To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

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

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of 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. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may 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.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or system 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be 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, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

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

R8C/2D Group

Program ROM Rewrite Using EW0 Mode

1. Abstract

This document describes the rewrite program setting process and application example of the program ROM using EW0 mode.

2. Introduction

The application example described in this document applies to the following MCU:

- MCU : R8C/2D Group

This program can be used with other R8C/Tiny Series MCUs which have the same special function registers (SFRs) as the R8C/2D Group. Check the manual for any additions and modifications to functions. Careful evaluation is recommended before using this application note.

3. Application Example

3.1 EW0 Mode Features

In EW0 mode, the user ROM area can be rewritten by transferring the CPU rewrite program to RAM, and executing the program command and erase command using the CPU rewrite program on RAM. Since the CPU operates during programming and erasing in EW0 mode, the peripheral function interrupt can be accepted during programming and erasing by allocating the vector and interrupt program on RAM.

The CPU rewrite program is transferred to RAM in the main process and a program which cannot accept an interrupt during programming and erasing is described in this application note.

3.2 Program Outline

Figure 3.1 shows the Program Outline Flowchart.

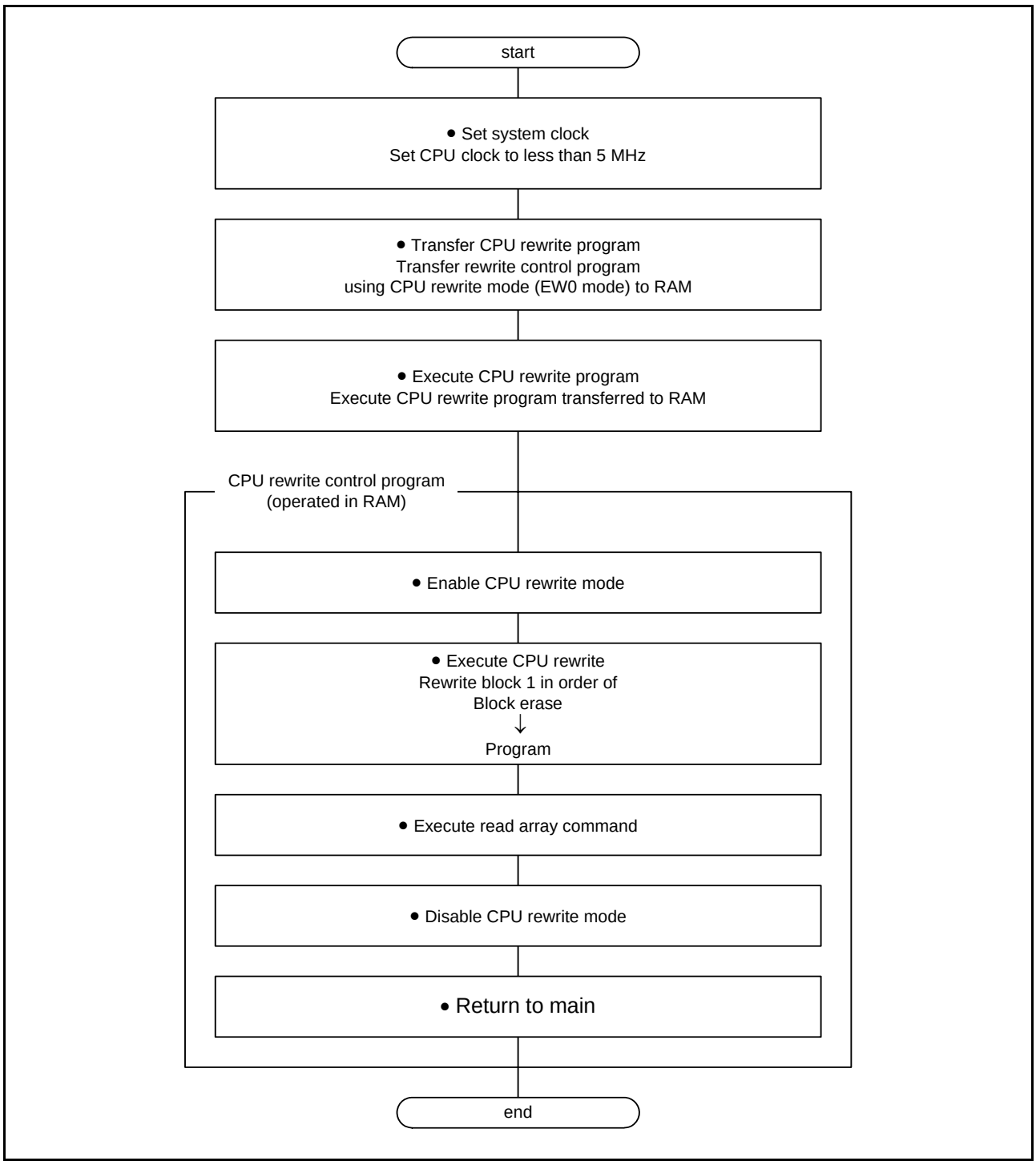


Figure 3.1 Program Outline Flowchart

3.3 Memory

Table 3.1 Memory

Memory	Size	Remarks
ROM	355 bytes	In the rej05b1103_src.c module
RAM	3 bytes	In the rej05b1103_src.c module However, the number of bytes of the variable “download_data[DATA_SIZE]” in which the write data is stored are excluded. 1,024 bytes are excluded here. The RAM size of program size transferred to RAM besides the above RAM size is necessary. 246 bytes are necessary for RAM size here.
Maximum user stack	18 bytes	main function: 3 bytes mcu_init function: 6 bytes set_data function: 7 bytes ew0_rewrite_control: 8 bytes full_sts_chk function: 7 bytes
Maximum interrupt stack	0 byte	Not used

The memory size varies depending on the C compiler version and compile options. The above applies to the following conditions:

C compiler: M16C/60, 30, 20, 10, and R8C/Tiny Series Compiler V.5.43 Release 00
Compile option: -c -finfo (see Note) -dir “$(CONFIGDIR)” -R8C
Note: -c -finfo cannot be used for the R8C/Tiny-only Free-version.

4. Setup

This section shows the initial setting procedures and values to set the example described in 3. Application Example. Refer to the R8C/2D Group Hardware Manual for details on individual registers.

4.1 Set System Clock

(1) Enable writing to registers CM0, CM1, OCD, FRA0, FRA1, and FRA2.

Protect Register

b7	b6	b5	b4	b3	b2	b1	b0
							1

Symbol	Address	After Reset
PRCR	000Ah	00h

Bit Symbol	Bit Name	Function	RW
PRC0	Protect bit 0	Writing to registers CM0, CM1, OCD, FRA0, FRA1, and FRA2 is enabled. 1 : Enables writing	RW

(2) Start the low-speed on-chip oscillator.

System Clock Control Register 1(1)

b7	b6	b5	b4	b3	b2	b1	b0
			0				

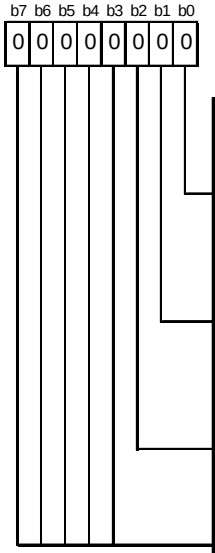
Symbol	Address	After Reset
CM1	0007h	00100000b

Bit Symbol	Bit Name	Function	RW
CM14	Low-speed on-chip oscillation stop bit(2, 3, 4)	0 : Low-speed on-chip oscillator on	RW

NOTES:
1. Set the PRC0 bit in the PRCR register to 1 (write enable) before rewriting the CM1 register.
2. When the OCD2 bit is set to 0 (XIN clock selected), the CM14 bit is set to 1 (low-speed on-chip oscillator stopped). When the OCD2 bit is set to 1 (on-chip oscillator clock selected), the CM14 bit is set to 0 (low-speed on-chip oscillator on). It remains unchanged even if 1 is written to it.
3. When using the voltage monitor 1 interrupt or voltage monitor 2 interrupt (when using the digital filter), set the CM14 bit to 0 (low-speed on-chip oscillator on).
4. In count source protect mode, the value remains unchanged even if bits CM10 and CM14 are set.

(3) Set the dividing ratio of the high-speed on-chip oscillator.

High-Speed On-Chip Oscillator Control Register 2(1)



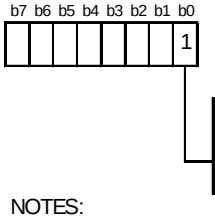
Symbol	Address	After Reset
FRA2	0025h	00h

Bit Symbol	Bit Name	Function	RW
FRA20	High-speed on-chip oscillator frequency switching bits(2)	Selects the dividing ratio for the high-speed on-chip oscillator clock. b2 b1 b0 0 0 0: Divide-by-2 mode	RW
FRA21			RW
FRA22			RW
— (b7-b3)	Reserved bits	Set to 0.	RW

NOTES:
1. Set the PRC0 bit in the PRCR register to 1 (write enable) before rewriting the FRA2 register.
2. Set these bits to 010b to 111b since the value after reset is 000b.

(4) Start the high-speed on-chip oscillator.

High-Speed On-Chip Oscillator Control Register 0(1)



Symbol	Address	After Reset
FRA0	0023h	00h

Bit Symbol	Bit Name	Function	RW
FRA00	High-speed on-chip oscillator enable bit	1 : High-speed on-chip oscillator on	RW

NOTES:
1. Set the PRC0 bit in the PRCR register to 1 (write enable) before rewriting the FRA0 register.

(5)Wait until oscillation stabilizes.

(6) Select the high-speed on-chip oscillator.

High-Speed On-Chip Oscillator Control Register 0(1)

b7	b6	b5	b4	b3	b2	b1	b0
						1	

Symbol	Address	After Reset
FRA0	0023h	00h

Bit Symbol	Bit Name	Function	RW
FRA01	High-speed on-chip oscillator select bit(2)	1 : Selects high-speed on-chip oscillator	RW

NOTES:
1. Set the PRC0 bit in the PRCR register to 1 (write enable) before rewriting the FRA0 register.
2. Change the FRA01 bit under the following conditions.
   - FRA00 = 1 (high-speed on-chip oscillator on)
   - The CM14 bit in the CM1 register = 0 (low-speed on-chip oscillator on)
   - Bits FRA22 to FRA20 in the FRA2 register:
     All divide ratio mode settings are supported when VCC = 3.0 V to 5.5 V 000b to 111b
     Divide ratio of 4 or more when VCC = 2.7 V to 5.5 V 010b to 111b (divide by 4 or more)
     Divide ratio of 8 or more when VCC = 2.2 V to 5.5 V 110b to 111b (divide by 8 or more)

(7) Set the system clock dividing ratio to divide-by-8 mode.

System Clock Control Register 0(1)

b7	b6	b5	b4	b3	b2	b1	b0
	1						

Symbol	Address	After Reset
CM0	0006h	01101000b

Bit Symbol	Bit Name	Function	RW
CM06	System clock division select bit 0(2)	1 : Divide-by-8 mode	RW

NOTES:
1. Set the PRC0 bit in the PRCR register to 1 (write enable) before rewriting the CM0 register.
2. When entering stop mode, the CM06 bit is set to 1 (divide-by-8 mode).

(8) Disable writing to registers CM0, CM1, OCD, FRA0, FRA1, and FRA2.

Protect Register

b7	b6	b5	b4	b3	b2	b1	b0
							0

Symbol	Address	After Reset
PRCR	000Ah	00h

Bit Symbol	Bit Name	Function	RW
PRC0	Protect bit 0	Writing to registers CM0, CM1, OCD, FRA0, FRA1, and FRA2 is enabled. 0 : Disables writing	RW

4.2 Transferring CPU Rewrite Control Program to RAM

The CPU rewrite control program needs to be operated on RAM. This application note describes an example of transferring the CPU rewrite control program to RAM using the smovf instruction in the main process. Figure 4.1 shows the Program Assignment.

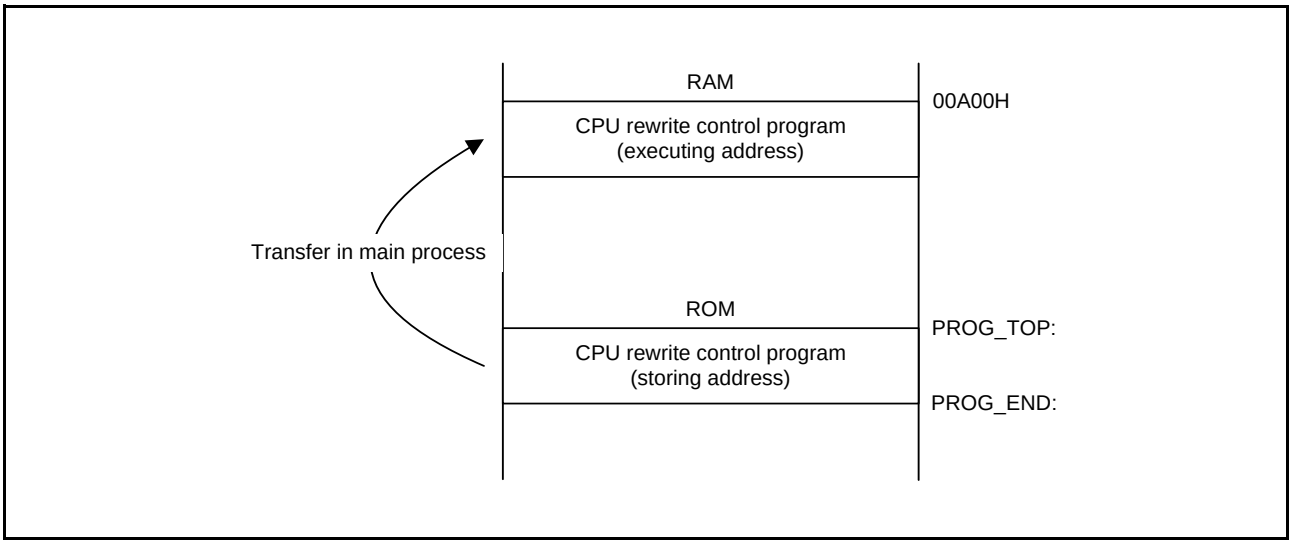


Figure 4.1 Program Assignment

(1) Set the ten-thousands digit of the CPU rewrite control program start address to r1h.

(2) Set the thousands digit of the CPU rewrite control program start address to a0.

(3) Set the RAM start address to which the CPU rewrite control program is transferred to a1. In this application note, set 00A00h as the start address.

(4) Set the CPU rewrite control program size to r3.

(5) Transfer the CPU rewrite control program to RAM area using the smovf instruction.

4.3 Processing During CPU Rewrite Control Program

4.3.1 CPU Rewrite Mode Enable Setting

To enable the CPU rewrite mode, follow the steps below.

Step 1: Set the FMR01 bit to 0. Do not generate an interrupt between setting the bit to 0 and setting it to 1.

Flash Memory Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	1

Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0 : Busy (writing or erasing in progress) 1 : Ready	RO
FMR01	CPU rewrite mode select bit(1)	0 : CPU rewrite mode disabled	RW
FMR02	Blocks 0 to 3 rewrite enable bit(2, 6)	0 : Disables rewrite	RW
FMSTP	Flash memory stop bit(3, 5)	0 : Enables flash memory operation	RW
— (b5-b4)	Reserved bits	Set to 0.	RW
FMR06	Program status flag(4)	0 : Completed successfully 1 : Terminated by error	RO
FMR07	Erase status flag(4)	0 : Completed successfully 1 : Terminated by error	RO

NOTES:

1. To set this bit to 1, set it to 1 immediately after setting it first to 0. Do not generate an interrupt between setting the bit to 0 and setting it to 1. Enter read array mode and set this bit to 0.
2. Set this bit to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1.
   Do not generate an interrupt between setting the bit to 0 and setting it to 1.
3. Set this bit by a program transferred to the RAM.
4. This bit is set to 0 by executing the clear status command.
5. This bit is enabled when the FMR01 bit is set to 1 (CPU rewrite mode enabled). When the FMR01 bit is set to 0, writing 1 to the FMSTP bit causes the FMSTP bit to be set to 1. The flash memory does not enter low-power consumption state nor is it reset.
6. When setting the FMR01 bit to 0 (CPU rewrite mode disabled), the FMR02 bit is set to 0 (disables rewrite).

Step 2: Set the FMR01 bit to 1.

Flash Memory Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	1	1

Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0 : Busy (writing or erasing in progress) 1 : Ready	RO
FMR01	CPU rewrite mode select bit(1)	1 : CPU rewrite mode enabled	RW
FMR02	Blocks 0 to 3 rewrite enable bit(2, 6)	0 : Disables rewrite	RW
FMSTP	Flash memory stop bit(3, 5)	0 : Enables flash memory operation	RW
— (b5-b4)	Reserved bits	Set to 0.	RW
FMR06	Program status flag(4)	0 : Completed successfully 1 : Terminated by error	RO
FMR07	Erase status flag(4)	0 : Completed successfully 1 : Terminated by error	RO

NOTES:

1. To set this bit to 1, set it to 1 immediately after setting it first to 0. Do not generate an interrupt between setting the bit to 0 and setting it to 1. Enter read array mode and set this bit to 0.
2. Set this bit to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1.
   Do not generate an interrupt between setting the bit to 0 and setting it to 1.
3. Set this bit by a program transferred to the RAM.
4. This bit is set to 0 by executing the clear status command.
5. This bit is enabled when the FMR01 bit is set to 1 (CPU rewrite mode enabled). When the FMR01 bit is set to 0, writing 1 to the FMSTP bit causes the FMSTP bit to be set to 1. The flash memory does not enter low-power consumption state nor is it reset.
6. When setting the FMR01 bit to 0 (CPU rewrite mode disabled), the FMR02 bit is set to 0 (disables rewrite).

Step 3: To enable rewriting to blocks 0 to 3, while the FMR01 bit is 1, set the FMR02 bit to 0. Do not generate an interrupt between Step 3 and Step 4.

Flash Memory Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	1	1

Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0 : Busy (writing or erasing in progress) 1 : Ready	RO
FMR01	CPU rewrite mode select bit(1)	1 : CPU rewrite mode enabled	RW
FMR02	Blocks 0 to 3 rewrite enable bit(2, 6)	0 : Disables rewrite	RW
FMSTP	Flash memory stop bit(3, 5)	0 : Enables flash memory operation	RW
— (b5-b4)	Reserved bits	Set to 0.	RW
FMR06	Program status flag(4)	0 : Completed successfully 1 : Terminated by error	RO
FMR07	Erase status flag(4)	0 : Completed successfully 1 : Terminated by error	RO

NOTES:

1. To set this bit to 1, set it to 1 immediately after setting it first to 0. Do not generate an interrupt between setting the bit to 0 and setting it to 1. Enter read array mode and set this bit to 0.
2. Set this bit to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1.
   Do not generate an interrupt between setting the bit to 0 and setting it to 1.
3. Set this bit by a program transferred to the RAM.
4. This bit is set to 0 by executing the clear status command.
5. This bit is enabled when the FMR01 bit is set to 1 (CPU rewrite mode enabled). When the FMR01 bit is set to 0, writing 1 to the FMSTP bit causes the FMSTP bit to be set to 1. The flash memory does not enter low-power consumption state nor is it reset.
6. When setting the FMR01 bit to 0 (CPU rewrite mode disabled), the FMR02 bit is set to 0 (disables rewrite).

Step 4: Set the FMR02 bit to 1.

Flash Memory Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	1	1	1

Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0 : Busy (writing or erasing in progress) 1 : Ready	RO
FMR01	CPU rewrite mode select bit(1)	1 : CPU rewrite mode enabled	RW
FMR02	Blocks 0 to 3 rewrite enable bit(2, 6)	1 : Enables rewrite	RW
FMSTP	Flash memory stop bit(3, 5)	0 : Enables flash memory operation	RW
— (b5-b4)	Reserved bits	Set to 0.	RW
FMR06	Program status flag(4)	0 : Completed successfully 1 : Terminated by error	RO
FMR07	Erase status flag(4)	0 : Completed successfully 1 : Terminated by error	RO

NOTES:

1. To set this bit to 1, set it to 1 immediately after setting it first to 0. Do not generate an interrupt between setting the bit to 0 and setting it to 1. Enter read array mode and set this bit to 0.
2. Set this bit to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1.
   Do not generate an interrupt between setting the bit to 0 and setting it to 1.
3. Set this bit by a program transferred to the RAM.
4. This bit is set to 0 by executing the clear status command.
5. This bit is enabled when the FMR01 bit is set to 1 (CPU rewrite mode enabled). When the FMR01 bit is set to 0, writing 1 to the FMSTP bit causes the FMSTP bit to be set to 1. The flash memory does not enter low-power consumption state nor is it reset.
6. When setting the FMR01 bit to 0 (CPU rewrite mode disabled), the FMR02 bit is set to 0 (disables rewrite).

Step 5: To enable rewriting to block 1, first set the FMR16 bit to 1.

Flash Memory Control Register 1

b7	b6	b5	b4	b3	b2	b1	b0
1	1	1	0	0	0	0	0

Symbol	Address	After Reset
FMR1	01B5h	1000000Xb

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved bit	When read, the content is undefined.	RO
FMR11	EW1 mode select bit(1, 2)	0 : EW0 mode	RW
— (b4-b2)	Reserved bits	Set to 0.	RW
FMR15	Block 0 rewrite disable bit(2,3)	1 : Disables rewrite	RW
FMR16	Block 1 rewrite disable bit(2,3)	1 : Disables rewrite	RW
— (b7)	Reserved bit	Set to 1.	RW

NOTES:

1. To set this bit to 1, set it to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1 (CPU rewrite mode enabled). Do not generate an interrupt between setting the bit to 0 and setting it to 1.
2. This bit is set to 0 by setting the FMR01 bit to 0 (CPU rewrite mode disabled).
3. When the FMR01 bit is set to 1 (CPU rewrite mode enabled), bits FMR15 and FMR16 can be written to.
   To set this bit to 0, set it to 0 immediately after setting it first to 1.
   To set this bit to 1, set it to 1.

Step 6: Set the FMR16 bit to 0.

Flash Memory Control Register 1

b7	b6	b5	b4	b3	b2	b1	b0
1	0	1	0	0	0	0	0

Symbol	Address	After Reset
FMR1	01B5h	1000000Xb

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved bit	When read, the content is undefined.	RO
FMR11	EW1 mode select bit(1, 2)	0 : EW0 mode	RW
— (b4-b2)	Reserved bits	Set to 0.	RW
FMR15	Block 0 rewrite disable bit(2,3)	1 : Disables rewrite	RW
FMR16	Block 1 rewrite disable bit(2,3)	1 : Enables rewrite	RW
— (b7)	Reserved bit	Set to 1.	RW

NOTES:

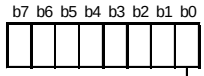
1. To set this bit to 1, set it to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1 (CPU rewrite mode enabled). Do not generate an interrupt between setting the bit to 0 and setting it to 1.
2. This bit is set to 0 by setting the FMR01 bit to 0 (CPU rewrite mode disabled).
3. When the FMR01 bit is set to 1 (CPU rewrite mode enabled), bits FMR15 and FMR16 can be written to.
   To set this bit to 0, set it to 0 immediately after setting it first to 1.
   To set this bit to 1, set it to 1.

4.3.2 Block Erase Processing

(1) The auto-erase operation (erase and erase verify) to the specified block starts by writing 20h at the first bus cycle, and D0h at the second bus cycle to a given address of a block. Block 1 is specified in this section.

(2) Wait until the auto-erase operation is completed. The completion of the auto-erase operation can be confirmed by the FMR00 bit in the FMR0 register.

Flash Memory Control Register 0

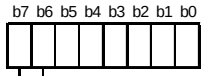


Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0 : Busy (writing or erasing in progress) 1 : Ready	RO

(3) To check the full status, check bits FMR06 and FMR07 in the FMR0 register. When an error occurs, the CPU rewrite process stops after writing 50h to the address to which the erase command was written.

Flash Memory Control Register 0



Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
FMR06	Program status flag(1)	0 : Completed successfully 1 : Terminated by error	RO
FMR07	Erase status flag(1)	0 : Completed successfully 1 : Terminated by error	RO

NOTES:
1. This bit is set to 0 by executing the clear status command.

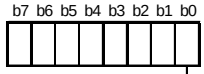
4.3.3 Program Processing

Write data to all applicable program areas in 1-byte units.

(1) The auto-programming operation (data program and verify) starts by writing 40h to the write address at the first bus cycle, and writing data at the second bus cycle. Set the same address for the address value at the first bus cycle as the specified write address at the second bus cycle.

(2) Wait until the auto-programming operation is completed. The completion of the auto-programming operation can be confirmed by the FMR00 bit in the FMR0 register.

Flash Memory Control Register 0

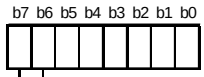


Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0 : Busy (writing or erasing in progress) 1 : Ready	RO

(3) To check the full status, check bits FMR06 and FMR07 in the FMR0 register. When an error occurs, the CPU rewrite process stops after writing 50h to the address to which the erase command was written.

Flash Memory Control Register 0



Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
FMR06	Program status flag(1)	0 : Completed successfully 1 : Terminated by error	RO
FMR07	Erase status flag(1)	0 : Completed successfully 1 : Terminated by error	RO

NOTES:
1. This bit is set to 0 by executing the clear status command.

4.3.4 CPU Rewrite Mode Disable Setting

(1) Issue the read array command. In this section, write FFh to a given address of block 1.

(2) To disable the CPU rewrite mode, set the FMR01 bit to 0.

Flash Memory Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	1

Symbol	Address	After Reset
FMR0	01B7h	00000001b

Bit Symbol	Bit Name	Function	RW
FMR00	RY/$\overline{BY}$ status flag	0 : Busy (writing or erasing in progress) 1 : Ready	RO
FMR01	CPU rewrite mode select bit(1)	0 : CPU rewrite mode disabled	RW
FMR02	Blocks 0 to 3 rewrite enable bit(2, 6)	0 : Disables rewrite	RW
FMSTP	Flash memory stop bit(3, 5)	0 : Enables flash memory operation	RW
— (b5-b4)	Reserved bits	Set to 0.	RW
FMR06	Program status flag(4)	0 : Completed successfully 1 : Terminated by error	RO
FMR07	Erase status flag(4)	0 : Completed successfully 1 : Terminated by error	RO

NOTES:

1. To set this bit to 1, set it to 1 immediately after setting it first to 0. Do not generate an interrupt between setting the bit to 0 and setting it to 1. Enter read array mode and set this bit to 0.
2. Set this bit to 1 immediately after setting it first to 0 while the FMR01 bit is set to 1.
   Do not generate an interrupt between setting the bit to 0 and setting it to 1.
3. Set this bit by a program transferred to the RAM.
4. This bit is set to 0 by executing the clear status command.
5. This bit is enabled when the FMR01 bit is set to 1 (CPU rewrite mode enabled). When the FMR01 bit is set to 0, writing 1 to the FMSTP bit causes the FMSTP bit to be set to 1. The flash memory does not enter low-power consumption state nor is it reset.
6. When setting the FMR01 bit to 0 (CPU rewrite mode disabled), the FMR02 bit is set to 0 (disables rewrite).

5. Function Table and Flowchart

5.1 Function Table

Declaration	void mcu_init(void)		
Outline	Process system clock setting process		
Argument	Argument name		Meaning
	None		–
Variable (global)	Variable name		Contents
	None		
Returned value	Type	Value	Meaning
	None	–	–
Function	Set system clock (high-speed on-chip oscillator)		

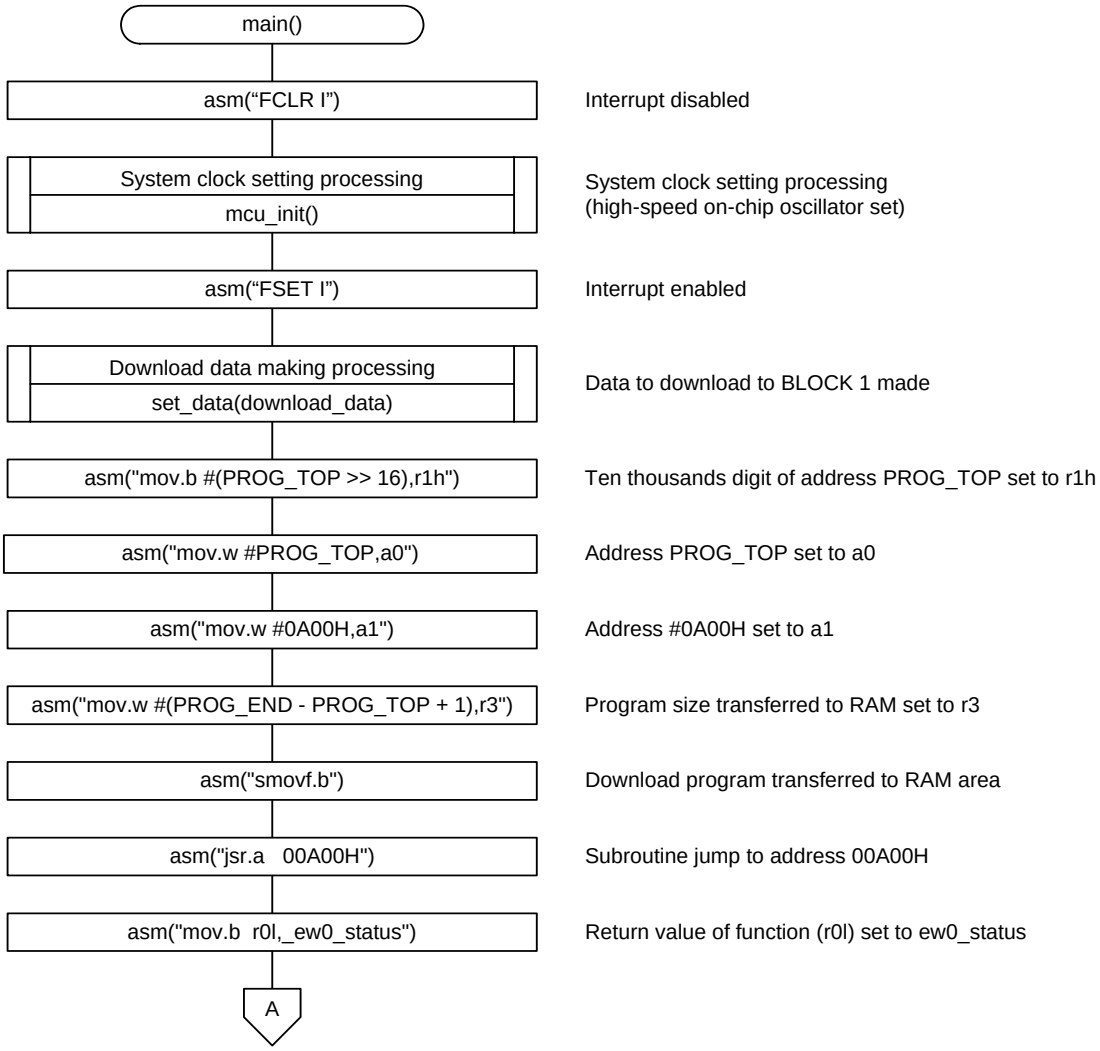
Declaration	void set_data(unsigned char *data)		
Outline	Make writing data		
Argument	Argument name		Meaning
	None		Table start address of write data
Variable (global)	Variable name		Contents
	None		
Returned value	Type	Value	Meaning
	None	–	–
Function	Make write record data for data flash. There is no process in this application note. Add the process if necessary.		

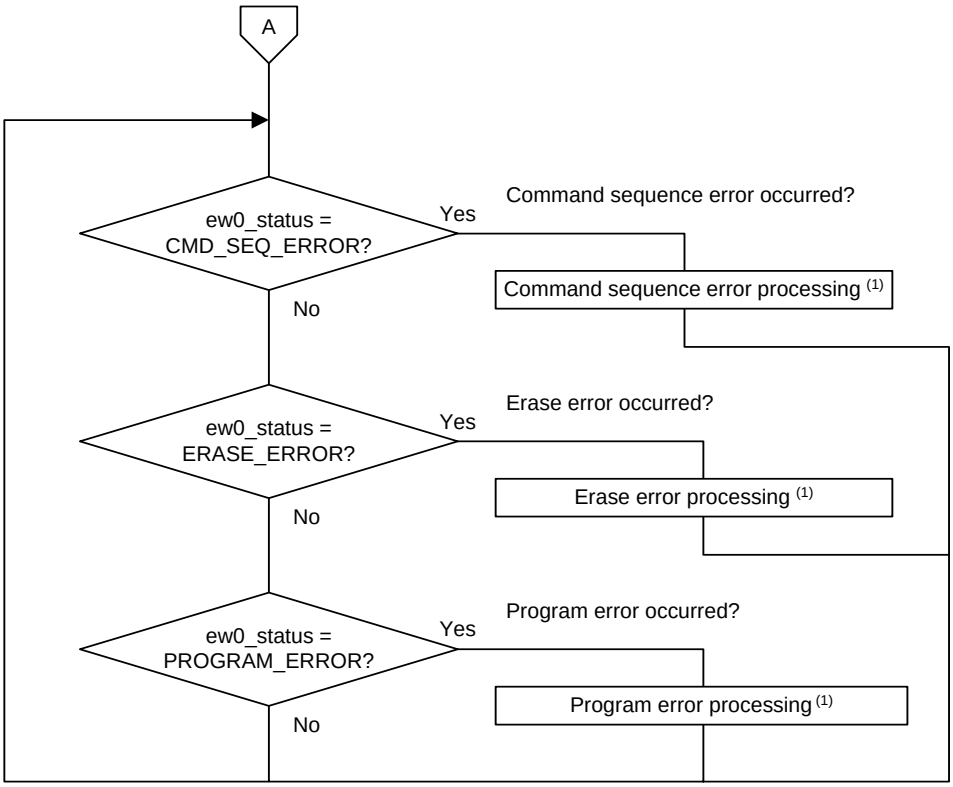
Declaration	unsigned char ew0_rewrite_control(void)		
Outline	CPU rewrite control processing		
Argument	Argument name		Meaning
	None		–
Variable (global)	Variable name		Contents
	unsigned char*wp		Address to which erase command and program command are written
	unsigned char download_data[DATA_SIZE]		Data array which stores writing data in block
Returned value	Type	Value	Meaning
	unsigned char	NORMAL	Completed successfully
		CMD_SEQ_ERROR	Command sequence error
		ERASE_ERROR	Erase error
		PROGRAM_ERROR	Program error
Function	Erase block 1 in EW0 mode and write data of download_data[DATA_SIZE].		

Declaration	unsigned char full_sts_chk (unsigned char *chk_adr)		
Outline	Full status check processing		
Argument	Argument name		Meaning
	unsigned char *chk_adr		Address to which erase command and program command are written
Variable (global)	Variable name		Contents
	None		–
Returned value	Type	Value	Meaning
	unsigned char	NORMAL	Completed successfully
		CMD_SEQ_ERROR	Command sequence error
		ERASE_ERROR	Erase error
		PROGRAM_ERROR	Program error
Function	Check full status		

5.2 Flow Chart

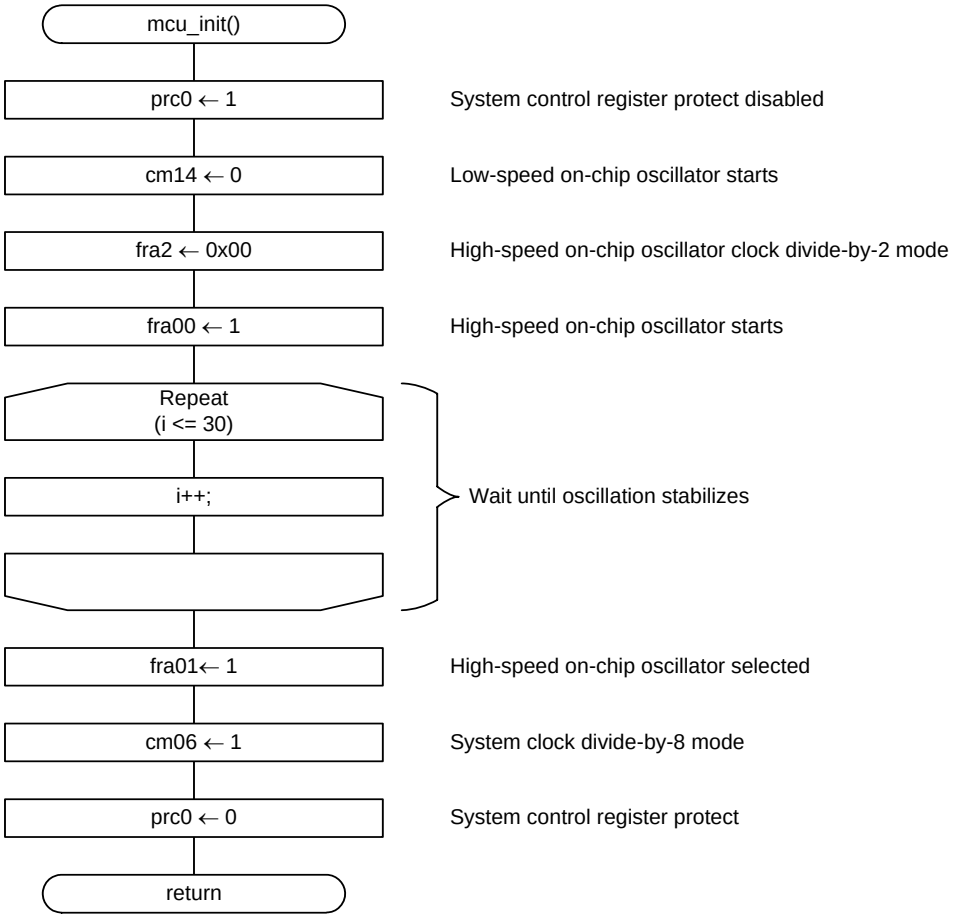
5.2.1 Main Function



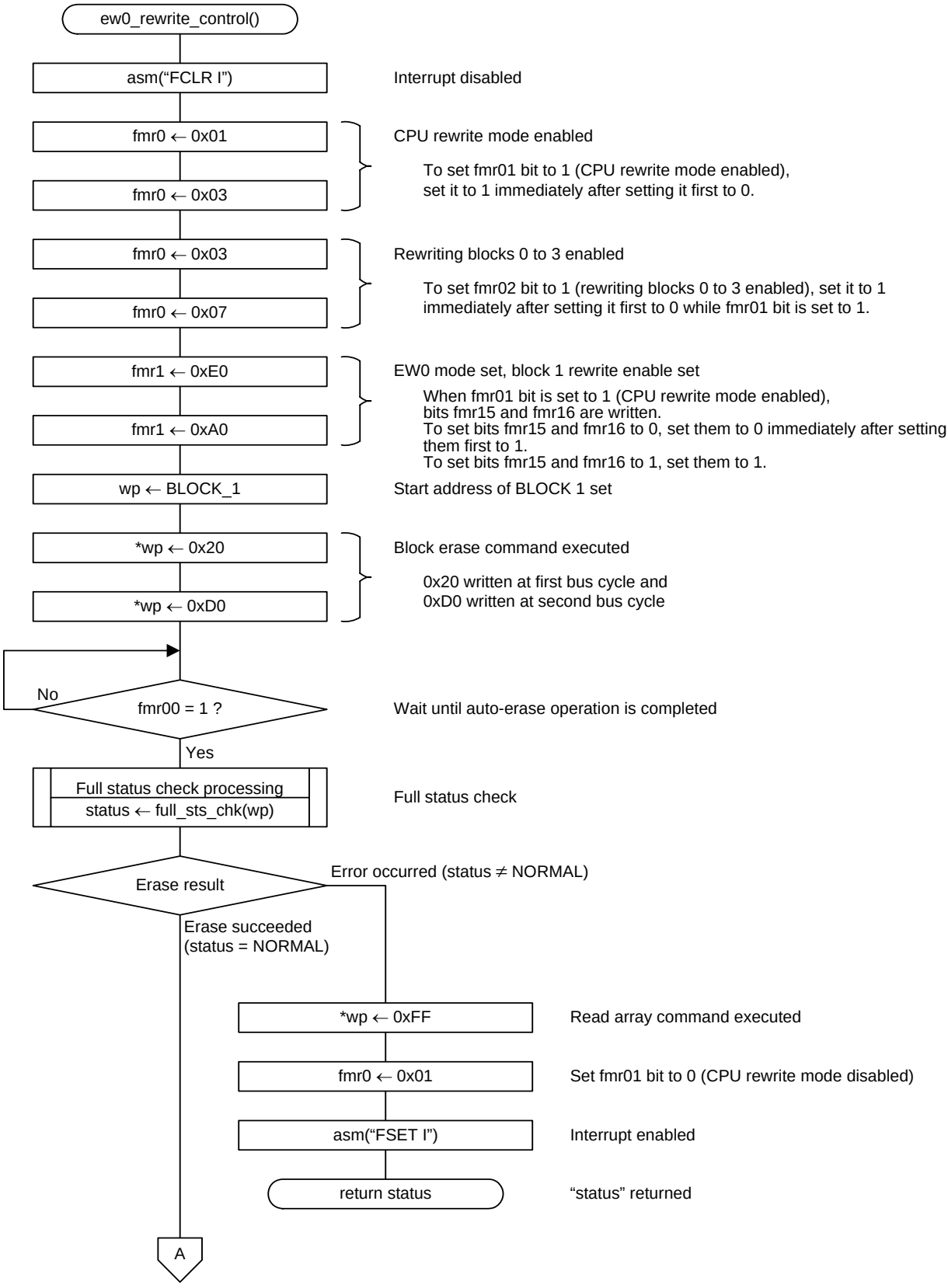


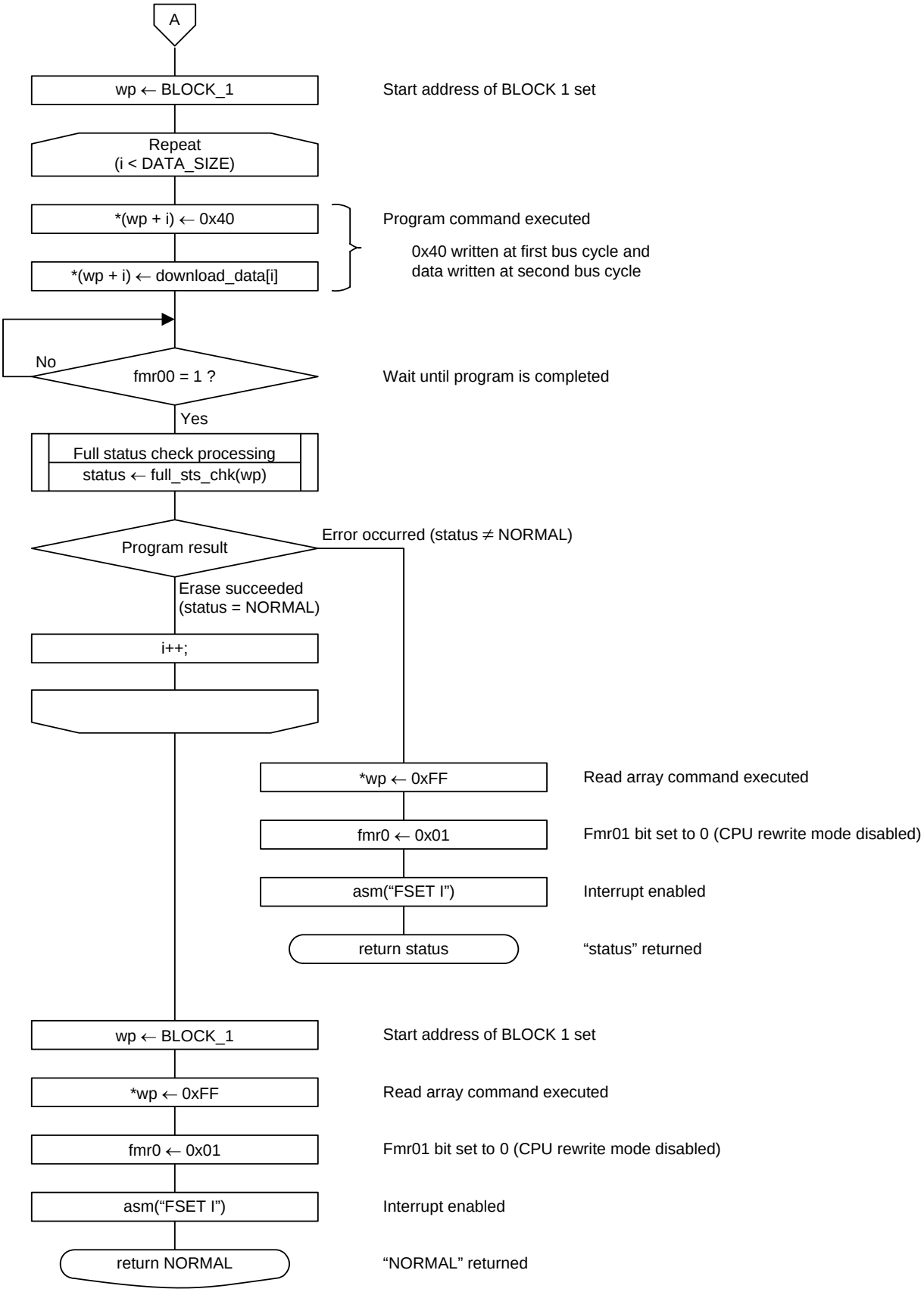
NOTE:
1. In this application note, the command sequence error processing, erase error processing, and program error processing are not executed. Execute the error processing if necessary.

5.2.2 System Clock Setting Process

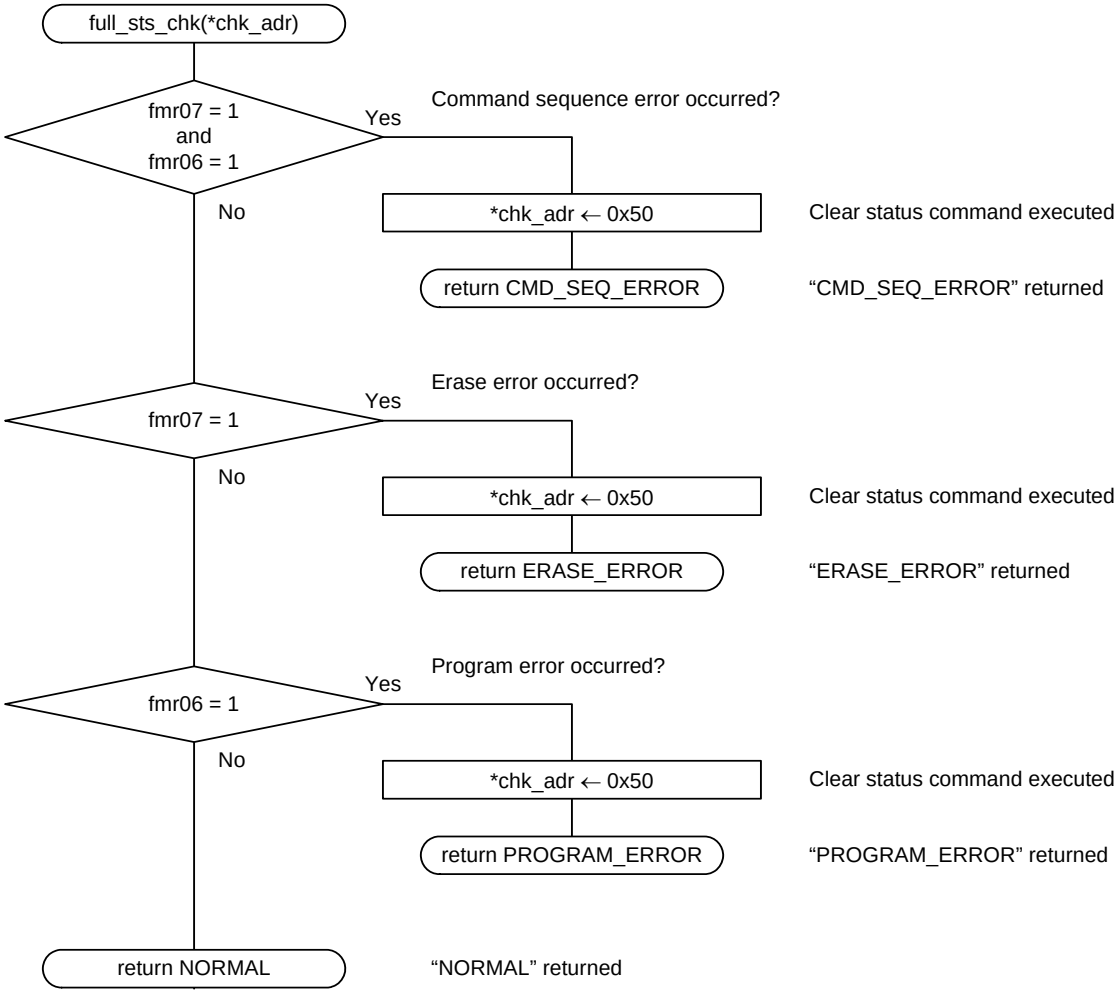


5.2.3 CPU Rewrite Control Process




A
wp ← BLOCK_1
Start address of BLOCK 1 set
Repeat
(i < DATA_SIZE)
*(wp + i) ← 0x40
Program command executed
0x40 written at first bus cycle and
data written at second bus cycle
*(wp + i) ← download_data[i]
No
fmr00 = 1 ?
Wait until program is completed
Yes
Full status check processing
status ← full_sts_chk(wp)
Program result
Error occurred (status ≠ NORMAL)
Erase succeeded
(status = NORMAL)
i++;
*wp ← 0xFF
Read array command executed
fmr0 ← 0x01
Fmr01 bit set to 0 (CPU rewrite mode disabled)
asm("FSET I")
Interrupt enabled
return status
"status" returned
wp ← BLOCK_1
Start address of BLOCK 1 set
*wp ← 0xFF
Read array command executed
fmr0 ← 0x01
Fmr01 bit set to 0 (CPU rewrite mode disabled)
asm("FSET I")
Interrupt enabled
return NORMAL
"NORMAL" returned

5.2.4 Full Status Check Process



6. Sample Programming Code

A sample program can be downloaded from the Renesas Technology Website.
To download, click “Application Notes” in the left-hand side menu of the R8C/Tiny Series page.

7. Reference Documents

Hardware Manual
R8C/2D Group Hardware Manual
The latest version can be downloaded from the Renesas Technology Website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Technology Website.

Website and Support

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

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

REVISION HISTORY	R8C/2D Group Program ROM Rewrite Using EW0 Mode

Rev.	Date	Description	
		Page	Summary
1.00	June 9, 2008	–	First Edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

© 2008. Renesas Technology Corp., All rights reserved. Printed in Japan.