selection vport[x] => RN6318#x BIT[0:3] => VD1 BIT[4:7] => VD3 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
mode	{ 2, 2, 2, 2 }	RN6318 video port output mode selection mode[x] => RN6318#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 1, 1, 1, 1 }	RN6318 video standard norm norm[x] => RN6318#x 0: PAL 1: NTSC 2: 960H_PAL 3: 960H_NTSC

Name	Default Value	Description
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => RN6318#x 0: Anyone for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control inv_clk[x] = > RN6318#x 0: Disable 1: Enable
data_swap	0	Data pin swap control of video port 0: None 1: Lo8Bit swap 2: Byte swap 3: Lo8Bit + Byte swap

### 9.3.19 vcap300\_rn6314 Module

This module is used to configure the video input interface for the Richnex RN6314 4-CH 720H/960H video decoders and audio codecs. Table 9-20 lists and describes the parameters when inserting this module.

**Table 9-20. Parameters of vcap300\_rn6314 Module**

Name	Default Value	Description
vport	{ 0x0010, 0x0030, 0x0050, 0x0070 }	RN6314 VPort link selection vport[x] => RN6314#x BIT[0:3] => VD1 BIT[4:7] => VD3 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported)

Name	Default Value	Description
		8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
mode	{ 2, 2, 2, 2 }	RN6314 video port output mode selection mode[x] => RN6314#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 1, 1, 1, 1 }	RN6314 video standard norm norm[x] => RN6314#x 0: PAL 1: NTSC 2: 960H_PAL 3: 960H_NTSC
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => RN6314#x 0: Anyone for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control inv_clk[x] = > RN6314#x 0: Disable 1: Enable

### 9.3.20 vcap300\_cx25930 Module

This module is used to configure the video input interface for the Conexant CX25930 4-CH SDI receiver. Table 9-21 lists and describes the parameters when inserting this module.

**Table 9-21. Parameters of vcap300\_cx25930 Module**

Name	Default Value	Description
vport	{ 0x1234, 0x5678, 0x0000, 0x0000 }	CX25930 VPort link selection vport[x] => CX25930#x BIT[0:3] => VOUT0 BIT[4:7] => VOUT1



Name	Default Value	Description
		BIT[8:11] => VOUT2 BIT[12:15] => VOUT3 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
yc_swap	{ 0x0000, 0x0000, 0x0000, 0x0000 }	Y/C swap control for video port BIT[0:3] => VOUT0 BIT[4:7] => VOUT1 BIT[8:11] => VOUT2 BIT[12:15] => VOUT3 0: None 1: YC swap 2: CbCr swap 3: YC + CbCr swap
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control inv_clk[x] = > CX25930#x BIT[0:3] => VOUT0 BIT[4:7] => VOUT1 BIT[8:11] => VOUT2 BIT[12:15] => VOUT3 0: Disable 1: Enable

### 9.3.21 vcap300\_mv4101 Module

This module is used to configure the video input interface for the Magnision MV4101 4-CH SDI receiver. Table 9-22 lists and describes the parameters when inserting this module.

**Table 9-22. Parameters of vcap300\_mv4101 Module**

Name	Default Value	Description
vport	{ 0x1234, 0x5678, 0x0000, 0x0000 }	MV4101 VPort link selection vport[x] => MV4101#x BIT[0:3] => VOUT0 BIT[4:7] => VOUT1 BIT[8:11] => VOUT2 BIT[12:15] => VOUT3 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
yc_swap	{ 0x0000, 0x0000, 0x0000, 0x0000 }	Y/C swap control for video port BIT[0:3] => VOUT0 BIT[4:7] => VOUT1 BIT[8:11] => VOUT2 BIT[12:15] => VOUT3 0: None 1: YC swap 2: CbCr swap 3: YC + CbCr swap
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control inv_clk[x] = > MV4101#x BIT[0:3] => VOUT0

Name	Default Value	Description
		BIT[4:7] => VOUT1
		BIT[8:11] => VOUT2
		BIT[12:15] => VOUT3
		0: Disable
		1: Enable

### 9.3.22 vcap300\_dh9901 Module

This module is used to configure the video input interface for the Dahua DH9901 4-CH HDCVI receiver. Table 9-23 lists and describes the parameters when inserting this module.

**Table 9-23. Parameters of vcap300\_dh9901 Module**

Name	Default Value	Description
vport	{ 0x1234, 0x5678, 0x0000, 0x0000 }	DH9901 VPort link selection vport[x] => DH9901#x BIT[0:3] => VOUT0 BIT[4:7] => VOUT1 BIT[8:11] => VOUT2 BIT[12:15] => VOUT3 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
yc_swap	{ 0x0000, 0x0000, 0x0000, 0x0000 }	Y/C swap control for video port BIT[0:3] => VOUT0 BIT[4:7] => VOUT1 BIT[8:11] => VOUT2 BIT[12:15] => VOUT3

Name	Default Value	Description
		0: None 1: YC swap 2: CbCr swap 3: YC + CbCr swap
inv_clk	{ 0x0000, 0x0000, 0x0000, 0x0000 }	Input pixel clock invert control inv_clk[x] = > DH9901#x BIT[0:3] => VOUT0 BIT[4:7] => VOUT1 BIT[8:11] => VOUT2 BIT[12:15] => VOUT3 0: Disable 1: Enable

### 9.3.23 vcap300\_nvp6114 Module

This module is used to configure the video input interface for the NextChip NVP6114 4-CH AHD receiver. Table 9-24 lists and describes the parameters when inserting this module.

**Table 9-24. Parameters of vcap300\_nvp6114 Module**

Name	Default Value	Description
vport	{ 0x0021, 0x0043, 0x0065, 0x0087 }	NVP6114 VPort link selection vport[x] => NVP6114#x BIT[0:3] => VPORTA BIT[4:7] => VPORTB 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)

Name	Default Value	Description
mode	{ 2, 2, 2, 2 }	NVP6114 video port output mode selection mode[x] => NVP6114#x 0: 1CH 1: 2CH 2: 4CH
norm	{ 5, 5, 5, 5 }	NVP6114 video standard norm norm[x] => NVP6114#x 0: PAL 1: NTSC 2: 960H_PAL 3: 960H_NTSC 4: 720P_PAL@25P 5: 720P_NTSC@30P
order	{ 1, 1, 1, 1 }	VI field order selection order[x] => NVP6114#x 0: Anyone for the progressive interface 1: Odd field first for the interlace interface 2: Even field first for the interlace interface
inv_clk	{ 0, 0, 0, 0 }	Input pixel clock invert control inv_clk[x] = > NVP6114#x 0: Disable 1: Enable

### 9.3.24 vcap300\_tp2802 Module

This module is used to configure the video input interface for the TechPoint TP2802 4-CH HDTV receiver. Table 9-25 lists and describes the parameters when inserting this module.

**Table 9-25. Parameters of vcap300\_tp2802 Module**

Name	Default Value	Description
vport	{ 0x1234, 0x5678, 0x0000, 0x0000 }	TP2802 VPort link selection vport[x] => TP2802#x BIT[0:3] => VOUT0 BIT[4:7] => VOUT1

Name	Default Value	Description
		BIT[8:11] => VOUT2 BIT[12:15] => VOUT3 0: None 1: X_CAP#0 2: X_CAP#1 3: X_CAP#2 (Not supported) 4: X_CAP#3 (Not supported) 5: X_CAP#4 (Not supported) 6: X_CAP#5 (Not supported) 7: X_CAP#6 (Not supported) 8: X_CAP#7 (Not supported) 9: X_CAPCAS (Not supported)
yc_swap	{ 0x0000, 0x0000, 0x0000, 0x0000 }	Y/C swap control for video port BIT[0:3] => VOUT0 BIT[4:7] => VOUT1 BIT[8:11] => VOUT2 BIT[12:15] => VOUT3 0: None 1: YC swap 2: CbCr swap 3: YC + CbCr swap
inv_clk	{ 0x0000, 0x0000, 0x0000, 0x0000 }	Input pixel clock invert control inv_clk[x] = > TP2802#x BIT[0:3] => VOUT0 BIT[4:7] => VOUT1 BIT[8:11] => VOUT2 BIT[12:15] => VOUT3 0: Disable 1: Enable

# Chapter 10

## Capture Proc Nodes

---

This chapter contains the following sections:

- 10.1 /proc/vcap300/version Proc Node
- 10.2 /proc/vcap300/vcap0 Proc Nodes
- 10.3 /proc/vcap300/vcap0/fcs Proc Nodes
- 10.4 /proc/vcap300/vcap0/denoise Proc Nodes
- 10.5 /proc/vcap300/vcap0/sharpness Proc Nodes
- 10.6 /proc/vcap300/vcap0/md Proc Nodes
- 10.7 /proc/vcap300/vcap0/cfg Proc Nodes
- 10.8 /proc/vcap300/vcap0/cfg Proc Nodes
- 10.9 /proc/vcap300/input\_module Proc Nodes
- 10.10 /proc/vcap300/vcap0/diagnostic Proc Nodes
- 10.11 /proc/vcap300/vcap0/dbg\_mode Proc Node
- 10.12 /proc/vcap300/vcap0/crop\_rule Proc Node
- 10.13 /proc/vcap300/vcap0/grab\_filter Proc Node
- 10.14 /proc/vcap300/vcap0/vi\_probe Proc Node

The capture driver module provides the proc nodes. Users can read the capture information or setup configuration through these nodes. A sample of these nodes is listed below.

```

/ # ls /proc/vcap300/
input_module  vcap0          version

/ # ls /proc/vcap300/vcap0/
ability      denoise      dump_reg     jobq         sharpness
cfg          diagnostic  dump_split  lli_info     status
crop_rule    dump_ch     fcs         md           vg_info
dbg_mode     dump_lli    grab_filter  presmooth    vi_probe

/ # ls /proc/vcap300/input_module/
generic.0    generic.1    table

```

**Figure 10-1. Proc Nodes of Capture Module**

The following table shows the proc nodes of the capture module.

/proc/vcap300	Version	
	vcap0	ability
		dump_ch
		dump_reg
		dump_lli
		lli_info
		jobq
		status
		vg_info
		dbg_mode
		crop_rule
		grab_filter
		vi_probe
	diagnostic	channel
		clear
		frame_cnt
		global



		fcs	ch
			enable
			param
		denoise	ch
			enable
			param
		sharpness	ch
			param
		presmooth	ch
			param
	md	ch	
		event	
		param	
		region	
	cfg	channel	
		global	
		mark	
		mask	
		osd	
input_module		table	

## 10.1 /proc/vcap300/version Proc Node

Users can use the /proc/vcap300/version node to get the driver version.

Usage:

- Get the current value

**cat /proc/vcap300/version**

```
/ # cat /proc/vcap300/version  
Version: 0.1.7
```

## 10.2 /proc/vcap300/vcap0 Proc Nodes

### 10.2.1 ability

Users can use the /proc/vcap300/vcap0/ability node to get the capture hardware ability.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/ability**

```

/# cat /proc/vcap300/vcap0/ability
HW_Version      : 20130528
HW_Revision     : 000
VI_Count        : 1
Cascade_Count   : 1
Mask_Win_Count  : 8
OSD_Win_Count   : 8
Mark_Win_Count  : 4
Scaler_Count    : 4
Scaler_Ability_UP : Yes Yes Yes No
Scaler_Ability_DOWN: Yes Yes Yes No
FCS_Support     : Yes
Denoise_Support : Yes
Sharpness_Support : Yes
VI_MD_Win_X_Num : 127 127
MD_IMG_SRC      : TC_Out
TC_X_Align      : 2

```

Name	Description
HW_Version	VCAP300 hardware version in this platform
HW_Revision	VCAP300 hardware revision in this platform
VI_Count	Number of video interface
Cascade_Count	Number of cascade video interface
Mask_Win_Count	Number of MASK window for each channel (MASK window apply to all path of each channel)
OSD_Win_Count	Number of OSD window for each channel (OSD window share for all path of each channel)
Mark_Win_Count	Number of MARK window for each channel (MARK window share for all path of each channel)
Scaler_Count	Number of scaler(path) for each channel
Scaler_Ability_UP	Scaling up ability for each scaler of each channel
Scaler_Ability_DOWN	Scaling down ability for each scaler of each channel
FCS_Support	1-D false color suppression support

Name	Description
Denoise_Support	1-D denoise support
Sharpness_Support	Line sharpness support
VI_MD_Win_X_Num	Maximus motion block number of horizontal for each VI. This number depends on VI mode setting, limited by hardware ability.
MD_IMG_SRC	Image source for motion detection hardware engine (SD_OUT/TC_OUT). Depend on different platform, limited by different VCAP300 hardware.

## 10.2.2 dump\_ch

Users can use the `/proc/vcap300/vcap0/dump_ch` node to dump the channel register of the capture hardware.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/dump\_ch**

```

/# cat /proc/vcap300/vcap0/dump_ch

=== CH#0 ===
[0]:0x0000: 0000000a 00000000 00000000 00000000
[0]:0x0010: 005a0140 00f002d0 00f002d0 00f002d0
[0]:0x0020: 02d00140 02d002d0 0c8c80fa 03842832
[0]:0x0030: 00000000 00000000 00000000 00000000
[0]:0x0040: 00000000 00000000 00000000 00000000
[0]:0x0050: 00000000 00000000 00000000 00000000
[0]:0x0060: 00000000 00000000 00000000 00000000
[0]:0x0070: 00000840 00000000 02aa0240 001aa000
[0]:0x0080: 30000000 1842ac22 01000100 00000000
[0]:0x0090: 30000000 1842ac22 01000100 00000000
[0]:0x00a0: 30000000 1842ac22 01000100 00000000
[0]:0x00b0: 30000000 1842ac22 30300008 000000c0
[0]:0x00c0: 00000000 00000000 00000000 00000000
[0]:0x00d0: 00000000 00000000 00000000 00000000

```

```

[0]:0x00e0: 00000000 00000000 00000000 00000000
[0]:0x00f0: 00000000 00000000 00000000 00000000
[0]:0x0100: 00000000 00000000 00000000 00000000
[0]:0x0110: 00000000 00000000 00000000 00000000
[0]:0x0120: 00000000 00000000 00000000 00000000
[0]:0x0130: 00000000 00000000 00000000 00000000
[0]:0x0140: 00000000 00000000 00000000 00000000
[0]:0x0150: 00000000 00000000 00000000 00000000
[0]:0x0160: 013f0000 00590000 02cf0000 00ef0000
[0]:0x0170: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x0180: 00000000 00000000 1ccb0020 ccf0b907
[0]:0x0190: 9003ffe0 ffe00020 04001000 003c0370
[0]:0x01a0: 00000000 00000000 00000000 00000000
[0]:0x01b0: 00000000 00000000 00000000 00000000
[0]:0x01c0: 01880079 03180250 018800b8 03180250
[0]:0x01d0: 00000000 00000000

```

- Set the value to switch the channel number  
**echo [value] > /proc/vcap300/vcap0/dump\_ch**  
Value: 0 ~ 4, others are for dumping all channels

### 10.2.3 dump\_reg

Users can use the `/proc/vcap300/vcap0/dump_reg` node to dump the global control register of the capture hardware.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/dump\_reg**

```
/# cat /proc/vcap300/vcap0/dump_reg

[0]:0x5000: 08602300 00f002d0 08602300 00f002d0
[0]:0x5010: 08602300 00f002d0 08602300 00f002d0
[0]:0x5020: 08602300 00f002d0 08602300 00f002d0
[0]:0x5030: 08602300 00f002d0 08602300 00f002d0
[0]:0x5040: 08000000 00000000 10000000 28000000
[0]:0x5050: d9ea1048 eec0523d 00000000 00000000
[0]:0x5100: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x5110: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x5120: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x5130: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x5140: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x5150: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x5160: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x5170: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x5180: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x5190: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x51a0: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x51b0: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x51c0: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x51d0: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x51e0: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x51f0: 02cf0000 00ef0000 02cf0000 00ef0000
[0]:0x5200: 00000000 00000000 00000000 00000000
```

## 10.2.4 dump\_lli

Users can use the `/proc/vcap300/vcap0/dump_lli` node to dump the channel link-list table of the capture hardware.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/dump\_lli**

```
/# cat /proc/vcap300/vcap0/dump_lli
[CH#00]
>> Path[0] Link List Table =====
[N : 0xcf9c00b8]
      (00) 0xa0040050 00000000
[00: 0xcf9c0050]
      (00) 0x20000082 c63e0005
      (01) 0x40780198 04001000
      (02) 0x4078019c 003c0370
      (03) 0x40080000 0000000a
      (04) 0x800400c0 00000000
[N : 0xcf9c00c0]
      (00) 0xa0040078 00000000
[01: 0xcf9c0078]
      (00) 0x20000000 00000000
      (01) 0x40780198 04001000
      (02) 0x4078019c 003c0370
      (03) 0x40080000 0000000a
      (04) 0x800400b0 00000000
[N : 0xcf9c00b0]
      (00) 0x00080000 00000000
>> Path[1] Link List Table =====
[N : 0xcf9c0188]
      (00) 0x00100000 00000000
>> Path[2] Link List Table =====
[N : 0xcf9c0250]
      (00) 0x00200000 00000000
>> Path[3] Link List Table =====
```

```
[N : 0xcf9c0318]
      (00) 0x00400000 00000000
```

- Set the value to switch the channel number

**echo [value] > /proc/vcap300/vcap0/dump\_lli**

Value: 0 ~ 4, others are for dumping all channels.

### 10.2.5 lli\_info

Users can use the /proc/vcap300/vcap0/lli\_info node to display the channel and global link-list table status information.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/lli\_info**

```
/# cat /proc/vcap300/vcap0/lli_info
[vcap#0]
Global_Update: 68
-----
CH#      PATH#      Normal      NULL      Start      End
=====
0         0         4           4         0x0000     0x00c7
          1         4           4         0x00c8     0x018f
          2         4           4         0x0190     0x0257
          3         4           4         0x0258     0x031f
```

- Set the value to switch the channel number

**echo [value] > /proc/vcap300/vcap0/lli\_info**

Value: 0 ~ 4, others are for dumping all channels.

Name	Description
CH#	Channel index
PATH#	Path index
Normal	Free normal LLI table count



Name	Description
NULL	Free null LLI table count
Start	Start offset of internal LLI memory for each path
End	End offset of internal LLI memory for each path
Global_Update	Free global LLI table count, share to use for all channels

## 10.2.6 jobq

Users can use the `/proc/vcap300/vcap0/jobq` node to display the ongoing and pending job IDs for jobs putted from middleware for each channel and path.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/jobq**

```

/# cat /proc/vcap300/vcap0/jobq
[CH#00]
Path  Job_ID
=====
0      Ongoing=> 70294593[r:70294593] [dummy] 70294594[r:70294594]
      Pending=> 70294625[r:70294625]
1      Ongoing=>
      Pending=>
2      Ongoing=>
      Pending=>
3      Ongoing=>
      Pending=>

```

- Set the value to switch the channel number

**echo [value] > /proc/vcap300/vcap0/jobq**

Value: 0 ~ 4, others are for dumping all channels.

Name	Description
Job_ID	Videograph job index (frame_buffer) r: Root job

Name	Description
	dummy: Dummy LLI table, block the DMA data output, which is used for the frame rate control
Ongoing	Job ready in the hardware LLI engine queue
Pending	Job ready in the software pending queue wait CPU to put to the hardware LLI engine queue

## 10.2.7 status

Users can use the `/proc/vcap300/vcap0/status` node to display the statuses of all channels and paths.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/status**

```

/# cat /proc/vcap300/vcap0/status
Dev      VI#      CH#      P0      P1      P2      P3
-----
(IDLE )
      0 (IDLE ) 0 (IDLE ) IDLE    IDLE    IDLE    IDLE
              1 (IDLE ) IDLE    IDLE    IDLE    IDLE
              2 (IDLE ) IDLE    IDLE    IDLE    IDLE
              3 (IDLE ) IDLE    IDLE    IDLE    IDLE
      1 (IDLE ) 0 (IDLE ) IDLE    IDLE    IDLE    IDLE

```

## 10.2.8 vg\_info

Users can use the `/proc/vcap300/vcap0/vg_info` node to display middleware and capture the mapping information of the channel.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/vg\_info**

```

/# cat /proc/vcap300/vcap0/vg_info
| VCH#  VCAP_CH#  Num_of_Split  Num_of_Path  VI_Mode  FD_Start  Resolution  VLOS
|-----|
| 0      0          0           4           0       0x10000000  640x480    No
| 1      4          0           4           0       0x10000200  1280x720   No

```

Name	Description
VCH#	Video channel index Hardware video connector index and videograph access channel index
VCAP_CH#	Capture hardware channel index (0 ~ 32) 32: Cascade
Num_of_Split	Image split count of split VI
Num_of_Path	Number of path for this channel, depending on the VI mode. Only two paths for split VI, and generic VI has four paths.
VI_Mode	VI operation mode 0: 1CH Bypass mode 1: 2CH Dual Edge mode 2: 4CH MUX mode 3: Bypass-Split mode
FD_Start	Tag for videograph job dispatch identify
Resolution	Video signal resolution for this channel
VLOS	Video loss status for this channel

### 10.2.9 crop\_rule

Users can use the /proc/vcap300/vcap0/crop\_rule node to display and capture the information of the horizontal cropping rule.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/crop\_rule**

```
/# cat /proc/vcap300/vcap0/crop_rule
Rule#  H_In    H_Out
-----
0      720     704
1      -      -
2      -      -
```

Name	Description
Rule#	Horizontal source cropping rule index  The source cropping will apply to all channels if the width of the video signals equal to H_IN. The width of the video image will center crop to H_out size.
H_IN	Video signal image input width
H_Out	Video signal image center crop width

## 10.3 /proc/vcap300/vcap0/fcs Proc Nodes

The capture hardware supports the false color suppression for each channel. Users can use the fcs proc nodes to adjust the fcs parameter for reducing the color noise.

### 10.3.1 ch

Users can use the /proc/vcap300/vcap0/fcs/ch node to display and switch the fcs control channel.

Usage:

- Get the current value

### **cat /proc/vcap300/vcap0/fcs/ch**

```
/# cat /proc/vcap300/vcap0/fcs/ch
FCS_Control_CH: 0 (0~4, other for all)
```

- Set the value to switch the channel number

**echo [value] > /proc/vcap300/vcap0/fcs/ch**

Value: 0 ~ 4, others are for all channels.

### **10.3.2 enable**

Users can use the /proc/vcap300/vcap0/fcs/enable node to enable/disable the channel fcs function.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/fcs/enable**

```
/# cat /proc/vcap300/vcap0/fcs/enable

=== [CH#00] ===
FCS_Enable: 0
```

- Set the value to disable/enable channel fcs

**echo [value] > /proc/vcap300/vcap0/fcs/enable**

Value: 0 for disable, 1 for enable

### **10.3.3 param**

Users can use the /proc/vcap300/vcap0/fcs/param node to get and set the parameter of the channel fcs.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/fcs/param**

```
/# cat /proc/vcap300/vcap0/fcs/param
```

```

=== [CH#00] FCS Parameter ===
[00]LV0_THRED : 0xfa
[01]LV1_THRED : 0xc8
[02]LV2_THRED : 0xc8
[03]LV3_THRED : 0x32
[04]LV4_THRED : 0x28
[05]GREY_THRED: 0x384

```

- Set the param\_id value to switch the parameter value of channel fcs  
**echo [param\_id] [value] > /proc/vcap300/vcap0/fcs/param**  
 param\_id:  
 0: LV0\_THRED, threshold for Y bandwidth + C bandwidth, [0 ~ 0xFFFF]  
 1: LV1\_THRED, threshold for first element of Y bandwidth, [0 ~ 0xFF]  
 2: LV2\_THRED, threshold for Y bandwidth, [0 ~ 0xFF]  
 3: LV3\_THRED, threshold for C bandwidth, [0 ~ 0xFF]  
 4: LV4\_THRED, threshold for C level, [0 ~ 0xFF]  
 5: GREY\_THRED, threshold for local grey area decision, [0 ~ 0xFF]

## 10.4 /proc/vcap300/vcap0/denoise Proc Nodes

The capture hardware supports 1-D de-noise for each channel. Users can use the de-noise proc nodes to adjust the de-noise parameter for reducing the noise.

### 10.4.1 ch

Users can use the /proc/vcap300/vcap0/denoise/ch node to display and switch the de-noise control channel.

Usage:

- Get the current value  
**cat /proc/vcap300/vcap0/denoise/ch**

```

/# cat /proc/vcap300/vcap0/denoise/ch
DeNoise_Control_CH: 0 (0~4, other for all)

```

- Set value to switch channel number

**echo [value] > /proc/vcap300/vcap0/denoise/ch**

Value: 0 ~ 4, others are for all channels.

### 10.4.2 enable

Users can use the /proc/vcap300/vcap0/denoise/enable node to enable/disable the channel de-noise function.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/denoise/enable**

```

/# cat /proc/vcap300/vcap0/denoise/enable

=== [CH#00] ===
DeNoise_Enable: 0

```

- Set the value to disable/enable channel fcs

**echo [value] > /proc/vcap300/vcap0/denoise/enable**

Value: 0 for disable, 1 for enable

### 10.4.3 param

Users can use the /proc/vcap300/vcap0/denoise/param node to get and set the parameter of the channel de-noise function.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/denoise/param**

```

/# cat /proc/vcap300/vcap0/denoise/param

=== [CH#00] DeNoise Parameter ===
[00]DN_GEOMATRIC      : 0x0

```

```
[01]DN_SIMILARITY      : 0x0
[02]DN_ADAPTIVE        : 0x1
[03]DN_ADAPTIVE_STEP   : 0x10
```

- Set the param\_id value to switch the parameter value of channel de-noise

**echo [param\_id] [value] > /proc/vcap300/vcap0/denoise/param**

param\_id:

- 0: DN\_GEOMATRIC, 1D DN strength according to distance. [0 ~ 0x7] 0 means bypass .
- 1: DN\_SIMILARITY, 1D DN strength according to difference. [0 ~ 0x7] 0 means bypass.
- 2: DN\_ADAPTIVE, disable/enable adaptive de-noise (0: Disable, 1: Enable)
- 3: DN\_ADAPTIVE\_STEP, de-noise adaptive step size (Size: 4/8/16/32/64 /128/256)

## 10.5 /proc/vcap300/vcap0/sharpness Proc Nodes

The capture hardware supports sharpness for each path of each channel. Users can use the sharpness proc nodes to strengthen the edges of an image.

### 10.5.1 ch

Users can use the /proc/vcap300/vcap0/sharpness/ch node to display and switch the sharpness control channel and path

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/sharpness/ch**

```
/# cat /proc/vcap300/vcap0/sharpness/ch
Sharpness_Control_CH   : 0 (0~4, other for all)
Sharpness_Control_PATH: 0 (0~3, other for all)
```

- Set ch and path to switch the channel and path numbers

**echo [ch] [path] > /proc/vcap300/vcap0/sharpness/ch**

ch: 0 ~ 4, others are for all channels.

path: 0 ~ 3, others are for all paths.



## 10.5.2 param

Users can use the `/proc/vcap300/vcap0/sharpness/param` node to get and set the parameter of path sharpness.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/sharpness/param**

```
/# cat /proc/vcap300/vcap0/sharpness/param

=== [CH#00] Sharpness Parameter ===
<Path#0>
[00]SHARP_ADAPTIVE_ENABLE      : 0x1
[01]SHARP_PARAM_RADIUS         : 0x0
[02]SHARP_PARAM_AMOUNT         : 0x16
[03]SHARP_PARAM_THRED          : 0x5
[04]SHARP_PARAM_ADAPTIVE_START : 0x2
[05]SHARP_PARAM_ADAPTIVE_STEP  : 0x3
```

- Set the param\_id value to switch the parameter value of path sharpness

**echo [param\_id] [value] > /proc/vcap300/vcap0/sharpness/param**

param\_id:

- 0: SHARP\_ADAPTIVE\_ENABLE, disable/enable adaptive sharpness (0: Disable, 1: Enable)
- 1: SHARP\_PARAM\_RADIUS, sharpness radius. [0 ~ 0x7] 0 means bypass.
- 2: SHARP\_PARAM\_AMOUNT, sharpness amount. [0 ~ 0x3F]
- 3: SHARP\_PARAM\_THRED, sharpness dn level. [0 ~ 0x3F]
- 4: SHARP\_PARAM\_ADAPTIVE\_START, sharpness adaptive starting strength. [0 ~ 0x3F]
- 5: SHARP\_PARAM\_ADAPTIVE\_STEP, sharpness adaptive step. [0 ~ 0x1F]

## 10.6 /proc/vcap300/vcap0/md Proc Nodes

### 10.6.1 ch

Users can use the /proc/vcap300/vcap0/md/ch node to display and switch the motion detection control channel.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/md/ch**

```
/# cat /proc/vcap300/vcap0/md/ch
MD_Control_CH: 0 (0~4, other for all)
```

- Set the value to switch the channel number

**echo [value] > /proc/vcap300/vcap0/md/ch**

Value: 0 ~ 4, others are for all channels.

### 10.6.2 param

Users can use the /proc/vcap300/vcap0/md/param node to get and set the parameter of the channel motion detection.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/md/param**

```
/# cat /proc/vcap300/vcap0/md/param

=== [CH#00] MD Parameter ===
[00]Alpha      : 0x20
[01]TBG        : 0x1ccb
[02]INIT_VAL   : 0x7
[03]TB         : 0x9
[04]SIGMA      : 0xb
[05]PRUNE      : 0xf
```

```
[06]TAU          : 0xcc
[07]ALPHA_ACCURACY: 0x9ffb0
[08]TG           : 0x9
[09]DXDY         : 0x20
[10]ONE_MIN_ALPHA : 0x7fe0
```

- Set the param\_id value to switch the parameter value of the channel motion detection

**echo [param\_id] [value] > /proc/vcap300/vcap0/md/param**

param\_id:

- 0: Alpha, [0 ~ 0xFFFF]
- 1: TBG, [0 ~ 0x1FFF]
- 2: INIT\_VAL, [0 ~ 0xFF]
- 3: TB, [0 ~ 0xF]
- 4: SIGMA, [0 ~ 0x1F]
- 5: PRUNE, [0 ~ 0xF]
- 6: TAU, [0 ~ 0xFF]
- 7: ALPHA\_ACCURACY, [0 ~ 0xFFFFFFFF]
- 8: TG, [0 ~ 0xF]
- 9: DXDY, [0 ~ 0x3F]
- 10: ONE\_MIN\_ALPHA, [0 ~ 0x1F]

### 10.6.3 region

Users can use the /proc/vcap300/vcap0/md/region node to display the region configuration of the channel motion detection.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/md/region**

```
/# cat /proc/vcap300/vcap0/md/region

=== [CH#00] MD Region ===
```

```

MD_GRP   : 0
MD_Active : 1
MD_Enable : 1
MD_Src    : 0
MD_X_Start : 0
MD_Y_Start : 0
MD_X_Size : 16
MD_Y_Size : 8
MD_X_Num  : 45
MD_Y_Num  : 30

```

Name	Description
MD_GRP	MD group index of this channel (4 channels combine to one MD group)  Only one MD group enables the MD hardware engine at the same time.
MD_Active	MD at the active stage, it means that users can apply to enable MD.  Driver will check the MD group to enable the MD hardware engine.
MD_Enable	MD hardware engine enable status
MD_Src	Path index for MD hardware engine image source
MD_X_Start	Motion block horizontal offset
MD_Y_Start	Motion block vertical offset
MD_X_Size	Motion block width (16/32)
MD_Y_Size	Motion block height
MD_X_Num	Horizontal count of motion block
MD_Y_Num	Vertical count of motion block

## 10.6.4 event

Users can use the `/proc/vcap300/vcap0/md/event` node to display the motion event of a channel. '0' indicates the foreground, '1' indicates the background.

Usage:

- Get the current value

[illegible]

## 10.7 /proc/vcap300/vcap0/cfg Proc Nodes

These nodes are used to debug the capture hardware setting. Users can use these proc nodes to get the current configuration of the capture hardware.

### 10.7.1 channel

Users can use the `/proc/vcap300/vcap0/cfg/channel` node to display the channel configuration.

Usage:

- Get the current value

```
cat /proc/vcap300/vcap0/cfg/channel
```

[illegible]

TC_Y	: 0	0	0	0
TC_W	: 640	640	640	640
TC_H	: 480	480	480	480
Grab_Pair	: 0	0	0	0
DMA_Channel	: 0	0	0	0
Frame2Field	: 0	0	0	0
P2I	: 0	0	0	0

- Set the value to switch the channel number

**echo [value] > /proc/vcap300/vcap0/cfg/channel**

Value: 0 ~ 4, others are for all channels/

Name	Description
VI_Format	Video interface format BT656 BT1120 SDI8BIT RGB888 BT601_8BIT BT601_16BIT ISP
VI_Prog	Video interface progressive/interlace
VI_TDM	Video interface channel time-division multiplexed mode Bypass Frame_Interleave 2CH_Byte_Interleave (Clock Dual Edge mode) 4CH_Byte_Interleave

Name	Description
VI_Cap_Style	Video interface capture style Anyone (Progressive) Odd_Field_First (Interlace) Even_Field_First (Interlace)
Src_W	Video source width
Src_H	Video source height
SC_Type	Source cropping channel type Normal Split
SC_Roing	Source cropping memory address rolling type 1024 2048 4096
SC_X	Source cropping horizontal offset
SC_Y	Source cropping vertical offset
SC_W	Source cropping width
SC_H	Source cropping height
Scaler_Enable	Capture scaler path hardware enable 0: Disable 1: Enable
Scaler_Bypass	Capture scaler path hardware bypass 0: Scaling up/down 1: Scaling bypass
Scaler_W	Capture scaler path output width
Scaler_H	Capture scaler path output height
TC_Enable	Target cropping hardware enable 0: Disable 1: Enable
TC_X	Target cropping horizontal offset
TC_Y	Target cropping vertical offset
TC_W	Target cropping output width
TC_H	Target cropping output height







OSD_V_NUM	: 0	0	0	0	0	0	0	0
OSD_Word_Addr	: 0	0	0	0	0	0	0	0
OSD_Win_Color_FG	: 1	1	1	1	1	1	1	1
OSD_Win_Color_BG	: 0	0	0	0	0	0	0	0
OSD_Alpha_Font	: 0	0	0	0	0	0	0	0
OSD_Alpha_BG	: 0	0	0	0	0	0	0	0
OSD_Border_Width	: 0	0	0	0	0	0	0	0
OSD_Border_Color	: 0	0	0	0	0	0	0	0
OSD_Border_Type	: BG	BG	BG	BG	BG	BG	BG	BG
Img_Border_Color	: 0							
Img_Border_Enb	: 0	0	0	0				
Img_Border_Width	: 0	0	0	0				

- Set the value to switch channel number

**echo [value] > /proc/vcap300/vcap0/cfg/osd**

Value: 0 ~ 4, others are for all channels.

Name	Description
OSD_Priority	OSD window priority MARK_ON_OSD: Mark window on OSD window OSD_ON_MARK: OSD window on Mark window
OSD_Smooth	OSD Font Smooth Disable Enable[X] => x means the smooth level. 0: Weak smoothing effect 1: Strong smoothing effect
OSD_Marq_Length	OSD marquee length 8192, 4096, 2048, 1024, 512, 256, 128, 64, 32, 16, 8, 4
OSD_Marq_Speed	OSD marquee speed 0 ~ 3 (0 means fastest)
OSD_Marq_Mode	OSD marquee mode None: No marquee effect HLINE: One horizontal line marquee effect VLINE: One vertical line marquee effect HFLIP: One horizontal line flip effect

Name	Description
OSD_WIN_Type	OSD window type Mask: window as mask Font: window as OSD
OSD_Enable	OSD window enable 0: Disable 1: Enable
OSD_Align	OSD window align method None TOP_L: Top left TOP_C: Top center TOP_R: Top right BOTTOM_L: Bottom left BOTTOM_C: Bottom center BOTTOM_R: Bottom right CENTER: Center
OSD_Path	OSD window apply to which scaler path 0 ~ 3
OSD_Zoom	OSD Font Zoom function 1x: Horizontal and vertical zoom 1x 2x: Horizontal and vertical zoom 2x 3x: Horizontal and vertical zoom 3x 4x: Horizontal and vertical zoom 4x 1/2x: Horizontal and vertical zoom 1/2x H2x_V1x: Horizontal zoom 2x and vertical zoom 1x H4x_V1x: Horizontal zoom 4x and vertical zoom 1x H4x_V2x: Horizontal zoom 4x and vertical zoom 4x H1x_V2x: Horizontal zoom 1x and vertical zoom 2x H1x_V4x: Horizontal zoom 1x and vertical zoom 4x H2x_V4x: Horizontal zoom 2x and vertical zoom 4x
OSD_X	OSD window horizontal offset
OSD_Y	OSD window vertical offset
OSD_W	OSD window width
OSD_H	OSD window height
OSD_H_SP	OSD window horizontal space

Name	Description
OSD_V_SP	OSD window vertical space
OSD_H_NUM	OSD window horizontal font number
OSD_V_NUM	OSD window vertical font number
OSD_Word_Addr	OSD window display font memory address base in the internal OSD display SRAM
OSD_Win_Color_FG	Palette index of OSD window foreground color 0 ~ 15
OSD_Win_Color_BG	Palette index of OSD window background color 0 ~ 15
OSD_Alpha_Font	Font transparency of OSD window 0%, 25%, 37.5%, 50%, 62.5%, 75% 87.5%, 100%
OSD_Alpha_BG	Background transparency of OSD window 0%, 25%, 37.5%, 50%, 62.5%, 75% 87.5%, 100%
OSD_Border_Width	Border width of OSD window n:0 ~ 7 => 4 x (n+1) pixel
OSD_Border_Color	Palette index of OSD border color 0 ~ 15
OSD_Border_Type	OSD border color type FG: Transparency as foreground BG: Transparency as background
Img_Border_Color	Palette index of image boarder color for all path 0 ~ 15
Img_Border_Enb	Image border enable for each path 0: Disable 1: Enable
Img_Border_Width	Image border width n: 0 ~ 7 => 4 x (n+1) pixel

### 10.7.4 mark

Users can use the `/proc/vcap300/vcap0/cfg/mark` node to display the mark window configuration of each channel.

Usage:

- Get the current value

```
cat /proc/vcap300/vcap0/cfg/mark
```

```
/# cat /proc/vcap300/vcap0/cfg/mark  

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
[CH#00]
Mark_Enable   : 0          0          0          0
Mark_Align    : None      None      None      None
Mark_Path     : 0         0         0         0
Mark_X        : 0         0         0         0
Mark_Y        : 0         0         0         0
Mark_X_DIM    : 64       64       64       64
Mark_Y_DIM    : 64       64       64       64
Mark_Zoom     : 1x       1x       1x       1x
Mark_Alpha    : 0         0         0         0
```

- Set the value to switch the channel number

```
echo [value] > /proc/vcap300/vcap0/cfg/mark
```

Value: 0 ~ 4, others are for all channels.

Name	Description
Mark_Enable	Mark window enable 0: Disable 1: Enable
Mark_Align	Mark window align method None TOP_L: Top left TOP_C: Top center TOP_R: Top right BOTTOM_L: Bottom left BOTTOM_C: Bottom center

Name	Description
	BOTTOM_R: Bottom right CENTER: Center
Mark_Path	Mark window apply to which path 0 ~ 3
Mark_X	Mark window horizontal offset
Mark_Y	Mark window vertical offset
Mark_X_DIM	Mark window horizontal dimension 16 pixels 32 pixels 64 pixels 128 pixels 256 pixels 512 pixels
Mark_Y_DIM	Mark window vertical dimension 16 pixels 32 pixels 64 pixels 128 pixels 256 pixels 512 pixels
Mark_Zoom	Mark window zoom 1x 2x 4x
Mark_Alpha	Transparency of mark window 0%, 25%, 37.5%, 50%, 62.5%, 75% 87.5%, 100%

### 10.7.5 mask

Users can use the /proc/vcap300/vcap0/cfg/mask node to display the mask window configuration of each channel.

Usage:

- Get the current value

```
cat /proc/vcap300/vcap0/cfg/mask
```

```
/# cat /proc/vcap300/vcap0/cfg/mask  
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  
[CH#00]  
  
Mask_Enable   : 0          0          0          0          0          0          0          0  
Mask_X        : 0          0          0          0          0          0          0          0  
Mask_Y        : 0          0          0          0          0          0          0          0  
Mask_W        : 0          0          0          0          0          0          0          0  
Mask_H        : 0          0          0          0          0          0          0          0  
Mask_Color    : 0          0          0          0          0          0          0          0  
Mask_Alpha    : 0          0          0          0          0          0          0          0
```

- Set the value to switch the channel number

```
echo [value] > /proc/vcap300/vcap0/cfg/mask
```

Value: 0 ~ 4, others are for all channels.

Name	Description
Mask_Enable	Mask window enable 0: Disable 1: Enable
Mask_X	Mask window horizontal offset
Mask_Y	Mask window vertical offset
Mask_W	Mask window width
Mask_H	Mask window height
Mask_Color	Palette index of mask color 0 ~ 15
Mask_Alpha	Transparency of mask window 0%, 25%, 37.5%, 50%, 62.5%, 75% 87.5%, 100%
Mask_Border_W	Mask window border width
Mask_Border_T	Mask window border type Hollow True



## 10.8 /proc/vcap300/vcap0/presmooth Proc Nodes

The capture hardware supports scaler presmooth for each path of each channel. Users can use the presmooth proc nodes to adjust the image scaling quality.

### 10.8.1 ch

Users can use the /proc/vcap300/vcap0/presmooth/ch node to display and switch the presmooth control channel and path.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/presmooth/ch**

```
/# cat /proc/vcap300/vcap0/presmooth/ch
Presmooth_Control_CH : 0 (0~4, other for all)
Presmooth_Control_PATH: 0 (0~3, other for all)
```

- Set ch and path to switch the channel and path numbers

**echo [ch] [path] > /proc/vcap300/vcap0/presmooth/ch**

ch: 0 ~ 4, others for all other channels

path: 0 ~ 3, others for all other paths

### 10.8.2 param

Users can use the /proc/vcap300/vcap0/presmooth/param node to get and set the parameter of path presmooth.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/presmooth/param**

```
/# cat /proc/vcap300/vcap0/presmooth/param

=== [CH#00] Presmooth Parameter ===
```

```
<Path#0>
[00]VCAP_PRESMO_NONE_AUTO : 0x0
[01]VCAP_PRESMO_H_STRENGTH: 0x3
[02]VCAP_PRESMO_V_STRENGTH: 0x5
```

- Set the param\_id value to switch the parameter value of path presmooth  
**echo [param\_id] [value] > /proc/vcap300/vcap0/presmooth/param**  
param\_id:

0: VCAP\_PRESMO\_NONE\_AUTO, disable/enable auto presmooth (1: Disable, 0: Enable)  
1: VCAP\_PRESMO\_H\_STRENGTH, horizontal strength 0 ~ 7  
2: VCAP\_PRESMO\_V\_STRENGTH, vertical strength 0 ~ 7

## 10.9 /proc/vcap300/input\_module Proc Nodes

The capture supports various input devices. Users can use these proc nodes to display the input device drivers that have been installed. After inserting the vcap300\_xxx.ko input module driver, the xxx proc node will be created and users can switch the device operation through these nodes.

```
/# cat /proc/vcap300/input_module/
generic.0/ generic.1/ table

/# cat /proc/vcap300/input_module/generic.0/norm
Norm: 640 x 480
```

### 10.9.1 table

Users can use the /proc/vcap300/input\_module/table node to display the input device and the capture video interface mapping information.

Usage:

- Get the current value  
**cat /proc/vcap300/input\_module/table**

```
/# cat /proc/vcap300/input_module/table
| VI#   Name                Type                Interface          Resolution    FPS_C    FPS_M    XCAP#
```

=====							
0	generic.0	generic	progressive	640x480	30	30	XCAP0
1	generic.1	generic	isp	1280x720	30	30	ISP

Name	Description
VI	Video interface number
Name	font_end device name generic.x => x: device id
Type	Font_end device type generic decoder sensor isp sdi cvi ahd tvi
Interface	Video port interface format Interlace Progressive RGB888 ISP
Resolution	Video interface input signal resolution
FPS_C	Current frame rate
FPS_M	Maximus frame rate
XCAP#	Font_end device video port link to X_CAP# XCAP#0 ~ XCAP#1 ISP

## 10.10 /proc/vcap300/vcap0/diagnostic Proc Nodes

The capture provides much useful error status for diagnostic hardware. Users can get channel and global diagnostic information from these proc nodes.

### 10.10.1 global

Users can use the /proc/vcap300/vcap0/diagnostic/global node to display hardware global diagnostic information.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/diagnostic/global**

```
===== [Global Diagnostic] =====
SD job count overflow           : 0
MD miss statistics done         : 0
MD job count overflow           : 0
LLI channel id mismatch         : 0
LLI command load too late       : 0
Fatal reset                     : 0
MD      reset                   : 0
DMA#0 overflow                  : 0
DMA#0 job count overflow         : 0
DMA#0 write response fail       : 0
DMA#0 read  response fail       : 0
DMA#0 commad prefix error       : 0
DMA#0 write block width zero: 0
DMA#1 overflow                  : 0
DMA#1 job count overflow         : 0
DMA#1 write response fail       : 0
DMA#1 commad prefix error       : 0
DMA#1 write block width zero: 0
VI#0 no clock                   : 0
VI#1 no clock                   : 0
```

Name	Description
SD job count overflow	Scaler engine job count overflow (Capture Fatal Error) 1. Scaler loading overflows or the bandwidth not enough Loss of frame on this error
MD miss statistics done	Motion detection engine miss statistics 1. Bus loading overflows or the bandwidth is not enough. 2. Some channel video signals are not stable. MD event does not update this error.
MD job count overflow	Motion detection engine job count overflow (MD Fatal Error) 1. Bus loading overflows or the bandwidth is not enough. 2. Some channel video signals are not stable. MD event does not update this error.
LLI channel id mismatch	Link-list engine detect channel id mismatch in channel command table Loss of frame on this error
LLI command load too late	Link-list engine load command table not completed before frame start 1. Video signal vertical blanking is not enough. Loss of frame on this error
DMA#0/1 overflow	Capture DMA overflow (Capture Fatal Error) 1. Bus loading overflows or the bandwidth is not enough. Please try to adjust the DDR priority or decrease the used channel. Frame loss some line on this error
DMA#0/1 job count overflow	Capture DMA job count overflow 1. Bus loading overflows or the bandwidth is not enough. Frame Loss some line on this error
DMA#0/1 write response fail	Capture DMA write channel response fail Loss of frame on this error
DMA#0 commad prefix error	Capture DMA command prefix error Loss of frame on this error
DMA#0/1 write block width zero	Capture DMA write zero block length Loss of frame on this error

Name	Description
VI#0~8 no clock	Capture hardware detects no pixel clock input or clock is not stable in the video interface.
	Loss of frame on this error

## 10.10.2 channel

Users can use the `/proc/vcap300/vcap0/diagnostic/channel` node to display hardware channel diagnostic information.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/diagnostic/channel**

```
===== [CH#0 Diagnostic] =====
VI fifo full           : 0
Pixel lack             : 0
Line lack              : 0
SD timeout             : 0
SD job overflow        : 0
SD SC memory overflow  : 0
SD line lack           : 0
SD pixel lack          : 0
SD parameter error     : 0
SD prefix decode error : 0
SD 1st field           : 0
MD read traffic jam    : 0
LL id mismatch         : 0
LL DMA no done         : 0
LL Split DMA no done   : 0
LL jump table update   : 0
LL Null table mismatch : 0
LL Null table not zero : 0
LL table lack          : 0      0      0      0
No_Job alarm           : 3      0      0      0
Job Timeout            : 0
```

- Set the value to switch the channel number

**echo [value] > /proc/vcap300/vcap0/diagnostic/channel**

Value: 0 ~ 4, others are for all channels.

Name	Description
VI fifo full	<p>Video interface FIFO full</p> <p>1. Video signal is not stable or the format is not recognized.</p> <p>Loss of frame on this error</p>
Pixel lack	<p>Video signal pixel lack (Recoverable)</p> <p>1. Video signal is not stable (Please check the output signal of the front-end video port)</p> <p>2. Video signal width does not match the video interface setting.</p> <p>Users can dump /proc/vcap300/vcap0/channel to check the src_w value.</p> <p>Frame loss some pixel on this error</p>
Line lack	<p>Video signal line lack</p> <p>1. Video signal is not stable (Please check the output signal of the front-end video port)</p> <p>2. Video signal height does not match the video interface setting.</p> <p>Users can dump /proc/vcap300/vcap0/channel to check the src_h value.</p> <p>Frame loss some line on this error</p>
SD timeout	<p>Scaler engine timeout</p> <p>Loss of frame on this error</p>
SD job overflow	<p>Scaler engine job overflow</p> <p>1. Scaler loading overflows or the bandwidth is not enough.</p> <p>Loss of frame on this error</p>
SD SC memory overflow	<p>Source cropping engine memory overflow</p> <p>1. Source cropping engine bandwidth overflow</p> <p>Loss of frame on this error</p>

Name	Description
SD line lack	<p>Scaler engine detect line lack</p> <ol style="list-style-type: none"> <li>1. Video signal is not stable (Please check the output signal of the front-end video port.)</li> <li>2. Video signal height does not match the video interface setting.</li> </ol> <p>Users can dump /proc/vcap300/vcap0/channel to check the src_h value.</p> <p>Frame loss some line on this error</p>
SD pixel lack	<p>Scaler engine detect pixel lack (Fatal pixel lack)</p> <ol style="list-style-type: none"> <li>1. Video signal is not stable (Please check output signal of the front-end video port.)</li> <li>2. Video signal width does not match the video interface setting.</li> </ol> <p>Users can dump /proc/vcap300/vcap0/channel to check the src_w value.</p> <p>Frame loss some pixel on this error</p>
SD parameter error	<p>Scaler engine parameter error</p> <p>Loss of frame on this error</p>
SD prefix decode error	<p>Scaler engine prefix decode error</p> <p>Loss of frame on this error</p>
SD 1st field	<p>Scaler engine detect first field error</p> <ol style="list-style-type: none"> <li>1. Video signal is not stable (Please check the output signal of the front-end video port.)</li> <li>2. Video signal resolution does not match the video interface setting.</li> </ol> <p>Loss of frame on this error</p>
MD read traffic jam	<p>Motion detection engine read traffic jam</p> <ol style="list-style-type: none"> <li>1. Bus bandwidth is not enough.</li> </ol> <p>MD event not update on this error</p>
LL id mismatch	<p>Link list channel id mismatch</p> <p>Loss of frame on this error</p>
LL DMA no done	<p>Link list DMA not write done</p> <p>Loss of frame on this error</p>
LL Split DMA no done	<p>Link list split channel DMA not write done</p> <p>Loss of frame on this error</p>



Name	Description
LL jump table update	Software detect link list engine update status to wrong table Loss of frame on this error
LL Null table mismatch	Software detect null table of link-list not matched Loss of frame on this error
LL Null table not zero	Software detect null table of link-list not zero value Loss of frame on this error
LL table lack	Software detect global update table of link-list not enough. Channel property update will be pending until the global update table is available. Loss of frame on this error
No_Job alarm	No frame buffer available, means path will drop frame Loss of frame on this error
Job Timeout	Unable to grab frame, video signal grab timeout Signal timeout second depends on the signal frame rate. $25\text{FPS} \Rightarrow (1000(\text{ms})/25) * 2 = 80 (\text{ms})$ $30\text{FPS} \Rightarrow (1000(\text{ms})/30) * 2 = 66 (\text{ms})$ Loss of frame on this error

### 10.10.3 frame\_cnt

Users can use the `/proc/vcap300/vcap0/diagnostic/frame_cnt` node to display channel frame count information.

Usage:

- Get the current value  
**cat /proc/vcap300/vcap0/diagnostic/frame\_cnt**

```
[CH#00]
Path      top      bottom
=====
0         4493      0
1         4494      0
2         4491      0
3         4491      0
HW_Count: 8988
```

- Set the value to switch the channel number

**echo [value] > /proc/vcap300/vcap0/diagnostic/channel**

Value: 0 ~ 4, others are for all channels.

For the frame rate statistic of each path, users can type the below command to do sample calculation.

```
#> cat /proc/vcap300/vcap0/diagnostic/channel;sleep 10; cat
/proc/vcap300/vcap0/diagnostic/channel
```

```
[CH#00]
Path      top      bottom
=====
0         4493      0
1         4494      0
2         4491      0
3         4491      0
HW_Count: 8988
```

```
[CH#00]
Path      top      bottom
=====
0         4793      0
1         4644      0
2         4591      0
3         4491      0
HW_Count: 9688
```

Path#0 frame rate =>  $(4793 - 4493)/10(s) = 30 \text{ FPS}$ , no frame rate control

Path#1 frame rate =>  $(4644 - 4494)/10(s) = 15 \text{ FPS}$ , do frame rate control 15/30

Path#2 frame rate =>  $(4591 - 4491)/10(s) = 10 \text{ FPS}$ , do frame rate control 10/30

Path#3 frame rate =>  $(4491 - 4491)/10(s) = 0 \text{ FPS}$ , path stopped or no frame buffer

The frame rate depend one frame buffer available count, if the buffer not enough the frame will be loss to grab. In the other hand, the real time frame rate statistic will be incorrect. User can check the `/proc/vcap300/vcap0/diagnostic/channel` to monitor "No Job alarm" counter for frame buffer unavailable event.

#### 10.10.4 clear

Users can use the `/proc/vcap300/vcap0/diagnostic/clear` node to clear diagnostic status.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/diagnostic/clear**

```
Clear Diagnostic Data
1: clear all
2: clear global
3: clear channel
```

- Set the value to clear diagnostic status

**echo [value] > /proc/vcap300/vcap0/diagnostic/clear**

Value:

- 1: Clear all diagnostic counters
- 2: Clear global diagnostic counter
- 3: Clear channel diagnostic counter

#### 10.11 /proc/vcap300/vcap0/dbg\_mode Proc Node

Users can use the `/proc/vcap300/vcap0/dbg_mode` node to disable/enable error message output to UART for debug capture hardware.

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/dbg\_mode**

```
Debug_Mode: 0
0: disable error message output
1: enable error message output
```

- Set the value to switch debug mode

**echo [value] > /proc/vcap300/vcap0/dbg\_mode**

Value:

- 0: Disable the error message output
- 1: Enable the error message output

## 10.12 /proc/vcap300/vcap0/crop\_rule Proc Node

Users can use the /proc/vcap300/vcap0/crop\_rule node to display source cropping rule information. Users can modify or add rule through module parameter "hcrop\_rule".

Usage:

- Get the current value

**cat /proc/vcap300/vcap0/crop\_rule**

```
Rule#  H_In    H_Out
-----
0      720     704
1      -      -
2      -      -
```

## 10.13 /proc/vcap300/vcap0/grab\_filter Proc Node

Users can use the /proc/vcap300/vcap0/grab\_filter node to display and set filter for fail frame drop rule.

Usage:

- Get the current value

### **cat /proc/vcap300/vcap0/grab\_filter**

```
Grab Filter => 0x00000040
```

```
-----
```

```
-[BIT0]: SD_JOB_OVF  
-[BIT1]: SD_SC_MEM_OVF  
-[BIT2]: SD_PARAM_ERR  
-[BIT3]: SD_PREFIX_ERR  
-[BIT4]: SD_TIMEOUT  
-[BIT5]: SD_PIXEL_LACK  
*[BIT6]: SD_LINE_LACK
```

- Set the value to setup garb filter mask

### **echo [value] > /proc/vcap300/vcap0/grab\_filter**

Value:

BIT0: SD\_JOB\_OVF, scaler job count overflow

BIT1: SD\_SC\_MEM\_OVF, scaler source cropping memory overflow

BIT2: SD\_PARAM\_ERR, scaler parameter error

BIT3: SD\_PREFIX\_ERR, scaler prefix decode error

BIT4: SD\_TIMEOUT, scaler timeout

BIT5: SD\_PIXEL\_LACK, scaler pixel lack

BIT6: SD\_LINE\_LACK, scaler line lack

0: Enable fail frame drop

1: Disable fail frame drop

## **10.14 /proc/vcap300/vcap0/vi\_probe Proc Node**

Users can use the /proc/vcap300/vcap0/vi\_probe node to measure resolution of video signal.

Usage:

- Get the current value

### **cat /proc/vcap300/vcap0/vi\_probe**

```
Mode: NONE          [0:NONE 1:ACTIVE_REGION]  
VI   : 0            [0 ~ 1]
```

- Set the mode and VI number to disable/enable VI probe function.

**echo [mode] [VI#]> /proc/vcap300/vcap0/vi\_probe**

```
#> echo 1 0 > /proc/vcap300/vcap0/vi_probe
[VCAP_INFO]: VI#0 Signal Probe Active => Width:720 Height:288 (T:48 ms)
[VCAP_INFO]: VI#0 Signal Probe Active => Width:720 Height:288 (T:48 ms)
[VCAP_INFO]: VI#0 Signal Probe Active => Width:720 Height:288 (T:32 ms)
[VCAP_INFO]: VI#0 Signal Probe Active => Width:720 Height:288 (T:48 ms)
#> echo 0 0 > /proc/vcap300/vcap0/vi_probe
```

Because the hardware limitations, the driver must stop VI to grab the frame after enabling the VI probe function.

# Chapter 11

## Front-end Driver Module

---

This chapter contains the following sections:

- 11.1 TW2968 Driver Module
- 11.2 NVP1118 Driver Module
- 11.3 NVP1918 Driver Module
- 11.4 CX26848 Driver Module
- 11.5 TW9900 Driver Module
- 11.6 TW2868 Driver Module
- 11.7 TW2964 Driver Module
- 11.8 NVP1918C Driver Module
- 11.9 NVP1914 Driver Module
- 11.10 NVP1914C Driver Module
- 11.11 RN6318 Driver Module
- 11.12 RN6314 Driver Module
- 11.13 CX25930 Driver Module
- 11.14 MV4101 Driver Module
- 11.15 DH9901 Driver Module

- 11.16 NVP6114 Driver Module
- 11.17 TP2802 Driver Module
- 11.18 MT9M131 Driver Module
- 11.19 MT9D131 Driver Module
- 11.20 HM1375 Driver Module
- 11.21 OV7725 Driver Module



The capture input module driver depends on the related front end driver module. Users must insert the front end driver before the capture driver module. In the GM8136 SDK release package, users can find the source code and kernel module of the front end from "/module/front\_end/". The front end driver provides the audio and video common initialization. Users can control the front-end device number and the external clock output frequency of platform through the module parameters.

## 11.1 TW2968 Driver Module

Table 11-1 lists and describes the parameters when inserting this module.

**Table 11-1. Parameters of TW2968 Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configurations.
iaddr	{0x50, 0x52, 0x54, 0x56}	Device I <sup>2</sup> C address
vmode	{2, 2, 2, 2}	Video output mode selection 0: NTSC 720H 1CH 1: NTSC 720H 2CH 2: NTSC 720H 4CH 3: NTSC 960H 1CH 4: NTSC 960H 2CH 5: NTSC 960H 4CH 6: PAL 720H 1CH 7: PAL 720H 2CH 8: PAL 720H 4CH 9: PAL 960H 1CH 10: PAL 960H 2CH 11: PAL 960H 4CH

Name	Default Value	Description
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1(Not supported) <b>-1: It means that do not change the current frequency settings.</b>
clk_src	0	Clock source of the external clock port 0: PLL1OUT1 (Not supported) 1: PLL1OUT1/2 (Not supported) 2: PLL4OUT2 (Not supported) 3: PLL4OUT1/2 (Not supported) 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as the device RSTB pin 0: Not used 1: X_CAP_RST as RSTB pin for the device hardware reset
sample_rate	0	Audio sample rate 0: 8K 1: 16K 2: 32K 3: 44.1K 5: 48K
sample_size	0	Audio sample size 0: 16bits 1: 8bits
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module tw2968 driver for the video interface to switch configuration.

Name	Default Value	Description
ch_map	0	Device video channel mapping table 0: GM8210 socket board, vport 4CH mode GM8287 socket board 1: GM8210 system board, vport 4CH mode 2: GM8287 system board, vport 4CH mode GM8283 socket board 3: Generic, vport 1CH mode 4: Generic, vport 2CH mode Users must select the mapping table based on the hardware circuit.
audio_chnum	8	Device audio channel number

### 11.1.1 TW2968 Proc Nodes

The tw2968 driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation/sharpness through these nodes. A sample of these nodes is listed below.

/# cat /proc/tw2968.0/				
brightness	hue	saturation_u	sharpness	vmode
contrast	output_ch	saturation_v	status	volume
/# cat /proc/tw2968.1/				
brightness	hue	saturation_u	sharpness	vmode
contrast	output_ch	saturation_v	status	volume
/# cat /proc/tw2968.2/				
brightness	hue	saturation_u	sharpness	vmode
contrast	output_ch	saturation_v	status	volume
/# cat /proc/tw2968.3/				
brightness	hue	saturation_u	sharpness	vmode
contrast	output_ch	saturation_v	status	volume

Figure 11-1. Proc Nodes of TW2968 Module

The following table shows the proc nodes of the tw2968 module.

/proc/tw2968.0	brightness
/proc/tw2968.1	contrast
/proc/tw2968.2	hue
/proc/tw2968.3	saturation_u
	saturation_v
	sharpness
	output_ch
	volume
	status
	vmode

### 11.1.1.1 brightness

Users can use the /proc/tw2968.x/brightness node to display and setup brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2968.x/brightness**

```

/# cat /proc/tw2968.0/brightness

[TW2968#0]
-----
VIN#      VCH#      BRIGHTNESS
-----
0         0         0x00
1         1         0x00
2         2         0x00
3         3         0x00
4         4         0x00
5         5         0x00
6         6         0x00
7         7         0x00

Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1

```

- Set the value to switch brightness

**echo [vin] [value] > /proc/tw2968.x/brightness**

vin: 0 ~ 7, others for all VIN

Value: 0x00 ~ 0xff

### 11.1.1.2 contrast

Users can use the /proc/tw2968.x/contrast node to display and setup contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2968.x/contrast**

```
/# cat /proc/tw2968.0/contrast

-----
VIN#      VCH#      CONTRAST
-----
0         0         0x64
1         1         0x64
2         2         0x64
3         3         0x64
4         4         0x64
5         5         0x64
6         6         0x64
7         7         0x64

Contrast[0% ~ 255%] ==> 0x00=0%, 0xff=255%
```

- Set the value to switch contrast

**echo [vin] [value] > /proc/tw2968.x/contrast**

vin: 0 ~ 7, others for all VIN

Value: 0x00 ~ 0xff

### 11.1.1.3 hue

Users can use the `/proc/tw2968.x/hue` node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2968.x/hue**

```
/# cat /proc/tw2968.0/hue

[TW2968#0]
-----
VIN#      VCH#      HUE
-----
0         0         0x00
1         1         0x00
2         2         0x00
3         3         0x00
4         4         0x00
5         5         0x00
6         6         0x00
7         7         0x00

Hue[0x00 ~ 0xff] ==> 0x00=0, 0x7f=90, 0x80=-90
```

- Set the value to switch hue

**echo [vin] [value] > /proc/tw2968.x/hue**

vin: 0 ~ 7, others for all VIN

Value: 0x00 ~ 0xff

### 11.1.1.4 saturation\_u

Users can use the `/proc/tw2968.x/saturation_u` node to display and setup saturation\_u of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2968.x/saturation\_u**

```

/# cat /proc/tw2968.0/saturation_u

[TW2968#0]
-----
VIN#      VCH#      SATURATION_U
-----
0         0         0x80
1         1         0x80
2         2         0x80
3         3         0x80
4         4         0x80
5         5         0x80
6         6         0x80
7         7         0x80

Saturation_U[0% ~ 200%] ==> 0x80=100%

```

- Set the value to switch saturation\_u  
**echo [vin] [value] > /proc/tw2968.x/saturation\_u**  
 vin: 0 ~ 7, others for all VIN  
 Value: 0x00 ~ 0xff

#### 11.1.1.5 saturation\_v

Users can use the /proc/tw2968.x/saturation\_v node to display and setup saturation\_v of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/tw2968.x/saturation\_v**

```

/# cat /proc/tw2968.0/saturation_v

[TW2968#0]
-----
VIN#      VCH#      SATURATION_V
-----
0         0         0x80
1         1         0x80
2         2         0x80
3         3         0x80
4         4         0x80
5         5         0x80
6         6         0x80
7         7         0x80

Saturation_V[0% ~ 200%] ==> 0x80=100%

```

- Set the value to switch saturation\_v  
**echo [vin] [value] > /proc/tw2968.x/saturation\_v**  
 vin: 0 ~ 7, others for all VIN  
 Value: 0x00 ~ 0xff

#### 11.1.1.6 status

Users can use the /proc/tw2968.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/tw2968.x/status**



```
/# cat /proc/tw2968.0/status
```

```
[TW2968#0]
```

```
-----  
VIN#      VCH#      NOVID  
-----  
0          0          Video_On  
1          1          Video_Loss  
2          2          Video_Loss  
3          3          Video_Loss  
4          4          Video_Loss  
5          5          Video_Loss  
6          6          Video_Loss  
7          7          Video_Loss
```

### 11.1.1.7 vmode

Users can use the `/proc/tw2968.x/vmode` node to display the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2968.x/vmode**

```
/# cat /proc/tw2968.0/vmode
```

```
[TW2968#0]
```

```
00: NTSC_720H_1CH
```

```
01: NTSC_720H_2CH
```

```
02: NTSC_720H_4CH
```

```
03: NTSC_960H_1CH
```

```
04: NTSC_960H_2CH
```

```
05: NTSC_960H_4CH
```

```
06: PAL_720H_1CH
```

```
07: PAL_720H_2CH
```

```
08: PAL_720H_4CH
```

```
09: PAL_960H_1CH
```

```
10: PAL_960H_2CH
```

```
11: PAL_960H_4CH
```

```
-----
```

```
Current==> NTSC_720H_4CH
```

- Set the value to switch device vport output mode

**echo [mode] > /proc/tw2968.x/vmode**

mode: 0 ~ 11

### 11.1.1.8 sharpness

Users can use the /proc/tw2968.x/sharpness node to display and setup sharpness of each channel

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2968.x/sharpness**

```

/# cat /proc/tw2968.0/sharpness

[TW2968#0]
-----
VIN#      VCH#      SHARPNESS
-----
0         0         0x1
1         1         0x1
2         2         0x1
3         3         0x1
4         4         0x1
5         5         0x1
6         6         0x1
7         7         0x1

Sharpness[0x0 ~ 0xf] - (16 levels) ==> 0x0:no effect, 0x1~0xf:sharpness enhancement
('0xf' being the strongest)

```

- Set the value to switch sharpness  
**echo [vin] [value] > /proc/tw2968.x/sharpness**  
 vin: 0 ~ 7, other for all channels  
 Value: 0x0 ~ 0xf

### 11.1.1.9 volume

Users can use the /proc/tw2968.x/volume node to display and setup audio volume

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/tw2968.x/volume**

```
/# cat /proc/tw2968.0/volume  
  
[tw2968#0]  
Volume[0x0~0xf] = 8
```

- Set the value to switch audio volume  
**echo [value] > /proc/tw2968.x/volume**  
Value: 0x0 ~ 0xf

#### 11.1.1.10 output\_ch

Users can use the /proc/tw2968.x/output\_ch node to display and setup audio playback output channel.

Usage:

- Get the current value, x is ranging from 0 to 3.  
**cat /proc/tw2968.x/output\_ch**

```
/# cat /proc/tw2968.0/output_ch  
  
[tw2968#0]  
Current[0x0~0x18]==> PLAYBACK first stage
```

- Set the value to switch audio playback output channel  
**echo [value] > /proc/tw2968.x/output\_ch**  
Value: 0x0 ~ 0x18

## 11.2 NVP1118 Driver Module

Table 11-2 lists and describes the parameters when inserting this module.

**Table 11-2. Parameters of NVP1118 Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable I <sup>2</sup> C bus support in the kernel configuration.
iaddr	{0x66, 0x64, 0x62, 0x60}	Device I <sup>2</sup> C address
vmode	{2, 2, 2, 2}	Video output mode selection 0: NTSC 720H 1CH 1: NTSC 720H 2CH 2: NTSC 720H 4CH 3: PAL 720H 1CH 4: PAL 720H 2CH 5: PAL 720H 4CH
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 (Not supported) <b>-1: It means that do not change the current frequency settings.</b>
clk_src	0	Clock source of the external clock port 0: PLL1OUT1 (Not supported) 1: PLL1OUT1/2 (Not supported) 2: PLL4OUT2 (Not supported) 3: PLL4OUT1/2 (Not supported) 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as the device RSTB pin 0: Not used 1: X_CAP_RST as the RSTB pin for the device hardware reset

Name	Default Value	Description
sample_rate	0	Audio sample rate 0: 8K 1: 16K
sample_size	0	Audio sample size 0: 16 bits 1: 8 bits
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module nvp1118 driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: GM8210 socket board, vport 4CH mode GM8287 socket board 1: GM8210 system board, vport 4CH mode 2: GM8287 system board, vport 4CH mode GM8283 socket board <b>Users must select the mapping table based on the hardware circuit.</b>
audio_chnum	8	Device audio channel number

### 11.2.1 NVP1118 Proc Nodes

The nvp1118 driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/nvp1118.0/
brightness hue saturation status volume
contrast output_ch sharpness vmode

/# cat /proc/nvp1118.1/

```

```
brightness hue saturation status volume
contrast output_ch sharpness vmode

/# cat /proc/nvp1118.2/
brightness hue saturation status volume
contrast output_ch sharpness vmode

/# cat /proc/nvp1118.3/
brightness hue saturation status volume
contrast output_ch sharpness vmode
```

Figure 11-2. Proc Nodes of NVP1118 Module

The following table shows the proc nodes of the nvp1118 module.

/proc/nvp1118.0	brightness
/proc/nvp1118.1	contrast
/proc/nvp1118.2	hue
/proc/nvp1118.3	saturation
	sharpness
	output_ch
	volume
	status
	vmode

11.2.1.1 brightness

Users can use the /proc/nvp1118.x/brightness node to display and setup brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1118.x/brightness**

```
/# cat /proc/nvp1118.0/brightness

[NVP1118#0]
-----
VIN#      VCH#      BRIGHTNESS
```

```

-----
0      0      0xf8
1      1      0xf8
2      2      0xf8
3      3      0xf8
4      4      0xf8
5      5      0xf8
6      6      0xf8
7      7      0xf8

```

```
Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1
```

- Set the value to switch brightness  
**echo [vin] [value] > /proc/nvp118.x/brightness**  
 vin: 0 ~ 7, others for all VIN  
 Value: 0x00 ~ 0xff

### 11.2.1.2 contrast

Users can use the /proc/nvp118.x/contrast node to display and setup contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp118.x/contrast**

```

/# cat /proc/nvp118.0/contrast

[NVP1118#0]
-----
VIN#      VCH#      CONTRAST
-----
0          0          0x76
1          1          0x76
2          2          0x76
3          3          0x76

```



```

4      4      0x76
5      5      0x76
6      6      0x76
7      7      0x76

Contrast[0x00 ~ 0xff] ==> 0x00=x0, 0x40=x0.5, 0x80=x1, 0xff=x2

```

- Set the value to switch contrast  
**echo [vin] [value] > /proc/nvp118.x/contrast**  
 vin: 0 ~ 7, other for all VIN  
 Value: 0x00 ~ 0xff

### 11.2.1.3 hue

Users can use the /proc/nvp118.x/hue node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp118.x/hue**

```

/# cat /proc/nvp118.0/hue

[NVP1118#0]
-----
VIN#    VCH#    HUE
-----
0        0      0x01
1        1      0x01
2        2      0x01
3        3      0x01
4        4      0x01
5        5      0x01
6        6      0x01
7        7      0x01

```

```
Hue[0x00 ~ 0xff] ==> 0x00=0, 0x40=90, 0x80=180, 0xff=360
```

- Set the value to switch hue

**echo [vin] [value] > /proc/nvp118.x/hue**

vin: 0 ~ 7, others for all VIN

Value: 0x00 ~ 0xff

#### 11.2.1.4 saturation

Users can use the /proc/nvp118.x/saturation node to display and setup saturation of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp118.x/saturation**

```
/# cat /proc/nvp118.0/saturation

[NVP1118#0]
-----
VIN#      VCH#      SATURATION
-----
0          0          0x80
1          1          0x80
2          2          0x80
3          3          0x80
4          4          0x80
5          5          0x80
6          6          0x80
7          7          0x80

Saturation[0x00 ~ 0xff] ==> 0x00=x0, 0x80=x1, 0xc0=x1.5, 0xff=x2
```

- Set the value to switch saturation

**echo [vin] [value] > /proc/nvp118.x/saturation**

vin: 0 ~ 7, others for all VIN

Value: 0x00 ~ 0xff

### 11.2.1.5 status

Users can use the `/proc/nvp1118.x/status` node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1118.x/status**

```
/# cat /proc/nvp1118.0/status

[NVP1118#0]
-----
VIN#      VCH#      NOVID
-----
0          0          Video_On
1          1          Video_Loss
2          2          Video_Loss
3          3          Video_Loss
4          4          Video_Loss
5          5          Video_Loss
6          6          Video_Loss
7          7          Video_Loss
```

### 11.2.1.6 vmode

Users can use the `/proc/nvp1118.x/vmode` node to display the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1118.x/vmode**

```
/# cat /proc/nvp1118.0/vmode
```

```
[NVP1118#0]
```

```
02: NTSC_720H_4CH
```

```
05: PAL_720H_4CH
```

```
-----
```

```
Current==> NTSC_720H_4CH
```

- Set the value to switch device video port output mode

**echo [mode] > /proc/nvp1118.x/vmode**

mode: 2 or 5

### 11.2.1.7 sharpness

Users can use the /proc/nvp1118.x/sharpness node to display and setup sharpness of each channel

Usage:

- Get the current value, x is ranging from 0 to 3.

**cat /proc/nvp1118.x/sharpness**

```

/# cat /proc/nvp1118.0/sharpness

[NVP1118#0]
-----
VIN#      VCH#      SHARPNESS
-----
0         8         0x80
1         9         0x80
2         10        0x80
3         11        0x80
4         12        0x80
5         13        0x80
6         14        0x80
7         15        0x80

H_Sharpness[0x0 ~ 0xf] - Bit[7:4] ==> 0x0:x0, 0x4:x0.5, 0x8:x1, 0xf:x2
V_Sharpness[0x0 ~ 0xf] - Bit[3:0] ==> 0x0:x1, 0x4:x2, 0x8:x3, 0xf:x4

```

- Set the value to switch sharpness  
**echo [vin] [value] > /proc/nvp1118.x/sharpness**  
 vin: 0 ~ 7, others are for all channels.  
 Value: 0x00~ 0xff

### 11.2.1.8 volume

Users can use the /proc/nvp1118.x/volume node to display and setup audio volume

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/nvp1118.x/volume**

```
/# cat /proc/nvp1118.0/volume
```

```
[NVP1118#0]
```

```
Volume[0x0~0xf] = 8
```

- Set the value to switch audio volume

**echo [value] > /proc/nvp1118.x/volume**

Value: 0x0 ~ 0xf

### 11.2.1.9 output\_ch

Users can use the /proc/nvp1118.x/output\_ch node to display and setup audio playback output channel.

Usage:

- Get the current value, x is ranging from 0 to 3.

**cat /proc/nvp1118.x/output\_ch**

```
/# cat /proc/nvp1118.0/output_ch
```

```
[NVP1118#0]
```

```
Current[0x0~0x18]==> FIRST PLAYBACK AUDIO
```

- Set the value to switch audio playback output channel

**echo [value] > /proc/nvp1118.x/output\_ch**

Value: 0x0 ~ 0x18

## 11.3 NVP1918 Driver Module

Table 11-3 lists and describes the module parameters when inserting this driver.

**Table 11-3. Parameters of NVP1918 Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configuration.
iaddr	{0x66, 0x64, 0x62, 0x60}	Device I <sup>2</sup> C address
vmode	{2, 2, 2, 2}	Video output mode selection 0: NTSC 720H 1CH 1: NTSC 720H 2CH 2: NTSC 720H 4CH 3: NTSC 960H 1CH 4: NTSC 960H 2CH 5: NTSC 960H 4CH 6: PAL 720H 1CH 7: PAL 720H 2CH 8: PAL 720H 4CH 9: PAL 960H 1CH 10: PAL 960H 2CH 11: PAL 960H 4CH
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 (Not supported) <b>-1: It means that do not change the current frequency settings.</b>
clk_src	0	Clock source of the external clock port 0: PLL1OUT1 (Not supported) 1: PLL1OUT1/2 (Not supported) 2: PLL4OUT2 (Not supported) 3: PLL4OUT1/2 (Not supported) 4: PLL3

Name	Default Value	Description
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: GPIO#33 as RSTB pin for the device hardware reset
sample_rate	0	Audio sample rate 0: 8K 1: 16K
sample_size	0	Audio sample size 0: 16bits 1: 8bits
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture input_module nvp1918 driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: GM8210 socket board, vport 4CH mode GM8287 socket board 1: GM8210 system board, vport 4CH mode 2: GM8287 system board, vport 4CH mode GM8283 socket board Users must select the mapping table based on the hardware circuit.
audio_chnum	8	Device audio channel number



### 11.3.1 NVP1918 Proc Nodes

The nvp1918 driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation/sharpness through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/nvp1918.0/
brightness hue saturation status volume
contrast output_ch sharpness vmode

/# cat /proc/nvp1918.1/
brightness hue saturation status volume
contrast output_ch sharpness vmode

/# cat /proc/nvp1918.2/
brightness hue saturation status volume
contrast output_ch sharpness vmode

/# cat /proc/nvp1918.3/
brightness hue saturation status volume
contrast output_ch sharpness vmode

```

Figure 11-3. Proc Nodes of NVP1918 Module

The following table shows the proc nodes of the nvp1918 module.

/proc/nvp1918.0	brightness
/proc/nvp1918.1	contrast
/proc/nvp1918.2	hue
/proc/nvp1918.3	saturation
	sharpness
	volume
	output_ch
	status
	vmode

### 11.3.1.1 brightness

Users can use the `/proc/nvp1918.x/brightness` node to display and setup brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1918.x/brightness**

```
/# cat /proc/nvp1918.0/brightness

[NVP1918#0]
-----
VIN#      VCH#      BRIGHTNESS
-----
0          0          0xf8
1          1          0xf8
2          2          0xf8
3          3          0xf8
4          4          0xf8
5          5          0xf8
6          6          0xf8
7          7          0xf8

Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1
```

- Set the value to switch brightness

**echo [vin] [value] > /proc/nvp1918.x/brightness**

vin: 0 ~ 7, others for all VIN

Value: 0x00 ~ 0xff

### 11.3.1.2 contrast

Users can use the `/proc/nvp1918.x/contrast` node to display and setup contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

### **cat /proc/nvp1918.x/contrast**

```
/# cat /proc/nvp1918.0/contrast

[NVP1918#0]
-----
VIN#      VCH#      CONTRAST
-----
0         0         0x76
1         1         0x76
2         2         0x76
3         3         0x76
4         4         0x76
5         5         0x76
6         6         0x76
7         7         0x76

Contrast[0x00 ~ 0xff] ==> 0x00=x0, 0x40=x0.5, 0x80=x1, 0xff=x2
```

- Set the value to switch contrast

**echo [vin] [value] > /proc/nvp1918.x/contrast**

vin: 0 ~ 7, others for all VIN

Value: 0x00 ~ 0xff

#### **11.3.1.3 hue**

Users can use the /proc/nvp1918.x/hue node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1918.x/hue**

```
/# cat /proc/nvp1918.0/hue
```

```
[NVP1918#0]
```

```
-----  
VIN#      VCH#      HUE  
-----
```

```
0          0          0x01
```

```
1          1          0x01
```

```
2          2          0x01
```

```
3          3          0x01
```

```
4          4          0x01
```

```
5          5          0x01
```

```
6          6          0x01
```

```
7          7          0x01
```

```
Hue[0x00 ~ 0xff] ==> 0x00=0, 0x40=90, 0x80=180, 0xff=360
```

- Set the value to switch hue

**echo [vin] [value] > /proc/nvp1918.x/hue**

vin: 0 ~ 7, others for all VIN

Value: 0x00 ~ 0xff

#### 11.3.1.4 saturation

Users can use the /proc/nvp1918.x/saturation node to display and setup saturation of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1918.x/saturation**

```

/# cat /proc/nvp1918.0/saturation

[NVP1918#0]
-----
VIN#      VCH#      SATURATION
-----
0         0         0x80
1         1         0x80
2         2         0x80
3         3         0x80
4         4         0x80
5         5         0x80
6         6         0x80
7         7         0x80

Saturation[0x00 ~ 0xff] ==> 0x00=x0, 0x80=x1, 0xc0=x1.5, 0xff=x2

```

- Set the value to switch saturation  
**echo [vin] [value] > /proc/nvp1918.x/saturation**  
 vin: 0 ~ 7, others for all channels  
 Value: 0x00 ~ 0xff

### 11.3.1.5 status

Users can use the /proc/nvp1918.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/nvp1918.x/status**

```
/# cat /proc/nvp1918.0/status
```

```
[NVP1918#0]
```

```
-----  
VIN#      VCH#      NOVID  
-----  
0          0          Video_On  
1          1          Video_Loss  
2          2          Video_Loss  
3          3          Video_Loss  
4          4          Video_Loss  
5          5          Video_Loss  
6          6          Video_Loss  
7          7          Video_Loss
```

### 11.3.1.6 vmode

Users can use the /proc/nvp1918.x/vmode node to display the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1918.x/vmode**

```
/# cat /proc/nvp1918.0/vmode
```

```
[NVP1918#0]
```

```
02: NTSC_720H_4CH
```

```
05: NTSC_960H_4CH
```

```
08: PAL_720H_4CH
```

```
11: PAL_960H_4CH
```

```
-----  
Current==> NTSC_720H_4CH
```

- Set the value to switch device video port output mode

**echo [mode] > /proc/nvp1918.x/vmode**

mode: 2, 5, 8, 11

### 11.3.1.7 sharpness

Users can use the /proc/nvp1918.x/sharpness node to display and setup sharpness of each channel

Usage:

- Get the current value, x is ranging from 0 to 3.

**cat /proc/nvp1918.x/sharpness**

```

/# cat /proc/nvp1918.0/sharpness

[NVP1918#0]
-----
VIN#      VCH#      SHARPNESS
-----
0         8         0x80
1         9         0x80
2         10        0x80
3         11        0x80
4         12        0x80
5         13        0x80
6         14        0x80
7         15        0x80

H_Sharpness[0x0 ~ 0xf] - Bit[7:4] ==> 0x0:x0, 0x4:x0.5, 0x8:x1, 0xf:x2
V_Sharpness[0x0 ~ 0xf] - Bit[3:0] ==> 0x0:x1, 0x4:x2, 0x8:x3, 0xf:x4

```

- Set the value to switch sharpness

**echo [vin] [value] > /proc/nvp1918.x/sharpness**

vin: 0 ~ 7, other for all channels

Value: 0x00~ 0xff

### 11.3.1.8 volume

Users can use the /proc/nvp1918.x/volume node to display and setup audio volume

Usage:

- Get the current value, x is ranging from 0 to 3.

**cat /proc/nvp1918.x/volume**

```
/# cat /proc/nvp1918.0/volume  
  
[NVP1918#0]  
Volume[0x0~0xf] = 8
```

- Set the value to switch audio volume

**echo [value] > /proc/nvp1918.x/volume**

Value: 0x0 ~ 0xf

### 11.3.1.9 output\_ch

Users can use the /proc/nvp1918.x/output\_ch node to display and setup audio playback output channel.

Usage:

- Get the current value, x is ranging from 0 to 3.

**cat /proc/nvp1918.x/output\_ch**

```
/# cat /proc/nvp1918.0/output_ch  
  
[NVP1918#0]  
Current[0x0~0x18]==> FIRST PLAYBACK AUDIO
```

- Set the value to switch audio playback output channel

**echo [value] > /proc/nvp1918.x/output\_ch**

Value: 0x0 ~ 0x18



## 11.4 CX26848 Driver Module

Table 11-4 lists and describes the module parameters when inserting this driver.

**Table 11-4. Parameters of CX26848 Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configuration.
iaddr	{0x88, 0x8a, 0x8c, 0x8e}	Device I <sup>2</sup> C address
vmode	{2, 2, 2, 2}	Video output mode selection 0: NTSC 720H 1CH 1: NTSC 720H 2CH 2: NTSC 720H 4CH 3: NTSC 960H 1CH 4: NTSC 960H 2CH 5: NTSC 960H 4CH 6: PAL 720H 1CH 7: PAL 720H 2CH 8: PAL 720H 4CH 9: PAL 960H 1CH 10: PAL 960H 2CH 11: PAL 960H 4CH
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 (Not supported) -1: It means that do not change the current frequency settings.

Name	Default Value	Description
clk_src	3	<p>Clock source of the external clock port</p> <p>0: PLL1OUT1 (Not supported)</p> <p>1: PLL1OUT1/2 (Not supported)</p> <p>2: PLL4OUT2 (Not supported)</p> <p>3: PLL4OUT1/2 (Not supported)</p> <p>4: PLL3</p>
clk_freq	24000000	<p>Clock frequency of the external clock port</p> <p>xxx Hz</p>
rstb_used	1	<p>Control GPIO pin as device RSTB pin</p> <p>0: Not used</p> <p>1: X_CAP_RST as the RSTB pin for the device hardware reset</p>
notify	1	<p>Device notify</p> <p>0: Disable</p> <p>1: Enable</p> <p>Notify the video status and video format to capture the input_module cx26848 driver for the video interface to switch configuration.</p>
ch_map	0	<p>Device video channel mapping table</p> <p>0: GM8210 socket board, vport 4CH mode GM8287 socket board</p> <p>1: GM8210 system board, vport 4CH mode</p> <p>2: GM8287 system board, vport 4CH mode GM8283 socket board</p> <p>3: Generic, vport 1CH mode</p> <p>4: Generic, vport 2CH mode</p> <p>Users must select the mapping table based on the hardware circuit.</p>

### 11.4.1 CX26848 Proc Nodes

The cx26848 driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/cx26848.0/
brightness hue saturation vmode
contrast output_ch status volume

/# cat /proc/cx26848.1/
brightness hue saturation vmode
contrast output_ch status volume

/# cat /proc/cx26848.2/
brightness hue saturation vmode
contrast output_ch status volume

/# cat /proc/cx26848.3/
brightness hue saturation vmode
contrast output_ch status volume
```

Figure 11-4. Proc Nodes of CX26848 Module

The following table shows the proc nodes of the cx26848 module.

/proc/cx26848.0	brightness
/proc/cx26848.1	contrast
/proc/cx26848.2	hue
/proc/cx26848.3	saturation
	output_ch
	volume
	status
	vmode

### 11.4.1.1 brightness

Users can use the `/proc/cx26848.x/brightness` node to display and setup brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/cx26848.x/brightness**

```
/# cat /proc/cx26848.0/brightness

[ CX26848#0 ]
-----
VIN#      VCH#      BRIGHTNESS
-----
0          0          6000
1          1          6000
2          2          6000
3          3          6000
4          4          6000
5          5          6000
6          6          6000
7          7          6000

Brightness[0 ~ 10000]
```

- Set the value to switch brightness

**echo [vin] [value] > /proc/cx26848.x/brightness**

vin: 0 ~ 7, others for all VIN

Value: 0 ~ 10000

### 11.4.1.2 contrast

Users can use the `/proc/cx26848.x/contrast` node to display and setup contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

### **cat /proc/cx26848.x/contrast**

```
/# cat /proc/cx26848.0/contrast

[CX26848#0]
-----
VIN#      VCH#      CONTRAST
-----
0          0          5000
1          1          5000
2          2          5000
3          3          5000
4          4          5000
5          5          5000
6          6          5000
7          7          5000

Contrast[0 ~ 10000]
```

- Set the value to switch brightness

**echo [vin] [value] > /proc/cx26848.x/contrast**

vin: 0 ~ 7, others for all VIN

Value: 0 ~ 10000

#### **11.4.1.3 hue**

Users can use the /proc/cx26848.x/hue node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/cx26848.x/hue**

```
/# cat /proc/cx26848.0/hue

[CX26848#0]
-----
```

VIN#	VCH#	HUE
-----		
0	0	5000
1	1	5000
2	2	5000
3	3	5000
4	4	5000
5	5	5000
6	6	5000
7	7	5000
Hue[0 ~ 10000]		

- Set the value to switch hue  
**echo [vin] [value] > /proc/cx26848.x/hue**  
vin: 0 ~ 7, others for all VIN  
Value: 0 ~ 10000

#### 11.4.1.4 saturation

Users can use the /proc/cx26848.x/saturation node to display and setup saturation of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/cx26848.x/hue**

```

/# cat /proc/cx26848.0/ saturation

[CX26848#0]
-----
VIN#      VCH#      SATURATION
-----
0         0         5000
1         1         5000
2         2         5000

```

3	3	5000
4	4	5000
5	5	5000
6	6	5000
7	7	5000

Saturation[0 ~ 10000]

- Set the value to switch saturation  
**echo [vin] [value] > /proc/cx26848.x/saturation**  
 vin: 0 ~ 7, others for all VIN  
 Value: 0 ~ 10000

### 11.4.1.5 status

Users can use the /proc/cx26848.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/cx26848.x/status**

```
/# cat /proc/cx26848.0/status
```

```
[CX26848#0]
```

```
-----
```

```
VIN#      VCH#      NOVID
```

```
-----
```

0	0	Video_On
1	1	Video_Loss
2	2	Video_Loss
3	3	Video_Loss
4	4	Video_Loss
5	5	Video_Loss
6	6	Video_Loss
7	7	Video_Loss

### 11.4.1.6 vmode

Users can use the `/proc/cx26848.x/vmode` node to display the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/cx26848.x/vmode**

```
/#
cat /proc/cx26848.0/vmode

[ CX26848#0 ]
00: NTSC_720H_1CH
01: NTSC_720H_2CH
02: NTSC_720H_4CH
03: NTSC_960H_1CH
04: NTSC_960H_2CH
05: NTSC_960H_4CH
06: PAL_720H_1CH
07: PAL_720H_2CH
08: PAL_720H_4CH
09: PAL_960H_1CH
10: PAL_960H_2CH
11: PAL_960H_4CH
-----
Current==> NTSC_720H_4CH
```

- Set the value to switch device video port output mode

**echo [mode] > /proc/cx26848.x/vmode**

mode: 2, 5, 8, 11



### 11.4.1.7 volume

Users can use the `/proc/cx26848.x/volume` node to display and setup audio volume

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/cx26848.x/volume**

```
/# cat /proc/cx26848.0/volume  
  
[CX26848#0]  
Volume[0 ~ 63] = 16
```

- Set the value to switch audio volume

**echo [value] > /proc/cx26848.x/volume**

Value: 0 ~ 63

### 11.4.1.8 output\_ch

Users can use the `/proc/cx26848.x/output_ch` node to display and setup audio playback output channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/cx26848.x/output\_ch**

```
/# cat /proc/cx26848.0/output_ch  
  
[CX26848#0]
```

- Set the value to switch audio playback output channel

**echo [value] > /proc/cx26848.x/output\_ch**

## 11.5 TW9900 Driver Module

Table 11-5 lists and describes the module parameters when inserting this driver.

**Table 11-5. Parameters of TW9900 Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configuration
iaddr	{0x88, 0x8a, 0x8c, 0x8e}	Device I <sup>2</sup> C address
vmode	{0, 0, 0, 0}	Video output mode selection 0: NTSC 720H 1CH 1: PAL 720H 1CH
clk_used	0x1	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 (Not supported) <b>-1: It means that do not change the current frequency settings.</b>
clk_src	4	Clock source of the external clock port 0: PLL1OUT1 (Not supported) 1: PLL1OUT1/2 (Not supported) 2: PLL4OUT2 (Not supported) 3: PLL4OUT1/2 (Not supported) 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as the device RSTB pin 0: Not used 1: X_CAP_RST as the RSTB pin for the device hardware reset 2: GPIO_59 as RTSB pin

Name	Default Value	Description
clk_sscg	0	External Clock SSCG control 0: Disable 1: SSCR MR0 2: SSCG MR1 3: SSCG MR2 4: SSCG MR3
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module tw9900 driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: Generic, vport 1CH mode  Users must select the mapping table based on the hardware circuit.

### 11.5.1 TW9900 Proc Nodes

The tw9900 driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/tw9900.0/
brightness    hue          saturation_v  vmode
contrast      saturation_u  status

```

**Figure 11-5. Proc Nodes of TW9900 Module**

The following table shows the proc nodes of the tw2968 module.

/proc/tw9900.0	brightness
/proc/tw9900.1	contrast

/proc/tw9900.2	hue
/proc/tw9900.3	saturation_u
	saturation_v
	status
	vmode

### 11.5.1.1 brightness

Users can use the /proc/tw9900.x/brightness node to display and setup brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw9900.x/brightness**

```

/# cat /proc/tw9900.0/brightness

[TW9900#0]
-----
VIN#      VCH#      BRIGHTNESS
-----
0          0          0x00

Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1

```

- Set the value to switch brightness

**echo [value] > /proc/tw9900.x/brightness**

Value: 0x00 ~ 0xff

### 11.5.1.2 contrast

Users can use the /proc/tw9900.x/contrast node to display/setup contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw9900.x/contrast**

```

/# cat /proc/tw9900.0/contrast

[TW9900#0]
-----
VIN#      VCH#      CONTRAST
-----
0          0          0x64

Contrast[0% ~ 255%] ==> 0x00=0%, 0xff=255%

```

- Set the value to switch contrast  
**echo [value] > /proc/tw9900.x/contrast**  
Value: 0x00 ~ 0xff

### 11.5.1.3 hue

Users can use the /proc/tw9900.x/hue node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/tw9900.x/hue**

```

/# cat /proc/tw9900.0/hue

[TW9900#0]
-----
VIN#      VCH#      HUE
-----
0          0          0x00

Hue[0x00 ~ 0xff] ==> 0x00=0, 0x7f=90, 0x80=-90

```

- Set the value to switch hue  
**echo [value] > /proc/tw9900.x/hue**  
Value: 0x00 ~ 0xff

### 11.5.1.4 saturation\_u

Users can use the `/proc/tw9900.x/saturation_u` node to display and setup `saturation_u` of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw9900.x/saturation\_u**

```
/# cat /proc/tw9900.0/saturation_u

[TW9900#0]
-----
VIN#      VCH#      SATURATION_U
-----
0          0          0x80

Saturation_U[0% ~ 200%] ==> 0x80=100%
```

- Set the value to switch `saturation_u`

**echo [value] > /proc/tw9900.x/saturation\_u**

Value: 0x00 ~ 0xff

### 11.5.1.5 saturation\_v

Users can use the `/proc/tw9900.x/saturation_v` node to display and setup `saturation_v` of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw9900.x/saturation\_v**

```

/# cat /proc/tw9900.0/saturation_v

[TW9900#0]
-----
VIN#      VCH#      SATURATION_V
-----
0          0          0x80

Saturation_V[0% ~ 200%] ==> 0x80=100%

```

- Set the value to switch saturation\_v  
**echo [value] > /proc/tw9900.x/saturation\_v**  
Value: 0x00 ~ 0xff

### 11.5.1.6 status

Users can use the /proc/tw9900.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw9900.x/status**

```

/# cat /proc/tw9900.0/status

[TW9900#0]
-----
VIN#      VCH#      NOVID
-----
0          0          Video_Loss

```

### 11.5.1.7 vmode

Users can use the /proc/tw9900.x/vmode node to display the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**Cat /proc/tw9900.x/vmode**

```
/# cat /proc/tw2968.0/vmode

[TW9900#0]
00: NTSC_720H_1CH
01: PAL_720H_1CH
-----
Current==> NTSC_720H_1CH
```

- Set the value to switch device vport output mode

**echo [mode] > /proc/tw9900.x/vmode**

mode: 0 ~ 1

## 11.6 TW2868 Driver Module

Table 11-6 lists and describes the module parameters when inserting this driver.

**Table 11-6. Parameters of TW2868 Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configuration.
iaddr	{0x50, 0x52, 0x54, 0x56}	Device I <sup>2</sup> C address
vmode	{2, 2, 2, 2}	Video output mode selection 0: NTSC 720H 1CH 1: NTSC 720H 2CH 2: NTSC 720H 4CH 3: PAL 720H 1CH 4: PAL 720H 2CH 5: PAL 720H 4CH



Name	Default Value	Description
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1  -1: It means that do not change the current frequency settings.
clk_src	0	Clock source of the external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as the device RSTB pin 0: Not used 1: X_CAP_RST as RSTB pin for the device hardware reset
sample_rate	0	Audio sample rate 0: 8K 1: 16K 2: 32K 3: 44.1K 5: 48K
sample_size	0	Audio sample size 0: 16bits 1: 8bits
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable  Notify the video status and video format to capture the input_module tw2868 driver for the video interface to switch configuration.

Name	Default Value	Description
ch_map	0	Device video channel mapping table 0: GM8210 socket board, vport 4CH mode GM8287 socket board 1: GM8210 system board, vport 4CH mode 2: GM8287 system board, vport 4CH mode GM8283 socket board 3: Generic, vport 1CH mode 4: Generic, vport 2CH mode Users must select the mapping table based on the hardware circuit.
audio_chnum	8	Device audio channel number

### 11.6.1 TW2868 Proc Nodes

The tw2868 driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation/sharpness through these nodes. A sample of these nodes is listed below.

/# cat /proc/tw2868.0/				
brightness	hue	saturation_u	sharpness	vmode
contrast	output_ch	saturation_v	status	volume
/# cat /proc/tw2868.1/				
brightness	hue	saturation_u	sharpness	vmode
contrast	output_ch	saturation_v	status	volume
/# cat /proc/tw2868.2/				
brightness	hue	saturation_u	sharpness	vmode
contrast	output_ch	saturation_v	status	volume
/# cat /proc/tw2868.3/				
brightness	hue	saturation_u	sharpness	vmode
contrast	output_ch	saturation_v	status	volume

**Figure 11-6. Proc Nodes of TW2868 Module**

The following table shows the proc nodes of the tw2868 module.

/proc/tw2868.0	brightness
/proc/tw2868.1	contrast
/proc/tw2868.2	hue
/proc/tw2868.3	saturation_u
	saturation_v
	sharpness
	output_ch
	volume
	status
	vmode

11.6.1.1 brightness

Users can use the /proc/tw2868.x/brightness node to display and setup brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2868.x/brightness**

```
/# cat /proc/tw2868.0/brightness

[TW2868#0]
-----
VIN#      VCH#      BRIGHTNESS
-----
0         0         0x00
1         1         0x00
2         2         0x00
3         3         0x00
4         4         0x00
5         5         0x00
6         6         0x00
7         7         0x00

Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1
```

- Set the value to switch brightness  
**echo [vin] [value] > /proc/tw2868.x/brightness**  
 vin: 0 ~ 7, other for all channels  
 value: 0x00 ~ 0xff

### 11.6.1.2 contrast

Users can use the /proc/tw2868.x/contrast node to display/setup the contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/tw2868.x/contrast**

```

/# cat /proc/tw2868.0/contrast

[TW2868#0]
-----
VIN#      VCH#      CONTRAST
-----
0         0         0x64
1         1         0x64
2         2         0x64
3         3         0x64
4         4         0x64
5         5         0x64
6         6         0x64
7         7         0x64

Contrast[0% ~ 255%] ==> 0x00=0%, 0xff=255%

```

- Set the value to switch contrast  
**echo [vin] [value] > /proc/tw2868.x/contrast**  
 vin: 0 ~ 7, other for all channels  
 Value: 0x00 ~ 0xff

### 11.6.1.3 hue

Users can use the `/proc/tw2868.x/hue` node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2868.x/hue**

```
/# cat /proc/tw2868.0/hue

[TW2868#0]
-----
VIN#      VCH#      HUE
-----
0          0          0x00
1          1          0x00
2          2          0x00
3          3          0x00
4          4          0x00
5          5          0x00
6          6          0x00
7          7          0x00

Hue[0x00 ~ 0xff] ==> 0x00=0, 0x7f=90, 0x80=-90
```

- Set the value to switch hue

**echo [vin] [value] > /proc/tw2868.x/hue**

vin: 0 ~ 7, other for all channels

Value: 0x00 ~ 0xff

### 11.6.1.4 saturation\_u

Users can use the `/proc/tw2868.x/saturation_u` node to display and setup saturation\_u of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2868.x/saturation\_u**

```
[TW2868#0]
-----
VIN#      VCH#      SATURATION_U
-----
0         0         0x80
1         1         0x80
2         2         0x80
3         3         0x80
4         4         0x80
5         5         0x80
6         6         0x80
7         7         0x80

Saturation_U[0% ~ 200%] ==> 0x80=100%
```

- Set the value to switch saturation\_u

**echo [vin] [value] > /proc/tw2868.x/saturation\_u**

vin: 0 ~ 7, other for all channels

Value: 0x00 ~ 0xff

### 11.6.1.5 saturation\_v

Users can use the /proc/tw2868.x/saturation\_v node to display and setup saturation\_v of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2868.x/saturation\_v**

```

/# cat /proc/tw2868.0/saturation_v

[TW2868#0]
-----
VIN#      VCH#      SATURATION_V
-----
0         0         0x80
1         1         0x80
2         2         0x80
3         3         0x80
4         4         0x80
5         5         0x80
6         6         0x80
7         7         0x80

Saturation_V[0% ~ 200%] ==> 0x80=100%

```

- Set the value to switch saturation\_v  
**echo [vin] [value] > /proc/tw2868.x/saturation\_v**  
 vin: 0 ~ 7, other for all channels  
 Value: 0x00 ~ 0xff

### 11.6.1.6 status

Users can use the /proc/tw2868.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/tw2868.x/status**

```
/# cat /proc/tw2868.0/status
```

```
[TW2868#0]
```

```
-----  
VIN#      VCH#      NOVID  
-----  
0         0         Video_Loss  
1         1         Video_Loss  
2         2         Video_Loss  
3         3         Video_Loss  
4         4         Video_Loss  
5         5         Video_Loss  
6         6         Video_Loss  
7         7         Video_Loss
```

### 11.6.1.7 vmode

Users can use the /proc/tw2868.x/vmode node to display and setup the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2868.x/vmode**

```
/# cat /proc/tw2868.0/vmode
```

```
[TW2868#0]
```

```
00: NTSC_720H_1CH
```

```
01: NTSC_720H_2CH
```

```
02: NTSC_720H_4CH
```

```
03: PAL_720H_1CH
```

```
04: PAL_720H_2CH
```

```
05: PAL_720H_4CH
```

```
-----  
Current==> NTSC_720H_4CH
```



- Set the value to switch device vport output mode

**echo [mode] > /proc/tw2868.x/vmode**

mode: 0 ~ 5

### 11.6.1.8 sharpness

Users can use the /proc/tw2868.x/sharpness node to display and setup sharpness of each channel

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2868.x/sharpness**

```

/# cat /proc/tw2868.0/sharpness

[TW2868#0]
-----
VIN#      VCH#      SHARPNESS
-----
0         0         0x1
1         1         0x1
2         2         0x1
3         3         0x1
4         4         0x1
5         5         0x1
6         6         0x1
7         7         0x1

Sharpness[0x0 ~ 0xf] - (16 levels) ==> 0x0:no effect, 0x1~0xf:sharpness enhancement
('0xf' being the strongest)

```

- Set the value to switch sharpness

**echo [vin] [value] > /proc/tw2868.x/sharpness**

vin: 0 ~ 7, other for all channels

Value: 0x0 ~ 0xf

### 11.6.1.9 volume

Users can use the `/proc/tw2868.x/volume` node to display and setup the audio volume.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2868.x/volume**

```
/# cat /proc/tw2868.0/volume  
  
[tw2868#0]  
Volume[0x0~0xf] = 8
```

- Set the value to switch audio volume

**echo [value] > /proc/tw2868.x/volume**

Value: 0x0 ~ 0xf

### 11.6.1.10 output\_ch

Users can use the `/proc/tw2868.x/output_ch` node to display and setup the audio playback output channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2868.x/output\_ch**

```
/# cat /proc/tw2868.0/output_ch  
  
[tw2868#0]  
Current[0x0~0x18]==> PLAYBACK first stage
```

- Set the value to switch the audio playback output channel

**echo [value] > /proc/tw2868.x/output\_ch**

Value: 0x0 ~ 0x18

## 11.7 TW2964 Driver Module

Table 11-7 lists and describes the parameters when inserting this module.

**Table 11-7. Parameters of TW2964 Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configuration.
iaddr	{0x50, 0x52, 0x54, 0x56}	Device I <sup>2</sup> C address
vmode	{2, 2, 2, 2}	Video output mode selection 0: NTSC 720H 1CH 1: NTSC 720H 2CH 2: NTSC 720H 4CH 3: NTSC 960H 1CH 4: NTSC 960H 2CH 5: NTSC 960H 4CH 6: PAL 720H 1CH 7: PAL 720H 2CH 8: PAL 720H 4CH 9: PAL 960H 1CH 10: PAL 960H 2CH 11: PAL 960H 4CH
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1  -1: It means that do not change the current frequency settings.

Name	Default Value	Description
clk_src	4	Clock source of the external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as the device RSTB pin 0: Not used 1: X_CAP_RST as the RSTB pin for the device hardware reset
sample_rate	0	Audio sample rate 0: 8K 1: 16K 2: 32K 3: 44.1K 5: 48K
sample_size	0	Audio sample size 0: 16bits 1: 8bits
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module tw2964 driver for the video interface to switch configuration.

Name	Default Value	Description
ch_map	0	Device video channel mapping table 0: GM8210 socket board, vport 4CH mode GM8287 socket board 1: Generic, vport 1CH mode 2: Generic, vport 2CH mode 3: Generic, vport 4CH mode Users must select the mapping table based on the hardware circuit.
audio_chnum	8	Device audio channel number

### 11.7.1 TW2964 Proc Nodes

The tw2964 driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation/sharpness through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/tw2964.0/
brightness    hue          saturation_u  sharpness    vmode
contrast     output_ch    saturation_v  status       volume

/# cat /proc/tw2964.1/
brightness    hue          saturation_u  sharpness    vmode
contrast     output_ch    saturation_v  status       volume

/# cat /proc/tw2964.2/
brightness    hue          saturation_u  sharpness    vmode
contrast     output_ch    saturation_v  status       volume

/# cat /proc/tw2964.3/
brightness    hue          saturation_u  sharpness    vmode
contrast     output_ch    saturation_v  status       volume

```

**Figure 11-7. Proc Nodes of TW2964 Module**

The following table shows the proc nodes of the tw2964 module.

/proc/tw2964.0	brightness
/proc/tw2964.1	contrast
/proc/tw2964.2	hue
/proc/tw2964.3	saturation_u
	saturation_v
	sharpness
	output_ch
	volume
	status
	vmode

### 11.7.1.1 brightness

Users can use the /proc/tw2964.x/brightness node to display and setup the brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2964.x/brightness**

```
/# cat /proc/tw2964.0/brightness

[TW2964#0]
-----
VIN#      VCH#      BRIGHTNESS
-----
0         0         0x00
1         1         0x00
2         2         0x00
3         3         0x00

Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1
```

- Set the value to switch brightness

**echo [vin] [value] > /proc/tw2964.x/brightness**

vin: 0 ~ 7, other for all channels

Value: 0x00 ~ 0xff

### 11.7.1.2 contrast

Users can use the /proc/tw2964.x/contrast node to display/setup the contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2964.x/contrast**

```
/# cat /proc/tw2964.0/contrast

[TW2964#0]
-----
VIN#      VCH#      CONTRAST
-----
0          0          0x64
1          1          0x64
2          2          0x64
3          3          0x64

Contrast[0% ~ 255%] ==> 0x00=0%, 0xff=255%
```

- Set the value to switch contrast

**echo [vin] [value] > /proc/tw2964.x/contrast**

vin: 0 ~ 7, other for all channels

Value: 0x00 ~ 0xff

### 11.7.1.3 hue

Users can use the /proc/tw2964.x/hue node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

### **cat /proc/tw2964.x/hue**

```
/# cat /proc/tw2964.0/hue

[TW2964#0]
-----
VIN#      VCH#      HUE
-----
0         0         0x00
1         1         0x00
2         2         0x00
3         3         0x00

Hue[0x00 ~ 0xff] ==> 0x00=0, 0x7f=90, 0x80=-90
```

- Set the value to switch hue  
**echo [vin] [value] > /proc/tw2964.x/hue**  
vin: 0 ~ 7, other for all channels  
Value: 0x00 ~ 0xff

#### **11.7.1.4 saturation\_u**

Users can use the /proc/tw2964.x/saturation\_u node to display and setup saturation\_u of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/tw2964.x/saturation\_u**



```
[TW2964#0]
```

```
-----  
VIN#      VCH#      SATURATION_U  
-----
```

```
0          0          0x80
```

```
1          1          0x80
```

```
2          2          0x80
```

```
3          3          0x80
```

```
Saturation_U[0% ~ 200%] ==> 0x80=100%
```

- Set the value to switch saturation\_u  
**echo [vin] [value] > /proc/tw2964.x/saturation\_u**  
vin: 0 ~ 7, other for all channels  
Value: 0x00 ~ 0xff

#### 11.7.1.5 saturation\_v

Users can use the /proc/tw2964.x/saturation\_v node to display and setup saturation\_v of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/tw2964.x/saturation\_v**

```

/# cat /proc/tw2964.0/saturation_v

[TW2964#0]
-----
VIN#      VCH#      SATURATION_V
-----
0         0         0x80
1         1         0x80
2         2         0x80
3         3         0x80

Saturation_V[0% ~ 200%] ==> 0x80=100%

```

- Set the value to switch saturation\_v  
**echo [vin] [value] > /proc/tw2964.x/saturation\_v**  
vin: 0 ~ 7, other for all channels  
Value: 0x00 ~ 0xff

#### 11.7.1.6 status

Users can use the /proc/tw2964.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/tw2964.x/status**

```
/# cat /proc/tw2964.0/status
```

```
[TW2964#0]
```

```
-----
```

```
VIN#      VCH#      NOVID
```

```
-----
```

```
0          0          Video_Loss
```

```
1          1          Video_Loss
```

```
2          2          Video_Loss
```

```
3          3          Video_Loss
```

### 11.7.1.7 vmode

Users can use the `/proc/tw2964.x/vmode` node to display and setup the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2964.x/vmode**

```
/# cat /proc/tw2964.0/vmode
```

```
[TW2964#0]
```

```
00: NTSC_720H_1CH
```

```
01: NTSC_720H_2CH
```

```
02: NTSC_720H_4CH
```

```
03: NTSC_960H_1CH
```

```
04: NTSC_960H_2CH
```

```
05: NTSC_960H_4CH
```

```
06: PAL_720H_1CH
```

```
07: PAL_720H_2CH
```

```
08: PAL_720H_4CH
```

```
09: PAL_960H_1CH
```

```
10: PAL_960H_2CH
```

```
11: PAL_960H_4CH
```

```
-----  
Current==> NTSC_720H_4CH
```

- Set the value to switch the device vport output mode

**echo [mode] > /proc/tw2964.x/vmode**

mode: 0 ~ 11

### 11.7.1.8 sharpness

Users can use the /proc/tw2964.x/sharpness node to display and setup sharpness of each channel

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2964.x/sharpness**

```
/# cat /proc/tw2964.0/sharpness
```

```
[TW2964#0]
```

```
-----  
VIN#      VCH#      SHARPNESS  
-----
```

```
0          0          0x1
```

```
1          1          0x1
```

```
2          2          0x1
```

```
3          3          0x1
```

```
Sharpness[0x0 ~ 0xf] - (16 levels) ==> 0x0:no effect, 0x1~0xf:sharpness enhancement  
( '0xf' being the strongest)
```

- Set the value to switch sharpness

**echo [vin] [value] > /proc/tw2964.x/sharpness**

vin: 0 ~ 7, others are for all channels.

Value: 0x0 ~ 0xf

### 11.7.1.9 volume

Users can use the /proc/tw2964.x/volume node to display and setup the audio volume.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2964.x/volume**

```
/# cat /proc/tw2964.0/volume
```

```
[tw2964#0]
```

```
Volume[0x0~0xf] = 8
```

- Set the value to switch the audio volume

**echo [value] > /proc/tw2964.x/volume**

Value: 0x0 ~ 0xf

### 11.7.1.10 output\_ch

Users can use the /proc/tw2964x/output\_ch node to display and setup the audio playback output channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tw2964.x/output\_ch**

```
/# cat /proc/tw2964.0/output_ch

[tw2964#0]
Current[0x0~0x18]==> PLAYBACK first stage
```

- Set the value to switch audio playback output channel

**echo [value] > /proc/tw2964.x/output\_ch**

Value: 0x0 ~ 0x18

## 11.8 NVP1918C Driver Module

Table 11-8 lists and describes the parameters when inserting this module.

**Table 11-8. Parameters of NVP1918C Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configuration
iaddr	{0x66, 0x64, 0x62, 0x60}	Device I <sup>2</sup> C address

Name	Default Value	Description
vmode	{2, 2, 2, 2}	Video output mode selection 0: NTSC 720H 1CH 1: NTSC 720H 2CH 2: NTSC 720H 4CH 3: NTSC 960H 1CH 4: NTSC 960H 2CH 5: NTSC 960H 4CH 6: PAL 720H 1CH 7: PAL 720H 2CH 8: PAL 720H 4CH 9: PAL 960H 1CH 10: PAL 960H 2CH 11: PAL 960H 4CH
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1  -1: It means that do not change the current frequency settings.
clk_src	4	Clock source of the external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as the device RSTB pin 0: Not used 1: GPIO#33 as the RSTB pin for the device hardware reset
sample_rate	0	Audio sample rate 0: 8K 1: 16K

Name	Default Value	Description
sample_size	0	Audio sample size 0: 16bits 1: 8bits
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module nvp1918c driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: GM8210 socket board, vport 4CH mode GM8287 socket board 1: GM8210 system board, vport 4CH mode 2: GM8287 system board, vport 4CH mode GM8283 socket board Users must select the mapping table based on the hardware circuit.
audio_chnum	8	Device audio channel number



11.8.1 NVP1918C Proc Nodes

The nvp1918c driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation/sharpness through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/nvp1918c.0/
brightness  hue          saturation  status      volume
contrast    output_ch    sharpness  vmode

/# cat /proc/nvp1918c.1/
brightness  hue          saturation  status      volume
contrast    output_ch    sharpness  vmode

/# cat /proc/nvp1918c.2/
brightness  hue          saturation  status      volume
contrast    output_ch    sharpness  vmode

/# cat /proc/nvp1918c.3/
brightness  hue          saturation  status      volume
contrast    output_ch    sharpness  vmode

```

Figure 11-8. Proc Nodes of NVP1918C Module

The following table shows the proc nodes of the nvp1918c module.

/proc/nvp1918c.0	brightness
/proc/nvp1918c.1	contrast
/proc/nvp1918c.2	hue
/proc/nvp1918c.3	saturation
	sharpness
	volume
	output_ch
	status
	vmode

### 11.8.1.1 brightness

Users can use the `/proc/nvp1918c.x/brightness` node to display and setup the brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1918c.x/brightness**

```
/# cat /proc/nvp1918c.0/brightness

[NVP1918C#0]
-----
VIN#      VCH#      BRIGHTNESS
-----
0          8          0xf8
1          9          0xf8
2         10          0xf8
3         11          0xf8
4         12          0xf8
5         13          0xf8
6         14          0xf8
7         15          0xf8

Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1
```

- Set the value to switch brightness

**echo [vin] [value] > /proc/nvp1918c.x/brightness**

vin: 0 ~ 7, others are for all channels.

Value: 0x00 ~ 0xff

### 11.8.1.2 contrast

Users can use the `/proc/nvp1918c.x/contrast` node to display and setup the contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

### **cat /proc/nvp1918c.x/contrast**

```
/# cat /proc/nvp1918c.0/contrast

[NVP1918C#0]
-----
VIN#      VCH#      CONTRAST
-----
0         8         0x76
1         9         0x76
2         10        0x76
3         11        0x76
4         12        0x76
5         13        0x76
6         14        0x76
7         15        0x76

Contrast[0x00 ~ 0xff] ==> 0x00=x0, 0x40=x0.5, 0x80=x1, 0xff=x2
```

- Set the value to switch contrast  
**echo [vin] [value] > /proc/nvp1918c.x/contrast**  
vin: 0 ~ 7, others are for all channels.  
Value: 0x00 ~ 0xff

#### **11.8.1.3 hue**

Users can use the /proc/nvp1918c.x/hue node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/nvp1918c.x/hue**

```
/# cat /proc/nvp1918c.0/hue
```

```
[NVP1918C#0]
```

```
-----  
VIN#      VCH#      HUE  
-----
```

0	8	0x01
1	9	0x01
2	10	0x01
3	11	0x01
4	12	0x01
5	13	0x01
6	14	0x01
7	15	0x01

```
Hue[0x00 ~ 0xff] ==> 0x00=0, 0x40=90, 0x80=180, 0xff=360
```

- Set the value to switch hue

```
echo [vin] [value] > /proc/nvp1918c.x/hue
```

vin: 0 ~ 7, others are for all channels.

Value: 0x00 ~ 0xff

#### 11.8.1.4 saturation

Users can use the /proc/nvp1918c.x/saturation node to display and setup the saturation of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

```
cat /proc/nvp1918c.x/saturation
```

```

/# cat /proc/nvp1918c.0/saturation

[NVP1918C#0]
-----
VIN#      VCH#      SATURATION
-----
0         8         0x80
1         9         0x80
2         10        0x80
3         11        0x80
4         12        0x80
5         13        0x80
6         14        0x80
7         15        0x80

Saturation[0x00 ~ 0xff] ==> 0x00=x0, 0x80=x1, 0xc0=x1.5, 0xff=x2

```

- Set the value to switch saturation  
**echo [vin] [value] > /proc/nvp1918c.x/saturation**  
vin: 0 ~ 7, others are for all channels.  
Value: 0x00 ~ 0xff

### 11.8.1.5 status

Users can use the /proc/nvp1918c.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/nvp1918c.x/status**

```
/# cat /proc/nvp1918c.0/status
```

```
[NVP1918C#0]
```

```
-----
```

VIN#	VCH#	NOVID
------	------	-------

```
-----
```

0	8	Video_Loss
---	---	------------

1	9	Video_Loss
---	---	------------

2	10	Video_Loss
---	----	------------

3	11	Video_Loss
---	----	------------

4	12	Video_Loss
---	----	------------

5	13	Video_Loss
---	----	------------

6	14	Video_Loss
---	----	------------

7	15	Video_Loss
---	----	------------

### 11.8.1.6 vmode

Users can use the /proc/nvp1918c.x/vmode node to display the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1918c.x/vmode**

```
/# cat /proc/nvp1918c.0/vmode
```

```
[NVP1918C#0]
```

```
02: NTSC_720H_4CH
```

```
05: NTSC_960H_4CH
```

```
08: PAL_720H_4CH
```

```
11: PAL_960H_4CH
```

```
-----
```

```
Current==> NTSC_720H_4CH
```

- Set the value to switch the device video port output mode

**echo [mode] > /proc/nvp1918c.x/vmode**

mode: 2, 5, 8, 11

### 11.8.1.7 sharpness

Users can use the /proc/nvp1918c.x/sharpness node to display and setup the sharpness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1918c.x/sharpness**

```
/# cat /proc/nvp1918c.0/sharpness

[NVP1918C#0]
-----
VIN#      VCH#      SHARPNESS
-----
0         8         0x80
1         9         0x80
2         10        0x80
3         11        0x80
4         12        0x80
5         13        0x80
6         14        0x80
7         15        0x80

H_Sharpness[0x0 ~ 0xf] - Bit[7:4] ==> 0x0:x0, 0x4:x0.5, 0x8:x1, 0xf:x2
V_Sharpness[0x0 ~ 0xf] - Bit[3:0] ==> 0x0:x1, 0x4:x2, 0x8:x3, 0xf:x4
```

- Set the value to switch sharpness

**echo [vin] [value] > /proc/nvp1918c.x/sharpness**

vin: 0 ~ 7, others are for all channels.

Value: 0x00~ 0xff

### 11.8.1.8 volume

Users can use the `/proc/nvp1918c.x/volume` node to display and setup the audio volume.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1918c.x/volume**

```
/# cat /proc/nvp1918c.0/volume  
  
[NVP1918C#0]  
Volume[0x0~0xf] = 8
```

- Set the value to switch the audio volume

**echo [value] > /proc/nvp1918c.x/volume**

Value: 0x0 ~ 0xf

### 11.8.1.9 output\_ch

Users can use the `/proc/nvp1918c.x/output_ch` node to display and setup the audio playback output channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1918c.x/output\_ch**

```
/# cat /proc/nvp1918c.0/output_ch  
  
[NVP1918C#0]  
Current[0x0~0x18]==> FIRST PLAYBACK AUDIO
```

- Set the value to switch the audio playback output channel

**echo [value] > /proc/nvp1918c.x/output\_ch**

Value: 0x0 ~ 0x18



## 11.9 NVP1914 Driver Module

Table 11-9 lists and describes the parameters when inserting this module.

**Table 11-9. Parameters of NVP1914 Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in kernel configuration
iaddr	{0x66, 0x64, 0x62, 0x60}	Device I <sup>2</sup> C address
vmode	{2, 2, 2, 2}	Video output mode selection 0: NTSC 720H 1CH 1: NTSC 720H 2CH 2: NTSC 720H 4CH 3: NTSC 960H 1CH 4: NTSC 960H 2CH 5: NTSC 960H 4CH 6: PAL 720H 1CH 7: PAL 720H 2CH 8: PAL 720H 4CH 9: PAL 960H 1CH 10: PAL 960H 2CH 11: PAL 960H 4CH
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 -1: It means that do not change the current frequency settings.

Name	Default Value	Description
clk_src	4	Clock source of external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: GPIO#33 as RSTB pin for the device hardware reset
sample_rate	0	Audio sample rate 0: 8K 1: 16K
sample_size	1	Audio sample size 0: 16bits 1: 8bits
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module nvp1914 driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: GM8210 socket board, vport 4CH mode GM8287 socket board 1: GM8210 system board, vport 4CH mode 2: GM8287 system board, vport 4CH mode GM8283 socket board <b>Users must select the mapping table based on the hardware circuit.</b>
audio_chnum	4	Device audio channel number

11.9.1 NVP1914 Proc Nodes

The nvp1914 driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation/sharpness through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/nvp1914.0/
brightness hue saturation status volume
contrast output_ch sharpness vmode

/# cat /proc/nvp1914.1/
brightness hue saturation status volume
contrast output_ch sharpness vmode

/# cat /proc/nvp1914.2/
brightness hue saturation status volume
contrast output_ch sharpness vmode

/# cat /proc/nvp1914.3/
brightness hue saturation status volume
contrast output_ch sharpness vmode

```

Figure 11-9. Proc Nodes of NVP1914 Module

The following table shows the proc nodes of the nvp1914 module.

/proc/nvp1914.0	brightness
/proc/nvp1914.1	contrast
/proc/nvp1914.2	hue
/proc/nvp1914.3	saturation
	sharpness
	volume
	output_ch
	status
	vmode

### 11.9.1.1 brightness

Users can use the `/proc/nvp1914.x/brightness` node to display and setup the brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914.x/brightness**

```
/# cat /proc/nvp1914.0/brightness

[NVP1914#0]
-----
VIN#      VCH#      BRIGHTNESS
-----
0         8         0xf8
1         9         0xf8
2         10        0xf8
3         11        0xf8

Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1
```

- Set the value to switch brightness

**echo [vin] [value] > /proc/nvp1914.x/brightness**

vin: 0 ~ 7, others are for all channels.

Value: 0x00 ~ 0xff

### 11.9.1.2 contrast

Users can use the `/proc/nvp1914.x/contrast` node to display and setup the contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914.x/contrast**

```
/# cat /proc/nvp1914.0/contrast
```

```
[NVP1914#0]
```

```
-----  
VIN#      VCH#      CONTRAST  
-----
```

```
0          8          0x76
```

```
1          9          0x76
```

```
2          10         0x76
```

```
3          11         0x76
```

```
Contrast[0x00 ~ 0xff] ==> 0x00=x0, 0x40=x0.5, 0x80=x1, 0xff=x2
```

- Set the value to switch contrast

**echo [vin] [value] > /proc/nvp1914.x/contrast**

vin: 0 ~ 7, others are for all channels.

Value: 0x00 ~ 0xff

### 11.9.1.3 hue

Users can use the /proc/nvp1914.x/hue node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914.x/hue**

```
/# cat /proc/nvp1914.0/hue
```

```
[NVP1914#0]
```

```
-----  
VIN#      VCH#      HUE  
-----
```

```
0          8          0x01
```

```
1          9          0x01
```

```
2          10         0x01
```

```
3          11         0x01
```

```
Hue[0x00 ~ 0xff] ==> 0x00=0, 0x40=90, 0x80=180, 0xff=360
```

- Set the value to switch hue

**echo [vin] [value] > /proc/nvp1914.x/hue**

vin: 0 ~ 7, others are for all channels.

Value: 0x00 ~ 0xff

### 11.9.1.4 saturation

Users can use the /proc/nvp1914.x/saturation node to display and setup saturation of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914.x/saturation**

```

/# cat /proc/nvp1914.0/saturation

[NVP1914#0]
-----
VIN#      VCH#      SATURATION
-----
0         8         0x80
1         9         0x80
2         10        0x80
3         11        0x80

Saturation[0x00 ~ 0xff] ==> 0x00=x0, 0x80=x1, 0xc0=x1.5, 0xff=x2

```

- Set the value to switch saturation  
**echo [vin] [value] > /proc/nvp1914.x/saturation**  
vin: 0 ~ 7, others are for all channels.  
Value: 0x00 ~ 0xff

### 11.9.1.5 status

Users can use the /proc/nvp1914.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/nvp1914.x/status**

```
/# cat /proc/nvp1914.0/status
```

```
[NVP1914#0]
```

```
-----  
VIN#      VCH#      NOVID  
-----  
0         8         Video_Loss  
1         9         Video_Loss  
2         10        Video_Loss  
3         11        Video_Loss
```

### 11.9.1.6 vmode

Users can use the `/proc/nvp1914.x/vmode` node to display the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914.x/vmode**

```
/# cat /proc/nvp1914.0/vmode
```

```
[NVP1914#0]
```

```
02: NTSC_720H_4CH
```

```
05: NTSC_960H_4CH
```

```
08: PAL_720H_4CH
```

```
11: PAL_960H_4CH
```

```
-----  
Current==> NTSC_720H_4CH
```

- Set the value to switch the device video port output mode

**echo [mode] > /proc/nvp1914.x/vmode**

mode: 2, 5, 8, 11



### 11.9.1.7 sharpness

Users can use the `/proc/nvp1914.x/sharpness` node to display and setup sharpness of each channel

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914.x/sharpness**

```
/# cat /proc/nvp1914.0/sharpness

[NVP1914#0]
-----
VIN#      VCH#      SHARPNESS
-----
0         8         0x80
1         9         0x80
2         10        0x80
3         11        0x80

H_Sharpness[0x0 ~ 0xf] - Bit[7:4] ==> 0x0:x0, 0x4:x0.5, 0x8:x1, 0xf:x2
V_Sharpness[0x0 ~ 0xf] - Bit[3:0] ==> 0x0:x1, 0x4:x2, 0x8:x3, 0xf:x4
```

- Set the value to switch sharpness

**echo [vin] [value] > /proc/nvp1914.x/sharpness**

vin: 0 ~ 7, others are for all channels.

Value: 0x00~ 0xff

### 11.9.1.8 volume

Users can use the `/proc/nvp1914.x/volume` node to display and setup the audio volume.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914.x/volume**

```
/# cat /proc/nvp1914.0/volume
```

```
[NVP1914#0]
```

```
Volume[0x0~0xf] = 8
```

- Set the value to switch audio volume  
**echo [value] > /proc/nvp1914.x/volume**  
Value: 0x0 ~ 0xf

#### 11.9.1.9 output\_ch

Users can use the /proc/nvp1914.x/output\_ch node to display and setup the audio playback output channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914.x/output\_ch**

```
/# cat /proc/nvp1914.0/output_ch
```

```
[NVP1914#0]
```

```
Current[0x0~0x18]==> FIRST PLAYBACK AUDIO
```

- Set the value to switch the audio playback output channel  
**echo [value] > /proc/nvp1914.x/output\_ch**  
Value: 0x0 ~ 0x18

## 11.10 NVP1914C Driver Module

Table 11-10 lists and describes the parameters when inserting this module.

**Table 11-10. Parameters of NVP1914C Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must to enable I <sup>2</sup> C bus support in kernel configuration
iaddr	{0x66, 0x64, 0x62, 0x60}	Device I <sup>2</sup> C address
vmode	{2, 2, 2, 2}	Video output mode selection 0: NTSC 720H 1CH 1: NTSC 720H 2CH 2: NTSC 720H 4CH 3: NTSC 960H 1CH 4: NTSC 960H 2CH 5: NTSC 960H 4CH 6: PAL 720H 1CH 7: PAL 720H 2CH 8: PAL 720H 4CH 9: PAL 960H 1CH 10: PAL 960H 2CH 11: PAL 960H 4CH
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 -1: It means that do not change the current frequency settings.
clk_src	4	Clock source of the external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3

Name	Default Value	Description
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: GPIO#33 as RSTB pin for the device hardware reset
sample_rate	0	Audio sample rate 0: 8K 1: 16K
sample_size	1	Audio sample size 0: 16bits 1: 8bits
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module nvp1914c driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: GM8210 socket board, vport 4CH mode GM8287 socket board 1: GM8210 system board, vport 4CH mode 2: GM8287 system board, vport 4CH mode GM8283 socket board <b>Users must select the mapping table based on the hardware circuit.</b>
audio_chnum	4	Device audio channel number

### 11.10.1 NVP1914C Proc Nodes

The nvp1914c driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation/sharpness through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/nvp1914c.0/
brightness  hue          saturation  status      volume
contrast    output_ch    sharpness  vmode

/# cat /proc/nvp1914c.1/
brightness  hue          saturation  status      volume
contrast    output_ch    sharpness  vmode

/# cat /proc/nvp1914c.2/
brightness  hue          saturation  status      volume
contrast    output_ch    sharpness  vmode

/# cat /proc/nvp1914c.3/
brightness  hue          saturation  status      volume
contrast    output_ch    sharpness  vmode

```

Figure 11-10. Proc Nodes of NVP1914C Module

The following table shows the proc nodes of the nvp1914c module.

/proc/nvp1914c.0	brightness
/proc/nvp1914c.1	contrast
/proc/nvp1914c.2	hue
/proc/nvp1914c.3	saturation
	sharpness
	volume
	output_ch
	status
	vmode

### 11.10.1.1 brightness

Users can use the `/proc/nvp1914c.x/brightness` node to display and setup brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914c.x/brightness**

```
/# cat /proc/nvp1914c.0/brightness

[NVP1914C#0]
-----
VIN#      VCH#      BRIGHTNESS
-----
0         8         0xf8
1         9         0xf8
2         10        0xf8
3         11        0xf8

Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1
```

- Set the value to switch brightness

**echo [vin] [value] > /proc/nvp1914c.x/brightness**

vin: 0 ~ 7, others are for all channels.

Value: 0x00 ~ 0xff

### 11.10.1.2 contrast

Users can use the `/proc/nvp1914c.x/contrast` node to display and setup the contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914c.x/contrast**

```

/# cat /proc/nvp1914c.0/contrast

[NVP1914C#0]
-----
VIN#      VCH#      CONTRAST
-----
0         8         0x76
1         9         0x76
2         10        0x76
3         11        0x76

Contrast[0x00 ~ 0xff] ==> 0x00=x0, 0x40=x0.5, 0x80=x1, 0xff=x2

```

- Set the value to switch contrast  
**echo [vin] [value] > /proc/nvp1914c.x/contrast**  
vin: 0 ~ 7, others are for all channels.  
Value: 0x00 ~ 0xff

### 11.10.1.3 hue

Users can use the /proc/nvp1914c.x/hue node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/nvp1914c.x/hue**

```
/# cat /proc/nvp1914c.0/hue
```

```
[NVP1914C#0]
```

```
-----  
VIN#      VCH#      HUE  
-----
```

```
0          8          0x01
```

```
1          9          0x01
```

```
2         10          0x01
```

```
3         11          0x01
```

```
Hue[0x00 ~ 0xff] ==> 0x00=0, 0x40=90, 0x80=180, 0xff=360
```

- Set the value to switch hue

**echo [vin] [value] > /proc/nvp1914c.x/hue**

vin: 0 ~ 7, others are for all channels.

Value: 0x00 ~ 0xff

#### 11.10.1.4 saturation

Users can use the /proc/nvp1914c.x/saturation node to display and setup the saturation of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914c.x/saturation**



```

/# cat /proc/nvp1914c.0/saturation

[NVP1914C#0]
-----
VIN#      VCH#      SATURATION
-----
0         8         0x80
1         9         0x80
2         10        0x80
3         11        0x80

Saturation[0x00 ~ 0xff] ==> 0x00=x0, 0x80=x1, 0xc0=x1.5, 0xff=x2

```

- Set the value to switch saturation  
**echo [vin] [value] > /proc/nvp1914c.x/saturation**  
vin: 0 ~ 7, others are for all channels.  
Value: 0x00 ~ 0xff

### 11.10.1.5 status

Users can use the /proc/nvp1914c.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/nvp1914c.x/status**

```
/# cat /proc/nvp1914c.0/status
```

```
[NVP1914C#0]
```

```
-----  
VIN#      VCH#      NOVID  
-----  
0         8         Video_Loss  
1         9         Video_Loss  
2         10        Video_Loss  
3         11        Video_Loss
```

### 11.10.1.6 vmode

Users can use the /proc/nvp1914c.x/vmode node to display the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914c.x/vmode**

```
/# cat /proc/nvp1914c.0/vmode
```

```
[NVP1914C#0]
```

```
02: NTSC_720H_4CH
```

```
05: NTSC_960H_4CH
```

```
08: PAL_720H_4CH
```

```
11: PAL_960H_4CH
```

```
-----  
Current==> NTSC_720H_4CH
```

- Set the value to switch the device video port output mode

**echo [mode] > /proc/nvp1914c.x/vmode**

mode: 2, 5, 8, 11

### 11.10.1.7 sharpness

Users can use the /proc/nvp1914c.x/sharpness node to display and setup the sharpness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914c.x/sharpness**

```
/# cat /proc/nvp1914c.0/sharpness

[NVP1914C#0]
-----
VIN#      VCH#      SHARPNESS
-----
0         8         0x80
1         9         0x80
2         10        0x80
3         11        0x80

H_Sharpness[0x0 ~ 0xf] - Bit[7:4] ==> 0x0:x0, 0x4:x0.5, 0x8:x1, 0xf:x2
V_Sharpness[0x0 ~ 0xf] - Bit[3:0] ==> 0x0:x1, 0x4:x2, 0x8:x3, 0xf:x4
```

- Set the value to switch sharpness

**echo [vin] [value] > /proc/nvp1914c.x/sharpness**

vin: 0 ~ 7, others are for all channels.

Value: 0x00~ 0xff

### 11.10.1.8 volume

Users can use the /proc/nvp1914c.x/volume node to display and setup the audio volume.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp1914c.x/volume**

```
/# cat /proc/nvp1914c.0/volume  
  
[NVP1914C#0]  
Volume[0x0~0xf] = 8
```

- Set the value to switch the audio volume  
**echo [value] > /proc/nvp1914c.x/volume**  
Value: 0x0 ~ 0xf

#### 11.10.1.9 output\_ch

Users can use the /proc/nvp1914c.x/output\_ch node to display and setup the audio playback output channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/nvp1914c.x/output\_ch**

```
/# cat /proc/nvp1914c.0/output_ch  
  
[NVP1914C#0]  
Current[0x0~0x18]==> FIRST PLAYBACK AUDIO
```

- Set the value to switch the audio playback output channel  
**echo [value] > /proc/nvp1914c.x/output\_ch**  
Value: 0x0 ~ 0x18

## 11.11 RN6318 Driver Module

Table 11-11 lists and describes the parameters when inserting this module.

**Table 11-11. Parameters of RN6318 Module**

Name	Default Value	Description
dev_num	2	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configuration
iaddr	{0x58, 0x5c}	Device I <sup>2</sup> C address
vmode	{2, 2, 2, 2}	Video output mode selection 0: NTSC 720H 1CH 1: NTSC 720H 2CH 2: NTSC 720H 4CH 3: NTSC 960H 1CH 4: NTSC 960H 2CH 5: NTSC 960H 4CH 6: PAL 720H 1CH 7: PAL 720H 2CH 8: PAL 720H 4CH 9: PAL 960H 1CH 10: PAL 960H 2CH 11: PAL 960H 4CH
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 <b>-1: It means that do not change the current frequency settings.</b>
clk_src	4	Clock source of external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3

Name	Default Value	Description
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: GPIO#33 as RSTB pin for the device hardware reset
sample_rate	0	Audio sample rate 0: 8K 1: 16K 2: 32K 3: 44K 4: 48K
sample_size	1	Audio sample size 0: 16bits 1: 8bits
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module rn6318 driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: GM8210 socket board, vport 4CH mode GM8287 socket board Users must select the mapping table based on the hardware circuit.
audio_chnum	16	Device audio channel number

### 11.11.1 RN6318 Proc Nodes

The rn6318 driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation through these nodes. A sample of these nodes is listed below.

```
# cat /proc/rn6318.0/
audio_volume      contrast          sample_rate      status
brightness        hue              sample_size      vmode
bypass_channel    playback_channel saturation

# cat /proc/rn6318.1/
audio_volume      contrast          sample_rate      status
brightness        hue              sample_size      vmode
bypass_channel    playback_channel saturation
```

**Figure 11-11. Proc Nodes of RN6318 Module**

The following table shows the proc nodes of the rn6318 module.

/proc/rn6318.0	brightness
/proc/rn6318.1	contrast
/proc/rn6318.2	hue
/proc/rn6318.3	saturation
	audio_volume
	sample_rate
	sample_size
	playback_channel
	bypass_channel
	status
	vmode

#### 11.11.1.1 brightness

Users can use the /proc/rn6318.x/brightness node to display and setup the brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 1.

### **cat /proc/rn6318.x/brightness**

```
/# cat /proc/rn6318.0/brightness

[RN6318#0]
-----
VIN#      VCH#      BRIGHTNESS
-----
0          0          0x00
1          1          0x00
2          2          0x00
3          3          0x00
4          4          0x00
5          5          0x00
6          6          0x00
7          7          0x00

Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1
```

- Set the value to switch brightness

**echo [vin] [value] > /proc/rn6318.x/brightness**

vin: 0 ~ 7, others are for all channels.

Value: 0x00 ~ 0xff

### **11.11.1.2 contrast**

Users can use the /proc/rn6318.x/contrast node to display and setup the contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 1.

**cat /proc/rn6318.x/contrast**



```
/# cat /proc/rn6318.0/contrast
```

```
[RN6318#0]
```

```
-----  
VIN#      VCH#      CONTRAST  
-----
```

0	0	0x80
1	1	0x80
2	2	0x80
3	3	0x80
4	4	0x80
5	5	0x80
6	6	0x80
7	7	0x80

```
Contrast[0x00 ~ 0xff]
```

- Set the value to switch contrast  
**echo [vin] [value] > /proc/rn6318.x/contrast**  
vin: 0 ~ 7, others are for all channels.  
Value: 0x00 ~ 0xff

### 11.11.1.3 hue

Users can use the /proc/rn6318.x/hue node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 1.  
**cat /proc/rn6318.x/hue**

```
/# cat /proc/rn6318.0/hue
```

```
[RN6318#0]
```

```
-----  
VIN#      VCH#      HUE  
-----
```

0	0	0x80
1	1	0x80
2	2	0x80
3	3	0x80
4	4	0x80
5	5	0x80
6	6	0x80
7	7	0x80

```
Hue[0x00 ~ 0xff]
```

- Set the value to switch hue

**echo [vin] [value] > /proc/rn6318.x/hue**

vin: 0 ~ 7, others for all channels.

Value: 0x00 ~ 0xff

#### 11.11.1.4 saturation

User can the use /proc/rn6318.x/saturation node to display and setup the saturation of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 1.

**cat /proc/rn6318.x/saturation**

```
/# cat /proc/rn6318.0/saturation
```

```
[RN6318#0]
```

```
-----  
VIN#    VCH#    SATURATION  
-----  
0        0        0x80  
1        1        0x80  
2        2        0x80  
3        3        0x80  
4        4        0x80  
5        5        0x80  
6        6        0x80  
7        7        0x80
```

```
Saturation[0x00 ~ 0xff]
```

- Set the value to switch saturation  
**echo [vin] [value] > /proc/rn6318.x/saturation**  
vin: 0 ~ 7, others are for all channels.  
Value: 0x00 ~ 0xff

### 11.11.1.5 status

Users can use the /proc/rn6318.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 1.  
**cat /proc/rn6318.x/status**

```
/# cat /proc/rn6318.0/status
```

```
[RN6318#0]
```

```
-----
```

VIN#	VCH#	NOVID
------	------	-------

```
-----
```

0	0	Video_Loss
---	---	------------

1	1	Video_Loss
---	---	------------

2	2	Video_Loss
---	---	------------

3	3	Video_Loss
---	---	------------

4	4	Video_Loss
---	---	------------

5	5	Video_Loss
---	---	------------

6	6	Video_Loss
---	---	------------

7	7	Video_Loss
---	---	------------

### 11.11.1.6 vmode

Users can use the /proc/rn6318.x/vmode node to display the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/rn6318.x/vmode**

```
/# cat /proc/rn6318.0/vmode
```

```
[RN6318#0]
```

```
02: NTSC_720H_4CH
```

```
05: NTSC_960H_4CH
```

```
08: PAL_720H_4CH
```

```
11: PAL_960H_4CH
```

```
-----
```

```
Current==> NTSC_720H_4CH
```

- Set the value to switch the device video port output mode

**echo [mode] > /proc/rn6318.x/vmode**

mode: 2, 5, 8, 11

### 11.11.1.7 audio\_volume

Users can use the /proc/rn6318.x/audio\_volume node to display and setup audio volume

Usage:

- Get the current value, and x is ranging from 0 to 1.

**cat /proc/rn6318.x/audio\_volume**

```
/# cat /proc/rn6318.0/audio_volume
```

```
[RN6318#0]
```

```
-----  
AIN#      ACH#      VOL  
-----
```

```
0          0          0x8  
1          1          0x8  
2          2          0x8  
3          3          0x8  
4          4          0x8  
5          5          0x8  
6          6          0x8  
7          7          0x8
```

```
Volume[0x0 ~ 0xF]
```

- Set the value to switch the audio volume

**echo [vin] [value] > /proc/rn6318.x/audio\_volume**

vin: 0 ~ 7

Value: 0x0 ~ 0xf

### 11.11.1.8 playback\_channel

Users can use the `/proc/rn6318.x/playback_channel` node to display and setup the audio playback channel.

Usage:

- Get the current value, and x is ranging from 0 to 1.

**cat /proc/rn6318.x/playback\_channel**

```
/# cat /proc/rn6318.0/playback_channel  
  
[RN6318#0]  
Current is not under playback mode!
```

- Set the value to switch the audio playback channel  
**echo [value] > /proc/rn6318.x/playback\_channel**

### 11.11.1.9 byapss\_channel

Users can use the `/proc/rn6318.x/bypass_channel` node to display and setup the audio bypass channel.

Usage:

- Get the current value, and x is ranging from 0 to 1.

**cat /proc/rn6318.x/bypass\_channel**

```
/# cat /proc/rn6318.0/bypass_channel  
  
[RN6318#0]  
Current Audio Playback Channel: 7
```

- Set the value to switch the audio bypass channel  
**echo [value] > /proc/rn6318.x/bypass\_channel**

### 11.11.1.10 sample\_rate

Users can use the `/proc/rn6318.x/sample_rate` node to display and setup the audio sample rate.

Usage:

- Get the current value, and x is ranging from 0 to 1.

**cat /proc/rn6318.x/sample\_rate**

```
/# cat /proc/rn6318.0/sample_rate  
  
[RN6318#0]  
Current Audio Sample Rate: 8 KBps
```

- Set the value to switch audio sample rate

**echo [value] > /proc/rn6318.x/sample\_rate**

Value: 8000, 16000, 32000, 44000, 48000

### 11.11.1.11 sample\_size

Users can use the `/proc/rn6318.x/sample_size` node to display and setup the audio sample size.

Usage:

- Get the current value, and x is ranging from 0 to 1.

**cat /proc/rn6318.x/sample\_size**

```
/# cat /proc/rn6318.0/sample_size  
  
[RN6318#0]  
Current Audio Sample Size: 16 Bit
```

- Set the value to switch audio sample size

**echo [value] > /proc/rn6318.x/sample\_size**

Value: 8, 16

## 11.12 RN6314 Driver Module

Table 11-12 lists and describes the module parameters when inserting this driver.

**Table 11-12. Parameters of RN6314 Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configuration
iaddr	{0x58, 0x5a, 0x5c, 0x5e}	Device I <sup>2</sup> C address
vmode	{2, 2, 2, 2}	Video output mode selection 0: NTSC 720H 1CH 1: NTSC 720H 2CH 2: NTSC 720H 4CH 3: NTSC 960H 1CH 4: NTSC 960H 2CH 5: NTSC 960H 4CH 6: PAL 720H 1CH 7: PAL 720H 2CH 8: PAL 720H 4CH 9: PAL 960H 1CH 10: PAL 960H 2CH 11: PAL 960H 4CH
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 <b>-1: It means that do not change the current frequency settings.</b>
clk_src	4	Clock source of the external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3



Name	Default Value	Description
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: GPIO#33 as RSTB pin for the device hardware reset
sample_rate	0	Audio sample rate 0: 8K 1: 16K 2: 32K 3: 44K 4: 48K
sample_size	1	Audio sample size 0: 16bits 1: 8bits
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module rn6314 driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: GM8210 socket board, vport 4CH mode GM8287 socket board Users must select the mapping table based on the hardware circuit.
audio_chnum	8	Device audio channel number

### 11.12.1 RN6314 Proc Nodes

The rn6314 driver module provides the proc nodes. Users can read the statuses of all channels or setup brightness/contrast/hue/saturation through these nodes. A sample of these nodes is listed below.

```
/# cat /proc/rn6314.0/
audio_volume      contrast          sample_rate      status
brightness        hue              sample_size      vmode
bypass_channel    playback_channel saturation

/# cat /proc/rn6314.1/
audio_volume      contrast          sample_rate      status
brightness        hue              sample_size      vmode
bypass_channel    playback_channel saturation

/# cat /proc/rn6314.2/
audio_volume      contrast          sample_rate      status
brightness        hue              sample_size      vmode
bypass_channel    playback_channel saturation

/# cat /proc/rn6314.3/
audio_volume      contrast          sample_rate      status
brightness        hue              sample_size      vmode
bypass_channel    playback_channel saturation
```

**Figure 11-12. Proc Nodes of RN6314 Module**

The following table shows the proc nodes of the rn6314 module.

/proc/rn6314.0	brightness
/proc/rn6314.1	contrast
/proc/rn6314.2	hue
/proc/rn6314.3	saturation
	audio_volume
	sample_rate
	sample_size
	playback_channel

	bypass_channel
	status
	vmode

### 11.12.1.1 brightness

Users can use the `/proc/rn6314.x/brightness` node to display and setup the brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/rn6314.x/brightness**

```

/# cat /proc/rn6314.0/brightness

[RN6314#0]
-----
VIN#      VCH#      BRIGHTNESS
-----
0         0         0x00
1         1         0x00
2         2         0x00
3         3         0x00

Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1

```

- Set the value to switch brightness

**echo [vin] [value] > /proc/rn6314.x/brightness**

vin: 0 ~ 3, other for all channels

Value: 0x00 ~ 0xff

### 11.12.1.2 contrast

Users can use the `/proc/rn6314.x/contrast` node to display and setup contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/rn6314.x/contrast**

```
/# cat /proc/rn6314.0/contrast

[RN6314#0]
-----
VIN#      VCH#      CONTRAST
-----
0         0         0x80
1         1         0x80
2         2         0x80
3         3         0x80

Contrast[0x00 ~ 0xff]
```

- Set the value to switch contrast

**echo [vin] [value] > /proc/rn6314.x/contrast**

vin: 0 ~ 3, others are for all channels.

Value: 0x00 ~ 0xff

### 11.12.1.3 hue

Users can use the /proc/rn6314.x/hue node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/rn6314.x/hue**

```
/# cat /proc/rn6314.0/hue
```

```
[RN6314#0]
```

```
-----  
VIN#      VCH#      HUE  
-----
```

```
0          0          0x80
```

```
1          1          0x80
```

```
2          2          0x80
```

```
3          3          0x80
```

```
Hue[0x00 ~ 0xff]
```

- Set the value to switch hue

**echo [vin] [value] > /proc/rn6314.x/hue**

vin: 0 ~ 3, others are for all channels.

Value: 0x00 ~ 0xff

#### 11.12.1.4 saturation

Users can use the /proc/rn6314.x/saturation node to display and setup saturation of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/rn6314.x/saturation**

```
/# cat /proc/rn6314.0/saturation
```

```
[RN6314#0]
```

```
-----  
VIN#      VCH#      SATURATION  
-----  
0         0         0x80  
1         1         0x80  
2         2         0x80  
3         3         0x80
```

```
Saturation[0x00 ~ 0xff]
```

- Set the value to switch saturation  
**echo [vin] [value] > /proc/rn6314.x/saturation**  
vin: 0 ~ 3, others are for all channels  
Value: 0x00 ~ 0xff

### 11.12.1.5 status

Users can use the /proc/rn6314.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/rn6314.x/status**

```
/# cat /proc/rn6314.0/status
```

```
[RN6314#0]
```

```
-----  
VIN#      VCH#      NOVID  
-----  
0          0          Video_Loss  
1          1          Video_Loss  
2          2          Video_Loss  
3          3          Video_Loss
```

### 11.12.1.6 vmode

Users can use the `/proc/rn6314.x/vmode` node to display the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/rn6314.x/vmode**

```
/# cat /proc/rn6314.0/vmode
```

```
[RN6314#0]
```

```
02: NTSC_720H_4CH
```

```
05: NTSC_960H_4CH
```

```
08: PAL_720H_4CH
```

```
11: PAL_960H_4CH
```

```
-----  
Current==> NTSC_720H_4CH
```

- Set the value to switch device video port output mode

**echo [mode] > /proc/rn6314.x/vmode**

mode: 2, 5, 8, 11

### 11.12.1.7 audio\_volume

Users can use the /proc/rn6314.x/audio\_volume node to display and setup the audio volume.

Usage:

- Get the current value, and x is ranging from 0 to 1.

**cat /proc/rn6314.x/audio\_volume**

```
/# cat /proc/rn6314.0/audio_volume
```

```
[RN6314#0]
```

```
-----  
AIN#      ACH#      VOL  
-----
```

```
0          0          0x8  
1          1          0x8  
2          2          0x8  
3          3          0x8
```

```
Volume[0x0 ~ 0xF]
```

- Set the value to switch audio volume

**echo [vin] [value] > /proc/rn6314.x/audio\_volume**

vin: 0 ~ 3

Value: 0x0 ~ 0xf

### 11.12.1.8 playback\_channel

Users can use the /proc/rn6314.x/playback\_channel node to display and setup audio playback channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/rn6314.x/playback\_channel**



```
/# cat /proc/rn6314.0/playback_channel  
  
[RN6314#0]  
Current is not under playback mode!
```

- Set the value to switch the audio playback channel  
**echo [value] > /proc/rn6314.x/playback\_channel**

#### 11.12.1.9 byapss\_channel

Users can use the /proc/rn6314.x/bypass\_channel node to display and setup the audio bypass channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/rn6314.x/bypass\_channel**

```
/# cat /proc/rn6314.0/bypass_channel  
  
[RN6314#0]  
Current Audio Playback Channel: 7
```

- Set the value to switch the audio bypass channel  
**echo [value] > /proc/rn6314.x/bypass\_channel**

#### 11.12.1.10 sample\_rate

Users can use the /proc/rn6314.x/sample\_rate node to display and setup the audio sample rate.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/rn6314.x/sample\_rate**

```
/# cat /proc/rn6314.0/sample_rate  
  
[RN6314#0]  
Current Audio Sample Rate: 8 KBps
```

- Set the value to switch the audio sample rate  
**echo [value] > /proc/rn6314.x/sample\_rate**  
Value: 8000, 16000, 32000, 44000, 48000

#### 11.12.1.11 sample\_size

Users can use the /proc/rn6314.x/sample\_size node to display and setup the audio sample size.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/rn6314.x/sample\_size**

```
/# cat /proc/rn6314.0/sample_size  
  
[RN6314#0]  
Current Audio Sample Size: 16 Bit
```

- Set the value to switch audio sample size  
**echo [value] > /proc/rn6314.x/sample\_size**  
Value: 8, 16

## 11.13 CX25930 Driver Module

Table 11-13 lists and describes the module parameters when inserting this driver.

**Table 11-13. Parameters of CX25930 Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configuration
iaddr	{0x88, 0x8a, 0x8c, 0x8e}	Device I <sup>2</sup> C address
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 <b>-1: It means that do not change the current frequency settings.</b>
clk_src	4	Clock source of external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: GPIO#33 as RSTB pin for the device hardware reset
input_rate	1	Input data rate 0: SD 1: HD 2: GEN3
vout_format	0	Video port output format 0: BT656 1: BT1120

Name	Default Value	Description
cab_len	0	Cable length 0: 0 ~ 50M (Auto) 1: 75 ~ 100M 2: 125 ~ 150M
sample_rate	4	Audio sample rate 0: 48K 1: 44K 2: 32K 3: 16K 4: 8K
sample_size	1	Audio sample size 0: 8bit 1: 16bit 2: 24bit
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module cx25930 driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: GM8210 socket/system board GM8287 socket board 1: GM8287 system board Users must select the mapping table based on the hardware circuit.

### 11.13.1 CX25930 Proc Nodes

The cx25930 driver module provides the proc nodes. Users can read the statuses of all channels through these nodes. A sample of these nodes is listed below.

```
/# cat /proc/cx25930.0/
loopback      reg_read      reg_write      status      video_format

/# cat /proc/cx25930.1/
loopback      reg_read      reg_write      status      video_format
```

**Figure 11-13. Proc Nodes of CX25930 Module**

The following table shows the proc nodes of the cx25930 module.

/proc/cx25930.0	loopback
/proc/cx25930.1	reg_read
/proc/cx25930.2	reg_write
/proc/cx25930.3	status
	video_format

#### 11.13.1.1 loopback

Users can use the /proc/cx25930.x/loopback node to setup the SDO loopback function.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/cx25930.x/brightness**

```
/ # cat /proc/cx25930.0/loopback
[CX25930#0]

-----

SDO_CH#0: TX Power Down
SDO_CH#1: SDI_CH#0 CDR Loopback
SDO_CH#2: SDI_CH#0 CDR Loopback
SDO_CH#3: SDI_CH#0 CDR Loopback

-----
```

```
echo [SDO_CH] [SDI_CH] [LBK_MODE] to node to control channel loopback function
[SDO_CH]    : 0~3
[SDI_CH]    : 0~3
[LBK_MODE]: 0: OFF, 1: EQ_LBK, 2: CDR_LBK
```

- Set the parameter to switch the SDO loopback mode

**echo [SDO\_CH] [SDI\_CH] [LBK\_MODE] > /proc/cx25930.x/loopback**

SDO\_CH: 0 ~ 3

SDI\_CH: 0 ~ 3

LBK\_MODE: 0: OFF 1: EQ\_LBK 2: CDR\_LBK

Example:

echo 0 1 2 > /proc/cx25930.0/loopback

SDO#0 is loopback from SDI#1 CDR

### 11.13.1.2 reg\_read

Users can use the /proc/cx25930.x/reg\_read node to read the cx25930 register.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/cx25930.x/reg\_read**

```

/ # cat /proc/cx25930.0/reg_read
[CX25930#0] 0x0000 => 0x0000
[CX25930#0] 0x0002 => 0x0000
[CX25930#0] 0x0004 => 0x00ff
[CX25930#0] 0x0006 => 0x0000
[CX25930#0] 0x0008 => 0x0000
[CX25930#0] 0x000a => 0x0000
[CX25930#0] 0x000c => 0x0000
[CX25930#0] 0x000e => 0x0000
[CX25930#0] 0x0010 => 0x0000
[CX25930#0] 0x0012 => 0x0000
-----
echo [reg] [read_count] to node to read data from register

```

- Set the reg and read\_count to specify the device read start from which register  
**echo [reg] [read\_count] > /proc/cx25930.x/reg\_read**  
reg: 0x0000 ~ 0xffff

### 11.13.1.3 reg\_write

Users can use the /proc/cx25930.x/reg\_write node to write data to the cx25930 register.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/cx25930.x/reg\_write**

```

/ # cat /proc/cx25930.0/reg_write
echo [reg] [data] to node to write data to register

```

- Set the reg and data to write data to cx25930 register  
**echo [reg] [data] > /proc/cx25930.x/reg\_write**  
reg: 0x0000 ~ 0xffff  
data: 0x0000 ~ 0xffff

#### 11.13.1.4 status

Users can use the `/proc/cx25930.x/status` node to display the video signal status of each SDI channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/cx25930.x/status**

```
/# cat /proc/cx25930.0/status
```

```
[CX25930#0]
```

```
-----
```

```
SDI      VCH      LOS
```

```
=====
```

```
0         0        NO
```

```
1         1        NO
```

```
2         2        YES
```

```
3         3        YES
```

#### 11.13.1.5 video\_format

Users can use the `/proc/cx25930.x/video_format` node to display the incoming video format of each SDI channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/cx25930.x/video\_format**



```

/# cat /proc/cx25930.0/video_format

[CX25930#0] Incoming Video Fromat
=====
SDI    VCH    Width  Height  Pixel_Rate(KHz)  Prog/Inter  Frame_Rate
=====
0      0      1920   1080    74250             Progressive  30
1      1      1920   1080    74250             Progressive  25
2      2      -       -        -                 -           -
3      3      -       -        -                 -           -

```

## 11.14 MV4101 Driver Module

Table 11-14 lists and describes the parameters when inserting this module.

**Table 11-14. Parameters of MV4101 Module**

Name	Default Value	Description
dev_num	4	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configuration
iaddr	{0x30, 0x32, 0x34, 0x36}	Device I <sup>2</sup> C address
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 <b>-1: It means that do not change the current frequency settings.</b>
clk_src	4	Clock source of external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3

Name	Default Value	Description
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: GPIO#33 as RSTB pin for the device hardware reset
vout_format	0	Video port output format 0: BT656 1: BT1120
eq_mode	1	EQ operation mode 0: Software auto 1: Hardware auto 2: Mode A line0 3: Mode A line1 4: Mode A line2 5: Mode A line3 6: Mode A line4 7: Mode A line5 8: Mode B line0 9: Mode B line1 10: Mode B line2 11: Mode B line3 12: Mode B line4 13: Mode B line5
spi_used	0	GPIO pin of SPI interface for device control 0: None 1: GPIO#47_48_51_52 2: GPIO#27_28_49_50 3: GPIO#49_50_27_28 4: GPIO#25_26_27_28 SCLK + CS + SDIN + SDOUT SPI pin
init	1	Device init control 0: No-init 1: init

Name	Default Value	Description
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module mv4101 driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: GM8210 socket/system board GM8287 socket board 1: GM8287 system board <b>Users must select the mapping table based on the hardware circuit.</b>

### 11.14.1 MV4101 Proc Nodes

The mv4101 driver module provides the proc nodes. Users can read the statuses of all channels through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/mv4101.0/
reg_read      reg_write      status      video_format  vout_sdi

/# cat /proc/mv4101.1/
reg_read      reg_write      status      video_format  vout_sdi

```

**Figure 11-14. Proc Nodes of MV4101 Module**

The following table shows the proc nodes of the mv4101 module.

/proc/mv4101.0	reg_read
/proc/mv4101.1	reg_write
/proc/mv4101.2	status
/proc/mv4101.3	video_format
	vout_sdi

### 11.14.1.1 reg\_read

Users can use the `/proc/mv4101.x/reg_read` node to read the mv4101 register.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/mv4101.x/reg\_read**

```
/ # cat /proc/mv4101.0/reg_read
[MV4101#0] 0x0000 => 0x78c1
[MV4101#0] 0x0001 => 0x0400
[MV4101#0] 0x0002 => 0x0000
[MV4101#0] 0x0003 => 0x0000
[MV4101#0] 0x0004 => 0x0077
[MV4101#0] 0x0005 => 0x8210
[MV4101#0] 0x0006 => 0x0044
[MV4101#0] 0x0007 => 0x0000
[MV4101#0] 0x0008 => 0x0000
[MV4101#0] 0x0009 => 0x0000
-----
echo [reg] [count] to node to read data from address
```

- Set the reg and count to specify device read start from which register

**echo [reg] [count] > /proc/mv4101.x/reg\_read**

reg: 0x0000 ~ 0xffff

### 11.14.1.2 reg\_write

Users can use the `/proc/mv4101.x/reg_write` node to write data to the mv4101 register.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/mv4101.x/reg\_write**

```
/ # cat /proc/mv4101.0/reg_write
echo [reg] [data] to node to write data to address
```

- Set the reg and data to write data to the mv4101 register

**echo [reg] [data] > /proc/mv4101.x/reg\_write**

reg: 0x0000 ~ 0xffff

data: 0x0000 ~ 0xffff

### 11.14.1.3 status

Users can use the /proc/mv4101.x/status node to display the video signal status of each SDI channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/mv4101.x/status**

```
/# cat /proc/mv4101.0/status

[MV4101#0]
-----
SDI      VCH      LOS      EQ_MODE
=====
0         0        NO       HW_AUTO
1         1        YES      HW_AUTO
2         2        YES      HW_AUTO
3         3        YES      HW_AUTO
```

### 11.14.1.4 video\_format

Users can use the /proc/mv4101.x/video\_format node to display the incoming video format of each SDI channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/mv4101.x/video\_format**

```
/# cat /proc/cx25930.0/video_format

[MV4101#0] Incoming Video Fromat
=====
SDI      VCH      Width    Height    Pixel_Rate(KHz)  Prog/Inter  Frame_Rate
=====
0        0        1920     1080     74250            Progressive  30
1        1        -        -        -                -           -
2        2        -        -        -                -           -
3        3        -        -        -                -           -
```

#### 11.14.1.5 vout\_sdi

Users can the use /proc/mv4101.x/vout\_sdi node to display and setup the SDI channel output to which video port.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/mv4101.x/vout\_sdi**

```

/# cat /proc/mv4101.0/vout_sdi

[MV4101#0]
-----

VOUT_CH#0: SDI_CH#0
VOUT_CH#1: SDI_CH#1
VOUT_CH#2: SDI_CH#2
VOUT_CH#3: SDI_CH#3
-----

echo [VOUT_CH] [SDI_CH] to node to control vout from which sdi input source
[VOUT_CH]: 0~3
[SDI_CH] : 0~3

```

- Set the VOUT\_CH and SDI\_CH to switch VOUT SDI channel.  
**echo [VOUT\_CH] [SDI\_CH] > /proc/mv4101.x/vout\_sdi**  
VOUT\_CH: 0 ~ 3  
SDI\_CH: 0 ~ 3

## 11.15 DH9901 Driver Module

Table 11-15 lists and describes the parameters when inserting this module.

**Table 11-15. Parameters of DH9901 Module**

Name	Default Value	Description
dev_num	1	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in the kernel configuration.
iaddr	{0x60, 0x62, 0x64, 0x66}	Device I <sup>2</sup> C address
clk_used	0	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 -1: It means that do not touch the current frequency settings.

Name	Default Value	Description
clk_src	4	Clock source of external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: GPIO#33 as RSTB pin for the device hardware reset
vout_format	0	Video port output format 0: BT656 1: BT1120
sample_rate	0	Audio sample rate 0: 8K 1: 16K
sample_szie	1	Audio sample size 0: 8bit 1: 16bit
init	1	Device init control 0: No-init 1: init
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module dh9901 driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: GM8210 socket/system board GM8287 socket board User must select the mapping table based on the hardware circuit.



11.15.1 DH9901 Proc Nodes

The dh9901 driver module provides the proc nodes. Users can read the statuses of all channels through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/dh9901.0/
audio_volume  clear_eq      status      video_format
cable_type    ptz          video_color video_pos

/# cat /proc/dh9901.1/
audio_volume  clear_eq      status      video_format
cable_type    ptz          video_color video_pos
```

Figure 11-15. Proc Nodes of DH9901 Module

The following table shows the proc nodes of the dh9901 module.

/proc/dh9901.0	audio_volume	
/proc/dh9901.1	clear_eq	
/proc/dh9901.2	status	
/proc/dh9901.3	video_format	
	cable_type	
	video_color	
	video_pos	
	ptz	cfg
		control

11.15.1.1 audio\_volume

Users can the use /proc/dh9901.x/audio\_volume node to display and setup the audio volume.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/dh9901.x/audio\_volume**

```

/# cat /proc/dh9901.0/audio_volume

[DH9901#0] Audio Volume
=====
echo [CVI#] [Volume(0~100)] for setup audio volume

```

- Set the CVI# and Volume to switch the audio volume.

**echo [CVI] [Volume] > /proc/dh9901.x/audio\_volume**

CVI: 0 ~ 3

Volume: 0 ~ 100

### 11.15.1.2 clear\_eq

Users can use the /proc/dh9901.x/clear\_eq node to enable channel EQ clear.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/dh9901.x/clear\_eq**

```

/# cat /proc/dh9901.0/clear_eq

[DH9901#0] Clear EQ
=====
echo [CVI#] for trigger channel EQ clear

```

- Set the CVI# to trigger channel EQ clear.

**echo [CVI] > /proc/dh9901.x/clear\_eq**

CVI: 0 ~ 3

### 11.15.1.3 status

Users can use the /proc/dh9901.x/status to display the video signal status of each CVI channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/dh9901.x/status**

```
/# cat /proc/dh9901.0/status

[DH9901#0]
-----
CVI      VCH      LOS
=====
0        0        YES
1        1        YES
2        2        YES
3        3        YES
```

#### 11.15.1.4 video\_format

Users can use /proc/dh9901.x/video\_format to display the incoming video format of each CVI channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/dh9901.x/video\_format**

```
/# cat /proc/dh9901.0/video_format

[DH9901#0] Incoming Video Fromat
=====
CVI      VCH      Width  Height  Prog/Inter  Frame_Rate
=====
0        0        1280   720     Progressive  25
1        1        1280   720     Progressive  25
2        2        1280   720     Progressive  25
3        3        1280   720     Progressive  25
```

### 11.15.1.5 cable\_type

Users can use /proc/dh9901.x/cable\_type to setup the cable type of each CVI channel

Usage:

- Get the current value, and x is ranging from 0 to 3.

#### **cat /proc/dh9901.x/cable\_type**

```
/# cat /proc/dh9901.0/cable_type

[DH9901#0] Cable Type
=====
[0]COAXIAL
[1]UTP_10OHM
[2]UTP_17OHM
[3]UTP_25OHM
[4]UTP_35OHM
=====
echo [CVI#] [TYPE#] for cable type setup
```

- Set the CVI# and TYPE# to switch CVI channel cable type.

#### **echo [CVI] [TYPE] > /proc/dh9901.x/cable\_type**

CVI: 0 ~ 3

TYPE:

- 0: COAXIAL
- 1: UTP\_10OHM
- 2: UTP\_17OHM
- 3: UTP\_25OHM
- 4: UTP\_35OHM

### 11.15.1.6 video\_color

Users can use /proc/dh9901.x/video\_color to display and setup video color of each CVI channel.

#### Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/dh9901.x/video\_color**

```
/# cat /proc/dh9901.0/video_color

[DH9901#0] Video Color
=====
CVI    VCH    BRI    CON    SAT    HUE    GAIN    W/B    SHARP
=====
0      0      50     50     50     50     0       0      1
1      1      50     50     50     50     0       0      1
2      2      50     50     50     50     0       0      1
3      3      50     50     50     50     0       0      1
=====

[0]Brightness : 0 ~ 100
[1]Contrast   : 0 ~ 100
[2]Saturation : 0 ~ 100
[3]Hue        : 0 ~ 100
[4]Gain       : Not support
[5]WhiteBalance: Not support
[6]Sharpness  : 0 ~ 15
=====

echo [CVI#] [PARAM#] [VALUE] for parameter setup
```

- Set the parameter to switch the video color of each CVI channel.

**echo [CVI] [PARAM] [VALUE] > /proc/dh9901.x/video\_color**

CVI: 0 ~ 3

PARAM:

- 0: Brightness, 0 ~ 100
- 1: Contrast, 0 ~ 100
- 2: Saturation, 0 ~ 100
- 3: Hue, 0 ~ 100
- 4: Gain, not support
- 5: White Balance, not support
- 6: Sharpness, 0 ~ 15

### 11.15.1.7 video\_pos

Users can use /proc/dh9901.x/video\_pos to display and setup the video position of each CVI channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/dh9901.x/video\_pos**

```
/# cat /proc/dh9901.0/video_pos

[DH9901#0] Video Position
-----
CVI      VCH      H_Offset  V_Offset
=====
0         0         5          0
1         1         5          0
2         2         5          0
3         3         5          0
=====
echo [CVI#] [H_Offset] [V_Offset] for video position setup
```

- Set the parameter to switch the video position of each CVI channel.

**echo [CVI] [H\_Offset] [V\_Offset] > /proc/dh9901.x/video\_pos**

CVI: 0 ~ 3

### 11.15.2 DH9901 PTZ Proc Nodes

Users can use these proc nodes to get the current PTZ configuration and control OSD menu of the remote camera with the specified PTZ protocol.

```
/# cat /proc/dh9901.0/ptz
cfg      control
/# cat /proc/dh9901.1/ptz
cfg      control
```

### 11.15.2.1 cfg

Users can use `/proc/dh9901.x/ptz/cfg` to display the PTZ protocol configuration of each CVI channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**`cat /proc/dh9901.x/ptz/cfg`**

```
/# cat /proc/dh9901.0/ptz/cfg

[DH9901#0] PTZ Configuration
-----
CVI    VCH    PROTOCOL  BAUD_RATE  PARITY_CHK
=====
0      0      DH_SD1    9600        Yes
1      1      DH_SD1    9600        Yes
2      2      DH_SD1    9600        Yes
3      3      DH_SD1    9600        Yes
```

### 11.15.2.2 control

Users can use `/proc/dh9901.x/ptz/control` to control remote camera.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**`cat /proc/dh9901.x/ptz/control`**

```

/# cat /proc/dh9901.0/ptz/control

[DH9901#0]
<< DH_SD1 Command List >>
=====
[0]Menu_CLOSE
[1]Menu_OPEN
[2]Menu_BACK
[2]Menu_NEXT
[4]Menu_UP
[5]Menu_DOWN
[6]Menu_LEFT
[7]Menu_RIGHT
[8]Menu_ENTER
=====
echo [CVI#] [CAMERA#(0~255)] [CMD#] for remote camera control

```

- Set the parameter to send command to remote camera.

**echo [CVI] [CAMERA\_ID] [CMD] > /proc/dh9901.x/ptz/control**

CVI: 0 ~ 3

CAMERA\_ID: 0 ~ 255

CMD: 0 ~ 8 for DH\_SD1 protocol

## 11.16 NVP6114 Driver Module

Table 11-16 lists and describes the module parameters when inserting this driver.

**Table 11-16. Parameters of NVP6114 Module**

Name	Default Value	Description
dev_num	2	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in kernel configuration
iaddr	{0x60, 0x62, 0x64, 0x66}	Device I <sup>2</sup> C address



Name	Default Value	Description
vmode	7	Device video mode
		0: NTSC_720H_1CH
		1: NTSC_720H_2CH
		2: NTSC_720H_4CH
		3: NTSC_960H_1CH
		4: NTSC_960H_2CH
		5: NTSC_960H_4CH
		6: NTSC_720P_2CH
		7: NTSC_720P_4CH
		8: PAL_720H_1CH
		9: PAL_720H_2CH
		10: PAL_720H_4CH
		11: PAL_960H_1CH
		12: PAL_960H_2CH
		13: PAL_960H_4CH
		14: PAL_720P_2CH
		15: PAL_720P_4CH
		16: NTSC_720H_720P_2CH
		17: NTSC_720P_720H_2CH
		18: NTSC_960H_720P_2CH
		19: NTSC_720P_960H_2CH
		20: PAL_720H_720P_2CH
		21: PAL_720P_720H_2CH
		22: PAL_960H_720P_2CH
		23: PAL_720P_960H_2CH
clk_used	0x3	External clock port selection
		BIT0: EXT_CLK0
		BIT1: EXT_CLK1
		-1: It means that do not change the current frequency settings.

Name	Default Value	Description
clk_src	1	Clock source of external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: GPIO#33 as RSTB pin for the device hardware reset
init	1	Device init control 0: No-init 1: init
sample_rate	0	Audio sample rate 0: 8K 1: 16K
sample_size	0	Audio sample size 0: 16bit 1: 8bit
audio_chnum	8	Audio channel number
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module nvp6114 driver for the video interface to switch configuration.

Name	Default Value	Description
ch_map	{0, 0}	Device video channel mapping table ch_map[0] for map index 0: Generic, vport 2CH mode 1: GM8210 socket board, vport 2CH mode GM8287 socket board, vport 2CH mode 2: Generic, vport 4CH mode <b>Users must select the mapping table based on the hardware circuit.</b> ch_map[1] for VCH start index

### 11.16.1 NVP6114 Proc Nodes

The nvp6114 driver module provides the proc nodes. Users can read the statuses of all channels through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/nvp6114.0/
brightness hue          sharpness  vmode
contrast   saturation    status

/# cat /proc/nvp6114.1/
brightness hue          sharpness  vmode
contrast   saturation    status

```

**Figure 11-16. Proc Nodes of NVP6114 Module**

The following table shows the proc nodes of the nvp6114 module.

/proc/nvp6114.0	brightness
/proc/nvp6114.1	contrast
/proc/nvp6114.2	hue
/proc/nvp6114.3	saturation
	sharpness
	status
	vmode

### 11.16.1.1 brightness

Users can use the `/proc/nvp6114.x/brightness` node to display and setup the brightness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp6114.x/brightness**

```
/# cat /proc/nvp6114.0/brightness

[NVP6114#0]
-----
VIN#      VCH#      BRIGHTNESS
-----
0         0         0x08
1         1         0x08
2         2         0x08
3         3         0x08

Brightness[-128 ~ +127] ==> 0x01=+1, 0x7f=+127, 0x80=-128, 0xff=-1
```

- Set the value to switch brightness

**echo [vin] [value] > /proc/nvp6114.x/brightness**

vin: 0 ~ 3, others are for all channels.

Value: 0x00 ~ 0xff

### 11.16.1.2 contrast

Users can use the `/proc/nvp6114.x/contrast` node to display and setup the contrast of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp6114.x/contrast**

```
/# cat /proc/nvp6114.0/contrast
```

```
[NVP6114#0]
```

```
-----  
VIN#      VCH#      CONTRAST  
-----
```

```
0          0          0x88
```

```
1          1          0x88
```

```
2          2          0x88
```

```
3          3          0x88
```

```
Contrast[0x00 ~ 0xff] ==> 0x00=x0, 0x40=x0.5, 0x80=x1, 0xff=x2
```

- Set the value to switch contrast

**echo [vin] [value] > /proc/nvp6114.x/contrast**

vin: 0 ~ 3, others are for all channels.

Value: 0x00 ~ 0xff

### 11.16.1.3 hue

Users can use the /proc/nvp6114.x/hue node to display and setup hue of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp6114.x/hue**

```
/# cat /proc/nvp6114.0/hue
```

```
[NVP6114#0]
```

```
-----  
VIN#      VCH#      HUE  
-----
```

```
0          0          0x00
```

```
1          1          0x00
```

```
2          2          0x00
```

```
3          3          0x00
```

```
Hue[0x00 ~ 0xff] ==> 0x00=0, 0x40=90, 0x80=180, 0xff=360
```

- Set the value to switch hue

**echo [vin] [value] > /proc/nvp6114.x/hue**

vin: 0 ~ 3, others for all channels.

Value: 0x00 ~ 0xff

### 11.16.1.4 saturation

Users can use the /proc/nvp6114.x/saturation node to display and setup saturation of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp6114.x/saturation**

```

/# cat /proc/nvp6114.0/saturation

[NVP6114#0]
-----
VIN#      VCH#      SATURATION
-----
0         0         0x84
1         1         0x84
2         2         0x84
3         3         0x84

Saturation[0x00 ~ 0xff] ==> 0x00=x0, 0x80=x1, 0xc0=x1.5, 0xff=x2

```

- Set the value to switch saturation  
**echo [vin] [value] > /proc/nvp6114.x/saturation**  
vin: 0 ~ 3, others are for all channels.  
Value: 0x00 ~ 0xff

### 11.16.1.5 sharpness

Users can the use /proc/nvp6114.x/sharpness node to display and setup the sharpness of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/nvp6114.x/sharpness**

```

/# cat /proc/nvp6114.0/sharpness

[NVP6114#0]
-----
VIN#      VCH#      SHARPNESS
-----
0         0         0x9f
1         1         0x9f
2         2         0x9f
3         3         0x9f

H_Sharpness[0x0 ~ 0xf] - Bit[7:4] ==> 0x0:x0, 0x4:x0.5, 0x8:x1,

```

- Set the value to switch sharpness  
**echo [vin] [value] > /proc/nvp6114.x/sharpness**  
vin: 0 ~ 3, others are for all channels.  
Value: 0x00 ~ 0xff

### 11.16.1.6 status

Users can use the /proc/nvp6114.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.  
**cat /proc/nvp6114.x/status**



```
/# cat /proc/nvp6114.0/status
```

```
[NVP6114#0]
```

```
-----  
VIN#      VCH#      NOVID      INPUT_VFMT  
-----  
0         0         Video_Loss  SD  
1         1         Video_Loss  SD  
2         2         Video_Loss  SD  
3         3         Video_Loss  Unknown
```

### 11.16.1.7 vmode

Users can use the `/proc/nvp6114.x/vmode` node to display and setup the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/nvp6114.x/vmode**

```
/# cat /proc/nvp6114.0/vmode
```

```
[NVP6114#0]
```

```
04: NTSC_960H_2CH
```

```
07: NTSC_720P_2CH
```

```
12: PAL_960H_2CH
```

```
15: PAL_720P_2CH
```

```
18: NTSC_960H_720P_2CH
```

```
19: NTSC_720P_960H_2CH
```

```
22: PAL_960H_720P_2CH
```

```
23: PAL_720P_960H_2CH  
-----
```

```
Current==> PAL_720P_2CH
```

- Set the value to switch device video port output mode

**echo [mode] > /proc/nvp6114.x/vmode**

mode:

04: NTSC\_960H\_2CH

→ CH#0 ~ 3 as 960H NTSC, video port output 2CH dual edge mode.

07: NTSC\_720P\_2CH

→ CH#0 ~ 3 as 720P NTSC, video port output 2CH dual edge mode.

12: PAL\_960H\_2CH

→ CH#0 ~ 3 as 960H NTSC, video port output 2CH dual edge mode.

15: PAL\_720P\_2CH

→ CH#0 ~ 3 as 720P PAL, video port output 2CH dual edge mode.

18: NTSC\_960H\_720P\_2CH

→ CH#0 ~ 1 as 960H NTSC, CH#2 ~ 3 as 720P NTSC, video port output 2CH dual edge mode.

19: NTSC\_720P\_960H\_2CH

→ CH#0 ~ 1 as 720P NTSC, CH#2 ~ 3 as 960H NTSC, video port output 2CH dual edge mode.

22: PAL\_960H\_720P\_2CH

→ CH#0 ~ 1 as 960H PAL, CH#2 ~ 3 as 720P PAL, video port output 2CH dual edge mode.

23: PAL\_720P\_960H\_2CH

→ CH#0 ~ 1 as 720P PAL, CH#2 ~ 3 as 960H PAL, video port output 2CH dual edge mode

## 11.17 TP2802 Driver Module

Table 11-17 lists and describes the module parameters when inserting this driver.

**Table 11-17. Parameters of TP2802 Module**

Name	Default Value	Description
dev_num	1	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in kernel configuration
iaddr	{0x88, 0x8a, 0x8c, 0x8e}	Device I <sup>2</sup> C address
clk_used	0x3	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 <b>-1: It means that do not change the current frequency settings.</b>
clk_src	1	Clock source of external clock port 0: PLL1OUT1 1: PLL1OUT1/2 2: PLL4OUT2 3: PLL4OUT1/2 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: GPIO#33 as RSTB pin for the device hardware reset
vout_format	0	Device video port output format 0: BT656 1: BT1120
init	1	Device init control 0: No-init 1: init

Name	Default Value	Description
notify	1	Device notify 0: Disable 1: Enable Notify the video status and video format to capture the input_module tp2802 driver for the video interface to switch configuration.
ch_map	0	Device video channel mapping table 0: GM8210 system board GM8287 system board 1: GM8210 socket board GM8287 socket board <b>Users must select the mapping table based on the hardware circuit.</b>

### 11.17.1 TP2802 Proc Nodes

The tp2802 driver module provides the proc nodes. Users can read the statuses of all channels through these nodes. A sample of these nodes is listed below.

```

/# cat /proc/tp2802.0/
status          video_format  vout_format

/# cat /proc/tp2802.1/
status          video_format  vout_format

```

**Figure 11-17. Proc Nodes of TP2802 Module**

The following table shows the proc nodes of the nvp6114 module.

/proc/tp2802.0	status
/proc/tp2802.1	video_format
/proc/tp2802.2	vout_format
/proc/tp2802.3	

### 11.17.1.1 status

Users can use the /proc/tp2802.x/status node to display the status of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tp2802.x/status**

```
/# cat /proc/tp2802.0/status
```

```
[TP2802#0]
```

```
-----
```

```
VIN      VCH      LOS
```

```
=====
```

```
0         0         NO
```

```
1         1         YES
```

```
2         2         YES
```

```
3         3         YES
```

### 11.17.1.2 video\_format

Users can use the /proc/tp2802.x/video\_format node to display the incoming video format of each channel.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tp2802.x/video\_format**

```

/# cat /proc/tp2802.0/video_format

[TP2802#0] Incoming Video Format
=====
VIN      VCH      Width  Height  Prog/Inter  Frame_Rate
=====
0        0        1280   720     progressive  30
1        1        -       -       -           -
2        2        -       -       -           -
3        3        -       -       -           -

```

### 11.17.1.3 vout\_format

Users can use the `/proc/tp2802.x/vout_format` node to display the video port output format.

Usage:

- Get the current value, and x is ranging from 0 to 3.

**cat /proc/tp2802.x/vout\_format**

```

/# cat /proc/tp2802.0/vout_format

[TP2802#0] Video Output Format
=====
VOUT     VCH     Width  Height  Prog/Inter  Frame_Rate
=====
0        0        1920   1080    Progressive  30
1        1        1920   1080    Progressive  30
2        2        1920   1080    Progressive  30
3        3        1920   1080    Progressive  30

```

## 11.18 MT9M131 Driver Module

Table 11-18 lists and describes the module parameters when inserting this driver.

**Table 11-18. Parameters of MT9M131 Module**

Name	Default Value	Description
dev_num	1	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in kernel configuration
iaddr	{0xBA, 0x90,}	Device I <sup>2</sup> C address
vmode	{0, 0}	Video output mode selection 0: VGA 640x480 1: QSXGA 640x512 2: SXGA 1280x1024
clk_used	0x1	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 (Not supported) <b>-1: It means that do not change the current frequency settings.</b>
clk_src	4	Clock source of the external clock port 0: PLL1OUT1 (Not supported) 1: PLL1OUT1/2 (Not supported) 2: PLL4OUT2 (Not supported) 3: PLL4OUT1/2 (Not supported) 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
inti	1	Device init control 0: No-init 1: init
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: X_CAP_RST as the RSTB pin for the device hardware reset

Name	Default Value	Description
ch_id	{0, 1}	Device video channel index

### 11.18.1 MT9M131 Proc Nodes

The MT9M131 driver module provides the proc nodes. User can read some information through these proc nodes. A sample of these nodes is listed below.

```

/# cat /proc/mt9m131.0/
vmode

```

Figure 11-18. Proc Nodes of MT9M131 Module

The following table shows the proc nodes of the MT9M131 module.

/proc/mt9m131.0	vmode
/proc/mt9m131.1	

#### 11.18.1.1 vmode

Users can use the /proc/mt9m131.x/vmode node to display the video output mode.

Usage:

- Get the current value, and x is ranging from 0 to 1.

**cat /proc/mt9m131.x/vmode**

```

/# cat /proc/mt9m131.0/vmode

[MT9M131#0]
00: VGA_640x480
01: QSXGA_640x512
02: SXGA_1280x1024
-----
Current==> VGA_640x480

```



## 11.19 MT9D131 Driver Module

Table 11-19 lists and describes the parameters when inserting this module.

**Table 11-19. Parameters of MT9D131 Module**

Name	Default Value	Description
dev_num	1	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in kernel configuration
iaddr	0x90	Device I <sup>2</sup> C address
vmode	0	Video output mode selection 0: SVGA 800x600 1: HD720 1280x720 2: WXGA 1280x800 3: UXGA 1600x1200
clk_used	0x1	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 (Not supported) <b>-1: It means that do not change the current frequency settings.</b>
clk_src	4	Clock source of the external clock port 0: PLL1OUT1 (Not supported) 1: PLL1OUT1/2 (Not supported) 2: PLL4OUT2 (Not supported) 3: PLL4OUT1/2 (Not supported) 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
inti	1	Device init control 0: No-init 1: init
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: X_CAP_RST as the RSTB pin for the device hardware reset

Name	Default Value	Description
ch_id	0	Device video channel index

### 11.19.1 MT9D131 Proc Nodes

The MT9D131 driver module provides the proc nodes. User can read some information through these proc nodes. A sample of these nodes is listed below.

```

/# cat /proc/mt9d131.0/
vmode

```

Figure 11-19. Proc Nodes of MT9D131 Module

The following table shows the proc nodes of the MT9D131 module.

/proc/mt9d131.0	vmode
-----------------	-------

#### 11.19.1.1 vmode

Users can use the /proc/mt9d131.x/vmode node to display the video output mode.

Usage:

- Get the current value, and x is 0.

**cat /proc/mt9d131.x/vmode**

```

/# cat /proc/mt9d131.0/vmode

[MT9D131#0]
00: SVGA_800x600
01: HD720_1280x720
02: WXGA_1280x800
03: UXGA_1600x1200
-----
Current==> HD720_1280x720

```

## 11.20 HM1375 Driver Module

Table 11-20 lists and describes the parameters when inserting this module.

**Table 11-20. Parameters of HM1375 Module**

Name	Default Value	Description
dev_num	1	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in kernel configuration.
iaddr	0x48	Device I <sup>2</sup> C address
vmode	0	Video output mode selection 0: VGA 640x480 1: HD720 1280x720 2: WXGA 1280x960 3: SXGA 1280x1024
clk_used	0x1	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 (Not supported) <b>-1: It means that do not change the current frequency settings.</b>
clk_src	4	Clock source of the external clock port 0: PLL1OUT1 (Not supported) 1: PLL1OUT1/2 (Not supported) 2: PLL4OUT2 (Not supported) 3: PLL4OUT1/2 (Not supported) 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
inti	1	Device init control 0: No-init 1: init
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: X_CAP_RST as RSTB pin for the device hardware reset

Name	Default Value	Description
is_50hz		0: 60Hz 1: 50Hz
ch_id	0	Device video channel index

### 11.20.1 HM1375 Proc Nodes

The HM1375 driver module provides the proc nodes. User can read some information through these proc nodes. A sample of these nodes is listed below.

```
/# cat /proc/hm1375.0/  
vmode
```

**Figure 11-20. Proc Nodes of HM1375 Module**

The following table shows the proc nodes of the HM1375 module.

/proc/hm1375.0	vmode
----------------	-------

#### 11.20.1.1 vmode

Users can use the /proc/hm1375.x/vmode node to display the video output mode.

Usage:

- Get the current value, and x is 0.

**cat /proc/hm1375.x/vmode**

```
/# cat /proc/hm1375.0/vmode  
  
[HM1375#0]  
00: VGA_640x480  
01: HD720_1280x720  
02: WXGA_1280x960  
03: SXGA_1280x1024  
-----  
Current==> HD720_1280x720 60Hz
```

## 11.21 OV7725 Driver Module

Table 11-21 lists and describes the module parameters when inserting this driver.

**Table 11-21. Parameters of HM1375 Module**

Name	Default Value	Description
dev_num	1	Device number
ibus	0	Device attached I <sup>2</sup> C bus number 0 ~ 4 Users must enable the I <sup>2</sup> C bus support in kernel configuration.
iaddr	0x42	Device I <sup>2</sup> C address
vmode	0	Video output mode selection 0: VGA 640x480
clk_used	0x1	External clock port selection BIT0: EXT_CLK0 BIT1: EXT_CLK1 (Not supported) <b>-1: It means that do not change the current frequency settings.</b>
clk_src	4	Clock source of the external clock port 0: PLL1OUT1 (Not supported) 1: PLL1OUT1/2 (Not supported) 2: PLL4OUT2 (Not supported) 3: PLL4OUT1/2 (Not supported) 4: PLL3
clk_freq	27000000	Clock frequency of the external clock port xxx Hz
inti	1	Device init control 0: No-init 1: init
rstb_used	1	Control GPIO pin as device RSTB pin 0: Not used 1: X_CAP_RST as the RSTB pin for the device hardware reset
ch_id	0	Device video channel index

### 11.21.1 OV7725 Proc Nodes

The OV7725 driver module provides the proc nodes. User can read some information through these proc nodes. A sample of these nodes is listed below.

```
/# cat /proc/ov7725.0/  
vmode
```

**Figure 11-21. Proc Nodes of OV7725 Module**

The following table shows the proc nodes of the OV7725 module.

/proc/hm1375.0	vmode
----------------	-------

#### 11.21.1.1 vmode

Users can use the /proc/ov7725.x/vmode node to display the video output mode.

Usage:

- Get the current value, and x is 0.

**cat /proc/ov7725.x/vmode**

```
/# cat /proc/ov7725.0/vmode  
  
[OV7725#0]  
00: VGA_640x480  
-----  
Current==> VGA_640x480
```

# Chapter 12

## Front-end and VCAP300

### Input\_Module Driver Porting

---

This chapter contains the following sections:

- 12.1 Video Channel Mapping
- 12.2 Watch Dog Thread
- 12.3 Summary of Driver Porting

In the GM8136 SDK release package, Grain Media collected all video/audio codec drivers in `/module/front_end` directory. Each front-end codec driver corresponds to one vcap300 input module driver in `/module/vcap300/input_module` directory. The purpose of the front-end codec driver is to initialize codec and provide the proc node, IOCTL device node in Linux user space for the user application to get the codec video signal status, switch video format, adjust video color, and so on. The codec driver applies a notify procedure to inform the corresponding vcap300 input\_module driver. The input\_module driver will dynamically switch the port setting of the capture video interface according to the codec driver notify information. Figure 12-1 is an example for the driver architecture of nvp1914 and vcap300\_nvp1914.



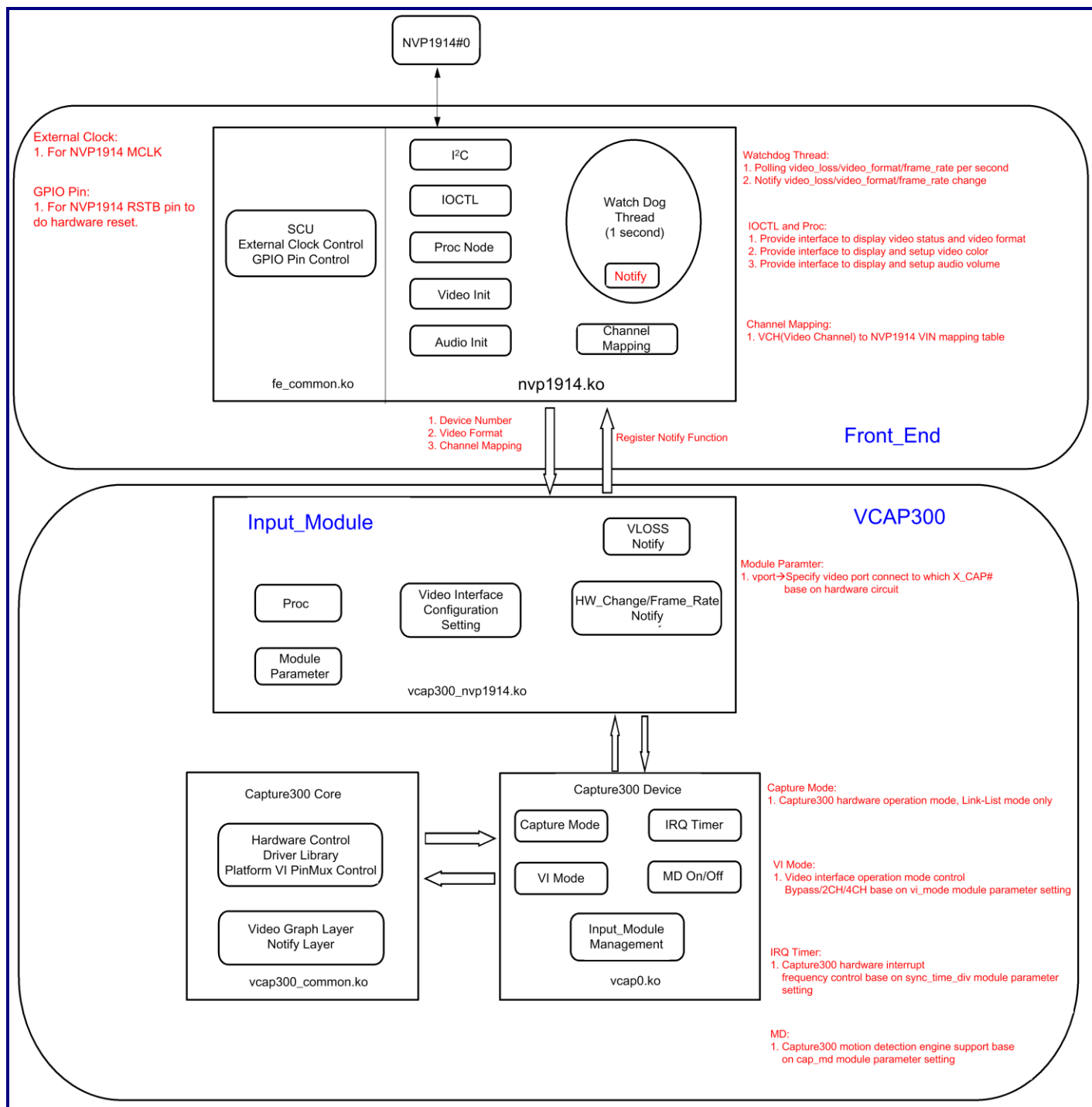


Figure 12-1. Driver Architecture of NVP1914 and VCAP300\_NVP1914

## 12.1 Video Channel Mapping

Because of the different circuit layouts in various hardware boards, the hardware video connector may not sequentially link to the codec video channel. Grain Media designs a video channel mapping mechanism in the front-end codec driver to transfer the video connector index to the codec video channel index and vcap300 capture channel hardware index. The user application only needs to concern the video connector index to get the video frame from the middleware API. Grain Media have created some mapping tables in the codec driver based on the EVB. Users can modify the codec driver to specify one mapping table. Users also can dump `/proc/vcap300/vcap0/vg_info` and `/proc/vcap300/input_module/table` to get the hardware channel mapping information. Figure 12-2 is an example of the mapping table for NVP1914 video channel with VI as 4CH MUX mode. Figure 12-3 is an example of the mapping table for OV7725 + MT9M136 video channel with VI as bypass mode. For the codec video output port and X\_CAP port number, users must setup the module parameter "vport" of the codec driver based on the circuit.

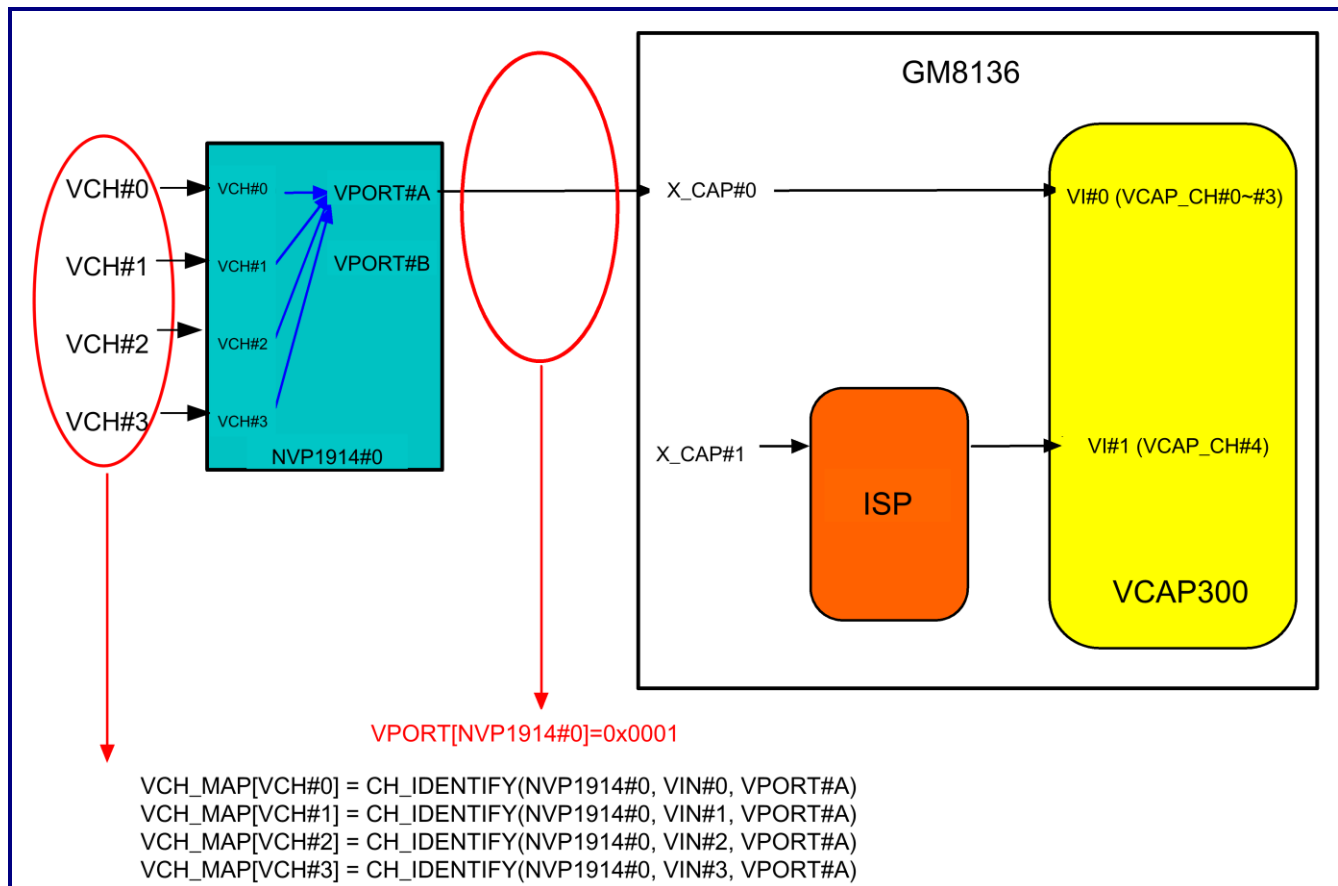


Figure 12-2. Mapping Table for NVP1914 Video Channel

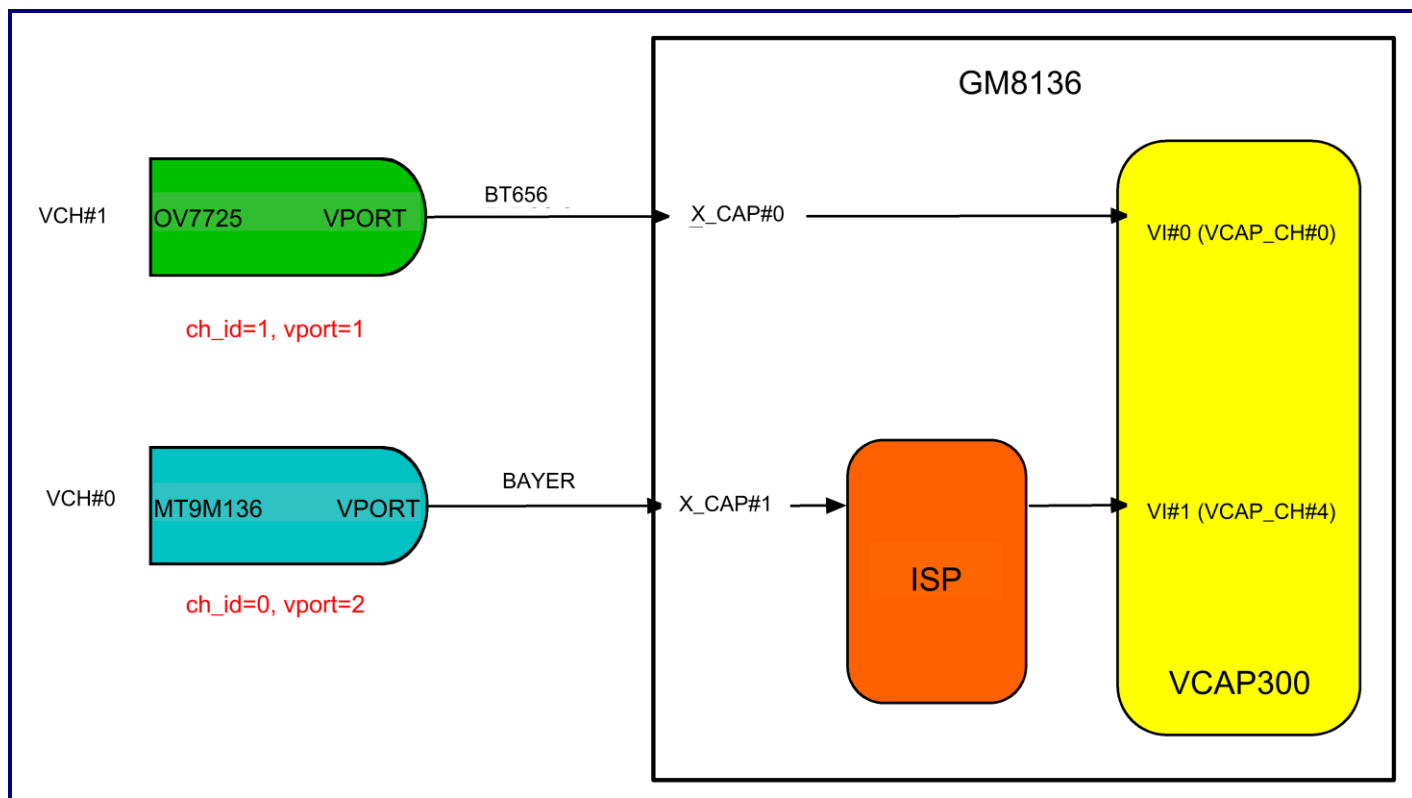


Figure 12-3. Mapping Table for OV7725 + MT9M136 Video Channel

## 12.2 Watch Dog Thread

In the front-end codec driver, Grain Media implements a kernel thread to polling channel video signal status and video format from codec. If the vcap300 input\_module driver has registered a notify callback function, this thread will notify the vcap300 input\_mode driver when the video channel status or format changed. Users can modify the code to change the thread polling time to speed up the video port to change configuration. The module parameter, "notify", can be used to turn on/off notify to the vcap300 input\_module driver. The input\_module driver will not auto-monitor the video channel status and format if users turn off the notify function. So, it is suggested for users to turn on the notify function for dynamically switching the video interface setting according to the video port output format from codec.

## 12.3 Summary of Driver Porting

Based on the SDK architecture, the front-end codec driver header file is located in `/module/include/front_end`. Users can create the header file in this directory for new codec. The codec driver is located in `/module/front_end` directory. Grain Media classifies the D1 codec to the decoder directory, SDI codec to `sdi` directory, AHD codec to `ahd` directory, CVI codec to `hdcvi` directory, TVI codec to `hdtvi` directory. The front-end driver provided the platform control related API in `fe_common.ko`. The platform header file is in `/module/front_end/platform.h`. Users can refer to this file to know which platform control API can be available, such as the external clock output frequency, driving, and GPIO assignment of the codec hardware reset pin. The front-end codec driver must implement the below items:

- Module parameter interface, for adjusting the codec setting in different platforms
- Register read/write interface (I2C, SPI, and so on)
- Video initial procedure
- Audio initial procedure
- Codec control library
- Watchdog thread, for monitor video status and format and issue notify
- Video channel mapping table
- Device node for IOCTL and proc node

The `vcap300 input_module` header file is located in `/module/vcap300/vcap_input.h`. The purpose of the `input_module` driver is used to provide the video interface setting according to the format of the codec video output port. Below items must be implemented in the codec `input_module` driver:

- Module parameter interface, for adjusting the settings in different platforms (`vport`, `inv_clk`, and so on)
- Codec video status and video format notification callback function
- Initialize of input module device structure (`struct vcap_input_dev_t`) for video interface setting