RX21A Group
Transition to Low Power Consumption Modes

Abstract
This document describes transition to low power consumption modes using the low power consumption function in the RX21A Group.

Products
- RX21A Group 100-pin package with a ROM size between 256 KB and 512 KB
- RX21A Group 80-pin package with a ROM size between 256 KB and 512 KB
- RX21A Group 64-pin package 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

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

3. Reference Application Note ................................................................................................................ 5

4. Hardware ............................................................................................................................................ 6
   4.1 Pins Used ..................................................................................................................................... 6

5. Software ............................................................................................................................................. 7
   5.1 Operation Overview ..................................................................................................................... 8
      5.1.1 Sleep Mode ........................................................................................................................... 8
      5.1.2 Software Standby Mode ....................................................................................................... 8
      5.1.3 Deep Software Standby Mode .............................................................................................. 9
      5.1.4 Entering and Exiting a Low Power Consumption Mode Using the IRQ ............................. 10
      5.1.5 Entering and Exiting a Low Power Consumption Mode Using the RTC ............................ 10
      5.1.6 Entering and Exiting a Low Power Consumption Mode Using the LVD ............................. 11
      5.2 File Composition ........................................................................................................................ 11
      5.3 Option-Setting Memory .............................................................................................................. 12
      5.4 Constants ................................................................................................................................... 12
      5.5 Variables .................................................................................................................................... 12
      5.6 Functions .................................................................................................................................... 13
      5.7 Function Specifications .............................................................................................................. 13
      5.8 Flowcharts .................................................................................................................................. 17
         5.8.1 Main Processing ................................................................................................................. 17
         5.8.2 Port Initialization ................................................................................................................. 18
         5.8.3 Peripheral Function Initialization ......................................................................................... 18
         5.8.4 Transition to Sleep Mode .................................................................................................... 19
         5.8.5 Transition to Software Standby Mode ................................................................................ 20
         5.8.6 Transition to Deep Software Standby Mode ....................................................................... 21
         5.8.7 IRQ Initialization .................................................................................................................. 24
         5.8.8 LVD Initialization ................................................................................................................. 25
         5.8.9 RTC Initialization ................................................................................................................. 27
         5.8.10 IRQ1 Interrupt Handling ..................................................................................................... 28
         5.8.11 LVD1 Interrupt Handling .................................................................................................. 28
         5.8.12 LVD2 Interrupt Handling .................................................................................................. 29
         5.8.13 RTC.PRD Interrupt Handling ............................................................................................. 29

6. Sample Code .................................................................................................................................... 30

7. Reference Documents ...................................................................................................................... 30
1. Specifications

The sample code performs processing to enter and exit a low power consumption mode by specifying a source for transition and exit. The MCU enters or exits a low power consumption mode when the specified source occurs.

- Low power consumption modes:
  Selectable from sleep mode, software standby mode, and deep software standby mode
- Source to enter or exit low power consumption mode: Selectable from IRQ1, LVD, and RTC
- Clock source: Selectable
- Operating power control mode: Selectable
- Sleep mode return clock source switching function: Not used

Note:
1. The clock source and operating power control mode are selected in r_init_clock.h. This application note uses the main clock as the clock source and middle-speed operating mode 1A for the operating power control mode. Refer to the RX21A Group Initial Setting Rev. 1.00 application note for details.

Table 1.1 lists the Peripheral Functions and Their Applications and Figure 1.1 shows the Block Diagram.

<table>
<thead>
<tr>
<th>Peripheral Function</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low power consumption function</td>
<td>Reduces power consumption.</td>
</tr>
<tr>
<td>External pin interrupt (IRQ)</td>
<td>Enters low power consumption mode.</td>
</tr>
<tr>
<td></td>
<td>Exits low power consumption mode.</td>
</tr>
<tr>
<td>Voltage detection circuit (LVD)</td>
<td>Enters low power consumption mode.</td>
</tr>
<tr>
<td></td>
<td>Exits low power consumption mode.</td>
</tr>
<tr>
<td>Realtime clock (RTC)</td>
<td>Enters low power consumption mode.</td>
</tr>
<tr>
<td></td>
<td>Exits low power consumption mode.</td>
</tr>
</tbody>
</table>
Figure 1.1 Block Diagram
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>R5F521A8BDFP (RX21A Group)</td>
</tr>
<tr>
<td>Operating frequencies</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Main clock: 20 MHz</td>
</tr>
<tr>
<td></td>
<td>- Sub-clock: 32.768 kHz</td>
</tr>
<tr>
<td></td>
<td>- System clock (ICLK): 20 MHz (main clock divided by 1)</td>
</tr>
<tr>
<td></td>
<td>- Peripheral module clock B (PCLKB): 20 MHz (main clock divided by 1)</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>1.8 to 3.3 V</td>
</tr>
<tr>
<td>Integrated development</td>
<td>Renesas Electronics Corporation</td>
</tr>
<tr>
<td>environment</td>
<td>High-performance Embedded Workshop Version 4.09.01</td>
</tr>
<tr>
<td>C compiler</td>
<td>Renesas Electronics Corporation</td>
</tr>
<tr>
<td></td>
<td>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=rx200 -output=obj=&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 1.0</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 1.00</td>
</tr>
<tr>
<td>Board used</td>
<td>Hokuto Electronic Co., Ltd.</td>
</tr>
<tr>
<td></td>
<td>HSB Series MCU board (product part no.: HSBRX21AP-B)</td>
</tr>
</tbody>
</table>

3. Reference Application Note

For additional information associated with this document, refer to the following application note.

- RX21A Group Initial Setting Rev. 1.00 (R01AN1486EJ0100_RX21A)

The sample code in this application note uses the initial setting functions in the reference application note with the settings changed as follows:
- The main clock is set as the clock source.
- The sub-clock is set to oscillate (RTC used).
- The operation power control mode is set to middle-speed operating mode 1A.

The revision number of the reference application note is the one when this application note is created. However the latest version is always recommended. Visit the Renesas Electronics Corporation website to check and download the latest version.
4. Hardware

4.1 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

The pins described here are for 100-pin products. When the product with less than 100-pin is used, select pins appropriate to the product used.

Table 4.1 Pins Used and Their Functions

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P31/IRQ1</td>
<td>Input</td>
<td>SW2 input (for entering or exiting low power consumption mode)</td>
</tr>
<tr>
<td>PH0</td>
<td>Output</td>
<td>LED1 output (turned on after the initial setting has been performed)</td>
</tr>
<tr>
<td>PH1</td>
<td>Output</td>
<td>LED2 output (turned on before entering low power consumption mode)</td>
</tr>
<tr>
<td>PH2</td>
<td>Output</td>
<td>LED3 output (turned on after exiting low power consumption mode)</td>
</tr>
</tbody>
</table>
5. Software

When the source for entering a low power consumption mode occurs, the MCU enters a low power consumption mode. When in a low power consumption mode, if the source for exit occurs, the MCU exits the mode it is in.

Sleep mode, software standby mode, or deep software standby mode can be selected as the low power consumption mode. The IRQ, LVD, or RTC can be used for the source to enter or exit a low power consumption mode.

Settings for the peripheral functions are as follows:

**IRQ**
- Detection method: Falling edge on the IRQ1 pin
- Digital filter: Disabled
- Interrupt priority level: Level 15

**LVD**
- Condition for LVD1 detection: VCC passed upward through Vdet1 (2.95 V)
- Condition for LVD2 detection: VCC passed downward through Vdet2 (2.80 V)
- Processing when LVD1 is detected: Voltage monitoring 1 interrupt (maskable)
- Processing when LVD2 is detected: Voltage monitoring 2 interrupt (maskable)
- Digital filter: Disabled
- Interrupt priority level: Level 15

**RTC**
- Initial date and time setting: 2013-01-01 (Tue.) 00:00:00
- Time mode: 24-hour mode
- Interrupt: Periodic interrupt (PRD) is used and generated every 2 seconds.
- Interrupt priority level: Level 15
5.1 Operation Overview

5.1.1 Sleep Mode
After the initial setting has been performed, turns on LED1 and waits until the source for transition to sleep mode occurs. When the source occurs, turns off LED1, turns on LED2, and enters sleep mode. When the source for exit occurs during sleep mode, exits sleep mode, turns off LED2, and turns on LED3.

Figure 5.1 shows the Operation Overview when Entering and Exiting Sleep Mode.

5.1.2 Software Standby Mode
After the initial setting has been performed, turns on LED1 and waits until the source for transition to software standby mode occurs. When the source occurs, turns off LED1, turns on LED2, and enters software standby mode. When the source for exit occurs during software standby mode, exits software standby mode, turns off LED2, and turns on LED3.

Figure 5.2 shows the Operation Overview when Entering and Exiting Software Standby Mode.
5.1.3 Deep Software Standby Mode

After the initial setting has been performed, turns on LED1 and waits until the source for transition to deep software standby mode occurs. When the source occurs, turns off LED1, turns on LED2, and enters deep software standby mode. When the source for exit occurs during deep software standby mode, exits deep software standby mode, and performs a reset. After the reset, performs the initial setting and turns on LED3.

Figure 5.3 shows the Operation Overview when Entering and Exiting Deep Software Standby Mode.

![Figure 5.3 Operation Overview when Entering and Exiting Deep Software Standby Mode](image-url)
5.1.4 Entering and Exiting a Low Power Consumption Mode Using the IRQ

When using the IRQ for the source to enter and exit a low power consumption mode, the MCU enters or exits a low power consumption mode by the IRQ interrupt request generation.

The IRQ interrupt request generated during a wait period for transition to a low power consumption mode becomes the source for transition, and the IRQ interrupt request generated during the low power consumption mode becomes the source for exit.

Figure 5.4 shows the Timing for Entering and Exiting a Low Power Consumption Mode Using the IRQ.

![Figure 5.4 Timing for Entering and Exiting a Low Power Consumption Mode Using the IRQ](image)

5.1.5 Entering and Exiting a Low Power Consumption Mode Using the RTC

When using the RTC for the source to enter and exit a low power consumption mode, the MCU enters or exits a low power consumption mode by the RTC.PRD interrupt request generated every 2 seconds.

The RTC.PRD interrupt request generated during a wait period for transition to a low power consumption mode becomes the source for transition, and the RTC.PRD interrupt request generated during the low power consumption mode becomes the source for exit.

Figure 5.5 shows the Timing of Entering and Exiting a Low Power Consumption Mode Using the RTC.

![Figure 5.5 Timing of Entering and Exiting a Low Power Consumption Mode Using the RTC](image)
5.1.6 Entering and Exiting a Low Power Consumption Mode Using the LVD

When using the LVD for the source to enter and exit a low power consumption mode, the MCU enters a low power consumption mode by the LVD2 interrupt request generation and exits the low power consumption mode by the LVD1 interrupt request generation. The LVD2 interrupt request is generated when ‘VCC < Vdet2’ is detected. The LVD1 interrupt request is generated when ‘VCC ≥ Vdet1’ is detected.

Figure 5.6 shows the Timing of Entering and Exiting a Low Power Consumption Mode Using the LVD.

![Figure 5.6 Timing of Entering and Exiting a Low Power Consumption Mode Using the LVD](image)

5.2 File Composition

Table 5.1 lists the Files Used in the Sample Code. Files generated by the integrated development environment are not included in this table.

Table 5.1 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>
5.3 Option-Setting Memory

Table 5.2 lists the Option-Setting Memory Configured in the Sample Code. When necessary, set a value suited to the user system.

Table 5.2 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 FF88h to FFFF FF88h</td>
<td>FFFF FFFFh</td>
<td>The voltage monitor 0 reset is disabled after a reset.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>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>

5.4 Constants

Table 5.3 lists the Constants Used in the Sample Code.

Table 5.3 Constants Used in the Sample Code

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_IRQ</td>
<td>0</td>
<td>Source to enter and exit a low power consumption mode: IRQ</td>
</tr>
<tr>
<td>L_LVD</td>
<td>1</td>
<td>Source to enter and exit a low power consumption mode: LVD</td>
</tr>
<tr>
<td>L_RTC</td>
<td>2</td>
<td>Source to enter and exit a low power consumption mode: RTC</td>
</tr>
<tr>
<td>L_SOURCE</td>
<td>L_IRQ</td>
<td>Selection of the source to enter and exit a low power consumption mode: IRQ</td>
</tr>
<tr>
<td>L_SLEEP</td>
<td>0</td>
<td>Low power consumption mode: Sleep mode</td>
</tr>
<tr>
<td>L_SOFT_STANDBY</td>
<td>1</td>
<td>Low power consumption mode: Software standby mode</td>
</tr>
<tr>
<td>L_DEEP_STANDBY</td>
<td>2</td>
<td>Low power consumption mode: Deep software standby mode</td>
</tr>
<tr>
<td>L_MODE</td>
<td>L_SLEEP</td>
<td>Selection of low power consumption mode: Sleep mode</td>
</tr>
<tr>
<td>WAIT_tdEA</td>
<td>300</td>
<td>td(E-A) wait time (max. 15 µs) Wait time ÷ ICLK (20 MHz) cycles = 15 ÷ 0.05 = 300</td>
</tr>
</tbody>
</table>

5.5 Variables

Table 5.4 lists the Global Variables.

Table 5.4 Global Variables

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function Used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>initial_end</td>
<td>Initial setting end flag</td>
<td>sleep_mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Processing</td>
<td>software_standby_mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Completed</td>
<td>deep_standby_mode</td>
</tr>
<tr>
<td></td>
<td>enable_low_power</td>
<td>Enable flag for transition to a low power consumption mode</td>
<td>Excep_ICU_IRQ1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Transition disabled</td>
<td>Excep_LVD_LVD2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Transition enabled</td>
<td>Excep_RTC_PRD</td>
</tr>
</tbody>
</table>
5.6 Functions
Table 5.5 lists the Functions Used in the Sample Code.

Table 5.5 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>port_init</td>
<td>Port initialization</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>peripheral_init</td>
<td>Peripheral function initialization</td>
</tr>
<tr>
<td>sleep_mode</td>
<td>Transition to sleep mode</td>
</tr>
<tr>
<td>software_standby_mode</td>
<td>Transition to software standby mode</td>
</tr>
<tr>
<td>deep_standby_mode</td>
<td>Transition to deep software standby mode</td>
</tr>
<tr>
<td>irq_init</td>
<td>IRQ initialization</td>
</tr>
<tr>
<td>lvd_init</td>
<td>LVD initialization</td>
</tr>
<tr>
<td>rtc_init</td>
<td>RTC initialization</td>
</tr>
<tr>
<td>Excep_ICU_IRQ1</td>
<td>IRQ1 interrupt handling</td>
</tr>
<tr>
<td>Excep_LVD_LVD1</td>
<td>LVD1 interrupt handling</td>
</tr>
<tr>
<td>Excep_LVD_LVD2</td>
<td>LVD2 interrupt handling</td>
</tr>
<tr>
<td>Excep_RTC_PRD</td>
<td>RTC_PRD interrupt handling</td>
</tr>
</tbody>
</table>

5.7 Function Specifications
The following tables list the sample code function specifications.

main

<table>
<thead>
<tr>
<th>Outline</th>
<th>Main processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void main(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Enters a low power consumption mode after the initial setting has been performed.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>

port_init

<table>
<thead>
<tr>
<th>Outline</th>
<th>Port initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void port_init(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Initializes ports.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return Value</td>
<td>None</td>
</tr>
</tbody>
</table>
### 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**  
Configures 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. Refer to the RX21A Group Initial Setting Rev. 1.00 application note for details on this function.

### R_INIT_NonExistentPort

**Outline**  
Nonexistent port initialization

**Header**  
r_init_non_existent_port.h

**Declaration**  
void R_INIT_NonExistentPort(void)

**Description**  
Initializes port direction registers for ports that do not exist in products with less than 100 pins.

**Arguments**  
None

**Return Value**  
None

**Remarks**  
The number of pins in the sample code is set for the 100-pin package (PIN_SIZE=100). 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.  
Refer to the RX21A Group Initial Setting Rev. 1.00 application note for details on this function.

### R_INIT_Clock

**Outline**  
Clock initialization

**Header**  
r_init_clock.h

**Declaration**  
void R_INIT_Clock(void)

**Description**  
Initializes the clock.

**Arguments**  
None

**Return Value**  
None

**Remarks**  
The sample code selects processing with the following settings:  
- System clock: Main clock
- Operating power control mode: Middle-speed operating mode 1A
- HOCO and PLL: Not used

Refer to the RX21A Group Initial Setting Rev. 1.00 application note for details on this function.

### peripheral_init

**Outline**  
Peripheral function initialization

**Header**  
None

**Declaration**  
void peripheral_init(void)

**Description**  
Initializes peripheral functions used.

**Arguments**  
None

**Return Value**  
None
### sleep_mode

**Outline**  
Transition to sleep mode

**Header**  
None

**Declaration**  
void sleep_mode(void)

**Description**  
Configures settings to enter sleep mode.

**Arguments**  
None

**Return Value**  
None

### software_standby_mode

**Outline**  
Transition to software standby mode

**Header**  
None

**Declaration**  
void software_standby_mode(void)

**Description**  
Configures settings to enter software standby mode.

**Arguments**  
None

**Return Value**  
None

### deep_standby_mode

**Outline**  
Transition to deep software standby mode

**Header**  
None

**Declaration**  
void deep_standby_mode(void)

**Description**  
Configures settings to enter deep software standby mode.

**Arguments**  
None

**Return Value**  
None

### irq_init

**Outline**  
IRQ initialization

**Header**  
None

**Declaration**  
void irq_init(void)

**Description**  
Performs the IRQ initialization.

**Arguments**  
None

**Return Value**  
None

### lvd_init

**Outline**  
LVD initialization

**Header**  
None

**Declaration**  
void lvd_init(void)

**Description**  
Performs the LVD initialization.

**Arguments**  
None

**Return Value**  
None

### rtc_init

**Outline**  
RTC initialization

**Header**  
None

**Declaration**  
void rtc_init(void)

**Description**  
Performs the RTC initialization.

**Arguments**  
None

**Return Value**  
None
<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Header</th>
<th>Declaration</th>
<th>Description</th>
<th>Arguments</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excep_ICU_IRQ1</td>
<td>IRQ1 interrupt handling</td>
<td>None</td>
<td>void Excep_ICU_IRQ1(void)</td>
<td>Performs the IRQ1 interrupt handling.</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Excep_LVD_LVD1</td>
<td>LVD1 interrupt handling</td>
<td>None</td>
<td>void Excep_LVD_LVD1(void)</td>
<td>Performs the LVD1 interrupt handling.</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Excep_LVD_LVD2</td>
<td>LVD2 interrupt handling</td>
<td>None</td>
<td>void Excep_LVD_LVD2(void)</td>
<td>Performs the LVD2 interrupt handling.</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Excep_RTC_PRD</td>
<td>RTC.LVD interrupt handling</td>
<td>None</td>
<td>void Excep_RTC_PRD(void)</td>
<td>Performs the RTC.LVD interrupt handling.</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>
5.8 Flowcharts

5.8.1 Main Processing

Figure 5.7 shows the Main Processing.

```
main

- Disable maskable interrupts
- I flag ← 0

- Port initialization
  port_init()

- Stop processing for active peripheral functions after a reset
  R_INIT_StopModule()

- Nonexistent port initialization
  R_INIT_NonExistentPort()

- Clock Initialization
  R_INIT_Clock()

- Peripheral function initialization
  peripheral_init()

- Enable maskable interrupts
  I flag ← 1

  When L_SLEEP (sleep mode) is set to L_MODE (low power consumption mode)
  Transition to sleep mode
  sleep_mode()

  When L_SOFT_STANDBY (software standby mode) is set to L_MODE
  Transition to software standby mode
  software_standby_mode()

  When L_DEEP_STANDBY (deep software standby mode) is set to L_MODE
  Transition to deep software standby mode
  deep_standby_mode()
```

Figure 5.7 Main Processing
### 5.8.2 Port Initialization

Figure 5.8 shows the Port Initialization.

```
port_init

Turn off LEDs

PORTH.PODR register
B0 bit ← 1: LED1: High
B1 bit ← 1: LED2: High
B2 bit ← 1: LED3: High

PORTH.PDR register
B0 bit ← 1: LED1: Output
B1 bit ← 1: LED2: Output
B2 bit ← 1: LED3: Output
```

Figure 5.8 Port Initialization

### 5.8.3 Peripheral Function Initialization

Figure 5.9 shows the Peripheral Function Initialization.

```
peripheral_init

When L_IRQ (IRQ) is set to the L_SOURCE (source to enter and exit low power consumption mode)

IRQ initialization
irq_init()

When L_LVD (LVD) is set to the L_SOURCE

LVD initialization
lvd_init()

When L_RTC (RTC) is set to the L_SOURCE

RTC initialization
rtc_init()

return
```

Figure 5.9 Peripheral Function Initialization
5.8.4 Transition to Sleep Mode

Figure 5.10 shows the Transition to Sleep Mode.

![Transition to Sleep Mode Diagram]

- **sleep_mode**
  - PORTH.PODR register
    - B0 bit ← 0: LED1: Low
  - Set the initial setting end flag
    - initial_end ← 1
  - Wait period for transition to sleep mode
    - I flag ← 0
  - Disable maskable interrupts
  - Disable write protection
  - MSTPCRA register
    - ACSE bit ← 0: All-module clock stop mode is disabled.
  - SBYCR register
    - SSBY bit ← 0: Transition to sleep mode is made after the WAIT instruction is executed.
  - Disable switching the clock source when exiting sleep mode
  - Enable write protection
  - PRCR register ← A502h
    - PRC1 bit = 1: Enables writing to the registers related to the low power consumption function.
  - RSTCKCR register ← 00h
    - RSTCKEN bit = 0: Clock source switching at sleep mode cancellation is disabled.
  - Turn off LED1 and turn on LED2 (1)
    - PORTH.PODR register
      - B0 bit ← 1: LED1: High
      - B1 bit ← 0: LED2: Low
  - Execute the WAIT instruction
    - Enters sleep mode.
  - Turn off LED2 and turn on LED3
    - PORTH.PODR register
      - B1 bit ← 1: LED2: High
      - B2 bit ← 0: LED3: Low
  - return

**Note:**
1. Read the register written immediately before the WAIT instruction is executed to confirm the written value can be read correctly.

Figure 5.10 Transition to Sleep Mode
### 5.8.5 Transition to Software Standby Mode

Figure 5.11 shows the Transition to Software Standby Mode.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Disable maskable interrupts&lt;br&gt;<code>i_flag ← 0</code></td>
</tr>
<tr>
<td>2.</td>
<td>Disable write protection&lt;br&gt;<code>PRC0 bit = 1</code>: Enables writing to the registers related to the clock generation circuit.&lt;br&gt;<code>PRC1 bit = 1</code>: Enables writing to the registers related to the low power consumption function.</td>
</tr>
<tr>
<td>3.</td>
<td>Set the transition destination after the&lt;br&gt;<code>WAIT</code> instruction is executed&lt;br&gt;<code>SBYCR</code> register&lt;br&gt;<code>SSBY bit ← 1</code>: Transition to software standby mode is made after the <code>WAIT</code> instruction is executed.&lt;br&gt;<code>DPSBYCR</code> register&lt;br&gt;<code>DPSBY bit ← 0</code>: Transition to software standby mode is made after the <code>WAIT</code> instruction is executed.</td>
</tr>
<tr>
<td>4.</td>
<td>Disable the oscillation stop detection&lt;br&gt;<code>OSTDCR</code> register&lt;br&gt;<code>OSTDE bit ← 0</code>: Oscillation stop detection function is disabled.</td>
</tr>
<tr>
<td>5.</td>
<td>Enable write protection</td>
</tr>
<tr>
<td>6.</td>
<td>Disable the DMAC and DTC&lt;br&gt;<code>DMST register&lt;br&gt;</code>DMST bit ← 0<code>: DMAC activation is disabled.&lt;br&gt;</code>DTCST register&lt;br&gt;<code>DTCST bit ← 0</code>: DTC module stop</td>
</tr>
<tr>
<td>7.</td>
<td>Turn off LED1 and turn on LED2&lt;br&gt;<code>PORTH.PODR register&lt;br&gt;</code>B0 bit ← 0<code>: LED1: Low&lt;br&gt;</code>B1 bit ← 0`: LED2: Low</td>
</tr>
<tr>
<td>8.</td>
<td>Execute the <code>WAIT</code> instruction&lt;br&gt;Enters software standby mode.</td>
</tr>
<tr>
<td>9.</td>
<td>Turn off LED2 and turn on LED3&lt;br&gt;<code>PORTH.PODR register&lt;br&gt;</code>B1 bit ← 1<code>: LED2: High&lt;br&gt;</code>B2 bit ← 0`: LED3: Low</td>
</tr>
</tbody>
</table>

**Note:**
1. Read the register written immediately before the `WAIT` instruction is executed to confirm the written value can be read correctly.

---

Figure 5.11 Transition to Software Standby Mode
5.8.6 Transition to Deep Software Standby Mode

Figure 5.12 to Figure 5.14 show the Transition to Deep Software Standby Mode.

![Diagram of Transition to Deep Software Standby Mode]

- Reads RSTSR0 register.
  - DPSRSTF flag: 0: Deep software standby mode cancellation not requested by an interrupt.
  - 1: Deep software standby mode cancellation requested by an interrupt.
- Has the MCU exited deep software standby mode?
  - Yes
  - No
  - Turn on LED1
  - Set the initial setting end flag
  - Wait period for transition to deep software standby mode
  - Disable the DMAC and DTC
  - Disable write protection
  - Disable the oscillation stop detection function
  - Set the transition destination after the WAIT instruction is executed
  - Turn off LED1 and turn on LED2
  - Configure pin states after exiting deep software standby mode

Figure 5.12 Transition to Deep Software Standby Mode (1/3)
Enable write protection (2)
PRCR register ← 0x500h
PRC0 bit = 0: Disables writing to the registers related to the clock generation circuit.
PRC1 bit = 0: Disables writing to the registers related to the low power consumption function.

Execute the WAIT instruction
Enters deep software standby mode.
return

Notes:
1. After reading registers DPSIFR0 and DPSIFR2, write 00h to these registers.
2. Read the register written immediately before the WAIT instruction is executed to confirm the written value can be read correctly.

Figure 5.13 Transition to Deep Software Standby Mode (2/3)
Determine the exit request (1)  
Read the DPSIFR0 register  
Read the DPSIFR2 register  

Turn off LED1 and LED2, and turn on LED3  
PORTH.PODR register  
B0 bit ← 1: LED1: High  
B1 bit ← 1: LED2: High  
B2 bit ← 0: LED3: Low  

Disable write protection  
PRCR register ← A502h  
PRC1 bit = 1: Enables writing to the registers related to the low power consumption function.  

Cancel the I/O port retention  
DPSBYCR register  
IOKEEP bit ← 0  

Enable write protection  
PRCR register ← A500h  
PRC1 bit = 0: Disables writing to the registers related to the low power consumption function.  

Processing according to the exit request (1)  

return

Note:  
1. Processing for exit requests is not performed in the sample code. Add a program as required.

Figure 5.14 Transition to Deep Software Standby Mode (3/3)
5.8.7 IRQ Initialization

Figure 5.15 shows the IRQ Initialization.

![Diagram of IRQ Initialization]

Figure 5.15 IRQ Initialization
### 5.8.8 LVD Initialization

Figure 5.16 and Figure 5.17 show the LVD Initialization.

1. **IER0B register**
   - IEN0 bit ← 0: LVD1 interrupt request is disabled.
   - IEN1 bit ← 0: LVD2 interrupt request is disabled.

2. **PRC3 register**
   - PRC3 bit = 1: Enables writing to registers related to the LVD.

3. **LVDLVL register** ← F8h
   - LVD1LVL[3:0] bits = 1000b: Voltage detection 1 level : 2.95 V
   - LVD2LVL[3:0] bits = 1001b: Voltage detection 2 level : 2.80 V

4. **Select the voltage detection level**
   - Set the LVCMPCR register
     - Set registers LVD1CR0 and LVD2CR0
       - LVD1CR0 register ← 02h
         - LVD1RIE bit = 0: Voltage monitoring 1 interrupt disabled
         - LVD1DFDIS bit = 1: Digital filter disable
         - LVD1CMPE bit = 0: Voltage monitoring 1 circuit comparison results output disabled
         - LVD1RI bit = 0: Voltage monitoring 1 interrupt at Vdet1 passage
       - LVD2CR0 register ← 02h
         - LVD2RIE bit = 0: Voltage monitoring 2 interrupt disabled.
         - LVD2DFDIS bit = 1: Digital filter disabled
         - LVD2CMPE bit = 0: Voltage monitoring 2 circuit comparison results output disabled
         - LVD2RI bit = 0: Voltage monitoring 2 interrupt at Vdet2 passage
     - Set registers LVD1CR1 and LVD2CR1
       - LVD1CR1 register ← 04h
         - LVD1DTSEL[1:0] bits = 00b: When VCC ≥ Vdet1 (rise) is detected
         - LVD1IRQSEL bit = 1: Maskable interrupt
       - LVD2CR1 register ← 05h
         - LVD2DTSEL[1:0] bits = 01b: When VCC < Vdet2 (drop) is detected
         - LVD2IRQSEL bit = 1: Maskable interrupt
     - Enable voltage detection 1 and 2 circuits
   - Wait at least td(E-A)

---

**Figure 5.16 LVD Initialization (1/2)**
Clear the voltage change detection flags of voltage monitoring 1 and 2.

LVD1SR register
LVD1DET flag ← 0
LVD2SR register
LVD2DET flag ← 0

Enable interrupt requests
IEN0 bit ← 1: LVD1 interrupt request is enabled.
IEN1 bit ← 1: LVD2 interrupt request is enabled.

Enable outputting comparison results of voltage monitoring 1 and 2 circuits
LVD1CR0 register ← 06h
LVD1CMPE bit = 1: Voltage monitoring 1 circuit comparison results output enabled
LVD2CR0 register ← 06h
LVD2CMPE bit = 1: Voltage monitoring 2 circuit comparison results output enabled

Enable voltage monitoring 1 and 2 interrupts
LVD1R0 register ← 07h
LVD1RIE bit = 1: Voltage monitoring 1 interrupt enabled
LVD2CR0 register ← 07h
LVD2RIE bit = 1: Voltage monitoring 2 interrupt enabled

Disable write protection
PRC3 bit = 0: Disables writing to registers related to the LVD.

Set the interrupt priority level
IPR088 register
IPR[3:0] bits ← 1111b: LVD1 interrupt priority level is set to level 15.
IPR089 register
IPR[3:0] bits ← 1111b: LVD2 interrupt priority level is set to level 15.

Clear interrupt requests
IR088 register
IR flag ← 0: LVD1 interrupt request is not generated.
IR089 register
IR flag ← 0: LVD2 interrupt request is not generated.

Wait for two or more PCLKB cycles
Holds two or more PCLKB cycles by reading the LVD1CR0 register.

Figure 5.17  LVD Initialization (2/2)
5.8.9 RTC Initialization

Figure 5.18 shows the RTC Initialization.

![Diagram of RTC Initialization](image)

Notes:
1. After writing to the RCR1 register and RCR2.START bit, confirm that the written value can be read correctly.
2. After writing to the RCR2.HR24 bit, dummy read the register three times.
5.8.10 IRQ1 Interrupt Handling

Figure 5.19 shows the IRQ1 Interrupt Handling.

![ IRQ1 Interrupt Handling Diagram ]

5.8.11 LVD1 Interrupt Handling

Figure 5.20 shows the LVD1 Interrupt Handling.

![ LVD1 Interrupt Handling Diagram ]
5.8.12 LVD2 Interrupt Handling
Figure 5.21 shows the LVD2 Interrupt Handling.

![Diagram of LVD2 Interrupt Handling]

- Excep_LVD_LVD2
  - Disable write protection
    - PRCR register ← A508h
      - PRC3 bit = 1: Enable writing to registers related to the LVD.
    - Disable the voltage monitoring 2 interrupt
      - LVD2CR0 register ← 06h
      - LVD2RIE bit = 0
  - Clear the voltage monitoring 2 voltage change detection flag
    - LVD2SR register
      - LVD2DET flag ← 0
  - Wait for two or more PCLKB cycles
    - Holds two or more PCLKB cycles by reading the LVD2CR0 register.
  - Enable the voltage monitoring 2 interrupt
    - LVD2CR0 register ← 07h
      - LVD2RIE bit = 1
  - Enable write protection
    - PRCR register ← A500h
      - PRC3 bit = 0: Disable writing to registers related to the LVD.

Has the initial setting completed?

- No
  - Set the enable flag for transition to low power consumption mode
  - Clear the enable flag for transition to low power consumption mode
- Yes
  - return

Figure 5.21 LVD2 Interrupt Handling

5.8.13 RTC.PRD Interrupt Handling
Figure 5.22 shows the RTC.PRD Interrupt Handling.

![Diagram of RTC.PRD Interrupt Handling]

- Excep_RTC_PRD
  - Has the initial setting completed?
    - No
      - Set the enable flag for transition to low power consumption mode
      - Clear the enable flag for transition to low power consumption mode
    - Yes
      - return

Figure 5.22 RTC.PRD Interrupt Handling
### 6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

### 7. Reference Documents

**User’s Manual: Hardware**
- RX21A Group User’s Manual: Hardware Rev.1.00 (R01UH0251EJ)
  - 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](http://www.renesas.com)

Inquiries
- [http://www.renesas.com/contact/](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>Dec. 16, 2013</td>
<td>First edition issued</td>
</tr>
</tbody>
</table>

---

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.

1. 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 LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

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

3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   - The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

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

5. Differences between Products
   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.
   - 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.
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 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 depend 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, 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.

Renesas Electronics products are neither intended nor 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 implants etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.

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 malfunction under certain use 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 microcomputer 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 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 relating to military applications or 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.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.