RX Family

JPEG Encoder Module

Firmware Integration Technology

R20AN0263EJ0102
Rev.1.02
Nov 15, 2024

Introduction

This material explains usage of JPEG Encoder (JPEGE).

JPEG Encoder is only for encoding, and there are the following two libraries.

JPEG Encode Library: DCT, Quantization and Huffman encoding.

JPEG File Compress Library: Compress JPEG data using JPEG Encode Library.

Normally, the API for the JPEG File Compress Library is used to compress bitmap images into the JPEG image format. But since the source code for the JPEG File Compress Library is provided, users can also change the specifications to match the particular needs of their applications.

Target Device

RX Family

Contents

1. Structure of product

Package name : JPEG Encoder for the RX Family V.1.02 Release 00

Table 1.1 Product Files of JPEG Encoder

File/Directory name	description
JPEGE FIT Module (r_jpege_rx_v.1.02.zip)	
JPEGE config (r_config)	
r_jpege_rx_config.h	JPEGE Config file (default setting)
JPEGE FIT Module body (r_jpege_rx)	
JPEGE document (doc)	
Japanese (ja)	
r20an0263jj0102_rx_jpege.pdf	Introduction Guide
r20uw0122jj0100_jpege.pdf	User's Manual
English (en)	
r20an0263ej0102_rx_jpege.pdf	Introduction Guide (this document)
r20uw0122ej0100_jpege.pdf	User's Manual
JPEGE Library (lib)	
jpege_rx600_little.lib jpege_rx600_big.lib jpege_rx200_little.lib jpege_rx200_big.lib r_jpeg.h	JPEG Encode Library and header file
compress_jpege_rx600_little.lib compress_jpege_rx600_big.lib compress_jpege_rx200_little.lib compress_jpege_rx200_big.lib r_compress_jpege.h	JPEG File Compress Library and header file
r_stdint.h	Data type header file
r_mw_version.h	Version data header file
JPEG File Compress Library make environment (make_lib)	
make_lib.zip	JPEG File Compress Library make environment (includes source code)
readme(readme.txt)	Readme

2. Specification of library

2.1 Structure of software stack

This figure explains structure of software stack of JPEG Encoder.

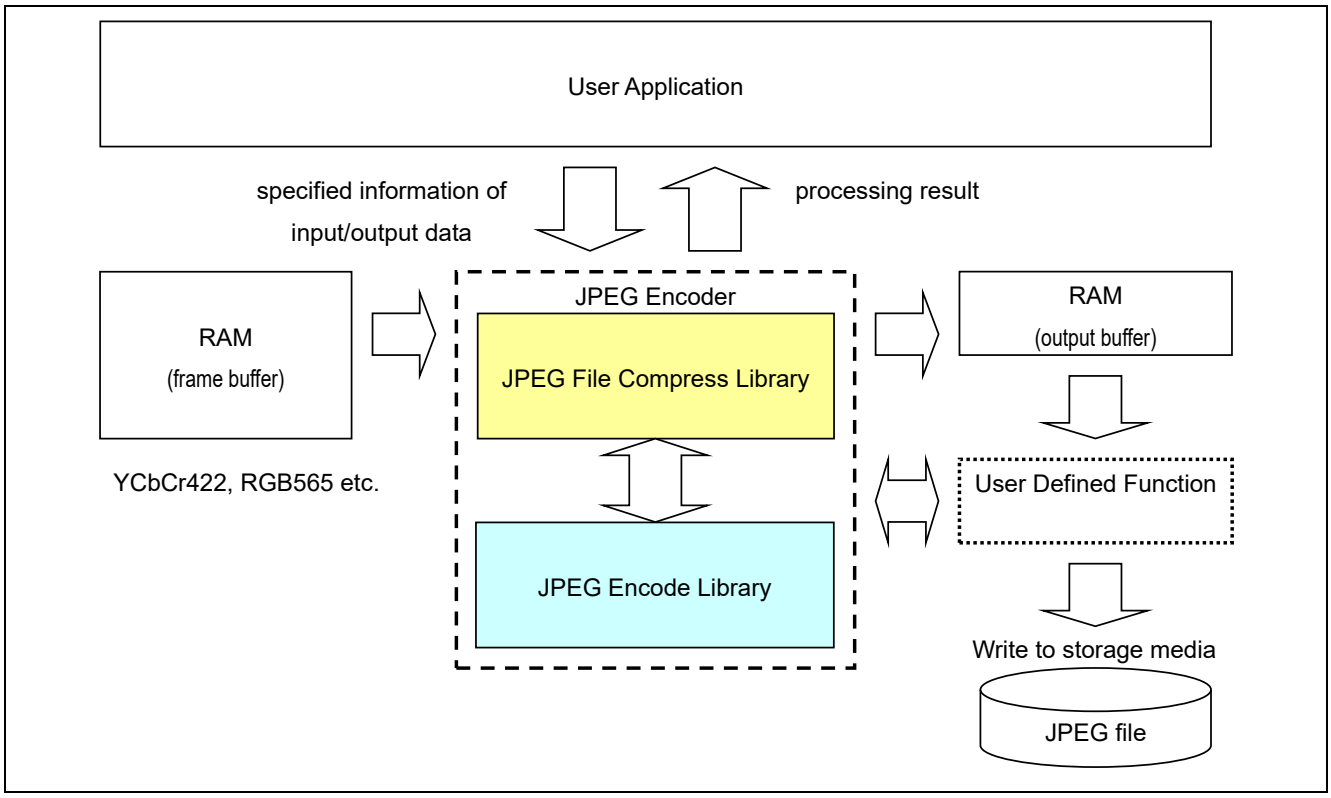

Figure 2.1 Structure of Software Stack

2.2 Specification of JPEG File Compress Library

Specification of this JPEG File Compress Library is below.

Table 2.1 specification of JPEG Compress Library

Items		Specifications
Output JPEG File	Support format	JFIF
	Elements of color	3 colors (YCbCr)
	Ratio of sample	4:2:2 (2x1,1x1,1x1) or 4:2:0 (2x2,1x1,1x1)
	Image quality	A value in the range 1 to 128 may be specified.
	Restart marker	Either none or an arbitrary interval may be set.
	Comment	none
	Exif	No support
	Progressive	No support
	Thumbnail	No support
	Output units	Output buffers with an arbitrary size may be provided and the data can be stored to various media in units of that size.
Input Image data	Image format	RGB565 (16bit color), RGB888 (24bit color), YCbCr 4:2:2
	Input units	Data is read as single unit. (Reading divided into smaller units is not possible.)

The source code of JPEG File Compress Library is attached so that a user can change specification.

2.3 Development environment

JPEG Encoder can run with this development environment below.

[IDE]

CS+ V8.12.00

e^{2}studio 2024-10

[C compiler]

C/C++ Compiler Package for RX Family V.1.02 Release 01 or later

Library file is built with default compile option.

- compile option (RX600 little endian)
  -cpu=rx600 -output=obj="$(CONFIGDIR)\$(FILELEAF).obj" -nologo
- compile option (RX600 big endian)
  Adding “-endian=big” to default option.

- compile option (RX200 little endian)
  -cpu=rx200 -output=obj="$(CONFIGDIR)\$(FILELEAF).obj" -nologo
- compile option (RX200 big endian)
  Adding “-endian=big” to default option.

2.4 for", "while" and "do while" statements

In this module, "for", "while" and "do while" statements (loop processing) are used in processing to wait for register to be reflected and so on. For these loop processing, comments with "WAIT_LOOP" as a keyword are described. Therefore, if user incorporates fail-safe processing into loop processing, user can search the corresponding processing with "WAIT_LOOP".

The following shows example of description.

```
while statement example :
/* WAIT_LOOP */
while(0 == SYSTEM.OSCOVFSR.BIT.PLOVF)
{
    /* The delay period needed is to make sure that the PLL has stabilized. */
}

for statement example :
/* Initialize reference counters to 0. */
/* WAIT_LOOP */
for (i = 0; i < BSP_REG_PROTECT_TOTAL_ITEMS; i++)
{
    g_protect_counters[i] = 0;
}

do while statement example :
/* Reset completion waiting */
do
{
    reg = phy_read(ether_channel, PHY_REG_CONTROL);
    count++;
} while ((reg & PHY_CONTROL_RESET) && (count < ETHER_CFG_PHY_DELAY_RESET)); /* WAIT_LOOP */
```

2.5 Specification of API

Specification of JPEG Compress Library APIs are below.

Table 2.2 API (JPEG File Compress Library)

function name	outline
R_compress_jpeg	compress bitmap images into JPEG files

Note: Refer to the JPEG Encoder User's Manual to know details.

Specification of JPEG Encode Library APIs are below.

Table 2.3 API (JPEG Encode Library)

function name	outline
R_jpeg_add_quant_table	Registers Quantization table
R_jpeg_DCT	Executes DCT and quantization
R_jpeg_encode_one_block	Executes Huffman encoding
R_jpeg_writeDRI	Writes DRI
R_jpeg_writeRST	Writes RSTm
R_jpeg_writeEOI	Writes EOI
R_jpeg_flush_bits	Forcibly writes Huffman encoded data

Note: Refer to the JPEG Encoder User's Manual to know details.

2.6 ROM size / RAM size / Stack size

JPEG Encoder requires ROM/RAM/Stack size as below.

Table 2.4 ROM/RAM size (JPEG File Compress Library)

kind	section name	Attribute, Alignment	size [byte]	
			RX600, RX200 little endian	RX600, RX200 big endian
ROM	P_jpeg_cmp_F (*)	code	3005	2990
	P_jpeg_cmp_S	code	2980	2980
	C_jpeg_cmp_F (*) C_jpeg_cmp_F_2 (*)	data, align=4 data, align=2	16 256	16 256
	C_jpeg_cmp_S	data, align=4	388	388
Total	-	-	6645	6630
RAM	B_jpeg_cmp_F (*) B_jpeg_cmp_F_2 (*) B_jpeg_cmp_F_1 (*)	data, align=4 data, align=2 data, align=1	1164 128 384	1164 128 384
Total	-	-	1676	1676

Note: (*): Recommends arranging to a high speed memory.

Table 2.5 ROM/RAM size (JPEG Encode Library)

Kind	section name	Attribute, Alignment	size [byte]	
			RX600, RX200 little endian	RX600, RX200 big endian
ROM	P_jpeg_enc_F (*) P_jpeg_enc_F_8 (*)	code code, align=8	1050 644	1050 644
	P_jpeg_enc_S	code	203	204
	C_jpeg_enc_F (*) C_jpeg_enc_F_1 (*)	data, align=4 data, align=1	2536 64	2536 64
	C_jpeg_enc_S	data, align=4	828	828
Total	-	-	5325	5326
RAM	-	-	0	0
Total	-	-	0	0

Note: (*): Recommends arranging to a high speed memory.

Table 2.6 Stack size (JPEG File Compress Library)

API function name	stack size [byte]	
	RX600, RX200 little endian	RX600, RX200 big endian
R_compress_jpeg	296	296

Table 2.7 Stack size (JPEG Encode Library)

API function name	stack size [byte]	
	RX600, RX200 little endian	RX600, RX200 big endian
R_jpeg_add_quant_table	36	36
R_jpeg_DCT	4	4
R_jpeg_encode_one_block	60	60
R_jpeg_writeDRI	12	12
R_jpeg_writeRST	36	36
R_jpeg_writeEOI	8	8
R_jpeg_flush_bits	24	24

2.7 Version information

Version information is stored in this library. Version information can be accessed if the header of this library is included. The data stored in this library is as follows.

2.7.1 RX600 (little endian)

JPEG File Compress Library (compress_jpege_rx600_little.lib)

```
#include "r_compress_jpegd.h"

Version infomation of a Library
  R_compress_jpege_version.library[] =
    "JPEG File Compress Library version 1.00 for the RX600 LITTLE endian.(Sep 12 2013, 11:40:46)"

Version infomation of a compiler
  R_expand_jpegd_version.complier = 0x01020100
```

JPEG Encode Library (jpege_rx600_little.lib)

```
#include "r_jpeg.h"

Version infomation of a Library
  R_jpege_version.library[] =
    "JPEG Encode Library version 1.01 for the RX600 LITTLE endian.(Feb 18 2016, 21:07:57)"

Version infomation of a compiler
  R_jpegd_version.complier = 0x01020100
```

2.7.2 RX600 (big endian)

JPEG File Compress Library (compress_jpege_rx600_big.lib)

```
#include "r_compress_jpege.h"

Version infomation of a Library
  R_compress_jpege_version.library[] =
    "JPEG File Compress Library version 1.00 for the RX600 BIG endian.(Sep 12 2013, 11:40:42)"

Version infomation of a compiler
  R_expand_jpegd_version.complier = 0x01020100
```

JPEG Encode Library (jpege_rx600_big.lib)

```
#include "r_jpeg.h"

Version infomation of a Library
  R_jpege_version.library[] =
    "JPEG Encode Library version 1.01 for the RX600 BIG endian.(Feb 18 2016, 21:08:09)"

Version infomation of a compiler
  R_jpegd_version.complier = 0x01020100
```

2.7.3 RX200 (little endian)

JPEG File Compress Library (compress_jpege_rx200_little.lib)

```
#include "r_compress_jpege.h"

Version infomation of a Library
  R_compress_jpege_version.library[] =
    "JPEG File Compress Library version 1.00 for the RX200 LITTLE endian.(Sep 12 2013, 11:40:37)"

Version infomation of a compiler
  R_expand_jpegd_version.complier = 0x01020100
```

JPEG Encode Library (jpege_rx200_little.lib)

```
#include "r_jpeg.h"

Version infomation of a Library
  R_jpege_version.library[] =
    "JPEG Encode Library version 1.01 for the RX200 LITTLE endian.(Feb 18 2016, 21:08:15)"

Version infomation of a compiler
  R_jpegd_version.complier = 0x01020100
```

2.7.4 RX200 (big endian)

JPEG File Compress Library (compress_jpege_rx200_big.lib)

```
#include "r_compress_jpege.h"

Version infomation of a Library
  R_compress_jpege_version.library[] =
    "JPEG File Compress Library version 1.00 for the RX200 BIG endian.(Sep 12 2013, 11:40:33)"

Version infomation of a compiler
  R_expand_jpegd_version.complier = 0x01020100
```

JPEG Encode Library (jpege_rx200_big.lib)

```
#include "r_jpeg.h"

Version infomation of a Library
  R_jpege_version.library[] =
    "JPEG Encode Library version 1.01 for the RX200 BIG endian.(Feb 18 2016, 21:08:22)"

Version infomation of a compiler
  R_jpegd_version.complier = 0x01020100
```

3. Usage of Libraries

3.1 Usage of JPEG File Compress library

Please link the library file to user application.

RX600 little endian:	jpege_rx600_little.lib, compress_jpege_rx600_little.lib
RX600 big endian:	jpege_rx600_big.lib, compress_jpege_rx600_big.lib
RX200 little endian:	jpege_rx200_little.lib, compress_jpege_rx200_little.lib
RX200 big endian:	jpege_rx200_big.lib, compress_jpege_rx200_big.lib

Please include the header files with library.

```
#include "r_compress_jpege.h"
```

3.2 Usage of JPEG Encode Library

Please link the library file to user application.

RX600 little endian:	jpege_rx600_little.lib
RX600 big endian:	jpege_rx600_big.lib
RX200 little endian:	jpege_rx200_little.lib
RX200 big endian:	jpege_rx200_big.lib

Please include the header files with library.

```
#include "r_jpeg.h"
```

4. Adding Library to Your Project

The lib folder has all Libraries for RX Family. Add the required library for your project.

4.1 Adding Library to CS+ Project

Please refer to the Adding Firmware Integration Technology Modules to Projects.

- Adding Firmware Integration Technology Modules to CS+ Projects (R01AN1826)

Adding Library

(1) Adding FIT modules to projects.

(2) Open: Project Tree >> File >> [FIT Module name] >> lib.

(3) All Libraries will be linked for building. Please remove the Libraries excluding your needing libraries. Right click on the removed library file name, select “Remove from Project”.

4.2 Adding Library to e² studio Project

Please refer to the Adding Firmware Integration Technology Modules to Projects.

- Adding Firmware Integration Technology Modules to Projects (R01AN1723)

Adding Library

(1) Adding FIT modules to projects.

(2) Right click on the project name in the Project Explorer, then choose “Properties” from the pop-up menu.

(3) The “Properties” dialog is shown. The right side of this dialog changes as you select categories on the left. Select the following in order:

”C/C++ Build” >> “Settings” >> “Linker” >> “Input”

Click the “Add…” button.

(4) Adding library.

5. Notes

- This library uses DSP instructions. Please push/pop accumulator register (ACC) in user interrupts function using accumulator, because DSP instructions uses accumulator (RX600).

- This library can be used with Microcontroller Options fint_register=0 (Fast interrupt vectorregister [None]). The default for this option is fint_register=0.

- This library is using Application Notes "Color Space Conversion Using the DSP Instructions" [R01AN0225EJ0100]. Refer to it to know details.

6. Software Update Information

Package version	Date	Description
V.1.00 Release 00	Mar 17, 2015	First release
V.1.01 Release 00	Apr 01, 2016	Updated version number with the xml file revision.
V.1.02 Release 00	Nov 15, 2024	Added support for adding WAIT_LOOP comments.

Website and Support

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

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

All trademarks and registered trademarks are the property of their respective owners.

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Mar.17.15	—	First edition issued
1.01	Apr.01.16	—	Update the xml file for FIT
1.02	Nov.15.24	6	Added 2.4“for”, “while” and “do while” statements

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

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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. 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.5.0-1 October 2020)

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/.

© 2024 Renesas Electronics Corporation. All rights reserved.