I2C Single-Master Communication Sample Code (Using CMSIS Driver Package) for RE01 1500KB Group, 256KB Group

I2C Sample Code Using CMSIS Driver Package

Summary

This application note describes a sample code using the RE01 1500KB Group, RE01 256KB Group CMSIS driver package. The sample code can be found in the project delivered with this application note.

The overview of this sample code is shown in the table below.

Table Overview of Sample Code

Overview of Sample Code Operation	Peripheral Module Mainly Used	Driver Module Mainly Used
Transfers ROM data to EEPROM using the I2C driver.	RIIC	I2C Driver

Target Device

RE01 1500KB Group

RE01 256KB Group

Note

When applying the sample code covered in this application note to another microcomputer, modify the code according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Related Document

RE01 1500KB, 256KB Group Startup Guide to Development Using CMSIS Package (R01AN4660)

Contents

1. Specifications

1.1 Description of Project

The following sample code projects are provided with this application note.

Sample code project for RE01 1500KB group : an4697_cmsis_iic_re

Sample code project for RE01 256KB group : r01an4697_cmsis_iic_re_256kb

The an4697_cmsis_iic_re project has been tested using the Evaluation Kit RE01 1500KB. This project is configured to match the settings of R7F0E015D2CFB mounted on the Evaluation Kit RE01 1500KB.

The r01an4697_cmsis_iic_re_256kb project has been tested using the Evaluation Kit RE01 256KB. This project is configured to match the settings of R7F0E01182CFP mounted on the Evaluation Kit RE01 256KB.

When using another device, change the device settings in the project to those of the target device.

1.2 Pins Used

The pins used by the sample code are shown below.

Table 1-1 The pins used by the sample code for RE01 1500KB group

Pin Used	Purpose of Use
P008	LED1
P009	LED0
P700	SDA1
P701	SCL1

Table 1-2 The pins used by the sample code for RE01 256KB group

Pin Used	Purpose of Use
P410	LED1
P210	LED0
P700	SDA1
P701	SCL1

1.3 Folder Structure

The folder structure of the sample code is shown below.

```
I2C sample program
 ├─CMSIS
 │  └─ Driver
 │      └─ Include
 │          └─ Driver_I2C.h: CMSIS driver header file
 ├─Device
 │  ├─ CMSIS_Driver
 │  │  ├─ Include
 │  │  │  ├─ R_Driver_I2C.h: CMSIS driver extended header file
 │  │  │  └─ r_i2c_cmsis.h: R_I2C driver header file
 │  │  └─ Src
 │  │      └─ r_i2c_cmsis.c: R_I2C driver source file
 │  ├─ Config
 │  │  ├─ r_i2c_cfg.h: R_I2C configuration definition file
 │  │  ├─ r_lpm_cfg.h: R_LPM configuration definition file
 │  │  └─ r_system_cfg.h: System configuration definition file
 │  ├─ Driver
 │  │  ├─ Include
 │  │  │  ├─ r_lpm_api.h: R_LPM driver header file
 │  │  │  └─ r_system_api.h: System driver header file
 │  │  └─ Src
 │  │      ├─ r_lpm
 │  │      │   └─ r_lpm_api.c: R_LPM driver source file
 │  │      └─ r_system
 │  │          └─ r_system_api.c: System driver source file
 │  ├─ pin.c: Pin setting file
 │  └─ pin.h: Pin setting header file
 │
 └─ Src
     └─ main.c: Sample program source file
```

Figure 1.1 Folder Structure

1.4 File Configuration

Table 1-3 shows the files that are added or modified for this sample code.

Table 1-3 Files Added or Modified for this Sample Code

File Name	Overview of Processing or Configuration	Remarks
main.c	Main processing	
r_system_cfg.h	System configuration	Registering I2C transmit interrupt to NVIC
r_i2c_cfg.h	I2C-related configuration	

1.5 Option-Setting Memory

Table 1-4 shows the option-setting memory setting for the sample code. Set suitable values for a user system if required.

Table 1-4 Option-Setting Memory Setting for Sample Code

Symbol	Address	Setting	Description
AWS	0100A164h to 0100A167h	FFFF FFFFh	No access window settings
OSIS	0100A150h to 0100A15Fh	FFFF FFFFh	No ID code protection (All FFh)
SECMPUxxx	00000408h to 0000043Bh	FFFF FFFFh	MPU is disabled.
OFS1	00000404h to 00000407h	FFFF FFFFh	After a reset, the voltage monitor 0 reset is disabled. After a reset, HOCO oscillation is disabled.
OFS0	00000400h to 00000403h	FFFF FFFFh	Automatic activation of IWDT is disabled. Automatic activation of WDT is disabled.

2. Operating Conditions

The operation of the sample code provided with this application note has been tested under the following conditions (Table 2-1, Table 2-2).

Table 2-1 Operating Conditions for RE01 1500KB group

Item		Description
Microcontroller used		R7F0E015D2CFB 144pin
Operating frequency	PLL is selected as the system clock	• Main clock: 32 MHz • PLL: 64 MHz (main clock frequency is divided by 4 and then multiplied by 8) • System clock (ICLK): 64 MHz (PLL) • Peripheral module clock A (PCLKA): 64 MHz (PLL frequency is not divided) • Peripheral module clock B (PCLKB): 32 MHz (PLL frequency is divided by 2)
Operating voltage		• 3.3V
Target board		Evaluation Kit RE01 1500KB (RTK70E015DSXXXXXBE)
Integrated Development Environment	GCC	Renesas e^2 studio Version 7
	IAR	IAR Embedded Workbench for ARM Version 8.32
C compiler	GCC	GCC ARM Embedded Version 6.3.1.20170620 GNU 6-2017-q2-update
	IAR	IAR C/C++ Compiler for ARM Version 8.32
Debugger		Segger J-Link OB
I/O header Version		Rev1.00
Sample code Version		Rev1.00

Table 2-2 Operating Conditions for RE01 256KB group

Item		Description
Microcontroller used		R7F0E01182CFP 100pin
Operating frequency	PLL is selected as the system clock	• HOCO: 64MHz • System clock (ICLK): 64 MHz (HOCO) • Peripheral module clock A (PCLKA): 64 MHz (HOCO is not divided) • Peripheral module clock B (PCLKB): 32 MHz (HOCO is divided by 2)
Operating voltage		• 3.3V
Target board		Evaluation Kit RE01 256KB (RTK70E0118CXXXXXBJ)
Integrated Development Environment	GCC	Renesas e^2 studio 2020-07
	IAR	IAR Embedded Workbench for ARM Version 8.40.2
C compiler	GCC	GCC ARM Embedded Version 6.3.1.20170620 GNU 6-2017-q2-update
	IAR	IAR C/C++ Compiler for ARM Version 8.40.2
Debugger		Segger J-Link OB
I/O header Version		Rev1.00
Sample code Version		Rev1.03

3. Description of Software

This sample code performs single-master communication with EEPROM using the I2C driver.

EEPROM (ATMEL AT24C256) is connected to RIIC.

The sample code performs the following operations.

- Sets the slave address and bit rate after released from the reset state.
- Performs a 5-byte page write to EEPROM as follows:
  - In the master transmit mode, sends a 2-byte EEPROM address (0x0000) and five bytes of data to be written to EEPROM to the device address (0x50).
- Waits for the EEPROM writing time (10 ms).
- Performs a 5-byte sequential read from EEPROM as follows:
  - The master transmit mode in the pending mode, sends a 2-byte EEPROM address to the device address (0x50).
  - After transmission has finished, enters the master receive mode and receives five bytes of data from the device address (0x50).
- Compares the write data with the read data and turns LED1 on if they match.

Table 3-1 Information of Sample Program Operation (I2C)

Item	Setting
Transfer rate	100 kbps
Size of data read from or written to EEPROM	5 bytes

3.1 System Configuration

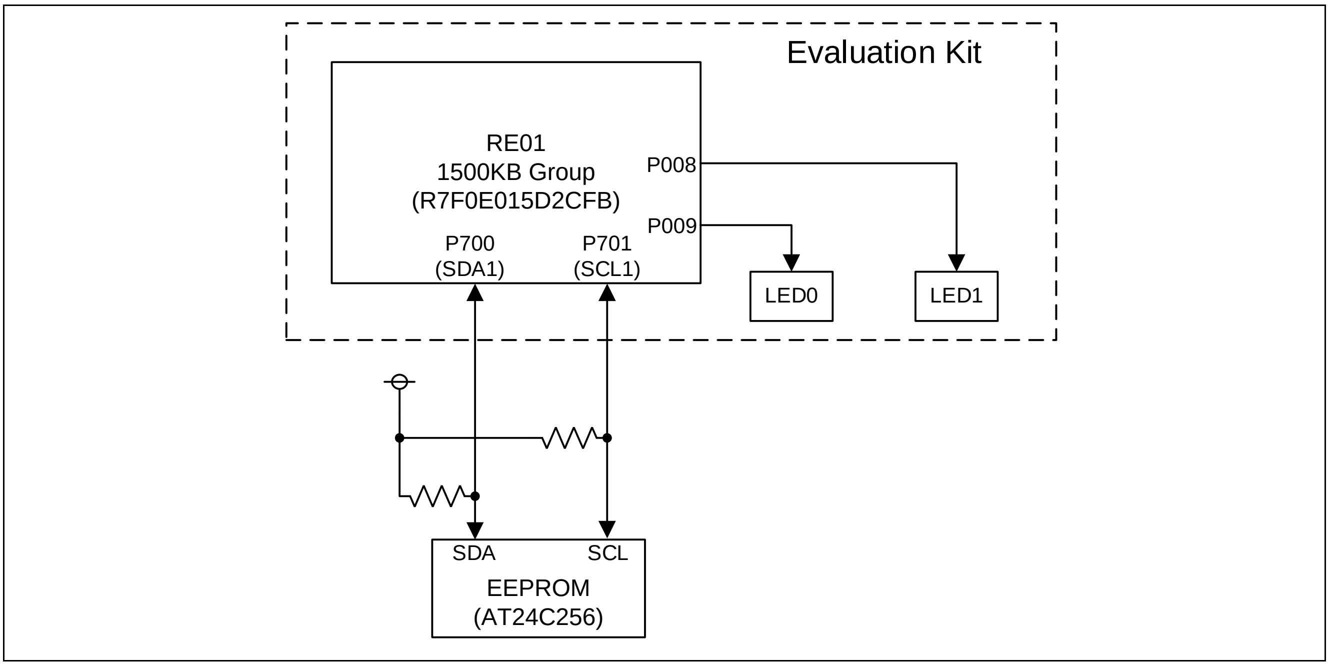


Figure 3.1 System Configuration for RE01 1500KB group

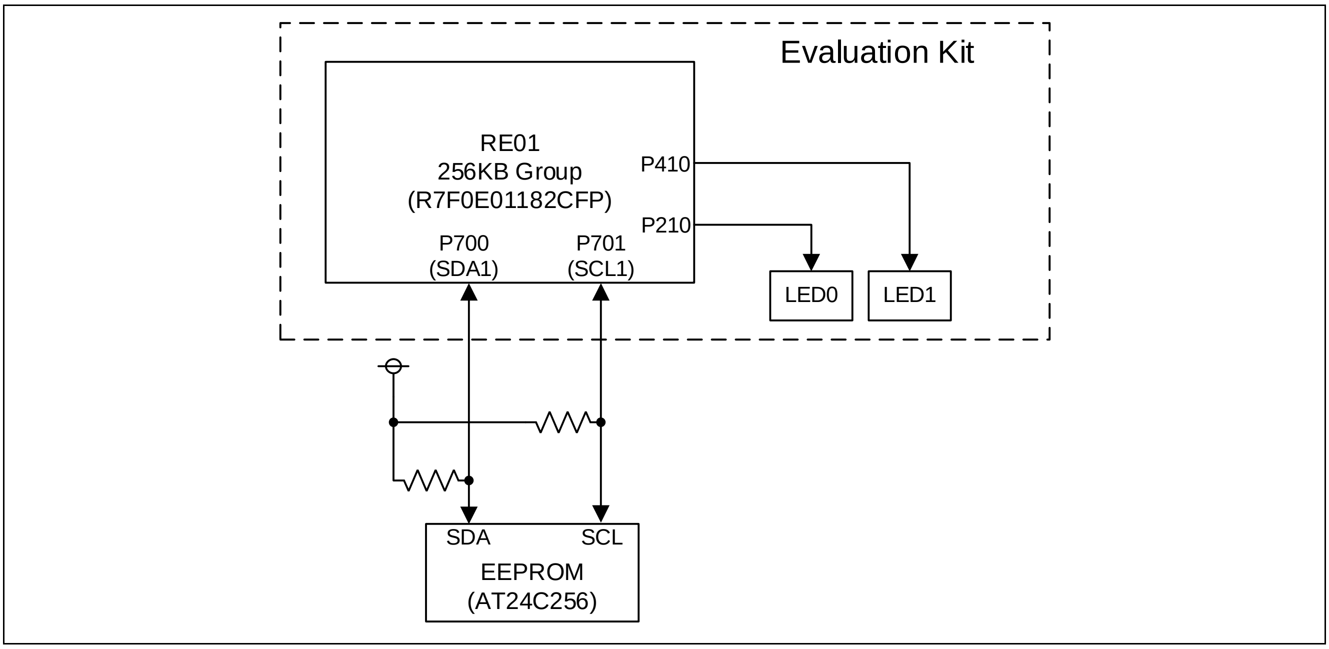


Figure 3.2 System Configuration for RE01 256KB group

3.2 Driver Configuration

Table 3-2 Driver Configuration

Item	Location of Change	Details of Change
Registering I2C receive data full interrupt to NVIC	[r_system_cfg.h] SYSTEM_CFG_EVENT_ NUMBER_IIC1_RXI	• Setting change. (SYSTEM_IRQ_EVENT_NUMBER0)
Registering I2C transmit end interrupt to NVIC	[r_system_cfg.h] SYSTEM_CFG_EVENT_ NUMBER_IIC1_TEI	• Setting change. (SYSTEM_IRQ_EVENT_NUMBER4)
Registering I2C transmit data empty interrupt to NVIC	[r_system_cfg.h] SYSTEM_CFG_EVENT_ NUMBER_IIC1_TXI	• Setting change. (SYSTEM_IRQ_EVENT_NUMBER1)
Registering I2C communication error/event generation interrupt to NVIC	[r_system_cfg.h] SYSTEM_CFG_EVENT_ NUMBER_IIC1_EEI	• Setting change. (SYSTEM_IRQ_EVENT_NUMBER5)
Set P701 as SCL1 pin. (Only RE01 256KB Group)	[pin.c] R_RIIC_Pinset_CH1() function	• Validate followings PFS->P701PFS_b.PMR = 0U; PFS->P701PFS_b.ASEL = 0U; PFS->P701PFS_b.ISEL = 0U; PFS->P701PFS_b.DSCR = 0U; PFS->P701PFS_b.PSEL = R_PIN_PRV_RIIC_PSEL; PFS->P701PFS_b.PMR = 1U;
	[pin.c] R_RIIC_Pinclr_CH1() function	• Validate followings PFS->P701PFS &= R_PIN_PRV_CLR_MASK;
Set P700 as SDA1 pin. (Only RE01 256KB Group)	[pin.c] R_RIIC_Pinset_CH1() function	• Validate followings PFS->P700PFS_b.PMR = 0U; PFS->P700PFS_b.ASEL = 0U; PFS->P700PFS_b.ISEL = 0U; PFS->P700PFS_b.DSCR = 0U; PFS->P700PFS_b.PSEL = R_PIN_PRV_RIIC_PSEL; PFS->P700PFS_b.PMR = 1U;
	[pin.c] R_RIIC_Pinclr_CH1() function	• Validate followings PFS->P700PFS &= R_PIN_PRV_CLR_MASK;

3.3 List of Functions

The functions added to the sample code are described here.

main

Overview	Main processing
Header	None
Declaration	void main (void)
Description	This function calls the system initialization function. Then, it sets up I2C communication and performs page write and sequential read to EEPROM. After the reading has finished, this function compares the transmit data and receive data.
Argument	None
Return Value	None

system_init

Overview	System initialization processing
Header	None
Declaration	static void system_init (void)
Description	This function initializes sections, the system, the R_LPM driver, and calls the IO power supply setting function.
Argument	None
Return Value	None

iic_callback

Overview	I2C transfer end callback processing	
Header	None	
Declaration	static void iic_callback (uint32_t event)	
Description	After the end of I2C communication, this function sets the transmission flag to 1 if the transmission has finished and sets the reception flag to 1 if the reception has finished.	
Argument	uint32_t event	Cause of callback ARM_I2C_EVENT_TRANSFER_DONE (Transfer end)
Return Value	None	

3.4 List of Constants

Table 3-3 shows a list of constants.

Table 3-3 Constants (User Changeable) Used in Sample Code

Constant Name	Setting	Description
IIC_DEVICE_ADDRESS	0x50	Device address for EEPROM
IIC_TRANSFER_DATA_SIZE	7	Transmit data size
IIC_RECEIVE_DATA_SIZE	5	Receive data size
IIC_DATA_ADDRESS_SIZE	2	Data address size

3.5 Flowcharts

Figure 3.3 shows a flowchart of the main processing.

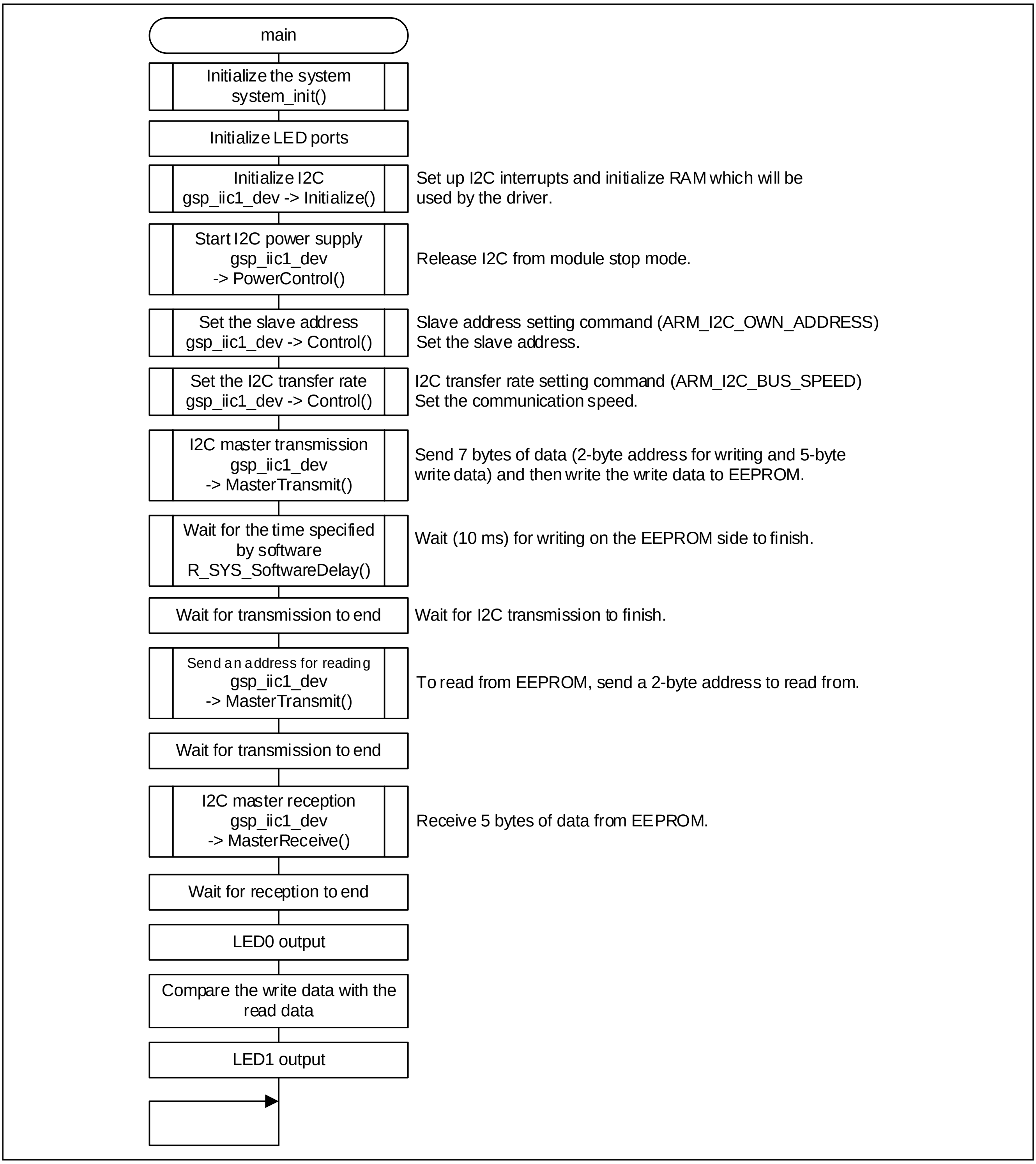


Figure 3.3 Main Processing

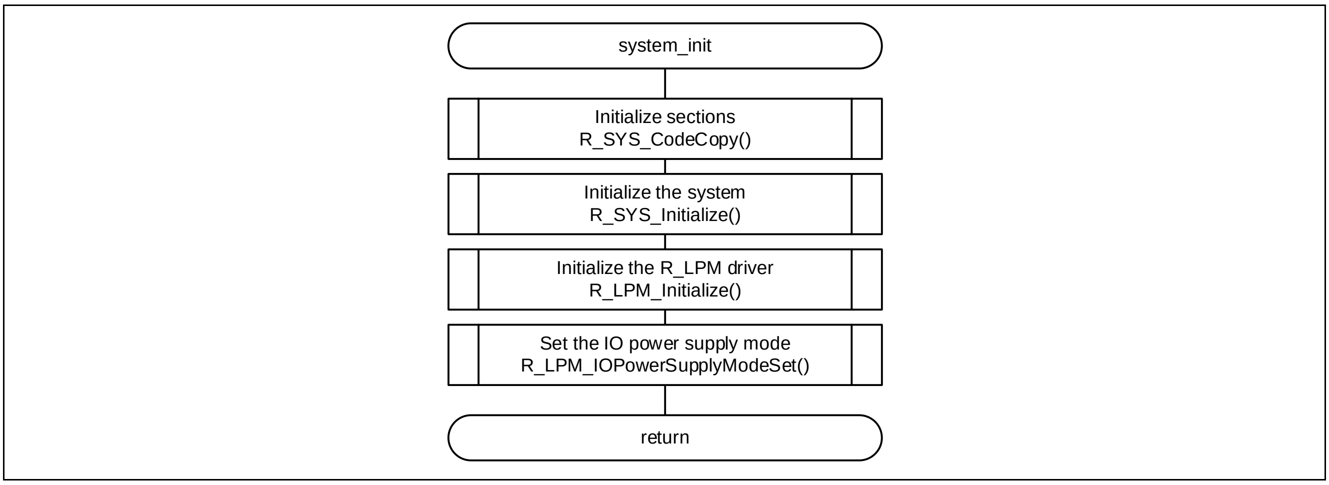


Figure 3.4 System Initialization Processing

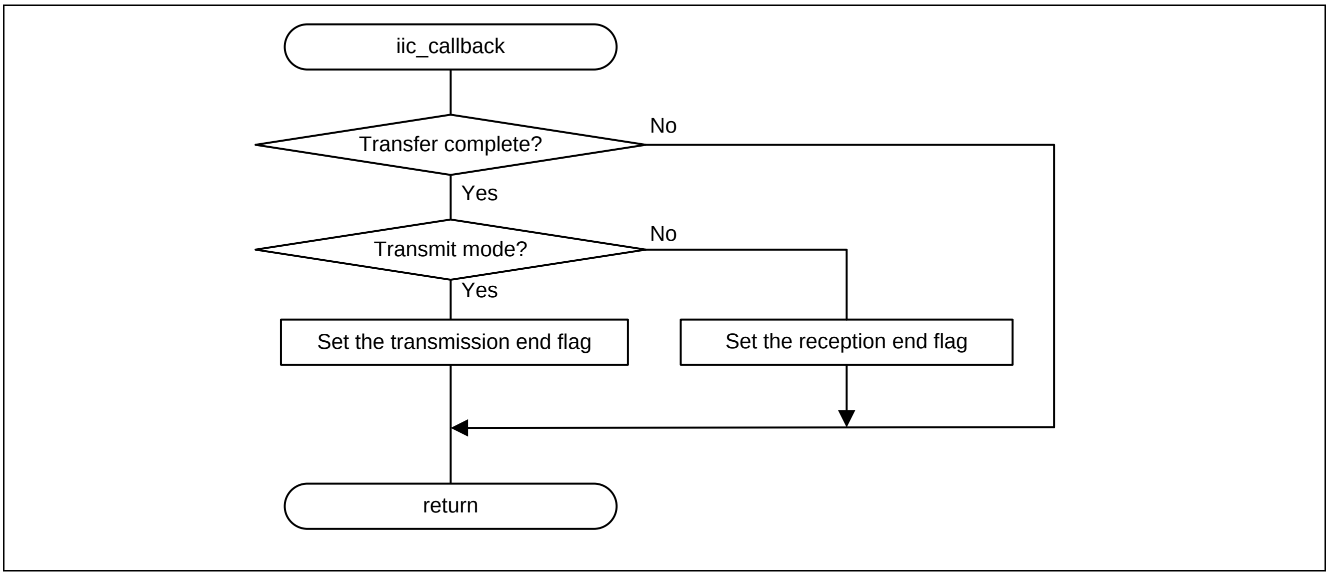


Figure 3.5 I2C Transfer End Callback Processing

4. Specifications of Driver APIs

4.1 External Specification

This driver contains documents that describes the external API specification. These files are contained in the Driver Specification folder within the Documents.

5. Usage Notes of I2C Driver

This chapter introduces the main points to concern regarding the I2C driver. Note that not all notes are given here.

For other notes, see the external specification document described in "4 Specifications of Driver APIs".

5.1 Registering Interrupts to NVIC

The I2C driver uses the receive data full interrupt (RXI), transmit end interrupt (TEI), transmit data empty interrupt (TXI), and communication error/event generation interrupt (EEI) in the transmit processing or receive processing. Register the interrupts to the NVIC in r_system_cfg.h.

Figure 5.1 shows an example of registering interrupts to the NVIC.

```
...
#define SYSTEM_CFG_EVENT_NUMBER_ADC140_WCMPM
     (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)  /*!< Numbers 0/4/8/12/16/20/24/28 only */
#define SYSTEM_CFG_EVENT_NUMBER_IIC0_RXI
     (SYSTEM_IRQ_EVENT_NUMBER0)  /*!< Numbers 0/4/8/12/16/20/24/28 only */
#define SYSTEM_CFG_EVENT_NUMBER_CCC_PRD
     (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)  /*!< Numbers 0/4/8/12/16/20/24/28 only */
...
#define SYSTEM_CFG_EVENT_NUMBER_ADC140_WCMPUM
     (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)  /*!< Numbers 1/5/9/13/17/21/25/29 only */
#define SYSTEM_CFG_EVENT_NUMBER_IIC0_TXI
     (SYSTEM_IRQ_EVENT_NUMBER1)  /*!< Numbers 1/5/9/13/17/21/25/29 only */
#define SYSTEM_CFG_EVENT_NUMBER_DOC_DOPCI
     (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)  /*!< Numbers 1/5/9/13/17/21/25/29 only */
...
#define SYSTEM_CFG_EVENT_NUMBER_ADC140_GCADI
     (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)  /*!< Numbers 2/6/10/14/18/22/26/30 only */
#define SYSTEM_CFG_EVENT_NUMBER_IIC0_TEI
     (SYSTEM_IRQ_EVENT_NUMBER2)  /*!< Numbers 2/6/10/14/18/22/26/30 only */
#define SYSTEM_CFG_EVENT_NUMBER_CAC_MENDI
     (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)  /*!< Numbers 2/6/10/14/18/22/26/30 only */
...
#define SYSTEM_CFG_EVENT_NUMBER_ACMP_CMPI
     (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)  /*!< Numbers 3/7/11/15/19/23/27/31 only */
#define SYSTEM_CFG_EVENT_NUMBER_IIC0_EEI
     (SYSTEM_IRQ_EVENT_NUMBER3)  /*!< Numbers 3/7/11/15/19/23/27/31 only */
#define SYSTEM_CFG_EVENT_NUMBER_CAC_OVFI
     (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)  /*!< Numbers 3/7/11/15/19/23/27/31 only */
```

Figure 5.1 Example of Registering Interrupts to NVIC (When Using RIIC0)

5.2 Note When Automatic Calculation of Bus Speed Is Enabled

When automatic calculation of the bus speed is enabled (RIIC_BUS_SPEED_CAL_ENABLE = 1), values that will satisfy the I2C bus specifications are calculated from the PCLKB frequency at the time the bus speed is set, and registers are set accordingly. The rise time and fall time of the SCL line are also considered at calculation. Therefore, the I2C-bus speed has to be recalculated when PCLKB is changed.

Table 5-1 shows the prescribed values of the SCL clock in the I2C bus specifications and the rise time and fall time of the SCL line, which are used for setting the bus speed.

Table 5-1 Prescribed Values for Setting Bus Speed

Bus Speed	Item	Value
Standard mode (100 kbps)	Prescribed low period of SCL clock	4.7 us (min.)
	Prescribed high period of SCL clock	4.0 us (min.)
	Rise time of SCL line	Defined in RIIC_STAD_SCL_UP_TIME (Initial value: 1000 ns)
	Fall time of SCL line	Defined in RIIC_STAD_SCL_DOWN_TIME (Initial value: 300 ns)
Fast mode (400 kbps)	Prescribed low period of SCL clock	1.3 us (min.)
	Prescribed high period of SCL clock	0.6 us (min.)
	Rise time of SCL line	Defined in RIIC_FAST_SCL_UP_TIME (Initial value: 300 ns)
	Fall time of SCL line	Defined in RIIC_FAST_SCL_DOWN_TIME (Initial value: 300 ns)

The calculated results satisfy the I2C bus specifications, but the error of the bus speed may increase if the PCLKB frequency is low.

Table 5-2 Examples of Register Settings by Automatic Calculation and Error (with 2-Stage Noise Filtering)

Bus Speed	Operating Frequency PCLKB	Register Settings			Expected Output	
		ICMR1. CKS[2:0]	ICBRH. BRH[4:0]	ICBRL. BRL[4:0]	Transfer Rate	Error
100 kbps	32 MHz	011b	12 (0Ch)	15 (0FH)	99.5 kbps	0.5%
	20 MHz	010b	16 (10h)	20 (14h)	99.0 kbps	1.0%
	8 MHz	001b	12 (0Ch)	15 (0Fh)	99.5 kbps	0.5%
	2 MHz	000b	3 (03h)	1 (01H)	120.5 kbps	20.5%
	1 MHz	000b	0 (00h)	0 (00h)	88.5 kbps	11.5%
400 kbps	32 MHz	001b	6 (06h)	17 (11h)	394.1 kbps	1.5%
	20 MHz	000b	7 (07h)	21 (15h)	400.0 kbps	0.0%
	8 MHz	000b	0 (00h)	5 (05h)	404.0 kbps	1.0%
	2 MHz	000b	0 (00h)	0 (00h)	178.6 kbps	55.4%
	1 MHz	000b	0 (00h)	0 (00h)	94.3 kbps	76.4%

Note: The transfer rate in the Expected Output column is calculated by the following formula.
tr: Rise time of SCL line, tf: Fall time of SCL line, nf: Number of stages of noise filtering
At standard speed: tr = 1000 ns, tf = 300 ns
At fast speed: tr = 300 ns, tf = 300 ns
(1) For CKS[2:0] = 000b
Transfer rate = 1/{[(BRH + 3 + nf) + (BRL + 3 + nf)]/(PCLKB) + tr + tf}
(2) For CKS[2:0] ≠ 000b
Transfer rate = 1/{[(BRH + 2 + nf) + (BRL + 2 + nf)]/ (PCLKB/Division ratio) + tr + tf}

Table 5-3 Examples of Register Settings by Automatic Calculation and Error (without Noise Filtering)

Bus Speed	Operating Frequency PCLKB	Register Settings			Expected Output	
		ICMR1. CKS[2:0]	ICBRH. BRH[4:0]	ICBRL. BRL[4:0]	Transfer Rate	Error
100 kbps	32 MHz	011b	14 (0Eh)	17 (11H)	99.5 kbps	0.5%
	20 MHz	010b	18 (12h)	22 (16h)	99.0 kbps	1.0%
	8 MHz	001b	14 (0Eh)	17 (11h)	99.5 kbps	0.5%
	2 MHz	000b	5 (05h)	6 (06H)	102.0 kbps	2.0%
	1 MHz	000b	1 (01h)	2 (02h)	97.1 kbps	2.9%
400 kbps	32 MHz	001b	7 (07h)	18 (12h)	414.5 kbps	3.6%
	20 MHz	000b	9 (09h)	23 (17h)	400.0 kbps	0.0%
	8 MHz	000b	2 (02h)	7 (07h)	404.0 kbps	1.0%
	2 MHz	000b	0 (00h)	0 (00h)	277.8 kbps	30.6%
	1 MHz	000b	0 (00h)	0 (00h)	151.5 kbps	62.1%

Note: The transfer rate in the Expected Output column is calculated by the following formula.
tr: Rise time of SCL line, tf: Fall time of SCL line
At standard speed: tr = 1000 ns, tf = 300 ns
At fast speed: tr = 300 ns, tf = 300 ns
(1) For CKS[2:0] = 000b
Transfer rate = 1/{[(BRH + 3) + (BRL + 3)]/ PCLKB + tr + tf}
(2) For CKS[2:0] ≠ 000b
Transfer rate = 1/{[(BRH + 2) + (BRL + 2)]/ (PCLKB/Division ratio) + tr + tf}

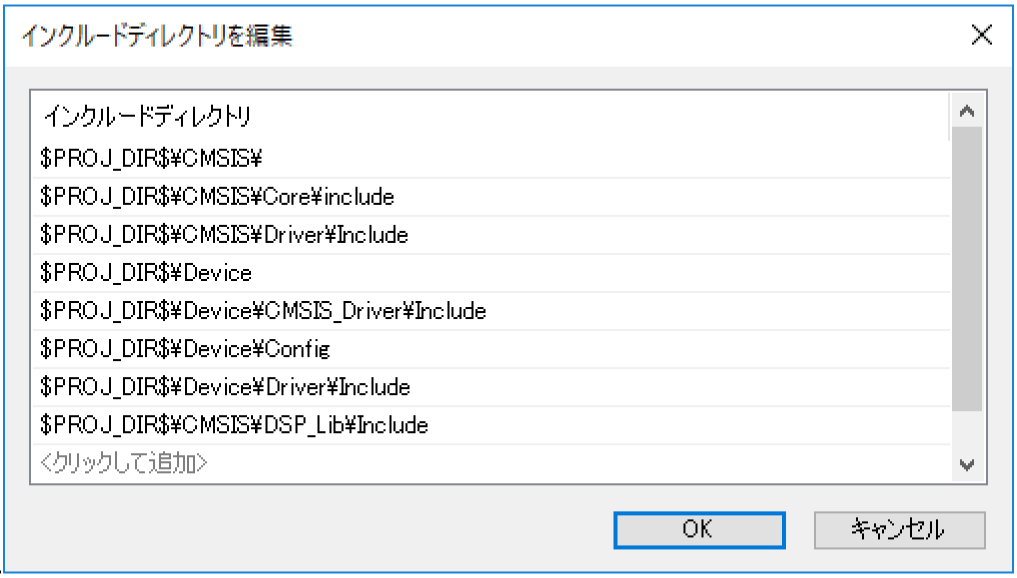
6. Troubleshooting

6.1 Occurrence of Build Error with IAR compiler

A-1) Have the include directories been specified correctly?

When using EWARM, we recommend that the include directories be specified as shown in the example below.

The include directories can be specified from IDE Options [C/C++ Compiler] → [Preprocessor]



6.2 Occurrence of HardFault Error when API of CMSIS Driver Is Called

A) The API has possibly not been copied to RAM.

Before calling an API function that is mapped to RAM, make sure that it has been copied to RAM by the R_SYS_CodeCopy function. For details, refer to the related document No. R01AN4660.

6.3 Peripheral Function Fails to Operate when API Is Called

A) Has the API been set up correctly?

Check the API's return value to see if an error has occurred.

In particular, errors are often caused by problems related to interrupts not being set in r_system_cfg.h. For details, refer to the related document No. R01AN4660.

6.4 Normal API Return Value But No Pin Output from Peripheral Function

A) Are the pin settings correct?

Check to make sure the pins have been set up correctly by the functions in pin.c.

For details, refer to the related document No. R01AN4660.

6.5 Peripheral Function's Input or Output Does Not Operate as Expected

A) Check to make sure the VOCR register has been set up correctly before making the initial settings for peripheral functions.

For details, refer to the related document No. R01AN4660.

7. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

8. Reference Documents

User's Manual: Hardware

RE01 1500KB Group User's Manual: Hardware R01UH0796

RE01 256KB Group User's Manual: Hardware R01UH0894

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

RE01 1500KB, 256KB CMSIS Package Startup Guide

RE01 1500KB, 256KB Group Startup Guide to Development Using CMSIS Package R01AN4660

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

Technical Update/Technical News

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

User's Manual: Development Tools

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

Revision History

Rev.	Date	Description	
		Page	**Summary**
1.00	Sep.19.2019	-	First edition issued
1.01	Jan.17.2020	-	Add RE01 256KB group.
1.02	Mar.19.2020	3,6,8	RE01 256KB group target board changed to Evaluation Kit RE01 256KB
		-	Clerical error correction
		Program (256KB)	Replaced CMSIS Driver Package - RE01 256KB: CMSIS Driver Package Rev.0.80
1.03	Jun.01.2020	3	Change the name of the sample code project
		6	Updated "Operating Conditions"
		Program (256KB)	Replaced CMSIS Driver Package - RE01 256KB: CMSIS Driver Package Rev.1.00

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/.