RZ/A1H Group

Video Utility

Introduction

This document describes the functional specifications of Renesas Video Application Interface (RVAPI) for a RZ/A1H Software Package that supports a Renesas Starter Kit + for RZ/A1H with an RZ/A series RZ/A1H group MCU.

Target Device / Target Board

Target Device : RZ/A1H

Target Board : Renesas Starter Kit+ for RZ/A1H (YR0K77210C000BE)

Contents

1. Specifications 6
2. Documents 6
2.1 Summaries of the Related Documents 6
3. Hardware Description 7
3.1 List of Pins That are Used 7
4. Software Description 8
4.1 File 8
4.2 Functions 9
5. Function Reference (VDC5) 10
5.1 R_RVAPI_InitializeVDC 10
5.1.1 Description 10
5.1.2 Parameter details 11
5.1.3 Setting up the panel clock 13
5.2 R_RVAPI_TerminateVDC 14
5.2.1 Description 14
5.3 R_RVAPI_DispControlVDC 15
5.3.1 Description 15
5.3.2 Parameter details 16
5.4 R_RVAPI_GraphCreateSurfaceVDC 18
5.4.1 Description 18
5.4.2 Parameter details 19
5.5 R_RVAPI_GraphChangeSurfaceVDC 22
5.5.1 Description 22
5.6 R_RVAPI_GraphDestroySurfaceVDC 23
5.6.1 Description 23
5.7 R_RVAPI_DispPortSettingVDC 24
5.7.1 Description 24
5.8 R_RVAPI_VideoControlVDC 25
5.8.1 Description 25
5.8.2 Parameter details 26
5.9 R_RVAPI_VideoCreateSurfaceVDC 28
5.9.1 Description 29
5.9.2 Parameter details 29
5.9.3 About the configuration of the video capture range 33
5.10 R_RVAPI_VideoDestroySurfaceVDC 34
5.10.1 Description 34
5.11 R_RVAPI_VideoPortSettingVDC 35

5.11.1 Description 35
5.12 R_RVAPI_InterruptEnableVDC 36
5.12.1 Description 36
5.12.2 Parameter details 36
5.13 R_RVAPI_InterruptDisableVDC 38
5.13.1 Description 38
5.14 R_RVAPI_AlphablendingRectVDC 39
5.14.1 Description 39
5.15 R_RVAPI_ChromakeyVDC 40
5.15.1 Description 40
5.16 R_RVAPI_DispCalibrationVDC 41
5.16.1 Description 41
5.16.2 Parameter details 42
5.17 R_RVAPI_DispGammaVDC 44
5.17.1 Description 44
5.18 R_RVAPI_VideoCalibrationVDC 45
5.18.1 Description 45
5.18.2 Parameter details 46
5.19 R_RVAPI_VideoSharpnessLtiVDC 47
5.19.1 Description 47
5.19.2 Parameter details 48
5.20 R_RVAPI_GraphChangeSurfaceConfigVDC 51
5.20.1 Description 51
5.20.2 Parameter details 52
5.21 R_RVAPI_AlphablendingVDC 54
5.21.1 Description 54
5.22 R_RVAPI_VideoCreateSurfaceIMRL2 55
5.22.1 Description 55
5.23 R_RVAPI_VideoCreateSurfaceIMRLD 56
5.23.1 Description 56

6. Function Reference (CEU) 57
6.1 R_RVAPI_InitializeCEU 57
6.1.1 Description 57
6.2 R_RVAPI_TerminateCEU 58
6.2.1 Description 58
6.3 R_RVAPI_PortSettingCEU 59
6.3.1 Description 59
6.4 R_RVAPI_OpenCEU 60
6.4.1 Description 60
6.4.2 Parameter details 61

6.4.3 About the configuration of the capture size 64
6.5 R_RVAPI_CaptureStartCEU 66
6.5.1 Description 66
6.6 R_RVAPI_CaptureStatusCEU 67
6.6.1 Description 67

List of Abbreviations and Acronyms

Abbreviation	Full Form
API	Application Programming Interface
ARGB8888	Alpha Red Green Blue 32-bit (8 bits for each of A, R, G, B)
CEU	Capture Engine Unit
CLUT	Color Look Up Table
CMOS	Complementary Metal-Oxide-Semiconductor
DC	Direct Current
GBR	Green Blue Red
HD	Horizontal sync signal
IDE	Integrated Development Environment
I/F	InterFace
I/O	Input / Output
LCD	Liquid Crystal Display
LPF	Low Pass Filter
LSB	Least Significant Bit
LTI	Luminance Transient Improvement
LVDS	Low-Voltage Differential Signaling
MCU	MicroController Unit
MHz	MegaHertz
NC	Not Connected
NTSC	National Television System Committee
PLL	Phase Locked Loop
QE	Quick and Easy
RAM	Random Access Memory
RGB565	Red Green Blue 16-bit, 5 bits for red and blue, 6 bits for green
RGB666	Red Green Blue 18-bit, 6 bits for each of red, green and blue
RGB888	Red Green Blue 24-bit, 8 bits for each of red, green and blue
RGBA8888	Red Green Blue Alpha 32-bit (8 bits for each of R, G, B, A)
RVAPI	Renesas Video Application Interface
VD	Vertical sync signal
VDC5	Video Display Controller 5
VDEC	Video DECoder
YCbCr	Y - Luma, Cb - blue chroma difference, red chroma difference
YCbCr422	Y - Luma (4-bit), Cb - blue chroma difference (2-bit), red chroma difference (2-bit)
YUV422	Y – Luma (4-bit), U - blue projection (2-bit), V - red projection (2-bit)

List of Abbreviations and Acronyms

1. Specifications

RVAPI implements the functions of controlling the display output and video input using the video display controller (VDC5) and various drivers for the NTSC all of which are mounted on the RZ/A1.

Table 1-1 shows the peripheral functions to be used and their uses

Peripheral Function	Use
RZ/A1H embedded VDC5 control	Display and video input control Display and image quality adjustment
RZ/A1H embedded NTSC control	NTSC camera video input control

Table 1-1 Peripheral Functions Used by RVAPI and their Uses

2. Documents

2.1 Summaries of the Related Documents

Summaries of the related documents follow.

- RZ/A1H Group, RZ/A1M Group User's Manual: Hardware (R01UH0403)
  This document describes the hardware specifications for RZ/A1H.
- Renesas Starter Kit + for RZ/A1H User's Manual For e^2 studio (R20UT3007)
  This document describes the specifications of Renesas Starter Kit + for RZ/A1H.

3. Hardware Description

3.1 List of Pins That are Used

Table 3-1 lists the pins to be used and their function.

Pin Name	Input/ Output	Description	Renesas Starter Kit + for RZ/A1H connection
LCD0_CLK	Output	Panel clock	CN44
LCD0_DATA23 to 0	Output	Video image data for panel	CN44
LCD0_TCON6 to 0	Output	Control signal for panel	CN44
LCD0_EXTCLK	Output	Panel clock source	NC
DV0_CLK	Input	External input clock	CN40
DV0_VSYNC	Input	External input Vsync	CN40
DV0_HSYNC	Input	External input Hsync	CN40
DV0_DATA23 to 0	Input	External input video image data	CN40
VIO_D7 to 0	Input	CEU data bus	CN40
VIO_CLK	Input	CEU clock	CN40
VIO_VD	Input	CEU vertical sync	CN40
VIO_HD	Input	CEU horizontal sync	CN40
VIO_FLD	Input	Field signal	NC

Note: Refer to the specifications for the individual evaluation board for details

Table 3-1 Pins Used and their Function

4. Software Description

4.1 File

Table 4-1 shows the files used by the RVAPI.

```
Top
└──RZA1H_Sample
    └──src
        └──renesas
            └──middleware
                └──video
                    ├ inc
                    │   r_rvapi_header.h      : RVAPI public header
                    │   r_rvapi_vdc.h         : RVAPI VDC5 public header
                    │   r_rvapi_vdec.h        : RVAPI DVDEC public header (not supported)
                    │   r_rvapi_ceu.h         : RVAPI CEU public header
                    │
                    └ src
                        r_rvapi_vdc.c         : IRVAPI VDC5 API implementation
                        r_rvapi_vdec.c        : RVAPI DVDEC API implementation (not supported)
                        r_rvapi_ceu.c         : RVAPI CEU API implementation
```

Table 4-1 Files Used by the RVAPI

4.2 Functions

Table 4-2 gives a list of RVAPI functions. The list also contains the functions that need configuration when providing 'display only', 'video input only', or 'video input and display' functions.

Display Only	Video Input	Video Display	Function Name	Section No.	Outline
VDC5 video input display function					
Required	Required	Required	R_RVAPI_InitializeVDC	5.1	VDC5 initialization clock setup
-	-	-	R_RVAPI_TerminateVDC	5.2	VDC5 termination setup
Required	-	Required	R_RVAPI_DispControlVDC	5.3	Display output setup
Required	-	-	R_RVAPI_GraphCreateSurfaceVDC	5.4	Display area generation
-	-	-	R_RVAPI_GraphChangeSurfaceVDC	5.5	Display area change
-	-	-	R_RVAPI_GraphDestroySurfaceVDC	5.6	Display area disposal
Required	-	Required	R_RVAPI_DispPortSettingVDC	5.7	Display output pin setup
-	Required	Required	R_RVAPI_VideoControlVDC	5.8	Video input setup
-	Required	Required	R_RVAPI_VideoCreateSurfaceVDC	5.9	Video and display area generation
-		-	R_RVAPI_VideoDestroySurfaceVDC	5.10	Video and display area cancellation
-	Required	Required	R_RVAPI_VideoPortSettingVDC	5.11	Video input pin setup
-	-	-	R_RVAPI_InterruptEnableVDC	5.12	VDC5 interrupt enable setup
-	-	-	R_RVAPI_InterruptDisableVDC	5.13	VDC5 interrupt disable setup
-	-	-	R_RVAPI_AlphablendingRectVDC	5.14	Rectangle alpha blend
			R_RVAPI_ChromakeyVDC	5.15	Transparency using chroma key
VDC5 image quality adjustment function					
-	-	-	R_RVAPI_DispCalibrationVDC	5.16	Screen output calibration processing
-	-	-	R_RVAPI_DispGammaVDC	5.17	Gamma calibration setup
-	-	-	R_RVAPI_VideoCalibrationVDC	5.18	Color matrix setup
-	-	-	R_RVAPI_VideoSharpnessLtiVDC	5.19	Image enhancement processing
-	-	-	R_RVAPI_GraphChangeSurfaceConfigVDC	5.20	Data read change processing
-	-	-	R_RVAPI_AlphablendingVDC	5.21	1-bit alpha blending setup
CEU video input functions (Note 1)					
-	Required	Required	R_RVAPI_InitializeCEU	6.1	CEU initialization setup
-	-	-	R_RVAPI_TerminateCEU	6.2	CEU termination setup
-	Required	Required	R_RVAPI_PortSettingCEU	6.3	Video input pin setup
-	Required	Required	R_RVAPI_OpenCEU	6.4	Image capturing setup
-	Required	Required	R_RVAPI_CaptureStartCEU	6.5	1-frame capture startup
			R_RVAPI_CaptureStatusCEU	6.6	Capture termination judgment

Note 1: Setup is required when using CEU for the video inputs

Table 4-2 List of Functions

5. Function Reference (VDC5)

5.1 R_RVAPI_InitializeVDC

R_RVAPI_Initialize			
Synopsis	VDC5 initialization clock setup		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_InitializeVDC(`
	`  const vdc_channel_t ch,`
	`  const clock_config_t * const c_cnf);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	clock_config_t * c_cnf	Clock configuration
Return value	VDC_OK:		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_PARAM_EXCEED_RANGE		Out-of-value-range error
	VDC_ERR_PARAM_CONDITION		Unauthorized condition error
	VDC_ERR_RESOURCE_LVDS_CLK		LVDS clock resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.1.1 Description

VDC5 can generate the panel clock from various input clocks as the source clocks. This function is used to set up that clock. Since the panel clock is used to control the display device, it is necessary to set up the clock according to the specifications of the display device to be used.

The following driver is used within this function:

- R_VDC_Initialize()

5.1.2 Parameter details

clock_config_t

The members of the clock_config_t structure are described below.

```
typedef struct
{
    vdc_panel_clksel_t     panel_clk;
    vdc_panel_clk_dcdr_t   panel_clk_div;
    const vdc_lvds_t     * lvds;
} clock_config_t;
```

Type / Member Name	Description
vdc_panel_clksel_t panel_clk	Selects the panel clock. • VDC_PANEL_ICKSEL_IMG (Note 1) Frequency-divided clock for video clock (VIDEO_X1) • VDC_PANEL_ICKSEL_IMG_DV Frequency-divided clock for video clock (DV_CLK) • VDC_PANEL_ICKSEL_EXT_0 (Note 1) Frequency-divided clock for peripheral clock 0 (LCD0_EXTCLK) • VDC_PANEL_ICKSEL_EXT_1 Frequency-divided clock of external clock 1 (LCD1_EXTCLK) • VDC_PANEL_ICKSEL_PERI Frequency-divided clock for peripheral clock 1 (P1φ) • VDC_PANEL_ICKSEL_LVDS: LVDS (Note 1) PLL clock • VDC_PANEL_ICKSEL_LVDS_DIV7 (Note 1) Clock generated by dividing frequency of LVDS PLL by 7
vdc_panel_clk_dcdr_t panel_clk_div	Specifies the clock frequency division ratio • VDC_PANEL_CLKDIV_1_1: 1/1 • VDC_PANEL_CLKDIV_1_2: 1/2 • VDC_PANEL_CLKDIV_1_3: 1/3 • VDC_PANEL_CLKDIV_1_4: 1/4 • VDC_PANEL_CLKDIV_1_5: 1/5 • VDC_PANEL_CLKDIV_1_6: 1/6 • VDC_PANEL_CLKDIV_1_7: 1/7 • VDC_PANEL_CLKDIV_1_8: 1/8 • VDC_PANEL_CLKDIV_1_9: 1/9 • VDC_PANEL_CLKDIV_1_12: 1/12 • VDC_PANEL_CLKDIV_1_16: 1/16 • VDC_PANEL_CLKDIV_1_24: 1/24 • VDC_PANEL_CLKDIV_1_32: 1/32 This parameter is invalid when the panel clock select (panel_icksel) is set to LVDS PLL (VDC_PANEL_ICKSEL_LVDS or VDC_PANEL_ICKSEL_LVDS_DIV7)
const vdc_lvds_t * lvds	LVDS-related parameter (Note 1) Specify NULL if this parameter is not required

Note 1: Configurable only on the RZ/A1H and RZ/A1M

The members of the vdc_lvds_t structure are described below.

```
typedef struct
{
    vdc_lvds_in_clk_sel_t  lvds_in_clk_sel;
    vdc_lvds_ndiv_t        lvds_idiv_set;
    uint16_t               lvdspll_tst;
    vdc_lvds_ndiv_t        lvds_odiv_set;
    vdc_channel_t          lvds_vdc_sel;
    uint16_t               lvdspll_fd;
    uint16_t               lvdspll_rd;
    vdc_lvds_pll_nod_t     lvdspll_od;
} vdc_lvds_t;
```

Type / Member Name	Description
vdc_lvds_in_clk_sel_t lvds_in_clk_sel	Selects the frequency divider 1 input • VDC_LVDS_INCLK_SEL_IMG: VIDEO_X1 • VDC_LVDS_INCLK_SEL_DV_0: DV0_CLK0 • VDC_LVDS_INCLK_SEL_DV_1: DV1_CLK1 • VDC_LVDS_INCLK_SEL_EXT_0: LCD0_EXTCLK • VDC_LVDS_INCLK_SEL_EXT_1: LCD1_EXTCLK • VDC_LVDS_INCLK_SEL_PERI: P1φ
vdc_lvds_ndiv_t lvds_idiv_set	Specifies the frequency divider 1 division ratio NIDIV • VDC_LVDS_NDIV_1: NIDIV = 1 • VDC_LVDS_NDIV_2: NIDIV = 2 • VDC_LVDS_NDIV_4: NIDIV = 4
uint16_t lvdspll_tst	Specifies the LVDS PLL internal parameter Specify 16
vdc_lvds_ndiv_t lvds_odiv_set	Specifies the frequency divider 2 division ratio NODIV • VDC_LVDS_NDIV_1: NODIV = 1 • VDC_LVDS_NDIV_2: NODIV = 2 • VDC_LVDS_NDIV_4: NODIV = 4
vdc_channel_t lvds_vdc_sel	Selects the LVDS VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1
uint16_t lvdspll_fd	Specifies the LVDS PLL feedback ratio NFD NFD = lvdspll_fd (24 to 2047) The following values are invalid, however: 28 to 31, 37 to 39, 46, 47, 55
uint16_t lvdspll_rd	Specifies the LVDS PLL input frequency division ratio NRD NRD = lvdspll_rd + 1 lvdspll_rd (0 to 31)
vdc_lvds_pll_nod_t lvdspll_od	Specifies the LVDS PLL output frequency division ratio NOD • VDC_LVDS_PLL_NOD_1: NOD = 1 • VDC_LVDS_PLL_NOD_2: NOD = 2 • VDC_LVDS_PLL_NOD_4: NOD = 4 • VDC_LVDS_PLL_NOD_8: NOD = 8

Note : Configurable only on the RZ/A1H and RZ/A1M

5.1.3 Setting up the panel clock

An example of VDC5 panel clock configuration is shown in Table 5-1. Since the clock generated by the LVDS's PLL can be used for purposes other than LVDS crystal output, it can be used to generate an arbitrary clock. Examples of VDC5 panel clock configuration using the LVDS's PLL are shown in Table 5-2 and Table 5-3.

Member Name	33.3 [MHz]	22.2 [MHz]
panel_icksel	VDC_LVDS_INCLK_SEL_PERI Peripheral clock 1 (P1φ) 66.6 [MHz]	
panel_dcdr	VDC_PANEL_CLKDIV_1_2	VDC_PANEL_CLKDIV_1_3

Note: Peripheral clock 1 (P1φ) is assumed to be 66.6 [MHz]

Table 5-1 Example of Panel Clock Configuration

Member Name	25.175 [MHz]	40.0 [MHz]	65.0 [MHz]	83.5 [MHz]	85.5 [MHz]
panel_icksel	VDC_PANEL_ICKSEL_LVDS				
lvds_in_clk_sel	VDC_LVDS_INCLK_SEL_PERI Peripheral clock 1 (P1φ) 66.6 [MHz]				
lvds_idiv_set	VDC_LVDS_NDIV_4				
lvds_odiv_set	VDC_LVDS_NDIV_4				
lvdspll_fd	145	384	312	481	82
lvdspll_rd	(3u-1u)	(5u-1u)	(5u-1u)	(6u-1u)	(1u-1u)
lvdspll_od	VDC_LVDS_PLL_NOD_8		VDC_LVDS_PLL_NOD_4		

Note: Configurable only on the RZ/A1H and RZ/A1M

Note: Peripheral clock 1 (P1φ) is assumed to be 66.6 [MHz]

Table 5-2 Example of Panel Clock Configuration Using LVDS PLL (Example 1)

Member Name	25.175 [MHz]	40.0 [MHz]	65.0 [MHz]	83.5 [MHz]	85.5 [MHz]
panel_icksel	VDC_PANEL_ICKSEL_LVDS				
lvds_in_clk_sel	VDC_LVDS_INCLK_SEL_PERI Peripheral clock 1 (P1φ) 64.0 [MHz]				
lvds_idiv_set	VDC_LVDS_NDIV_4				
lvds_odiv_set	VDC_LVDS_NDIV_4				
lvdspll_fd	151	80	65	167	171
lvdspll_rd	(3u-1u)	(1u-1u)	(1u-1u)	(2u-1u)	(2u-1u)
lvdspll_od	VDC_LVDS_PLL_NOD_8		VDC_LVDS_PLL_NOD_4		

Note: Configurable only on the RZ/A1H and RZ/A1M

Note: Peripheral clock 1 (P1φ) is assumed to be 64.0 [MHz]

Table 5-3 Example of Panel Clock Configuration Using LVDS PLL (Example 2)

5.2 R_RVAPI_TerminateVDC

R_RVAPI_TerminateVDC			
Synopsis	VDC5 termination setup		
Header	r_rvapi_vdc.h		
Declaration	`vdc_error_t R_RVAPI_TerminateVDC(` `  const vdc_channel_t ch);`		
Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.2.1 Description

This function performs the VDC5 driver termination processing. It carries out VDC5 interrupt and panel clock disable processing.

The following driver is used within this function:

- R_VDC_Terminate()

5.3 R_RVAPI_DispControlVDC

R_RVAPI_DispControlVDC			
Synopsis	Display output setup		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_DispControlVDC(`
	`  const vdc_channel_t ch,`
	`  const vdc_onoff_t res_vs_sel,`
	`  const qe_config_t * const q_cnf);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_onoff_t res_vs_sel	Selects the vertical sync signal to be output (self-running sync signal). • VDC_OFF (Note 2) The vertical sync video input signal is used as the vertical sync signal for the liquid crystal. • VDC_ON Internally generated self-running vertical sync signal
	[IN]	qe_config_t * q_cnf	Display output configuration
Return value	VDC_OK:		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_EXCEED_RANGE		Out-of-value-range error
	VDC_ERR_RESOURCE_CLK		Clock resource error
	VDC_ERR_RESOURCE_INPUT		Input signal resource error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_PARAM_CONDITION		Unauthorized condition error
	VDC_ERR_RESOURCE_VSYNC		Vertical sync signal resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

Note 2: Must not be configured if no video input is present

5.3.1 Description

This function makes settings with respect to the display output. You can use the settings that are generated by the ‘RZ/A Display Compatible Development Support Tool QE for Display’ of the solution tool kit which runs in the e² studio IDE. Visit the Renesas web site for the ‘RZ/A Display Compatible Development Support Tool QE for Display’. A header file generated by the tool contains macro named VDC5_xxxx. They are treated as VDC_xxxx in the RVAPI header file.

The following driver is used within this function:

- R_VDC_SyncControl()
- R_VDC_DisplayOutput()

5.3.2 Parameter details

qe_config_t

The members of the qe_config_t structure are shown below.

```
typedef struct
{
    uint16_t              vps;
    uint16_t              vpw;
    uint16_t              vs;
    uint16_t              vdp;
    uint16_t              hps;
    uint16_t              hpw;
    uint16_t              hs;
    uint16_t              hdp;
    uint16_t              vtp;
    uint16_t              htp;
    vdc_lcd_tcon_pin_t    tcon_vsync;
    vdc_lcd_tcon_pin_t    tcon_hsync;
    vdc_lcd_tcon_pin_t    tcon_de;
    vdc_sig_pol_t         tcon_vsync_inv;
    vdc_sig_pol_t         tcon_hsync_inv;
    vdc_sig_pol_t         tcon_de_inv;
    uint16_t              tcon_half;
    uint16_t              tcon_ofset;
    vdc_edge_t            lcd_data_out_edge;
    vdc_lcd_outformat_t   lcd_outformat;
} qe_config_t;
```

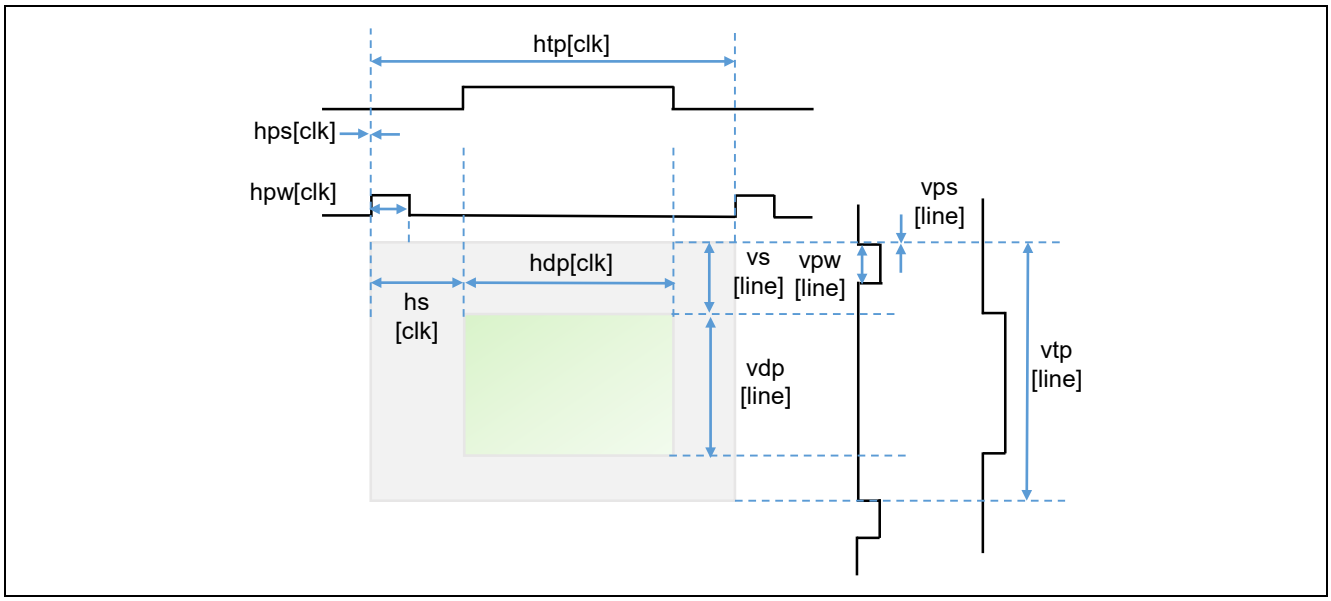


Figure 5-1 Signal Configuration Parameter Diagram

Type / Member Name		Description
uint16_t	vps	Vsync pulse start position [in lines]
uint16_t	vpw	Vsync pulse width [in lines]
uint16_t	vs	Display area vertical start position [in lines]
uint16_t	vdp	Vertical display period [in lines]
uint16_t	hps	Hsync pulse start position [in clks]
uint16_t	hpw	Hsync pulse width [in clks]
uint16_t	hs	Display area horizontal start position [in clks]
uint16_t	hdp	Horizontal display period [in clks]
uint16_t	vtp	Vertical total period [in lines]
uint16_t	htp	Horizontal total period [in clks]
vdc_lcd_tcon_pin_t tcon_vsync vdc_lcd_tcon_pin_t tcon_hsync vdc_lcd_tcon_pin_t tcon_de		LCD TCON output pin select • VDC_LCD_TCON_PIN_NON (-1): No output • VDC_LCD_TCON_PIN_0 (0): LCD_TCON0 is output • VDC_LCD_TCON_PIN_1 (1): LCD_TCON1 is output • VDC_LCD_TCON_PIN_2 (2): LCD_TCON2 is output • VDC_LCD_TCON_PIN_3 (3): LCD_TCON3 is output • VDC_LCD_TCON_PIN_4 (4): LCD_TCON4 is output • VDC_LCD_TCON_PIN_5 (5): LCD_TCON5 is output • VDC_LCD_TCON_PIN_6 (6): LCD_TCON6 is output
vdc_sig_pol_t	tcon_vsync_inv tcon_hsync_inv tcon_de_inv	Horizontal signal operating reference select • VDC_LCD_TCON_REFSEL_HSYNC (0): Horizontal sync signal reference • VDC_LCD_TCON_REFSEL_OFFSET_H (1): Horizontal sync signal reference after offset
uint16_t tcon_half		Specify htp
uint16_t tcon_ofset		Specify 0
vdc_edge_t lcd_data_out_edge		LCD_DATA23 to LCD_DATA0 pin output phase control • VDC_EDGE_RISING: Output on rising edge of LCD_CLK pin signal • VDC_EDGE_FALLING: Output on falling edge of LCD_CLK pin signal
vdc_lcd_outformat_t lcd_outformat		Output format select • VDC_LCD_OUTFORMAT_RGB888 (0): RGB888 • VDC_LCD_OUTFORMAT_RGB666 (1): RGB666 • VDC_LCD_OUTFORMAT_RGB565 (2): RGB565

5.4 R_RVAPI_GraphCreateSurfaceVDC

R_RVAPI_GraphCreateSurfaceVDC			
Synopsis	Display area generation		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_GraphCreateSurfaceVDC(`
	`  const vdc_channel_t ch,`
	`  const gr_surface_disp_config_t * const gr_disp_cnf);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	gr_surface_disp_config_t * gr_disp_cnf	Graphics display area settings
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_LAYER_ID		Invalid layer ID error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_PARAM_EXCEED_RANGE		Out-of-value-range error
	VDC_ERR_PARAM_CONDITION		Unauthorized condition error
	VDC_ERR_RESOURCE_LAYER		Layer resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.4.1 Description

This function make settings for displaying the memory contents allocated in the buffer.

The following driver is used within this function:

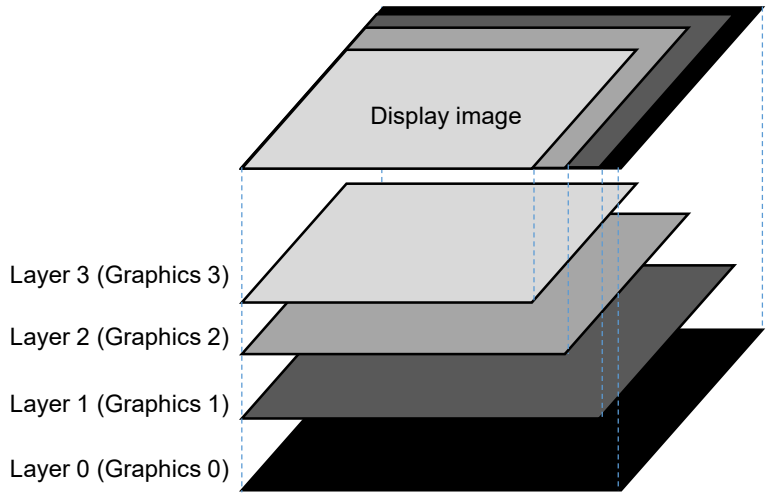
- R_VDC_ReadDataControl()
- R_VDC_CLUT()
- R_VDC_StartProcess()

5.4.2 Parameter details

gr_surface_disp_config_t

The members of the gr_surface_disp_config_t structure are shown below.

```
typedef struct
{
    vdc_layer_id_t         layer_id;
    vdc_pd_disp_rect_t     disp_area;
    void                  *fb_buff;
    uint32_t               fb_stride;
    vdc_gr_format_t        read_format;
    uint32_t              *clut_table;
    vdc_gr_ycc_swap_t      read_ycc_swap;
    vdc_wr_rd_swa_t        read_swap;
    vdc_gr_disp_sel_t      disp_mode;
} gr_surface_disp_config_t;
```



Note: Layer1 (Graphics 1) is configurable only on the RZ/A1H and RZ/A1M

Figure 5-2 Layer Configuration

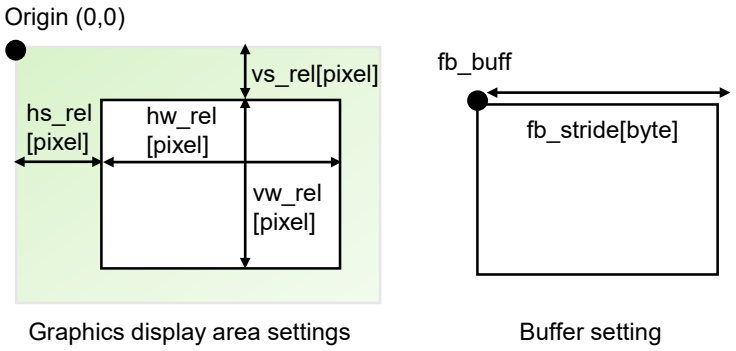


Figure 5-3 Graphics Parameter Diagram

Type/Member Name	Description
vdc_layer_id_t layer_id	Display layer (see Figure 5-2) • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_1_RD (Note 1) • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
vdc_pd_disp_rect_t disp_area	Graphics display area [in pixels] (see Figure 5-3) • disp_area.vs_rel / vw_rel: Vertical display start position / vertical display size • disp_area.hs_rel / hw_rel: Horizontal display start position / horizontal display size vs_rel = hs_rel = 0 causes the display to start at the origin
void * fb_buff	Frame buffer base address (see Figure 5-3) Do not specify NULL
uint32_t fb_stride	Frame buffer line offset address [in bytes] (see Figure 5-3) Specify a multiple of 32 [bytes]
vdc_gr_format_t read_format	Frame buffer read signal format • VDC_GR_FORMAT_RGB565 (0): RGB565 • VDC_GR_FORMAT_ARGB8888 (4): ARGB8888 • VDC_GR_FORMAT_CLUT8 (5): CLUT8 • VDC_GR_FORMAT_CLUT4 (6): CLUT4 • VDC_GR_FORMAT_CLUT1 (7): CLUT1 • VDC_GR_FORMAT_YCBCR422 (8): YCbCr422 (Note 2) • VDC_GR_FORMAT_RGBA8888 (11): RGBA8888
uint32_t * clut_table	Color lookup table This parameter is valid only when the value that is set in read_format is VDC_GR_FORMAT_CLUT8/4/1. Specify the address of the area of a size enough to store as many CLUT data blocks (ARGB8888) as the number of colors. If NULL is selected, the default CLUT data is set up. (Default) CLUT8 (256 colors): CLUT Nos. 0-255 Monochrome (black → white) CLUT4 (16 colors): CLUT Nos. 0-15 Black, red, green, cyan, blue, pink, brown, dark green, lightgoldenrod2, dark blue, violet, gray, orange, white, transparent color CLUT1 (2 colors): CLUT Nos. 0-1 black, white
vdc_gr_ycc_swap_t read_ycc_swap	YCbCr422 format mode buffer read data swap control This parameter is valid only when the value specified in read_format is VDC_GR_FORMAT_YCBCR422 • VDC_GR_YCCSWAP_CBY0CRY1 (0): CbY0/Cr/Y1 • VDC_GR_YCCSWAP_Y0CBY1CR (1): Y0/Cb/Y1/Cr • VDC_GR_YCCSWAP_CRY0CBY1 (2): Cr/Y0/Cb/Y1 • VDC_GR_YCCSWAP_Y0CRY1CB (3): Y0/Cr/Y1/Cb • VDC_GR_YCCSWAP_Y1CRY0CB (4): Y1/Cr/Y0/Cb • VDC_GR_YCCSWAP_CRY1CBY0 (5): Cr/Y1/Cb/Y0 • VDC_GR_YCCSWAP_Y1CBY0CR (6): Y1/Cb/Y0/Cr • VDC_GR_YCCSWAP_CBY1CRY0 (7): Cb/Y1/Cr/Y0
vdc_wr_rd_swa_t read_swap	Makes 8-bit / 16-bit / 32-bit swap setting • VDC_WR_RD_WRSWA_NON (0): No swap 1-2-3-4-5-6-7-8 • VDC_WR_RD_WRSWA_8BIT (1): 8-bit swap 2-1-4-3-6-5-8-7 • VDC_WR_RD_WRSWA_16BIT (2): 16-bit swap 3-4-1-2-7-8-5-6

	• VDC_WR_RD_WRSWA_16_8BIT (3): 16-bit + 8-bit swap 4-3-2-1-8-7-6-5 • VDC_WR_RD_WRSWA_32BIT (4): 32-bit swap 5-6-7-8-1-2-3-4 • VDC_WR_RD_WRSWA_32_8BIT (5): 32-bit + 8-bit swap 6-5-8-7-2-1-4-3 • VDC_WR_RD_WRSWA_32_16BIT (6): 32-bit + 16-bit swap 7-8-5-6-3-4-1-2 • VDC_WR_RD_WRSWA_32_16_8BIT (7): 16-bit + 8-bit swap 8-7-6-5-4-3-2-1
vdc_gr_disp_sel_t disp_mode	Graphics display settings • VDC_DISPSEL_BACK: Background color display • VDC_DISPSEL_LOWER: Lower layer graphics display • VDC_DISPSEL_CURRENT: Current graphics display • VDC_DISPSEL_BLEND : — Blended display of lower layer and current graphics

Note 1: Configurable only on RZ/A1H and RZ/A1M

Note 2: Layer 0 and 1 are configurable on the RZ/A1H and RZ/A1M
On the other platforms, only Layer 0 is configurable

5.5 R_RVAPI_GraphChangeSurfaceVDC

R_RVAPI_GraphChangeSurfaceVDC			
Synopsis	Display area change		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_GraphChangeSurfaceVDC(`
	`  const vdc_channel_t ch,`
	`  const vdc_layer_id_t layer_id,`
	`  void* const fb_buff);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_layer_id_t layer_id	Layer ID • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_1_RD (Note 1) • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
	[IN]	void * framebuff	Frame buffer base address
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_LAYER_ID		Invalid layer ID error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_PARAM_EXCEED_RANGE		Out-of-value-range error
	VDC_ERR_RESOURCE_LAYER		Layer resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.5.1 Description

This function changes the address of the data read buffer.

The following driver is used within this function:

- R_VDC_ChangeReadProcess()

5.6 R_RVAPI_GraphDestroySurfaceVDC

R_RVAPI_GraphDestroySurfaceVDC			
Synopsis	Display area disposal		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_GraphDestroySurfaceVDC(`
	`  const vdc_channel_t ch,`
	`  const vdc_layer_id_t layer_id);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_layer_id_t layer_id	Layer ID • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_1_RD (Note 1) • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_LAYER_ID		Invalid layer ID error
	VDC_ERR_RESOURCE_LAYER		Layer resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.6.1 Description

This function performs stop processing on the specified layer. It stops reading data from the frame buffer and returns the layer's graphics display settings to their initial values.

The following driver is used within this function:

- R_VDC_StopProcess()
- R_VDC_ReleaseDataControl()

5.7 R_RVAPI_DispPortSettingVDC

R_RVAPI_DispPortSettingVDC			
Synopsis	Display output pin setup		
Header	r_rvapi_vdc.h		
Declaration	`void R_RVAPI_DispPortSettingVDC( const vdc_channel_t ch, void (* const port_func)(uint32_t));`		
Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	void (*port_func) (uint32_t)	Pointer of the function to set the display control pins
Return value	None		
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.7.1 Description

The callback function to be set up with this function must configure the pins that are necessary for display output. This function must be called after making all VDC5 display settings as shown in Figure 5-4. A control signal of an unexpected period may be output if pin configuration is made before making display settings.

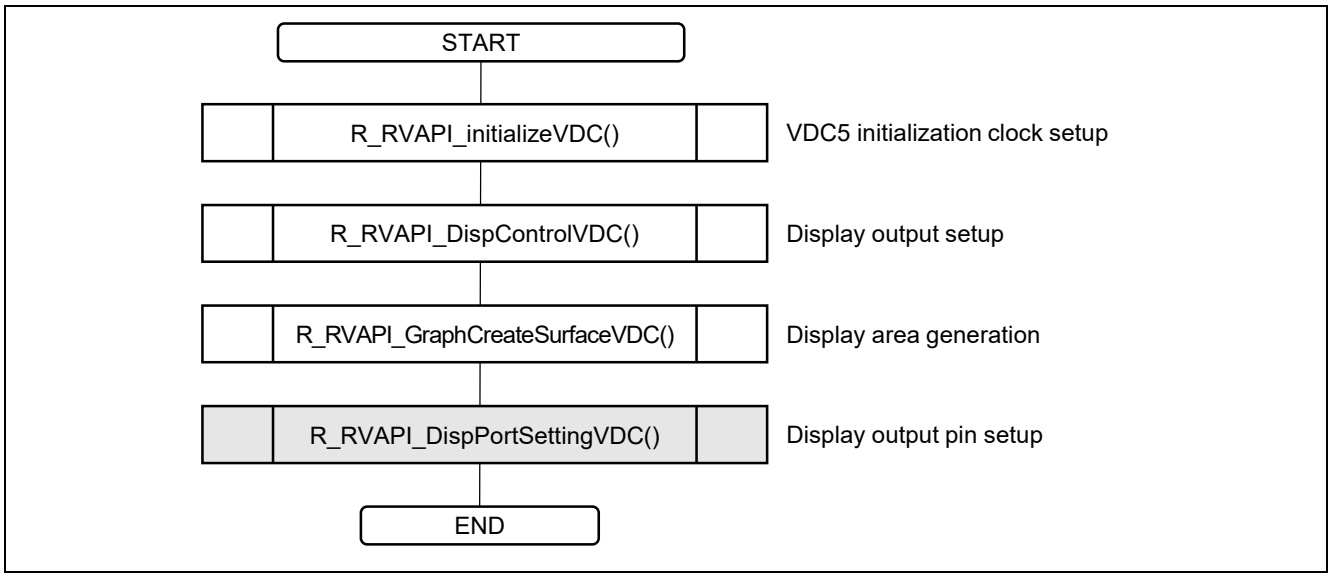


Figure 5-4 Display Output Pin Configuration Timing

5.8 R_RVAPI_VideoControlVDC

R_RVAPI_VideoControlVDC			
Synopsis	Video input setup		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_VideoControlVDC(`
	`  const vdc_channel_t ch,`
	`  const digital_in_t * const digital);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	digital_in_t * digital	Digital video settings Do not specify NULL
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_PARAM_EXCEED_RANGE		Out-of-value-range error
	VDC_ERR_PARAM_CONDITION		Unauthorized condition error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.8.1 Description

This function makes video input settings. For the VDC5, make settings for the digital video input such as that from the CMOS camera.

If there is no digital video input(When the argument ‘digital’ is ‘NULL’), set the analog video input (Note 1).

The following driver is used within this function:

- R_VDC_VideoInput()

5.8.2 Parameter details

digital_in_t

The members of the digital_in_t structure are shown below.

```
typedef struct
{
    vdc_extin_format_t     inp_format;
    vdc_edge_t             inp_pxd_edge;
    vdc_onoff_t            inp_endian_on;
    vdc_onoff_t            inp_swap_on;
    vdc_sig_pol_t          inp_vs_inv;
    vdc_sig_pol_t          inp_hs_inv;
    vdc_extin_ref_hsync_t  inp_h_edge_sel;
    vdc_extin_input_line_t inp_f525_625;
    vdc_extin_h_pos_t      inp_h_pos;
} digital_in_t;
```

Type/Member Name	Description
vdc_extin_format_t inp_format	Selects the format of the external input. • VDC_EXTIN_FORMAT_RGB888 (0): RGB888 • VDC_EXTIN_FORMAT_RGB666 (1): RGB666 • VDC_EXTIN_FORMAT_RGB565 (2): RGB565 • VDC_EXTIN_FORMAT_BT656 (3): BT656 • VDC_EXTIN_FORMAT_BT601 (4): BT601 • VDC_EXTIN_FORMAT_YCBCR422 (5): YCbCr422 • VDC_EXTIN_FORMAT_YCBCR444 (6): YCbCr444
vdc_edge_t inp_pxd_edge	Selects the edge on which the external input video signal DV_DATA is to be sampled into the input stage. • VDC_EDGE_RISING: Rising edge • VDC_EDGE_FALLING: Falling edge
vdc_onoff_t inp_endian_on	Sets the bit endian mode of the external inputs • VDC_OFF • VDC_ON
vdc_onoff_t inp_swap_on	Switches the external input B/R signal • VDC_OFF • VDC_ON
vdc_sig_pol_t inp_vs_inv vdc_sig_pol_t inp_hs_inv	Exercises inversion control of the sync external input signals DV_VSYNC / DV_HSYNC. • VDC_SIG_POL_NOT_INVERTED: Not inverted (positive polarity) • VDC_SIG_POL_INVERTED: Inverted (negative polarity)
vdc_extin_ref_hsync_t inp_h_edge_sel	Selects the reference for the BT656 horizontal sync signal for the external input system. Valid only when inp_format is set to VDC_EXTIN_FORMAT_BT656. • VDC_EXTIN_REF_H_EAV (0): EAV reference • VDC_EXTIN_REF_H_SAV (1): SAV reference
vdc_extin_input_line_t inp_f525_625	Specifies the number of lines for the BT656 input mode for the external input system. Valid only when inp_format is set to VDC_EXTIN_FORMAT_BT656. • VDC_EXTIN_LINE_525 (0): 525 lines

	• VDC_EXTIN_LINE_625 (1): 625 lines
vdc_extin_h_pos_t inp_h_pos	Specifies the data stream start timing with respect to the horizontal sync. The following settings are possible when inp_format is set to VDC_EXTIN_FORMAT_BT656 or VDC_EXTIN_FORMAT_BT601: • VDC_EXTIN_H_POS_CBYCRY (0): Cb/Y/Cr/Y • VDC_EXTIN_H_POS_YCRYCB (1): Y/Cr/Y/Cb • VDC_EXTIN_H_POS_CRYCBY (2): Cr/Y/Cb/Y • VDC_EXTIN_H_POS_YCBYCR (3): Y/Cb/Y/Cr The following settings are possible when inp_format is set to VDC_EXTIN_FORMAT_YCBCR422: • VDC_EXTIN_H_POS_CBYCRY (0): Cb/Y/Cr/Y • VDC_EXTIN_H_POS_YCBYCR (3): Y/Cb/Y/Cr

5.9 R_RVAPI_VideoCreateSurfaceVDC

R_RVAPI_VideoCreateSurfaceVDC

Synopsis: Video and display area generation

Header: r_rvapi_vdc.h

Declaration:

```
vdc_error_t R_RVAPI_VideoCreateSurfaceVDC(
                const vdc_channel_t ch,
                const v_surface_config_t * const v_cnf,
                const v_surface_disp_config_t * const v_disp_cnf);
```

Arguments

[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
[IN]	v_surface_config_t * v_cnf l	Video input area settings Specify NULL when making no video input
[IN]	v_surface_disp_config_t * v_g_cnf	Video input area display settings Specify NULL when making no display

Return value

VDC_OK	Normal termination
VDC_ERR_PARAM_CHANNEL	Channel invalid error
VDC_ERR_PARAM_NULL	NULL specification error
VDC_ERR_PARAM_BIT_WIDTH	Bit width error
VDC_ERR_PARAM_UNDEFINED	Undefined parameter specification error
VDC_ERR_PARAM_EXCEED_RANGE	Out-of-value-range error
VDC_ERR_PARAM_CONDITION	Unauthorized condition error
VDC_ERR_RESOURCE_LVDS_CLK	LVDS clock resource error

Remarks

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.9.1 Description

This function sets as the video input area settings, the video capture timing and buffer write size. It also makes settings for the display of the video input. When performing only video capture, there is no need to make display settings for the video input area.

The following driver is used within this function:

- R_VDC_WriteDataControl()
- R_VDC_ReadDataControl()
- R_VDC_StartProcess()

5.9.2 Parameter details

v_surface_config_t

The members of the v_surface_config_t structure are shown below.

```
typedef struct
{
    vdc_layer_id_t      layer_id;
    vdc_period_rect_t   cap_area;
    void               *fb_buff;
    uint32_t            fb_stride;
    uint32_t            fb_offset;
    uint32_t            fb_num;
    vdc_res_md_t        write_format;
    uint16_t            write_fb_vw;
    uint16_t            write_fb_hw;
    vdc_wr_rd_swa_t     write_swap;
    vdc_wr_md_t         write_rot;
    vdc_res_inter_t     res_inter;
} v_surface_config_t;
```

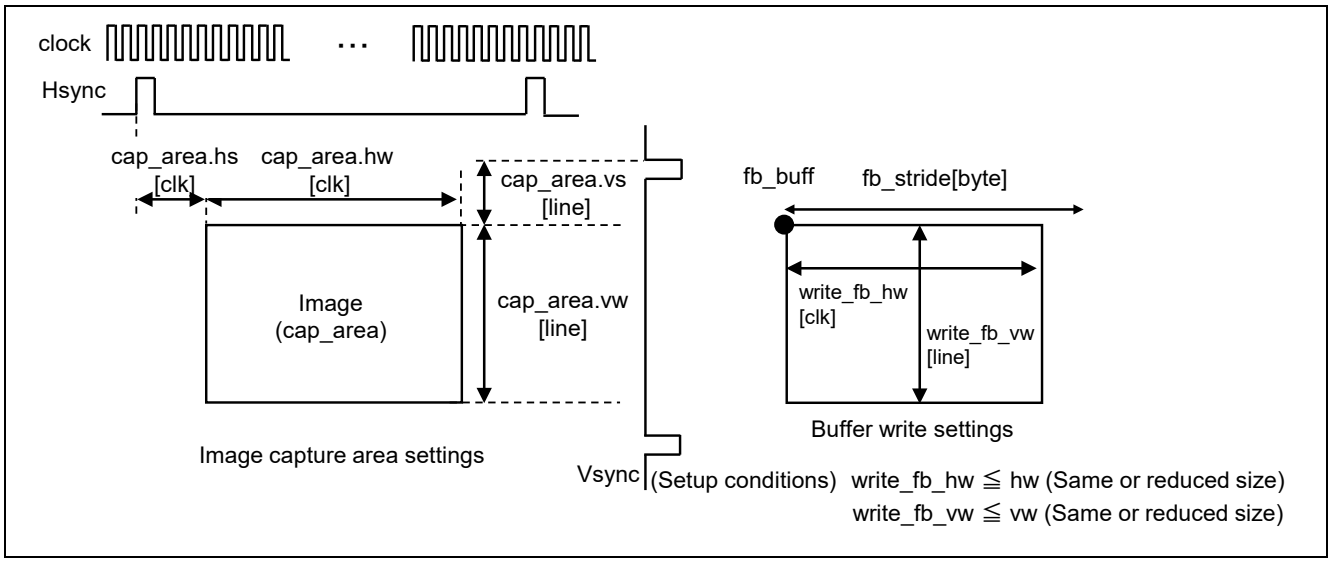


Figure 5-5 Video Input Area Parameter Diagram

Type/Member Name	Description
vdc_layer_id_t layer_id	Layer ID • VDC_LAYER_ID_0_WR • VDC_LAYER_ID_1_WR (Note 1) • VDC_LAYER_ID_OIR_WR (Note 1)
vdc_period_rect_t cap_area	Image capturing range: Horizontal [in clocks] Vertical [in lines] (see Figure 5-5) cap_area.vs / vw: Vertical capture start position/vertical capture size cap_area.hs / hw: Horizontal capture start position/horizontal capture size
void * fb_buff	Frame buffer base address (see Figure 5-5) Specify an address that is aligned on a 32 [byte] boundary
uint32_t fb_stride	Frame buffer line offset address (see Figure 5-5) Specify a multiple of 32 [lines]
uint32_t fb_offset	Frame buffer frame offset address This parameter is invalid when the number of frames is 1 (fb_num is set to '1'). Specify a multiple of 32
uint32_t fb_num	Number of write frame buffer frames Specify 1 or 2.
vdc_res_md_t write_format	Frame buffer write video format • VDC_RES_MD_YCBCR422 (0): YCbCr422 • VDC_RES_MD_RGB565 (1): RGB565 • VDC_RES_MD_RGB888 (2): RGB888 • VDC_RES_MD_YCBCR444 (3): YCbCr444
uint16_t write_fb_vw	Buffer write vertical size [in pixels] 0x0000 to 0x07FF Specify a size that is aligned on a 4 [line] boundary and that is not greater than the value of cap_area.res.vw. Data whose size is equal to or smaller than the specified size is written into the buffer.
uint16_t write_fb_hw	Buffer write horizontal size [in clocks] 0x0000 to 0x07FF Specify a size that is aligned on a 4 [pixel] boundary and that is not greater than the value of cap_area.hw. Data whose size is equal to or smaller than the specified size is written into the buffer.
vdc_wr_rd_swa_t write_swap	8-bit / 16-bit / 32-bit swap setting (Note 2) • VDC_WR_RD_WRSWA_NON (0): No swap 1-2-3-4-5-6-7-8 • VDC_WR_RD_WRSWA_8BIT (1): 8-bit swap 2-1-4-3-6-5-8-7 • VDC_WR_RD_WRSWA_16BIT (2): 16-bit swap 3-4-1-2-7-8-5-6 • VDC_WR_RD_WRSWA_16_8BIT (3): 16-bit + 8-bit swap 4-3-2-1-8-7-6-5 • VDC_WR_RD_WRSWA_32BIT (4): 32-bit swap 5-6-7-8-1-2-3-4 • VDC_WR_RD_WRSWA_32_8BIT (5): 32-bit + 8-bit swap 6-5-8-7-2-1-4-3 • VDC_WR_RD_WRSWA_32_16BIT (6): 32-bit + 16-bit swap 7-8-5-6-3-4-1-2 • VDC_WR_RD_WRSWA_32_16_8BIT (7): 32-bit + 16-bit + 8-bit swap 8-7-6-5-4-3-2-1
vdc_wr_rd_swa_t write_swap	Frame buffer writing mode for image processing • VDC_WR_MD_NORMAL (0): Normal • VDC_WR_MD_MIRROR (1): Horizontal mirroring • VDC_WR_MD_ROT_90DEG (2): 90-degree rotation • VDC_WR_MD_ROT_180DEG (3): 180-degree rotation

	• VDC_WR_MD_ROT_270DEG (4): 270-degree rotation Setting this parameter to 90-degree, 180-degree, or 270-degree rotation is valid only when frame buffer video-signal writing format (write_format) is set to YCbCr422 or RGB565.
vdc_res_inter_t res_inter	Specifies the field operation mode. • VDC_RES_INTER_PROGRESSIVE (0): Progressive • VDC_RES_INTER_INTERLACE (1): Interlace

Note 1: These layers are configurable only on the RZ/A1H and RZ/A1M.
Note 2: When write_format is set to YCbCr422 or RGB565, be sure to specify a 0 (no swap).

v_surface_disp_config_t

The members of the v_surface_disp_config_t structure are shown below.

```
typedef struct
{
    vdc_period_rect_t   disp_area;
    vdc_gr_ycc_swap_t   read_ycc_swap;
    vdc_wr_rd_swa_t     read_swap;
} v_surface_disp_config_t;
```

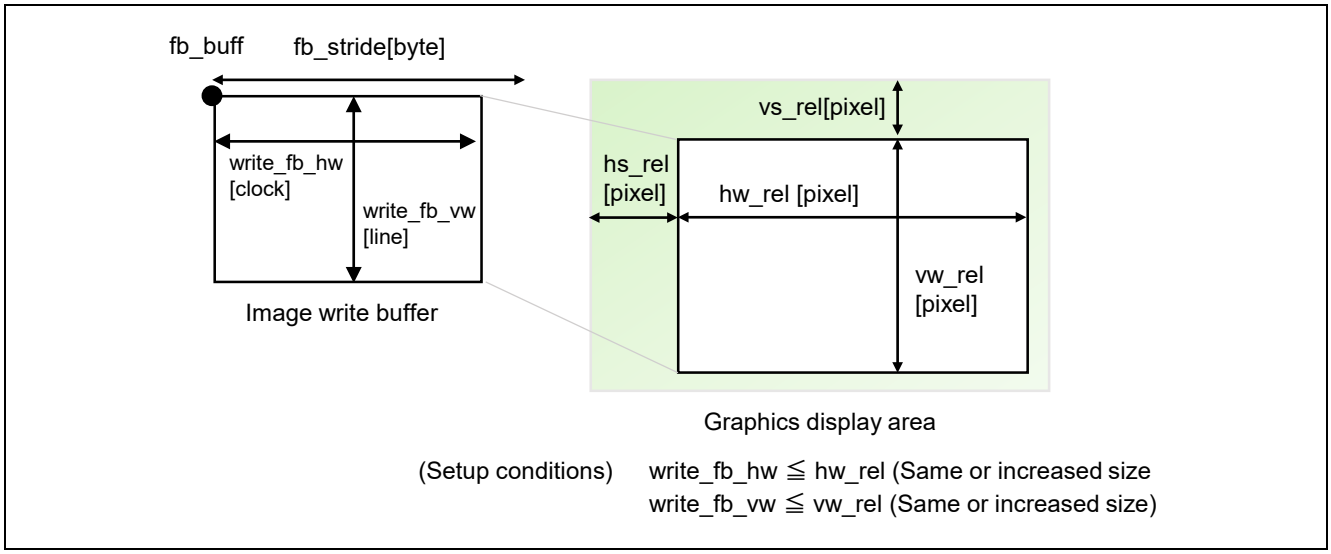


Figure 5-6 Video Input Area Display Parameter Diagram

Type/Member Name	Description
vdc_pd_disp_rect_t disp_area	Graphics display area [in pixels] (see Figure 5-6) • disp_area.vs_rel / vw_rel: Vertical display start position / vertical display size • disp_area.hs_rel / hw_rel: Horizontal display start position / horizontal display size
vdc_gr_ycc_swap_t read_ycc_swap	YCbCr422 format mode buffer read data swap control This parameter is valid only when the value specified in read_format is VDC_GR_FORMAT_YCBCR422. • VDC_GR_YCCSWAP_CBY0CRY1 (0): CbY0/Cr/Y1 • VDC_GR_YCCSWAP_Y0CBY1CR (1): Y0/Cb/Y1/Cr • VDC_GR_YCCSWAP_CRY0CBY1 (2): Cr/Y0/Cb/Y1 • VDC_GR_YCCSWAP_Y0CRY1CB (3): Y0/Cr/Y1/Cb • VDC_GR_YCCSWAP_Y1CRY0CB (4): Y1/Cr/Y0/Cb • VDC_GR_YCCSWAP_CRY1CBY0 (5): Cr/Y1/Cb/Y0 • VDC_GR_YCCSWAP_Y1CBY0CR (6): Y1/Cb/Y0/Cr • VDC_GR_YCCSWAP_CBY1CRY0 (7): Cb/Y1/Cr/Y0
vdc_wr_rd_swa_t read_swap	Makes 8-bit / 16-bit / 32-bit swap setting. • VDC_WR_RD_WRSWA_NON (0): No swap 1-2-3-4-5-6-7-8 • VDC_WR_RD_WRSWA_8BIT (1): 8-bit swap 2-1-4-3-6-5-8-7 • VDC_WR_RD_WRSWA_16BIT (2): 16-bit swap 3-4-1-2-7-8-5-6 • VDC_WR_RD_WRSWA_16_8BIT (3): 16-bit + 8-bit swap 4-3-2-1-8-7-6-5 • VDC_WR_RD_WRSWA_32BIT (4): 32-bit swap 5-6-7-8-1-2-3-4 • VDC_WR_RD_WRSWA_32_8BIT (5): 32-bit + 8-bit swap 6-5-8-7-2-1-4-3 • VDC_WR_RD_WRSWA_32_16BIT (6): 32-bit + 16-bit swap 7-8-5-6-3-4-1-2 • VDC_WR_RD_WRSWA_32_16_8BIT (7): 32-bit +16-bit + 8-bit swap 8-7-6-5-4-3-2-1

5.9.3 About the configuration of the video capture range

Examples of video capture range configuration are summarized in Table 5-4.

(Example of digital input)

VGA (640 x 480) size progressive input

Writing VGA (640 x 480) size input to buffer in YCbCr422 format with no reduction

The display size is increased from VGA (640 x 480) to SVGA (800 x 600).

Structure Name	Member Name	Digital input 24/18/16 bit I/F	Digital input 8-bit I/F
digital_in_t	inp_format	RGB888/666/565 YCbCr422/444	BT6556 BT601
v_surface _config_t	layer_id	VDC_LAYER_ID_0_WR	
	cap_area.vs	Arbitrary	
	cap_area vw	480u	
	cap_area.hs	Arbitrary	
	cap_area.hw	640u x 1u 1 [pixel] / 1 [clock]	640u x 2u (Note 1) 1 [pixel] / 2 [clock]
	fb_buff	Internal RAM area	
	fb_stride	640u x 2u (as per YCbCr422)	
	fb_num	2 planes	
	write_format	YCbCr422	
	write_fb_vw	480u	
	write_fb_hw	640u	640u (Note 2)
	res_inter	Progressive	
	fb_offset	Buffer offset	
v_surface _disp_config_t	disp_area.vs_rel	0u	
	disp_area.vw_rel	800u (640u if equal size)	
	disp_area.hs_rel	0u	
	disp_area.hw_rel	600u (480u if equal size)	

Note 1: The capture width clock differs according to the I/F for the external input (1 [pixel] / 1 [clock] and 1 [pixel] / 2 [clocks]).

Note 2: Horizontal reduction is required for BT.656/601 because the same image data is captured twice as per VDC5 specifications. 640u, which is the half of the buffer write setting (write_fb_hw), is set for the capture width clock (cap_area.hw = 640u x 2u).

Table 5-4 Examples of Video Capture Range Configuration

5.10 R_RVAPI_VideoDestroySurfaceVDC

R_RVAPI_VideoDestroySurfaceVDC			
Synopsis	Video and display area cancellation		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_VideoDestroySurfaceVDC(`
	`  const vdc_channel_t ch,`
	`  const vdc_layer_id_t layer_id);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_layer_id_t layer_id	Layer ID • VDC_LAYER_ID_0_WR • VDC_LAYER_ID_1_WR (Note 1)
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_PARAM_EXCEED_RANGE		Out-of-value-range error
	VDC_ERR_PARAM_CONDITION		Unauthorized condition error
	VDC_ERR_RESOURCE_LVDS_CLK		LVDS clock resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.10.1 Description

This function performs stop processing on the specified layer. It stops reading data from the frame buffer and returns the layer's graphics display settings to their initial values.

The following driver is used within this function:

- R_VDC_StopProcess()
- R_VDC_ReleaseDataControl()

5.11 R_RVAPI_VideoPortSettingVDC

R_RVAPI_VideoPortSettingVDC			
Synopsis	Video input pin setup		
Header	r_rvapi_vdc.h		

Declaration	`void R_RVAPI_VideoPortSettingVDC(`
	`const vdc_channel_t ch,`
	`void (* const port_func)(uint32_t));`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	void (* const port_func) (uint32_t)	Pointer of function to set the video input pins
Return value	None		
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.11.1 Description

The callback function to be set up with this function must configure the video input pins. This function must have been called by the time the video area is generated as shown in Figure 5-7.

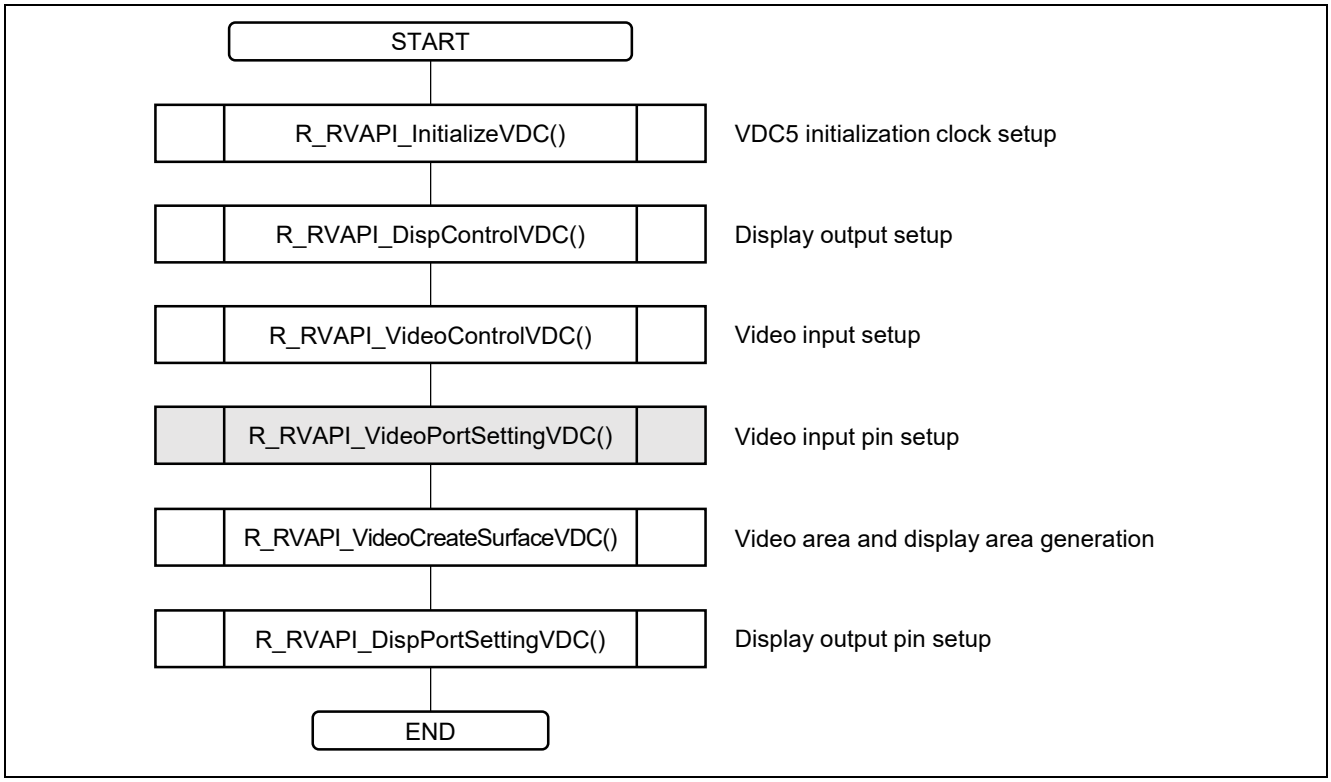


Figure 5-7 Timing of Configuring the Video Input Pins

5.12 R_RVAPI_InterruptEnableVDC

R_RVAPI_InterruptEnableVDC			
Synopsis	VDC5 interrupt enable setup		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_InterruptEnableVDC(`
	`const vdc_channel_t ch,`
	`const vdc_int_type_t flag,`
	`const uint16_t line_num,`
	`void (* const callback)(vdc_int_type_t int_type,`
	`uint32_t buff));`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_int_type_t flag	VDC5 interrupt type
	[IN]	uint16_t line_num	Sets up the line interrupt Valid only for VDC_INT_TYPE_VLINE
	[IN]	void (*callback) (vdc_int_type_t, void * buff)	Interrupt callback function pointer
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_RESOURCE_CLK:		Clock resource error
	VDC_ERR_RESOURCE_VSYNC		Vertical sync signal resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.12.1 Description

This function enables the interrupts of the VDC5 interrupt types described in Table 5-5 and registers the specified callback function.

The following driver is used within this function:

- R_VDC_CallbackISR()

5.12.2 Parameter details

The VDC5 interrupt types are listed in Table 5-5.

Enumeration Constant	Value	Description
VDC_INT_TYPE_S0_VI_VSYNC	0	Vertical sync signal input to scaling 0
VDC_INT_TYPE_S0_LO_VSYNC	1	Vertical sync signal output from scaling 0
VDC_INT_TYPE_S0_VSYNCERR	2	Missing vertical sync signal of scaling 0
VDC_INT_TYPE_VLINE	3	Graphics (3) panel output designation line signal
VDC_INT_TYPE_S0_VFIELD	4	End of field signal of the scaling 0 record function
VDC_INT_TYPE_IV1_VBUFERR	5	Scaling 0 frame buffer write overflow signal
VDC_INT_TYPE_IV3_VBUFERR	6	Graphics (0) frame buffer read underflow signal
VDC_INT_TYPE_IV5_VBUFERR	7	Graphics (2) frame buffer read underflow signal
VDC_INT_TYPE_IV6_VBUFERR	8	Graphics (3) frame buffer read underflow signal
VDC_INT_TYPE_S1_VI_VSYNC (Note 1)	10	Vertical sync signal input to scaling 1
VDC_INT_TYPE_S1_LO_VSYNC (Note 1)	11	Vertical sync signal output from scaling 1
VDC_INT_TYPE_S1_VSYNCERR (Note 1)	12	Missing vertical sync signal of scaling 1
VDC_INT_TYPE_S1_VFIELD (Note 1)	13	End of field signal of the scaling 1 record function
VDC_INT_TYPE_IV2_VBUFERR (Note 1)	14	Scaling 1 frame buffer write overflow signal
VDC_INT_TYPE_IV4_VBUFERR (Note 1)	15	Graphics (1) frame buffer read underflow signal
VDC_INT_TYPE_OIR_VI_VSYNC (Note 1)	17	Vertical sync signal input to the output image generator
VDC_INT_TYPE_OIR_LO_VSYNC (Note 1)	18	Vertical sync signal output from the output image generator
VDC_INT_TYPE_OIR_VLINE (Note 1)	19	Output image generator panel output designation line signal
VDC_INT_TYPE_OIR_VFIELD (Note 1)	20	End of field signal of the output image generator record function
VDC_INT_TYPE_IV7_VBUFERR (Note 1)	21	Output image generator frame buffer write overflow signal
VDC_INT_TYPE_IV8_VBUFERR (Note 1)	22	Graphics (OIR) frame buffer read underflow signal

Note 1: Configurable only on the RZ/A1H and RZ/A1M

Table 5-5 VDC5 Interrupt Type

5.13 R_RVAPI_InterruptDisableVDC

R_RVAPI_InterruptDisableVDC			
Synopsis	VDC5 interrupt disable setup		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_InterruptDisableVDC(`
	`  const vdc_channel_t ch,`
	`  const vdc_int_type_t flag);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_int_type_t flag	VDC5 interrupt type
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_RESOURCE_CLK		Clock resource error
	VDC_ERR_RESOURCE_VSYNC		Vertical sync signal resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.13.1 Description

This function disables the interrupts of the VDC5 interrupt types described in Table 5-5.

The following driver is used within this function:

- R_VDC_CallbackISR()

5.14 R_RVAPI_AlphablendingRectVDC

R_RVAPI_AlphablendingRectVDC				
Synopsis	Rectangle alpha blend			
Header	r_rvapi_vdc.h			

Declaration	`vdc_error_t R_RVAPI_AlphablendingRectVDC(`
	`  const vdc_channel_t ch,`
	`  const vdc_layer_id_t layer_id,`
	`  const vdc_onoff_t alpha_onoff,`
	`  const vdc_pd_disp_rect_t * const alpha_area,`
	`  const uint8_t alpha_value);`			

Arguments	[IN]	vdc_channel_t	ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_layer_id_t	layer_id,	Layer ID • VDC_LAYER_ID_1_RD (Note 1) • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
	[IN]	vdc_onoff_t	alpha_onoff	Rectangle alpha blend ON / OFF setting • VDC_ON • VDC_OFF
	[IN]	vdc_pd_disp_rect_t *	alpha_area	Rectangle alpha blend area [in pixels]
	[IN]	uint8_t	alpha_value	Alpha value (0 to 255) 0: Perfect transparency
Return value	VDC_OK			Normal termination
	VDC_ERR_PARAM_CHANNEL			Channel invalid error
	VDC_ERR_PARAM_LAYER_ID			Invalid layer ID error
	VDC_ERR_PARAM_BIT_WIDTH			Bit width error
	VDC_ERR_PARAM_EXCEED_RANGE			Out-of-value-range error
	VDC_ERR_IF_CONDITION			Interface condition error
	VDC_ERR_RESOURCE_LAYER			Layer resource error
Remarks				

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.14.1 Description

This function turns on and off rectangular area alpha blending, sets up a rectangular area, and sets an alpha value. The following driver is used within this function:

- R_VDC_AlphaBlendingRect()

5.15 R_RVAPI_ChromakeyVDC

R_RVAPI_ChromakeyVDC			
Synopsis	Transparency using chroma key		
Header	r_vapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_ChromakeyVDC(`
	`  const vdc_channel_t ch,`
	`  const vdc_layer_id_t layer_id,`
	`  const vdc_onoff_t gr_ck_on,`
	`  const uint32_t ck_color,`
	`  const uint8_t rep_alpha);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_layer_id_t layer_id,	Layer ID • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_1_RD (Note 1) • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
	[IN]	vdc_onoff_t gr_ck_on	Chroma key ON / OFF setting • VDC_ON • VDC_OFF
	[IN]	uint32_t ck_color	Color signal subject to chroma keying Specify with the color format that is used for the target layer (LSB justified)
	[IN]	uint8_t rep_alpha	Alpha value after chroma key replacement (0 to 255)
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_LAYER_ID		Invalid layer ID error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_IF_CONDITION		Interface condition error
	VDC_ERR_RESOURCE_LAYER		Layer resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.15.1 Description

This function turns on and off chroma keying and sets the color signal to be subjected to chroma keying and a post-replacement alpha value. The following driver is used within this function:

- R_VDC_Chromakey()

5.16 R_RVAPI_DispCalibrationVDC

R_RVAPI_DispCalibrationVDC			
Synopsis	Screen output calibration processing		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_DispCalibrationVDC(`
	`const vdc_channel_t ch,`
	`const vdc_calibr_route_t route,`
	`const vdc_calibr_bright_t * const bright,`
	`const vdc_calibr_contrast_t * const contrast,`
	`const vdc_calibr_dither_t * const panel_dither);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_calibr_route_t route	Calibration circuit sequence control • VDC_CALIBR_ROUTE_BCG • Brightness ⇒ Contrast ⇒ Gamma calibration • VDC_CALIBR_ROUTE_GBC • Gamma calibration ⇒ Brightness ⇒ Contrast
	[IN]	vdc_calibr_bright_t * bright	Brightness (DC) adjustment parameter Specify NULL if there is no need to change
	[IN]	vdc_calibr_contrast_t * contrast	Contrast (gain) adjustment parameter Specify NULL if there is no need to change
	[IN]	vdc_calibr_dither_t * panel_dither	Panel dithering parameter Specify NULL if there is no need to change
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_RESOURCE_OUTPUT		Output resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.16.1 Description

This function makes settings for panel brightness, contrast adjustment, panel dithering, and panel output calibration circuit control. The settings made by this function remain valid until a hardware reset is effected or they are overwritten by other settings made through this function.

The following driver is used within this function:

- R_VDC_DisplayCalibration()

5.16.2 Parameter details

vdc_calibr_bright_t

The members of the vdc_calibr_bright_t structure are shown below.

```
typedef struct
{
    uint16_t    pbrt_g;
    uint16_t    pbrt_b;
    uint16_t    pbrt_r;
} vdc_calibr_bright_t;
```

Type / Member Name	Initial Value	Description
uint16_t pbrt_g	512	G signal brightness (DC) adjustment 0x0000 (-512) to 0x03FF (+511)
uint16_t pbrt_b	512	B signal brightness (DC) adjustment 0x0000 (-512) to 0x03FF (+511)
uint16_t pbrt_r	512	R signal brightness (DC) adjustment 0x0000 (-512) to 0x03FF (+511)

vdc_calibr_contrast_t

The members of the vdc_calibr_contrast_t structure are shown below.

```
typedef struct
{
    uint8_t     cont_g;
    uint8_t     cont_b;
    uint8_t     cont_r;
} vdc_calibr_contrast_t;
```

Type/Member Name	Initial Value	Description
uint8_t cont_g	128	G signal contrast (gain) adjustment 0x0000 (0 / 128 [times]) to 0x00FF (255 / 128 [times])
uint8_t cont_b	128	B signal contrast (gain) adjustment 0x0000 (0 / 128 [times]) to 0x00FF (255 / 128 [times])
uint8_t cont_r	128	R signal contrast (gain) adjustment 0x0000 (0 / 128 [times]) to 0x00FF (255 / 128 [times])

vdc_calibr_dither_t

The members of the vdc_calibr_dither_t structure are shown below.

```
typedef struct
{
    vdc_panel_dither_md_t   pdth_sel;
    uint8_t                 pdth_pa;
    uint8_t                 pdth_pb;
    uint8_t                 pdth_pc;
    uint8_t                 pdth_pd;
} vdc_calibr_dither_t;
```

Type/Member Name	Initial Value	Description
vdc_panel_dither_md_t pdth_sel	0	Panel dithering mode • VDC_PDTH_MD_TRU (0): Truncation • VDC_PDTH_MD_RDOF (1): Rounding • VDC_PDTH_MD_2X2 (2): 2x2 pattern dithering • VDC_PDTH_MD_RAND (3): Random pattern dithering
uint8_t pdth_pa	3	2x2 pattern dithering pattern value 0 to 3 Referenced only when pdth_sel is set to VDC_PDTH_MD_2X2
uint8_t pdth_pb	0	2x2 pattern dithering pattern value (B) 0 to 3 Referenced only when pdth_sel is set to VDC_PDTH_MD_2X2
uint8_t pdth_pc	2	2x2 pattern dithering pattern value (C) 0 to 3 Referenced only when pdth_sel is set to VDC_PDTH_MD_2X2
uint8_t pdth_pd	1	2x2 pattern dithering pattern value (D) 0 to 3 Referenced only when pdth_sel is set to VDC_PDTH_MD_2X2

5.17 R_RVAPI_DispGammaVDC

R_RVAPI_DispGammaVDC			
Synopsis	Gamma calibration setup		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_DispGammaVDC(`
	`const vdc_channel_t ch,`
	`const vdc_onoff_t gam_on,`
	`const uint16_t * const gam_r_gain,`
	`const uint8_t * const gam_r_th,`
	`const uint16_t * const gam_g_gain,`
	`const uint8_t * const gam_g_th,`
	`const uint16_t * const gam_b_gain,`
	`const uint8_t * const gam_b_th);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_onoff_t gam_on	Gamma correction ON / OFF setting • VDC_ON • VDC_OFF
	[IN]	uint16_t * gam_r_gain,	Gain adjustment for the R signal areas 0 to 31 Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [time])
	[IN]	uint8_t * gam_r_th	Starting threshold value for the R signal areas 1 to 31 Unsigned (0 to 255 [LSB])
	[IN]	uint16_t * gam_g_gain	Gain adjustment for the G signal areas 0 to 31 Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [time])
	[IN]	uint8_t * gam_g_th	Starting threshold value for the G signal areas 1 to 31 Unsigned (0 to 255 [LSB])
	[IN]	uint16_t * gam_b_gain	Gain adjustment for the B signal areas 0 to 31 Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [time])
	[IN]	uint8_t * gam_b_th	Starting threshold value for the B signal areas 1 to 31 Unsigned (0 to 255 [LSB])
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_RESOURCE_OUTPUT		Output resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.17.1 Description

This function turns on and off gamma calibration and sets the gamma calibration values and gamma calibration starting threshold values of the G / B / R signals. For gamma calibration processing, the user can configure gamma calibration ON / OFF control and gamma calibration parameter setup separately. The gamma calibration parameter values, once set, is valid until a hardware reset is effected or they are overwritten by other settings.

The following driver is used within this function:

- R_VDC_GammaCorrection()

5.18 R_RVAPI_VideoCalibrationVDC

R_RVAPI_VideoCalibrationVDC			
Synopsis	Color matrix setup		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_VideoCalibrationVDC(`
	`  const vdc_channel_t ch,`
	`  const vdc_color_matrix_t * const color_matrix);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_color_matrix_t * color_matrix	Color matrix setup parameter
Return value	VDC_OK:		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_PARAM_CONDITION		Unauthorized condition error
	VDC_ERR_RESOURCE_LAYER		Layer resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.18.1 Description

This function sets up the specified color matrix. This color matrix is used to adjust the contrast and brightness of the video input.

The following driver is used within this function:

- R_VDC_ImageColorMatrix()

5.18.2 Parameter details

vdc_color_matrix_t

The members of the vdc_color_matrix_t structure are shown below.

```
typedef struct
{
    vdc_colormtx_module_t   module;
    vdc_colormtx_mode_t     mtx_mode;
    uint16_t                offset[VDC_COLORMTX_OFFST_NUM];
    uint16_t                gain[VDC_COLORMTX_GAIN_NUM];
} vdc_color_matrix_t;
```

Type / Member Name	Description
vdc_colormtx_module_t module	Selects the module to be subjected to color matrix setup. • Input controller • VDC_COLORMTX_ADJ_0 (1): Image quality enhancer 0 • VDC_COLORMTX_ADJ_1 (2): Image quality enhancer 1 (Note 1)
vdc_colormtx_mode_t mtx_mode	Specifies the color matrix operating mode. • VDC_COLORMTX_GBR_GBR:GBR ⇒ GBR • VDC_COLORMTX_GBR_YCBCR:GBR ⇒ YCbCr (Note 2) • VDC_COLORMTX_YCBCR_GBR:YCbCr ⇒ GBR • VDC_COLORMTX_YCBCR_YCBCR: YCbCr ⇒ YCbCr (Note 2)
uint16_t offset[VDC_COLORMTX_OFFST_NUM]	Y / G, B, and R signal offset (DC) adjustment 0x0000 (-128) to 0x0080 (0) to 0x00FF (+127)
uint16_t gain[VDC_COLORMTX_GAIN_NUM]	GG, GB, GR, BG, BB, BR, RG, RB, and RR gain adjustment Signed (2's complement) -1024 to +1023 [LSB], 256 [LSB] = 1.0 [times]

Note 1: Configurable only on the RZ/A1H and RZ/A1M

Note 2: The operating mode in which conversion to YCbCr is performed is made available only when the input controller (VDC_COLORMTX_IMGCNT) is specified in module.

5.19 R_RVAPI_VideoSharpnessLtiVDC

R_RVAPI_VideoSharpnessLtiVDC			
Synopsis	Image enhancement processing		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_VideoSharpnessLtiVDC(`
	`const vdc_channel_t ch,`
	`const vdc_imgimprv_id_t imgimprv_id,`
	`const vdc_onoff_t shp_h_on,`
	`const vdc_enhance_sharp_t * const sharp_param,`
	`const vdc_onoff_t lti_h_on,`
	`const vdc_enhance_lti_t * const lti_param,`
	`const vdc_period_rect_t * const enh_area);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_imgimprv_id_t imgimprv_id	image quality enhancer ID • VDC_IMG_IMPRV_0: Image quality enhancer 0 • VDC_IMG_IMPRV_1: Image quality enhancer 1 (Note 1)
	[IN]	vdc_onoff_t shp_h_on	Sharpness ON / OFF setting
	[IN]	vdc_enhance_sharp_t * sharp_param	Sharpness parameter
	[IN]	vdc_onoff_t lti_h_on	LTI ON / OFF setting
	[IN]	vdc_enhance_lti_t * lti_param	LTI parameter
	[IN]	vdc_period_rect_t * enh_area	Image quality enhancement area parameter
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_PARAM_EXCEED_RANGE		Out-of-value-range error
	VDC_ERR_IF_CONDITION		Interface condition error
	VDC_ERR_RESOURCE_LAYER		Layer resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.19.1 Description

This function sets up the sharpness ON / OFF setting and sharpness parameters, LTI ON / OFF setting and LTI parameters, and the rectangular area where sharpness and LTI are to be applied.

The following driver is used within this function:

- R_VDC_ImageEnhancement()

5.19.2 Parameter details

vdc_enhance_sharp_t

The members of the vdc_enhance_sharp_t structure are shown below.

```
typedef struct
{
    vdc_onoff_t             shp_h2_lpf_sel;
    vdc_sharpness_ctrl_t    hrz_sharp[VDC_IMGENH_SHARP_NUM];
} vdc_enhance_sharp_t;
```

Type / Member Name	Initial Value	Description
vdc_onoff_t shp_h2_lpf_sel	VDC_OFF (0)	Selects the LPF to be used for fold removal before H2 edge detection. • VDC_OFF: Without LPF • VDC_ON: With LPF
vdc_sharpness_ctrl_t hrz_sharp [VDC_IMGENH_SHARP_NUM]	-	Sharpness control parameter Horizontal sharpness (H1, H2, H3)

vdc_sharpness_ctrl_t

The members of the vdc_sharpness_ctrl_t structure are shown below.

```
typedef struct
{
    uint8_t     shp_clip_o;
    uint8_t     shp_clip_u;
    uint8_t     shp_gain_o;
    uint8_t     shp_gain_u;
    uint8_t     shp_core;
} vdc_sharpness_ctrl_t;
```

Type / Member Name	Initial Value	Description
uint8_t shp_clip_o	0	Sharpness correction value clip (overshoot side) 0x0000 to 0x00FF
uint8_t shp_clip_u	0	Sharpness correction value clip (undershoot side) 0x0000 to 0x00FF
uint8_t shp_gain_o	0	Specifies the gain for sharpness edge amplitude value (overshoot side) 0x0000 (0 time) to 0x0040 (1 time) to 0x00FF (approx. 4 times)
uint8_t shp_gain_u	0	Specifies the gain for sharpness edge amplitude value (undershoot side) 0x0000 (0 time) to 0x0040 (1 time) to 0x00FF (approx. 4 times)
uint8_t shp_core	0	Specifies the active sharpness area. 0x0000 to 0x007F

vdc_enhance_lti_t

The members of the vdc_enhance_lti_t structure are shown below.

```
typedef struct
{
    vdc_onoff_t             lti_h2_lpf_sel;
    vdc_lti_mdfil_sel_t     lti_h4_median_tap_sel;
    vdc_lti_ctrl_t          lti[VDC_IMGENH_LTI_NUM];
} vdc_enhance_lti_t;
```

Type / Member Name	Initial Value	Description
vdc_onoff_t lti_h2_lpf_sel	VDC_OFF(0)	Selects the LPF to be used for fold removal before H2 edge detection. • VDC_OFF: Without LPF • VDC_ON: With LPF
vdc_lti_mdfil_sel_t lti_h4_median_tap_sel	0	Selects the median filter pixel to be referenced • VDC_LTI_MDFIL_SEL_ADJ2 (0): Reference to 2 adjacent pixels • VDC_LTI_MDFIL_SEL_ADJ1 (1): Reference to 1 adjacent pixel
vdc_lti_ctrl_t lti [VDC_IMGENH_LTI_NUM]	-	LTI control parameter Horizontal LTI (H2, H4)

vdc_lti_ctrl_t

The members of the vdc_lti_ctrl_t structure are shown below.

```
typedef struct
{
    uint8_t     lti_inc_zero;
    uint8_t     lti_gain;
    uint8_t     lti_core;
} vdc_lti_ctrl_t;
```

Type / Member Name	Initial Value	Description
uint8_t lti_inc_zero	10	Specifies the LTI correction threshold for the median filter 0x0000 to 0x00FF
uint8_t lti_gain	0	Specifies the gain for the LTI edge amplitude value. 0x0000 (0 time) to 0x0040 (1 time) to 0x00FF (approx. 4 times)
uint8_t lti_core	0	LTI coring 0x0000 to 0x00FF

vdc_period_rect_t

The members of the vdc_period_rect_t structure are shown below.

```
typedef struct
{
    uint16_t    vs;
    uint16_t    vw;
    uint16_t    hs;
    uint16_t    hw;
} vdc_period_rect_t;
```

Type / Member Name	Initial Value	Description
uint16_t vs	0	Specifies the start position of the effective vertical image area in the enhancer effective area (in lines) Specify 2 lines or more
uint16_t vw	0	Specifies the width of the effective vertical image area in the enhancer effective area (in lines)
uint16_t hs	0	Specifies the start position of the effective horizontal image area in the enhancer effective area (in clocks) Specify 4 clocks or more
uint16_t hw	0	Specifies the width of the effective horizontal image area in the enhancer effective area (in clocks)

5.20 R_RVAPI_GraphChangeSurfaceConfigVDC

R_RVAPI_GraphChangeSurfaceConfigVDC			
Synopsis	Data read change processing		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_GraphChangeSurfaceVDC(`
	`const vdc_channel_t ch,`
	`const vdc_layer_id_t layer_id,`
	`void* const fb_buff,`
	`vdc_period_rect_t * const gr_grc,`
	`vdc_width_read_fb_t * const width_read_fb,`
	`vdc_gr_disp_sel_t * const gr_disp_sel);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_layer_id_t layer_id	Layer ID • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_1_RD (Note 1) • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
	[IN]	void * framebuff	Frame buffer base address
	[IN]	vdc_period_rect_t * gr_grc	Graphics display area
	[IN]	vdc_width_read_fb_t * width_read_fb	Size of the frame buffer to be read
	[IN]	vdc_gr_disp_sel_t * r_disp_sel	Graphics display mode
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_LAYER_ID		Invalid layer ID error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_PARAM_EXCEED_RANGE		Out-of-value-range error
	VDC_ERR_RESOURCE_LAYER		Layer resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.20.1 Description

This function changes the address of the data read buffer.

The following driver is used within this function:

- R_VDC_ChangeReadProcess()

5.20.2 Parameter details

vdc_period_rect_t

vdc_period_rect_t is a structure for representing the horizontal/vertical timing of the VDC signals.

```
typedef struct
{
    uint16_t    vs;
    uint16_t    vw;
    uint16_t    hs;
    uint16_t    hw;
} vdc5_period_rect_t;
```

Type Member Name	Description
uint16_t vs	Vertical signal start position from the reference signal (lines) vs = 0 causes the display to start at the origin.
uint16_t vw	Vertical signal width (lines)
uint16_t hs	Horizontal signal start position from the reference signal (clock cycles) hs = 0 causes the display to start at the origin.
uint16_t hw	Horizontal signal width (clock cycles)

vdc_width_read_fb

The members of the vdc_width_read_fb_t structure is described below.

```
typedef struct
{
    uint16_t                in_vw;
    uint16_t                in_hw;
} vdc_width_read_fb_t;
```

Type Member Name	Description
uint16_t in_vw	Number of lines in a frame (lines) 0x0000 to 0x07FF
uint16_t in_hw	Width of the horizontal valid period (pixels) 0x0000 to 0x07FF

vdc_gr_disp_sel_t

vdc5_gr_disp_sel_t is an enumeration type for representing the graphics display modes.

```
typedef enum
{
    VDC_DISPSEL_IGNORED    = -1,
    VDC_DISPSEL_BACK       = 0,
    VDC_DISPSEL_LOWER      = 1,
    VDC_DISPSEL_CURRENT    = 2,
    VDC_DISPSEL_BLEND      = 3,
    VDC_DISPSEL_NUM        = 4
} vdc_gr_disp_sel_t;
```

Enumeration constant	Description
VDC_DISPSEL_IGNORED	Ignored, no change made
VDC_DISPSEL_BACK	Background color display
VDC_DISPSEL_LOWER	Lower-layer graphics display
VDC_DISPSEL_CURRENT	Current graphics display
VDC_DISPSEL_BLEND	Blended display of lower-layer graphics and current graphics
VDC_DISPSEL_NUM	Number of graphics display modes

5.21 R_RVAPI_AlphablendingVDC

R_RVAPI_GraphChangeSurfaceVDC			
Synopsis	1bit alpha blending setup		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_AlphablendingVDC(`
	`const vdc_channel_t ch,`
	`const vdc_layer_id_t layer_id,`
	`uint8_t alpha_value0,`
	`uint8_t alpha_value1);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	vdc_layer_id_t layer_id	Layer ID • VDC_LAYER_ID_0_RD • VDC_LAYER_ID_1_RD (Note 1) • VDC_LAYER_ID_2_RD • VDC_LAYER_ID_3_RD
	[IN]	uint8_t alpha_value0	Alpha signal of the ARGB1555/RGBA5551 format Alpha signal when alpha is set to ‘0’ 0 to 255
	[IN]	uint8_t alpha_value1	Alpha signal of the ARGB1555/RGBA5551 format Alpha signal when alpha is set to ‘1’ 0 to 255
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_LAYER_ID		Invalid layer ID error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_RESOURCE_LAYER		Layer resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.21.1 Description

This function changes the address of the data read buffer.

The following driver is used within this function:

- R_VDC_AlphaBlending()

5.22 R_RVAPI_VideoCreateSurfaceIMRL2

R_RVAPI_VideoCreateSurfaceIMRL2			
Synopsis	Video create surface processing		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_AlphablendingVDC(`
	`const vdc_channel_t ch,`
	`v_surface_config_t * const v_cnf,`
	`const v_surface_disp_config_t * const v_disp_cnf);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	v_surface_config_t * v_cnf l	Video input area settings Specify NULL when making no video input
	[IN]	v_surface_disp_config_t * v_g_cnf	Video input area display settings Specify NULL when making no display
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_PARAM_EXCEED_RANGE		Out-of-value-range error
	VDC_ERR_PARAM_CONDITION		Unauthorized condition error
	VDC_ERR_RESOURCE_LVDS_CLK		LVDS clock resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.22.1 Description

Please refer to [R_RVAPI_VideoCreateSurfaceVDC] except using YUV422 format.

5.23 R_RVAPI_VideoCreateSurfaceIMRLD

R_RVAPI_VideoCreateSurfaceIMRLD			
Synopsis	Video create surface processing		
Header	r_rvapi_vdc.h		

Declaration	`vdc_error_t R_RVAPI_AlphablendingVDC(`
	`const vdc_channel_t ch,`
	`v_surface_config_t * const v_cnf,`
	`const v_surface_disp_config_t * const v_disp_cnf);`		

Arguments	[IN]	vdc_channel_t ch	VDC5 channel • VDC_CHANNEL_0 • VDC_CHANNEL_1 (Note 1)
	[IN]	v_surface_config_t * v_cnf l	Video input area settings Specify NULL when making no video input.
	[IN]	v_surface_disp_config_t * v_g_cnf	Video input area display settings Specify NULL when making no display.
Return value	VDC_OK		Normal termination
	VDC_ERR_PARAM_CHANNEL		Channel invalid error
	VDC_ERR_PARAM_NULL		NULL specification error
	VDC_ERR_PARAM_BIT_WIDTH		Bit width error
	VDC_ERR_PARAM_UNDEFINED		Undefined parameter specification error
	VDC_ERR_PARAM_EXCEED_RANGE		Out-of-value-range error
	VDC_ERR_PARAM_CONDITION		Unauthorized condition error
	VDC_ERR_RESOURCE_LVDS_CLK		LVDS clock resource error
Remarks			

Note 1: Configurable only on the RZ/A1H and RZ/A1M

5.23.1 Description

Please refer to [R_RVAPI_VideoCreateSurfaceVDC] except using YUV422 format.

6. Function Reference (CEU)

6.1 R_RVAPI_InitializeCEU

R_RVAPI_InitializeCEU	
Synopsis	CEU initialization setup
Header	r_rvapi_ceu.h
Declaration	`void R_RVAPI_InitializeCEU(void);`
Arguments	[IN] None
Return value	None
Remarks	

6.1.1 Description

This function releases the CEU standby mode, enables interrupts, and sets up the interrupt handler.

The following driver is used within this function:

- R_CEU_Initialize()

6.2 R_RVAPI_TerminateCEU

R_RVAPI_TerminateCEU	
Synopsis	CEU termination setup
Header	r_rvapi_ceu.h
Declaration	`void R_RVAPI_TerminateCEU(void);`
Arguments	None
Return value	None
Remarks	

6.2.1 Description

This function enables the CEU standby mode, disables interrupts, and releases the interrupt handler.

The following driver is used within this function:

- R_CEU_InterruptDisable()
- R_CEU_Terminate()

6.3 R_RVAPI_PortSettingCEU

R_RVAPI_PortSettingCEU			
Synopsis	Video input pin setup		
Header	r_rvapi_ceu.h		
Declaration	`void R_RVAPI_PortSettingCEU(` `  void (* const port_func)(uint32_t));`		
Arguments	[IN]	void (* const port_func) (uint32_t)	Pointer to function for setting video input pins
Return value	None		
Remarks			

6.3.1 Description

The callback function to be set up with this function must configure the pins that are necessary for the CEU to capture the video image. This function must have been called by the time the CEU starts image capturing as shown in Figure 6-1.

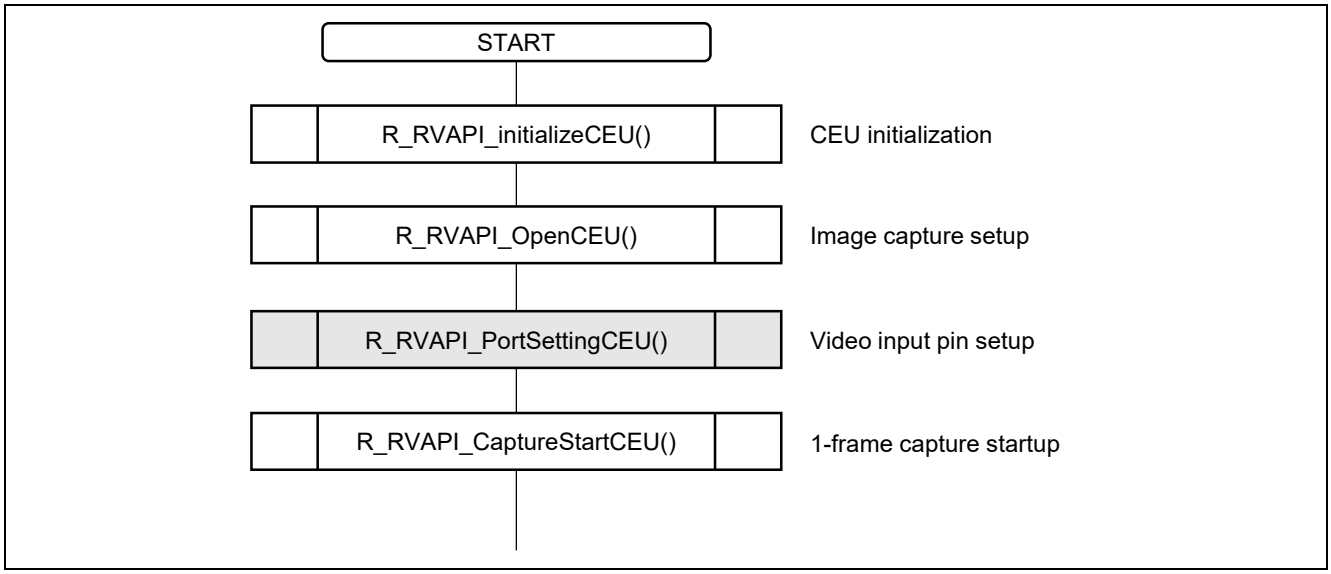


Figure 6-1 Timing When Configuring the CEU's Video Input Pins

6.4 R_RVAPI_OpenCEU

R_RVAPI_OpenCEU			
Synopsis	Image capturing setup		
Header	r_rvapi_ceu.h		
Declaration	`ceu_error_t R_RVAPI_OpenCEU(` `  const ceu_config_t * const config);`		
Arguments	[IN]	ceu_config_t * config	Configuration Do not specify NULL
Return value	CEU_OK		Normal termination
	CEU_ERR_PARAM		config or cap is set to NULL, cap and clp values are out of valid range
Remarks			

6.4.1 Description

This function is used to select the CEU capture mode, set up the capture size, and set up the interface with the external module. There are some parameters that need no configuration depending on the capture mode selected. Table 6-1 lists the parameters that may not be set up.

Capture Mode Selection ceu_jpg_t jpg	Image Capture Mode	Data Synchronous Fetch Mode	Data Enable Fetch Mode
ceu_dtif_t dtif	✓	✓	✓
ceu_sig_pol_t vdpol	✓	✓	Need not be set.
ceu_sig_pol_t hdpol	✓	✓	Need not be set.
ceu_dtary_t dtary	✓	✓ (Note 1)	✓ (Note 1)
ceu_cap_rect_t * cap	✓	✓	Need not be set.
ceu_clp_t * clp	✓	Need not be set (Note 2)	Need not be set.
ceu_onoff_t cols/ cows/ cobs	✓	✓	✓

Note 1: CEU_CB0_Y0_CR0_Y1 must be set up by the driver.
Note 2: The driver must set vfclp to vwdth and hfclp to (hwdth / 2) for the 8-bit interface.
For the 16-bit interface, the driver must set vfclp to vwdth and hfclp to hwdth.

Table 6-1 Parameters that need not be Set up Depending on the Selected Capture Mode

The following driver is used within this function:

- R_CEU_Open()
- R_CEU_InterruptEnable()

6.4.2 Parameter details

ceu_config_t

The members of the ceu_config_t structure are shown below.

```
typedef struct
{
    ceu_jpg_t           jpg;
    ceu_dtif_t          dtif;
    ceu_sig_pol_t       vdpol;
    ceu_sig_pol_t       hdpol;
    ceu_dtary_t         dtary;
    ceu_cap_rect_t    * cap;
    ceu_clp_t         * clp;
    ceu_onoff_t         cols;
    ceu_onoff_t         cows;
    ceu_onoff_t         cobs;
} ceu_config_t;
```

Type / Member Name	Description
ceu_jpg_t jpg	Capture mode selection • CEU_IMAGE_CAPTURE_MODE Image capture mode • CEU_DATA_SYNC_MODE Data synchronous fetch mode • CEU_DATA_ENABLE_MODE (Note 1) Data enable fetch mode
ceu_dtif_t dtif	Specifies the pins to be used to input the digital image to be captured. • CEU_8BIT_DATA_PINS 8-bit interface • CEU_16BIT_DATA_PINS (Note 2) 16-bit interface
ceu_sig_pol_t vdpol	Specifies the sensing polarity of the vertical sync signal from the external module. • CEU_HIGH_ACTIVE Senses the vertical sync signal from the external module (VD) as a high active signal. • CEU_LOW_ACTIVE Senses the vertical sync signal from the external module (VD) as a low active signal.
ceu_sig_pol_t hdpol	Specifies the sensing polarity of the horizontal sync signal from the external module. • CEU_HIGH_ACTIVE Senses the horizontal sync signal from the external module (HD) as a high active signal. • CEU_LOW_ACTIVE Senses the vertical sync signal from the external module (HD) as a low active signal.
ceu_dtary_t dtary	Specifies the order in which the luminance and color difference components are to be input. Specify CEU_CB0_Y0_CR0_Y1 for the data synchronous and data enable fetch modes. (With the 8-bit interface) • CEU_CB0_Y0_CR0_Y1 The image input data is fetched in the order of Cb0, Y0, Cr0, and Y1. • CEU_CR0_Y0_CB0_Y1

	The image input data is fetched in the order of Cr0, Y0, Cb0, and Y1. • CEU_Y0_CB0_Y1_CR0 The image input data is fetched in the order of Y0, Cb0, Y1, and Cr0. • CEU_Y0_CR0_Y1_CB0 The image input data is fetched in the order of Y0, Cr0, Y1, and Cb0. (With the 16-bit interface) • CEU_CB0_Y0_CR0_Y1 The image input data is fetched in the order of {Cb0, Y0} and {Cr0, Y1}. • CEU_CR0_Y0_CB0_Y1 The image input data is fetched in the order of {Cr0, Y0} and {Cb0, Y1}. • CEU_Y0_CB0_Y1_CR0 The image input data is fetched in the order of {Y0, Cb0} and {Y1, Cr0}. • CEU_Y0_CR0_Y1_CB0 The image input data is fetched in the order of {Y0, Cr0} and {Y1, Cb0}.
ceu_cap_rect_t * cap	Specifies the capture size. This member need be set up when the image capture mode or data synchronous fetch mode is selected. Specify NULL if the member need not be set up.
ceu_clp_t * clp	Filter size clip setting. This member need be set up when the image capture mode is selected. Specify NULL if the member need not be set up.
ceu_onoff_t cols	32-bit swap
ceu_onoff_t cows	16-bit swap
ceu_onoff_t cobs	8-bit swap

Note 1: Must not be set up for the RZ/A1H and RZ/A1M

Note 2: Configurable only on the RZ/A1H and RZ/A1M

ceu_cap_rect_t

The members of the ceu_cap_rect_t structure are shown below. These members need be set up when the image capture mode or data synchronous fetch mode is selected.

```
typedef struct
{
    uint32_t    vofst;
    uint32_t    vwdth;
    uint32_t    hofst;
    uint32_t    hwdth;
} ceu_cap_rect_t;
```

Type / Member Name		Description
uint32_t	vofst	Specifies the capture position with the number of HDs from the vertical sync signal [in 1HD units]. Specify a number 4095 or smaller.
uint32_t	vwdth	Specifies the capture period in the vertical direction [in 4HD units]. Specify a number not greater than 1920.
uint32_t	hofst	Specifies the capture position with the number of cycles from the horizontal sync signal [in 1 cycle units]. Specify a number 8191 or smaller.
uint32_t	hwdth	Specifies the capture period in the horizontal direction. (With the 8-bit interface) In image capture mode: [8 cycle units]: 5,120 cycles or smaller In data synchronous fetch mode: [4 cycle units]: 2,560 or smaller (With the 16-bit interface) In image capture mode: [4 cycle units]: 2,560 cycles or smaller In data synchronous fetch mode: [2 cycle units]: 1,280 or smaller

ceu_clp_t

The members of the ceu_clp_t structure are shown below.

These members need be set up when the image capture mode is selected.

```
typedef struct
{
    uint32_t    vfclp;
    uint32_t    hfclp;
} ceu_clp_t;
```

Type/Member Name		Description
uint32_t	vfclp	Clip value of the vertical direction filter output size [in 4 pixel units]
uint32_t	hfclp	Clip value of the horizontal direction filter output size [in 4 pixel units]

6.4.3 About the configuration of the capture size

Given below is an explanation of the capture size configuration (cap) to be made when connecting a CMOS camera which generates YCbCr422 format video output.

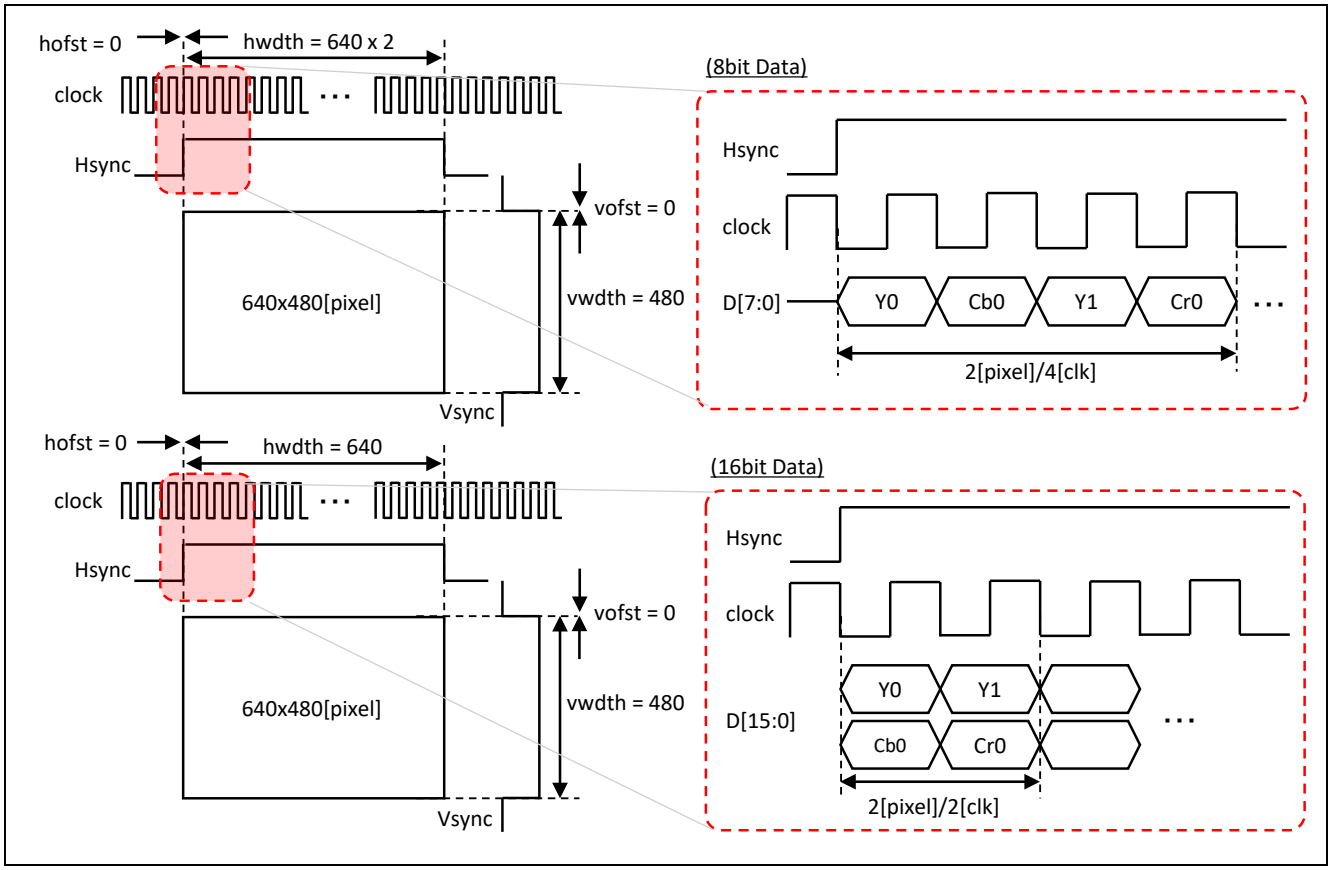


Figure 6-2 Timing of the Signals Output from the Camera

The timing of the camera-output signals is shown in Figure 6-2. This figure shows that since the image data is output from the camera at the same timing when the horizontal sync signals (Hsync) / vertical sync signal (Vsync) rise, hofst / vofst which indicates the image capture position are set to 0.

While the value of vwdth indicating the vertical image capture period is 480 which is the same as the height of the image, the value of hwdth, which indicates the horizontal image capture period, varies depending on the number of clocks that are required to capture 1 pixel.

When an 8-bit interface is attached, since the number of clocks required to capture 2 [pixels] is 4 [clks] (twice), the value of hwdth turns to 640 x 2 [clks].

When a 16-bit interface is attached, since the number of clocks required to capture 2 [pixels] is 2 [clk] (the same value), the value of hwdth turns to 640 [clks].

Figure 6-3 shows a configuration example for a 8-bit interface.

```
Image capture mode
ceu_config_t     config;
ceu_cap_rect_t cap;
ceu_clp_t        clp;

config.jpg    =
CEU_IMAGE_CAPTURE_MODE;

cap.hofst     = 0u;
cap.vofst     = 0u;
cap.hwdth     = 640u* 2u;
cap.vwdth     = 480u;
config.cap    = &cap;

clp.hfclp     = 640u;
clp.vfclp     = 480u;
config.clp    = &clp;
```

```
Data synchronous fetch mode
ceu_config_t     config;
ceu_cap_rect_t cap;

config.jpg    =
CEU_DATA_SYNC_MODE;

cap.hofst     = 0u;
cap.vofst     = 0u;
cap.hwdth     = 640u* 2u;
cap.vwdth     = 480u;
config.cap    = &cap;

config.clp    = NULL;
```

```
Data enable fetch mode
ceu_config_t   config;

config.jpg    =
CEU_DATA_ENABLE_MODE;

config.cap    = NULL;

config.clp    = NULL;
```

Figure 6-1 Sample Parameter Settings (8-bit Interface)

Figure 6-4 shows a configuration example for a 16-bit interface.

```
Image capture mode
ceu_config_t     config;
ceu_cap_rect_t cap;
ceu_clp_t        clp;

config.jpg    =
CEU_IMAGE_CAPTURE_MODE;

cap.hofst     = 0u;
cap.vofst     = 0u;
cap.hwdth     = 640u;
cap.vwdth     = 480u;
config.cap    = &cap;

clp.hfclp     = 640u;
clp.vfclp     = 480u;
config.clp    = &clp;
```

```
Data synchronous fetch mode
ceu_config_t     config;
ceu_cap_rect_t cap;

config.jpg    =
CEU_DATA_SYNC_MODE;

cap.hofst     = 0u;
cap.vofst     = 0u;
cap.hwdth     = 640u;
cap.vwdth     = 480u;
config.cap    = &cap;

config.clp    = NULL;
```

```
Data enable fetch mode
ceu_config_t   config;

config.jpg    =
CEU_DATA_ENABLE_MODE;

config.cap    = NULL;

config.clp    = NULL;
```

Figure 6-2 Sample Parameter Settings (16-bit Interface)

6.5 R_RVAPI_CaptureStartCEU

R_RVAPI_CaptureStartCEU			
Synopsis	1-frame capture startup		
Header	r_rvapi_ceu.h		

Declaration	`ceu_error_t R_RVAPI_CaptureStartCEU(`
	`const void * cayr,`
	`const void * cacr,`
	`uint32_t chdw);`		

Arguments	[IN]	void * cayr	Data storage area address specification 1 Do not specify NULL • In image capture mode Address of the area for storing the capture data luminance component data [in 4-byte units] • Data synchronous fetch mode Address of data storage area [in 4-byte units] • In data enable fetch mode Address of data storage area [in 32-byte units]
	[IN]	void * cacr	Data storage area address specification 2 • This member need be set up when the image capture mode is selected. Address of the area for storing the capture data color difference component data [in 4-byte units]
	[IN]	uint32_t chdw	Data buffer stride [bytes] • In image capture mode Capture data buffer stride [in 4-byte units] • Data synchronous fetch mode (for the 8-bit interface) Specify horizontal capture period (hwdth) (for the 16-bit interface) Specify horizontal capture period (hwdth) x 2
Return value	CEU_OK		Normal termination
	CEU_ERR_PARAM		cayr / cacr set to NULL (Note 1) cayr / cacr values are out of valid range (Note 1) chdw value is out of valid range The function is called again during capture processing
Remarks			

6.5.1 Description

This function starts capturing one frame. Since this function is of asynchronous type, it is necessary to use function described in ‘6.6 R_RVAPI_CaptureStatusCEU()’ to identify the completion of the 1-frame capturing.

The following driver is used within this function:

- R_CEU_Execute()

6.6 R_RVAPI_CaptureStatusCEU

R_RVAPI_CaptureStatusCEU		
Synopsis	Capture termination judgment	
Header	r_rvapi_ceu.h	
Declaration	`cap_status_t R_RVAPI_CaptureStatusCEU(void);`	
Arguments	None	
Return value	CAP_BUSY	Capturing in progress
	CAP_END	Capturing has ended
Remarks		

6.6.1 Description

This function returns the end of 1-frame capture status. It returns the end of capture status even when no capturing is started.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Nov. 29, 2019	-	First Edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.