RX111 Group

Using the RTCA to Exit Software Standby Mode Firmware Integration Technology

R01AN1821EJ0100
Rev. 1.00
Dec. 16, 2013

Abstract

This document describes a process that uses the firmware integration technology (FIT) module. This document describes how to exit software standby mode at regular intervals, and to obtain the current time information using the RTC module.

Products

RX111 Group, 64-Pin Package, ROM Capacities: 16 Kbytes to 128 Kbytes
RX111 Group, 48-Pin Package, ROM Capacities: 16 Kbytes to 128 Kbytes
RX111 Group, 40-Pin Package, ROM Capacities: 16 Kbytes to 64 Kbytes

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 4
3. Reference Application Notes 4
4. Hardware 5
4.1 Hardware Configuration 5
4.2 Pins Used 5
5. Software 6
5.1 Operation Overview 7
5.2 File Composition 9
5.3 Constants 10
5.4 Structure/Union List 11
5.5 Functions 12
5.6 Function Specifications 13
5.7 Flowcharts 19
5.7.1 Main Processing 19
5.7.2 Alarm (ALM) Interrupt Handling 20
5.7.3 Processing Before Entering Software Standby Mode 21
5.7.4 IRQ0 Interrupt Handling (Processing After a Switch is Pressed) 21
6. Appendix 22
6.1 Compile Option Settings 22
7. Sample Code 23
8. Reference Documents 23

1. Specifications

The RTC can be used to obtain the time information as well as intermittently exit software standby mode.

After a reset, if the MCU performs cold start processing, the sub-clock oscillator and the RTC are initialized. Then, the input level of the interrupt request pin is monitored. If the signal is low, the MCU enters software standby mode. Subsequently, the MCU exits software standby mode when one of the following occurs:

- An alarm (ALM) interrupt is generated every minute
- The input level to the interrupt request pin becomes low again

Until the input level to the interrupt request pin becomes low again, an alarm (ALM) interrupt is generated every minute, and the MCU exits software standby mode. Then, the RTC obtains the time information, and the MCU re-enters software standby mode.

Restrictions

This application note is not applicable when performing a warm start.

- RTC count source: Sub-clock
- VBATT pin: Connect to the VCC pin
- Intermittent period: 1 minute

Table 1.1 lists the Peripheral Functions and Their Applications.

In this document, when not in software standby mode, the MCU is assumed to be in normal mode.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
RTCA	The RTCA acts as a clock and initiates the MCU's exit from software standby mode.
IRQ0	The IRQ0 is an external input used to enter and exit software standby mode.

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

Item	Contents
MCU used	R5F51115ADFM (RX111 Group)
Operating frequencies	• Main clock: 16 MHz • Sub-clock: 32.768 kHz • PLL: 32 MHz (main clock divided by 4 and multiplied by 8) • LOCO: 4 MHz • System clock (ICLK): 32 MHz (PLL divided by 1) • Peripheral module clock B (PCLKB): 48 MHz (PLL divided by 1)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics Corporation e2 studio Version 2.0.1.6
C compiler	Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.2.00
	Compile options Refer to the appendix in chapter 6.
iodefine.h version	V.0.9A
Endian	Little endian
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Sample code version	Version 1.00
FIT modules used	r_bsp: Version 2.30 r_cgc_rx: Version 1.00 r_irq_rx: Version 1.00 r_mpc_rx: Version 1.00 r_lpc_rx: Version 1.00 r_rtc_rx: Version 1.00
Board used	Renesas Start Kit for RX111 (product part no.: R0K505111S001BE)

3. Reference Application Notes

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

- RX Family Board Support Package Module Using Firmware Integration Technology Rev.2.30 (R01AN1685EU)
- RX Family Clock Generation Circuit Module Using Firmware Integration Technology Rev.1.00 (R01AN1727EU)
- RX Family IRQ (External Pin Interrupt Request) Module Using Firmware Integration Technology Rev.1.00 (R01AN1668EU)
- RX Family Multi-Function Pin Controller Module Using Firmware Integration Technology Rev.1.00 (R01AN1724EU)
- RX Family Low Power Consumption Module Using Firmware Integration Technology Rev.1.00 (R01AN1728EU)
- RX Family RTC Module Using Firmware Integration Technology Rev.1.00 (R01AN1817ES)

The API functions of the above FIT modules are used in the sample code for this application note. The revision numbers of the reference application notes are current as of the creation of this application note. However the latest version is always recommended. Visit the Renesas Electronics Corporation website to check and download the latest version.

4. Hardware

4.1 Hardware Configuration

Figure 4.1 shows a Connection Example.

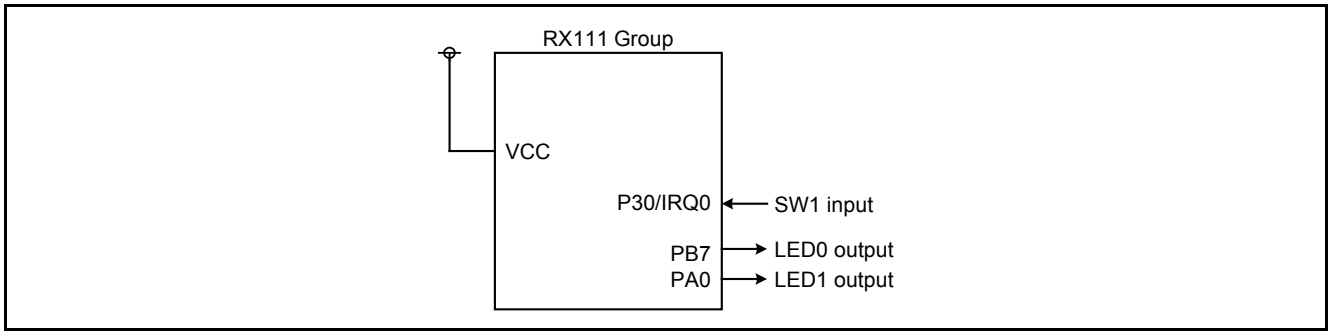


Figure 4.1 Connection Example

4.2 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

This document assumes the 64-pin package is used. When using products with less than 64-pins, select pins applicable to that product.

Table 4.1 Pins Used and Their Functions

Pin Name	I/O	Function
P30/IRQ0	Input	Input from SW1 for entering and exiting software standby mode
PB7	Output	LED0 output: ON in normal mode, OFF in software standby mode
PA0	Output	LED1 output: Inverted every minute

5. Software

In the sample code accompanying this document, after a reset, the IR flag for the IRQ0 interrupt is monitored. If the number of IRQ0 interrupts is odd number of times, the MCU enters software standby mode. If the number of IRQ0 interrupts is even number of times, the MCU returns to normal mode.

The alarm (ALM) interrupt is generated every minute regardless of the MCU being in normal mode or software standby mode, the time information is obtained, and the alarm time is updated. When the MCU exits software standby mode by an alarm (ALM) interrupt, it re-enters software standby mode.

Settings for the peripheral functions used are listed in Table 5.1 and Table 5.2.

Table 5.1 RTC Settings

Setting	Function
Count source	Operates using the sub-clock
Initial time setting	00:00:00, January 1, 2013 (Tuesday)
Hours mode	24-hour mode
RTCOUT output	Do not use
Error adjustment	Not used
Interrupt	Alarm (ALM) interrupt: Generated every minute

Table 5.2 IRQ0 Settings

Setting	Function
Detection method	Falling edge
Digital filter	Enabled
Sampling clock for the digital filter	PCLK/64
Interrupt	IRQ0 external interrupt used

5.1 Operation Overview

(1) Initial settings
After a reset, if the RSTSR1.CWSF flag is 0 (cold start), the sub-clock oscillator and RTC are initialized. In the RTC initialization, the time and alarm are set, and the alarm (ALM) interrupt request is enabled. Settings are also performed for the MCU to enter software standby mode, and LED0 and LED1 are turned on. Subsequently, an alarm (ALM) interrupt is generated every minute, the time information is obtained, the alarm time is updated, and the LED1 signal is inverted. The IR flag of the IRQ0 interrupt is monitored.

(2) Entering software standby mode
When the IR flag of the IRQ0 interrupt becomes 1, LED0 turns off, the WAIT instruction is executed and the MCU enters software standby mode.

(3) Exiting software standby mode
The MCU exits software standby mode if an alarm (ALM) interrupt is generated, or if an IRQ0 interrupt is generated while monitoring the IR flag of the IRQ0 interrupt, and the number of the IRQ0 interrupts is even number of times.

(3-1) Exit initiated by an alarm (ALM) interrupt
After the MCU exits software standby mode, LED0 turns on, the signal for LED1 is inverted, and after 1/128 seconds elapses, the RTC time information is read. Also, when the IR flag of the IRQ0 interrupt is not 1, the MCU re-enters software standby mode. Step (3) is repeated until an IRQ0 interrupt is generated an even number of times.

(3-2) Exit initiated by an IRQ0 interrupt
After the MCU exits software standby mode, LED0 is turned on.

Figure 5.1 shows the Timing Diagram.

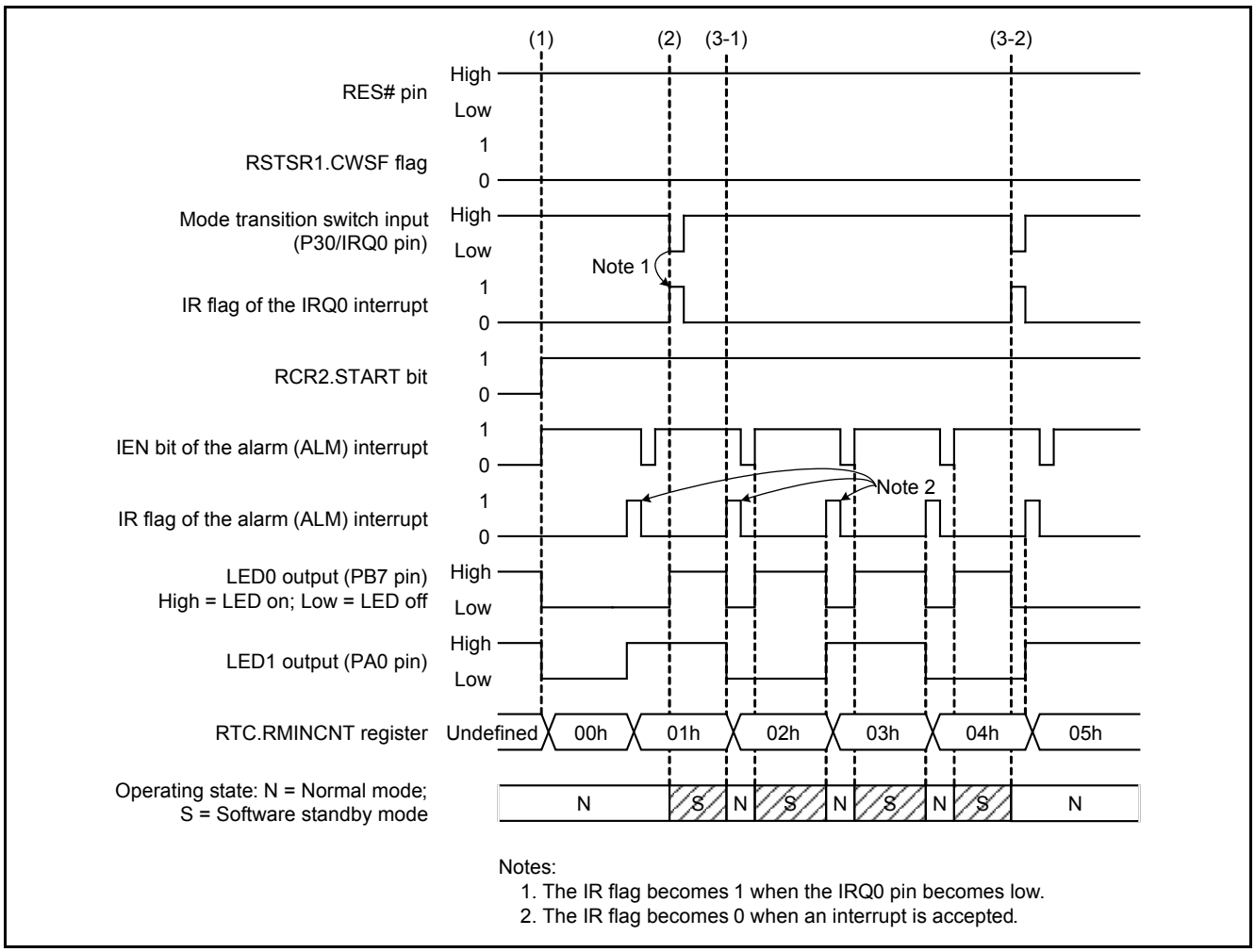


Figure 5.1 Timing Diagram

5.2 File Composition

Table 5.3 lists the File Used in the Sample Code, and Table 5.4 lists the FIT Modules Used in the Sample Code. Files generated by the integrated development environment are not included in this table.

Table 5.3 File Used in the Sample Code

File Name	Outline	Remarks
main.c	Main processing	

Table 5.4 FIT Modules Used in the Sample Code

Module Name	Outline	Remarks
CGC module	Module associated with the clock generation circuit	Used for setting and changing clocks
IRQ module	Module associated with external interrupts	Used for the IRQ0 interrupt
MPC module	Module associated with the multi-function pin controller	Used for the SW1 pin settings
RTC module	Module associated with the RTC	Used for setting the time and alarms
LPC module	Module associated with low power consumption	Used for entering and exiting software standby mode

5.3 Constants

Table 5.5 to Table 5.8 list the constants used in the sample code. For setting values of constants not listed below, their default values are used.

Table 5.5 Constants Used in the Sample Code for the BSP (r_bsp_config.h)

Constant Name	Setting Value	Contents
BSP_CFG_USE_CGC_MODULE	0	CGC module not used for the clock initial settings
BSP_CFG_CLOCK_SOURCE	4	System clock at a rising edge: PLL
BSP_CFG_XTAL_HZ	1600000	Main clock frequency
BSP_CFG_PLL_DIV	4	PLL division ratio
BSP_CFG_PLL_MUL	8	PLL multiplication factor
BSP_CFG_ICK_DIV	1	ICLK division ratio
BSP_CFG_PCKB_DIV	1	PCLKB division ratio
BSP_CFG_PLKD_DIV	1	PCLKD division ratio
BSP_CFG_FCK_DIV	1	FCLK division ratio

Table 5.6 Constants Used in the Sample Code for the CGC Module (r_cgc_rx111_config.h)

Constant Name	Setting Value	Contents
CGC_CFG_SUBCLOCK_DRIVE	0x02	Low drive capacity

Table 5.7 Constants Used in the Sample Code for the RTC Module (r_rtc_rx_config.h)

Constant Name	Setting Value	Contents
RTC_CFG_CALCULATE_YDAY	0	Year is omitted

Table 5.8 Constants Used in the Sample Code for the IRQ Module (r_irq_rx_config.h)

Constant Name	Setting Value	Contents
IRQ_CFG_USE_IRQ0	1	IRQ0: Used
IRQ_PORT_IRQ0	PORT3	IRQ0 port: Port 3 used
IRQ_PORT_BIT_IRQ0	IRQ_BIT0	IRQ0 bit: Bit 0 used
IRQ_CFG_FILT_EN_IRQ0	1	IRQ0 digital filter: Enabled
IRQ_CFG_FILT_PLCK_IRQ0	IRQ_CFG_PCLK_DIV64	Sampling clock for the digital filter of IRQ0: Divided by 64

5.4 Structure/Union List

Figure 5.2 to Figure 5.4 show the structures used in the sample code.

```
typedef struct
{
   int tm_sec;         /* Seconds (0-59) */
   int tm_min;         /* Minute (0-59) */
   int tm_hour;        /* Hour (0-23) */
   int tm_mday;        /* Day of the month (1-31) */
   int tm_mon;         /* Month (0-11, 0 = January) */
   int tm_year;        /* Year since 1900 */
   int tm_wday;        /* Day of the week (0-6, 0 = Sunday) */
   int tm_ydat;        /* Day of the year (0-365) */
   int tm_isdst;       /* Day Light Saving enable (>0), disabled (=0), or unknown (<0) */
}  tm_t;
```

Figure 5.2 Structure of the Time and Alarm Settings (R_RTC_Open and R_RTC_Control Functions)

```
typedef struct
{
   uint32_t;              configuration;
   uint16_t;              periodic;
   uint8_t;               int_priority_alm;     // alm INT priority; 1 = low, 15 = high
   uint8_t;               int_priority_prd;     // prd INI priority; 1 = low, 15 = high
   RTCCallBackFunc;       p_callback;
}  rtc_init_t;
```

Figure 5.3 Structure of the RTC Initial Settings (R_RTC_Open Function)

```
typedef struct
{
   uint32_t;              configuration;
   uint16_t;              periodic;
   uint16_t;              update_selection;
}  rtc_cfg_t;
```

Figure 5.4 Structure of the RTC Information Update (R_RTC_Control Function)

Figure 5.5 shows the structure used in the MPC module.

```
typedef struct
{
   uint8_t    pin_function;    //which peripheral function is assigned to this pin
   bool       irq_enable       //This pin is used as IRQ pin.
   bool       analog_enable    //This pin is used as ADC input, DAC output, or for LVD (CMPA2)
}  mpc_config_t
```

Figure 5.5 Structure of the External Interrupt Pin Settings (R_MPC_Write Function)

5.5 Functions

Table 5.9 lists the Functions.

Table 5.9 Functions

Function Name	Outline
main	Main processing
rtc_callback	Processing when an alarm (ALM) interrupt is generated
lpc_callback	Processing before entering software standby mode
irq_callback	IRQ0 interrupt handling
R_CGC_Open	CGC initialization (sub-clock drive capacity settings)
R_CGC_ClockStart	Start clock oscillation
R_RTC_Open	RTC initialization
R_RTC_Control	Update RTC information
R_RTC_Read	Read RTC information
R_MPC_Write	External interrupt pin settings
R_IRQ_Open	IRQ initialization
R_LPC_LowPowerModeConfigure	Settings to enter low power consumption mode
R_LPC_LowPowerModeActivate	Enter low power consumption mode

5.6 Function Specifications

The following tables list the sample code function specifications.

main	
Outline	Main processing
Header	None
Declaration	void main(void)
Description	After the sub-clock starts oscillating, the RTC is initialized, and the external interrupts are initialized, the settings to enter software standby mode are configured, and LED0 and LED1 are turned on. Then, when SW1 is pressed, the MCU enters software standby mode. The alarm (ALM) interrupt is generated every minute regardless of the MCU being in normal mode or software standby mode, the RTC time information is read, and the alarm time is updated. When the MCU exits software standby mode by the alarm (ALM) interrupt, the MCU re-enters software standby mode.
Arguments	None
Return Value	None

rtc_callback	
Outline	Processing when an alarm (ALM) interrupt is generated
Header	r_rtc_rx_if.h
Declaration	void rtc_callback (void * p_flag)
Description	This processing is performed after an alarm (ALM) interrupt is generated. In this processing, LED1 output is inverted, LED0 is turned on if the MCU exits software standby mode due to an alarm (ALM) interrupt while the MCU is entering software standby mode, and a software loop is performed for 1/128 second. Then, the RTC information is read, and the alarm time is updated after one minute elapses.
Arguments	p_flag: This pointer indicates when an alarm (ALM) interrupt or a periodic interrupt is generated.
Return Value	None

lpc_callback	
Outline	Processing before entering software standby mode
Header	r_lpc_rx_if.h
Declaration	void lpc_callback (void * pdata)
Description	LED0 is turned off and the software standby transition flag becomes 1 (the software standby mode flag determines whether or not the MCU is in software standby mode).
Arguments	pdata: Pointer for the function that is called before the MCU enters low power consumption mode.
Return Value	None

irq_callback	
Outline	IRQ0 interrupt handling
Header	r_irq_rx_if.h
Declaration	void irq_callback (void * pargs)
Description	After SW1 is pushed, the counter that counts the number of times SW1 is pushed increments, LED0 is turned on, and the software standby transition flag is set to 0.
Arguments	pargs: Pointer for the callback function.
Return Value	None

R_CGC_Open	
Outline	CGC initialization (sub-clock drive capacity settings)
Header	None
Declaration	cgc_err_t R_CGC_Open(void)
Description	For the CGC initialization, the sub-clock is stopped, the sub-clock drive capacity is set, the HOCO oscillation stabilization wait time is set, the main clock oscillation stabilization wait time is set, and the main clock drive capacity is set.
Arguments	None
Return Value	Success value or error value CGC_SUCCESS: When the drive capacity value configuration is complete
Remarks	In the sample code, the sub-clock drive capacity is set to low. For details on this function, refer to the RX Family Clock Generation Circuit Module Using Firmware Integration Technology application note, Rev.1.00 (R01AN1727EU).

R_CGC_ClockStart	
Outline	Start clock oscillation
Header	None
Declaration	cgc_err_t R_CGC_(cgc_clock_t clock_source, cgc_clock_config_t* pClock_config)
Description	Start oscillating the clock specified by the first argument.
Arguments	cgc_clock_t clock_source: Clock to be oscillated cgc_clock_config_t* pClock_config: PLL division ratio and multiplication factor
Return Value	Success value or error value CGC_SUCCESS: Oscillation started successfully CGC_ERR_NOT_STABILIZED: Clock is not oscillating CGC_ERR_CLOCK_ACTIVE: Clock is already oscillating CGC_MAIN_OCO_INACTIVE: Main clock is stopped when PLL oscillates CGC_ERR_ILL_PARAM: Parameter error CGC_ERR_NULL_PTR: Second argument is NULL
Remarks	In the sample code, the sub-clock oscillates. For details on this function, refer to the RX Family Clock Generation Circuit Module Using Firmware Integration Technology application note, Rev.1.00 (R01AN1727EU).

R_RTC_Open	
Outline	RTC initialization
Header	r_rtc_if.h
Declaration	rtc_err_t R_RTC_Open(rtc_init * p_init tm_t * p_current tm_t * p_alarm)
Description	For the RTC initialization, the RTC is initialized, the initial time is set, and the alarm time is set.
Arguments	rtc_init * p_init: Pointer for the initialization structure. tm_t * p_current: Pointer for the time setting structure. tm_t * p_alarm: Pointer for the alarm time setting structure.
Return Value	Success value or error value CGC_SUCCESS: RTC is initialized successfully CGC_ERR_INIT: RTC is already open RTC_ERR_SUBCLOCK: Sub-clock is not oscillating RTC_ERR_ILL_PARAM: Parameter error RTC_ERR_ILL_CALLBACK: Callback function is incorrect RTC_ERR_ILL_PRIORITY: Interrupt priority level is incorrect RTC_ERR_ILL_CURRENT: Set time is incorrect RTC_ERR_ILL_ALARM: Alarm time is incorrect
Remarks	In the first argument of the sample code for this document, the minute alarm is enabled, periodic interrupts are disabled, and the alarm (ALM) interrupt priority level is set to 15. The time in the second argument is set to 00:00:00, Tuesday, January 1, 2013. The alarm time in the third argument is set to 00:01:00, Tuesday, January 1, 2013. For details on this function, refer to the RX Family RTC Module Using Firmware Integration Technology application note, Rev.1.00 (R01AN1817ES).

R_RTC_Control	
Outline	Update RTC information
Header	r_rtc_if.h
Declaration	rtc_err_t R_RTC_Control(rtc_cgc_t * p_cfg tm_t * p_current tm_t * p_alarm)
Description	RTC setting information is updated.
Arguments	rtc_cgc_t * p_cfg: Pointer for the RTC setting information structure. tm_t * p_current: Pointer for the time setting structure. tm_t * p_alarm: Pointer for the alarm time setting structure.
Return Value	Success value or error value RTC_SUCCESS: RTC is initialized successfully RTC_ERR_UNINIT: RTC is not initialized RTC_ERR_ILL_PARAM: Parameter error RTC_ERR_ILL_CURRENT: Set time is incorrect RTC_ERR_ILL_ALARM: Alarm time is incorrect
Remarks	In the sample code, after the RTC time information is read, the time setting is not changed, so FIT_NO_PTR is set to the second argument. The third argument passes the pointer for the structure which has the alarm time with 1-minute update. For details on this function, refer to the RX Family RTC Module Using Firmware Integration Technology application note, Rev.1.00 (R01AN1817ES).

R_RTC_Read	
Outline	Read RTC information
Header	r_rtc_if.h
Declaration	rtc_err_t R_RTC_Read(int8_t * status, tm_t * p_current tm_t * p_alarm)
Description	The current time and alarm time are read.
Arguments	int8_t * status: Storage for the RTC counter status (stopped/operating) tm_t * p_current: Pointer for the current time information storage. tm_t * p_alarm: Pointer for the alarm time storage.
Return Value	Success value or error value RTC_SUCCESS: RTC is initialized successfully RTC_ERR_UNINIT: RTC is not initialized RTC_ERR_READ: Data exceeds readable size
Remarks	In the sample code, this function is used when reading the RTC time information. The RTC counter status, time information, and alarm time are all read, so the pointer for each storage area is passed. For details on this function, refer to the RX Family RTC Module Using Firmware Integration Technology application note, Rev.1.00 (R01AN1817ES).

R_MPC_Write	
Outline	External interrupt pin settings
Header	r_mpc_rx_if.h
Declaration	rtc_err_t R_MPC_Write (gpio_port_pin_t pin, mpc_config_t * pconfig)
Description	Functions for external interrupts used are selected, and settings are made to configure pins as IRQ input pins or configure pins as analog pins.
Arguments	gpio_port_pin_t pin: Pin used mpc_config_t * pconfig: Pointer of the storage for information on the pin used
Return Value	Success value or error value MPC_SUCCESS: Pin settings for external interrupts were completed successfully MPC_ERR_INVALID_CFG: Input value is invalid
Remarks	In the sample code, SW1 is used. The P30 pin for SW1 is set in the first argument. As SW1 is used as an IRQ input pin, the second arguments are specified as follows: the function to 0 (high-impedance is set), the IRQ input interrupt to true (used), and usage as an analog pin to false (not used). For details on this function, refer to the RX Family Multi-Function Pin Controller Module Using Firmware Integration Technology application note, Rev.1.00 (R01AN1724EU).

R_IRQ_Open	
Outline	IRQ initialization
Header	r_irq_rx_if.h
Declaration	irq_err_t R_RTC_Open (uint8_t irq_number irq_trigger_t trigger, irq_prio_t priority irq_handle_t *phandle void (*const pcallback)(void *pargs))
Description	For the IRQ initialization, settings such as the interrupt trigger level and priority level are configured.
Arguments	uint8_t irq_number: Initialized IRQ irq_trigger_t trigger: Used trigger level irq_prio_t priority: Interrupt priority level irq_handle_t *phandle: Pointer for the IRQ setting structure void (*const pcallback)(void *pargs)): Pointer of the callback function for exiting an interrupt
Return Value	Success value or error value IRQ_SUCCESS: IRQ initialization successful IRQ_ERR_BAD_NUM: IRQ number was incorrect IRQ_ERR_NOT_CLOSED: R_IRQ_Close() is executed first IRQ_ERR_NULL_PTR: Pointer set was NULL IRQ_ERR_INVALID_ARG: Parameter error IRQ_ERR_LOCK: Selected IRQ number cannot be locked
Remarks	In the sample code, 0 is set to the first argument, low level detection is set to the second argument, and interrupt level 15 is set to the third argument. For details on this function, refer to the RX Family IRQ (External Pin Interrupt Request) Module Using Firmware Integration Technology application note, Rev.1.00 (R01AN1668EU).

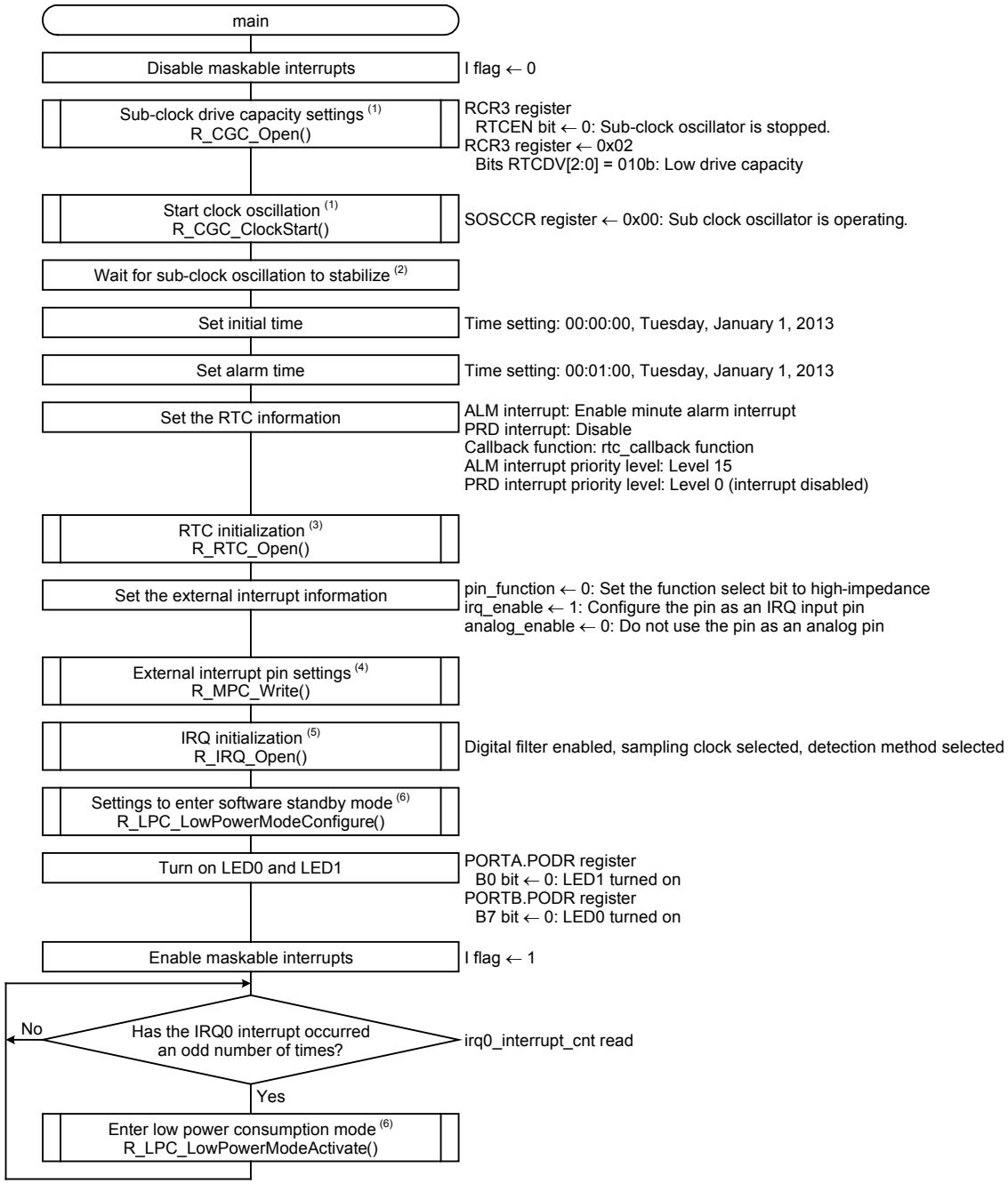
R_LPC_LowPowerModeConfigure	
Outline	Settings to enter low power consumption mode
Header	r_lpc_rx111_if.h
Declaration	lpc_err_t R_LPC_LowPowerModeConfigure (lpc_low_power_mode_t e_mode)
Description	Settings are configured for the MCU to enter low power consumption mode.
Arguments	r lpc_low_power_mode_t e_mode: Low power consumption mode for transition
Return Value	Success value or error value LPC_SUCCESS: Configuration successful for entering low power consumption mode LPC_ERR_ILL_PARAM: Parameter error
Remarks	In the sample code, LPC_LP_SW_STANDBY (software standby mode) is set to the first argument. For details on this function, refer to the RX Family Low Power Consumption Module Using Firmware Integration Technology application note, Rev.1.00 (R01AN1728EU).

R_LPC_LowPowerModeActivate	
Outline	Enter low power consumption mode
Header	r_lpc_rx111_if.h
Declaration	lpc_err_t R_LPC_LowPowerModeActivate((*pcallback)(void *pdata))
Description	The WAIT instruction is executed, and the MCU enters low power consumption mode.
Arguments	(*pcallback)(void *pdata): Pointer of the callback function for processing before the MCU enters low power consumption mode.
Return Value	Success value or error value LPC_SUCCESS: MCU successfully entered low power consumption mode LPC_ERR_OSC_STOP_ENABLED: When entering software standby mode, oscillation stop detection is enabled LPC_ERR_ILL_CLOCK_SOURCE: ▪ When entering sleep mode, the sub-clock is selected as the clock after exiting sleep mode. ▪ When the HOCO clock is selected as the clock after exiting sleep mode, the MCU is in middle speed mode. ▪ When the main clock with middle speed mode is selected as the clock after exiting sleep mode, the ICLK becomes 8 or 12 MHz.
Remarks	In the sample code, the callback function is set to the first argument. For details on this function, refer to the RX Family Low Power Consumption Module Using Firmware Integration Technology application note, Rev.1.00 (R01AN1728EU).

5.7 Flowcharts

5.7.1 Main Processing

Figure 5.6 shows the Main Processing.



Note 1: For more information, refer to the RX Family Clock Generation Circuit Module Using Firmware Integration Technology application note (R01AN1727EU).
Note 2: Set the value applied the optimization in the current compiler version (RXC2.00). When updating the compiler version, change this value as needed.
Note 3: For more information, refer to the RX Family RTC Module Using Firmware Integration Technology application note (R01AN1817ES).
Note 4: For more information, refer to the RX Family Multi-Function Pin Controller Module Using Firmware Integration Technology application note (R01AN1724EU).
Note 5: For more information, refer to the RX Family IRQ (External Pin Interrupt Request) Module Using Firmware Integration Technology application note (R01AN1668EU).
Note 6: For more information, refer to the RX Family Low Power Consumption Module Using Firmware Integration Technology application note (R01AN1728EU).

Figure 5.6 Main Processing

5.7.2 Alarm (ALM) Interrupt Handling

Figure 5.7 shows Alarm (ALM) Interrupt Handling.

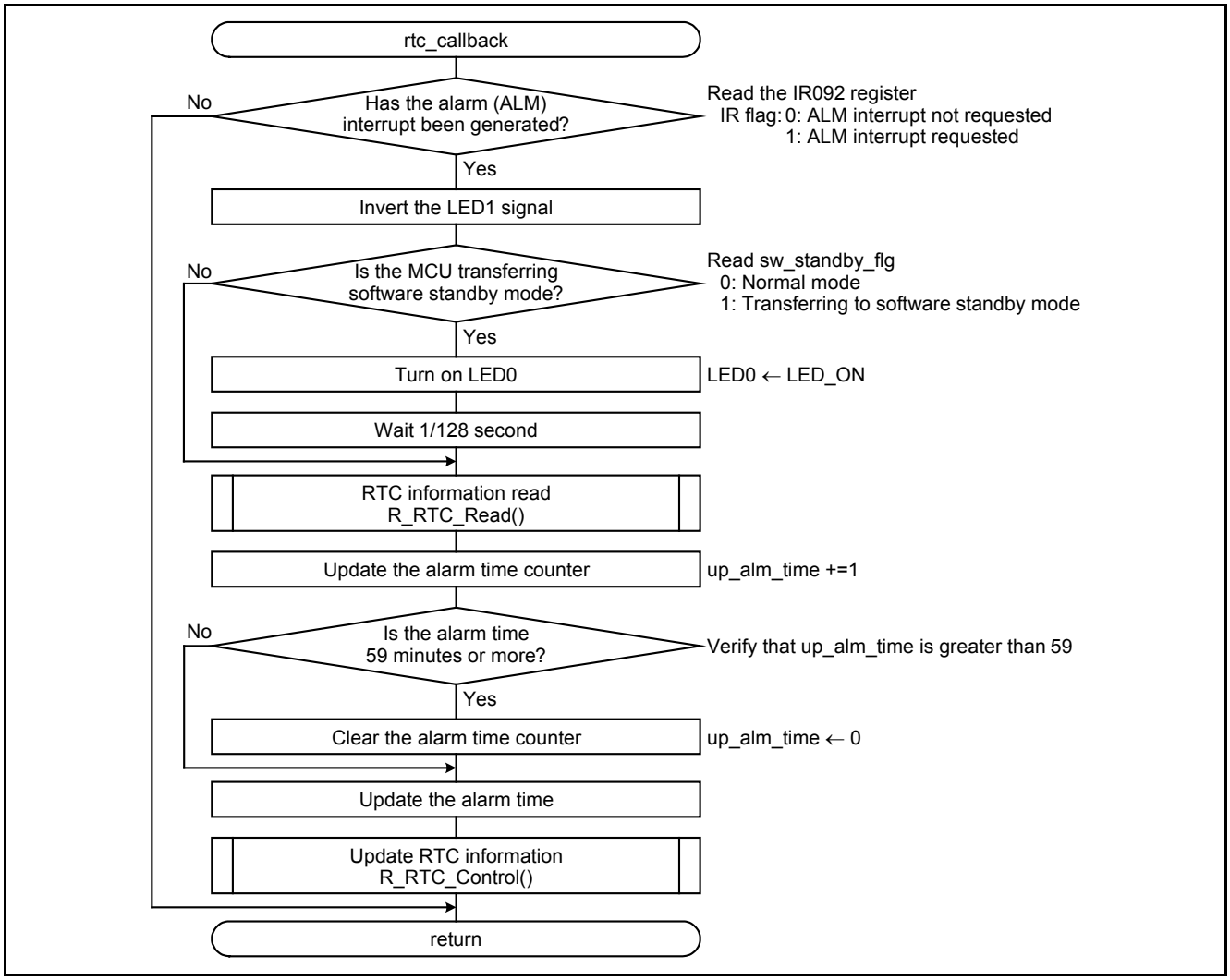


Figure 5.7 Alarm (ALM) Interrupt Handling

5.7.3 Processing Before Entering Software Standby Mode

Figure 5.8 shows Processing Before Entering Software Standby Mode.

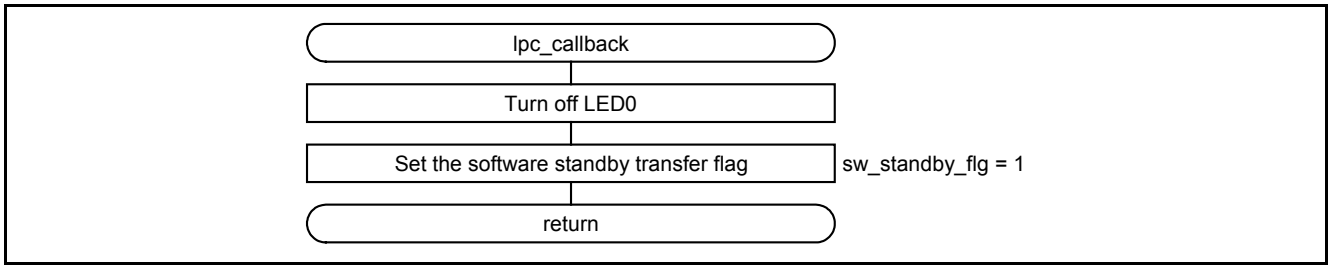


Figure 5.8 Processing Before Entering Software Standby Mode

5.7.4 IRQ0 Interrupt Handling (Processing After a Switch is Pressed)

Figure 5.9 shows IRQ0 Interrupt Handling.

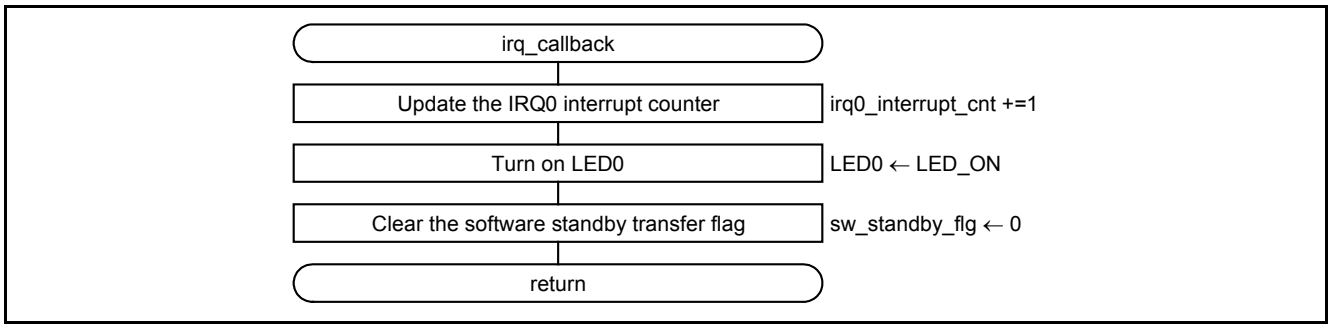


Figure 5.9 IRQ0 Interrupt Handling

6. Appendix

6.1 Compile Option Settings

Figure 6.1 shows Path Settings for the Compile Options. Configure the path settings as shown in the figure. For information on section allocation, refer to chapter 6 in the RX Family Board Support Package Module Using firmware Integration Technology application note Rev.2.30 (R01AN1685EU).

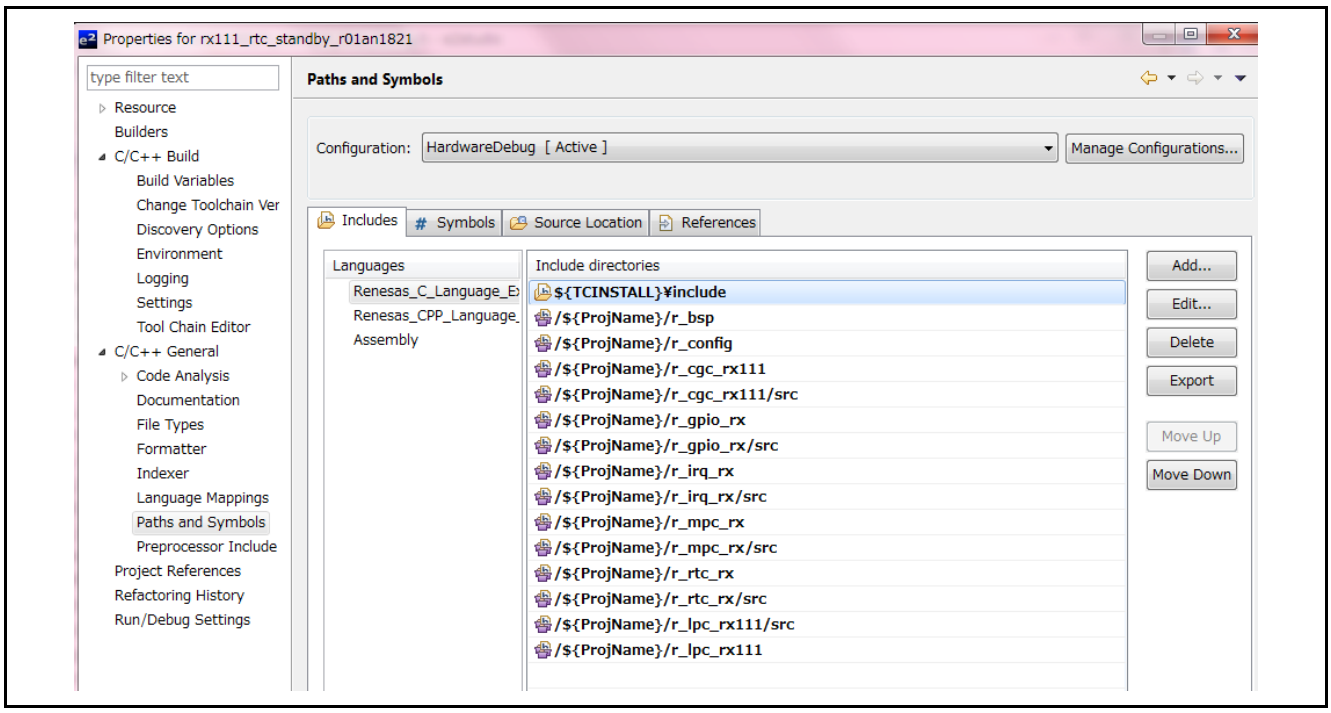


Figure 6.1 Path Settings for the Compile Options

7. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

8. Reference Documents

User's Manual: Hardware
RX111 Group User's Manual: Hardware Rev.1.00 (R01UH0365EJ)
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.2.00 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	RX111 Group Application Note Using the RTCA to Exit Software Standby Mode Firmware Integration Technology

Rev.	Date	Description	
		Page	Summary
1.00	Dec. 16, 2013	—	First edition issued

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 depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; 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 implantations 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 malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or 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 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.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

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

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

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

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

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

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 LanGao Rd., Putuo District, Shanghai, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

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

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

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 3.0