1. Overview

Overview

This manual explains the driver module (this module) that controls the VIN on R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H.

1.1

Reference

Standard

1.2 The following table shows the standard that this module corresponds.

1.2.1 **Table 1.1 Standard of V4L2 API (R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H)**

Title	location
Linux Media Infrastructure userspace API	https://linuxtv.org/downloads/v4l-dvb-apis/

Related Document

1.2.2 The following table shows the document related to this module.

Table 1.2 Related documents (R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H)

Number	Issue	Title	Edition	Date
-	Renesas Electronics	R-Car Series, 3rd Generation User's Manual: Hardware	Rev.2.20	Jun. 30, 2020
-	Renesas Electronics	R-Car V3U Series User's Manual	Rev.0.5	Jul. 31, 2020
-	Renesas Electronics	R-CarH3-SiP System Evaluation Board Salvator-X RTP0RC7795SIPB0011S	Rev.1.09	May. 11, 2017
-	Renesas Electronics	R-CarM3-SiP System Evaluation Board Salvator-X RTP0RC7796SIPB0011S	Rev.0.04	Oct. 3, 2016
-	Renesas Electronics	R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board Salvator-XS Hardware Manual	Rev.2.04	Jul. 17, 2018
-	Renesas Electronics	R-CarE3 System Evaluation Board Ebisu Hardware Manual RTP0RC77990SEB0010S	Rev.0.03	Apr. 11, 2018
-	Renesas Electronics	R-CarE3 System Evaluation Board Ebisu-4D (E3 board 4xDRAM) Hardware Manual	Rev.1.01	Jul. 19, 2018
-	Renesas Electronics	R-CarV3U System Evaluation Board Falcon Hardware Manual	Rev.0.01	Sep. 11, 2020
-	Renesas Electronics	R-Car V3H_2 Additional Document for User's Manual: Hardware	Rev.0.50	Jul. 31, 2020
-	Renesas Electronics	R-CarV3H System Evaluation Board Condor-I Hardware Manual	Rev.0.02	Nov. 11, 2020

Table 1.3 Related documents (R-Car H3 / M3 / M3N / E3)

Issue	Title	Edition	Date
Analog Devices	ADV7482 Data Sheet	Rev.0	Jun. 2014
Analog Devices	ADV7481 Reference Manual UG-747	Rev.0	Dec. 2014
Analog Devices	ADV7481 Required Settings *1	Rev. v3.6	Oct. 24, 2014

*1 Please refer to http://www.analog.com/media/en/engineering-tools/design-tools/ADV7481ES3C-VER.3.6c.txt

Table 1.4 Related documents (R-Car D3)

Issue	Title	Edition	Date
Analog Devices	ADV7612 Data Sheet	Rev.E	Feb. 23, 2017
Analog Devices	ADV7612 User Guide UG-216	Rev.C	-
Analog Devices	ADV7180 Data Sheet	Rev.J	May. 09, 2017

Table 1.5 Related documents (R-Car V3U)

Please get the data sheet for the MAX96712 yourself.

Table 1.6 Related documents (R-Car V3H)

Please get the data sheet for the MAX9286 yourself

1.3 Restrictions

There are no restrictions.

1.4 Notice

- This module supports only the V4L2 APIs for capture. This module does not guarantee the undescribed V4L2 APIs in this document.
- The channel number of VIN that can operate simultaneously depends on the channel number of CSI2. In the Salvator-X/XS board, VIN can operate simultaneously up to 2 channels, so 2 channels of CSI40 and CSI20 are used.
- R-Car E3 has single 2 lane CSI2.
- CSI2 module name of R-Car E3 is CSI40 (CSI4LNK0), but the module is 2 lane CSI2.
- It is prohibited to simultaneously use the NV12 format with VIN0 and VIN1 by H/W specification. It also applies to VIN4 and VIN5.
- ISP module is not supported (this module control channel selector only).
- The supported camera device in this module is LI-AR0231-AP0200-GMSL2, other cameras devices is not supported.
- The VIN function of BPS and UDS are not supported in V3U/V3H by H/W specification.
- The supported camera for this module is ov10635 of OmniVision.

- The R-CarV3H incorporates two MIPI-CSI2 interfaces.
- R-Car D3 has no CSI2. Therefore, the functions of CSI2 described in this document cannot be used with R-Car D3.

2. Terminology

The following table shows the terminology related to this module.

Table 2.1 Terminology

Terms	Explanation
V4L2	Video For Linux Two
VIN	Video Input module
MIPI	Mobile Industry Processor Interface
CSI-2	Camera Serial Interface 2
CSI40	4 Lane of CSI-2 LINK 0
CSI41	4 Lane of CSI-2 LINK 1
CSI42	4 Lane of CSI-2 LINK 2
CSI43	4 Lane of CSI-2 LINK 3
CSI20	2 Lane of CSI-2 LINK 0
NTSC	National Television System Committee
STP	Shielded Twisted Pair
UDS	Up Down Scalar
BPS	Color Space Conversion **B**y**p**a**s**s Mode
VC	Virtual Channel
EMB	CSI2 Embedded of Virtual Channel 0

3. Operating Environment

Hardware Environment

The following table lists the hardware needed to use this module.

Table 3.1 Hardware environment (R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H)

3.1

Name	Version	Manufacturer
R-CarH3-SiP System Evaluation Board Salvator-X	-	Renesas Electronics
R-CarM3-SiP System Evaluation Board Salvator-X	-	Renesas Electronics
R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board Salvator-XS	-	Renesas Electronics
R-CarE3 System Evaluation Board Ebisu	-	Renesas Electronics
R-CarE3 System Evaluation Board Ebisu-4D	-	Renesas Electronics
R-CarV3U System Evaluation Board Falcon	-	Renesas Electronics
R-CarV3H System Evaluation Board Condor-I	-	Renesas Electronics
R-CarD3 System Evaluation Board Draak	-	Renesas Electronics

3.2 Module Configuration

The following figure shows the configuration of this module.

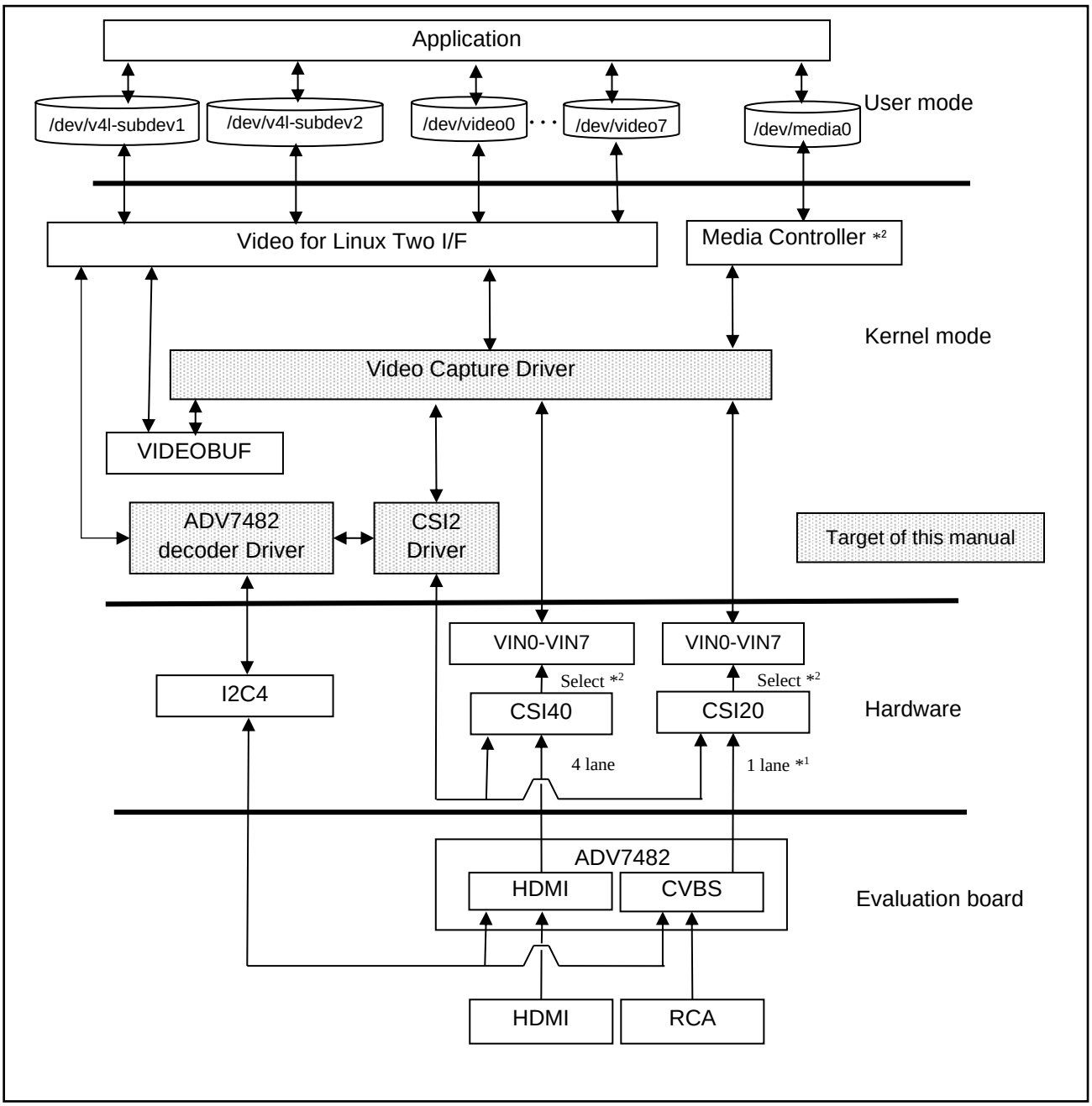


Figure 3.1 Module Configuration (R-Car H3 / M3 / M3N)

*1 CSI20 which has 2 lane is connected to 1 lane only by R-Car H3 / M3 / M3N evaluation board specification.

*2 The channel of the VIN can be selected from CSI2 driver by Media Controller API. Please refer to 4.1 Connected Device in detail and 5.2Media Controller API

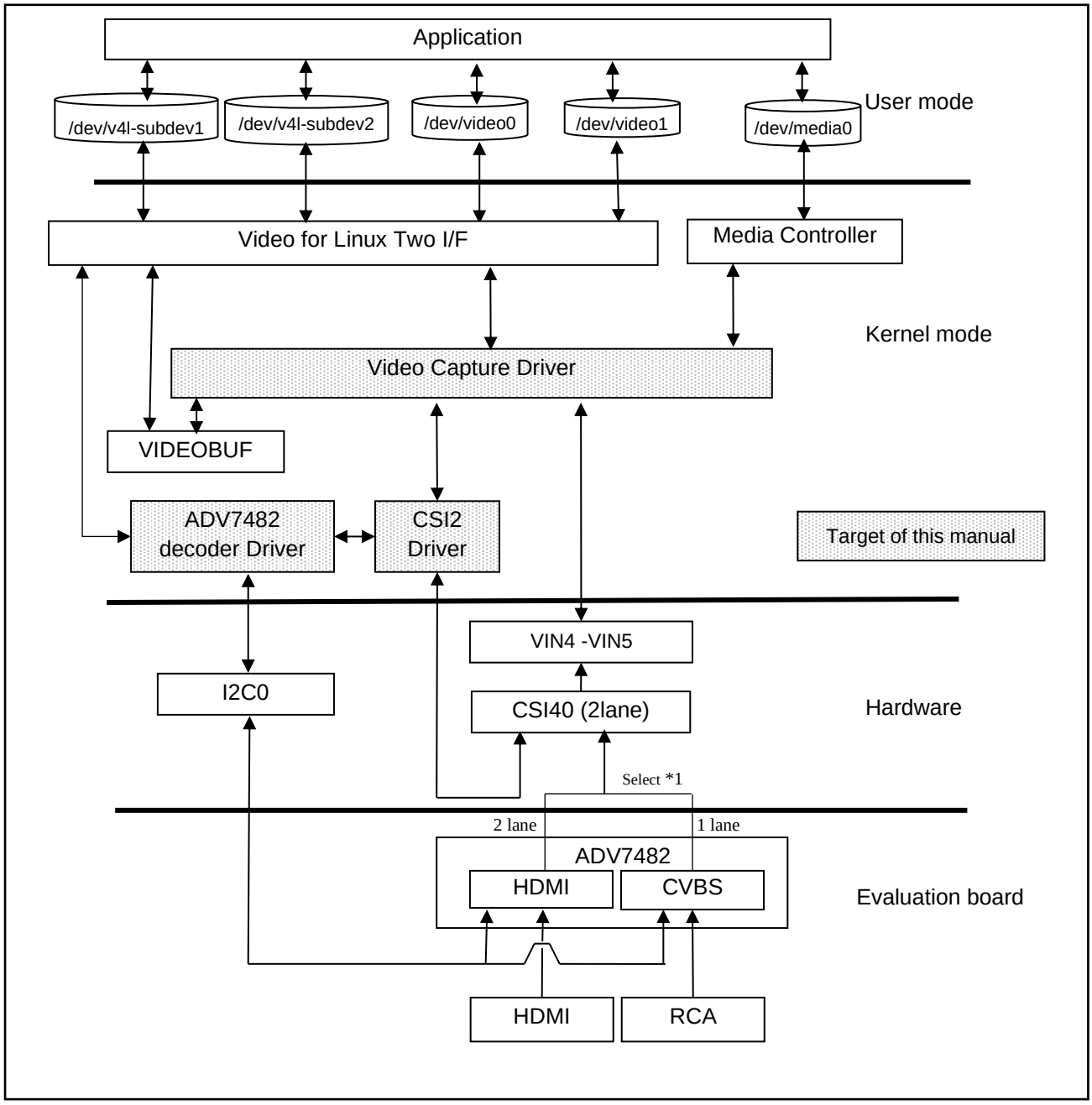


Figure 3.2 Module Configuration (R-Car E3)

*1 The input (HDMI / CVBS) of ADV7482 can be selected by device tree. Please refer to 6.3.1 Module Parameters.

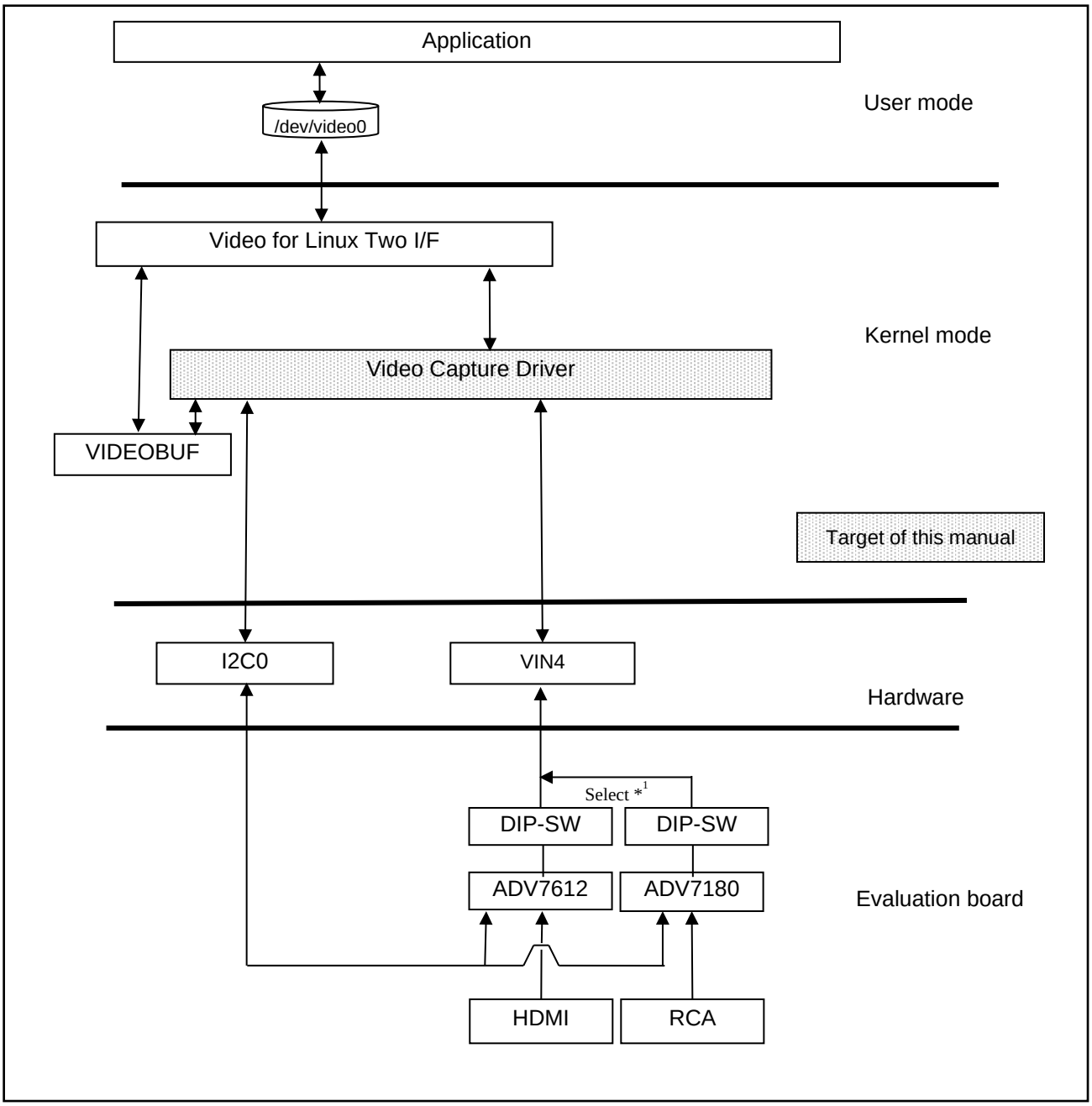


Figure 3.3 Module Configuration (R-Car D3)

*1 Input signal enables one side by the DIP-SW. please refer to [R-Car D3] in 6.3.1 Module Parameters in detail.

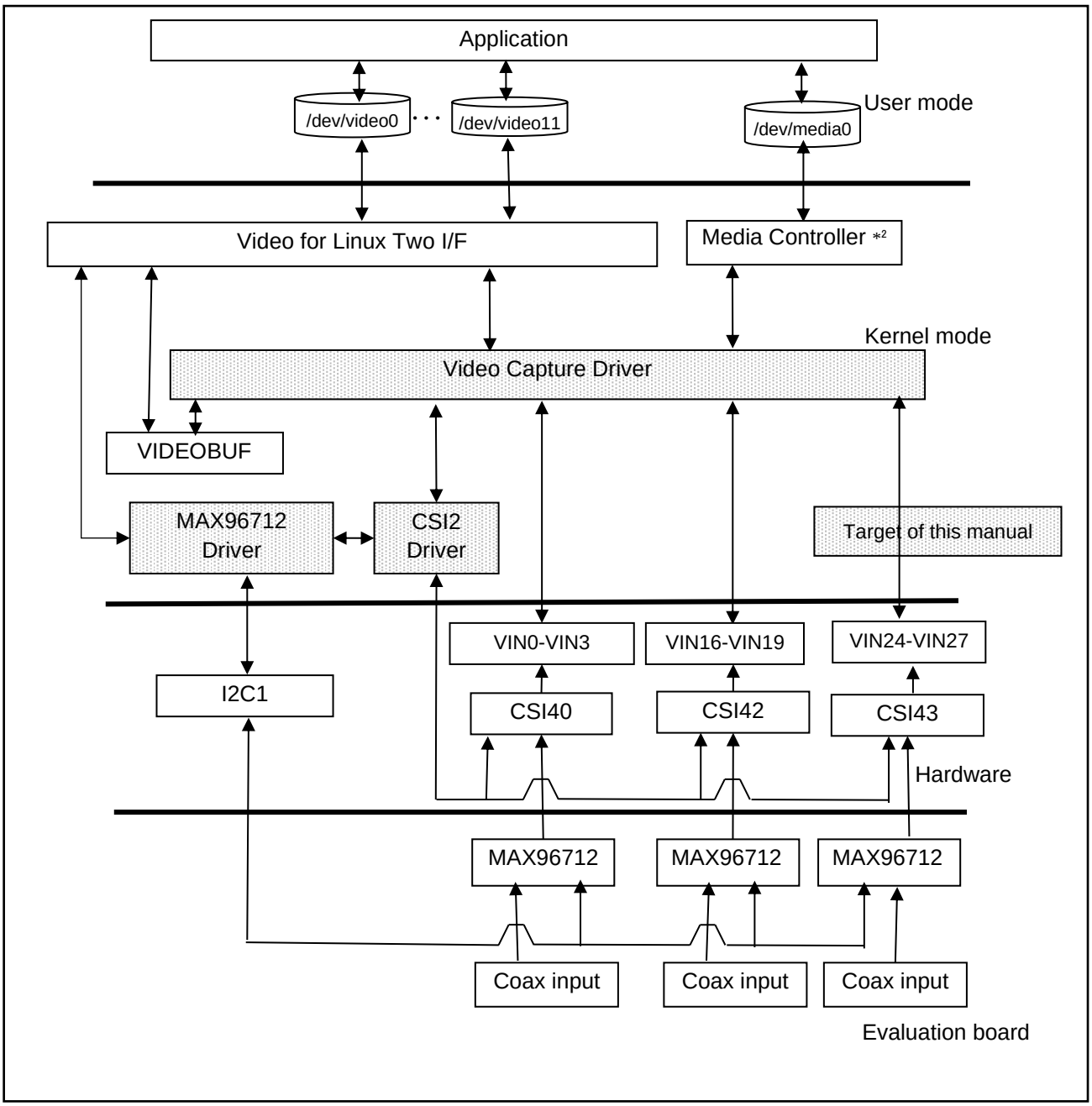


Figure 3.4 Module Configuration (R-Car V3U)

The following figure shows the configuration of this module

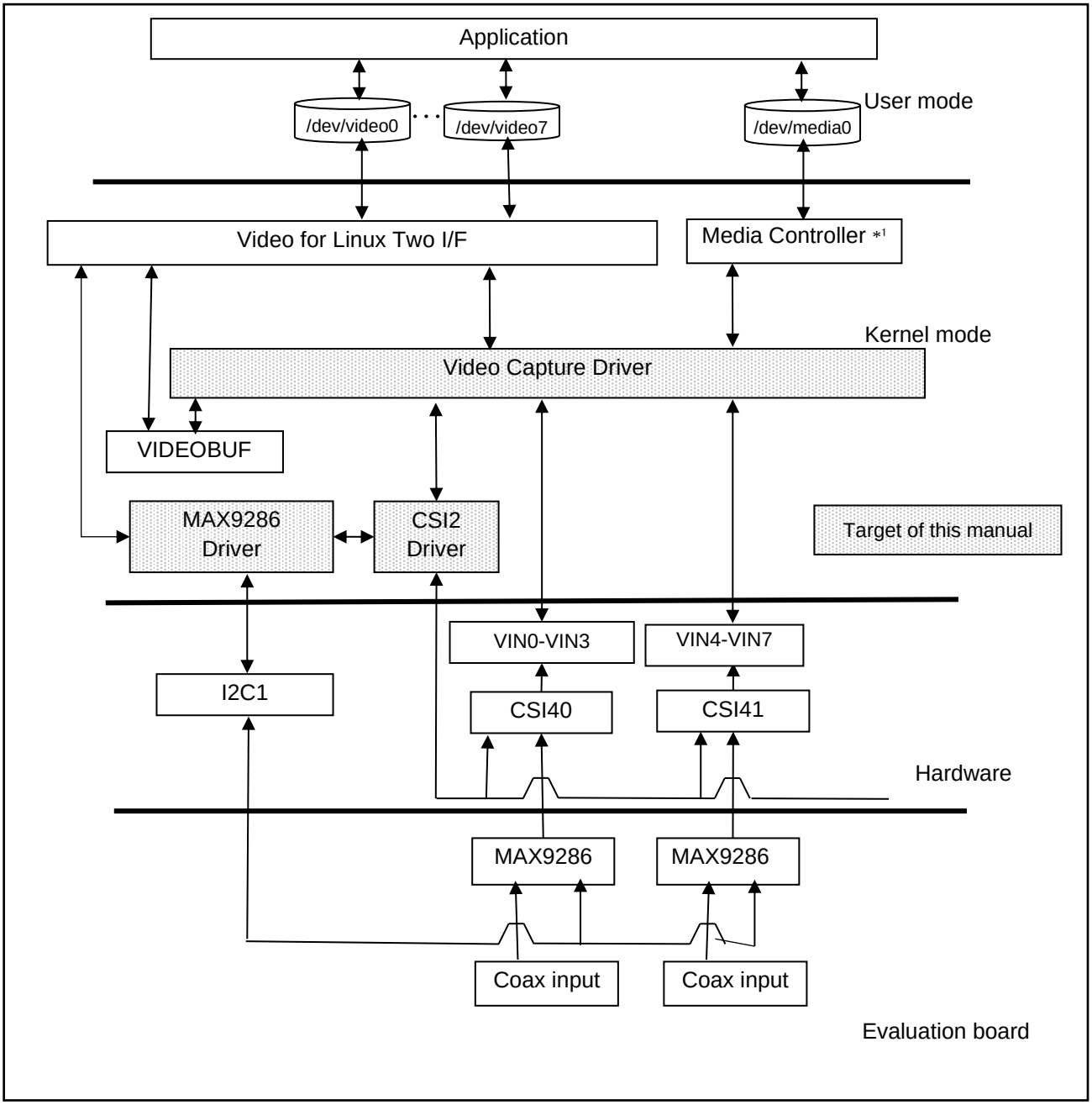


Figure 3.5 Module Configuration (R-Car V3H)

3.3

*1 The channel of the VIN can be selected from CSI2 driver by Media Controller API. Please refer to 4.1 Connected Device in detail and 5.2Media Controller API

State Transition Diagram

There is no state transition diagram for this module.

4. Function

This module controls the VIN and CSI2 on R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H and supports the video capture function. This module supports signals that the NTSC signal, the PAL signal and the HD digital signal that is decoded from the ADV7482 video decoder (R-Car H3 / M3 / M3N / E3).

This module supports serial output from the MAX96712 Deserializer(R-Car V3U).

This module supports serial output from MAX9286 Deserializer(R-Car V3H).

This module supports signals that the NTSC signal, the PAL signal (there are decoded from the ADV7180) and the HD digital signal (from the ADV7612). (R-Car D3)

The input data from the video decoder is transferred to the VIN through MIPI CSI-2 interface. Standard of ITU-R BT.601/BT.656/BT.709/BT.1358 is not supported in V3U.

Capture mode is decided by buffer number, and it becomes continuous frame capture mode by four or more buffer number, and it becomes single frame capture mode by three or less buffer number.

*Buffer number can be specified by 5.1.2 ioctl(VIDIOC_REQBUFS).

4.1 Connected Device

The following tables specify connector connected to Video Capture on the R-Car H3 / M3 / M3N evaluation board. CSI40 is used at CN20. CSI20 is used at CN21.

Table 4.1 Video Capture connection (R-Car H3)

Channel	Video Input Connector	Supporting Status	Scaling (UDS) support	Remark
VIN0	CN20 or CN21	Yes	Yes*1	HDMI / CVBS Receiver ADV7482
VIN1	CN20 or CN21	Yes	Yes*1	HDMI / CVBS Receiver ADV7482
VIN2	CN20 or CN21	Yes	No	HDMI / CVBS Receiver ADV7482
VIN3	CN20 or CN21	Yes	No	HDMI / CVBS Receiver ADV7482
VIN4	CN21	Yes	Yes*1	CVBS Receiver ADV7482
VIN5	CN21	Yes	Yes*1	CVBS Receiver ADV7482
VIN6	CN21	Yes	No	CVBS Receiver ADV7482
VIN7	CN21	Yes	No	CVBS Receiver ADV7482

Table 4.2 Video Capture connection (R-Car M3 / M3N)

Channel	Video Input Connector	Supporting Status	Scaling (UDS) support	Remark
VIN0	CN20 or CN21	Yes	Yes*1	HDMI / CVBS Receiver ADV7482
VIN1	CN20 or CN21	Yes	Yes*1	HDMI / CVBS Receiver ADV7482
VIN2	CN20 or CN21	Yes	No	HDMI / CVBS Receiver ADV7482
VIN3	CN20 or CN21	Yes	No	HDMI / CVBS Receiver ADV7482
VIN4	CN20 or CN21	Yes	Yes*1	HDMI / CVBS Receiver ADV7482
VIN5	CN20 or CN21	Yes	Yes*1	HDMI / CVBS Receiver ADV7482
VIN6	CN20 or CN21	Yes	No	HDMI / CVBS Receiver ADV7482

Channel	Video Input Connector	Supporting Status	Scaling (UDS) support	Remark
VIN7	CN20 or CN21	Yes	No	HDMI / CVBS Receiver ADV7482

Table 4.3 Video Capture connection (R-Car E3)

Channel	Video Input Connector	Supporting Status	Scaling (UDS) support	Remark
VIN4	CN20 or CN21	Yes	Yes*1	HDMI / CVBS Receiver ADV7482
VIN5	CN20 or CN21	Yes	Yes*1	HDMI / CVBS Receiver ADV7482

Note *1 The UDS module has two in VIN hardware. The first UDS is used in common by VIN0 and VIN1. The second UDS is used in common by VIN4 and VIN5. For use in common, it is prohibited that VIN0 and VIN1 is scaling at the same time. It is similar about VIN4 and VIN5.

Table 4.4 Video Capture connection (R-Car D3)

Channel	Video Input Connector	Supporting Status	Scaling (UDS) support	Remark
VIN4	CN42 or CN51	Yes	Yes	HDMI Receiver ADV7612 Video Processor ADV7180 Input signal is select by the DIP-SW.

Table 4.5 Video Capture connection (R-Car V3U)

Channel	Video Input Connector	Supporting Status on Falcon board	Scaling (UDS) support	Remark
VIN0	CN4	Yes	No	Max96712 Deserializer
VIN1	CN4	Yes	No	Max96712 Deserializer
VIN2	CN4	Yes	No	Max96712 Deserializer
VIN3	CN4	Yes	No	Max96712 Deserializer
VIN4	-	No	No	
VIN5	-	No	No	
VIN6	-	No	No	
VIN7	-	No	No	
VIN8	-	No	No	
VIN9	-	No	No	
VIN10	-	No	No	
VIN11	-	No	No	
VIN12	-	No	No	
VIN13	-	No	No	
VIN14	-	No	No	
VIN15	-	No	No	

Channel	Video Input Connector	Supporting Status on Falcon board	Scaling (UDS) support	Remark
VIN16	CN5	Yes	No	Max96712 Deserializer
VIN17	CN5	Yes	No	Max96712 Deserializer
VIN18	CN5	Yes	No	Max96712 Deserializer
VIN19	CN5	Yes	No	Max96712 Deserializer
VIN20	-	No	No	
VIN21	-	No	No	
VIN22	-	No	No	
VIN23	-	No	No	
VIN24	CN6	Yes	No	Max96712 Deserializer
VIN25	CN6	Yes	No	Max96712 Deserializer
VIN26	CN6	Yes	No	Max96712 Deserializer
VIN27	CN6	Yes	No	Max96712 Deserializer
VIN28	-	No	No	
VIN29	-	No	No	
VIN30	-	No	No	
VIN31	-	No	No	

Table 4.6 Video Capture connection (R-Car V3H)

Channel	Video Input Connector	Supporting Status	Scaling (UDS) support	Remark
VIN0	CN6	Yes	No	Max9286 Deserializer
VIN1	CN7	Yes	No	Max9286 Deserializer
VIN2	CN8	Yes	No	Max9286 Deserializer
VIN3	CN9	Yes	No	Max9286 Deserializer
VIN4	CN27	Yes	No	Max9286 Deserializer
VIN5	CN28	Yes	No	Max9286 Deserializer
VIN6	CN29	Yes	No	Max9286 Deserializer
VIN7	CN30	Yes	No	Max9286 Deserializer
VIN8	-	No	No	
VIN9	-	No	No	
VIN10	-	No	No	
VIN11	-	No	No	
VIN12	-	No	No	
VIN13	-	No	No	
VIN14	-	No	No	
VIN15	-	No	No	

The following table shows the connection table of the VIN, CSI2 and virtual channel. This module supports only the following connection by H/W specification. Please refer to 5.2 Media Controller API and 6.3.1 Module Parameters for the selection method.

About the combination of VIN3 from VIN0, please choose from No.5 to No.1. About the combination of VIN7 from VIN4, please choose from No.10 to No.6. If you select No.1, it determines connection of VIN0 (CSI40/VC0), VIN1 (CSI20/VC0), VIN2 (CSI21/VC0) and VIN3 (CSI40/VC1) automatically.

Make sure to set the VIN and CSI routing with media-ctl before executing capture.

Table 4.7 Connection of Video Capture and CSI2 (R-Car H3)

No.	VIN0	VIN1	VIN2	VIN3	CSI_CHSEL bit value
1	CSI40/VC0	CSI20/VC0	CSI20/VC1	CSI40/VC1	0
2	CSI20/VC0	CSI40/VC1	CSI40/VC0	CSI20/VC1	1
3	CSI40/VC1	CSI40/VC0	CSI20/VC0	CSI20/VC1	2
4	CSI40/VC0	CSI40/VC1	CSI40/VC2	CSI40/VC3	3
5	CSI20/VC0	CSI20/VC1	CSI20/VC2	CSI20/VC3	4
No.	VIN4	VIN5	VIN6	VIN7	CSI_CHSEL bit value
6	CSI41/VC0	CSI20/VC0	CSI20/VC1	CSI41/VC1	0
7	CSI20/VC0	CSI41/VC1	CSI41/VC0	CSI20/VC1	1
8	CSI41/VC1	CSI41/VC0	CSI20/VC0	CSI20/VC1	2
9	CSI41/VC0	CSI41/VC1	CSI41/VC2	CSI41/VC3	3
10	CSI20/VC0	CSI20/VC1	CSI20/VC2	CSI20/VC3	4

Table 4.8 Connection of Video Capture and CSI2 (R-Car M3)

No.	VIN0	VIN1	VIN2	VIN3	CSI_CHSEL bit value
1	CSI40/VC0	CSI20/VC0	-	CSI40/VC1	0
2	CSI20/VC0	-	CSI40/VC0	CSI20/VC1	1
3	-	CSI40/VC0	CSI20/VC0	-	2
4	CSI40/VC0	CSI40/VC1	CSI40/VC2	CSI40/VC3	3
5	CSI20/VC0	CSI20/VC1	CSI20/VC2	CSI20/VC3	4
No.	VIN4	VIN5	VIN6	VIN7	CSI_CHSEL bit value
6	CSI40/VC0	CSI20/VC0	-	CSI40/VC1	0
7	CSI20/VC0	-	CSI40/VC0	CSI20/VC1	1
8	-	CSI40/VC0	CSI20/VC0	-	2
9	CSI40/VC0	CSI40/VC1	CSI40/VC2	CSI40/VC3	3
10	CSI20/VC0	CSI20/VC1	CSI20/VC2	CSI20/VC3	4

Table 4.9 Connection of Video Capture and CSI2 (R-Car M3N)

No.	VIN0	VIN1	VIN2	VIN3	CSI_CHSEL bit value
1	CSI40/VC0	CSI20/VC0	CSI20/VC1	CSI40/VC1	0
2	CSI20/VC0	CSI40/VC1	CSI40/VC0	CSI20/VC1	1

No.	VIN0	VIN1	VIN2	VIN3	CSI_CHSEL bit value
1	CSI40/VC0	CSI20/VC0	CSI20/VC1	CSI40/VC1	0
3	CSI40/VC1	CSI40/VC0	CSI20/VC0	CSI20/VC1	2
4	CSI40/VC0	CSI40/VC1	CSI40/VC2	CSI40/VC3	3
5	CSI20/VC0	CSI20/VC1	CSI20/VC2	CSI20/VC3	4
No.	VIN4	VIN5	VIN6	VIN7	CSI_CHSEL bit value
6	CSI40/VC0	CSI20/VC0	CSI20/VC1	CSI40/VC1	0
7	CSI20/VC0	CSI40/VC1	CSI40/VC0	CSI20/VC1	1
8	CSI40/VC1	CSI40/VC0	CSI20/VC0	CSI20/VC1	2
9	CSI40/VC0	CSI40/VC1	CSI40/VC2	CSI40/VC3	3
10	CSI20/VC0	CSI20/VC1	CSI20/VC2	CSI20/VC3	4

Table 4.10 Connection of Video Capture and CSI2 (R-Car E3)

No.	VIN4	VIN5	CSI_CHSEL bit value
1	CSI40/VC0	-	0
2	-	CSI40/VC1	1
3	CSI40/VC1	CSI40/VC0	2
4	CSI40/VC0	CSI40/VC1	3

Table 4.11 Connection of Video Capture and CSI2 (R-Car V3U)

No.	VIN0	VIN1	VIN2	VIN3
1	CSI40/VC0	CSI40/VC1	CSI40/VC2	CSI40/VC3
No.	VIN16	VIN17	VIN18	VIN19
1	CSI42/VC0	CSI42/VC1	CSI42/VC2	CSI42/VC3
No.	VIN24	VIN25	VIN26	VIN27
1	CSI43/VC0	CSI43/VC1	CSI43/VC2	CSI43/VC3

Above is default setting statically on Falcon board (R-Car V3U).

4.2

Input / Output Format

The following table shows the Input/output format for this module.

Table 4.12 Input/output format (R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H)

Input format for VIN / CSI2		HDMI output formats from ADV7482	CVBS output formats from ADV7482	HDMI output formats from ADV7612	CVBS output formats from ADV7180	Output formats from MAX96712	Media bus pixel code that this module supports
Width of bits	Data format						
8bit	YCbCr422	No	Yes	No	Yes	No	MEDIA_BUS_FMT_UYVY8_2X8
10bit	RAW10	No	No	No	No	Yes	MEDIA_BUS_FMT_Y10_1X10 *1
24bit	RGB-888	Yes	No	Yes	No	No	MEDIA_BUS_FMT_RGB888_1X24
8bit	8-bit user defined data	No	No	No	No	No	-

*1 This is the MEDIA_BUS_FMT flag that is tentatively used instead of RAW10 input format in the V4L2 framework.

Table 4.13 Output format (R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H)

Output formats from VIN *4	Output formats for this module	Pixel format definition macro in V4L2
RGB565	Yes	V4L2_PIX_FMT_RGB565
YUYV	Yes	V4L2_PIX_FMT_YUYV
UYVY	Yes	V4L2_PIX_FMT_UYVY
ARGB1555	Yes	V4L2_PIX_FMT_ARGB555
RGB888 (32bits/pixel)	Yes	V4L2_PIX_FMT_XBGR32
ARGB8888	Yes	V4L2_PIX_FMT_ABGR32
NV16*1	Yes	V4L2_PIX_FMT_NV16
NV12*2 *3	Yes	V4L2_PIX_FMT_NV12
RAW10 *5	Yes	V4L2_PIX_FMT_Y10 *6

Notes: *1 At the time of NV16 format specification, the capture output width should be specified the value of the multiple of 32 by the specification of H/W. If it is not a multiple of 32, round it to a multiple of 32.

*2 At the time of NV12 format specification, the capture output width should be specified the value of the multiple of 32 by the specification of H/W. If it is not a multiple of 32, round it to a multiple of 32. the capture output height should be specified the vertical value of the input image size. Scaling is forbidden with NV12 format by the specification of H/W.

*3 Use of NV12 format is prohibited in VIN2, VIN3, VIN6 and VIN7 by the specification of H/W.

*4 This module is not supported other than output format format of above table.

*5 RAW10 is supported at V3U only.

*6 This is the V4L2_PIX_FMT flag that is tentatively used instead of RAW10 output format in the V4L2 framework..

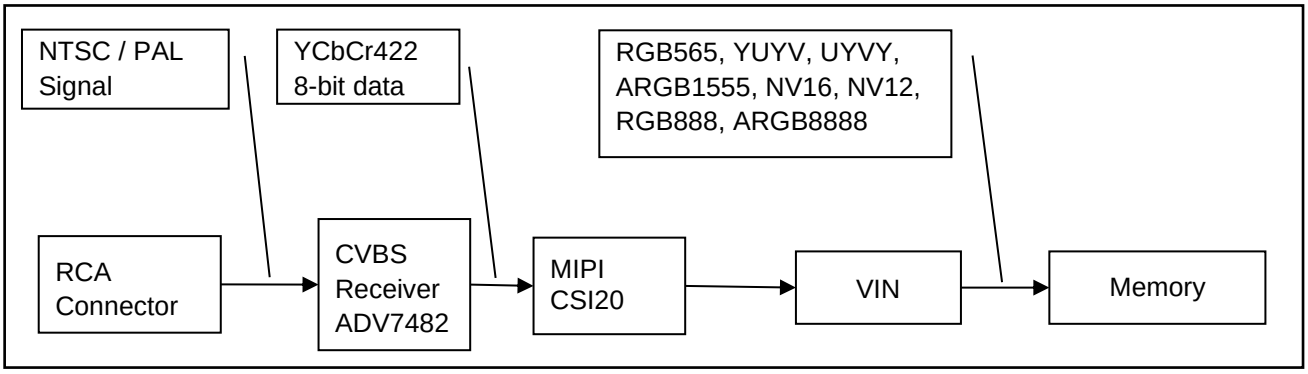


Figure 4.1 Flow of analog data (R-Car H3 / M3 / M3N and Salvator-X board)

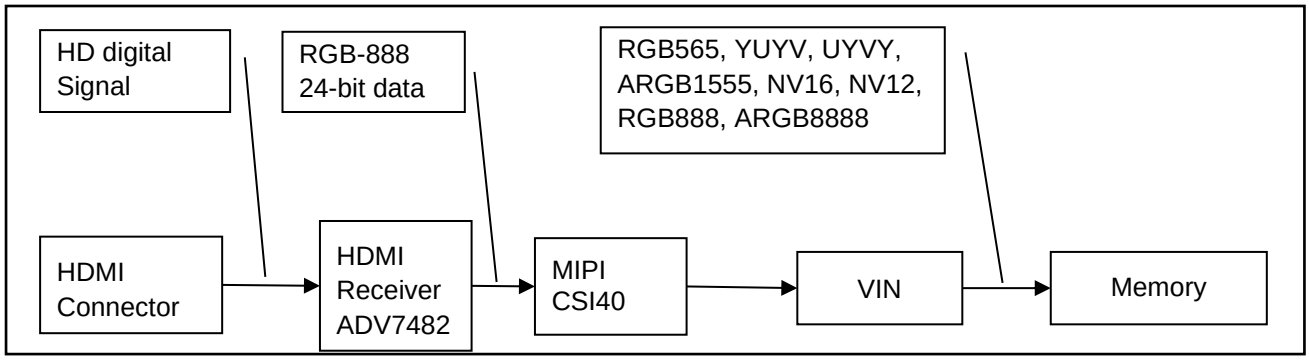


Figure 4.2 Flow of digital data (R-Car H3 / M3 / M3N and Salvator-X board)

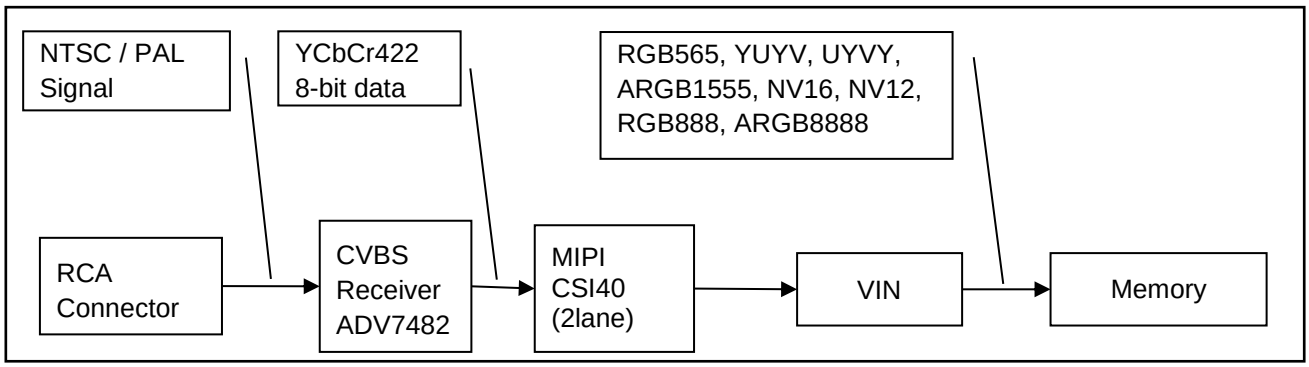


Figure 4.3 Flow of analog data (R-Car E3 and Ebisu board)

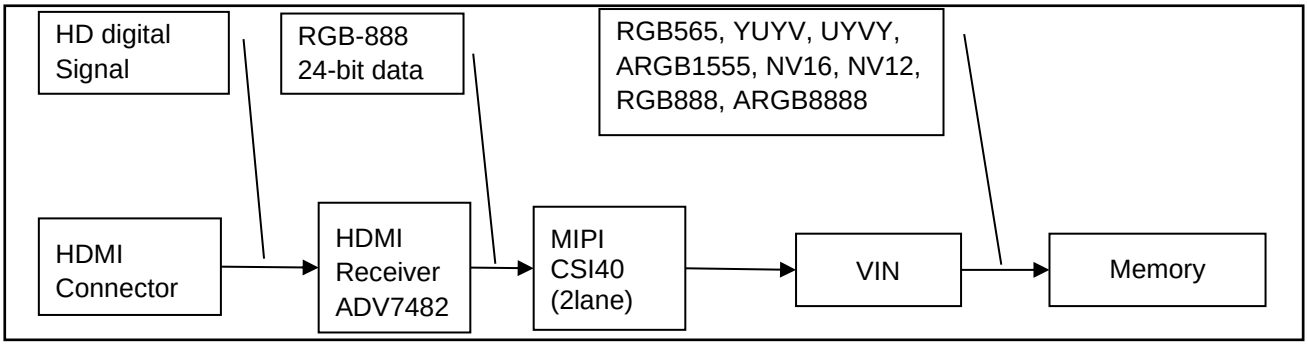


Figure 4.4 Flow of digital data (R-Car E3 and Ebisu board)

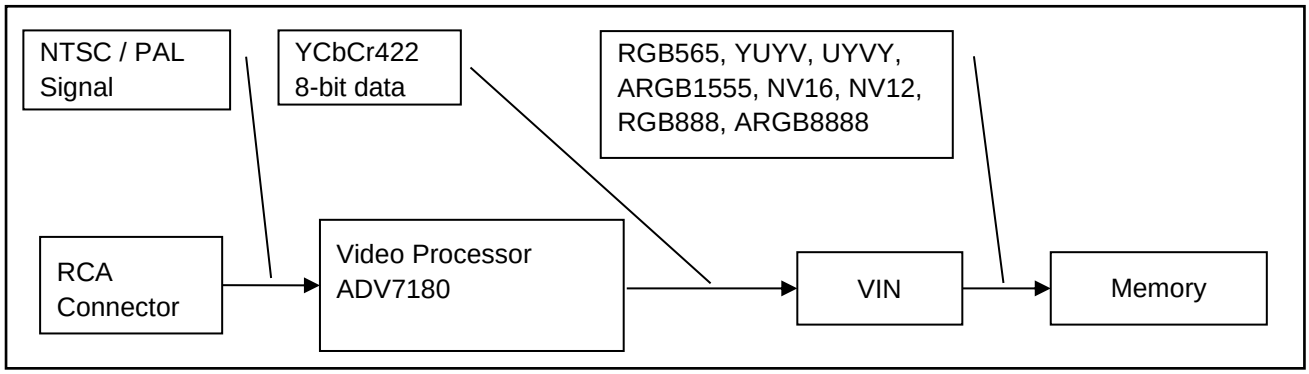


Figure 4.5 Flow of analog data (R-Car D3 and Draak board)

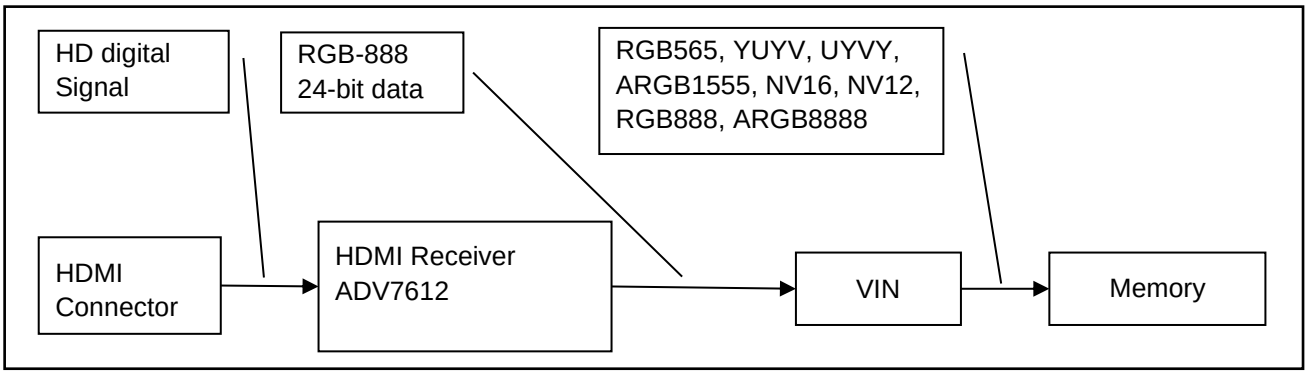


Figure 4.6 Flow of digital data (R-Car D3 and Draak board)

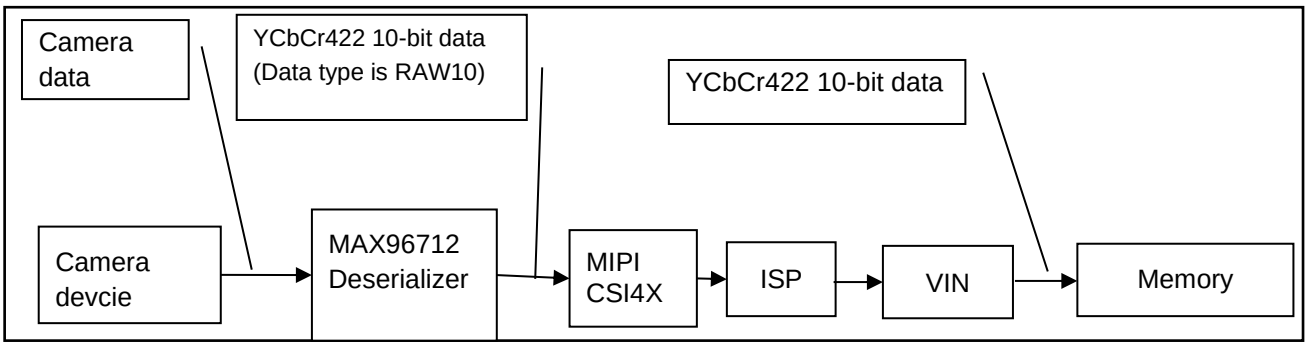


Figure 4.7 Flow of digital data (R-Car V3U and evaluation board)

This is the case when the Camera device is LI-AR0231-AP0200-GMSL2.

Notice: The datetype of LI-AR0231-AP0200-GMSL2 is RAW10, but the data inside is YCbCr422 10-bit data.

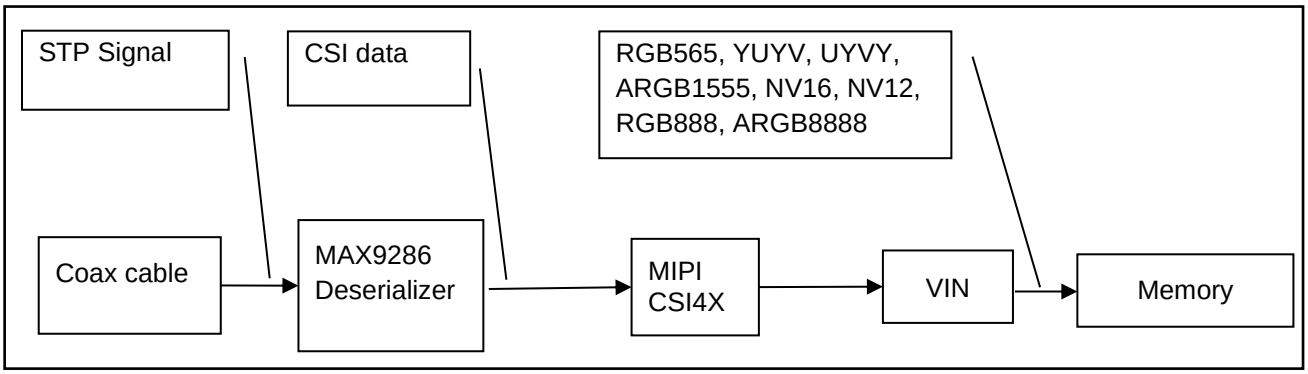


Figure 4.8 Flow of digital data (R-Car V3H and evaluation board)

Input Resolution

The following table shows the input resolution for this module.

Table 4.14 Input resolution [R-Car H3 / M3 / M3N / E3 / D3]

Input resolution for this module	HDMI connector	RCA connector
1920x1080p @ 60Hz	Yes*1	No
1920x1080i @ 60Hz	Yes	No
1280x720p @ 60Hz	Yes	No
720x480p @ 60Hz	Yes	No
640x480p @ 60Hz	Yes	No
720x576p @ 50Hz	Yes	No
720x480i @ 60Hz (NTSC signal)	No	Yes
720x576i @ 50Hz (PAL signal)	No	Yes

Notes: *1 1920x1080p@60Hz is not supported in R-Ca E3 / D3.

Table 4.15 Input resolution [R-Car V3U]

Input resolution for this module	Coax Cable
1920x1020p @ 30Hz*1	Yes

Notes: *1 Other resolutions may be able to captured, but the camera device on Falcon board can only evaluate and support 1920x1020 resolution. The refresh rate may decrease when operating multiple channels at the same time.

Table 4.16 Input resolution [R-Car V3H]

Input resolution for this module	Coax Cable
1280x1080p @ 30Hz	Yes*1

Notes: *1 Other resolutions can be captured, but the Condor I board can only evaluate and support 1280x1080 resolution.

Clipping and Scaling

This module supports clipping and scaling using the hardware function of the VIN. As shown in the Figure 4.9, the VIN input image is clipped first and then scaled to the size of output image. The scaling function by NV12 format is forbidden by the specification of H/W. The horizontal scaling should be specified the value of multiple of 16 (NV16 format should be specified the value of multiple of 32. If it is not a multiple of 32, round it to a multiple of 32.).

4.4

- VIN input image

The input resolution of the VIN input image (maximum 1920x1080) depends on the input device such as video decoder.

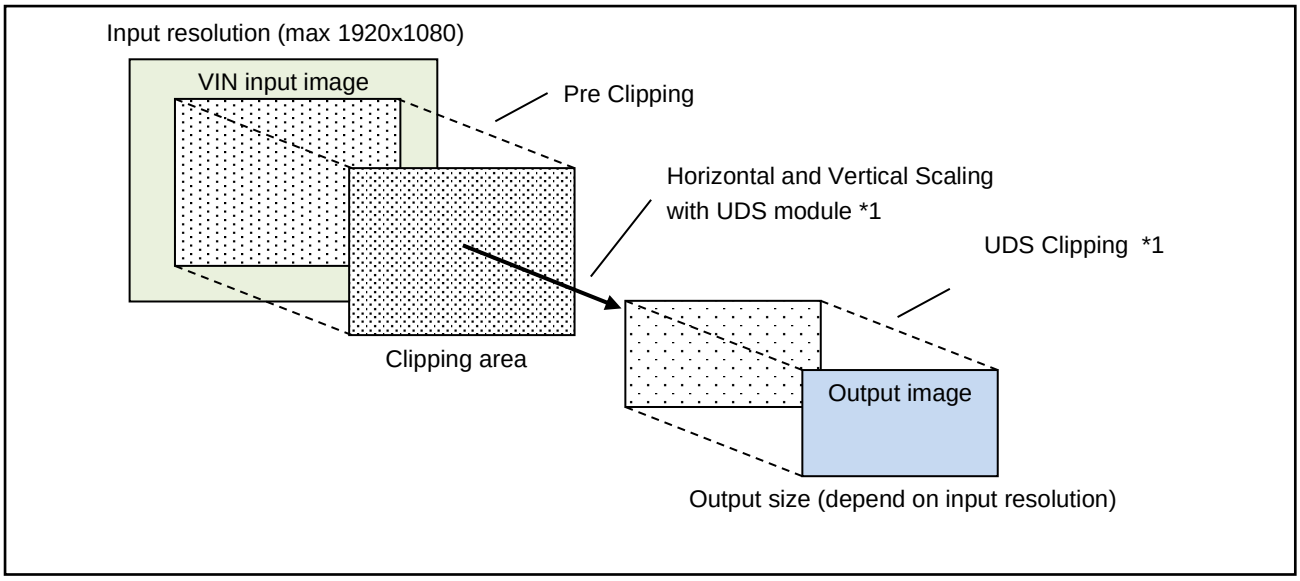


Figure 4.9 Clipping and Scaling function (R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H)

*1 UDS is not supported in V3U by H/W specification.

Scaling size is dependent on the input resolution. It indicates that information in the following.

Table 4.17 scaling size on the input resolution

Input resolution	Scale down minimum size of output image	Scale up maximum size of output image
1920x1080p @ 60Hz (cannot scaling up)	128x68p	1920x1080p
1920x1080i @ 60Hz (cannot scaling up)	128x68i	1920x1080i
1280x720p @ 60Hz	96x45p	1920x1080p
1280x800p @ 60Hz	96x50p	1920x1080p
720x480p @ 60Hz	64x30p	1440x1080p
640x480p @ 60Hz	64x30p	1280x1080p
720x576p @ 50Hz	64x36p	1440x1080p
720x480i @ 60Hz (NTSC signal)	64x30i	1440x1080i
720x576i @ 50Hz (PAL signal)	64x36i	1440x1080i

- Clipping area

  The clipping area is set with VIDIOC_S_CROP interface of V4L2. Relations of the input image and VIDIOC_S_CROP are described below.

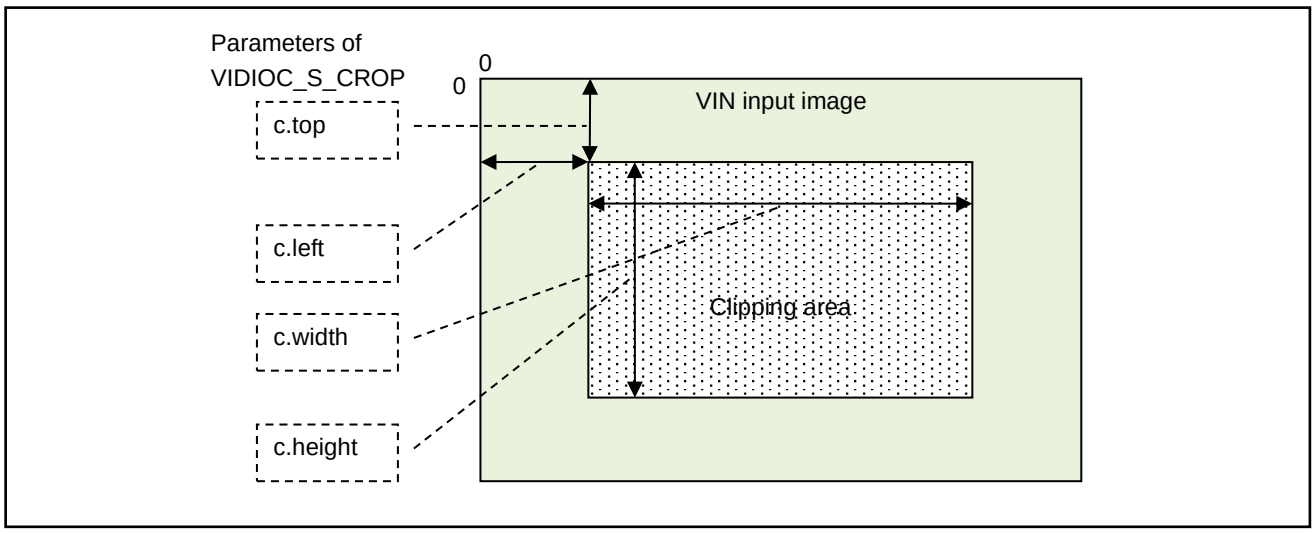


Figure 4.10 Relations of the VIN input image and VIDIOC_S_CROP (R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H)

- Output image

  It is the image stored in the capture buffer. The output image size is set by VIDIOC_S_FMT interface of V4L2. Relations of the output image and VIDIOC_S_FMT are described below.

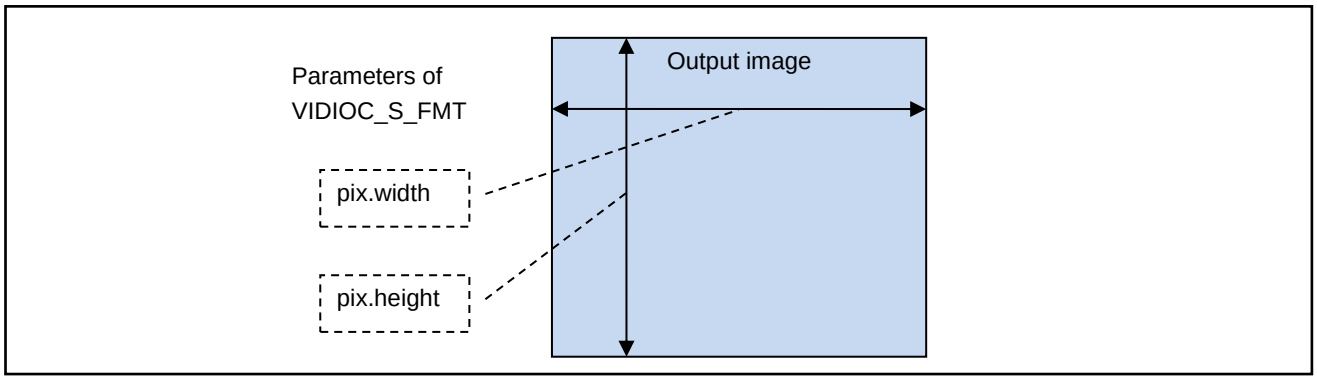


Figure 4.11 Relations of the VIN input image and VIDIOC_S_FMT (R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H)

At the time of scaling, Please refer to **Table 4.17** about the setting value of pix.width and pix.height.

This module performs scaling against the clipping area to fit the output image.

Notice: At the time of horizontal scaling, pix.width should be specified the value of multiple of 16 (NV16 format should be specified the value of multiple of 32. If it is not a multiple of 32, round it to a multiple of 32.). Scaling of NV12 format is prohibited by H/W specification. pix.height should be specified the vertical value of input image size. If the pix.height value is greater than the input size, the extra data will be captured. If the pix.height value is set smaller than the input size, the capture data will be distorted or clipped.

Hardware Parameters

This module supports VIDIOC_S_CTRL. Using this interface, the hardware control parameters of the video decoder (adv7482) can be set. The following tables show the hardware control parameters which can be set.

Table 4.18 Hardware CVBS input control of ADV7482 receiver (R-Car H3 / M3 / M3N / E3)

4.5

Item	V4L2 Command ID	Minimum	Default	Maximum
Contrast	V4L2_CID_CONTRAST	0	128	255
Brightness	V4L2_CID_BRIGHTNESS	-128	0	127
Hue	V4L2_CID_HUE	-127	0	128
Saturation	V4L2_CID_SATURATION	0	128	255

Table 4.19 Hardware HDMI input control of ADV7482 receiver (R-Car H3 / M3 / M3N / E3)

Item	V4L2 Command ID	Minimum	Default	Maximum
Contrast	V4L2_CID_CONTRAST	0	128	255
Brightness	V4L2_CID_BRIGHTNESS	-128	0	127
Hue	V4L2_CID_HUE	0	0	255
Saturation	V4L2_CID_SATURATION	0	128	255

Table 4.20 Hardware CVBS input control of ADV7180 video input processor (R-Car D3)

Item	V4L2 Command ID	Minimum	Default	Maximum
Contrast	V4L2_CID_CONTRAST	0	128	255
Brightness	V4L2_CID_BRIGHTNESS	-128	0	127
Hue	V4L2_CID_HUE	-127	0	128
Saturation	V4L2_CID_SATURATION	0	128	255

Table 4.21 Hardware HDMI input control of ADV7612 receiver (R-Car D3)

Item	V4L2 Command ID	Minimum	Default	Maximum
Contrast	V4L2_CID_CONTRAST	0	128	255
Brightness	V4L2_CID_BRIGHTNESS	-128	0	127
Hue	V4L2_CID_HUE	0	0	255
Saturation	V4L2_CID_SATURATION	0	128	255

Field order

This module supports interlaced image in addition to progressive image. The setting value shown in the Table 4.22. The setting value can be specified to use VIDIOC_S_FMT interface.

Table 4.22 Field order (R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H)

4.6

Setting Value	Content
V4L2_FIELD_NONE*1	Images are in progressive format, not interlaced. Output the image in 1 frame unit.
V4L2_FIELD_INTERLACED_TB*2	Images contain both fields, interleaved line by line, top field first. The top field is transmitted first. Top field is set odd field. (Full interlace capture mode)
V4L2_FIELD_INTERLACED_BT*2	Images contain both fields, interleaved line by line, top field first. The bottom field is transmitted first. Top field is set even field. (Full interlace capture mode)
V4L2_FIELD_TOP	Images consist of the top field only. (Odd-field capture mode)
V4L2_FIELD_BOTTOM	Images consist of the bottom field only. (Even-field capture mode)
V4L2_FIELD_SEQ_TB*3	Images contain both fields, the top field lines are stored first in memory, immediately followed by the bottom field lines. Fields are always stored in temporal order, the older one first in memory. Image sizes refer to the frame, not fields.
V4L2_FIELD_SEQ_BT*3	Images contain both fields, the bottom field lines are stored first in memory, immediately followed by the top field lines. Fields are always stored in temporal order, the older one first in memory. Image sizes refer to the frame, not fields.
V4L2_FIELD_INTERLACED*2	Capture with top field first or bottom field first depending on the input signal. (Full interlace capture mode)
V4L2_FIELD_ALTERNATE*3	The two fields of a frame are passed in separate buffers

Note : 1: This module prohibits to set the value of V4L2_FIELD_NONE in interlaced input.

2: H/W limitation on vertical scaling. For vertical scaling and full interlace composition, the capture lines are inverted in some cases depending on the scaling ratio because the scaling processing is applied before interlace composition in memory. Be sure to evaluate the image quality before practical application.
In addition, in full interlace composition mode (Full interlace capture mode), horizontal stripe noise (such as combing noise) is generated in composite images as fields based on different timelines are combined in memory due to the interlacing method.

3: This field is supported with single frame capture mode only.

Initialization Process

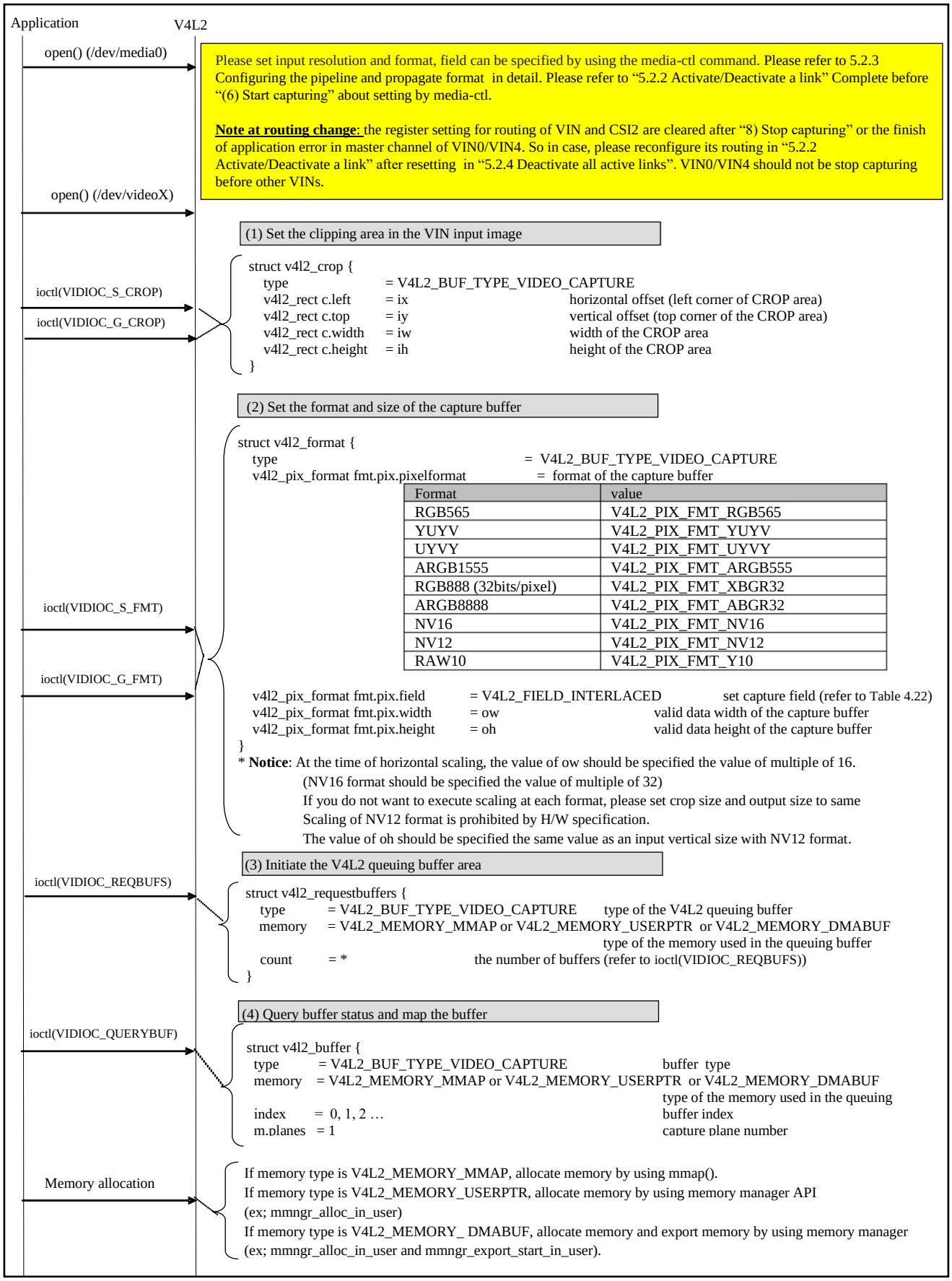
It is necessary to initialize this module, before starting video capturing.

1. Set the clipping area
   Set V4L2_BUF_TYPE_VIDEO_CAPTURE to the “type” and specify a part or all valid area of the VIN input image, in VIDIOC_S_CROP.
4.7 2. Set the format and size of the capture buffer
   Set V4L2_BUF_TYPE_VIDEO_CAPTURE to the “type” and specify the output image resolution, in VIDIOC_S_FMT.
3. Initiate the V4L2 queuing buffer area
   Set V4L2_BUF_TYPE_VIDEO_CAPTURE to the “type”, V4L2_MEMORY_MMAP or V4L2_MEMORY_DMABUF or V4L2_MEMORY_USERPTR to the “memory”, and the necessary buffer number to the “count”, in VIDIOC_REQBUFS.
4. Query buffer status and map the buffer
   Query the status of buffer using VIDIOC_QUERYBUF. And then, using the offset and length of the buffer, map the buffer into application address space with the system call mmap() or memory manager API.

Capture Process

4.8
This module captures video image in 1 frame unit. This module captures video image in 1 frame unit. The following figure shows the sequence of capturing video using this module.

Before using the V4L2 API by opening /dev/videoX, it is necessary to set the resolution and input format by using media-ctl. Please refer to 5.2 Media Controller API.


Application
V4L2
open() (/dev/media0)
Please set input resolution and format, field can be specified by using the media-ctl command. Please refer to 5.2.3 Configuring the pipeline and propagate format in detail. Please refer to "5.2.2 Activate/Deactivate a link" Complete before "(6) Start capturing" about setting by media-ctl.
Note at routing change: the register setting for routing of VIN and CSI2 are cleared after "8) Stop capturing" or the finish of application error in master channel of VIN0/VIN4. So in case, please reconfigure its routing in "5.2.2 Activate/Deactivate a link" after resetting in "5.2.4 Deactivate all active links". VIN0/VIN4 should not be stop capturing before other VINs.
open() (/dev/videoX)
(1) Set the clipping area in the VIN input image
ioctl(VIDIOC_S_CROP)
ioctl(VIDIOC_G_CROP)
struct v4l2_crop {
type = V4L2_BUF_TYPE_VIDEO_CAPTURE
v4l2_rect c.left = ix horizontal offset (left corner of CROP area)
v4l2_rect c.top = iy vertical offset (top corner of the CROP area)
v4l2_rect c.width = iw width of the CROP area
v4l2_rect c.height = ih height of the CROP area
}
(2) Set the format and size of the capture buffer
ioctl(VIDIOC_S_FMT)
ioctl(VIDIOC_G_FMT)
struct v4l2_format {
type = V4L2_BUF_TYPE_VIDEO_CAPTURE
v4l2_pix_format fmt.pix.pixelformat = format of the capture buffer
Format value
RGB565 V4L2_PIX_FMT_RGB565
YUYV V4L2_PIX_FMT_YUYV
UYVY V4L2_PIX_FMT_UYVY
ARGB1555 V4L2_PIX_FMT_ARGB555
RGB888 (32bits/pixel) V4L2_PIX_FMT_XBGR32
ARGB8888 V4L2_PIX_FMT_ABGR32
NV16 V4L2_PIX_FMT_NV16
NV12 V4L2_PIX_FMT_NV12
RAW10 V4L2_PIX_FMT_Y10
v4l2_pix_format fmt.pix.field = V4L2_FIELD_INTERLACED set capture field (refer to Table 4.22)
v4l2_pix_format fmt.pix.width = ow valid data width of the capture buffer
v4l2_pix_format fmt.pix.height = oh valid data height of the capture buffer
}
* Notice: At the time of horizontal scaling, the value of ow should be specified the value of multiple of 16.
(NV16 format should be specified the value of multiple of 32)
If you do not want to execute scaling at each format, please set crop size and output size to same
Scaling of NV12 format is prohibited by H/W specification.
The value of oh should be specified the same value as an input vertical size with NV12 format.
(3) Initiate the V4L2 queuing buffer area
ioctl(VIDIOC_REQBUFS)
struct v4l2_requestbuffers {
type = V4L2_BUF_TYPE_VIDEO_CAPTURE type of the V4L2 queuing buffer
memory = V4L2_MEMORY_MMAP or V4L2_MEMORY_USERPTR or V4L2_MEMORY_DMABUF
type of the memory used in the queuing buffer
count = * the number of buffers (refer to ioctl(VIDIOC_REQBUFS))
}
(4) Query buffer status and map the buffer
ioctl(VIDIOC_QUERYBUF)
struct v4l2_buffer {
type = V4L2_BUF_TYPE_VIDEO_CAPTURE buffer type
memory = V4L2_MEMORY_MMAP or V4L2_MEMORY_USERPTR or V4L2_MEMORY_DMABUF
type of the memory used in the queuing
index = 0, 1, 2 … buffer index
m.planes = 1 capture plane number
Memory allocation
If memory type is V4L2_MEMORY_MMAP, allocate memory by using mmap().
If memory type is V4L2_MEMORY_USERPTR, allocate memory by using memory manager API
(ex; mmngr_alloc_in_user)
If memory type is V4L2_MEMORY_ DMABUF, allocate memory and export memory by using memory manager
(ex; mmngr_alloc_in_user and mmngr_export_start_in_user).

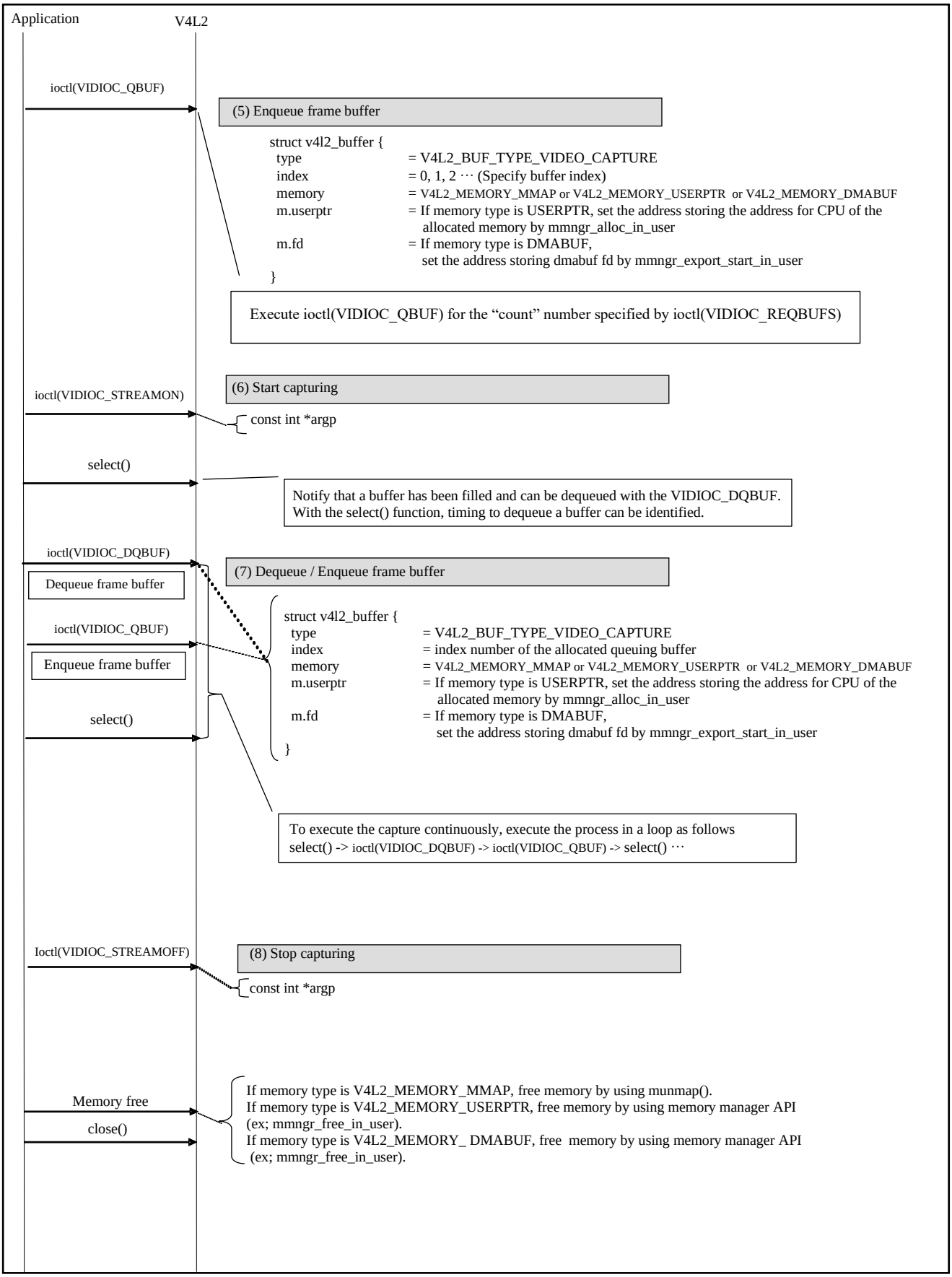


Figure 4.12 Initialization sequence of V4L2 (In case of /dev/videoX)

Capture Control (enqueue / dequeue buffer)

The buffer used to store the captured video image is enqueued by VIDIOC_QBUF from the application. The application needs to dequeue this buffer when it has been filled. VIDIOC_DQBUF will dequeue the buffer in turn, start from the oldest one. The application use system call select() to judge the capability to call VIDIOC_DQBUF.

Start or Stop Capturing

4.8.1 To start capturing, call the VIDIOC_STREAMON. To stop capturing, call the VIDIOC_STREAMOFF or close the device.

Pause capturing

4.8.2 To pause capturing, stop calling VIDIOC_QBUF and let the buffer queue of this module become empty. Do not call the VIDIOC_STREAMOFF or close the device.

4.8.3

Changing Output Image Size

4.8.4 Output image size cannot be changed during capture process. Please set the output size using VIDIOC_S_FMT in the initialization process.

Changing CROP of Captured Image

4.8.5 CROP cannot be changed during capture process. Please set the CROP using VIDIOC_S_CROP in the initialization process.

5. External Interface

The external interface of this module is based on Video for Linux Two API. Device node of this module is shown below.

Table 5.1 VIN device node (R-Car H3 / M3 / M3N / E3 / V3U / V3H)

Video decoder Input (VIN Channel supported)					Device node	Major number	Minor Number (H3/M3/M3N/E3/V3H)	Minor Number (V3U)	Remark
M3 / M3N	H3	E3	V3H	V3U					
HDMI / CVBS	HDMI / CVBS	HDMI / CVBS	Coax	Coax	/dev/video0	81	6(4)*1	18	UDS can be used. UDS shares with/dev/video0 and /dev/video1 *2
HDMI / CVBS	HDMI / CVBS	HDMI / CVBS	Coax	Coax	/dev/video1	81	7(5)*1	19	
HDMI / CVBS	HDMI / CVBS	-	Coax	Coax	/dev/video2	81	8	20	
HDMI / CVBS	HDMI / CVBS	-	Coax	Coax	/dev/video3	81	9	21	
HDMI / CVBS	CVBS	-	Coax	Coax	/dev/video4	81	10	22	UDS can be used. UDS shares with/dev/video4 and /dev/video5 *2
HDMI / CVBS	CVBS	-	Coax	Coax	/dev/video5	81	11	23	
HDMI / CVBS	CVBS	-	Coax	Coax	/dev/video6	81	12	24	
HDMI / CVBS	CVBS	-	Coax	Coax	/dev/video7	81	13	25	
-	-	-	Coax		/dev/video8	81		26	
-	-	-	Coax		/dev/video9	81		27	
-	-	-	Coax		/dev/video10	81		28	
-	-	-	Coax		/dev/video11	81		29	

*1 () is number at E3.

*2 V3U is not supported with UDS

Table 5.2 VIN device node (R-Car D3)

Video decoder Input (VIN Channel supported)	Device node	Major number	Minor number	Remark
HDMI / CVBS	/dev/video0	81	1	UDS can be used.

Table 5.3 media controller device node (R-Car H3 / M3 / M3N / E3)

Device node	Major number	Minor number
/dev/media0	250	0

Table 5.4 media controller device node (R-Car V3U)

Device node	Major number	Minor number
/dev/media0	250	0

Table 5.5 media controller device node (R-Car V3H)

Device node	Major number	Minor number
/dev/media0	250	0

Table 5.6 subdevice node (R-Car H3 / M3 / M3N / E3)

Device node	Major number	Minor number	Remark
/dev/v4l-subdev1	81	1	for controlling ADV7482(CVBS IN)
/dev/v4l-subdev2	81	2	for controlling ADV7482(HDMI IN)

Other subdevice node can be set via meida-ctl. list the necessary device file only for the application.

Video for Linux Two API

5.1 This module supports a part of Video for Linux Two API as follows.

Table 5.7 List of external interface (R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H)

Chapter	Function Name	Description	Remarks
5.1.1	ioctl(VIDIOC_QUERYCAP)	Query device capabilities.	#1
5.1.2	ioctl(VIDIOC_REQBUFS)	Buffer demand and transmission mode select.	#1
5.1.3	ioctl(VIDIOC_G_FMT)	Get a format.	#1
5.1.4	ioctl(VIDIOC_S_FMT)	Set scaling and the data format	#1
5.1.5	ioctl(VIDIOC_TRY_FMT)	Try a format.	#1
5.1.6	ioctl(VIDIOC_QUERYCTRL)	Enumerate controls and menu control items.	#3
5.1.7	ioctl(VIDIOC_G_CTRL)	Get the value of a decoder control.	#3
5.1.8	ioctl(VIDIOC_S_CTRL)	Set the value of a decoder control.	#3
5.1.9	ioctl(VIDIOC_CROPCAP)	Information about the video cropping and scaling abilities.	#1
5.1.10	ioctl(VIDIOC_G_CROP)	Get the current cropping rectangle.	#1
5.1.11	ioctl(VIDIOC_S_CROP)	Set the current cropping rectangle.	#1
5.1.12	ioctl(VIDIOC_QUERYBUF)	Query the status of a buffer.	#2
5.1.13	ioctl(VIDIOC_DQBUF)	Dequeue an empty buffer.	#2
5.1.14	ioctl(VIDIOC_QBUF)	Enqueue a filled buffer.	#2
5.1.15	ioctl(VIDIOC_STREAMON)	Start streaming I/O.	#2
5.1.16	ioctl(VIDIOC_STREAMOFF)	Stop streaming I/O.	#2

Notes: #1 This module provides these ioctls.
#2 VIDEOBUF provides these ioctls.
#3 These ioctls is provided by subdevice (/dev/v4l-subdev1 or /dev/v4l-subdev2).

ioctl(VIDIOC_QUERYCAP)

5.1.1

[Function]	int ioctl(int fd, int request, struct v4l2_capability *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_QUERYCAP
	argp	: Pointer of v4l2_capability
[Returns]	0	: Success
	-1	: Error
[Error number]	-EINVAL	: The device is not compatible with this specification.
[Description]	Query device capabilities.	

ioctl(VIDIOC_REQBUFS)

5.1.2

[Function]	int ioctl(int fd, int request, struct v4l2_requestbuffers *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_REQBUFS
	argp	: Pointer of v4l2_requestbuffers
[Returns]	0	: Success
	-1	: Error
[Error number]	-EBUSY	: The driver supports multiple opens and I/O is already in progress, or reallocation of buffers was attempted although one or more are still mapped.
	-EINVAL	: The buffer type (type field) or the requested I/O method (memory) is not supported.
[Description]	The buffer structure is allocated according to the number of the specified buffer.	
	Moreover, Transmission mode is chosen by the number of a buffer.	
	Transmission mode is chosen by the member "count" of a v4l2_requestbuffers structure.	
	The count value of three or less: Single frame capture mode	
	(Single transfer is up to max 15 fps by H/W specification, even if it is executed continuously.)	
	The count value of four or more: Continuous frame capture mode	
	The count value of zero: free all buffers	

5.1.3

ioctl(VIDIOC_G_FMT)

[Function]	int ioctl(int fd, int request, struct v4l2_format *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_G_FMT
	argp	: Pointer of v4l2_format
[Returns]	0	: Success
	-1	: Error
[Error number]	-EINVAL	: The struct v4l2_format type field is invalid or the requested buffer type not supported.
[Description]	Get a format.	

ioctl(VIDIOC_S_FMT)

[Function]	int ioctl(int fd, int request, struct v4l2_format *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_S_FMT
	argp	: Pointer of v4l2_format
[Returns]	0	: Success
	-1	: Error
[Error number]	-EBUSY	: The data format cannot be changed at this time, for example because I/O is already in progress.
	-EINVAL	: The struct v4l2_format type field is invalid or the requested buffer type not supported.
[Description]	Set the data format.	
[Notice]	At the time of horizontal scaling, the value of width in v4l2_pix_format structure of v4l2_format should be specified the value of multiple of 16 (NV16 format should be specified the value of multiple of 32. If it is not a multiple of 32, round it to a multiple of 32.). Scaling of NV12 format is prohibited by H/W specification. At the time of NV12 format specification, the value of height in v4l2_pix_format structure of v4l2_format should be specified the same value as an input vertical size.	

5.1.4

ioctl(VIDIOC_TRY_FMT)

5.1.5

[Function]	int ioctl(int fd, int request, struct v4l2_format *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_TRY_FMT
	argp	: Pointer of v4l2_format
[Returns]	0	: Success
	-1	: Error
[Error number]	-EINVAL	: The setting value of struct v4l2_format is invalid.
[Description]	Try a format.	

5.1.6

ioctl(VIDIOC_QUERYCTRL)

[Function]	int ioctl(int fd, int request, struct v4l2_queryctrl *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_QUERYCTRL
	argp	: Pointer of v4l2_queryctrl
[Returns]	0	: Success
	-1	: Error
[Error number]	-EINVAL	: The struct v4l2_queryctrl id is invalid. The struct v4l2_querymenu id or index is invalid.
	-EACCES	: An attempt was made to read a write-only control.
[Description]	Enumerate controls.	

ioctl(VIDIOC_G_CTRL)

5.1.7

[Function]	int ioctl(int fd, int request, struct v4l2_control *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_G_CTRL
	argp	: Pointer of v4l2_control
[Returns]	0	: Success
	-1	: Error
[Error number]	-EINVAL	: The struct v4l2_control id is invalid.
	-ERANGE	: The struct v4l2_control value is out of bounds.
	-EBUSY	: The control is temporarily not changeable, possibly because another applications took over control of the device function this control belongs to.
	-EACCES	: Attempt to get a write-only control.
[Description]	Get the value of a video decoder control. Please refer to Table 4.18 and Table 4.19 about id of v4l2_control structure member.	

ioctl(VIDIOC_S_CTRL)

5.1.8

[Function]	int ioctl(int fd, int request, struct v4l2_control *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_S_CTRL
	argp	: Pointer of v4l2_control
[Returns]	0	: Success
	-1	: Error
[Error number]	-EINVAL	: The struct v4l2_control id is invalid.
	-ERANGE	: The struct v4l2_control value is out of bounds.
	-EBUSY	: The control is temporarily not changeable, possibly because another applications took over control of the device function this control belongs to.
	-EACCES	: Attempt to set a read-only control.
[Description]	Set the value of a video decoder control. Please refer to Table 4.18 and Table 4.19 about id of v4l2_control structure member.	

5.1.9

ioctl(VIDIOC_CROPCAP)

[Function]	int ioctl(int fd, int request, struct v4l2_cropcap *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_CROPCAP
	argp	: Pointer of v4l2_cropcap
[Returns]	0	: Success
	-1	: Error
[Error number]	-EINVAL	: v4l2_cropcap type is invalid.
[Description]	Information about the video cropping and scaling abilities	

ioctl(VIDIOC_G_CROP)

[Function]	int ioctl(int fd, int request, struct v4l2_crop *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_G_CROP
	argp	: Pointer of v4l2_crop
[Returns]	0	: Success
	-1	: Error
[Error number]	-EINVAL	: Cropping is not supported.
[Description]	Get the current cropping rectangle.	

5.1.10

ioctl(VIDIOC_S_CROP)

[Function]	int ioctl(int fd, int request, struct v4l2_crop *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_S_CROP
	argp	: Pointer of v4l2_crop
[Returns]	0	: Success
	-1	: Error
[Error number]	-EINVAL	: Cropping is not supported.
[Description]	Set the current cropping rectangle.	

5.1.11

ioctl(VIDIOC_QUERYBUF)

[Function]	int ioctl(int fd, int request, struct v4l2_buffer *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_QUERYBUF
	argp	: Pointer of v4l2_buffer
[Returns]	0	: Success
	-1	: Error
[Error number]	-EINVAL	: The specified buffer type is not supported.
[Description]	Query the status of a buffer.	

5.1.12

ioctl(VIDIOC_DQBUF)

[Function]	int ioctl(int fd, int request, struct v4l2_buffer *argp)	
[Arguments]	fd	: File descriptors
	request	: VIDIOC_DQBUF
	argp	: Pointer of v4l2_buffer
[Returns]	0	: Success
	-1	: Error
[Error number]	-EAGAIN	: Non-blocking I/O is selected.
	-EINVAL	: The specified buffer type is not supported.
	-EIO	: VIDIOC_DQBUF failed due to an internal error.
[Description]	Dequeue a filled buffer from the driver's outgoing queue	

5.1.13

ioctl(VIDIOC_QBUF)

[Function] int ioctl(int fd, int request, struct v4l2_buffer *argp)
[Arguments] fd : File descriptors
request : VIDIOC_QBUF
argp : Pointer of v4l2_buffer
5.1.14 [Returns] 0 : Success
-1 : Error
[Error number] -EINVAL : The specified buffer type is not supported.
[Description] Enqueue an empty buffer in the driver's incoming queue.

ioctl(VIDIOC_STREAMON)

[Function] int ioctl(int fd, int request, const int *argp)
5.1.15 [Arguments] fd : File descriptors
request : VIDIOC_STREAMON
argp : Pointer of const int area for v4l2_buftype
[Returns] 0 : Success
-1 : Error
[Error number] -EINVAL : The buffer type is not supported, or no buffers have been allocated (memory mapping) or enqueued (output) yet.
-EPIPE : the pipeline configuration is invalid.
[Description] Start streaming I/O.

5.1.16

ioctl(VIDIOC_STREAMOFF)

[Function] int ioctl(int fd, int request, const int *argp)
[Arguments] fd : File descriptors
request : VIDIOC_STREAMOFF
argp : Pointer of const int area for v4l2_buftype
[Returns] 0 : Success
-1 : Error
[Error number] -EINVAL : The buffer type is not supported, or no buffers have been allocated (memory mapping) or enqueued (output) yet.
[Description] Stop streaming I/O and stop H/W operation.

Media Controller API

This ability not only work with a local digital subdevice directly attached to a VIN instance in a 1:1 mapping but to be part of a CSI-2 group which share a set of video decoders and CSI-2.

In this mode of operation each video decoder source is connected to a CSI-2 which in turn can be routed to the different VIN instances depending on one of the predetermined routing setups for that particular SoC, it's not possible to go outside the routing tables provided by the hardware.

Show current routing

Examine the current routing setup with 'media-ctl -d /dev/mediaX -p'.

media-ctl -d /dev/media0 -p

Example)

```
Media controller API version 4.14.70

Media device information
------------------------
driver          rcar_vin
model           renesas,vin-r8a7795
serial
bus info        platform:e6ef0000.video
hw revision     0x0
driver version  4.14.70

Device topology
- entity 1: rcar_csi2 feaa0000.csi2 (5 pads, 9 links)
            type V4L2 subdev subtype Unknown flags 0
            device node name /dev/v4l-subdev0
        pad0: Sink
                [fmt:RGB888_1X24/720x480 field:none]
                <- "adv748x 4-0070 txa":1 [ENABLED,IMMUTABLE]
        pad1: Source
                [fmt:RGB888_1X24/720x480 field:none]
                -> "VIN0 output":0 [ENABLED]
                -> "VIN1 output":0 []
                -> "VIN2 output":0 []
        pad2: Source
                [fmt:RGB888_1X24/720x480 field:none]
                -> "VIN0 output":0 []
                -> "VIN1 output":0 []
                -> "VIN3 output":0 [ENABLED]
        pad3: Source
                [fmt:RGB888_1X24/720x480 field:none]
                -> "VIN2 output":0 []
        pad4: Source
                [fmt:RGB888_1X24/720x480 field:none]
                -> "VIN3 output":0 []

~~~~~~~~ omission ~~~~~~

- entity 80: VIN7 output (1 pad, 2 links)
             type Node subtype V4L flags 0
             device node name /dev/video7
        pad0: Sink
                <- "rcar_csi2 fea80000.csi2":2 []
                <- "rcar_csi2 fea80000.csi2":4 []
```

Activate/Deactivate a link

5.2.2

The Media Controller framework allows user-space to enable/disable a link and that way control the routing of video data from a CSI-2 to a VIN instance. Not all CSI-2 and Virtual Channel are possible to route to any and all VIN instances, this is limited by the SoC hardware and is specific for each SoC version. Look at the previous section on how to show which links are possible on your particular SoC. Furthermore not all links can be activated independent of other links, this is also a limitation set by the hardware. If you try to enable a link which would not be possible with regard to other already active links the operation will fail with a EBUSY error. Try deactivating some other link to create more routing possibilities and try again, all possible routing setups for a specific SoC are documented in the datasheet.

A link is always configured from a CSI-2 instance to a VIN instance, the same way the video data is flowing. To enable a link use the media-ctl utility from v4l-utils package:

To enable a link use the media-ctl utility from v4l-utils package: (VIN0 is set CSI20/VC0)

```
# media-ctl -d /dev/media0 -l "'rcar_csi2 fea80000.csi2':1 -> 'VIN0 output':0 [1]"
```

```
# media-ctl -d /dev/media0 -l "'rcar_csi2 fea80000.csi2':1 -> 'VIN0 output':0 [1]"
```

Select CSI2 device to set VIN device	VC number 1~4: (VC0~VC3)	Select VIN device	Set Connection [1]: Enable [0]: Disable

To disable the same link use:

```
# media-ctl -d /dev/media0 -l "'rcar_csi2 fea80000.csi2':1 -> 'VIN0 output':0 [0]"
```

Once a link is enabled you can access the /dev/videoX node associated with the VIN instance to start capturing video data. Using the link from the example above you would be able to access video data from CSI20/VC0 on the VIN0 instance.

[Example setting in R-Car H3/M3/M3N]

How to capture HDMI in HDMI connector by setting CSI_CHSEL[2:0]= 0 about VIN 0.

```
# media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':1 -> 'VIN0 output':0 [1]"
```

How to capture NTSC in RCA connector by setting CSI_CHSEL[2:0]= 0 about VIN 5.

```
# media-ctl -d /dev/media0 -l "'rcar_csi2 fea80000.csi2':1 -> 'VIN5 output':0 [1]"
```

[Example setting in R-Car M3/M3N]

How to capture NTSC in RCA connector by setting CSI_CHSEL[2:0]= 1 about VIN 4.

```
# media-ctl -d /dev/media0 -l "'rcar_csi2 fea80000.csi2':1 -> 'VIN4 output':0 [1]"
```

[Example setting in R-Car E3]

How to capture in HDMI connector by setting CSI_CHSEL[2:0] = 2 about VIN 5.

```
# media-ctl -r /dev/media0

# media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':1 -> 'VIN5 output':0 [1]"

# media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':2 -> 'VIN4 output':0 [1]"
```

Configuring the pipeline and propagate format

Once the user has configured a pipeline using 'media-ctl' as described 5.2.2 the format needs to be propagated in the pipeline before streaming can start (The capture cannot be performed unless it is set). The following shows an example of the execution command.

5.2.3 [HDMI IN]

[H3/M3/M3N/E3]

Please change command description in case of R-Car E3

adv748x 4-0070 -> adv748x 0-0070, VIN0 -> VIN4

- 1920x1080p (This resolution is not support in R-Car E3)

  # media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':1 -> 'VIN0 output':0 [1]"

  # media-ctl -d /dev/media0 -l "'adv748x 4-0070 hdmi':1 -> 'adv748x 4-0070 txa':0 [1]"

  # media-ctl -d /dev/media0 -V "'rcar_csi2 feaa0000.csi2':1 [fmt:RGB888_1X24/1920x1080 field:none]"

  # media-ctl -d /dev/media0 -V "'adv748x 4-0070 txa':0 [fmt:RGB888_1X24/1920x1080 field:none]"

- 1920x1080i

  # media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':1 -> 'VIN0 output':0 [1]"

  # media-ctl -d /dev/media0 -l "'adv748x 4-0070 hdmi':1 -> 'adv748x 4-0070 txa':0 [1]"

  # media-ctl -d /dev/media0 -V "'rcar_csi2 feaa0000.csi2':1 [fmt:RGB888_1X24/1920x540 field:alternate]"

  # media-ctl -d /dev/media0 -V "'adv748x 4-0070 txa':0 [fmt:RGB888_1X24/1920x540 field:alternate]"

- 1280x720p

  # media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':1 -> 'VIN0 output':0 [1]"

  # media-ctl -d /dev/media0 -l "'adv748x 4-0070 hdmi':1 -> 'adv748x 4-0070 txa':0 [1]"

  # media-ctl -d /dev/media0 -V "'rcar_csi2 feaa0000.csi2':1 [fmt:RGB888_1X24/1280x720 field:none]"

  # media-ctl -d /dev/media0 -V "'adv748x 4-0070 txa':0 [fmt:RGB888_1X24/1280x720 field:none]"

- 720x576p

  # media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':1 -> 'VIN0 output':0 [1]"

  # media-ctl -d /dev/media0 -l "'adv748x 4-0070 hdmi':1 -> 'adv748x 4-0070 txa':0 [1]"

  # media-ctl -d /dev/media0 -V "'rcar_csi2 feaa0000.csi2':1 [fmt:RGB888_1X24/720x576 field:none]"

  # media-ctl -d /dev/media0 -V "'adv748x 4-0070 txa':0 [fmt:RGB888_1X24/720x576 field:none]"

- 720x480p

  # media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':1 -> 'VIN0 output':0 [1]"

  # media-ctl -d /dev/media0 -l "'adv748x 4-0070 hdmi':1 -> 'adv748x 4-0070 txa':0 [1]"

  # media-ctl -d /dev/media0 -V "'rcar_csi2 feaa0000.csi2':1 [fmt:RGB888_1X24/720x480 field:none]"

media-ctl -d /dev/media0 -V "'adv748x 4-0070 txa':0 [fmt:RGB888_1X24/720x480 field:none]"

- 640x480p

media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':1 -> 'VIN0 output':0 [1]"

media-ctl -d /dev/media0 -l "'adv748x 4-0070 hdmi':1 -> 'adv748x 4-0070 txa':0 [1]"

media-ctl -d /dev/media0 -V "'rcar_csi2 feaa0000.csi2':1 [fmt:RGB888_1X24/640x480 field:none]"

media-ctl -d /dev/media0 -V "'adv748x 4-0070 txa':0 [fmt:RGB888_1X24/640x480 field:none]"

[CVBS IN]

[H3/M3/M3N]

- NTSC (720x480i)

media-ctl -d /dev/media0 -l "'rcar_csi2 fea80000.csi2':1 -> 'VIN5 output':0 [1]"

media-ctl -d /dev/media0 -l "'adv748x 4-0070 afe':8 -> 'adv748x 4-0070 txb':0 [1]"

media-ctl -d /dev/media0 -V "'rcar_csi2 fea80000.csi2':1 [fmt:UYVY2X8/720x240 field:alternate]"

media-ctl -d /dev/media0 -V "'adv748x 4-0070 afe':8 [fmt:UYVY2X8/720x240 field:alternate]"

- PAL (720x576i)

media-ctl -d /dev/media0 -l "'rcar_csi2 fea80000.csi2':1 -> 'VIN5 output':0 [1]"

media-ctl -d /dev/media0 -l "'adv748x 4-0070 afe':8 -> 'adv748x 4-0070 txb':0 [1]"

media-ctl -d /dev/media0 -V "'rcar_csi2 fea80000.csi2':1 [fmt:UYVY2X8/720x288 field:alternate]"

media-ctl -d /dev/media0 -V "'adv748x 4-0070 afe':8 [fmt:UYVY2X8/720x288 field:alternate]"

[E3]

- NTSC (720x480i)

please refer to [Change data lane for cvbs input] in chapter 6.3.1

[In case of using VIN4]

media-ctl -d /dev/media0 -r

media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':1 -> 'VIN4 output':0 [1]"

media-ctl -d /dev/media0 -l "'adv748x 0-0070 afe':8 -> 'adv748x 0-0070 txa':0 [1]"

media-ctl -d /dev/media0 -V "'rcar_csi2 feaa0000.csi2':1 [fmt:UYVY2X8/720x240 field:alternate]"

media-ctl -d /dev/media0 -V "'adv748x 0-0070 txa':0 [fmt:UYVY2X8/720x240 field:alternate]"

- PAL (720x576i)

[In case of using VIN4]

media-ctl -d /dev/media0 -r

media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':1 -> 'VIN4 output':0 [1]"

media-ctl -d /dev/media0 -l "'adv748x 0-0070 afe':8 -> 'adv748x 0-0070 txa':0 [1]"

```
# media-ctl -d /dev/media0 -V "'rcar_csi2 feaa0000.csi2':1 [fmt:UYVY2X8/720x288 field:alternate]"

# media-ctl -d /dev/media0 -V "'adv748x 0-0070 txa':0 [fmt:UYVY2X8/720x288 field:alternate]"
```

[V3U]

[In case of LI-AR0231-AP0200-GMSL2 camera]

```
# media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':1 -> 'VIN0 output':0 [1]"

# media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':2 -> 'VIN1 output':0 [1]"

# media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':3 -> 'VIN2 output':0 [1]"

# media-ctl -d /dev/media0 -l "'rcar_csi2 feaa0000.csi2':4 -> 'VIN3 output':0 [1]"

# media-ctl -d /dev/media0 -V "'rcar_csi2 feaa0000.csi2':1 [fmt:Y10_1X10/1920x1020 field:none]"


# media-ctl -d /dev/media0 -l "'rcar_csi2 fed60000.csi2':1 -> 'VIN16 output':0 [1]"

# media-ctl -d /dev/media0 -l "'rcar_csi2 fed60000.csi2':2 -> 'VIN17 output':0 [1]"

# media-ctl -d /dev/media0 -l "'rcar_csi2 fed60000.csi2':3 -> 'VIN18 output':0 [1]"

# media-ctl -d /dev/media0 -l "'rcar_csi2 fed60000.csi2':4 -> 'VIN19 output':0 [1]"

# media-ctl -d /dev/media0 -V "'rcar_csi2 fed60000.csi2':1 [fmt:Y10_1X10/1920x1020 field:none]"


# media-ctl -d /dev/media0 -l "'rcar_csi2 fed70000.csi2':1 -> 'VIN24 output':0 [1]"

# media-ctl -d /dev/media0 -l "'rcar_csi2 fed70000.csi2':2 -> 'VIN25 output':0 [1]"

# media-ctl -d /dev/media0 -l "'rcar_csi2 fed70000.csi2':3 -> 'VIN26 output':0 [1]"

# media-ctl -d /dev/media0 -l "'rcar_csi2 fed70000.csi2':4 -> 'VIN27 output':0 [1]"

#  media-ctl -d /dev/media0 -V "'rcar_csi2 fed70000.csi2':1 [fmt:Y10_1X10/1920x1020 field:none]"
```

5.2.4 [D3]

There is no need to config VIN in D3 through media-ctl

Deactivate all active links

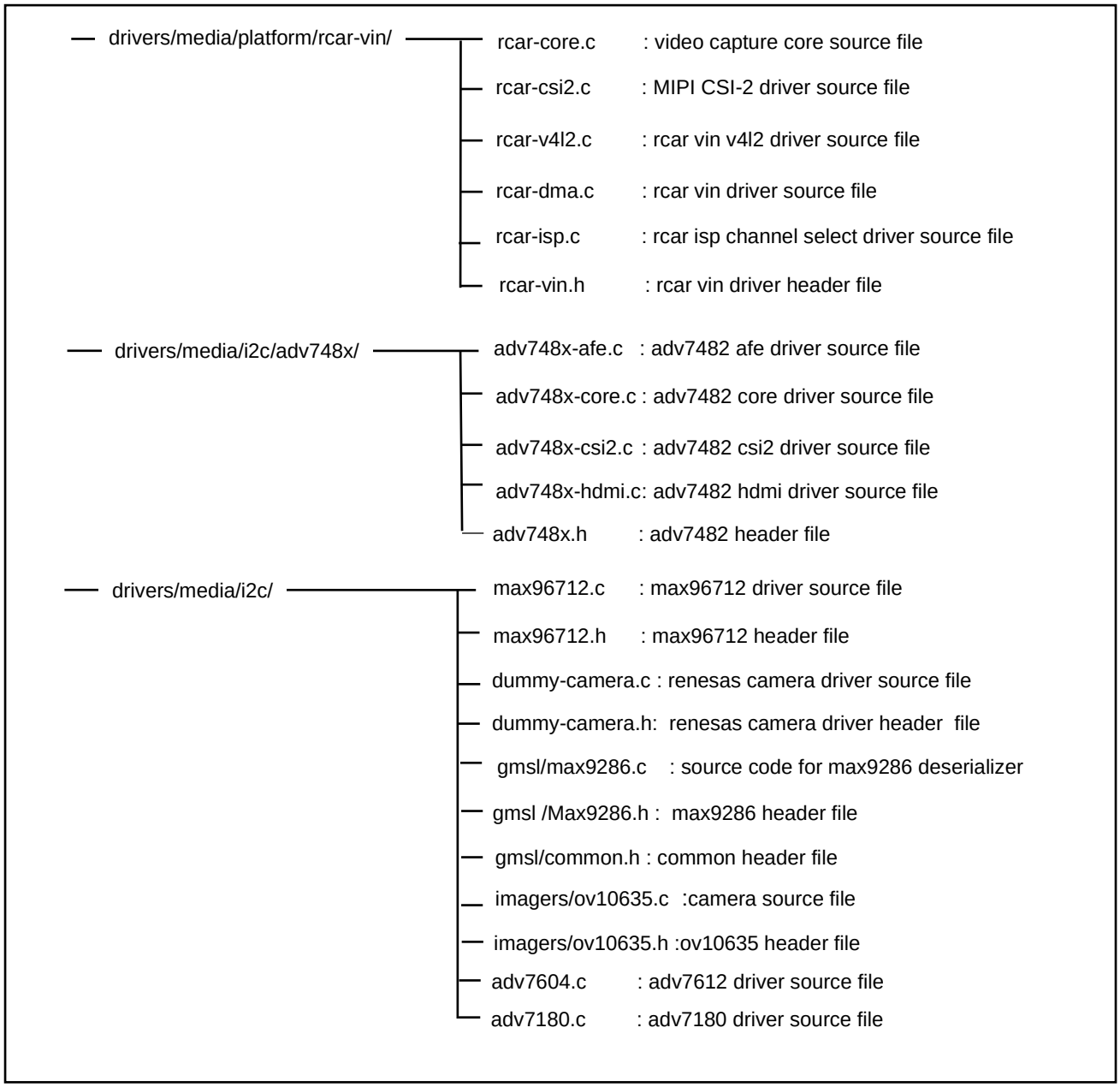
This is a useful command to reset all links before you start enabling new links to make sure you got the biggest possible routing space to start out with.

```
# media-ctl -d /dev/media0 -r
```

6.Integration

Directory Configuration

The directory configuration is shown below



6.1

Figure 6.1 Directory Configuration (R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H)

Integration Procedure

To enable the function of this module, make the following setting with Kernel Configuration.

Video Capture Driver [R-Car H3 / M3 / M3N / E3 / D3 / V3U / V3H]

```
Device Drivers  --->
    <*> Multimedia support  --->
        Media device types  --->
            [*]  Cameras/video grabbers support
        Media core support  ---> *2
            [*]  Media Controller API
        Video4Linux options  --->
            [*]  V4L2 sub-device userspace API
        Media drivers  --->
            [*]  V4L platform devices  --->
                <*>  R-Car Image Signal Processor (ISP)
                <*>  R-Car MIPI CSI-2 Receiver
                <*>  R-Car Video Input (VIN) Driver
                 [ ]  R-Car VIN overflow debug messages  *1
        [ ]  Autoselect ancillary drivers (tuners, sensors, i2c, spi, frontends)
        Media ancillary drivers  --->
            Video decoders --->
                <*> Analog Devices ADV7180 decoder
                <*> Analog Devices ADV748x decoder
                <*> Analog Devices ADV7604 decoder
                <*> Maxim MAX96712 GMSL2 deserializer support
                <*> Dummy camera support
                <*>  MAXIM MAX9286 GMSL deserializer support
                <*>  LVDS camera support
```

note: *1 This configuration is enabled debug message when VIN overflow occurring.
In addition, please step on the following steps after kernel starting.

1. # echo 1 > /sys/module/rcar_vin/parameters/debug (debug message enable and VIN overflow count starts)
2. # cat /sys/module/rcar_vin/parameters/overflow_video (Check VIN overflow count)
3. # 0,0,0,0,0,0,0,0 … (From left to right, /dev/video0, /dev/video1 …. : value shows overflow count)

For overflow interrupt is enabled, it is a possibility that it becomes impossible to properly capture.

*2 Please checkout Filter media drivers option to show Media core support

I2C Driver

```
Device Drivers  --->
    I2C support  --->
        I2C Hardware Bus support  --->
        <*> Renesas R-Car I2C Controller
```

Option Setting

Module Parameters

[R-Car H3 / M3 /M3N]

This module option is controlled by modifying the DT (Device Tree) file (arch/arm64/boot/dts/renesas/salvator-common.dtsi). This section explains how to change the virtual channel of VIN, CSI2 and ADV7482. (Please refer to Documentation\devicetree\bindings\media/rcar-vin.txt about device node definition method)

6.3

6.3.1

```
&vin0 {                                                    // Set VIN channel node (vin0-vin7)
        status = "okay";
};
```

```
&csi40 {                                        // Set csi2 channel node (&csi20 / &csi41 / &csi40)
        status = "okay";

        ports {
                port@0 {
                        reg = <0>;

                        csi40_in: endpoint {
                                clock-lanes = <0>;
                                data-lanes = <1 2 3 4>;          // Set data lane number
                                                                   <1 2 3 4> = 4 lane (csi40 or csi41 only)
                                                                   <1 2> = 2lane
                                                                   <1> = 1 lane
                                remote-endpoint = <&adv7482_txa>; // Set video encoder node
                        };
                };
        };
```

```
...
        video-receiver@70 {
                compatible = "adi,adv7482";

...
                port@a {
                        reg = <10>;

                        adv7482_txa: endpoint {
                                virtual-channel = <0>;           // Set virtual channel. It is not specified by default.
                                                                 // If not set, virtual-channel is set 0.
                                                                  ( 0:VC0,  1:VC1,  2:VC2,  3:VC3)
                                clock-lanes = <0>;
                                data-lanes = <1 2 3 4>;          // Set data lane number <1> or <1 2> or
                                                                  <1 2 3 4>. (hdmi in only)
                                remote-endpoint = <&csi40_in>;   // Set csi2 encoder node
                        };
                };
                port@b {
                        reg = <11>;

                        adv7482_txb: endpoint {
                                virtual-channel = <0>;           // Set virtual channel. It is not specified by default.
                                                                 // If not set, virtual-channel is set 0.
                                                                 ( 0:VC0,  1:VC1,  2:VC2,  3:VC3)
                                clock-lanes = <0>;
                                data-lanes = <1>;                // Set data lane number <1>
                                remote-endpoint = <&csi20_in>;   // Set csi2 encoder node
                        };
                };
        }
```

Please set the virtual-channel to the same value for both txb and txa.

[R-Car E3]

This module option is controlled by modifying the DT (Device Tree) file (arch/arm64/boot/dts/renesas/r8a77990-ebisu.dts, r8a77990-es10-ebisu.dts). This section explains how to change input. About the virtual channel and lane (1 or 2) can be set for lane in csi40 is as described above (in case of R-Car H3/M3/M3N).

- **Set vin4/vin5 channel node for hdmi input**

Device node of video decoder, csi40 and vin is OK with default description.

- **Change data lane for cvbs input**

Please add and correct about following highlight.

```
&csi40 {
        status = "okay";

        ports {
                port@0 {
                        reg = <0>;

                        csi40_in: endpoint {
                                clock-lanes = <0>;
-                               data-lanes = <1 2>;
+                               data-lanes = <1>;
                                remote-endpoint = <&adv7482_txa>;
                        };
                };
        };
};

&i2c0 {
 ・ ・ ・ ・
        video-receiver@70 {
 ・ ・ ・ ・
                port@a {
                        reg = <10>;

                        adv7482_txa: endpoint {

                                clock-lanes = <0>;
-                               data-lanes = <1 2>;
+                               data-lanes = <1>;
                                remote-endpoint = <&csi40_in>;
                        };
                };
        };
};
```

[R-Car D3]

This module option is controlled by modifying the DT (Device Tree) file (arch/arm64/boot/dts/renesas/r8a77995-draak.dts). This section explains how to change ADV7180 and ADV7612.

Set vin4 channel node for hdmi input
The HDMI input is set by default on dts, so no need to modify dts in this case.

In case of this case, also change DIP switch. (R-Car D3 only)

DIP switch	Configuration
49	ALL ON
50	ALL ON
51	ALL ON
52	ALL ON
53	ALL OFF
54	ALL OFF

Set vin4 channel node for cvbs input.

```
composite-in@20 {
        …
        ports {
                …
                port@3 {
                        reg = <3>;
                        …
                        adv7180_out: endpoint {
+                               remote-endpoint = <&vin4_in>;
                        };
                };

hdmi-decoder@4c {
        …
        ports {
                …
                port@2 {                        …
                        adv7612_out: endpoint {
-                               remote-endpoint = <&vin4_in>;
                        };
                };

&vin4 {
        …
        ports {
                port {
                        vin4_in: endpoint {
                                pclk-sample = <0>;
-                               hsync-active = <0>;
-                               vsync-active = <0>;
-                               remote-endpoint = <&adv7612_out>;
+                               remote-endpoint = <&adv7180_out>;
                        };
                };
```

In case of this case, also change DIP switch. (R-Car D3 only)

DIP switch	Configuration
49	ALL OFF
50	ALL OFF
51	ALL OFF
52	ALL OFF
53	ALL ON
54	ALL ON

Kernel Parameters

There are no kernel parameters

6.3.2