RL78/G23

ELCL Multiple Parameter Monitoring Function

Introduction

This application note describes how to monitor multiple input signals (parameters) using the logic and event link controller (ELCL). By using ELCL, it is possible to reduce the resources (external parts, ROM, RAM, etc.) because the functions realized by external parts or software can be realized by hardware.

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 describes an example of linking an output signal generated by ORing four input signals using ELCL to a specific peripheral function.

Figure 1-1 shows the ELCL configuration and Figure 1-2 shows the timing chart.

Link the four input signals (INPUT A, INPUT B, INPUT C, INPUT D) to the three ORs (L1L0, L1L1, L2L0), and link the generated signal to OUTPUT E. In addition, link the four input signals to the flip-flops (L2F0, L2F1, L3F0, L3F1), and link the output of the flip-flops to the monitor register (ELOMONI). Thereby it is possible to confirm the state of the input signal by reading the monitor register.

For example, link four ports (P20 to P23) that do not have an external interrupt function to INPUT A to INPUT D, and set the link destination of OUTPUT E to event link interrupt (INTELCL). As a result, INTELCL is generated when any one of P20 to P23 changes, and the port without external interrupt function can be treated as an interrupt.

Figure 1-1 ELCL Configuration

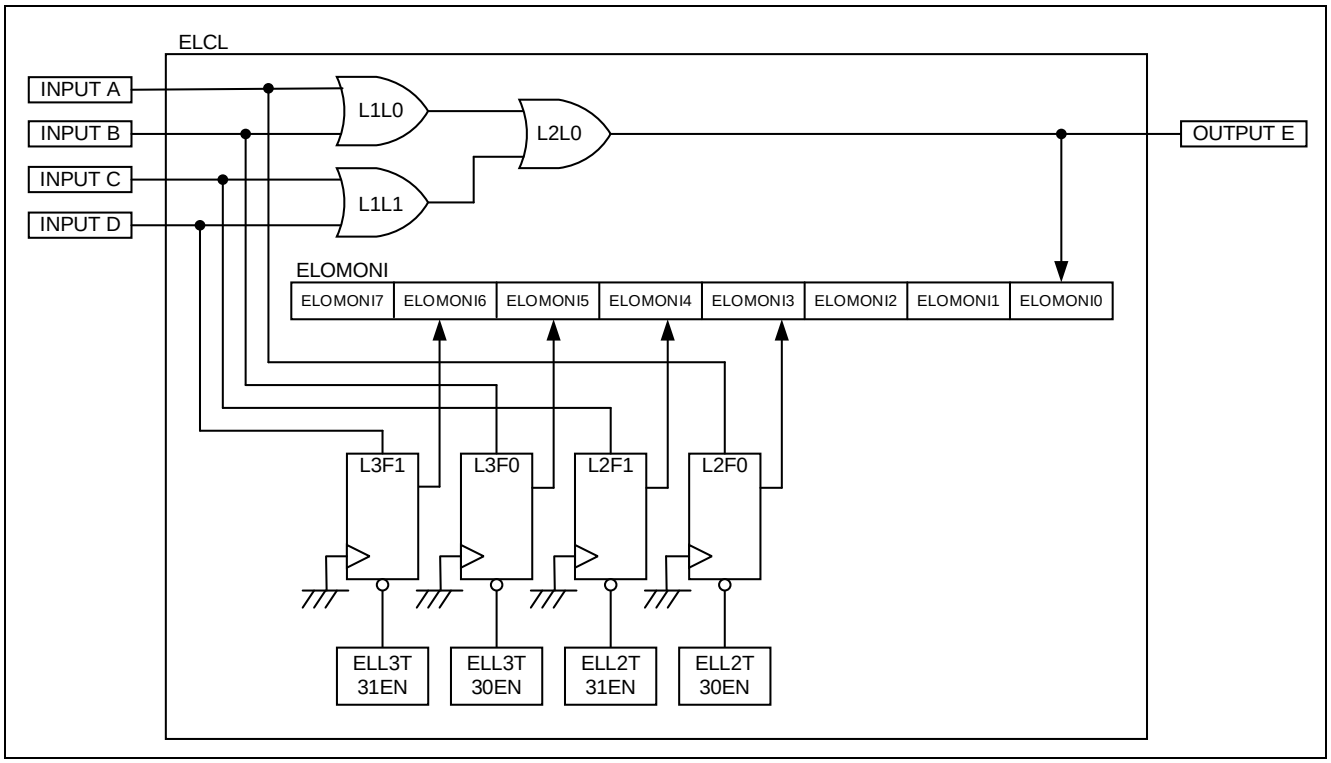
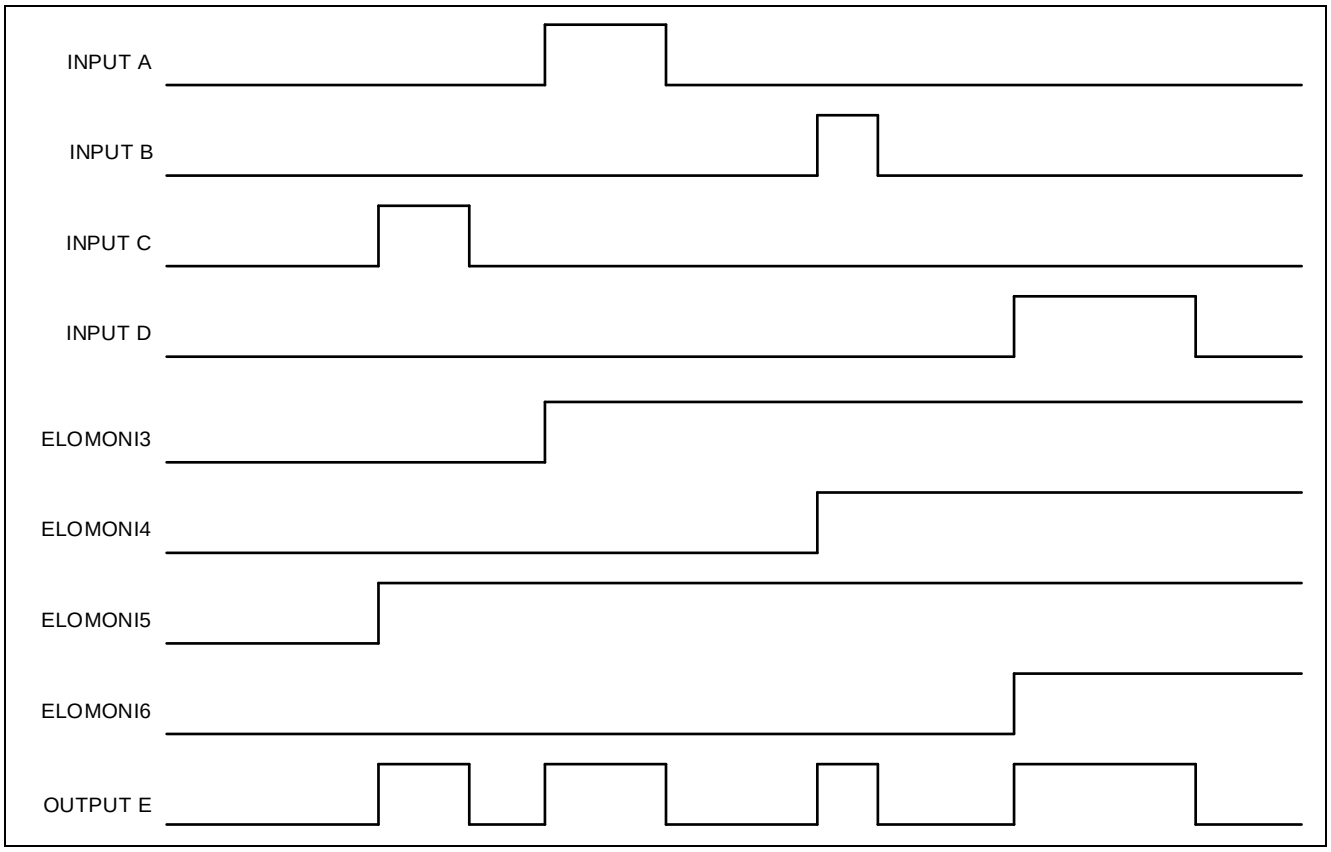


Figure 1-2 Timing chart



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

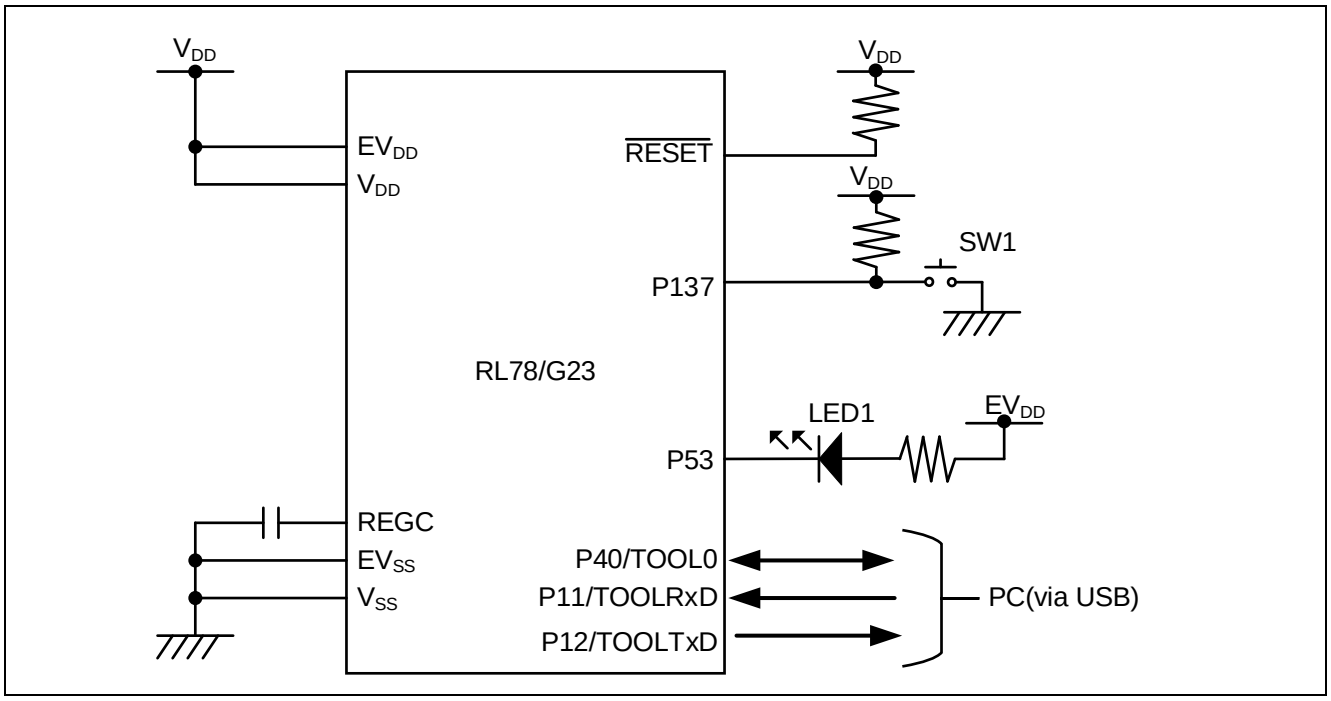
Items	Contents
MCU	RL78/G23 (R7F100GLG)
Operating frequencies	• High-speed on-chip oscillator clock: 32 MHz • CPU/peripheral hardware clock: 32 MHz
Operating voltage	• 3.3V • LVD0 operations (V_{LVD0}) : Reset mode Rising edge TYP.1.90V Falling edge TYP.1.86V
Integrated development environment (CS+)	CS+ for CC V8.07.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.11 from Renesas Electronics Corp.
Integrated development environment (e^2 studio)	e^2 studio 2022-01 (22.01.0) from Renesas Electronics Corp.
C compiler (e^2 studio)	CC-RL V1.11 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.1.2.0
Board support package (r_bsp)	V.1.13
Emulator	CS＋, e^2 studio: COM port IAR: E2 Emulator Lite
Board	RL78/G23 Fast Prototyping Board (RTK7RLG230CLG000BJ)

3. Hardware

3.1 Example of Hardware Configuration

Figure 3-1 shows an example of the hardware configuration in this application.

Figure 3-1 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.

3.2 Used Pins

Table 3-1 shows list of used pins and assigned functions.

Table 3-1 List of Pins and Functions

Pin Name	Input/Output	Function
P53	Output	LED1 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.

4. Software

4.1 Overview of the sample program

In this sample code, four timer (TAU0 channel 0, TAU0 channel 1, TAU0 channel 2 and TAU0 channel 3) are operated in interval timer mode. ELCL ORs the four timer completion interrupt (INTTM00, INTTM01, INTTM02 and INTTM03) as input signals and outputs the result as INTELCL. When INTELCL occurs, the output of the port (P53) is inverted and LED1 turns on or off. Since the combination of four timer outputs changes each time SW1 is pressed, the lighting interval of LED1 changes accordingly.

Table 4-1 shows operation overview. The operations from No. (1) to (10) are repeated by pressing SW1.

Table 4-1 Operation overview (Combination of timers)

No.	g_timer_mode	SW1	TAU0				LED1
			Channel 0	Channel 1	Channel 2	Channel 3	
(1)	0	-	STOP	STOP	STOP	STOP	OFF
(2)	1	Press	STOP	STOP	STOP	STOP	OFF
(3)	1	-	Count	STOP	STOP	STOP	Blink (About 2s intervals)
(4)	2	Press	STOP	STOP	STOP	STOP	OFF
(5)	2	-	Count	Count	STOP	STOP	Blink (About 1s intervals)
(6)	3	Press	STOP	STOP	STOP	STOP	OFF
(7)	3	-	Count	Count	Count	STOP	Blink (About 0.5s intervals)
(8)	4	Press	STOP	STOP	STOP	STOP	OFF
(9)	4	-	Count	Count	Count	Count	Blink (About 0.25s intervals)
(10)	0	Press	STOP	STOP	STOP	STOP	OFF

Figure 4-1 shows operation overview, Figure 4-2 shows timing chart.

Figure 4-1 Operation overview

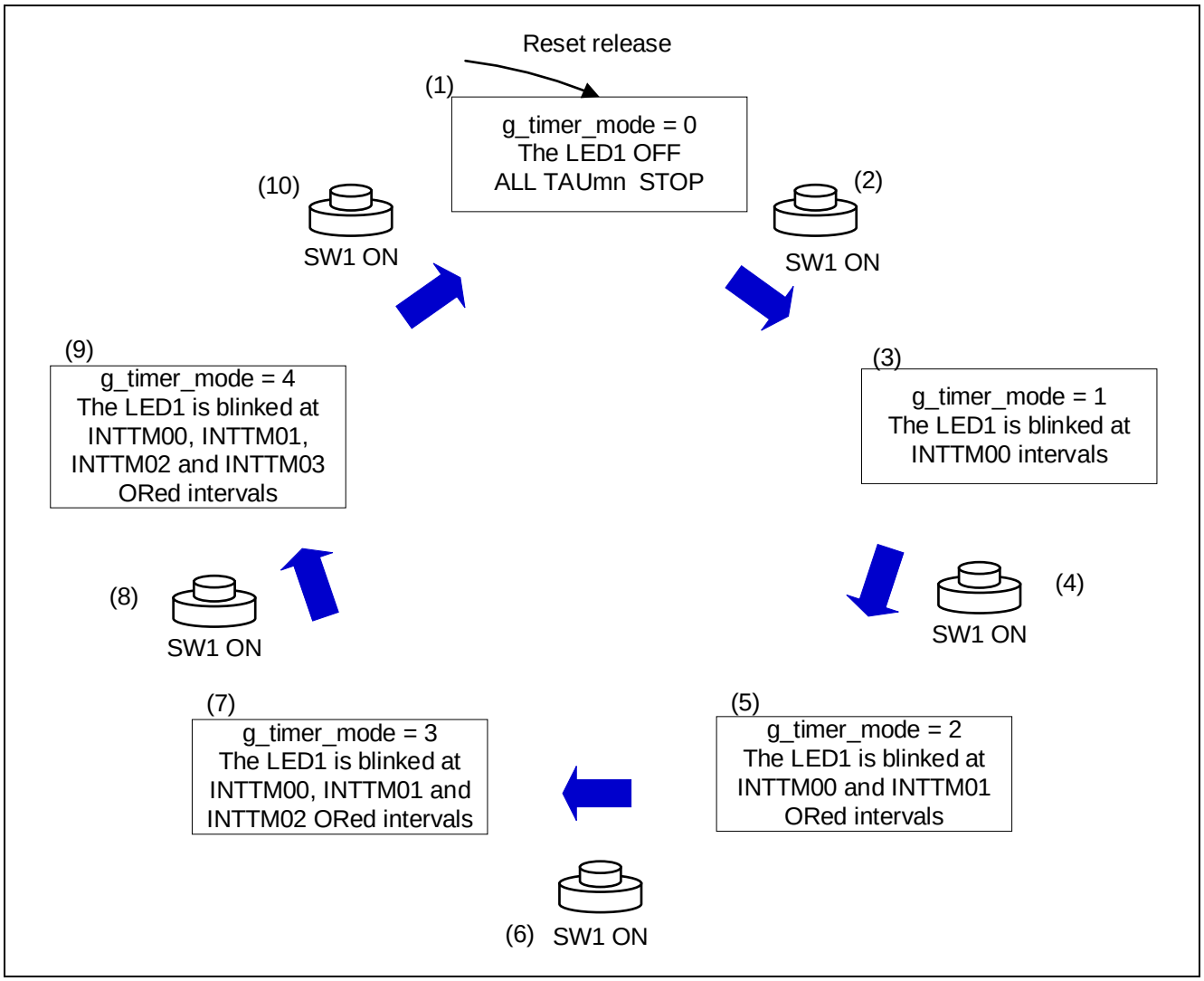


Figure 4-2 Timing chart of the sample code

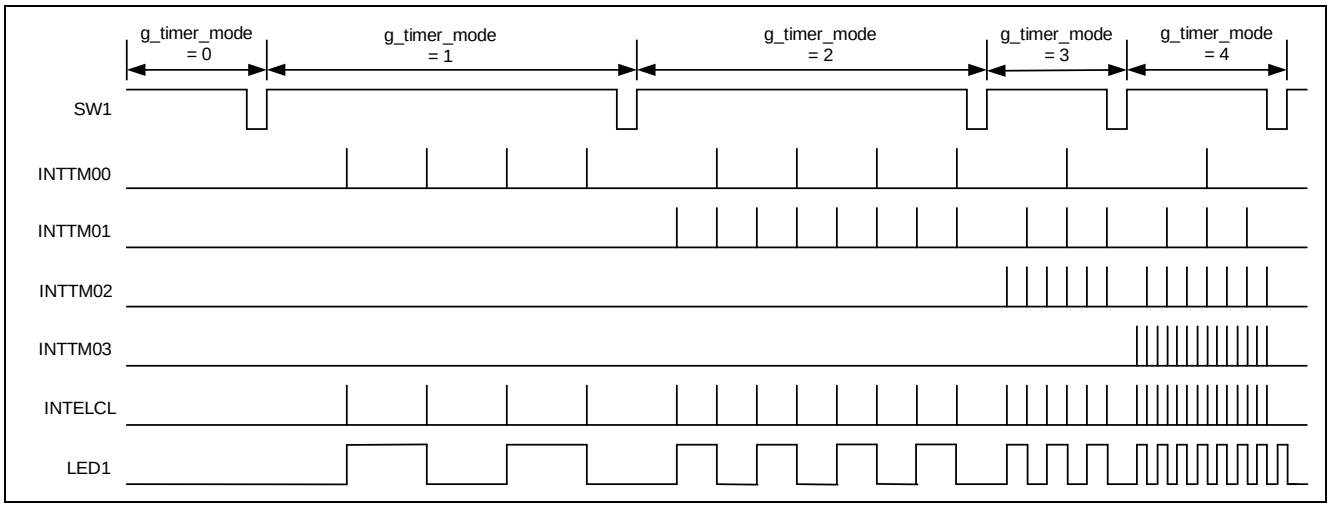
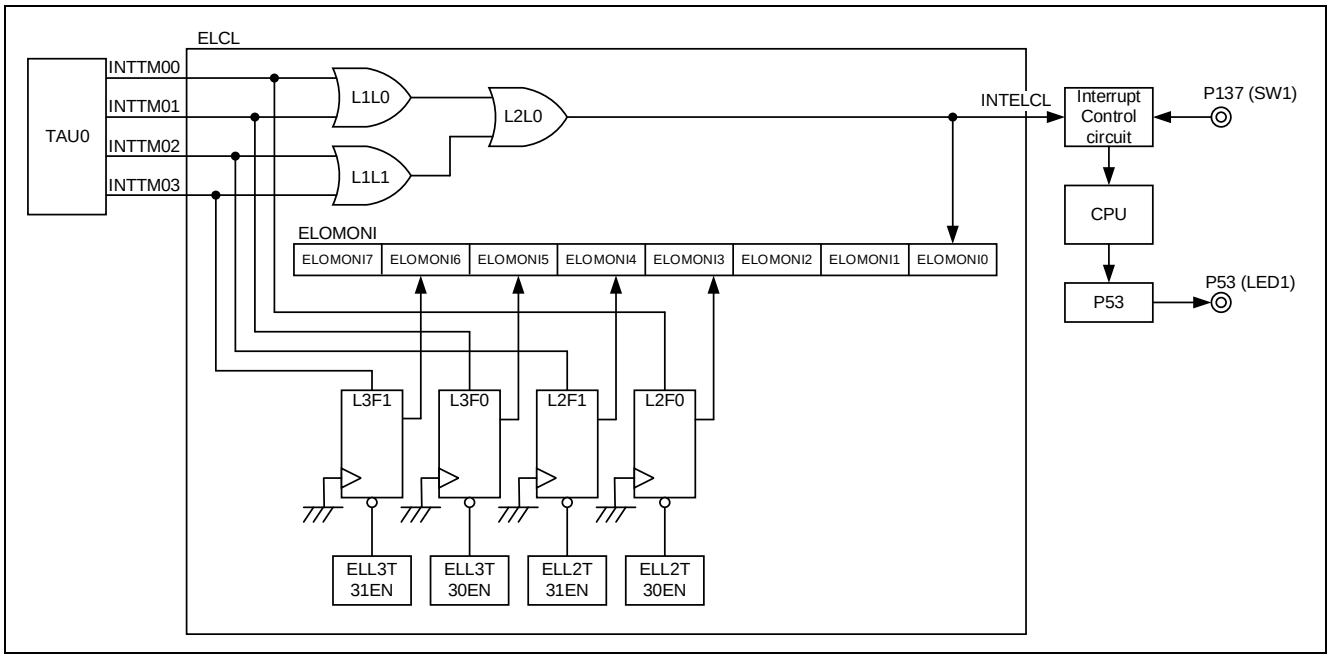


Figure 4-3 shows the system configuration of the sample code.

Select INTTM00, INTTM01, INTTM02 and INTTM03 as the ELCL input signal and INTELCL as the ELCL output signal.

Figure 4-3 System configuration of the sample code



4.2 Folder Configuration

Table 4-2 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 4-2 Folder configuration (1/2)

Folder/File configuration	Outline	Created by Smart configurator
¥r01an5615_elcl_parameter<DIR>	Root folder of this sample code	
¥src<DIR>	Folder for program source	
main.c	Sample code source file	
support_functions.c	Source file for functions	
support_functions.h	Header file for functions	
¥smc_gen<DIR>	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 2
¥Config_MultipleParameterMonitoring<DIR>	Folder for ELCL program	√
Config_MultipleParameterMonitoring.c	Source file for ELCL	√
Config_MultipleParameterMonitoring.h	Source file for ELCL	√
Config_MultipleParameterMonitoring_user.c	Interrupt source file for ELCL	√Note 2
¥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_TAU0_0<DIR>	Folder for TAU0 channel 0 program	√
Config_TAU0_0.c	Source file for TAU0 channel 0	√
Config_TAU0_0.h	Header file for TAU0 channel 0	√
Config_TAU0_0_user.c	Interrupt source file for TAU0 channel 0	√Note 1
¥Config_TAU0_1<DIR>	Folder for TAU0 channel 1 program	√
Config_TAU0_1.c	Source file for TAU0 channel 1	√
Config_TAU0_1.h	Header file for TAU0 channel 1	√
Config_TAU0_1_user.c	Interrupt source file for TAU0 channel 1	√Note 1
¥Config_TAU0_2<DIR>	Folder for TAU0 channel 2 program	√
Config_TAU0_2.c	Source file for TAU0 channel 2	√
Config_TAU0_2.h	Header file for TAU0 channel 2	√
Config_TAU0_2_user.c	Interrupt source file for TAU0 channel 2	√Note 1
¥Config_TAU0_3<DIR>	Folder for TAU0 channel 3 program	√
Config_TAU0_3.c	Source file for TAU0 channel 3	√
Config_TAU0_3.h	Header file for TAU0 channel 3	√
Config_TAU0_3_user.c	Interrupt source file for TAU0 channel 3	√Note 1

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.

Table 4-3 Folder configuration (2/2)

Folder/File configuration	Outline	Created by Smart configurator
¥r01an5615_elcl_parameter<DIR> Note 3	Root folder of this sample code	
¥src<DIR>	Folder for program source	
¥smc_gen<DIR>	Folder created by Smart Configurator	√
¥Config_TAU0_7<DIR>	Folder for TAU0 channel 7 program	√
Config_TAU0_7.c	Source file for TAU0 channel 7	√
Config_TAU0_7.h	Header file for TAU0 channel 7	√
Config_TAU0_7_user.c	Interrupt source file for TAU0 channel 7	√Note 2
¥general<DIR>	Folder for initialize or common program	√
¥r_bsp<DIR>	Folder for BSP program	√
¥r_config<DIR>	Folder for program	√

Note. <DIR> means directory.
Note 2. Added the interrupt handling routine to the file generated by the Smart Configurator.
Note 3. The IAR version of the sample code contains r01an5615_elcl_parameter.ipcf. For the ipcf file, refer to "RL78 Smart Configurator User Guide: IAR (R20AN0581)".

4.3 Option Byte Settings

Table 4-4 shows the option byte settings.

Table 4-4 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 1.90V Falling edge 1.86V
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

4.4 Constants

Table 4-5 shows the constants that are used in this sample code.

Table 4-5 Constants used in the sample code

Constant Name	Setting Value	Contents	File
LED1	P5_bit.no3	P53	main.c
LED_ON	0	Setting value for turning on the LED	main.c
LED_OFF	1	Setting value for turning off the LED	main.c
CHATTA_WAIT	100	Chattering prevention time (100ms)	Config_INTC_user.c

4.5 Variables

Table 4-6 shows the global variables used in this sample code.

Table 4-6 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
volatile uint8_t	g_led_flag	INTELCL generation flag Polling with the main function and toggle the state of the LED	main r_Config_MultipleParameterMonitoring_interrupt
uint8_t	g_timer_mode	Timer operation mode	r_timer_update r_Config_MultipleParameterMonitoring_interrupt

4.6 Functions

Table 4-7 shows the functions used in the sample code. However, the unchanged functions generated by the Smart Configurator are excluded.

Table 4-7 Functions

Function name	Outline	Source file
main	Main process	main.c
r_timer_update	Channel change process for TAU0	support_functions.c
r_timer_reset	TAU0 stop process	support_functions.c
r_Config_MultipleParameterMonitoring_interrupt	ELCL interrupt process	Config_MultipleParameterMonitoring_user.c
r_elcl_reset_flipflop	ELCL flip-flop reset process	Config_MultipleParameterMonitoring_user.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

4.7 Function Specifications

This part describes function specifications of the sample code.

[Function name] main

Outline	Main process
Header	r_smc_entry.h, support_functions.h
Declaration	void main (void);
Description	This function initializes ELCL, sets ELCL output, and sets interrupts. Every time it returns from INTELCL, P53 is inverted and LED1 turns ON/OFF. Set the flag of ELCL interrupt generation g_led_flag to 0.
Arguments	None
Return value	None
Remarks	None

[Function name] r_timer_update

Outline	TAU0 operation channel change process
Header	r_smc_entry.h, support_functions.h
Declaration	void timer_update (void);
Description	Depending on the value of g_timer_mode, the combination of channels of the TAU to be operated is changed, and the TAU is started and stopped.
Arguments	None
Return value	None
Remarks	None

[Function name] r_timer_reset

Outline	TAU0 reset process
Header	r_smc_entry.h, support_functions.h
Declaration	void timer_reset (void);
Description	Stop TAU0, reset the counter, and wait for the start state.
Arguments	None
Return value	None
Remarks	None

[Function name] r_Config_MultipleParameterMonitoring_interrupt

Outline	ELCL interrupt process
Header	platform.h
Declaration	#pragma interrupt r_Config_MultipleParameterMonitoring_interrupt (vect=INTELCL)
Description	This function sets the flag of ELCL interrupt generation g_led_flag to 1.
Arguments	None
Return value	None
Remarks	None

[Function name] r_elcl_reset_flipflop

Outline	ELCL flip-flop reset process
Header	platform.h
Declaration	void r_elcl_reset_flipflop (void);
Description	This function resets the flip-flops to reset the ELOMONI flag.
Arguments	None
Return value	None
Remarks	Flip-flops are reset when bit 6 and bit 7 of ELLnCTL are set to 0.

[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
Declaration	#pragma interrupt r_Config_INTC_intp0_interrupt (vect=INTP0)
Description	Sets g_timer_mode to indicate the state of the timer. Sets the INTP0 interrupt generation flag g_timer_flag to 1. Execute the wait process to prevent chattering when pressing SW1.
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

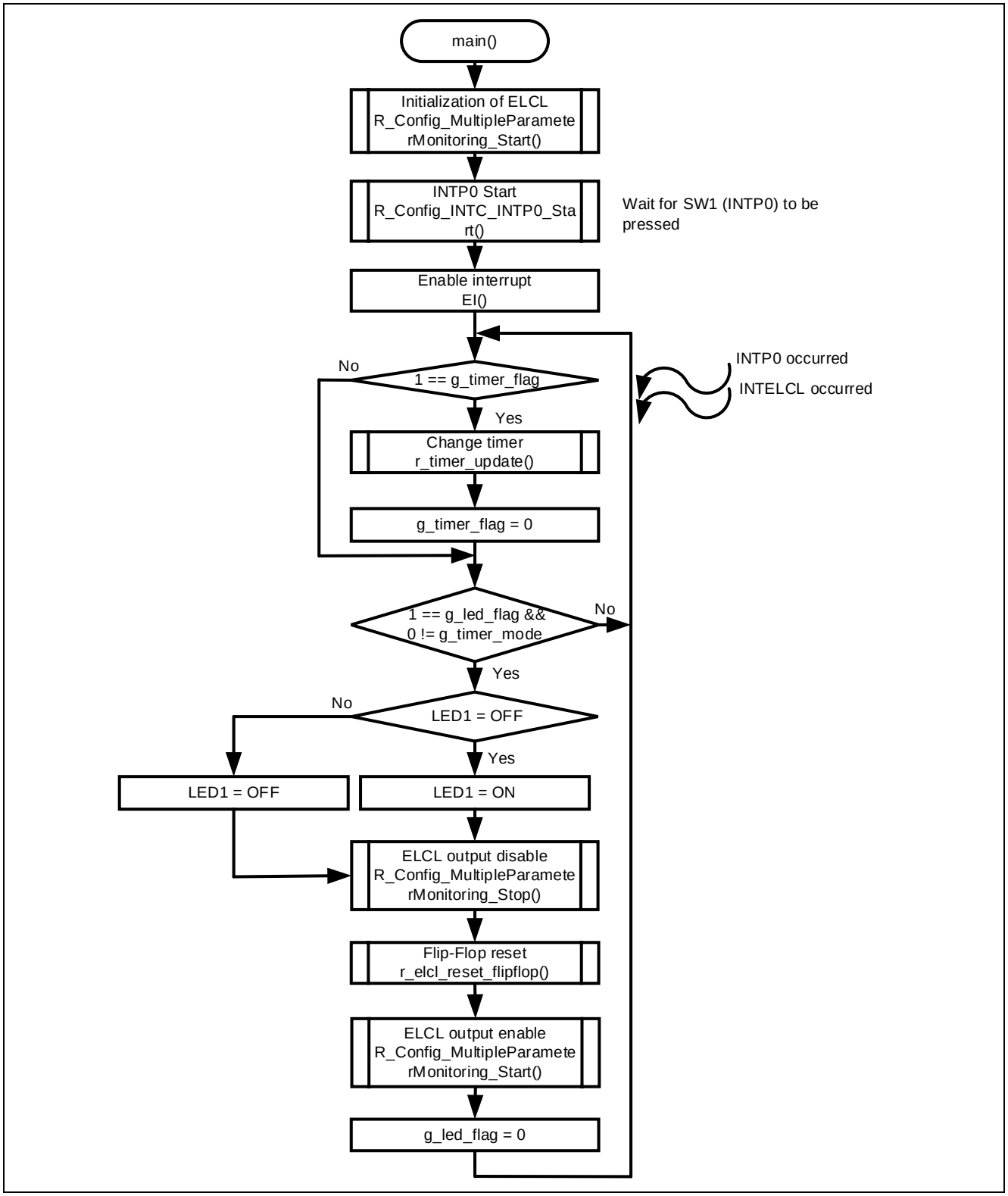
Outline	TAU0 channel 7 interrupt process
Header	r_cg_macrodriver.h, r_cg_userdefine.h, Config_TAU0_7.h
Declaration	#pragma interrupt r_Config_TAU0_7_interrupt (vect=INTTM07)
Description	This function is an interrupt process by INTTM07 on TAU0 channel 7. Counts up g_ms_timer.
Arguments	None
Return value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

4.8 Flow Charts

4.8.1 Main Process

Figure 4-4 shows flowchart of main process.

Figure 4-4 Main process



4.8.2 Channel change process for TAU0

Figure 4-5 shows flowchart of the channel change process for TAU0.

Figure 4-5 Channel change process for TAU0 (1/2)

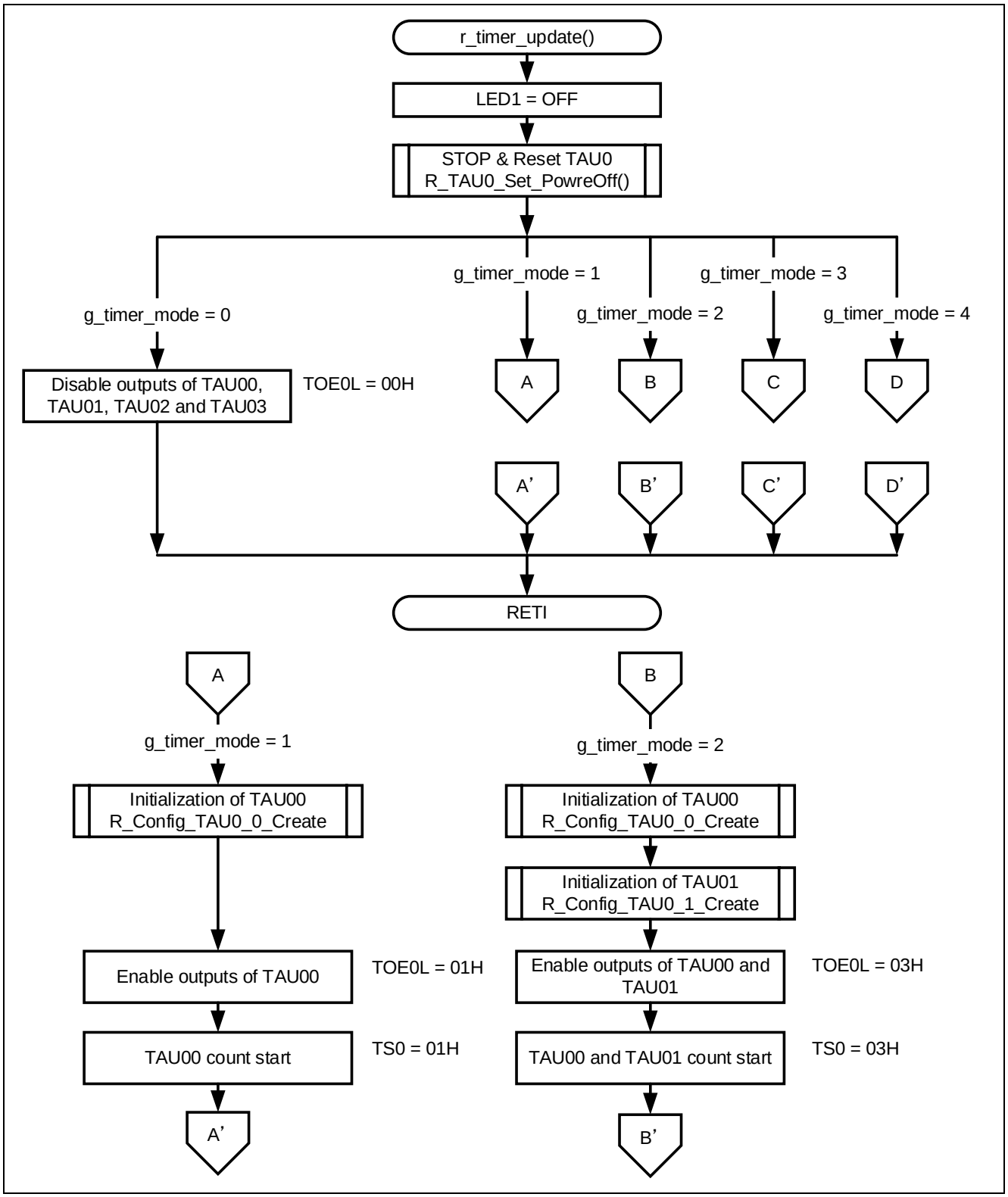
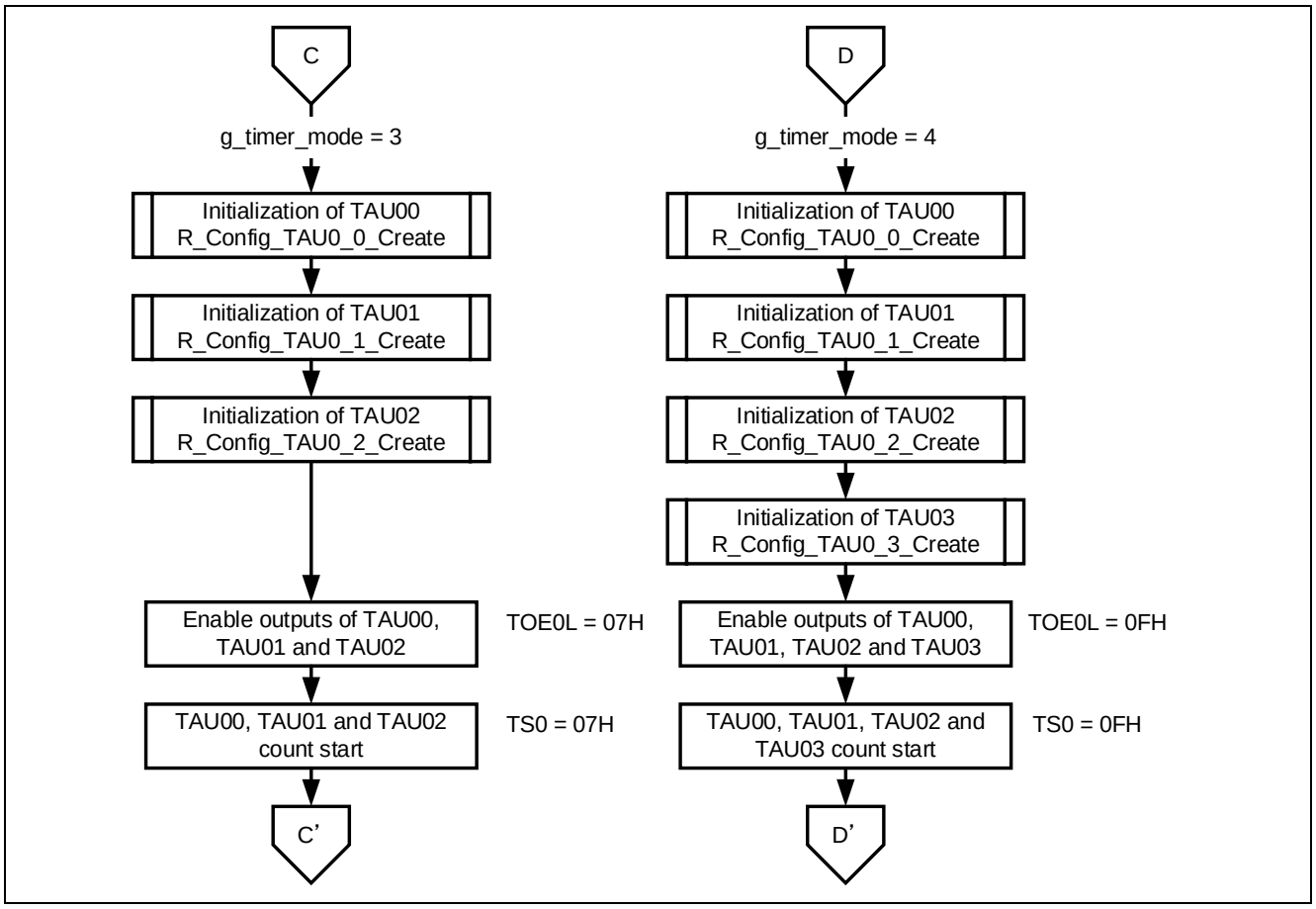


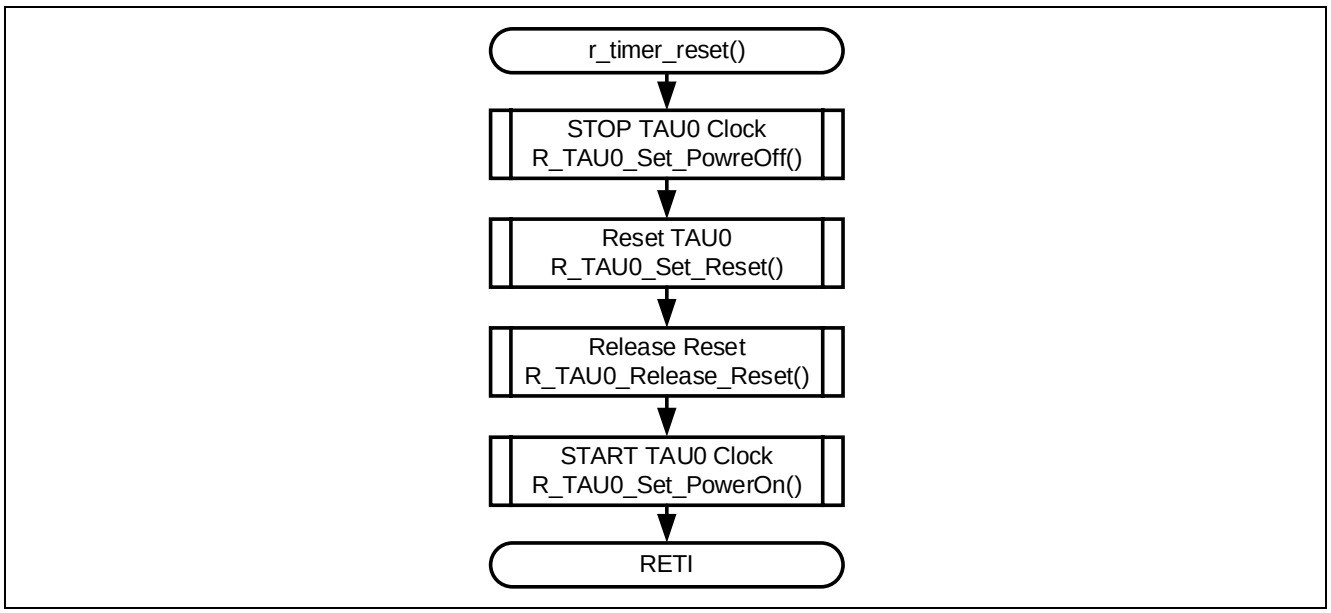
Figure 4-6 Channel change process for TAU0 (2/2)



4.8.3 TAU0 reset process

Figure 4-7 shows flowchart of TAU0 reset process.

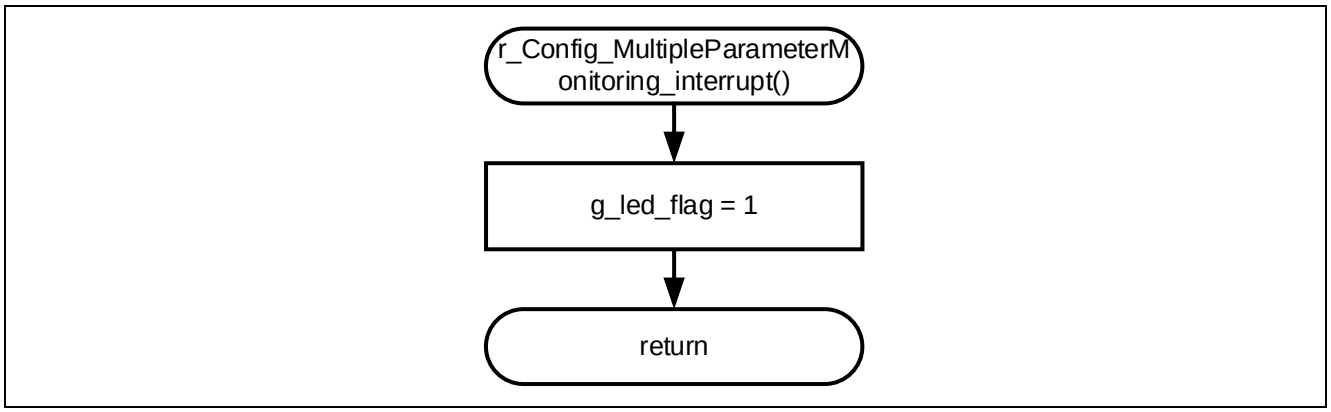
Figure 4-7 TAU0 reset process



4.8.4 ELCL interrupt process

Figure 4-8 shows flowchart of ELCL interrupt process.

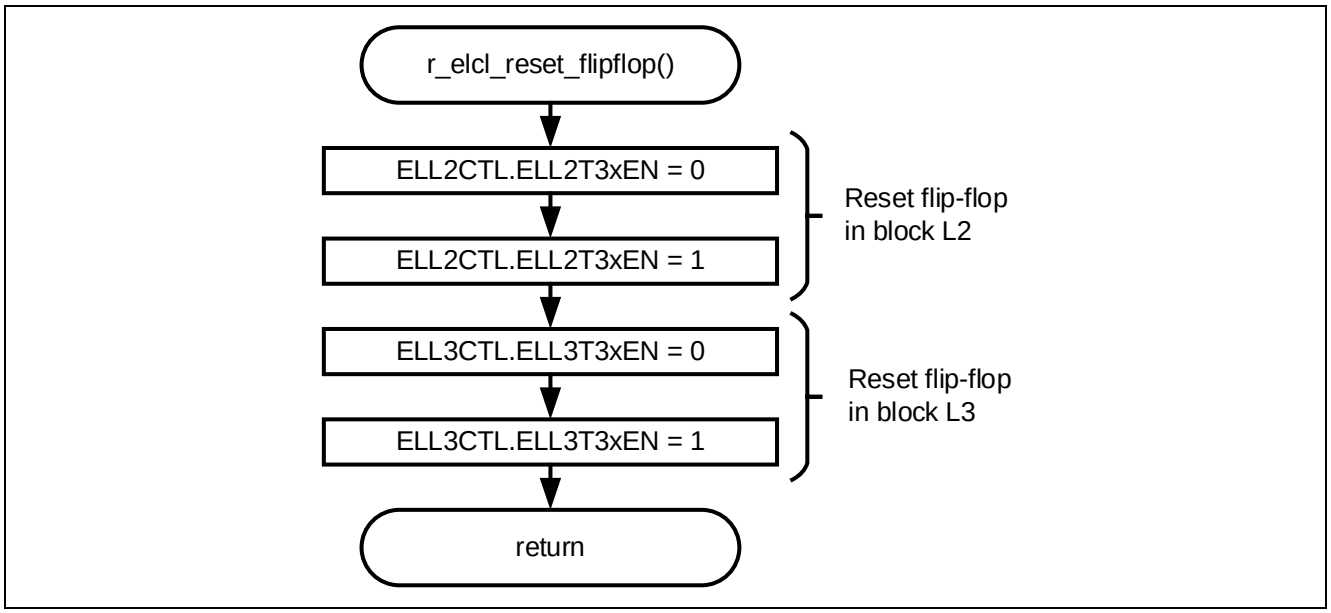
Figure 4-8 ELCL interrupt process



4.8.5 ELCL flip-flop reset process

Figure 4-9 shows flowchart of ELCL flip-flop reset process.

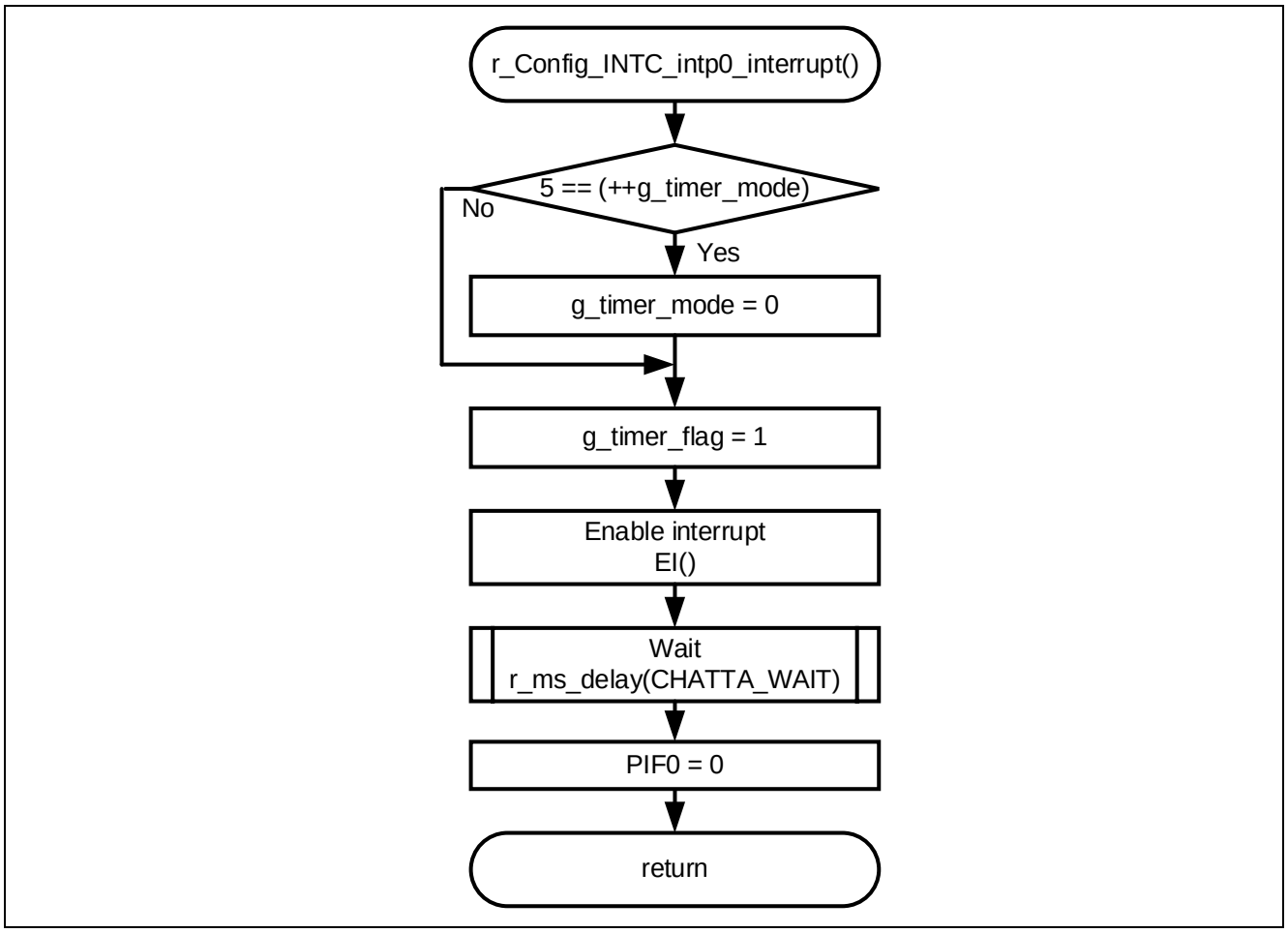
Figure 4-9 ELCL flip-flop reset process



4.8.6 INTP0 interrupt process

Figure 4-10, shows flowchart of INTP0 interrupt process.

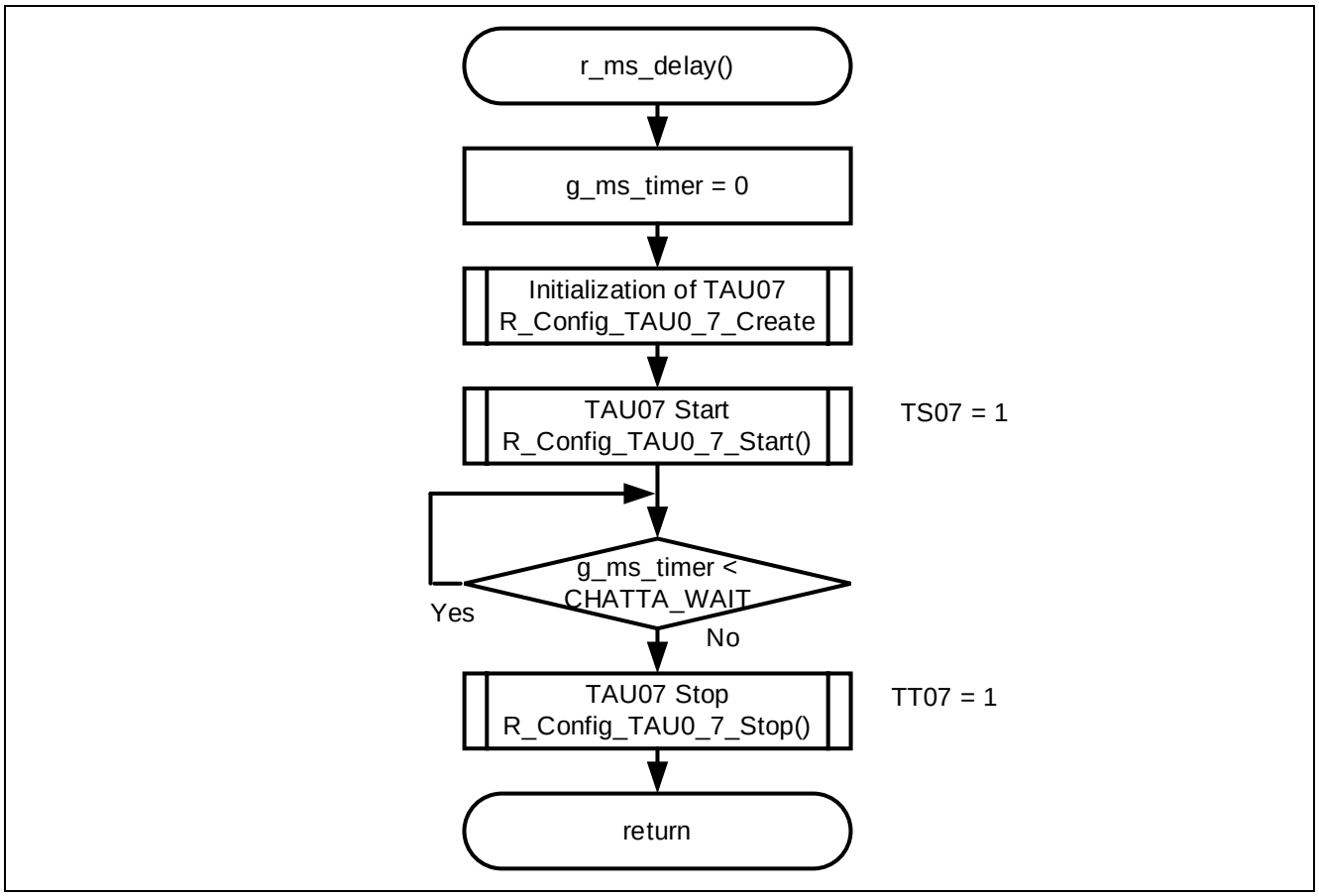
Figure 4-10 INTP0 interrupt process



4.8.7 Wait process

Figure 4-11 shows flowchart of Wait process.

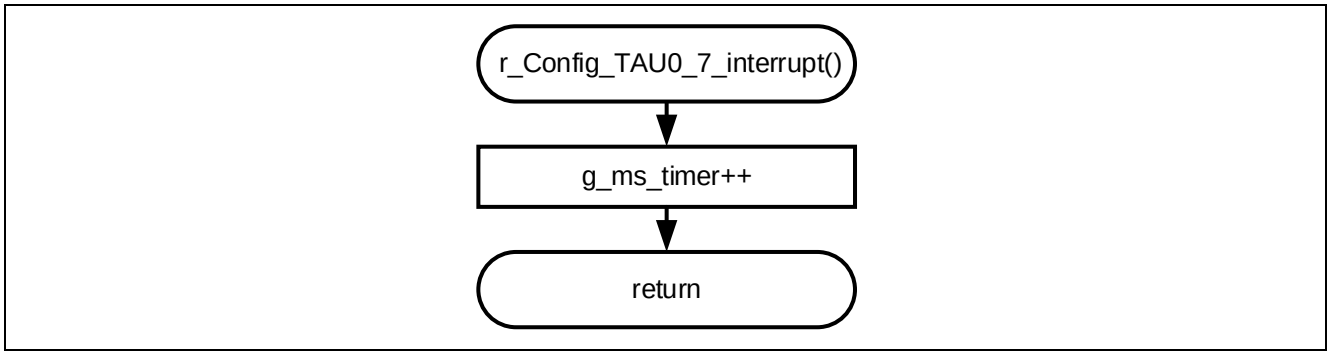
Figure 4-11 Wait process



4.8.8 TAU0 channel 7 interrupt process

Figure 4-12 shows flowchart of TAU0 channel 7 interrupt process.

Figure 4-12 TAU0 channel 7 interrupt process



5. Application example

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

r01an5615_elcl_parameter.scfg

The following is a description of the file and examples of settings and notes for use.

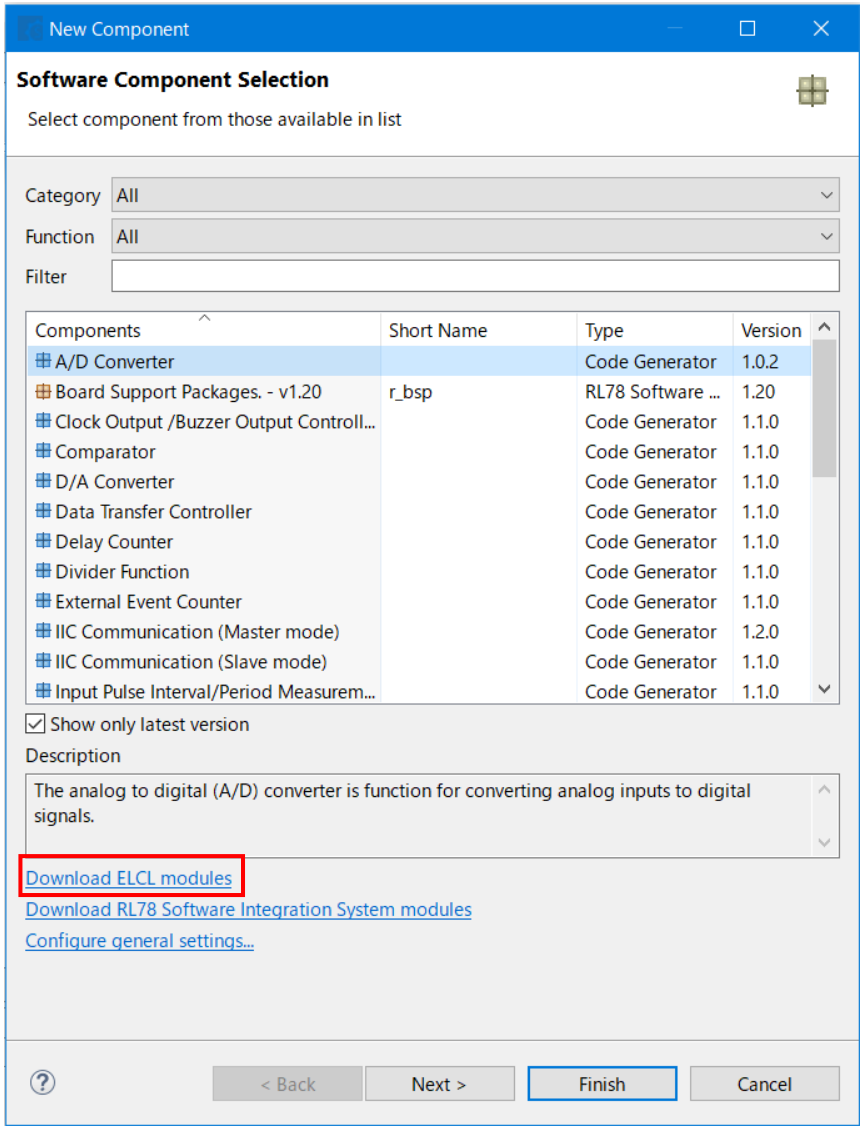
5.1 Setting up the ELCL components

To use the ELCL component, you need to install the ELCL content file.

The procedure is shown below.

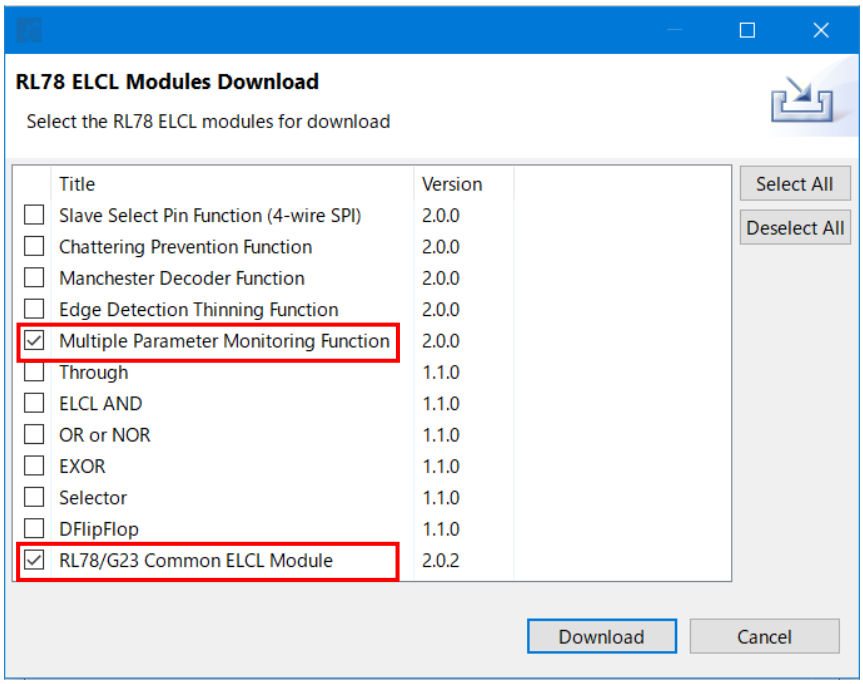
1. Start the Smart Configurator.
2. Click on the "Components" tag, and then click "Add component".
3. When the "New Component" window shown in Figure 5-1 opens, click on "Download ELCL modules".

Figure 5-1 Add component



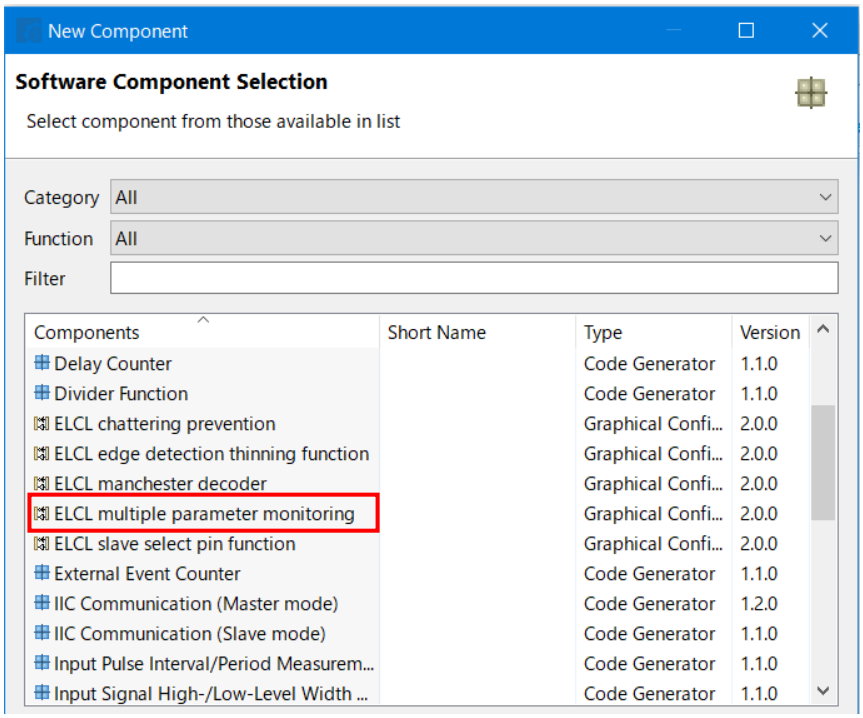
4. Select "Multiple Parameter Monitoring Function" and download it. Please download the common setting file "RL78/G23 Common ELCL Module" as well.

Figure 5-2 Download the module



5. After the download is complete, make sure that " ELCL multiple parameter monitoring" is available for selection.

Figure 5-3 Select the module



5.2 r01an5615_elcl_parameter.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 5-1 Parameters of Smart Configurator (1/2)

Tag name	Component	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}: 16MHz (High-speed on-chip oscillator) f_{SXP}: 32.768kHz (Low-speed on-chip oscillator)
System	-	On-chip debug operation setting: COM port Note 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}): 1.86 (V)
	Config_INTC	INTP0 setting: use Valid edge: Falling edge Priority: Level 3

Note. When using IAR, use by the following settings.
On-chip debug operation setting: Use emulator
Emulator setting: E2 Emulator Lite

Table 5-2 Parameters of Smart Configurator (2/2)

Tag name	Component	Contents
Components	Config_TAU0_0	Components: Interval timer Operating mode: 16 bit count mode Resource: TAU0_0 Operation clock: CK00 Clock source: $f_{CLK}/2^{12}$ Interval value: 8000 count Interrupt setting: unused
	Config_TAU0_1	Components: Interval timer Operating mode: 16 bit count mode Resource: TAU0_1 Operation clock: CK00 Clock source: $f_{CLK}/2^{12}$ Interval value: 4000 count Interrupt setting: unused
	Config_TAU0_2	Components: Interval timer Operating mode: 16 bit count mode Resource: TAU0_2 Operation clock: CK00 Clock source: $f_{CLK}/2^{12}$ Interval value: 2000 count Interrupt setting: unused
	Config_TAU0_3	Components: Interval timer Operating mode: 16 bit count mode Resource: TAU0_3 Operation clock: CK00 Clock source: $f_{CLK}/2^{12}$ Interval value: 1000 count Interrupt setting: unused
	Config_TAU0_7	Components: Interval timer Operating mode: 16 bit count mode Resource: TAU0_3 Operation clock: CK00 Clock source: $f_{CLK}/2^{12}$ Interval value: 1 ms Interrupt setting: Level 2
	Config_MultipleParameterMonitoring	Components: ELCL multiple parameter monitoring Source A：ELISEL_6 , INTTM00 Source B：ELISEL_7 , INTTM01 Source C：ELISEL_8 , INTTM02 Source D：ELISEL_9 , INTTM03 Output signal selector: INTELCL
	Config_PORT	Components: Port Port selection: PORT5 P53: Out (Output 1)

5.2.1 Clocks

Set the clock used in the sample code.

5.2.2 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 4-4 Option Byte Settings". Note that changing the settings.

5.2.3 r_bsp

Set the startup of the sample code.

5.2.4 Config_LVD0

Set the power management of the sample code.

Affects "Setting of LVD0" in "Table 4-4 Option Byte Settings". Note that changing the settings.

5.2.5 Config_INTC

Set the interrupt settings of the sample code.

The sample code sets the external maskable interrupt (INTP0); delete it if you do not use INTP0.

5.2.6 Config_TAU0_0、Config_TAU0_1、Config_TAU0_2、Config_TAU0_3

Set the TAU00, TAU01, TAU02, and TAU03 of the sample code.

In the sample code, it is used as an "interval timer" and the interval time is changed for each channel.

Changing the interval time will change the interval at which the LED blinks.

TAU00, TAU01, TAU02, TAU03, and individual interrupts are not used.

5.2.7 Config_TAU0_7

Set the TAU07 of the sample code.

In the sample code, it is used as a chattering countermeasure for INTP0; delete it if you do not use INTP0 or do not need the chattering countermeasure.

5.2.8 Config_MultipleParameterMonitoring

Initialize the ELCL in the sample code.

In the sample code, INTTM00, INTTM01, INTTM02, and INTTM03 are selected for input, and INTELCL is selected for output.

See Section 5.3 Component “ELCL multiple parameter monitoring” for details.

5.2.9 Config_PORT

Set the port of the sample code.

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

5.3 Component “ELCL multiple parameter monitoring”

Figure 5-4 shows the component “ELCL multiple parameter monitoring” and Table 5-3 shows the options for this component.

Figure 5-4 Component "ELCL multiple parameter monitoring"

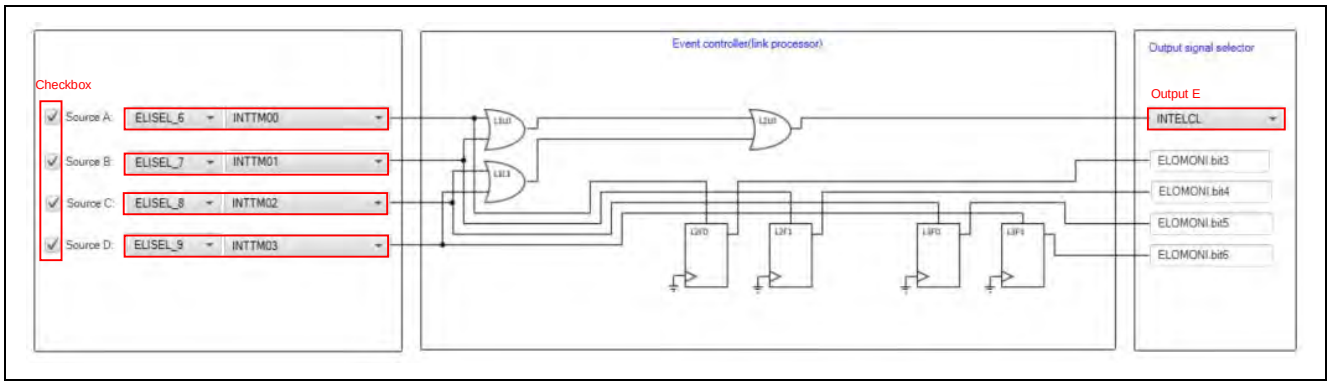


Table 5-3 Choices of component "ELCL multiple parameter monitoring"

Item	Choices		Description
Checkbox	Check: Input factor is valid Uncheck: Input factor is invalid (Low level Clamp)		Select valid / invalid (Low level Clamp) input factor
Source A Source B	ELISEL_0 - 11[Note 1]	Signals listed in Table 5-4, Table 5-5, Table 5-6	Select an input factor
Source C Source D	ELISEL_6 - 11[Note 1]	Signals listed in Table 5-4, Table 5-6	
Output E	INTELCL		Select the output destination
	DTC startup Trigger		
	SMS startup Trigger		
	A / D converter hardware trigger		
	D / A converter 0 hardware trigger		
	D / A converter 1 hardware trigger		
	CTSU hardware trigger		
	ITL capture Trigger		

Note 1. ELISEL_0 - 11 and ELISEL_6 - 11 correspond to ELISEL0 - 11 and ELISEL6 - 11, respectively.
Note 2. Source A, B must select the input factor and select ELISEL_0 - 11 to link the input factor. Similarly, Source C, D must select ELISEL_6 - 11. Refer to Table 5-4 to Table 5-6 and make the correct settings. If the selected combination is incorrect, a warning will be displayed.

Table 5-4 Choices of input parameter (ELISEL 0 - 11)

Suitable ELISELn register	Input parameter
ELISEL 0 - 11	SAU0 Channel0 Data Output
	SAU0 Channel1 Data Output
	SAU0 Channel0 Communication Clock Output
	SAU0 Channel1 Communication Clock Output
	P11
	P50
	UARTA0 Communication Clock Output
	TAU0 Channel0 Output
	TAU0 Channel1 Output
	TAU0 Channel2 Output
	TAU0 Channel3 Output
	TAU0 Channel4 Output
	TAU0 Channel5 Output
	TAU0 Channel6 Output
	TAU0 Channel7 Output
	P10
	P51
	comparator ch0
	comparator ch1
	UARTA0 Data Output
	P12

Table 5-5 Choices of input parameter (ELISEL 0 - 5)

Suitable ELISELn register	Input parameter
ELISEL0	P20
	P120
	INTP6
	INTURE0/INTTM10
	INTTM16
	INTUT1
ELISEL1	P21
	P121
	INTP7
	INTURE1/INTTM11
	INTTM17
	INTUR1
ELISEL2	P22
	P122
	INTP8
	INTTM12
	INTST3/INTCSI30INTIIC30
	INTIICA1
ELISEL3	P23
	P147
	INTP9
	INTSRE3/INTTM13H
	INTSR3/INTCSI31/INTIIC31
ELISEL4	P30
	P00
	INTKR
	INTTM14
	INTUT0
ELISEL5	P31
	P01
	INTTM13
	INTTM15
	INTUR0
ELISEL 0 - 5	P137

Table 5-6 Choices of input parameter (ELISEL 6 - 11)

Suitable ELISELn register	Input parameter
ELISEL6	INTP0
	INTTM00
	INTTM06
	INTST2/INTCSI20/INTIIC20
	INTSR1/INTCSI11/INTIIC11
	INTSMSE
ELISEL7	INTP1
	INTTM01
	INTITL
	INTSR2/INTCSI21/INTIIC21
	INTSRE1/INTTM03H
	INTP10/INTCMP0
ELISEL8	INTP2
	INTTM02
	INTWDTI
	INTSRE2/INTM11H
	INTREMC
	INTP11/INTCMP1
ELISEL9	INTP3
	INTTM03
	INTRTC
	INTST0/INTCSI00/INTIIC00
	INTSR0/INTCSI01/INTIIC01
	INTCTSUWR
ELISEL10	INTP4
	INTTM04
	INTTM07
	INTSRE0/INTTM01H
	INTLVI
	INTCTSURD
ELISEL11	INTP5
	INTTM05
	INTIICA0
	INTST1/INTCSI10/INTIIC10
	INTAD
	INTCTSUFN
ELISEL 6 - 11	CSC PCLK
	HOCO HCLK_out
	MOCO MCLK_out
	SUB_CLK SUBCLK_out

5.3.1 Setting the ELCL Register

Table 5-7 to show Table 5-11 the initial settings of the ELCL register, and Figure 5-5 to Figure 5-9 show the ELCL configuration at that time. Refer to Figure 4-3 for the overall ELCL configuration.

Table 5-7 ELCL register settings (Inputs)

Register Symbol	Register Name	Setting	Description
ELISEL6	Input signal select register 6	16H	Select INTTM00
ELISEL7	Input signal select register 7	16H	Select INTTM01
ELISEL8	Input signal select register 8	16H	Select INTTM02
ELISEL9	Input signal select register 9	16H	Select INTTM03

Figure 5-5 Setting of ELCL input

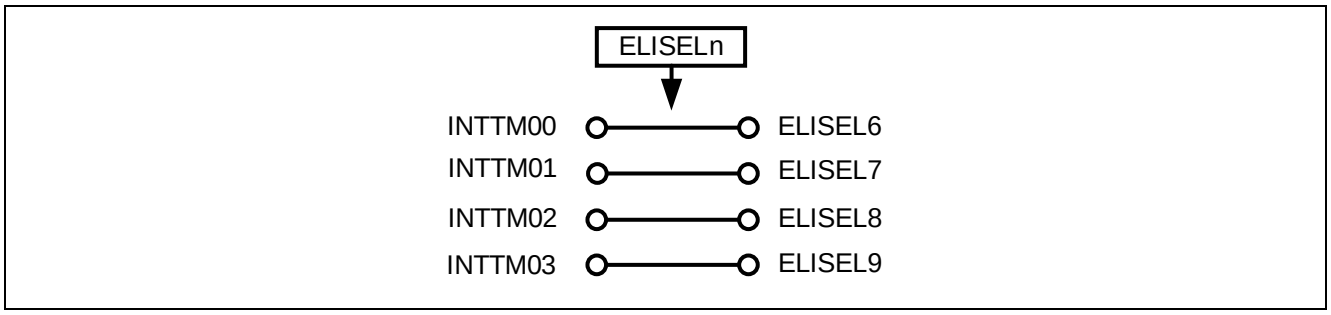


Table 5-8 ELCL register settings (Logic cell block L1)

Register Symbol	Register Name	Setting	Description
ELL1SEL0	Event link L1 signal select register 0	07H	Select the signal selected by ELISEL6 as the link target of L1
ELL1SEL1	Event link L1 signal select register 1	08H	Select the signal selected by ELISEL7 as the link target of L1
ELL1SEL2	Event link L1 signal select register 2	09H	Select the signal selected by ELISEL8 as the link target of L1
ELL1SEL3	Event link L1 signal select register 3	0AH	Select the signal selected by ELISEL9 as the link target of L1
ELL1LNK0	Event link L1 output select register 0	01H	Link target selected by ELL1SEL0 to input 0 of logic cell 0 in logic cell block L1
ELL1LNK1	Event link L1 output select register 1	02H	Link target selected by ELL1SEL1 to input 1 of logic cell 0 in logic cell block L1
ELL1LNK2	Event link L1 output select register 2	03H	Link target selected by ELL1SEL2 to input 0 of logic cell 1 in logic cell block L1
ELL1LNK3	Event link L1 output select register 3	04H	Link target selected by ELL1SEL3 to input 1 of logic cell 1 in logic cell block L1
ELL1CTL	Logic cell block L1 control register	0AH	Logic cell 0 selects OR circuit Logic cell 1 selects OR circuit

Figure 5-6 Setting of logic cells L1

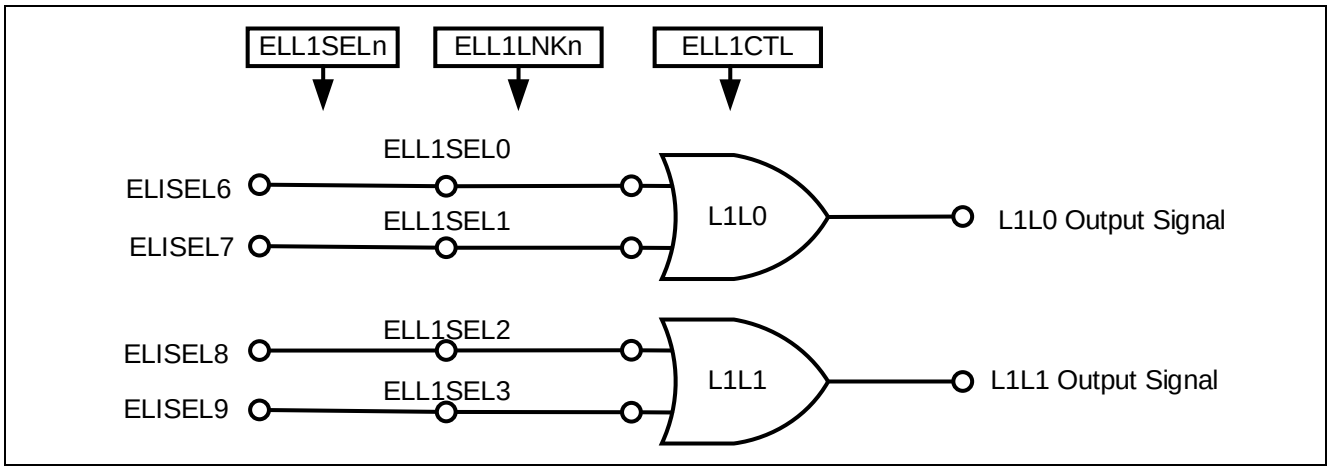


Table 5-9 ELCL register settings (Logic cell block L2)

Register Symbol	Register Name	Setting	Description
ELL2SEL0	Event link L2 signal select register 0	07H	Select the signal selected by ELISEL6 as the link target of L2
ELL2SEL1	Event link L2 signal select register 1	0DH	Output signal 0 in logic cell block L1 is selected as the link target of L2
ELL2SEL2	Event link L2 signal select register 2	0EH	Output signal 1 in logic cell block L1 is selected as the link target of L2
ELL2SEL4	Event link L2 signal select register 4	03H	Select the signal selected by ELISEL8 as the link target of L2
ELL2SEL6	Event link L2 signal select register 6	00H	No selection (fixed to 0)
ELL2LNK0	Event link L2 output select register 0	0AH	Link target selected by ELL2SEL0 to set control of flip-flop 0 in logic cell block L2
ELL2LNK1	Event link L2 output select register 1	01H	Link target selected by ELL2SEL1 to input 0 of logic cell 0 in logic cell block L2
ELL2LNK2	Event link L2 output select register 2	02H	Link target selected by ELL2SEL2 to input 1 of logic cell 0 in logic cell block L2
ELL2LNK4	Event link L2 output select register 4	01H	Link target selected by ELL2SEL4 to set control of flip-flop 1 in logic cell block L2
ELL2LNK6	Event link L2 output select register 6	03H	Link target selected by ELL2SEL6 to clock of flip-flop 0 and 1 in logic cell block L2
ELL2CTL	Logic cell block L2 control register	C2H	Enable use of logic cell block L2 flip-flops 0 and 1, logic cell 0 selects OR circuit

Figure 5-7 Setting of logic cells L2

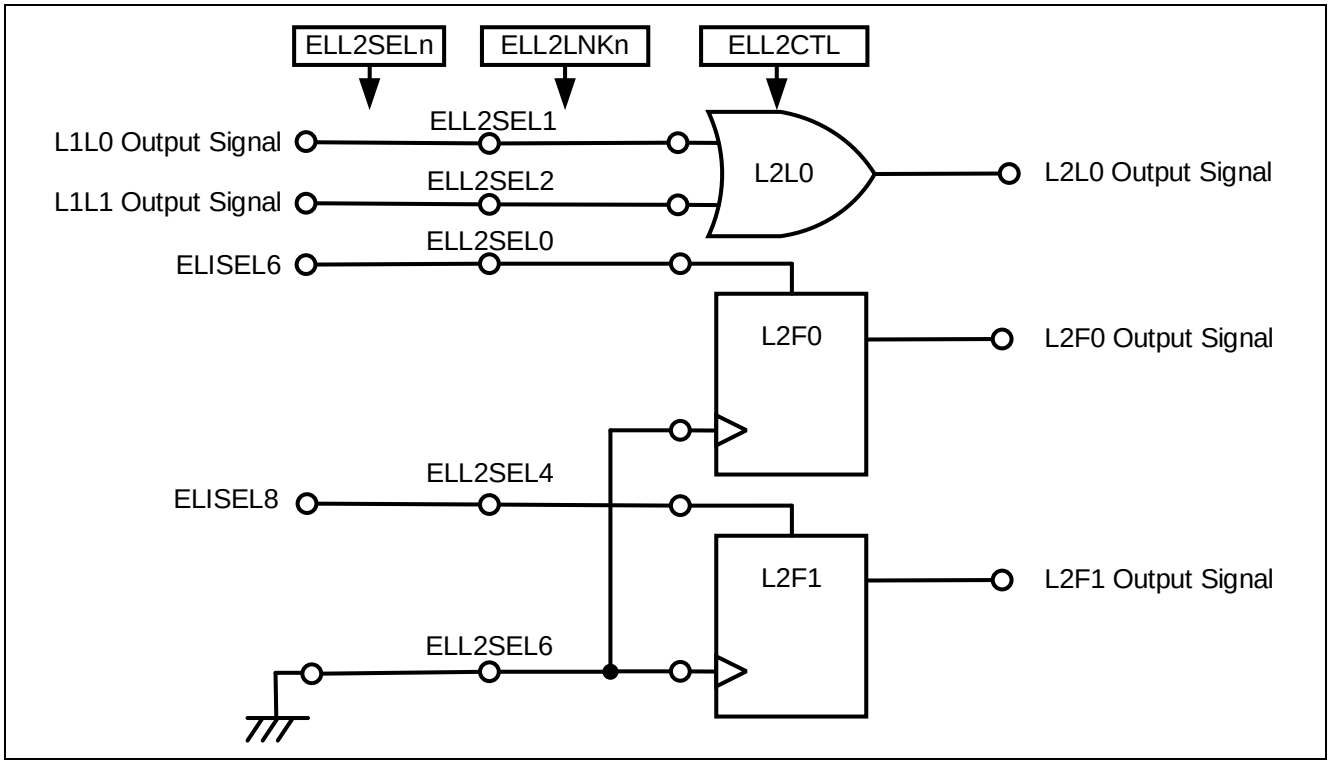


Table 5-10 ELCL register settings (Logic cell block L3)

Register Symbol	Register Name	Setting	Description
ELL3SEL0	Event link L3 signal select register 0	08H	Select the signal selected by ELISEL7 as the link target of L3
ELL3SEL4	Event link L3 signal select register 4	04H	Select the signal selected by ELISEL9 as the link target of L3
ELL3SEL6	Event link L3 signal select register 6	00H	No selection (fixed to 0)
ELL3LNK0	Event link L3 output select register 0	0AH	Link target selected by ELL3SEL0 to set control of flip-flop 0 in logic cell block L3
ELL3LNK4	Event link L3 output select register 4	01H	Link target selected by ELL3SEL4 to set control of flip-flop 1 in logic cell block L3
ELL3LNK6	Event link L3 output select register 6	03H	Link target selected by ELL3SEL6 to clock of flip-flop 0 and 1 in logic cell block L3
ELL3CTL	Logic cell block L3 control register	C0H	Enable use of logic cell block L3 flip-flops 0 and 1

Figure 5-8 Setting of logic cells L3

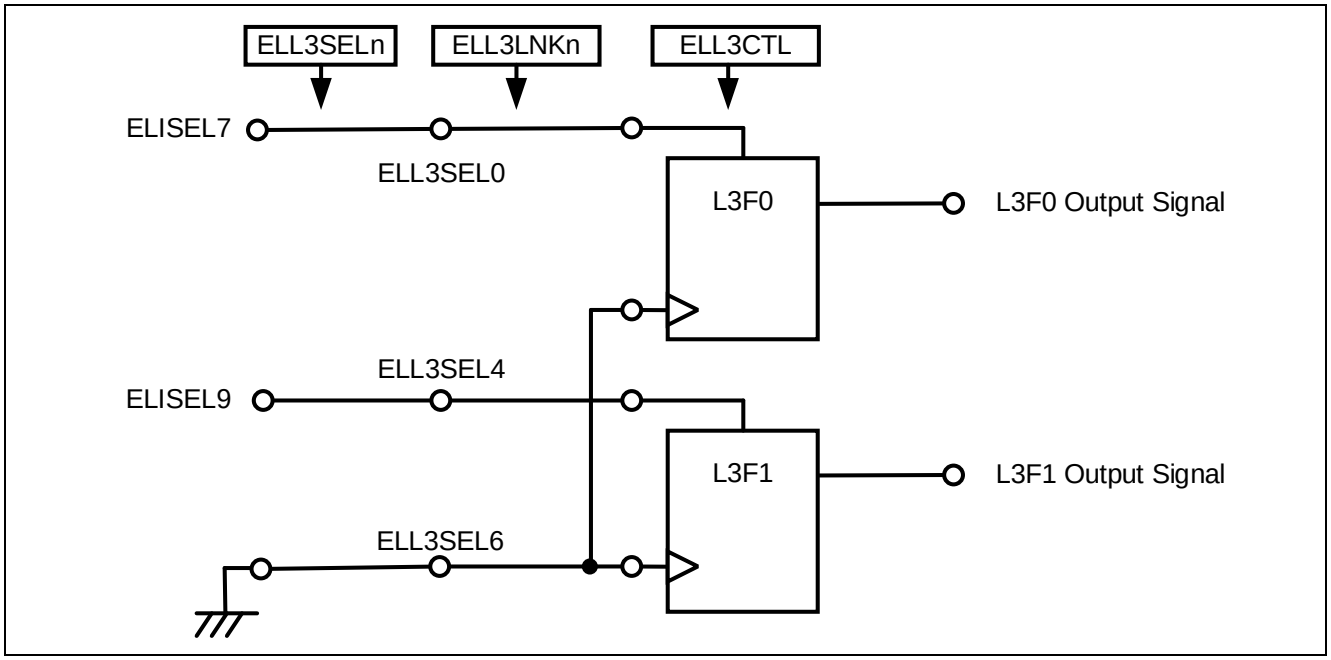
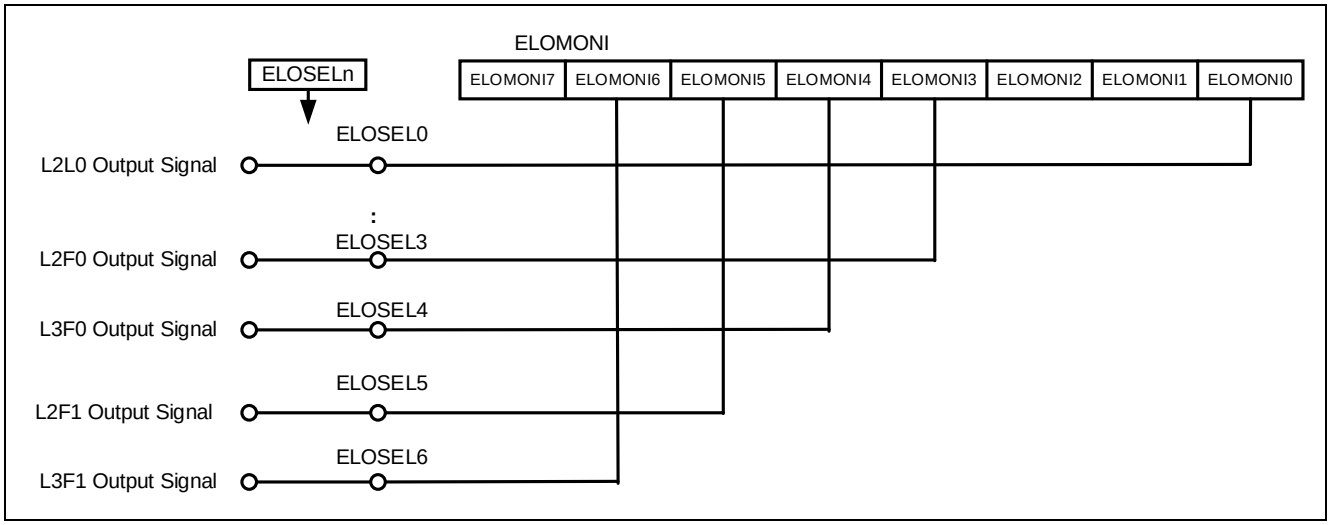


Table 5-11 ELCL register settings (Outputs)

Register Symbol	Register Name	Setting	Description
ELOSEL0	Output signal select register 0	06H	Select the output signal [0] from logic cell block L2.
ELOSEL3	Output signal select register 3	09H	Select the output signal [3] from logic cell block L2.
ELOSEL4	Output signal select register 4	0EH	Select the output signal [3] from logic cell block L3.
ELOSEL5	Output signal select register 5	0AH	Select the output signal [4] from logic cell block L2.
ELOSEL6	Output signal select register 6	0FH	Select the output signal [4] from logic cell block L3.

Figure 5-9 Settings of ELCL output

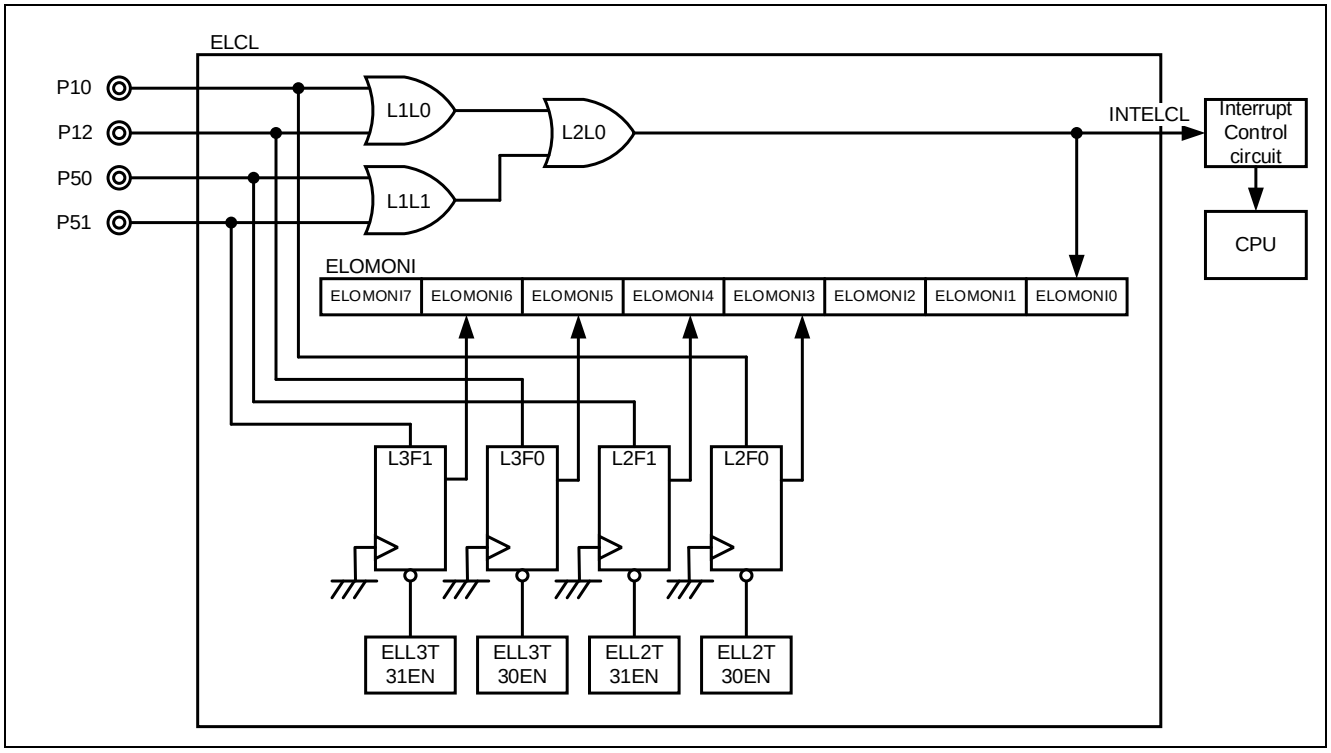


5.3.2 Example of using four ports with no external interrupt function as interrupts

The four ports that do not have external interrupt functions can be used as interrupts (INTELCL).

Figure 5-10 shows an example configuration.

Figure 5-10 ELCL Configuration



Note that the interrupt function realized in this configuration differs from the external interrupt function (INTPx) in the following respects.

INTELCL is an ORed signal of the input pins, and an interrupt is generated when the ORed signal changes from "Low" to "High".

If the code is output from the smart configurator as is, an interrupt will be generated on the rising edge of the input pin. If you want to generate an interrupt on the falling edge, change the input logic to negative logic when inputting the code to the OR circuit. For details, see Section 5.3.3, "How to Make Inputs Negative Logic".

It is not possible to set both rising and falling edges for one input at the same time. Please set one of them.

5.3.3 How to Make Inputs Negative Logic

The component "ELCL multiple parameter monitoring" ORs four inputs with positive logic. The following sample code shows how to change the input to negative logic.

Change the yellow part of the code output by the smart configurator as follows. When the program is automatically generated by the Smart Configurator again, it will be overwritten with the output value of the Smart Configurator (before the change). Please change the value each time you automatically generate the program.

Function name：R_Config_MultipleParameterMonitoring_Create	
change before	**after change**

void
R_Config_MultipleParameterMonitoring_Create
(void)
{
uint8_t p_isel0;
uint8_t p_isel1;
uint8_t p_isel2;
uint8_t p_isel3;
uint8_t p_isel2u;
uint8_t p_isel3u;
uint8_t p_osel0;
uint8_t p_osel1;
uint8_t p_osel2;
ELISEL6=0x16U;
p_isel0=0x07U;
ELISEL7=0x16U;
p_isel1=0x08U;
ELISEL8=0x16U;
p_isel2=0x09U;
p_isel2u=0x03U;
ELISEL9=0x16U;
p_isel3=0x0AU;
p_isel3u=0x04U;
ELL1SEL0=p_isel0;
ELL1SEL1=p_isel1;
ELL1SEL2=p_isel2;
ELL1SEL3=p_isel3;
ELL1LNK0=0x01U;
:
	void
	R_Config_MultipleParameterMonitoring_Create
	(void)
	{
	uint8_t p_isel0;
	uint8_t p_isel1;
	uint8_t p_isel2;
	uint8_t p_isel3;
	uint8_t p_isel2u;
	uint8_t p_isel3u;
	uint8_t p_osel0;
	uint8_t p_osel1;
	uint8_t p_osel2;
	ELISEL6=0x16U;
	p_isel0=0x17U;
	ELISEL7=0x16U;
	p_isel1=0x18U;
	ELISEL8=0x16U;
	p_isel2=0x19U;
	p_isel2u=0x03U;
	ELISEL9=0x16U;
	p_isel3=0x1AU;
	p_isel3u=0x04U;
	ELL1SEL0=p_isel0;
	ELL1SEL1=p_isel1;
	ELL1SEL2=p_isel2;
	ELL1SEL3=p_isel3;
	ELL1LNK0=0x01U;
	:

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference

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

RL78 Family User's Manual: Software (R01US0015E)

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)

(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
2.00	Mar.24.22	6	Table 2-1 Operation Confirmation Conditions Operating voltage Rising edge TYP.1.875V -> 1.90V Falling edge TYP.1.835V -> 1.86V
		6	Updated tool version Table 2-1 Operation Confirmation Conditions Integrated development environment (CS+) : E8.05.00f -> V8.07.00 C compiler (CS+) : V1.09.00 -> V1.11 Integrated development environment (e^2 studio) : 2021-01 (21.01.0) -> 2022-01 (22.1.0) C compiler (e^2 studio) : V1.09.00 -> V1.11 Integrated development environment (IAR) : V4.20.1 -> V4.21.1 Smart Configurator : V.1.0.0 -> V.1.2.0 Board support package (r_bsp) : V.1.0.0 -> V.1.13
		6-7, 24	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 3-1 Hardware Configuration Added P11/TOOLRxD and P12/TOOLTxD Table 5-1 Smart Configurator Settings Note added
		11-12	Updated the folder structure in Table 4-2 due to the sample program update. Added Note 3 due to the update of the IAR version sample code.
		12	Table 4-4 Option byte setting Detection voltage Rise 1.875V / Fall 1.835V -> Rise 1.90V / Fall 1.86V
		13	Updated Table 4-6 Global variables used in the sample code g_led_flag contents Toggle LED -> Polling it in main function to toggle LED
		13	Update with component "ELCL multiple parameter monitoring" update g_led_flag, g_timer_mode Functions used in r_elcl_interrupt -> r_Config_MultipleParameterMonitoring_interrupt
		13	Update with component "ELCL multiple parameter monitoring" update Updated Table 4-7 Functions
		14-15	Update with component "ELCL multiple parameter monitoring" update Updated 4.7 Function Specifications

Rev.	Date	Description	
		Page	Summary
2.00	Mar.24.22	16-23, 28	Updated some figures as follows due to the component "ELCL multiple parameter monitoring" update Figure 4-4 Main process Function name: R_Config_MultipleParameterMonitoring_Create () -> R_Config_ MultipleParameterMonitoring_Start () ELCLIF = 0 ELCLMK = 0 -> deleted Figure 4-10 ELCL interrupt process Function name: r_elcl_interrupt() -> r_Config_MultipleParameterMonitoring_interrupt() Figure 4-8 ELCL output start process Figure 4-9 ELCL output stop process Delete flowchart Figure 4-10 ELCL interrupt process Figure number: Figure 4-10 -> Figure 4-8 Figure 4-11 ELCL flip-flop reset process Figure number: Figure 4-11 -> Figure 4-9 Figure 4-12 INTP0 interrupt process Figure number: Figure 4-12 -> Figure 4-10 Figure 4-13 Wait process Figure number: Figure 4-13 -> Figure 4-11 Figure 4-14 TAU0 channel 7 interrupt process Figure number: Figure 4-14 -> Figure 4-12 Figure 5-1 Add component Figure 5-2 Download the module Figure 5-3 Select the module Figure 5-4 Component "ELCL multiple parameter monitoring" Figure update
		23, 25, 27-28	Updated the component name to the latest. ELCL Multiple Parameter Monitoring Function -> ELCL multiple parameter monitoring
		24	Table 5-1 Parameters of Smart Configurator (1/2) Clock: f_{SXL} -> f_{SXP} Component: Config_LVD0 Reset generation voltage (V_{LVD0}): 1.835 (V) -> 1.86 (V)
		25	Updated the contents in Table 5-2 with the component "ELCL multiple parameter monitoring" update. Table 5-2 Parameters of Smart Configurator (2/2) Component: Config_MultipleParameterMonitoring Source A：INTTM00 -> ELISEL_6 , INTTM00 Source B：INTTM01 -> ELISEL_7 , INTTM01 Source C：INTTM02 -> ELISEL_8 , INTTM02 Source D：INTTM03 -> ELISEL_9 , INTTM03

Rev.	Date	Description	
		Page	Summary
2.00	Mar.24.22	28	Updated the contents in Table 5-3 with the component "ELCL multiple parameter monitoring" update. Table 5-3 Choices of component "ELCL multiple parameter monitoring" Item Added Checkbox Choices Added Source A, B, C, D, Output E choices Added Note 1,Note 2
		29-31	Added the tables of input parameter choices due to the component "ELCL multiple parameter monitoring" update. Table 5-4 Choices of input parameter (common to ELISEL 0 to 11) Table 5-5 Choices of input parameter (common to ELISEL 0 to 5) Table 5-6 Choices of input parameter (common to ELISEL 6 to 11)
		38	Updated the smart configurator output code in line with the update of the component "ELCL Multiple Parameter Monitoring". 5.3.3 How to Make Inputs Negative Logic Updated to match the current smart configurator output code
		39	Added of RL78 Smart Configurator User's Guide 7. 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)

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/.