RZ/A2M Group

Capture Engine Unit Sample Driver

R01AN4474EJ0101
Rev.1.01
Dec 28, 2018

Introduction

This document describes the functional specification of Capture Engine Unit (CEU) driver for RZ/A series RZ/A2M group MCU.

Target Device

RZ/A2M

Contents

1. Specifications 3
2. Operation Confirmation Conditions 5
3. Reference Application Notes 7
4. Hardware Description 7
4.1 Hardware Configuration 7
4.2 List of Pins to Be Used 7
5. Software Description 8
5.1 Enumeration Definitions 8
5.2 Error Codes 11
5.3 Restrictions 11
5.4 Functions 11
6. Functions Reference 12
6.1 R_CEU_Initialize 12
6.2 R_CEU_Open 13
6.3 R_CEU_Execute 19
6.4 R_CEU_Stop 20
6.5 R_CEU_Terminate 21
6.6 R_CEU_InterruptEnable 22
6.7 R_CEU_InterruptDisable 23
6.8 CEU_Isr 24
6.9 R_CEU_OnInitialize 25
6.10 R_CEU_OnFinalize 25
7. Processing Flow of the Application Using the CEU Sample Driver 26
8. Reference Documents 27

1. Specifications

This driver uses the RZ/A2M group on-chip CEU to capture the video data and transfer the memory.

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

Table 1-1 Peripheral Functions to Be Used and Their Uses

Classification	Item	Implemented Function	Description	Remarks
Attachable camera	Sample sizes	5M pixels	2,560 pixels ×1,920 line	Horizontal: In 4 pixel units Vertical: In 4 line units (Note 1) Sizes of image that can be input Horizontal 2,560pixels to 128 pixels Vertical 1,920lines to 96 lines
		UXGA	1,600 pixels ×1,200 lines	
		SXGA	1,280 pixels ×1,024 lines	
		XGA	1,024 pixels × 768 lines	
		SVGA	800 pixels ×600 lines	
		VGA	640 pixels ×480 lines	
		Sub-QCIF	128 pixels × 96 lines	
	Input format	YCbCr422 8 bits	Cb0, Y0, Cr0, Y1…	Supports the 1-to-1 clock ratio.
			Cr0, Y0, Cb0, Y1…	
			Y0, Cb0, Y1, Cr0…	
			Y0, Cr0, Y1, Cb0…	
		YCbCr422 16 bits	{Y0, Cb0}, {Y1, Cr0}…	
			{Y0, Cr0}, {Y1, Cb0}…	
		Binary data	Data of the specified size is captured starting at an edge of the sync signal.	Written sequentially.
			Captured using the horizontal sync signal as the enable signal (not configurable on the RZ/A1H and RZ/A1M).	
	Horizontal/vertical sync signal polarity	Optional	Active high Active low	
	Capture start position	Optional	Specifiable in camera input clock units.	Horizontal: In 1 cycle units Vertical: In 1 HD (horizontal sync signal) units
	Number of captured pixels	Optional	Specifiable in units of 4 pixels horizontally and 4 lines vertically.	
	Interlace (Note 2)	Both field capture	Stored as field image.	Capture: In 2 VD (vertical sync signal) units
			Stored as frame image.	
		Single field capture	Top field and bottom field are specifiable.	Capture: In 1 VD units
Memory write	Output format (Note 2)	YCbCr422 YCbCr420	Simple thinning for YCbCr420	
Filter function	Same size, reduced size (Note 2)	Capture image reduction	Arbitrary magnification between 1/16 and 1(The size of the reduced image is VGA or less.)	

Low-pass filter	High frequency component rejection	Applicable only in horizontal direction.

Note 1: Dependent on the device to be attached, its AC characteristics, the frame rate of the device to be attached, and the rate at which data is to be transferred to RAM.

Note 2: This driver does not support it.

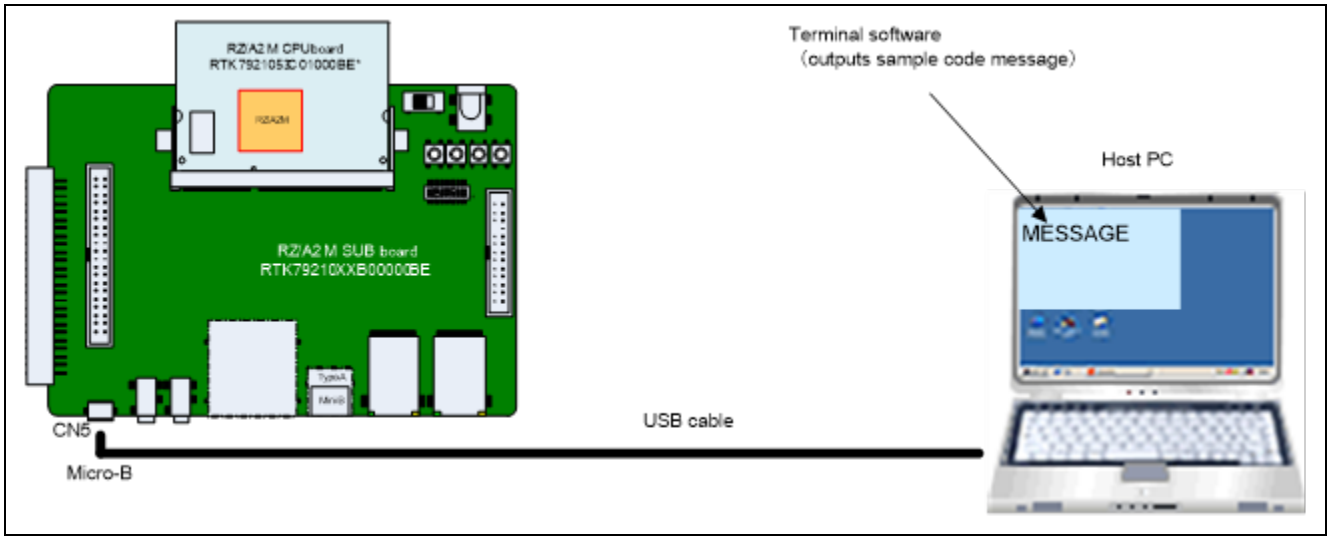

Figure 1.1 Operation check conditions

2. Operation Confirmation Conditions

The sample code of this application note has been confirmed to operate under the following conditions.

Table 2.1 Peripheral device used(1/2)

Peripheral device	Usage
Microcomputer used	RZ/A2M
Operating frequency[MHz] (Note)	CPU Clock (Iφ) : 528MHz Image processing clock (Gφ) : 264MHz Internal Bus Clock (Bφ) : 132MHz Peripheral Clock 1 (P1φ) : 66MHz Peripheral Clock 0 (P0φ) : 33MHz QSPI0_SPCLK : 66MHz CKIO : 132MHz
Operating voltage	Power supply voltage (I/O): 3.3 V Power supply voltage (either 1.8V or 3.3V I/O (PVcc SPI)) : 3.3V Power supply voltage (internal): 1.2 V
Integrated development environment	e2 studio V7.3.0
C compiler	"GNU Arm Embedded Tool chain 6-2017-q2-update" compiler options(except directory path) Release: -mcpu=cortex-a9 -march=armv7-a -marm -mthumb-interwork -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Os -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -Wstack-usage=100 -fabi-version=0 Hardware Debug: -mcpu=cortex-a9 -march=armv7-a -marm -mthumb-interwork -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Og -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -g3 -Wstack-usage=100 -fabi-version=0

Note: The operating frequency used in clock mode 1 (Clock input of 24MHz from EXTAL pin)

Table 2.2 Peripheral device used(2/2)

Operation mode	Boot mode 3 (Serial Flash boot 3.3V)
Terminal software communication settings	Communication speed: 115200bps Data length: 8 bits Parity: None Stop bits: 1 bit Flow control: None
Board to be used	RZ/A2M CPU board RTK7921053C00000BE RZ/A2M SUB board RTK79210XXB00000BE
Device (functionality to be used on the board)	Serial flash memory allocated to SPI multi-I/O bus space (channel 0) Manufacturer : Macronix Inc. Model Name : MX25L51245GXD RL78/G1C (This device communications the host PC by convert USB Communication and Serial Communication.)

3. Reference Application Notes

The application notes related to this application note are shown below.

Nothing.

4. Hardware Description

4.1 Hardware Configuration

Please refer to the manual of RZ / A2M evaluation board for hardware configuration.

4.2 List of Pins to Be Used

Table 4-1 lists the pins to be used and describes their functionalities.

Table 4-1 Pins to Be Used and Their Functions

Pin name	I/O	Description	RZ / A2M evaluation board connection
VIO7~VIO0	Input	CEU data bus	PE_6-1, PH_1-0
VIO_CLK	Input	CEU clock	P6_1
VIO_VD	Input	CEU vertical sync	P6_2
VIO_HD	Input	CEU horizontal sync	P6_3
VIO_FLD	Input	Field signal	NC

5. Software Description

This section describes each enum type definition in the code. For details on error codes, refer to "5.2 Error Codes".

5.1 Enumeration Definitions

(1) **ceu_onoff_t**

ceu_onoff_t defines the ON/OFF state of a CEU function. The CEU sample driver uses this constant to configure bit swapping among 32, 16, and 8 bits.

```
typedef enum
{
    CEU_OFF    = 0,
    CEU_ON     = 1
} ceu_onoff_t;
```

Enumerator	Value	Description
CEU_OFF	0	OFF (Disables the function.)
CEU_ON	1	ON (Enables the function.)

(2) **ceu_jpg_t**

ceu_jpg_t defines the CEU's capture mode.

```
typedef enum
{
    CEU_IMAGE_CAPTURE_MODE = 0,
    CEU_DATA_SYNC_MODE,
    CEU_DATA_ENABLE_MODE
} ceu_jpg_t;
```

Enumerator	Value	Description
CEU_IMAGE_CAPTURE_MODE	0	Image capture mode
CEU_DATA_SYNC_MODE	1	Data synchronous fetch mode
CEU_DATA_ENABLE_MODE	2	Data enable fetch mode

(3) **ceu_dtif_t**

ceu_dtif_t defines the CEU's input interface.

```
typedef enum
{
    CEU_8BIT_DATA_PINS = 0,
    CEU_16BIT_DATA_PINS
} ceu_dtif_t;
```

Enumerator	Value	Description
CEU_8BIT_DATA_PINS	0	8-bit interface
CEU_16BIT_DATA_PINS	1	16-bit interface

(4) **ceu_sig_pol_t**

ceu_sig_pol_t defines the sense polarity of the sync signal from the external module.

```
typedef enum
{
    CEU_HIGH_ACTIVE = 0,
    CEU_LOW_ACTIVE
} ceu_sig_pol_t;
```

Enumerator	Value	Description
CEU_HIGH_ACTIVE	0	Senses the sync signal from the external module as a high active signal.
CEU_LOW_ACTIVE	1	Senses the sync signal from the external module as a low active signal.

(5) **ceu_dtary_t**

ceu_dtary_t defines the order in which the luminance and color difference components are to be input.

```
typedef enum
{
    CEU_CB0_Y0_CR0_Y1 = 0,
    CEU_CR0_Y0_CB0_Y1,
    CEU_Y0_CB0_Y1_CR0,
    CEU_Y0_CR0_Y1_CB0
} ceu_dtary_t;
```

Enumerator	Value	Description
CEU_CB0_Y0_CR0_Y1	0	With the 8-bit interface • The image input data is fetched in the order of Cb0, Y0, Cr0, and Y1. With the 16-bit interface • The image input data is fetched in the order of {Cb0, Y0} and {Cr0, Y1}.
CEU_CR0_Y0_CB0_Y1	1	With the 8-bit interface • The image input data is fetched in the order of Cr0, Y0, Cb0, and Y1. With the 16-bit interface • The image input data is fetched in the order of {Cr0, Y0} and {Cb0, Y1}.
CEU_Y0_CB0_Y1_CR0	2	With the 8-bit interface • The image input data is fetched in the order of Y0, Cb0, Y1, and Cr0. With the 16-bit interface • The image input data is fetched in the order of {Y0, Cb0} and {Y1, Cr0}.
CEU_Y0_CR0_Y1_CB0	3	With the 8-bit interface • The image input data is fetched in the order of Y0, Cr0, Y1, and Cb0. With the 16-bit interface • The image input data is fetched in the order of {Y0, Cr0} and {Y1, Cb0}.

(6) **ceu_int_type_t**

ceu_int_type_t defines the types of CEU interrupt to be enabled. When using two or more types of interrupts, specify the following definitions separated by ORs in the function "6.5R_CEU_InterruptEnable()."

```
typedef enum
{
    CEU_INT_CPEIE   = (0x00000001u),
    CEU_INT_CFEIE   = (0x00000002u),
    CEU_INT_IGEWIE  = (0x00000010u),
    CEU_INT_HDIE    = (0x00000100u),
    CEU_INT_VDIE    = (0x00000200u),
    CEU_INT_CPBE1IE = (0x00001000u),
    CEU_INT_CPBE2IE = (0x00002000u),
    CEU_INT_CPBE3IE = (0x00004000u),
    CEU_INT_CPBE4IE = (0x00008000u),
    CEU_INT_CDTOFIE = (0x00010000u),
    CEU_INT_IGHSIE  = (0x00020000u),
    CEU_INT_IGVSIE  = (0x00040000u),
    CEU_INT_VBPIE   = (0x00100000u),
    CEU_INT_FWFIE   = (0x00800000u),
    CEU_INT_NHDIE   = (0x01000000u),
    CEU_INT_NVDIE   = (0x02000000u)
} ceu_int_type_t;
```

Error Code	Value	Description (Error type)
CEU_INT_CPEIE	0x00000001u	Enables end of 1-frame capture interrupts.
CEU_INT_CFEIE	0x00000002u	Enables end of 1-field capture interrupts.
CEU_INT_IGEWIE	0x00000010u	Enables register access during capture interrupts (error related).
CEU_INT_HDIE	0x00000100u	Enables HD interrupts.
CEU_INT_VDIE	0x00000200u	Enables VD interrupts.
CEU_INT_CPBE1IE	0x00001000u	Enables CPBE1 interrupts. (Bundle write related)
CEU_INT_CPBE2IE	0x00002000u	Enables CPBE2 interrupts. (Bundle write related)
CEU_INT_CPBE3IE	0x00004000u	Enables CPBE3 interrupts. (Bundle write related)
CEU_INT_CPBE4IE	0x00008000u	Enables CPBE4 interrupts. (Bundle write related)
CEU_INT_CDTOFIE	0x00010000u	Enables CDTOF interrupts. (Error related)
CEU_INT_IGHSIE	0x00020000u	Enables IGHS interrupts. (Error related)
CEU_INT_IGVSIE	0x00040000u	Enables IGVS interrupts. (Error related)
CEU_INT_VBPIE	0x00100000u	Enables VBP interrupts. (Error related)
CEU_INT_FWFIE	0x00800000u	Enables FWF interrupts. (Error related)
CEU_INT_NHDIE	0x01000000u	Enables non-HD interrupts. (Error related) (Note 1)
CEU_INT_NVDIE	0x02000000u	Enables non-VD interrupts. (Error related) (Note 1)

Note 1: This type of interrupts must be disabled in data enable fetch mode.

5.2 Error Codes

Table 5-1 shows the error code lists of the CEU driver.

Table 5-1 CEU driver error code list

Error Code	Value	Description (Error type)
CEU_OK	0	Normal termination
CEU_ERR_PARAM	1	Parameter error • Data enable fetch mode was specified on the RZ/A1H or RZ/A1M. • 16-bit interface was specified for a platform other than the RZ/A1H and RZ/A1M. • cap is set to NULL, or cap or clp value is out of valid range. • cayr/ cacr value is set to NULL. (Note 1) • cayr/cacr value is out of valid range. (Note 1) • chdw value is out of valid range.

Note 1: cacr is checked only in image capture mode.

5.3 Restrictions

(1) **Reentrancy**

The functions of the CEU sample driver are not reentrant. An unexpected driver operation may result if a CEU sample driver function is called asynchronously by two or more tasks or interrupt processing routines.

5.4 Functions

Table 5-2 shows the API function lists of the CEU driver.

Table 5-2 List of RVAPI Functions

Function Name	Outline	Header file
R_CEU_Initialize	Initialization processing	r_ceu.h
R_CEU_Open	CEU configuration	r_ceu.h
R_CEU_Execute	Frame capture startup processing	r_ceu.h
R_CEU_Stop	Stop the Continuous capture	r_ceu.h
R_CEU_Terminate	CEU termination processing	r_ceu.h
R_CEU_InterruptEnable	Interrupt enable setup	r_ceu.h
R_CEU_InterruptDisable	Interrupt disable setup	r_ceu.h
CEU_Isr	Interrupt handler	r_ceu.h
R_CEU_OnInitialize	Sample for releasing CEU standby state and registering the interrupt handler	r_ceu_user.h
R_CEU_OnFinalize	Sample for setting up CEU standby state and releasing the interrupt handler	r_ceu_user.h

6. Functions Reference

6.1 R_CEU_Initialize

R_CEU_Initialize			
Synopsis	Initialization processing		
Header	r_ceu.h		
Declaration	`void R_CEU_Initialize( void (* const init_func)( uint32_t ), const uint32_t user_num );`		
Arguments	[IN]	void (* init_func)(uint32_t)	: callback function to be registered Specify NULL if not necessary.
	[IN]	uint32_t user_num	: Argument to the callback function Set up according to the application.
Return value	None		
Remarks			

(1) **Description**

Since the CEU sample driver will perform neither CEU module standby release processing nor interrupt handler registration processing, it is necessary to add those processing using the callback function specified in this function. "6.8 R_CEU_OnInitialize()" is available as a sample function for adding those processing. Add the required processing while referring to that sample.

This function takes the following actions:

- To initialize the internal variables to be used by the sample driver.
- To call the callback function specified in the argument.

6.2 R_CEU_Open

R_CEU_Open			
Synopsis	CEU configuration		
Header	r_ceu.h		
Declaration	`ceu_error_t R_CEU_Open( 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, or cap or clp value is out of valid range.
Remarks	•		

(1) **Description**

This function is used to select the CEU capture mode and size and to set up the interface with the external module. According to the capture mode, there are some parameters that do not need to be set. Table 6-1lists that parameters that need not be set up.

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

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.

This function takes the following actions:

- Selects the capture mode (jpg).
- Sets up the pins for inputting the digital image to be captured (dtif).
- Specifies the sensing polarity of the sync signal from the external module (vdpol/ hdpol).
- Specifies the order in which the luminance and color difference components are to be input (dtary).
- Capture size setting (cap)
- Filter size clip setting (clp)
- 32/16/8-bit swap settings (cols/ cows/ cobs)

(2) **Parameter details**

(a) **ceu_config_t**

ceu_config_t structure is described below.

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

Type/Member Name	Description
ceu_cap_rect_t * cap	Specifies the capture size. This member needs to 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 needs to be set up when the image capture mode is selected. Specify NULL if the member need not be set up.
ceu_jpg_t jpg	Selects the capture size. The RZ/A1H or RZ/A1M do not allow the data enable fetch mode to be selected. • CEU_IMAGE_CAPTURE_MODE Image capture mode • CEU_DATA_SYNC_MODE Data synchronous fetch mode • CEU_DATA_ENABLE_MODE Data enable fetch mode
ceu_dtif_t dtif	Specifies the pins to be used to input the digital image to be captured. The user can also select a 16-bit interface for the RZ/A1H and RZ/A1M. • CEU_8BIT_DATA_PINS 8-bit interface • CEU_16BIT_DATA_PINS 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 horizontal 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_Y1for 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_onoff_t cols	32-bit swap
ceu_onoff_t cows	16-bit swap
ceu_onoff_t cobs	8-bit swap

(b) **ceu_cap_rect_t**

The members of the ceu_cap_rect_t structure are shown below. This member needs to 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 1cycle 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 (in 8 cycle units) : 5,120 cycles maximum • In data synchronous fetch mode (in 4 cycle units) : 2,560 cycles maximum (With the 16-bit interface) • In image capture mode (in 4 cycle units) : 2,560 cycles maximum • In data synchronous fetch mode (in 2 cycle units): 1,280 cycles maximum

(c) **ceu_clp_t**

The members of the ceu_clp_t structure are shown below.

These members need to 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)

(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.

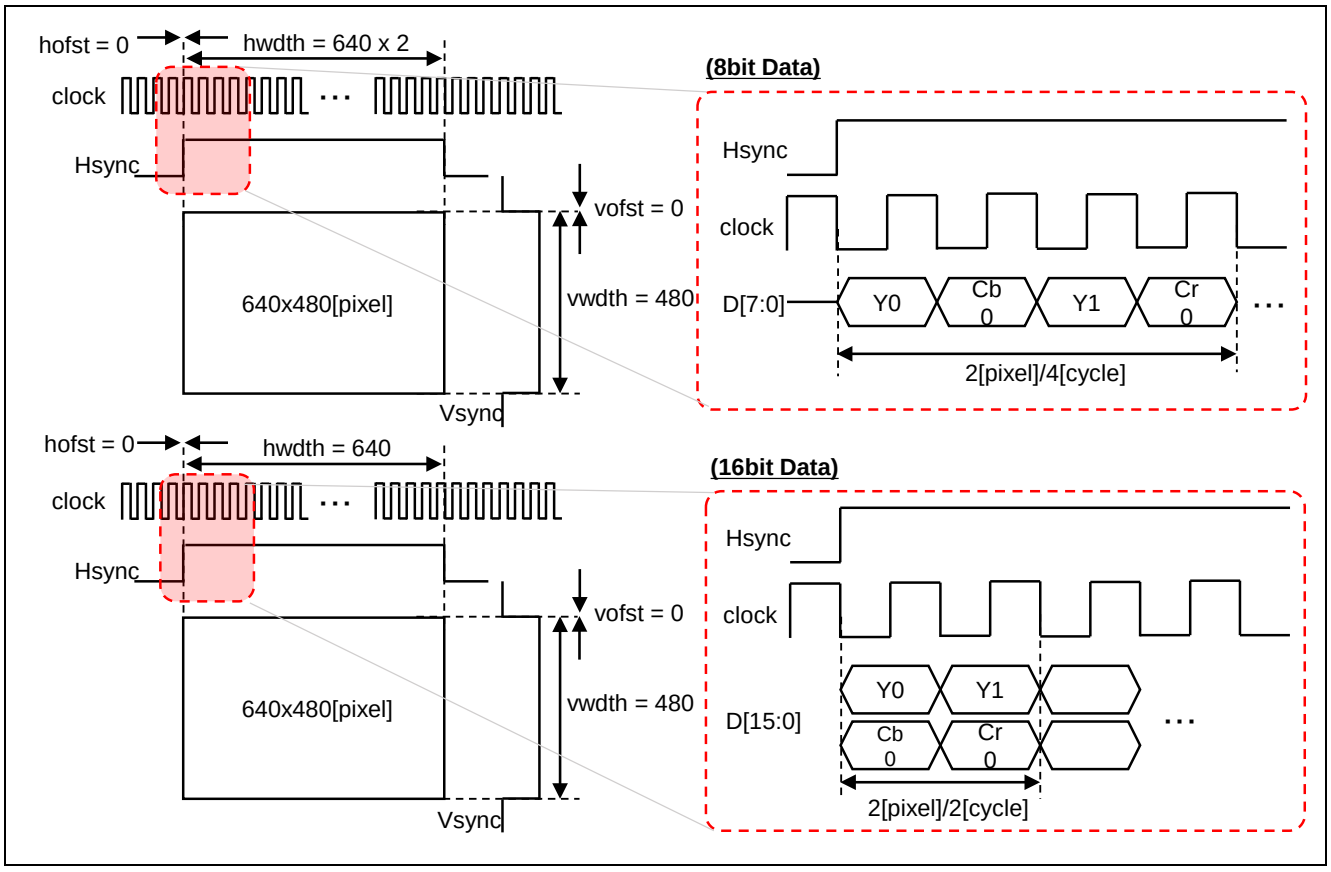

Figure 6-1 Timing of the Signals Output from the Camera

The timing of the camera-output signals is shown in Figure 6-1. 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 [cycles] (twice), the value of hwdth turns to 640 x 2 [cycles].

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

Figure 6-2 shows a configuration example for an 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 6-2 Sample R_CEU_Open Function's Parameter Settings (8-bit Interface)

Figure 6-3 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 6-3 Sample R_CEU_Open Function's Parameter Settings (16-bit Interface)

6.3 R_CEU_Execute

R_CEU_Execute			
Synopsis	Frame capture startup processing		
Header	r_ceu.h		
Declaration	`ceu_error_t R_CEU_Execute( const void * cayr, const void * cacr, uint32_t chdw ceu_onoff_t auto_capture);`		
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) • In 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 needs to 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 (in bytes) • In image capture mode Capture data buffer stride (in 4 byte units) • In 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.
	[IN]	ceu_onoff_t auto_capture	: Continuous capture
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.
Remarks	•		

[Note 1] cacr is checked only in image capture mode.

(1) **Description**

This function starts the image capture processing according to the settings made with the function described in Section 6.2, R_CEU_Open(). After starting image capturing, the end of 1-frame capture interrupt must be used to check for the completion of image capturing.

This function takes the following actions:

- To specify the address of the area for storing data.
- To specify the data buffer stride.

6.4 R_CEU_Stop

R_CEU_Stop		
Synopsis	Stop the Continuous capture	
Header	r_ceu.h	
Declaration	`ceu_error_t R_CEU_Stop(void);`	
Arguments	[IN] void	
Return value	CEU_OK	: Normal termination
Remarks		

(1) **Description**

In this function, clear the CE bit of the capture start register. Use this function to stop capture during continuous capture operation.

This function takes the following actions:

- Clear the CE bit of the capture start register

6.5 R_CEU_Terminate

R_CEU_Terminate			
Synopsis	CEU termination processing		
Header	r_ceu.h		
Declaration	`ceu_error_t R_CEU_Terminate( void (* const quit_func)( uint32_t ), const uint32_t user_num );`		
Arguments	[IN]	void (* quit_func)(uint32_t)	: Callback function to be registered Specify NULL if not necessary.
	[IN]	uint32_t user_num	: Argument to the callback function Set up according to the application.
Return value	CEU_OK		: Normal termination
Remarks			

(1) **Description**

This function performs a CEU software reset. Since the CEU sample driver performs neither CEU module standby configuration processing nor interrupt handler release processing, it is necessary to add those processing using the callback function specified in this function.

"6.9 R_CEU_OnFinalize ()" is available as a sample function for adding those processing. Add the required processing while referring to that sample.

This function takes the following actions:

- To perform a CEU software reset
- To call the callback function specified in the argument.

6.6 R_CEU_InterruptEnable

R_CEU_InterruptEnable			
Synopsis	Interrupt enable setup		
Header	r_ceu.h		

Declaration	`void R_CEU_InterruptEnable(`
	`const ceu_int_type_t int_type,`
	`void (* const callback)(ceu_int_type_t) );`		

Arguments	[IN]	ceu_int_type_t int_type	: CEU interrupt type selection
	[IN]	void (*callback)(ceu_int_type_t)	: Callback function to be registered Specify NULL if not necessary.
Return value	None		
Remarks			

(1) **Description**

This function takes the following actions: When using two or more types of interrupts, specify the correct ceu_int_type_t type definitions separated by ORs. The types of interrupts specified in the argument of the callback function will become identifiable. The interrupt priorities are checked by the function described in Section 6.8, "R_CEU_OnInitialize()" which is introduced as an example of registering an interrupt handler. If two or more callback functions are registered, only the last registered one can be used, and others are ignored. .

- To enable the types of CEU interrupts specified as an argument
- To register the callback function registered as an argument

6.7 R_CEU_InterruptDisable

R_CEU_InterruptDisable		
Synopsis	Interrupt disable setup	
Header	r_ceu.h	
Declaration	`void R_CEU_InterruptDisable( void );`	
Arguments	[IN]	None
Return value	None	
Remarks		

(1) **Description**

This function takes the following actions:

- To disable all types of CEU interrupts.
- To clear the registered callback function.

6.8 CEU_Isr

CEU_Isr			
Synopsis	Interrupt handler		
Header	r_ceu.h		
Declaration	`void CEU_Isr( const uint32_t int_sense );`		
Arguments	[IN]	uint32_t int_sense	: Interrupt Request Edge/Level
Return value	None		
Remarks			

(1) **Description**

This function serves as the interrupt handler to be used by the CEU sample driver.

The function is registered as the CEU interrupt handler through the function described in Section 6.8, "R_CEU_OnInitialize()" which is introduced as an example of interrupt handler registration processing.

6.9 R_CEU_OnInitialize

R_CEU_OnInitialize			
Synopsis	Sample processing for releasing the CEU standby state and registering an interrupt handler		
Header	r_ceu_user.h		
Declaration	`void R_CEU_OnInitialize ( const uint32_t user_num );`		
Arguments	[IN]	uint32_t user_num	: User parameter
Return value	None		
Remarks			

(1) **Description**

This function is introduced as an example of releasing the CEU module standby state and registering an interrupt handler. This function takes the following actions:

- To perform CEU standby release processing
- To register an interrupt handler
- To set up interrupt priorities

6.10 R_CEU_OnFinalize

R_CEU_OnFinalize			
Synopsis	Sample for setting up CEU standby state and releasing the interrupt handler		
Header	r_ceu_user.h		
Declaration	`void R_CEU_OnFinalize( const uint32_t user_num );`		
Arguments	[IN]	uint32_t user_num	: User parameter
Return value	None		
Remarks			

(1) **Description**

This function is introduced as an example of setting up the CEU module standby state and releasing the interrupt handler. This function takes the following actions:

- To set up the CEU standby state
- To release the interrupt handler

7. Processing Flow of the Application Using the CEU Sample Driver

Figure 7-1 shows the processing flow of the application using the CEU sample driver.

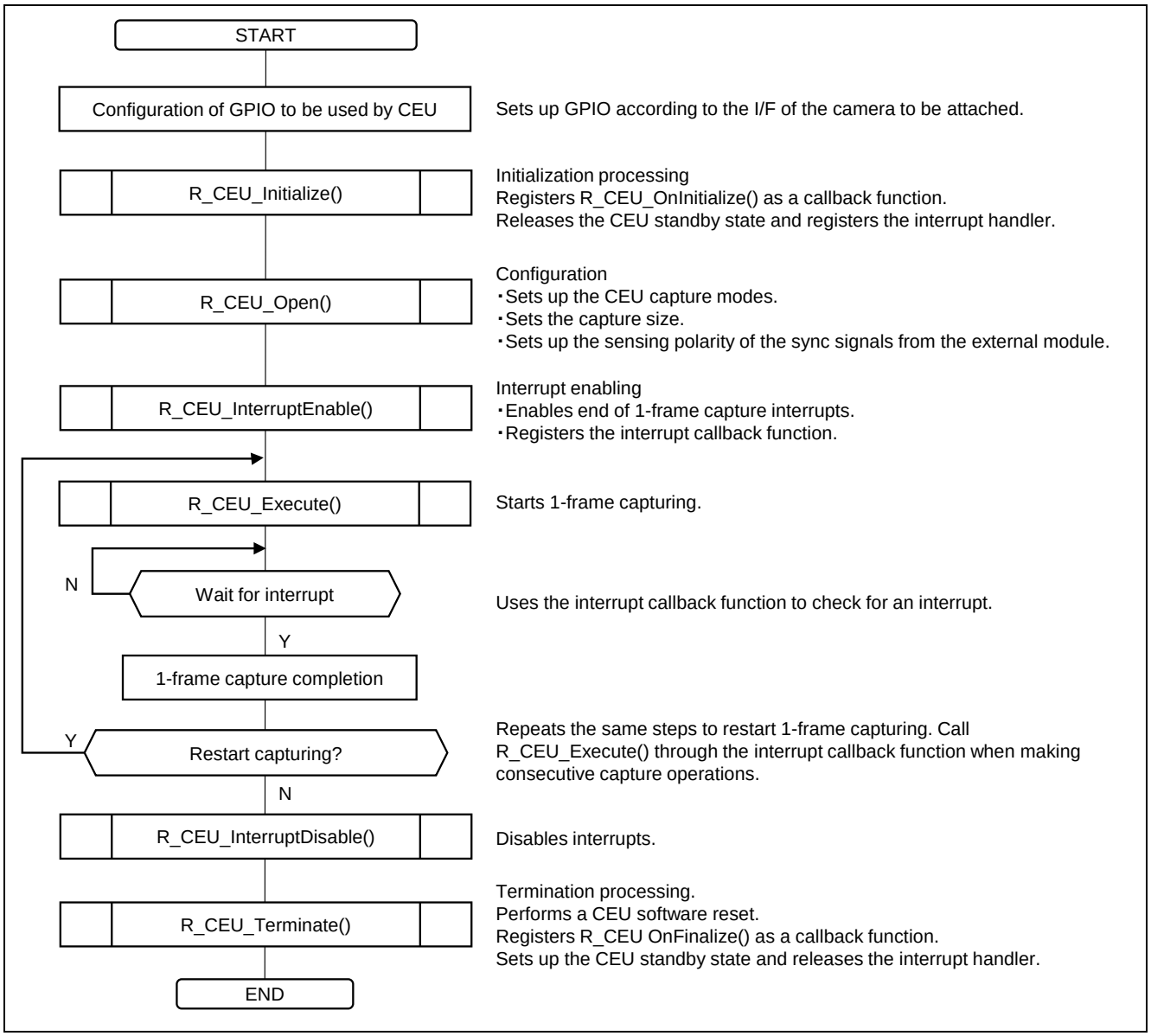

Figure 7-1 Processing Flow of the Application Using the CEU Sample Driver

The GPIO configuration of the pins to be used by the CEU must be accomplished by the user application since it is not carried out by the CEU sample driver.

8. Reference Documents

User's Manual: Hardware

RZ/A2M Group User's Manual: Hardware

The latest version can be downloaded from the Renesas Electronics website.

RTX921053C00000BE (RZ/A2M CPU board) User's Manual

The latest version can be downloaded from the Renesas Electronics website.

RTK79210XXB00000BE (RZ/A2M SUB board) User's Manual

The latest version can be downloaded from the Renesas Electronics website.

ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C

The latest version can be downloaded from the ARM website.

ARM Cortex™-A9 (Revision: r4p1) Technical Reference Manual

The latest version can be downloaded from the ARM website.

ARM Generic Interrupt Controller Architecture Specification - Architecture version 2.0

The latest version can be downloaded from the ARM website.

ARM CoreLink™ Level 2 Cache Controller L2C-310 (Revision: r3p3) Technical Reference Manual

The latest version can be downloaded from the ARM website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

User's Manual: Development Tools

Integrated development environment e2studio User’s Manual can be downloaded from the Renesas Electronics website.

The latest version can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Sep 14, 2018	-	First edition issued
1.01	Dec.28, 2018	p.28	Addition “6.4 R_CEU_Stop”

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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the 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.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.2