RX72T Group

Initial Settings Example

Introduction

This application note describes the initial setting that must be performed after a reset according to usage conditions specified in the header file in the RX72T Group. The initial setting for the RX72T Group includes clock settings, stop processing for peripheral modules operating after a reset, and nonexistent port setting.

Target Device

- RX72T Group100-/144-pin version, ROM capacity: 512KB to 1 MB

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Specifications

The sample code provides processing to stop peripheral modules operating after a reset and configure nonexistent ports and clock settings. This application note assumes processing at power-on (cold start).

1.1 Project Description

This application note includes the project “r01an4398_rx72t_clock”.

r01an4398_rx72t_clock is the project specific to Renesas Starter Kit forRX72T. This project contains files that were generated automatically by e^2 studio. In addition, the project settings match the device mounted on the RSK board (ROM capacity: 1MB, pin count: 144, programmable gain amplifier (PGA) with pseudo-differential input, and with USB). When using another device, change the project settings as necessary.

1.2 Stop Processing for Peripheral Modules Operating After a Reset

The module stop function is disabled for some peripheral modules after power-on. Processing to stop the functioning of the following peripheral modules is provided:

- DMAC, DTC, ECCRAM, and RAM

Note that this processing is disabled in the sample code. To enable this processing, change the appropriate constants.

1.3 Nonexistent Port Setting

The direction control bits for nonexistent ports must be set as described in 20.4.1, “Initialization of the Port Direction Register (PDR),” in the hardware manual. The initial settings contained in the sample code accompanying this application note are appropriate for 144-pin products with PGA pseudo-differential input and with USB.

Change constants appropriate to the product used.

1.4 Clock Settings

1.4.1 Overview

Clock settings are performed in the following order:

1. Main clock setting
2. HOCO clock setting
3. PLL clock setting
4. System clock switching

In this application note, clock settings can be switched by changing constants defined in r_init_clock.h.

The sample code uses PLL as the system clock. Select the clock to be used by changing the appropriate constant.

1.4.2 Clock Specifications in the Sample Code

Table 1.1 lists Clock Specifications in the Sample Code.

Table 1.1 Clock Specifications in the Sample Code

Clock	Oscillation Frequency	Oscillation Stabilization Time	Remarks
Main clock oscillator	8 MHz	4.2 ms (2)	Crystal
PLL clock	200 MHz (Main clock × 1/1 × 25)	— (3)	—
HOCO clock	20 MHz (1)	— (3)	—

Notes: 1. The clock is stopped in the sample code.
2. The oscillation stabilization time of each crystal/ceramic resonator differs depending on wiring pattern, conditions of oscillation parameters, and other settings in the user system. To determine the appropriate oscillation stabilization time, ask the crystal/ceramic resonator manufacturer to evaluate the user system.
3. Refer to the Electrical Characteristics chapter in the User's Manual: Hardware.

1.4.3 Clock Selection

In the sample code, the clock source of the system clock and enable/disable setting of each clock can be selected by changing constants defined in r_init_clock.h.

Table 1.2 lists an Example of Clock Selection. In the sample code No. 1 is selected.

Table 1.2 Example of Clock Selection

No.	1	2	3	4
System clock	PLL	PLL	HOCO	Main clock
PLL clock	Oscillating	Oscillating	Stopped	Stopped
Main clock	Oscillating	Stopped	Stopped	Oscillating
HOCO clock	Stopped	Oscillating	Oscillating	Stopped
Memory wait cycle (1)	1 wait	1 wait	0 wait	0 wait
Constant (2)				
SEL_SYSCLK	CLK_PLL	CLK_PLL	CLK_HOCO	CLK_MAIN
SEL_PLL	B_USE_PLL_MAIN	B_USE_PLL_HOCO	B_NOT_USE	B_NOT_USE
SEL_MAIN	B_USE	B_NOT_USE	B_NOT_USE	B_USE
SEL_HOCO	B_NOT_USE	B_USE	B_USE	B_NOT_USE
REG_MEMWAIT	MEMWAIT_1WAIT	MEMWAIT_1WAIT	MEMWAIT_0WAIT	MEMWAIT_0WAIT

Notes: 1. When ICLK is faster than 120 MHz, set the REG_MEMWAIT constant to MEMWAIT_1WAIT (1 wait).
2. For the values of constants, refer to Table 3.10, Table 3.11, and Table 3.12, “Constants Used in the Sample Code (User Changeable),” and Table 3.13, “Constants Used in the Sample Code (Not User Changeable).”

1.5 Notes on Voltage Level Setting Register (VOLSR)

1.5.1 Notes on Setting VOLSR

For RX72T, the voltage level setting register (VOLSR) needs to be set properly according to conditions and voltages used.

Conditions and voltages:

- Whether to use the USB
- Whether to use PGA pseudo-differential input, and when used, voltage level of analog power supply (AVCC)
- Whether to use the RIIC, and when used, voltage level of power supply (VCC)

In this application note, the voltage level setting register (VOLSR) is set to initial values shown in the following table assuming to use Renesas Starter Kit for RX72T with factory settings.

Table 1.3 Initial Values Set in the Voltage Level Setting Register (VOLSR)

Symbol	Bit Name	Function	Initial Value	Description of Why the Initial Value is Used
USBVON	USB power supply control bit	0: When USB not in use (Deep software standby mode included) 1: When USB in use	0	USB is not used in RX72T Group Initialize Example.
PGAVLS	PGA operating condition setting bit	0: The AVCC voltage is at least 4.0 V, and the pseudo-differential inputs of the PGAs are enabled and negative voltages is to be input to the pins. 1: The AVCC voltage is lower than 4.0 V or negative voltages are not to be input to the pins.	1	Factory setting of analog power supply voltage on Renesas Starter Kit for RX72T is 3.3 V.
RICVLS	RIIC operating voltage setting bit	0: VCC ≥ 4.5 V 1: VCC < 4.5 V	1	Factory setting of power supply on Renesas Starter Kit for RX72T is 3.3 V.

Specify the voltage level setting register (VOLSR) according to the usage conditions. If setting values are inappropriate, the operation cannot be guaranteed. In particular, if PGA pseudo-differential input is used while the PGA operating condition setting bit (PGAVLS) is set to 1 and 5 V is applied to the analog power supply voltage, the device could be destroyed.

2. Operation Confirmation Conditions

The sample code (No. 1 to 4 in Table 1.2) accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 lists the Operation Confirmation Conditions for r01an4398_rx72t_clock.

Table 2.1 Operation Confirmation Conditions for r01an4398_rx72t_clock

Item		Contents
MCU used		R5F572TKCDFB (RX72T Group)
Operating frequency	PLL selected as system clock, and main clock used as PLL input (No. 1 in Table 1.2)	• Main clock: 8 MHz • PLL: 200 MHz(main clock ×1/1 ×25) • System clock (ICLK): 200 MHz (PLL ×1/1) • Peripheral module clock A (PCLKA): 100 MHz (PLL ×1/2) • Peripheral module clock B, D (PCLKB, PCLKD): 50 MHz (PLL ×1/4) • Peripheral module clock C (PCLKC): 200 MHz (PLL ×1/1) • Flash-IF clock (FCLK): 50 MHz (PLL ×1/4) • External bus clock (BCLK): 50 MHz (PLL ×1/4)
	PLL selected as system clock, and HOCO used as PLL input (No. 2 in Table 1.2)	• HOCO: 20 MHz • PLL: 200 MHz (HOCO ×1/2 ×10) • System clock (ICLK): 200 MHz (PLL ×1/1) • Peripheral module clock A (PCLKA): 100 MHz (PLL ×1/2) • Peripheral module clock B, D (PCLKB, PCLKD): 50 MHz (PLL ×1/4) • Peripheral module clock C (PCLKC): 200 MHz (PLL ×1/1) • Flash-IF clock (FCLK): 50 MHz (PLL ×1/4) • External bus clock (BCLK): 50 MHz (PLL ×1/4)
	HOCO clock selected as system clock (No. 3 in Table 1.2)	• HOCO: 20 MHz • System clock (ICLK): 20 MHz (HOCO ×1/1) • Peripheral module clock A to D (PCLKA to PCLKD): 20 MHz (HOCO clock ×1/1) • Flash-IF clock (FCLK): 20 MHz (HOCO clock ×1/1) • External bus clock (BCLK): 20 MHz (HOCO ×1/1)
	Main clock selected as system clock (No. 4 in Table 1.2)	• Main clock: 8 MHz • System clock (ICLK): 8 MHz (Main clock ×1/1) • Peripheral module clock A to D (PCLKA to PCLKD): 8 MHz (Main clock ×1/1) • Flash-IF clock (FCLK): 8 MHz (Main clock ×1/1) • External bus clock (BCLK): 8 MHz (Main clock ×1/1)
Operating voltage		3.3 V
Integrated development environment		Renesas Electronics e^2 studio Version 7.3.0
C compiler		Renesas Electronics C/C++ Compiler Package for RX Family V3.01
		Compiler option The default settings in the integrated development environment are used.
iodefine.h version		V1.00
Endian		Little endian, big endian
Operating mode		Single-chip mode
Processor mode		Supervisor mode
Sample code version		Version 1.00
Board used		Renesas Starter Kit for RX72T(product No.: RTK5572TKCSxxxxxBE)

3. Software

In the sample code, peripheral functions operating after a reset are stopped, nonexistent ports are configured, and then clock settings are configured.

3.1 Stop Processing for Peripheral Modules Operating After a Reset

Peripheral modules that are operating after a reset are stopped in this processing.

Table 3.1 lists Peripheral Modules Released from the Module-Stop State After a Reset.

To enable transitioning to the module-stop state after a reset, set the corresponding module stop bit to 1 (transition to the module-stop state). Power consumption can be reduced by entering the module-stop state.

In the sample code, the MSTP_STATE_“target module” constant is set to 0 (MODULE_STOP_DISABLE) and the target module does not enter the module-stop state. When the system requires a module to enter the module-stop state, set the constant in r_init_stop_module.h to 1 (MODULE_STOP_ENABLE).

Table 3.1 Peripheral Modules Released from the Module-Stop State After a Reset

Peripheral Module	Module Stop Setting Bit	Value After Reset	Setting When Not Using the Module
DMAC/DTC	MSTPCRA.MSTPA28 bit	0 (Release from the module-stop state)	1 (Transition to the module-stop state is made)
ECCRAM	MSTPCRC.MSTPC6 bit		
RAM	MSTPCRC.MSTPC0 bit		

3.2 Nonexistent Port Setting

3.2.1 Processing Overview

Bits corresponding to nonexistent ports in the PDR register are set to 0 (input) or 1 (output). Values are set according to 20.4 “Initialization of the Port Direction Register (PDR)” in the User’s Manual: Hardware. After calling this function, when writing in byte units to the PDR registers which include nonexistent ports, set the I/O select bits corresponding to nonexistent ports according to 20.4 “Initialization of the Port Direction Register (PDR)” in the User’s Manual: Hardware, and when writing in byte units to the PODR registers, set the port output data storage bits corresponding to nonexistent ports to 0.

Table 3.2 and Table 3.3list Nonexistent Ports on Devices with PGA Pseudo-Differential Input, and Table 3.4 lists Nonexistent Ports on Devices without PGA Pseudo-Differential Input.

Table 3.2 Nonexistent Ports on Devices with PGA Pseudo-Differential Input (1)

Port Symbol	144-Pin Package, With USB	Number of Pins
PORT0	P02 to P07	6
PORT1	—	—
PORT2	—	—
PORT3	—	—
PORT4	—	—
PORT5	P56, P57	2
PORT6	P66, P67	2
PORT7	P77	1
PORT8	P83 to P87	5
PORT9	P97	1
PORTA	—	—
PORTB	—	—
PORTC	PC7	1
PORTD	—	—
PORTE	PE7	1
PORTF	PF4 to PF7	4
PORTG	PG3 to PG7	5
PORTH	—	—
PORTK	PK3 to PK7	5

Table 3.3 Nonexistent Ports on Devices with PGA Pseudo-Differential Input (2)

Port Symbol	100-Pin Package, With USB	Number of Pins	100-Pin Package, Without USB	Number of Pins
PORT0	P02 to P07	6	P02 to P07	6
PORT1	P12 to P17	6	P12 to P17	6
PORT2	P25, P26	2	P25, P26	2
PORT3	P34, P35	2	P34, P35	2
PORT4	—	—	—	—
PORT5	P50, P51, P56, P57	4	P50, P51, P56, P57	4
PORT6	P66, P67	2	P66, P67	2
PORT7	P77	1	P77	1
PORT8	P83 to P87	5	P83 to P87	5
PORT9	P97	1	P97	1
PORTA	PA6, PA7	2	PA6, PA7	2
PORTB	PB7	1	—	—
PORTC	PC0 to PC7	8	PC0 to PC7	8
PORTD	PD0, PD1	2	—	—
PORTE	PE6, PE7	2	PE6, PE7	2
PORTF	PF0 to PF7	8	PF0 to PF7	8
PORTG	PG0 to PG7	8	PG0 to PG7	8
PORTH	PH1 to PH3, PH5 to PH7	6	PH1 to PH3, PH5 to PH7	6
PORTK	PK0 to PK7	8	PK0 to PK7	8

Table 3.4 Nonexistent Ports on Devices without PGA Pseudo-Differential Input

Port Symbol	100-Pin Package, Without USB	Number of Pins
PORT0	P02 to P07	6
PORT1	P12 to P17	6
PORT2	P25 to P27	3
PORT3	P34, P35	2
PORT4	—	—
PORT5	P56, P57	2
PORT6	P66, P67	2
PORT7	P77	1
PORT8	P83 to P87	5
PORT9	P97	1
PORTA	PA6, PA7	2
PORTB	—	—
PORTC	PC0 to PC7	8
PORTD	—	—
PORTE	PE6, PE7	2
PORTF	PF0 to PF7	8
PORTG	PG0 to PG7	8
PORTH	PH0 to PH7	8
PORTK	PK0 to PK7	8

3.2.2 Specifying the Number of Pins

In the sample code the pin count is set to 144 (PIN_SIZE=144). The sample code is also compatible with pin counts of 100. When using a product with a pin count other than 144, change PIN_SIZE in r_init_port_initialize.h to match the pin count of your device.

3.2.3 With/Without PGA Pseudo-Differential Input Setting

The sample code contains a setting specifying PGA pseudo-differential input (PGA_DEFAMP=WITH). When using a product without PGA pseudo-differential input, change the PGA_DEFAMP setting value in r_init_port_initialize.h to WITHOUT.

3.2.4 With/Without USB Setting

The sample code contains a setting specifying no USB (USB_MODULE=WITH). When using a product without USB, change the USB_MODULE setting value in r_init_port_initialize.h to WITHOUT.

3.3 Clock Settings

3.3.1 Clock Setting Procedure

Table 3.5 lists the clock setting procedure with each processing and the default setting in the sample code. The main clock and PLL are operating, and HOCO is stopped by default in the sample code.

Table 3.5 Clock Setting Procedure

Step	Processing	Description		Setting in Sample Code
1	Main clock setting (2)	Not used	No setting is required.	The main clock is used.
		Used	Sets the main clock drive capacity, sets the wait time to the MOSCWTCR register (wait time is time until the main clock output is provided to the internal clock), and enables main clock oscillation. Then waits for the oscillation stabilization waiting time (1) by hardware.	
2.	HOCO clock setting (2)	Not used	HOCO is turned off.	The HOCO clock is not used.
		Used	Sets the HOCO frequency and enables HOCO clock oscillation. Then waits for the oscillation stabilization waiting time (1) by hardware.	
3	PLL clock setting (2)	Not used	No setting is required.	The PLL clock is used.
		Used	Sets the PLL input frequency division ratio and frequency multiplication factor, and enables PLL clock oscillation. Then wait for the clock oscillation stabilization waiting time (1) by hardware.	
4	Clock division ratio settings	Sets the clock division ratios.		• ICLK: ×1/1 • PCLKA: ×1/2 • PCLKC: ×1/1 • PCLKB, PCLKD, BCLK, FCLK: ×1/4 • BCLK: Stopped
5	System clock switching	Switches the system clock according to the user system.		Switches to PLL clock.

Notes: 1. The target bit in the oscillation stabilization flag register (OSCOVFSR) is checked to see if the bit has become 1.
2. Change the constant value in r_init_clock.h to set clocks to be used or not to be used.

3.4 Section Configuration

Table 3.6 lists the Information of the Section Changed in the Sample Code (r01an4398_rx72t).

To add, change, or delete sections, refer to the latest version of the RX Family CC-RX Compiler User's Manual.

Table 3.6 Information of the Section Changed in the Sample Code (r01an4398_rx72t)

Section Name	Change	Address	Description
End_of_RAM	Add	0001 FFFCh	End address of the on-chip RAM
End_of_ECCRAM	Add	00FF FFFCh	End address of the ECCRAM

3.5 File Composition

Table 3.7 lists Files Used in the Sample Code. Files generated by the integrated development environment are not included in this table.

Table 3.7 Files Used in the Sample Code

File Name	Overview	Remarks
main.c	Main processing	
r_init_stop_module.c	Stop processing for peripheral modules operating after a reset	
r_init_stop_module.h	Header file for r_init_stop_module.c	
r_init_port_initialize.c	Nonexistent port initialization	
r_init_port_initialize.h	Header file for r_init_port_initialize.c	
r_init_clock.c	Clock initialization	
r_init_clock.h	Header file for r_init_clock.c	
r_init_rom_cache.c	ROM cache initialization	
r_init_rom_cache.h	Header file for r_init_rom_cache.c	

3.6 Option-Setting Memory

Table 3.8 lists the Option-Setting Memory Used in the Sample Code. When necessary, set a value suited to the user system.

Table 3.8 Option-Setting Memory Used in the Sample Code

Symbol	Address	Setting Value	Contents
OFS0	0012 0068h to 0012 006Bh	FFFF FFFFh	The IWDT is stopped after a reset. The WDT is stopped after a reset.
OFS1	0012 006Ch to 0012 006Fh	FFFF FFFFh	The voltage monitor 0 reset is disabled after a reset. HOCO oscillation is disabled after a reset.
MDE	0012 0064h to 0012 0067h	FFFF FFFFh	Little endian

3.7 Constants

Table 3.9 to Table 3.11 list Constants Used in the Sample Code (User Changeable), and Table 3.12 lists Constants Used in the Sample Code (Not User Changeable).

Table 3.13 to Table 3.16 show constants for the various product versions.

Table 3.9 Constants Used in the Sample Code (User Changeable) (1/4)

Constant Name	Setting Value	Contents
SEL_MAIN (1)	B_USE	Main clock enable/disable selection: - B_USE: Used (main clock oscillating) - B_NOT_USE: Not used (main clock stopped)
MAIN_CLOCK_Hz (1)	8000000L	Oscillation frequency of a crystal/ceramic resonator for the main clock (Hz)
REG_MOFCR (1)	30h	Drive capacity setting of the main clock oscillator (setting value of the MOFCR register)
REG_MOSCWTCR (1)	53h	Setting value of the main clock oscillator wait control register
SEL_PLL (1)	B_USE_PLL_MAIN	PLL clock enable/disable selection: - B_USE_PLL_MAIN: Used (main) - B_USE_PLL_HOCO: Used (HOCO) - B_NOT_USE: Not used (PLL clock stopped)
REG_PLLCR (1, 2, 3)	1310h (HOCO as PLL clock source) 3100h (Other than above)	PLL input division ratio and frequency multiplication factor settings (setting value of the PLLCR register) 1310h: Clock source: HOCO, ×1/1, ×10 3100h: Clock source: Main clock, ×1/1, ×25
REG_SCKCR (1, 2, 3)	2082 1202h (PLL selected) 0080 0000h (HOCO selected) 0080 0000h (Other than above)	Internal clock division ratio and BCLK control setting (setting value of the SCKCR register)
REG_VOLSR (1, 3)	C0h	Settings for USB usage, PGA differential input usage and the voltage level, and VCC voltage level when using RIIC (setting value of the VOLSR register)
REG_ MEMWAIT (1, 4)	MEMWAIT_1WAIT	Selection of the memory wait cycle: - MEMWAIT_0WAIT: 0 wait - MEMWAIT_1WAIT: 1 wait

Notes: 1. Change the constant value in r_init_clock.h according to the user system.
2. The setting value differs depending on the clock source of the selected system clock.
3. When changing setting values in this register, follow the User's Manual: Hardware.
4. Specify 1 wait cycle if ICLK is faster than 120 MHz.

Table 3.10 Constants Used in the Sample Code (User Changeable) (2/4)

SEL_SYSCLK (1)	CLK_PLL	System clock source selection: - CLK_PLL: PLL clock - CLK_HOCO: HOCO clock - CLK_MAIN: Main clock

Note: 1. Change the constant value in r_init_clock.h according to the user system.

Table 3.11 Constants Used in the Sample Code (User Changeable) (3/4)

Constant Name	Setting Value	Contents
MSTP_STATE_DMACDTC (1)	MODULE_STOP_DISABLE	Selection of module stop state for DMAC and DTC - MODULE_STOP_DISABLE: Release - MODULE_STOP_ENABLE: Transition
MSTP_STATE_ECCRAM (1)	MODULE_STOP_DISABLE	Selection of module stop state for ECCRAM - MODULE_STOP_DISABLE: Release - MODULE_STOP_ENABLE: Transition
MSTP_STATE_RAM (1)	MODULE_STOP_DISABLE	Selection of module stop state for RAM - MODULE_STOP_DISABLE: Release - MODULE_STOP_ENABLE: Transition
PIN_SIZE (2)	144	Number of pins for the product used
PGA_DEFAMP (2)	WITH	Selection of the product with/without PGA pseudo-differential input: - WITHOUT: Product without PGA pseudo-differential input - WITH: Product with PGA pseudo-differential input
USB_MODULE (2)	WITH	Selection of product with or without USB - WITHOUT: Product without USB - WITH: Product with USB
SEL_ROM_CACHE (3)	CACHE_ENABLE	Selection of ROM cache enabled/disabled - CACHE_ENABLE: ROM cache enabled - CACHE_DISABLE: ROM cache disabled
SEL_NON_CACHEABLE_AREA0	NON_CACHEABLE_AREA_DISABLE	Selection of non-cacheable area 0 enabled/disabled - NON_CACHEABLE_AREA_ENABLE: Enabled - NON_CACHEABLE_AREA_DISABLE: Disabled
SEL_NON_CACHEABLE_AREA1	NON_CACHEABLE_AREA_DISABLE	Selection of non-cacheable area 1 enabled/disabled - NON_CACHEABLE_AREA_ENABLE: Enabled - NON_CACHEABLE_AREA_DISABLE: Disabled

Notes: 1. Change the setting value in r_init_stop_module.h according to the user system.
2. Change the setting value in r_init_port_initialize.h according to the user device (package).
3. Change the setting value in r_init_rom_cache.h according to the user system.

Table 3.12 Constants Used in the Sample Code (Not User Changeable) (4/4)

Constant Name	Setting Value	Contents
B_NOT_USE	0	Clock not used
B_USE	1	Clock used
B_USE_PLL_MAIN	2	PLL clock used (clock source: main clock)
B_USE_PLL_HOCO	3	PLL clock used (clock source: HOCO)
FREQ_16MHz	00h	HOCO frequency: 16 MHz
FREQ_18MHz	01h	HOCO frequency: 18 MHz
FREQ_20MHz	02h	HOCO frequency: 20 MHz
CLK_PLL	0400h	Clock source: PLL
CLK_HOCO	0100h	Clock source: HOCO
CLK_MAIN	0200h	Clock source: Main clock
MEMWAIT_0WAIT	0	Memory wait cycle: 0 wait
MEMWAIT_1WAIT	1	Memory wait cycle: 1 wait
REG_SCKCR2	0011h	USB clock division ratio (setting value when USB is not used)
WITH	1	Function implemented
WITHOUT	0	Function not implemented
MODULE_STOP_ENABLE	1	Transition to the module-stop state is made.
MODULE_STOP_DISABLE	0	Release from the module-stop state
CACHE_ENABLE	1	ROM cache enabled
CACHE_DISABLE	0	ROM cache disabled
NON_CACHEABLE_AREA_DISABLE	0	Non-cacheable area disabled
NON_CACHEABLE_AREA_ENABLE	1	Non-cacheable area enabled

Table 3.13 Constants for 144-Pin Products with PGA Pseudo-Differential Input and with USB (PIN_SIZE=144, PGA_DEFAMP=WITH, USB_MODULE=WITH)

Constant Name	Setting Value	Contents
DEF_P0PDR	00h	Setting value for the port P0 direction register
DEF_P1PDR	00h	Setting value for the port P1 direction register
DEF_P2PDR	00h	Setting value for the port P2 direction register
DEF_P3PDR	00h	Setting value for the port P3 direction register
DEF_P4PDR	00h	Setting value for the port P4 direction register
DEF_P5PDR	00h	Setting value for the port P5 direction register
DEF_P6PDR	00h	Setting value for the port P6 direction register
DEF_P7PDR	00h	Setting value for the port P7 direction register
DEF_P8PDR	00h	Setting value for the port P8 direction register
DEF_P9PDR	00h	Setting value for the port P9 direction register
DEF_PAPDR	00h	Setting value for the port PA direction register
DEF_PBPDR	00h	Setting value for the port PB direction register
DEF_PCPDR	00h	Setting value for the port PC direction register
DEF_PDPDR	00h	Setting value for the port PD direction register
DEF_PEPDR	00h	Setting value for the port PE direction register
DEF_PFPDR	00h	Setting value for the port PF direction register
DEF_PGPDR	00h	Setting value for the port PG direction register
DEF_PHPDR	00h	Setting value for the port PH direction register
DEF_PKPDR	00h	Setting value for the port PK direction register

Table 3.14 Constants for 100-Pin Products with PGA Pseudo-Differential Input and with USB (PIN_SIZE=100, PGA_DEFAMP=WITH, USB_MODULE=WITH)

Constant Name	Setting Value	Contents
DEF_P0PDR	00h	Setting value for the port P0 direction register
DEF_P1PDR	FCh	Setting value for the port P1 direction register
DEF_P2PDR	60h	Setting value for the port P2 direction register
DEF_P3PDR	30h	Setting value for the port P3 direction register
DEF_P4PDR	00h	Setting value for the port P4 direction register
DEF_P5PDR	03h	Setting value for the port P5 direction register
DEF_P6PDR	00h	Setting value for the port P6 direction register
DEF_P7PDR	00h	Setting value for the port P7 direction register
DEF_P8PDR	00h	Setting value for the port P8 direction register
DEF_P9PDR	00h	Setting value for the port P9 direction register
DEF_PAPDR	C0h	Setting value for the port PA direction register
DEF_PBPDR	80h	Setting value for the port PB direction register
DEF_PCPDR	7Fh	Setting value for the port PC direction register
DEF_PDPDR	03h	Setting value for the port PD direction register
DEF_PEPDR	40h	Setting value for the port PE direction register
DEF_PFPDR	0Fh	Setting value for the port PF direction register
DEF_PGPDR	07h	Setting value for the port PG direction register
DEF_PHPDR	EEh	Setting value for the port PH direction register
DEF_PKPDR	07h	Setting value for the port PK direction register

Table 3.15 Constants for 100-Pin Products with PGA Pseudo-Differential Input and without USB (PIN_SIZE=100, PGA_DEFAMP=WITH, USB_MODULE=WITHOUT)

Constant Name	Setting Value	Contents
DEF_P0PDR	00h	Setting value for the port P0 direction register
DEF_P1PDR	FCh	Setting value for the port P1 direction register
DEF_P2PDR	60h	Setting value for the port P2 direction register
DEF_P3PDR	30h	Setting value for the port P3 direction register
DEF_P4PDR	00h	Setting value for the port P4 direction register
DEF_P5PDR	03h	Setting value for the port P5 direction register
DEF_P6PDR	00h	Setting value for the port P6 direction register
DEF_P7PDR	00h	Setting value for the port P7 direction register
DEF_P8PDR	00h	Setting value for the port P8 direction register
DEF_P9PDR	00h	Setting value for the port P9 direction register
DEF_PAPDR	C0h	Setting value for the port PA direction register
DEF_PBPDR	00h	Setting value for the port PB direction register
DEF_PCPDR	7Fh	Setting value for the port PC direction register
DEF_PDPDR	00h	Setting value for the port PD direction register
DEF_PEPDR	40h	Setting value for the port PE direction register
DEF_PFPDR	0Fh	Setting value for the port PF direction register
DEF_PGPDR	07h	Setting value for the port PG direction register
DEF_PHPDR	EEh	Setting value for the port PH direction register
DEF_PKPDR	07h	Setting value for the port PK direction register

Table 3.16 Constants for 100-Pin Products without PGA Pseudo-Differential Input and without USB (PIN_SIZE=100, PGA_DEFAMP=WITHOUT, USB_MODULE=WITHOUT)

Constant Name	Setting Value	Contents
DEF_P0PDR	00h	Setting value for the port P0 direction register
DEF_P1PDR	FCh	Setting value for the port P1 direction register
DEF_P2PDR	E0h	Setting value for the port P2 direction register
DEF_P3PDR	30h	Setting value for the port P3 direction register
DEF_P4PDR	00h	Setting value for the port P4 direction register
DEF_P5PDR	00h	Setting value for the port P5 direction register
DEF_P6PDR	00h	Setting value for the port P6 direction register
DEF_P7PDR	00h	Setting value for the port P7 direction register
DEF_P8PDR	00h	Setting value for the port P8 direction register
DEF_P9PDR	00h	Setting value for the port P9 direction register
DEF_PAPDR	C0h	Setting value for the port PA direction register
DEF_PBPDR	00h	Setting value for the port PB direction register
DEF_PCPDR	7Fh	Setting value for the port PC direction register
DEF_PDPDR	00h	Setting value for the port PD direction register
DEF_PEPDR	40h	Setting value for the port PE direction register
DEF_PFPDR	0Fh	Setting value for the port PF direction register
DEF_PGPDR	07h	Setting value for the port PG direction register
DEF_PHPDR	EEh	Setting value for the port PH direction register
DEF_PKPDR	07h	Setting value for the port PK direction register

3.8 Functions

Table 3.21 lists the Functions.

Table 3.17 Functions

Function Name	Outline
main	Main processing
R_INIT_StopModule	Stop processing for peripheral modules operating after a reset
R_INIT_Port_Initialize	Nonexistent port initialization
R_INIT_Clock	Clock initialization
R_INIT_ROM_Cache	ROM cache initialization
CGC_oscillation_main	Main clock oscillation setting
CGC_oscillation_PLL	PLL clock oscillation setting
CGC_oscillation_HOCO	HOCO clock oscillation setting

3.9 Function Specifications

The following tables list the specifications of sample code functions.

main	
Outline	Main processing
Header	None
Declaration	void main(void)
Description	Calls the following functions: Stop processing for peripheral modules operating after a reset, nonexistent port initialization, clock initialization, and ROM cache initialization.
Arguments	None
Return Value	None

R_INIT_StopModule	
Outline	Stop processing for peripheral modules operating after a reset
Header	r_init_stop_module.h
Declaration	void R_INIT_StopModule(void)
Description	Configures the setting to transition to the module stop state.
Arguments	None
Return Value	None
Remarks	Transition to the module-stop state is not performed in the sample code.

R_INIT_Port_Initialize	
Outline	Nonexistent port initialization
Header	r_init_port_initialize.h
Declaration	void R_INIT_Port_Initialize(void)
Description	Initializes port direction registers corresponding to nonexistent port pins.
Arguments	None
Return Value	None
Remarks	The settings in the sample code are for a 144-pin product with PGA pseudo-differential input (PIN_SIZE=144, PGA_DEFAMP=WITH). After this function is called, when writing in byte units to the PDR or PODR registers that include nonexistent ports, set the I/O Select bits corresponding to nonexistent ports as indicated in 20.4, “Initialization of the Port Direction Register (PDR),” in the hardware manual. Also, set the output data store bits corresponding to ports set as output ports to 0.

R_INIT_Clock	
Outline	Clock initialization
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Initializes clocks and specifies the memory wait cycle.
Arguments	None
Return Value	None
Remarks	In the sample code, PLL is selected as the system clock and the memory wait cycle is set to 1 wait.

CGC_oscillation_main	
Outline	Main clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_main (void)
Description	Sets the drive capacity of the main clock and sets the MOSCWTCR register, and enables main clock oscillation. Then waits for the main clock oscillation stabilization waiting time by hardware.
Arguments	None
Return Value	None

CGC_oscillation_PLL	
Outline	PLL clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_PLL (void)
Description	Sets the PLL input frequency division ratio and frequency multiplication factor, and enables PLL clock oscillation. Then waits for the PLL clock oscillation stabilization waiting time by hardware.
Arguments	None
Return Value	None

CGC_oscillation_HOCO	
Outline	HOCO clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_HOCO (void)
Description	Sets the HOCO frequency and enables HOCO clock oscillation. Then waits for the HOCO clock oscillation stabilization waiting time by hardware.
Arguments	None
Return Value	None

R_INIT_ROM_Cache	
Outline	ROM cache initialization
Header	r_init_rom_cache.h
Declaration	void R_INIT_ROM_Cache (void)
Description	After specifying the non-cacheable areas, enables the ROM cache.
Arguments	None
Return Value	None
Outline	In the sample code, this function only makes it possible for the ROM cache to operate. It is assumed that this function will be called while the ROM cache is in the disabled state after the system starts. To specify non-cacheable areas after the ROM cache has been enabled, first disable the ROM cache and then call this function.

3.10 Flowcharts

3.10.1 Main Processing

Figure 3.1 shows the Main Processing.

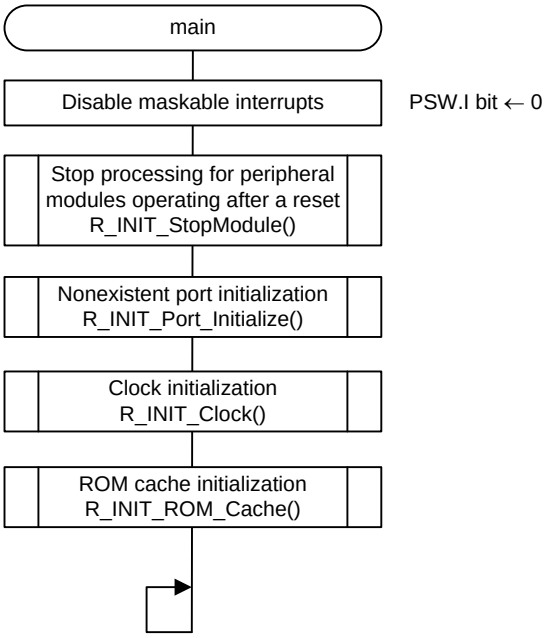
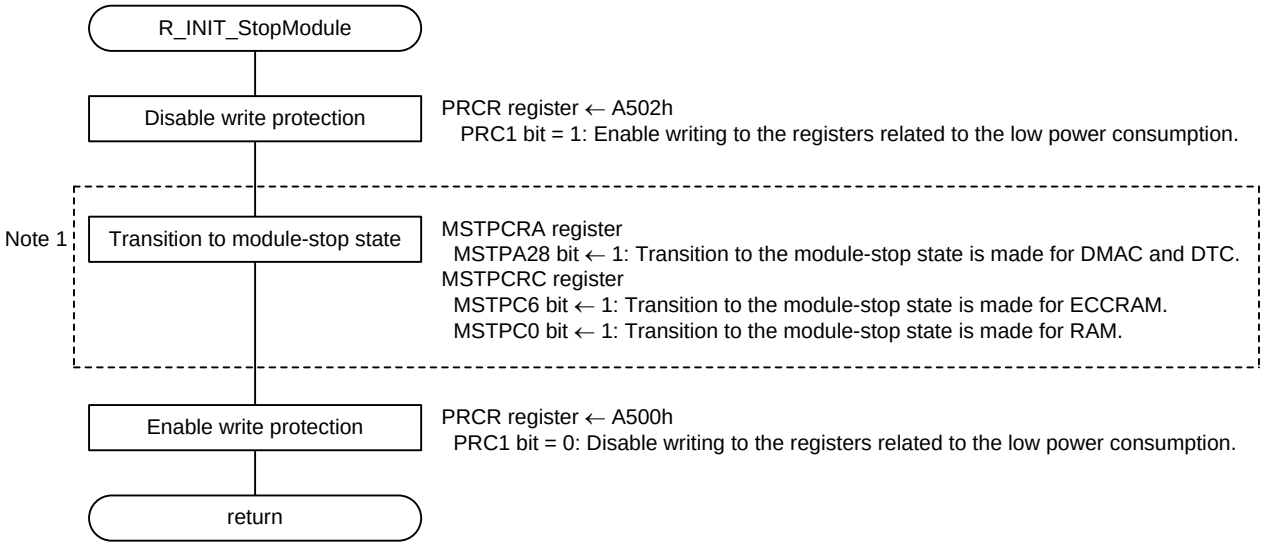


Figure 3.1 Main Processing

3.10.2 Stop Processing for Peripheral Modules Operating After a Reset

Figure 3.2 shows the Stop Processing for Peripheral Modules Operating After a Reset.



Note 1. The module-stop state is canceled in the sample code. To enter the module-stop state for any peripheral modules, set "#define MSTP_STATE_MSTP_STATE_target module" constant to 1.

Figure 3.2 Stop Processing for Peripheral Module Operating After a Reset

3.10.3 Nonexistent Port Initialization

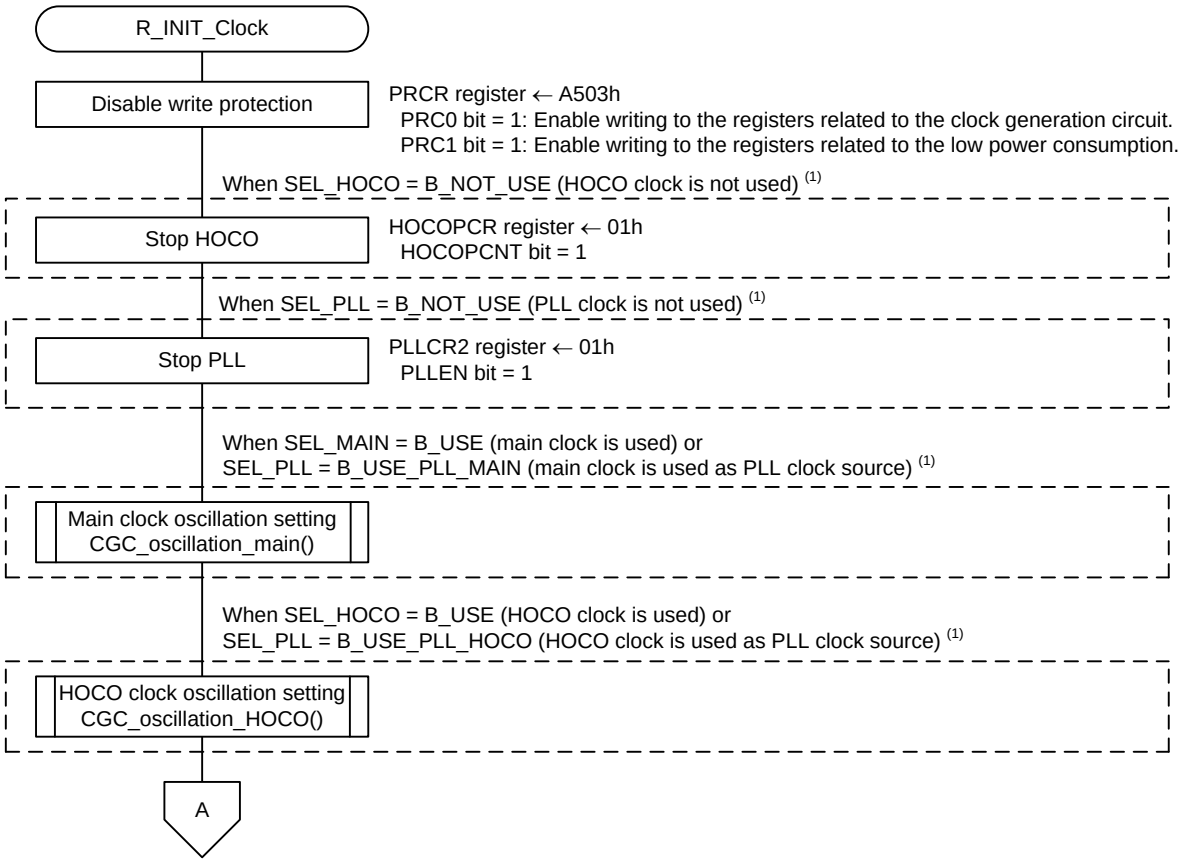
Figure 3.3 shows the Nonexistent Port Initialization.

Note 1. Port initialization processing is not executed if all pins in the register are present (the processing is skipped when compiling).

Figure 3.3 Nonexistent Port Initialize

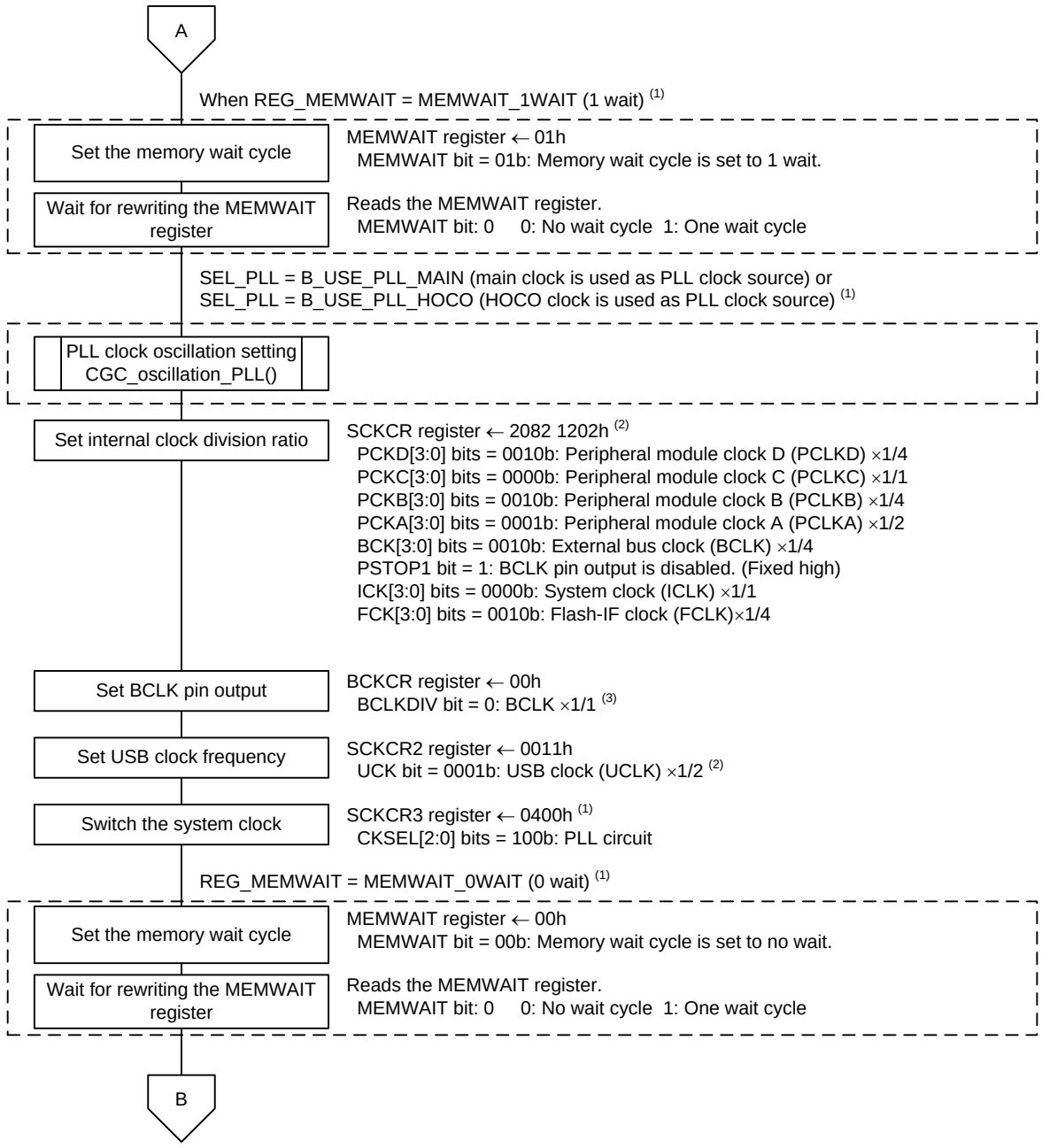
3.10.4 Clock Initialization

Figure 3.4 to Figure 3.6 show the clock initialization.



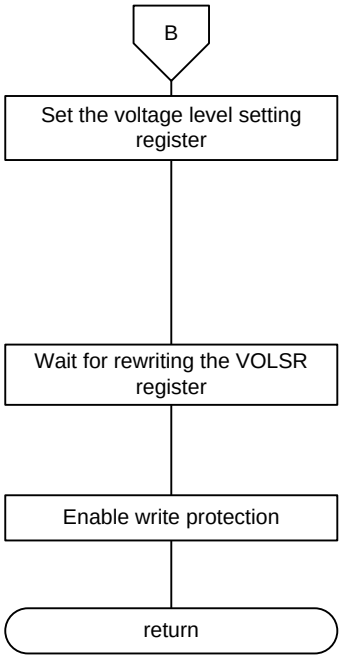
Note 1. Change the constant value depending on the user system.

Figure 3.4 Clock Initialization (1/3)



Notes 1. Change the constant value depending on the user system.
2. The setting value varies depending on the system clock selected by the constant.
3. When the bus clock is set to more than 40 MHz, the BCLKDIV bit in the BCKCR register must be set to 1.

Figure 3.5 Clock Initialization (2/3)



VOLSR register ← C0h (1)
USBVON bit = 0 0: When USB not in use (Deep software standby mode included)
1: When USB in used
PGAVLS bit = 1 0: AVCC voltage of 4.0 V or above, PGA pseudo-differential input enabled, negative voltage input to pin
1: AVCC voltage of less than 4.0 V, or no negative voltage input to pin
RICVLS bit = 1 0: VCC ≥ 4.5 V
1: VCC < 4.5 V

Reads the VOLSR register.
Checks whether the register has been rewritten correctly.

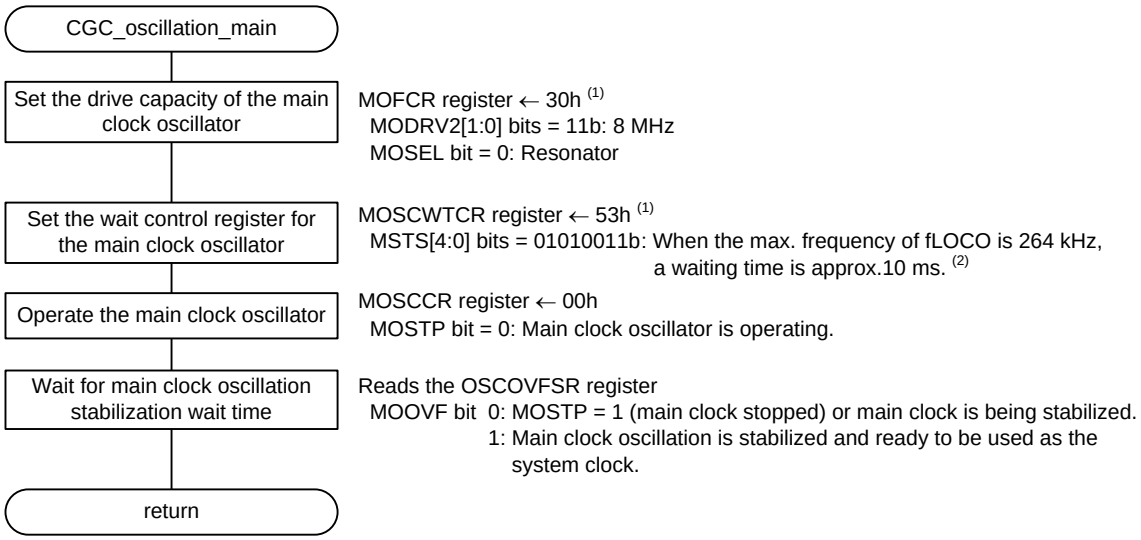
PRCR register ← A500h
PRC0 bit = 0: Prohibit writing to registers related to the clock generation circuit.
PRC1 bit = 0: Prohibit writing to registers related to low power consumption.

Note 1. Change the constant value depending on the user system.

Figure 3.6 Clock Initialization (3/3)

3.10.5 Main Clock Oscillation Setting

Figure 3.7 shows the Main Clock Oscillation Setting.

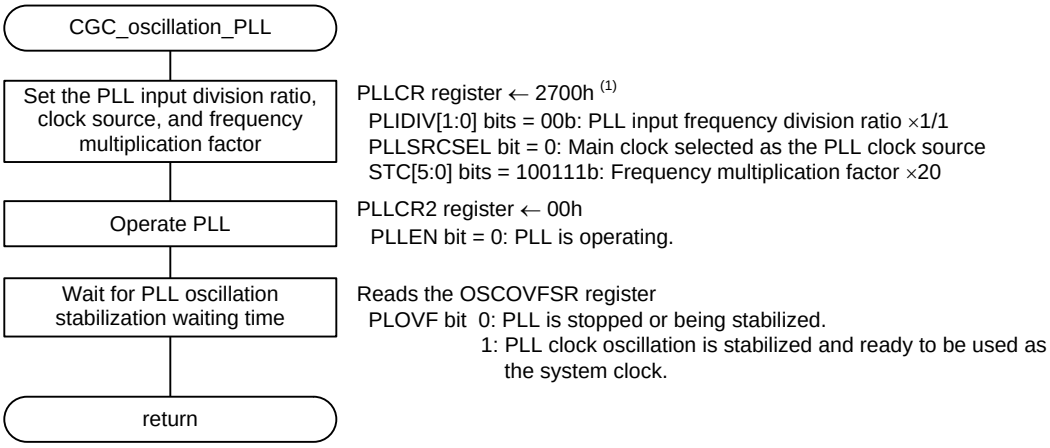


Notes 1. Change the constant value depending on the user system.
2. The initial value for the register is used in this application note.

Figure 3.7 Main Clock Oscillation Setting

3.10.6 PLL Clock Oscillation Setting

Figure 3.8 shows the PLL Clock Oscillation Setting.

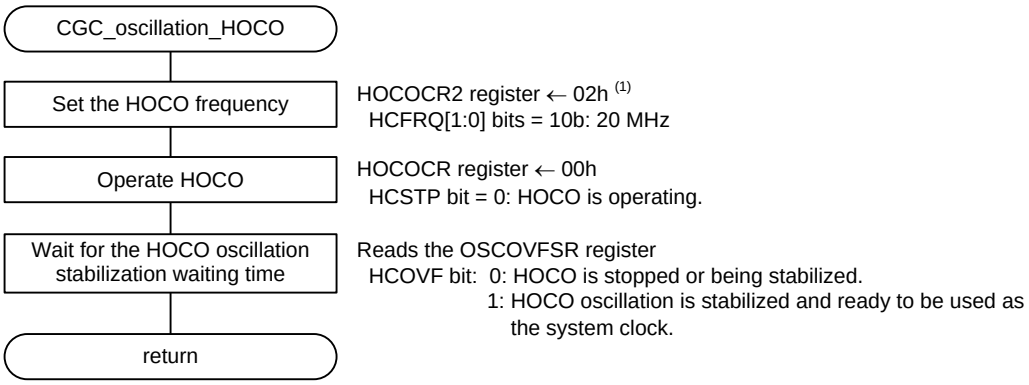


Note 1. Change the constant value depending on the user system.

Figure 3.8 PLL Clock Oscillation Setting

3.10.7 HOCO Clock Oscillation Setting

Figure 3.9 shows the HOCO Clock Oscillation Setting.

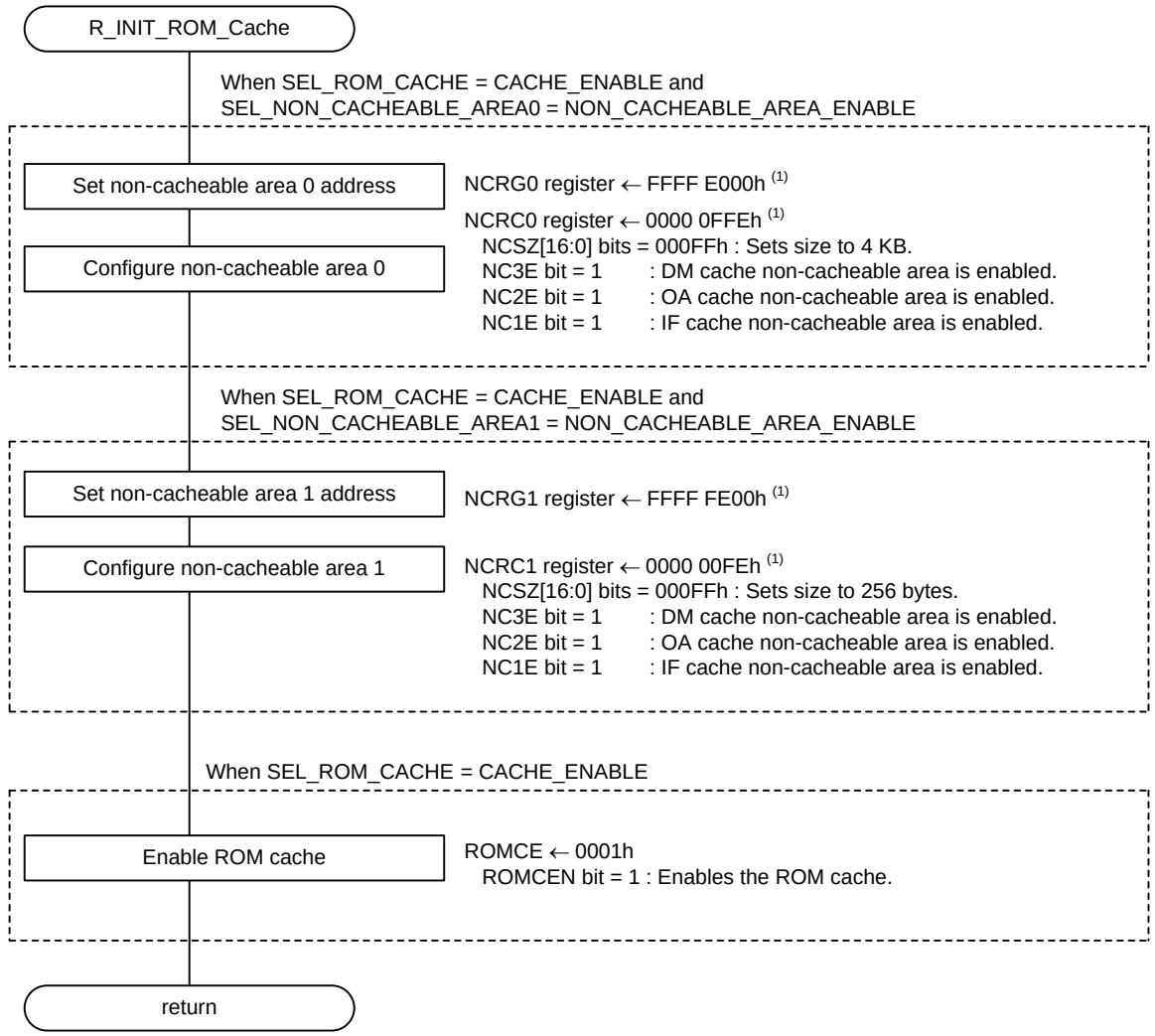


Note 1. Change the constant value depending on the user system.

Figure 3.9 PLL Clock Oscillation Setting

3.10.8 ROM Cache Initialization

Figure 3.10 shows the ROM Cache Initialization.



Note 1. Set the non-cacheable areas to match the characteristics of your system.
In the sample code it is assumed that this function will be called while the ROM cache is in the disabled state after the system starts. To specify non-cacheable areas after the ROM cache has been enabled, first disable the ROM cache and then call this function.

Figure 3.10 ROM Cache Initialization

4. Importing a Project

The sample code is provided as the e² studio project. This section describes importing a project into the e² studio and CS+. After importing a project, confirm that the build settings and the debug settings are correct.

4.1 Importing a Project into the e² studio

Follow the steps below to import your project into the e² studio.
(Windows/dialogs may differ depending on the e² studio version used.)

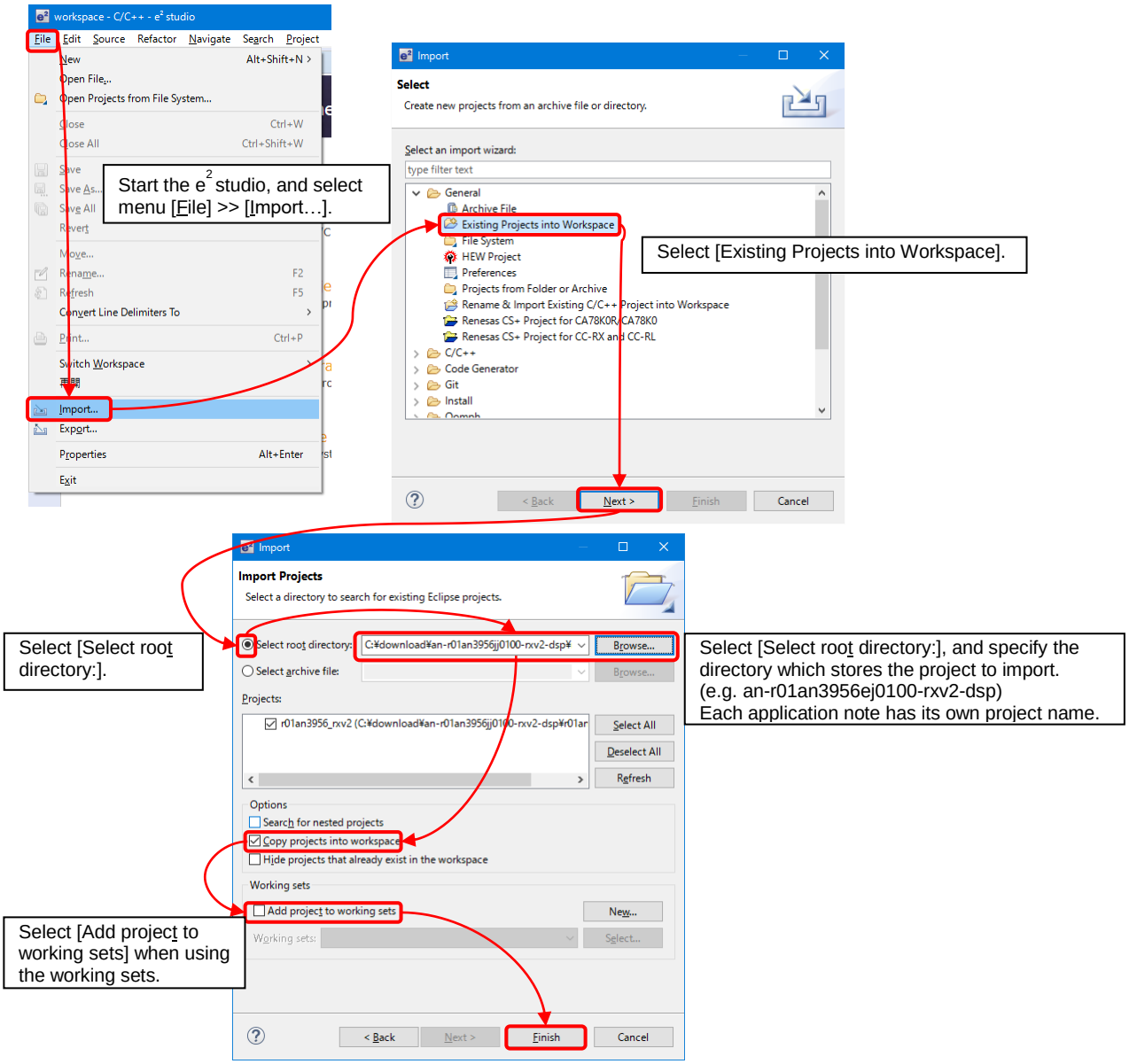


Figure 4.1 Importing a Project into the e² studio

4.2 Importing a Project into CS+

Follow the steps below to import your project into CS+.
(Windows/dialogs may differ depending on the CS+ version used.)

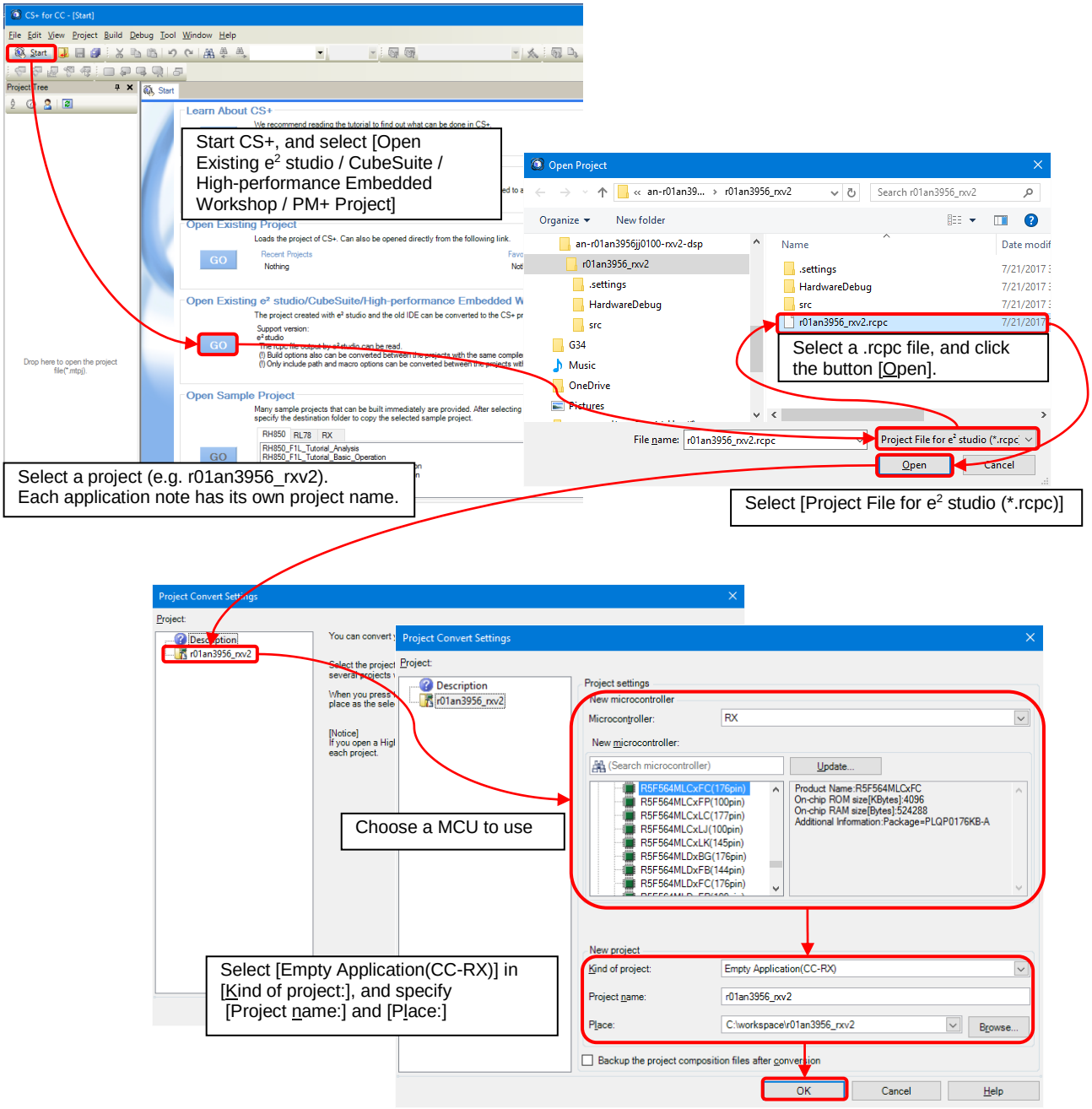


Figure 4.2 Importing a Project into CS+

5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

User's Manual: Hardware
RX72T Group User's Manual: Hardware (R01UH0803)
(The latest version can be downloaded from the Renesas Electronics website.)

Technical Update/Technical News
(The latest version can be downloaded from the Renesas Electronics website.)

User's Manual: Development Tools
RX C/C++ Compiler CC-RX User's Manual (R20UT3248)
(The latest version can be downloaded from the Renesas Electronics website.)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar.29.19	—	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.