RX630 Group

Deep Software Standby Mode

R01AN0592EJ0100
Rev.1.00
Dec 13, 2011

Introduction

This application note describes setting up deep software standby mode using the RX630 LVD (voltage detection circuit) module.

Target Device

RX630 Group

This application note can also be used with other RX family microcontrollers that have the same I/O registers (peripheral unit control registers) as the RX630 Group products. Note, however, that since there are changes between devices, such as additional functionality in certain functions, these operations must be verified with the manual for the device actually used. When using the methods described in this application note, full testing in the actual user system is required.

Contents

1. Specifications

In deep software standby mode, all functions of the CPU, the built-in peripheral functions, internal RAM 1, and the oscillators are stopped. Furthermore, since power supply to these circuits is stopped, power consumption is reduced dramatically. At this time, all data in the CPU, the built-in peripheral function registers, and internal RAM 1 becomes undefined. The states of the address bus and bus control signals can be selected with deep standby controller register (DPSBYCR) settings.

Whether or not power is supplied to internal RAM 0 can be selected with deep standby controller register settings. If internal power is supplied, the internal RAM 0 data will be retained. If power is not supplied to internal RAM 0, power consumption is reduced even further. In this case, however, data in internal RAM 0 will be undefined.

The deep standby backup registers (DPSBKRm, m = 0 to 31) are 32-byte read/write registers whose contents are retained in deep software standby mode. Even in a deep software standby mode in which no internal RAM data is retained, the data in the deep standby backup registers will be retained.

The LVD (voltage detection circuit) monitors the voltage applied to the VCC pin. The voltage applied to the VCC pin can be monitored in software.

The enabled/disabled state of the voltage monitor 0 reset function can be selected with the option function selection register 1 (OFS) after a reset.

The voltage monitor 0 reset, voltage monitor 1 reset/interrupt, and the voltage monitor 2 reset/interrupt functions can be used.

This application note presents an example of deep software standby mode in which voltage monitor 1 is used.

Table 1.1 lists the peripheral functions used and their uses.

Table 1.1 Peripheral Functions and their Uses

Peripheral Function	Use
Power consumption reduction functions	• Deep software standby mode • Clear factor: voltage monitor 1 reset • Power is not supplied to internal RAM 0 or the USB resume detection block in deep software standby mode. • The functions of the I/O ports is retained even after deep software standby mode is cleared. • Internal power supply to internal RAM 0 is stopped.
Voltage detection circuit	• Voltage detection circuit 1 is enabled • Voltage level at transition to deep software standby mode (when the voltage falls): 2.95 V • When deep software standby mode is cleared (when the voltage rises): Voltage detection level: 2.95 V Voltage monitor 1 reset: enabled

2. Conditions of Checking the Operation of the Software

The sample code described in this application note has been confirmed to run normally under the operating conditions given below.

Table 2.1 Operating Conditions

Item	Description
MCU	RX630 (R5F5630EDDFP)
Memory used for evaluation	Xin clock: 12 MHz Subclock: 32.768 kHz
Operating voltage	3.3 V
Integrated development environment	Version 4.09.00.007
C compiler	RX Standard Toolchain (V.1.0.1.0) C/C++ compiler package for RX family V.1.01.00
	Compiler options: The default settings for the integrated development environment are used.
Operating mode	Single chip mode
Version of the sample code	Version 1.40
Board used	R0K505630C001BR

3. Software Description

3.1 Operation Overview

In this application example, deep software standby mode is cleared by the voltage monitor 1 voltage detection circuit.

The address bus and bus control signal states are set to high impedance in deep software standby mode.

Table 3.1 lists the settings and conditions for deep software standby mode and table 3.2 lists the settings and conditions for the voltage detection circuit.

Table 3.1 Deep Software Standby Mode Settings and Conditions

Transition destination after a WAIT instruction is executed	Deep software standby mode
Output port enable	The address bus and bus control signal states are set to high impedance in deep software standby mode.
Internal power supply	Power is not supplied to internal RAM 0 and the USB resume detection block in deep software standby mode.
I/O port retention	The I/O port states are retained even after deep software standby mode is cleared.
Clear factor	Deep software standby mode is cleared by the voltage monitor 1 signal.

Table 3.2 Voltage Detection Circuit Settings and Conditions

Circuit used	Voltage detection 1
Setting for transition to deep software standby mode	• Voltage detection level: 2.95 V • Voltage detection 1 monitors the circuit comparison result
Setting for clearing deep software standby mode	• Voltage detection level: 2.95 V • Voltage detection 1 reset: enabled

Notes: 1. The WAIT instruction is used for deep software standby mode. The WAIT instruction is a privileged instruction. Therefore, if the processor mode is set to user mode, the WAIT instruction will cause a privileged instruction exception and the device will not transition to deep software standby mode. The processor must be set to supervisor more in advance to use deep software standby mode.

2. The transition to deep software standby mode passes through software standby mode. If the clear factor for software standby mode is set and there is a contention between the occurrence of the software standby mode clear factor and the transition to deep software standby mode, software standby mode will be cleared and exception handling for the interrupt for the set clear factor will start. Here, the device will not transition to deep software standby mode.
Since no software standby mode clear factor is set in this application example, this phenomenon will not occur.

3. The DMAC DMSCNT.DMST bit and the DTC DTCST.DTCST bit must be cleared to 0 before executing the WAIT instruction. If the WDT module is used as a watchdog timer, the device will not transition to software standby mode. Stop the WDT before executing the WAIT instruction.

3.2 File Structure

Table 3.3 lists the files used in the sample code.

Table 3.3 File Structure

File Name	Function	Notes
main_dpstby.c	• The main processing • Deep software standby mode related processing	
dbsct.c	B and R section settings	File automatically generated by the IDE
intprg.c	Interrupt handling (The SCI interrupt handler, which is used by this program, has been removed from this file.)	File automatically generated by the IDE
resetprg.c	Reset handling	File automatically generated by the IDE
sbrk.c	sbrk() function	File automatically generated by the IDE
vecttbl.c	Vector table related processing	File automatically generated by the IDE to which option and memory settings have been added
iodefine.h	I/O register related header file	
lowsrc.h	I/O streams related header file	File automatically generated by the IDE
sbrk.h	sbrk() function header file	File automatically generated by the IDE
stacksct.h	Stack area header file	File automatically generated by the IDE
typedefine.h	Integer type definitions header file	File automatically generated by the IDE
vect.h	Vector table related header file	File automatically generated by the IDE

3.3 Constants

Table 3.4 lists the constants used by the sample code.

Table 3.4 Constants Used in the Sample Code

Constant	Set Value	Description
None		

3.4 Variables

Table 3.5 lists the global variables.

Table 3.5 Global Variables

Type	Name	Description	Functions Where Used
unsigned char	dpstby_count	Count of number of times deep software standby mode has been cleared	deepstandby_setup

3.5 Functions

Table 3.6 lists the functions defined in the sample code.

Table 3.6 Functions

Function Name	Overview
mcu_init	CPU initialization
clock_setting	CPU clock settings
peripheral_init	Peripheral function initialization
clear_PSW_I	Clears the PSW I bit
LVD1_init	Initializes voltage detection 1
LVD1_setup	Sets voltage detection 1 to clear deep software standby mode
deepstandby_setup	Sets up deep software standby mode

3.6 Function Specifications

This section lists the specifications of the functions in the sample code.

Name	mcu_init
Overview	CPU initialization
Header	Iodefine.h
Declaration	void mcu_init(void)
Description	Sets the CPU clock.
Arguments	None
Return values	None
Notes	

Name	clock_setting
Overview	Sets the CPU clock.
Header	Iodefine.h
Declaration	void clock_setting(void)
Description	• Stops the subclock oscillator. • Stops the high-speed clock oscillator. • Sets the oscillator stabilization time for the main clock oscillator to 131,072 cycles. • Sets the PLL oscillator stabilization time to 4,194,304 cycles. • Sets the PLL frequency multiplier to 16x. • Sets the main clock oscillator to the operating state. • Sets the PLL circuit to the operating state. • Sets the system clock to divided by 2, the FlashIF clock to divided by 4, the external bus clock to divided by 4, and the peripheral module clock to divided by 4. • Sets the clock source to be the PLL circuit.
Arguments	None
Return values	None
Notes	

Name	peripheral_init
Overview	Peripheral function initialization
Header	Iodefine.h
Declaration	void peripheral_init(void)
Description	• Clears the PSW I bit • Sets up (initializes) deep software standby mode. • Initializes voltage detection 1
Arguments	None
Return values	None
Notes	

Name	clear_PSW_I
Overview	Clears the PSW I bit
Header	machine.h
Declaration	void clear_PSW_I(void)
Description	Clears the PSW I bit and sets the system to the interrupts disabled state.
Arguments	None
Return values	None
Notes	

Name	LVD1_init
Overview	Initializes voltage detection 1
Header	Iodefine.h
Declaration	void LVD1_init(void)
Description	• Sets the voltage detection 1 detection voltage to 2.95 V. • Enables output of the voltage detection 1 circuit comparison result. • Enables the voltage detection 1 circuit.
Arguments	None
Return values	None
Notes	

Name	LVD1_setup
Overview	Sets voltage detection 1 to clear deep software standby mode.
Header	Iodefine.h
Declaration	void LVD1_setup(void)
Description	• Disables the voltage detection 1 circuit. • Disables the voltage detection 1 circuit interrupt and reset. • Disables output of the voltage detection 1 circuit comparison result. • Sets the voltage detection 1 detection voltage to 2.95 V. • Disables the voltage detection 1 digital filter. • Sets the voltage detection 1 circuit mode to voltage monitor 1 reset when Vdet1 is passed. • Sets the reset negate type to negate after a fixed time (tLVD1) has passed after detecting VCC > Vdet1. • Enables output of the voltage detection 1 circuit comparison result. • Enables the voltage detection 1 circuit interrupt and reset. • Enables the voltage detection 1 circuit.
Arguments	None
Return values	None
Notes	

Name	deepstandby_setup
Overview	Sets up deep software standby mode.
Header	Iodefine.h
Declaration	void deepstandby_setup(void)
Description	Pin reset (Sets the deep software standby mode registers) • Sets the post-WAIT instruction execution transition target (SBYCR, DPSBYCR) • Sets the RAM 0 power supply state (DPSBYCR) • Sets the clear factor (DPSIER2) • Clears the LVD1 clear request flag (DPSIFR2) • Selects the edge type for the clear factor (DPSIEGR2) • Clears the reset status flag (RSTSR0) Clear of deep software standby mode • Clears the reset status flag (RSTSR0) • Clears the LVD1 clear request flag (DPSIFR2) • Sets the deep software standby mode registers again (SBYCR, DPSBYCR)
Arguments	None
Return values	None
Notes	

3.7 Flowcharts

3.7.1 Main Processing

Figure 3.1 shows the main processing flowchart.

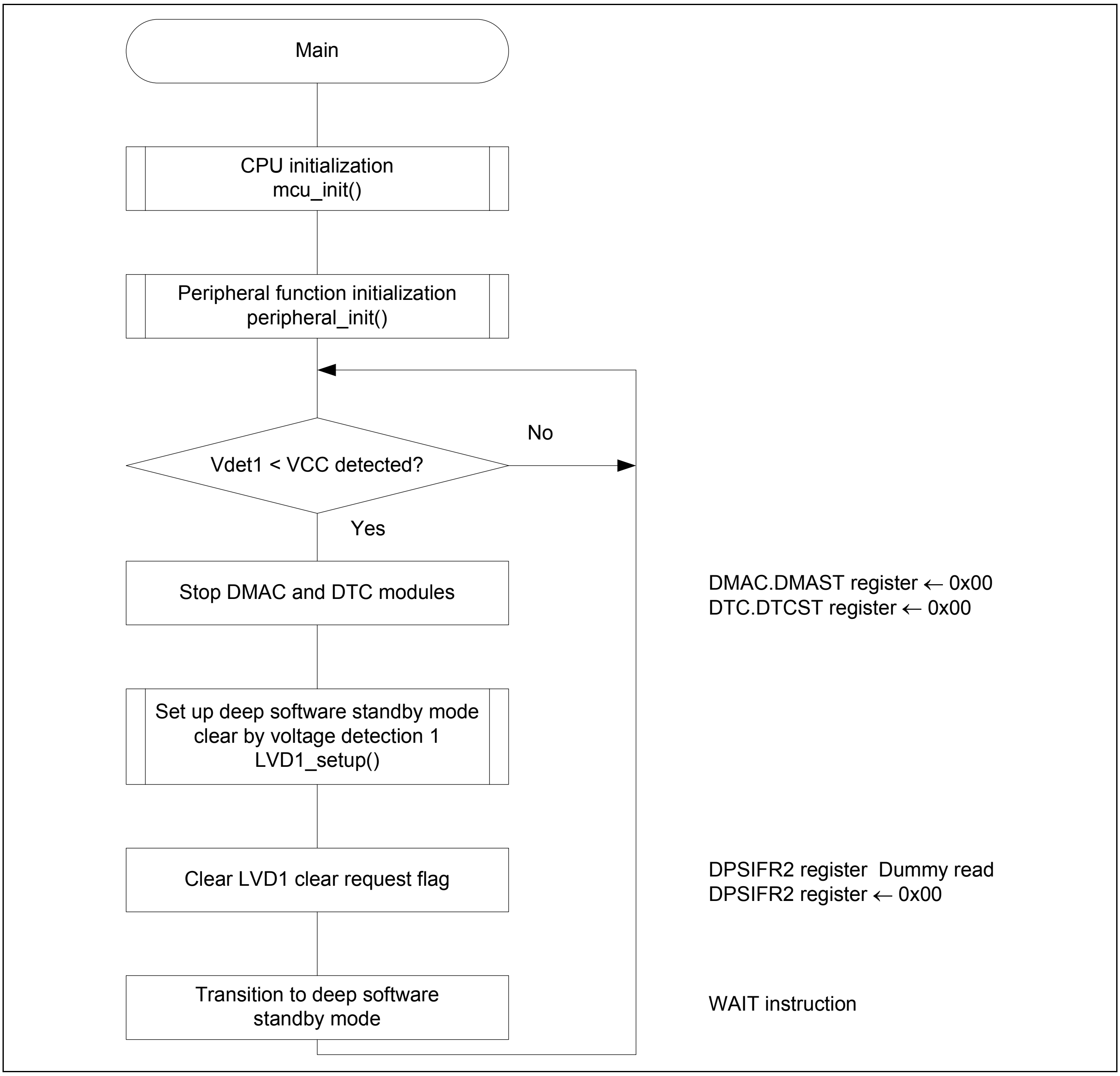

Figure 3.1 Main Processing

3.7.2 CPU Initialization

Figure 3.2 shows the flowchart for CPU initialization.

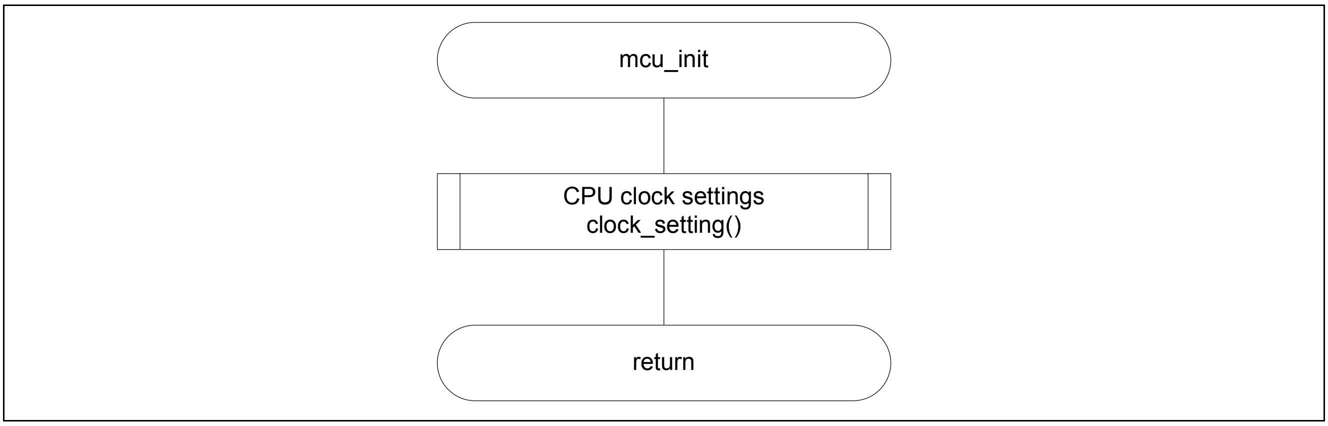

Figure 3.2 CPU Initialization

3.7.3 CPU Clock Settings

Figure 3.3 shows the flowchart for the CPU clock settings.

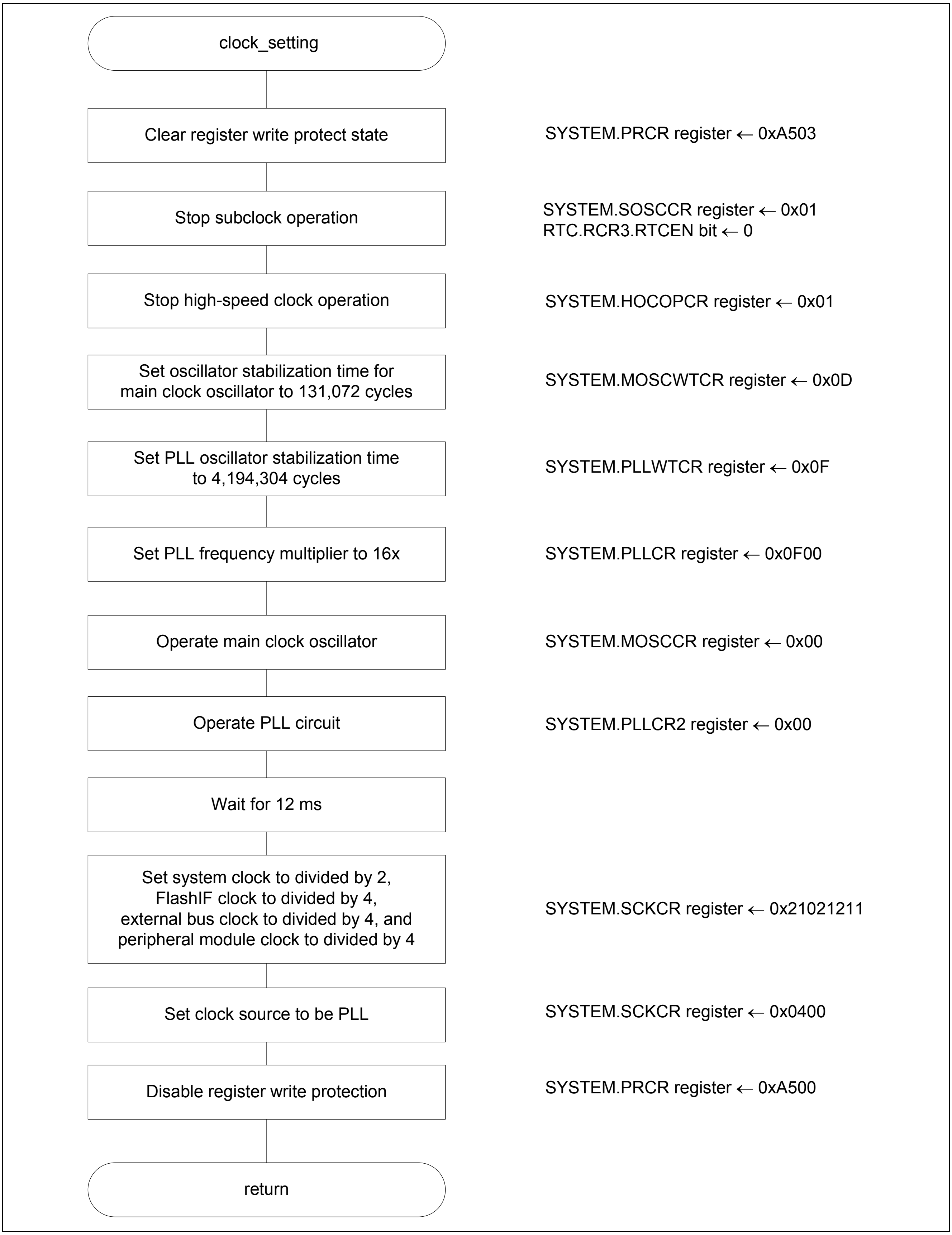

Figure 3.3 CPU Clock Settings

3.7.4 Peripheral Function Initialization

Figure 3.4 shows the flowchart for peripheral function initialization.

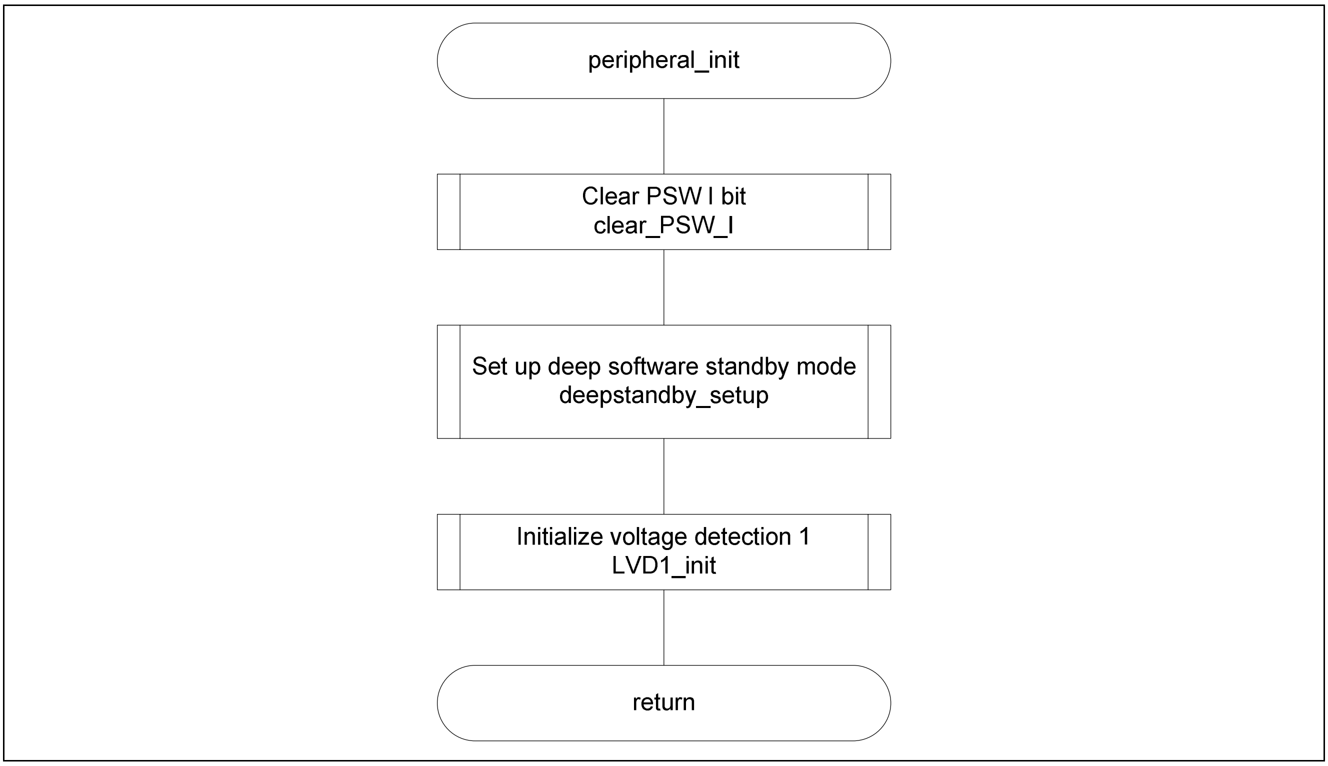

Figure 3.4 Peripheral Function Initialization

3.7.5 Clearing PSW I Bit

Figure 3.5 shows the flowchart for clearing the I bit in the PSW.

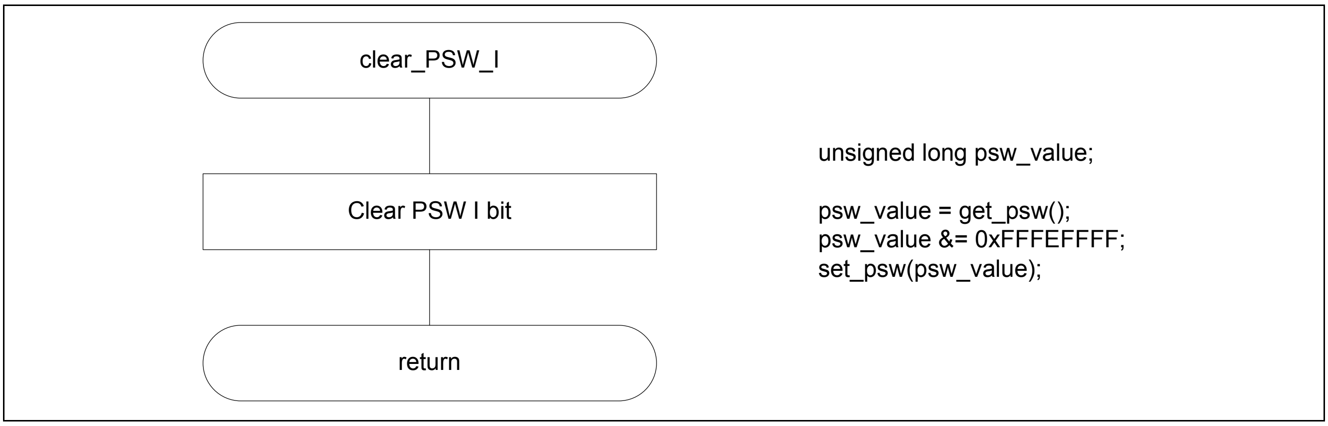

Figure 3.5 Clearing PSW I Bit

3.7.6 Initializing Voltage Detection 1

Figure 3.6 shows the flowchart for initializing voltage detection 1.

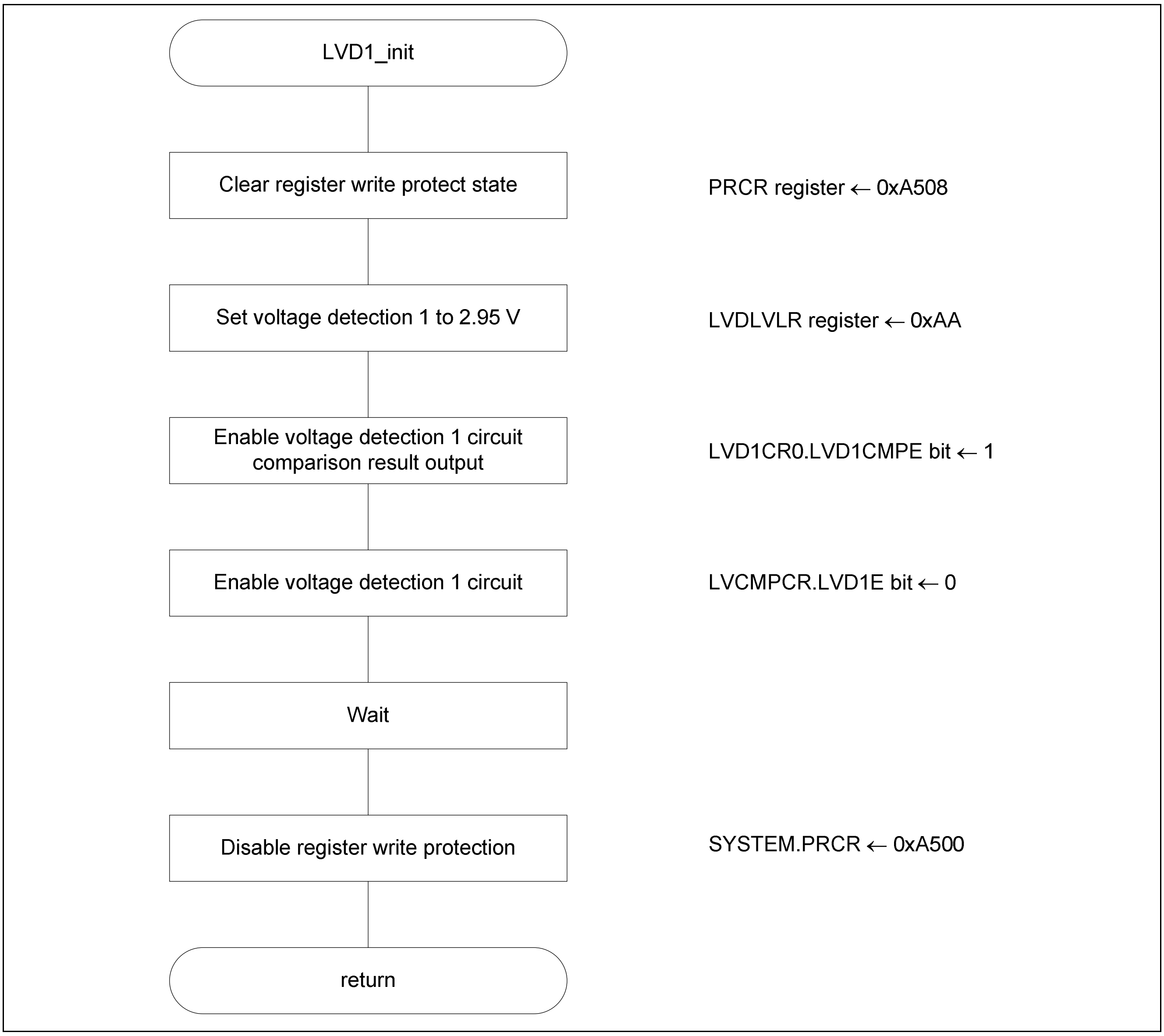

Figure 3.6 Initializing Voltage Detection 1

3.7.7 Setting Voltage Detection 1 Deep Software Standby Mode Clear

Figure 3.7 shows the flowchart for setting up clearing deep software standby mode with voltage detection 1.

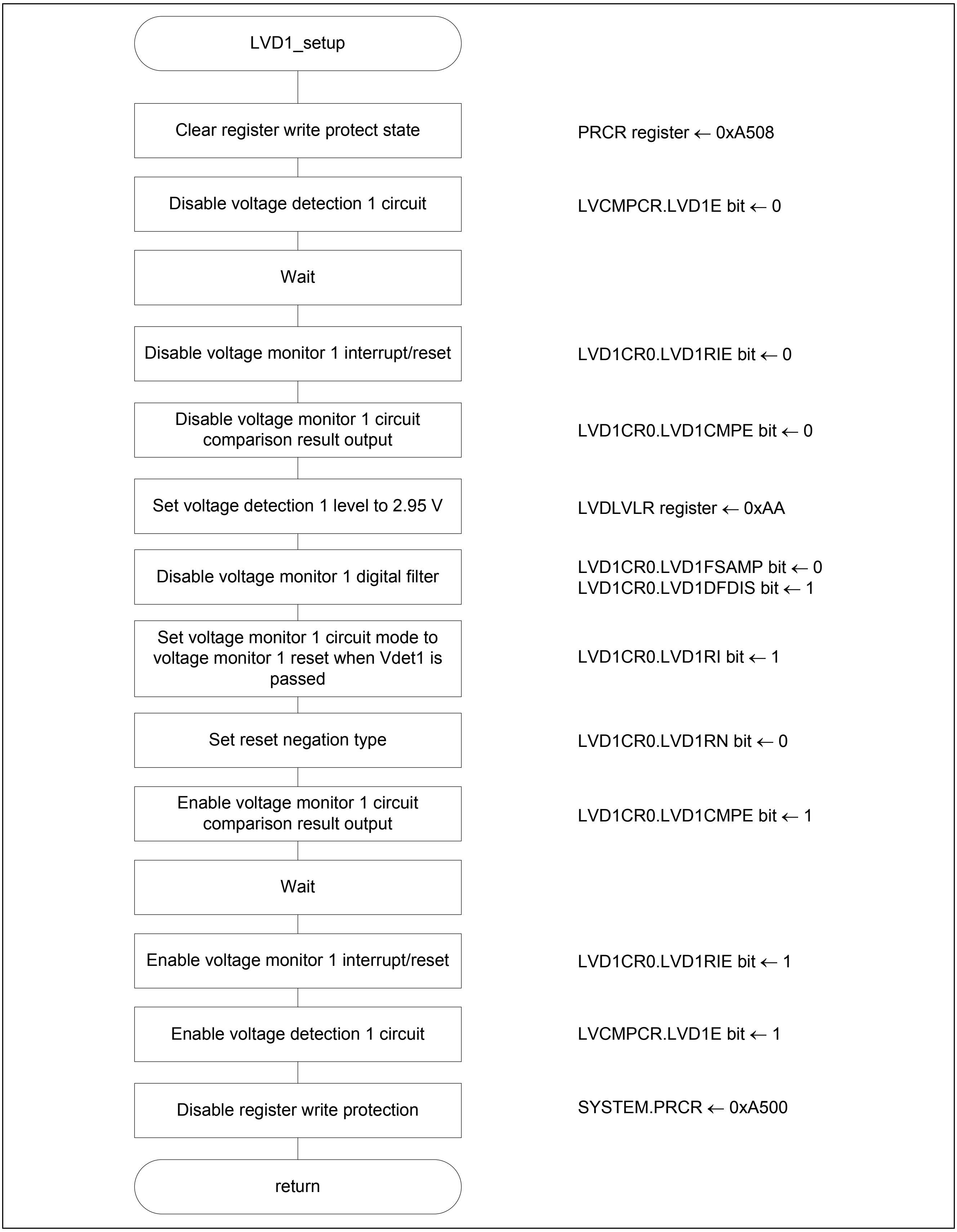

Figure 3.7 Setting Voltage Detection 1 Deep Software Standby Mode Clear

3.7.8 Deep Software Standby Mode Settings

Figures 3.8 and 3.9 show the flowchart for setting up deep software standby mode.

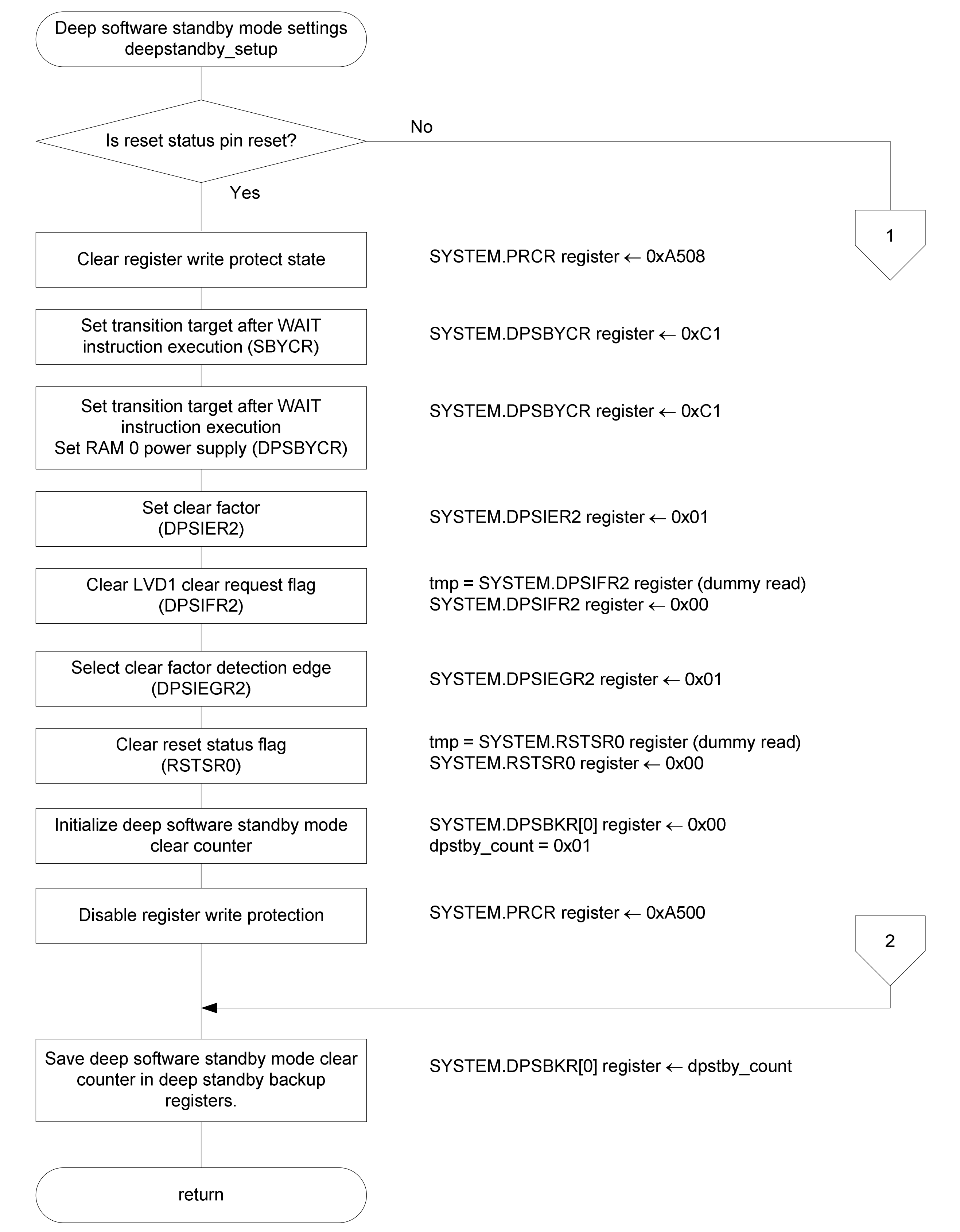

Figure 3.8 Deep Software Standby Mode Settings (1)

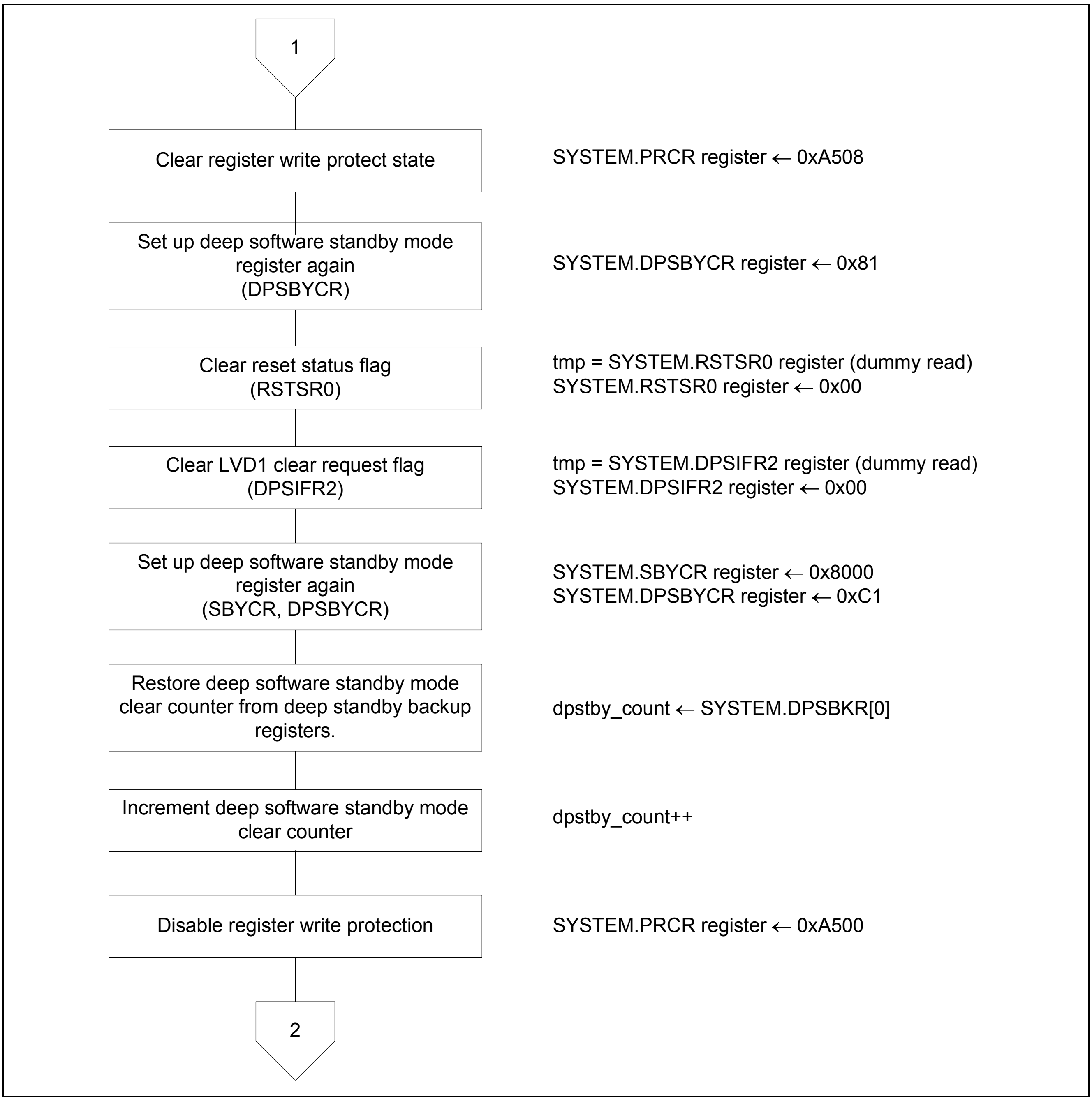

Figure 3.9 Deep Software Standby Mode Settings (2)

4. Sample Programs

The sample program can be downloaded from the Renesas Electronics Web site.

5. Reference Documents

- RX630 Group User's Manual: Hardware, Rev.1.00
  (The latest version can be downloaded from the Renesas Electronics Web site.)
- Technical Updates/Technical News
  (The latest information can be downloaded from the Renesas Electronics Web site.)
- C Compiler Manual
  RX Family C/C++ Compiler Package User's Manual V.1.0.1.0
  (The latest version can be downloaded from the Renesas Electronics Web site.)

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Dec.13.11	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.
   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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

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

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

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

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

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

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

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

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

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

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

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

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1