RX26T Group

Initial Settings Example

Introduction

This application note describes the tasks that must be performed according to the usage conditions specified in the header file after a reset occurs. These tasks include setting the clocks for the RX26T Group, stopping the peripheral modules that are still operating after a reset, and configuring the nonexistent ports.

Target Device

- 48-pin, 64-pin, 80-pin, or 100-pin version of the RX26T Group with a ROM capacity of 128 to 512 KB

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 also covers "r01an6567_rx26t".

"r01an6567_rx26t" is a project that is applicable to the Renesas Flexible Motor Control Kit for RX26T. This project contains files that were generated automatically by e^2 studio. The settings of this project are adapted for the device mounted on the MCK(Renesas Flexible Motor Control Kit for RX26T) board (a 100-pin device with a ROM capacity of 512 KB). When using another device, change the project settings as necessary.Refer to the following URL for details.
https://en-support.renesas.com/knowledgeBase/18696526

1.2 Stop Processing for Peripheral Modules Operating After a Reset

The module stop function is disabled for some peripheral modules after power-on. In order to stop these modules, the following processing is provided:

- Processing to stop the functionality of the DMAC, DTC, and RAM modules

Note that this processing is disabled in the sample code. To enable this processing, change the appropriate constants.Refer to Table 3.10 for details.

1.3 Nonexistent Port Setting

The direction control bits for nonexistent ports must be set as described in 20.5.1, “Initialization of the Port Direction Register (PDR),” in the User’s Manual: Hardware. In the sample code for this application note, the initial values are adapted for a 100-pin device with a ROM capacity of 512 KB.

Change constants appropriate to the product used.Refer to Chapter 3.2 and Tables 3.12 to 3.15 in Chapter 3.5 for details.

1.4 Clock Settings

1.4.1 Overview

Clock settings are configured in the following order:

(1) Setting the main clock
(2) Setting the HOCO clock
(3) Setting the PLL clock
(4) Switching the system clock

In this application note, the settings of clock 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.Refer to 1.4.3 Clock Selection for details.

1.4.2 Clock Specifications in the Sample Code

Table 1.1 lists Clock Specifications Assumed in the Sample Code.

Table 1.1 Clock Specifications Assumed in the Sample Code

Clock	Oscillation Frequency	Oscillation Stabilization Time	Remarks
Main clock resonator	10 MHz	10 ms (Note 2)	Crystal
PLL clock	240 MHz (Main clock × 1/1 × 24)	— (Note 3)	—
HOCO clock	20 MHz (Note 1)	— (Note 3)	—

Notes: 1. Oscillation is stopped in the sample code.
2. The time required before oscillation of each resonator is stabilized differs depending on the conditions, such as the wiring pattern and oscillation parameters, in the actual system. To determine the appropriate oscillation stabilization time, ask the crystal/ceramic resonator manufacturer to evaluate the user system.
3. Refer to "Electrical Characteristics" in the User’s Manual: Hardware.

1.4.3 Clock Selection

In the sample code, the clock source of the system clock and whether to enable oscillation of each clock can be changed by changing constants defined in r_init_clock.h.

Table 1.2 shows Clock Configuration Examples. In the sample code, the settings of example 1 are configured.

Table 1.2 Clock Configuration Examples

No.		**1**	**2**	**3**	**4**
System clock		PLL	PLL	HOCO	Main clock
PLL clock		Enabled	Enabled	Disabled	Disabled
Main clock		Enabled	Disabled	Disabled	Enabled
HOCO clock		Disabled	Enabled	Enabled	Disabled
Constants (Note 1)	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

Note: 1. For details about the constant settings, refer to Table 3.8, Table 3.9, , and Table 3.11.

1.5 Notes on Voltage Level Setting Register (VOLSR)

1.5.1 Notes on Setting VOLSR

For the RX26T, the voltage level setting register (VOLSR) needs to be set properly according to the power supply voltage (VCC) when using the RIIC.:

In this application note, the initial value in the following table is used for the voltage level setting register (VOLSR) on the assumption that the Renesas Flexible Motor Control Kit for RX26T is used with the 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
RICVLS	RIIC operating voltage setting bit	0: VCC ≥ 4.5 V 1: VCC < 4.5 V	1	The voltage of the power supply of the Renesas Flexible Motor Control Kit for RX26T is set to 3.3 V in the factory settings.

If necessary, change the settings of the voltage level setting register (VOLSR) according to the actual usage conditions. If setting values are inappropriate, the operation cannot be guaranteed.

2. Operation Confirmation Conditions

For the four example settings of the sample code for this application note (Nos. 1 to 4 in Table 1.2), operation was verified under specific conditions. Table 2.1 shows Conditions Under Which Operation of r01an6567_rx26t Was Verified.

Table 2.1 Conditions Under Which Operation of r01an6567_rx26t Was Verified

Item		**Contents**
MCU used		R5F526TFCDFP (RX26T Group)
Operating frequency	If PLL is selected as the system clock and the main clock is used for PLL input (No. 1 in Table 1.2)	• Main clock: 10 MHz • PLL: 240 MHz (main clock × 1/1 × 24) • System clock (ICLK): 120 MHz (PLL × 1/2) • Peripheral module clock A (PCLKA): 120 MHz (PLL × 1/2) • Peripheral module clocks B (PCLKB) and D (PCLKD): 60 MHz (PLL × 1/4) • Peripheral module clock C (PCLKC): 120 MHz (PLL × 1/2) • FlashIF clock (FCLK): 60 MHz (PLL × 1/4)
	If PLL is selected as the system clock and HOCO is used for PLL input (No. 2 in Table 1.2)	• HOCO: 20 MHz • PLL: 240 MHz (HOCO × 1/1 × 12) • System clock (ICLK): 120 MHz (PLL × 1/2) • Peripheral module clock A (PCLKA): 120 MHz (PLL × 1/2) • Peripheral module clocks B (PCLKB) and D (PCLKD): 60 MHz (PLL × 1/4) • Peripheral module clock C (PCLKC): 120 MHz (PLL × 1/2) • FlashIF clock (FCLK): 60 MHz (PLL × 1/4)
	If HOCO is selected as the system clock (No. 3 in Table 1.2)	• HOCO: 20 MHz • System clock (ICLK): 20 MHz (HOCO × 1/1) • Peripheral module clocks A to D (PCLKA to PCLKD): 20 MHz (HOCO ×1/1) • FlashIF clock (FCLK): 20MHz (HOCO × 1/1)
	If the main clock is selected as the system clock (No. 4 in Table 1.2)	• Main clock: 10 MHz • System clock (ICLK): 10 MHz (main clock × 1/1) • Peripheral module clocks A to D (PCLKA to PCLKD): 10 MHz (main clock × 1/1) • Flash-IF clock (FCLK): 10 MHz (main clock × 1/1)
Operating voltage		5.0 V
Integrated development environment		Renesas Electronics e^2 studio Version: 2023-01
C compiler		Renesas Electronics C/C++ Compiler Package for RX Family V3.05.00
		Compiler option The default settings in the integrated development environment are used.
Version of iodefine.h		V1.00a
Endian		Little endian, big endian
Operating mode		Single-chip mode
Processor mode		Supervisor mode
Sample code version		Version 1.00
Board used		Renesas Flexible Motor Control Kit for RX26T (Product No.:RTK0EMXE70S00020BJ)

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 Stopping Peripheral Modules Still Operating After a Reset

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

Table 3.1 shows List of Peripheral Modules That Do Not Enter the Module-Stop State after a Reset.

For a module to enter the module-stop state after a reset, set the module stop bit to 1 (changes the state to "module-stop"). Power consumption can be reduced by entering the module-stop state.

In the sample code, the constant "MSTP_STATE_target module name" in constant r_init_stop_module.h is set to "0 (MODULE_STOP_DISABLE)" and the target module does not transition to the module stop state.If there are modules whose state must change to "module-stop" in your system, set "1 (MODULE_STOP_ENABLE)" for the corresponding constants.

Table 3.1 List of Peripheral Modules That Do Not Enter 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)
RAM	MSTPCRC.MSTPC0 bit		

3.2 Configuring Nonexistent Ports

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 20.5.1, "Initialization of the Port Direction Register (PDR)" in the User’s Manual: Hardware. If data is written on a byte basis to the PDR register including nonexistent ports after calling the R_INIT_Port_Initialize function in main.c,, make sure that the direction control bits for the nonexistent ports are set as indicated in 20.5.1, "Initialization of the Port Direction Register (PDR)", in the User’s Manual: Hardware. To perform byte-wise writes to the PODR register, set 0 for the port output data storage bit.

Table 3.2 and Table3.3 list nonexistent ports.

Table 3.2 Nonexistent Ports

Port Symbol	100-pin version	Pins	80-pin version	Pins
PORT0	P02 to P07	6	P02 to P07	6
PORT1	P12 to P17	6	P12 to P17	6
PORT2	P25, P26	2	P23 to P26	4
PORT3	P34, P35	2	P32 to P35	4
PORT4	—	—	—	—
PORT5	P56, P57	2	P56, P57	2
PORT6	P66, P67	2	P61 to P63, P66, and P67	5
PORT7	P77	1	P77	1
PORT8	P83 to P87	5	P80 to P87	8
PORT9	P97	1	P97	1
PORTA	PA6, PA7	2	PA0 to PA2, PA4, PA6, and PA7	6
PORTB	—	—	PB7	1
PORTD	—	—	PD0, PD1	2
PORTE	PE6, PE7	2	PE0, PE1, and PE5 to PE7	5
PORTN	PN0 to PN5	6	PN0 to PN5	6

Table3.3 Nonexistent Ports

Port Symbol	64-pin version	Pins	48-pin version	Pins
PORT0	P02 to P07	6	P01 to P07	7
PORT1	P10 and P12 to P17	7	P12 to P17	6
PORT2	P23 to P27	5	P22 to P27	6
PORT3	P30 to P35	6	P30 to P35	6
PORT4	-	-	P45 to P47	3
PORT5	P50, P51, P55, P56, P57	5	P50, P51, and P54 to P57	6
PORT6	P60 to P63, P66, and P67	6	P60, P61, and P63 to P67	7
PORT7	P77	1	P70, P77	2
PORT8	P80 to P87	8	P80 to P87	8
PORT9	P97	1	P96, P97	2
PORTA	PA0 to PA7	8	PA0 to PA7	8
PORTB	PB7	1	PB7	1
PORTD	PD0 to PD2	3	PD0 to PD2, PD4, and PD6	5
PORTE	PE0, PE1, and PE3 to PE7	7	PE0, PE1, and PE3 to PE7	7
PORTN	PN0 to PN5	6	PN0 to PN5 and PN7	7

3.2.2 Specifying the Number of Pins

In the sample code, use of the 100-pin version of a product is set (PIN_SIZE=100). The sample code also supports the 80-pin, 64-pin, and 48-pin versions of products. To use a device other than the 100-pin version, set the number of pins of the device for PIN_SIZE in r_init_port_initialize.h.

3.3 Clock Settings

3.3.1 Clock Setting Procedure

Table 3.4 shows Clock Setting Procedure. It also describes the processing of each step and shows the default settings configured in the sample code. The sample code configured with the default settings enables the main clock and the PLL module, and disables the HOCO module.

Table 3.4 Clock Setting Procedure

Step	Processing	Description		Setting in Sample Code
1	Setting the main clock (Note 2)	If not used	No setting is required.	The main clock is used.
		If used	Set the drive capability of the main clock in the MOFCR register, set the waiting time until the output of the main clock is supplied to the internal clock in the MOSCWTCR register, and then oscillate the main clock.Then waits until oscillation is stabilized (for the oscillation stabilization time). (Note 1)	
2	Setting the HOCO clock (Note 2)	If not used	The HOCO clock is turned off.	The HOCO clock is not used.
		If used	Sets the HOCO frequency and starts HOCO clock oscillation. Then waits until oscillation is stabilized (for the oscillation stabilization time). (Note 1)	
3	Setting the PLL clock (Note 2)	If not used	No setting is required.	The PLL clock is used.
		If used	Sets the PLL input frequency division ratio and frequency multiplication factor, and then starts PLL clock oscillation. Then waits until oscillation is stabilized (for the oscillation stabilization time). (Note 1)	
4	Setting the clock frequency division ratios	Sets the clock division ratios.		• ICLK, PCLKA, and PCLKC: × 1/2 • PCLKB, PCLKD, and FCLK: × 1/4
5	Switching the system clock	Switches the system clock according to the user system.		Switches to the PLL clock.

Notes: 1. During the wait time, the system checks whether the corresponding bit in the oscillation stabilization flag register (OSCOVFSR) is 1.

2. Whether to use a clock can be changed by changing the setting of the corresponding constant in r_init_clock.h. Change the constant settings as necessary.Refer to section 3.7 for constants.

3.4 Section Configuration

Table 3.5 shows Information of the Section Changed in the Sample Code (r01an6567_rx26t).

For details about how to add, change, or delete sections, refer to the latest version of the RX Family CC-RX Compiler User’s Manual.

Table 3.5 Information of the Section Changed in the Sample Code (r01an6567_rx26t)

Section Name	Type	Address	Description
End_of_RAM	Addition	0000 FFFCh	End address of the on-chip RAM

3.5 File Composition

Table 3.6 shows Files Used in the Sample Code. Files generated by the integrated development environment are not included in this table.

Table 3.6 Files Used in the Sample Code

File Name	Outline	Remarks
main.c	Main processing	
r_init_stop_module.c	Stopping the peripheral modules that are still operating after a reset occurs	
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_rom_cache.c	

3.6 Option-Setting Memory

Table 3.7 shows Status of the Option-Setting Memory Used in the Sample Code. When necessary, set a value suited to the user system.

Table 3.7 Status of the Option-Setting Memory Used in the Sample Code

Symbol	Address	Setting Value	Description
OFS0	0012 0068h to 0012 006Bh	FFFF FFFFh	IWDT is stopped after a reset. 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, linear mode

3.7 Constants

Table 3.8 to 3.11 showsConstants Used in the Sample Code .. Note that the constants in Table 3.11 cannot be changed by the user.

Table 3.12 to Table 3.15 show constants for the various product versions. (The specification is to select a macro definition with #ifdef according to the PIN_SIZE value set in Table 3.10.)

Table 3.8 Constants Used in the Sample Code (User Changeable) (1/4)

Constant Name	Setting Value	Description
SEL_MAIN (Note 1)	B_USE	Main clock enable/disable selection: B_USE: Enable (enables oscillation of the main clock) B_NOT_USE: Disable (disables the main clock)
MAIN_CLOCK_Hz (Note 1)	10000000L	Oscillation frequency of the resonator for the main clock (Hz)
REG_MOFCR (Note 1)	20h	Drive capacity setting of the main clock oscillator (setting value of the MOFCR register)
REG_MOSCWTCR (Note 1)	53h	Setting value of the main clock oscillator wait control register
SEL_PLL (Note 1)	B_USE_PLL_MAIN	PLL clock enable/disable selection: B_USE_PLL_MAIN: Used (with the main clock) B_USE_PLL_HOCO: Used (with the HOCO clock) B_NOT_USE: Not used (PLL clock stopped)
REG_PLLCR (Note 1, Note 2, Note 3)	1710h (HOCO as PLL clock source) 2F00h (Other than above)	PLL input frequency division ratio and frequency multiplication factor settings (setting value of the PLLCR register) 1710h: Clock source: HOCO × 1/1 × 12 2F00h: Clock source: Main clock × 1/1 × 24
REG_SCKCR (Note 1, Note 2, Note 3)	2101 1212h (PLL selected) 0000 0000h (HOCO selected) 0000 0000h (Other than above)	Internal clock frequency division ratio (setting value of the SCKCR register)
REG_VOLSR (Note 1, Note 3)	80h	Setting of the voltage level for VCC when RIIC is used (setting value of the VOLSR register)

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

2. The value to be set differs depending on the clock source of the selected system clock. Refer to the setting value column in Table 3.8 for details.

3. When changing the values set in this register, follow the User’s Manual: Hardware.

Table 3.9 Constants Used in the Sample Code (User Changeable) (2/4)

Constant Name	Setting Value	Description
SEL_HOCO (Note 1)	B_NOT_USE	HOCO clock enable/disable selection: B_USE: Enable (Enables oscillation of the HOCO clock.) B_NOT_USE: Disable (Disables the HOCO clock.)
REG_HOCOCR2 (Note 1)	FREQ_20MHz	HOCO clock frequency selection: FREQ_16MHz: 16 MHz FREQ_18MHz: 18 MHz FREQ_20MHz: 20 MHz
SEL_SYSCLK (Note 1)	CLK_PLL	System clock source selection: CLK_PLL: PLL CLK_HOCO: HOCO CLK_MAIN: Main clock

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

Table 3.10 Constants Used in the Sample Code (User Changeable) (3/4)

Constant Name	Setting Value	Description
MSTP_STATE_DMACDTC (Note 1)	MODULE_STOP_DISABLE	Selection of the module-stop state for DMAC and DTC: MODULE_STOP_DISABLE: Disables state transition. MODULE_STOP_ENABLE: Enables state transition.
MSTP_STATE_RAM (Note 1)	MODULE_STOP_DISABLE	Selection of the module-stop state for RAM: MODULE_STOP_DISABLE: Disables state transition. MODULE_STOP_ENABLE: Enables state transition.
PIN_SIZE (Note 2)	100	Number of pins for the product used

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). It is also necessary to change the port settings that do not exist in the device (package) to be used. Refer to Chapter 3.2 for details.

Table 3.11 Constants Used in the Sample Code (Not User Changeable) (4/4)

Constant Name	Setting Value	Description
B_NOT_USE	0	Disable
B_USE	1	Enable
B_USE_PLL_MAIN	2	The PLL clock is used. (clock source: main clock)
B_USE_PLL_HOCO	3	The PLL clock is 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
REG_SCKCR2	2011h	Peripheral clock (CANFD clock) frequency division ratio (value set in the SCKCR2 register)
MODULE_STOP_ENABLE	1	Transition to the module-stop state is made.
MODULE_STOP_DISABLE	0	Release from the module-stop state

Table 3.12 Constants for 100-Pin Products (PIN_SIZE=100)

Constant Name	Setting Value	Description
DEF_P0PDR	00h	Value set in the direction register for port P0
DEF_P1PDR	00h	Value set in the direction register for port P1
DEF_P2PDR	00h	Value set in the direction register for port P2
DEF_P3PDR	00h	Value set in the direction register for port P3
DEF_P4PDR	00h	Value set in the direction register for port P4
DEF_P5PDR	00h	Value set in the direction register for port P5
DEF_P6PDR	00h	Value set in the direction register for port P6
DEF_P7PDR	00h	Value set in the direction register for port P7
DEF_P8PDR	00h	Value set in the direction register for port P8
DEF_P9PDR	00h	Value set in the direction register for port P9
DEF_PAPDR	00h	Value set in the direction register for port PA
DEF_PBPDR	00h	Value set in the direction register for port PB
DEF_PDPDR	00h	Value set in the direction register for port PD
DEF_PEPDR	00h	Value set in the direction register for port PE
DEF_PNPDR	00h	Value set in the direction register for port PN

Table 3.13 Constants for 80-Pin Products (PIN_SIZE=80)

Constant Name	Setting Value	Description
DEF_P0PDR	00h	Value set in the direction register for port P0
DEF_P1PDR	00h	Value set in the direction register for port P1
DEF_P2PDR	18h	Value set in the direction register for port P2
DEF_P3PDR	0Ch	Value set in the direction register for port P3
DEF_P4PDR	00h	Value set in the direction register for port P4
DEF_P5PDR	00h	Value set in the direction register for port P5
DEF_P6PDR	0Eh	Value set in the direction register for port P6
DEF_P7PDR	00h	Value set in the direction register for port P7
DEF_P8PDR	07h	Value set in the direction register for port P8
DEF_P9PDR	00h	Value set in the direction register for port P9
DEF_PAPDR	17h	Value set in the direction register for port PA
DEF_PBPDR	80h	Value set in the direction register for port PB
DEF_PDPDR	03h	Value set in the direction register for port PD
DEF_PEPDR	23h	Value set in the direction register for port PE
DEF_PNPDR	00h	Value set in the direction register for port PN

Table 3.14 Constants for 64-Pin Products (PIN_SIZE=64)

Constant Name	Setting Value	Description
DEF_P0PDR	00h	Value set in the direction register for port P0
DEF_P1PDR	01h	Value set in the direction register for port P1
DEF_P2PDR	98h	Value set in the direction register for port P2
DEF_P3PDR	0Fh	Value set in the direction register for port P3
DEF_P4PDR	00h	Value set in the direction register for port P4
DEF_P5PDR	23h	Value set in the direction register for port P5
DEF_P6PDR	0Fh	Value set in the direction register for port P6
DEF_P7PDR	00h	Value set in the direction register for port P7
DEF_P8PDR	07h	Value set in the direction register for port P8
DEF_P9PDR	00h	Value set in the direction register for port P9
DEF_PAPDR	3Fh	Value set in the direction register for port PA
DEF_PBPDR	80h	Value set in the direction register for port PB
DEF_PDPDR	07h	Value set in the direction register for port PD
DEF_PEPDR	3Bh	Value set in the direction register for port PE
DEF_PNPDR	00h	Value set in the direction register for port PN

Table 3.15 Constants for 48-Pin Products (PIN_SIZE=48)

Constant Name	Setting Value	Description
DEF_P0PDR	02h	Value set in the direction register for port P0
DEF_P1PDR	00h	Value set in the direction register for port P1
DEF_P2PDR	9Ch	Value set in the direction register for port P2
DEF_P3PDR	0Fh	Value set in the direction register for port P3
DEF_P4PDR	E0h	Value set in the direction register for port P4
DEF_P5PDR	33h	Value set in the direction register for port P5
DEF_P6PDR	3Bh	Value set in the direction register for port P6
DEF_P7PDR	01h	Value set in the direction register for port P7
DEF_P8PDR	07h	Value set in the direction register for port P8
DEF_P9PDR	41h	Value set in the direction register for port P9
DEF_PAPDR	3Fh	Value set in the direction register for port PA
DEF_PBPDR	80h	Value set in the direction register for port PB
DEF_PDPDR	57h	Value set in the direction register for port PD
DEF_PEPDR	3Bh	Value set in the direction register for port PE
DEF_PNPDR	00h	Value set in the direction register for port PN

3.8 Functions

Table 3.16 shows Functions.

Table 3.16 Functions

Function Name	Outline
main	Main Processing
R_INIT_StopModule	Stopping Peripheral Modules Still Operating After a Reset
R_INIT_Port_Initialize	Initialization of nonexistent ports
R_INIT_Clock	Clock 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 functions that stop peripheral modules still operating after a reset, initialize nonexistent ports, and initialize clocks.
Arguments	None
Return Value	None

R_INIT_StopModule	
Outline	Stopping the peripheral modules that are still operating after a reset occurs
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	Initialization of nonexistent ports
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	In the sample code, use of the 100-pin version of a product is set (PIN_SIZE=100). If data is written on a byte basis to the PDR register including nonexistent ports after this function is called, make sure that the direction control bits for the nonexistent ports are set as indicated in 20.5, "Initialization of the Port Direction Register (PDR)", in the User's Manual: Hardware. 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.
Arguments	None
Return Value	None
Remarks	In the sample code, PLL is selected as the system clock.

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 then starts oscillation of the main clock. Then waits until oscillation of the main clock is stabilized (for the oscillation stabilization time).
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 then starts PLL clock oscillation. Then waits until oscillation of the PLL clock is stabilized (for the oscillation stabilization time).
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 then starts HOCO clock oscillation. Then waits until oscillation of the HOCO clock is stabilized (for the oscillation stabilization time).
Arguments	None
Return Value	None

3.10 Flowcharts

3.10.1 Main Processing

Figure 3.1 shows the flowchart of Main Processing.

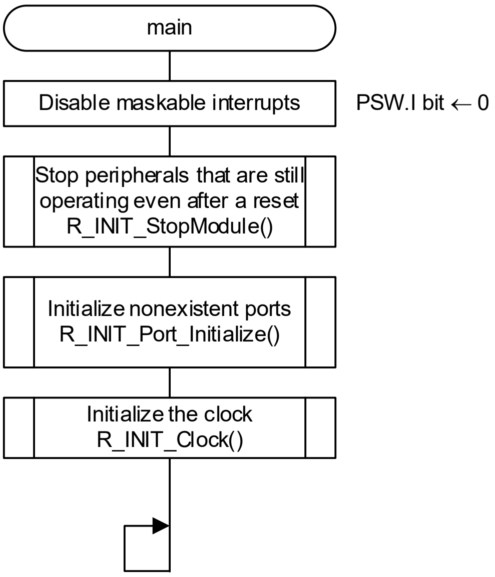
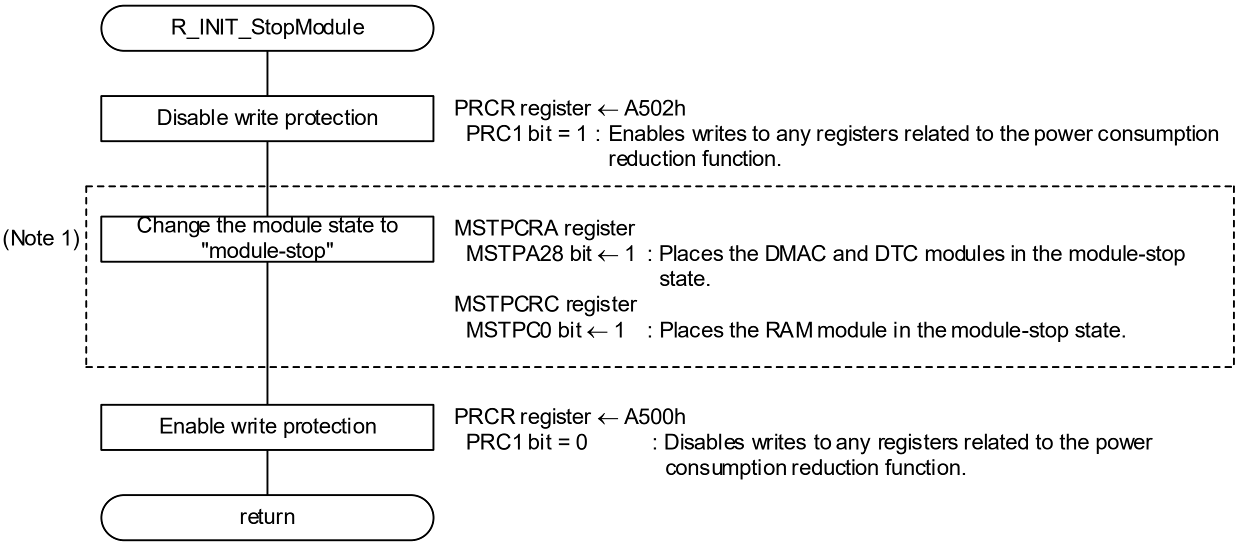

main
Disable maskable interrupts
PSW.I bit ← 0
Stop peripherals that are still operating even after a reset
R_INIT_StopModule()
Initialize nonexistent ports
R_INIT_Port_Initialize()
Initialize the clock
R_INIT_Clock()

Figure 3.1 Main Processing

3.10.2 Stopping Peripheral Modules Still Operating After a Reset

Figure 3.2 shows the flowchart of Stopping Peripheral Modules Still Operating After a Reset.



PRCR register ← A502h
PRC1 bit = 1 : Enables writes to any registers related to the power consumption reduction function.

MSTPCRA register
MSTPA28 bit ← 1 : Places the DMAC and DTC modules in the module-stop state.
MSTPCRC register
MSTPC0 bit ← 1 : Places the RAM module in the module-stop state.

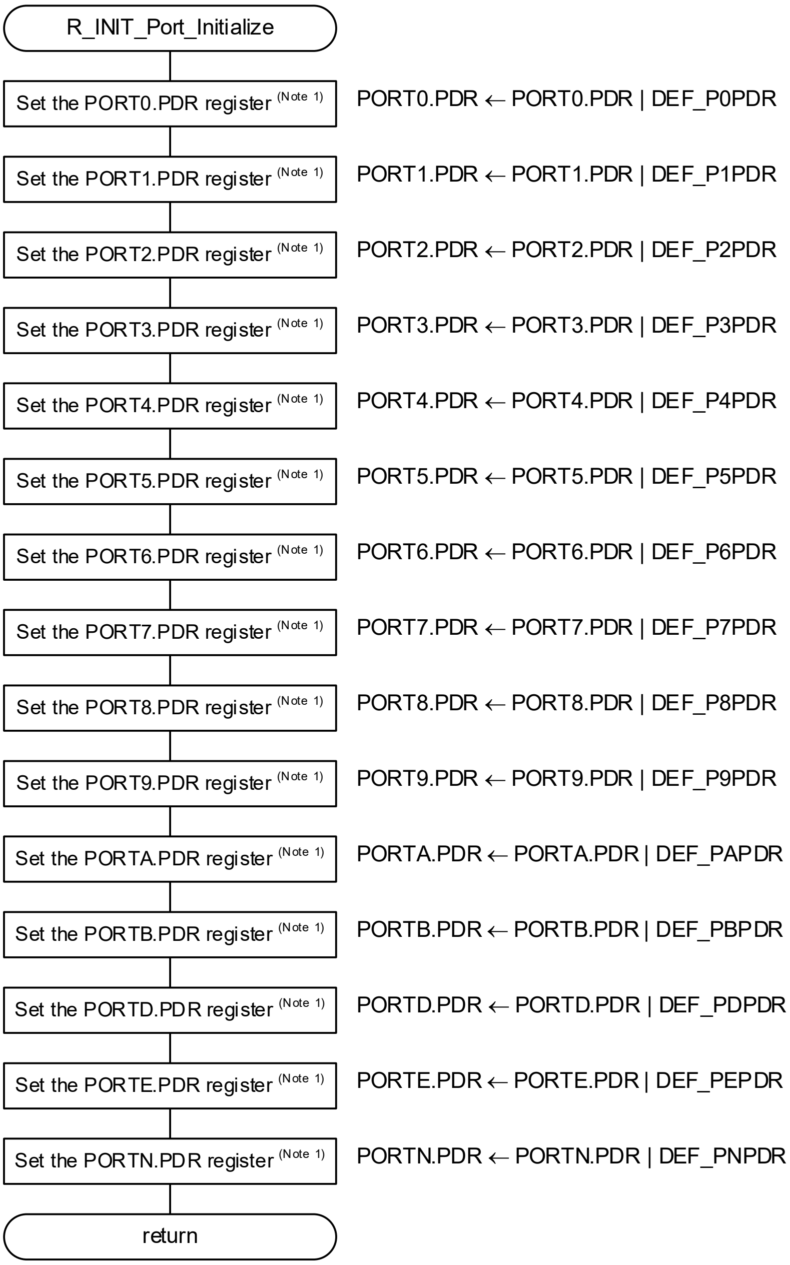
PRCR register ← A500h
PRC1 bit = 0 : Disables writes to any registers related to the power consumption reduction function.

Note 1. In the sample code, no modules enter the module-stop state. For a module to enter the module-stop state, set 1 for the following constant: #define MSTP_STATE_<module-name>

Figure 3.2 Stopping Peripheral Modules Still Operating After a Reset

3.10.3 Initialization of Nonexistent Ports

Figure 3.3 shows the flowchart of Nonexistent Port Initialization.

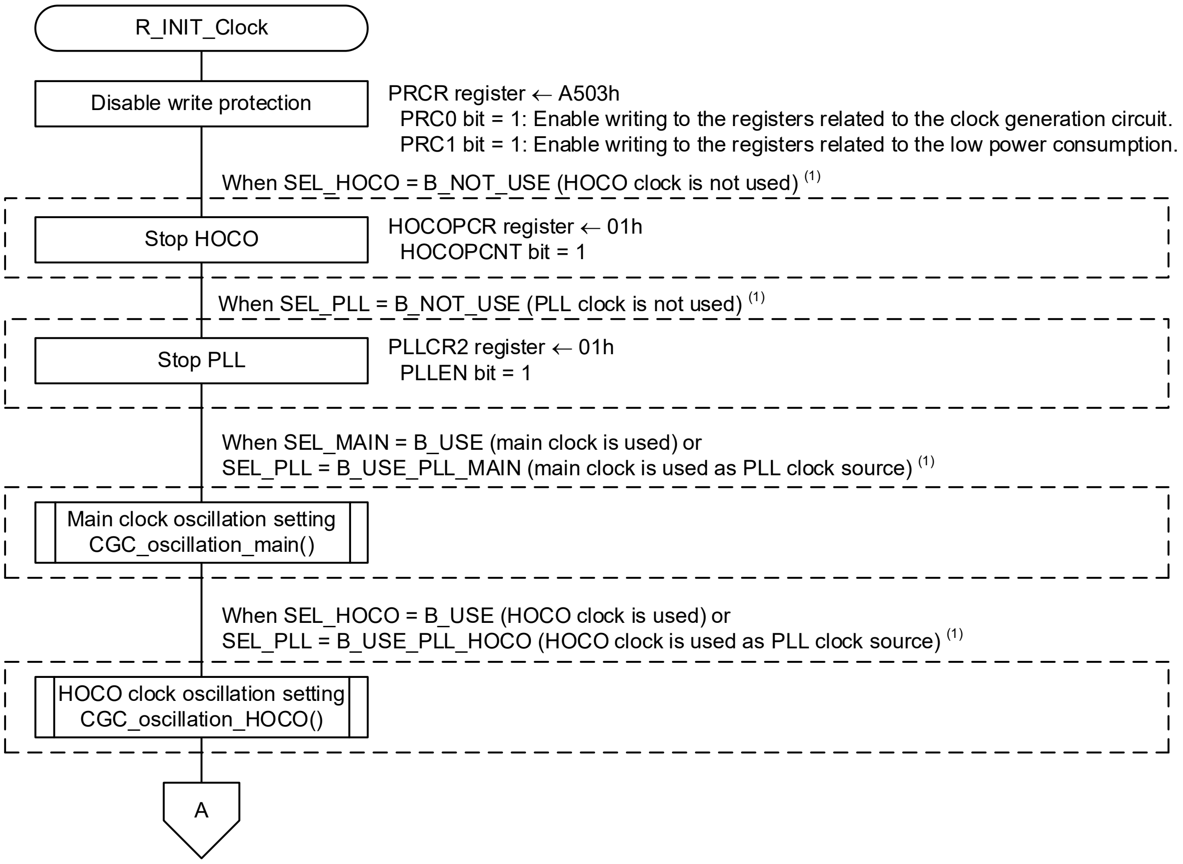

R_INIT_Port_Initialize
Set the PORT0.PDR register (Note 1)
PORT0.PDR ← PORT0.PDR | DEF_P0PDR
Set the PORT1.PDR register (Note 1)
PORT1.PDR ← PORT1.PDR | DEF_P1PDR
Set the PORT2.PDR register (Note 1)
PORT2.PDR ← PORT2.PDR | DEF_P2PDR
Set the PORT3.PDR register (Note 1)
PORT3.PDR ← PORT3.PDR | DEF_P3PDR
Set the PORT4.PDR register (Note 1)
PORT4.PDR ← PORT4.PDR | DEF_P4PDR
Set the PORT5.PDR register (Note 1)
PORT5.PDR ← PORT5.PDR | DEF_P5PDR
Set the PORT6.PDR register (Note 1)
PORT6.PDR ← PORT6.PDR | DEF_P6PDR
Set the PORT7.PDR register (Note 1)
PORT7.PDR ← PORT7.PDR | DEF_P7PDR
Set the PORT8.PDR register (Note 1)
PORT8.PDR ← PORT8.PDR | DEF_P8PDR
Set the PORT9.PDR register (Note 1)
PORT9.PDR ← PORT9.PDR | DEF_P9PDR
Set the PORTA.PDR register (Note 1)
PORTA.PDR ← PORTA.PDR | DEF_PAPDR
Set the PORTB.PDR register (Note 1)
PORTB.PDR ← PORTB.PDR | DEF_PBPDR
Set the PORTD.PDR register (Note 1)
PORTD.PDR ← PORTD.PDR | DEF_PDPDR
Set the PORTE.PDR register (Note 1)
PORTE.PDR ← PORTE.PDR | DEF_PEPDR
Set the PORTN.PDR register (Note 1)
PORTN.PDR ← PORTN.PDR | DEF_PNPDR
return

Note 1. Processing is skipped (omitted during compilation) if all pins in the register are present.

Figure 3.3 Nonexistent Port Initialization

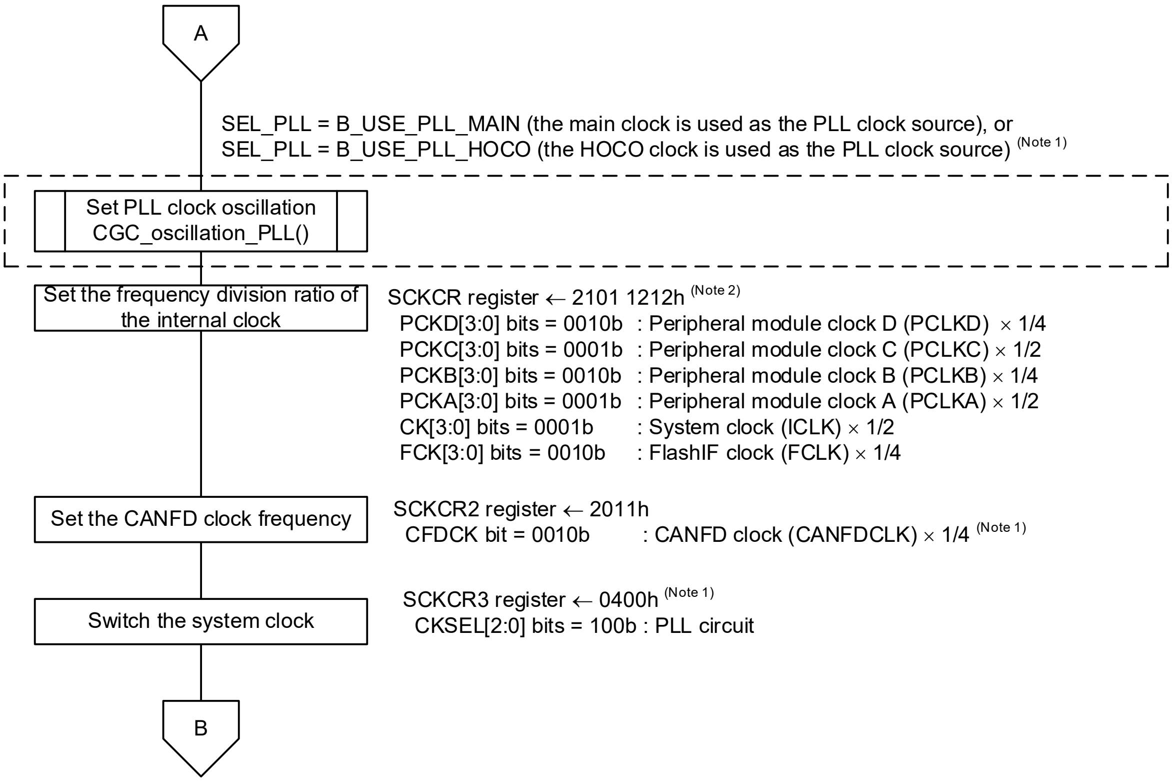
3.10.4 Clock Initialization

Figures 3.4 to 3.6 shows the clock setting flowchart.


R_INIT_Clock
Disable write protection
PRCR register ← A503h
PRC0 bit = 1: Enable writing to the registers related to the clock generation circuit.
PRC1 bit = 1: Enable writing to the registers related to the low power consumption.
When SEL_HOCO = B_NOT_USE (HOCO clock is not used) (1)
Stop HOCO
HOCOPCR register ← 01h
HOCOPCNT bit = 1
When SEL_PLL = B_NOT_USE (PLL clock is not used) (1)
Stop PLL
PLLCR2 register ← 01h
PLLEN bit = 1
When SEL_MAIN = B_USE (main clock is used) or
SEL_PLL = B_USE_PLL_MAIN (main clock is used as PLL clock source) (1)
Main clock oscillation setting
CGC_oscillation_main()
When SEL_HOCO = B_USE (HOCO clock is used) or
SEL_PLL = B_USE_PLL_HOCO (HOCO clock is used as PLL clock source) (1)
HOCO clock oscillation setting
CGC_oscillation_HOCO()
A

Note 1.Change the SEL_MAIN, SEL_PLL, and SEL_HOCO constant settings in "r_init_clock.h" according to the user system.
Refer to Tables 3.8 and 3.9 for details.

Figure 3.4 Clock Initialization (1/3)


A
Set PLL clock oscillation
CGC_oscillation_PLL()
Set the frequency division ratio of the internal clock
Set the CANFD clock frequency
Switch the system clock
B

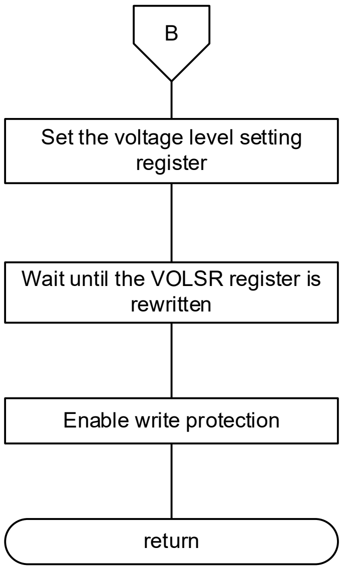
SEL_PLL = B_USE_PLL_MAIN (the main clock is used as the PLL clock source), or
SEL_PLL = B_USE_PLL_HOCO (the HOCO clock is used as the PLL clock source) (Note 1)

```
SCKCR register ← 2101 1212h (Note 2)
  PCKD[3:0] bits = 0010b  : Peripheral module clock D (PCLKD)  × 1/4
  PCKC[3:0] bits = 0001b  : Peripheral module clock C (PCLKC) × 1/2
  PCKB[3:0] bits = 0010b  : Peripheral module clock B (PCLKB) × 1/4
  PCKA[3:0] bits = 0001b  : Peripheral module clock A (PCLKA) × 1/2
  CK[3:0] bits = 0001b    : System clock (ICLK) × 1/2
  FCK[3:0] bits = 0010b   : FlashIF clock (FCLK) × 1/4

SCKCR2 register ← 2011h
  CFDCK bit = 0010b       : CANFD clock (CANFDCLK) × 1/4 (Note 1)

SCKCR3 register ← 0400h (Note 1)
  CKSEL[2:0] bits = 100b : PLL circuit
```

Notes 1. Change the REG_SCKCR2 and REG_SCKCR3 constant settings in "r_init_stop_module.h" according to the user system.

2. The value that is set differs depending on the system clock selected by the REG_SCKCR Constant in "r_init_clock.h" differs.

Figure 3.5 Clock Initialization (2/3)


B
Set the voltage level setting register
Wait until the VOLSR register is rewritten
Enable write protection
return

VOLSR register ← 80h (Note 1)
RICVLS bit = 1 0: VCC ≥ 4.5 V
1: VCC < 4.5 V

Reads the VOLSR register to confirm that it is rewritten.

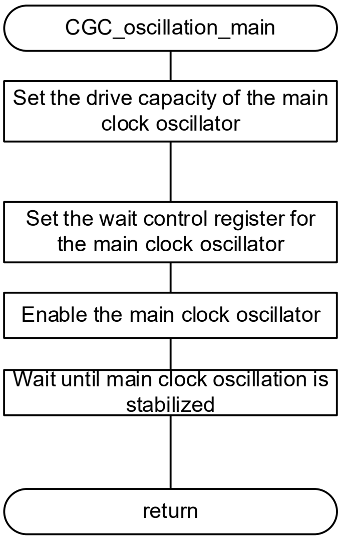
PRCR register ← A500h
PRC0 bit = 0 : Prohibits writes to any registers related to clock generation circuits.
PRC1 bit = 0 : Prohibits writes to any registers related to the power consumption reduction function.

Note 1. Change the REG_VOLSR comstant settings in "r_init_clock.h" according to the user system.Refer toTable 3.8 for details.

Figure 3.6 Clock Initialization (3/3)

3.10.5 Main Clock Oscillation Setting

Figure 3.7 shows the flowchart of Main Clock Oscillation Setting.


CGC_oscillation_main
Set the drive capacity of the main clock oscillator
Set the wait control register for the main clock oscillator
Enable the main clock oscillator
Wait until main clock oscillation is stabilized
return

MOFCR register ← 20h (Note 1)
MODRV2[1:0] bits = 10b : 8.1 to 16 MHz
MOSEL bit = 0 : Resonator

MOSCWTCR register ← 53h (Note 1)
MSTS[7:0] bits = 01010011b : When the max. frequency of fLOCO is 264 kHz, the wait time is approx.10 ms. (Note 2)

MOSCCR register ← 00h
MOSTP bit = 0 : Enables the main clock oscillator.

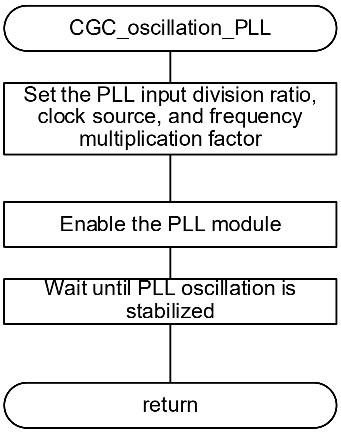
Reads the OSCOVFSR register.
MOOVF bit: 0: MOSTP = 1 (stopped) or waiting until oscillation is stabilized
1: Oscillation is stabilized and can be used as the system clock.

Notes 1. Change the REG_MOFCR and REG_MOSCWTCR constant settings in "r_init_clock.h" according to the user system.
2. In this application note, the initial value of the register is used.

Figure 3.7 Main Clock Oscillation Setting

3.10.6 PLL Clock Oscillation Setting

Figure 3.8 shows the flowchart of PLL Clock Oscillation Setting.


CGC_oscillation_PLL
Set the PLL input division ratio, clock source, and frequency multiplication factor
Enable the PLL module
Wait until PLL oscillation is stabilized
return

PLLCR register ← 2F00h (Note 1)
PLIDIV[1:0] bits = 00b : PLL input frequency division ratio = ×1/1
PLLSRCSEL bit = 0 : Main clock is selected as the PLL clock source
STC[5:0] bits = 101111b : Frequency multiplication factor = ×24

PLLCR2 register ← 00h
PLLEN bit = 0 : Enables the PLL module.

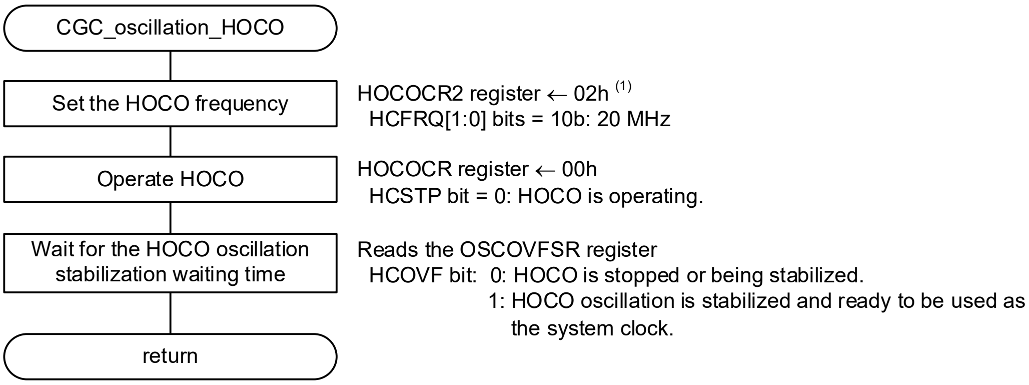
Reads the OSCOVFSR register
PLOVF bit: 0: PLL is stopped or waiting until oscillation is stabilized.
1: Oscillation is stabilized and can be used as the system clock.

Note1. Change the REG_PLLCR constant settings in "r_init_clock.h" according to the user system. Refer to Table 3.8 for details.

Figure 3.8 PLL Clock Oscillation Setting

3.10.7 HOCO Clock Oscillation Setting

Figure 3.9 shows the flowchart of HOCO Clock Oscillation Setting.


CGC_oscillation_HOCO
Set the HOCO frequency
HOCOCR2 register ← 02h (1)
HCFRQ[1:0] bits = 10b: 20 MHz
Operate HOCO
HOCOCR register ← 00h
HCSTP bit = 0: HOCO is operating.
Wait for the HOCO oscillation stabilization waiting time
Reads the OSCOVFSR register
HCOVF bit: 0: HOCO is stopped or being stabilized.
1: HOCO oscillation is stabilized and ready to be used as the system clock.
return

Note 1. Change the REG_HOCOCR2 constant settings in "r_init_clock.h" according to the user system. Refer to Table 3.9 for details.

Figure 3.9 HOCO Clock Oscillation Setting

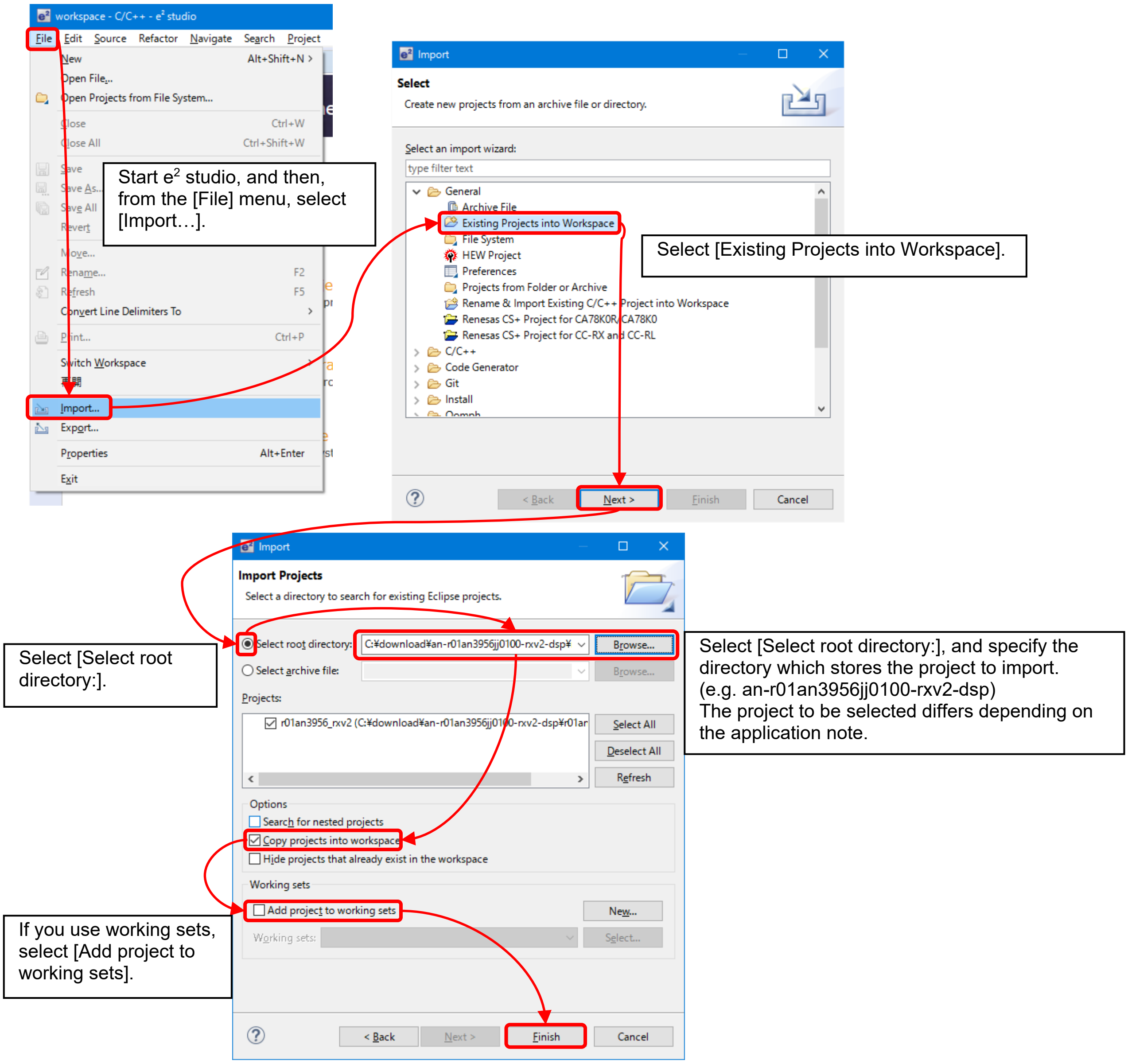
4. Importing a Project

The sample code is provided in the form of an e^2 studio project. This section describes the procedures for importing a project into the e^2 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 e^2 studio

If you use the project with e^2 studio, use the procedure described below to import the project to e^2 studio.

(The windows and dialogs shown in the following procedure may slightly differ from the actually displayed ones, depending on the version of e^2 studio you use.)



Start e^2 studio, and then, from the [File] menu, select [Import…].

Select [Existing Projects into Workspace].

Select [Select root directory:].

Select [Select root directory:], and specify the directory which stores the project to import.
(e.g. an-r01an3956jj0100-rxv2-dsp)
The project to be selected differs depending on the application note.

If you use working sets, select [Add project to working sets].

Select root directory: C:¥download¥an-r01an3956jj0100-rxv2-dsp¥

Projects:
☑ r01an3956_rxv2 (C:¥download¥an-r01an3956jj0100-rxv2-dsp¥r01an

Options
☐ Search for nested projects
☑ Copy projects into workspace
☐ Hide projects that already exist in the workspace

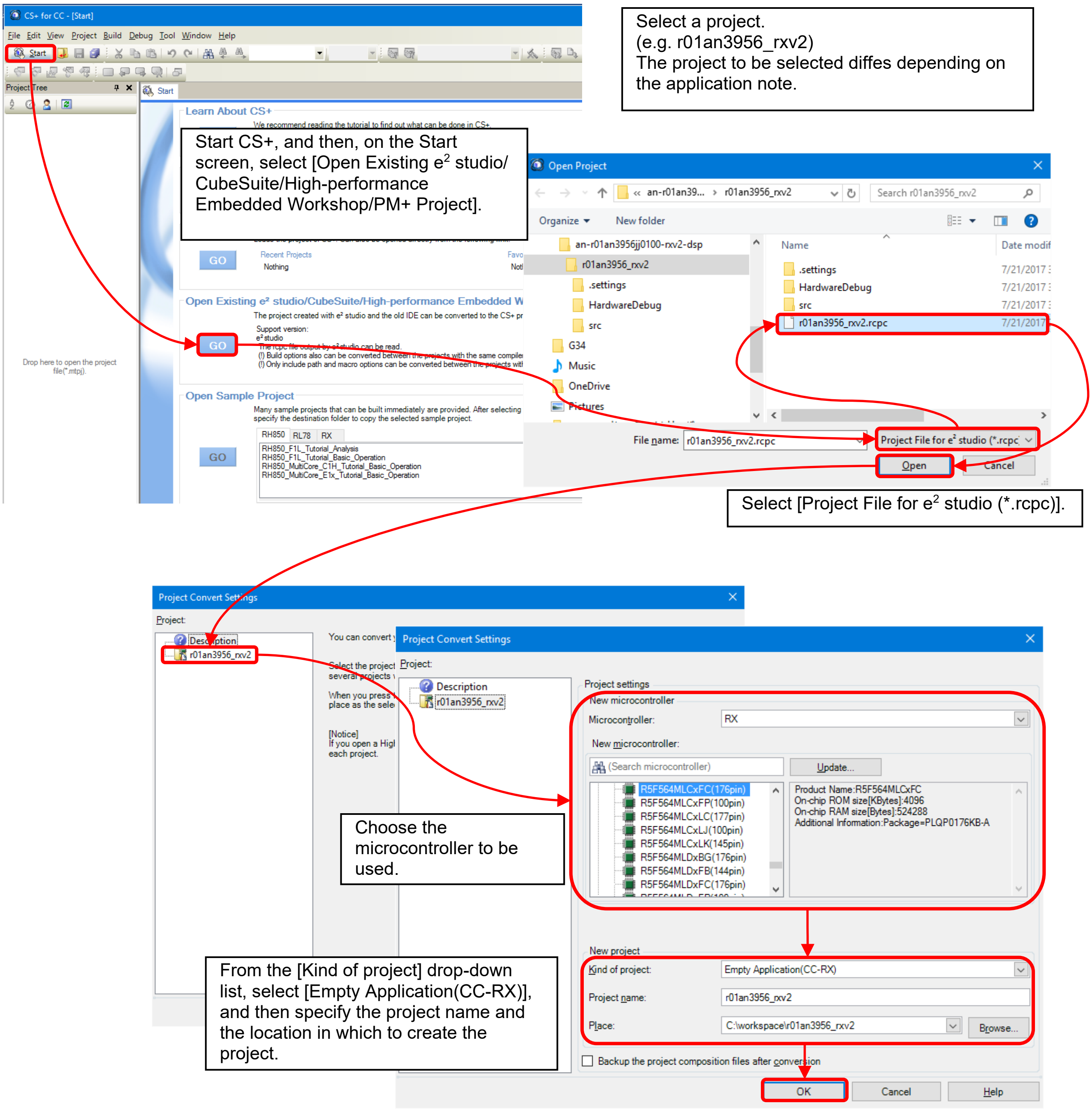
Working sets
☐ Add project to working sets

Figure 4.1 Importing a Project into the e^2 studio

4.2 Importing a Project into CS+

If you use the project with CS+, use the procedure described below to import the project to CS+.

(The windows and dialogs shown in the following procedure may slightly differ from the actually displayed ones, depending on the version of CS+ you use.)



Start CS+, and then, on the Start screen, select [Open Existing e^2 studio/ CubeSuite/High-performance Embedded Workshop/PM+ Project].

Select a project.
(e.g. r01an3956_rxv2)
The project to be selected diffes depending on the application note.

Select [Project File for e^2 studio (*.rcpc)].

Choose the microcontroller to be used.

From the [Kind of project] drop-down list, select [Empty Application(CC-RX)], and then specify the project name and the location in which to create the project.

File name: r01an3956_rxv2.rcpc

Microcontroller: RX

Kind of project: Empty Application(CC-RX)

Project name: r01an3956_rxv2

Place: C:\workspace\r01an3956_rxv2

□ Backup the project composition files after conversion

Figure 4.2 Importing a Project into CS+

5. Sample Code

Download the desired sample code from the Renesas Electronics website.

6. Reference Documents

User’s Manual: Hardware

RX26T Group User’s Manual: Hardware (R01UH0979)
(The latest versions of these documents are available at the Renesas Electronics website.)

Technical Update/Technical News

(The latest information is available at the Renesas Electronics website.)

User’s Manual: Development environment

RX Family CC-RX Compiler User’s Manual (R20UT3248)
(The latest versions of these documents are available at the Renesas Electronics website.)

Revision History

Rev.	Date	Description	
		Page	**Summary**
1.00	Apr.27.23	—	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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. 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.5.0-1 October 2020)

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/.