RL78/G23

SMS Button Long Press/Short Press Judgment

Introduction

This application note describes how to build a button long press/short press judgment system using the SNOOZE mode sequencer.

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

This application note shows how to measure the button (SW1) press time and judge long press/short press and start the CPU when the judgment result is long press, in SMS processing.

Set in advance the process that SMS judges as long pressed if SW1 is pressed for the specified time or longer, and short pressed otherwise. After shifting to STOP mode, when SW1 is pressed, INTP0 is detected and SMS is started via ELCL. SMS executes the long press/short press judgment process for SW1, and if it is long pressed, it issues a WAKEUP request (INTSMSE) from SMS and starts the CPU.

Figure 1-1 shows an example of the system configuration, and Figure 1-2 shows the flowchart of the entire system.

Figure 1-1 System Configuration

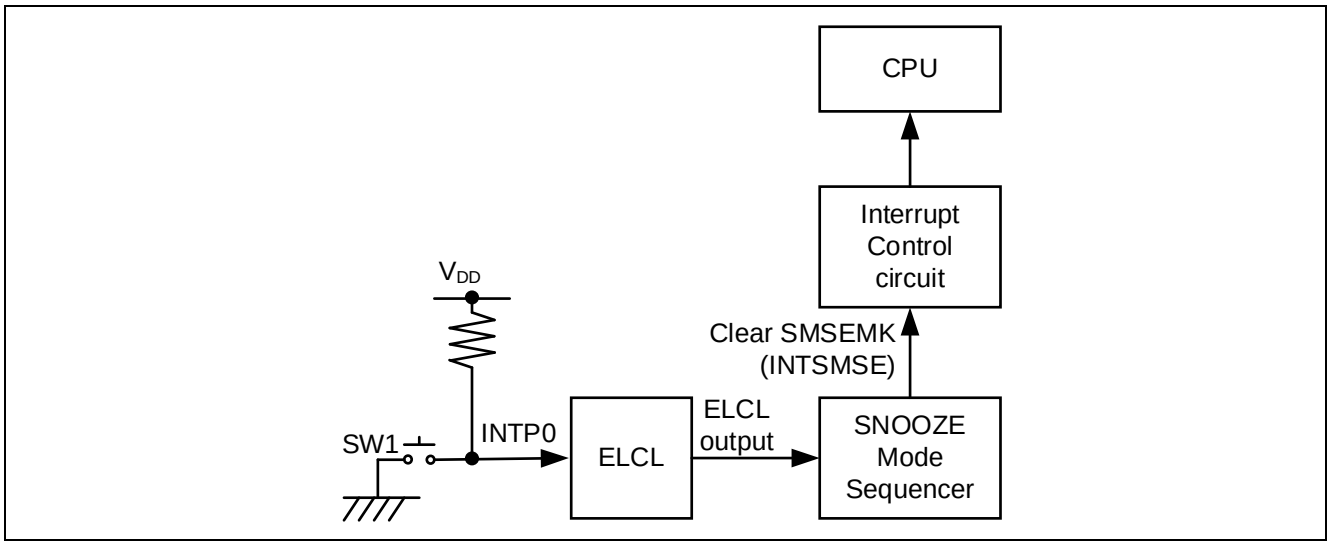
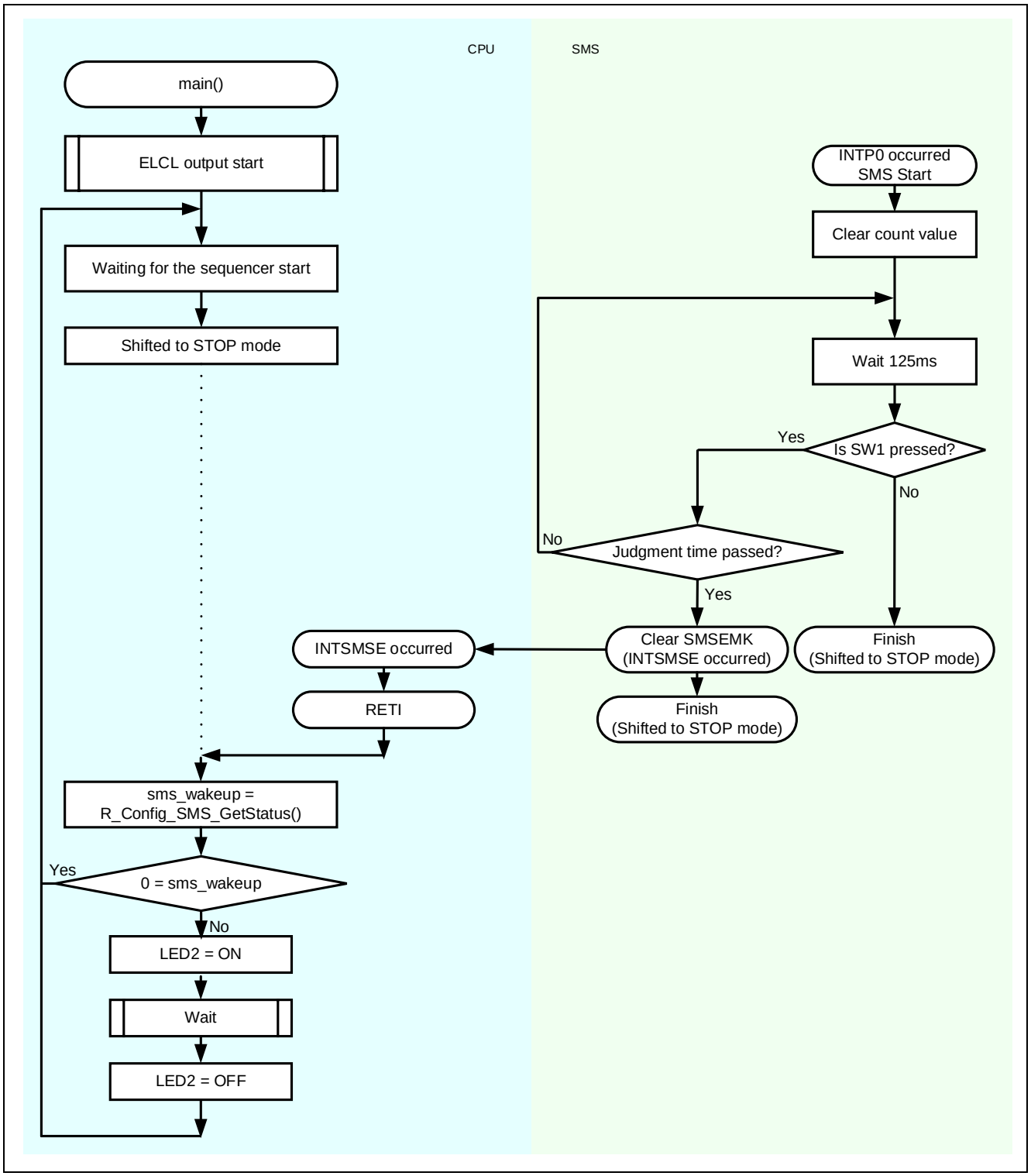


Figure 1-2 Entire Flowchart



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 V8.11.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.13 from Renesas Electronics Corp.
Integrated development environment (e^2 studio)	e^2 studio 2024-04 (24.4.0) from Renesas Electronics Corp.
C compiler (e^2 studio)	CC-RL V1.14 from Renesas Electronics Corp.
Integrated development environment (IAR)	IAR Embedded Workbench for Renesas RL78 v5.10.3 from IAR Systems
C compiler (IAR)	
Smart Configurator	V.1.10.0
Board support package (r_bsp)	V.1.62
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.

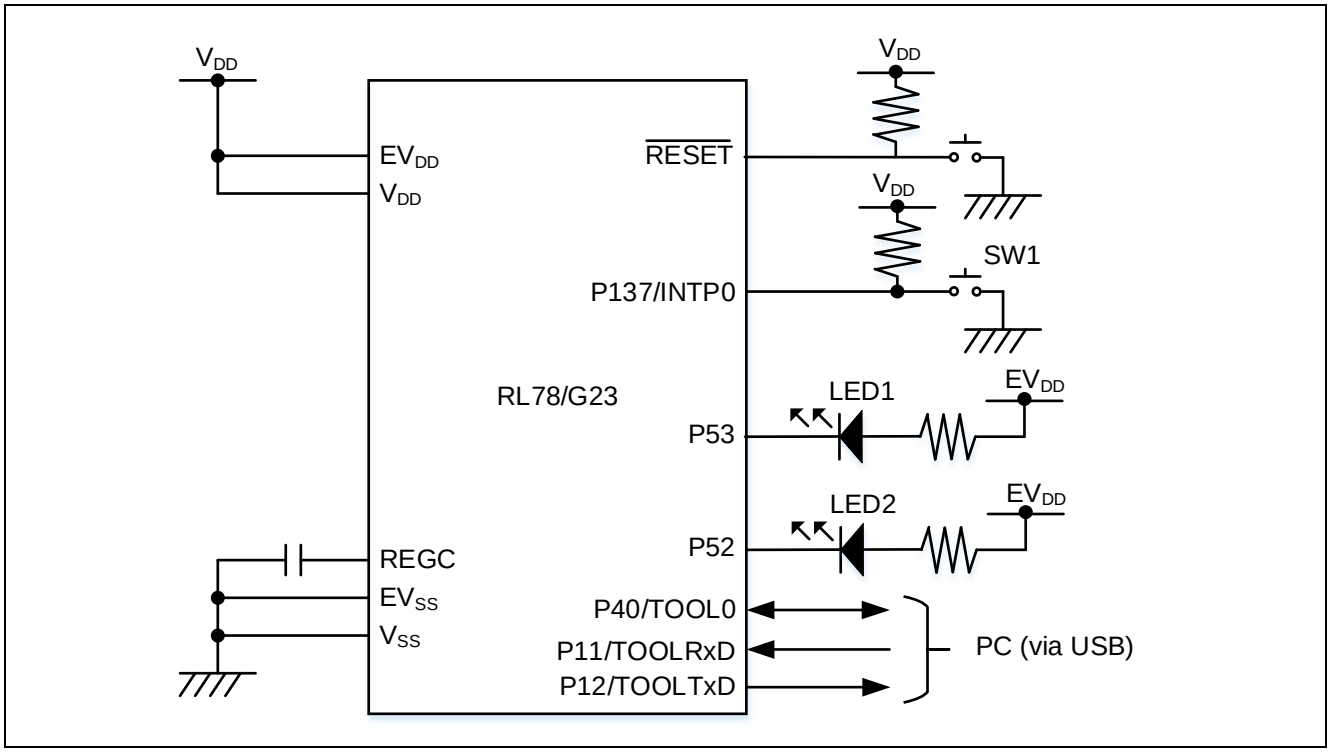
Please refer to them as well.

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

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, after shifting to STOP mode, shifts from STOP mode to SNOOZE mode due to INTP0 generated by pressing SW1. The SNOOZE mode sequencer (SMS) processes the measurement of the SW1 pressed time and the long press/short press judgment.

By selecting INTP0 for the ELCL input, SMS can be started by the occurrence of INTP0 via ELCL. SMS measures the SW1 pressed time, judges that it is a long press if the measurement result is longer than the specified time, shifts from SNOOZE mode to normal operation, and starts the CPU. In addition, mask the INTP0 interrupt to prevent from canceling STOP mode.

LED1 turns on when SMS starts, and turns off when SMS processing is complete. In addition, LED2 turns on for 1 second if SW1 is pressed and held.

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 following is an overview of the processing performed by this sample code.

(1) Sets ELCL, SMS.
(2) Shifts to STOP mode.
(3) By pressing SW1, INTP0 starts SMS via ELCL and shifts to SNOOZE mode.
(4) Turns on LED1.
(5) Measures the pressing time of SW1.
(6) Branches to (7) if the press time does not exceed the specified time (short press), else branches (9).
(7) Turns off LED1.
(8) Returns to (2).
(9) Turns off LED1 and starts the CPU.
(10) Shifts to normal operation from SNOOZE mode.
(11) Turns on LED2 for 1 second.
(12) Returns to (2).

(4) to (9) are processed by SMS.

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
¥r01an5609_sms_push_judgment<DIR>	Root folder of this sample code	
¥src<DIR>	Folder for program source	
main.c	Sample code source file	
¥elcl<DIR>	Folder for ELCL program	
elcl.c	Source file for ELCL	
elcl.h	Header for ELCL	
¥smc_gen<DIR>Note 4	Folder created by Smart Configurator	√
¥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 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 r01an5609_sms_push_judgment.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	11101 111B (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
JUDGMENT_COUNT	24	Number of times to wait 125ms The following time using this constant is the long press judgment time. Long press judgment time = 125ms × this constant In this sample program, the long press judgment time is set to 3 seconds according to the setting value on the left.	main.c
DELAY_MILLSECS	1000	LED2 lighting time (unit: ms)	
LED2	P5_bit.no2	P52	
LED_ON	0	Setting value for turning on the LED	
LED_OFF	1	Setting value for turning off the LED	

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

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_elcl_create	ELCL initialize process	elcl.c
r_elcl_start	ELCL output start process	elcl.c
r_ms_delay	LED2 lighting time wait process	Config_TAU0_7_user.c
r_Config_TAU0_7_interrupt	TAU0 channel 7 interrupt process (For LED wait time count)	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, elcl.h
Declaration	void main (void);
Description	This function initializes ELCL, sets output, and sets the waiting time for long press judgment of SMS, and shifts to STOP mode. LED2 is turned on when returning from SNOOZE mode. After waiting for the lighting time with the r_ms_delay () function, LED2 is turned off and the CPU shifts to STOP mode again.
Arguments	None
Return value	None
Remarks	None

[Function name] r_elcl_create

Outline	ELCL initialize process
Header	r_cg_macrodriver.h、r_cg_userdefine.h、elcl.h
Declaration	void r_elcl_create (void);
Description	This function does the ELCL initial setting. ELCL selects INTP0 as the ELCL output and SMS selects the ELCL output as the start trigger.
Arguments	None
Return value	None
Remarks	None

[Function name] r_elcl_start

Outline	ELCL operation start process
Header	r_cg_macrodriver.h、r_cg_userdefine.h、elcl.h
Declaration	void r_elcl_start (void);
Description	This function enables the ELCL output.
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 msec, completes wait process if more than msec.
Arguments	msec
Return value	None
Remarks	None

[Function name] r_Config_TAU0_7_interrupt

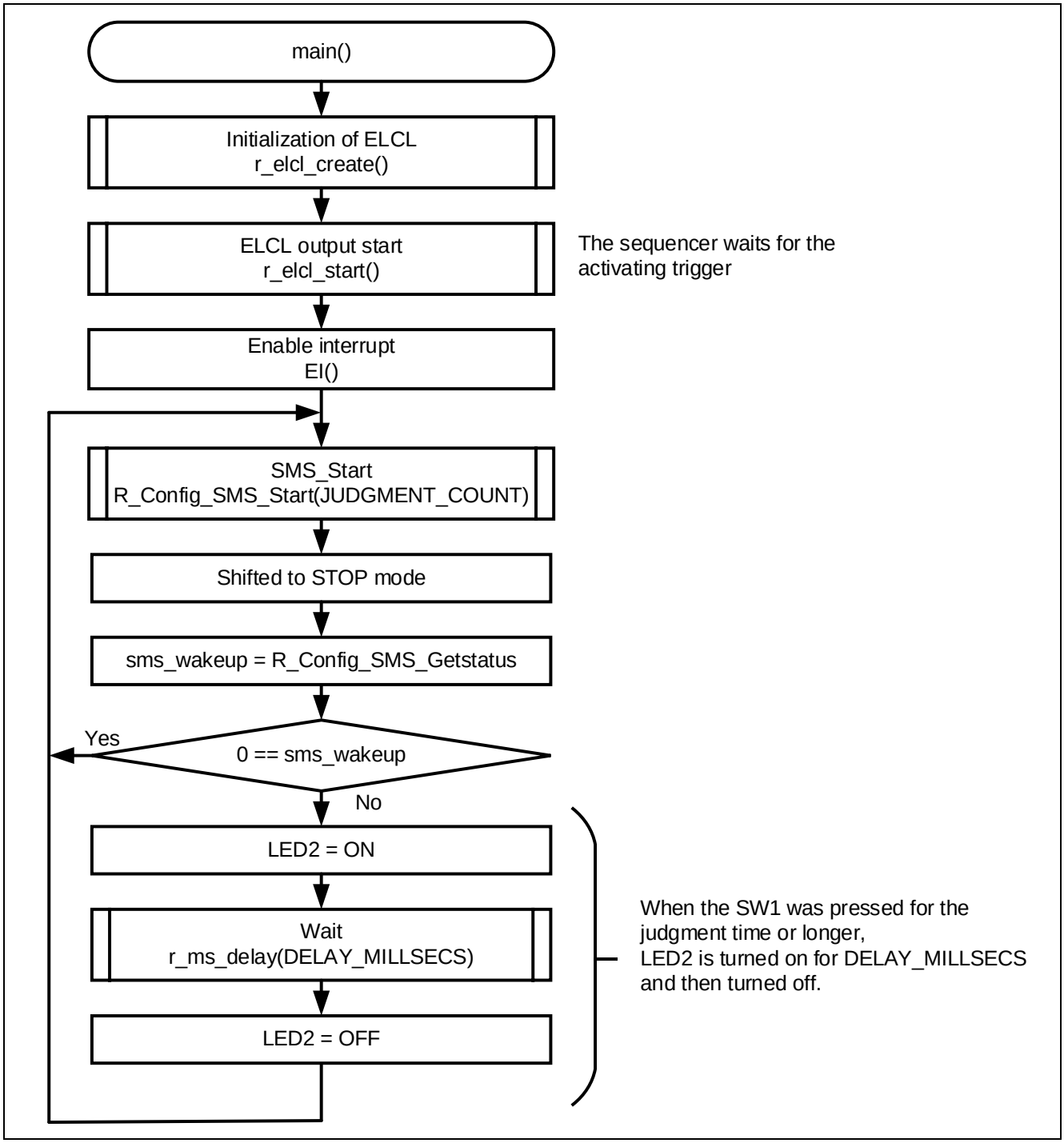
Outline	TAU0 channel 7 interrupt process
Header	r_cg_macrodriver.h, r_cg_userdefine.h, Config_TAU0_7.h
Declaration	#pragma interrupt r_Config_TAU0_7_interrupt (vect=INTTM07)
Description	This function is an interrupt process by INTTM07 on TAU0 channel 7. Counts up g_ms_timer.
Arguments	None
Return value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

5.8 Flow Charts

5.8.1 Main Process

Figure 5-1 shows flowchart of main process.

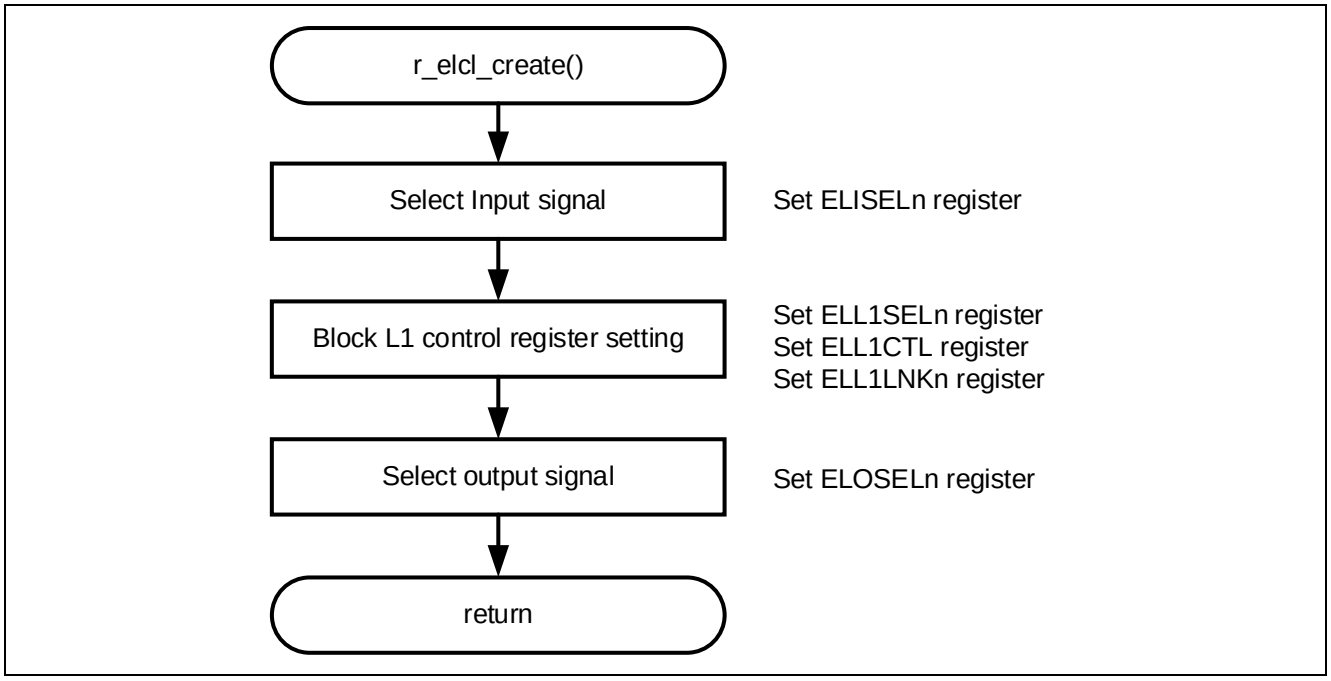
Figure 5-1 Main process



5.8.2 ELCL initialize process

Figure 5-2 shows flowchart of initialize process for ELCL.

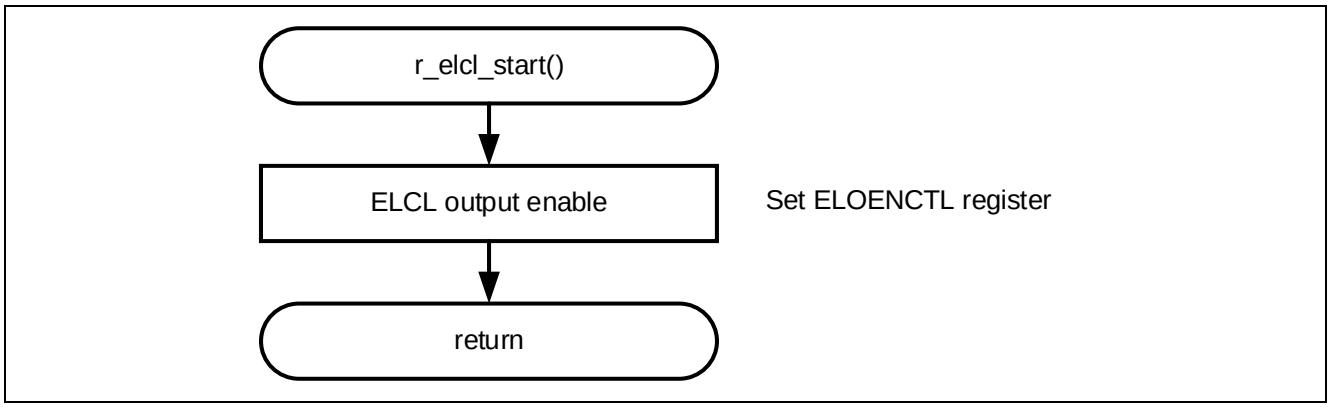
Figure 5-2 ELCL initialize process



5.8.3 ELCL operation start process

Figure 5-3 shows flowchart of operation start process for ELCL.

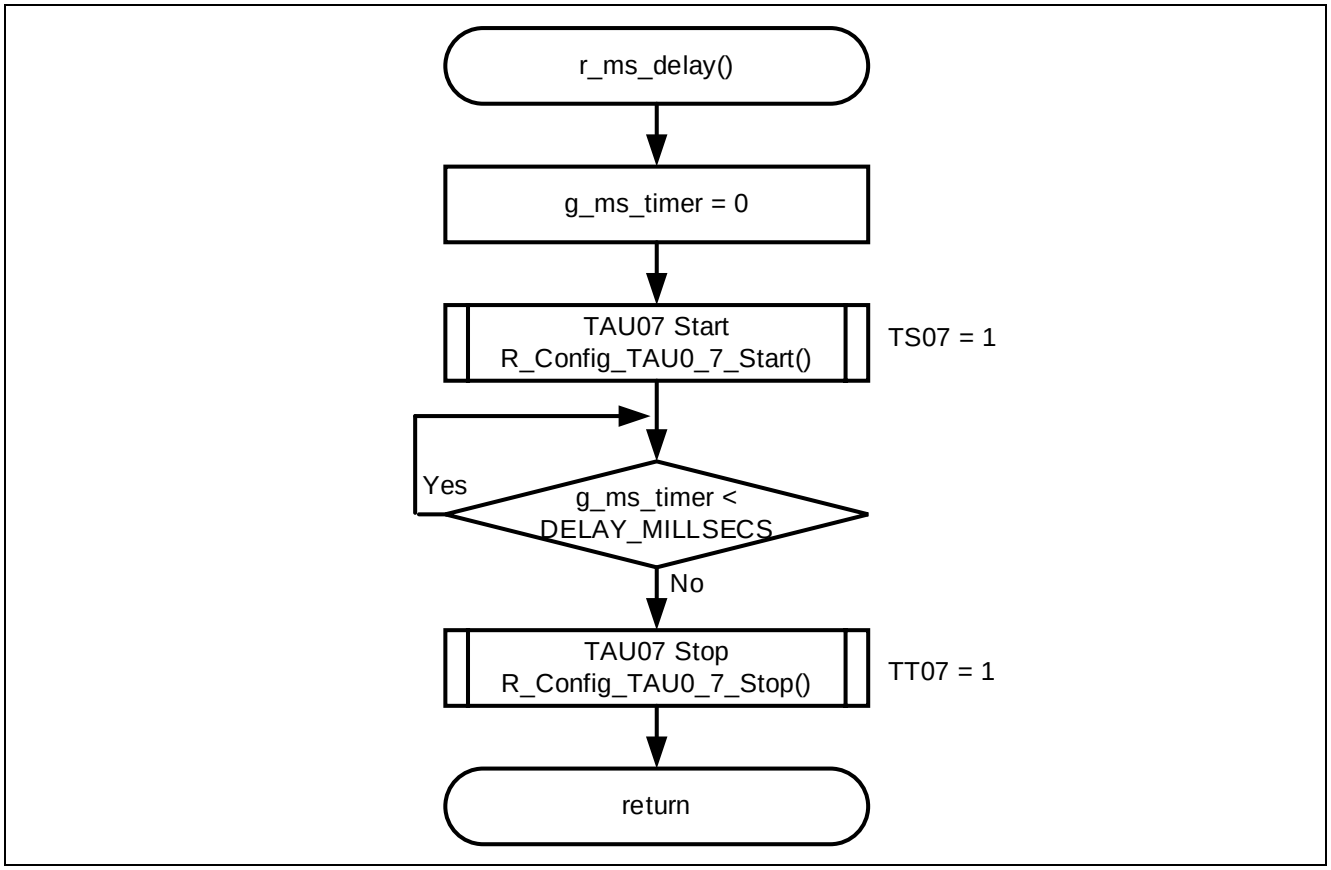
Figure 5-3 ELCL operation start process



5.8.4 Wait process

Figure 5-4 shows flowchart of wait process.

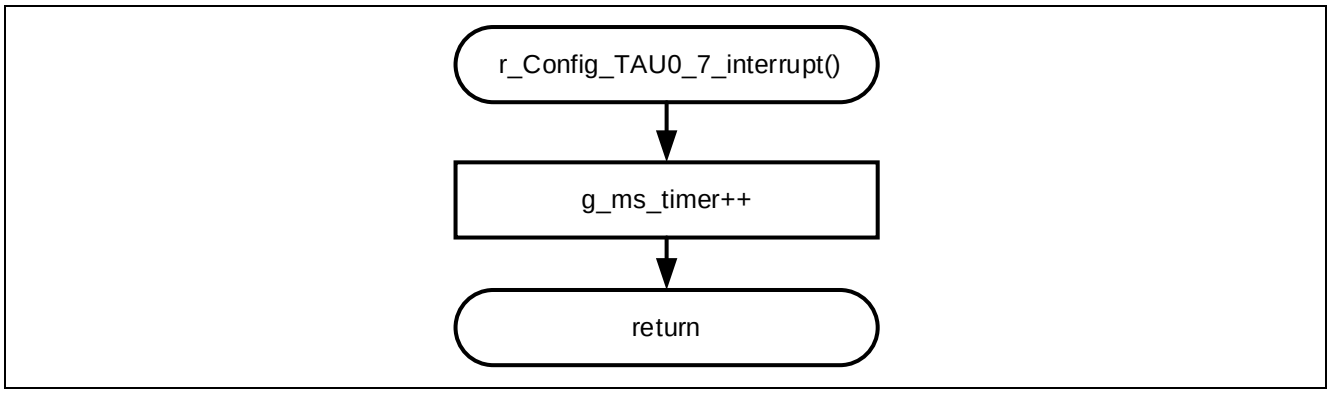
Figure 5-4 Wait process



5.8.5 TAU0 channel 7 interrupt process

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

Figure 5-5 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 (SMSI0-31) in order. When executing a processing command, the Sequencer general-purpose register (SMSG0-15) is used to store the source address, destination address, calculated data, and so on.

SMSI0-31 and SMSG0-15 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	When INTP0 is detected by pressing SW1, SMS is started via ELCL, and the pressing time of SW1 is measured and long press/short press judgment is executed. After starting SMS, wait 125ms and check if SW1 is pressed. If it is not in the pressed state, it is judged as a short press, and the mode returns to STOP mode. If it is pressed, add 1 to the count value of the wait number, and then check if the count value is equal to or higher than the threshold value. If it is above the threshold value, it returns to the 125ms wait process and repeats the series of processes. If it is above the threshold value, it is judged as a long press and INTSMSE is generated. LED1 turns on when SMS starts and turns off after the SMS process is completed.
Arguments Note1	val_cnt_th
Return value	None
Remarks	In this sample code, the process of turning on LED1 during SMS processing is added. 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 6.2.6

Figure 5-6 shows the SMS processing flowchart.
Table 5-6 to Table 5-8 show the register settings that control the SNOOZE mode sequencer.

Figure 5-6 SMS process

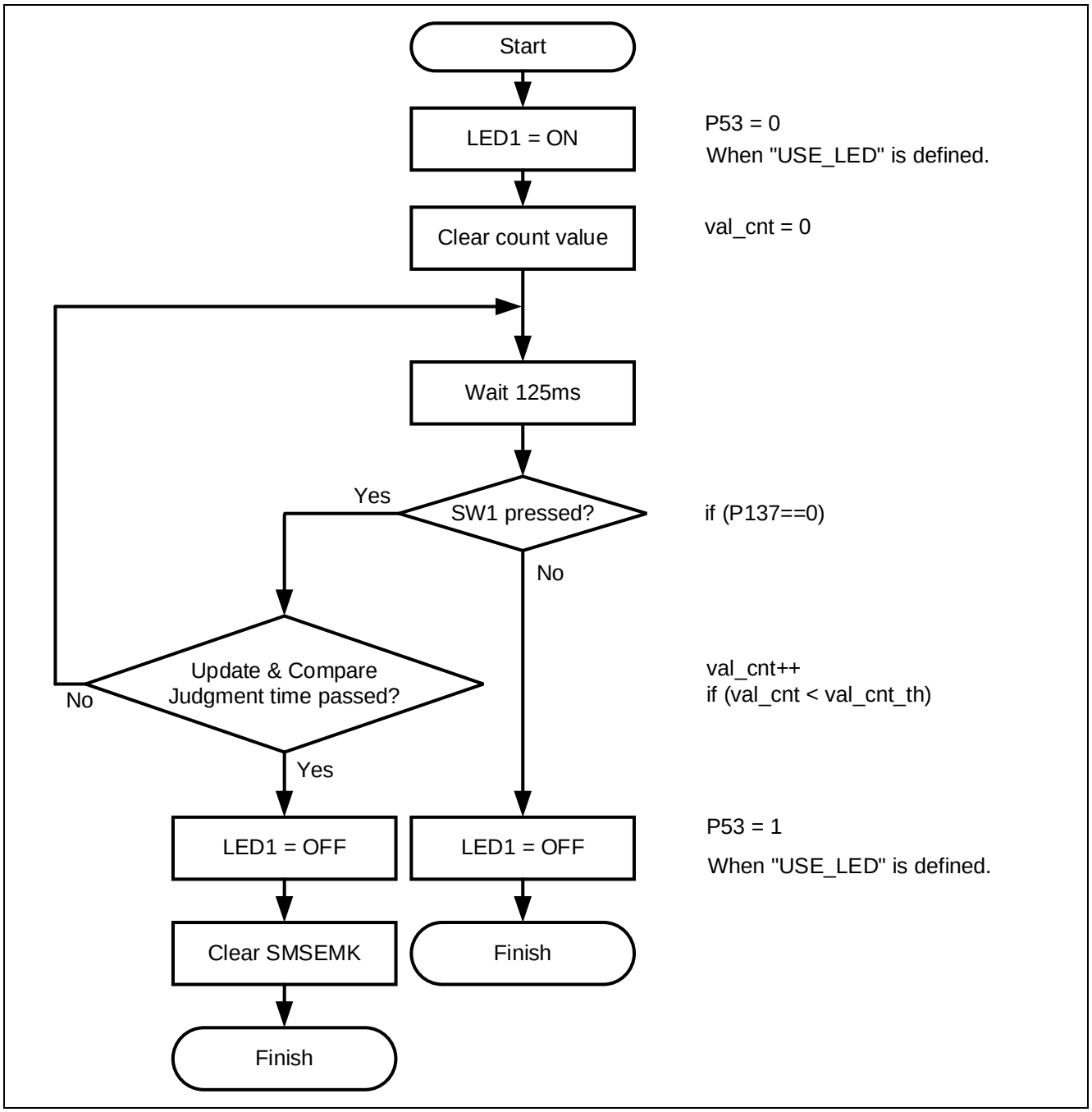


Table 5-6 Sequencer general-purpose registers 0-15

Register Symbol	Setting	Remark
SMSG0	0000H	fixed value: 0000H
SMSG1	0	Wait number count value: val_cnt
SMSG2	0000H	Threshold of wait number count value: val_cnt_th
SMSG3	&P13	P13 address
SMSG4	0xFFE5	MK0H address
SMSG5	1	fixed value: 1
SMSG6	&smsInitValue0	Address to store the initial value of the variable (initial value: 0)
SMSG7	0000H	unused
SMSG8	0000H	unused
SMSG9	0000H	unused
SMSG10	0000H	unused
SMSG11	0000H	unused
SMSG12	0000H	unused
SMSG13	0000H	unused
SMSG14	FF05H	P5 register address (Used only when USE_LED is defined)
SMSG15	FFFFH	fixed value: FFFFH

Table 5-7 Sequencer instruction registers 0-31 (When USE_LED is defined)

Register Symbol	Setting	Remark
SMSI0	5E30H	CLR1 [SMSG14+0].3
SMSI1	3610H	MOVW SMSG1, [SMSG6+0]
SMSI2	9805H	WAIT 128, 5
SMSI3	6370H	MOV1 SCY, [SMSG3+0].7
SMSI4	8031H	BNC $__portinbranch_finish
SMSI5	4E30H	SET1 [SMSG14+0].3
SMSI6	F000H	FINISH
SMSI7	7150H	ADDW SMSG1, SMSG5
SMSI8	7122H	CMPW SMSG1, SMSG2
SMSI9	8F90H	BC $__updbranch_do
SMSI10	4E30H	SET1 [SMSG14+0].3
SMSI11	5440H	CLR1 [SMSG4+0].4
SMSI12	F000H	FINISH
SMSI13-31	0000H	unused

Table 5-8 Sequencer instruction registers 0-31 (When USE_LED is undefined)

Register Symbol	Setting	Remark
SMSI0	3610H	MOVW SMSG1, [SMSG6+0]
SMSI1	9805H	WAIT 128, 5
SMSI2	6370H	MOV1 SCY, [SMSG3+0].7
SMSI3	8021H	BNC $__portinbranch_finish
SMSI4	F000H	FINISH
SMSI5	7150H	ADDW SMSG1, SMSG5
SMSI6	7122H	CMPW SMSG1, SMSG2
SMSI7	8FA0H	BC $__updbranch_do
SMSI8	5440H	CLR1 [SMSG4+0].4
SMSI9	F000H	FINISH
SMSI10-31	0000H	unused

6. Application example

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

r01an5609_sms_push_jadgement.scfg

r01an5609_sms_push_jadgement.sms

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

6.1 r01an5609_sms_push_jadgement.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: $1.8V \leqq EV_{DD0} < 5.5V$ High-speed on-chip oscillator: 32MHz f_{IHP}: 32MHz f_{CLK}: 32MHz (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_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_SMS	Components: SNOOZE Mode Sequencer Start trigger: ELCL output signal
	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.

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_TAU0_7

Set TAU07 of the sample code.

In the sample code, it is used to count the lighting time of LED2. When the INTP0 is not used, delate it.

6.1.7 Config_SMS

Set the sample code SMS.

For details, refer to "6.2 r01an5609_sms_push_jadgement.sms".

6.1.8 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.

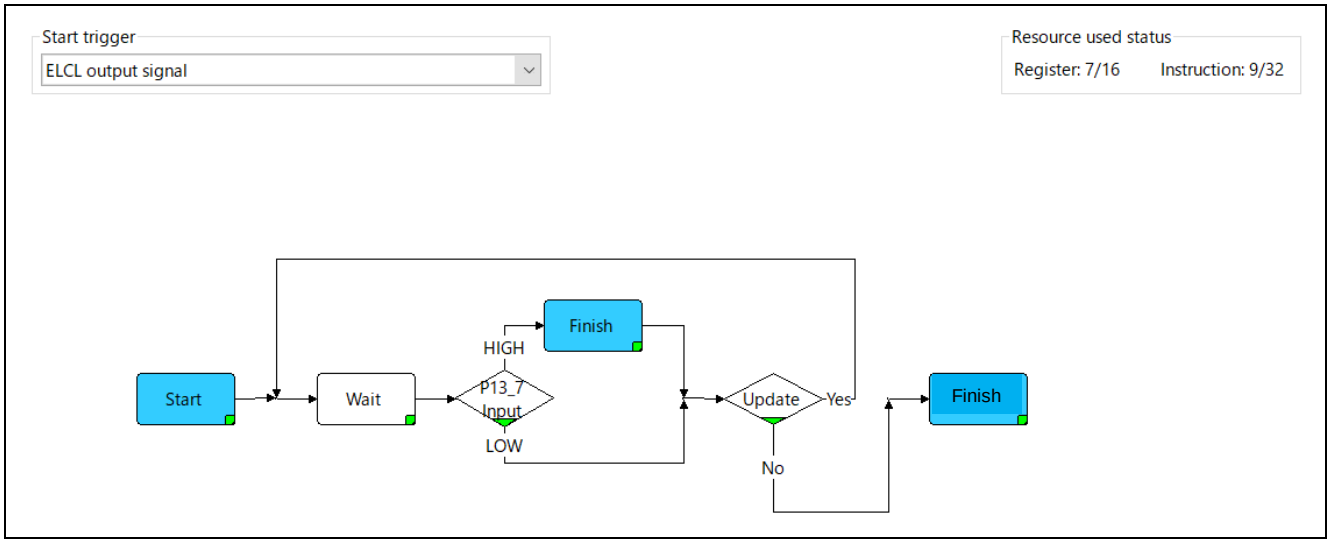
6.2 r01an5609_sms_push_jadgement.sms

This is the data for Config_SMS alone. In the sample code, the ELCL output signal is used to start the SMS. Please note that you need to configure ELCL separately.

The r01an5609_sms_push_jadgement.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 "r01an5609_sms_push_jadgement.sm" 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 5-6 SMS process".

Figure 6-1 Config_SMS flow chart

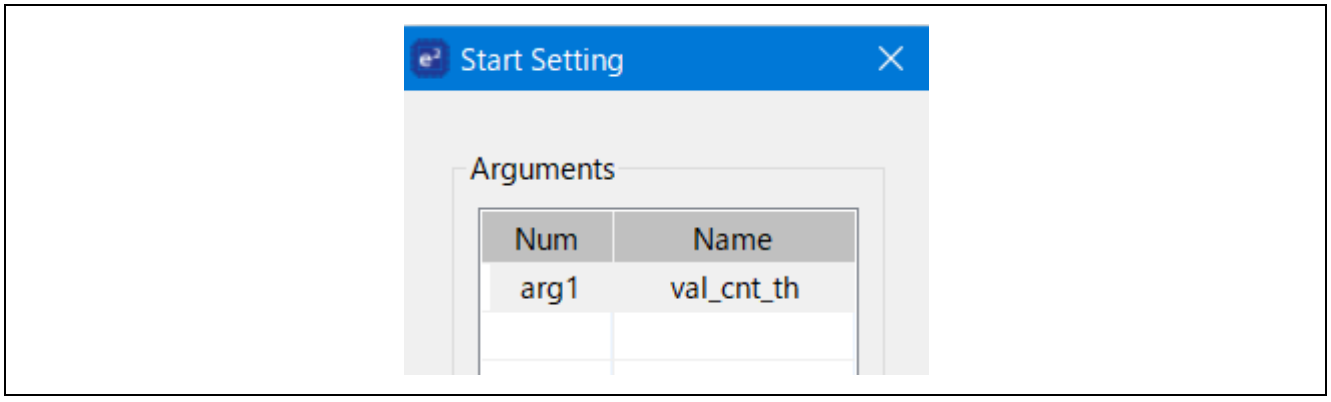


A description of each block is shown below.

6.2.1 Start

When the SMS starts, The value of JUDGMENT_COUNT passed as an argument in the SMS start function (R_Config_SMS_Start function) is set to val _cnt_th (count threshold of wait count).

Figure 6-2 Start Setting

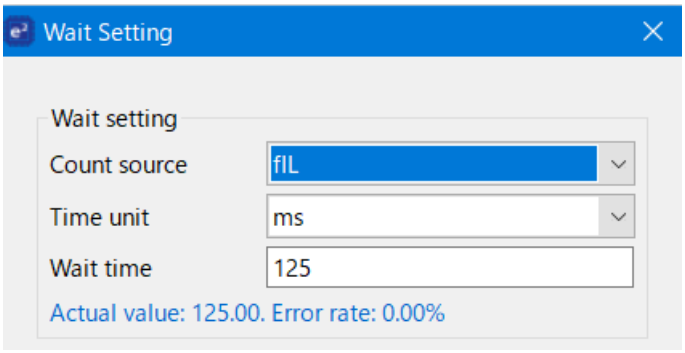


6.2.2 Wait

Wait for the SMS processing for the set wait time. In the sample code, the counting source (fIL) waits for 125ms for processing.

When changing the waiting time, if you set a value that cannot be set, it will be in red. Note that changing.

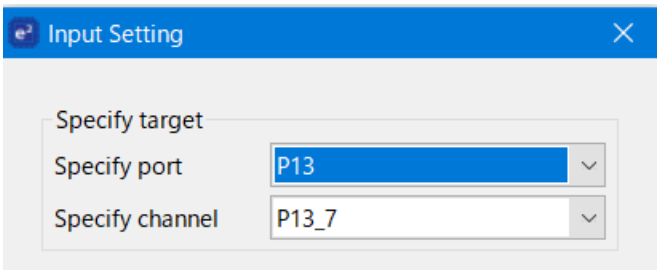
Figure 6-3 Wait Setting



6.2.3 P13_7 Input

Check the input value of the target specified port and branch the process. In the sample code, the value of P13_7 is used as the target pin. To set the port (e.g., change to input mode), use the Config_Port component. The P13_7 used in the sample code does not need to be set because the input/output mode is fixed to input.

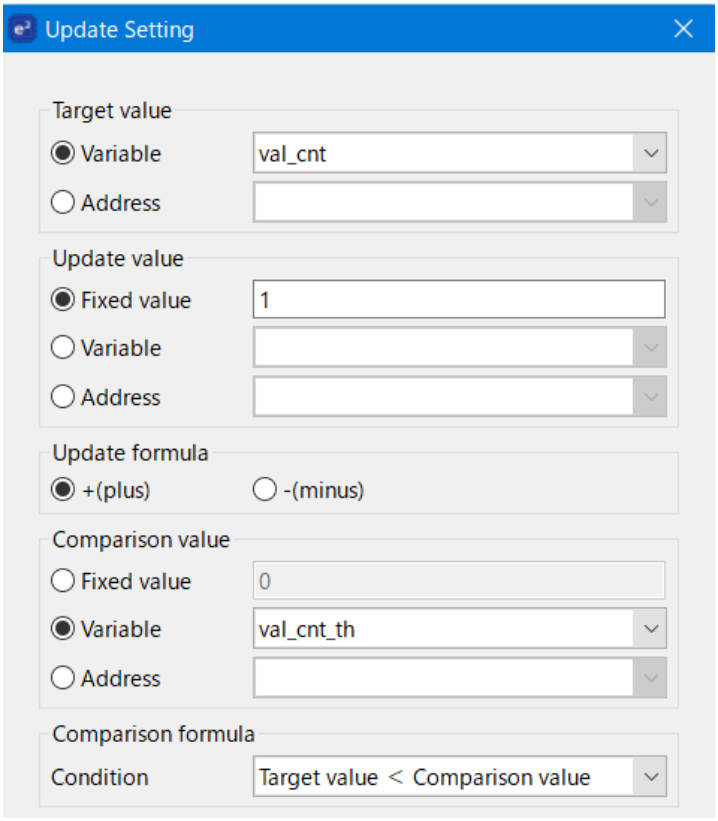
Figure 6-4 Input Setting



6.2.4 Update

Add the fixed value "1" to the value "0" (initial value) set in the variable "val_cnt", and compare this value with the comparison value "val_cnt_th. The value is compared with the expression (val_cnt < val_cnt_th). If the value matches the comparison, it returns to the wait, at which time val_cnt is overwritten with the updated value "1". If the values match, proceed.

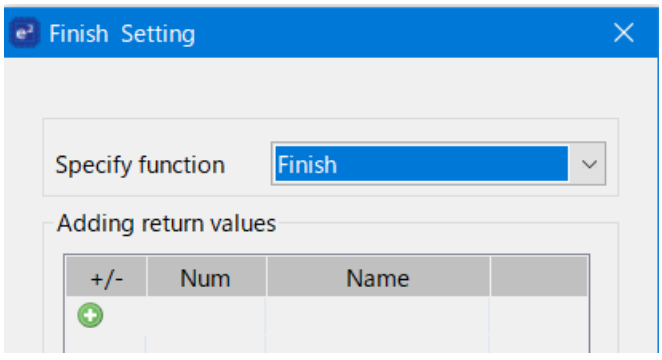
Figure 6-5 Input Setting



6.2.5 Finish

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

Figure 6-6 Finish Setting



6.2.6 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_cnt	Initialize every time with SMS	0	Stores the count value of the number of wait.
val_cnt_th	Pass argument via SMS start function	-	Stores the count threshold of the number of wait. The value of JUDGMENT_COUNT is set as an argument in the R_Config_SMS_Start function.

6.3 How to change the judgment time

The following formula is the judgment time for the sample code. This section explains how to change the judgment time.

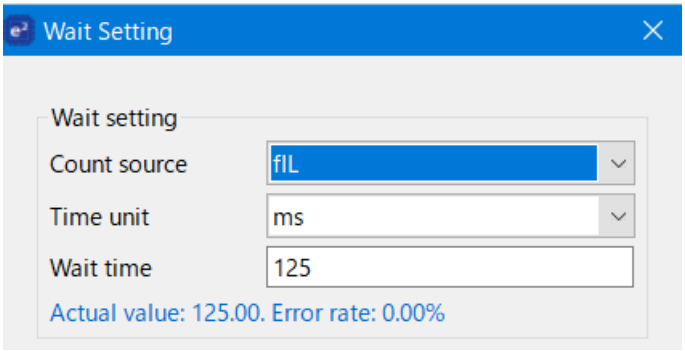
Judgment time = 125ms x JUDGMENT_COUNT (ms)

The constant "JUDGMENT_COUNT" is set as follows.

```
#define JUDGMENT_COUNT    (24U)
```

As shown in Figure 6-7 Wait Setting, 125ms is set in the Wait block of the SMS.

Figure 6-7 Wait Setting



The judgment time can be changed by changing the value of JUDGMENT_COUNT and the Wait setting.

7. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

8. Reference

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

RL78 Family User's Manual: Software (R01US0015E)

SMS assembler User's Manual【Preliminary version】(R20UT4792E)

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.)

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 -> V8.05.01 C compiler (CS+) : V1.09.00 -> V1.10 Integrated development environment (e² studio) : 2021-01 (21.01.0) -> 2021-04 (21.4.0) C compiler (e² 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 20, 21	Changed due to COM port support Table 2-1 Operation Confirmation Conditions Emulator: E2 Emulator Lite -> CS＋, e² studio: COM port IAR: E2 Emulator Lite Figure 4-1 Hardware Configuration Added P11/TOOLRxD and P12/TOOLTxD
		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 e² studio, CS+ : r_smc_entry.h, elc.h IAR : ior7f100g.h, ior7f100g_ext.h, r_cg_macrodriver.h, Config_SMS.h, Config_TAU0_7.h, elc.h -> r_smc_entry.h, elc.h
		20	Changed clock abbreviation Table 6-1 Parameters of Smart Configurator Clocks : f_{SXL} -> f_{SXP}
		28	Added of RL78 Smart Configurator User's Guide 8. Reference 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)
1.20	Feb.15.22	5	Modified Figure 1-2 Entire Flowchart
		18	Modified Figure 5-6 SMS process
1.30	Jan.9.24	-	Changed the flowchart for SMS processing
1.40	Jul.24.24	6	Modified Table 2-1 Operation Confirmation Conditions
		19	Modified Table 5-7 Sequencer instruction registers 0-31 (When USE_LED is defined)
		20	Modified Table 5-8 Sequencer instruction registers 0-31 (When USE_LED is undefined)

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/.