H8S/2400 Series

Programming Flash Memory by User Programming Mode (EW0 Mode)

R01AN0770EJ0102
Rev.1.02
Sep 07, 2011

Introduction

This application note describes an example of programming the on-chip flash memory in the user programming mode.

This mode is provided for the on-board programming of flash memory in H8S/2400-series products.

Target Device

H8S/2456, H8S/2456R, H8S/2454 Group

Preface

This application note was prepared using the H8S/2456, H8S/2456R, H8S/2454 Group, one of the devices on which operation has been confirmed, as the basis.

This program can be used with other H8S/2400 Series MCUs that have the same internal I/O registers as the devices on which operation has been confirmed. Check the latest version of the manual for any additions and modifications to functions.

Careful evaluation is recommended before using this application note.

Contents

1. Specifications

This application note describes an example of erasing a block of the on-chip flash memory (erasure), and writing data to all address regions within a block that has been erased (programming).

The following are the detailed specifications of the application covered by this application note.

- The erasure control program erases the contents of block 1 of the on-chip flash memory (user ROM: 64 Kbytes).
- The programming control program writes 256 bytes of data to all 256-byte ranges within block 1.
- The write data is provided by software at addresses from H'00 to H'FF in the on-chip RAM.
- Whether contents of the on-chip flash memory match the expected data after erasure and programming is verified.
- The erasure and programming sequences are started by controlling SW1 and SW2, respectively.
- The LED is turned on or off as described in table 1, Specifications for LED Operations.

Note: This application include the programming control program within block 0. Do not program this block.

Figure 1 is an overview of the application.

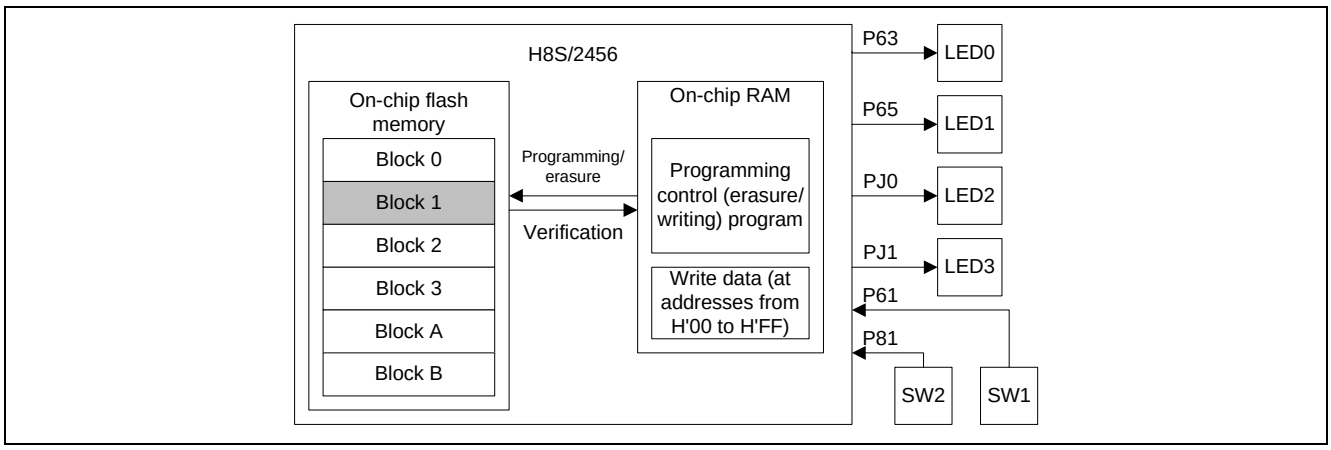


Figure 1 Overview

Table 1 gives the specifications for LED operation.

Table 1 Specification for LED Operations

On: ○, Off: ●, Blinking: ⊙

State			LED0 (Erasure)	LED1 (Programming)	LED2 (Error/no error)	LED3 (Check error)
At startup			●	●	●	●
Erasure	Start (Erasure in progress)		○	●	●	●
	Completed		⊙	●	●	●
	Error	Erasure error	○	●	○	●
		Blank check error	○	●	○	○
Program Operation	Start (programming in progress)		●	○	●	●
	Completed		●	⊙	●	●
	Error	Programming error	●	○	○	●
		Verification error	●	○	○	○

2. Conditions for Application

Table 2 shows the conditions for running the application.

Table 2 Conditions for Application

Item		Contents	Remarks
Operating frequency	Input clock	16 MHz	
	System clock (ϕ)	32 MHz (16 MHz multiplied by 2)	
Operating voltage		3.3 V	
Operating mode		On-chip ROM enabled expanded mode Mode 4 (MD2 = 1, MD1 = 0, MD0 = 0)	
Integrated development environment		High-performance Embedded Workshop	Version 4.09.00.007
Evaluation board		Renesas Electronics Renesas Starter Kit for H8S/2456R R0K524568S000BE	
C/C++ compiler		H8S,H8/300 C/C++ Compiler	V.6.02.02
Optimizing linkage editor		Optimizing Linkage Editor	V.9.05.00
Compile options		-cpu=2600A:24 -nolist -chgincpath -nologo	
Linker options		-noprelink -rom=D=R,PFWRMAIN=RFWRMAIN(*) -nomessage -nooptimize - start=PResetPRG,PIntPRG/0400,P,C,C$DSEC,C$ BSEC,D/0800,PFWRMAIN/02000,B,R/0FEC000,R FWRMAIN/0FF0000,S/0FFBE00 -nologo - form=stype -exit	

Note: * For this application, the ROM-support function (rom option) is used to allocate a RFWRMAIN section of the same size as the PFWRMAIN section and to relocate the symbols specified in the PFWRMAIN section to addresses in the RFWRMAIN section. The programming control program is allocated to the PFWRMAIN section.

3. Description of Modules Used

In this application note, flash memory is used in user programming mode.

In the user programming mode, the flash memory can be programmed by the CPU through execution of software commands. In this mode, the user ROM and data flash can be programmed without using a ROM programmer with the microcomputer mounted on a system board.

The programming and block erase commands should be executed only in each block area of the user ROM and data flash.

The user programming mode provides the erase/write 0 mode (EW0 mode).

3.1 EW0 Mode

Setting the FMCMDEN bit in Flash memory control register 1 (FLMCR1) to 1 shifts the flash memory into the user programming mode, in which commands can be accepted.

Programming and erasure are controlled through software commands. The flash memory state after programming or erasure can be checked through FLMSTR or the status register.

3.2 Software Commands

The following describes the software commands. A command or data should be read or written in 16-bit units at an even address in the user ROM or data flash area. When a command code is written, the lower eight bits (D7 to D0) are ignored.

Table 3 List of Software Commands

	First Bus Cycle			Second Bus Cycle			Third Bus Cycle		
Software Command	**Mode**	**Address**	**Data (D15 to D0)**	**Mode**	**Address**	**Data (D15 to D0)**	**Mode**	**Address**	**Data (D15 to D0)**
Read array	Write	×	H'FFxx						
Read status register	Write	×	H'70xx	Read	×	SRD			
Clear status register	Write	×	H'50xx						
Program	Write	WA0	H'41xx	Write	WA0	WD0	Write	WA1	WD1
Block erase	Write	×	H'20xx	Write	BA	H'D0xx			
Block blank check	Write	×	H'25xx	Write	BA	H'D0xx			

[Legend]

SRD: Status register data (D7 to D0)
WA0: Address to write the lower word (the address for the first bus cycle must be the same even address as that for the second bus cycle).
WA1: Address to write the upper word
WD0: Lower word of write data (16 bits)
WD1: Upper word of write data (16 bits)
BA: Start address of the block (see table 4, List of Start Addresses of Blocks)
×: A desired even address in program ROM, or data flash.
xx: Lower eight bits of command code (ignored)

Table 4 List of Start Addresses of Blocks

On-chip ROM	**User ROM**				**Data Flash**	
Block	Block 0	Block 1	Block 2	Block 3	Block A	Block B
Address specified to erase the block	H'000000	H'010000	H'020000	H'030000	H'F00000	H'F01000

3.2.1 Read Array

This command reads the flash memory.

Write H'FFxx in the first bus cycle to shift the flash memory into the read array mode. Specify the target read address in the next bus cycle, and data is read from the address in 16-bit units.

As the flash memory stays in the read array mode until another command is issued, multiple addresses can be read in sequence.

3.2.2 Read Status Register

This command reads the status register.

Write H'70xx in the first bus cycle, and the status register can be read in the second bus cycle (see section 3.2.6, Status Register). Specify an even address in the user ROM, or data flash to read the status register.

3.2.3 Clear Status Register

This command clears the status register.

Write H'50xx in the first bus cycle, and the FMERSF and FMPRSF bits in FLMSTR are cleared to 0.

3.2.4 Program

This command writes data to the flash memory in 2-word units.

Write H'41xx in the first bus cycle and write data to the target address in the second and third bus cycles; the flash memory starts automatic writing (programming and verifying data). The address value specified in the first bus cycle should be the same even address as that specified in the second bus cycle.

Completion of automatic writing can be checked through the FMRDY bit in FLMSTR. The FMRDY bit is 0 (busy) during automatic writing and becomes 1 (ready) when writing is completed.

After automatic writing is completed, the result can be checked through the FMPRSF bit in FMRSTR (see section 3.2.6, Status Register).

Once an address is programmed, no additional data can be written to the address.

In the EW0 mode, the read status register mode is entered as soon as automatic writing starts, and the status register can be read. The SR7 bit in the status register becomes 0 when automatic writing starts and returns to 1 when writing is completed. In this case, the flash memory stays in the read status register mode until a read array command is issued. After automatic writing is completed, the result of writing can be checked by reading the status register.

3.2.5 Block Erase

Write H'20xx in the first bus cycle and H'D0xx to the start address (see table 4, List of Start Addresses of Blocks) of the target block in the second cycle; automatic erasure (erasing data and verifying the erased status) starts in the specified block.

Completion of automatic erasure can be checked through the FMRDY bit in FLMSTR.

The FMRDY bit is 0 (busy) during automatic erasure and becomes 1 (ready) when erasure is completed.

After automatic erasure is completed, the result can be checked through the FMERSF bit in FLMSTR (see section 3.2.6, Status Register).

In the EW0 mode, the read status register mode is entered as soon as automatic erasure starts, and the status register can be read. The SR7 bit in the status register becomes 0 when automatic erasure starts and returns to 1 when erasure is completed. In this case, the flash memory stays in the read status register mode until a read array command is issued. If an erase error occurs, repeat a sequence of the clear status register command to block erase command at least three times until no erase error occurs.

3.2.6 Status Register

The status register indicates the state of flash memory operation and whether erasure or programming has ended successfully or with an error. The status register contents can be read through the FMRDY, FMPRSF, and FMERSF bits in FLMSTR.

Table 5 shows the status register.

In the EW0 mode, the status register can be read with the following timing.

- When a read status register command is issued and then an even address in the user ROM or data flash is read
- When a program command, a block erase command, or a block blank check command is issued and then an even address in the user ROM or data flash is read before a read array command is issued

Table 5 Status Register

Bits in Status Register	Bits in FMLSTR	Status Name	Status		Value after Reset
			0	1	
SR0 (D0)	—	Reserved	—	—	—
SR1 (D1)	—	Reserved	—	—	—
SR2 (D2)	—	Reserved	—	—	—
SR3 (D3)	—	Reserved	—	—	—
SR4 (D4)	FMPRSF	Programming status	Completed successfully	Ended with error	0
SR5 (D5)	FMERSF	Erase status	Completed successfully	Ended with error	0
SR6 (D6)	—	Reserved	—	—	—
SR7 (D7)	FMRDY	Sequencer status	Busy	Ready	1

[Legend]

D0 to D7: Data bus from which the bit is read when a read status register command is issued.

Note: The FMERSF (SR5) and FMPRSF (SR4) bits are cleared to 0 by a clear status register command. When the FMERSF (SR5) or FMPRSF (SR4) bit is 1, the program, block erase, and block blank check commands are not accepted.

4. Description of Operation

Erasure and programming are started by pushing switches according to the following procedure.

(1) After start-up, the programming control program is copied to the on-chip RAM, and the data for writing (256 bytes of data stored at addresses from H'00 to H'FF) is created in the on-chip RAM.
(2) Pushing SW1 starts erasure of block 1, after which a blank-check command is executed to check if all addresses within block 1 contain H'FF.
(3)Pushing SW2 starts writing of the 256 bytes of data to all 256-byte ranges within block 1, and after this is completed, verification of whether or not the 256 bytes of data have been written to all 256-byte ranges within block 1 proceeds.
(4) The application returns to step (2).
If an error is found in step (2) or (3), an LED is turned on and execution goes into an infinite loop.

Figure 2 shows the operation described in this application note.

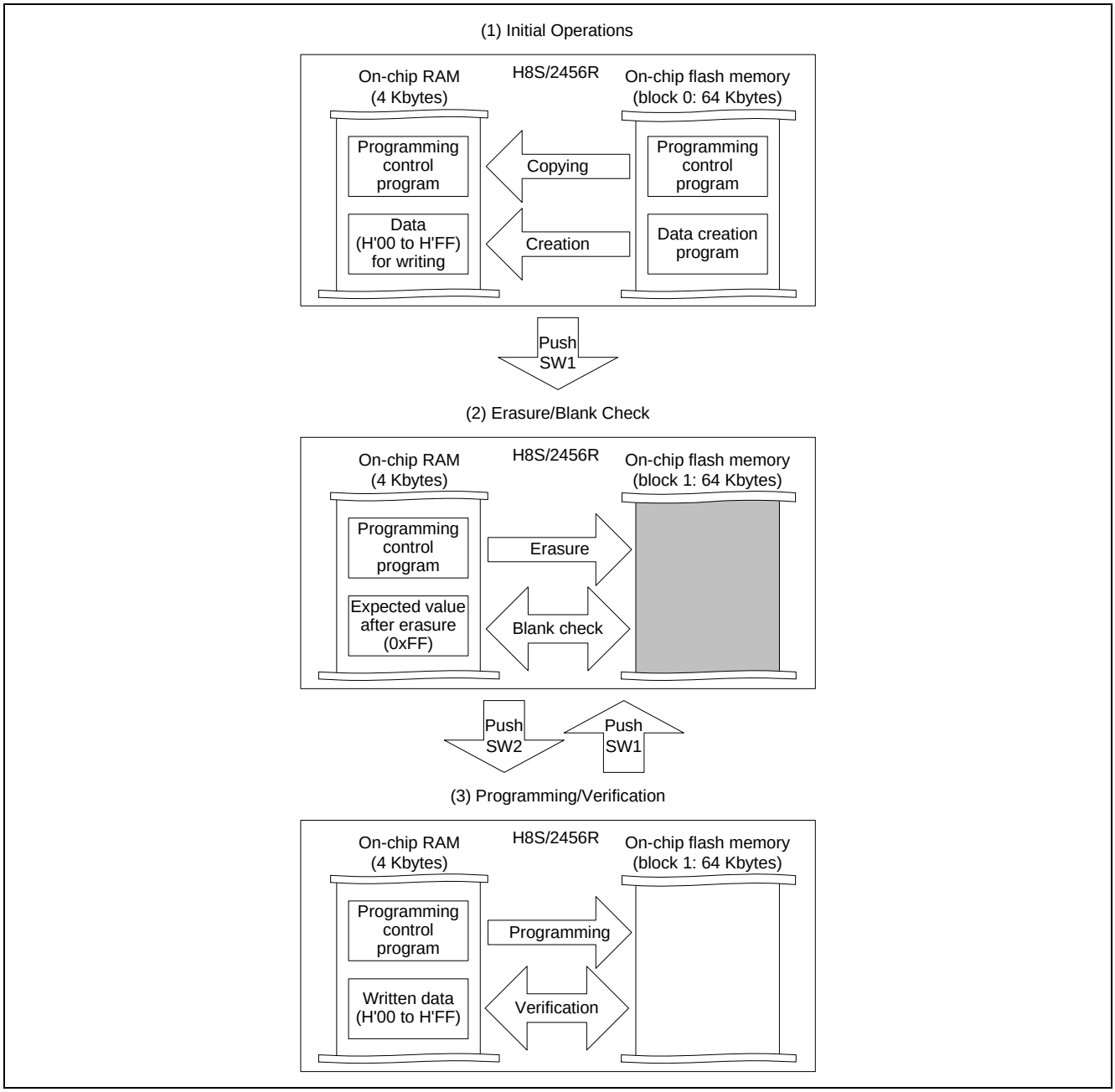


Figure 2 Description of Operation

In this application note, the ROM-support function of the optimizing linkage editor and the initialization routine (_INYSCT function) for sections that is provided as a standard library function are used for relocation in the on-chip RAM and copying of the programming control program provided in the on-chip flash memory to the on-chip RAM.

The ROM-support function is an option of the optimizing linkage editor that reserves ROM and RAM areas in the initialized data area and relocates symbols defined in the ROM section at the specified address range in the RAM section. Note that this function only handles relocation and does not copy the program. Therefore, the programming control program must be separately copied from the on-chip ROM to the on-chip RAM.

The _INITSCT function initializes the non-initialized data section to 0, and also copies the initialized data from on-chip ROM in the initialized data section to the on-chip RAM.

In this application note, the _INITSCT function is used to copy the programming control program from the on-chip ROM to the on-chip RAM (relocation to which is handled by the ROM-support function).

Note that, in order to copy the program from the on-chip ROM to the on-chip RAM by using the _INITSCT function, the source address, the amount of source data, and the destination address must be added to the tables for section initialization (DTBL, BTBL).

High-performance Embedded Workshop (HEW) automatically generates the tables for section initialization (DTBL, BTBL) in the dbsct.c file as part of a project for creating executable programs that include an initialization routine.

The following shows an example of addition to the tables for section initialization.

```
#pragma section $DSEC
static const struct {
  _UBYTE *rom_s;       /* Start address of the initialized data section in ROM */
  _UBYTE *rom_e;       /* End address of the initialized data section in ROM   */
  _UBYTE *ram_s;       /* Start address of the initialized data section in RAM */
}DTBL[]= {
  {__sectop("D"), __secend("D"), __sectop("R")},
  {__sectop("PFWRMAIN"), __secend("PFWRMAIN"), __sectop("RFWRMAIN")}, (*)
// {__sectop("$ABS8D"), __secend("$ABS8D"), __sectop("$ABS8R")},
// {__sectop("$ABS16D") , __secend("$ABS16D") , __sectop("$ABS16R") }
};
#pragma section $BSEC
static const struct {
  _UBYTE *b_s;         /* Start address of non-initialized data section */
  _UBYTE *b_e;         /* End address of non-initialized data section */
}BTBL[]= {
  {__sectop("B"), __secend("B")},
// {__sectop("$ABS8B"), __secend("$ABS8B")},
// {__sectop("$ABS16B"), __secend("$ABS16B")}
};
```

***:** Added in this application note.

For details on the ROM-support function and the _INITSCT function, see the development environment manual indicated in section 6, Documents for Reference.

5. Description of Software

5.1 Symbolic Constants

Table 6 List of Symbolic Constants

Constant Name	Setting	Description	Used by Functions
BLOCK0	0x01	Block 0 of user ROM	(Not used)
BLOCK1	0x02	Block 1 of user ROM	main
BLOCK2	0x04	Block 2 of user ROM	(Not used)
BLOCK3	0x08	Block 3 of user ROM	(Not used)
BLOCKA	0x10	Block A of data flash	(Not used)
BLOCKB	0x20	Block B of data flash	(Not used)
BLOCK_MAX	6	Number of blocks	flash_erase
PWR_ON_RST	0x00	Initialization is complete.	main set_led
ERASING	0x01	Erasure in progress	main set_led
ERASE_END	0x02	Erasure is complete.	main set_led
ERASE_ERROR	0x04	An erase error was found.	main set_led
ERASE_COMPARE_ERROR	0x05	A comparison-check error was found after erasure.	main set_led
PROGRAMING	0x06	Programming in progress	main set_led
PROGRAM_END	0x07	Programming is complete.	main set_led
PROGRAM_ERROR	0x09	A programming error was found.	main set_led
PROGRAM_COMPARE_ERROR	0x0A	A comparison-check error was found after programming	main set_led
LED0	P6.DR.BIT.B3	LED0 of RSK	set_led
LED1	P6.DR.BIT.B5	LED1 of RSK	set_led
LED2	PJ.DR.BIT.B0	LED2 of RSK	set_led
LED3	PJ.DR.BIT.B1	LED3 of RSK	set_led
LED_ON	0	LED is turned on.	set_led
LED_OFF	1	LED is turned off.	set_led
MCUFLASH_OK	0	Erasure/programming was successfully completed.	main flash_erase flash_program block_erase program_256_bytes full_chk
MCUFLASH_NG	–1	Erasure/programming ended with an error.	main flash_erase flash_program block_erase program_256_bytes full_chk

5.2 ROM Variables

Table 7 List of ROM Variables

Type	Variable Name	Setting	Description	Used by Functions
const unsigned long	erase_address[6]	0x00000000ul, 0x00010000ul, 0x00020000ul, 0x00030000ul, 0x00F00000ul, 0x00F01000ul	Addresses for specifying erasure of the on-chip flash memory	flash_erase

5.3 RAM Variables

Table 8 List of RAM Variables

Type	Variable Name	Description	Used by Functions
unsigned char	write_data[256]	Data to be written to the on-chip flash memory	main mcu_init
volatile_ evenacce ss unsigned short *	wp	Address at the time the command is issued	block_erase program_256_bytes full_chk
unsigned char	Imask	Saves the interrupt mask level before the transition to the user programming mode	enter_EW0mode exit_EW0mode

5.4 List of Functions

Table 9 List of Functions

Function Name	Descriptions
PowerOn_Reset	• Initial settings function Initializes status pointer (SP), sets interrupt mask bits, sets uninitialized/initialized data, and calls main function.
main	• Main function Calls the mcu_init function and executes erasure and programming.
mcu_init	• MCU initialization function Initializes registers, and generates the data for writing.
led_set	• LED controlling function Turns on, off, or blinks LEDs
flash_erase	• Erasure function for the on-chip flash memory Erases the specified block of the on-chip flash memory.
flash_program_256bytes	• Programming function of the on-chip flash memory Writes the specified data to the specified address range within the flash memory.
enter_EW0mode	• User programming mode enabling function Saves the interrupt mask level and enables the user programming mode.
exit_EW0mode	• User programming mode disabling function Restores the interrupt mask level and disables the user programming mode.
block_erase	• Block erasure function Erases the specified block.
program_256_bytes	• 256-byte programming function Executes programming to the range from the specified address.
full_chk	• Full status checking function Runs a status check to determine whether erasure was or was not successfully completed.

5.5 Functions

5.5.1 PowerON_Reset

(1) Functional overview

The PowerON_Reset function initializes the status pointer (SP) and uses embedded functions and standard library functions to set interrupt mask bits and set uninitialized/initialized data. Then it calls the main function.

(2) Arguments

None

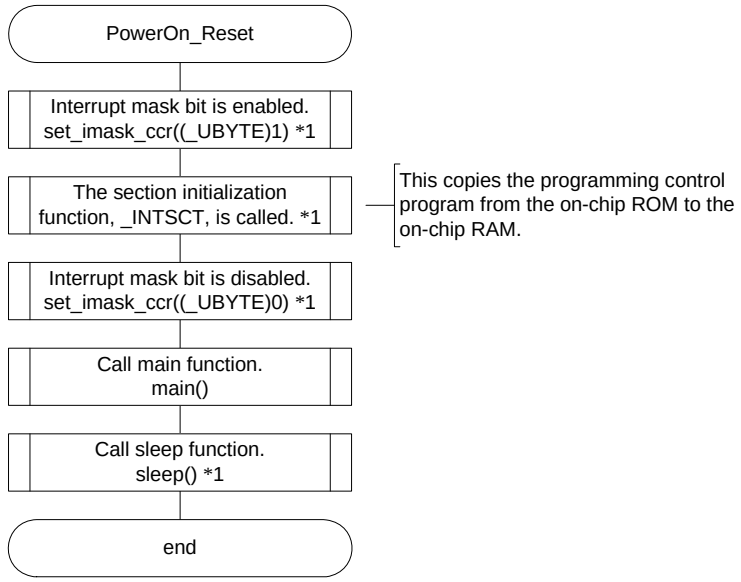
(3) Returned values

None

(4) Description of internal I/O registers used

None

(5) Flowchart



Note: *1 set_imask_ccr function: Intrinsic function
_INITSCT function: Standard library function
sleep function: Intrinsic function

For details, please refer to the *Development Environment Manual* in the "6. Documents for Reference".

Figure 3 Flowchart (PowerON_Reset)

5.5.2 main Function

(1) Functional overview

The main function calls the mcu_init function and executes erasure and programming.

(2) Arguments

None

(3) Returned values

None

(4) Description of internal I/O registers used

The internal I/O registers used by this function are shown below.

Note that the setting values shown are those used in this application note and differ from the initial values.

- Port 6 Register (PORT6) Number of bits: 8 Address: H'FFFFFF55

Bit	Bit Name	Setting	R/W	Descriptions
1	P61	—	R/W	Reading this register reads out the value of P6DR for bits where the corresponding bits in P6DDR are set and the pin states for bits where the corresponding bits in P6DR are not set (i.e. where the bits in P6DR are clear).

- Port 8 Register (PORT8) Number of bits: 8 Address: H'FFFFFF57

Bit	Bit Name	Setting	R/W	Descriptions
1	P81	—	R/W	

(5) Flowchart

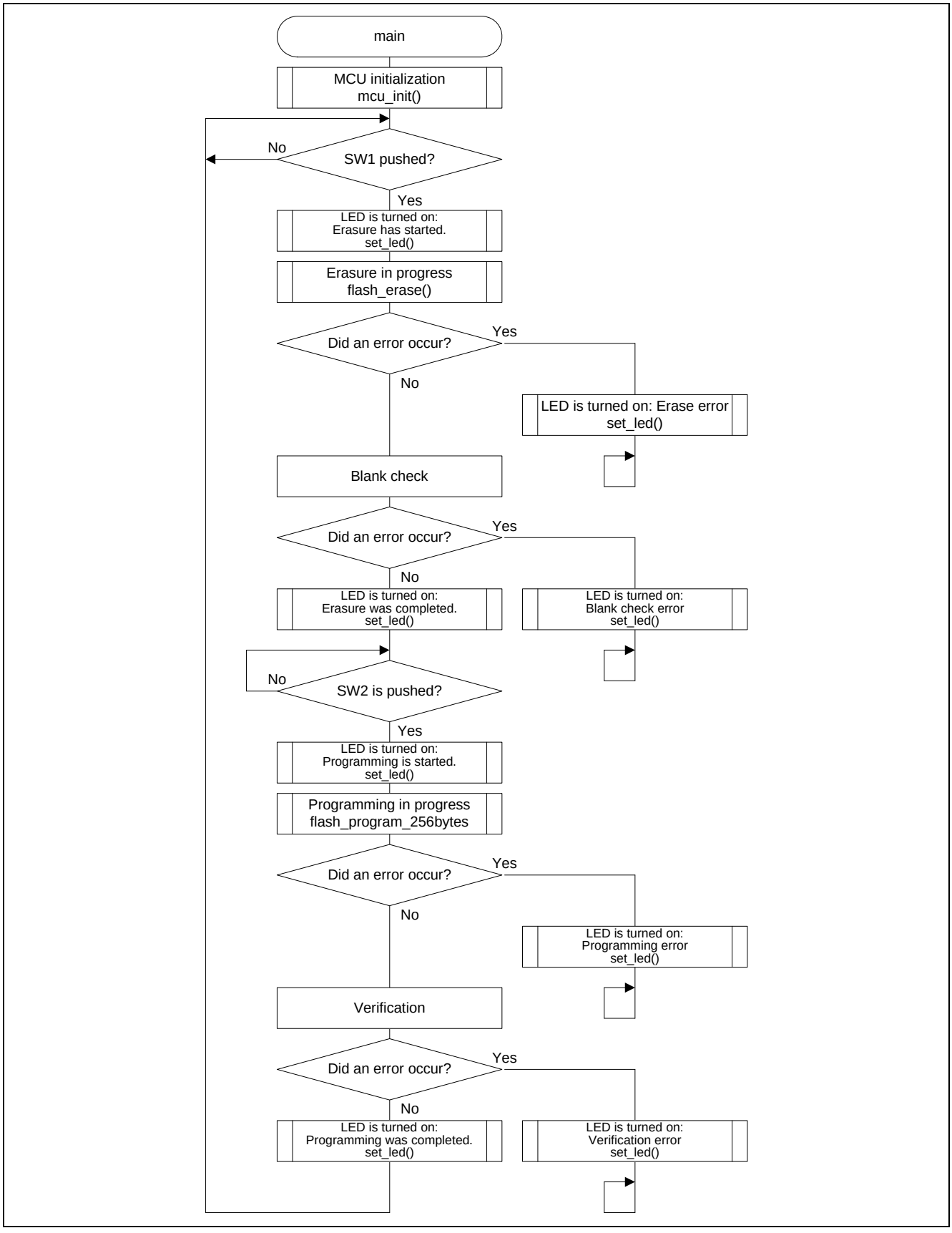


Figure 4 Flowchart (main)

5.5.3 mcu_init Function

(1) Functional overview

The mcu_init function sets the clock, initializes registers, and generates the data for writing.

(2) Arguments

None

(3) Returned values

None

(4) Description of internal I/O registers used

The internal I/O registers used by this function are shown below.

Note that the setting values shown are those used in this application note and differ from the initial values.

- Mode Control Register (MDCR) Number of bits: 8 Address: H'FFFFFF3E

Bit	Bit Name	Setting	R/W	Descriptions
2	MDS2	—*	R	Mode Select 2 to 0
1	MDS1	—*	R	These bits indicate the input levels at mode pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to pins MD2 to MD0, respectively. These bits are read-only bits and so they cannot be modified. The input levels of the MD2 to MD0 pins are latched into these bits when MDCR is read. These latches are canceled by a reset.
0	MDS0	—*	R	

Note: * Determined by the settings of pins MD2 to MD0.

- Port 6 Data Direction Register (P6DDR) Number of bits: 8 Address: H'FFFFFE25

Bit	Bit Name	Setting	R/W	Descriptions
5	P65DDR	1	W	When a pin function is specified as a general purpose I/O, setting this bit to 1 makes the corresponding pin an output port, while clearing this bit to 0 makes the corresponding pin an input port.
4	P64DDR	0	W	
3	P63DDR	1	W	
2	P62DDR	0	W	
1	P61DDR	0	W	
0	P60DDR	0	W	

- Port J Data Direction Register (PJDDR) Number of bits: 8 Address: H'FFFFFF75

Bit	Bit Name	Setting	R/W	Descriptions
1	PJ1DDR	1	W	When a pin function is specified as a general purpose I/O, setting this bit to 1 makes the corresponding pin an output port, while clearing this bit to 0 makes the corresponding pin an input port.
0	PJ0DDR	1	W	

- Module Stop Control Registers H (MSTPCRH) Number of bits: 8 Address: H'FFFFFF42

Bit	Bit Name	Setting	R/W	Descriptions
14	MSTP14	1	R/W	EXDMA controller (EXDMAC)*
13	MSTP13	1	R/W	DMA controller (DMAC)
12	MSTP12	1	R/W	Data transfer controller (DTC)

Note: * Not supported by the H8S/2454 Group.

- System Clock Control Register (SCKCR) Number of bits: 8 Address: H'FFFFFF3B

Bit	Bit Name	Setting	R/W	Descriptions
3	STCS	1	R/W	Frequency Multiplication Factor Switching Mode Select Selects the operation when the PLLCR register setting is changed. 1: Specified multiplication factor is valid immediately after STC1 and STC0 bits are rewritten.

- PLL Control Register (PLLCR) Number of bits: 8 Address: H'FFFFFF45

Bit	Bit Name	Setting	R/W	Descriptions
1	STC1	0	R/W	Frequency Multiplication Factor for System-Clock PLL Circuit and System Clock Divider Setting The STC bits specify the frequency multiplication factor and dividing ratio with respect to the oscillator frequency. 01:× 2
0	STC0	1	R/W	

- Interrupt Control Register (INTCR) Number of bits: 8 Address: H'FFFFF31

Bit	Bit Name	Setting	R/W	Descriptions
5	INTM1	1	R/W	Interrupt Control Select Mode 1 and 0 These bits select either of two interrupt control modes for the interrupt controller. 10: Interrupt control mode 2 Interrupts are controlled by bits I2 to I0, and IPR.
4	INTM0	0	R/W	

- System Control Register (SYSCR) Number of bits: 8 Address: H'FFFFF3D

Bit	Bit Name	Setting	R/W	Descriptions
3	FLSHE	1	R/W	Flash Memory Control Register Enable Controls CPU access to the flash memory control registers (FLMCR1, FLMDBPR, and FLMSTR). If this bit is set to 1, the flash memory control registers can be read from and written to. If this bit is cleared to 0, the flash memory control registers are not selected. At this time, the contents of the flash memory control registers are retained. 0 should be written to this bit in LSIs other than the flash memory version. 1: Flash memory control registers are selected for addresses H'FFFEB0 to H'FFFEB3

(5) Flowchart

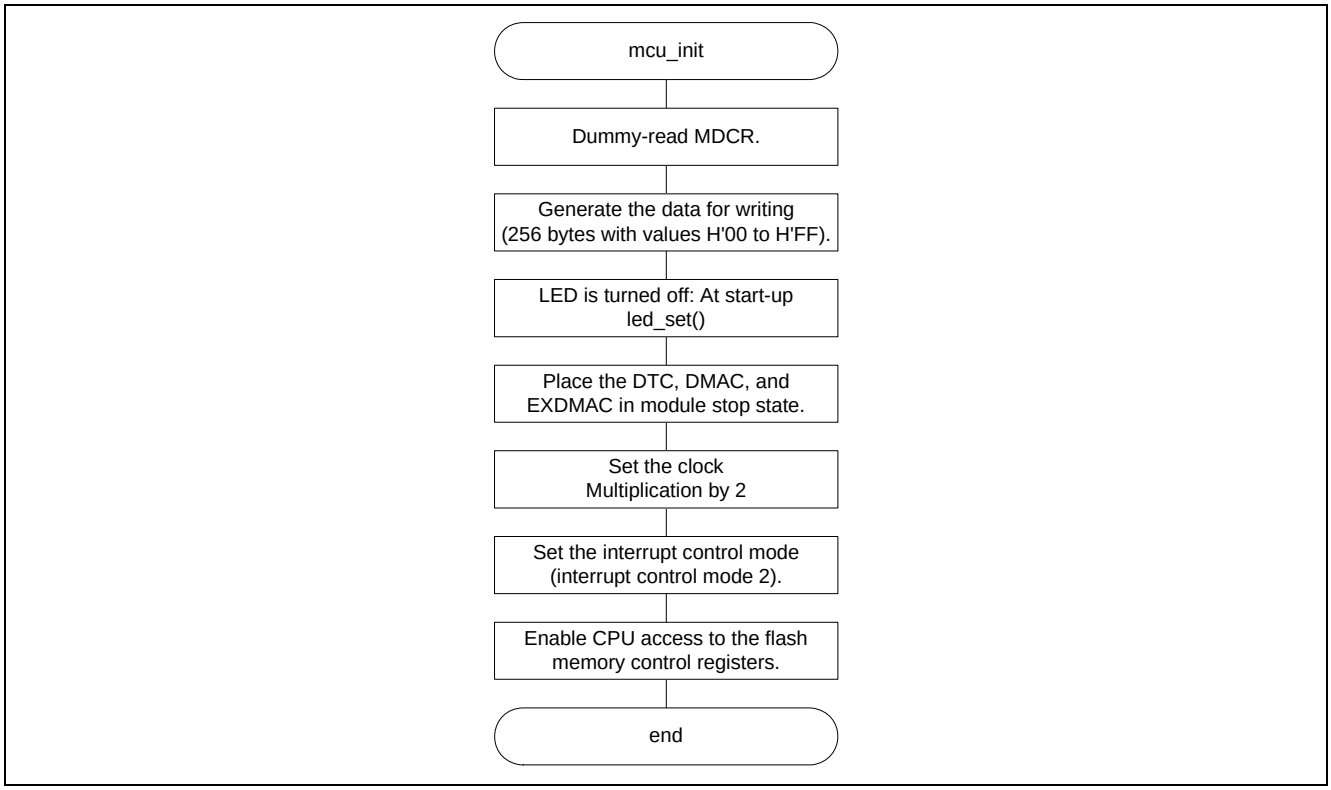


Figure 5 Flowchart (mcu_init)

5.5.4 led_set Function

(1) Functional overview

The led_set function turns the LEDs on or off according to the specified argument.

(2) Arguments

Argument	Type	Argument Name	Argument Value	Description
First argument	unsigned char	led_mode	PWR_ON_RST ERASING ERASE_END ERASE_ERROR ERASE_COMPARE_ERROR PROGRAMING PROGRAM_END PROGRAM_ERROR PROGRAM_COMPARE_ERROR	Turns the LEDs on or off according to the specified argument. For the specifications for LED operations, refer to table 1, Specifications for LED Operations. For details on the argument value, refer to table 6, List of Symbol Constants.

(3) Returned values

None

(4) Description of internal I/O registers used

None

(5) Flowchart

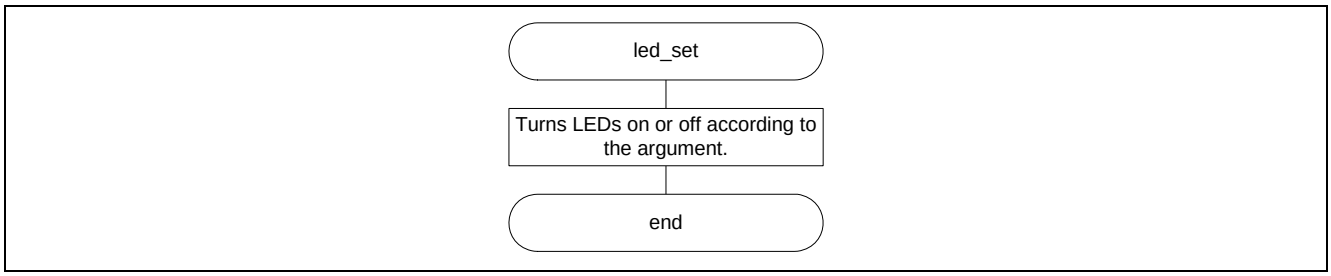


Figure 6 Flowchart (led_set)

5.5.5 flash_erase Function

(1) Functional overview

The flash_erase function erases the specified block of the on-chip flash memory.

(2) Arguments

Argument	Type	Argument Name	Argument Value	Description
First argument	unsigned char	block	BLOCK0 BLOCK1 BLOCK2 BLOCK3 BLOCKA BLOCKB	Erases the block specified by the argument. Specifying more than one block is possible.

(3) Returned values

Type	Returned Value Name	Returned Value	Description
char	status	MCUFLASH_OK	MCUFLASH_OK: Erasure was successfully completed.
		MCUFLASH_NG	MCUFLASH_NG: Erasure ended with an error.

(4) Description of internal I/O registers used

None

(5) Flowchart

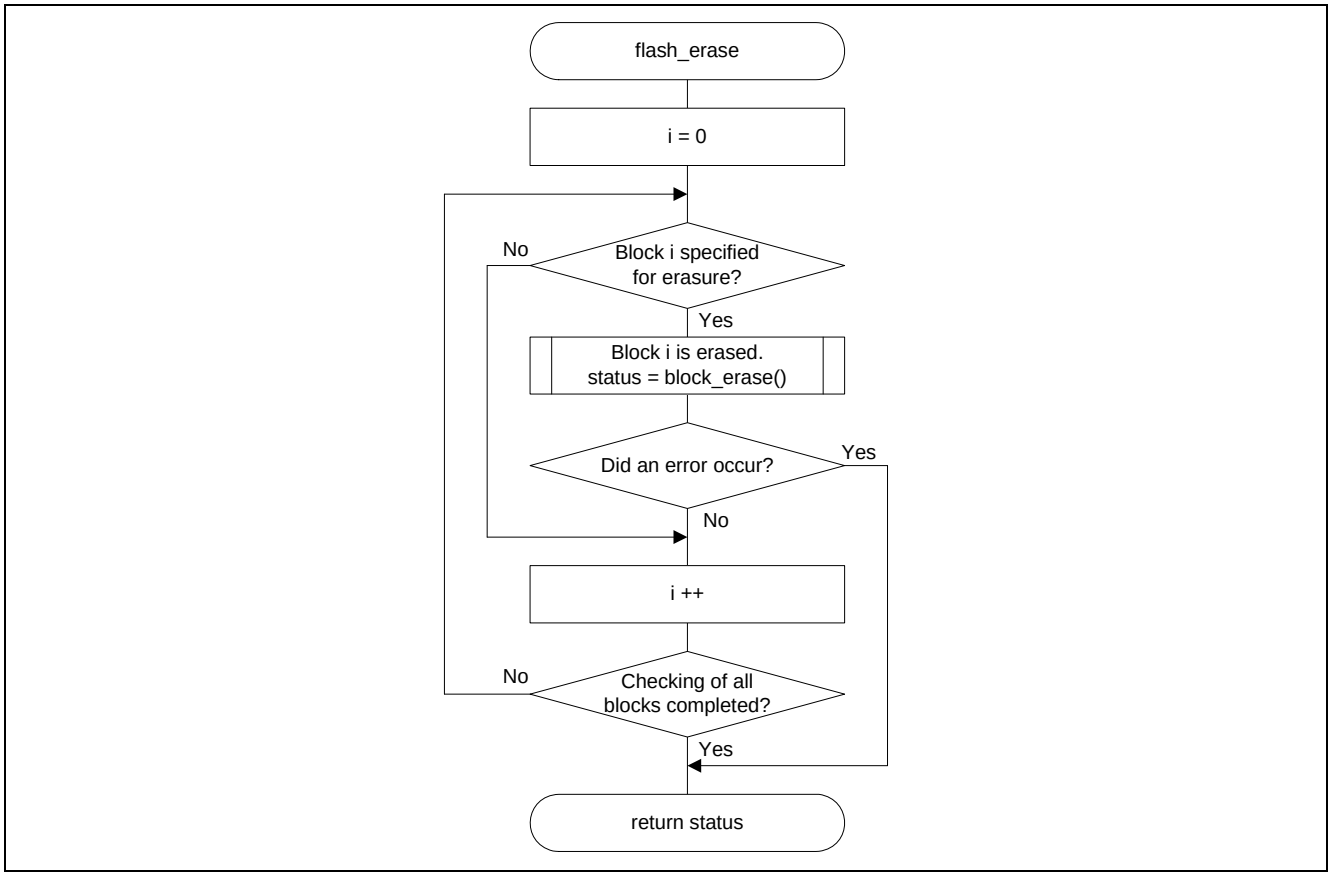


Figure 7 Flowchart (flash_erase)

5.5.6 flash_program_256bytes Function

(1) Functional overview

The flash_program_256bytes function writes the specified data to the specified address range within the flash memory.

(2) Arguments

Argument	Type	Argument Name	Argument Value	Description
First argument	unsigned long	addr	Address to which the data is to be written	Specifies the address from which the data are to be written.
Second argument	unsigned short *	data	Pointer to the data for writing	Specifies the pointer to the area where the data for writing are stored.

(3) Returned values

Type	Returned Value Name	Returned Value	Description
char	status	MCUFLASH_OK	MCUFLASH_OK: Programming was successfully completed.
		MCUFLASH_NG	MCUFLASH_NG: Programming ended with an error.

(4) Description of internal I/O registers used

None

(5) Flowchart

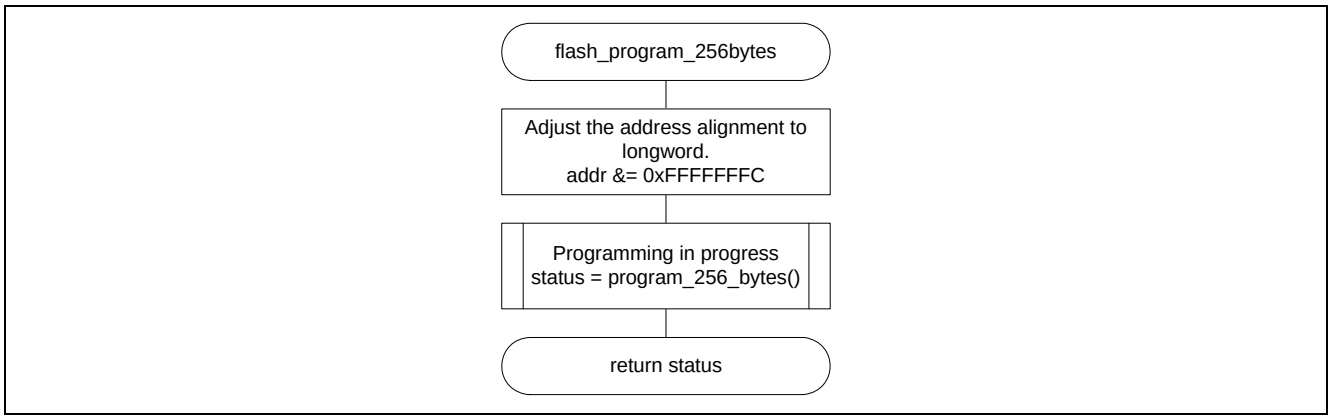


Figure 8 Flowchart (flash_program_256bytes)

5.5.7 enter_EW0mode Function

(1) Functional overview

The enter_EW0mode function saves the interrupt mask level (EXR) and enables the user programming mode.

(2) Arguments

None

(3) Returned values

None

(4) Description of internal I/O registers used

The internal I/O registers used by this function are shown below.

Note that the setting values shown are those used in this application note and differ from the initial values.

- Flash Memory Control Register 1 (FLMCR1) Number of bits: 8 Address: H'FFFFFEB0

Bit	Bit Name	Setting	R/W	Descriptions
6	CBIDB	0	R/W	CPU Programming Mode Select Setting this bit to 0 (CPU programming mode) enables command acceptance. 0: CPU programming mode enabled
0	FMCMDEN	1	R/W	Flash Memory Software Command Enable Setting this bit to 1 (CPU programming mode) enables command acceptance. 1: Flash memory software commands enabled To set this bit to 1, be sure to write 0 and then write 1 in a row.

(5) Flowchart

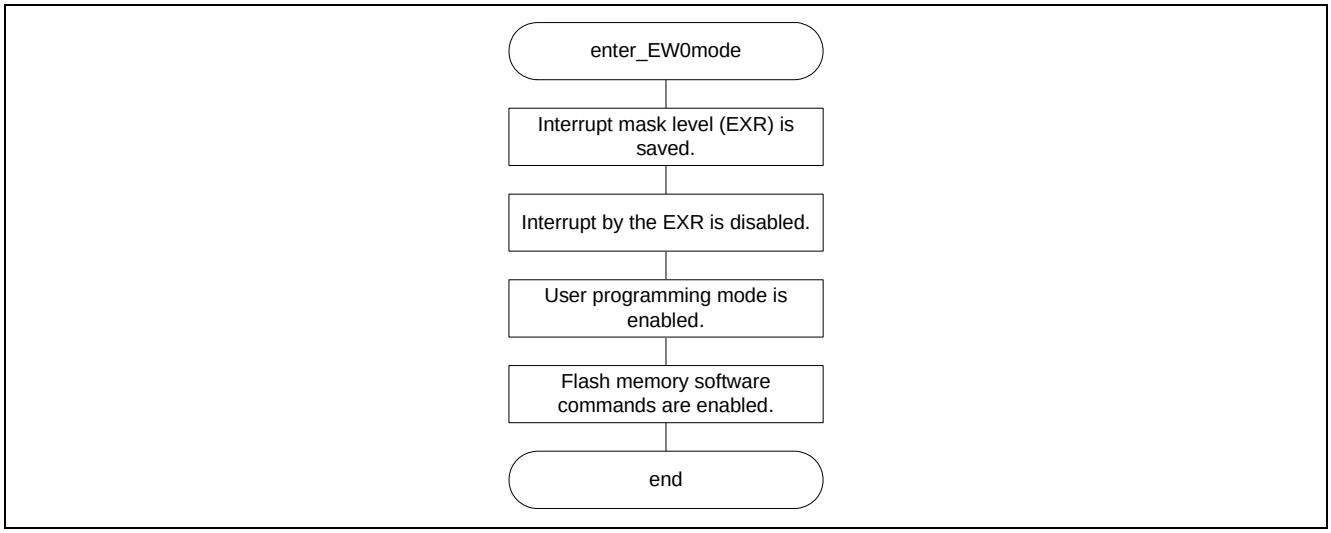


Figure 9 Flowchart (enter_EW0mode)

5.5.8 exit_EW0mode Function

(1) Functional overview

The exit_EW0mode function restores the interrupt mask level (EXR) and disables the user programming mode.

(2) Arguments

None

(3) Returned values

None

(4) Description of internal I/O registers used

The internal I/O registers used by this function are shown below.

Note that the setting values shown are those used in this application note and differ from the initial values.

- Flash Memory Control Register 1 (FLMCR1) Number of bits: 8 Address: H'FFFFFEB0

Bit	Bit Name	Setting	R/W	Descriptions
6	CBIDB	1	R/W	CPU Programming Mode Select Setting this bit to 0 (CPU programming mode) enables command acceptance. 1: CPU programming mode disabled
0	FMCMDEN	0	R/W	Flash Memory Software Command Enable Setting this bit to 1 (CPU programming mode) enables command acceptance. 0: Flash memory software commands disabled To set this bit to 1, be sure to write 0 and then write 1 in a row.

(5) Flowchart

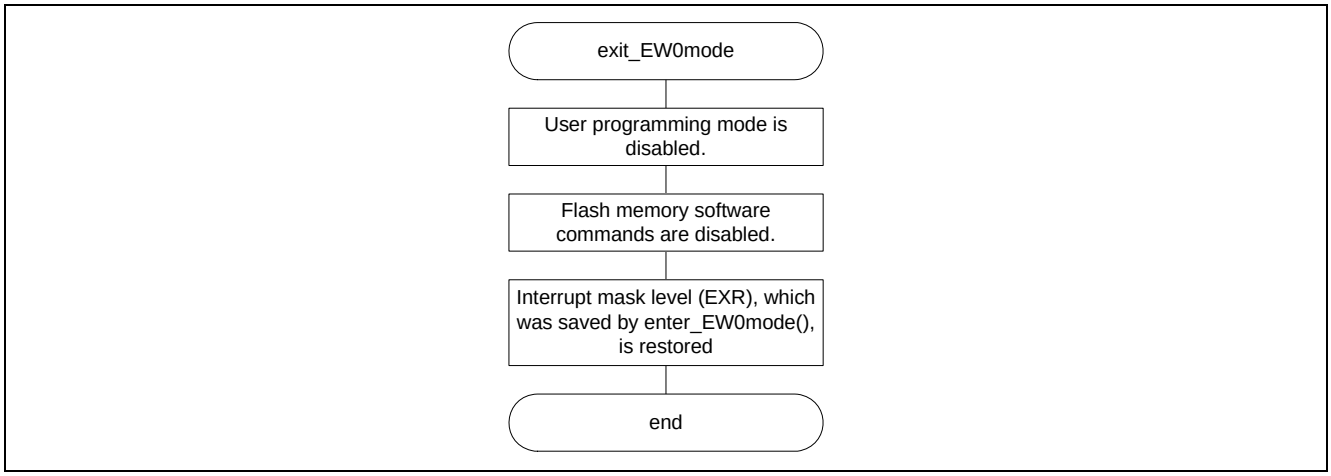


Figure 10 Flowchart (exit_EW0mode)

5.5.9 block_erase Function

(1) Functional overview

The block_erase function erases the specified block (see table 4, List of Start Addresses of Blocks).

(2) Arguments

Argument	Type	Argument Name	Argument Value	Description
First argument	unsigned long	addr	Address of the block to be erased	Specifies the address where the block to be erased starts.

(3) Returned values

Type	Returned Value Name	Returned Value	Description
char	status	MCUFLASH_OK	MCUFLASH_OK: Erasure was successfully completed.
		MCUFLASH_NG	MCUFLASH_NG: Erasure ended with an error.

(4) Description of internal I/O registers used

The internal I/O registers used by this function are shown below.

Note that the setting values shown are those used in this application note and differ from the initial values.

- Flash Memory Status Register (FLMSTR) Number of bits: 8 Address: H'FFFFFEB3

Bit	Bit Name	Setting	R/W	Descriptions
0	FMRDY	—	R	Flash Memory Ready/Busy Status 0: Busy (Interrupt processing or erasure is in progress.) 1: Ready

(5) Flowchart

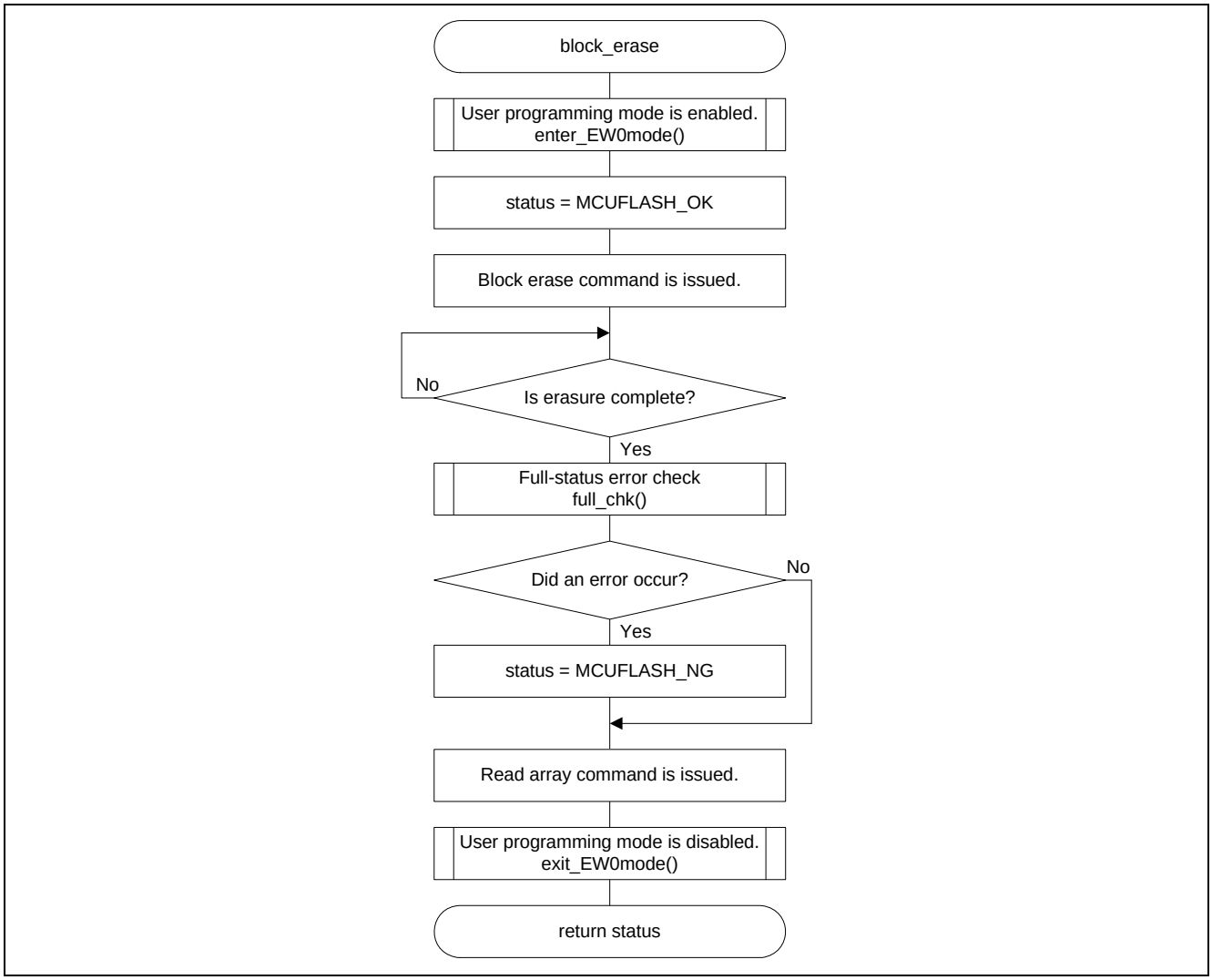


Figure 11 Flowchart (block_erase)

5.5.10 program_256_bytes Function

(1) Functional overview

The program_256_bytes function executes programming to the range from the specified address.

(2) Arguments

Argument	Type	Argument Name	Argument Value	Description
First argument	unsigned long	addr	Address to which the data is to be written	Specifies the address from which the data are to be written.
Second argument	unsigned short *	data	Pointer to the data for writing	Specifies the pointer to the area where the data for writing are stored.

(3) Returned values

Type	Returned Value Name	Returned Value	Description
char	status	MCUFLASH_OK	MCUFLASH_OK: Erasure was successfully completed.
		MCUFLASH_NG	MCUFLASH_NG: Erasure ended with an error.

(4) Description of internal I/O registers used

The internal I/O registers used by this function are shown below.

Note that the setting values shown are those used in this application note and differ from the initial values.

- Flash Memory Status Register (FLMSTR) Number of bits: 8 Address: H'FFFFFEB3

Bit	Bit Name	Setting	R/W	Descriptions
0	FMRDY	—	R	Flash Memory Ready/Busy Status 0: Busy (Interrupt processing or erasure is in progress.) 1: Ready

(5) Flowchart

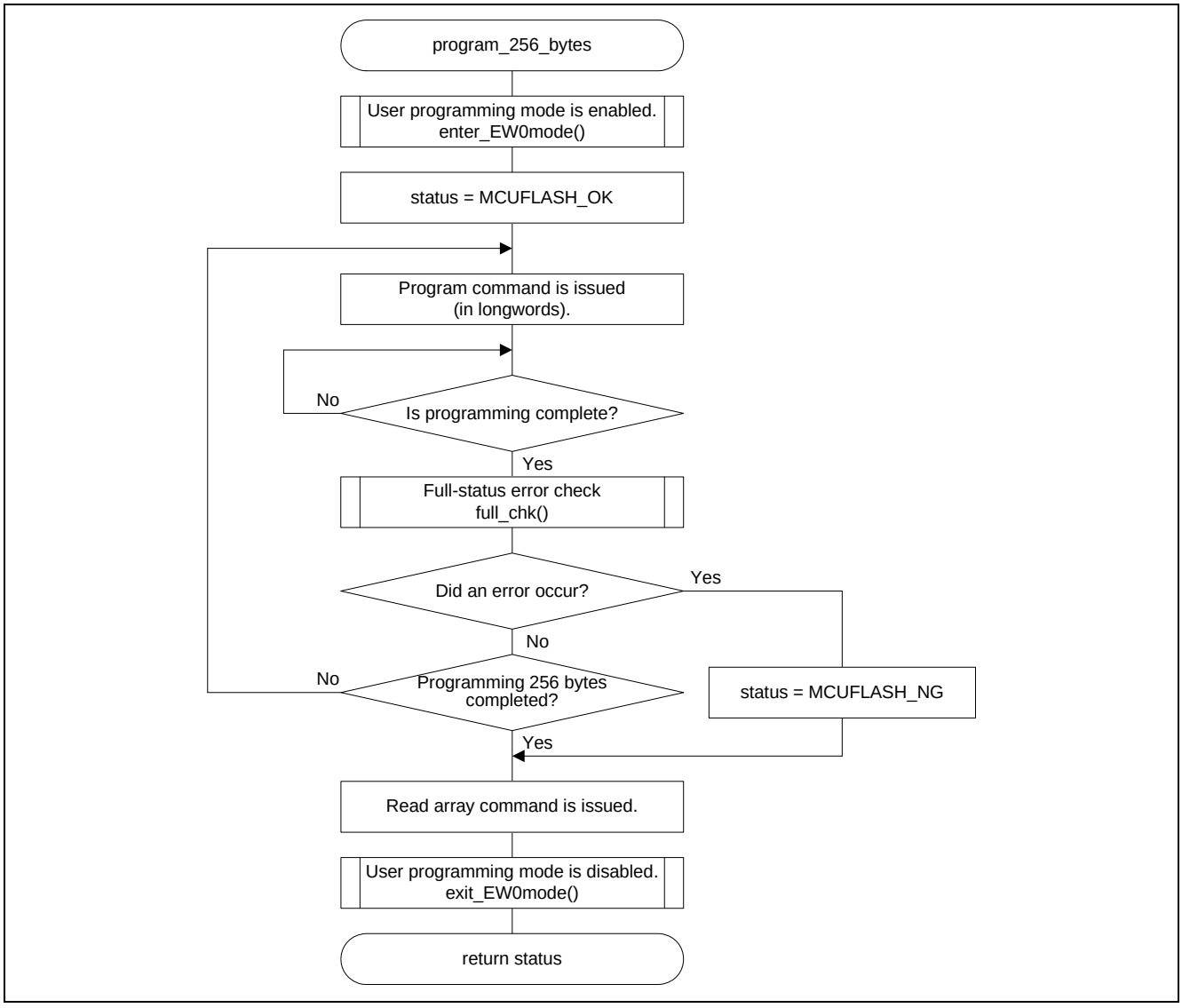


Figure 12 Flowchart (program_256_bytes)

5.5.11 full_chk Function

(1) Functional overview

The full_chk function runs a status check to determine whether erasure was or was not successfully completed.

(2) Arguments

None

(3) Returned values

Type	Returned Value Name	Returned Value	Description
char	status	MCUFLASH_OK	MCUFLASH_OK: Erasure was successfully completed.
		MCUFLASH_NG	MCUFLASH_NG: Erasure ended with an error.

(4) Description of internal I/O registers used

The internal I/O registers used by this function are shown below.

Note that the setting values shown are those used in this application note and differ from the initial values.

- Flash Memory Status Register (FLMSTR) Number of bits: 8 Address: H'FFFFFEB3

Bit	Bit Name	Setting	R/W	Descriptions
5	FMERSF	—	R	Erase or Blank Check Status Flag 0: Successfully completed 1: Ended with an error
3	FMPRSF	—	R	Program Status Flag 0: Successfully completed 1: Ended with an error

(5) Flowchart

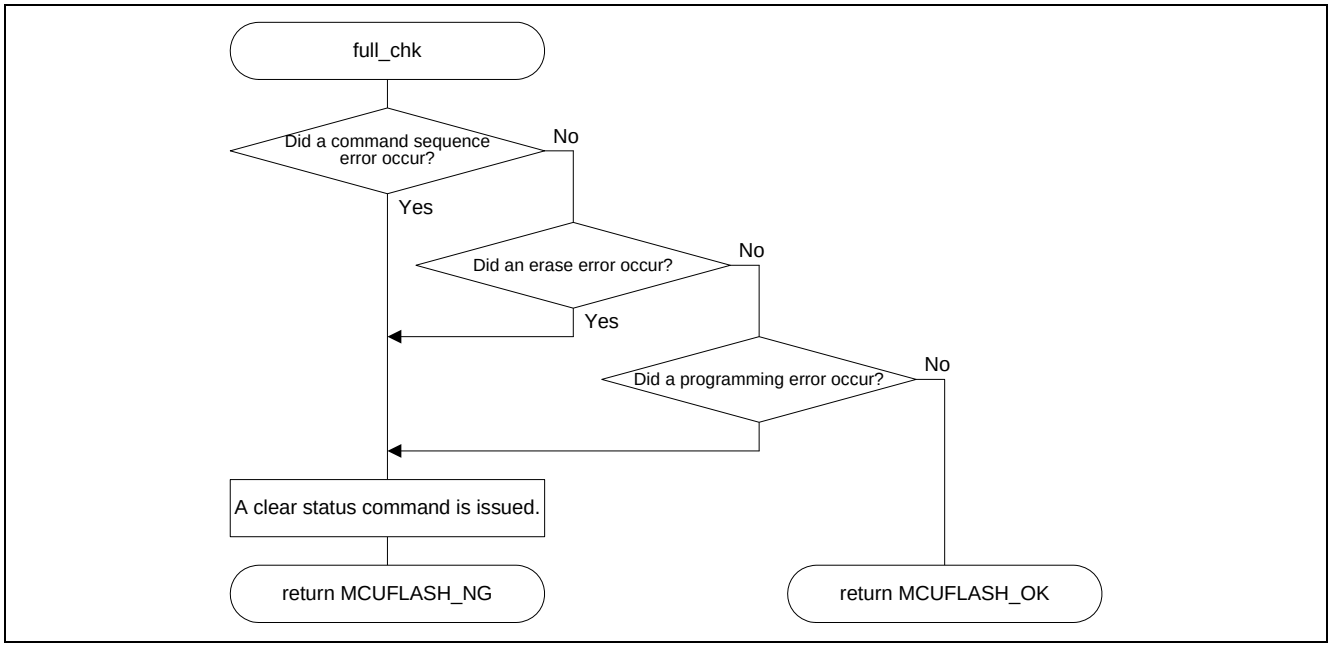


Figure 13 Flowchart (full_chk)

6. Documents for Reference

- Hardware Manual
  H8S/2456, H8S/2456R, H8S/2454 Group Hardware Manual (REJ09B0467)
  (The most up-to-date version of this document is available on the Renesas Electronics Website.)
- Development Tool Manual
  H8S H8/300 Series C/C++ Compiler Package User's Manual (REJ10J2039)
  (The most up-to-date version of this document is available on the Renesas Electronics Website.)
- Technical Update
  (The most up-to-date version of this document is available on the Renesas Electronics Website.)

Website and Support

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

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

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Jun 07, 10	—	First edition issued
1.01	Jun 15, 10	3	Evaluation board is corrected.
1.02	Sep 07, 11	2	Note is added.
		3	Conditions for application are changed.
		10	volatile_evenaccess is added as a RAM variable.
		16	Flow charts and tables are changed due to the revision.
		—	Source files are revised. (flash.c and flash_main.c)

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord 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 one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

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.

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.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, 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 Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1