

RL78/G23

SMS Power Supply Monitoring

Introduction

This application note describes how to build a power supply monitoring system using the SNOOZE mode sequencer. By processing the power supply voltage measurement and measurement result judgment required for power supply voltage monitoring by the SNOOZE mode sequencer instead of the CPU, it is possible to monitor with less power consumption.

Target Device

RL78/G23

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Contents

1. Specifications	4
2. Conditions for Operation Confirmation Test	6
3. Related application note	7
4. Hardware	8
4.1 Example of Hardware Configuration	8
4.2 Used Pins	8
5. Software	9
5.1 Overview of the sample program	9
5.2 Folder Configuration	10
5.3 Option Byte Settings	11
5.4 Constants	11
5.5 Variables	12
5.6 Functions	12
5.7 Function Specifications	12
5.8 Flow Charts	14
5.8.1 Main Process	14
5.8.2 INTPO interrupt process	15
5.8.3 Wait process	16
5.8.4 TAU0 channel 7 interrupt process	16
5.9 SNOOZE Mode Sequencer settings	17
6. Application example	22
6.1 r01an5605_sms_power_monitoring.scfg	22
6.1.1 Clocks	24
6.1.2 System	24
6.1.3 r_bsp	24
6.1.4 Config_LVD0	24
6.1.5 Config_INTC	24
6.1.6 Config_IT000_ITL001	24
6.1.7 Config_TAU0_7	24
6.1.8 Config_ADC	25
6.1.9 Config_SMS	25
6.1.10 Config_PORT	25
6.2 r01an5605_sms_power_monitoring.sms	26
6.2.1 Start	26
6.2.2 2byte transfer	27
6.2.3 A/D Change channel	27

6.2.4	A/D Start	27
6.2.5	A/D Get voltage	28
6.2.6	Update	28
6.2.7	A/D End	30
6.2.8	A/D Change channel	30
6.2.9	Compare	30
6.2.10	Finish	31
6.2.11	Variable Setting	32
7.	Sample Code	33
8.	Reference	33
	Revision History	34

1. Specifications

This application note shows how to measure the power supply voltage and judge the measurement result by the processing of the SNOOZE mode sequencer (SMS) during STOP mode, and start the CPU when the power supply voltage drops.

Power supply voltage is measured by A/D converter (ADC). Sets A/D conversion and the process of judging the conversion result in SMS in advance. The interval for measuring the power supply voltage is set using the interval timer (TML32), and SMS is started by the TML32 interrupt (INTITL). When SMS is started, A/D conversion and the process of judging the conversion result are executed in sequence, and if it is detected the power supply voltage drops, an interrupt request signal (INTSMSE) is generated from SMS to start the CPU.

Figure 1-1 shows the system configuration and Figure 1-2 shows the entire flowchart.

Figure 1-1 System Configuration

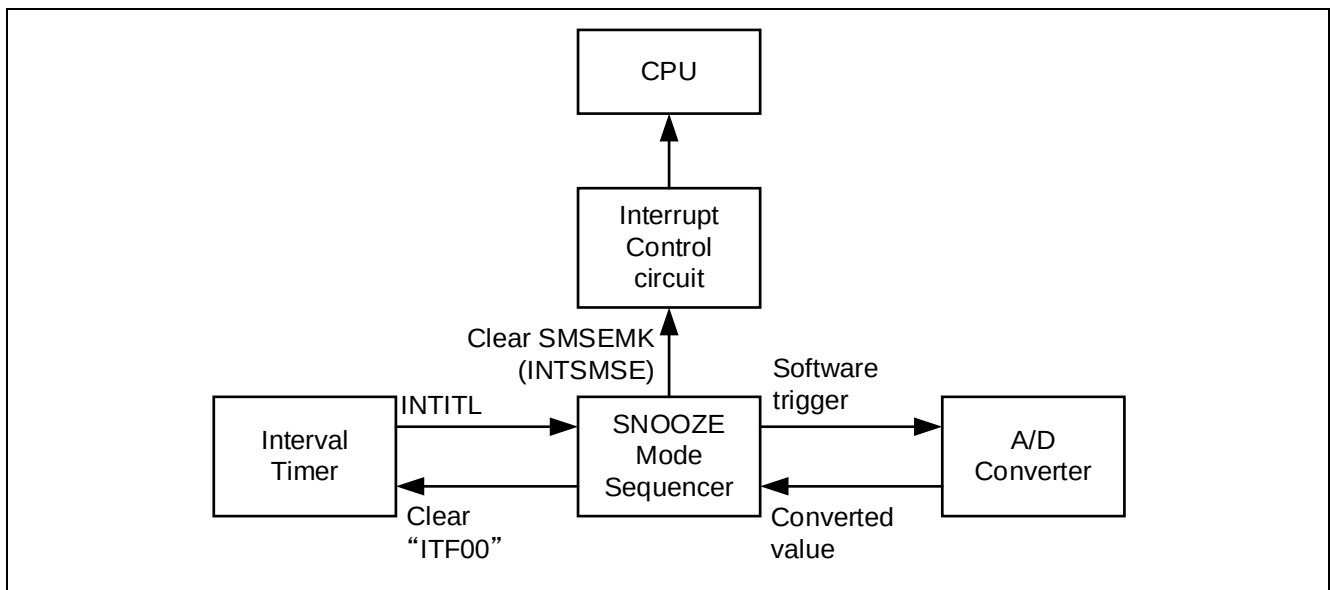
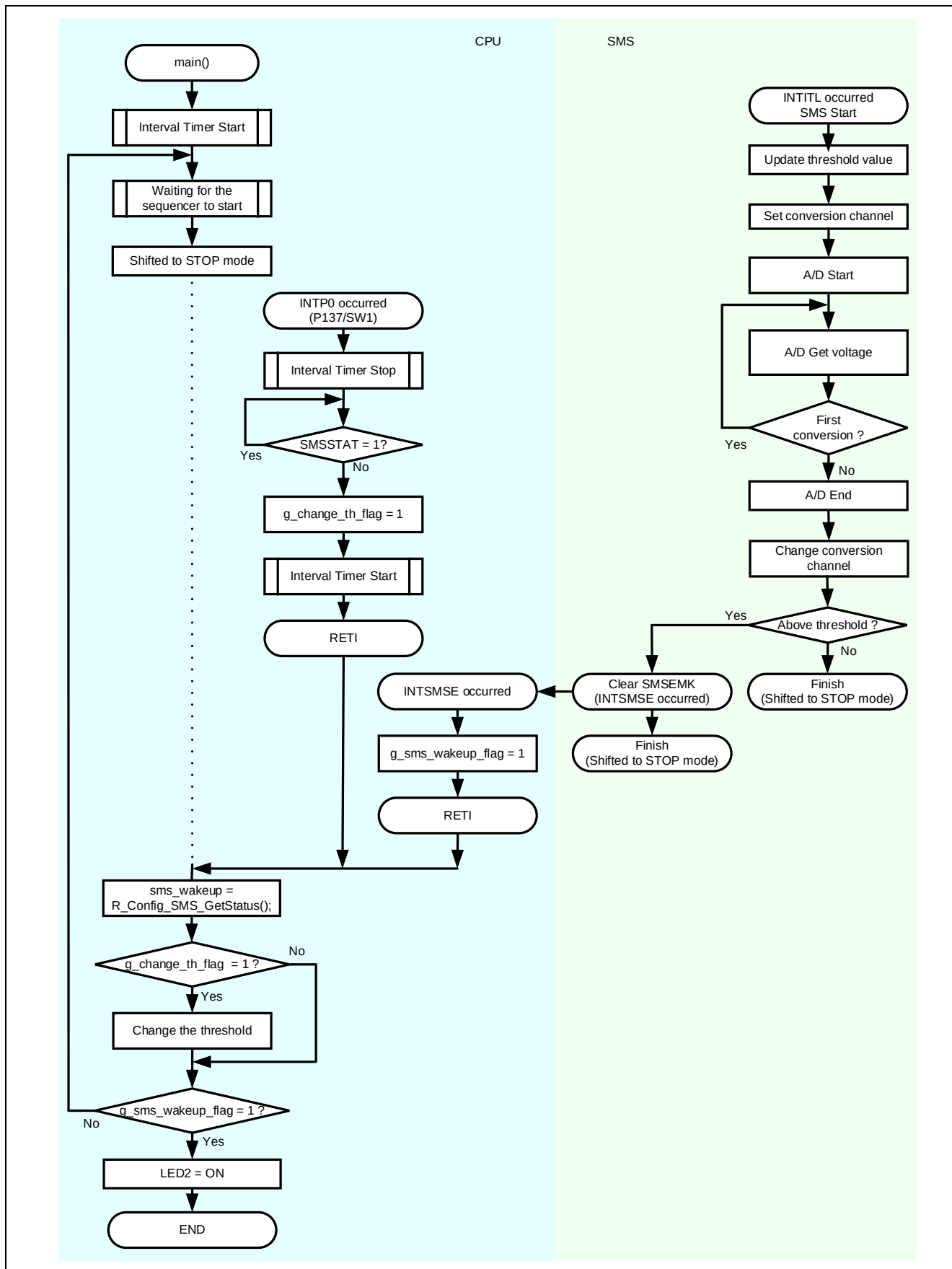


Figure 1-2 Entire Flowchart



2. Conditions for Operation Confirmation Test

The sample code with this application note runs properly under the condition below.

Table 2.1 Operation Confirmation Conditions

Items	Contents
MCU	RL78/G23 (R7F100GLG)
Operating frequencies	- High-speed on-chip oscillator clock: 32 MHz - CPU/peripheral hardware clock: 32 MHz
Operating voltage	3.3V - LVD0 operations (V_{LVD0}) : Reset mode Rising edge TYP.1.875V Falling edge TYP.1.835V
Integrated development environment (CS+)	CS+ for CC V8.05.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.10 from Renesas Electronics Corp.
Integrated development environment (e ² studio)	e ² studio 2021-04 (21.4.0) from Renesas Electronics Corp.
C compiler (e ² studio)	CC-RL V1.10 from Renesas Electronics Corp.
Integrated development environment (IAR)	IAR Embedded Workbench for Renesas RL78 v4.21.1 from IAR Systems
C compiler (IAR)	
Smart Configurator	V.1.0.1
Board support package (r_bsp)	V.1.10
Emulator	CS+, e ² studio: COM port IAR: E2 Emulator Lite
Board	RL78/G23 Fast Prototyping Board (RTK7RLG230CLG000BJ)

3. Related application note

The following application note is related to this application note.

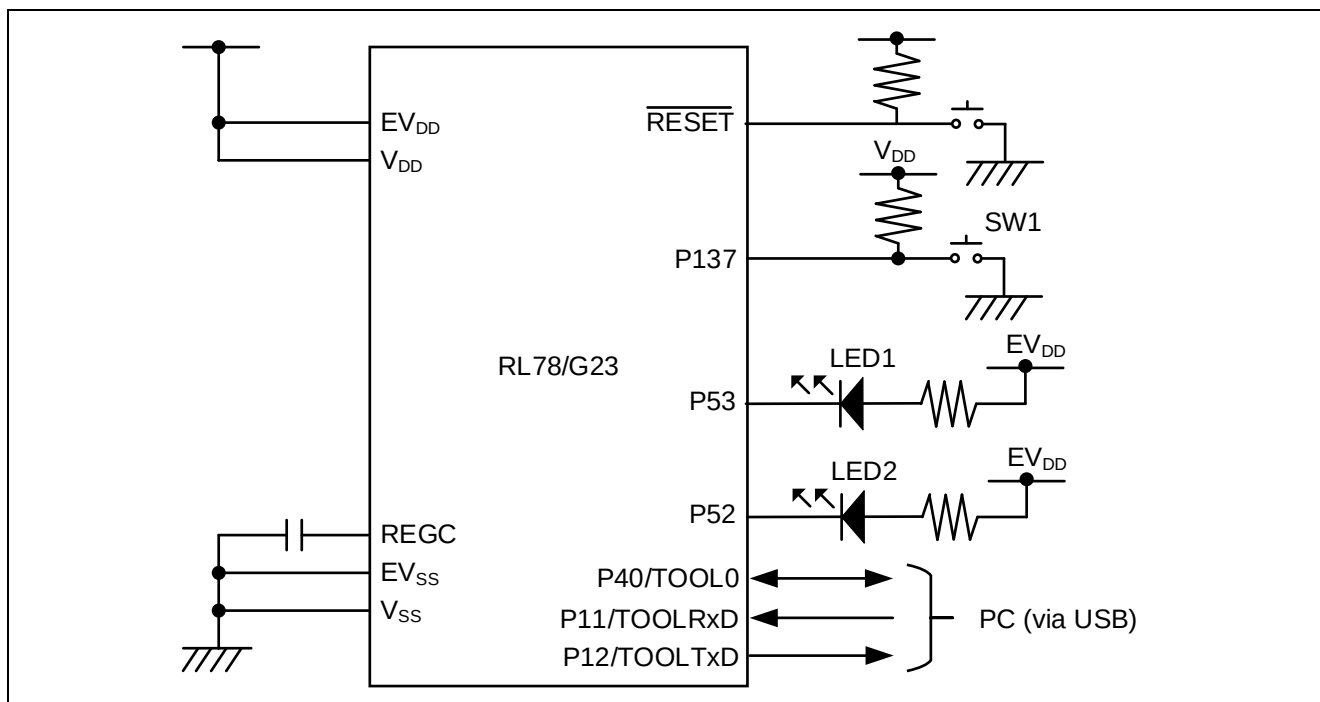
Please refer to them as well.

4. Hardware

4.1 Example of Hardware Configuration

Figure 4-1 shows an example of the hardware configuration in this application.

Figure 4-1 Hardware Configuration



Caution 1. This simplified circuit diagram was created to show an overview of connections only. When actually designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements. (Connect each input-only port to V_{DD} or V_{SS} through a resistor.)

Caution 2. Connect the EV_{SS} pin to V_{SS} and the EV_{DD} pin to V_{DD} .

Caution 3. V_{DD} must be held at not lower than the reset release voltage (V_{LVD0}) that is specified as LVD.

4.2 Used Pins

Figure 4-1 Table 4.1 shows list of used pins and assigned functions.

Table 4.1 List of Pins and Functions

Pin name	Input/Output	Function
P53	Output	LED1 lights (Low Active)
P52	Output	LED2 lights (Low Active)
P137	Input	SW1 (Low Active)

Caution. In this application note, only the used pins are processed. When actually designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.

5. Software

5.1 Overview of the sample program

In this sample code, the RL78/G23 shifts from STOP mode to SNOOZE mode by the interrupt request (INTITL) of the 32-bit interval timer (TML32), and the SNOOZE mode sequencer (SMS) performs A/D conversion and judgment of the conversion result.

A/D is measured by setting the power supply voltage (V_{DD}) to the reference voltage and setting the internal reference voltage to the analog input channel. The internal reference voltage is constant (TYP. 1.48V) regardless of the power supply voltage. Therefore, as the power supply voltage drops, the A/D conversion result becomes larger. SMS compares the A/D conversion result with the threshold value, and if the result exceeds the threshold value, an interrupt request signal (INTSMSE) is generated to start the CPU.

LED1 turns ON/OFF each time SMS completes the process. In addition, the threshold value is changed from 1.7V to 2.0V, 2.5V, 3.0V, 3.5V, 4.0V, 4.5V, 5.0V each time SW1 is pressed. If the threshold is exceeded, LED2 lights up.

Caution. When using the RL78/G23 and LED with the same power supply as in the hardware configuration shown in this application note, the LED may not meet the forward voltage standard of the LED and the LED may not light.

The outline of the processing performed by this sample code is shown below.

- (1) Starts counting TML32
- (2) Shifts to STOP mode
- (3) Shifts to SNOOZE mode with TML32 compare match
- (4) Performs A/D conversion
- (5) Branches to (6) if conversion result does not exceed threshold, else branches (8)
- (6) Switches on/off LED1^{Note}
- (7) Returns to (2)
- (8) Turns off LED1 and start the CPU
- (9) Shifts to normal operation from SNOOZE mode
- (10) Turns on LED2

Note. When it is off, it lights up, and when it is on, it goes out.

(4) to (8) are processed by SMS.

[Threshold change process]

- (1) INTP0 interrupt occurred by pressing SW1
- (2) Threshold changes process
- (3) Shifts STOP mode

5.2 Folder Configuration

Table 5.1 shows folder configuration of source file and header files using by sample code except the files generated by integrated development environment and the files in the bsp environment.

Table 5.1 Folder configuration

Folder/File configuration	Outline	Created by Smart configurator
¥r01an5605_sms_power_monitoring<DIR>	Root folder of this sample code	
¥src<DIR>	Folder for program source	
main.c	Sample code source file	
¥smc_gen<DIR> ^{Note 4}	Folder created by Smart Configurator	√
¥Config_ADC<DIR>	Folder for ADC program	√
Config_ADC.c	Source file for ADC	√
Config_ADC.h	Header file for ADC	√
Config_ADC_user.c	Interrupt source file for ADC	√ ^{Note 1}
¥Config_INTC<DIR>	Folder for interrupt program	√
Config_INTC.c	Source file for INTP0 (SW1)	√
Config_INTC.h	Header file for INTP0	√
Config_INTC_user.c	Interrupt source file for INTP0	√ ^{Note 2}
¥Config_ITL000_ITL001<DIR>	Folder for TML32 program	√
Config_ITL000_ITL001.c	Source file for TML32	√
Config_ITL000_ITL001.h	Header file for TML32	√
Config_ITL000_ITL001_user.c	Interrupt source file for TML32	√ ^{Note 1}
¥Config_PORT<DIR>	Folder for PORT program	√
Config_PORT.c	Source file for PORT	√
Config_PORT.h	Header file for PORT	√
Config_PORT_user.c	Interrupt source file for PORT	√ ^{Note 1}
¥Config_SMS<DIR>	Folder for SMS program	√
Config_SMS.c	Source file for SMS	√
Config_SMS.h	Header file for SMS	√
Config_SMS_ASM.smsasm	ASM source file for SMS	√ ^{Note 3}
Config_SMS_user.c	Interrupt source file for SMS	√
¥Config_TAU0_7<DIR>	Folder for TAU program	√
Config_TAU0_7.c	Source file for TAU	√
Config_TAU0_7.h	Header file for TAU	√
Config_TAU0_7_user.c	Interrupt source file for TAU	√ ^{Note 2}
¥general<DIR>	Folder for initialize or common program	√
¥r_bsp<DIR>	Folder for BSP program	√
¥r_config<DIR>	Folder for BSP_CFG program	√

Note. <DIR> means directory.

Note 1. Not used in this sample code.

Note 2. Added the interrupt handling routine to the file generated by the Smart Configurator.

Note 3. Added the LED1 ON/OFF process to the file generated by the Smart Configurator.

Note 4. The sample code of the IAR version has a different configuration. Check the sample code of the IAR version for details. In addition, stores r01an5605_sms_power_monitoring.ipcf. For details, refer to "RL78 Smart Configurator User's Guide: IAREW (R20AN0581)".

5.3 Option Byte Settings

Table 5.2 shows the option byte settings.

Table 5.2 Option Byte Settings

Address	Setting Value	Contents
000C0H/040C0H	1110 1111B (EFH)	Operation of Watchdog timer is stopped (counting is stopped after reset)
000C1H/040C1H	1111 1110B (FEH)	LVD0 operating mode: reset mode Detection voltage: Rising edge 1.875V Falling edge 1.835V
000C2H/040C2H	1110 1000B (E8H)	Flash operating mode: HS mode High-speed on-chip oscillator clock: 32MHz
000C3H/040C3H	1000 0101B (85H)	On-chip debugging is enabled

5.4 Constants

Table 5.3 shows the constants that are used in this sample code.

Table 5.3 Constants used in the sample code

Constant Name	Setting Value	Contents	File
LED2	P5_bit.no2	P52	main.c
LED_ON	0	Setting value for turning on the LED	main.c
THRESHOLD_PATTE RN	8	Number of changes of the A/D conversion result threshold	main.c
CHATTA_WAIT	100	CHATTA_WAIT	Config_INTC_user.c
g_adcr_th[]	0DEDH	A/D conversion result threshold ($V_{DD} = 1.7V$) ^{Note}	main.c
	0BD6H	A/D conversion result threshold ($V_{DD} = 2.0V$) ^{Note}	
	0978H	A/D conversion result threshold ($V_{DD} = 2.5V$) ^{Note}	
	07E4H	A/D conversion result threshold ($V_{DD} = 3.0V$) ^{Note}	
	06C3H	A/D conversion result threshold ($V_{DD} = 3.5V$) ^{Note}	
	05EBH	A/D conversion result threshold ($V_{DD} = 4.0V$) ^{Note}	
	0542H	A/D conversion result threshold ($V_{DD} = 4.5V$) ^{Note}	
	04BCH	A/D conversion result threshold ($V_{DD} = 5.0V$) ^{Note}	

Note. The A/D conversion result threshold sets the theoretical value of the A/D conversion result when the reference voltage is set to V_{DD} , the analog input channel is set to the internal reference voltage (TYP. 1.48V), and the reference V_{DD} voltage fluctuates from 1.7V to 5.0V.

5.5 Variables

Table 5.4 shows the global variables used in this sample code.

Table 5.4 Global variables used in the sample code

Type	Variable name	contents	Functions used in
volatile uint16_t	g_ms_timer	Count value of the wait process	r_ms_delay, r_Config_TAU0_7_interrupt
uint8_t	g_change_th_flag	SW1 press flag	main.c Config_INTC_user.c

5.6 Functions

Table 5.5 shows the functions used in the sample code. However, the unchanged functions generated by the Smart Configurator are excluded.

Table 5.5 Functions

Function name	Outline	Source file
main	Main process	main.c
r_Config_INTC_intp0_interrupt	INTP0 interrupt process	Config_INTC_user.c
r_ms_delay	Wait process to prevent chattering	Config_TAU0_7_user.c
r_Config_TAU0_7_interrupt	TAU0 channel 7 interrupt process (For chattering prevention)	Config_TAU0_7_user.c

5.7 Function Specifications

This part describes function specifications of the sample code.

[Function name] main

Outline	Main process
Header	r_smc_entry.h
Declaration	void main (void);
Description	This function sets the threshold for starting the CPU from standby mode, then starts TML32 operation and shifts to STOP mode. When the CPU returns from standby mode, if the SW1 press flag is not set, the process shifts to checking the CPU start request flag. If the SW1 press flag is set, the threshold for starting the CPU is changed and the confirmation of the CPU starts. If the CPU start request flag^{Note} is not set, the CPU shifts to STOP mode again, and if the CPU start request flag is set, LED2 lights up. The threshold value is changed from the initial value of 1.7V to 2.0V, 2.5V, 3.0V, 3.5V, 4.0V, 4.5V, 5.0V.
Arguments	None
Return value	None
Remarks	None

Note. Variable flags generated by the Smart Configurator.

[Function name] r_Config_INTC_intp0_interrupt

Outline	INTP0 interrupt process
Header	r_cg_macrodriver.h, r_cg_userdefine.h, Config_INTC.h, Config_TAU0_7.h, Config_ITL000_ITL001.h, Config_SMS.h
Declaration	#pragma interrupt r_Config_INTC_intp0_interrupt (vect=INTP0)
Description	This function is an interrupt process by INTP0 that occurs when SW1 is pressed. Executes the wait process to prevent chattering when SW1 is pressed.
Arguments	None
Return value	None
Remarks	None

[Function name] r_ms_delay

Outline	Wait process
Header	r_cg_macrodriver.h, r_cg_userdefine.h, Config_TAU0_7.h
Declaration	void r_ms_delay (uint16_t msec);
Description	This function waits for the time (ms) specified by the argument msec. This function counts using channel 7. Polls if g_ms_timer is less than CHATTA_WAIT, completes wait process if more than CHATTA_WAIT.
Arguments	msec
Return value	None
Remarks	None

[Function name] r_Config_TAU0_7_interrupt

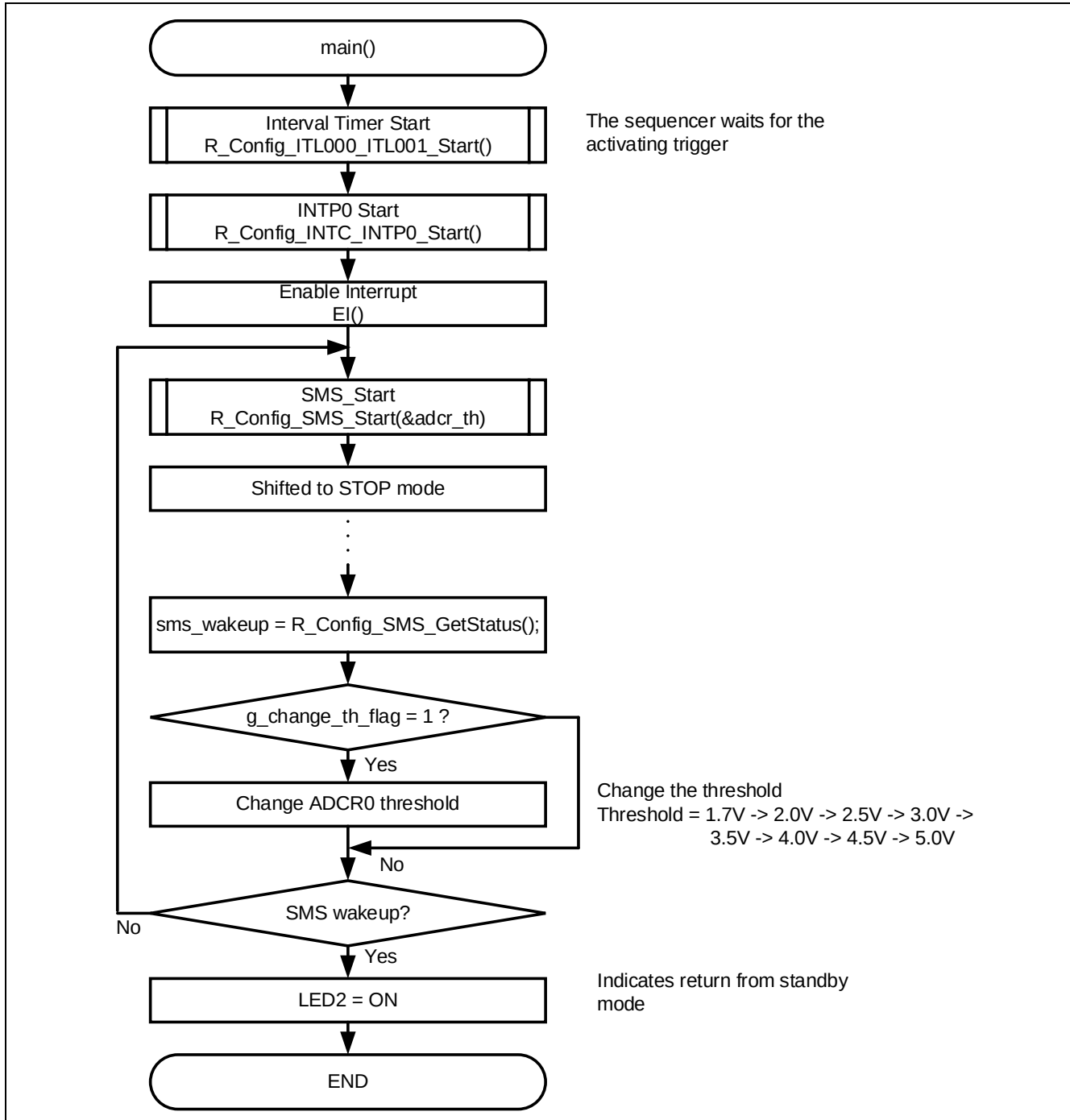
Outline	TAU0 channel 7 interrupt process
Header	r_cg_macrodriver.h, r_cg_userdefine.h, Config_TAU0_7.h
Declaration	#pragma interrupt r_Config_TAU0_7_interrupt (vect=INTTM07)
Description	This function is an interrupt process by INTTM07 on TAU0 channel 7. Counts up g_ms_timer.
Arguments	None
Return value	None
Remarks	None

5.8 Flow Charts

5.8.1 Main Process

Figure 5-1 shows flowchart of main process.

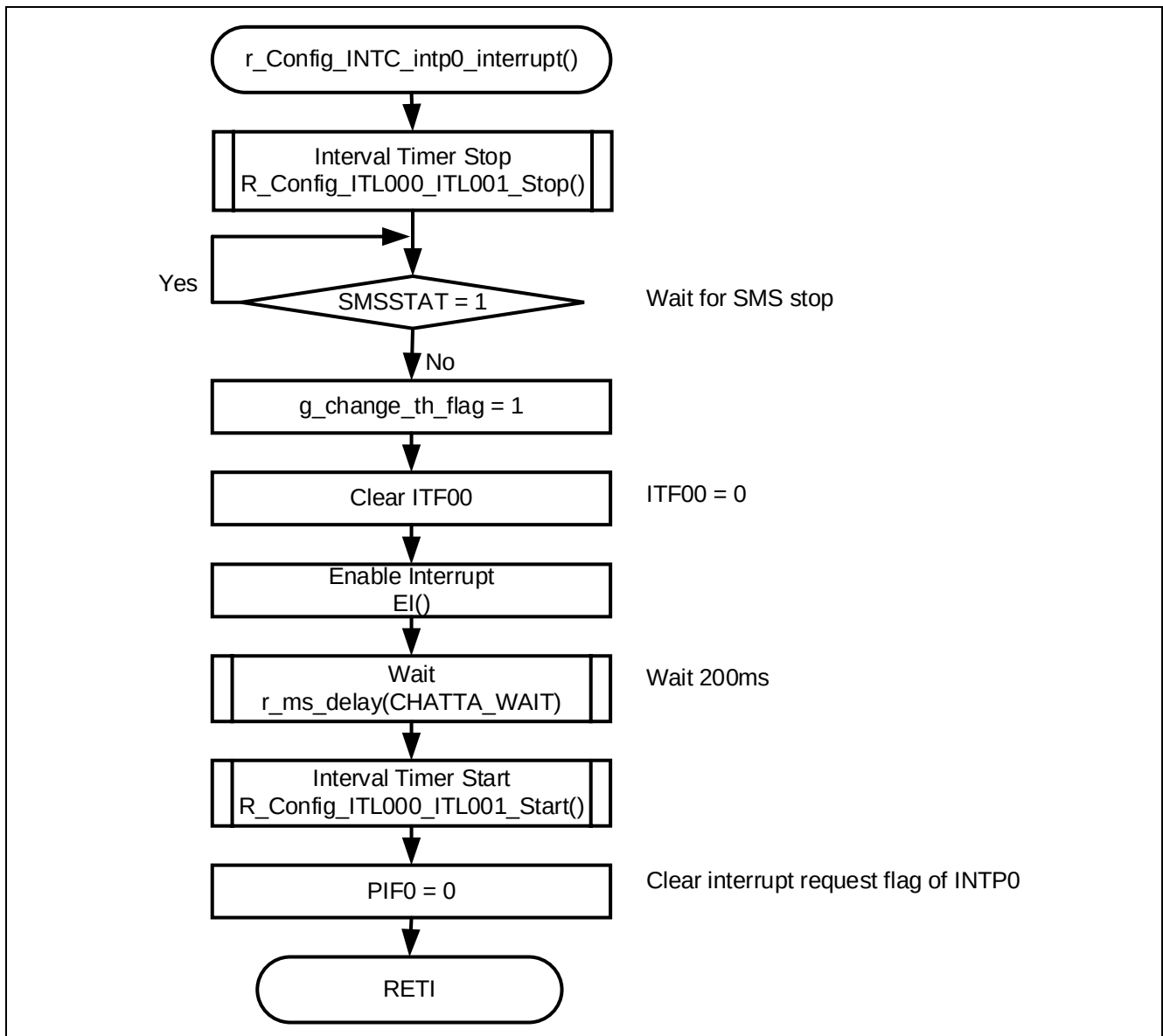
Figure 5-1 Main process



5.8.2 INTP0 interrupt process

Figure 5-2 shows flowchart of INTP0 interrupt process

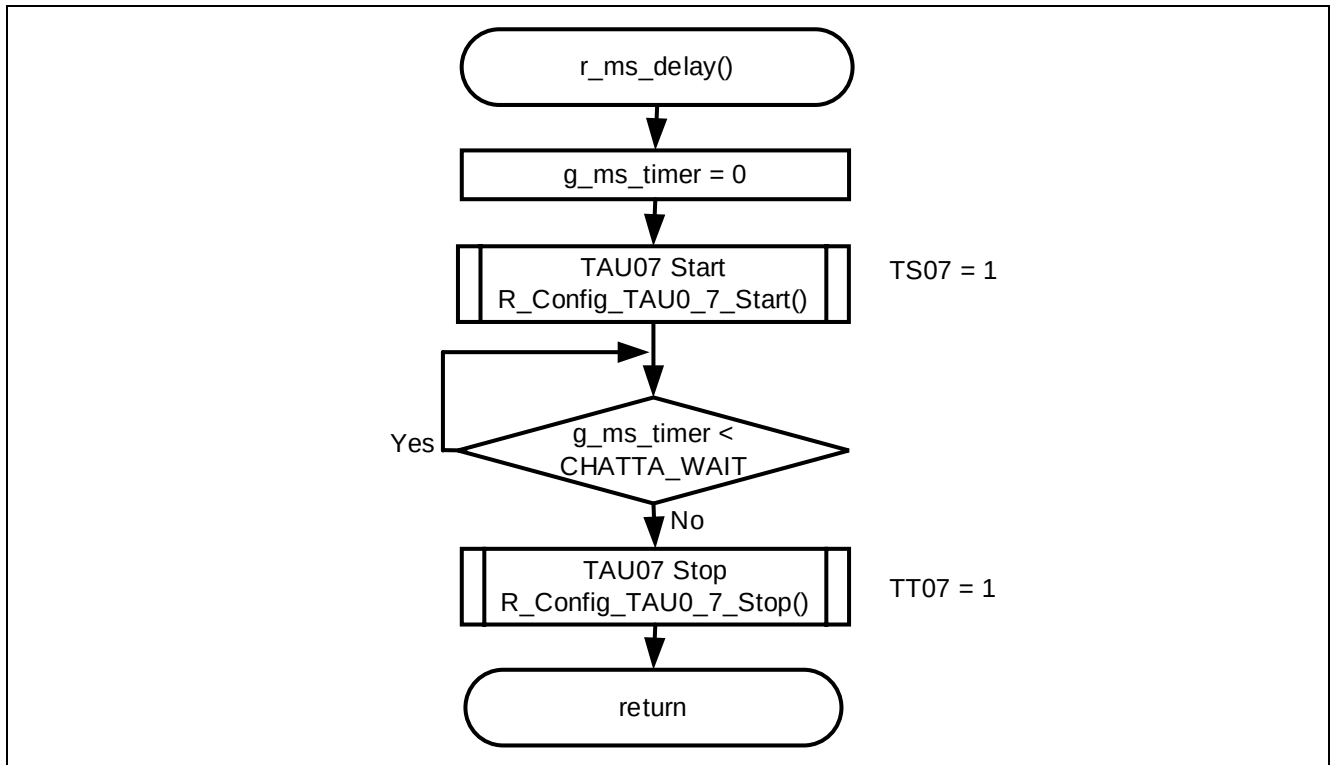
Figure 5-2 INTP0 interrupt process



5.8.3 Wait process

Figure 5-3 shows flowchart of Wait process

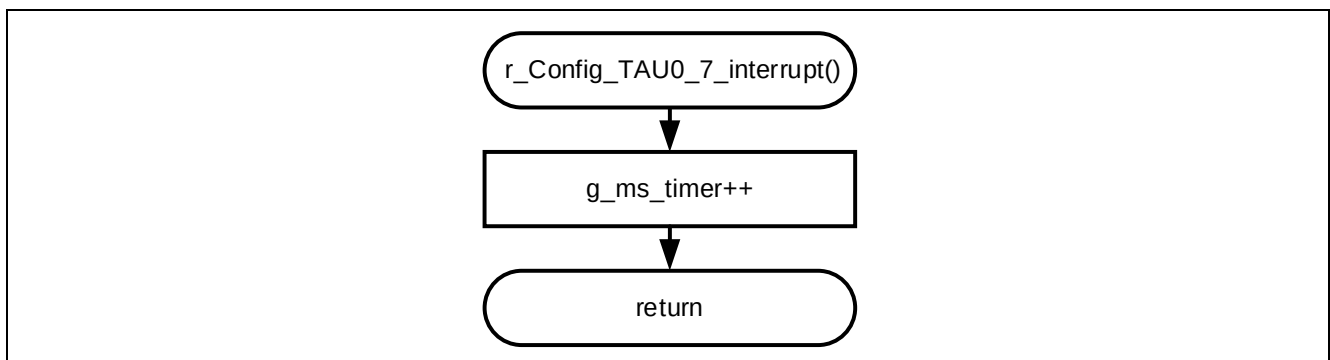
Figure 5-3 Wait process



5.8.4 TAU0 channel 7 interrupt process

Figure 5-4 shows flowchart of TAU0 channel 7 interrupt process

Figure 5-4 TAU0 channel 7 interrupt process



5.9 SNOOZE Mode Sequencer settings

When the event set in the start trigger occurs, SMS executes the processing commands stored in the sequencer instruction register (SMSIO-31) in order. When executing a processing command, the Sequencer general-purpose register (SMSGO-15) is used to store the source address, destination address, calculated data, and so on.

SMSIO-31 and SMSGO-15 are set by writing the SMS program (.SMSASM file) in assembly language. The SMS program can also be created by combining processing blocks using the SNOOZE Mode sequencer component of the Smart Configurator. The created SMS program is converted to a C language file by the SMS assembler and incorporated into the program.

The specifications of SMS processing executed by the sample code are shown below.

Outline	SMS process
Description	SMS is started by a TML32 interrupt, sets the ADC and performs A/D conversion. If the internal reference voltage is selected for the A/D channel, the A/D conversion will be performed twice because the result of the first conversion cannot be used. After A/D conversion, the ADC is stopped and the A/D channel is returned to the initial value. If the conversion result is equal to or higher than the ADCR threshold ^{Note} , the CPU is started, else if it is smaller than the ADCR threshold, the SMS process is finished and shifts to STOP mode again.
Arguments ^{Note1}	addr_adcr_th
Return value	None
Remarks	In this sample code, the process of switching LED1 ON/OFF is added every time the SMS process is completed. The LED1 ON/OFF switching process is valid when "USE_LED" is defined by the .DEFINE command in the Config_SMS_ASM.smsasm file. If you do not want to use LED1, comment out ".DEFINE USE_LED".

Note1. Argument to be specified in the R_Config_SMS_Start function setting. For details, refer to 6.2.1 and 6.2.12.

Note. In the sample code, the ADCR threshold is set to the following values.

Initial setting value	0DEDH (V _{DD} =1.7V)
SW1 pressed 1st time	0BD6H (V _{DD} =2.0V)
SW1 pressed 2nd time	0978H (V _{DD} =2.5V)
SW1 pressed 3rd time	07E4H (V _{DD} =3.0V)
SW1 pressed 4th time	06C3H (V _{DD} =3.5V)
SW1 pressed 5th time	05EBH (V _{DD} =4.0V)
SW1 pressed 6th time	0542H (V _{DD} =4.5V)
SW1 pressed 7th time	04BCH (V _{DD} =5.0V)

Figure 5-5 shows the flow chart of SMS process.

Table 5.6 to Table 5.8 show the register settings that control the SNOOZE mode sequencer.

Figure 5-5 SMS process

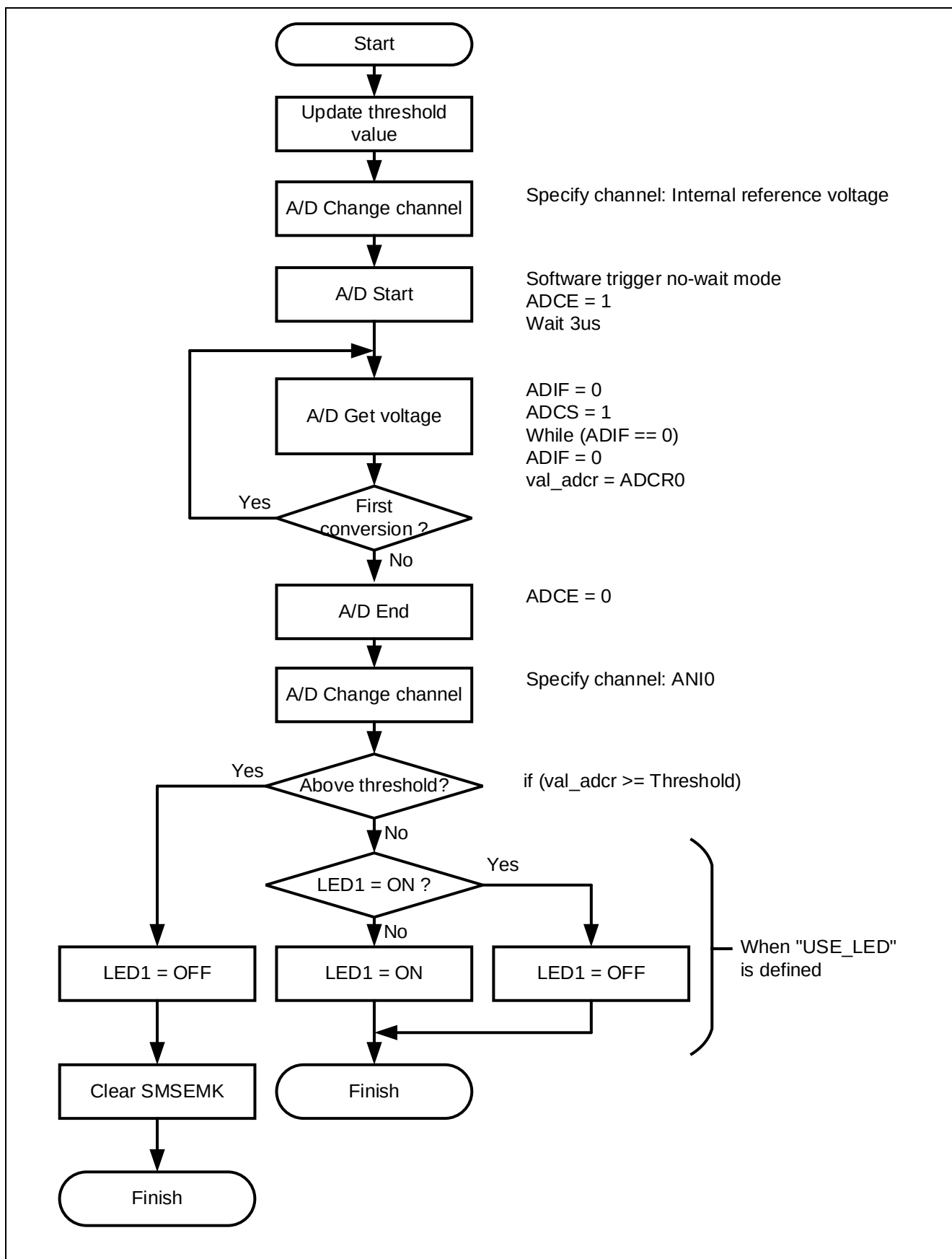


Table 5.6 Sequencer general-purpose registers 0-15

Register Symbol	Setting	Remark
SMSG0	0000H	fixed value: 0000H
SMSG1	0000H	Threshold address of ADCR: val_adcr
SMSG2	0000H	Threshold of ADCR: val_adcr_th
SMSG3	0000H	Variable for storing ADCR value: val_adcr
SMSG4	2	SMS internal variable: val_loop
SMSG5	0xFF05	P5 address
SMSG6	0xFFE5	MK0H address
SMSG7	&ITLS0	ITLS0 address
SMSG8	FF05H	P5 address (Used only when USE_LED is defined)
SMSG9	&ADCR0	ADCR0 address
SMSG10	&ADM0	ADM0 address
SMSG11	&IF1H	IF1H address
SMSG12	129	fixed value: 129
SMSG13	1	fixed value: 1
SMSG14	&smsInitValue0	Address to store the initial value of the variable (initial value: 2)
SMSG15	FFFFH	fixed value: FFFFH

Table 5.7 Sequencer instruction registers 0-31 (When USE_LED is defined)

Register Symbol	Setting	Remark
SMSI0	3E40H	MOVW SMSG4, [MSG14+0]
SMSI1	0700H	MOV [MSG7+0], MSG0
SMSI2	3120H	MOVW MSG2, [MSG1+0]
SMSI3	0AC1H	MOV [MSG10+1], MSG12
SMSI4	4A00H	SET1 [MSG10+0].0
SMSI5	9600H	WAIT 96, 0
SMSI6	5B00H	CLR1 [MSG11+0].0
SMSI7	4A70H	SET1 [MSG10+0].7
SMSI8	BB00H	WHILE0 [MSG11+0].0
SMSI9	5B00H	CLR1 [MSG11+0].0
SMSI10	3930H	MOVW MSG3, [MSG9+0]
SMSI11	74D1H	SUBW MSG4, MSG13
SMSI12	7402H	CMPW MSG4, MSG0
SMSI13	8F93H	BNZ \$ __updbbranch_do
SMSI14	5A00H	CLR1 [MSG10+0].0
SMSI15	0A01H	MOV [MSG10+1], MSG0
SMSI16	7322H	CMPW MSG3, MSG2
SMSI17	8081H	BNC \$ __endbranch_yes
SMSI18	6530H	MOV1 SCY, [MSG5+0].3
SMSI19	8041H	BNC \$ __bitbranch_yes
SMSI20	5830H	CLR1 [MSG8+0].3
SMSI21	7002H	CMPW MSG0, MSG0
SMSI22	8022H	BZ \$ __bitbranch_finish
SMSI23	4830H	SET1[MSG8+].3
SMSI24	F000H	FINISH
SMSI25	4830H	SET1 [MSG8+0].3
SMSI26	5640H	CLR1 [MSG6+0].4
SMSI27	F000H	FINISH
SMSI28-31	0000H	Unused

Table 5.8 Sequencer instruction registers 0-31 (When USE_LED is undefined)

Register Symbol	Setting	Remark
SMSI0	3E40H	MOVW SMSG4, [SMSG14+0]
SMSI1	0700H	MOV [SMSG7+0], SMSG0
SMSI2	3120H	MOVW SMSG2, [SMSG1+0]
SMSI3	0AC1H	MOV [SMSG10+1], SMSG12
SMSI4	4A00H	SET1 [SMSG10+0].0
SMSI5	9600H	WAIT 96, 0
SMSI6	5B00H	CLR1 [SMSG11+0].0
SMSI7	4A70H	SET1 [SMSG10+0].7
SMSI8	BB00H	WHILE0 [SMSG11+0].0
SMSI9	5B00H	CLR1 [SMSG11+0].0
SMSI10	3930H	MOVW SMSG3, [SMSG9+0]
SMSI11	74D1H	SUBW SMSG4, SMSG13
SMSI12	7402H	CMPW SMSG4, SMSG0
SMSI13	8F93H	BNZ \$__updbbranch_do
SMSI14	5A00H	CLR1 [SMSG10+0].0
SMSI15	0A01H	MOV [SMSG10+1], SMSG0
SMSI16	7322H	CMPW SMSG3, SMSG2
SMSI17	8081H	BNC \$__endbranch_yes
SMSI18	6530H	MOV1 SCY, [SMSG5+0].3
SMSI19	8041H	BNC \$__bitbranch_yes
SMSI20	5830H	CLR1 [SMSG8+0].3
SMSI21	7002H	CMPW SMSG0, SMSG0
SMSI22	8022H	BZ \$__bitbranch_finish
SMSI23	4830H	SET1[SMSG8+].3
SMSI24	F000H	FINISH
SMSI25	4830H	SET1 [SMSG8+0].3
SMSI26	5640H	CLR1 [SMSG6+0].4
SMSI27	F000H	FINISH
SMSI28-31	0000H	Unused

6. Application example

In addition to the sample code, this application note stores the following Smart Configurator configuration files.

r01an5605_sms_power_monitoring.scfg

r01an5605_sms_power_monitoring.sms

The following is a description of the file and setting examples and precautions for use.

6.1 r01an5605_sms_power_monitoring.scfg

This is the Smart Configurator configuration file used in the sample code. It contains all the features configured in the Smart Configurator. The sample code settings are as follows.

Table 6.1 Parameters of Smart Configurator

Tag name	Components	Contents
Clocks	-	Operation mod: High-speed main mode 2.4 (V) ~ 5.5 (V) EV _{DD} setting: $1.8V \leq EV_{DD0} < 5.5V$ High-speed on-chip oscillator: 32MHz f _{IHP} : 32MHz f _{CLK} : 32000kHz (High-speed on-chip oscillator) f _{SXP} : 32.768kHz (Low-speed on-chip oscillator)
System	-	On-chip debug operation setting: COM port ^{Note 1} Pseudo-RRM/DMM function setting: Used Start/Stop function setting: Unused Trace function setting: Used Security ID setting: Use security ID Security ID : 0x00000000000000000000 Security ID authentication failure setting: Do not erase flash memory data
Components	r_bsp	Start up select : Enable (use BSP startup) Control of invalid memory access detection : Disable RAM guard space (GRAM0-1) : Disabled Guard of control registers of port function (GPORT) : Disabled Guard of registers of interrupt function (GINT) : Disabled Guard of control registers of clock control function, voltage detector, and RAM parity error detection function (GCSC) : Disabled Data flash access control (DFLEN) : Disables Initialization of peripheral functions by Code Generator/Smart Configurator : Enable API functions disable : Enable Parameter check enable : Enable Setting for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode : High-speed Enable user warm start callback (PRE) : Unused Enable user warm start callback (POST) : Unused Watchdog Timer refresh enable : Unused
	Config_LVD0	Operation mode setting: Reset mode Voltage detection setting: Reset generation level (V _{LVD0}): 1.835 (V)

Table 6.2 Parameters of Smart Configurator

Tag name	Components	Contents
Components	Config_INTC	INTP0 setting: use Valid edge: Falling edge Priority: Level 3
	Config_ITL000_ITL001	Components: Interval timer Operation mode: 16 bit count mode Resource: ITL000_ITL001 Operation clock: f_{SXP} Clock source: $f_{ITL0}/128$ Interval value: 1000 ms Interrupt setting: unused
	Config_TAU0_7	Components: Interval timer Operating mode: 16 bit count mode Resource: TAU0_7 Operation clock: CK00 Clock source: f_{CLK} Interval value: 1 ms Interrupt setting: use Priority: Level 2
	Config_ADC	Components: A/D Converter Comparator operation setting: Stop Resolution setting: 12 bits VREF (+) setting: V_{DD} VREF (-) setting: V_{SS} Trigger mode setting: Software trigger no-wait mode Operation mode setting: One-shot select mode A/D channel selection: ANI0 Conversion time mode: Normal 1 Conversion time: $66/f_{CLK}$ Conversion result upper/lower bound value setting: Generates an interrupt request (INTAD) when $ADLL \leq ADCRn \leq ADUL$ Upper bound (ADUL) value: 255 Lower bound (ADLL) value: 0 Interrupt setting: unused
	Config_SMS	Components: SNOOZE Mode Sequencer Start trigger: Interval detection interrupt (INTITL)
	Config_PORT	Components: Port Port selection: PORT5 P52: Out (Output 1) P53: Out (Output 1)

Note 1. When using IAR, use the following settings.

On-chip debug operation setting: Use emulator

Emulator setting: E2 Emulator Lite

6.1.1 Clocks

Set the clock used in the sample code.

In this sample code, 32000KHz is set for f_{CLK} and the conversion time mode is set to "Standard 1 ($2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$)" with Config_ADC, so the operation mode is "High-speed main mode 2.4 (V) ~ 5.5 (V)". Note that changing the settings.

6.1.2 System

Set the on-chip debug of the sample code.

"Control of on-chip debug operation" and "Security ID authentication failure setting" affect "On-chip debugging is enabled" in "Table 5.2 Option Byte Settings". Note that changing the settings.

6.1.3 r_bsp

Set the startup of the sample code.

6.1.4 Config_LVD0

Set the power management of the sample code.

Affects "Setting of LVD0" in "Table 5.2 Option Byte Settings". Note that changing the settings

6.1.5 Config_INTC

Set the interrupt used in the sample code.

In the sample code, set an external maskable interrupt (INTP0). When the INTP0 is not used, delete it.

6.1.6 Config_IT000_ITL001

Initialize the interval timer for the sample code.

The interval timer interrupt (INTITL) is used to start the SMS in the sample code. Therefore, "Interrupt setting" is set to "Not used". "Interrupt Settings" can also be changed to "Use".

Since INTITL is masked by the R_Config_SMS_Start function, the CPU will not start even if INTITL is generated during STOP or SNOOZE mode. After returning from STOP mode and SNOOZE mode, INTITL is in a masked state, so unmask INTITL if necessary.

6.1.7 Config_TAU0_7

Set TAU07 of the sample code.

In the sample code, it is used as a chattering prevention against INTP0. When the INTP0 is not used, or chattering prevention is not needed, delete it.

6.1.8 Config_ADC

Initialize the ADC for the sample code.

In the sample code, "VREF (+) setting" is set to V_{DD} and "A/D channel selection" is set to ANI0. It is also possible to change "A/D channel selection" to another ANI pin. And "the internal reference voltage" or "the temperature sensor output voltage" can be selected too. However, the A/D converter reference voltage current and temperature sensor operating current will flow during STOP mode in this case.

In the sample code, A/D conversion is not performed when the device is not in SNOOZE mode, so "Interrupt Settings" is set to "Not Used". "Interrupt Settings" can also be changed to "Use". Since INTAD is masked by the R_Config_SMS_Start function, the CPU will not start even if INTAD is generated during STOP or SNOOZE mode. After returning from STOP mode and SNOOZE mode, INTAD is in a masked state, so unmask INTAD if necessary.

6.1.9 Config_SMS

Set the sample code SMS.

For details, refer to "6.2 r01an5605_sms_power_monitoring.sms".

6.1.10 Config_PORT

Set the port of the sample code.

In the sample code, P53 is used to control LED1 and P52 is used to control LED2.

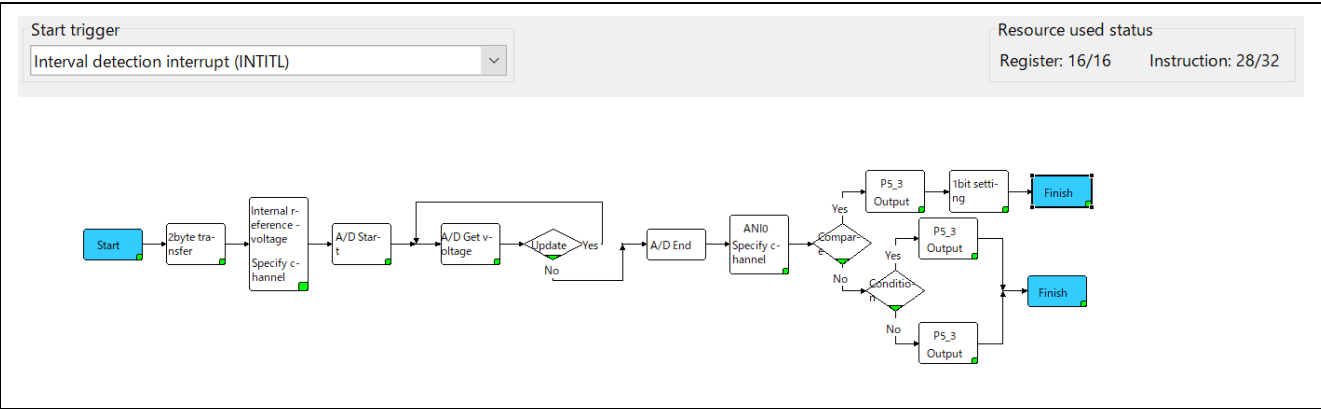
6.2 r01an5605_sms_power_monitoring.sms

This is the data for Config_SMS alone. In the sample code, the interrupt of the interval timer is used to start SMS, and A/D is used in the operation of SMS. Note that it is necessary to set the interval timer and A/D separately.

The r01an5605_sms_power_monitoring.sms can also be imported into the Smart Configurator of another project. After setting up the SMS component in another project, go to [Import SMS Sequence] -> [Browse] and select "r01an5605_sms_power_monitoring.sms" to import it.

When imported into the smart configurator, the flow chart will be as shown in Figure 6-1. This flow chart is the same as "Figure 5-5 SMS process".

Figure 6-1 Config_SMS flow chart

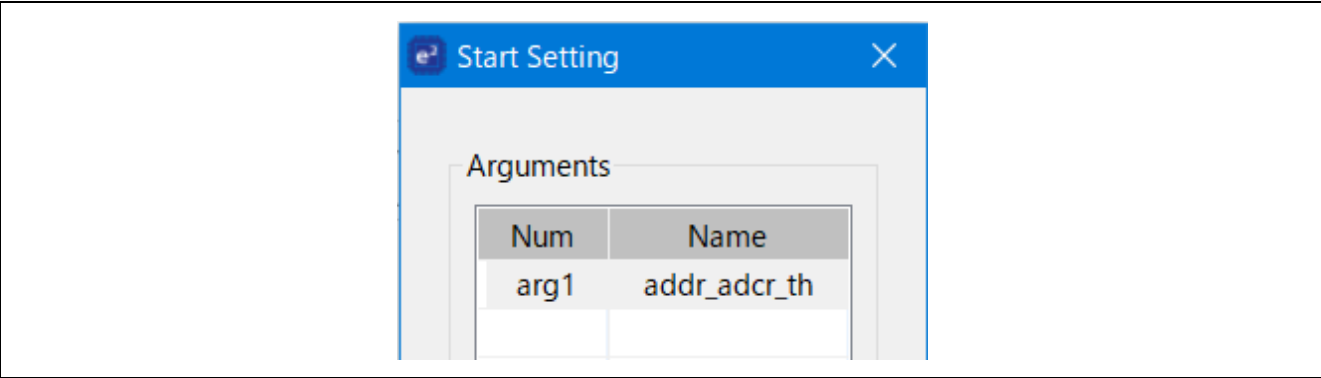


A description of each block is shown below.

6.2.1 Start

When the SMS starts, the address of g_adcr_th[0] passed as an argument in the SMS start function (R_Config_SMS_Start function) is set to addr_adcr_th (ADCR threshold address).

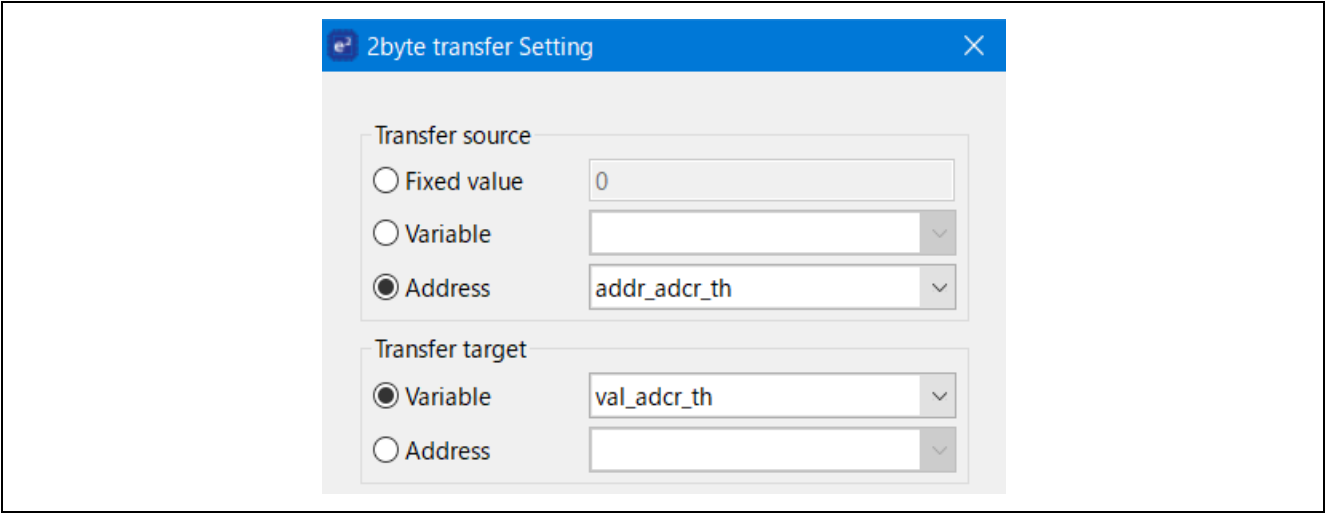
Figure 6-2 Start Setting



6.2.2 2byte transfer

Store the value of the argument `addr_adcr_th` (ADCR threshold address) in the transfer target `val_adcr_th`.

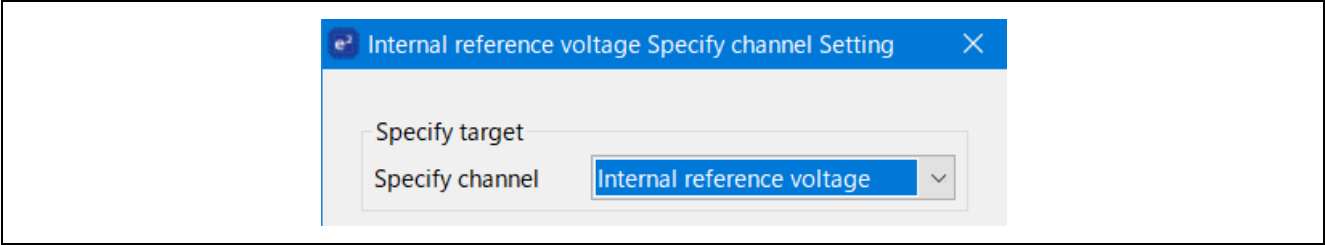
Figure 6-3 2byte transfer Setting



6.2.3 A/D Change channel

Set "Internal reference voltage" as the channel specification for A/D conversion. A/D conversion can also be targeted to other channels.

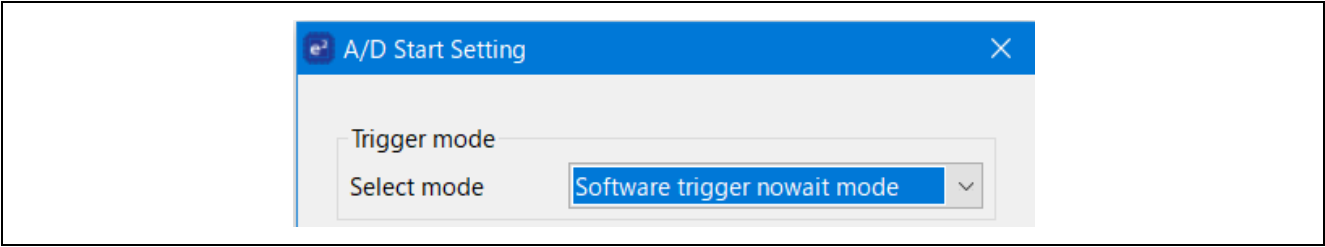
Figure 6-4 A/D Change channel Setting



6.2.4 A/D Start

Set the A/D trigger mode. Waiting time is automatically added according to the mode.

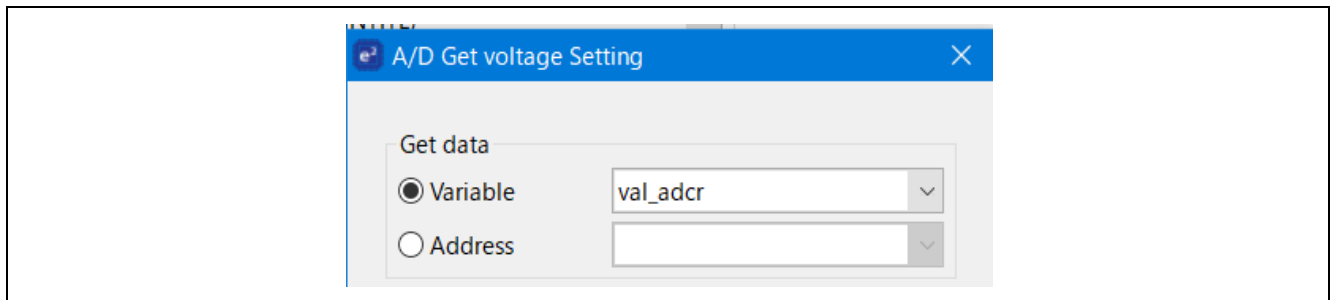
Figure 6-5 A/D Start Setting



6.2.5 A/D Get voltage

Converts A/D and stores the value of the A/D conversion result (ADCR0) in the variable val_adcr.

Figure 6-6 A/D Get voltage Setting



6.2.6 Update

In the sample code, "Internal reference voltage output" is selected as the target of A/D conversion, the first A/D conversion result cannot be used, and A/D conversion is performed twice.

The value "2" set in the variable val_loop is subtracted by the fixed value "1" and this value is compared with the comparison value "0" by the comparison expression (val_loop != comparison value). If the value does not match the comparison value, it returns to operation of getting AD conversion result, at which time val_loop is overwritten with the updated value "1". If the value does match the comparison value, then proceed.

When selecting an A/D conversion target other than "internal reference voltage", A/D conversion can be performed only once.

Figure 6-7 Update Setting

Update Setting

Target value

☒ Variable

val_loop

☐ Address

Update value

☒ Fixed value

1

☐ Variable

☐ Address

Update formula

☐ +(plus)

☒ -(minus)

Comparison value

☒ Fixed value

0

☐ Variable

☐ Address

Comparison formula

Condition

Target value != Comparison value

6.2.7 A/D End

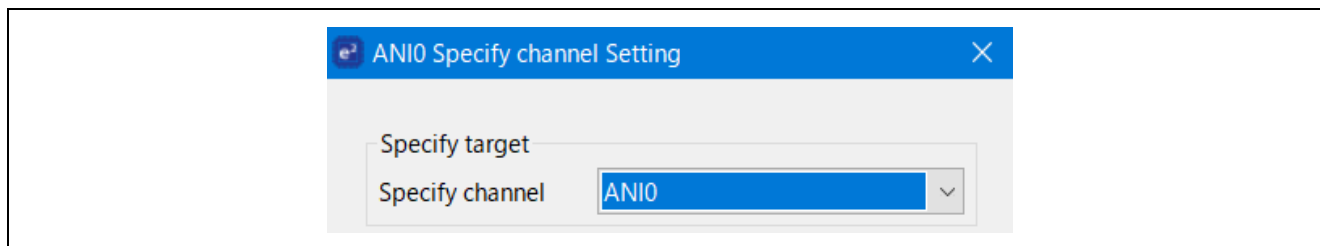
End the A/D conversion.

6.2.8 A/D Change channel

Change the A/D conversion target to ANI0.

Since ADISS = 1 must not be set when shifting to STOP mode, change the AD conversion target from "Internal reference voltage" to "ANI0". Any channel from ANI0 to ANI19 can be changed.

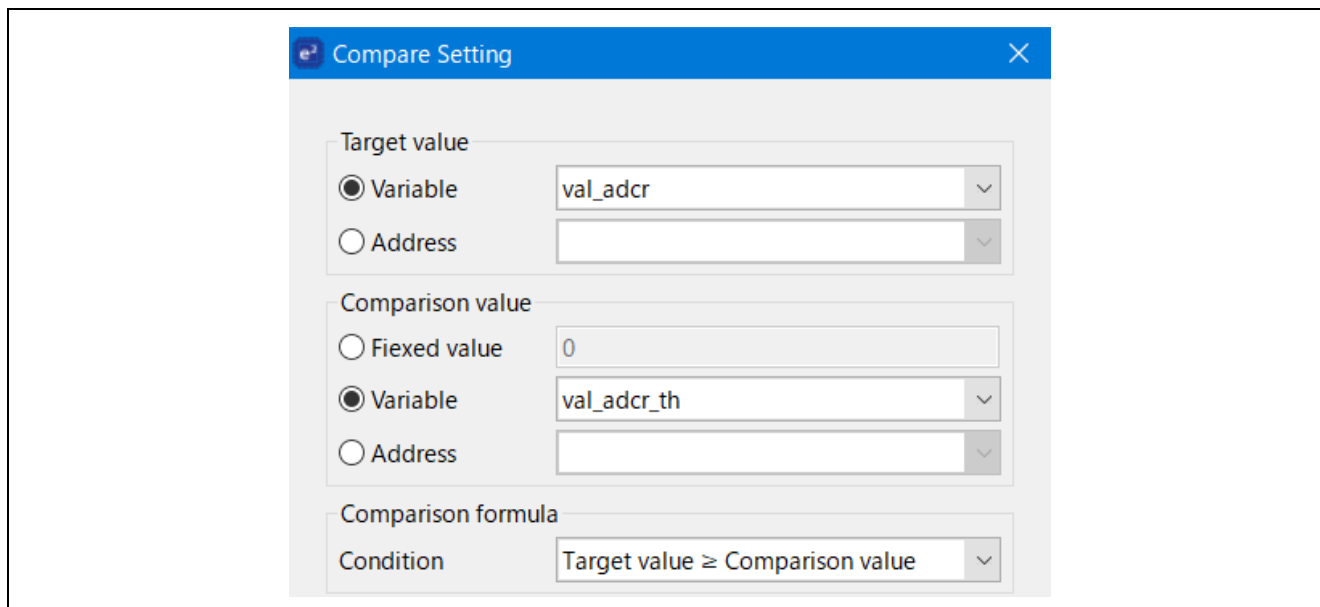
Figure 6-8 A/D Change channel Setting



6.2.9 Compare

This function compares whether the A/D conversion result stored in the variable val_adcr is greater than the threshold value stored in the variable val_adcr_th. If val_adcr is greater than the threshold value, it judges that the power supply voltage is less than the arbitrary value and returns to the CPU operation mode from the SNOOZE mode. val_adcr is less than the threshold value, the supply voltage is judged to be greater than or equal to the arbitrary value, and the mode shifts to STOP mode.

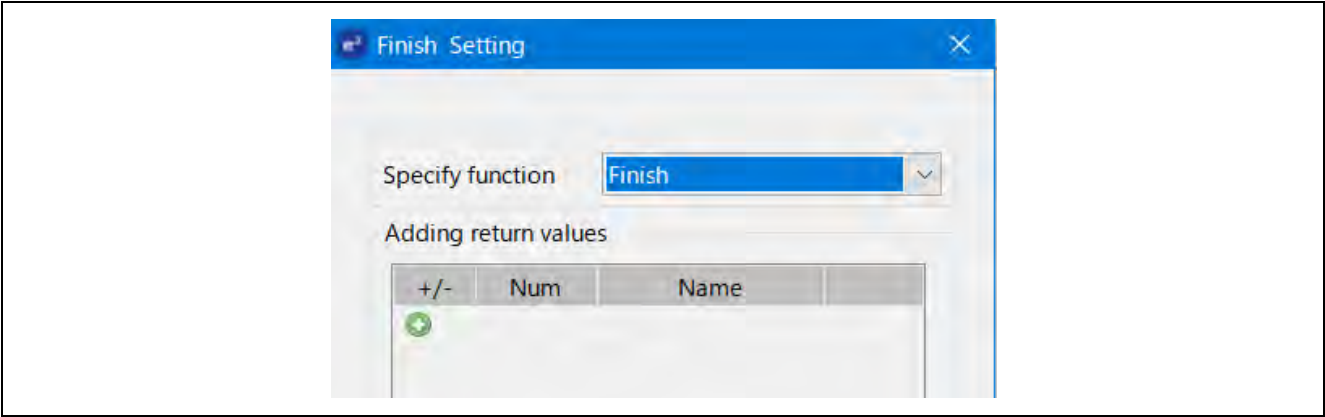
Figure 6-9 Compare Setting



6.2.10 Finish

It shifts to STOP mode. In the sample code, the return value is not used.

Figure 6-10 Finish Setting



6.2.11 Variable Setting

The settings of the variables used in SMS are shown below.

Table 6.3 Variables used in SMS

Data name	Initialization mode	Initial value	Description
val_adcr	No initialization	-	Stores the A/D conversion result.
val_adcr_th	No initialization	-	Stores the threshold.
val_loop	Initialize every time with SMS	2	This sets the number of loops. In the sample code, it loops twice.
addr_adcr_th	Pass argument via SMS start function	-	Stores the address of the threshold. The address of g_adcr_th[0] is set as an argument in the R_Config_SMS_Start function.

7. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

Furthermore, if you regenerate the code with Smart Configurator, please modify the R_Config_SMS_Start function as indicated in the red frame below.

```
void R_Config_SMS_Start(uint16_t addr_adcr_th)
{
    /* Set the sms data from arguments */
    SMSG1 = addr_adcr_th;
    /* Initialize SMS data */
    SMSG6 = 65509U;
    SMSG5 = 65285U;
    /* Disable related interrupts */
    ITLMK = 1U;
    ADMK = 1U;
    /* Start sms */
    //SMSEIF = 0U; /* clear INTSMSE interrupt flag */
    //SMSEMK = 0U; /* enable INTSMSE interrupt */
    SMSEMK = 1U; /* disable INTSMSE interrupt */
    SMSEIF = 1U; /* set INTSMSE interrupt flag */
    g_sms_wakeup_flag = 0U;
    ITLS0 = _00_INTITL_CLEAR;
    SMSSTART = 1U;
}
```

8. Reference

RL78/G23 User's Manual: Hardware (R01UH0896E)

RL78 Family User's Manual: Software (R01US0015E)

SMS assembler User's Manual (R20UT4792J)

RL78 Smart Configurator User's Guide: CS+ (R20AN0580E)

RL78 Smart Configurator User's Guide: e² studio (R20AN0579E)

RL78 Smart Configurator User's Guide: IAREW (R20AN0581E)

(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.)

All trademarks and registered trademarks are the property of their respective owners.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Apr.13.21	-	First edition
1.10	Jun.1.21	6	Updated tool version Table 2-1 Operation Confirmation Conditions Integrated development environment (CS+) : E8.05.00f -> V8.05.00 C compiler (CS+) : V1.09.00 -> V1.10 Integrated development environment (e ² studio) : 2021-01 (21.01.0) -> 2021-04 (21.4.0) C compiler (e ² studio) : V1.09.00 -> V1.10 Integrated development environment (IAR) : V4.20.1 -> V4.21.1 Smart Configurator : V.1.0.0 -> V.1.0.1 Board support package (r_bsp) : V.1.0.0 -> V.1.10
		6, 8 22, 23	Changed due to COM port support Table 2-1 Operation Confirmation Conditions Emulator: E2 Emulator Lite -> CS+, e ² studio: COM port IAR: E2 Emulator Lite Figure 4-1 Hardware Configuration Added P11/TOOLRxD and P12/TOOLTxD Table 6-1 Smart Configurator Settings Note 1 added
		10, 12	Changed due to IAR version sample code update Table 5-1 Folder configuration Added the note about reference documents in folder configuration 5.7 Function Specifications [Function name] main, Header e ² studio, CS+ : r_smc_entry.h IAR : ior7f100g.h, ior7f100g_ext.h, r_cg_macrodriver.h, Config_SMS.h, Config_ITL000_ITL001.h, Config_INTC.h -> r_smc_entry.h
		22, 23	Changed clock abbreviation Table 6-1 Parameters of Smart Configurator Clocks : f _{SXL} -> f _{SXP} Table 6-2 Parameters of Smart Configurator Operation clock : f _{SXL} -> f _{SXP}
		32	Added of RL78 Smart Configurator User's Guide 8. Reference RL78 Smart Configurator User's Guide: CS+ (R20AN0580E) RL78 Smart Configurator User's Guide: e ² studio (R20AN0579E) RL78 Smart Configurator User's Guide: IAREW (R20AN0581E)
1.20	Jan.9.24	-	Changed the flowchart for SMS processing.

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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. 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.5.0-1 October 2020)

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/.