# Overview

## Overview

R-Car H3 has video signal processing IP. The IP are VSP for Image (VSPI), VSP for Blend with Color Management (VSPBC), and VSP for Blend with DRC(VSPBD). R-Car M3/M3N/E3 has video signal processing IP. The IP are VSP for Image (VSPI), VSP for Blend (VSPB). These IP are controlled with V4L2 API and Media Controller API on VSP2 Driver. This manual explains how to use VSPI, VSPBC, VSPBD and VSPB with V4L2 API and Media Controller API on VSP2 Driver.

## Reference

### Standard

The following table shows the standard that VSP2 Driver corresponds.

Table 1‑1 Standard

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

### Related Document

The following table shows the document related to VSP2 Driver.

Table 1‑2 Related Document (R-Car H3/M3/M3N/E3)

| Number | Issue | Title |
| --- | --- | --- |
| - | Renesas Electronics | R-Car Series, 3rd Generation User’s Manual:  Hardware |
| - | Renesas Electronics | R-CarH3-SiP System Evaluation Board Salvator-X  Hardware Manual RTP0RC7795SIPB0011S |
| - | Renesas Electronics | R-CarM3-SiP System Evaluation Board Salvator-X  Hardware Manual RTP0RC7796SIPB0011S |
| - | Renesas Electronics | R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board Salvator-XS  Hardware Manual |
|  | Renesas Electronics | R-CarE3 System Evaluation Board Ebisu  Hardware Manual RTP0RC77990SEB0010S |
|  | Renesas Electronics | R-CarE3 System Evaluation Board  Ebisu-4D (E3 board 4xDRAM) Hardware Manual |
| - | Renesas  Electronics | VSP Manager for Linux User’s Manual:  Software |

## Terminology

The following table shows the terminology related to VSP2 Driver.

Table 1‑3 Terminology

| Terms | Explanation |
| --- | --- |
| DRM | Direct Rendering Manager |
| KMS | Kernel Mode Setting |
| DRI | Direct Rendering Infrastructure |
| VSPI | VSP for Image |
| VSPBC | VSP for Blend with Color Management |
| VSPBD | VSP for Blend with DRC |
| VSPB | VSP for Blend |
| VSPM | VSP Manager |
| MMNGR | Memory Manager |
| RPF | Read pixel formatter |
| WPF | Write pixel formatter |
| UDS | Up down scaler |
| BRU | Blend engine unit |
| LUT | Look up table |
| CLU | Cubic look up table |
| HGO | Histogram one dimension |

# Operating Environment

## Hardware Environment

The following table shows the hardware needed to use VSP2 Driver.

Table 2-1 Hardware Specification

| Name | Version | Manufacture |
| --- | --- | --- |
| 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-SiP System Evaluation Board ( Ebisu ) | - | Renesas Electronics |
| R-CarE3 System Evaluation Board Ebisu-4D | - | Renesas Electronics |

## Software Configuration

Figure 2-1 shows the software configuration in which VSP2 Driver is used. VSP2 Driver uses VSP Manager developed by Renesas.

User

Kernel

Application

V4L2 Core

**VSP2 Driver (vsp2.ko)**

VSP Manager (vspm.ko) \*1

S/W

H/W

VSPI

H3 x 2

M3 x1

M3N x1

E3 x1

VSPB

H3 x2

M3 x1

M3N x1

E3 x1

---- VSP2 Driver translates V4L2 request into VSPM request

V4L2 API

media-ctl (libmediactl)

media-ctl

**VSP2 Driver, hereinafter, this is called this module**

Media Controller API

Media Controller Kernel Interface

media-ctl (libv4l2subdev)

Media Controller API for v4l2 subdev

Figure 2‑1 Software Configuration

Note:

\*1 Some parameters of VSP Manager are fixed in this module.

# Function

VSPI, VSPBC, VSPB and VSPBD have some image processing modules. The function of this module is to provide V4L2 and Media Controller API to control image processing modules. V4L2 API controls buffer, start and stop. Media Controller API controls image processing link and each image processing. The targets of the image processing on this module are as follows.

Table 3‑1　Target of Image Processing Module

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ○：Implemented  ―：Not implemented | |  |  | |
|  | R-Car H3  VSPI | R-Car H3  VSPBC | R-Car H3  VSPBD | R-Car M3/M3N/E3  VSPI | R-Car M3/M3N/E3  VSPB | |
| Read pixel formatter (RPF) | ○ (1 RPF) | ○ (5 RPF) | ○ (5 RPF) | 〇  (1RPF) | 〇  (5RPF) | |
| Write pixel formatter (WPF) | ○ | ○ | ○ | ○ | ○ | |
| Up down scaler (UDS) | ○ | ― | ― | ○ | ― | |
| Blend engine unit (BRU) | ― | ○ | ○ | ― | ○ | |
| Look up table (LUT) | ○ | ○ | ― | ○ | ○ | |
| Cubic look up table (CLU) | ○ | ○ | ― | ○ | ○ | |
| Histogram one dimension　(HGO) | ○ | ○ | ― | ○ | ○ | |

Note:

・It’s not available to use VSPI function and VSPB function at the same request.

(e.g. UDS (VSPI) and BRU (VSPB) can’t be operated at the same request. )

・Refer to figure and explanation of Figure 6-3 in VSP Manager User’s Manual about color space specification of image processing modules.

・Refer to 6.2 in VSP Manager User’s Manual about Input/Output image limited size of image processing modules except for input minimum size of YUV formats. Input minimum size of YUV formats is width 2, height 2.

・This module does not support cropping in WPF.

・When using HGO refer to section 32.3.7.8 of the H/W manual.

# V4L2 API

This module should be accessed through the “/dev/video(number)” . To see detail the number of “/dev/video(number)”, refer to 6. media/video/v4l-subdev. To see detail API specification, refer to <https://linuxtv.org/downloads/v4l-dvb-apis/index.html>.

## List of V4L2 API

The following list shows the V4L2 API available in this module.

To see detail API specification, refer to https://linuxtv.org/downloads/v4l-dvb-apis/userspace-api/index.html

Table 4‑1 List of V4L2 API

|  |  |
| --- | --- |
| Function Name | Overview of Processing |
| open | Opens the V4L2 device |
| close | Closes the V4L2 device |
| select | Checks if DQBUF can be executed |
| mmap | Maps a buffer in user space |
| munmap | Unmaps a buffer in user space |
| ioctl(VIDIOC\_QBUF) | Registers a buffer |
| ioctl(VIDIOC\_DQBUF) | Releases a buffer |
| ioctl(VIDIOC\_G\_FMT) | Acquires the data format |
| ioctl(VIDIOC\_S\_FMT) | Specifies the data format |
| ioctl(VIDIOC\_REQBUFS) | Requests a buffer area |
| ioctl(VIDIOC\_QUERYBUF) | Queries the status of a buffer |
| ioctl(VIDIOC\_STREAMON) | Starts streaming |
| ioctl(VIDIOC\_STREAMOFF) | Stops streaming |
| ioctl(VIDIOC\_QUERYCAP) | Inquire about the device capabilities |
| ioctl(VIDIOC\_EXPBUF) | Export a buffer as a DMABUF file descriptor |
| ioctl(VIDIOC\_S\_CTRL) | Set the value of a control |
| ioctl(VIDIOC\_S\_EXT\_CTRLS) | Set the value of several controls |

Note:

・ioctl(VIDIOC\_S\_CTRL) is used to set global alpha value. If you set global alpha, call this function before ioctl(VIDIOC\_STREAMON). If this function is not used, global alpha is not used.

・ioctl(VIDIOC\_S\_EXT\_CTRLS) is used to set near-lossless enable/disable. If you set near-lossless enable/disable, call this function before ioctl(VIDIOC\_STREAMON). The default setting is disable.

・When using the color space conversion, you can be specified the method with ioctl(VIDIOC\_S\_FMT).

Specify V4L2\_QUANTIZATION\_FULL\_RANGE or V4L2\_QUANTIZATION\_LIM\_RANGE for quantization variable and V4L2\_YCBCR\_ENC\_601 or V4L2\_YCBCR\_ENC\_709 for ycbcr\_enc variable.

The setting of ycbcr\_enc is effective on the YUV format side.

・The data\_offset parameter of struct vb2\_plane is not supported , such as when calling ioctl(VIDIOC\_QBUF).

## List of V4L2 Structure

The following list shows the list of V4L2 structures available in this module.

To see detail Structure specification, refer to https://linuxtv.org/downloads/v4l-dvb-apis/driver-api/v4l2-core.html.

Table 4‑2　List of V4L2 Structure Definition

|  |
| --- |
| Structure Name |
| v4l2\_buffer |
| v4l2\_plane |
| v4l2\_format |
| v4l2\_requestbuffers |
| v4l2\_capability |
| v4l2\_exportbuffer |
| v4l2\_ext\_control |
| v4l2\_ext\_controls |
| v4l2\_control |

## Buffer Allocation

There are three types to allocate a buffer for input or output.

- Memory mapping

- User pointer

- DMA Buffers

Application can select the type with VIDIOC\_REQBUFS.

To see a buffer allocation in details, refer to https://linuxtv.org/downloads/v4l-dvb-apis/userspace-api/v4l/buffer.html

### Memory Mapping

The V4L2 driver allocates a physical contiguous buffer in kernel space. Memory mapping is primarily intended to map buffers in device memory into the application's address space.

### User Pointer

Application allocates a physical contiguous buffer with KMS or MMNGR.

Note: Use the address in user space, when application sets the buffers with VIDIOC\_QBUF.

### DMA Buffers

Application allocates a physical contiguous buffer with KMS or MMNGR and creates an identifier from the buffer with DRM or MMNGR. The identifier is called DMA buffer.

## List of Format of Image Buffer

The following list shows the list of formats available in this module. To see detail format specification, refer to https://linuxtv.org/downloads/v4l-dvb-apis/userspace-api/v4l/pixfmt.html

Table 4‑3 List of Format of Image Buffer

|  |
| --- |
| Image Format |
| V4L2\_PIX\_FMT\_RGB332 |
| V4L2\_PIX\_FMT\_RGB444 |
| V4L2\_PIX\_FMT\_XRGB444 |
| V4L2\_PIX\_FMT\_RGB555 |
| V4L2\_PIX\_FMT\_XRGB555 |
| V4L2\_PIX\_FMT\_RGB565 |
| V4L2\_PIX\_FMT\_BGR24 |
| V4L2\_PIX\_FMT\_RGB24 |
| V4L2\_PIX\_FMT\_BGR32 |
| V4L2\_PIX\_FMT\_RGB32 |
| V4L2\_PIX\_FMT\_XBGR32 |
| V4L2\_PIX\_FMT\_XRGB32 |
| V4L2\_PIX\_FMT\_ABGR32 |
| V4L2\_PIX\_FMT\_ARGB32 |
| V4L2\_PIX\_FMT\_UYVY |
| V4L2\_PIX\_FMT\_VYUY |
| V4L2\_PIX\_FMT\_YUYV |
| V4L2\_PIX\_FMT\_YVYU |
| V4L2\_PIX\_FMT\_NV12M |
| V4L2\_PIX\_FMT\_NV21M |
| V4L2\_PIX\_FMT\_NV16M |
| V4L2\_PIX\_FMT\_NV61M |
| V4L2\_PIX\_FMT\_YUV420M |
| V4L2\_PIX\_FMT\_YVU420M |
| V4L2\_PIX\_FMT\_YUV422M |
| V4L2\_PIX\_FMT\_YVU422M |
| V4L2\_PIX\_FMT\_YUV444M |
| V4L2\_PIX\_FMT\_YVU444M |

Note:

・V4L2\_PIX\_FMT\_RGB444, V4L2\_PIX\_FMT\_RGB555, V4L2\_PIX\_FMT\_BGR32 and V4L2\_PIX\_FMT\_RGB32 are deprecated and must not be used by new drivers. Therefore these formats will be not supported in this module.

・The top address of UV must be 4k aligned in two-plane versions of the YUV format.

・Size

Variation of YUV420: 2-pixel units both horizontally and vertically.

Variation of YUV422: 2-pixel units horizontally and 1-pixel units vertically.

・Crop size and position

Variation of YUV: 2-pixel units both horizontally and vertically.　Concretely, it is from V4L2\_PIX\_FMT\_UYVY to V4L2\_PIX\_FMT\_YVU444M in Table 4-3.

・V4L2\_PIX\_FMT\_ARGB32, YUV444M, YUV422M, YUV420M, YUYV are only support, when near-lossless is enabled.

・The top address and the stride of each plane must be a multiple of 256, when near-lossless is enabled.

# Media controller API

## List of Media Controller Kernel Interface

The following list shows the list of Media Controller Kernel Interface available in this module.

The purpose of the following interfaces is to set up links b/w image processing in VSPI, VSPBC, VSPBD and VSPB.

Table 5‑1 List of Media Controller Kernel Interface

|  |  |
| --- | --- |
| Function Name | Overview of Processing |
| open | Open a media device |
| close | Close a media device |
| ioctl(MEDIA\_IOC\_DEVICE\_INFO) | Query device information |
| ioctl(MEDIA\_IOC\_ENUM\_ENTITIES) | Enumerate entities and their properties |
| ioctl(MEDIA\_IOC\_ENUM\_LINKS) | Enumerate all pads and links for a given entity |
| ioctl(MEDIA\_IOC\_SETUP\_LINK) | Modify the properties of a link |

To see detail interface specification, refer to https://linuxtv.org/downloads/v4l-dvb-apis/userspace-api/mediactl/media-funcs.html. Application can also use these interfaces via Media Controller API in libmediactl.

Note:

・The values set with Media Controller API are reflected at ioctl(VIDIOC\_STREAMON).

## List of Media Controller API

### Media Controller API

The following list shows Media Controller API available in this module.

The purpose of the following API is to set up links b/w image processing in VSPI, VSPBC, VSPBD and VSPB.

Table 5‑2 List of Media Controller API

|  |  |
| --- | --- |
| Function Name | Overview of Processing |
| media\_device\_new() | Create a new media device |
| media\_device\_enumerate() | Enumerate the device topology |
| media\_device\_unref() | Release a reference to the device |
| media\_reset\_links() | Reset all links to the disabled state. |
| media\_parse\_link() | Parse string to a link on the media device. |
| media\_setup\_link() | Configure a link. |
| media\_parse\_pad() | Parse string to a pad on the media device. |
| media\_get\_entity\_by\_name() | Find an entity by its name |
| media\_entity\_get\_devname() | Get the device node name for an entity |

Note:

・The values set with Media Controller API are reflected at ioctl(VIDIOC\_STREAMON).

The following list shows Media Controller API for v4l2 subdev available in this module.

The purpose of the following API is to control image processing in VSPI, VSPBC, VSPBD and VSPB.

Table 5‑3 List of Media Controller API for v4l2 subdev

|  |  |
| --- | --- |
| Function Name | Overview of Processing |
| v4l2\_subdev\_set\_format() | Set the format on a pad. |
| v4l2\_subdev\_set\_selection() | Set a selection rectangle on a pad. |
| v4l2\_subdev\_get\_selection() | Get a selection rectangle on a pad |
| v4l2\_subdev\_get\_format() | Retrieve the format on a pad. |

Note:

・The values set with Media Controller API are reflected at ioctl(VIDIOC\_STREAMON).

### Media Controller Structure

The following list shows the list of Media Controller structures available in this module

Table 5‑4 List of Media Controller Structure Definition

|  |
| --- |
| Structure Name |
| media\_device |
| media\_link |
| media\_entity |
| v4l2\_mbus\_framefmt |
| v4l2\_rect |

## Media Controller Application

Media Controller application, media-ctl is a user application tool used in terminal. Source code is in the following URL.

Table 5‑5 URL of Media Controller Application

|  |
| --- |
| Git |
| git://linuxtv.org/v4l-utils.git |

When you create a filesystem using R-Car H3/M3/M3N/E3 Yocto recipe package files, media-ctl will be installed in it automatically.

# media/video/v4l-subdev

The control targets of V4L2 API and Media Controller API are implemented in device files (/dev/media, /dev/video or /dev/v4l-subdev). Media Controller uses /dev/media to select VSPI, VSPBC, VSPBD and VSPB. V4L2 API uses /dev/video to select the control targets of inputs and output in VSPI, VSPBC, VSPBD and VSPB. Media Controller API uses /dev/v4l-subdev to select the control target of image processing in VSPI, VSPBC, VSPBD and VSPB. The device files have Entity, Pad and Link.

Table 6‑1 List of Component

|  |  |
| --- | --- |
| Component | Explanation |
| Entity | A basic media hardware or software building block |
| Pad | A connection endpoint through which an entity can interact with other entities. Data produced by an entity flows from the entity's output to one or more entity inputs. |
| Link | A point-to-point oriented connection between two pads, either on the same entity or on different entities. Data flows from a source pad to a sink pad. |

Note:

Control device files (/dev/media, /dev/video or /dev/v4l-subdev) related to one IP (ex. VSPI1) by one thread from open to close of the files.

[R-Car H3]

There are information For VSPBC(/dev/media1).

Table 6‑2 Device/Subdevice file of VSPBC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Device/Subdevice file | Entity | Pad | Functionality |
| /dev/media1 | /dev/video(8~12) | fe920000.vsp rpf.(0-4) input | source | Input image |
| /dev/v4l-subdev6 | fe920000.vsp bru | source, sink | Blend image |
| /dev/v4l-subdev7 | fe920000.vsp lut | source, sink | LUT |
| /dev/v4l-subdev8 | fe920000.vsp clu | source, sink | 3D-LUT |
| /dev/v4l-subdev9 | fe920000.vsp hgo | source, sink | Histogram one dimension |
| /dev/v4l-subdev(10~14) | fe920000.vsp rpf.(0-4) | source, sink | Color format conversion  Pre cropping |
| /dev/v4l-subdev15 | fe920000.vsp wpf.0 | source, sink | Color format conversion  Post cropping |
| /dev/video13 | fe920000.vsp wpf.0 output | sink | Output image (For memory access) |

There are information For VSPBD(/dev/media2).

Table 6‑3 Device/Subdevice file of VSPBD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Device/Subdevice file | Entity | Pad | Functionality |
| /dev/media2 | /dev/video(14~18) | fe960000.vsp rpf.(0-4) input | source | Input image |
| /dev/v4l-subdev16 | fe960000.vsp bru | source, sink | Blend image |
| /dev/v4l-subdev(17~21) | fe960000.vsp rpf.(0-4) | source, sink | Color format conversion  Pre cropping |
| /dev/v4l-subdev22 | fe960000.vsp wpf.0 | source, sink | Color format conversion  Post cropping |
| /dev/video19 | fe960000.vsp wpf.0 output | sink | Output image (For memory access) |

There are information For VSPI0(/dev/media3).

Table 6‑4 Device/Subdevice file of VSPI0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Device/Subdevice file | Entity | Pad | Functionality |
| /dev/media3 | /dev/video20 | fe9a0000.vsp rpf.0 input | source | Input image |
| /dev/v4l-subdev23 | fe9a0000.vsp lut | source, sink | LUT |
| /dev/v4l-subdev24 | fe9a0000.vsp clu | source, sink | 3D-LUT |
| /dev/v4l-subdev25 | fe9a0000.vsp hgo | source, sink | Histogram one dimension |
| /dev/v4l-subdev26 | fe9a0000.vsp hgt | source, sink | Histogram one dimension (Not Support) |
| /dev/v4l-subdev27 | fe9a0000.vsp rpf.0 | source, sink | Color format conversion  Pre cropping |
| /dev/v4l-subdev28 | fe9a0000.vsp uds.0 | source, sink | Scaling |
| /dev/v4l-subdev29 | fe9a0000.vsp wpf.0 | source, sink | Color format conversion  Post cropping |
| /dev/video21 | fe9a0000.vsp wpf.0 output | sink | Output image (For memory access) |

There are information For VSPI1 (/dev/media4).

Table 6‑5 Device/Subdevice file of VSPI1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Device/Subdevice file | Entity | Pad | Functionality |
| /dev/media4 | /dev/video22 | fe9b0000.vsp rpf.0 input | source | Input image |
| /dev/v4l-subdev30 | fe9b0000.vsp lut | source, sink | LUT |
| /dev/v4l-subdev31 | fe9b0000.vsp clu | source, sink | 3D-LUT |
| /dev/v4l-subdev32 | fe9b0000.vsp hgo | source, sink | Histogram one dimension |
| /dev/v4l-subdev33 | fe9b0000.vsp hgt | source, sink | Histogram two dimension  (Not support) |
| /dev/v4l-subdev34 | fe9b0000.vsp rpf.0 | source, sink | Color format conversion  Pre cropping |
| /dev/v4l-subdev35 | fe9b0000.vsp uds.0 | source, sink | Scaling |
| /dev/v4l-subdev36 | fe9b0000.vsp wpf.0 | source, sink | Color format conversion  Post cropping |
| /dev/video23 | fe9b0000.vsp wpf.0 output | sink | Output image (For memory access) |

Note:

・The number from /dev/video8 to /dev/video25 depends on the other v4l2 driver (ex. VIN).

・The number from /dev/media1 to /dev/media5 depends on the other driver.

・The number from /dev/v4l-subdev6 to /dev/v4l-subdev36 depends on the other driver.

・The range of the ratio of uds is 1/16 < ratio <= 16.

・In VSPBC and VSPBD, when bru is not used, use rpf.0 only. When bru is used, use rpf(0-4) in ascending order.

[R-Car M3/M3N]

There are information For VSPB(/dev/media1).

Table 6‑7 Device/Subdevice file of VSPB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Device/Subdevice file | Entity | Pad | Functionality |
| /dev/media1 | /dev/video(8~12) | fe960000.vsp rpf.(0-4) input | source | Input image |
| /dev/v4l-subdev6 | fe960000.vsp bru | source, sink | Blend image |
| /dev/v4l-subdev7 | fe960000.vsp lut | source, sink | LUT |
| /dev/v4l-subdev8 | fe960000.vsp clu | source, sink | 3D-LUT |
| /dev/v4l-subdev9 | fe960000.vsp hgo | source, sink | Histogram one dimension |
| /dev/v4l-subdev(10~14) | fe960000.vsp rpf.(0-4) | source, sink | Color format conversion  Pre cropping |
| /dev/v4l-subdev15 | fe960000.vsp wpf.0 | source, sink | Color format conversion  Post cropping |
| /dev/video13 | fe960000.vsp wpf.0 output | sink | Output image (For memory access) |

There are information For VSPI(/dev/media2).

Table 6‑8 Device/Subdevice file of VSPI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Device/Subdevice file | Entity | Pad | Functionality |
| /dev/media2 | /dev/video14 | fe9a0000.vsp rpf.0 input | source | Input image |
| /dev/v4l-subdev16 | fe9a0000.vsp lut | source, sink | LUT |
| /dev/v4l-subdev17 | fe9a0000.vsp clu | source, sink | 3D-LUT |
| /dev/v4l-subdev18 | fe9a0000.vsp hgo | source, sink | Histogram one dimension |
| /dev/v4l-subdev19 | fe9a0000.vsp hgt | source, sink | Histogram two dimension (Not Support) |
| /dev/v4l-subdev20 | fe9a0000.vsp rpf.0 | source, sink | Color format conversion  Pre cropping |
| /dev/v4l-subdev21 | fe9a0000.vsp uds.0 | source, sink | Scaling |
| /dev/v4l-subdev22 | fe9a0000.vsp wpf.0 | source, sink | Color format conversion  Post cropping |
| /dev/video15 | fe9a0000.vsp wpf.0 output | sink | Output image (For memory access) |

Note:

・The number from /dev/video8 to /dev/video15 depends on the other v4l2 driver (ex. VIN).

・The number from /dev/media1 to /dev/media2 depends on the other driver.

・The number from /dev/v4l-subdev6 to /dev/v4l-subdev22 depends on the other driver.

・The range of the ratio of uds is 1/16 < ratio <= 16.

・In VSPB, when bru is not used, use rpf.0 only. When bru is used, use rpf(0-4) in ascending order.

[R-Car E3]

There are information For VSPB(/dev/media1).

Table 6‑9 Device/Subdevice file of VSPB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Device/Subdevice file | Entity | Pad | Functionality |
| /dev/media1 | /dev/video(2~6) | fe960000.vsp rpf.(0-4) input | source | Input image |
| /dev/v4l-subdev4 | fe960000.vsp bru | source, sink | Blend image |
| /dev/v4l-subdev5 | fe960000.vsp lut | source, sink | LUT |
| /dev/v4l-subdev6 | fe960000.vsp clu | source, sink | 3D-LUT |
| /dev/v4l-subdev7 | fe960000.vsp hgo | source, sink | Histogram one dimension |
| /dev/v4l-subdev(8~12) | fe960000.vsp rpf.(0-4) | source, sink | Color format conversion  Pre cropping |
| /dev/v4l-subdev13 | fe960000.vsp wpf.0 | source, sink | Color format conversion  Post cropping |
| /dev/video7 | fe960000.vsp wpf.0 output | sink | Output image (For memory access) |

There are information For VSPI(/dev/media2).

Table 6‑10 Device/Subdevice file of VSPI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Device/Subdevice file | Entity | Pad | Functionality |
| /dev/media2 | /dev/video8 | fe9a0000.vsp rpf.0 input | source | Input image |
| /dev/v4l-subdev14 | fe9a0000.vsp lut | source, sink | LUT |
| /dev/v4l-subdev15 | fe9a0000.vsp clu | source, sink | 3D-LUT |
| /dev/v4l-subdev16 | fe9a0000.vsp hgo | source, sink | Histogram one dimension |
| /dev/v4l-subdev17 | fe9a0000.vsp hgt | source, sink | Histogram two dimension (Not Support) |
| /dev/v4l-subdev18 | fe9a0000.vsp rpf.0 | source, sink | Color format conversion  Pre cropping |
| /dev/v4l-subdev19 | fe9a0000.vsp uds.0 | source, sink | Scaling |
| /dev/v4l-subdev20 | fe9a0000.vsp wpf.0 | source, sink | Color format conversion  Post cropping |
| /dev/video9 | fe9a0000.vsp wpf.0 output | sink | Output image (For memory access) |

Note:

・The number from /dev/video2 to /dev/video9 depends on the other v4l2 driver (ex. VIN).

・The number from /dev/media1 to /dev/media2 depends on the other driver.

・The number from /dev/v4l-subdev3 to /dev/v4l-subdev19 depends on the other driver.

・The range of the ratio of uds is 1/16 < ratio <= 16.

・In VSPB, when bru is not used, use rpf.0 only. When bru is used, use rpf(0-4) in ascending order.

# Integration

## Directory Configuration

About V4l2 Core Directory Configuration, please refer to linux kernel (linux/drivers/media/v4l2-core).

The directory configuration of this module is shown below.

Note :

・This module is put on <https://github.com/renesas-rcar/vsp2driver>

・This module uses VSP Manager. Therefore this module doesn’t work without VSP Manager.

vsp2driver

├── GPL-COPYING

├── MIT-COPYING

├── README

└── vsp2driver

├── Makefile

├── document

│ └── renesas,vsp2.txt

├── linux

│ └── vsp2.h

├── vsp2\_addr.c

├── vsp2\_addr.h

├── vsp2\_brs.c

├── vsp2\_brs.h

├── vsp2\_bru.c

├── vsp2\_bru.h

├── vsp2\_clu.c

├── vsp2\_clu.h

├── vsp2\_debug.c

├── vsp2\_debug.h

├── vsp2\_device.h

├── vsp2\_drv.c

├── vsp2\_entity.c

├── vsp2\_entity.h

├── vsp2\_hgo.c

├── vsp2\_hgo.h

├── vsp2\_hgt.c

├── vsp2\_hgt.h

├── vsp2\_lut.c

├── vsp2\_lut.h

├── vsp2\_pipe.c

├── vsp2\_pipe.h

├── vsp2\_regs.h

├── vsp2\_rpf.c

├── vsp2\_rwpf.c

├── vsp2\_rwpf.h

├── vsp2\_uds.c

├── vsp2\_uds.h

├── vsp2\_video.c

├── vsp2\_video.h

├── vsp2\_vspm.c

├── vsp2\_vspm.h

└── vsp2\_wpf.c

Figure 7‑1 Directory Configuration (R-Car H3/M3/M3N/E3)

# Sample Source Code

<https://github.com/renesas-rcar/vsp2driver_tp>