RZ/T1 Group

NOR Flash Sample Program

R01AN3011EJ0130
Rev.1.30
Jun. 07, 2018

Summary

This application note describes the sample program for reading from, writing to, and erasing the NOR flash memory on the evaluation board of the RZ/T1.

The feature of the NOR sample program:

- The board can be connected to the host computer with a USB cable and data can be read from and written to the NOR flash memory by a menu-driven program.

Restrictions

The following restrictions apply to the sample program.

(1) Only the area of the NOR flash memory is supported. Reading from or writing to other areas is not allowed.

(2) Only data in the memory can be written to the NOR flash memory.

Devices for Checking Operation

RZ/T1

When applying the program covered in this application note to another microcontroller, modify the program according to the specifications for the target microcontroller and extensively evaluate the modified program.

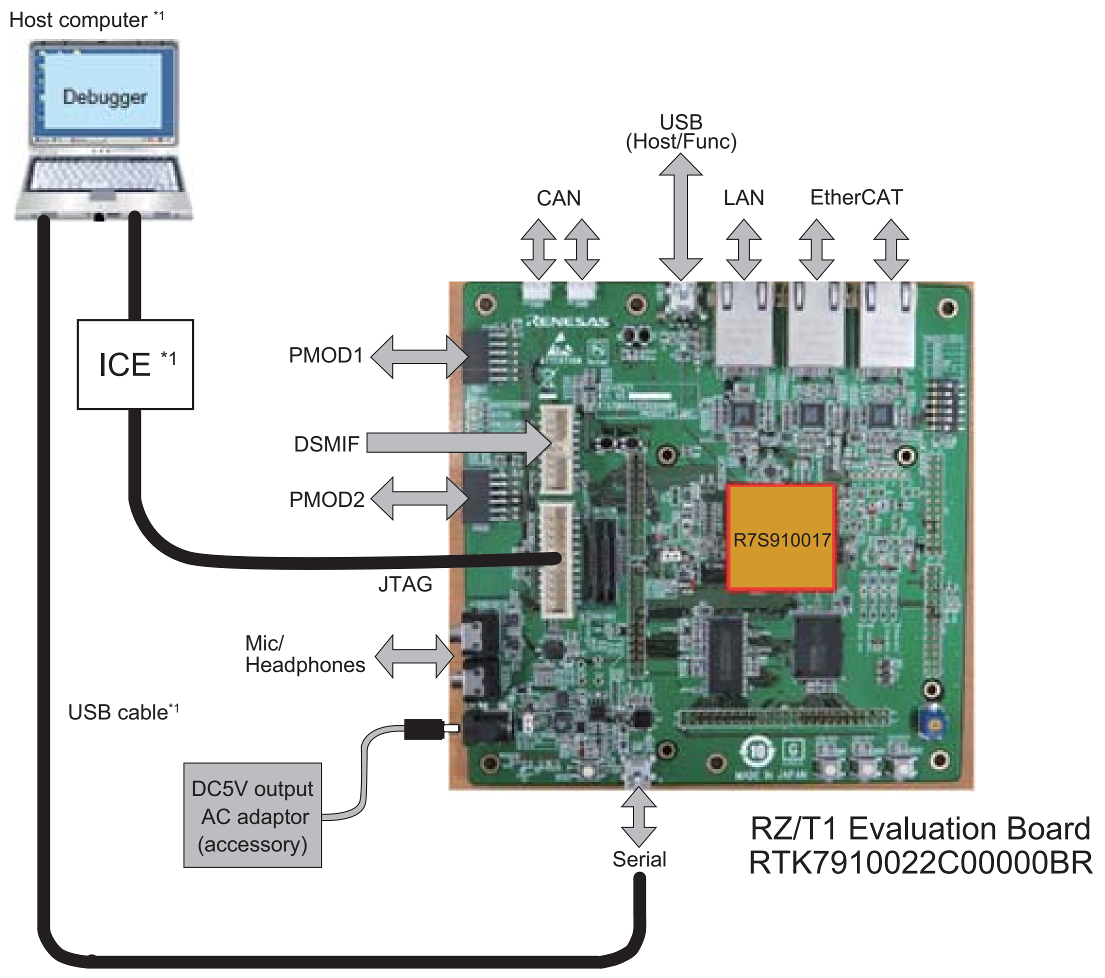
Table of Contents

1. Specifications

Table 1.1 lists the peripheral modules to be used and their applications and Figure 1.1 shows the operating environment when the sample program is being executed.

Table 1.1 Peripheral Modules and Applications

Peripheral Module	Application
Clock pulse generator (CPG)	The CPG produces the CPU clock and low-speed on-chip oscillator clock signals.
FIFO integrated serial communications interface (SCIFA)	Asynchronous communications of the SCIFA is used for COM port communications by using an RS-232C interface



Note 1. Indicates the device that the user needs to prepare.

Figure 1.1 Operating Environment

2. Operating Environment

The sample code covered in this application note is for the environment below.

Table 2.1 Operating Environment

Item	Description
MCU used	RZ/T1 Group
Operating frequency	CPUCLK = 450 MHz
Operating voltage	3.3 V
Integrated development environment	• Embedded Workbench® for Arm Version 8.20.2 from IAR Systems • Arm® Integrated Development Environment Arm Development Studio 5 (DS-5TM) Version 5.26.2 • Renesas e2studio Version: 6.1.0
Operating mode	SPI boot mode 16-bit bus boot mode
Communications settings in the terminal software	• Transfer rate: 115200 bps • Data length: 8 bits • Parity: None • Stop bit length: 1 bit • Flow control: None • New-line code (reception): CR • New-line code (transmission): CR
Board used	RZ/T1 evaluation board (RTK7910022C00000BR)
Devices used (functions to be used on the board)	• Serial interface (USB-Mini B connector J8) • NOR flash memory (connected to the CS0 or CS1 space) Manufacturer: Macronix International Co., Ltd. Product type name: MX29GL512FLT2I-10Q • Serial flash memory Manufacturer: Macronix International Co., Ltd. Product type name: MX25L51245G

3. Related Application Notes

The application notes related to the descriptions in this application note are listed below. Also consult the following documents along with this application note.

- RZ/T1 Group Initial Settings (R01AN2554EJ)
- RZ/T1 Group FIFO Integrated Serial Communication Interface (SCIFA) (R01AN2577EJ)

Note: For registers not covered in this application note, the values set in the RZ/T1 Group Initial Settings and RZ/T1 Group FIFO Integrated Serial Communication Interface application notes are used without change.

4. Peripheral Modules

For the basics of the clock pulse generator (CPG), interrupt controller (ICUA), bus state controller (BSC), SPI multi-I/O controller (SPIBSC), error control module (ECM), reset, and general I/O ports, refer to the RZ/T1 Group User's Manual: Hardware.

5. Hardware

5.1 Example of Hardware Configuration

Figure 5.1 shows an example of the hardware configuration.

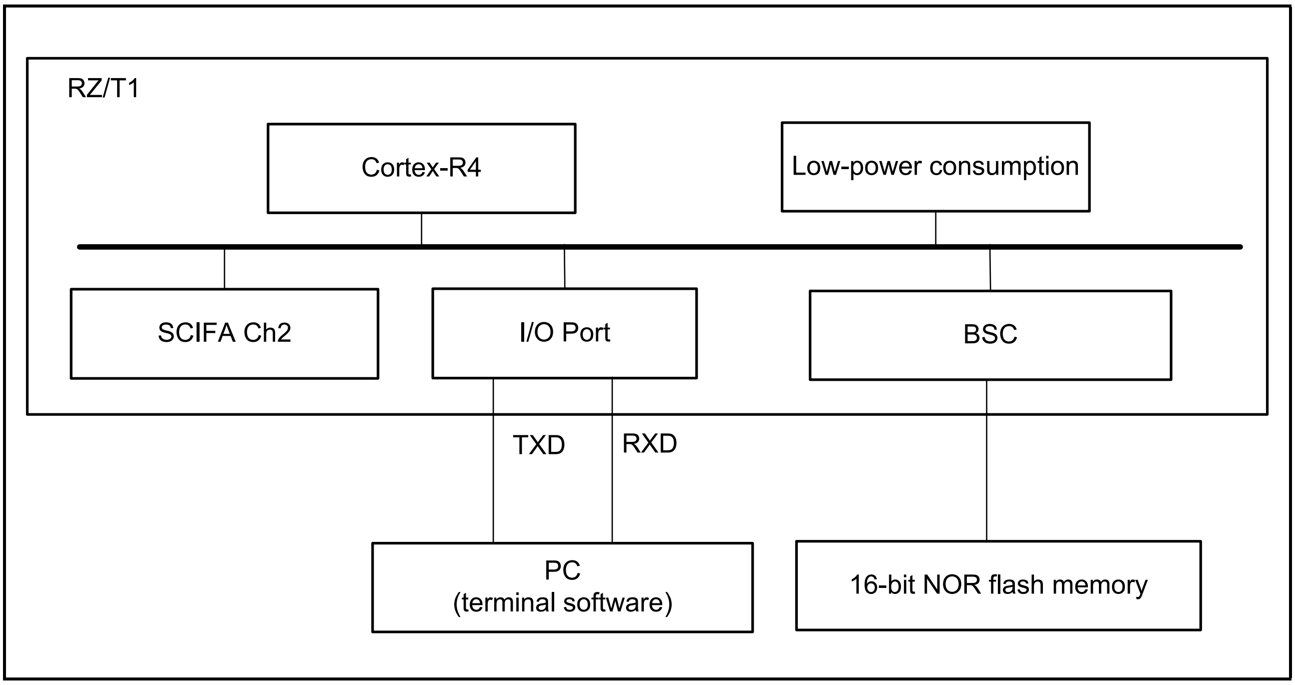


Figure 5.1 Example of Hardware Configuration

5.2 Pins

Table 5.1 lists the pins used and their functions.

Table 5.1 Pins Used and Their Functions

Pin Name	I/O	Description
MD0	Input	Selection of the operating mode MD0 = “L”, MD1 = “L”, MD2 = “L” (SPI boot mode) MD0 = “L”, MD1 = “H”, MD2 = “L” (16-bit bus boot mode)
MD1	Input	
MD2	Input	
TXD2	Output	Serial transmission data signal
RXD2	Input	Serial reception data signal
D15 to D0	I/O	BSC data (for connection of the NOR flash memory)
A25 to A1	Output	BSC addresses (for connection of the NOR flash memory)
RD#	Output	BSC read strobe signal (for connection of the NOR flash memory)
WE#	Output	BSC write strobe signal (for connection of the NOR flash memory)
CS0#	Output	BSC chip select signal (for connection of the NOR flash memory)
CS1#	Output	BSC chip select signal (for connection of the NOR flash memory)

6. Software

6.1 Outline of Operation

This software includes a driver for use of a NOR flash memory and API functions for erasure and programming of the NOR flash memory.

It also includes a sample program for COM port communications between a host PC and an RS-232 interface by using asynchronous communications through a serial communications interface incorporating an FIFO buffer (SCIFA) to read from, write to, and erase the NOR flash memory through terminal software on the host PC.

6.1.1 Project Settings

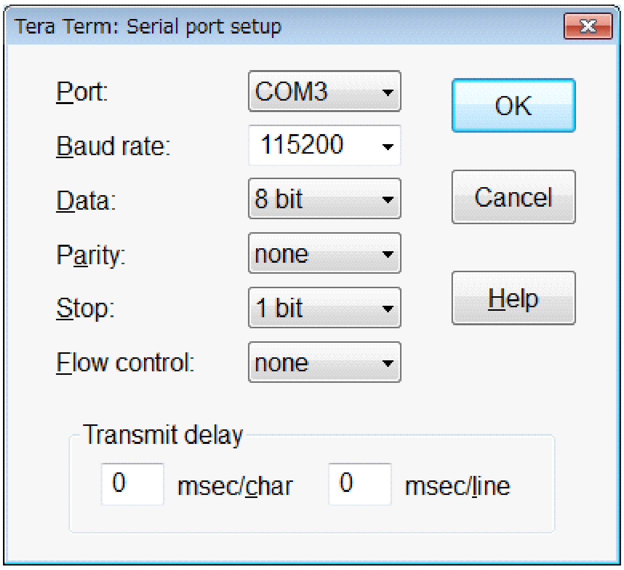
Project settings for use in the EWARM, DS-5, or e2studio as the development environment are described in the RZ/T1 Group Initial Settings application note.

Note: The contents of the sample program have been changed from the initial setting. For details, see Section 6.2.1.

6.1.2 Preparing to Run the Program

This sample program handles processing for transfer to and from the PC and the following describes the preparation for running the program.

(1) Start the terminal software on the host PC and make settings for the serial port as follows (the following is the case for Tera Term on COM3).



Tera Term: Serial port setup

Port: COM3

Baud rate: 115200

Data: 8 bit

Parity: none

Stop: 1 bit

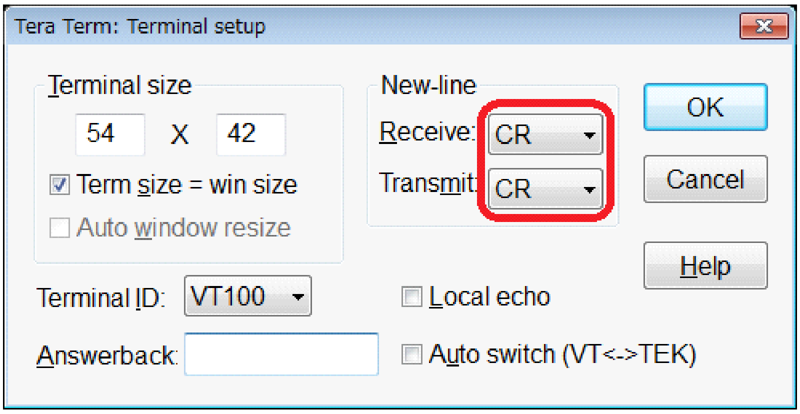
Flow control: none

Transmit delay

0 msec/char 0 msec/line

Figure 6.1 Settings for the Serial Port

(2) In the terminal setup for the terminal software, set the new-line codes for both transmission and reception to “CR”.



Terminal size

54 X 42

☑ Term size = win size

□ Auto window resize

New-line

Receive: CR

Transmit: CR

Terminal ID: VT100

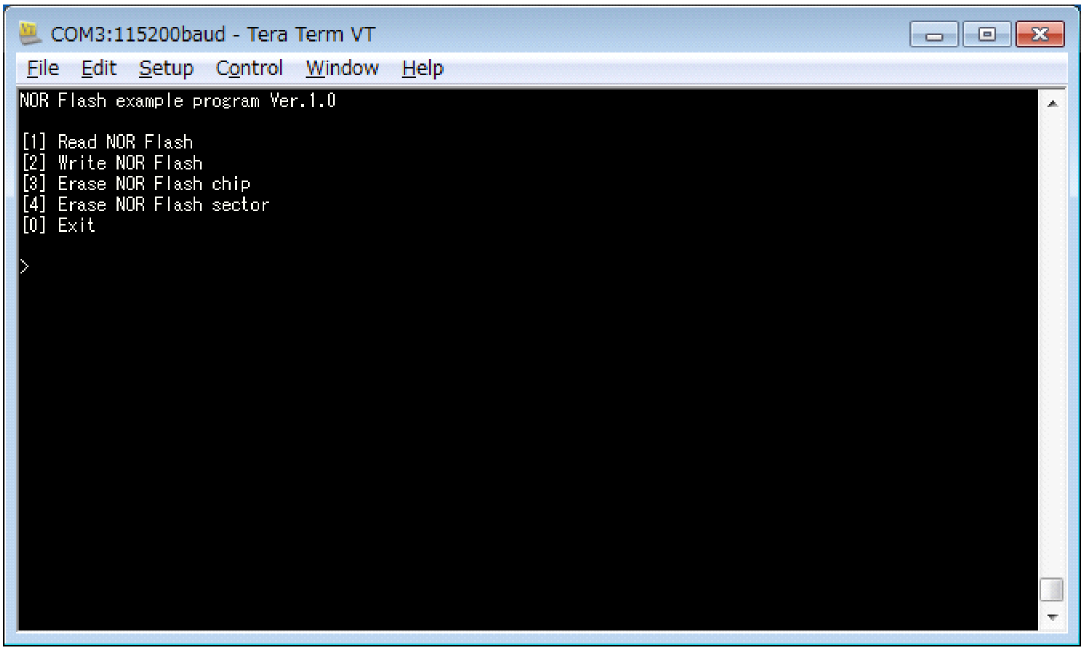
□ Local echo

Answerback:

□ Auto switch (VT<->TEK)

Figure 6.2 Display in the Terminal Software after Setting the SCIFA

(3) When the sample program is run and ready to handle transfer, the data received form the sample program are displayed in the terminal software as shown below.


COM3:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
NOR Flash example program Ver.1.0

[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash chip
[4] Erase NOR Flash sector
[0] Exit

>
```

Figure 6.3 Display in the Terminal Software after the Sample Program is Run

6.2 Memory Map

For the address space of the RZ/T1 Group and a memory map of the RZ/T1 evaluation board, refer to the RZ/T1 Group Initial Settings application note.

6.2.1 Assignment to Sections of Sample Program

Refer to the Application Note: RZ/T1 Group Initial Settings for the sections to be used in the program, assignment to sections (loading view) of the sample program in its initial state, and assignment to sections of the sample program following the application of scatter loading (execution view).

Note: In this sample program, the following have been changed from the initial settings.

1) The heap area defined as the RAM area has been changed to that defined as the SDRAM area.
2) The size of the heap area has been changed to 0x50000.
3) Only in the NOR boot version for the EWARM environment, rodata section defined as the NOR flash area has been changed to that defined as the RAM area.

The following is the description of Note 3) above.

Since the sample program of the NOR boot version for the EWARM environment (RZ_T1_nor_nor_boot.eww) does not update the contents of rodata section when the NOR flash memory is modified, the rodata section is placed in the “A” tightly-coupled memory (ATCM). When placing, "RZ_T1_init_nor_boot.icf" is changed as shown below (bold indicates added text, whereas double-strikethrough indicates deleted text).

```
define block USER_PRG_RBLOCK { ro code**, section .rodata_init** };
define block USER_PRG_WBLOCK { rw code**, section .rodata** };
              :
initialize manually { ro code object loader_init.o,
                      ro code object loader_init2.o,
                      ro code object r_atcm_init.o,
                      ro code object r_cpg.o,
                      ro code object r_ram_init.o,
                      ro code object r_mpc.o,
                      ro code object bus_init_nor_boot.o,
                      ro code object r_reset.o,
                      ro code object vector.o,
                      ro code**,**
                      **section .rodata**
                    };
              :
place in ROM_region { block USER_PRG_RBLOCK~~, section .rodata~~ };
```

6.2.2 MPU Settings

For the MPU settings, refer to the RZ/T1 Group Initial Settings application note.

6.2.3 Exception Processing Vector Table

For the vector table for exception processing, refer to the RZ/T1 Group Initial Settings application note.

6.3 Fixed-Width Integers

Table 6.1 lists the fixed width integers used in the sample program.

Table 6.1 Fixed-Width Integers Used in the Sample Program

Symbol	Description
int8_t	8-bit signed integer (defined in the standard library)
int16_t	16-bit signed integer (defined in the standard library)
int32_t	32-bit signed integer (defined in the standard library)
int64_t	64-bit signed integer (defined in the standard library)
uint8_t	8-bit unsigned integer (defined in the standard library)
uint16_t	16-bit unsigned integer (defined in the standard library)
uint32_t	32-bit unsigned integer (defined in the standard library)
uint64_t	64-bit unsigned integer (defined in the standard library)

6.4 Constants and Error Codes

Table 6.2 and Table 6.3 list the constants used in the sample program and the error codes, respectively.

Table 6.2 Constants Used in the Sample Program

Constant	Setting	Description
R_NOR_MEM_CHIP1	1U	Chip 1
R_NOR_MEM_CHIP2	2U	Chip 2
R_NOR_MEM_TOP1	0x60000000U	Address where the installed NOR flash 1 starts
R_NOR_MEM_END1	0x63FFFFFFU	Address where the installed NOR flash 1 ends
R_NOR_MEM_TOP2	0x64000000U	Address where the installed NOR flash 2 starts
R_NOR_MEM_END2	0x67FFFFFFU	Address where the installed NOR flash 2 ends
R_NOR_MEM_SIZE	0x04000000U	Size of the installed NOR flash memory
R_NOR_MEM_ADDR_LIMIT	0x68000000U	Boundary value of the address of the installed NOR flash memory
R_NOR_READ_SIZE	128U	Number of words that can be read in one round of reading
R_NOR_WRITE_SIZE	0x04000000U	Number of words that can be written in one round of writing
R_NOR_SECTOR_SIZE	64U * 1024U	Size of one sector (signed)
R_NOR_SECTOR_SIZE_INT	64 * 1024	Size of one sector (unsigned)
R_NOR_SECTOR_NO_SIZE	512U	Total number of sectors
R_NOR_SECT_ADDR_MSK	0x0001FFFFU	Mask value to set the address to 0x20000
R_NOR_SECT_NO_MSK	0x03FFFFFFU	Mask value to calculate the sector number

Table 6.3 Error Codes in the Sample Program

Constant	Setting	Description
R_NOR_SUCCESS	0U	The function call was executed successfully.
R_NOR_ERROR_PARAM	1U	The arguments of the function are not correct.
R_NOR_ERROR_FLASH_READ	2U	Failure in reading flash memory
R_NOR_ERROR_FLASH_WRITE	3U	Failure in writing to flash memory
R_NOR_ERROR_FLASH_ERASE	4U	Failure in erasing flash memory
R_NOR_ERROR_FLASH_INIT	5U	Failure in initializing flash memory

6.5 Global Variables

Table 6.4 lists the global variables.

Table 6.4 Global Variables

Type	Variable	Description	Function
char	sbuff[16]	Data for transmission to the terminal software	main
char	rbuff[16]	Data received from the terminal software	main

6.6 Functions

Table 6.5 lists the functions to be used.

Table 6.5 List of the Functions

Function	Summary	Page Number
nor_flash_init	Driver for initialization of the NOR flash memory	14
nor_flash_write_buf	Driver for writing to the NOR flash memory (multiple words)	14
nor_flash_write_unit	Driver for writing to the NOR flash memory (one word)	15
nor_flash_erase_sector	Driver for erasing a sector of the NOR flash memory	15
nor_flash_erase_chip	Driver for erasing a NOR flash memory chip	15
R_NOR_FLASH_Open	For starting the driver API function	16
R_NOR_FLASH_Erase_Chip	API for erasing a NOR flash memory chip	16
R_NOR_FLASH_Erase_Sector	API for erasing a sector of the NOR flash memory	16
R_NOR_FLASH_Write_Buf	API for writing to the NOR flash memory (multiple words)	17
R_NOR_FLASH_Close	For ending the driver API function	17
nor_flash_command	Function for issuing commands	17
main	Main function of the sample program	18

6.7 Specifications of Functions

The specifications of the functions of the sample program are listed below.

6.7.1 nor_flash_init

nor_flash_init		
Synopsis	Initialization of the NOR flash memory	
Header	nor_flash.h	
Declaration	int32_t nor_flash_write(uinit32_t address, uint32_t* data, uint32_t size)	
Description	This function initializes the NOR flash memory.	
Arguments	uint32_t chip_number	Specifies the chip number of flash memory to be initialized.
Return value	R_NOR_ SUCCESS:	Initialization has succeeded.
	R_NOR_ERROR_FLASH_INIT:	Initialization has failed.
	R_NOR_ ERR_PARAM:	Argument error
Supplement	None	

6.7.2 nor_flash_write_buf

nor_flash_write_buf		
Synopsis	Writing to the NOR flash memory (multiple words)	
Header	r_nor_flash.h	
Declaration	uint32_t nor_flash_write_buf(uint32_t address, uint16_t *data, uint32_t cnt);	
Description	This function is for writing multiple bytes of data to the NOR flash memory.	
Arguments	uinit32_t address	Specifies the address where writing of data is to start.
	uint16_t *data	Specifies the pointer to the data to be written.
	uint32_t cnt	Specifies the amount of data to be written.
Return value	R_NOR_ SUCCESS:	Writing has succeeded.
	R_NOR_ERROR_FLASH_WRITE:	Writing has failed.
	R_NOR_ ERR_PARAM:	Argument error
Supplement	This function can be called to write data without erasing a sector and without the need to recognize the chip and sector boundaries.	

6.7.3 nor_flash_write_unit

nor_flash_write_unit		
Synopsis	Writing to the NOR flash memory (one word)	
Header	r_nor_flash.h	
Declaration	uint32_t nor_flash_write_unit(uint32_t address, uint16_t data)	
Description	This function is for writing a word of data to the NOR flash memory.	
Arguments	uinit32_t address	Specifies the address where writing is to start.
	uint16_t data	Specifies data to be written.
Return value	R_NOR_ SUCCESS:	Writing has succeeded.
	R_NOR_ERROR_FLASH_WRITE:	Writing has failed.
	R_NOR_ ERR_PARAM:	Argument error
Supplement	Before using this command, erase the corresponding sector.	

6.7.4 nor_flash_erase_sector

nor_flash_erase_sector		
Synopsis	Erasing a sector of the NOR flash memory	
Header	r_nor_flash.h	
Declaration	uint32_t nor_flash_erase_sector(uint32_t chip_number, uint32_t sect_no)	
Description	This function is for erasing one sector of the NOR flash memory.	
Arguments	uint8_t chip_number:	Specifies the chip number to be erased.
	uint32_t sect_no:	Specifies the sector number to be erased.
Return value	R_NOR_ SUCCESS:	Erasure has succeeded.
	R_NOR_ERROR_FLASH_ERASE:	Erasure has failed.
	R_NOR_ ERR_PARAM:	Argument error
Supplement	None	

6.7.5 nor_flash_erase_chip

nor_flash_erase_chip		
Synopsis	Erasing a NOR flash memory chip	
Header	r_nor_flash.h	
Declaration	uint32_t nor_flash_erase_chip(uint32_t chip_number)	
Description	This function is for erasing all data in one NOR flash memory chip.	
Arguments	uint8_t chip_number:	Specifies the chip number of the NOR flash memory.
Return value	R_NOR_ SUCCESS:	Erasure has succeeded.
	R_NOR_ERROR_FLASH_ERASE:	Erasure has failed.
	R_NOR_ ERR_PARAM:	Argument error
Supplement	None	

6.7.6 R_NOR_FLASH_Open

R_NOR_FLASH_Open		
Synopsis	Starting the driver API function	
Header	r_nor_flash.h	
Declaration	int32_t R_NOR_FLASH_Open(void)	
Description	This function is for starting the driver API function.	
Arguments	None	
Return value	R_NOR_ SUCCESS:	Processing for the flash memory has started.
	R_NOR_ERROR_FLASH_INIT:	Initialization of the flash memory has failed.
Supplement	None	

6.7.7 R_NOR_FLASH_Erase_Chip

R_NOR_FLASH_Erase_Chip		
Synopsis	Erasing a NOR flash memory chip	
Header	r_nor_flash.h	
Declaration	uint32_t R_NOR_FLASH_Erase_Chip (uint32_t chip_number);	
Description	This function is for erasing the whole NOR flash memory chip.	
Arguments	uint32_t chip_number	
Return value	R_NOR_ SUCCESS:	Erasure of flash memory has succeeded.
	R_NOR_ERROR_FLASH_INIT:	Initialization of flash memory has failed.
	R_NOR_ERROR_FLASH_ERASE:	Erasure of flash memory has failed.
Supplement	Issuing this command takes several minutes since the whole chip is to be erased.	

6.7.8 R_NOR_FLASH_Erase_Sector

R_NOR_FLASH_Erase_Sector		
Synopsis	Erasing a sector of the NOR flash memory	
Header	r_nor_flash.h	
Declaration	uint32_t R_NOR_FLASH_Erase_Sector(uint32_t chip_number, uint32_t sectorAddr);	
Description	This function erases one sector of the installed NOR flash memory.	
Arguments	uint32_t chip_number:	Specifies the chip number.
	uint32_t sectorAddr:	Specifies the sector number.
Return value	R_NOR_ SUCCESS:	Erasure of flash memory has succeeded.
	R_NOR_ERROR_FLASH_INIT:	Initialization of flash memory has failed.
	R_NOR_ERROR_FLASH_ERASE:	Erasure of flash memory has failed.
Supplement	None	

6.7.9 R_NOR_FLASH_Write_Buf

R_NOR_FLASH_Write_Buf		
Synopsis	Writing to the NOR flash memory (multiple words)	
Header	r_nor_flash.h	
Declaration	int32_t R_NOR_FLASH_Write_Buf(uinit32_t address, uint16_t* data, uint32_t size)	
Description	This is an API function for writing multiple words of data to the NOR flash memory. This function can be called to write data without the need to recognize the chip and sector boundaries.	
Arguments	uinit32_t address:	Specifies the address where writing of data is to start.
	uint16 *data	Specifies the pointer to the data to be written.
	uint32_t size:	Specifies the number of bytes of data to be written.
Return value	R_NOR_SUCCESS:	Writing has succeeded.
	R_NOR_ERR_PARAM:	Argument error
Supplement	None	

6.7.10 R_NOR_FLASH_Close

R_NOR_FLASH_Close	
Synopsis	Ending the driver API function
Header	r_nor_flash.h
Declaration	void R_NOR_FLASH_Close (void)
Description	This function is for ending the driver API function.
Arguments	None
Return value	None
Supplement	None

6.7.11 nor_flash_command

nor_flash_command	
Synopsis	Function for issuing commands
Header	r_nor_flash.h
Declaration	void nor_flash_command (void)
Description	This function handles the commands for reading from and writing to the NOR flash memory in the form of a menu and responses in the terminal software.
Arguments	None
Return value	None
Supplement	None

6.7.12 main

main	
Synopsis	Main function of the sample program
Header	—
Declaration	void main(void)
Description	This is for main processing of the sample program. For the details of processing, refer to Section 7.1, Main Processing.
Arguments	None
Return value	None
Supplement	None

7. Flowcharts

7.1 Main Processing

Figure 7.1 shows a flowchart of main processing of the sample program.

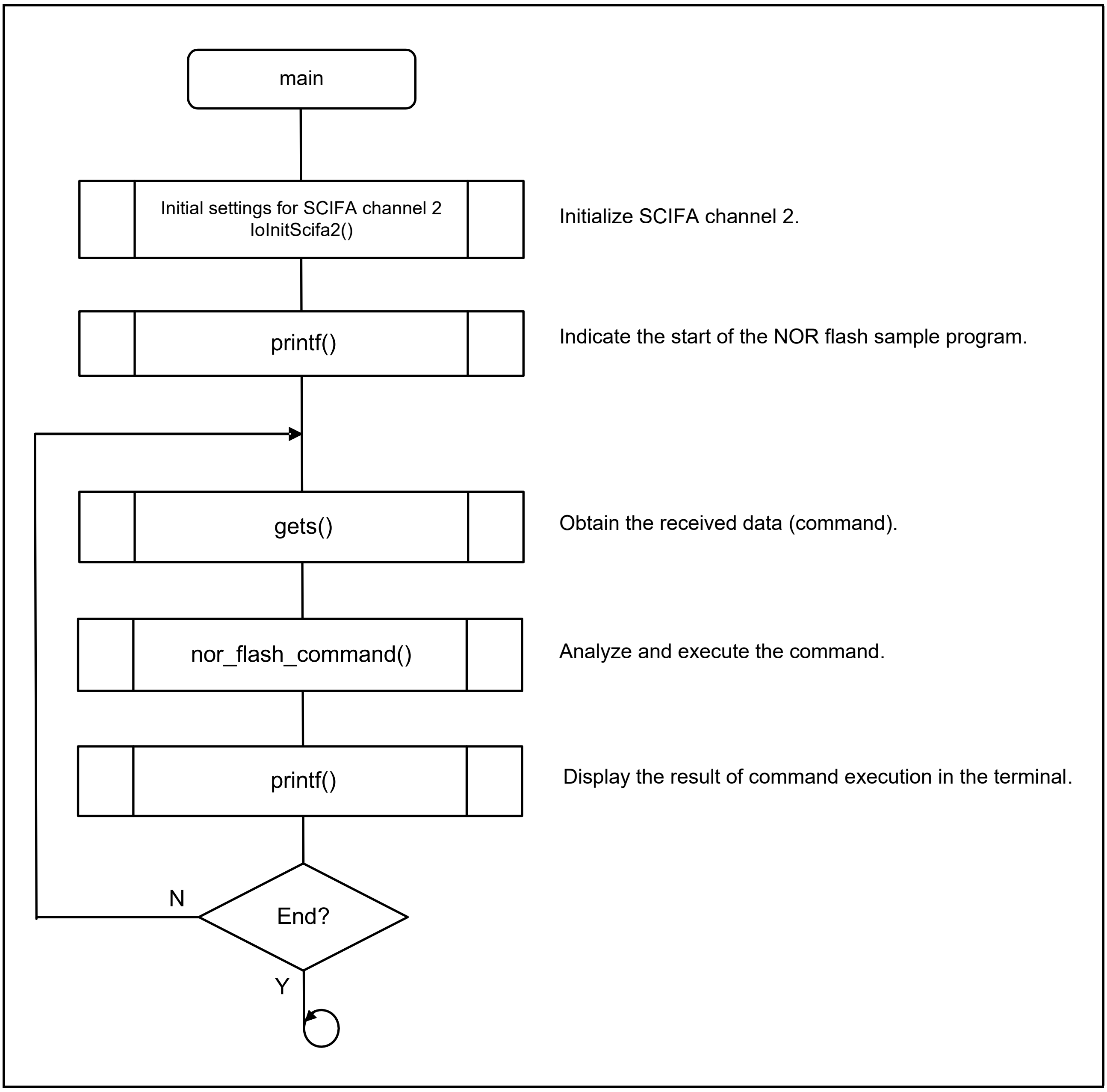

main
Initial settings for SCIFA channel 2
IoInitScifa2()
Initialize SCIFA channel 2.
printf()
Indicate the start of the NOR flash sample program.
gets()
Obtain the received data (command).
nor_flash_command()
Analyze and execute the command.
printf()
Display the result of command execution in the terminal.
End?
N
Y

Figure 7.1 Main Processing

8. Example Operation of the Sample Program

The following shows how the sample program operates in the terminal software on the PC.

The following menu is displayed when the power of the evaluation board is switched on or following a reset.

```
NOR Flash example program

[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash Chip
[4] Erase NOR Flash sector
[0] Exit

>
```

If you select [1], you will be asked to input the address and size. The maximum size is 128 words.

```
NOR Flash example program

[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash Chip
[4] Erase NOR Flash sector
[0] Exit

>1[Enter]

Input the top address and data size

>
```

If you input the address and size as 6000F000 and 16 respectively, 16 words of data will be displayed as shown below.

```
Input the top address and data size

>6000F000 16[Enter]

          00   02   04   06   08   0A   0C   0E
6000F000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
6000F010 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF

[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase all NOR Flash
[0] Exit

>
```

After the data that have been read are displayed, the menu screen will have been displayed again. Select [2] this time.

```
[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash Chip
[4] Erase NOR Flash sector
[0] Exit

>2[Enter]

Input the top address by hex (60000000 - 67FFFFFE)

>
```

Input the address where writing of data is to start.

```
Input the top address by hex (60000000 - 67FFFFFE)

>60000000[Enter]

Input the size by hex (0 - 04000000 words)
```

Input the size of data to be written.

```
Input the size by hex (0 - 04000000 words)

>20[Enter]

Input the pointer address by hex (0 - FFFFFFFE)
```

Input the pointer address of the data to be written.

```
Input the pointer address by hex (0 - FFFFFFFE)

>0[Enter]

Wrote NOR Flash
```

If the data have been written successfully, the message “Wrote NOR Flash” will be displayed.

Next, select [3] in the menu. You will be asked whether you really want to erase the NOR flash memory, so select [y].

```
[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash Chip
[4] Erase NOR Flash sector
[0] Exit

>3[Enter]

Do you really want to erase NOR Flash? <y/n>y[Enter]
```

Next, you will be asked the chip number to erase, so select 1.

```
[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash chip
[4] Erase NOR Flash sector
[0] Exit

>3

Do you really want to erase NOR Flash? <y/n>y

Input the chip number (1 or 2)

>1
```

Erasing one chip will take several minutes. If the chip was erased successfully, “Erased NOR Flash” will be displayed.

```
[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash chip
[4] Erase NOR Flash sector
[0] Exit

>3

Do you really want to erase NOR Flash? <y/n>y

Input the chip number (1 or 2)

>1

Please wait few minutes to erase

 Erased NOR Flash
```

Next, select [4] in the menu. You will be asked whether you really want to erase the NOR flash memory, so select [y].

```
[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash Chip
[4] Erase NOR Flash sector
[0] Exit

>4[Enter]

Do you really want to erase NOR Flash? <y/n>y[Enter]
```

Next, you will be asked the chip number to erase, so select 1.

```
Do you really want to erase NOR Flash? <y/n>y

Input the chip number (1 or 2)

>1
```

Then, you will be asked the sector number to erase, so select 0.

```
Input the chip number (1 or 2)

>1

Input the sector number by dec (0 - 511)

 >0
```

If the chip was erased successfully, “Erased NOR Flash sector” will be displayed.

```
Input the chip number (1 or 2)

>1

Erased NOR Flash sector
```

9. Obtaining the Sample Program

The sample program is available on the Renesas Electronics website.

10. Documents for Reference

- User's Manual: Hardware
  RZ/T1 Group User's Manual: Hardware
  (Download the latest version from the Renesas Electronics website.)

  RZ/T1 Evaluation Board RTK7910022C00000BR User's Manual
  (Download the latest version from the Renesas Electronics website.)

- Technical Update and Technical News
  (Download the latest version from the Renesas Electronics website.)

- User's Manual: Development Environment
  For the IAR integrated development environment (IAR Embedded Workbench® for Arm), visit the IAR Systems website.
  (Download the latest version from the IAR Systems website.)

  For the Arm software development tools (Arm Compiler toolchain, Arm DS-5, etc.), visit the Arm ltd. website.
  (Download the latest version from the Arm ltd. website.)

  For the Renesas Electronics software development tools (e2studio, etc.), visit the Renesas Electronics website.
  (Download the latest version from the Renesas Electronics website.)

Website and Support

Renesas Electronics website

http://www.renesas.com/

Inquiries

http://www.renesas.com/inquiry

Revision History	Application Note: NOR Flash Sample Program

Rev.	Date	Description	
		Page	Summary
1.00	Dec. 22, 2015	—	First Edition issued
1.10	Jun. 14, 2016	Summary	
		1	Restrictions modified
		2. Operating Environment	
		4	Table 2.1 Operating Environment Integrated development environment, format modified, entries added
		6.1.1 Project Settings	
		8	Information that is changed from the initial setting, added
		6.2.1 Assignment to Sections of Sample Program	
		10	Section placement / size change information that are changed from the initial setting, added
		6.4 Constants and Error Codes	
		12	Table 6.3 Constants Used in the Sample Program, values modified, entries added
		10. Documents for Reference	
		25	Where to find information about the operating environment added
		All	
		—	Errors corrected
1.20	Apr. 05, 2017	2. Operating Environment	
		4	Table 2.1 Operating Environment: Integrated Development Environment, modified
		6. Software	
		—	6.2.4 Required Memory Size, deleted
1.30	Jun. 07, 2018	2. Operating Environment	
		4	Table 2.1 Operating Environment: The description on the integrated development environment, modified
		5. Hardware	
		7	Figure 5.1 Hardware configuration example: The name of module, modified
		10. Documents for Reference	
		25	"ARM" changed to "Arm"

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 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, Fax: +44-1628-651-804

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 7.0