RE01 256KB Group Low Power Mode Transition Example

RE01 256KB Group CMSIS Driver Package Low Power Mode Sample Code

Introduction

This application note explains about the sample code that uses the RE01 256KB Group CMSIS software package. For the sample code, refer to the projects included.

The summary of this sample code is shown below.

Table 1 The summary of this sample code

Sample code operation	Peripheral functions	Drivers used in the sample code
Transition to software standby (SSTBY) mode and wake-up from SSTBY mode. In SSTBY mode, the lowest power consumption in each power supply mode is realized.	Power-saving functions	R_SYSTEM R_LPM

Although sample code projects for each power supply mode are included, this application note only explains the sample code for MINPWON mode, unless otherwise noted. If you use the sample code for EXFPWON mode, please read “MINPWON” as “EXFPWON”.

Target Device

RE01 256KB Group

Caution

If you apply this application note to another device, it is necessary to modify the software according to the specification of the MCU you use and evaluate it adequately.

Contents

1. Specification

1.1 Project Description

This application note includes two sample code projects shown below.

- r01an5337_re_hal_lpm_minpwon.zip : Transition and return for MINPWON SSTBY mode
- r01an5337_re_hal_lpm_exfpwon.zip : Transition and return for EXFPWON SSTBY mode

These sample code projects are confirmed to operate on the Evaluation Kit RE01 256KB. The setting of this project is adjusted to R7F0E01182CFP implemented on Evaluation Kit RE01 256KB. Modify the device setting in the project when you use other device.

1.2 List of Used Ports

Table 1-1 lists the used ports and functions in the sample code.

Table 1-1 Used ports in the sample code

Used ports	Functions
P210	LED0 : Turn on in operation mode after reset release Turn off in standby mode
P508	SW2 : IRQ4 interrupt

1.3 File Configuration

Table 1-2 lists the files added or changed in the sample code.

Table 1-2 Files added or changed in the sample code

Name	Outline	Remarks
main.c	Main processing	-
main_cfg.h	Constants used in sample code	Added file
nop.c	NOP processing	Added file
nop.h	Definition of NOP processing function	Added file
r_core_cfg.h	The setting of Main-clock oscillator (MOSC)	Low consumption oscillation function : Enable
	The setting of Middle-speed on-chip oscillator (MOCO)	Stop (Disable)
	The setting of Low-speed on-chip oscillator (LOCO)	Stop (Disable)
	The setting of Sub-clock oscillator (SOSC)	Drive capability : Low CL Noise filter : Disable
	The setting of System Clock Source	Select the Main-clock oscillator
r_system_cfg.h Note	The IRQ event link numbers used in this sample code are set in this file	IRQ event channel of IRQ4 : 4
pin.c	The pin setting used in this sample code	P508 : IRQ4

Note: For interrupt settings in r_system_cfg.h, see the section “6.3 Interrupt Control” in RE01 1500KB, 256KB CMSIS Package Startup Guide.

1.4 Option-Setting Memory

Table 1-3 lists the Option-setting memory settings of the sample code. Set the optimal value for the system as necessary.

Table 1-3 Option-setting memory settings of the sample code

Symbol	Address	Setting Value	
AWS	0100A164h to 0100A167h	FFFF FFFFh	No access window setting
OSIS	0100A150h to 0100A15Fh	FFFF FFFFh	No protection of ID code (all bytes are FFh)
SECMPUxxx	00000408h to 0000043Bh	FFFF FFFFh	Security MPU disabled
OFS1	00000404h to 00000407h	FFFF FFFFh	LVD0 reset invalid after reset release HOCO oscillation stop after reset release
OFS0	00000400h to 00000403h	FFFF FFFFh	IWDT stop WDT stop after reset release

2. Conditions for Confirming Operation

Sample code is confirmed to operate under the conditions shown in Table 2-1.

Table 2-1 Confirmed operating conditions

Item		**Conditions**
MCU		R7F0E01182CFP 100pin
Operating frequency	After reset release (Initial value)	• Main-clock oscillator (MOSC) : 32MHz • System clock (ICLK) : 32MHz (MOSC * 1/1) • Peripheral module clock A (PCLKA) : 32MHz (MOSC * 1/1) • Peripheral module clock B (PCLKB) : 32MHz (MOSC * 1/1)
Operating voltage		3.3V
Target board		Evaluation Kit RE01 256KB (RTK70E0118CXXXXXBJ)
Integrated Development Environment	GCC	Renesas e^2 studio 2020-07
	IAR	IAR Embedded Workbench® for Arm® Version 8.40
C Compiler	GCC	GCC Arm® Embedded Version 6.3.1.20170620 GNU 6-2017-q2-update
	IAR	IAR C/C++ Compiler for Arm® Version 8.40
Debugger		SEGGER J-Link OB
CMSIS Driver Package Version		Rev1.00
Sample code Version		Rev1.00

Note: The current consumption can be measured by modifying the target board as follows.
J9 : Open, Between T3-T6 : Connect to ammeter.
For details, refer to the RE01 Group Evaluation Kit RE01 256KB (EK-RE01 256KB) User's manual.

3. Software Description

The sample code operates as follows by using the R_SYSTEM and R_LPM driver functions.

1. After reset release, set the system clock to MOSC 32MHz (state-1), and the LED flashes three times.
2. The sample code then turns off the LED Note and changes mode to SSTBY mode (state-2).
3. When an IRQ4 interrupt occurs, caused by pressing SW2 during SSTBY mode, return to the operation mode before transition (state-1) and turns on the LED Note.
4. Repeat these transitions by pressing SW2.

Note: By the main_cfg.h file settings, the control of the LED can be disabled and so removes the current flowing to LED.

The operation conditions of the sample code is shown in Table 3-1.

Furthermore, transition and return for MINPWON SSTBY mode is shown in Figure 3-1 and transition and return for EXFPWON SSTBY mode is shown in Figure 3-2. Figure 3-1 shows the transition operation of sample code project “r01an5337_re_hal_lpm_minpwon.zip” and Figure 3-2 shows the transition operation of sample code project “r01an5337_re_hal_lpm_exfpwon.zip”.

Table 3-1 The operation conditions of the sample code

Item	**state-1** **(In the operation mode before transition to SSTBY)**	**state-2** **(In SSTBY mode)**
Oscillation Clock	MOSC SOSC Note	Stopped SOSC Note
System clock (ICLK)	MOSC 32MHz	Stopped
Power supply mode	ALLPWON mode	MINPWON mode
Power control mode	NORMAL mode	VBB mode
Low power mode	- (OPE mode)	SSTBY mode

Note: MOSC low consumption oscillation function is set to enable, SOSC noise filter is set to disabled and SOSC drive capability is set to Low CL for reducing power consumption.

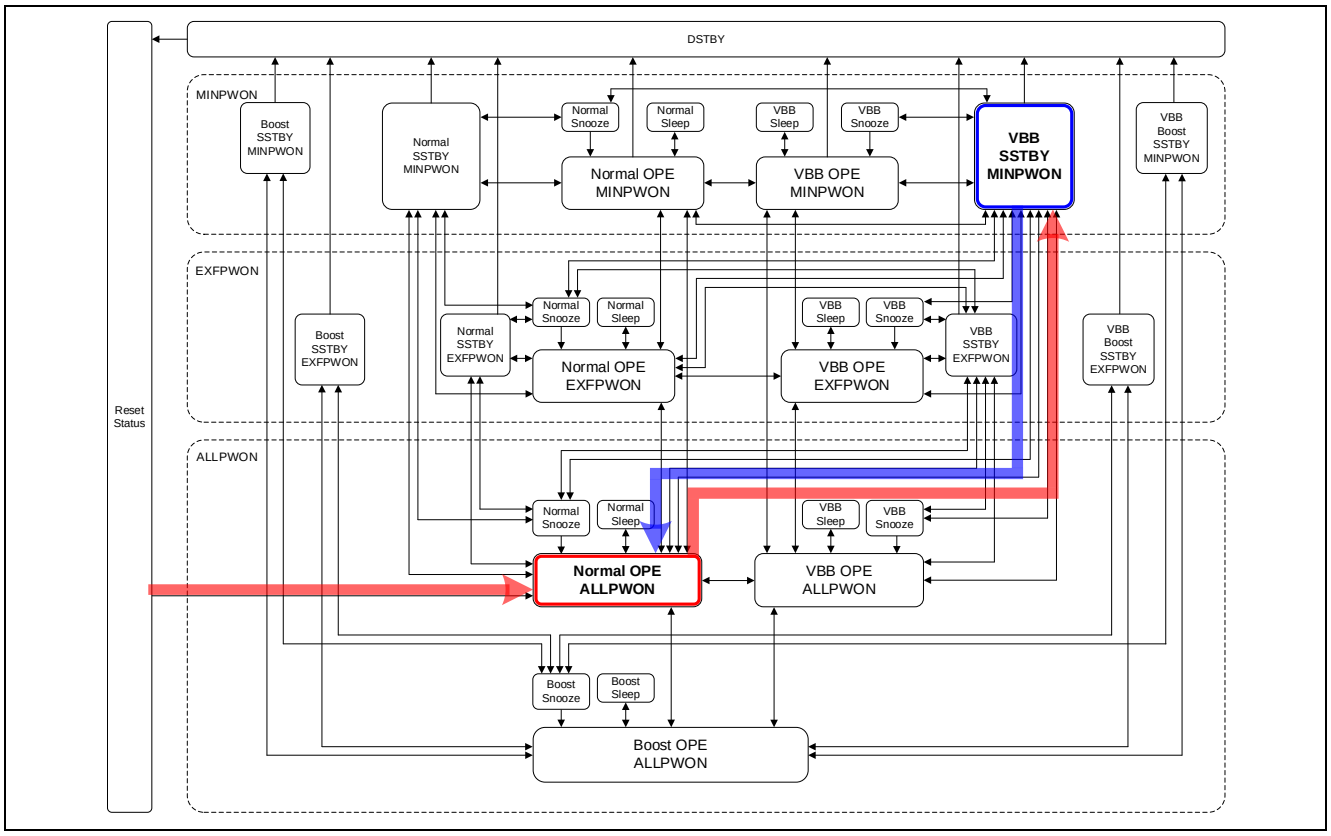


Figure 3-1 Transition and return for MINPWON SSTBY mode

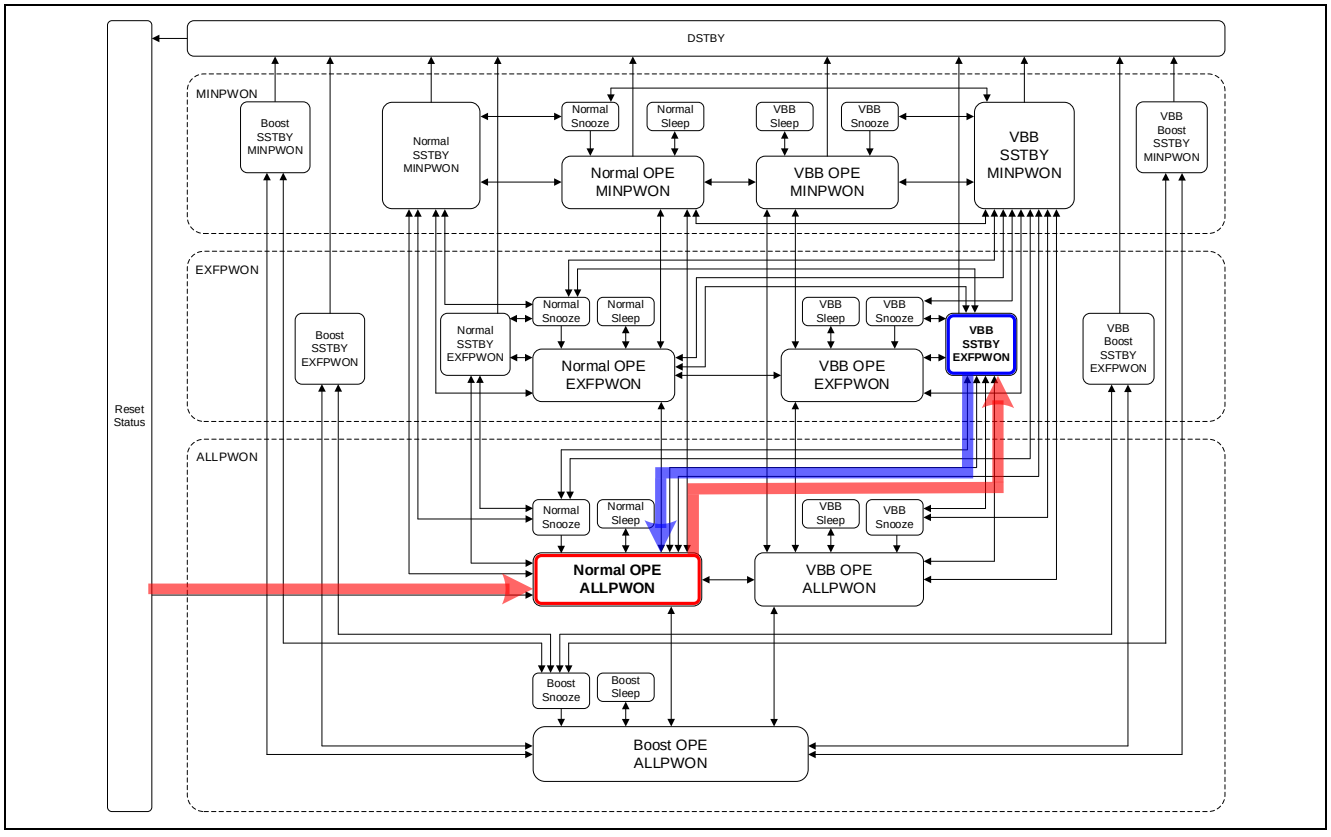


Figure 3-2 Transition and return for EXFPWON SSTBY mode

3.1 List of Functions

The specifications for functions added in this sample code are shown below.

main	
Outline	Main processing
Header	None
Declaration	int main(void)
Description	Stops the modules not used after the release of reset. Repeat transition between SSTBY mode and the operation mode after reset release, triggered by pressing SW2.
Argument	None
Return value	0

demo_error	
Outline	Error processing
Header	None
Declaration	static void demo_error(void)
Description	Execute endless loop in case that return value of function is error.
Argument	None
Return value	None

irq4_wait	
Outline	Wait for IRQ4
Header	None
Declaration	static void irq4_wait(void)
Description	Execute endless loop until IRQ4 interrupt occurs.
Argument	None
Return value	None

led_port_init	
Outline	Initial setting for LED port
Header	None
Declaration	static void led_port_init(void)
Description	Set the initial setting for port using as LED, and flash LED three times.
Argument	None
Return value	None

Note: The enable or disable of LED controlling can be selected in main_cfg.h file.

irq4_setup	
Outline	Setting for IRQ4 interrupt
Header	None
Declaration	static void irq4_setup(void)
Description	Set the IRQ4 interrupt
Argument	None
Return value	None

led_set		
Outline	Controlling LED	
Header	None	
Declaration	static void led_set(uint8_t flag)	
Description	Turn on or turn off the LED	
Argument	flag	0: Turn on 1: Turn off
Return value	None	

Note: The enable or disable of LED controlling can be selected in main_cfg.h file.

irq4_int	
Outline	Interrupt handling of IRQ4
Header	None
Declaration	static void irq4_int(void)
Description	Execute the interrupt handler by IRQ4 interrupt
Argument	None
Return value	None

Note: This function is allocated in RAM.

BoardInit	
Outline	Terminal settings on the using board
Header	None
Declaration	void BoardInit(void)
Description	This function is called by SystemInit() function after reset release
Argument	None
Return value	None

3.2 List of Constants

The modifiable constant is shown in Table 3-2.

Table 3-2 The modifiable constant

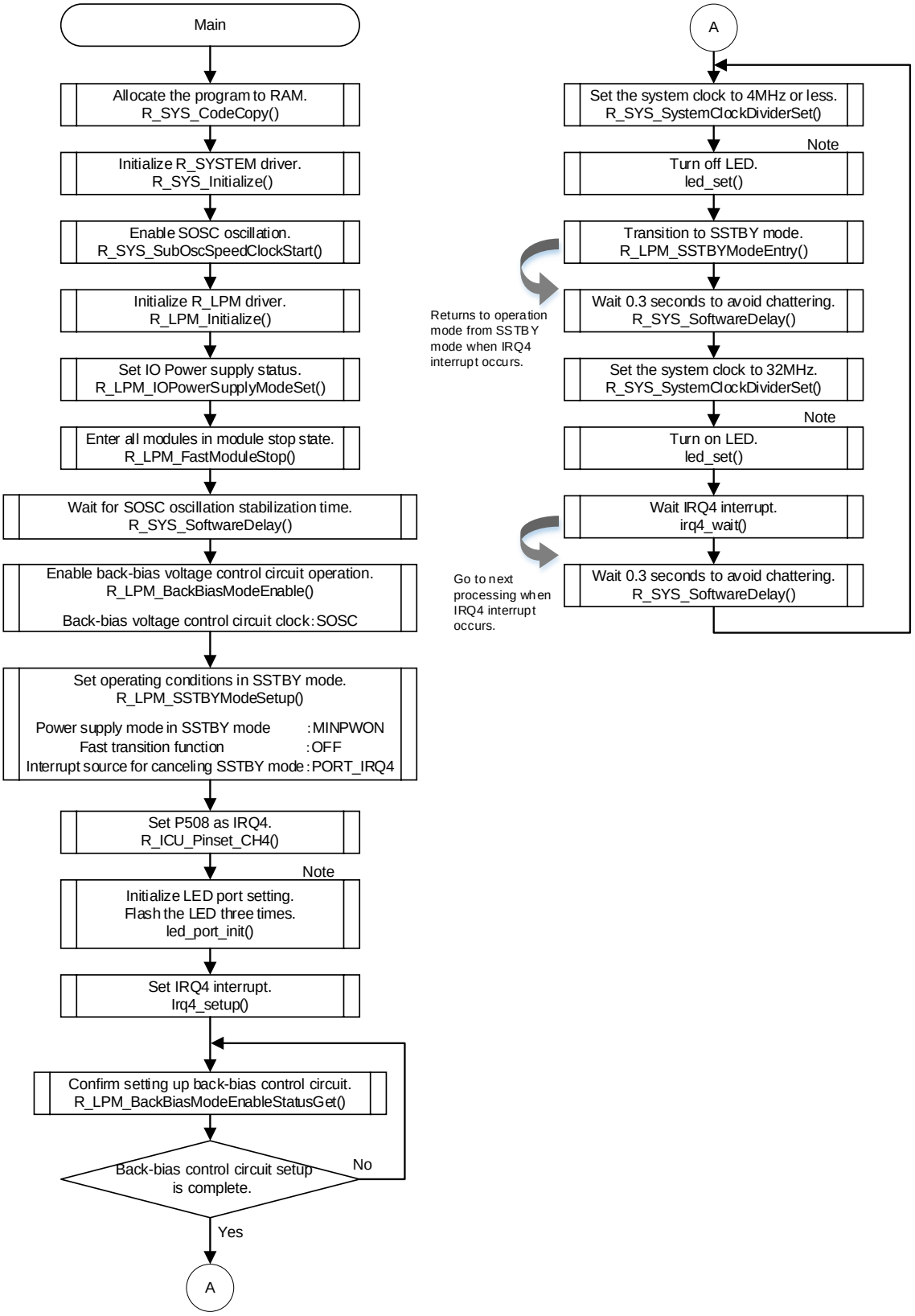
Contents	Initial value	Setting	File
MAIN_CFG_LED_ENABLE	1	0: Disable LED control Note 1: Enable LED control	main_cfg.h

Note: This setting can reduce the current flowing to LED.

3.3 Flowcharts

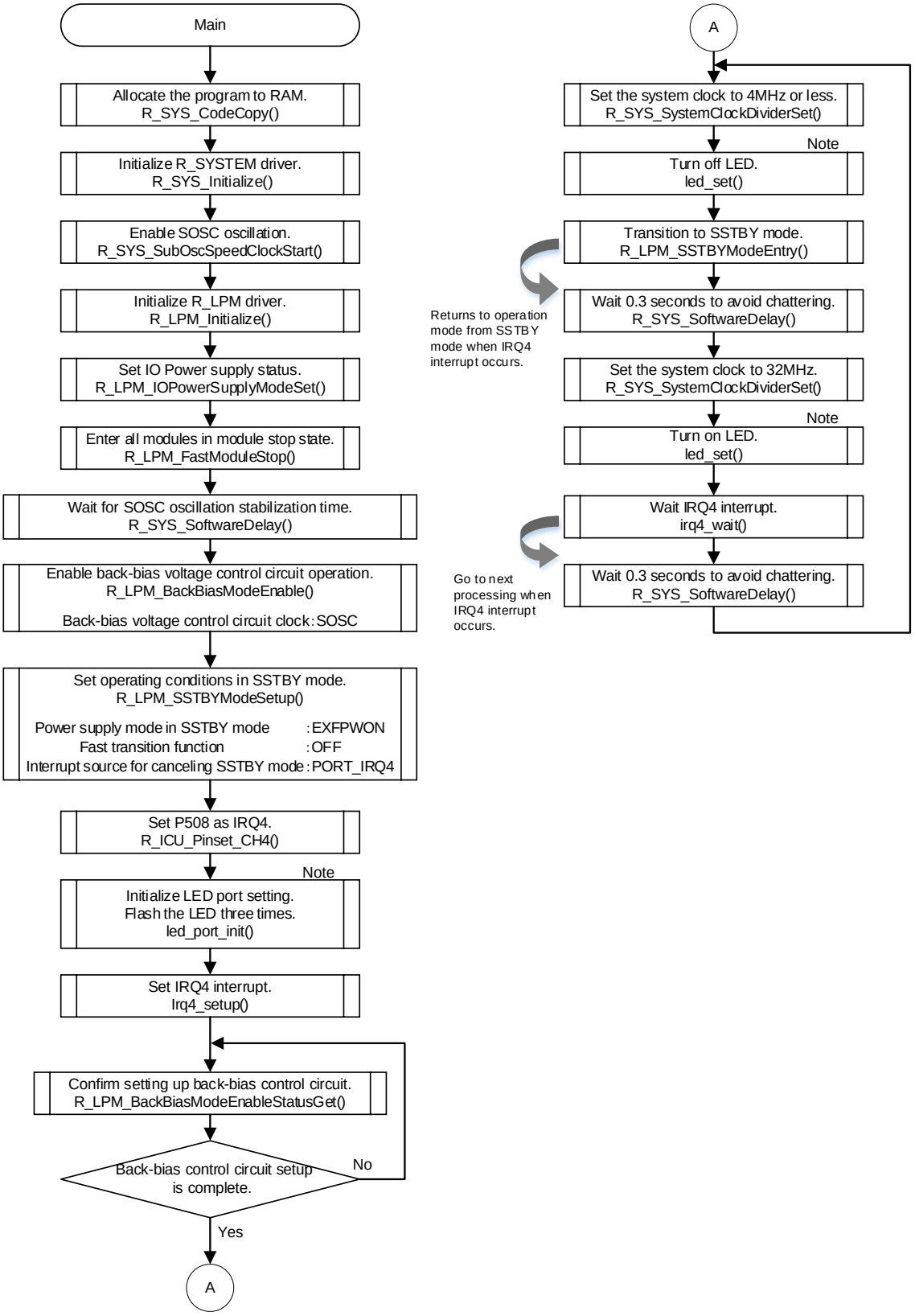
The main processing of the transition to and from in MINPWON SSTBY mode is shown in Figure 3-3.

The main processing of the transition to and from in EXFPWON SSTBY mode is shown in Figure 3-4.



Note: The enabling or disabling of LED can be selected in main_cfg.h file.

Figure 3-3 Main processing of the transition and return for MINPWON SSTBY mode



Note: The enabling or disabling of LED can be selected in main_cfg.h file.

Figure 3-4 Main processing of the transition and return for EXFPWON SSTBY mode

4. API Specification of Drivers

4.1 External Specification

This driver contains documents that describes the external API specification. These files are contained in the Driver Specification folder within the Documents.

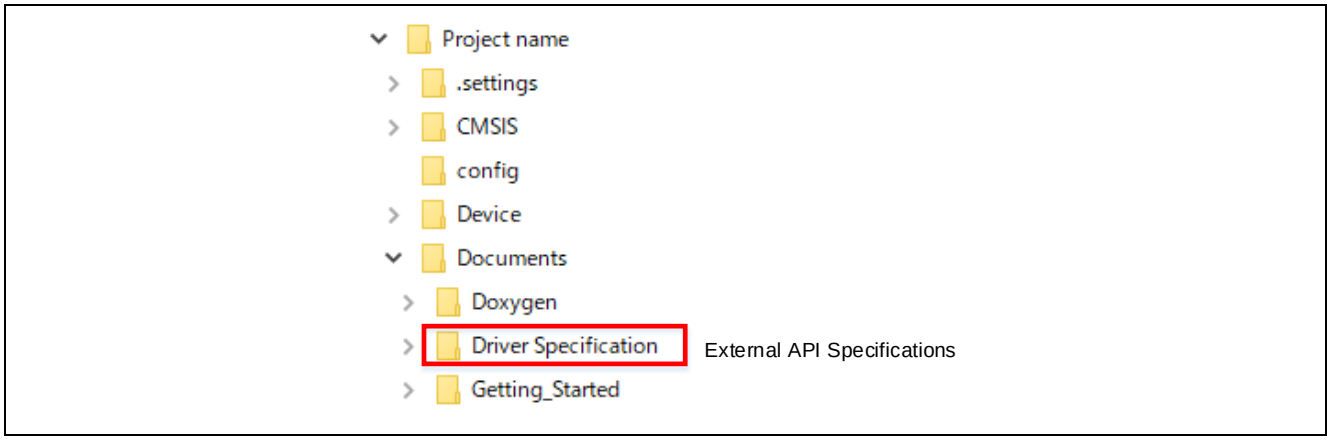


Figure 4-1 Location of External API Specifications

5. Usage Note on R_SYSTEM Driver

This section explains the main points about R_SYSTEM driver. For other notes, refer to the external specification introduced in section 4.1.

5.1 Clock Initialization

When you configure the clock settings by using R_SYSTEM driver, clock initialization is required in r_core_cfg.h file.

(In r_core_cfg.h of this sample code, MOSC low consumption oscillation function is set to enable, MOCO, LOCO and SOSC noise filter are set to stop and SOSC oscillation drive capability is set to Low CL.)

5.2 Changing the System Clock Source

When you change the system clock source, do not select a stopped clock.

5.3 Stopping the Clock Oscillation

When you stop the clock, do not stop it if it is selected as the system clock source. Stop the clock after changing the system clock source to another clock.

5.4 Switching the Power Control Mode

When you switch the power control mode, confirm that the system clock frequency is within the specified range before and after switching the mode.

6. Usage Note on R_LPM Driver

This section explains the main points about R_LPM driver. For other notes, refer to the external specification introduced in section 4.1.

6.1 R_SYSTEM Driver Initialization

R_SYSTEM initialization is required before using R_LPM driver.

6.2 Preparations before Transition to VBB mode

When you transition to VBB mode, setup the back-bias voltage control circuit is required.

First, enable the clock used as back-bias voltage control circuit clock. After that, enable the back-bias voltage control circuit operation and confirm the setup is completed. Because setup of the back-bias voltage control circuit takes time, we recommend that you enable the operation in advance.

6.3 Preparations before Transition to SSTBY mode

When you transition to SSTBY mode, set the power supply modes and VBB mode of during SSTBY mode in advance by using the R_LPM_SSTBYModeSetup function.

6.4 Preparations before Transition to EXFPWON Mode and MINPWON Mode

Both EXFPWON and MINPWON modes disable flash memory power-domain. Before transition to these power supply modes, it is crucial that all needed code must be copied from the flash memory into the RAM. Then ensure all constants and variables are also located in the RAM area, and run EXFPWON / MINPWON modes sequence from code within the RAM. Otherwise unpredictable behavior will cause.

6.4.1 Example of Allocating User Program to RAM Area

This section describes procedure to allocate functions or variables used in the user program in the RAM area.

1. Allocate functions or variables to the specific RAM area by using __attribute__ ((section("xxxx"))). Sample code uses sections below.
   - section " .ramdata" : variables to be allocated to RAM area
   - section " .ramfunc" : functions to be allocated to RAM area

   Example)

```
static const int32_t sample_data __attribute__ ((section(".ramdata")));
static void  sample_function(void) __attribute__
((section(".ramfunc")));
```

   __attribute__ ((section("xxxx"))) can be used in both IAR compiler and GCC compiler to allocate section. In IAR compiler, #pragma location also can be used for section allocation. Example of the way of section allocation by using #pragma location is shown below:

   Example)

```
#pragma location = ".ramdata"
static const int32_t sample_data = 0;

#pragma location = ".ramfunc"
static void sample_function(void)
{
      ;
}
```

2. Vector table and functions or variables set at 1 are allocated to the RAM area by executing R_SYS_CodeCopy() in R_SYSTEM driver. Execute R_SYS_CodeCopy() before device switches to EXFPWON mode or MINPWON mode.

7. Trouble Shooting

7.1 Build Error Occurs

A) Confirm the settings of include directory have been correctly entered.

<For EWARM>

The include directory can be set in project Option [C/C++ Compiler] -> [Preprocessor]. (See Figure 7-1)

<For e² studio>

The include directory can be set in project Properties [C/C++ General] -> [Paths and Symbols] -> [Includes]. (See Figure 7-2)

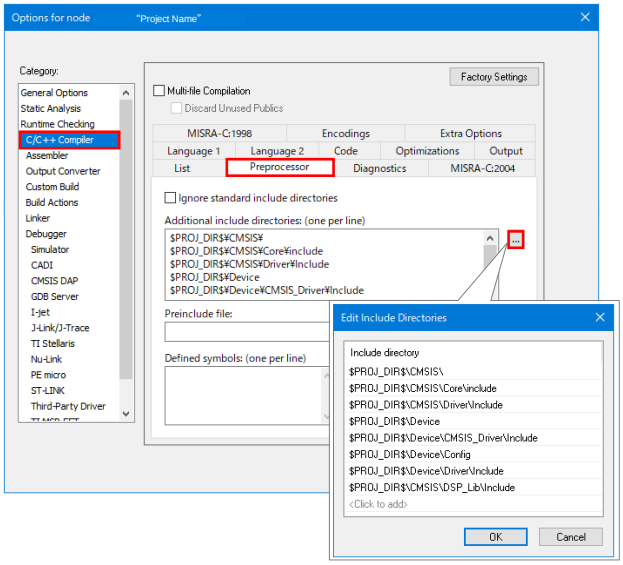


Figure 7-1 Setting of include directory (EWARM)

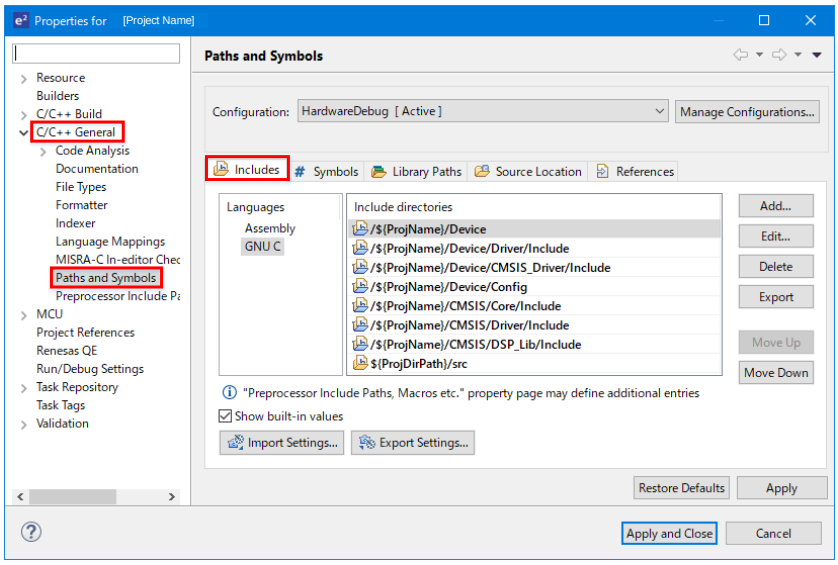


Figure 7-2 Setting of include directory (e² studio)

7.2 Hard Fault Error Occurs When Calling the API of CMSIS Driver

A) It is possible that the API has not been copied to RAM.
Before calling an API that is allocated to RAM, confirm the API is copied to RAM by R_SYS_CodeCopy() function. For details, see section 6.4 and the section “6.6.1 RAM Placement Method Using RAM Placement Section” in RE01 1500KB, 256KB CMSIS Package Startup Guide.

7.3 Peripheral Functions Do Not Work When the API Called

A) Confirm the return value from API is not an error value. In particular, there are many cases in which an error value is returned because the interrupt settings in r_system_cfg.h are insufficient. For details, see the section “6.3 Interrupt Control” in RE01 1500KB, 256KB CMSIS Package Startup Guide.

7.4 Although the Return Value of API is OK, Peripheral Function Ports Do Not Work

A) Confirm the port settings are made correctly by the function in pin.c. For details, see the section “6.5 Pin settings” in RE01 1500KB, 256KB CMSIS Package Startup Guide.

7.5 Does Not Operate after Power Supply Mode is Switched to EXFPWON or MINPWON From ALLPWON.

A) The functions or variables used by the user program at EXFPWON/MINPWON may not be allocated in the RAM area. Confirm that used functions and variables are allocated to RAM area with reference to section 6.4.

7.6 Fail to connect the debugger

A) Confirm the settings of debugger connection.

<For EWARM>

In EWARM, confirm that the following in project Option [I-jet/JTAGjet].

- [Connect during reset (default)] is selected in [setup] -> [Reset] (See Figure 7-3)
- [SWD] is selected in [Interface] -> [Interface] (See Figure 7-4).

<For e^2 studio>

In e^2 studio, confirm that the following in Debug Configurations [Renesas GDB Hardware Debugging] -> [(project name) Hardware Debug] -> [Debugger] -> [Connection Settings]. (See Figure 7-5).

- [SWD] is selected in [Interface] -> [Type]
- [Yes] is selected in [Connection] -> [Reset on connection]

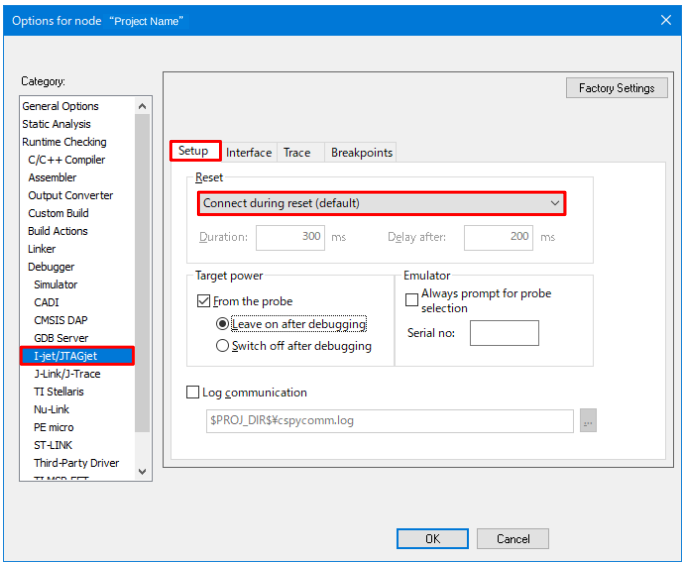


Figure 7-3 Setting of IAR EWARM (1/2)

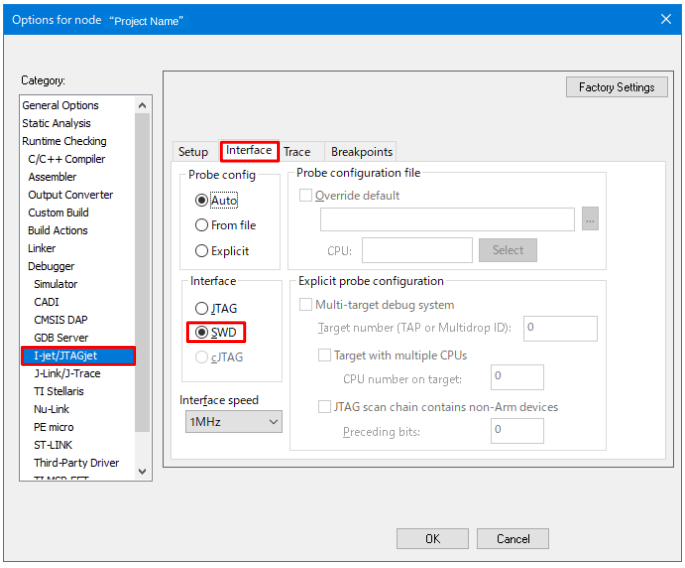


Figure 7-4 Setting of IAR EWARM (2/2)

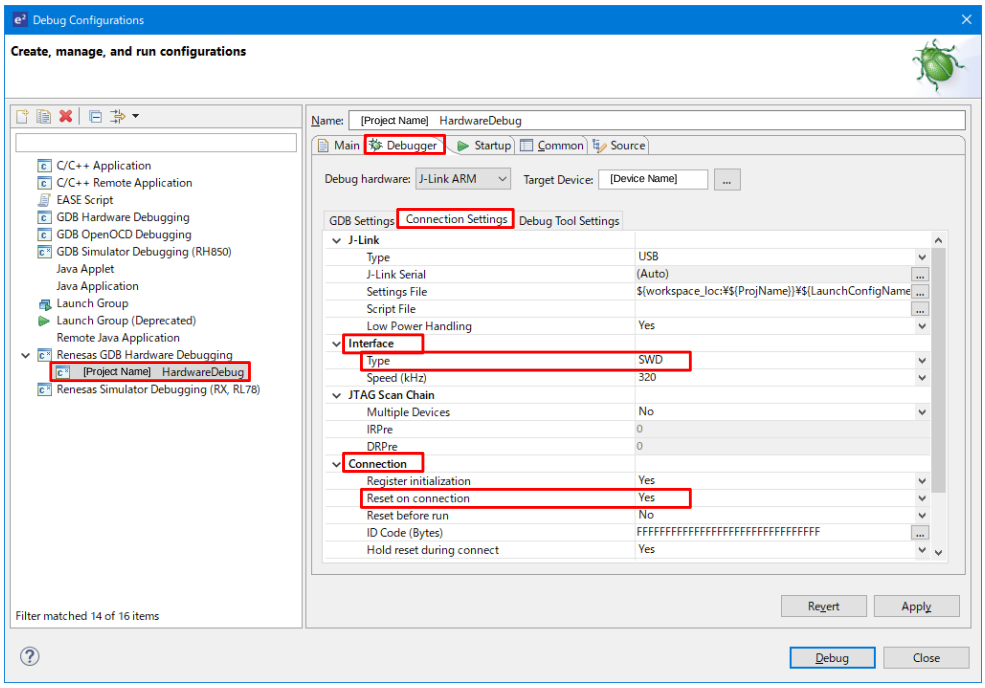


Figure 7-5 Setting of e² studio

7.7 When the Functions in R_SYSTEM Driver and R_LPM Driver Are Used, an Error Occurs During Compilation

A) Confirm that r_system_api.h and r_lpm_api.h are included as header file inside the execution file.

7.8 I/O Ports for Peripheral Function, Analog Input, or General I/O Ports Do Not Work

A) The power supply for IOVCC0/IOVCC1/AVCC0 are shut off after reset release. Before using these pins, SYSTEM.VOCR register has to be set with the R_LPM_IOPowerSupplyModeSet() function.

7.9 Registers Related to Clock Generation Circuit or Power-Saving Functions Cannot Be Set

A) When registers classified into SYSTEM such as registers in Clock Generation Circuit or Power-Saving Functions are set without R_SYSTEM driver or R_LPM driver, it is necessary to disable register write protect. For details about Register write protect, see 15. Register Write Protection in user's manual.

7.10 The Setup of Back-bias Voltage Control Circuit Is Not Completed When SOSC Clock Is Selected As the Back-bias Voltage Control Circuit Clock (VBBST.VBBSTUP Flag does not become to 1)

A) SOSC must be oscillation stable before setting the back-bias voltage control circuit to enable. Because the SOSC oscillation stabilization time is depend on the using oscillator and the board characteristics, adequate evaluation is necessary.

7.11 Debugger console displays an error that “driver error. After the error occurred, the program counter (PC) was <unknown>. ” after transition to SSTBY mode in case of debugging with IAR EWARM and I-jet.

A) This error is output because the debugger cannot acquire the program counter value when the MCU transitions to SSTBY mode. Debugging operation can be continued by canceling SSTBY mode.

8. Sample Codes

Sample code can be downloaded from the Renesas Electronics website.

9. References

User’s Manual: Hardware

RE01 256KB Group User’s Manual Hardware R01UH0894

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

Technical Update/Technical News

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

User’s Manual: Development Tools

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

RE01 1500KB, 256KB CMSIS Package Startup Guide

RE01 1500KB, 256KB Group Getting Started Guide to Development Using CMSIS Package R01AN4660

User’s Manual: Evaluation Kit RE01 256KB

RE01 Group Evaluation Kit RE01 256KB (EK-RE01 256KB) User's manual, R20UT4801

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

Appendix

Table Low Power Settings

Category	Method	Setting register	Effect Note	Remarks
Clock setting	Stop clock supply to unused peripheral functions.	Corresponding bit of MSTPCRA - MSTPCRD = 1. (module-stop state)	small	It depends largely on the operating frequency.
	Stop unused clock source.	LOCOCR.LCSTP = 1 (LOCO Stop) or HOCOCR.HCSTP = 1 (HOCO Stop)	small	
	Lower the PCLKB frequency.	SCKDIVCR.PCLKB[2:0]	large	Increase the division ratio and slow down the frequency.
	Enable low power consumption oscillation function of MOSC.	MOMCR.OSCLPEN = 1 (Enable)	-	It largely depends on the connected load resistance and operating frequency. Please check in your environment.
	Disable the noise filter for the SOSC.	SOMCR.SONFSTP = 1 (Noise filter is disabled)	medium	
	Lower the drive capability of the SOSC.	SOMCR.SODRV = 1 and SOMCR.SODRV0=0 (Low CL)	small	The effect is large in the standby mode and small in the normal operation mode.
Power-saving mode setting	Set the optimum Power control mode (BOOST/ High-speed/Low-speed/ VBB) according to the operating frequency.	PWSTCR.PWST[2:0] OPCCR.OPCM	large	For details such as setting to Low-speed mode at 2MHz operation and setting to VBB mode at 32kHz operation, refer to "13.5.2 Setting Power Control Mode (BOOST/ NORMAL/VBB)" in User's Manual Hardware.
	Set the optimum Power supply mode (ALLPWON/EXFPWON/ MINPWON) according to the function to be used.	PWSTCR.PWST[2:0]	-	Shut down unused power domains. For details, refer to "13.5.1 Setting Power Supply Mode (ALLPWON/EXFPWON/ MINPWON)" in User's Manual Hardware. The effect of reducing the current consumption largely depends on the operating frequency. Please check in your environment.
	Set the optimum Low power consumption mode (SLEEP/SSTBY/ DSTBY) according to the function usage status.	SBYCR.SSBY DPSBYCR.DPSBY	large	The supply clocks are stopped and some power domains are shut off (only DSTBY) according to the usage status of the CPU and peripheral functions. For details, refer to "13.6 Low Power Consumption Mode" in User's Manual Hardware.
AGT setting	When using AGT as a cycle timer, set AGT to low power consumption mode.	AGTMR2.LPM = 1 (Low power consumption mode)	small	

Note: Effect : Power consumption reduction effect (expected).
large : more than 100uA
medium : about several uA
small : about several dozen nA to several hundred nA

Revision History

Rev.	Date	Description	
		Page	Summary
0.40	Jan.31.2020	–	First edition, issued
0.80	Apr.20.2020	1, 20	Changed due to the RE01 256KB UMH Rev0.50 -> Rev1.00. Table 1 The summary of this sample code Power-save function -> Power-saving functions Title of 7.9 Power Save Function -> Power-Saving Functions
		3, 5	Changed due to the target board was change to Evaluation Kit RE01 256KB. RE01 256KB Simple Board -> Evaluation Kit RE01 256KB Table 1-1 Used ports in the sample code Used ports : P510 -> P210 Functions : LED2 -> LED0 Table 2-1 Confirmed operating conditions Target board : RE01 256KB Simple Board -> Evaluation Kit RE01 256KB(RTK70E0118CXXXXXBJ) Debugger : Integrated Segger J-Link OB Note JP2 -> J9, Between TP3-TP4 -> Between T3-T6
		5	Changed due to the CMSIS Driver Package Rev0.80. Table 2-1 Confirmed operating conditions I/O header Version : Rev0.40 -> CMSIS Driver Package Version : Rev0.80 Sample code Version: Rev0.40 -> Rev0.80
		11, 12	Modified according to the operation procedure. Figure 3-3 Main processing of the transition and return for MINPWON SSTBY mode Figure 3-4 Main processing of the transition and return for EXFPWON SSTBY mode Changed the order of "Set system clock to 32MHz" and "Light LED".
		22	Added Appendix. Table “Low Power Settings”.
1.00	Jun.25.2020	5, 22	Changed due to the CMSIS Driver Package Rev1.00. Table 2-1 Confirmed operating conditions CMSIS Driver Package Version :Rev0.80 -> Rev1.00 Sample code Version :Rev0.80 -> Rev1.00 Appendix: Table Low Power Settings Added "SOMCR.SODRV0=0" to the setting register that lowers the drive capacity of the SOSC.
		5	Updated the e^2 studio version. Table 2-1 Confirmed operating conditions Integrated Development Environment GCC: e^2 studio Version7 -> e^2 studio 2020-07
		5, 21	Added the information of Evaluation Kit RE01 256KB User's manual. Table 2-1 Note 9. References

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/.