RX63N Group, RX631 Group

Initial Setting

Abstract

This document describes settings required after a reset such as clock settings, stop processing for active peripheral functions after a reset, or nonexistent port initialization according to the conditions selected in the header file.

Products

- RX63N Group 177-pin and 176-pin packages with a ROM size between 768 KB and 2 MB
- RX63N Group 145-pin and 144-pin packages with a ROM size between 768 KB and 2 MB
- RX63N Group 100-pin package with a ROM size between 768 KB and 2 MB
- RX631 Group 177-pin and 176-pin packages with a ROM size between 256 KB and 2 MB
- RX631 Group 145-pin and 144-pin packages with a ROM size between 256 KB and 2 MB
- RX631 Group 100-pin package with a ROM size between 256 KB and 2 MB
- RX631 Group 64-pin and 48-pin packages with a ROM size between 256 KB and 512 KB

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 ......................................................................................................................... 3
   1.1 Stopping Peripheral Functions Operating after a Reset ................................................... 3
   1.2 Configuring Nonexistent Ports .................................................................................... 3
   1.3 Setting Clocks ............................................................................................................... 4
      1.3.1 Overview ................................................................................................................. 4
      1.3.2 Clock Specifications Used in the Sample Code .................................................. 4

2. Operation Confirmation Conditions ......................................................................................... 5

3. Software ................................................................................................................................. 6
   3.1 Stop Processing for Active Peripheral Functions after a Reset .................................. 6
   3.2 Nonexistent Port Initialization ....................................................................................... 7
      3.2.1 Overview ................................................................................................................. 7
      3.2.2 Selecting the Number of Pins .............................................................................. 8
   3.3 Clock Settings ................................................................................................................. 9
      3.3.1 Clock Setting Procedure ....................................................................................... 9
      3.3.2 Sub-Clock Setting Patterns ................................................................................ 10
      3.3.3 Oscillation Stabilization Wait Time for Each Clock ............................................ 11
         3.3.3.1 Main Clock Oscillation Stabilization Wait Time .......................................... 11
         3.3.3.2 PLL Clock Oscillation Stabilization Wait Time ........................................... 12
         3.3.3.3 Sub-Clock Oscillation Stabilization Wait Time (Except 64-Pin Packages) ... 13
         3.3.3.4 Sub-Clock Oscillation Stabilization Wait Time (for 64-Pin Packages) ....... 14
   3.4 File Composition .......................................................................................................... 15
   3.5 Option-Setting Memory ................................................................................................. 15
   3.6 Constants ......................................................................................................................... 16
   3.7 Functions ......................................................................................................................... 20
   3.8 Function Specifications ................................................................................................. 21
   3.9 Flowcharts ...................................................................................................................... 26
      3.9.1 Main Processing ...................................................................................................... 26
      3.9.2 Stop Processing for Active Peripheral Functions after a Reset ......................... 26
      3.9.3 Nonexistent Port Initialization ............................................................................ 27
      3.9.4 Clock Initialization ............................................................................................... 28
      3.9.5 Main Clock Oscillation Setting .......................................................................... 29
      3.9.6 PLL Clock Oscillation Setting .......................................................................... 29
      3.9.7 HOCO Clock Oscillation Setting ...................................................................... 30
      3.9.8 Sub-Clock Setting ............................................................................................... 31
      3.9.9 Wait Processing .................................................................................................... 36

4. Appendices ............................................................................................................................ 37
   4.1 Clock Oscillation Stabilization Wait Time ..................................................................... 37
      4.1.1 When Operating PLL before Main Clock Oscillation Stabilizes ....................... 37
      4.1.2 When the Sub-Clock Oscillator Start-Up Time is Greater than the Sub-Clock Oscillation Stabilization Wait Offset Time .................................................. 38

5. Sample Code .......................................................................................................................... 39

6. Reference Documents ............................................................................................................ 39
1. Specifications

In the sample code, peripheral functions operating after a reset are stopped, and nonexistent port and clock settings are configured. The application note assumes processing at power-on (cold start).

1.1 Stopping Peripheral Functions Operating after a Reset

Some peripheral functions operate at power-on, and the module-stop function is disabled for some. These include the DMAC, DTC, EXDMAC, RAM0, RAM1. Although the sample code includes processing for stopping these peripheral functions, it is not executed in the sample code. Change the oscillation parameters as required to execute processing.

Also the RTC needs to be stopped when it is not used. The setting to stop the RTC is performed in the clock setting part of the sample code.

1.2 Configuring Nonexistent Ports

Ports which are not connected to pins must be set as output for products with less than 176 pins. In the sample code, initial values are set for 176-pin products. Change the value according to the product used.
1.3 Setting Clocks

1.3.1 Overview
The clock setting procedure is as follows:

1. Main clock setting
2. Sub-clock setting
3. PLL clock setting
4. System clock switching

Processing of the sub-clock setting differs depending on sub-clock usage including processing when the sub-clock is not used. Five setting patterns are introduced in the application note (see section 3.3.2. for more information).

In the sample code, the PLL clock is used as the system clock without using the sub-clock.

1.3.2 Clock Specifications Used in the Sample Code
Table 1.1 lists the Clock Specifications Used in the Sample Code. Values such as the oscillation stabilization wait time are calculated using values listed on Table 1.1.

Table 1.2 lists the Peripheral Function and Its Application.

<table>
<thead>
<tr>
<th>Clock</th>
<th>Oscillation Frequency</th>
<th>Oscillation Start-Up Time</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal/ceramic resonator for the main clock</td>
<td>12 MHz</td>
<td>4.2 ms (2)</td>
<td></td>
</tr>
<tr>
<td>Crystal for the sub-clock</td>
<td>32.768 kHz (1)</td>
<td>1.3 sec. (2)</td>
<td>For standard clock loads</td>
</tr>
<tr>
<td>PLL clock</td>
<td>192 MHz (main clock divided by 1 and multiplied by 16)</td>
<td>Maximum of 500 µs (3)</td>
<td></td>
</tr>
</tbody>
</table>

Notes:
1. Sub-clock oscillation is disabled in the sample code.
2. The start-up time of a crystal/ceramic resonator differs depending on the wiring pattern, conditions of oscillation parameters, and other settings in the user system. Ask the crystal/ceramic resonator manufacturer to evaluate the user system and provide an appropriate start-up time.

<table>
<thead>
<tr>
<th>Peripheral Function, channel 0 (CMT0)</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare match timer</td>
<td>Measuring the clock oscillation stabilization wait time (1)</td>
</tr>
</tbody>
</table>

Note:
1. When using OS, select a channel for a timer that is not being used by OS.
2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU used</td>
<td>R5F563NBDDFC (RX63N Group)</td>
</tr>
<tr>
<td>Operating frequencies</td>
<td>- Main clock: 12 MHz</td>
</tr>
<tr>
<td></td>
<td>- Sub-clock: 32.768 kHz (stopped in patterns A and B, and a 48-pin package)</td>
</tr>
<tr>
<td></td>
<td>- PLL: 192 MHz (main clock divided by 1 and multiplied by 16)</td>
</tr>
<tr>
<td></td>
<td>- HOCo: Stopped</td>
</tr>
<tr>
<td></td>
<td>- System clock (ICLK): 96 MHz (PLL divided by 2)</td>
</tr>
<tr>
<td></td>
<td>- Peripheral module clock A (PCLKA): 96 MHz (PLL divided by 2)</td>
</tr>
<tr>
<td></td>
<td>- Peripheral module clock B (PCLKB): 48 MHz (PLL divided by 4)</td>
</tr>
<tr>
<td></td>
<td>- External bus clock (BCLK): 48 MHz (PLL divided by 4)</td>
</tr>
<tr>
<td></td>
<td>- FlashIF clock (FCLK): 48 MHz (PLL divided by 4)</td>
</tr>
<tr>
<td></td>
<td>- IEBUS clock (IECLK): 48 MHz (PLL divided by 4)</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>3.3 V</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>Renesas Electronics Corporation</td>
</tr>
<tr>
<td></td>
<td>High-performance Embedded Workshop Version 4.09.01</td>
</tr>
<tr>
<td>C compiler</td>
<td>Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.1.02 Release 01</td>
</tr>
<tr>
<td></td>
<td>Compile options</td>
</tr>
<tr>
<td></td>
<td>-cpu=rx600 -output=&quot;$(CONFIGDIR)$(FILELEAF).obj&quot; -debug -nologo</td>
</tr>
<tr>
<td></td>
<td>(The default setting is used in the integrated development environment.)</td>
</tr>
<tr>
<td>iodefine.h version</td>
<td>Version 1.6A</td>
</tr>
<tr>
<td>Endian</td>
<td>Little endian</td>
</tr>
<tr>
<td>Operating mode</td>
<td>Single-chip mode</td>
</tr>
<tr>
<td>Processor mode</td>
<td>Supervisor mode</td>
</tr>
<tr>
<td>Sample code version</td>
<td>Version 1.10</td>
</tr>
<tr>
<td>Board used</td>
<td>Renesas Starter Kit+ for RX63N (product part no.: R0K50563NC000BE)</td>
</tr>
</tbody>
</table>
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 Active Peripheral Functions after a Reset

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

The module-stop state is canceled after a reset only for modules listed in the Table 3.1. To enter the module-stop state, set the module stop bit to 1 (transition to the module-stop state is made). Power consumption can be reduced by entering the module-stop state.

In the sample code, set the MSTP_STATE_ “target module” constant to 0 (MODULE_STOP_DISABLE), so 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 lists the Peripheral Modules whose Module-Stop States are Canceled after a Reset.

<table>
<thead>
<tr>
<th>Peripheral Module</th>
<th>Module Stop Bit</th>
<th>Value after a Reset</th>
<th>Value when not Using the Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMAC/DTC</td>
<td>MSTPCRA.MSTPA28 bit</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>EXDMAC</td>
<td>MSTPCRA.MSTPA29 bit</td>
<td>(module-stop state is canceled)</td>
<td>(transition to the module-stop state is made)</td>
</tr>
<tr>
<td>RAM0</td>
<td>MSTPCRC.MSTPC0 bit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAM1</td>
<td>MSTPCRC.MSTPC1 bit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.2 Nonexistent Port Initialization

3.2.1 Overview

When using a product with less than 176 pins, set the corresponding bits of nonexistent ports in the PDR register to 1 (output). After the nonexistent port initialization function is called, when writing in byte units to the PDR registers or PODR registers which have nonexistent ports, set the corresponding bits for nonexistent ports as follows: set the I/O select bits in the PDR registers to 1 and set the output data store bits in the PODR registers to 0.

Table 3.2 and Table 3.3 list Nonexistent Ports.

<table>
<thead>
<tr>
<th>Port Symbol</th>
<th>144-Pin and 145-Pin Packages</th>
<th>No. of Pins</th>
<th>100-Pin Package</th>
<th>No. of Pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORT0</td>
<td>—</td>
<td>—</td>
<td>P00 to P03</td>
<td>4</td>
</tr>
<tr>
<td>PORT1</td>
<td>P10 and P11</td>
<td>2</td>
<td>P10 and P11</td>
<td>2</td>
</tr>
<tr>
<td>PORT2</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PORT3</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PORT4</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PORT5</td>
<td>P57</td>
<td>1</td>
<td>P56 and P57</td>
<td>2</td>
</tr>
<tr>
<td>PORT6</td>
<td>—</td>
<td>—</td>
<td>P60 to P67</td>
<td>8</td>
</tr>
<tr>
<td>PORT7</td>
<td>—</td>
<td>—</td>
<td>P70 to P77</td>
<td>8</td>
</tr>
<tr>
<td>PORT8</td>
<td>P84 and P85</td>
<td>2</td>
<td>P80 to P87</td>
<td>8</td>
</tr>
<tr>
<td>PORT9</td>
<td>P94 to P97</td>
<td>4</td>
<td>P90 to P97</td>
<td>8</td>
</tr>
<tr>
<td>PORTA</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PORTB</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PORTC</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PORTD</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PORTE</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PORTF</td>
<td>PF0 to PF4</td>
<td>5</td>
<td>PF0 to PF5</td>
<td>6</td>
</tr>
<tr>
<td>PORTG</td>
<td>PG0 to PG7</td>
<td>8</td>
<td>PG0 to PG7</td>
<td>8</td>
</tr>
<tr>
<td>PORTJ</td>
<td>—</td>
<td>—</td>
<td>PJ5</td>
<td>1</td>
</tr>
</tbody>
</table>
### Table 3.3 Nonexistent Ports (2/2)

<table>
<thead>
<tr>
<th>Port Symbol</th>
<th>64-Pin Package</th>
<th>No. of Pins</th>
<th>48-Pin Package</th>
<th>No. of Pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORT0</td>
<td>P00 to P03, P07</td>
<td>5</td>
<td>P00 to P03, P05, P07</td>
<td>6</td>
</tr>
<tr>
<td>PORT1</td>
<td>P10 to P13</td>
<td>4</td>
<td>P10 to P13</td>
<td>4</td>
</tr>
<tr>
<td>PORT2</td>
<td>P20 to P25</td>
<td>6</td>
<td>P20 to P25</td>
<td>6</td>
</tr>
<tr>
<td>PORT3</td>
<td>P32 to P34</td>
<td>3</td>
<td>P32 to P34</td>
<td>3</td>
</tr>
<tr>
<td>PORT4</td>
<td>P45 and P47</td>
<td>2</td>
<td>P43 to P45, P47</td>
<td>4</td>
</tr>
<tr>
<td>PORT5</td>
<td>P50 to P53, P56, P57</td>
<td>6</td>
<td>P50 to P57</td>
<td>8</td>
</tr>
<tr>
<td>PORT6</td>
<td>P60 to P67</td>
<td>8</td>
<td>P60 to P67</td>
<td>8</td>
</tr>
<tr>
<td>PORT7</td>
<td>P70 to P77</td>
<td>8</td>
<td>P70 to P77</td>
<td>8</td>
</tr>
<tr>
<td>PORT8</td>
<td>P80 to P87</td>
<td>8</td>
<td>P80 to P87</td>
<td>8</td>
</tr>
<tr>
<td>PORT9</td>
<td>P90 to P97</td>
<td>8</td>
<td>P90 to P97</td>
<td>8</td>
</tr>
<tr>
<td>PORTA</td>
<td>PA2, PA5, PA7</td>
<td>3</td>
<td>PA0, PA2, PA5, PA7</td>
<td>4</td>
</tr>
<tr>
<td>PORTB</td>
<td>PB2, PB4</td>
<td>2</td>
<td>PB2, PB4, PB6, PB7</td>
<td>4</td>
</tr>
<tr>
<td>PORTC</td>
<td>PC0, PC1</td>
<td>2</td>
<td>PC0 to PC3</td>
<td>4</td>
</tr>
<tr>
<td>PORTD</td>
<td>PD0 to PD7</td>
<td>8</td>
<td>PD0 to PD7</td>
<td>8</td>
</tr>
<tr>
<td>PORTE</td>
<td>PE6, PE7</td>
<td>2</td>
<td>PE0, PE5 to PE7</td>
<td>4</td>
</tr>
<tr>
<td>PORTF</td>
<td>PF0 to PF5</td>
<td>6</td>
<td>PF0 to PF5</td>
<td>6</td>
</tr>
<tr>
<td>PORTG</td>
<td>PG0 to PG7</td>
<td>8</td>
<td>PG0 to PG7</td>
<td>8</td>
</tr>
<tr>
<td>PORTJ</td>
<td>PJ3, PJ5</td>
<td>2</td>
<td>PJ3, PJ5</td>
<td>2</td>
</tr>
</tbody>
</table>

### 3.2.2 Selecting the Number of Pins

The number of pins in the sample code is set for the 176-pin package (PIN_SIZE=176). This application note covers 177-pin, 176-pin, 145-pin, 144-pin, 100-pin, 64-pin, and 48-pin packages. When using products with less than 176 pins, change PIN_SIZE in r_init_non_existent_port.h to the number of pins on the package.
3.3 Clock Settings

3.3.1 Clock Setting Procedure

Table 3.4 lists the Clock Setting with each processing and setting in the sample code.

In procedure 2. Sub-clock and RTC configuration, select a pattern from Table 3.5 according to the user system.

Steps 1 to 6 are all performed in the sample code. In processing, operate the main clock, PLL, and HOCO. Set the main clock as the RTC count source, switch the system clock to HOCO, and stop the sub-clock in the sub-clock setting (pattern A). Then switch the system clock to PLL and stop HOCO.

Table 3.4 Clock Setting Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Processing</th>
<th>Details</th>
<th>Setting in the Sample Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Main clock oscillation setting</td>
<td>Set the MOSCWTCR register with a wait time until the main clock output is provided to the internal clock, and enable main clock oscillation. Then wait for the main clock oscillation stabilization wait time (1) by software.</td>
<td>Main clock oscillator: Operating</td>
</tr>
<tr>
<td>2</td>
<td>Sub-clock and RTC configuration</td>
<td>Select the sub-clock setting pattern from five patterns listed in Table 3.5 according to the user system and configure the settings accordingly. - Patterns A and B, and the pattern for a 48-pin package Operate HOCO, set the RTC count source, switch the system clock to HOCO, and then configure the sub-clock and RTC. - Patterns C, D and E Set the RTC count source, and configure the sub-clock and RTC.</td>
<td>Sub-clock oscillator: Stopped (pattern A)</td>
</tr>
<tr>
<td>3</td>
<td>PLL oscillation setting (2)</td>
<td>Set the PLL input frequency division ratio and frequency multiplication factor, set the PLLWTCR register with a wait time until the PLL clock output is provided to the internal clock, and enable PLL clock oscillation. Then wait for the PLL clock oscillation stabilization wait time (1) by software.</td>
<td>PLL: Operating</td>
</tr>
<tr>
<td>4</td>
<td>Clock division ratio setting (3)</td>
<td>Set the clock division ratio. - ICLK,PCLKA: Divided by 2 - PCLKB, BCLK, FCLK.IECLK: Divided by 4 - SDCLK, BCLK: Stopped - UCLK: Not used</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Switching the system clock</td>
<td>Switch the system clock according to the user system.</td>
<td>Clock switched to: PLL</td>
</tr>
<tr>
<td>6</td>
<td>HOCO setting</td>
<td>Set HOCO to be operating or stopped. When not using the HOCO clock, power consumption can be reduced by turning off the HOCO power supply.</td>
<td>- HOCO: Stopped - HOCO power supply: OFF</td>
</tr>
</tbody>
</table>

Notes:

1. Refer to 3.3.3 Oscillation Stabilization Wait Time for Each Clock for details on the oscillation stabilization wait time.
2. When not using PLL, the PLL clock setting is not necessary.
3. Do not set divide-by-1 and divide-by-2 when selecting the main clock as the system clock.
3.3.2 Sub-Clock Setting Patterns

The following five patterns are provided for step 2 in Table 3.4.

A. Neither the sub-clock nor the RTC is used. Or the package used is a 48-pin package.
B. The sub-clock is not used and the RTC uses the main clock as the count source.
C. The sub-clock is used as the system clock, and the RTC is not used.
D. The sub-clock is used as the RTC count source.
E. The sub-clock is used as both the system clock and RTC count source.

Table 3.5 lists the Sub-Clock Setting Patterns. Select a pattern from patterns A to E according to the user system. For a 48-pin package, select pattern A.

Table 3.5 Sub-Clock Setting Patterns

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Sub-Clock</th>
<th>RTC</th>
<th>PCLKB (1) when Setting the Sub-Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Crystal Usage</td>
<td>System Clock</td>
<td>Usage</td>
</tr>
<tr>
<td>A</td>
<td>Not used</td>
<td>—</td>
<td>Not used</td>
</tr>
<tr>
<td>B</td>
<td>Not used</td>
<td>—</td>
<td>Used</td>
</tr>
<tr>
<td>C</td>
<td>Used</td>
<td>Used</td>
<td>Not used</td>
</tr>
<tr>
<td>D</td>
<td>Used</td>
<td>Not used</td>
<td>Used</td>
</tr>
<tr>
<td>E</td>
<td>Used</td>
<td>Used</td>
<td>Used</td>
</tr>
</tbody>
</table>

Note:
1. The set value for PCLKB must satisfy the following:
   
   PCLKB ≥ RTC count source.
3.3.3 Oscillation Stabilization Wait Time for Each Clock
This section describes the wait control registers and oscillation stabilization wait times for the main clock, PLL and sub-clock. This section also describes values for the oscillation stabilization wait times specified in the sample code.

3.3.3.1 Main Clock Oscillation Stabilization Wait Time
Figure 3.1 shows the Main Clock Oscillation Stabilization Wait Time and Table 3.6 lists the Setting Value for the MOSCWTCR Register and Oscillation Stabilization Wait Time.

Set the main clock oscillator wait control register (MOSCWTCR) to a value greater than or equal to the main clock oscillator start-up time (tMAINOSC) recommended by the crystal/ceramic resonator manufacturer. Set the main clock oscillation stabilization wait time (tMAINOSCWT) to a value greater than ‘tMAINOSC recommended by the crystal/ceramic resonator manufacturer + (wait time set in the MOSCWTCR register + 16384 cycles)’.

$t_{MAINOSC}$ used in the sample code is 4.2 ms, thus the setting value in the MOSCWTCR register is 0Ch (approximately 5.46 ms), and the setting value for $t_{MAINOSCWT}$ is approximately 11.026 ms.

![Main Clock Oscillation Stabilization Wait Time Diagram](image)

**Figure 3.1 Main Clock Oscillation Stabilization Wait Time**

**Table 3.6 Setting Value for the MOSCWTCR Register and Oscillation Stabilization Wait Time**

<table>
<thead>
<tr>
<th>Condition of Setting Value and Calculation Method</th>
<th>Setting Value in the Sample Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOSCWTCR.MSTS[4:0] bits Value greater than or equal to $t_{MAINOSC}$ recommended by the crystal/ceramic resonator manufacturer</td>
<td>0Ch (approx. 5.46 ms)</td>
</tr>
<tr>
<td>Oscillation stabilization wait time ($t_{MAINOSCWT}$) When $n$ is the wait time selected by the MOSCWTCR.MSTS[4:0] bits: $t_{MAINOSC} + \frac{n + 16384}{f_{MAIN}}$</td>
<td>Approx. 11.026 ms</td>
</tr>
</tbody>
</table>
3.3.3.2 PLL Clock Oscillation Stabilization Wait Time
(When enabling PLL oscillation after the main clock oscillation stabilization wait time elapses)

Figure 3.2 shows the PLL Clock Oscillation Stabilization Wait Time and Table 3.7 lists the Setting Value of the PLLWTCR Register and Oscillation Stabilization Wait Time.

Set a value greater than or equal to PLL lock time (tPLL1 (max. 500 µs)) to the PLL wait control register (PLLWTCR). Set the PLL clock oscillation stabilization wait time (tPLLWT1) to a value greater than ‘tPLL1 (500 µs) + (wait time set in the PLLWTCR register + 131072 cycles)’.

tPLL1 is a maximum of 500 µs, thus the setting value in the PLLWTCR register is 0Ah (approximately 681.6 µs), and the setting value for the tPLLWT1 is approximately 1.865 ms.

Table 3.7 Setting Value of the PLLWTCR Register and Oscillation Stabilization Wait Time

<table>
<thead>
<tr>
<th>Condition of Setting Value and Calculation Method</th>
<th>Setting Value in the Sample Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLLWTCR.PSTS[4:0] bits Value greater than or equal to tPLL1 (max. 500 µs)</td>
<td>0Ah (approx. 681.6 µs)</td>
</tr>
<tr>
<td>Oscillation stabilization wait time (tPLLWT1) When n is the wait time selected by the PLLWTCR.PSTS[4:0] bits: $tPLL1 + \frac{n + 131072}{fPLL}$</td>
<td>Approx. 1.865 ms</td>
</tr>
</tbody>
</table>
3.3.3.3 Sub-Clock Oscillation Stabilization Wait Time (Except 64-Pin Packages)

Figure 3.3 shows the Sub-Clock Oscillation Stabilization Wait Time and Table 3.8 lists the Setting Value of the SOSCWTCCR Register and Oscillation Stabilization Wait Time.

Set the sub-clock oscillator wait control register (SOSCWTCCR) to a value greater than or equal to the sub-clock oscillator start-up time (tSUBOSC) recommended by the crystal/ceramic resonator manufacturer minus the minimum value of the sub-clock oscillation stabilization wait offset time (tSUBOSCWT0 (1.8 sec.)). Set the sub-clock oscillation stabilization wait time (tSUBOSCWT) to a value greater than ‘tSUBOSC recommended by the crystal/ceramic resonator manufacturer’ or ‘maximum value of tSUBOSCWT0 (2.6 sec.)’, whichever is greater plus the wait time set in the SOSCWTCCR register.

tSUBOSC used in the sample code is 1.3 seconds, thus the setting value in the SOSCWTCCR register is 00h (approximately 61 µs), and the setting value for tSUBOSCWT is approximately 2.6 seconds.

---

### Figure 3.3 Sub-Clock Oscillation Stabilization Wait Time

![Diagram showing sub-clock oscillator stability](image)

### Table 3.8 Setting Value of the SOSCWTCCR Register and Oscillation Stabilization Wait Time

<table>
<thead>
<tr>
<th>Condition of Setting Value and Calculation Method</th>
<th>Setting Value in the Sample Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOSCWTCCR.SSTS[4:0] bits</td>
<td>Value greater than or equal to tSUBOSC recommended by the crystal/ceramic resonator manufacturer minus the minimum value of tSUBOSCWT0 (1.8 sec.)</td>
</tr>
<tr>
<td>Oscillation stabilization wait time (tSUBOSCWT)</td>
<td>When n is the wait time selected by the SOSCWTCCR.SSTS[4:0] bits: Maximum value of tSUBOSCWT0 (2.6 sec.) + ( \frac{n}{f_{SUB}} )</td>
</tr>
</tbody>
</table>
### 3.3.3.4 Sub-Clock Oscillation Stabilization Wait Time (for 64-Pin Packages)

The sub-clock oscillation stabilization wait offset time (tSUBOSCWT0) is not included in 64-pin packages.

Figure 3.4 shows the Sub-Clock Oscillation Stabilization Wait Time (for 64-Pin Packages) and Table 3.9 lists the Setting Value of the SOSCWTCR Register and Oscillation Stabilization Wait Time (for 64-Pin Packages).

Set the sub-clock oscillator start-up time (tSUBOSC) recommended by the crystal/ceramic resonator manufacturer. Set the sub-clock oscillation stabilization wait time (tSUBOSCWT) to a value greater than ‘tSUBOSC recommended by the crystal/ceramic resonator manufacturer + the wait time set in the SOSCWTCR register’.

tSUBOSC used in the sample code is 1.3 seconds, thus the setting value in the SOSCWTCR register is 0Ch (approximately 2 sec.), and the setting value for tSUBOSCWT is approximately 3.3 seconds.

![Figure 3.4 Sub-Clock Oscillation Stabilization Wait Time (for 64-Pin Packages)](image)

<table>
<thead>
<tr>
<th>Condition of Setting Value and Calculation Method</th>
<th>Setting Value in the Sample Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOSCWTCR.SSTS[4:0] bits Value greater than or equal to tSUBOSC recommended by the crystal/ceramic resonator manufacturer</td>
<td>0Ch (approx. 2 sec.)</td>
</tr>
<tr>
<td>Oscillation stabilization wait time (tSUBOSCWT) When n is the wait time selected by the SOSCWTCR.SSTS[4:0] bits: $tSUBOSC + \frac{n}{fSUB}$</td>
<td>Approx. 3.3 sec.</td>
</tr>
</tbody>
</table>
3.4 File Composition

Table 3.10 lists the Files Used in the Sample Code. Files generated by the integrated development environment are not included in this table.

Table 3.10 Files Used in the Sample Code

<table>
<thead>
<tr>
<th>File Name</th>
<th>Outline</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>main.c</td>
<td>Main processing</td>
<td></td>
</tr>
<tr>
<td>r_init_stop_module.c</td>
<td>Stop processing for active peripheral functions after a reset</td>
<td></td>
</tr>
<tr>
<td>r_init_stop_module.h</td>
<td>Header file for r_init_stop_module.c</td>
<td></td>
</tr>
<tr>
<td>r_init_non_existent_port.c</td>
<td>Nonexistent port initialization</td>
<td></td>
</tr>
<tr>
<td>r_init_non_existent_port.h</td>
<td>Header file for r_init_non_existent_port.c</td>
<td></td>
</tr>
<tr>
<td>r_init_clock.c</td>
<td>Clock initialization</td>
<td></td>
</tr>
<tr>
<td>r_init_clock.h</td>
<td>Header file for r_init_clock.c</td>
<td></td>
</tr>
</tbody>
</table>

3.5 Option-Setting Memory

Table 3.11 lists the Option-Setting Memory Configured in the Sample Code. When necessary, set a value suited to the user system.

Table 3.11 Option-Setting Memory Configured in the Sample Code

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Address</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFS0</td>
<td>FFFF FF8Fh to FFFF FF8Ch</td>
<td>FFFF FFFFh</td>
<td>The IWDT is stopped after a reset. The WDT is stopped after a reset.</td>
</tr>
<tr>
<td>OFS1</td>
<td>FFFF FF8Bh to FFFF FF88h</td>
<td>FFFF FFFFh</td>
<td>The voltage monitor 0 reset is disabled after a reset. HOCO oscillation is disabled after a reset.</td>
</tr>
<tr>
<td>MDES</td>
<td>FFFF FF83h to FFFF FF80h</td>
<td>FFFF FFFFh</td>
<td>Little endian</td>
</tr>
</tbody>
</table>
3.6 Constants

Table 3.12 and Table 3.13 list the Constants Used in the Sample Code, Table 3.14 lists the Constants when a 177-Pin or 176-Pin Package is Used (PIN_SIZE=177 or PIN_SIZE=176), Table 3.15 lists the Constants when a 145-Pin or 144-Pin Package is Used (PIN_SIZE=145 or PIN_SIZE=144), Table 3.16 lists the Constants when a 100-Pin Package is Used (PIN_SIZE=100), Table 3.17 lists the Constants when a 64-Pin Package is Used (PIN_SIZE=64), and Table 3.18 lists the Constants when a 48-Pin Package is Used (PIN_SIZE=48).

Table 3.12 Constants Used in the Sample Code (1/2)

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN_CLOCK_Hz (1)</td>
<td>12,000,000 L</td>
<td>Oscillation frequency of a crystal/ceramic resonator for the main clock (Hz)</td>
</tr>
<tr>
<td>SUB_CLOCK_Hz (1)</td>
<td>32,768 L</td>
<td>Oscillation frequency of a crystal for the sub-clock (Hz)</td>
</tr>
<tr>
<td>WAIT_TIME_FOR_MAIN_OSCILLATION (1)</td>
<td>11,026,000 L</td>
<td>Main clock oscillation stabilization wait time (ns)</td>
</tr>
<tr>
<td>WAIT_TIME_FOR_SUB_OSCILLATION (1)</td>
<td>2,600,000,000 L</td>
<td>Sub-clock oscillation stabilization wait time (ns) (except 64-pin packages)</td>
</tr>
<tr>
<td>WAIT_TIME_FOR_SUB_OSCILLATION (1)</td>
<td>3,300,000,000 L</td>
<td>Sub-clock oscillation stabilization wait time (ns) (for 64-pin packages)</td>
</tr>
<tr>
<td>WAIT_TIME_FOR_PLL_OSCILLATION (1)</td>
<td>1,865,000 L</td>
<td>PLL clock oscillation stabilization wait time (ns)</td>
</tr>
<tr>
<td>WAIT_TIME_FOR_HOCO_OSCILLATION</td>
<td>2,000,000 L</td>
<td>HOCO clock oscillation stabilization wait time (ns)</td>
</tr>
<tr>
<td>PATTERN_A</td>
<td>1</td>
<td>Sub-clock setting pattern A</td>
</tr>
<tr>
<td>PATTERN_B</td>
<td>2</td>
<td>Sub-clock setting pattern B</td>
</tr>
<tr>
<td>PATTERN_C</td>
<td>3</td>
<td>Sub-clock setting pattern C</td>
</tr>
<tr>
<td>PATTERN_D</td>
<td>4</td>
<td>Sub-clock setting pattern D</td>
</tr>
<tr>
<td>PATTERN_E</td>
<td>5</td>
<td>Sub-clock setting pattern E</td>
</tr>
<tr>
<td>PATTERN_48 (PATTERN_A)</td>
<td></td>
<td>Sub-clock setting pattern for a 48-pin package</td>
</tr>
<tr>
<td>SELECT_SUB (1)</td>
<td>PATTERN_A</td>
<td>Selection of the sub-clock setting pattern</td>
</tr>
<tr>
<td>LOW_CL (1)</td>
<td>—</td>
<td>Low driving ability is used.</td>
</tr>
<tr>
<td>MSTP_STATE_DM AC DTC (2)</td>
<td>MODULE_STOP_DISABLE</td>
<td>Cancel the module-stop state for DMAC and DTC</td>
</tr>
<tr>
<td>MSTP_STATE_EXDMAC (2)</td>
<td>MODULE_STOP_DISABLE</td>
<td>Cancel the module-stop state for EXDAMC</td>
</tr>
<tr>
<td>MSTP_STATE_RAM0 (2)</td>
<td>MODULE_STOP_DISABLE</td>
<td>Cancel the module-stop state for RAM0</td>
</tr>
<tr>
<td>MSTP_STATE_RAM1 (2)</td>
<td>MODULE_STOP_DISABLE</td>
<td>Cancel the module-stop state for RAM1</td>
</tr>
<tr>
<td>PIN_SIZE (3)</td>
<td>176</td>
<td>Number of pins of the product used</td>
</tr>
<tr>
<td>MAIN_CLOCK_CYCLE</td>
<td>(1/MAIN_CLOCK_Hz)*10^9</td>
<td>Main clock cycle (ns)</td>
</tr>
<tr>
<td>SUB_CLOCK_CYCLE</td>
<td>(1/SUB_CLOCK_Hz)*10^9</td>
<td>Sub-clock cycle (ns)</td>
</tr>
</tbody>
</table>

Notes:
1. Change the setting value in r_init_clock.h according to the user system.
2. Change the setting value in r_init_stop_module.h according to the user system.
3. Change the setting value in r_init_non_existent_port.h according to the user system.
### Table 3.13 Constants Used in the Sample Code (2/2)

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOR_CMT0_TIME (when the RTC count source is the sub-clock)</td>
<td>1/LOCO*32</td>
<td>Count period (ns) of the CMT0 timer for the oscillation stabilization wait time (LOCO = 143.75 kHz (max.), PCLKB divided by 32)</td>
</tr>
<tr>
<td>FOR_CMT0_TIME (when the RTC count source is the main clock)</td>
<td>1/HOCO*32</td>
<td>Count period (ns) of the CMT0 timer for the oscillation stabilization wait time (HOCO = 55 MHz (max.), PCLKB divided by 32)</td>
</tr>
<tr>
<td>FOR_CMT0_LOCO (when the RTC count source is the main clock)</td>
<td>1/LOCO*32</td>
<td>Count period (ns) of the CMT0 timer for the main clock and PLL oscillation stabilization wait times when the RTC count source is the main clock. (LOCO = 143.75 kHz (max.), PCLKB divided by 32)</td>
</tr>
<tr>
<td>MODULE_STOP_ENABLE</td>
<td>1</td>
<td>Module stop-state</td>
</tr>
<tr>
<td>MODULE_STOP_DISABLE</td>
<td>0</td>
<td>Module stop-state is canceled</td>
</tr>
</tbody>
</table>

### Table 3.14 Constants when a 177-Pin or 176-Pin Package is Used (PIN_SIZE=177 or PIN_SIZE=176)

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEF_P0PDR</td>
<td>0x00</td>
<td>Setting value for the port P0 direction register</td>
</tr>
<tr>
<td>DEF_P1PDR</td>
<td>0x00</td>
<td>Setting value for the port P1 direction register</td>
</tr>
<tr>
<td>DEF_P2PDR</td>
<td>0x00</td>
<td>Setting value for the port P2 direction register</td>
</tr>
<tr>
<td>DEF_P3PDR</td>
<td>0x00</td>
<td>Setting value for the port P3 direction register</td>
</tr>
<tr>
<td>DEF_P4PDR</td>
<td>0x00</td>
<td>Setting value for the port P4 direction register</td>
</tr>
<tr>
<td>DEF_P5PDR</td>
<td>0x00</td>
<td>Setting value for the port P5 direction register</td>
</tr>
<tr>
<td>DEF_P6PDR</td>
<td>0x00</td>
<td>Setting value for the port P6 direction register</td>
</tr>
<tr>
<td>DEF_P7PDR</td>
<td>0x00</td>
<td>Setting value for the port P7 direction register</td>
</tr>
<tr>
<td>DEF_P8PDR</td>
<td>0x00</td>
<td>Setting value for the port P8 direction register</td>
</tr>
<tr>
<td>DEF_P9PDR</td>
<td>0x00</td>
<td>Setting value for the port P9 direction register</td>
</tr>
<tr>
<td>DEF_PAPDR</td>
<td>0x00</td>
<td>Setting value for the port PA direction register</td>
</tr>
<tr>
<td>DEF_PBPD</td>
<td>0x00</td>
<td>Setting value for the port PB direction register</td>
</tr>
<tr>
<td>DEF_PCPDR</td>
<td>0x00</td>
<td>Setting value for the port PC direction register</td>
</tr>
<tr>
<td>DEF_PDPR</td>
<td>0x00</td>
<td>Setting value for the port PD direction register</td>
</tr>
<tr>
<td>DEF_PEPDR</td>
<td>0x00</td>
<td>Setting value for the port PE direction register</td>
</tr>
<tr>
<td>DEF_PFPDR</td>
<td>0x00</td>
<td>Setting value for the port PF direction register</td>
</tr>
<tr>
<td>DEF_PGPD</td>
<td>0x00</td>
<td>Setting value for the port PG direction register</td>
</tr>
<tr>
<td>DEF_PJPDR</td>
<td>0x00</td>
<td>Setting value for the port PJ direction register</td>
</tr>
</tbody>
</table>
Table 3.15 Constants when a 145-Pin or 144-Pin Package is Used (PIN_SIZE=145 or PIN_SIZE=144)

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEF_P0PDR</td>
<td>0x00</td>
<td>Setting value for the port P0 direction register</td>
</tr>
<tr>
<td>DEF_P1PDR</td>
<td>0x03</td>
<td>Setting value for the port P1 direction register</td>
</tr>
<tr>
<td>DEF_P2PDR</td>
<td>0x00</td>
<td>Setting value for the port P2 direction register</td>
</tr>
<tr>
<td>DEF_P3PDR</td>
<td>0x00</td>
<td>Setting value for the port P3 direction register</td>
</tr>
<tr>
<td>DEF_P4PDR</td>
<td>0x00</td>
<td>Setting value for the port P4 direction register</td>
</tr>
<tr>
<td>DEF_P5PDR</td>
<td>0x80</td>
<td>Setting value for the port P5 direction register</td>
</tr>
<tr>
<td>DEF_P6PDR</td>
<td>0x00</td>
<td>Setting value for the port P6 direction register</td>
</tr>
<tr>
<td>DEF_P7PDR</td>
<td>0x00</td>
<td>Setting value for the port P7 direction register</td>
</tr>
<tr>
<td>DEF_P8PDR</td>
<td>0x30</td>
<td>Setting value for the port P8 direction register</td>
</tr>
<tr>
<td>DEF_P9PDR</td>
<td>0xF0</td>
<td>Setting value for the port P9 direction register</td>
</tr>
<tr>
<td>DEF_PA0PDR</td>
<td>0x00</td>
<td>Setting value for the port PA direction register</td>
</tr>
<tr>
<td>DEF_PB0PDR</td>
<td>0xF0</td>
<td>Setting value for the port PB direction register</td>
</tr>
<tr>
<td>DEF_PC0PDR</td>
<td>0x00</td>
<td>Setting value for the port PC direction register</td>
</tr>
<tr>
<td>DEF_PD0PDR</td>
<td>0x00</td>
<td>Setting value for the port PD direction register</td>
</tr>
<tr>
<td>DEF_PE0PDR</td>
<td>0x1F</td>
<td>Setting value for the port PE direction register</td>
</tr>
<tr>
<td>DEF_PF0PDR</td>
<td>0xFF</td>
<td>Setting value for the port PF direction register</td>
</tr>
<tr>
<td>DEF_PG0PDR</td>
<td>0xFF</td>
<td>Setting value for the port PG direction register</td>
</tr>
<tr>
<td>DEF_PJ0PDR</td>
<td>0x00</td>
<td>Setting value for the port PJ direction register</td>
</tr>
</tbody>
</table>

Table 3.16 Constants when a 100-Pin Package is Used (PIN_SIZE=100)

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEF_P0PDR</td>
<td>0x0F</td>
<td>Setting value for the port P0 direction register</td>
</tr>
<tr>
<td>DEF_P1PDR</td>
<td>0x03</td>
<td>Setting value for the port P1 direction register</td>
</tr>
<tr>
<td>DEF_P2PDR</td>
<td>0x00</td>
<td>Setting value for the port P2 direction register</td>
</tr>
<tr>
<td>DEF_P3PDR</td>
<td>0x00</td>
<td>Setting value for the port P3 direction register</td>
</tr>
<tr>
<td>DEF_P4PDR</td>
<td>0x00</td>
<td>Setting value for the port P4 direction register</td>
</tr>
<tr>
<td>DEF_P5PDR</td>
<td>0xC0</td>
<td>Setting value for the port P5 direction register</td>
</tr>
<tr>
<td>DEF_P6PDR</td>
<td>0xFF</td>
<td>Setting value for the port P6 direction register</td>
</tr>
<tr>
<td>DEF_P7PDR</td>
<td>0xFF</td>
<td>Setting value for the port P7 direction register</td>
</tr>
<tr>
<td>DEF_P8PDR</td>
<td>0xFF</td>
<td>Setting value for the port P8 direction register</td>
</tr>
<tr>
<td>DEF_P9PDR</td>
<td>0xFF</td>
<td>Setting value for the port P9 direction register</td>
</tr>
<tr>
<td>DEF_PA0PDR</td>
<td>0x00</td>
<td>Setting value for the port PA direction register</td>
</tr>
<tr>
<td>DEF_PB0PDR</td>
<td>0x00</td>
<td>Setting value for the port PB direction register</td>
</tr>
<tr>
<td>DEF_PC0PDR</td>
<td>0x00</td>
<td>Setting value for the port PC direction register</td>
</tr>
<tr>
<td>DEF_PD0PDR</td>
<td>0x00</td>
<td>Setting value for the port PD direction register</td>
</tr>
<tr>
<td>DEF_PE0PDR</td>
<td>0x00</td>
<td>Setting value for the port PE direction register</td>
</tr>
<tr>
<td>DEF_PF0PDR</td>
<td>0x3F</td>
<td>Setting value for the port PF direction register</td>
</tr>
<tr>
<td>DEF_PG0PDR</td>
<td>0xFF</td>
<td>Setting value for the port PG direction register</td>
</tr>
<tr>
<td>DEF_PJ0PDR</td>
<td>0x20</td>
<td>Setting value for the port PJ direction register</td>
</tr>
</tbody>
</table>
### Table 3.17 Constants when a 64-Pin Package is Used (PIN_SIZE=64)

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEF_P0PDR</td>
<td>0x8F</td>
<td>Setting value for the port P0 direction register</td>
</tr>
<tr>
<td>DEF_P1PDR</td>
<td>0x0F</td>
<td>Setting value for the port P1 direction register</td>
</tr>
<tr>
<td>DEF_P2PDR</td>
<td>0x3F</td>
<td>Setting value for the port P2 direction register</td>
</tr>
<tr>
<td>DEF_P3PDR</td>
<td>0x1C</td>
<td>Setting value for the port P3 direction register</td>
</tr>
<tr>
<td>DEF_P4PDR</td>
<td>0xA0</td>
<td>Setting value for the port P4 direction register</td>
</tr>
<tr>
<td>DEF_P5PDR</td>
<td>0xCF</td>
<td>Setting value for the port P5 direction register</td>
</tr>
<tr>
<td>DEF_P6PDR</td>
<td>0xFF</td>
<td>Setting value for the port P6 direction register</td>
</tr>
<tr>
<td>DEF_P7PDR</td>
<td>0xFF</td>
<td>Setting value for the port P7 direction register</td>
</tr>
<tr>
<td>DEF_P8PDR</td>
<td>0xFF</td>
<td>Setting value for the port P8 direction register</td>
</tr>
<tr>
<td>DEF_P9PDR</td>
<td>0xFF</td>
<td>Setting value for the port P9 direction register</td>
</tr>
<tr>
<td>DEF_PA PDR</td>
<td>0xA4</td>
<td>Setting value for the port PA direction register</td>
</tr>
<tr>
<td>DEF_PB PDR</td>
<td>0x14</td>
<td>Setting value for the port PB direction register</td>
</tr>
<tr>
<td>DEF_PC PDR</td>
<td>0x03</td>
<td>Setting value for the port PC direction register</td>
</tr>
<tr>
<td>DEF_PD PDR</td>
<td>0xFF</td>
<td>Setting value for the port PD direction register</td>
</tr>
<tr>
<td>DEF_PE PDR</td>
<td>0xC0</td>
<td>Setting value for the port PE direction register</td>
</tr>
<tr>
<td>DEF_PF PDR</td>
<td>0x3F</td>
<td>Setting value for the port PF direction register</td>
</tr>
<tr>
<td>DEF_PG PDR</td>
<td>0xFF</td>
<td>Setting value for the port PG direction register</td>
</tr>
<tr>
<td>DEF_PJ PDR</td>
<td>0x28</td>
<td>Setting value for the port PJ direction register</td>
</tr>
</tbody>
</table>

### Table 3.18 Constants when a 48-Pin Package is Used (PIN_SIZE=48)

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEF_P0PDR</td>
<td>0xAF</td>
<td>Setting value for the port P0 direction register</td>
</tr>
<tr>
<td>DEF_P1PDR</td>
<td>0x0F</td>
<td>Setting value for the port P1 direction register</td>
</tr>
<tr>
<td>DEF_P2PDR</td>
<td>0x3F</td>
<td>Setting value for the port P2 direction register</td>
</tr>
<tr>
<td>DEF_P3PDR</td>
<td>0x1C</td>
<td>Setting value for the port P3 direction register</td>
</tr>
<tr>
<td>DEF_P4PDR</td>
<td>0xB8</td>
<td>Setting value for the port P4 direction register</td>
</tr>
<tr>
<td>DEF_P5PDR</td>
<td>0xFF</td>
<td>Setting value for the port P5 direction register</td>
</tr>
<tr>
<td>DEF_P6PDR</td>
<td>0xFF</td>
<td>Setting value for the port P6 direction register</td>
</tr>
<tr>
<td>DEF_P7PDR</td>
<td>0xFF</td>
<td>Setting value for the port P7 direction register</td>
</tr>
<tr>
<td>DEF_P8PDR</td>
<td>0xFF</td>
<td>Setting value for the port P8 direction register</td>
</tr>
<tr>
<td>DEF_P9PDR</td>
<td>0xFF</td>
<td>Setting value for the port P9 direction register</td>
</tr>
<tr>
<td>DEF_PA PDR</td>
<td>0xA5</td>
<td>Setting value for the port PA direction register</td>
</tr>
<tr>
<td>DEF_PB PDR</td>
<td>0x4D</td>
<td>Setting value for the port PB direction register</td>
</tr>
<tr>
<td>DEF_PC PDR</td>
<td>0x0F</td>
<td>Setting value for the port PC direction register</td>
</tr>
<tr>
<td>DEF_PD PDR</td>
<td>0xFF</td>
<td>Setting value for the port PD direction register</td>
</tr>
<tr>
<td>DEF_PE PDR</td>
<td>0xE1</td>
<td>Setting value for the port PE direction register</td>
</tr>
<tr>
<td>DEF_PF PDR</td>
<td>0x3F</td>
<td>Setting value for the port PF direction register</td>
</tr>
<tr>
<td>DEF_PG PDR</td>
<td>0xFF</td>
<td>Setting value for the port PG direction register</td>
</tr>
<tr>
<td>DEF_PJ PDR</td>
<td>0x28</td>
<td>Setting value for the port PJ direction register</td>
</tr>
</tbody>
</table>
3.7 Functions
Table 3.19 lists the Functions Used in the Sample Code.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>main</td>
<td>Main processing</td>
</tr>
<tr>
<td>R_INIT_StopModule</td>
<td>Stop processing for active peripheral functions after a reset</td>
</tr>
<tr>
<td>R_INIT_NonExistentPort</td>
<td>Nonexistent port initialization</td>
</tr>
<tr>
<td>R_INIT_Clock</td>
<td>Clock initialization</td>
</tr>
<tr>
<td>CGC_oscillation_main</td>
<td>Main clock oscillation setting</td>
</tr>
<tr>
<td>CGC_oscillation_PLL</td>
<td>PLL clock oscillation setting</td>
</tr>
<tr>
<td>CGC_oscillation_HOCO</td>
<td>HOCO clock oscillation setting</td>
</tr>
<tr>
<td>CGC_no_use_subclk</td>
<td>Sub-clock setting pattern A (when the sub-clock is not used as the system clock or RTC count source, or when the package sued is a 48-pin package)</td>
</tr>
<tr>
<td>CGC_disable_subclk_RTC_use_mainclk</td>
<td>Sub-clock setting pattern B (when the sub-clock is not used and the RTC operates using the main clock)</td>
</tr>
<tr>
<td>CGC_subclk_as_sysclk</td>
<td>Sub-clock setting pattern C (when the sub-clock is used as the system clock and not used as the RTC count source)</td>
</tr>
<tr>
<td>CGC_subclk_as_RTC</td>
<td>Sub-clock setting pattern D (when the sub-clock is used as the RTC count source and not used as the system clock)</td>
</tr>
<tr>
<td>CGC_subclk_as_sysclk_RTC</td>
<td>Sub-clock setting pattern E (when the sub-clock is used as both the system clock and RTC count source)</td>
</tr>
<tr>
<td>disable_subclk</td>
<td>Disabling the sub-clock</td>
</tr>
<tr>
<td>oscillation_subclk</td>
<td>Sub-clock oscillation setting</td>
</tr>
<tr>
<td>no_use_subclk_as_sysclk</td>
<td>Processing when the sub-clock is not used as the system clock</td>
</tr>
<tr>
<td>resetting_wtcr_mainclk</td>
<td>Resetting the wait control register (RTC count source is the main clock)</td>
</tr>
<tr>
<td>resetting_wtcr_subclk</td>
<td>Resetting the wait control register (RTC count source is the sub-clock)</td>
</tr>
<tr>
<td>enable_RTC</td>
<td>Initialization when using the RTC</td>
</tr>
<tr>
<td>disable_RTC_mainclk</td>
<td>Initialization when not using the RTC (RTC count source is the main clock)</td>
</tr>
<tr>
<td>disable_RTC_subclk</td>
<td>Initialization when not using the RTC (RTC count source is the sub-clock)</td>
</tr>
<tr>
<td>cmt0_wait</td>
<td>Wait processing</td>
</tr>
</tbody>
</table>
3.8 Function Specifications
The following tables list the sample code function specifications.

| main |
| Outline | Main processing |
| Header | None |
| Declaration | void main(void) |
| Description | Call the following functions: Stop processing for active peripheral functions after a reset, nonexistent port initialization, and clock initialization. |
| Arguments | None |
| Return Value | None |

R_INIT_StopModule
| Outline | Stop processing for active peripheral functions after a reset |
| Header | r_init_stop_module.h |
| Declaration | void R_INIT_StopModule(void) |
| Description | Configure the setting to enter 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_NonExistentPort
| Outline | Nonexistent port initialization |
| Header | r_init_non_existent_port.h |
| Declaration | void R_INIT_NonExistentPort(void) |
| Description | Initialize port direction registers for ports that do not exist in products with less than 176 pins. |
| Arguments | None |
| Return Value | None |
| Remarks | The number of pins in the sample code is set for the 176-pin package (PIN_SIZE=176). After this function is called, when writing in byte units to the PDR registers or PODR registers which have nonexistent ports, set the corresponding bits for nonexistent ports as follows: set the I/O select bits in the PDR registers to 1 and set the output data store bits in the PODR registers to 0. |

R_INIT_Clock
| Outline | Clock initialization |
| Header | r_init_clock.h |
| Declaration | void R_INIT_Clock(void) |
| Description | Initialize the clock. |
| Arguments | None |
| Return Value | None |
| Remarks | The sample code selects processing which uses PLL as the system clock without using the sub-clock. |
### CGC_oscillation_main

<table>
<thead>
<tr>
<th>Outline</th>
<th>Main clock oscillation setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_init_clock.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void CGC_oscillation_main(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Set the MOSCWTCR register, and enable main clock oscillation. Then wait for the main clock oscillation stabilization wait time by software.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### CGC_oscillation_PLL

<table>
<thead>
<tr>
<th>Outline</th>
<th>PLL clock oscillation setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_init_clock.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void CGC_oscillation_PLL(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Set the PLL input frequency division ratio and frequency multiplication factor, set the PLLWTCR register, and enable PLL clock oscillation. Then wait for the PLL clock oscillation stabilization wait time by software.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
<tr>
<td>Remarks</td>
<td>This processing is not necessary if PLL is not used as the system clock.</td>
</tr>
</tbody>
</table>

### CGC_oscillation_HOCO

<table>
<thead>
<tr>
<th>Outline</th>
<th>HOCO clock oscillation setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_init_clock.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void CGC_oscillation_HOCO(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Oscillate HOCO and then wait for HOCO oscillation stabilization wait time by software.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### CGC_no_use_subclk

<table>
<thead>
<tr>
<th>Outline</th>
<th>Sub-clock setting pattern A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_init_clock.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void CGC_no_use_subclk(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Configure the setting when the sub-clock is not used as the system clock or RTC count source, or configure the setting for a 48-pin package.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### CGC_disable_subclk_RTC_use_mainclk

<table>
<thead>
<tr>
<th>Outline</th>
<th>Sub-clock setting pattern B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_init_clock.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void CGC_disable_subclk_RTC_use_mainclk(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Configure the setting when the sub-clock is not used and the RTC operates using the main clock.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>
### CGC_subclk_as_sysclk

<table>
<thead>
<tr>
<th>Outline</th>
<th>Sub-clock setting pattern C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_init_clock.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void CGC_subclk_as_sysclk(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Configure the setting when the sub-clock is used as the system clock and not used as the RTC count source.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### CGC_subclk_as_RTC

<table>
<thead>
<tr>
<th>Outline</th>
<th>Sub-clock setting pattern D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_init_clock.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void CGC_subclk_as_RTC(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Configure the setting when the sub-clock is used as the RTC count source and not used as the system clock.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### CGC_subclk_as_sysclk_RTC

<table>
<thead>
<tr>
<th>Outline</th>
<th>Sub-clock setting pattern E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_init_clock.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void CGC_subclk_as_sysclk_RTC(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Configure the setting when the sub-clock is used as both the system clock and RTC count source.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### disable_subclk

<table>
<thead>
<tr>
<th>Outline</th>
<th>Disabling the sub-clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>static void disable_subclk(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Configure the setting when the sub-clock is not used as the system clock or RTC count source.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### oscillation_subclk

<table>
<thead>
<tr>
<th>Outline</th>
<th>Sub-clock oscillation setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>static void oscillation_subclk(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Configure the sub-clock oscillation.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>
### no_use_subclk_as_sysclk

<table>
<thead>
<tr>
<th>Outline</th>
<th>Processing when the sub-clock is not used as the system clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>static void no_use_subclk_as_sysclk(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Stop the sub-clock for processing when the sub-clock is used only as the RTC count source.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### resetting_wtcr_mainclk

<table>
<thead>
<tr>
<th>Outline</th>
<th>Resetting the wait control register (RTC count source is the main clock)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>static void resetting_wtcr_mainclk(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Reset the wait control register when exiting from software standby mode. Set the minimum value to the wait control register.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### resetting_wtcr_subclk

<table>
<thead>
<tr>
<th>Outline</th>
<th>Resetting the wait control register (RTC count source is the sub-clock)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>static void resetting_wtcr_subclk(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Reset the wait control register when exiting from software standby mode. Set the minimum value to the wait control register.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### enable_RTC

<table>
<thead>
<tr>
<th>Outline</th>
<th>Initialization when using the RTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>static void enable_RTC(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Initialize the settings when using the RTC (setting for clock provision and RTC software reset).</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

### disable_RTC_mainclk

<table>
<thead>
<tr>
<th>Outline</th>
<th>Initialization when not using the RTC (RTC count source is the main clock)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>static void disable_RTC_mainclk(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Initialize the settings when not using the RTC.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>
disable_RTC_subclk

Outline: Initialization when not using the RTC (RTC count source is the sub-clock)
Header: None
Declaration: static void disable_RTC_subclk(void)
Description: Initialize the settings when not using the RTC.
Arguments: None
Return Value: None

cmt0_wait

Outline: Wait processing
Header: None
Declaration: static void cmt0_wait(uint32_t cnt)
Description: This function is used when waiting for the oscillation stabilization wait time.
Arguments:  

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| uint32_t cnt | Oscillation stabilization wait time  
|            | cnt = oscillation stabilization wait time (ns) \(^{(1)} \div \) FOR_CMT0_TIME \(^{(2)} \) |
Return Value: None
Remarks: 
1. The oscillation stabilization wait time varies depending on the crystal/ceramic resonator. Set the value referring to 3.3.3 Oscillation Stabilization Wait Time for Each Clock.
2. The value of FOR_CMT0_TIME varies depending on the RTC count source. FOR_CMT0_TIME is calculated using 55 MHz (max.) as a HOCO value when the RTC count source is the main clock, and 143.75 kHz (max.) as a LOCO value when the RTC count source is the sub-clock. The actual wait time may differ according to the HOCO or LOCO frequency.
3.9 Flowcharts

3.9.1 Main Processing

Figure 3.5 shows the Main Processing.

![Flowchart of Main Processing](image)

3.9.2 Stop Processing for Active Peripheral Functions after a Reset

Figure 3.6 shows the Stop Processing for Active Peripheral Functions after a Reset.

![Flowchart of Stop Processing](image)

Note 1

1. The module-stop state is canceled in the sample code. When entering the module-stop state for any peripheral functions, set the MSTP_STATE_{"target module"} constant to 1.
3.9.3 Nonexistent Port Initialization

Figure 3.7 shows the Nonexistent Port Initialization.

```
Figure 3.7   Nonexistent Port Initialization

Note:
1. Port initialization processing is not executed for registers which have all pins
   (the processing is skipped when compiling).

```

```
R_INIT_NonExistentPort

Set the PORT0.PDR register (1) PORT0.PDR ← PORT0.PDR | DEF_P0PDR

Set the PORT1.PDR register (1) PORT1.PDR ← PORT1.PDR | DEF_P1PDR

Set the PORT2.PDR register (1) PORT2.PDR ← PORT2.PDR | DEF_P2PDR

Set the PORT3.PDR register (1) PORT3.PDR ← PORT3.PDR | DEF_P3PDR

Set the PORT4.PDR register (1) PORT4.PDR ← PORT4.PDR | DEF_P4PDR

Set the PORT5.PDR register (1) PORT5.PDR ← PORT5.PDR | DEF_P5PDR

Set the PORT6.PDR register (1) PORT6.PDR ← PORT6.PDR | DEF_P6PDR

Set the PORT7.PDR register (1) PORT7.PDR ← PORT7.PDR | DEF_P7PDR

Set the PORT8.PDR register (1) PORT8.PDR ← PORT8.PDR | DEF_P8PDR

Set the PORT9.PDR register (1) PORT9.PDR ← PORT9.PDR | DEF_P9PDR

Set the PORTA.PDR register (1) PORTA.PDR ← PORTA.PDR | DEF_PAPDR

Set the PORTB.PDR register (1) PORTB.PDR ← PORTB.PDR | DEF_PBPDR

Set the PORTC.PDR register (1) PORTC.PDR ← PORTC.PDR | DEF_PCPDR

Set the PORTD.PDR register (1) PORTD.PDR ← PORTD.PDR | DEF_PDPDR

Set the PORTE.PDR register (1) PORTE.PDR ← PORTE.PDR | DEF_PEPDR

Set the PORTF.PDR register (1) PORTF.PDR ← PORTF.PDR | DEF_PFPDR

Set the PORTG.PDR register (1) PORTG.PDR ← PORTG.PDR | DEF_PGPDR

Set the PORTJ.PDR register (1) PORTJ.PDR ← PORTJ.PDR | DEF_PJPDR

return

```
3.9.4 Clock Initialization

Figure 3.8 shows the Clock Initialization.

```
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.

Main clock oscillation setting
CGC_oscillation_main()

Sub-clock setting (1)
CGC_no_use_subclk()
Sub-clock default setting: The sub-clock is not used as the system clock or RTC count source (pattern A).
- The sub-clock is not used and the RTC operates using the main clock (pattern B): CGC_disable_subclk_RTC_use_main
- The sub-clock is used as the system clock and not used as the RTC count source (pattern C): CGC_subclk_as_sysclk
- The sub-clock is used as the RTC count source and not used as the system clock (pattern D): CGC_subclk_as_RTC
- The sub-clock is used as both the system clock and RTC count source (pattern E): CGC_subclk_as_sysclk_RTC

PLL Clock oscillation setting (2)
CGC_oscillation_PLL()

Set the division ratio for internal clocks (3)
SCKCR register ← 21C2 1211h
PCKB[3:0] bits = 0010b: Divide-by-4 is selected for the peripheral module clock B (PCLKB).
PCKA[3:0] bits = 0001b: Divide-by-2 is selected for the peripheral module clock A (PCLKA).
BCK[3:0] bits = 0010b: Divide-by-4 is selected for the external bus clock (BCLK).
PSTOP0 bit = 1: SDCLK pin output is disabled (fixed high).
PSTOP1 bit = 1: BCLK pin output is disabled (fixed high).
ICK[3:0] bits = 0001b: Divide-by-2 is selected for the system clock (ICLK).
FCK[3:0] bits = 0010b: Divide-by-4 is selected for the FlashIF clock (FCLK).
SCKCR2 register ← 0012h
IEBCK[3:0] bits = 0010b: Divide-by-4 is selected for the IEBUS clock (IECLK).
UCK[3:0] bits = 0001b: USB is not in use.

Set the BCLK pin output (3)
BCKCR register ← 00h
BCLKDIV bit = 0: BCLK (no division)

Switch the system clock (3)
SCKCR3 register ← 0400h
CKSEL[2:0] bits = 100b: PLL circuit

Turn off the HO CO power supply (3)
HOCCOR register ← 01h
HCSTP bit = 1: HO CO is stopped.
HOCCOCR register ← 01h
HOCCOPCNT bit = 1: Turn the power supply of the HO CO off.

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

return
```

Notes:
1. Select the sub-clock pattern according to the user system.
2. The PLL oscillation setting is not necessary when PLL is not used.
3. Set an appropriate value according to the user system.

Figure 3.8 Clock Initialization
### 3.9.5 Main Clock Oscillation Setting

Figure 3.9 shows the Main Clock Oscillation Setting.

```
CGC_oscillation_main

Set the wait control register for the main clock oscillator
MOSCWTCR register ← 0Ch
MSTS[4:0] bits = 01100b: Waiting time = 65536 cycles (approx. 5.46 ms)

Operate the main clock oscillator
MOSCCR register ← 00h
MOSTP bit = 0: Main clock oscillator is operating.

Wait for the oscillation stabilization wait time using CMT0
wait()
Wait for the main clock oscillation stabilization wait time (approx. 11.026 ms).
```

Figure 3.9 Main Clock Oscillation Setting

### 3.9.6 PLL Clock Oscillation Setting

Figure 3.10 shows the PLL Clock Oscillation Setting.

```
CGC_oscillation_PLL

Set the PLL input division ratio and frequency multiplication factor
PLLWTCR register ← 0Ah
PSTS[4:0] bits = 01010b: Waiting time = 131072 cycles (approx. 681.6 µs)

Set the wait control register for the PLL clock oscillator
PLLCR register ← 0F00h
PLIDIV[1:0] bits = 00b: Divided by 1
STC[5:3] bits = 001111b: Multiplied by 16

Operate PLL
PLLCR2 register ← 00h
PLLEN bit = 0: PLL is operating.

Wait for the oscillation stabilization wait time using CMT0
wait()
Wait for the PLL clock oscillation stabilization wait time (approx. 1.865 ms).
```

Figure 3.10 PLL Clock Oscillation Setting
3.9.7 HOCO Clock Oscillation Setting

Figure 3.11 shows the HOCO Clock Oscillation Setting.

![Flowchart of HOCO Clock Oscillation Setting]

- CGC_oscillation_HOCO
- Operate HOCO
- Wait for the oscillation stabilization wait time using CMT0 cmt0_wait()
- return

HOCOCR register ← 00h
HCSTP bit = 0: HOCO is operating.

Wait for the HOCO clock oscillation stabilization wait time (approx. 2.0 ms).

Figure 3.11 HOCO Clock Oscillation Setting
3.9.8 **Sub-Clock Setting**

Figure 3.12 to Figure 3.16 show the sub-clock setting patterns A to E.

Figure 3.12 shows Sub-Clock Setting Pattern A (when the Sub-Clock is not Used as the System Clock or RTC Count Source).

- **CGC_no_use_subclk**
  - CGC_oscillation_HOCO()
  - Switch the system clock
  - Set the RTC count source
  - Wait processing cmt0_wait()

- **Setting when not using the sub-clock**
  Function: disable_subclk

- **Initialization when not using the RTC**
  (RTC count source is the main clock)
  Function: disable_RTC_mainclk

- **Stop the sub-clock oscillator**
  - SOCCR register ← 0100h
  - CKSEL[2:0] bits = 001b: HOCO is selected.

- **Stop the prescaler and counters**
  - RCR4 register
  - RCKSEL bit ← 1: The main clock is selected as the count source.

- **Wait processing**
  cmt0_wait()

- **RTC software reset**
  - RCR2 register ← RCR2 register & 7Eh
  - START bit = 0: Prescaler and counters are stopped.
  - RCR2 register
  - RESET bit ← 1: The prescaler and target registers are reset by RTC software reset.

- **Has the RTC software reset been completed?**
  - Yes
    - Disable interrupt requests
      - RCR1 register ← 00h
      - AIE bit = 0: An alarm interrupt request is disabled.
      - CIE bit = 0: A carry interrupt request is disabled.
      - PIE bit = 0: A periodic interrupt request is disabled.
      - IR0/62 register
        - IR flag ← 0: No carry interrupt request is generated.
      - IR0/92 register
        - IR flag ← 0: No alarm interrupt request is generated.
      - IR0/93 register
        - IR flag ← 0: No periodic interrupt request is generated.

  - No
    - Stop the sub-clock oscillator
      - SOSCCR register ← 01h
      - SOSTP bit = 1: Sub-clock oscillator is stopped.
      - RCR3 register
        - RTCEN bit ← 0: Sub-clock oscillator is stopped.
      - Supply 6 clocks of the main clock using CMT0.

- **Switch the system clock**
  - SCKCR3 register ← 0100h
  - CKSEL[2:0] bits = 000b: Main clock is selected.

- **Set the RTC count source**
  - RCR4 register
    - RCKSEL bit ← 1: The main clock is selected as the count source.

- **Wait processing**
  cmt0_wait()

*Figure 3.12 Sub-Clock Setting Pattern A (when the Sub-Clock is not Used as the System Clock or RTC Count Source)*
Figure 3.13 shows the Sub-Clock Setting Pattern B (when the Sub-Clock is Stopped and the RTC Operates Using the Main Clock).

Figure 3.13 Sub-Clock Setting Pattern B (when the Sub-Clock is Stopped and the RTC Operates Using the Main Clock)
Figure 3.14 shows the Sub-Clock Setting Pattern C (when the Sub-Clock is Used as the System Clock and not Used as the RTC Count Source).

**Sub-clock oscillation setting**

Function: `oscillation_subclk`

- **CRCG_subclk_as_sysclk**
  - Set the RTC count source
  - RCR4 register
  - RTCSEL bit = 0: The sub-clock is selected as the count source.

- **Stop the sub-clock oscillator**
  - SOSCCCR register ← 01h
  - SOSTP bit = 1: Sub-clock oscillator is stopped.
  - RCR3 register
  - RTCEN bit = 0: Sub-clock oscillator is stopped.

- **Wait for 6 cycles of the sub-clock using CMT0**
  - Wait for 5 cycles of the sub-clock (approx. 152 μs).

- **Set the sub-clock driving ability**
  - RCR3 register ← 0Ch
  - RTCDV[2:0] bits = 110b: Standard driving ability
  - RTCDV[2:0] bits = 001b: Low driving ability

- **Set the sub-clock oscillator wait control register**
  - SOSCWTCCR register ← 00h
  - SSTS[4:0] bits = 00000b: Wait for 2 cycles (approx. 61 μs)

- **Operate the sub-clock oscillator**
  - SOSCCCR register ← 00h
  - SOSTP bit = 0: Sub-clock oscillator is operating.

- **Wait for the sub-clock oscillation stabilization wait time using CMT0**
  - Wait for the sub-clock oscillation stabilization wait time (2.6 sec.) (except 64-pin packages)
  - Wait for the sub-clock oscillation stabilization wait time (3.3 sec.) (for 64-pin packages)

**Initialization when not using the RTC (RTC count source is the sub-clock)**

Function: `disable_RTC_subclk`

- **RCR2 register**
  - RCR2 register & 7Eh
  - START bit = 0: Prescaler and counters are stopped.

- **RCR1 register**
  - ← 00h
  - AIE bit = 0: An alarm interrupt request is disabled.
  - CIE bit = 0: A carry interrupt request is disabled.
  - PIE bit = 0: A periodic interrupt request is disabled.

- **RCR3 register**
  - RTCEN bit = 1: Sub-clock oscillator is running.

- **Stop the prescaler and counters**
  - RCR2 register ← RCR2 register & 7Eh
  - START bit = 0: Prescaler and counters are stopped.

- **RTC software reset**
  - RCR2 register
  - RESET bit = 1: The prescaler and target registers are reset by RTC software reset.

- **Has the RTC software reset been completed?**
  - Yes
    - **Disable interrupt requests**
      - RCR1 register ← 00h
      - ASE bit = 0: An alarm interrupt request is disabled.
      - CIE bit = 0: A carry interrupt request is disabled.
      - PIE bit = 0: A periodic interrupt request is disabled.
    - **Clear IR flags**
      - IR092 register
      - IR flag ← 0: No alarm interrupt request is generated.
      - IR093 register
      - IR flag ← 0: No periodic interrupt request is generated.

- **Reset the RCR3 register**
  - RCR3 register ← 0Ch
  - RTCEN bit = 0: Sub-clock oscillator is stopped.

- **return**

**Notes:**

1. Set the sub-clock oscillator wait control register to a value greater than or equal to the sub-clock oscillator start-up time recommended by the crystal/ceramic resonator manufacturer minus the minimum value of the sub-clock oscillation stabilization wait offset time (1.8 sec.).
2. The time includes six clocks of the RTC count source.
3. The driving ability is set to drive a standard clock load in the sample code. When using the low driving ability, comment out the definition of LOW_CL in r_init_clock.h.
Figure 3.15 shows the Sub-Clock Setting Pattern D (when the Sub-Clock is Used as the RTC Count Source and not Used as the System Clock).

**Sub-clock oscillation setting**

Function: oscillation_subclk

- **RCR4 register**
  - RCKSEL bit ← 0: The sub-clock is selected as the count source.

- **SOSCCR register**
  - <01h>
  - SOSTP bit = 1: Sub-clock oscillator is stopped.
  - RTCEN bit = 0: Sub-clock oscillator is stopped.

- **RCR2 register**
  - &<7Eh>
  - START bit = 0: Prescaler and counters are stopped.

- **RTC software reset**
  - Reset the RCR3 register

- **RCR3 register**
  - UART bit ← 0: Sub-clock oscillator is running.

- **SOSCWTCR register**
  - <00h>
  - SST[4:0] bits = <00000b>: Wait for 2 cycles (approx. 61 µs).

- **RCR2 register**
  - RCR2 register & <7Eh>
  - RESET bit ← 1: The prescaler and target registers are reset by RTC software reset.

- **RCR3 register**
  - &<0D>
  - RTCEN bit = 1: Sub-clock oscillator is running.

- **SOSCCR register**
  - <01h>
  - SOSTP bit = 1: Sub-clock oscillator is stopped.

- **Wait for 5 cycles of the sub-clock using CMT0 wait()**

- **RCKSEL bit**
  - <001b>
  - RTCDV[2:0] bits = 000b: Standard driving ability
  - RTCDV[2:0] bits = 001b: Low driving ability

- **Wait for the sub-clock oscillation stabilization wait time (2.6 sec.) (except 64-pin packages)**

- **SOSCWTCR register**
  - <00h>
  - SST[4:0] bits = <00000b>: Wait for 2 cycles (approx. 61 µs).

**Notes:**

1. Set the sub-clock oscillator wait control register to a value greater than or equal to the sub-clock oscillator start-up time recommended by the crystal/ceramic resonator manufacturer minus the minimum value of the sub-clock oscillation stabilization wait offset time (1.8 sec.).
2. The time includes six clocks of the RTC count source.
3. The driving ability is set to drive a standard clock load in the sample code. When using the low driving ability, comment out the definition of LOW_CL in r_init_clock.h.

**Figure 3.15 Sub-Clock Setting Pattern D (when the Sub-Clock is Used as the RTC Count Source and not Used as the System Clock)**
Figure 3.16 shows the Sub-Clock Setting Pattern E (when the Sub-Clock is Used as Both the System Clock and RTC count source).

<table>
<thead>
<tr>
<th>CGIC_subclk_as_sysclk_RTC</th>
<th>RCR4 register RCKSEL bit ← 0: The sub-clock is selected as the count source.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stop the sub-clock oscillator</td>
</tr>
<tr>
<td></td>
<td>Wait for 5 cycles of the sub-clock using CMT0 cmt0_wait()</td>
</tr>
<tr>
<td></td>
<td>Set the sub-clock driving ability</td>
</tr>
<tr>
<td></td>
<td>Set the sub-clock oscillator wait control register (1)</td>
</tr>
<tr>
<td></td>
<td>Operate the sub-clock oscillator</td>
</tr>
<tr>
<td></td>
<td>Wait for the sub-clock oscillation stabilization wait time using CMT0 (2)</td>
</tr>
<tr>
<td></td>
<td>Set the sub-clock oscillator wait control register (3)</td>
</tr>
<tr>
<td></td>
<td>Operate the sub-clock oscillator</td>
</tr>
<tr>
<td></td>
<td>Wait for 2 cycles of the sub-clock using CMT0 cmt0_wait()</td>
</tr>
<tr>
<td></td>
<td>return</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RCR3 register RTGEN bit ← 1: Sub-clock oscillator is running.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCR2 register ← RCR2 register &amp; 7Eh START bit ← 0: Prescaler and counters are stopped.</td>
</tr>
<tr>
<td>RCR2 register ← 1: The prescaler and target registers are reset by RTC software reset.</td>
</tr>
<tr>
<td>RCR3 register ← 00D RTCEN bit ← 1: Sub-clock oscillator is running.</td>
</tr>
<tr>
<td>RCR3 register ← 00h RCKSEL bit ← 0: The sub-clock is selected as the count source.</td>
</tr>
<tr>
<td>RCR3 register ← 00h RCKSEL bit ← 0: Sub-clock oscillator is operating.</td>
</tr>
<tr>
<td>RCR3 register ← 00h SOSTP bit = 1: Sub-clock oscillator is stopped.</td>
</tr>
<tr>
<td>RCR3 register ← 00h SOSTP bit = 0: Sub-clock oscillator is operating.</td>
</tr>
<tr>
<td>RCR3 register ← 00h SOSTP bit = 1: Sub-clock oscillator is running.</td>
</tr>
</tbody>
</table>

**Notes:**
1. Set the sub-clock oscillator wait control register to a value greater than or equal to the sub-clock oscillator start-up time recommended by the crystal/ceramic resonator manufacturer minus the minimum value of the sub-clock oscillation stabilization wait offset time (1.8 sec.).
2. The time includes six clocks of the RTC count source.
3. When resetting the wait control register, set the minimum value (00000b) to the register.
4. The driving ability is set to drive a standard clock load in the sample code. When using the low driving ability, comment out the definition of LOW_CL in r_init_clock.h.

Figure 3.16 Sub-Clock Setting Pattern E (when the Sub-Clock is Used as Both the System Clock and RTC count source)
3.9.9 Wait Processing

Figure 3.17 shows the Wait Processing.

```
Figure 3.17 Wait Processing
```

Note:
1. When the counter of the watchdog timer (WDT) or independent watchdog timer (IWDT) is operating, refresh the counter in this loop as required.

```
RX63N Group, RX631 Group
Initial Setting

3.9.9 Wait Processing

Figure 3.17 shows the Wait Processing.

```
```
4. Appendices

4.1 Clock Oscillation Stabilization Wait Time

4.1.1 When Operating PLL before Main Clock Oscillation Stabilizes

When oscillating the main clock and PLL clock, their oscillation stabilization wait times can be combined into a single wait time.

Figure 4.1 shows the PLL Oscillation Stabilization Wait Time (when Operating PLL before the Main Clock Stabilizes) and Table 4.1 lists the Setting Value for the PLL Clock Wait Control Register and Oscillation Stabilization Wait Time (when Operating PLL before Main Clock Oscillation Stabilizes).

![Diagram of PLL Oscillation Stabilization Wait Time](image)

### Table 4.1 Setting Value for the PLL Clock Wait Control Register and Oscillation Stabilization Wait Time (when Operating PLL before Main Clock Oscillation Stabilizes)

<table>
<thead>
<tr>
<th>Condition of Setting Value and Calculation Method</th>
<th>Value greater than or equal to the main clock oscillator start-up time recommended by the crystal/ceramic resonator manufacturer plus tPLL1 (max. 500 µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLL wait control register (PLLWTCR.PSTS[4:0] bits)</td>
<td>When n is the wait time selected by the PLLWTCR.PSTS[4:0] bits: ( t_{MAINOSC} + t_{PLL1} + \frac{n + 131072}{f_{PLL}} )</td>
</tr>
</tbody>
</table>
### 4.1.2 When the Sub-Clock Oscillator Start-Up Time is Greater than the Sub-Clock Oscillation Stabilization Wait Offset Time

The sample code assumes that the sub-clock oscillator start-up time (tSUBOSC) is less than the sub-clock oscillation stabilization wait offset time (tSUBOSCWT0).

Figure 4.2 shows the Sub-Clock Oscillation Stabilization Wait Time (when tSUBOSC is Greater than tSUBOSCWT0) and Table 4.2 lists the Setting Value for the Sub-Clock Oscillator Wait Control Register and Oscillation Stabilization Wait Time (when tSUBOSC is Greater than tSUBOSCWT0).

<table>
<thead>
<tr>
<th>Condition of Setting Value and Calculation Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-clock oscillator wait control register (SOSCWTCR.SSTS[4:0] bits)</td>
</tr>
<tr>
<td>Oscillation stabilization wait time (tSUBOSCWT)</td>
</tr>
</tbody>
</table>

**Notes:**
1. Contact the crystal/ceramic resonator manufacturer to determine the start-up time of a crystal/ceramic resonator for the user system. The start-up time is not a condition for MCU operation but for a crystal/ceramic resonator to start oscillation.
5. Sample Code
Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents
User’s Manual: Hardware
RX63N Group, RX631 Group User’s Manual: Hardware Rev.1.70 (R01UH0041EJ)
The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

User’s Manual: Development Tools
RX Family C/C++ Compiler Package V.1.01 User’s Manual Rev.1.00 (R20UT0570EJ)
The latest version can be downloaded from the Renesas Electronics website.

Website and Support
Renesas Electronics website
http://www.renesas.com

Inquiries
http://www.renesas.com/contact/
## REVISION HISTORY

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Mar. 1, 2013</td>
<td>First edition issued</td>
</tr>
<tr>
<td>1.10</td>
<td>Jan. 6, 2014</td>
<td>- Covered 64-pin and 48-pin packages. - Modified the wait processing with CMT0 to support 32-bit argument.</td>
</tr>
</tbody>
</table>

Table 2.1 Operation Confirmation Conditions:
- Operating frequencies: Added the description regarding 48-pin version.
- iodefine.h version: Updated the version to 1.6A

Table 3.3 Nonexistent Ports (2/2):
Added the nonexistent port list for 48-pin and 64-pin packages.

Table 3.4 Clock Setting Procedure:
Added the description of a 48-pin package in step 2.

3.3.3.3 Sub-Clock Oscillation Stabilization Wait Time (Except 64-Pin Packages): Changed the specifications to support packages except 64-pin packages.

3.3.3.4 Sub-Clock Oscillation Stabilization Wait Time (for 64-Pin Packages): Added.

Table 3.12 Constants Used in the Sample Code:
- Added the WAIT_TIME_FOR_SUB_OSCILLATION constant for 64-pin package.
- Added the PATTERN_48 constant.
- Added the LOW_CL constant.

Table 3.17 Constants when a 60-Pin Package is Used (PIN_SIZE=60): Added.

Table 3.18 Constants when a 48-Pin Package is Used (PIN_SIZE=48): Added.

3.7 Functions: Added the description of a 48-pin package to the CGC_no_use_subclk function.

33 to 35 Figure 3.14 Sub-Clock Setting Pattern C, Figure 3.15 Sub-Clock Setting Pattern D, and Figure 3.16 Sub-Clock Setting Pattern E: Added descriptions regarding the low driving ability and 64-pin packages.

36 3.9.9 Wait Processing: Revised.

---

All trademarks and registered trademarks are the property of their respective owners.
### General Precautions in the Handling of MPU/MCU Products

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

<table>
<thead>
<tr>
<th>1. Handling of Unused Pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.</td>
</tr>
<tr>
<td>- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. Processing at Power-on</th>
</tr>
</thead>
<tbody>
<tr>
<td>The state of the product is undefined at the moment when power is supplied.</td>
</tr>
<tr>
<td>- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.</td>
</tr>
<tr>
<td>In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.</td>
</tr>
<tr>
<td>In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. Prohibition of Access to Reserved Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access to reserved addresses is prohibited.</td>
</tr>
<tr>
<td>- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. Clock Signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.</td>
</tr>
<tr>
<td>- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5. Differences between Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.</td>
</tr>
<tr>
<td>- The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the 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.</td>
</tr>
</tbody>
</table>
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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.

5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the products 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, and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.

7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, the control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcontroller software alone is very difficult, please evaluate 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose related to military use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.

11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Notice 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

Reneses Electronics Corporation
http://www.renesas.com

SALES OFFICES

Renesas Electronics America Inc.,
2880 Scott Boulevard Santa Clara, CA 95054, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited,
1101 Nicholson Road, Nepean, Ontario L3Y 9C3, Canada
Tel: +1-613-898-5441, Fax: +1-613-898-3220

Reneses Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-304

Reneses Electronics Europe GmbH
Arcadaplatz 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1227

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

Reneses Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Taiwei, 555 LinShao Rd., Pudong District, Shanghai, China
Tel: +86-21-2226-0568, Fax: +86-21-2226-0569

Reneses Electronics Hong Kong Limited
Unit 203-206, Tower 3, Grand Century Place, 193 Prince Edward Road West, Mongklok, Kowloon, Hong Kong
Tel: +852-2866-9318, Fax: +852-2866-9044

Reneses Electronics Taiwan Co., Ltd.
12F, No. 66, Section 1, Shong North Road, Taipei, Taiwan
Tel: +886-2-8175-9000, Fax: +886-2-8175-9070

Reneses Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit 805-02 HuXin Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0200

Reneses Electronics Malaysia Sdn.Bhd.
Unit 905-10, Menara Ambank, 6th Floor, 2, Jalan Perdepatan, Jelutong Trade Centre, No. 18, Jln Persiaran Barat, 40900 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +603-7955-9300, Fax: +603-7955-9510

Reneses Electronics Korea Co., Ltd.
12F, 234 Tahan-ro, Gangnam-gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-3141

© 2014 Renesas Electronics Corporation. All rights reserved.

Colophon 3.0