RZ/A2M Group

Use Example for Low Power Mode

Introduction

This application note describes an explanation of use examples for the RZ/A2M low power modes (sleep mode, software standby mode, and deep standby mode).

Target device

RZ/A2M

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

RZ/A2M supports power-down modes (sleep mode, software standby mode and deep standby mode, and module standby). The sample code provided here uses sleep mode, software standby mode and deep standby mode and performs the processes required for transition to and canceling low power mode.

In the sample code, after the start-up message is output, a command input from the terminal on the host PC causes a transition from the program execution state to the power-down mode, and processing of the program stops. In the sample code, the PJ_1 (IRQ0) pin falling edge and the NMI pin falling edge are set as the cancel factor of power-down mode. Accordingly, the power-down modes are canceled after accepting the cancel request from the PJ_1 (IRQ0) pin and NMI pin.

In this application note, the SPI multi-I/O bus controller is referred to as the SPIBSC, the Clock pulse generator as the CPG, the Interrupt controller as the INTC, the OS timer as the OSTM, the Serial communication interface with FIFO as the SCIFA, the General I/O ports as the GPIO, the low power modes as the STB, the Memory management unit as the MMU, and the watch dog timer as WDT.

Table 1.1 summarizes Peripheral Functions and Their Applications, and Figure 1.1 shows Operating Environment for the sample code.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
SPI multi-I/O bus controller (SPIBSC)	When set to external address space read mode, it generates signals that enable the CPU to directly read from serial flash memory connected to the SPI multi-I/O bus space.
Clock pulse generator (CPG)	Generate the operating clocks of the RZ/A2M.
Interrupt controller (INTC)	Used to control OSTM channel 0, OSTM channel 2 and SCIFA channel 4 interrupts. Used for control of the NMI pin and IRQ0 pin interrupts when canceling sleep mode and software standby mode
OS timer (OSTM)	Uses OSTM channel 0 and channel 2 • OSTM channel 0 Generate the intervals at which the LED are turned on and off. • OSTM channel 2 Used for time management via OS Abstraction Layer
FIFO on-board serial communication interface (SCIFA)	Control the communication between RZ/A2M and host PC using SCIFA channel 4
General purpose I/O ports (GPIO)	Used to switch multiplexed pin functions for SCIFA channel 4, and used to control pin for LED on/off. PJ_1 (IRQ0) pin is used as the low power mode cancellation pin
Power-down modes (STB)	Used to cancel the RZ/A2M peripheral I/O module standby state, and enable writing to the on-chip data retention RAM. Used for transition to and cancellation of sleep mode, software standby mode, and deep standby mode Used to set the oscillation stabilization time during transition to deep standby mode
Memory management unit (MMU), L1 cache, L2 cache	Generates translation tables such as specifying valid area of L1 cache or specifying memory type in the RZ/A2M external address area. Set to enable L1 cache and L2 cache
Watchdog timer (WDT)	Used to set the oscillation stabilization time during transition to software standby mode

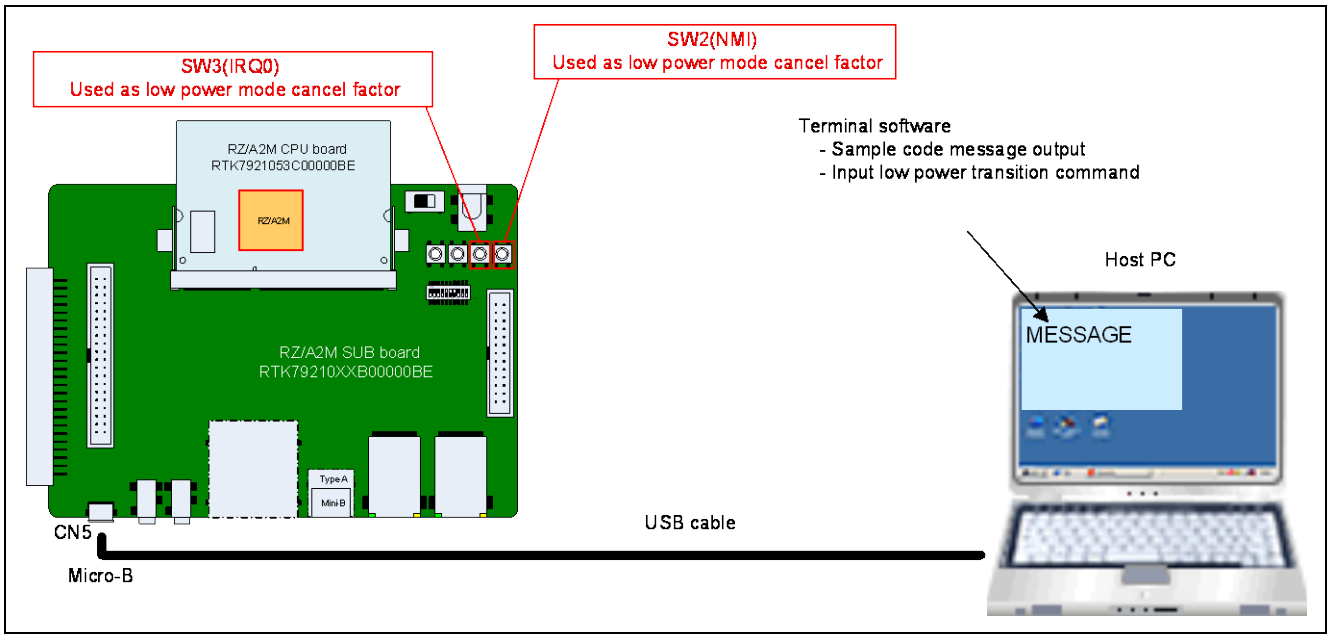


Figure 1.1 Operating Environment

2. Operation Confirmation Conditions

The operation of the sample code accompanying this application note has been confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions (1/2)

Item	Description
MCU used	RZ/A2M
Operating frequency (Note)	CPU clock (Iϕ): 528MHz Image processing clock (Gϕ): 264MHz Internal bus clock (Bϕ): 132MHz Peripheral clock 1 (P1ϕ): 66MHz Peripheral clock 0 (P0ϕ): 33MHz QSPI0_SPCLK: 66MHz CKIO: 132MHz
Operating voltage	Power supply voltage (I/O): 3.3V Power supply voltage (1.8/3.3V switching I/O (PVcc_SPI)): 3.3V Power supply voltage (internal): 1.2V
Integrated development environment	e2 studio V7.5.0
C compiler	GNU Arm Embedded Toolchain 6-2017-q2-update
	Compiler options (addition of directory path excluded) Release configuration: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Os -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -Wstack-usage=100 -fabi-version=0 Hardware Debug configuration: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Og -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -g3 -Wstack-usage=100 -fabi-version=0

Note: The operating frequency used in clock mode 1 (Clock input of 24MHz from EXTAL pin)

Table 2.2 Operation Confirmation Conditions (2/2)

Item	Contents
Operating mode	Boot mode 3 (Serial flash boot 3.3V)
Communications settings of the terminal software	• Communication speed: 115200bps • Data length: 8 bit • Parity: None • Stop bit length: 1 bit • Flow control: None
Boards used	RZ/A2M CPU board RTK7921053C00000BE RZ/A2M SUB board RTK79210XXB00000BE
Devices used (functions used on the board)	• Serial flash memory allocated to SPI multi-I/O bus space Manufacturer name: Macronix, Model name: MX25L51245GXD • RL78/G1C (Converts between USB communication and serial communication to communicate with the host PC.) • LED1 • SUB board SW2 and SW3 SW2: Signal input to NMI pin SW3: Signal input to PJ_1 (IRQ0) pin

Note: Because the debugger interface clock is also stopped in software standby mode and deep standby mode, this sample code may not operate with an emulator and the board connected, depending on the emulator used. If this kind of emulator is being used, download the program to the serial flash memory on the board, disconnect the emulator and the board, then run the sample code on just the board.

3. Reference Application Notes

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

- RZ/A2M Group: Example of Initialization (R01AN4321EJ)
- RZ/A2M Group: Example of Booting from Serial Flash Memory (R01AN4333EJ)

4. Hardware

4.1 Summary of Power-down Modes

RZ/A2M supports the following power-down modes. In power-down mode the amount of power consumed is decreased by stopping CPU operation, the clock, and the operation of on-chip memory and some on-chip peripheral modules, and turning the LSI internal power off.

- Sleep mode
- Software standby mode
- Deep standby mode
- Module standby mode

This application note provides an explanation mostly for sleep mode, software standby mode, and deep standby mode. The RZ/A2M processing status is composed of 3 states: reset state, program execution state, and power-down mode state. Figure 4.1 shows the Processing State Transition Diagram.

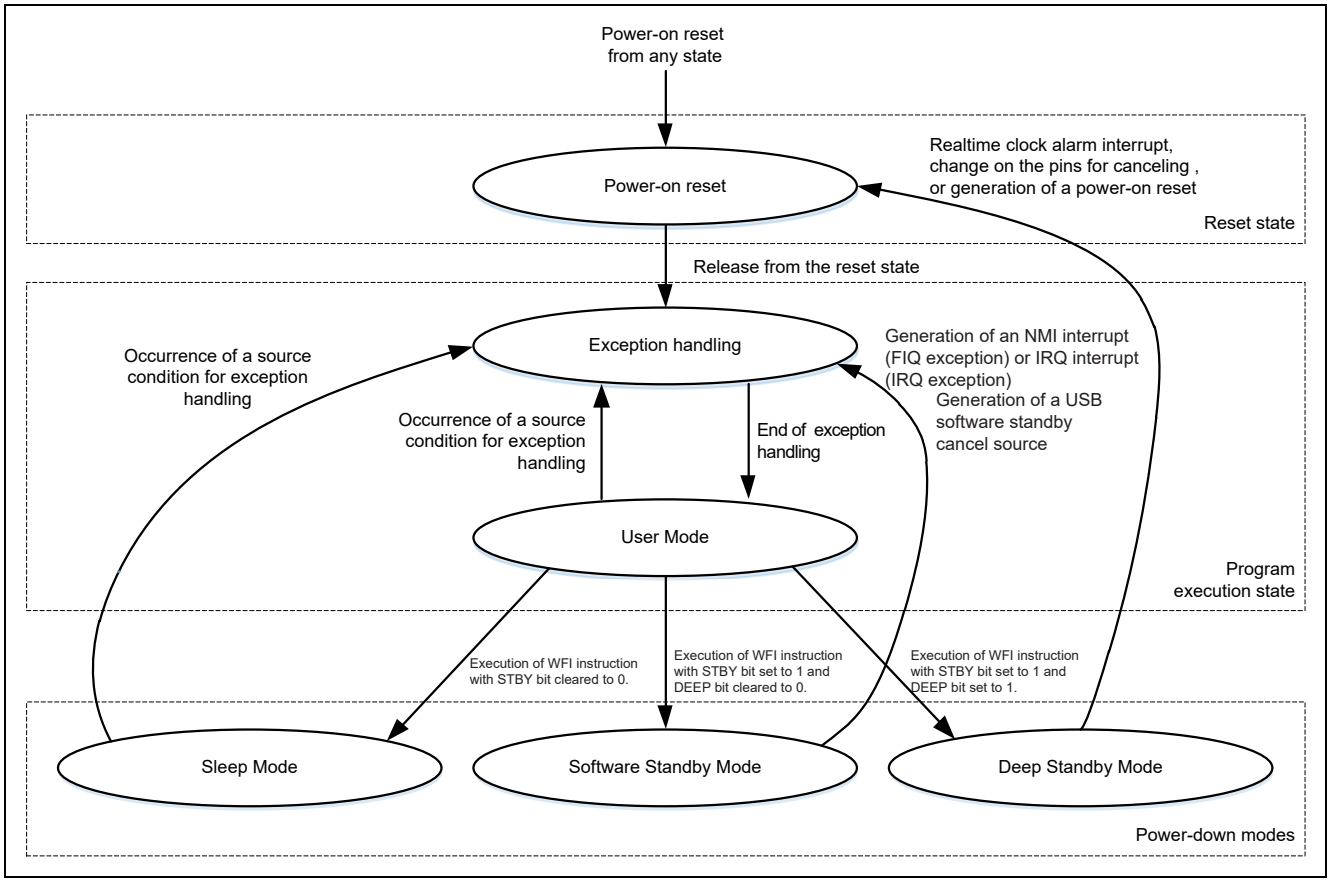


Figure 4.1 Processing State Transition Diagram

Table 4.1 shows the Summary of Power-down Modes.

Table 4.1 Summary of Power-down Modes

Power-down mode	Transition condition	State					Cancellation method
		CPU, CPU register, Primary cache TLB	Secondary cache	Large-capacity on-chip RAM (including on-chip data retention RAM)	On-chip peripheral modules	LSI internal power supply	
Sleep mode	In STBCR1=0x00 (or 0x40) state, execute WFI instruction	Halted Contents are held.	Running	Running	Running	Applied	• Interrupt • Power-on reset
Software standby mode	In STBCR1=0x80 state, execute WFI instruction	Halted Contents are held.	Halted Contents are held.	Halted Contents are held.	Halted	Applied	• NMI and IRQ interrupts • Power-on reset • Change on the USB Software Standby Cancel factor signal
Deep standby mode	In STBCR1=0xC0 state, execute WFI instruction	Halted Contents are not held.	Halted Contents are not held.	Halted Contents in on-chip data retention RAM are held, and the other contents in large-capacity on-chip RAM are not held	Halted	Shut off	• Power-on reset • Realtime clock alarm interrupt 0 and 1 • Change on the pins for canceling • USB2.0 host/function module channel interrupt 0 and 1
Module standby	MSTP bits of STBCR10 to STBCR2 are set to 1.	Running	Running	Running	Specified module halted	Applied	• MSTP bit cleared to 0

4.1.1 Sleep Mode

When transition to sleep mode, power consumption is reduced by stopping CPU operations. During transition to sleep mode, the contents of the CPU register, the large-capacity on-chip RAM, and caches are retained, so after sleep mode is canceled, it is possible to continue processing from instruction immediately after sleep mode transition. Furthermore, the CKIO output states in the program execution state and during sleep mode transition will be different in accordance with the FRQCR register's CKOEN[1:0] bit setting value.

Table 4.2 CKIO Output State in Normal State and Sleep Mode

CKOEN[1:0] bit setting value of FRQCR register	Output state during program execution state	Output state during sleep mode
b'00	Output	Output
b'01	Output	Output
b'10	Output	Output
b'11	Output off (Hi-z)	Output off (Hi-z)

(1) Transition to Sleep Mode

For RZ/A2M, if the WFI instruction is executed when the STBCR1 register STBY bit is 0, it will make transition from the program execute state to sleep mode.

(2) Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI pin, IRQ pin, on-chip peripheral function) or a power on reset. Processing after sleep mode cancellations differs according to the cancellation method. Table 4.3 shows the Sleep Mode Cancellation Method and Processing after Cancellation.

Table 4.3 Sleep Mode Cancellation Method and Processing after Cancellation

Cancellation Method	Processing after Cancellation
Interrupt (IRQ pin, peripheral module)	After an IRQ exception occurs, after the IRQ exception processing and interrupt processing in accordance with the cancel factor are performed, processing from the next instruction of the WFI instruction when making transition to sleep mode is continued.
NMI pin	After an FIQ exception occurs, after the FIQ exception processing and NMI interrupt processing are performed, processing from the next instruction of the WFI instruction when making transition to sleep mode is continued.
Power on reset	After the reset cancellation, the program is executed from the reset vector (Address H'FFFF_0000).

4.1.2 Software Standby Mode

When making transition to software standby mode, because the CPU operation, clock, and peripheral modules are stopped, it is possible to reduce the consumed power even lower than sleep mode. During software standby mode, the contents of the CPU and peripheral module registers, the large-capacity on-chip RAM and caches are retained, so after the software standby mode is canceled, it is possible to resume the software processing before the transition. Furthermore, the CKIO output states in the program execution state and during software standby mode transition will be different in accordance with the FRQCR register's CKOEN[1:0] bit setting value.

Table 4.4 CKIO output state for Software Standby Mode

CKOEN[1:0] bit setting value of FRQCR register	Output state during program execution state	Output state during software standby mode
b'00	Output	Output off (Hi-z)
b'01	Output	Low level output
b'10	Output	Output (Unstable clock)
b'11	Output off (Hi-z)	Output off (Hi-z)

(1) Transition to Software Standby Mode

For RZ/A2M, if the WFI instruction is executed when the STBCR1 register STBY bit is 1 and DEEP bit is 0, it will make transition from the program execute state to software standby mode.

(2) Canceling Software Standby Mode

Software standby mode is canceled by an interrupt (NMI pin, IRQ pin), USB software standby cancel factor or a reset (power on reset). After the cancel factor is detected, after the software standby recovery oscillation stabilization time (NOTE) has passed, the standby mode is canceled. Processing after software standby mode cancellations differs according to the cancellation method. Table 4.5 shows the Software Standby Mode Cancellation Method and Processing after Cancellation. After software standby cancellation, the clock is output from the CKIO pin (when CKOEN[1:0] bit is set to anything other than b'11).

Note: Software standby recovery oscillation stabilization time
Set via the WDT's WTCSR register CKS [3:0] bit (clock source selection) and the WTCNT register (counter setting value). The time until the WTCNT register counter overflow occurs must be set to anything greater than the clock oscillation stabilization time in customer's system.

Table 4.5 Software Standby Mode Cancellation Method and Processing after Cancellation

Cancellation Method	Processing after Cancellation
Interrupt (IRQ pin, USB)	After an IRQ exception occurs, after the IRQ exception processing and interrupt processing (IRQ interrupt processing or USB interrupt processing) in accordance with the cancel factor are performed, processing from the next instruction of the WFI instruction when making transition to software standby mode is resumed.
NMI pin	After an FIQ exception occurs, after the FIQ exception processing and NMI interrupt processing are performed, processing from the next instruction of the WFI instruction when making transition to software standby mode is resumed.
Power on reset	After the reset cancellation, the program is executed from the reset vector (Address H'FFFF_0000).

4.1.3 Deep Standby Mode

When making transition to deep standby mode, because the CPU operation, clock and peripheral modules are stopped, and the LSI internal power is shut off, it is possible to greatly reduce the consumed power even lower than sleep mode and software standby mode. However, during deep standby mode, the contents of the CPU and peripheral module registers, the large-capacity on-chip RAM and caches are not retained, and processing starts from the reset state after the cancellation of the deep standby mode, so it is necessary to re-initialize the peripheral modules and the contents of the memory used by any program. Furthermore, the CKIO output states in the program execution state and during deep standby mode transition will be different in accordance with the FRQCR register's CKOEN[1:0] bit setting value.

Table 4.6 CKIO output state for Deep Standby Mode

FRQCR register CKOEN[1:0] bit setting value	Output state during program execution state	Output state during deep standby mode
b'00	Output	Output off (Hi-z)
b'01	Output	Low level output
b'10	Output	Low level output or high level output
b'11	Output off (Hi-z)	Output off (Hi-z)

(1) Transition to Deep Standby Mode

For RZ/A2M, if the WFI instruction is executed when the STBCR1 register STBY bit is 1 and DEEP bit is 1, it will make transition from the program execute state to deep standby mode.

(2) Canceling Deep Standby Mode

Deep standby mode is canceled by interrupt (NMI pin, real time clock 0,1 alarm interrupt, or USB2.0 host/function module channel 0,1 cancel factor), change in the cancel pin or reset (power on reset). After the cancel factor is detected, after the deep standby recovery oscillation stabilization time (NOTE) has passed, the standby mode is canceled. For processing after deep standby mode cancellation, regardless of cancellation method, program execution starts from the reset vector (H'FFFF_0000) after the cancellation.

If the deep standby mode was canceled via external pin, f It is not necessary set to function assignment to the pin used as the cancel factor by the general purpose input/output port (GPIO).

Note: Deep standby recovery oscillation stabilization time
Set via DSCNT register. The time until the DSCNT register counter overflow occurs must be set to anything greater than the clock oscillation stabilization time in customer's system.

Operation after deep standby mode cancellation differs according to the setting value of the EBUSKEEPE bit and RAMBOOT bit. Table 4.7 shows the Startup Method and Pin State after Deep Standby Mode Cancellation.

Table 4.7 Startup Method and Pin State after Deep Standby Mode Cancellation

EBUSKEEPE bit	RAMBOOT bit	Startup method	Pin state after deep standby mode cancellation
0	0	Startup from external memory (Startup from the address where the loader program is located according to the boot mode at power on.)	The states of the external memory control pins are not retained. After cancellation of deep standby mode, the retention of the external memory control pin states is cancelled. For other pins, the retention of their states is cancelled when the IOKEEP bit is cleared.
0	1	Start from on-chip data retention RAM Address H'8000_0000	The states of the external memory control pins are not retained. After cancellation of deep standby mode, the retention of the external memory control pin states is cancelled. For other pins, the retention of their states is cancelled when the IOKEEP bit is cleared.
1	0	-	Setting prohibited
1	1	Start from on-chip data retention RAM Address H'8000_0000	The states of the external memory control pin are retained. The retention of the states of the external memory control pins and other pins is canceled when the IOKEEP bit is cleared.

(3) Determine Deep Standby Mode Cancel Factor

If the deep standby mode is canceled by any factor other than the power on reset, it is possible to determined which cancel factor caused the cancellation, based on the DSFR register and the USBDSFR register.

Figure 4.2 shows the Deep Standby Mode Cancellation Flow.

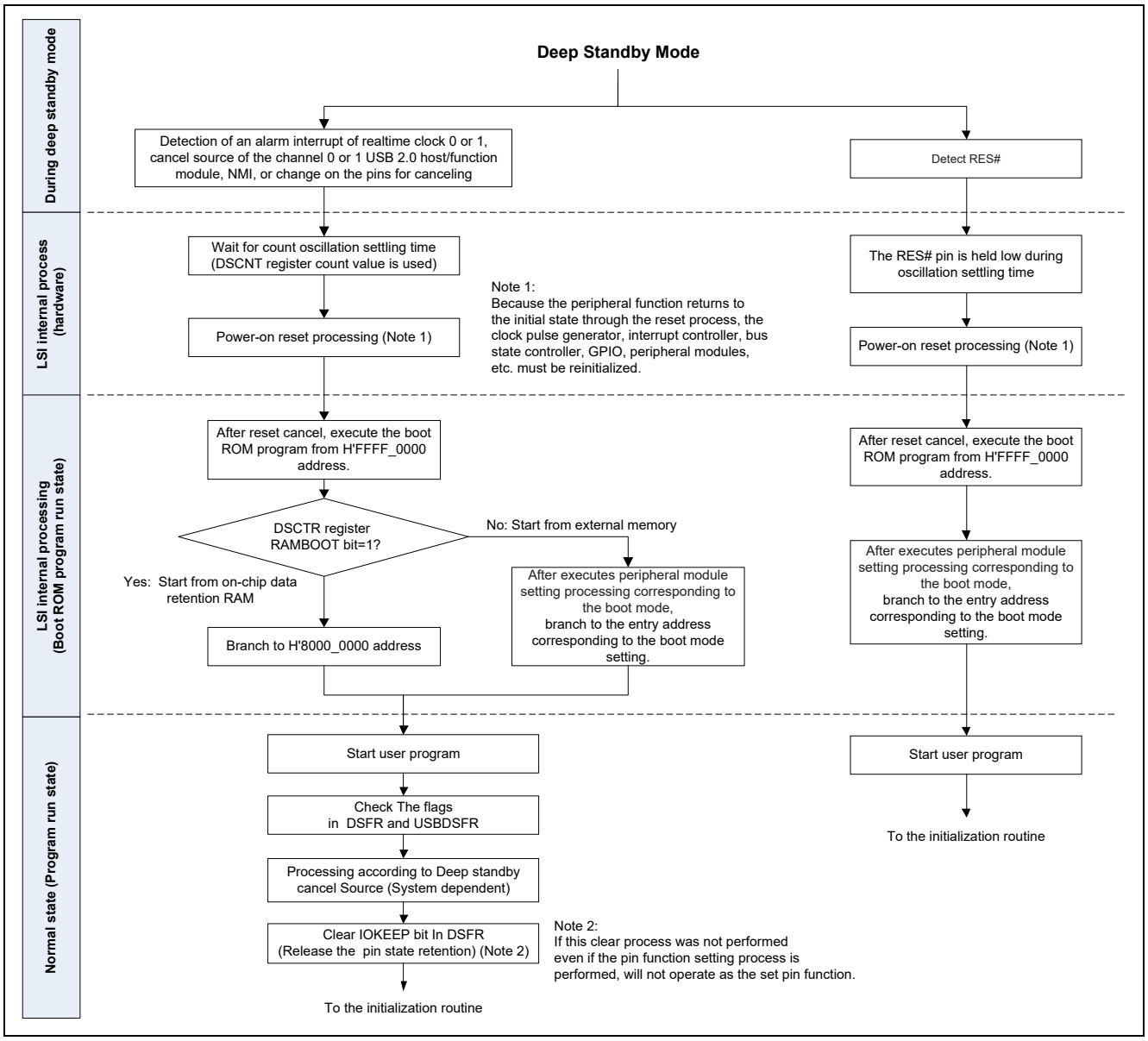


Figure 4.2 Deep Standby Mode Cancellation Flow

4.2 Notes Regarding Transition to Power-down Mode

In the software standby mode and the deep standby mode, after a WFI instruction is issued, the hardware automatically stops the bus master following the wait for completion of the issuing-finished request from the bus master and the transition to software standby proceeds. Since the transition to software standby is not possible if completion of the issuing-finished request is not possible at this time, do not proceed with access to the registers of modules in the module-standby state and so on. To suppress an unintended request from a bus master, use the software to stop bus masters before transition to the software standby mode.

4.3 Pins Used

Table 4.8 shows the Pins Used and Their Functions.

Table 4.8 Pins Used and Their Functions

Pin Name	I/O	Function
MD_BOOT2	Input	Select boot mode (set to boot mode 3) MD_BOOT2: "L", MD_BOOT1: "H", MD_BOOT0: "H"
MD_BOOT1	Input	
MD_BOOT0	Input	
QSPI0_SPCLK	Output	Serial flash memory clock
QSPI0_SSL	Output	Serial flash memory slave select
QSPI0_IO0	I/O	Serial flash memory data 0
QSPI0_IO1	I/O	Serial flash memory data 1
QSPI0_IO2	I/O	Serial flash memory data 2
QSPI0_IO3	I/O	Serial flash memory data 3
RPC_RESET#	Output	Serial flash memory reset
P6_0	Output	Turns on and off LED
RxD4 (P9_1)	Input	Serial receive data signal
TxD4 (P9_0)	Output	Serial transmit data signal
NMI	Input	NMI signal
IRQ0 (PJ_1)	Input	IRQ0 (PJ_1) signal

Note: # is the symbol that indicates negative logic (or active low).

5. Software

5.1 Operation Overview

The sample code provided here uses sleep mode, software standby mode, and deep standby mode and performs the processes required for transition to and canceling power-down mode. The process details for each of the power-down mode is explained below.

In the sample code, IRQ0 pin interrupt is used as the standby cancel factor for sleep mode and software standby mode, but processes which set PJ_1 pin to the IRQ0 pin function are performed using the main function's GPIO driver "Open function" process (The NMI pin is a specialized pin, so the pin function cannot be set). Implement the processes necessary for the cancellation, depending on the cancel factor used.

5.1.1 Sleep Mode

In the sample code, after the startup message is output, when the "LPM_SM" command is input from the terminal, the Sample_LPM_Sleep_Mode function used to make transition to sleep mode is executed, and the RZ/A2M makes transition to sleep mode.

While in sleep mode, if SW2 (NMI) or SW3 (IRQ0) on the SUB board is pressed, sleep mode is canceled, and the processes before transition to the sleep mode are continued.

Messages are output to the terminal before transition to sleep mode and after cancellation. For the messages output by sleep mode, refer to "5.10.1 Sleep Mode Transition Command".

(1) Processes before Transition to Sleep Mode

Before making transition to sleep mode, set the interrupt used as the cancel factor.

In the sample code, setting to enable the use of the IRQ0 pin interrupt and the NMI pin interrupt is performed. Setting to allow receipt of interrupt requests is performed, and the interrupt handler function that is execute during receipt of an IRQ0 pin interrupt or NMI pin interrupt is registered. The Sample_LPM_Sleep_Mode function uses the R_INTC_RegistIntFunc function, and registers the interrupt handler function shown in Table 5.1.

Table 5.1 Functions Used to Cancel Sleep Mode

Interrupt	Interrupt ID	Registered function
NMI	-	Sample_NMI_Interrupt function
IRQ0	36	Sample_IRQ0_Interrupt function

In the sample code, the interrupt mask level is controlled in order to prevent sleep mode cancellation by anything other than the IRQ0 pin interrupt or the NMI pin interrupt. By setting the IRQ0 pin interrupt priority level to "1", and changing the interrupt mask level from "31" to "2" using the R_INTC_SetMaskLevel function before transition to sleep mode, any interrupts at priority level 2 or lower (the OSTM and SCIFA interrupts in the sample code) are not generated.

The interrupt request detection method (edge sense or level sense) for the IRQ0 pin interrupt and NMI pin interrupt used as the cancel factor is set.

Table 5.2 Sleep Mode Cancel Factor Setting Contents

Cancel factor	Resources on board	Setting value
NMI	SUB board SW2	Interrupt request detected by NMI input falling edge
IRQ0	SUB board SW3	Interrupt request detected by IRQ0 input falling edge

(2) Processes during Transition to Sleep Mode

Transition to sleep mode is performed by a call to the R_LPM_SleepTransition function. In the R_LPM_SleepTransition function, after the STBCR1 register STBY bit is set to 0 and the DEEP bit is set to 0, the WFI instruction is executed.

(3) Processes after Sleep Mode Cancellation

Operation after sleep mode cancellation differs according to the cancel factor. Table 5.3 shows theProcessing after Sleep Mode Cancellation.

After sleep mode cancellation, the interrupt mask level is set to "31" that is the same as before transition to the sleep mode, so that IRQ exceptions with a priority level higher than 31 can be received.

Table 5.3 Processing after Sleep Mode Cancellation

Cancel factor	Executed exception	Operation after sleep mode cancellation
NMI	FIQ exception	After the Sample_NMI_Interrupt function called from the FIQ exception process is executed, return to normal routine, and continue processing from the next instruction after the WFI instruction.
IRQ0	IRQ exception	After the Sample_IRQ0_Interrupt function called from the IRQ exception process is executed, return to normal routine, and continue processing from the next instruction after the WFI instruction.
Reset	Reset exception	The program is executed from the reset vector (Address H'FFFF_0000). Processing executed same as power-on.

5.1.2 Software Standby Mode

In the sample code, after the startup message is output, when the "LPM_SS" command is input from the terminal, the Sample_LPM_Software_Standby function used to make transition to sleep mode is executed, and the RZ/A2M makes transition to software standby mode.

While in software standby mode, if SW2 (NMI) or SW3 (IRQ0) on the SUB board is pressed, software standby mode is canceled, and the processes before transition to the software standby mode are resumed.

The FRQCR register's CKOEN[1:0] bit is set to b'00, to turn the output state of the CKIO pin during software standby mode transition to off (Hi-z).

Messages are output to the terminal before transition to software standby mode and after cancellation. For the messages output by software standby mode, refer to "5.10.2 Software Standby Mode Transition Command Operation".

(1) Processes before Transition to Software Standby Mode.

Before transition to software standby mode, set the interrupt used as the cancel factor.

In the sample code, setting to enable the use of the IRQ0 pin interrupt and the NMI pin interrupt is performed. Setting to allow receipt of interrupt requests is performed, and the interrupt handler function that is execute during receipt of an IRQ0 pin interrupt or NMI pin interrupt is registered. The Sample_LPM_Software_Standby function uses the R_INTC_RegistIntFunc function, and registers the interrupt handler function shown in Table 5.4.

Table 5.4 Functions Used to Cancel Software Standby Mode

Interrupt	Interrupt ID	Registered function
NMI	-	Sample_NMI_Interrupt function
IRQ0	36	Sample_IRQ0_Interrupt function

In the sample code, in order to prevent the occurrence of interrupt requests other than the IRQ0 pin interrupt or NMI pin interrupt being executed at the same time as WFI instruction, the IRQ0 pin interrupt priority level is set to "1", and by changing the interrupt mask level from "31" to "2", any interrupts at priority level 2 or lower (the OSTM and SCIFA interrupts in the sample code) are not generated.

The interrupt request detection method (edge sense or level sense) for the IRQ0 pin interrupt and NMI pin interrupt used as the cancel factor is set.

Table 5.5 Software Standby Mode Cancel Factor Setting Contents

Cancel factor	Resources on board	Setting value
NMI	SUB board SW2	Interrupt request detected by NMI input falling edge
IRQ0	SUB board SW3	Interrupt request detected by IRQ0 input falling edge

It is possible to specify IRQ7 pin to IRQ0 pin or the USB module as the cancel factor via the contents of the configuration table (LPM_SC_TABLE[ ]) setting. It is necessary to separately prepare the software required for USB module operation.

(2) Processes during Transition to Software Standby Mode.

Transition to software standby mode is performed by a call to the R_LPM_SStandbyTransition function. In the R_LPM_SStandbyTransition function, the data of the configuration table for the low power mode specified in the argument is used, and after the register setting process for software standby mode cancel factor and standby recovery oscillation stabilization time, etc., is performed, STBCR1 register STBY bit is set to 1 and DEEP bit is set to 0, then the WFI instruction is executed.

In this sample code, it is possible to set the pin state during software standby mode transition, cancel factor, software standby mode recovery oscillation stabilization time based on the setting values in the configuration table (LPM_SC_TABLE[]). For details regarding configuration table LPM_SC_TABLE[], refer to Table 5.19 to Table 5.23.

Table 5.6 List of Software Standby Mode Items Settable via Configuration Table

Setting	Settable values	Setting content in sample code
Pin state	Pin state during transition to software standby mode. • LPM_PIN_STATUS_HIZ Do not retain pin state (Hi-z) • LPM_PIN_STATUS_RETAINED Pin state is retained	Do not retain pin state during transition to software standby mode (Hi-z)
Cancel factor	Specifies whether a IRQ pin (IRQ7 to IRQ0) will be used as the software standby mode cancel factor. • LPM_SSTANDBY_DISABLE_IRQ IRQ pin is not used as the cancel factor • LPM_SSTANDBY_ENABLE_IRQ IRQ pin is used as the cancel factor	An IRQ pin (IRQ7 to IRQ0) is used as the software standby mode cancel factor.
	Specifies whether USB0 or USB1 interrupt request will be used as the software standby mode cancel factor. • LPM_SS_CANCEL_FACTOR_DISABLED USB not used as the cancel factor • LPM_SS_CANCEL_FACTOR_ENABLED USB is used as the cancel factor	Do not use USB0 or USB1 as the software standby mode cancel factor
Software standby mode recovery oscillation stabilization time	Specifies ms unit for software standby mode recovery oscillation stabilization time (Up to 2 digits after the decimal point allowed).	7.0ms

(3) Processes after Software Standby Mode Cancellation

Operation after software standby mode cancellation differs according to the cancel factor. Table 5.7 shows the Processes after Software Standby Mode Cancellation.

After software standby mode cancellation, the interrupt mask level is set to "31" that is the same as before transition to the software standby mode, so that IRQ exceptions with a priority level higher than 31 can be received.

Table 5.7 Processes after Software Standby Mode Cancellation

Cancel factor	Executed exception	Operation after software standby mode cancellation
NMI	FIQ exception	After the Sample_NMI_Interrupt function called from the FIQ exception process is executed, return to normal routine, and resume processing from the next instruction after the WFI instruction.
IRQ0	IRQ exception	After the Sample_IRQ0_Interrupt function called from the IRQ exception process is executed, return to normal routine, and resume processing from the next instruction after the WFI instruction.
Reset	Reset exception	The program is executed from the reset vector (Address H'FFFF_0000). Processing executed same as power-on.

Note: In the transition process to software standby mode, in order to set the software standby recovery oscillation time, stop the WDT counter and change the counter value. If WDT is used after the software standby mode cancellation, it is necessary to reinitialize the WDT.

5.1.3 Deep Standby Mode

In the sample code, after the startup message is output, when the "LPM_DS" + "1" or "LPM_DS" + "2" command are input from the terminal, the Sample_LPM_Deep_Standby function used to make transition to deep standby mode is executed, and the RZ/A2M makes transition to deep standby mode.

While in deep standby mode, if SW2 (NMI) or SW3 (IRQ0) on the SUB board is pressed, deep standby mode is canceled, and the processes after the cancellation of deep standby mode are executed from the reset vector (Address H'FFFF_0000).

The FRQCR register's CKOEN[1:0] bit is set to b'00, to turn the output state of the CKIO pin during deep standby mode transition to off (Hi-z).

Messages are output to the terminal before transition to deep standby mode and after cancellation. For the messages output by deep standby mode, refer to "5.10.3 Deep Standby Mode Transition Command Operation".

(1) Processes before Transition to Deep Standby Mode

Before transition to deep standby mode, the cancel factor is set.

In the sample code, the changes for PJ_1 pin and NMI pin are set as the cancel factor.

Table 5.8 Deep Standby Mode Cancel Factor Setting Contents

Cancel factor	Resources on board	Setting value
NMI	SUB board SW2	When NMI pin falling edge detected
PJ_1	SUB board SW3	When PJ_1 pin falling edge detected

It is possible to specify the cancel pin, RTC alarm interrupt or USB module interrupt as the cancel factor via the contents of the configuration table (LPM_SC_TABLE[ ]) setting. It is necessary to separately prepare the software required for RTC and USB module operation.

(2) Processes during Transition to Deep Standby Mode

Transition to deep standby mode is performed by a call to the R_LPM_DStandbyTransition function. In the R_LPM_DStandbyTransition function, the data of the configuration table for the low power mode (LPM_SC_TABLE[]) specified in the argument is used, and after the register setting process for deep standby mode cancel factor and standby recovery oscillation stabilization time, etc., is performed, STBCR1 register STBY bit is set to 1 and DEEP bit is set to 1, then the WFI instruction is executed.

In this sample code, it is possible to set the pin state during deep standby mode transition, cancel factor, deep standby mode recovery oscillation stabilization time based on the setting values in the configuration table (LPM_SC_TABLE[]). For details regarding configuration table LPM_SC_TABLE[], refer to Table 5.19 to Table 5.23.

Table 5.9 List of Deep Standby Mode Items Settable via Configuration Table

Setting	Settable values	Setting content in sample code
Pin state	Pin state during deep standby mode transition • LPM_PIN_STATUS_HIZ Do not retain pin state (Hi-z) • LPM_PIN_STATUS_RETAINED Pin state is retained	Do not retain pin state during transition to deep standby mode (Hi-z)
Cancel factor	Selection of PK_4, PK_2, PJ_5, PJ_1, PH_0, PG_6, NMI, PG_2, PH_1, PE_1, P6_2, P3_3, P3_1 as deep standby mode cancel factor and setting of detection method • LPM_PIN_CANCEL_FACTOR_DISABLED Do not use as cancel factor • LPM_PIN_CANCEL_FACTOR_FALLING_EDGE Cancel by falling edge • LPM_PIN_CANCEL_FACTOR_RISING_EDGE Cancellation by rising edge	PJ_1: Falling edge NMI: Falling edge Sets other pins to not be used as the cancel factor
	Select enable/disable for use of each DP pins and DM pins for USB0, USB1 as the deep standby mode cancel factor • LPM_DS_CANCEL_FACTOR_DISABLED Do not use as cancel factor • LPM_DS_CANCEL_FACTOR_ENABLED Used signal change as cancel factor	USB0 and USB1 not used as cancel factor
	Select enable/disable for use RTC0 and RTC1 alarm interrupts as the deep standby mode cancel factor • LPM_DS_CANCEL_FACTOR_DISABLED Alarm interrupt is not used as the cancel factor • LPM_DS_CANCEL_FACTOR_ENABLED Alarm interrupt is used as the cancel factor	RTC1 and RTC0 not used as cancel factor setting
Retain/do not retain contents of on-chip data retention RAM during transition to standby	For on-chip data retention RAM pages 0 to 3, select whether the on-chip data retention RAM areas are retained during deep standby • LPM_RETENTION_RAM_NOT_RETAINED Not retained • LPM_RETENTION_RAM_RETAINED Retained	On-chip data retention RAM page 0: Retained On-chip data retention RAM page 1: Not retained On-chip data retention RAM page 2: Not retained On-chip data retention RAM page 3: Not retained
Deep standby mode recovery oscillation stabilization time	Specifies ms unit for deep standby mode recovery oscillation stabilization time (Up to 2 digits after the decimal point allowed).	7.0ms
Program startup method after deep standby mode cancellation	Selects the program startup method after deep standby mode cancellation • LPM_REBOOT_TYPE_EXTERNAL_MEMORY_BOOT Start program from external memory • LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_0 Start program from on-chip data retention RAM (Retain external memory pin state) • LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_1 Start program from on-chip data retention RAM (Do not retain external memory pin state)	Program start from external memory or program start from on-chip data retention RAM

(3) Processes after Deep Standby Mode Cancellation

Figure 5.1 shows the Sample Code Operation after Deep Standby Mode Cancellation. After deep standby mode cancellation, regardless of cancel factor, the boot program located at the reset vector (Address H'FFFF_0000) is executed. The operation after cancellation differs according to the settings of the RAMBOOT bit and the EBUSKEEPE bit in the Deep Standby Control Register (DSCTR) set before transition to deep standby mode.

- External memory startup mode
  In the case of making transition to deep standby mode when the RAMBOOT bit is set to "0", after standby cancellation, the boot program is executed, and the RZ/A2M starts up from external memory in accordance with the boot mode. In the sample code, Boot Mode 3 is used, and after the boot program is executed, the loader program branched to Address H'2000_0000 is executed. After the loader program is executed, it branches to the application program located at Address H'2001_0000. If the RAMBOOT is "0", the pin state for non-external memory control pins is held until the IOKEEP bit is cleared, so in the sample code, the loader program performs the IOKEEP bit clear process.
- On-chip data retention RAM startup mode
  In the case of making transition to deep standby mode when the RAMBOOT bit is set to "1", after standby cancellation, the boot program is executed, and it branches to the start address for the on-chip data retention RAM (Address H'8000_0000). In the sample code, the LPM_DStandby_ReturnRamBoot function located in the on-chip data retention RAM's start address is executed. If the RAMBOOT bit is "1", it is necessary to place the program that will operate after deep standby mode cancellation to the on-chip data retention RAM area (An area of 128KB from address H'8000_0000) prior to deep standby mode transition. In the sample code, the program to execute after deep standby mode cancellation is transferred to the on-chip data retention RAM using the INITSCT function's section initialization process.

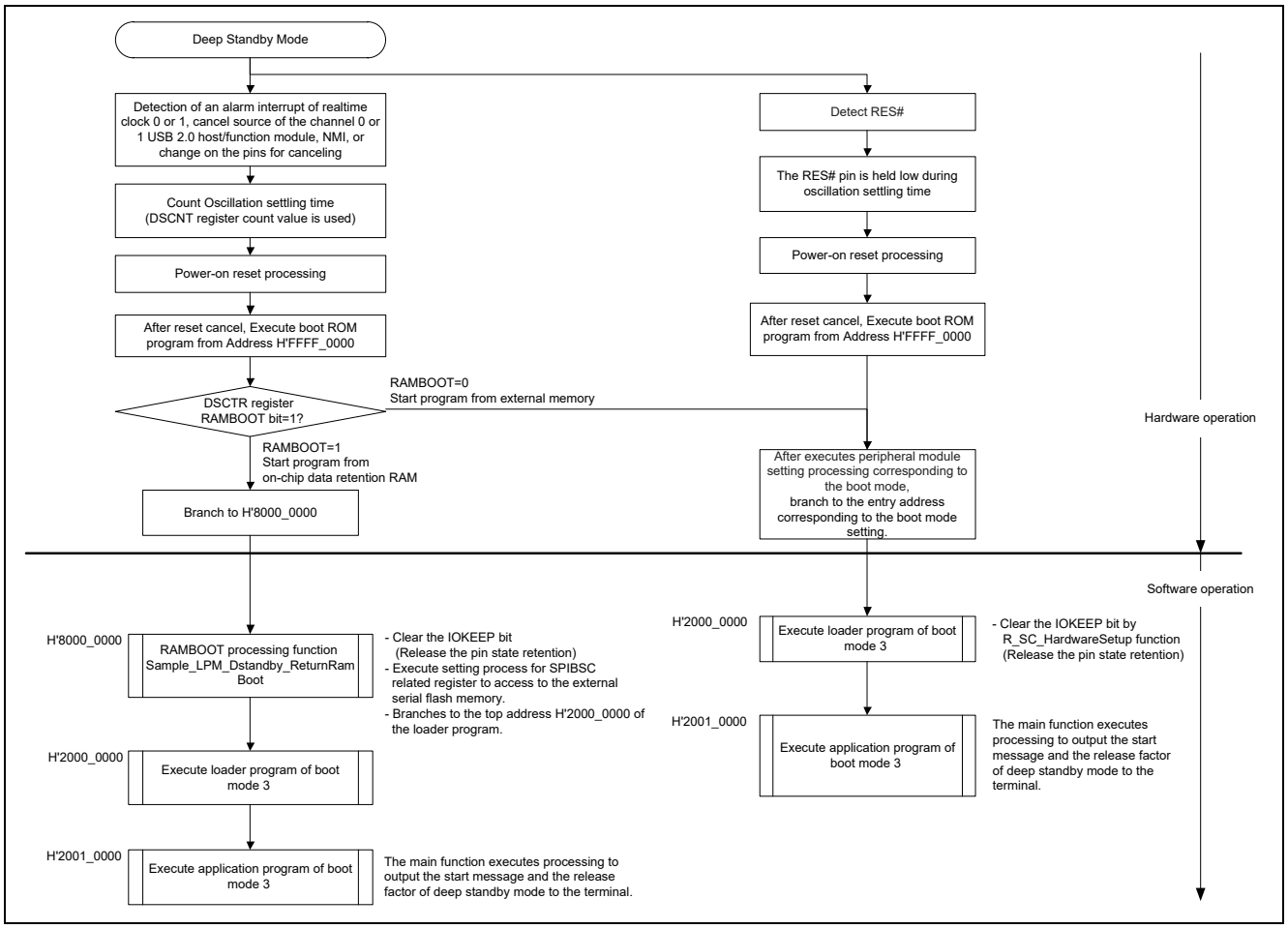


Figure 5.1 Sample Code Operation after Deep Standby Mode Cancellation

Figure 5.2 shows the Illustration of Program Placement when Starting from On-chip Data Retention RAM after Deep Standby Mode Cancellation. If the RAMBOOT bit is set to "1" and deep standby mode is canceled, the program entry point branched after executing the boot program stored in on-chip ROM at the deep standby mode cancellation, is Address H'8000_0000, which is the start address in the on-chip data retention RAM. Therefore, it is necessary to store the program which will start after the standby cancellation to Address H'8000_0000 prior to making transition to deep standby mode.

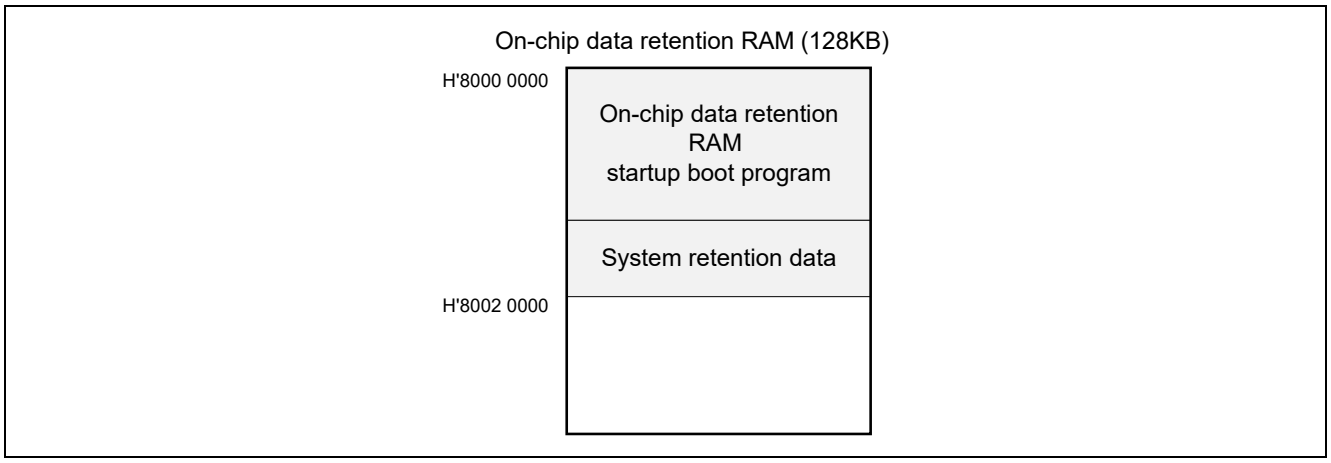


Figure 5.2 Illustration of Program Placement when Starting from On-chip Data Retention RAM after Deep Standby Mode Cancellation

If the RAMBOOT bit is set to 1 and deep standby mode is canceled, for the on-chip ROM boot program, because the peripheral module register setting process is not performed in accordance with the boot mode, it is therefore necessary to store the process for external memory access to Address H'8000_0000. For systems operating in Boot Mode 3, in order to allow access to the serial flash memory connected to the SPI multi-I/O bus controller, it is necessary to execute the peripheral module register setting processes shown in Table 5.10.

Table 5.10 Peripheral Module Setting For Serial Flash Memory Access When Recovering from On-chip Data Retention RAM

Peripheral module	Register	Register setting
Power-down mode	STBCR8	H'F7
Clock pulse generator	SCLKSEL	H'0000
SPI multi-I/O bus controller	CMNCR	H'01AA A200
	SSLDR	H'0000 0000
	DRCR	H'0003 0100
	DRCMR	H'0003 0000
	DRENR	H'0000 4700
	DRDMCR	H'0000 0000
	DRDRENR	H'0000 0000
	PHYCNT	H'0000 0260
	PHYOFFSET1	H'3151 1144
	PHYINT	H'0707 0002
General input/output port	PPOC	H'0000 0101
	PSPIBSC	H'0555 5555

5.2 Peripheral Functions and Memory Map Allocation in Sample Code

5.2.1 Setting for Peripheral Functions

Table 5.11 and Table 5.12 shows Setting for Peripheral Functions when the sample code is executed.

Table 5.11 Setting for Peripheral Functions (1/2)

Module	Setting value
CPG	CPU clock: Set to 1/2 the PLL circuit frequency Internal bus clock: Set to 1/8 the PLL circuit frequency Peripheral clock 1 (P1ϕ): Set to 1/16 the PLL circuit frequency If the input clock is 24MHz in clock mode 1 (divider 1: ×1/2, PLL circuit: ×88), set to the following frequencies • CPU clock (Iϕ): 528MHz • Image processing clock (Gϕ): 264MHz • Internal bus clock (Bϕ): 132MHz • Peripheral clock 1 (P1ϕ): 66MHz • Peripheral clock 0 (P0ϕ): 33MHz • QSPI0_SPCLK: 66MHz (When Bϕ is selected) • CKIO clock: 132MHz (When Bϕ is selected)
SPIBSC	When set to the external address space read mode, it generates the signals which enable the CPU to read directly from the serial flash memory connected to the SPI multi-I/O bus space.
OSTM	Sets Channel 0 and Channel 2 to interval timer mode • Channel 0 Sets the timer count to have interrupt request generated every 500ms when P1ϕ=66MHz • Channel 2 Sets the timer count to have interrupt request generated every 1ms when P1ϕ=66MHz
SCIFA	Sets the channel 4 in asynchronous communication mode • Data length: 8 bit • Stop bit length: 1 bit • Parity: None • Data transfer direction: LSB first transfer When P1ϕ is 66.67MHz, do not divide the clock source, set to operate with baud rate generator in double speed mode, and the bit rate to 8 times base clock. Set the bit rate to 71 so that the bit rate is 115200bps. (Difference: -0.53%)

Table 5.12 Setting for Peripheral Functions (2/2)

Module	Setting value
STB	• Write permission to on-chip data retention RAM • Clock provided to OSTM0, OSTM2, SCIFA4 and SPIBSC via STBCR3, STBCR4, and STBCR8 • Used for transition and cancellation processes for sleep mode, software standby mode, and deep standby mode • PJ_1 pin (falling edge) and NMI pin (falling edge) set as the deep standby mode cancel factor. • Deep standby mode recovery oscillation stabilization time set to 7ms
GPIO	Set PORT6, PORT9 and PORTJ shared pin functions • P6_0: Turns on and off LED • P9_1: RxD4, P9_0: TxD4 • PJ_1: Set to IRQ0 pin function, used for sleep mode and software standby mode cancellation. Also, PJ_1 pin is also used as the deep standby mode cancel factor
INTC	Initializes INTC, registers and executes for interrupts handler listed below • OSTM channel 0 interrupt (Interrupt ID: 88) • OSTM channel 2 interrupt (Interrupt ID: 90) • SCIFA channel 4 interrupt (Interrupt ID: 322,323) • IRQ0 pin interrupt (Interrupt ID: 36)
WDT	Software standby mode recovery oscillation stabilization time set to 7ms

5.2.2 Memory Mapping

Figure 5.3 shows the RZ/A2M Group Address Space and RZ/A2M CPU board memory map.

In the sample code, the code and data that use the ROM area are assigned to the serial flash memory connected to the SPI multi-I/O bus, and the code and data that use the RAM area are assigned to the large-capacity on-chip RAM. Program processing when executing from on-chip data retention RAM after deep standby mode cancellation is assigned to Address H'8000_0000 and subsequent area in the on-chip data retention RAM (128KB).

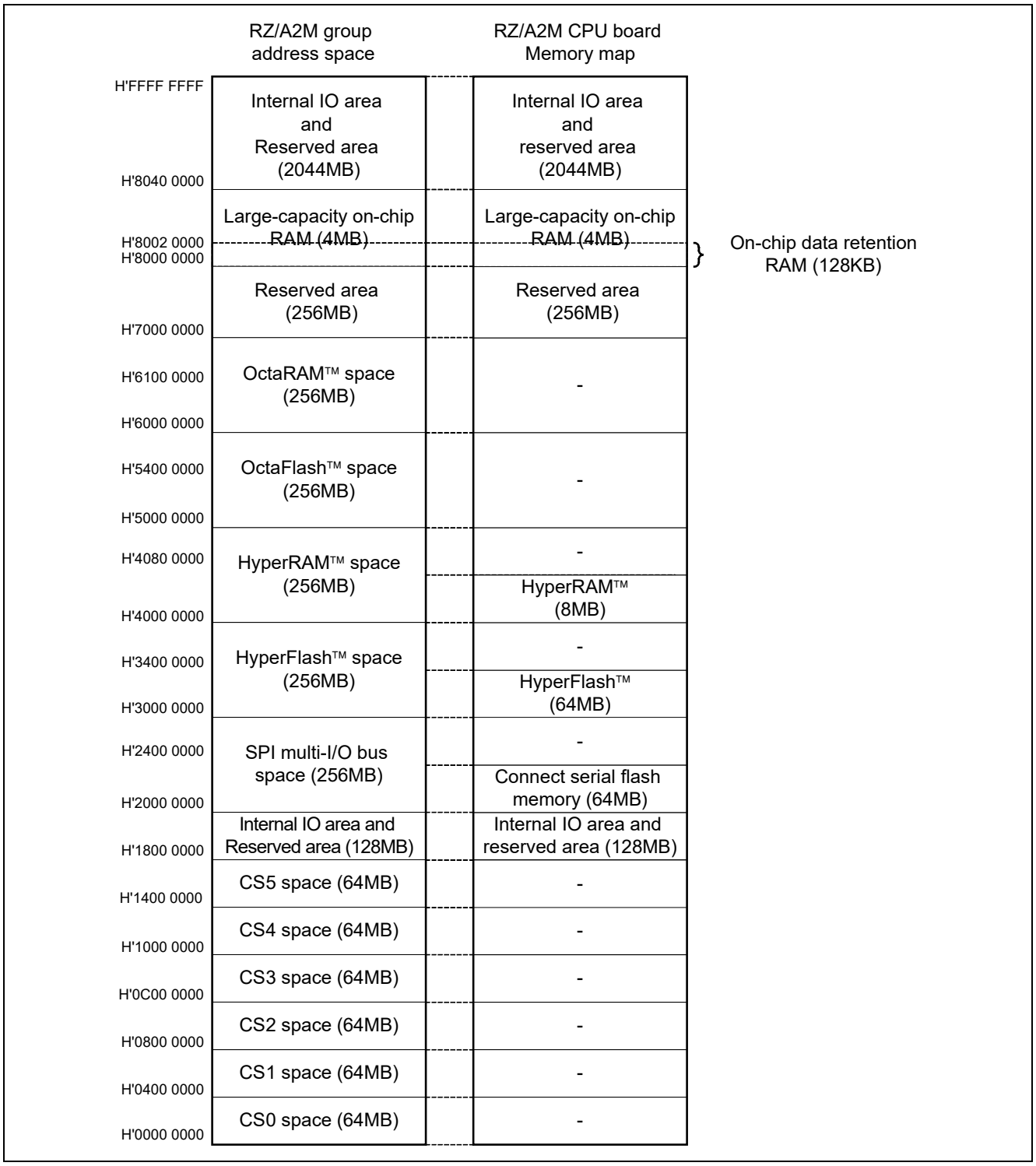


Figure 5.3 Memory Mapping

5.2.3 Section Assignment in Sample Code

In the sample code, in order to accelerate the interrupt process, the exception process vector table and the IRQ interrupt handler are placed in the large-capacity on-chip RAM, and these processes are executed on the large-capacity on-chip RAM. The process to transfer the exception process vector table and IRQ interrupt handler program code from the serial flash memory area to the large capacity on-chip RAM area, the zero clear process for the data without initial values section and the initialization process for the data with initial values section, is performed using the INITSCT function. The INISCT function, refers to the section initialization table data defined in section.c, and performs the initialization process on each section. The program data placement is recorded in the linker script (linker_script.ld).

The process to transfer the program processing when executing from on-chip data retention RAM after deep standby mode cancellation to the on-chip data retention RAM is also performed using the INTSCT function.

Table 5.13 shows the List of Sections and Objects Used in the Sample Code. The sample code initial state section allocation (Load view) and section allocation after INITSCT function execution (Execution view) is shown in Figure 5.4.

Table 5.13 List of Sections and Objects Used in the Sample Code

Output Section Name	Input Section Name Input Object Name	Description	Loading Area	Execution Area
LOAD_MODULE1	VECTOR_TABLE	Exception processing vector table	FLASH	FLASH
LOAD_MODULE2	*/r_cpg/*.o (.text .rodata .data)	CPG settings processing	FLASH	LRAM
	*/rza_io_regrw.o (.text .rodata .data)	I/O register access processing		
	/hwsetup.o (.text .rodata .data)	HardwareSetup setting processing		
LOAD_MODULE3	*/r_cpg/*.o (.bss)	Data area without default initial values for CPG settings processing	-	LRAM
	*/rza_io_regrw.o (.bss)	Data area without default initial values for I/O register access processing		
LOAD_MODULE4	RESET_HANDLER	Reset processing	FLASH	FLASH
	INIT_SECTION */sections.o	Section initialization processing		
LOAD_MODULE5	*/lpm_deepstandby_ramboot.o (.text .rodata .data)	Process to execute from on-chip data retention RAM after deep standby mode is canceled	FLASH	LRAM (Note 3)
.data	VECTOR_MIRROR_TABLE	Exception processing vector table	FLASH	LRAM
	*/r_intc_*.o (.text .rodata .data)	Code area for INTC driver processing		
	IRQ_FIQ_HANDLER	IRQ/FIQ handler processing		
.bss	None	None	-	LRAM
.uncached_RAM	*/r_cache_*.o (.bss)	Data area without default initial values for L1 and L2 cache setting processing (note 2)	-	LRAM
	UNCACHED_BSS	Data area without default initial values (non-cache settings)		
.uncached_RAM2	*/r_cache_*.o (.text .rodata .data)	L1 and L2 cache setting processing (note 2)	FLASH	LRAM
	UNCACHED_DATA	Data area with default initial values (non-cache settings)		
.mmu_page_table	None	MMU translation table area	-	LRAM
.stack	None	Stack area for system mode	-	LRAM
		Stack area for IRQ mode		
		Stack area for FIQ mode		
		Stack area for SVC mode		
		Abort (ABT) mode stack area		
.text2	* (.text .text.*)	Default code area	FLASH	FLASH
	* (.rodata .rodata.*)	Default constant data area		
.data2	* (.data .data.*)	Default area for data with initial values	FLASH	LRAM
.bss2	* (.bss .bss.*) * (COMMON)	Default area for data without initial values	-	LRAM
.heap	None	Heap area	-	LRAM

Note: 1. "FLASH" and "LRAM" shown in Load Area and Execution Area indicate the serial flash memory area and the large-capacity on-chip RAM area respectively.
2. This section must be allocated to a cache disabled area.
3. Indicates the on-chip data retention RAM (Address H'8000_0000 to Address H'8001_FFFF) for the large-capacity on-chip RAM.

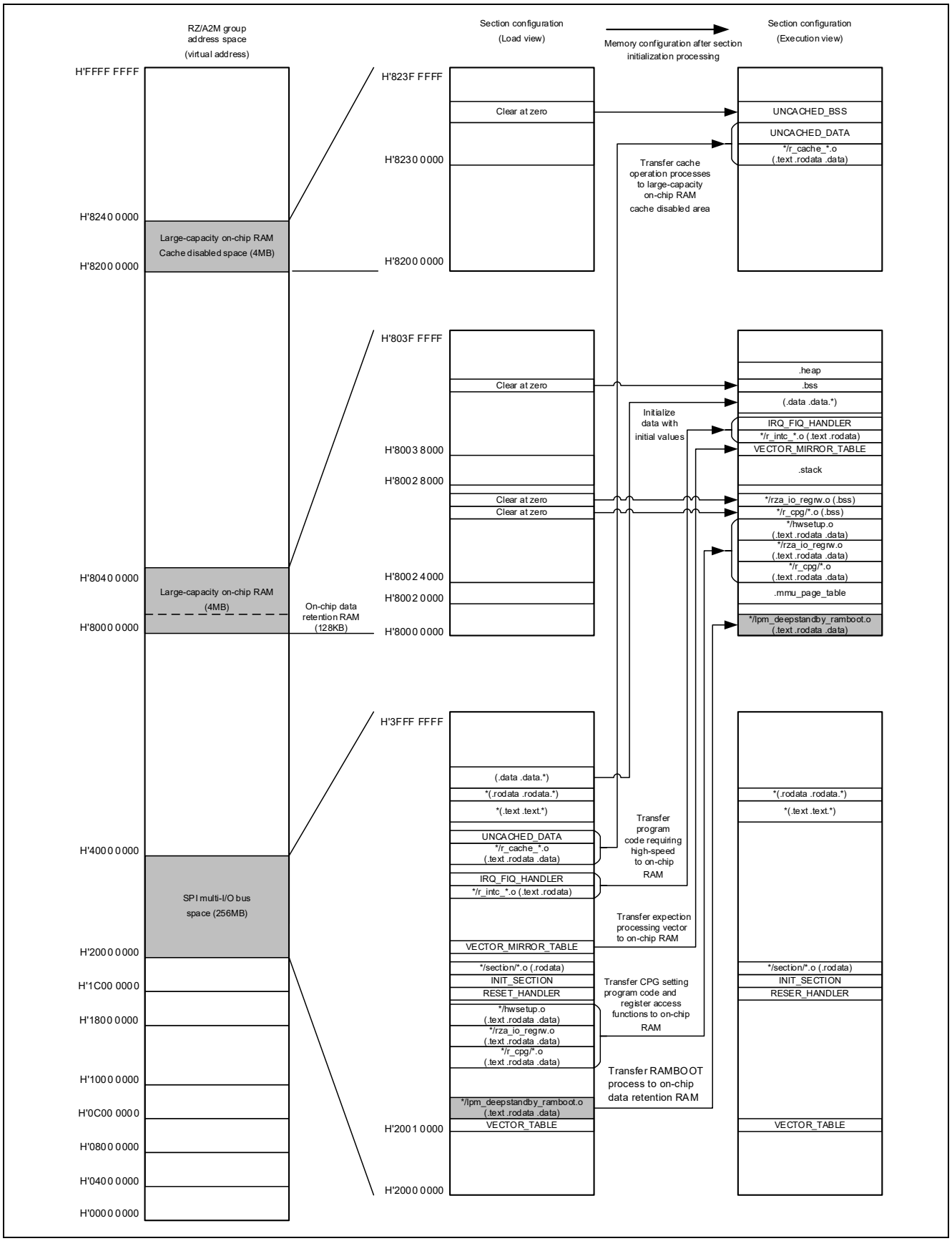


Figure 5.4 Section Allocation

5.3 Interrupt Used

Table 5.14 shows the Interrupts Used in the Sample Code.

Table 5.14 Interrupts Used in the Sample Code

Interrupt Factor (Factor ID)	Priority	Processing Summary
OSTM0 (88)	3	Generate interrupt every 500ms
OSTM2 (90)	30	Generate interrupt every 1ms (Used for time management via OS Abstraction Layer)
RXI4 (322)	30	Generate SCIFA TXI4 interrupt
TXI4 (323)	30	Generate SCIFA RXI4 interrupt
IRQ0 (36)	1	Generate interrupt when IRQ0 pin falling edge is detected. Used for sleep mode and software standby mode cancel factor
NMI	—	Generate interrupt when NMI pin falling edge is detected. Used for sleep mode and software standby mode cancel factor

Note: During transition to sleep mode or software standby mode, set the interrupt mask level to 2 before transition to the standby mode in order to prevent receipt of any non-NMI or IRQ0 interrupt requests. After sleep mode or software standby mode have been canceled, return the interrupt mask level to 31 in order to permit other interrupts.

5.4 Data Types

Table 5.15 shows the Data Types Used in the Sample Code.

Table 5.15 Data Types Used in the Sample Code

Symbol	Description
char_t	8-bit character
bool_t	Boolean type. Value is true (1) or false (0).
int_t	Fast integer, signed, 32-bit integer in this sample code
int8_t	8-bit integer, signed (defined in standard library stdint.h)
int16_t	16-bit integer, signed (defined in standard library stdint.h)
int32_t	32-bit integer, signed (defined in standard library stdint.h)
int64_t	64-bit integer, signed (defined in standard library stdint.h)
uint8_t	8-bit integer, unsigned (defined in standard library stdint.h)
uint16_t	16-bit integer, unsigned (defined in standard library stdint.h)
uint32_t	32-bit integer, unsigned (defined in standard library stdint.h)
uint64_t	64-bit integer, unsigned (defined in standard library stdint.h)
float32_t	32-bit float
float64_t	64-bit float
float128_t	128-bit float

5.5 Constants

Table 5.16 and Table 5.18 list the constants used by the low power mode sample program. These constants are used when setting the member variables in the configuration table LPM_SC_TABLE[]. For details regarding configuration table LPM_SC_TABLE[], refer to Table 5.19 to Table 5.23.

Table 5.16 Constants Used in Low Power Mode Sample Program (1/3)

Constant	Setting Value	Description
LPM_MODE_SLEEP	(0)	Used in setting the mode member for the configuration table LPM_SC_TABLE[]. Specifies the table format for the low power mode being used. • For transition to sleep mode Set LPM_MODE_SLEEP • For transition to software standby mode Set LPM_MODE_SOFTWARE_STANDBY • For transition to deep standby mode Set LPM_MODE_DEEP_STANDBY
LPM_MODE_SOFTWARE_STANDBY	(1)	
LPM_MODE_DEEP_STANDBY	(2)	
LPM_SSTANDBY_DISABLE_IRQ	(0)	Used in setting ss_cancel_by_irq member for the configuration table LPM_SC_TABLE[]. Specifies whether the IRQ pin will be used for the software standby mode cancel factor. • If the IRQ pin is not used as the cancel factor LPM_SSTANDBY_DISABLE_IRQ • If the IRQ pin is used as the cancel factor LPM_SSTANDBY_ENABLE_IRQ
LPM_SSTANDBY_ENABLE_IRQ	(1)	
LPM_PIN_CANCEL_FACTOR_DISABLED	(0)	Used to set the ds_cancel_by_pk_4, ds_cancel_by_pk_2, ds_cancel_by_pj_5, ds_cancel_by_pj_1, ds_cancel_by_ph_0, ds_cancel_by_pg_6, ds_cancel_by_pg_2, ds_cancel_by_ph_1, ds_cancel_by_pe_1, ds_cancel_by_p6_2, ds_cancel_by_p3_3, ds_cancel_by_p3_1 members in the configuration table LPM_SC_TABLE[]. Specifies whether pin change is used for the deep standby mode cancel factor, and specifies detection method if it is used. • If pin change is not used as the cancel factor LPM_PIN_CANCEL_FACTOR_DISABLED • If pin change is used as cancel factor and set to falling edge LPM_PIN_CANCEL_FACTOR_FALLING_EDGE • If pin change is used as cancel factor and set to rising edge LPM_PIN_CANCEL_FACTOR_RISING_EDGE
LPM_PIN_CANCEL_FACTOR_FALLING_EDGE	(1)	
LPM_PIN_CANCEL_FACTOR_RISING_EDGE	(2)	

Table 5.17 Constants Used in Low Power Mode Sample Program (2/3)

Constant	Setting Value	Description
LPM_SS_CANCEL_FACTOR_DISABLED	(0)	Used in setting the configuration table LPM_SC_TABLE[] ss_cancel_by_usb0_cc2_rd, ss_cancel_by_usb0_cc1_rd, ss_cancel_by_usb0_ovrclr, ss_cancel_by_usb0_vbusin, ss_cancel_by_usb0_dm, ss_cancel_by_usb0_dp, ss_cancel_by_usb1_cc2_rd, ss_cancel_by_usb1_cc1_rd, ss_cancel_by_usb1_ovrclr, ss_cancel_by_usb1_vbusin, ss_cancel_by_usb1_dm, ss_cancel_by_usb1_dp members. Specifies whether USB module interrupt will be used for the software standby mode cancel factor. • If the USB module is not used as the cancel factor LPM_SS_CANCEL_FACTOR_DISABLED • If the USB module is used as the cancel factor LPM_SS_CANCEL_FACTOR_ENABLED
LPM_SS_CANCEL_FACTOR_ENABLED	(1)	
LPM_DS_CANCEL_FACTOR_DISABLED	(0)	Used in setting the ds_cancel_by_rtc0, ds_cancel_by_rtc1, ds_cancel_by_usb0_dm, ds_cancel_by_usb0_dp, ds_cancel_by_usb1_dm ,ds_cancel_by_usb1_dp members for the configuration table LPM_SC_TABLE[]. Specifies whether RTC alarm interrupt and USB module interrupt will be used for the deep standby mode cancel factor. • If the RTC and USB module are not used as the cancel factor LPM_DS_CANCEL_FACTOR_DISABLED • If the RTC and USB module are used as the cancel factor LPM_DS_CANCEL_FACTOR_ENABLED
LPM_DS_CANCEL_FACTOR_ENABLED	(1)	
LPM_PIN_STATUS_RETAINED	(0)	Used in setting the pin status member for the configuration table LPM_SC_TABLE[]. Specifies the pin state during transition to software standby mode and deep standby mode. • If pin state is retained during transition to standby mode LPM_PIN_STATUS_RETAINED • If pin state is set to Hi-z state during transition to standby mode LPM_PIN_STATUS_HIZ
LPM_PIN_STATUS_HIZ	(1)	

Table 5.18 Constants Used in Low Power Mode Sample Program (3/3)

Constant	Setting Value	Description
LPM_RETENTION_RAM_NOT_RETAINED	(0)	Used to set the ds_retention_ram_page0 to ds_retention_ram_page3 members in the LPM_SC_TABLE[] configuration table. Specifies whether the contents of on-chip data retention RAM are retained during deep standby mode transition. ● If contents of on-chip data retention RAM are not retained LPM_RETENTION_RAM_NOT_RETAINED ● If contents of on-chip data retention RAM are retained LPM_RETENTION_RAM_RETAINED
LPM_RETENTION_RAM_RETAINED	(1)	
LPM_REBOOT_TYPE_EXTERNAL_MEMORY_BOOT	(0)	Used in setting the ds_reboot member for the configuration table LPM_SC_TABLE[]. Specifies startup method after deep standby mode cancellation. ● If starting from external memory LPM_REBOOT_TYPE_EXTERNAL_MEMORY_BOOT ● If starting from on-chip data retention RAM (external memory pin state not retained) LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_0 ● If starting from on-chip data retention RAM (external memory pin state is retained) LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_1
LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_0	(1)	
LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_1	(2)	

5.6 List of Structures/Unions

The low power mode configuration tables are shown in Table 5.19 to Table 5.23. The configuration table is used to specify the standby mode cancel factor, state during standby transition, and operation after cancellation. Only one low power mode can be defined in the configuration table for each of sleep mode, software standby mode, and deep standby mode.

Table 5.19 Configuration Structure (st_r_drv_lpm_sc_config_t) (1/5)

Classification	Member	Description
Common to all modes	e_r_drv_lpm_mode_t mode	Specifies the table format for the low power mode being used. ● Settable values: LPM_MODE_SLEEP LPM_MODE_SOFTWARE_STANDBY LPM_MODE_DEEP_STANDBY
Common to software standby mode and deep standby mode	e_r_drv_lpm_pin_ status_t pin_status	Specifies the pin state during transition to software standby mode and deep standby mode. ● Settable values: LPM_PIN_STATUS_RETAINED LPM_PIN_STATUS_HIZ
Software standby mode	e_r_drv_lpm_cancel_ factor_irq_t ss_cancel_by_irq	Specifies whether the IRQ pin will be used for the cancel factor. ● Settable values: LPM_SSTANDBY_DISABLE_IRQ LPM_SSTANDBY_ENABLE_IRQ
	e_r_drv_lpm_cancel_ factor2_t ss_cancel_by_usb0_ cc2_rd	Specifies whether USB channel 0 CC2_RD signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_cancel_ factor2_t ss_cancel_by_usb0_ cc1_rd	Specifies whether USB channel 0 CC1_RD signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_cancel_ factor2_t ss_cancel_by_usb0_ ovrclr	Specifies whether USB channel 0 OVRCLR signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_cancel_ factor2_t ss_cancel_by_usb0_ vbusin	Specifies whether USB channel 0 VBUSIN signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_cancel_ factor2_t ss_cancel_by_usb0_ dm	Specifies whether USB channel 0 DM signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED

Table 5.20 Configuration Structure (st_r_drv_lpm_sc_config_t) (2/5)

Classification	Member	Description
Software standby mode	e_r_drv_lpm_cancel_factor2_t ss_cancel_by_usb0_dp	Specifies whether USB channel 0 DP signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_cc2_rd	Specifies whether USB channel 1 CC2_RD signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_cc1_rd	Specifies whether USB channel 1 CC1_RD signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_ovrclr	Specifies whether USB channel 1 OVRCLR signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_vbusin	Specifies whether USB channel 1 VBUSIN signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_dm	Specifies whether USB channel 1 DM signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ss_cancel_factor_t ss_cancel_by_usb1_dp	Specifies whether USB channel 1 DP signal change will be used for the cancel factor. ● Settable values: LPM_SS_CANCEL_FACTOR_DISABLED LPM_SS_CANCEL_FACTOR_ENABLED
	float64_t ss_osc_stab_time	Specifies ms unit for software standby mode recovery oscillation stabilization time (Up to 2 digits after the decimal point allowed). ● Settable values: Decimal fraction for 1 or more

Table 5.21 Configuration Structure (st_r_drv_lpm_sc_config_t) (3/5)

Classification	Member	Description
Deep Standby mode	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pk_4	Specifies whether the cancel factor uses PK_4 pin signal change, and specifies detection method if it is used. • Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pk_2	Specifies whether the cancel factor uses PK_2 pin signal change, and specifies detection method if it is used. • Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pj_5	Specifies whether the cancel factor uses PJ_5 pin signal change, and specifies detection method if it is used. • Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pj_1	Specifies whether the cancel factor uses PJ_1 pin signal change, and specifies detection method if it is used. • Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_ph_0	Specifies whether the cancel factor uses PH_0 pin signal change, and specifies detection method if it is used. • Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pg_6	Specifies whether the cancel factor uses PG_6 pin signal change, and specifies detection method if it is used. • Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_nmi	Specifies whether the cancel factor uses NMI pin signal change, and specifies detection method if it is used. • Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_cancel_factor_t ds_cancel_by_rtc1	Specifies whether RTC channel 1 alarm interrupt will be used for the cancel factor. • Settable values: LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED

Table 5.22 Configuration Structure (st_r_drv_lpm_sc_config_t) (4/5)

Classification	Member	Description
Deep Standby mode	e_r_drv_lpm_ds_cancel_factor_t ds_cancel_by_rtc0	Specifies whether RTC channel 0 alarm interrupt will be used for the cancel factor. ● Settable values: LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pg_2	Specifies whether the cancel factor uses PG_2 pin signal change, and specifies detection method if it is used. ● Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_ph_1	Specifies whether the cancel factor uses PH_1 pin signal change, and specifies detection method if it is used. ● Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_pe_1	Specifies whether the cancel factor uses PE_1 pin signal change, and specifies detection method if it is used. ● Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDG
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_p6_2	Specifies whether the cancel factor uses P6_2 pin signal change, and specifies detection method if it is used. ● Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_p3_3	Specifies whether the cancel factor uses P3_3 pin signal change, and specifies detection method if it is used. ● Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_p3_1	Specifies whether the cancel factor uses P3_1 pin signal change, and specifies detection method if it is used. ● Settable values: LPM_PIN_CANCEL_FACTOR_DISABLED LPM_PIN_CANCEL_FACTOR_FALLING_EDGE LPM_PIN_CANCEL_FACTOR_RISING_EDGE
	e_r_drv_lpm_ds_pin_cancel_factor_t ds_cancel_by_usb1_dm	Specifies whether USB channel 1 DM signal change will be used for the cancel factor. ● Settable values: LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED

Table 5.23 Configuration Structure (st_r_drv_lpm_sc_config_t) (5/5)

Classification	Member	Description
Deep Standby mode	e_r_drv_lpm_ds_cancel_factor_t ds_cancel_by_usb1_dp	Specifies whether USB channel 1 DP signal change will be used for the cancel factor. ● Settable values: LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ds_cancel_factor_t ds_cancel_by_usb0_dm	Specifies whether USB channel 0 DM signal change will be used for the cancel factor. ● Settable values: LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_ds_cancel_factor_t ds_cancel_by_usb0_dp	Specifies whether USB channel 0 DP signal change will be used for the cancel factor. ● Settable values: LPM_DS_CANCEL_FACTOR_DISABLED LPM_DS_CANCEL_FACTOR_ENABLED
	e_r_drv_lpm_retention_ram_t ds_retention_ram_page3	Specifies whether the contents of on-chip data retention RAM page 3 are retained during deep standby mode transition. ● Settable values: LPM_RETENTION_RAM_NOT_RETAINED LPM_RETENTION_RAM_RETAINED
	e_r_drv_lpm_retention_ram_t ds_retention_ram_page2	Specifies whether the contents of on-chip data retention RAM page 2 are retained during deep standby mode transition. ● Settable values: LPM_RETENTION_RAM_NOT_RETAINED LPM_RETENTION_RAM_RETAINED
	e_r_drv_lpm_retention_ram_t ds_retention_ram_page1	Specifies whether the contents of on-chip data retention RAM page 1 are retained during deep standby mode transition. ● Settable values: LPM_RETENTION_RAM_NOT_RETAINED LPM_RETENTION_RAM_RETAINED
	e_r_drv_lpm_retention_ram_t ds_retention_ram_page0	Specifies whether the contents of on-chip data retention RAM page 0 are retained during deep standby mode transition. ● Settable values: LPM_RETENTION_RAM_NOT_RETAINED LPM_RETENTION_RAM_RETAINED
	float64_t ds_osc_stab_time	Specifies ms unit for deep standby mode recovery oscillation stabilization period (Up to 2 digits after the decimal point allowed). ● Settable values: Decimal fraction for 1 or more
	e_r_drv_lpm_reboot_type_t ds_reboot_type	Specifies startup method after deep standby mode cancellation. ● Settable values: LPM_REBOOT_TYPE_EXTERNAL_MEMORY_BOOT LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_0 LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEPE_1

5.7 Functions

This sample code consists of the interface functions to use the on-chip peripheral functions (API functions), and the sample functions required to operate the sample code.

Sample Functions List is shown in Table 5.24 and API Functions List is shown in Table 5.25.

Table 5.24 Sample Functions List

Function	Description
Sample_LPM_Sleep_Mode	Sleep mode transition sample function
Sample_LPM_Software_Standby	Software standby mode transition sample function
Sample_LPM_Deep_Standby	Deep standby mode transition sample function
Sample_NMI_Interrupt	NMI pin interrupt sample function
Sample_IRQ0_Interrupt	IRQ0 pin interrupt sample function
Sample_LPM_DStandby_ReturnRamBoot	Process from on-chip data retention RAM after deep standby mode is canceled

Table 5.25 API Functions List

Function	Description
R_LPM_Transition	Low power mode transition function
R_LPM_SleepTransition	Sleep mode transition function
R_LPM_SStandbyTransition	Software standby mode transition function
R_LPM_DStandbyTransition	Deep standby mode transition function
R_INTC_SetIrqMask	Mask control for software standby mode IRQ pin interrupt request
R_INTC_SetNMIConfig	Sets the NMI pin interrupt request detection method
R_IRQ_SetSense	Sets the IRQ pin interrupt request detection method

5.8 Function Specifications

Specifications of the functions of the sample code are listed below.

5.8.1 Sample Function Specification

Sample_LPM_Sleep_Mode		
Outline	Sleep mode transition sample function	
Declaration	int_t Sample_LPM_Sleep_Mode(void)	
Description	This is the sample function to make transition to sleep mode. This function will execute when the command string "LPM_SM" is input to the terminal. The IRQ0 pin interrupt and NMI pin interrupt are set as the sleep mode cancel factor, the R_LPM_Transition function is called, and the transition to sleep mode process is executed. (Calling R_LPM_SleepTransition function in the above function.)	
Argument	None	
Return Value	LPM_SUCCESS	: Completed normally
	LPM_ERROR	: Completed with error

Sample_LPM_Software_Standby		
Outline	Software standby mode transition sample function	
Declaration	int_t Sample_LPM_Software_Standby(st_r_drv_lpm_sc_config_t *p_config_tbl)	
Description	This is a sample function for software standby mode transition. This function will execute when the command string "LPM_SS" is input to the terminal. The IRQ0 pin interrupt and NMI pin interrupt are set as the software standby mode cancel factor, the R_LPM_Transition function is called, and the transition to software standby mode process is executed. (Calling R_LPM_SStandbyTransition function in the above function.)	
Argument	st_r_drv_lpm_sc_config_t *p_config_tbl	Configuration table pointer
Return Value	LPM_SUCCESS	: Completed normally
	LPM_ERROR	: Completed with error

Sample_LPM_Deep_Standby		
Outline	Deep standby mode transition sample function	
Declaration	int_t Sample_LPM_Deep_Standby(st_r_drv_lpm_sc_config_t *p_config_tbl)	
Description	This is a sample function for deep standby mode transition. This function will execute when the command string "LPM_DS"+"1" or "LPM_DS"+"2" is input to the terminal. The PJ_1 pin falling edge and NMI pin falling edge are set as the deep standby mode cancel factor, the R_LPM_Transition function is called, and the transition to deep standby mode process is executed. (Calling R_LPM_DStandbyTransition function in the above function.)	
Argument	st_r_drv_lpm_sc_config_t *p_config_tbl	Configuration table pointer
Return Value	LPM_SUCCESS	: Completed normally
	LPM_ERROR	: Completed with error

Sample_NMI_Interrupt	
Outline	NMI pin interrupt sample function
Declaration	void Sample_NMI_Interrupt(uint32_t int_sense)
Description	This is the sample NMI pin interrupt function executed when sleep mode or software standby mode are canceled. In the sample code, "0" is written to the interrupt control register 0 (IRQ0) NMIF bit, and then the NMI interrupt request clear process is executed.
Argument	uint32_t int_sense : Not used
Return Value	None

Sample_IRQ0_Interrupt	
Outline	IRQ0 pin interrupt sample function
Declaration	void Sample_IRQ0_Interrupt(uint32_t int_sense)
Description	This is the sample IRQ0 pin interrupt function executed when sleep mode or software standby mode are canceled. In the sample code, "0" is written to the IRQ interrupt request register (IRQRR) IRQ0F bit, and then the IRQ0 interrupt request clear process is executed.
Argument	uint32_t int_sense : Not used
Return Value	None
Precautions	When using multiple IRQ pin interrupt requests with different interrupt priority levels, if register setting value to clear the IRQnF bit is generated from the read value of IRQRR and the value is written to IRQRR, the another pending IRQ pin interrupt requests may be cleared depend on the write timing to IRQRR. In the sample code, the IRQ0F bit clear value is no generated from the IRQRR read value, and so that only IRQ0F is cleared, "1" is written to IRQ7F to IRQ1F bits, and "0" is written to the IRQ0F bit.

Sample_LPM_DStandby_ReturnRamBoot	
Outline	Process from on-chip data retention RAM after deep standby mode is canceled
Declaration	void Sample_LPM_DStandby_ReturnRamBoot(void)
Description	Sample function for the process to execute from on-chip data retention RAM after deep standby mode is canceled. If the Deep Standby Control register (DSCTR) RAMBOOT bit is set to 1 and transition to deep standby mode is canceled, this is called after the on-chip ROM boot program is executed. In the sample code, the pin state retention is canceled to enable clearing of the IOKEEP bit, peripheral module register setting is performed to use external memory and it branches to the loader program start address (Address H'2000_0000).
Argument	None
Return Value	None

5.8.2 API Function Specifications

R_LPM_Transition	
Outline	Low power mode transition function
Declaration	int_t R_LPM_Transition(e_r_drv_lpm_mode_t mode, e_r_drv_lpm_reboot_type_t ds_reboot_type)
Description	Performs process for transition to low power mode. Call the API function to transition to the low power mode specified by the argument mode. When LPM_MODE_SOFTWARE_STANDBY or LPM_MODE_DEEP_STANDBY is specified for the argument mode, the configuration table specified by the argument mode and the argument ds_reboot_type is used. (If LPM_MODE_SLEEP is specified for the argument mode, the configuration table is not used.)
Argument	e_r_drv_lpm_mode_t mode — Low power mode used LPM_MODE_SLEEP : Sleep mode LPM_MODE_SOFTWARE_STANDBY : Software standby mode LPM_MODE_DEEP_STANDBY : Deep standby mode e_r_drv_lpm_reboot_type_t ds_reboot_type — How to booting after canceling deep standby mode LPM_REBOOT_TYPE_EXTERNAL_MEMORY_BOOT : Booting from external memory LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEP_0 : Booting from on-chip data retention RAM (The states of the external memory control pins are not retained.) LPM_REBOOT_TYPE_RAMBOOT_EBUSKEEP_1 : Booting from on-chip data retention RAM (The states of the external memory control pins are retained.)
Return Value	DRV_SUCCESS : Completed normally DRV_ERROR : Completed with error
Precautions	Before calling this function and making transition to low power mode, set the interrupt function for cancel factor of used low power mode in accordance with the system. If the IRQ7 to IRQ0 pin interrupts are used for the software standby mode cancel factor, prior to calling this function, set the pin function to IRQ7 to IRQ0 pin.

R_LPM_SleepTransition	
Outline	Sleep mode transition function
Declaration	int_t R_LPM_SleepTransition(void)
Description	Performs process for transition to sleep mode.
Argument	None
Return Value	DRV_SUCCESS : Completed normally DRV_ERROR : Completed with error
Precautions	Before calling this function and making transition to sleep mode, set the interrupt function for sleep mode cancel factor in accordance with the system. If multiple low power modes are defined in the configuration table, call the R_LPM_Transition function instead of calling this function directly.

R_LPM_SStandbyTransition	
Outline	Software standby mode transition function
Declaration	int_t R_LPM_SStandbyTransition(st_r_drv_lpm_sc_config_t *p_config_tbl)
Description	Perform process for transition to software standby mode. Using the configuration table specified in the argument (refer to Table 5.19 to Table 5.23), the pin status during software standby mode transition, the software standby mode cancel factor and standby recovery oscillation stabilization time are set, and makes transition to software standby mode.
Argument	st_r_drv_lpm_sc_config_t *p_config_tbl : Configuration table pointer
Return Value	DRV_SUCCESS : Completed normally DRV_ERROR : Completed with error
Precautions	Before calling this function and making transition to software standby mode, set the interrupt function for software standby mode cancel factor in accordance with the system. If the IRQ7 to IRQ0 pin interrupts are used for the software standby mode cancel factor, prior to calling this function, set the pin function to IRQ7 to IRQ0 pin. If multiple low power modes are defined in the configuration table, call the R_LPM_Transition function instead of calling this function directly.

R_LPM_DStandbyTransition	
Outline	Deep standby mode transition function
Declaration	int_t R_LPM_DStandbyTransition(st_r_drv_lpm_sc_config_t *p_config_tbl)
Description	Performs process for transition to deep standby mode. Using the configuration table specified in the argument (refer to Table 5.19 to Table 5.23), the pin status during deep standby mode transition, the deep standby mode cancel factor, the retention setting for on-chip data retention RAM contents and standby recovery oscillation stabilization time are set, the program startup method after cancellation is set, and it makes transition to deep standby mode.
Argument	st_r_drv_lpm_sc_config_t *p_config_tbl : Configuration table pointer
Return Value	DRV_SUCCESS : Completed normally DRV_ERROR : Completed with error
Precautions	Before calling this function and making transition to deep standby mode, set the module that is the deep standby mode cancel factor in accordance with the system. If multiple low power modes are defined in the configuration table, call the R_LPM_Transition function instead of calling this function directly.

R_INTC_SetIrqMask	
Outline	Mask control for software standby mode IRQ pin interrupt request
Declaration	e_r_drv_intc_err_t R_INTC_SetIrqMask(e_r_drv_irq_mask_t mask)
Description	Performs the process to specify whether an IRQ pin (IRQ7 to IRQ0) interrupt will be used for the software standby mode cancel factor. The value specified in the mask argument is used, and the IRQ interrupt request register (IRQRR) IRQMSK bit is set. In the sample code, IRQ_MASK_OFF is specified in the mask argument, this function is called, and the IRQ pin interrupt is set as the software standby mode cancel factor.
Argument	e_r_drv_irq_mask_t mask : Mask value IRQ_MASK_ON (Mask IRQ cancel factor) IRQ_MASK_OFF (Do not mask IRQ cancel factor)
Return Value	INTC_SUCCESS : Completed normally INTC_ERR_INVALID : Abnormal end
Precautions	When the GPIO ports are switched to IRQ7 to IRQ0 pin functions, the software standby mode cancel factor may be set. If the IRQ7 to IRQ0 pins are used for the software standby mode cancel factor,Set the IRQMSK bit to "0" before switching the GPIO port to the IRQ7 to IRQ0 pin function. By setting the mask argument for this function to IRQ_MASK_ON, it is possible to set the IRQMSK bit to "0".

R_INTC_SetNMIConfig	
Outline	Sets the NMI pin interrupt request detection method
Declaration	e_r_drv_intc_err_t R_INTC_SetNMIConfig(const st_r_drv_nmi_cfg_t * p_nmi_config)
Description	When the pin state specified in the p_nmi_config argument is becamed, the NMI pin interrupt request can be detected.
Argument	const st_r_drv_nmi_cfg_t * p_nmi_config : NMI pin interrupt generate factor NMI_SENSE_FALLINGEDGE (falling edge) NMI_SENSE_RISINGEDGE (rising edge)
Return Value	INTC_SUCCESS : Completed normally

R_IRQ_SetSense	
Outline	Sets the IRQ pin interrupt request detection method
Declaration	e_r_drv_intc_err_t R_IRQ_SetSense(e_r_drv_irq_num_t irq_num, e_r_drv_irq_sense_t sense)
Description	Sets the IRQ pin interrupt request detection method This function sets the IRQ7 to IRQ0 pins specified by the argument irq_num to detect an IRQ pin interrupt request in the pin state specified by the argument sense.
Argument	e_r_drv_irq_num_t irq_num : IRQ number INTC_IRQ0 (IRQ0 pin) INTC_IRQ1 (IRQ1 pin) INTC_IRQ2 (IRQ2 pin) INTC_IRQ3 (IRQ3 pin) INTC_IRQ4 (IRQ4 pin) INTC_IRQ5 (IRQ5 pin) INTC_IRQ6 (IRQ6 pin) INTC_IRQ7 (IRQ7 pin) e_r_drv_irq_sense_t sense : IRQ pin interrupt generate factor IRQ_SENSE_LEVEL (low level) IRQ_SENSE_FALLINGEDGE (falling edge) IRQ_SENSE_RISINGEDGE (rising edge) IRQ_SENSE_BOTHEDGE (both edges)
Return Value	INTC_SUCCESS : Completed normally INTC_ERR_INVALID_NUM : The irq_num argument is out of range INTC_ERR_INVALID_SENSE : Argment sense is out of range

5.9 Flowcharts

5.9.1 Sleep Mode Transition Sample Function

The flow chart for Sleep Mode Transition Sample Function is shown in Figure 5.5 and Figure 5.6.

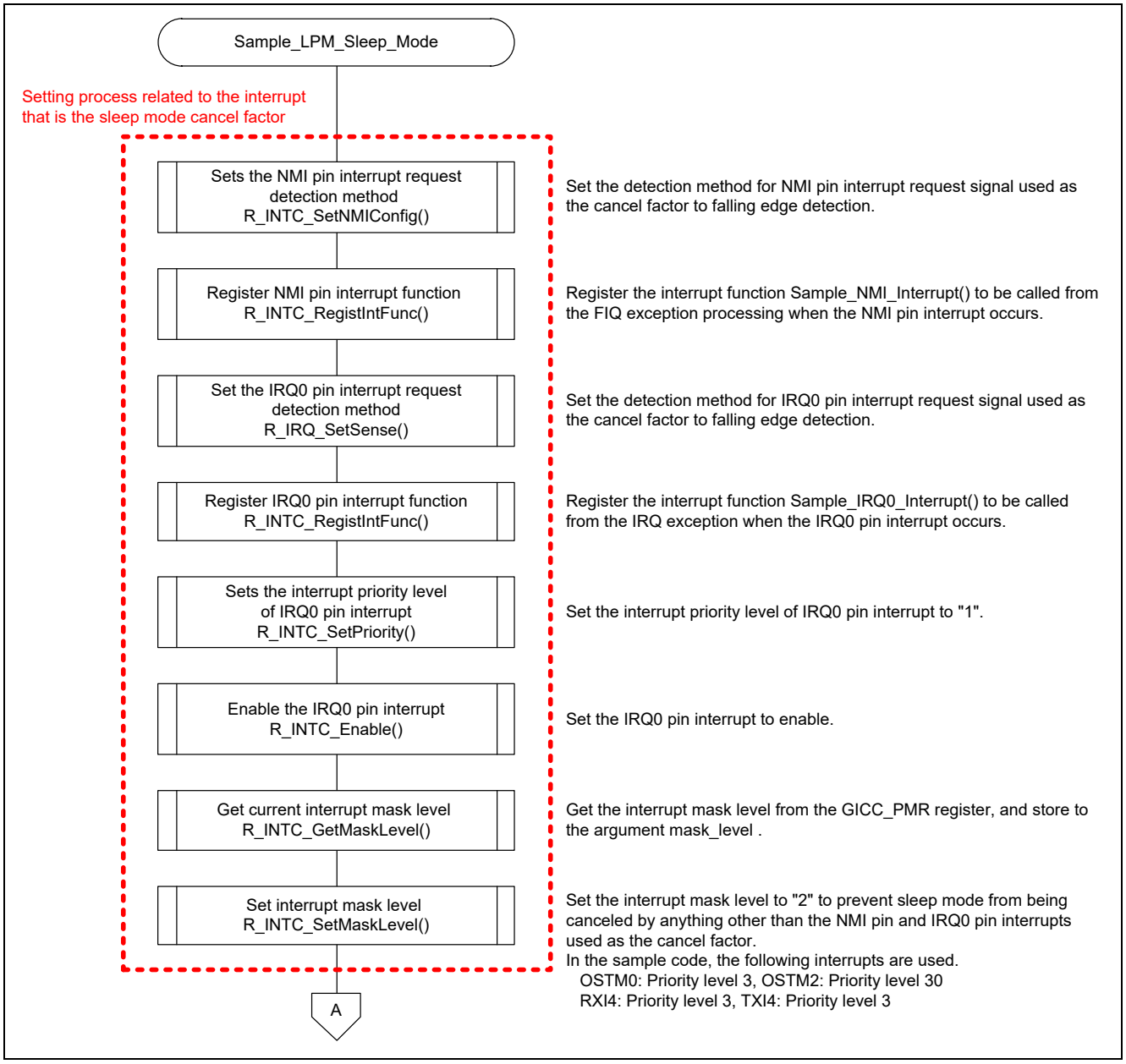


Figure 5.5 Sleep Mode Transition Sample Function (1/2)

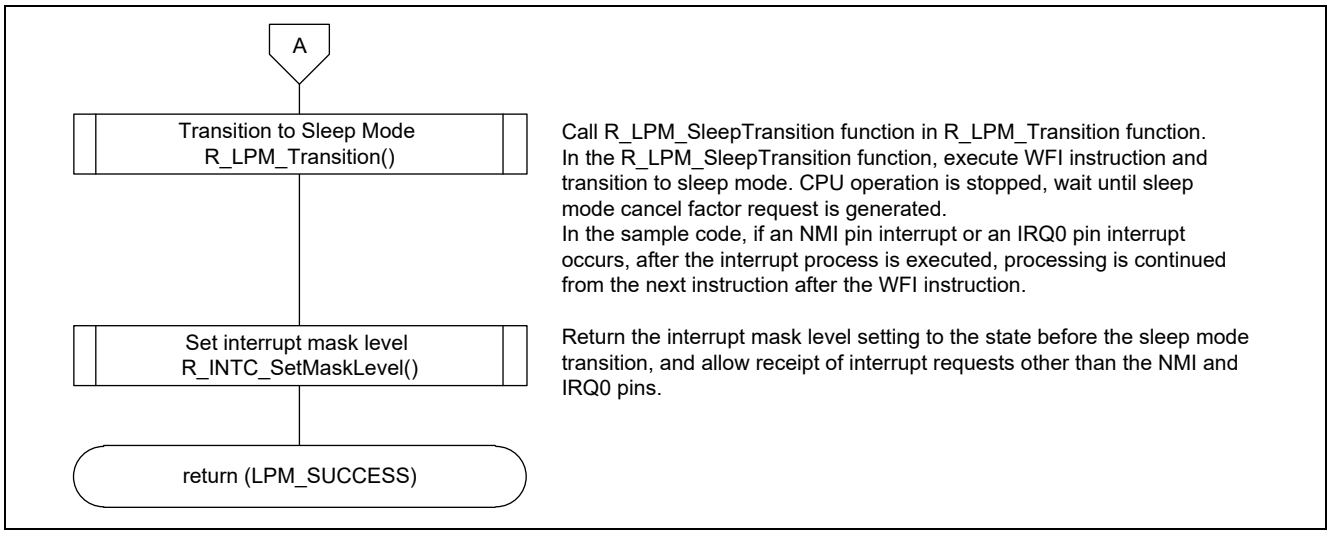


Figure 5.6 Sleep Mode Transition Sample Function (2/2)

5.9.2 Sleep Mode Transition Function

The flow chart for Sleep Mode Transition Function is shown in Figure 5.7.

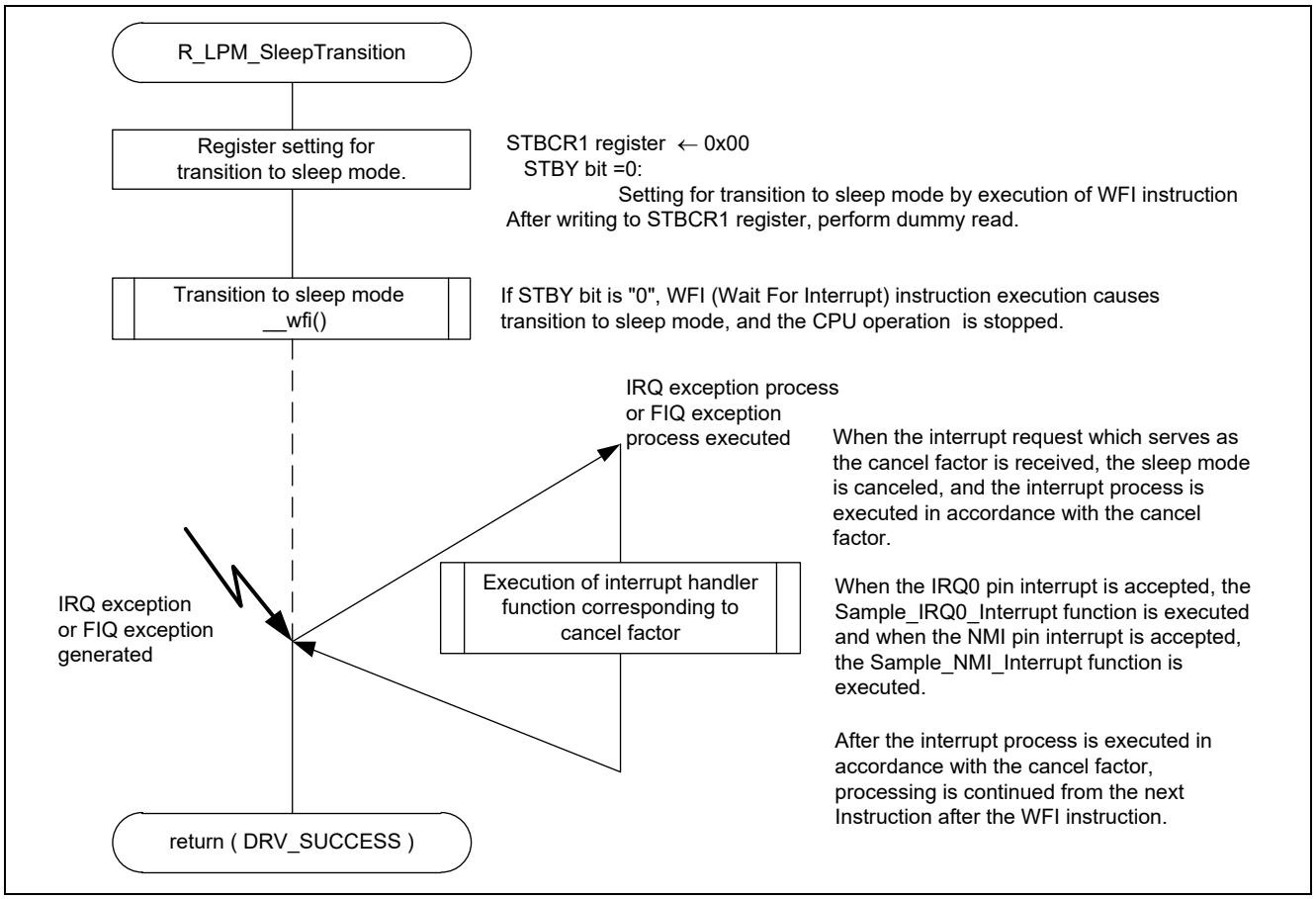


Figure 5.7 Sleep Mode Transition Function

5.9.3 Software Standby Mode Transition Sample Function

The flow chart for Software Standby Mode Transition Sample Function is shown in Figure 5.8 and Figure 5.9.

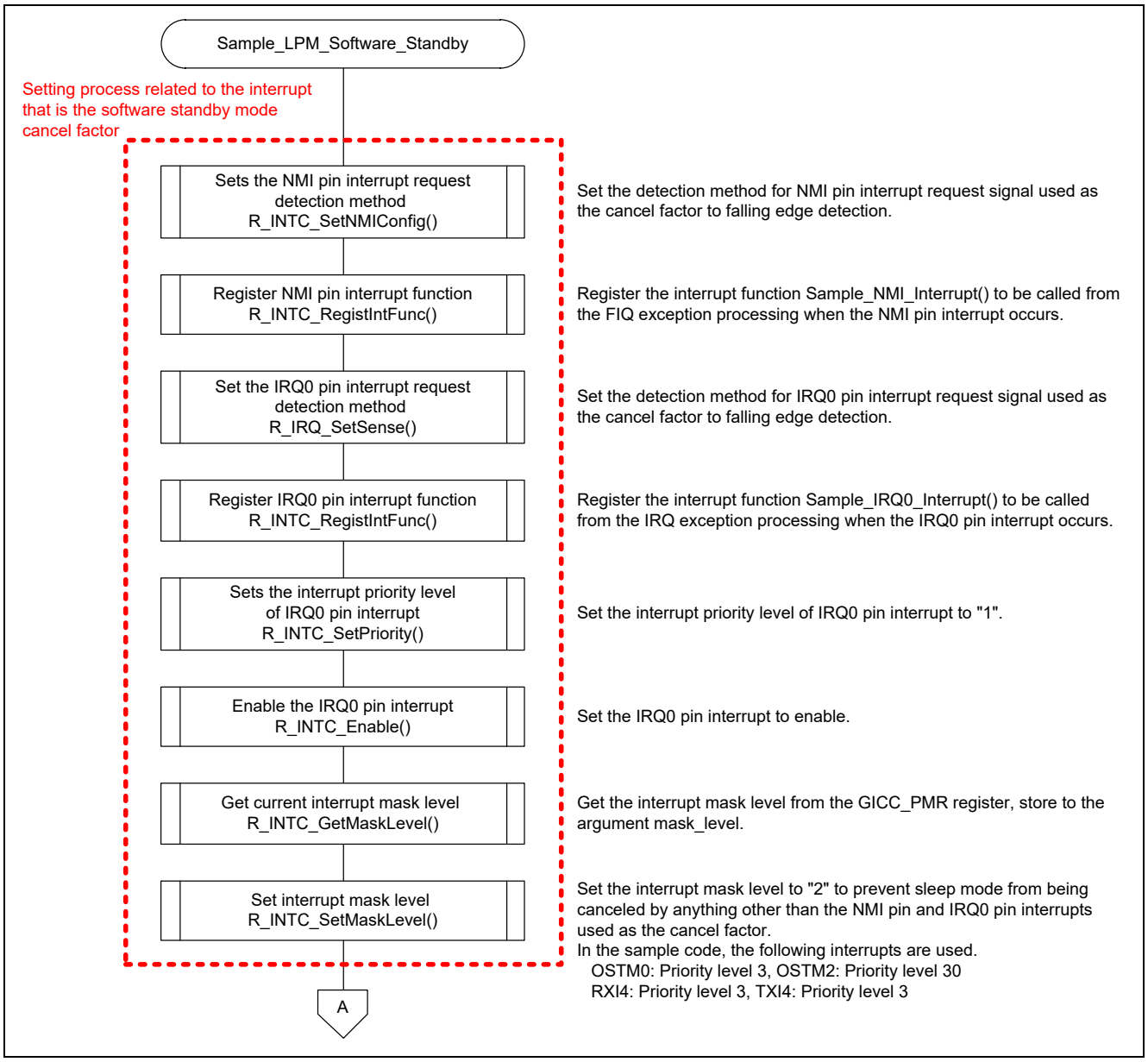


Figure 5.8 Software Standby Mode Transition Sample Function (1/2)

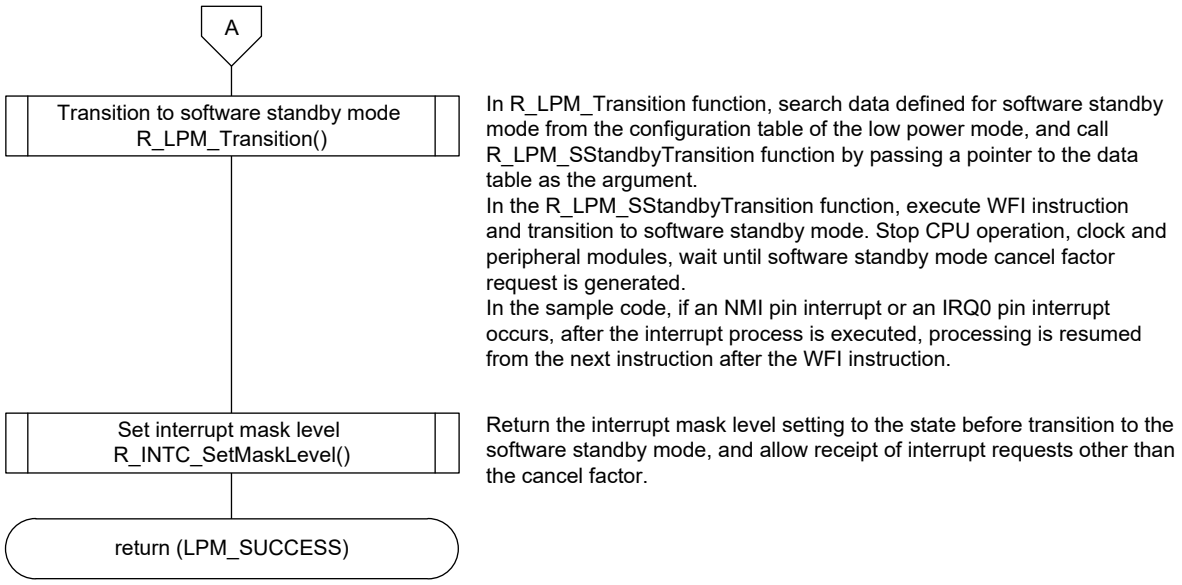
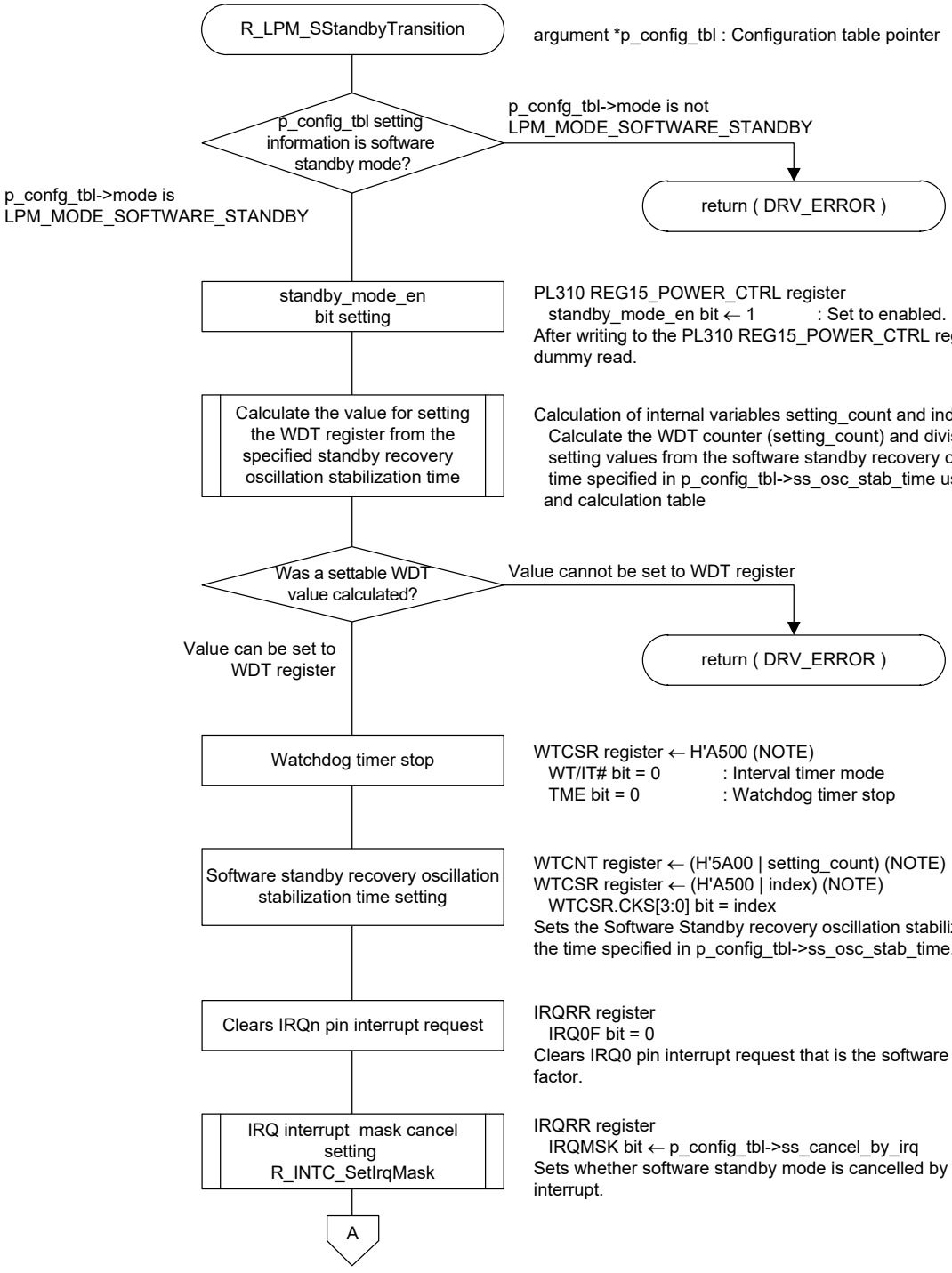


Figure 5.9 Software Standby Mode Transition Sample Function (2/2)

5.9.4 Software Standby Mode Transition Function

The software standby mode transition function flow chart is shown in Figure 5.10 and Figure 5.11.



Note: When writing to WTSCR or when writing to WTCNT, it is necessary to write in 16 bit units, with the upper byte in the cases set to H'A5 or H'5A respectively.

Figure 5.10 Software Standby Mode Transition Function (1/2)

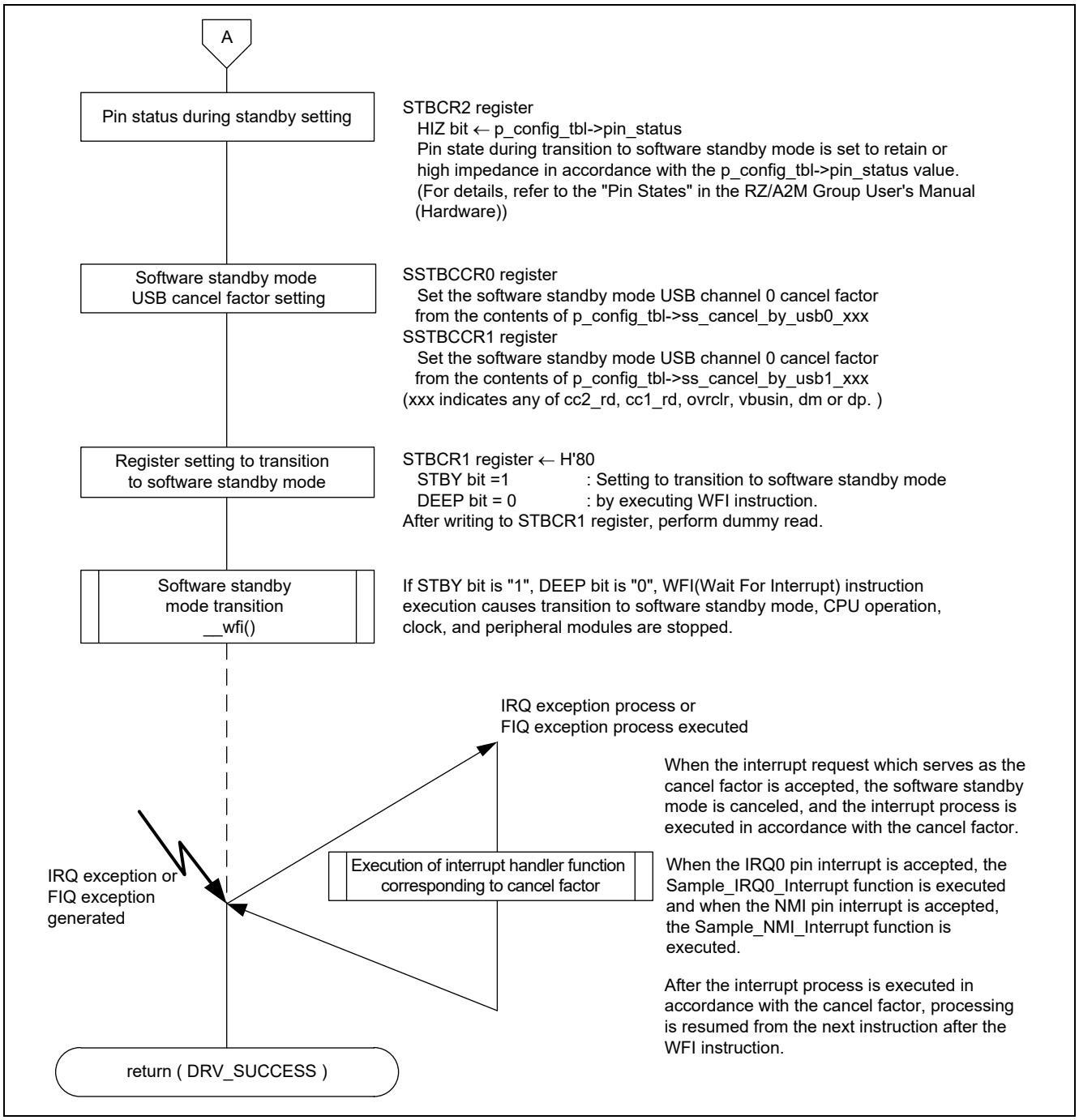


Figure 5.11 Software Standby Mode Transition Function (2/2)

5.9.5 Deep Standby Mode Transition Sample Function

The flow chart for Deep Standby Mode Transition Sample Function is shown in Figure 5.12.

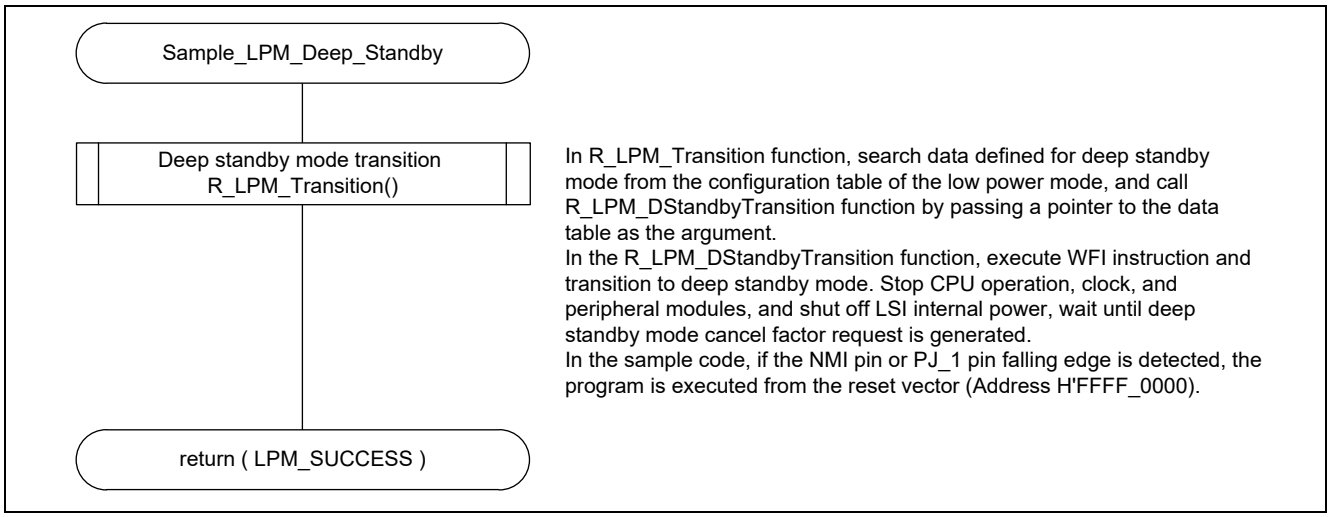


Figure 5.12 Deep Standby Mode Transition Sample Function

5.9.6 Deep Standby Mode Transition Function

The deep standby mode transition function flow chart is shown in Figure 5.13 and Figure 5.14.

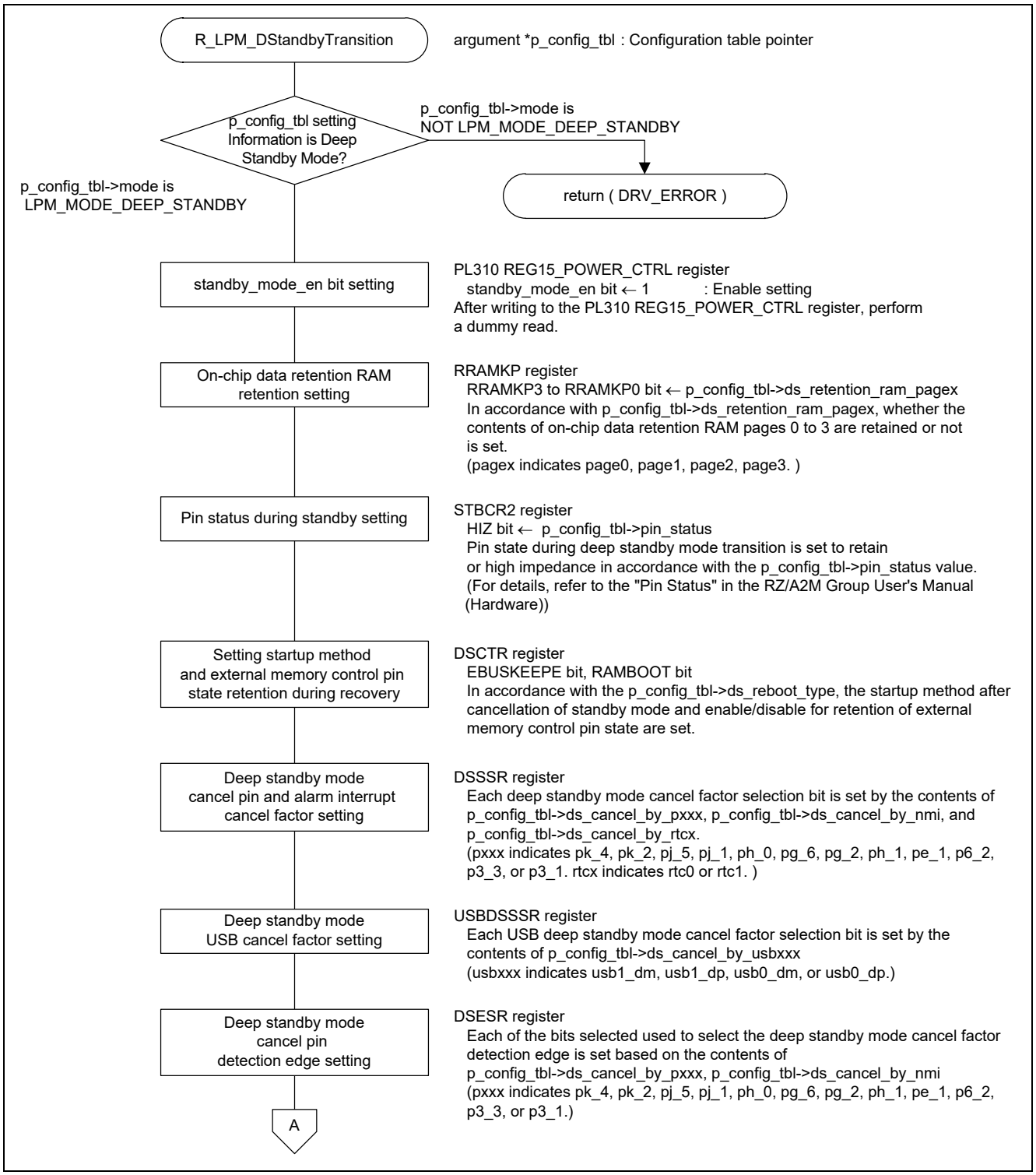


Figure 5.13 Deep Standby Mode Transition Function (1/2)

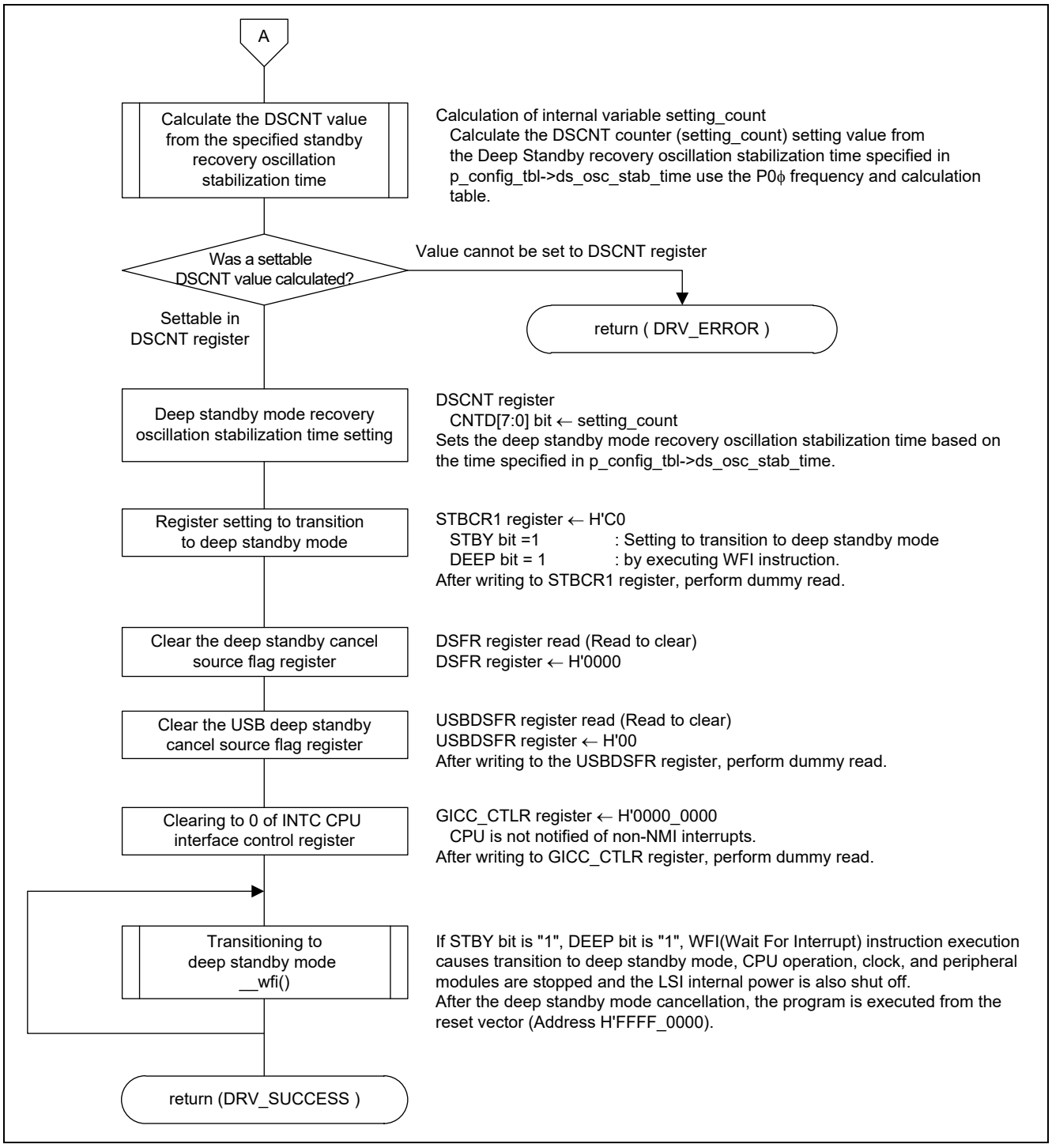


Figure 5.14 Deep Standby Mode Transition Function (2/2)

5.9.7 Process to Start from On-chip data retention RAM after Deep Standby Mode is Canceled

The flow chart for Process to Start from On-chip Data Retention RAM after Deep Standby Mode is Canceled is shown in Figure 5.15.

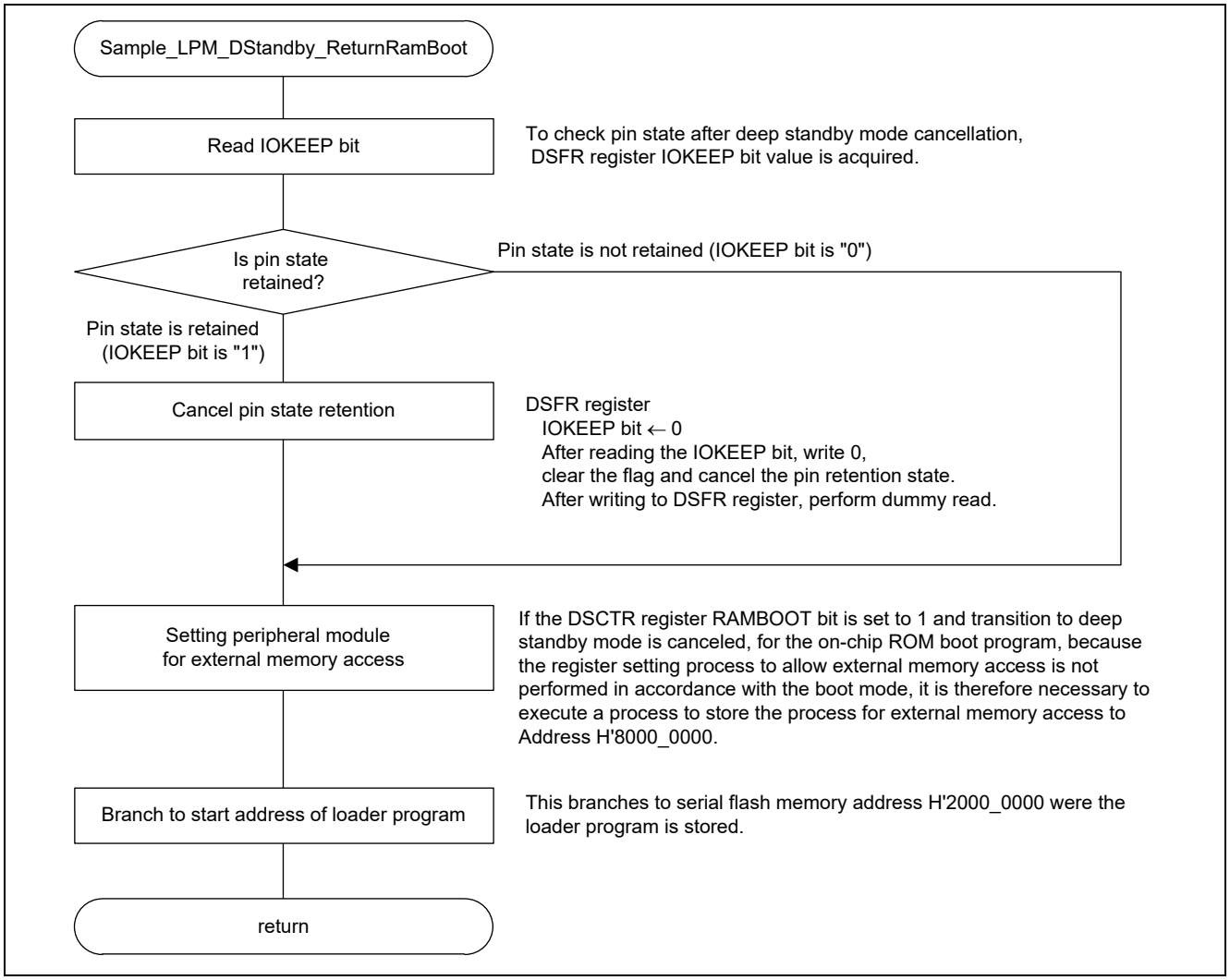


Figure 5.15 Process to Start from On-chip Data Retention RAM after Deep Standby Mode is Canceled

5.9.8 NMI Pin Interrupt Sample Function

The flow chart for NMI Pin Interrupt Sample Function is shown in Figure 5.16.

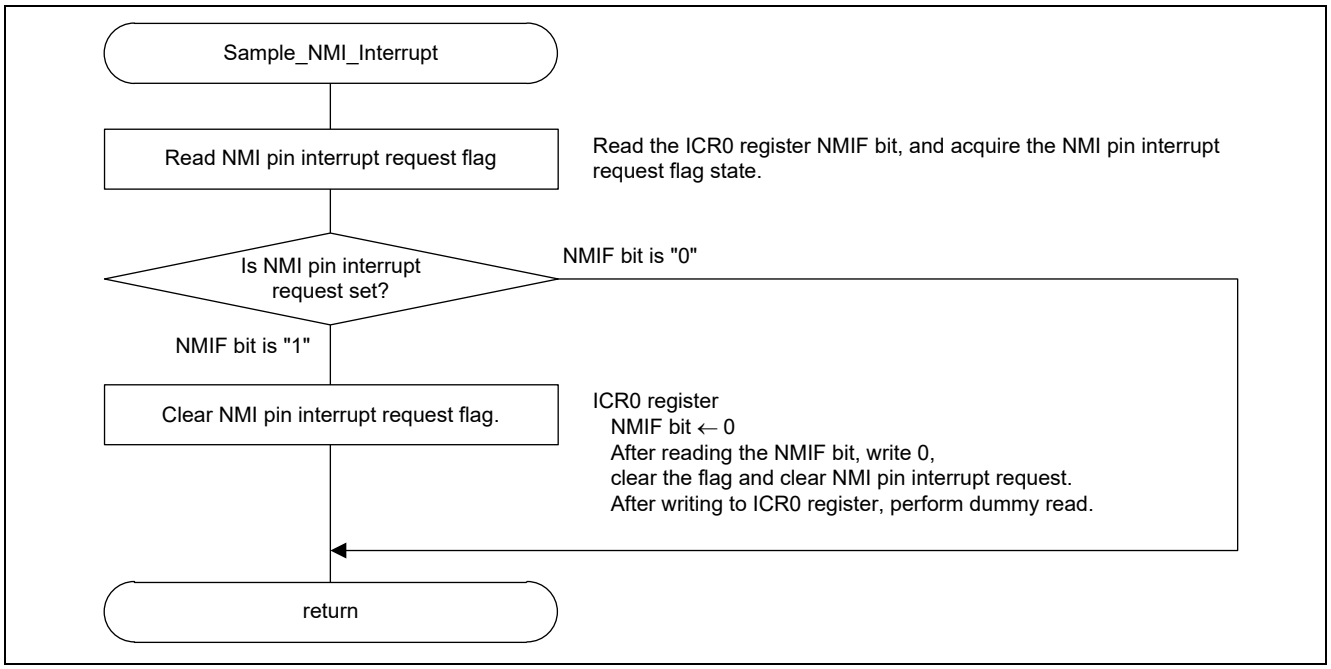


Figure 5.16 NMI Pin Interrupt Sample Function

5.9.9 IRQ0 Pin Interrupt Sample Function

The flow chart for IRQ0 Pin Interrupt Sample Function is shown in Figure 5.17.

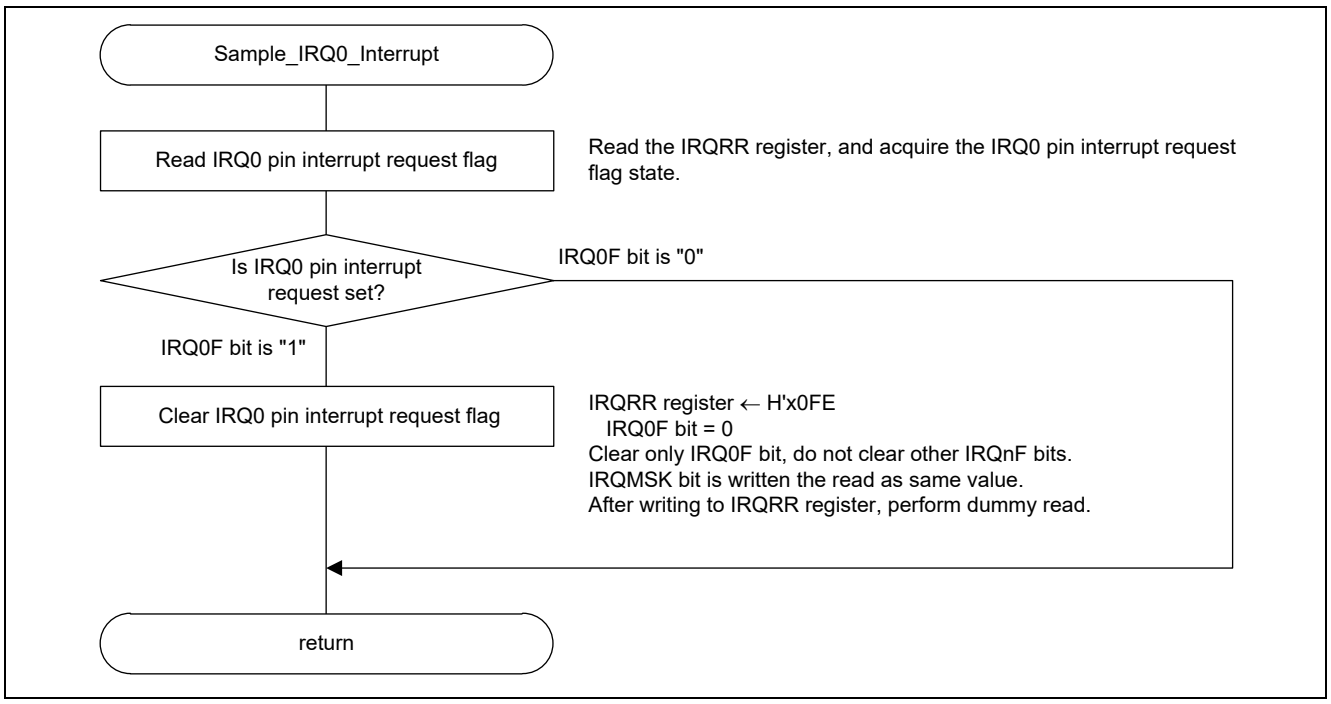


Figure 5.17 IRQ0 Pin Interrupt Sample Function

5.9.10 Low Power Mode Transition Function

The flow chart for Low Power Mode Transition Function is shown in Figure 5.18

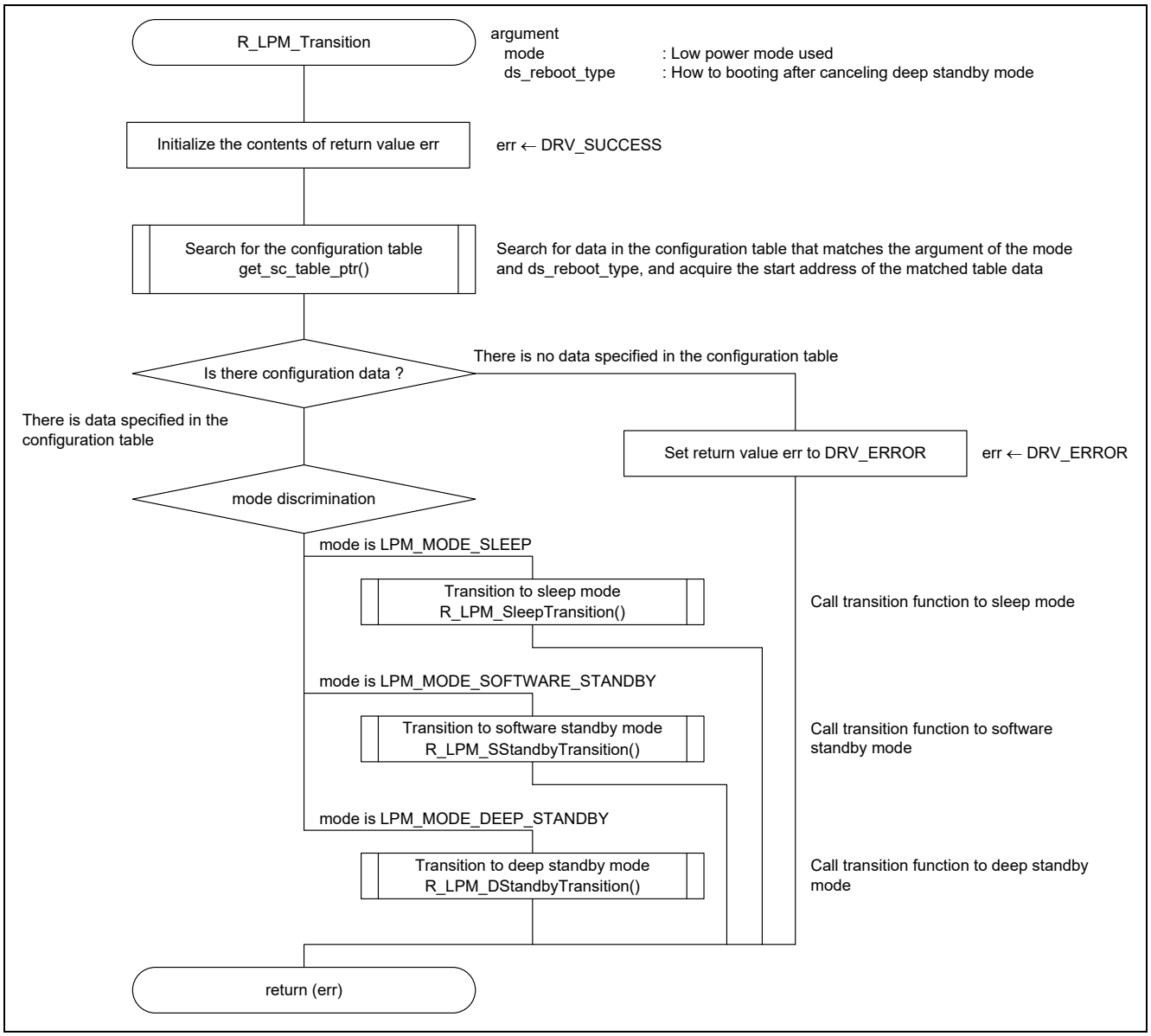


Figure 5.18 Low Power Mode Transition Function

5.10 Starting Up Sample Code

In the sample code, when a command is entered in the terminal, the RZ/A2M will make transition to a power-down mode. Table 5.26 shows the Sample Command Operation List.

Table 5.26 Sample Command Operation List

Command		Command function name	Description
First parameter	Second parameter		
LPM_SM	—	sapmle_lpm_ cmd_sleep_mode	Transition to sleep mode. After SW2 or SW3 press is detected, sleep mode is canceled.
LPM_SS	—	sapmle_lpm_ cmd_software_standby	Transition to software standby mode. After SW2 or SW3 press is detected, software standby mode is canceled.
LPM_DS	1	sapmle_lpm_ cmd_deep_standby	Transition to deep standby mode. When SW2 or SW3 press is detected, and after deep standby mode is canceled, the on-chip ROM boot program is executed, then the loader program is executed (external memory startup mode).
LPM_DS	2	sapmle_lpm_ cmd_deep_standby	Transition to deep standby mode. When SW2 or SW3 press is detected, and after deep standby mode is canceled, the on-chip ROM boot program is executed, then the program stored in on-chip data retention RAM is executed (on-chip data retention RAM startup mode).

After turning on the board power, the message shown in Figure 5.19 (1) is output. Ver.x.xx and Copyright (C) yyyy shown in Figure 5.19 indicate the main processing version of sample code and the issued year of the application note respectively. After the "SAMPLE>" prompt is output, it is possible to input commands.

Displayed Message
(1) RZ/A2M Evaluation Board Osless Low Power Mode Sample Program. Ver.x.xx Copyright (C) yyyy Renesas Electronics Corporation. All rights reserved. SAMPLE>

Figure 5.19 Example Terminal Display During Sample Code Startup

5.10.1 Sleep Mode Transition Command Operation

In the sample code, after the sample code starts, if "LPM_SM"+"Enter key" is input from the terminal software, the sleep mode transition command will operate. If the sleep mode transition command is operated, message shown in Figure 5.20 (1) is output. After "Please Press Enter Key" is output, when the "Enter key" is input, it will make transition to sleep mode. After transition to sleep mode, it will wait until the cancel factor is input.

If the switches that serve as the sleep mode cancel factor (SW2 or SW3) are pressed, the sleep mode is canceled, and message shown in Figure 5.20 (2) will be output.

Displayed Message
(1) SAMPLE>LPM_SM Transition to Sleep Mode Press the enter key on the terminal to enter sleep mode Sleep mode can be cancelled by pressing SW3 or SW2 on the board. (SW3 : Canceled by IRQ0, SW2: Canceled by NMI) Please Press Enter Key
(2) ********** Return from Sleep Mode ********** SAMPLE>

Figure 5.20 Example Terminal Display when Sleep Mode Transition Command is Executed

5.10.2 Software Standby Mode Transition Command Operation

In the sample code, after the sample code starts, if "LPM_SS"+"Enter key" is input from the terminal software, the software standby mode transition command will operate. If the software standby mode transition command is operated, message shown in Figure 5.21 (1) is output. After "Please Press Enter Key" is output, when the "Enter key" is input, it will make transition to software standby mode. After transition to software standby mode, it will wait until the cancel factor is input.

If the switches that serve as the software standby mode cancel factor (SW2 or SW3) are pressed, the software standby mode is canceled, message shown in Figure 5.21 (2) will be output.

Displayed Message
(1) SAMPLE> LPM_SS Transition to Software Standby Mode Press the enter key on the terminal to enter sleep mode Software stanby mode can be cancelled by pressing SW3 or SW2 on the board. (SW3 : Canceled by IRQ0, SW2 : Canceled by NMI) Please Press Enter Key
(2) ** Return from Software Standby Mode **   SAMPLE>

Figure 5.21 Example Terminal Display When Software Standby Mode Transition Command Is Executed

5.10.3 Deep Standby Mode Transition Command Operation

In the sample code, after the sample code starts, if "LPM_DS"+"1"+"Enter key" or "LPM_DS"+"2"+"Enter key" are input from the terminal software, the deep standby mode transition command will operate.

(1) External Memory Startup Mode

If "LPM_DS"+"1"+"Enter key" are entered, message shown in Figure 5.22 (1) will be output. After "Please Press Enter Key" is output, when the "Enter key" is input, it will make transition to deep standby mode. After transition to deep standby mode, it will wait until the cancel factor is input.

If the switches that serve as the deep standby mode cancel factor (SW2 or SW3) are pressed, the deep standby mode is canceled, the RZ/A2M will start in external memory startup mode, and message shown in Figure 5.22 (2) will be output.

Displayed Message

(1)

```
SAMPLE>LPM_DS 1
      Transition to Deep Standby Mode(EBUSKEEPE=0, RAMBOOT=0)

      Press the enter key on the terminal to enter deep standby mode

      Deep standby mode can be cancelled by pressing SW3 or SW2 on the board.
      (SW3 : Canceled by PJ_1, SW2 : Canceled by NMI)

      Please Press Enter Key
```

(2)

```
********************************************************************************
      Return from Deep Standby Mode
********************************************************************************


RZ/A2M Evaluation Board Osless Low Power Mode Sample Program. Ver.x.xx
Copyright (C) yyyy Renesas Electronics Corporation. All rights reserved.


SAMPLE>
```

Figure 5.22 Example Terminal Display when Deep Standby Mode Transition Command Is Executed (External Memory Startup Mode)

(2) On-chip Data Retention RAM Startup Mode

If "LPM_DS"+"2"+"Enter key" are entered, message shown in Figure 5.23 (1) will be output. After "Please Press Enter Key" is output, when the "Enter key" is input, it will make transition to deep standby mode. After transition to deep standby mode, it will wait until the cancel factor is input.

If the switches that serve as the deep standby mode cancel factor (SW2 or SW3) are pressed, the deep standby mode is canceled, the RZ/A2M will start in on-chip data retention RAM startup mode, and message shown in Figure 5.23 (2) will be output.

Displayed Message

(1)

```
SAMPLE>LPM_DS 2
      Transition to Deep Standby Mode(EBUSKEEPE=0, RAMBOOT=1)

      Press the enter key on the terminal to enter deep standby mode

      Deep standby mode can be cancelled by pressing SW3 or SW2 on the board.
      (SW3 : Canceled by PJ_1, SW2 : Canceled by NMI)

      Please Press Enter Key
```

(2)

```
********************************************************************************
      Return from Deep Standby Mode
********************************************************************************


RZ/A2M Evaluation Board Osless Low Power Mode Sample Program. Ver.x.xx
Copyright (C) yyyy Renesas Electronics Corporation. All rights reserved.


SAMPLE>
```

Figure 5.23 Example Terminal Display when Deep Standby Mode Transition Command is Executed (On-chip Data Retention RAM Startup Mode)

5.10.4 Help Command Operation

In the sample code, after the sample code starts, if "Help"+"Enter key" is input from the terminal software, the contents of the command supported by the sample code is displayed as shown in (1) of Figure 5.24.

Displayed Message

```
                                                                    (1)
SAMPLE> HELP

[LPM_SM command]
    Transition to Sleep Mode
command format:
    LPM_SM
argument:
    None

[LPM_SS command]
    Transition to Software Standby Mode
command format:
    LPM_SS
argument:
    None

[LPM_DS command]
    Transition to Deep Standby Mode
command format:
    LPM_DS 'value'
argument:
    'value' - 1 or 2
              (1:External Memory Startup Mode,
               2:On-chip Data Retention RAM Startup Mode)
```

Figure 5.24 Example Terminal Display when Help Command is Executed

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

User's Manual: Hardware

RZ/A2M Group User's Manual: Hardware
(The latest version can be downloaded from the Renesas Electronics website.)

RTK7921053C00000BE (RZ/A2M CPU board) User's Manual
(The latest version can be downloaded from the Renesas Electronics website.)

RTK79210XXB00000BE (RZ/A2M SUB board) User's Manual
(The latest version can be downloaded from the Renesas Electronics website.)

Arm Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C
(The latest version can be downloaded from the Arm website.)

Arm Cortex™-A9 Technical Reference Manual Revision: r4p1
(The latest version can be downloaded from the Arm website.)

Arm Generic Interrupt Controller Architecture Specification - Architecture version2.0
(The latest version can be downloaded from the Arm website.)

Arm CoreLink™ Level 2 Cache Controller L2C-310 Technical Reference Manual Revision: r3p3
(The latest version can be downloaded from the Arm website.)

Technical Update/Technical News

(The latest information can be downloaded from the Renesas Electronics website.)

User's Manual: Integrated development environment

The e2 studio Integrated Development Environment user's manual can be downloaded from the Renesas Electronics website.
(The latest version can be downloaded from the Renesas Electronics website.)

Revision History

Rev.	Date Issued	Revision Details	
		Page	Summary
Rev.1.00	Apr.15.19	—	First edition issued
Rev.1.10	May.17.19	P7	Table 2.1 Operation Confirmation Conditions (1/2) Remove compiler option "-mthumb-interwork"
Rev.1.11	Aug.28.19	P44 P47 P64	Added R_LPM_Transition function in API function • Added the function in "Table 5.25 API Functions List" • Added the function specification in "5.8.2 API Function Specifications" • Added the function flow in "Figure 5.18 Low Power Mode Transition Function"
		P48 P48 P49	Added precautions when calling the following API functions • R_LPM_SleepTransition function • R_LPM_SStandbyTransition function • R_LPM_DStandbyTransition function
		P45 P45 P45 P53 P56 P59	Changed the function called by the sample function for transition to each low power mode to R_LPM_Transition function • Modified the description of Sample_LPM_Sleep_Mode function • Modified the description of Sample_LPM_Software_Standby function • Modified the description of Sample_LPM_Deep_Standby function • Figure 5.6 Sleep Mode Transition Sample Function (2/2) Changed R_LPM_SleepTransition function to R_LPM_Transition function • Figure 5.9 Software Standby Mode Transition Sample Function (2/2) Changed R_LPM_SStandbyTransition function to R_LPM_Transition function • Figure 5.12 Deep Standby Mode Transition Sample Function Changed R_LPM_DStandbyTransition function to R_LPM_Transition function
		P66 to P70	Changed the output message when executing command in the sample code • 5.10.1 Sleep Mode Transition Command Operation • 5.10.2 Software Standby Mode Transition Command Operation • 5.10.3 Deep Standby Mode Transition Command Operation • 5.10.4 Help Command Operation
Rev.1.12	Nov.13.19	P33 P34	Changed the following figure and table because an input section for processing of the R_SC_HardwareSetup function was added. • Table 5.13 List of Sections and Objects Used in the Sample Code • Figure 5.4 Section Allocation

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Notice

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

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

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

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

Trademarks

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

Contact information

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