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/13 Group

## Flash Rewrite Using EW1 Mode (Data "FFh" Search)

## 1. Abstract

This application note shows a data flash rewrite program using EW1.
In this application note, Block A and Block B in flash memory are shown as data flash.

## 2. Introduction

An example described in this document supports the following microcomputer.

Microcomputer : R8C/13 Group

This program can also be used when operating other microcomputers within the R8C/Tiny series, provided they have the same SFR (Special Function Registers) and data flash as the R8C/13 microcomputers. However, some functions may have been modified. Refer to the User's Manual for details.

Use functions covered in this Application Note only after careful evaluation.

## 3. Description of Application Example

### 3.1 Flash Memory Overview

A flash memory has two rewrite modes; CPR rewrite mode and standard serial I/O mode.
A CPU rewrite mode is used in this application note.

**Table 1. Overview of Flash Memory Performance**

| Item | | Performance |
|---|---|---|
| Flash Memory Operation Mode | | 2 Modes (CPU Rewrite Mode, Standard Serial I/O Mode) |
| Erase Block | | See **Figure 1 R8C/13 Memory Map** |
| Program Method | | Byte Unit |
| Erase Method | | Block Erase |
| Program, Erase Control Method | | Program and erase controlled by software command |
| Protect Method | | Blocks 0 and 1 protected by the FMR02 bit in the FMR0 register |
| | | Individual protects to Block 0 and 1 by the FMR15 and 16 bits in the FMR1 register |
| Number of Commands | | 5 commands |
| Program and Erase Endurance (Block Erase Endurance) | Block 0 and Block 1 (Program Area) | 100 times |
| | Block A and Block B (Data Area) | 10,000 times |
| ROM Code Protect | | Applicable for Standard Serial I/O Mode |

**Table 2. Overview of Flash Memory Rewrite Mode**

| Flash Memory Rewrite Mode | CPU Rewrite Mode | Standard Serial I/O Mode |
|---|---|---|
| Function | User ROM area is rewritten by executing software commands from the CPU<br>EW0 Mode : Rewritable in any area other than flash memory<br>EW1 Mode : Rewritable in flash memory | User ROM area is rewritten by using a dedicated serial programmer<br>Standard Serial I/O Mode 1: Clock Synchronous Serial I/O<br>Standard Serial I/O Mode 2: UART |
| Area can be rewritten | User ROM Area | User ROM Area |
| Operation Mode | Single Chip Mode | Boot Mode |
| ROM Programmer | - | Serial Programmer |

## 3.2 Memory Map

A user ROM area built in R8C/13 consists of 4 blocks; block A, B, 0 and 1. Figure 1 shows a memory map.
In this application note, a data flash ROM area of block A and B is used.

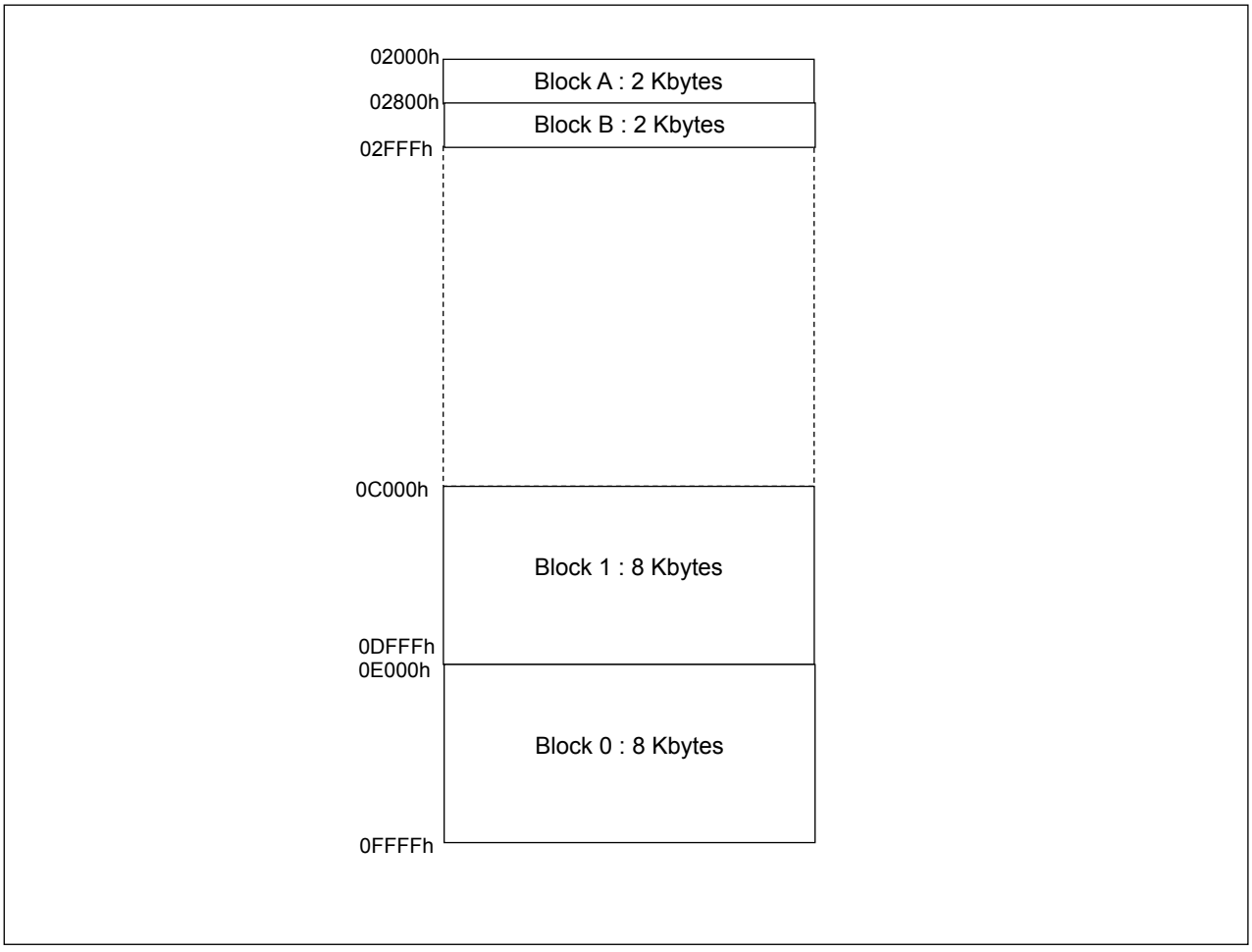



**Figure 1. R8C/13 Memory Map**

## 3.3 CPU Rewrite Mode

In CPU rewrite mode, the user ROM area can be rewritten by executing software commands from the CPU. Therefore, the user ROM area can be rewritten directly without using a ROM programmer, etc Execute the program and the Block Erase commands only on each block in the user ROM area.
In CPU rewrite mode, the user ROM area can be operated in either erase write 0 mode (EW0 mode) and erase write 1 mode (EW1 mode). Table 3 shows the difference between EW0 mode and EW1 mode.
EW1 mode is used in this application note.

**Table 3. EW0 Mode and EW1 Mode**

| Item | EW0 Mode | EW1Mode |
|---|---|---|
| Operation Mode | Single Chip Mode | Single Chip Mode |
| Area where rewrite control program can be placed | User ROM Area | User ROM Area |
| Area where rewrite control program can be executed | Rewrite control program needs to be transferred to any area other than the flash memory (e.g., RAM) | The rewrite control program can be executed in the user ROM area |
| Area which can be rewritten | User ROM Area | User ROM Area<br>(This excludes blocks with the rewrite control program) |
| Software Command Restriction | None | ·Program and Block Erase<br>(Execution disabled to blocks with the rewrite control program)<br>·Read status register command execution disabled |
| Modes after Programming or Erasing | Read Status Register Mode | Read Array Mode |
| CPU Status during Auto Write and Auto Erase | Operation | Maintains Hold State<br>(I/O ports maintain the state before the command is executed) |
| Flash Memory Status Detection | · Read the FMR00, FMR06 and FMR07 bits in the FMR0 register by a program<br>· Execute the read status register command and read the SR7, SR5 and SR4 bits in the SRD register | Read the FMR00, FMR06 and FMR07 bits in the FMR0 register by a program |

### 3.3.1 EW0 Mode

The microcomputer enters CPU rewrite mode by setting the FMR01 bit in the FMR0 register to “1” (CPU rewrite mode enabled) and is ready to accept software commands. EW0 mode is selected by setting the FMR11 bit in the FMR1 register to “0”. The software commands control programming and erasing. The FMR0 register or the SRD register indicates whether a programming or erasing operation is completed.

### 3.3.2 EW1 Mode

EW1 mode is selected by setting the FMR11 bit to “1” (EW1 mode) after the FMR01 bit is set to “1” (CPU rewrite mode enabled). The FMR0 register indicates whether a programming or erasing operation is completed. In EW1 mode, do not execute the software command in the read status register.

## 3.4 Associated Register Configuration

Figure 2 shows the required registers to control flash memory

### Flash Memory Control Register 0

| b7 b6 b5 b4 b3 b2 b1 b0 | Symbol | Address | After reset |
|---|---|---|---|
| 0 0 | FMR0 | 01B7h | XX00 0001b |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| FMR00 | RY/$\overline{BY}$ Status Flag | 0 : Busy (During writing or erasing)<br>1 : Ready | RO |
| FMR01 | CPU Rewrite Mode Select Bit(1) | 0 : CPU rewrite mode disabled<br>1 : CPU rewrite mode enabled | RW |
| FMR02 | Block 0 and 1 Rewrite Enable Bit(2) | 0 : Rewrite Disabled<br>1 : Rewrite Enabled | RW |
| FMSTP | Flash Memory Stop Bit(3,5) | 0 : Starts flash memory<br>1 : Stops the flash memory<br>(Enters low-power consumption state and flash memory is reset) | RW |
| ——— (b5-b4) | Reserved Bit | Set to “0”. | RW |
| FMR06 | Program Status Flag(4) | 0 : Completed successfully<br>1 : Teminated by error | RO |
| FMR07 | Erase Status Flag(4) | 0 : Completed successfully<br>1 : Teminated by error | RO |

NOTES :
1. When setting this bit to “1”, set 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”. Set the microcomputer in read array mode before writing to this bit.
2. Set the FMR01 bit 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 this bit to “0” and setting it to “1”.
3. Set this bit by a program in a space other than the flash memory.
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). The FMSTP bit is set to “1” when the FMR01 bit is set to “0” (CPU rewrite mode disabled) and the FMSTP bit is set to “1”. The flash memory does not enter low-power consumption state nor is reset.

### Flash Memory Control Register 1

| b7 b6 b5 b4 b3 b2 b1 b0 | Symbol | Address | After reset |
|---|---|---|---|
| 1 0 0 0 | FMR1 | 01B5h | 1000 000Xb |

| Bit Symbol | Bit Name | Function | |
|---|---|---|---|
| ——— (b0) | Reserved Bit | When read, its content is indeterminate. | RO |
| FMR11 | EW1 Mode Select Bit(1,2) | 0 : EW0 mode<br>1 : EW1 mode | RW |
| ——— (b4-b2) | Reserved Bit | Set to “0”. | RW |
| FMR15 | Block 0 Rewrite Disabled Bit(2,3) | 0 : Rewrite enabled<br>1 : Rewrite disabled | RW |
| FMR16 | Block 1 Rewrite Disabled Bit(2,3) | 0 : Rewrite enabled<br>1 : Rewrite disabled | RW |
| ——— (b7) | Reserved Bit | Set to “1”. | RW |

NOTES :
1. Set the FMR01 bit 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 this bit to “0” and setting it to “1”.
2. When setting the FMR01 bit to “0” (CPU rewrite mode disabled), this bit is set to “0”.
3. When the FMR01 bit is set to “1” (CPU rewrite mode enabled), the FMR15 and FMR16 bits are written. When setting this bit to “0”, set this bit to “0” immediately after setting it first to “1”. When setting this bit to “1”, set this bit to “1”.

**Figure 2. Associated Register**

## 3.5 Software Command

The software commands are described below. Read or write the command code and data in 8-bit unit.

**Table 4. Software Commands**

| Software Commands | First Bus Cycle | | | Second Bus Cycle | | |
|---|---|---|---|---|---|---|
| | Mode | Address | Data ($D_7$ to $D_0$) | Mode | Address | Data ($D_7$ to $D_0$) |
| Read Array | Write | × | FFh | | | |
| Read Status Flag | Write | × | 70h | Read | × | SRD |
| Clear Status Flag | Write | × | 50h | | | |
| Program | Write | WA | 40h | Write | WA | WD |
| Block Erase | Write | × | 20h | Write | BA | D0h |

SRD : Status Register ($D_7$ to $D_0$)

WA : Address to be written (The address specified in the first bus cycle is the same address specified in the second bus cycle)

WD : 8-bit write data

BA : Highest -order block address

× : An even address in the user ROM area

### 3.5.1 Read Array Command

The read array command reads the flash memory.

By writing “FFh” in the first bus cycle, read array mode is entered. Content of a specified address can be read in 8-bitunit after the next bus cycle. The microcomputer remains in read array mode until another command is written. Therefore, contents from multiple addresses can be read consecutively.

### 3.5.2 Read Status Register Command

The read status register command reads the SRD register.

By writing “70h” in the first bus cycle, the SRD register can be read in the second bus cycle. Read an address in the user ROM area. In EW1 mode, do not execute this command.

### 3.5.3 Clear Status Register Command

The clear status register command reads the SRD register.

By writing “50h” in the first bus cycle, the FMR06 to FMR07 bits in the FMR0 register and the SR4 to SR5 bits in the SRD register are set to “0”.

### 3.5.4 Program

The program command writes 1-byte data to the flash memory.
By writing “40h” in the first bus cycle and data to the write address in the second bus cycle, an auto programming operation (data program and verify) start. Set the address value specified in the first bus cycle to the same address as the write address specified in the second bus cycle.
The FMR00 bit in the FMR0 register indicates whether an auto programming operation is completed. The FMR00 bit is set to“0” during auto programming and “1” when an auto programming operation is completed.
After an auto programming is completed, the FMR06 bit in the FMR0 register indicates whether an auto programming operation is completed. Do not write additions to the programmed address. When the FMR02 bit in the FMR0 register is set to “0” (rewrite disabled) or the FMR02 bit is set to “1” (rewrite enabled) and the FMR15 bit in the FMR1 register is set to “1” (rewrite disable), the program command on the block0 can not be acknowledged. When the FMR16 bit is set to “1” (rewrite disabled), the program command on the block1 can not be acknowledged. In EW1 mode, do not execute this command to any address at which the rewrite control program is located.

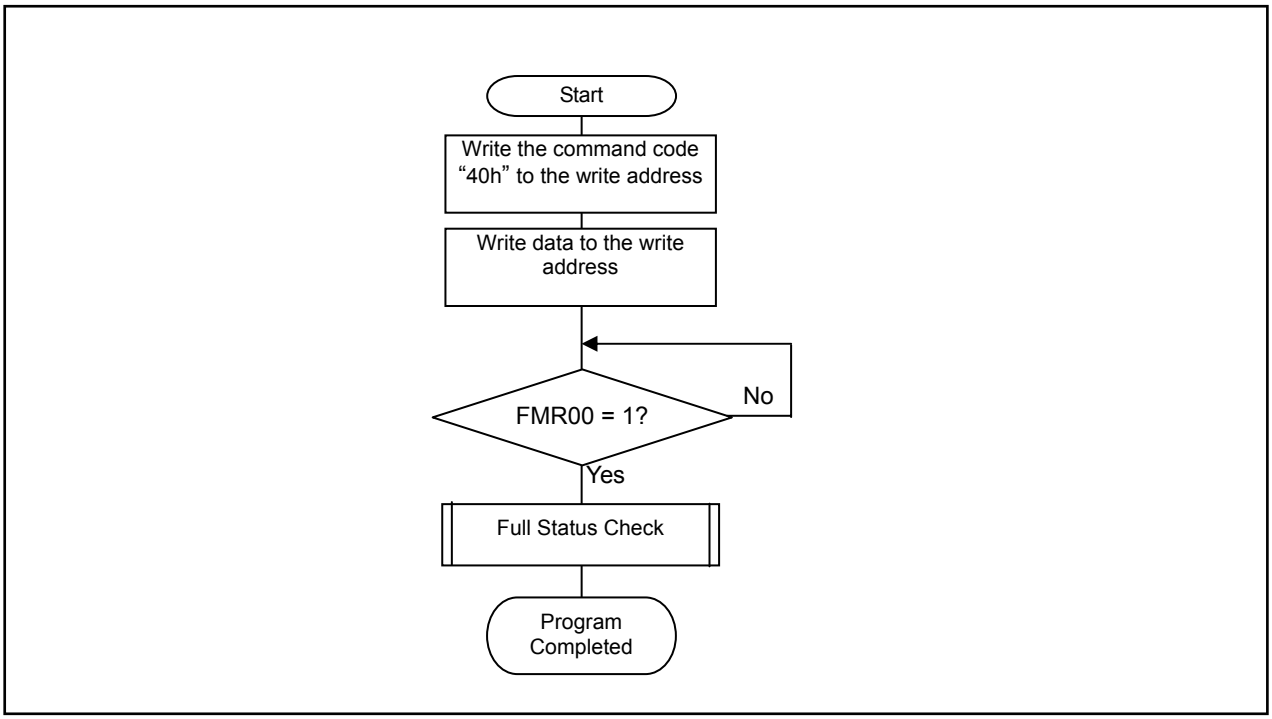



**Figure 3. Program Flow Chart**

### 3.5.5 Block Erase

By writing “20h” in the first bus cycle and “D0h” in the second bus cycle to the highest-order address of a block, an auto erasing operation (erase and erase verify) starts in the specified block. The FMR00 bit in the FMR0 register indicates whether an auto erasing operation is completed. The FMR00 bit is set to “0” during auto erasing operation and “1” when the auto erasing operation is completed.

After the auto erasing operation is completed, the FMR07 bit in the FMR0 register indicates whether the auto erasing operation is completed. When the FMR02 bit in the FMR0 register is set to “0” (rewrite disabled) or the FMR02 bit is set to “1” (rewrite enabled) and the FMR15 bit in the FMR1 register is set to “1” (rewrite disabled), the block erase command on the block0 can not be acknowledged. When the FMR16 bit is set to “1” (rewrite disabled), the block erase command on the block1 can not be acknowledged. Figure 4 shows block erase command when the erase-suspend function is not used. In EW1 mode, do not execute this command on any address at which the rewrite control program is located.

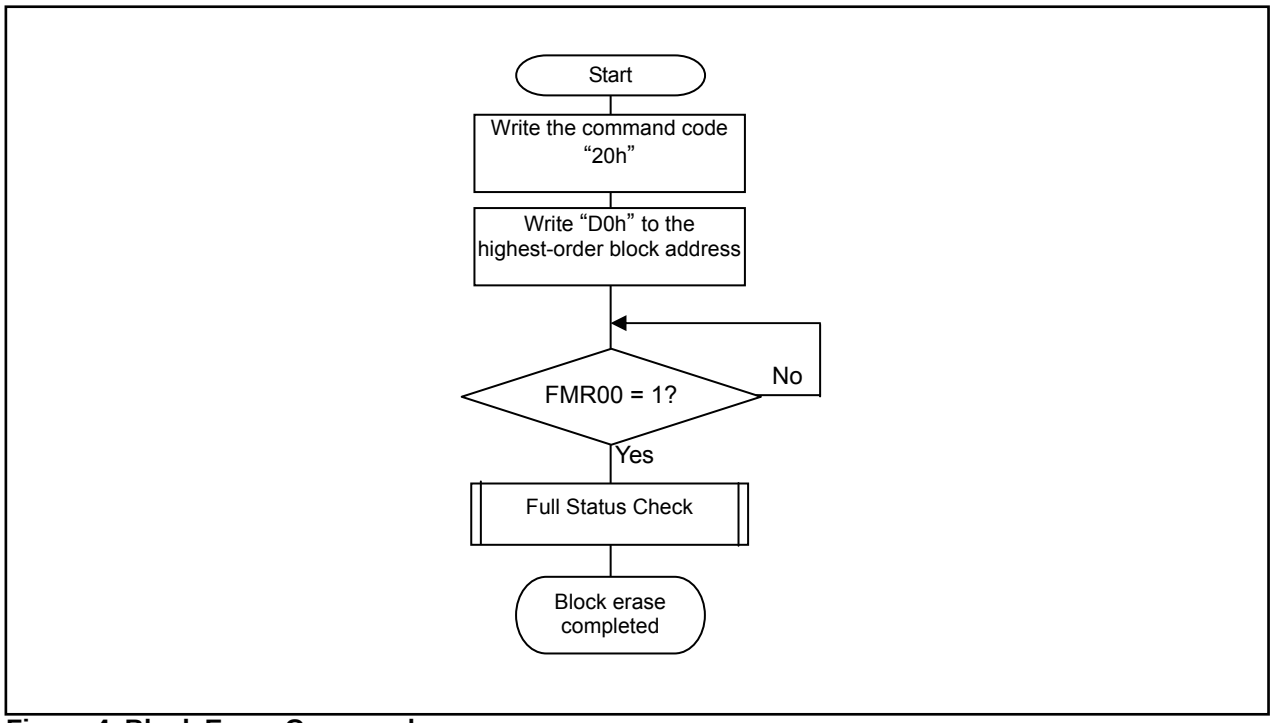



**Figure 4. Block Erase Command**

### 3.5.6 Status Register (SRD Register)

The SRD register indicates the flash memory operation state and whether an erasing or programming operation is completed successfully or in error. The FMR00, FMR06 to FMR07 bits in the FMR0 register indicate the SRD register states. Table 5 shows the SRD register. In EW0 mode, the SRD register can be read in the following cases:

(1) Any given addresses in user ROM area are read after writing the read status register command.

(2) Any given addresses in user ROM area are read when the program or block erase commands is executed until the read array command is executed.

**Table 5. SRD Register**

| Bits in SRD Register | Bits in FMR0 Register | Status Name | Definition | | After Reset |
|---|---|---|---|---|---|
| | | | "0" | "1" | |
| SR7($D_7$) | FMR00 | Sequence Status | Busy | Ready | 1 |
| SR6($D_6$) | - | Reserved Bit | - | - | - |
| SR5($D_5$) | FMR07 | Erase Status | Completed successfully | Completed in error | 0 |
| SR4($D_4$) | FMR06 | Program Status | Completed successfully | Completed in error | 0 |
| SR3($D_3$) | - | Reserved Bit | - | - | - |
| SR2($D_2$) | - | Reserved Bit | - | - | - |
| SR1($D_1$) | - | Reserved Bit | - | - | - |
| SR0($D_0$) | - | Reserved Bit | - | - | - |

$D_0$ to $D_7$ : Data buses are read when the read status register command is executed.
FMR07 (SR5) to FMR06 (SR4) bits are set to "0" by executing the clear status register command.
When the FMR07 bit (SR5) or FMR06 bit (SR4) is set to "1", the program and block erase command can not be acknowledged.

#### ●Sequence Status (SR7, FMR00 bits)

The sequence status indicates the flash memory state. This bit is set to "0" (busy) during an auto programming or auto erasing. It is set to "1" (ready) as soon as these operations are completed.

#### ●Erase Status (SR5, FMR07 bits)

Refer to **3.5.7 Full Status Check**.

#### ●Program Status (SR4, FMR06 bits)

Refer to **3.5.7 Full Status Check**.

### 3.5.7 Full Status Check

If an error occurs, the FMR06 to FMR07 bits in the FMR0 register are set to “1”, indicating a specific error. Therefore, Checking these status (full status check) indicates whether an erasing or programming operation is completed. Table 6 shows the errors and FMR0 register state. Figure 5 shows a flow chart of the full status check and handling procedure for each error.

**Table 6. Errors and FMR0 Register State**

| FMR0 register (SRD register) State | | Error | Error Occurrence Conditions |
|---|---|---|---|
| FMR07(SR5) | FMR06(SR4) | | |
| 1 | 1 | Command sequence error | · Command is written incorrectly<br>· A value other than “D0h” or “FFh” is written in the second bus cycle of the block erase command(1) |
| 1 | 0 | Erase error | · When the block erase command is executed and auto erasing operation is not completed correctly. |
| 0 | 1 | Program error | · When the program command is executed and auto programming operation is not completed correctly. |

NOTES:
1. The flash memory enters read array mode by writing command code “FFh” in the second bus cycle of these commands. The command code written in the first bus cycle becomes disabled.

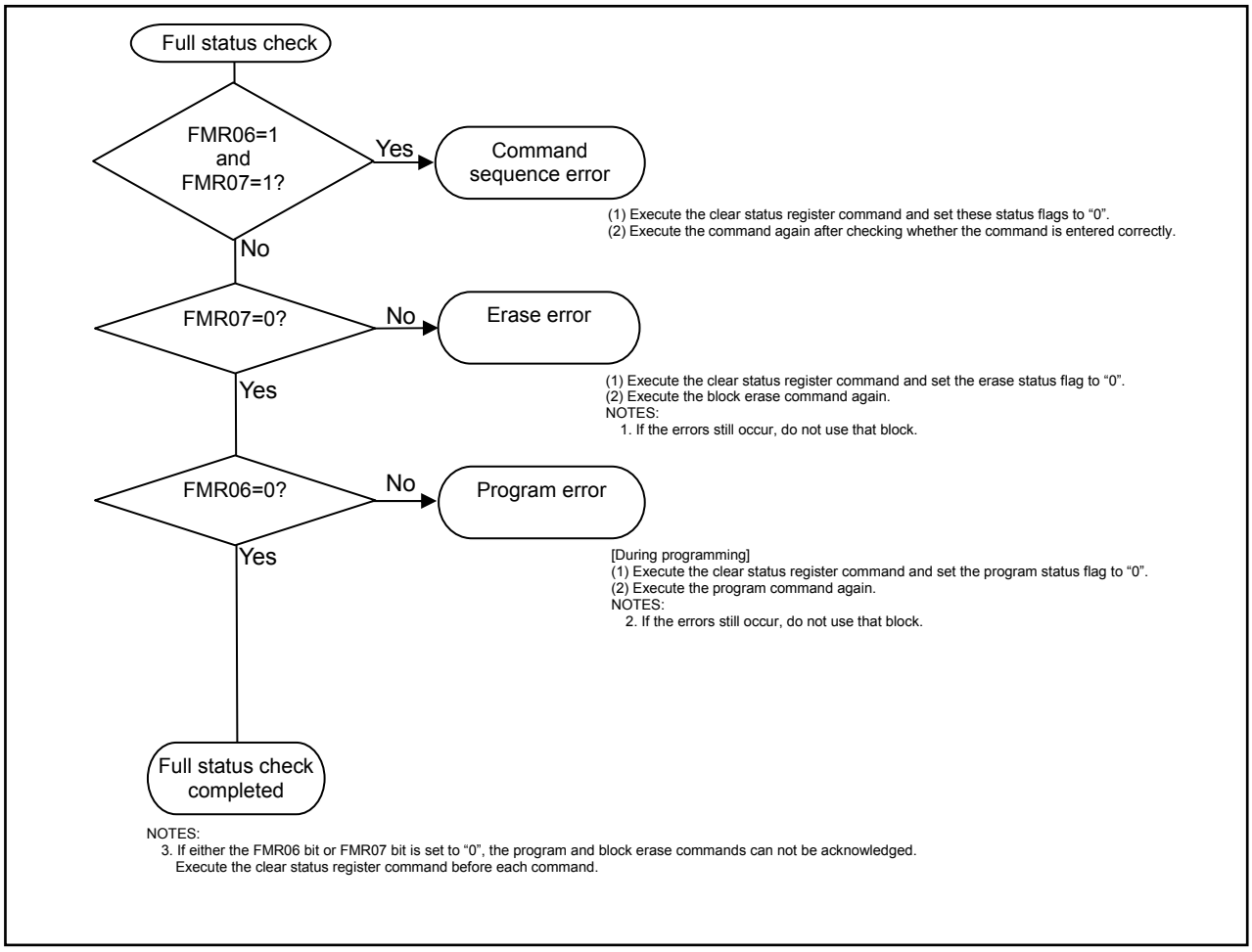



**Figure 5. Full Status Check and Handling Procedure for Each Error**

## 4. Program Overview

Dividing the data flash areas in record-unit and how to write data sequentially are described below.
This application note assumes that one record is made into 128 bytes. The block A, which is divided into 16 records from record 0 to 15, and the block B, which is divided into 16 records from record 16 to 31, are used as data area. Figure 6 shows the connection between data flash and records.

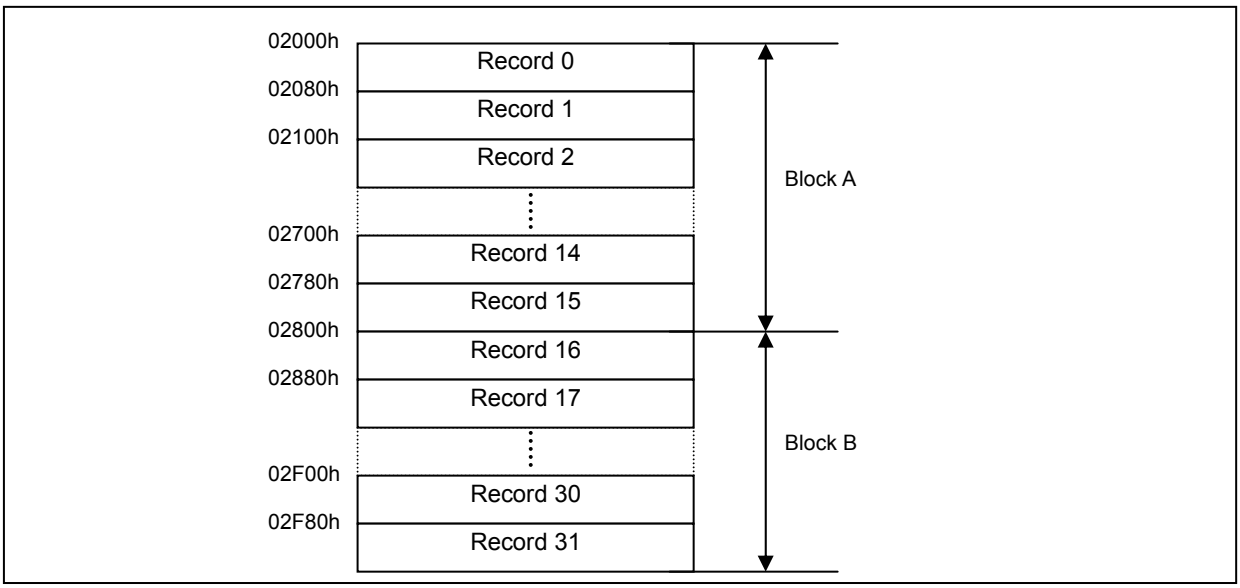



**Figure 6. Connection between Data Flash and Records**

When writing data, write from the record 0 in record-unit. Erase (block erase) all contents in the block B after writing to the record 15 and erase all contents in the block A after writing to the record 31. When writing data for the next step, write to the record 0. The data which is written into the data flash is maintained after turning the power off. Therefore, in this application note, all records (empty records) which contain data of “FFh” are searched after reset start. The following is described how to search empty records.

(1) Set the search pointer to the starting address in the record 0(Figure 7)

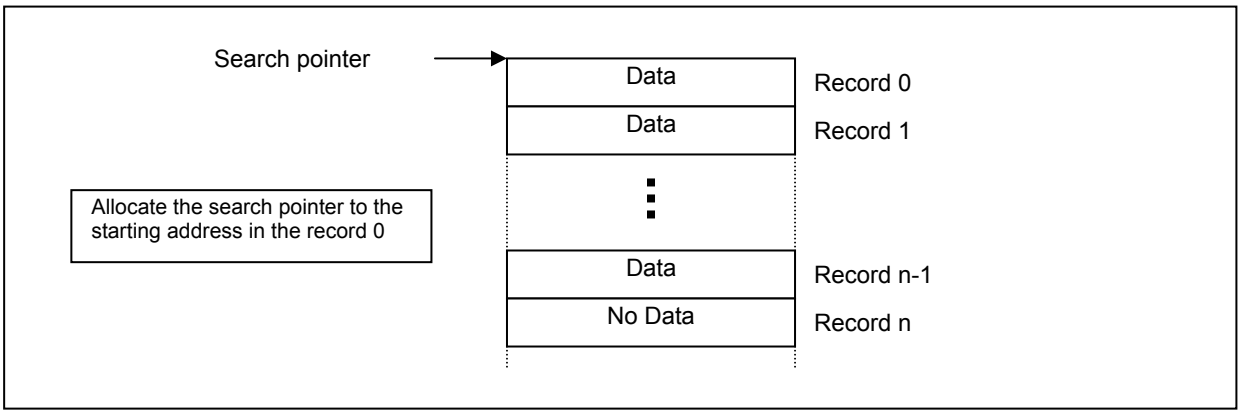



**Figure 7. Allocate the Search Pointer to the Starting Address in the Record 0**

(2) Check whether the records which the search pointer indicates are empty records (ALL“FFh”)

(3) When the records are not empty records, set the search pointer to the starting address of the next record. (Figure 8)

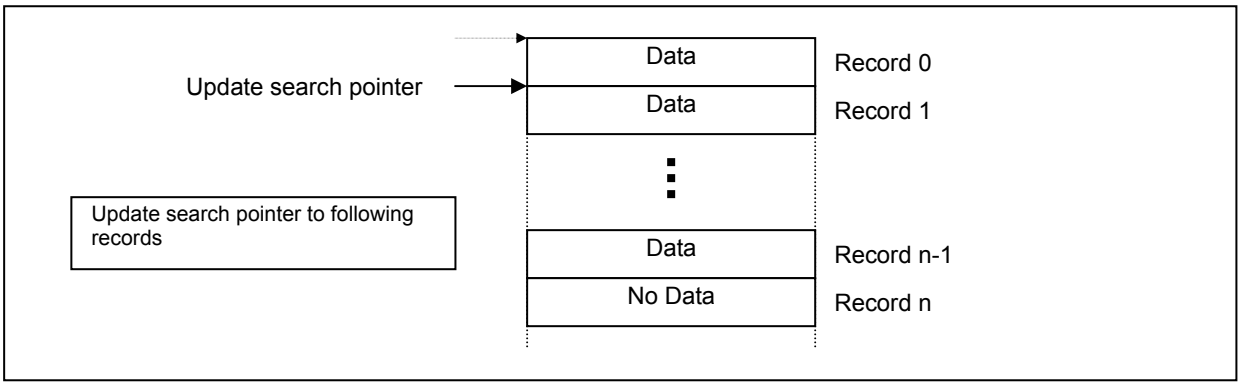



**Figure 8. Update Search Pointer**

(4) Execute (2) and (3) repeatedly until the empty records can be recognized or all records can be checked.

(5) When the empty records are recognized, set the starting address of empty records to the data write address and memorize the block which stores the records as a block select (Figure 9).

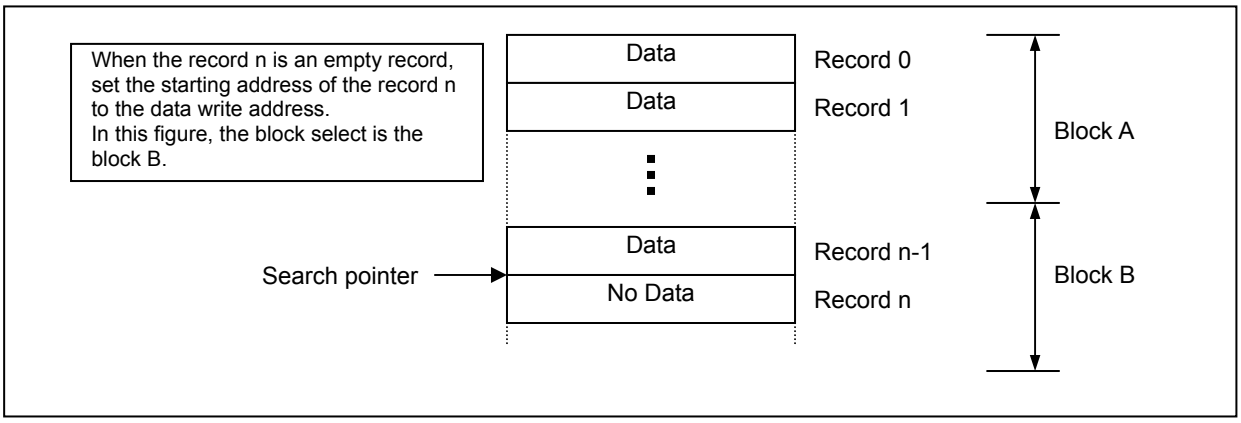



**Figure 9. When the Empty Records are Recognized**

(6) When the empty record are not recognized in the both block A and B, erase the block A and set the starting address of the record 0 to the data write address. The block A is memorized as a block select.

## 4.1 Function Table

| | | | |
|---|---|---|---|
| Declaring | void write_address_init(void) | | |
| Outline | Reset write address to default | | |
| Argument | Argument Type | | Meaning |
| | None | | |
| Use Variable (Global) | Variable Type | | Contents of Use |
| | unsigned char *write_addr | | Initial Setting |
| | unsigned char block_select | | Initial Setting |
| Return Value | Return Value Type | Value | Meaning |
| | None | | |
| Function Description | Search the empty records. Set the write address (write_addr) and block select (block_select) | | |

| | | | |
|---|---|---|---|
| Declaring | unsigned char flash_write(unsigned char *data) | | |
| Outline | Data Write Control | | |
| Argument | Argument Type | | Meaning |
| | unsigned char *data | | Table Starting Address of Write Data |
| Use Variable (Global) | Variable Type | | Contents of Use |
| | unsigned char *write_addr | | Reference/Setting |
| | unsigned char block_select | | Reference/Setting |
| Return Value | Return Value Type | Value | Meaning |
| | unsigned char | COMPLETE | Completed successfully |
| | | PROGRAM_ERR | Write Error(1) |
| | | ERASE_ERR | Erase Error(1) |
| Function Description | Update the write address (write_addr) after writing the reecord data. When the block select has no empty records, erase unused blocks and change the block select (block_select).<br>NOTES: 1. When write or erase error occurs, PROGRAM_ERR+ERASE_ERR is indicated. | | |

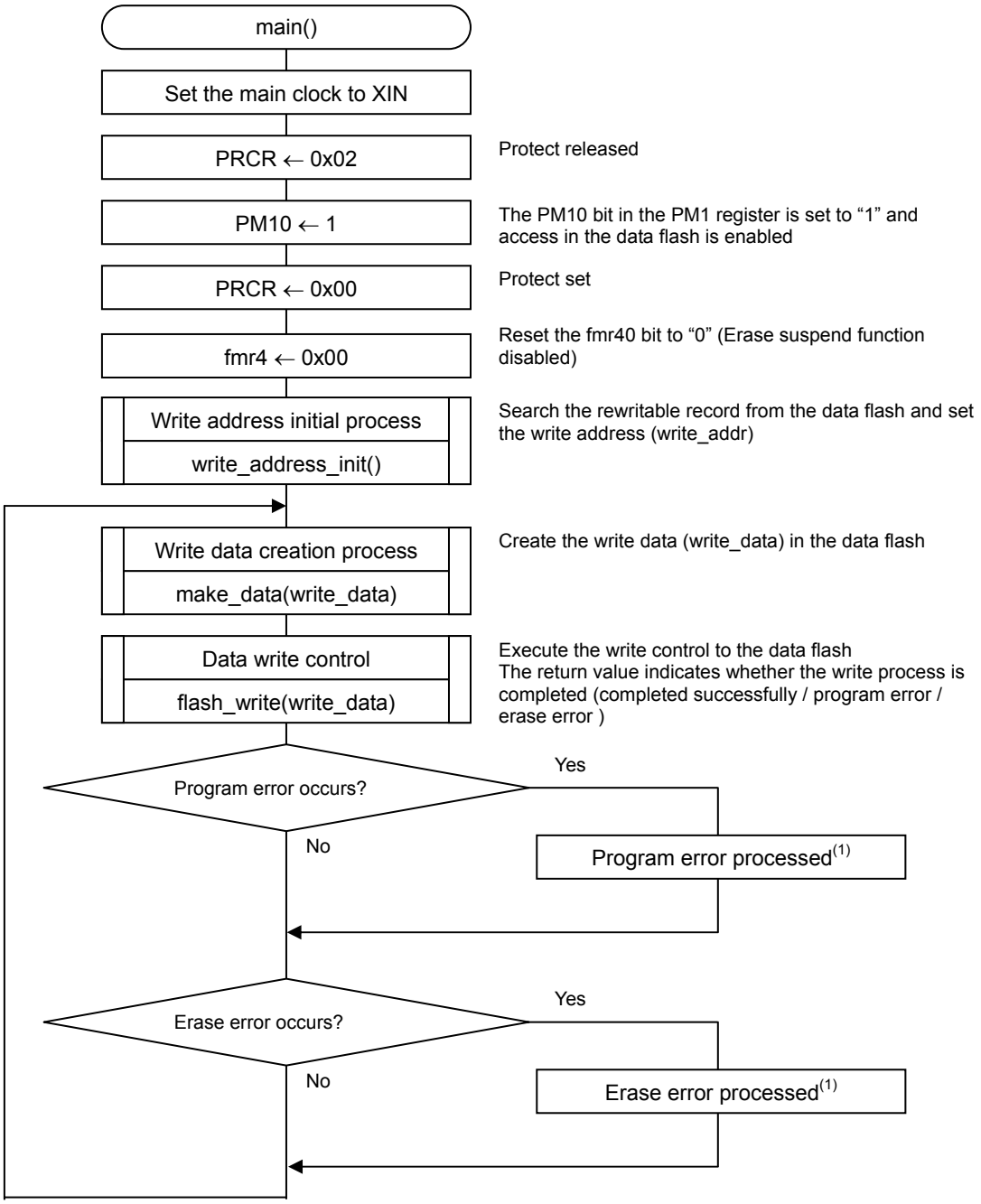
| | | | |
|---|---|---|---|
| Declaring | unsigned char block_erase(unsigned char *ers_addr) | | |
| Outline | Block Erase Process | | |
| Argument | Argument Type | | Meaning |
| | unsigned char *ers_addr | | Starting Address of Erase Block |
| Use Variable (Global) | Variable Type | | Contents of Use |
| | None | | |
| Return Value | Return Value Type | Value | Meaning |
| | unsigned char | COMPLETE | Completed successfully |
| | | ERASE_ERR | Erase Error |
| Function Description | Erase the specified block in EW1 mode | | |

| | | | |
|---|---|---|---|
| Declaring | unsigned char data_write(unsigned char *write_data) | | |
| Outline | Data Write Process | | |
| Argument | Argument Type | | Meaning |
| | unsigned char *write_data | | Table Starting Address of Write Data |
| Use Variable (Global) | Variable Type | | Contents of Use |
| | unsigned char *write_addr | | Reference |
| Return Value | Return Value Type | Value | Meaning |
| | unsigned char | COMPLETE | Completed successfully |
| | | PROGRAM_ERR | Write Error |
| Function Description | Write data for one record from write address (write_addr) in EW1 mode. | | |

| | | |
|---|---|---|
| Declaring | void make_data(unsigned char *write_data) | |
| Outline | Create the write record data in the data flash | |
| Argument | Argument Type | Meaning |
| | unsigned char *data | Table Starting Address of Write Data |
| Use Variable | Variable Type | Contents of Use |
| Return Value | None | |
| | Return Value Value | Meaning |
| | None | |
| Function Description | Create the write record data in the data flash. In this application note, nothing is processed. Add the processes if needed. | |

## 4.2 Flow Chart

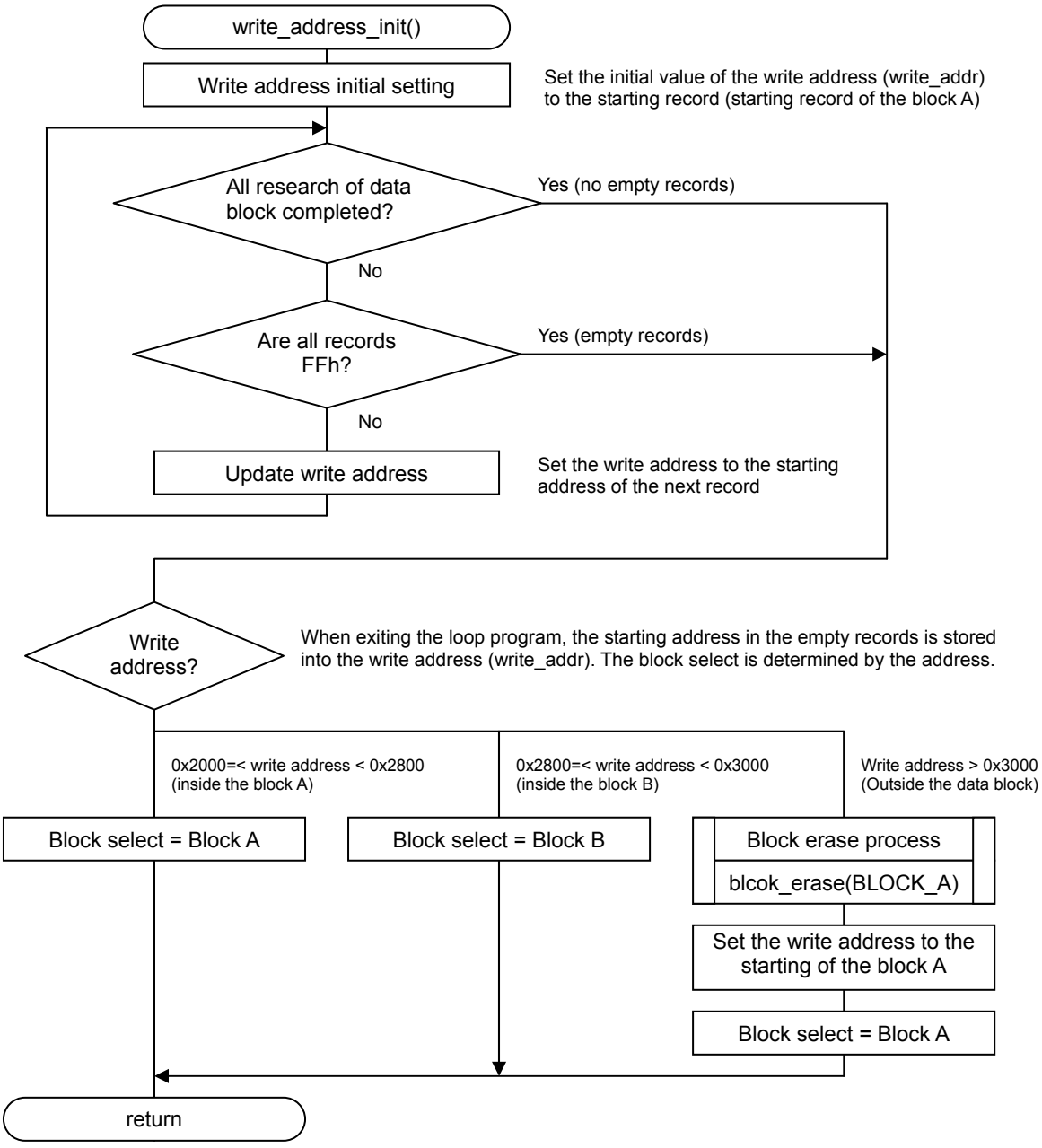
### 4.2.1 Main Function



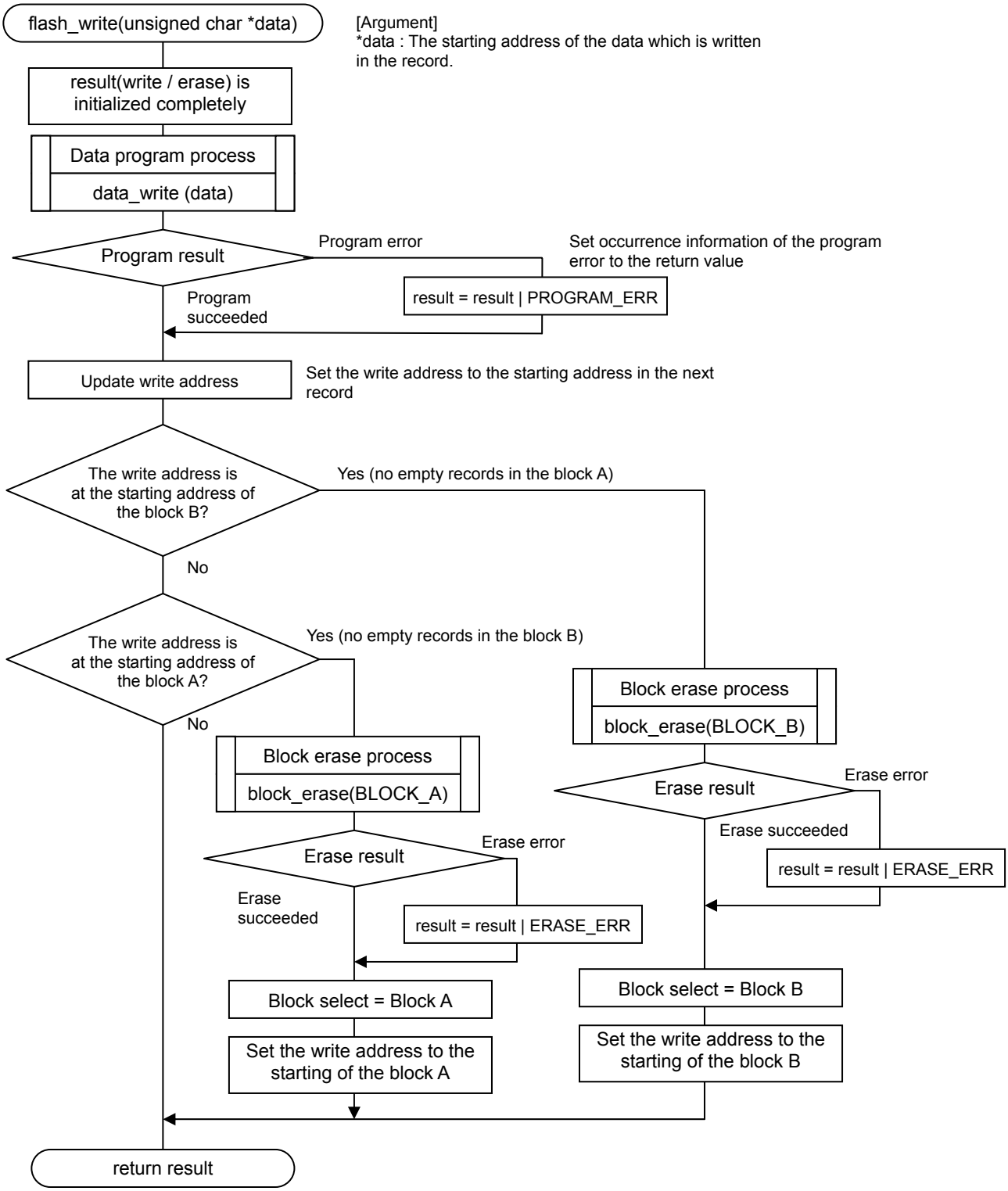

NOTES:

In this application note, error processes are not performed. Process errors if needed.

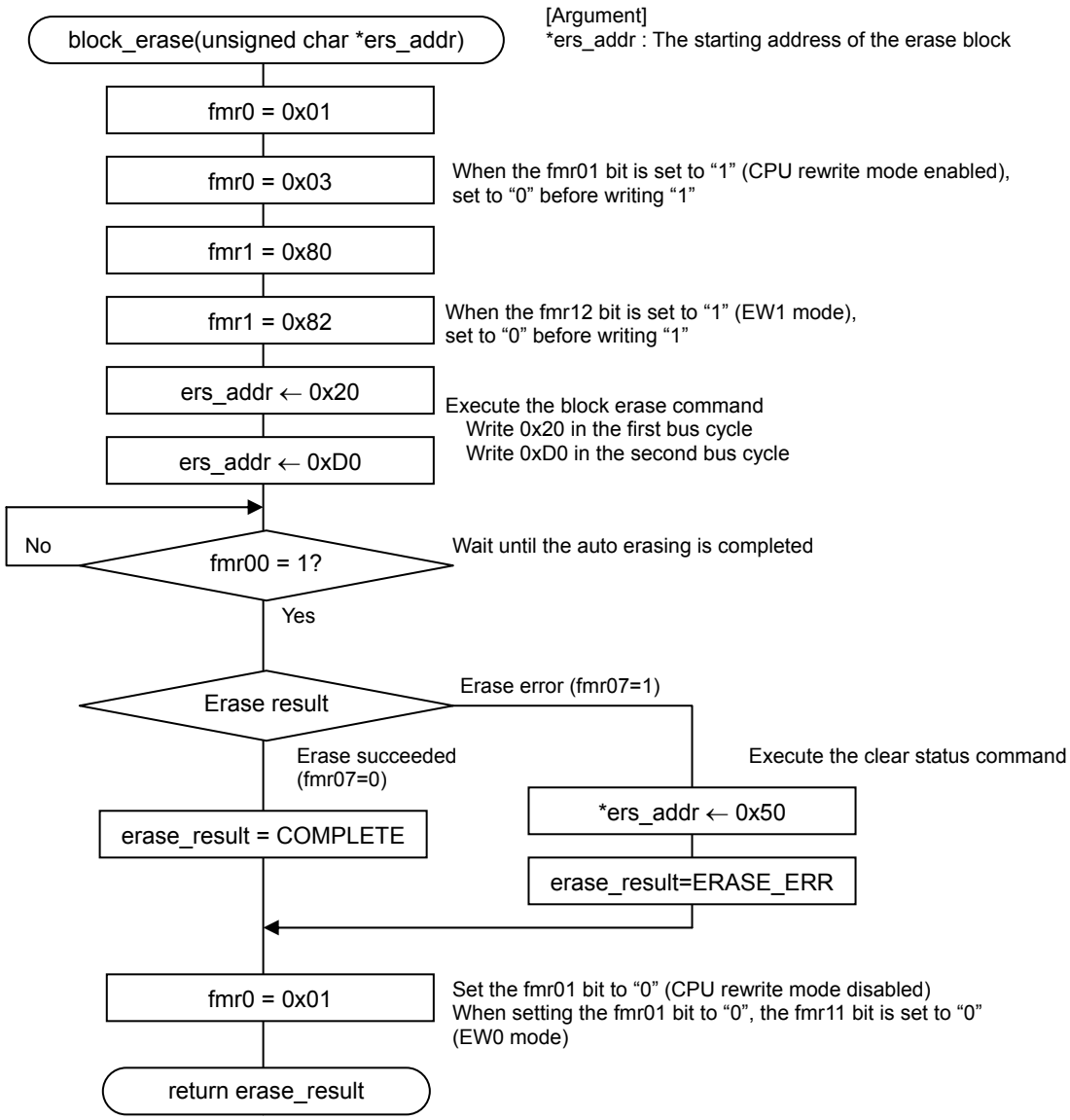
### 4.2.2 Write Address Initial Function



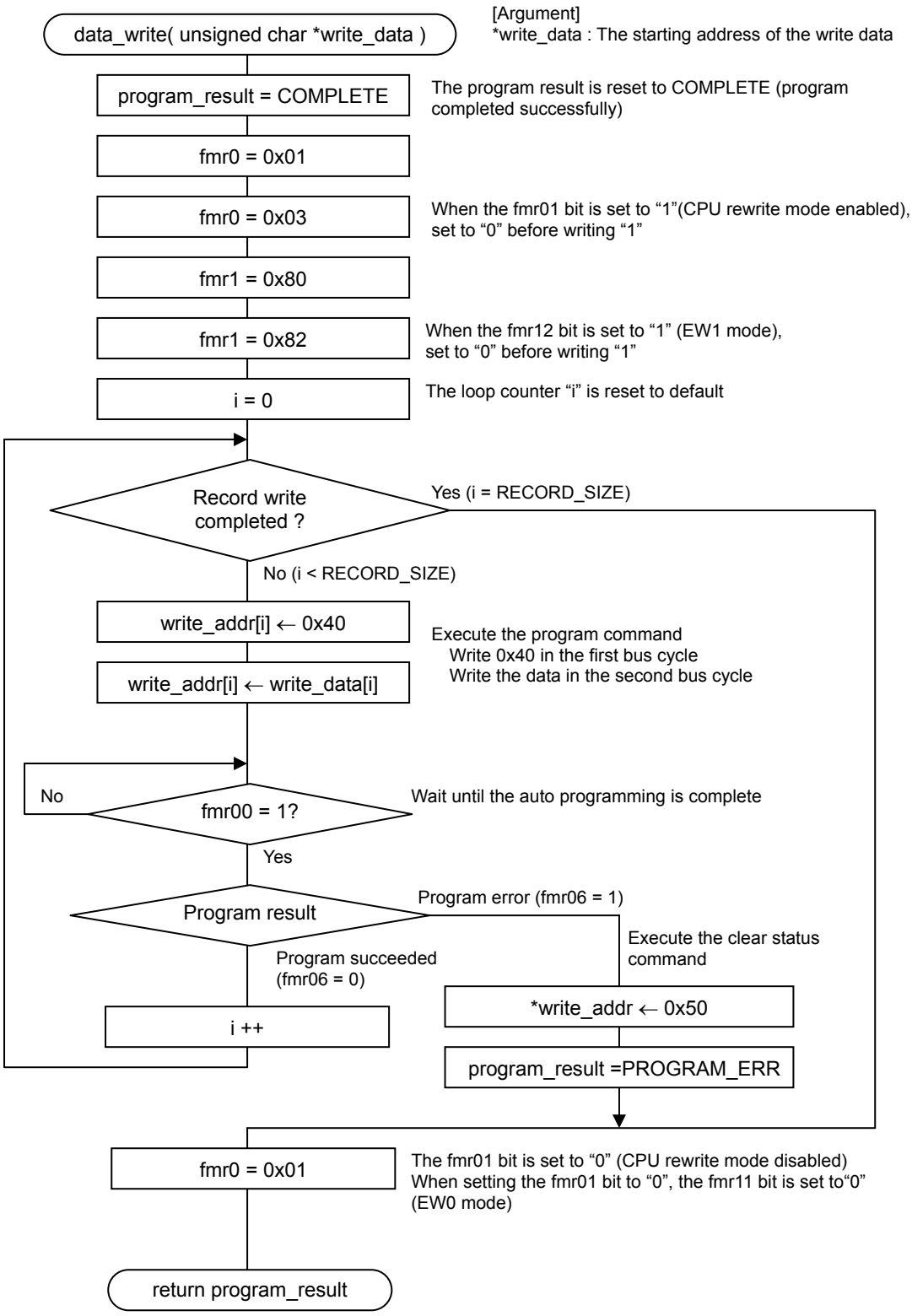
### 4.2.3 Data Write Control Function



### 4.2.4 Block Erase Function



### 4.2.5 Data Program Function



## 5. Sample Programming Code

Download a sample program from the Renesas Technology website.
To download, click “Application Notes” in the left-hand side menu on the top page of the R8C/Tiny Series.

## 6. Reference Documents

Hardware Manual
R8C/13 Group Hardware Manual
(Download the latest version from the Renesas Technology website.)

Technical News/Technical Update
(Download the latest information from the Renesas Technology website.)

## Website and Support

Renesas Website

http://www.renesas.com

Inquiries

http://www.renesas.com/inquiry

csc@renesas.com

| REVISION HISTORY | R8C/13 Group<br>Flash Rewrite Using EW1 Mode (Data "FFh" Search) |
|---|---|

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.00 | Mar 22, 2004 | - | First edition issued |
| 1.10 | Aug 23, 2004 | - | Sample Program Comment is revised |
| 1.20 | Apr 01, 2005 | 19 | Sample Programming Code is deleted |
| 2.00 | Jul 24, 2006 | 17,18 | The set value of FMR17 bit (reserved bit) is changed to “1” |

## Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

## Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.