RL78/G23

WS2812B LED Control Using ELCL

Introduction

This application note describes how to use logic and event link controller (ELCL) to implement full-color serial LED control (WS2812B). Using ELCL and the 3-wire serial SPI enables conversion of transmission data to meet the WS2812B standard.

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

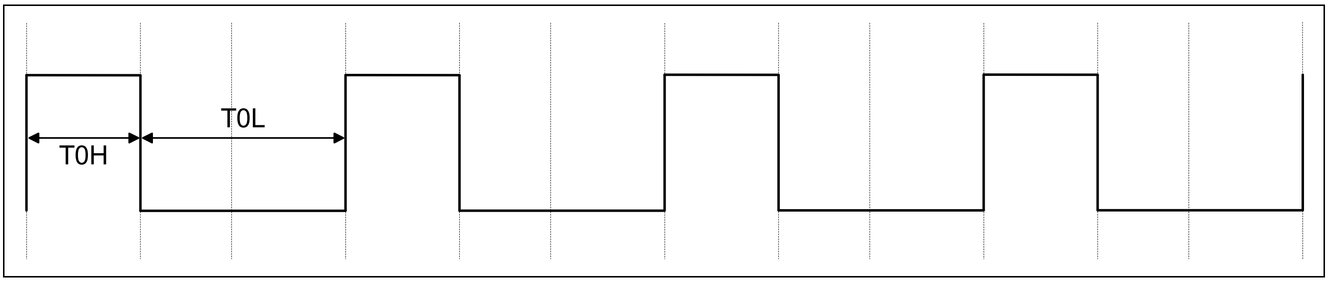
1. Specifications

This application note describes how to use ELCL to implement WS2812B (LED sheet) control.

For the WS2812B communication specifications, LEDs are controlled by 0-code shown in Figure 1-1, 1-code shown in Figure 1-2, and reset code shown in Figure 1-3. Waveforms of the three codes are generated by using timer array unit 0 (TAU0). Channels 1, 2, and 3 are used for 0-code generation, 1-code generation, and reset code generation, respectively. The code waveforms generated by TAU0 and a pattern selection signal generated in the Simlified SPI channel 0 (CSI00) of the serial array unit (SAU) are input to ELCL to control LEDs.

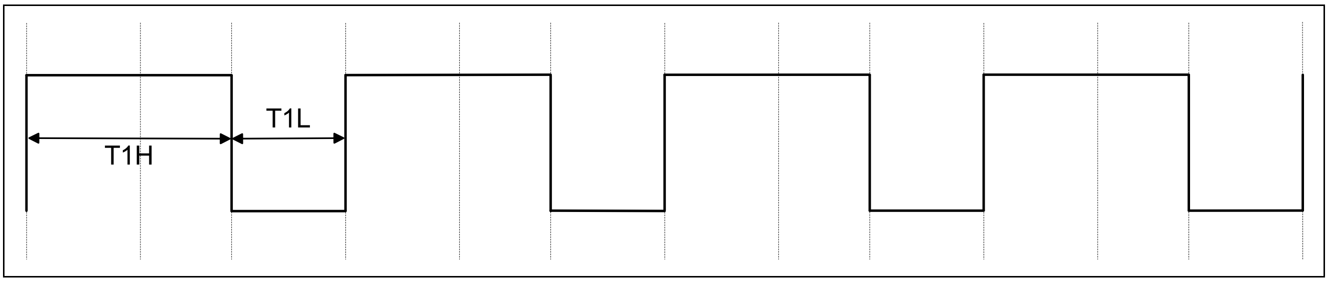
In this system, pressing a switch or detection of a motion sensor turns on the LED sheet. In switch pressing mode, pressing the switch turns on LEDs in the lighting pattern associated with this mode. In motion sensor mode, detection of the detection target by the motion sensor turns on LEDs in the lighting pattern associated with this mode. In addition, if the switch is pressed while the LEDs are lit in motion sensor mode, the lighting pattern changes to that of switch pressing mode.

Figure 1-1 0 code



In Figure 1-1, T0H is in the range from 220 ns to 380 ns (set to 342 ns in this sample code), and T0L is in the range from 580 ns to 1 µs (set to 908 ns in this sample code).

Figure 1-2 1 code



In Figure 1-2, T1H is in the range from 580 ns to 1 µs (set to 800 ns in this sample code), and T1L is in the range from 580 ns to 1 µs (set to 450 ns in this sample code).

Figure 1-3 Reset code

In Figure 1-3, the reset code is 280 µs or longer.

Figure 1-4 shows the system configuration that implements LED control of WS2812B communication by using ELCL.

Figure 1-4 System configuration

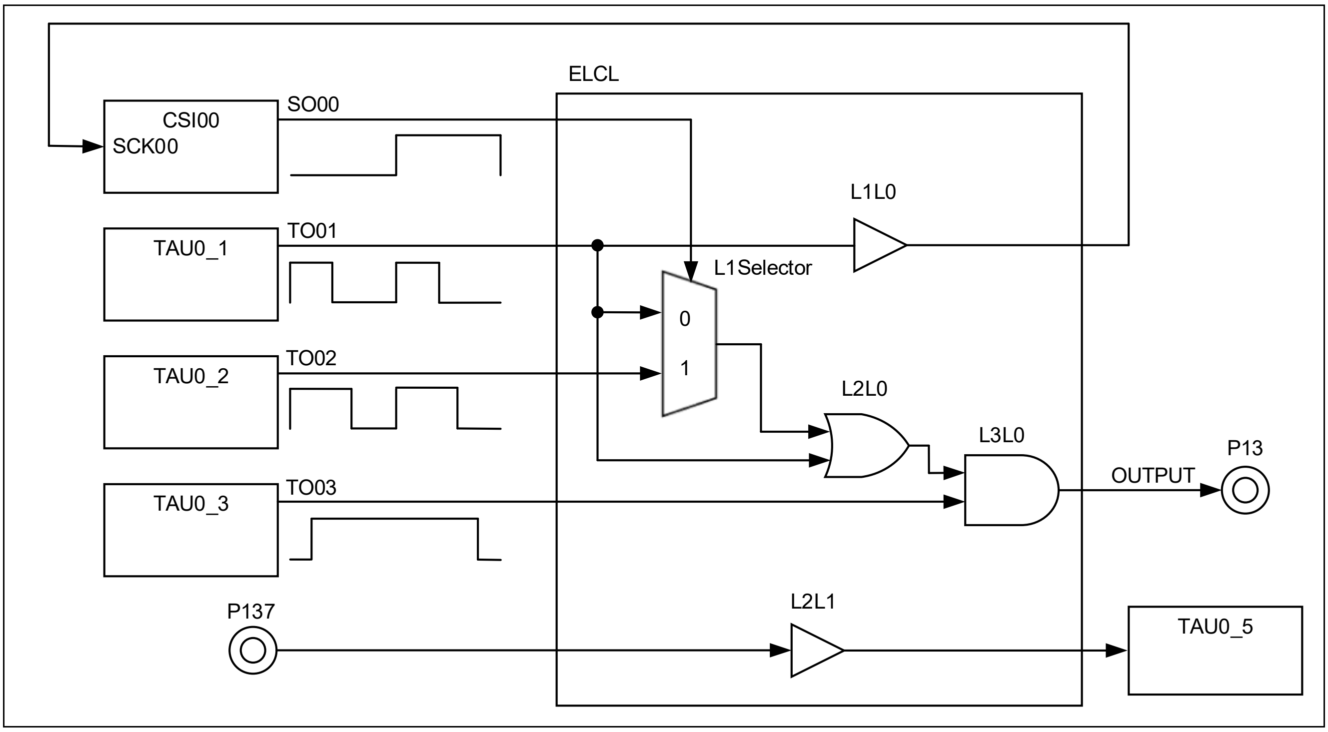
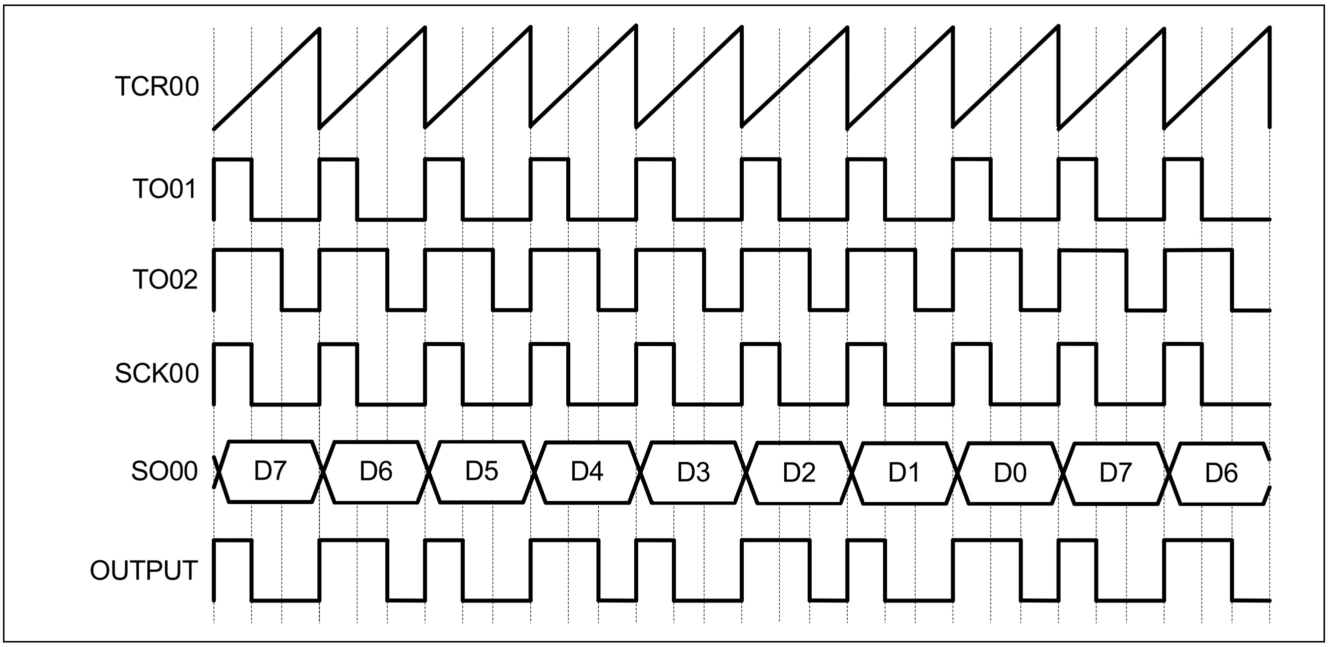


Figure 1-5 shows the timing chart.

(1) TAU0 channel 0 is used as the master channel and channels 1 and 2 are used as slave channels to configure the multiple PWM output function.
(2) TO01 is input to SCI SCK00 via ELCL, and SO00 outputs data according to the LED lighting pattern.
(3) OUTPUT is output according to the value of SO00.

Figure 1-5 Timing chart



2. Operating Confirmation Condition

The operation of the sample code provided with this application note has been tested under the following conditions.

Table 2-1 Operation Confirmation Conditions

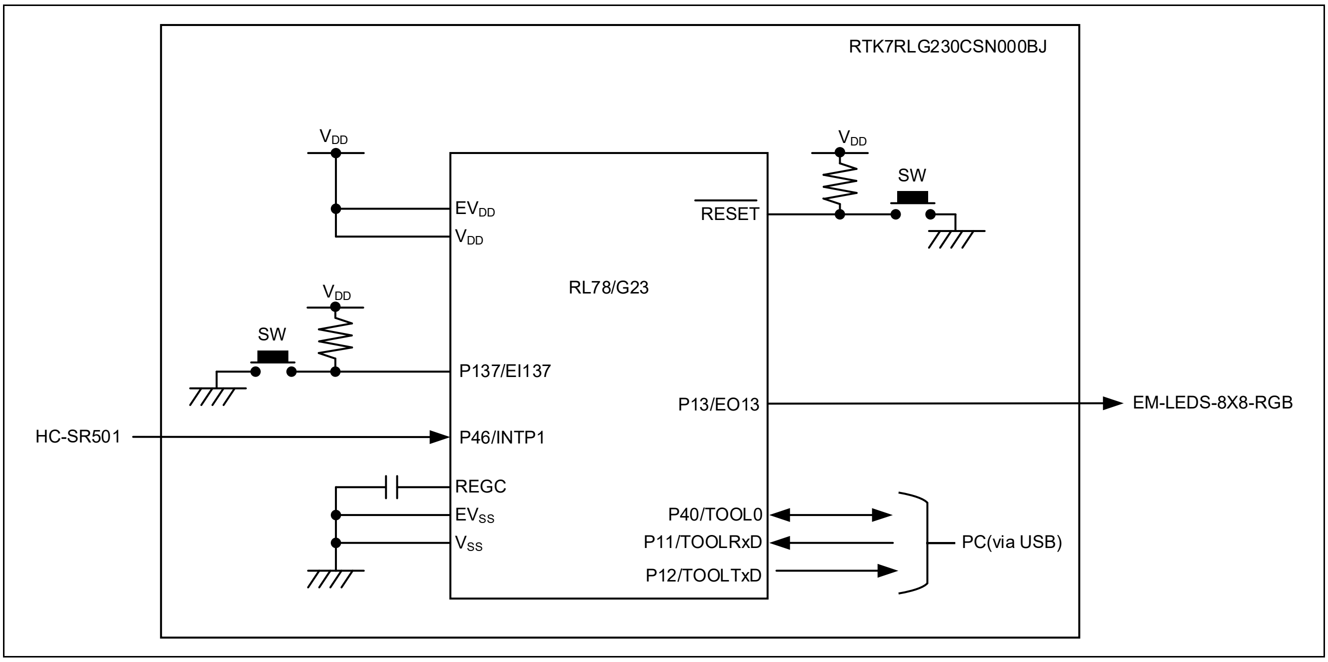
Item	Description
MCU used	RL78/G23（R7F100GSN）
Operating frequency	・ High-speed on-chip oscillator clock (fIH): 32 MHz ・ CPU/peripheral hardware clock: 32 MHz
Operating voltage	・ 5.0V
Integrated development environment (CS+)	CS+ for CC 8.11.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.11.00 from Renesas Electronics Corp.
Integrated development environment (e2 studio)	e2 studio 2024-01.1 (24.1.1) from Renesas Electronics Corp.
C compiler (e2 studio)	CC-RL V1.11.00 from Renesas Electronics Corp.
Integrated development environment (IAR)	IAR Embedded Workbench for Renesas RL78 V 5.10 from IAR Systems Corp.
C compiler (IAR)	
Smart configurator (SC)	V1.9.0
Board support package (r_bsp)	V.1.62
Emulator	e2 studio : COM port IAR : E2 Emulator Lite
Board used	RL78/G23 Fast Prototyping Board (RTK7RLG230CSN000BJ)

3. Hardware Description

3.1 Example of Hardware Configuration

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

Figure 3-1 Hardware Configuration



Note 1. This schematic circuit diagram is simplified to show the outline of connections. When creating actual circuits, design them using appropriate pin processing so that the circuits meet electrical characteristics. (Connect input-only ports to VDD or VSS individually through a resistor.)

Note 2. Connect pins (with a name beginning with EVSS), if any, to VSS, and connect pins (with a name beginning with EVDD), if any, to VDD.

Note 3. Set VDD to a voltage not less than the reset release voltage (VLVD0) set by the LVD0.

3.2 List of Pins to be Used

Figure 3-2 lists the pins to be used and their functions.

Figure 3-2 Pins to be Used and Their Functions

Pin name	I/O	Function
P13/EO13	Output	ELCL output signal (for WS2812 control)
P51/EO51	Output	ELCL output signal (Or output for monitor)
P10/EO10	Output	ELCL output signal (selector output for monitor)
P137/EI137	Input	ELCL output signal (switch)
P46/INTP1	Input	Motion sensor input pin
P16/TO01、P15/TO02、P14/TO03	Output	Monitor pin
P17/SO00	Output	Monitor pin

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 Explanation

4.1 Overview of Operation

This sample code generates control signals for the LED sheet (WS2812B) by using the following four ELCL modules (see Figure 4-1) for the PWM output from TAU0 and the CSI00 data output. The generated signals are output to P13.

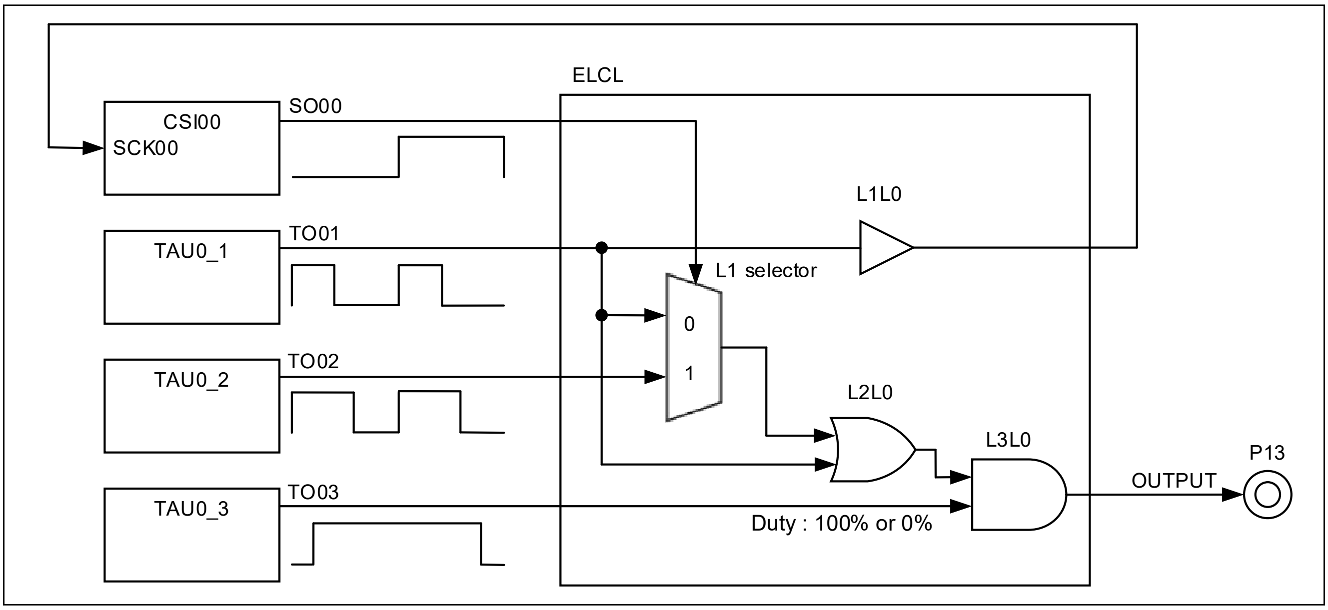
Figure 4-1 shows the system configuration for generating LED sheet (WS2812B) control signals.

Three PWM output signals (TO01, TO02, TO03) from TAU0 are used as input signals to ELCL. SO00 output from CSI00 is also used as input signal to ELCL. TO01 and TO02 are set for PWM signals for 0 code and 1 code, respectively. TO01 is set to the CSI00 clock input as the link destination in Through. TO01 and TO02 are used as input to the ELCL selector, and SO00 is used as the selection control signal of the selector in the link setting.

Output signal of the preceding selector and TO01 are combined by using OR as input signals to L2 of ELCL to prevent glitch noise during selector switching.

TO03 is set as PWM signal for the reset code (duty: 100% or 0%). The preceding OR output signals and TO03 are combined by AND as input signals to L3 of ELCL, and the PWM duty of TO03 is set to 0% so that the reset code can be output. Control signals for the LED sheet (WS2812B) are output from P13 by linking the AND output signals with P13.

Figure 4-1 System Configuration for Generating Control Signals for LED Sheet (WS2812B)



The chattering elimination function is implemented for switch input on the MCU board. P137 input is connected to TI5 of TAU0 by using the Through link. Chattering of P137 input is eliminated by using the TAU0 delay counter function. After chattering is eliminated, processing for switch input is performed by the timer interrupt (INTTM05) vector.

Figure 4-2 System Configuration for the Chattering Elimination Function

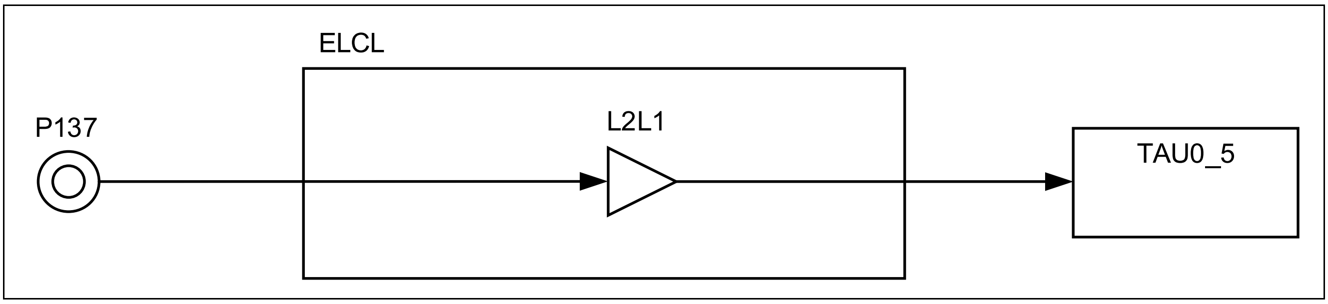
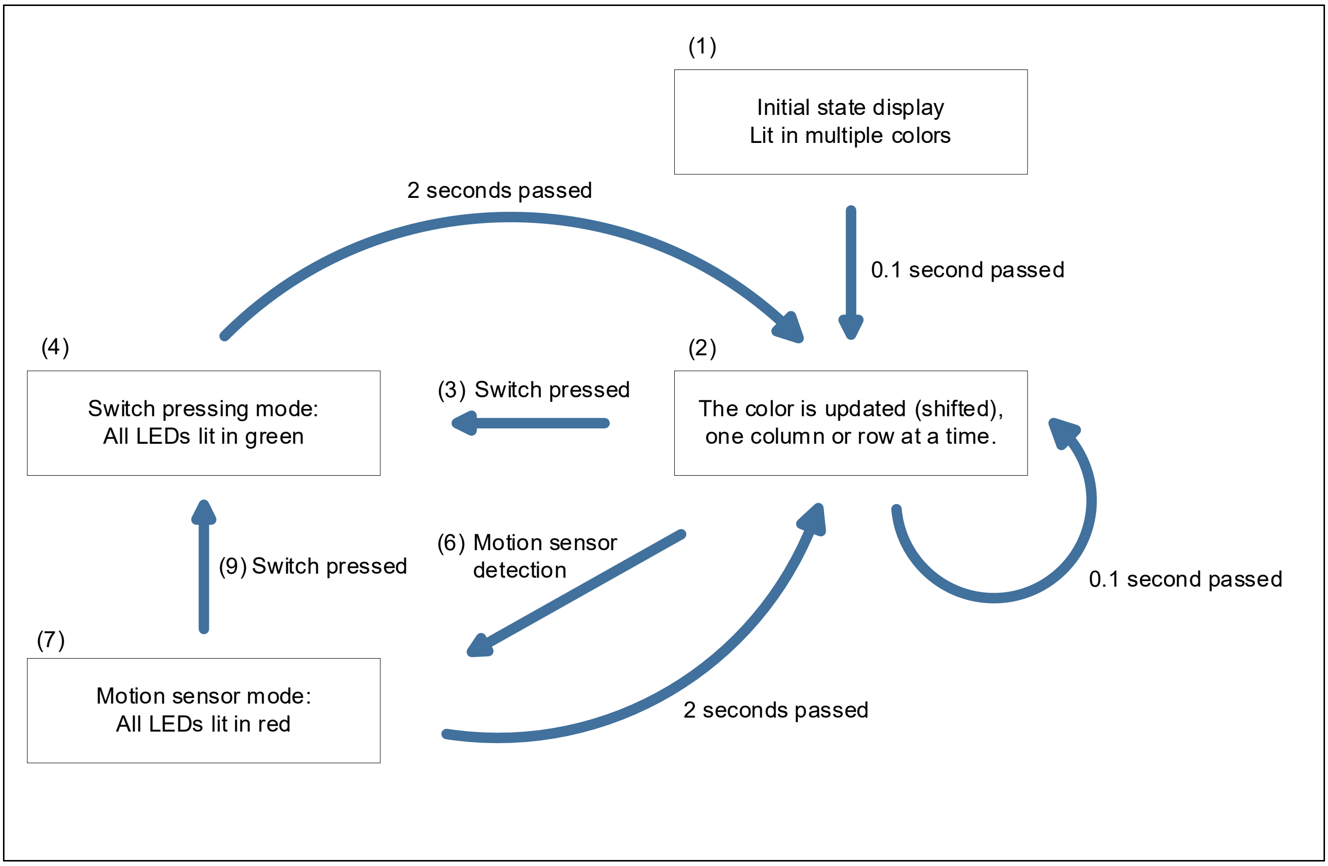


Figure 4-3 shows an overview of operation.

(1) When the operation starts, the LED sheet lights in the initial display state.
(2) The color pattern changes on one row or one column every 0.1 second.
(3) Press the switch.
(4) The operation in (3) causes the system to enter switch pressing mode, and all LEDs are lit in green for 2 seconds.
(5) As with (2), the color pattern changes on one row or one column every 0.1 second.
(6) Make the motion sensor react.
(7) The operation in (6) causes the system to enter to motion sensor mode, and all LEDs are lit in red for 2 seconds.
(8) As with (2), the color pattern changes on one row or one column every 0.1 second.
(9) When pressing of the switch is detected in motion sensor mode, the mode changes to switch pressing mode.

Figure 4-3 Overview of operation



4.2 Folder structure

Table 4-1 and Table 4-2 show the configuration of source files and header files used in this sample code. Note that files that are automatically generated in the integrated development environment and files in the bsp environment are excluded.

Table 4-1 Folder structure (1/2)

Folder/file name	Description	Smart Configurator file
¥r01an7320_elcl<DIR>[注 2]	Folder of this sample code	
¥src<DIR>	Program folder	
main.c	Sample code source file	
main.h	Source code header file	
¥smc_gen<DIR>	Folder created by Smart Configurator	√
¥Config_AND<DIR>	Program folder for AND	√
Config_AND.c	Source file for AND	√
Config_AND.h	Header file for AND	√
Config_AND_user.c	Interrupt source file for AND	√[Note 1]
¥Config_CSI00<DIR>	Program folder for CSI00	√
Config_CSI00.c	Source file for CSI00	√
Config_CSI00.h	Header file for CSI00	√
Config_CSI00_user.c	Interrupt source file for CSI00	√[Note 1]
¥Config_INTC<DIR>	Program folder for INTC	√
Config_INTC.c	Source file for INTC	√
Config_INTC.h	Header file for INTC	√
Config_INTC_user.c	Interrupt source file for INTC	√[Note 1]
¥Config_ITL000_ITL001_ITL012_ITL013<DIR>	Program folder for ITL000_ITL001_ITL012_ITL013	√
Config_ITL000_ITL001_ITL012_ITL013.c	Source file for ITL000_ITL001_ITL012_ITL013	√
Config_ITL000_ITL001_ITL012_ITL013.h	Header file for ITL000_ITL001_ITL012_ITL013	√
Config_ITL000_ITL001_ITL012_ITL013_user.c	Interrupt source file for ITL000_ITL001_ITL012_ITL013	√[Note1]
¥Config_OR<DIR>	Program folder for OR	√
Config_OR.c	OR source file	√
Config_OR.h	OR header file	√
Config_OR_user.c	OR interrupt source file	√[Note1]
¥Config_Selector<DIR>	Program folder for Selector	√
Config_Selector.c	Source file for Selector	√
Config_Selector.h	Header file for Selector	√
Config_Selector_user.c	Interrupt source file for Selector	√[注 1]

Table 4-2 Folder structure (2/2)

Folder/file name	Description	Smart Configurator file
¥Config_TAU0_0<DIR>	Program folder for TAU00	√
Config_TAU0_0.c	Source file for TAU00	√
Config_TAU0_0.h	Header file for TAU00	√
Config_TAU0_0_user.c	Interrupt source file for TAU00	√Note 1
¥Config_TAU0_5<DIR>	Program folder for TAU05	√
Config_TAU0_5.c	Source file for TAU05	√
Config_TAU0_5.h	Header file for TAU05	√
Config_TAU0_5_user.c	Interrupt source file for TAU05	√Note 1
¥Config_Through<DIR>	Program folder for Through	√
Config_Through.c	Source file for Through	√
Config_Through.h	Header file for Through	√
Config_Through_user.c	Interrupt source file for Through	√Note 1
¥general<DIR>	Folder for initialization and common programs	√
¥r_bsp<DIR>	Program folder for BSP	√
¥r_config<DIR>	Program folder	√

Supplementary note: <DIR> means a directory.

Note 1. Not used in this sample code.

Note 2. The IAR version of the sample code contains r01an7320_elcl.ipcf. For details about the ipcf file, refer to "RL78 Smart Configurator User's Guide: IAR (R20AN0581)".

4.3 List of Option Byte Setting

Table 4-3 shows option byte settings.

Table 4-3 Option Byte Setting

Address	Setting value	Contents
000C0H/040C0H	1110 1111B (EFH)	Watchdog time counter operation disable (Counting stopped after reset)
000C1H/040C1H	0011 1010B (3AH)	LVD0 off
000C2H/040C2H	1110 1000B (E8H)	HS mode, High-speed on-chip oscillator clock (fIH): 32 MHz
000C3H/040C3H	1000 0100B (84H)	Enables on-chip debugging

4.4 List of constants

Table 4-4 lists constants

Table 4-4 Constants

Constant name	Setting value	Contents
LED_W	8	LED sheet width
LED_H	8	LED sheet height
RGB	3	Three colors (red, green, and blue)
DISP_NORMAL_TIME	1	LED update time in normal operation
DISP_EVENT_TIME	20	LED update time when an event occurs

4.5 List of Variables

Table 4-5lists global variables.

Table 4-5 Global Variables

Type	Variable Name	Description	Function used
uint8_t	g_buf_csi	CSI transmission buffer (array)	main、shift_row、led_data_send
uint8_t	g_flag_send_end	Transmission completion flag	main、r_Config_CSI00_callback_sendend、led_data_send
uint8_t	g_flag_constperiod	Constant-period-elapsed flag	main、R_Config_ITL000_ITL001_ITL012_ITL013_Callback_Shared_Interrupt
uint8_t	g_flag_switch	Switch pressing flag	main、r_Config_TAU0_5_interrupt
uint8_t	g_flag_sensor	Motion sensor reaction flag	main、r_Config_INTC_intp1_interrupt
uint8_t	g_flag_send_timing	Display update flag	main
uint16_t	g_cnt_time	LED display time counter	main
uint16_t	g_time_disp	LED display time	main
uint8_t	g_row_column	Update target (row or column) indicator	main

4.6 List of Functions

Table 4-6 lists the functions used in the sample code. However, it excludes functions generated by the Smart Configurator that have not been modified.

Table 4-6 Functions

Function name	Outline
main()	Main process
set_color()	Setting LED color data
change_all()	Updating all LED display data
shift_column()	Shifting LED display data by one column
change_column()	Updating a column of LED display data
shift_row()	Shifting LED display data by one row
change_row()	Updating a row of LED display data
led_data_send()	Sending LED display data
r_Config_CSI00_callback_sendend()	CSI transmission completion interrupt
r_Config_INTC_intp1_interrupt()	Motion sensor detection interrupt (external pin)
R_Config_ITL000_ITL001_ITL012_ITL013_Callback_Shared_Interrupt()	Interval timer interrupt
r_Config_TAU0_5_interrupt()	Switch input detection interrupt (chattering filtered)

4.7 Specification of Functions

The function specifications of the sample code are shown below.

main

Outline	Main processing
Header	main.h、r_smc_entry.h
Declaration	int main (void);
Description	This function initializes ELCL, specifies ELCL output setting, and configures interrupts. It then starts operation of the external interrupt, interval timer, CSI00, and TAU0.
Argument	None
Return Vlaue	None
Remark	None

set_color

Outline	Setting LED color data
Header	main.h、r_smc_entry.h
Declaration	void set_color(E_COLOR e_color, uint8_t *g_val, uint8_t *r_val, uint8_t *b_val)
Description	This function sets the specified color data to the specified variable.
Argument	e_color, g_val, r_val, b_val
Return Vlaue	None
Remark	None

change_all

Outline	Updating all LED display
Header	main.h、r_smc_entry.h
Declaration	void change_all(uint8_t buf[][LED_W][RGB], E_COLOR e_color);
Description	This Function update all LED display data with the specified color
Argument	buf[][LED_W][RGB], e_color)
Return Value	None
Remark	None

shift_column

Outline	Shifting LED display data by one column
Header	main.h、r_smc_entry.h
Declaration	void shift_column(uint8_t buf[][LED_W][RGB])
Description	This function shifts LED display data by one column.
Argument	buf[][LED_W][RGB]
Return Value	None
Remark	None

change_column

Outline	Updating a column of LED display data
Header	main.h、r_smc_entry.h
Declaration	void change_column(uint8_t buf[][LED_W][RGB], uint8_t column, E_COLOR e_color);
Description	This function updates the LED display data of the specified column to the specified color.
Argument	None
Return Value	None
Remark	None

shift_row

Outline	Shifting LED display data by one row
Header	main.h、r_smc_entry.h
Declaration	void shift_row(uint8_t buf[][LED_W][RGB])
Description	This function shifts the LED display data by one row.
Argument	buf[][LED_W][RGB]
Return Value	None
Remark	None

change_row

Outline	Updating a row of LED display data
Header	main.h、r_smc_entry.h
Declaration	void change_row(uint8_t buf[][LED_W][RGB], uint8_t row, E_COLOR e_color);
Description	This function updates the LED display data of the specified row to the specified color.
Argument	buf[][LED_W][RGB], row, e_color
Return Value	None
Remark	None

led_data_send

Outline	Sending LED display data
Header	main.h、r_smc_entry.h
Declaration	void led_data_send(void);
Description	This function sends LED display data.
Argument	None
Return Value	None
Remark	None

r_Config_CSI00_callback_sendend

Outline	CSI transmission completion interrupt
Header	r_cg_macrodriver.h
Declaration	static void r_Config_CSI00_callback_sendend(void);
Description	This function sets the transmission completion flag to 1.
Argument	None
Return Value	None
Remark	None

r_Config_INTC_intp1_interrupt

Outline	Motion sensor detection interrupt (external pin)
Header	r_cg_macrodriver.h、r_cg_userdefine.h、Config_INTC.h
Declaration	static void __near r_Config_INTC_intp1_interrupt(void);
Description	This function sets the motion sensor reaction flag to 1 by means of motion sensor detection.
Argument	None
Return Value	None
Remark	None

R_Config_ITL000_ITL001_ITL012_ITL013_Callback_Shared_Interrupt

Outline	Interval timer interrupt
Header	Config_ITL000_ITL001_ITL012_ITL013.h
Declaration	void R_Config_ITL000_ITL001_ITL012_ITL013_Callback_Shared_Interrupt(void);
Description	This function sets the constant-period-elapsed flag to 1.
Argument	None
Return Value	None
Remark	None

r_Config_TAU0_5_interrupt

Outline	Switch input detection interrupt (chattering filtered)
Header	r_cg_macrodriver.h、r_cg_userdefine.h、Config_TAU0_5.h
Declaration	static void __near r_Config_TAU0_5_interrupt(void);
Description	This function sets the switch pressing flag to 1 when the switch is pressed.
Argument	None
Return Value	None
Remark	None

4.8 Flowcharts

4.8.1 Main Processing

Figure 4-4、Figure 4-5、Figure 4-6 および Figure 4-7 shows the flowchart of the main processing.

Figure 4-4 Main Processing (1/4)

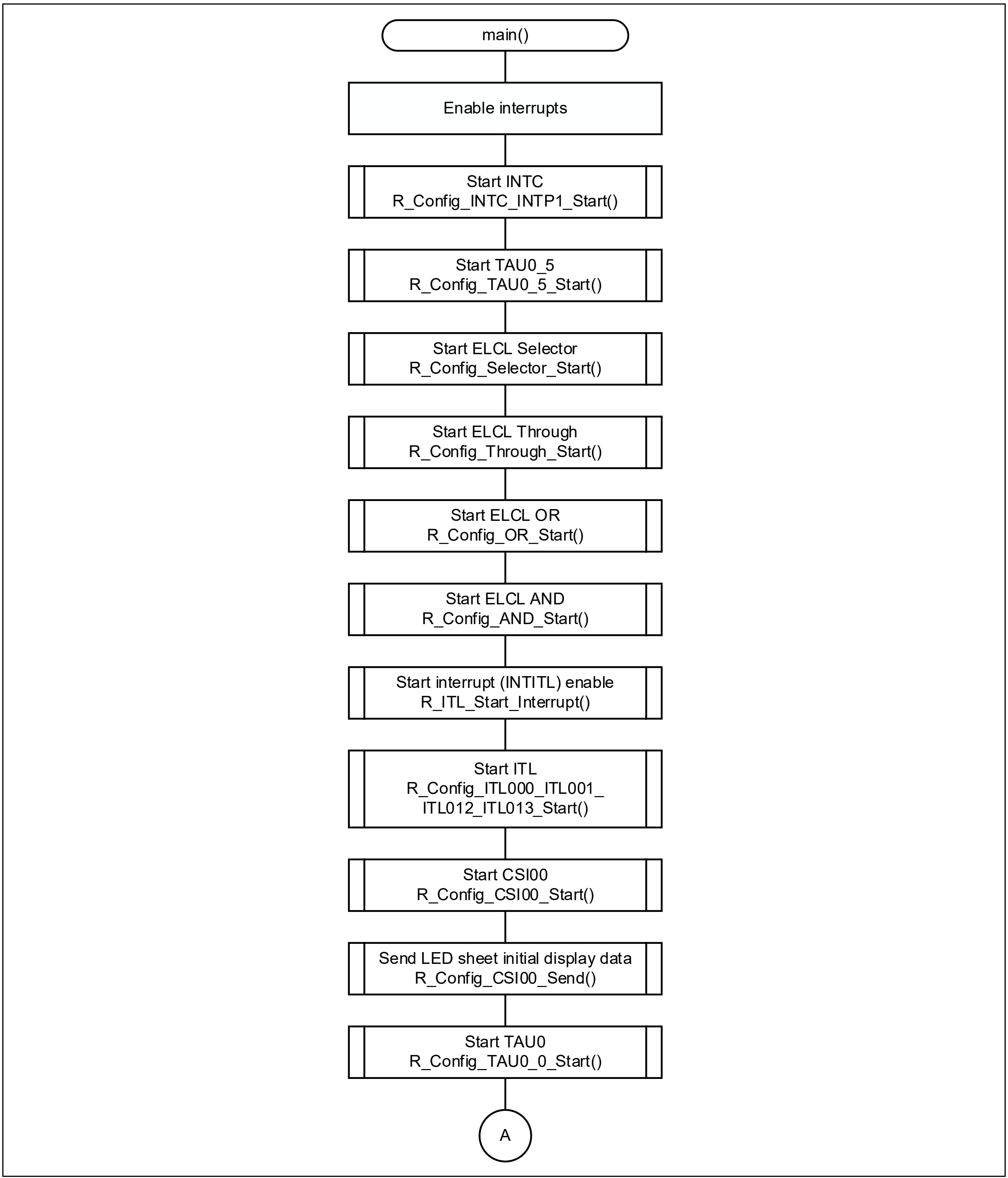


Figure 4-5 Main Processing (2/4)

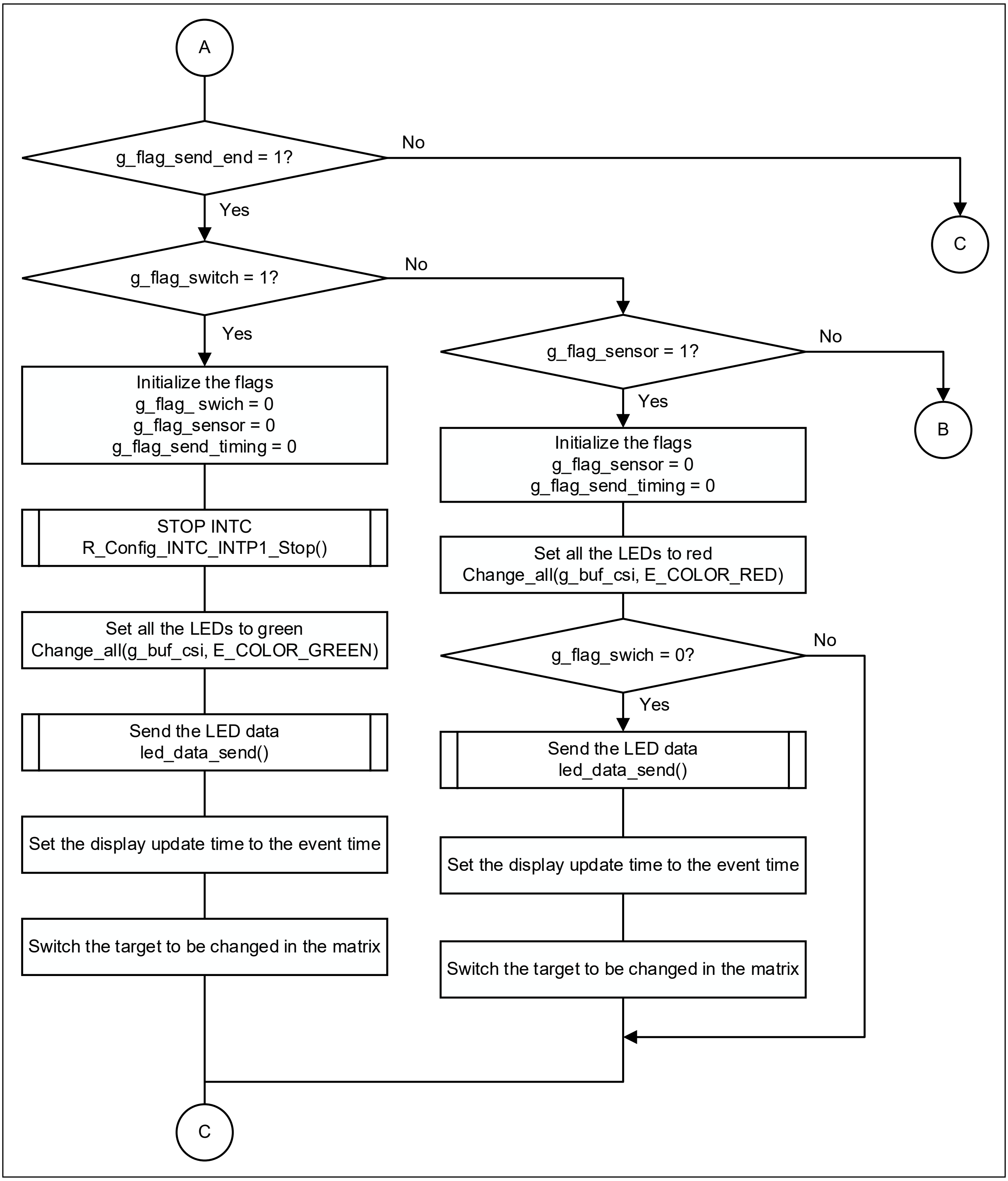


図 4-6 Main Processing (3/4)

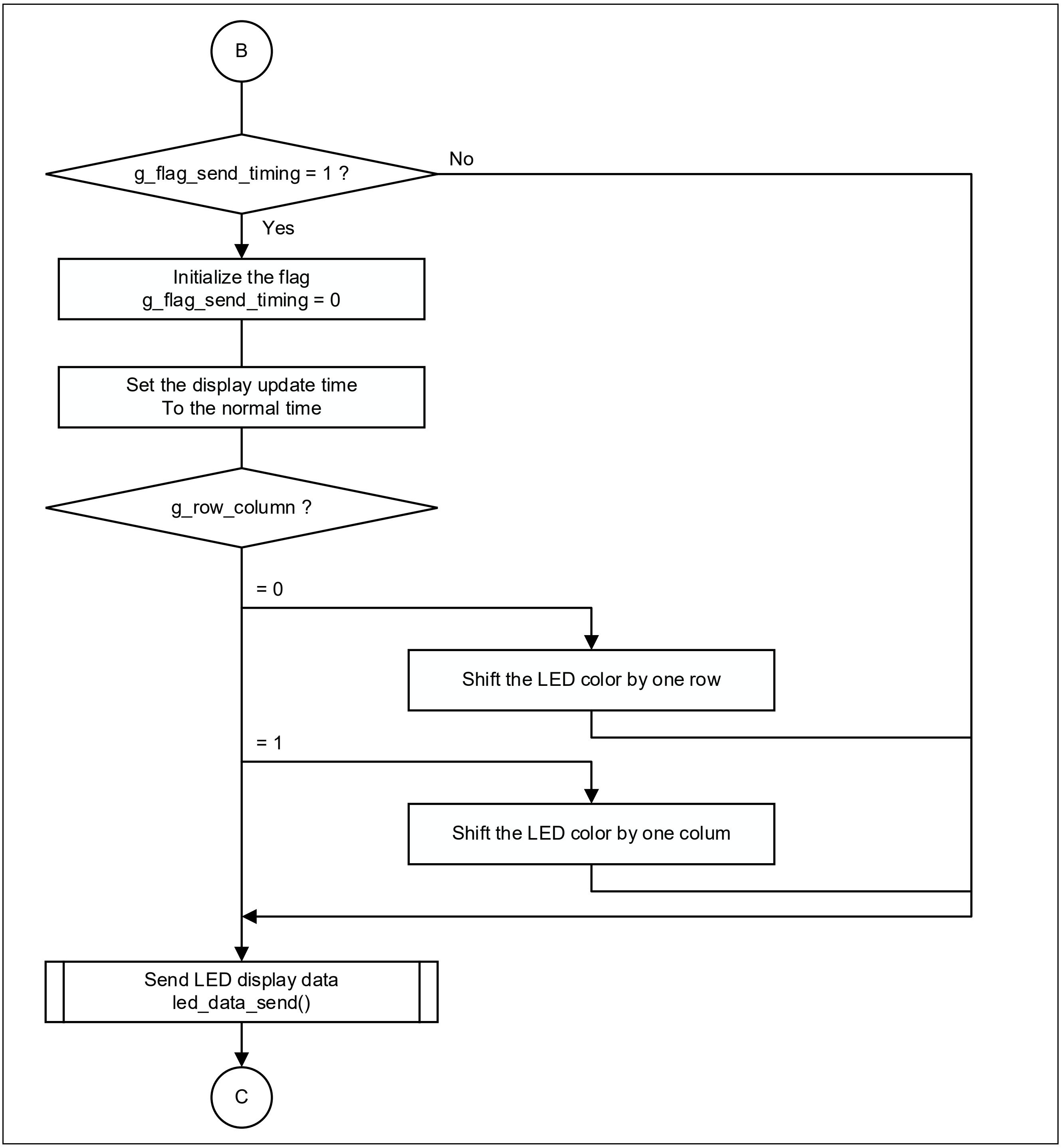
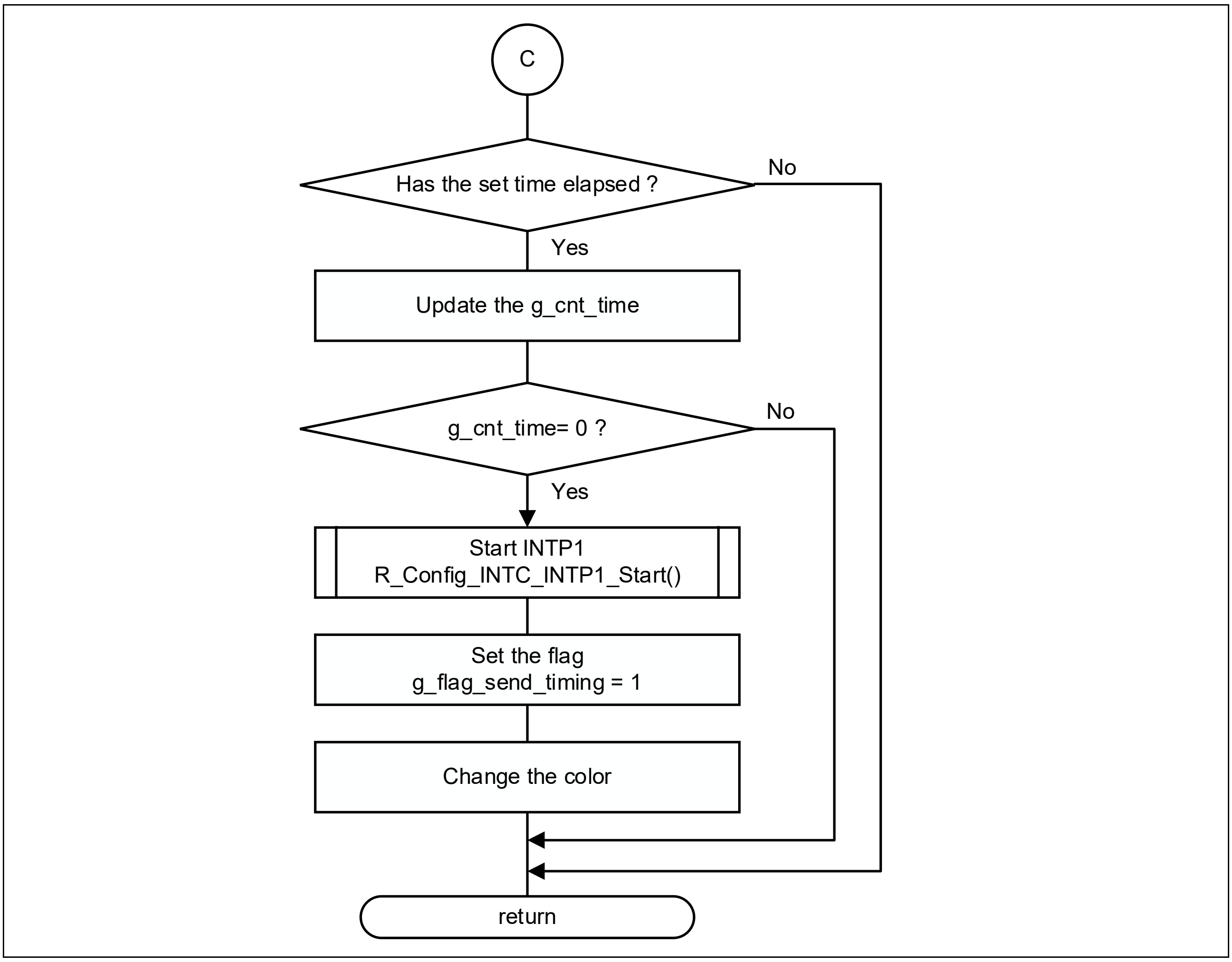


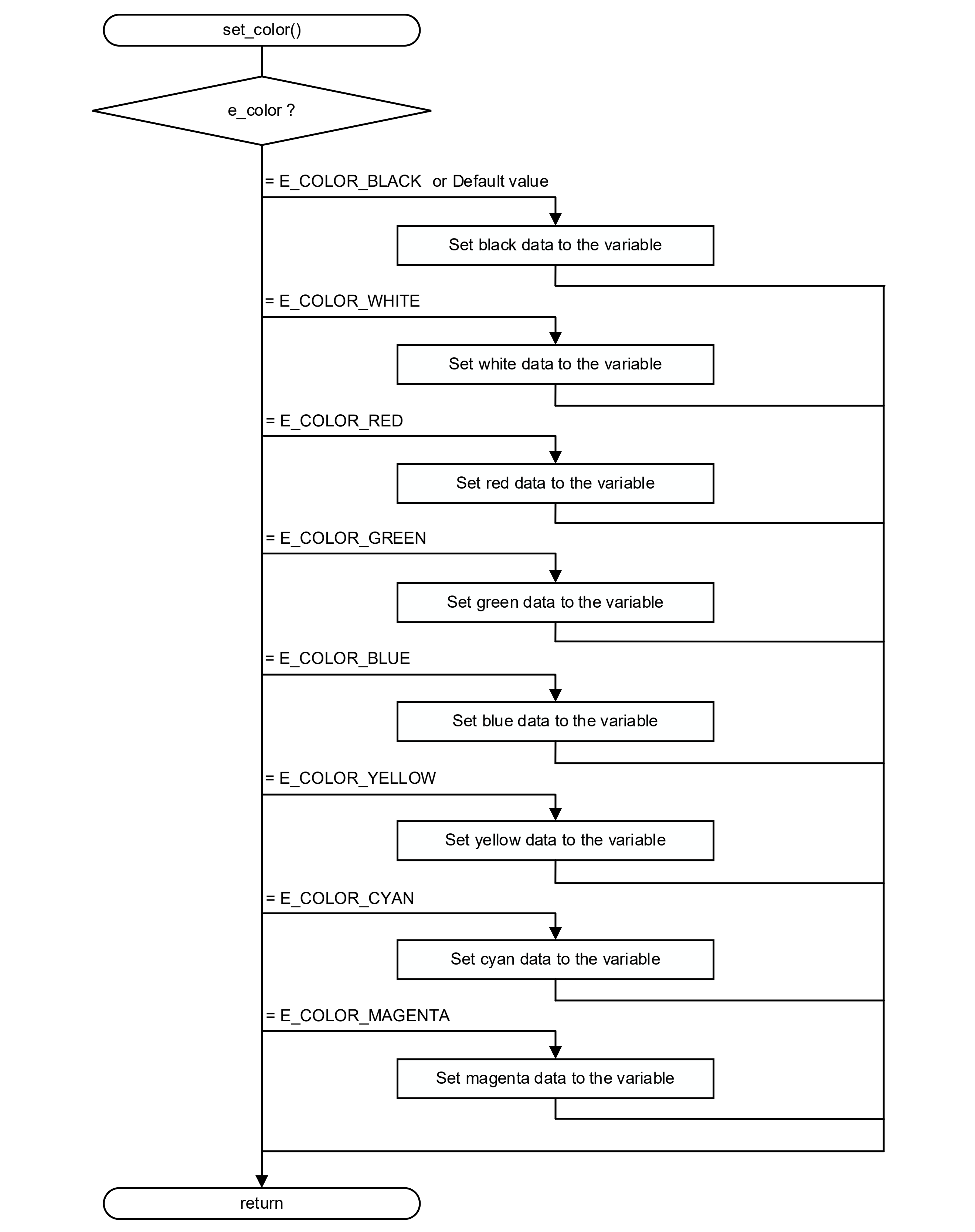
Figure 4-7 Main Processing (4/4)



4.8.2 Setting LED color data

Figure 4-8 shows the flowchart for setting LED color data.

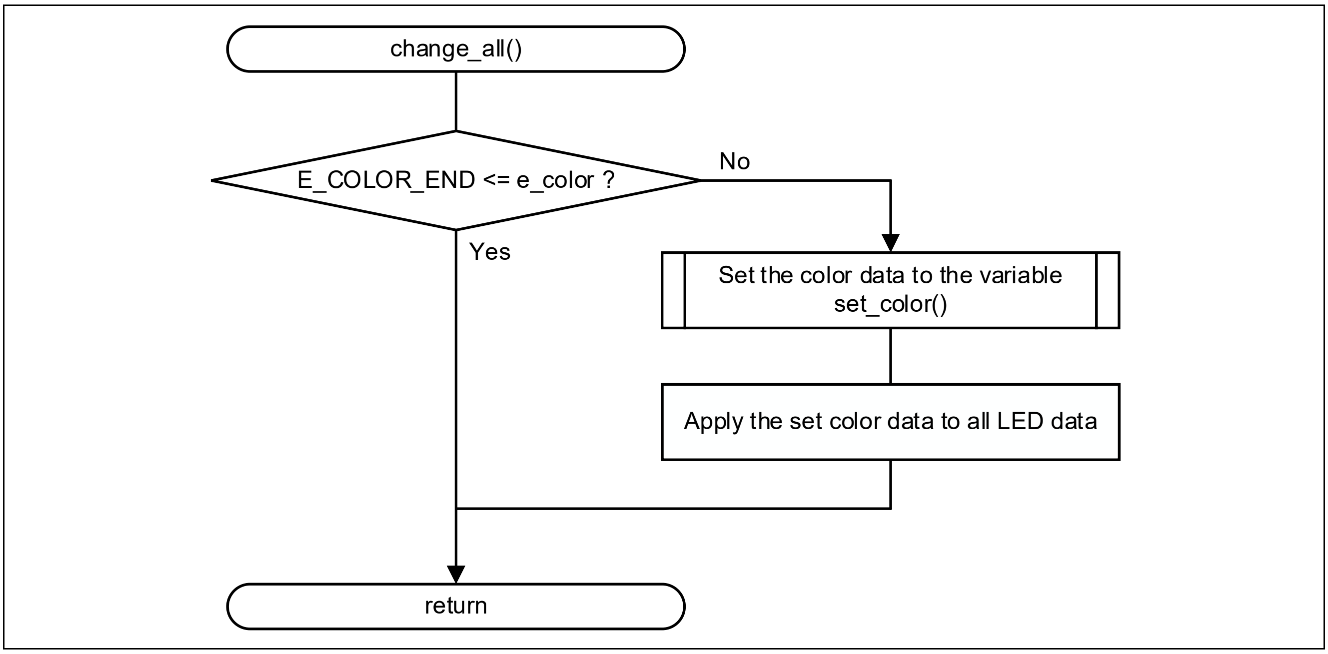
Figure 4-8 Setting LED Color Data



4.8.3 Updating all LED display data

Figure 4-9 shows the flowchart for updating all LED display data.

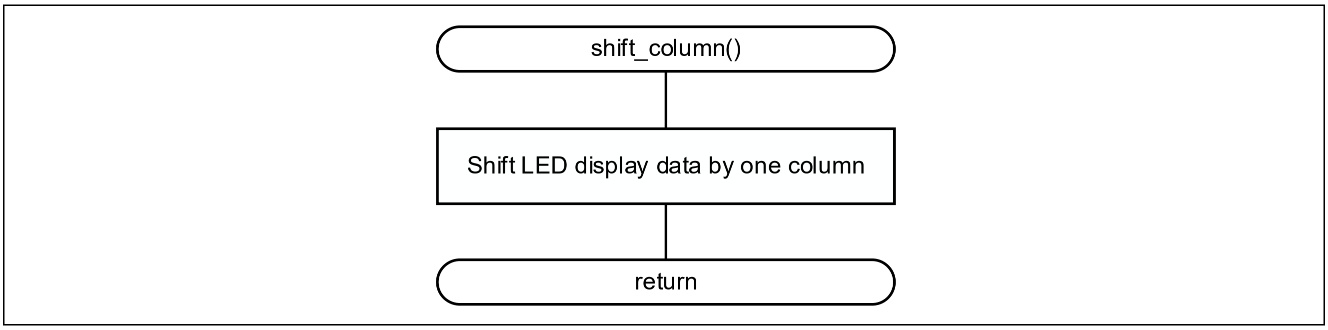
Figure 4-9 Updating All LED Display Data



4.8.4 Shifting LED display data by one column

Figure 4-10 shows the flowchart for shifting LED display data by one column.

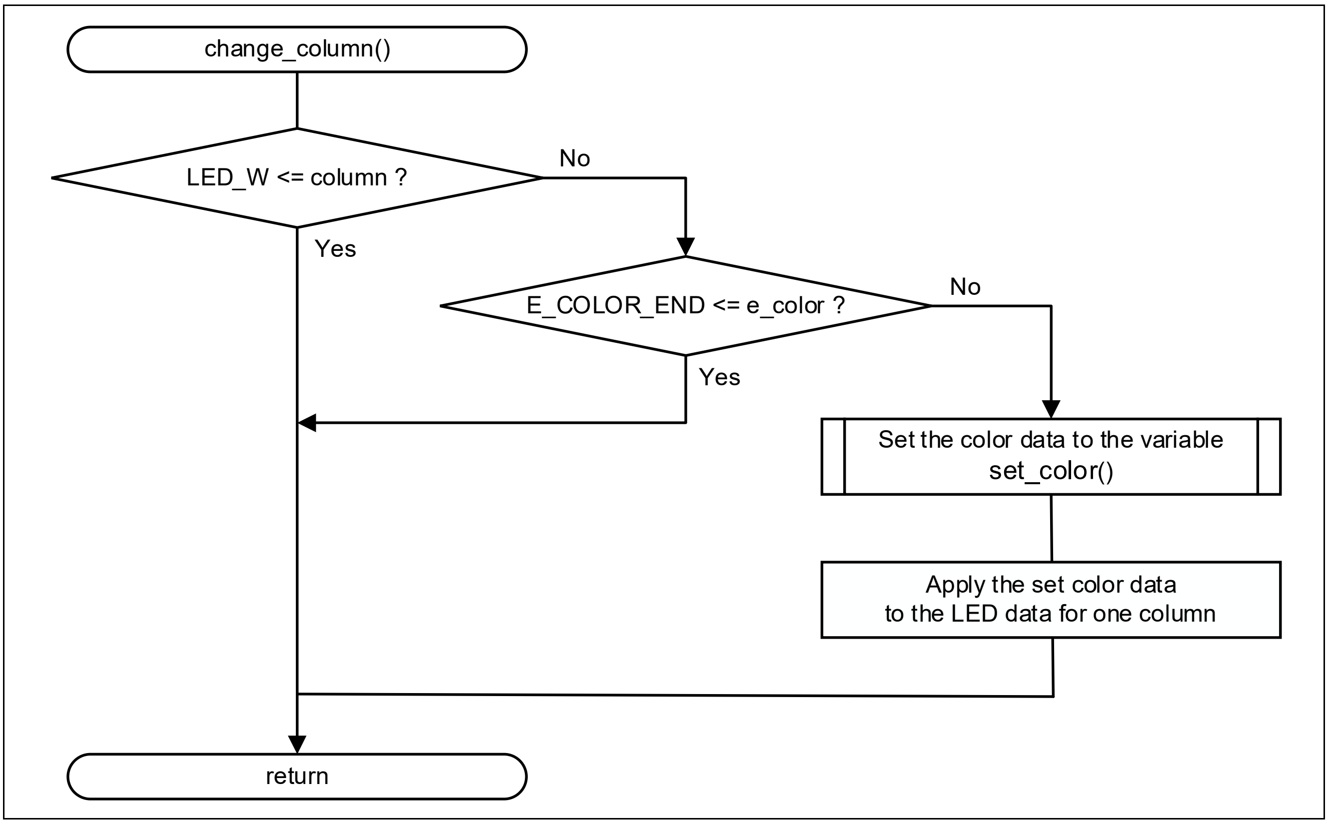
Figure 4-10 Shifting LED Display Data by One Column



4.8.5 Updating a column of LED display data

Figure 4-11 shows the flowchart for updating a column of LED display data.

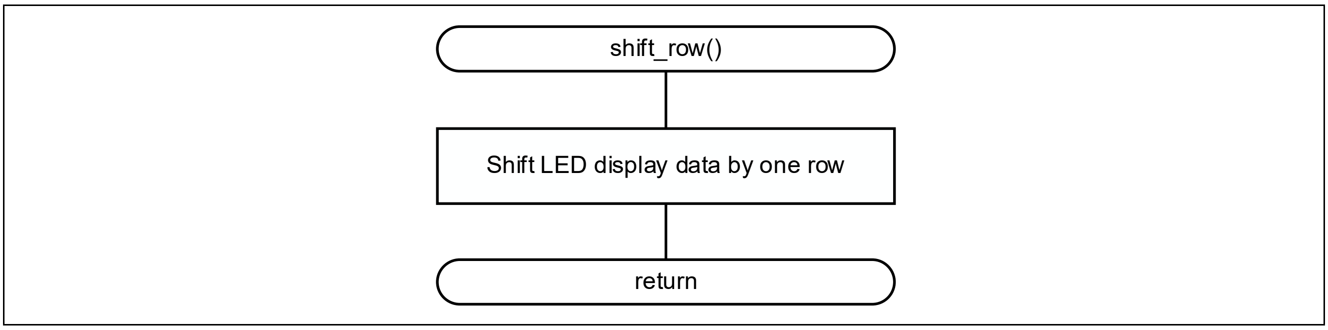
Figure 4-11 Updating a Column of LED Display Data



4.8.6 Shifting LED display data by one row

Figure 4-12 shows the flowchart for shifting LED display data by one row.

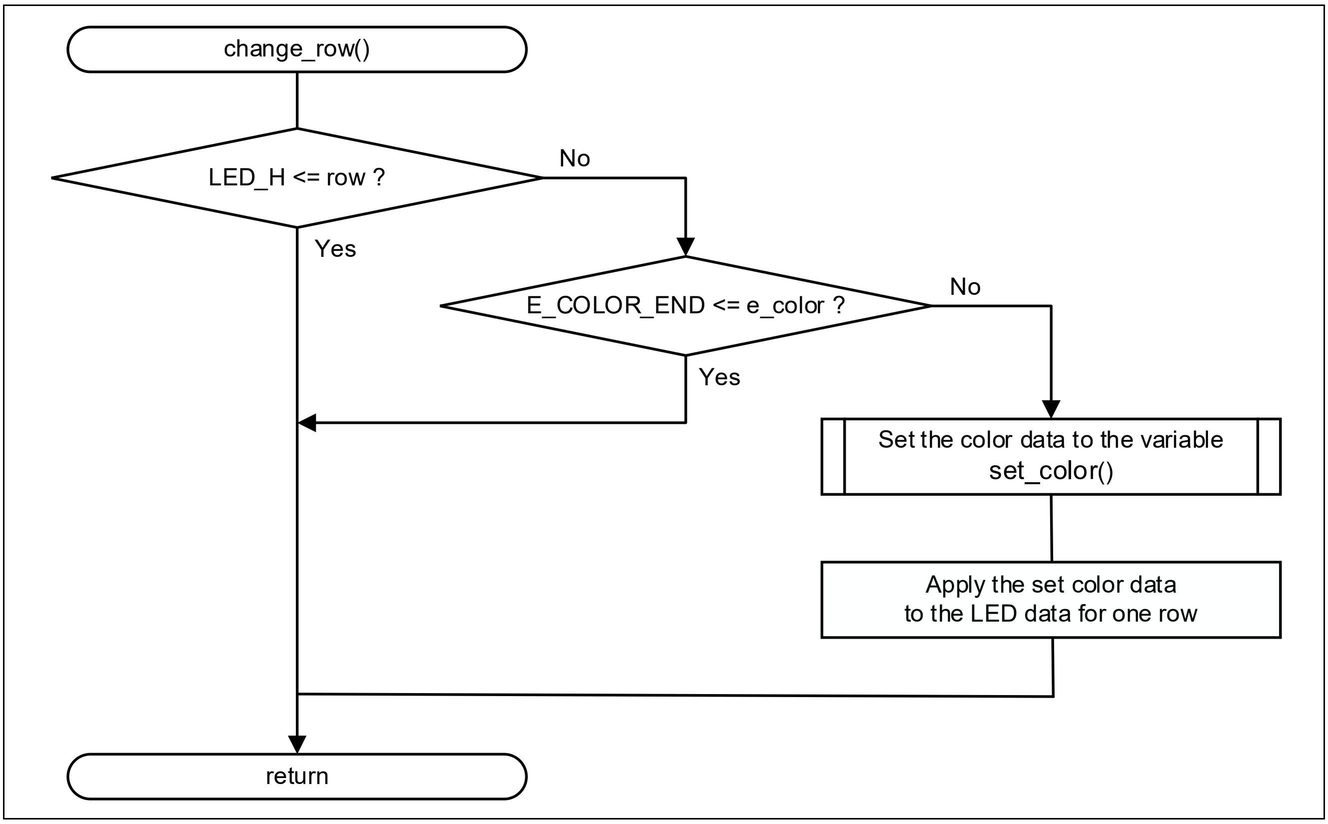
Figure 4-12 Shifting LED Display Data by One Row



4.8.7 Updating a row of LED display data

Figure 4-13 shows the flowchart for updating a row of LED display data.

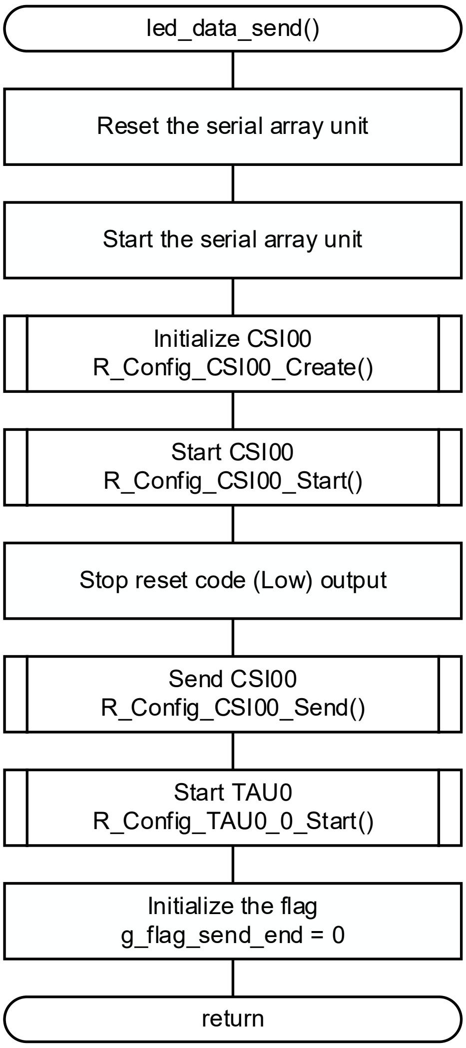
Figure 4-13 Updating a Row of LED Display Data



4.8.8 Sending LED display data

Figure 4-14 shows the flowchart for sending LED display data.

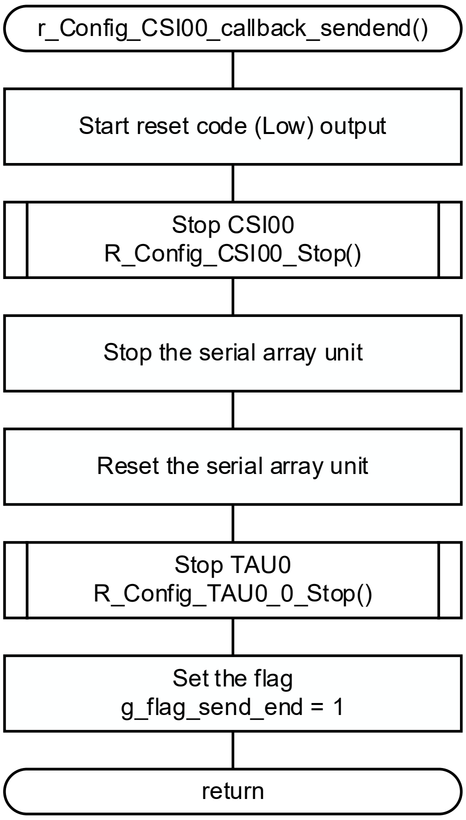
Figure 4-14 Sending LED Display Data



4.8.9 CSI transmission completion interrupt

Figure 4-15 shows the flowchart of a CSI transmission completion interrupt.

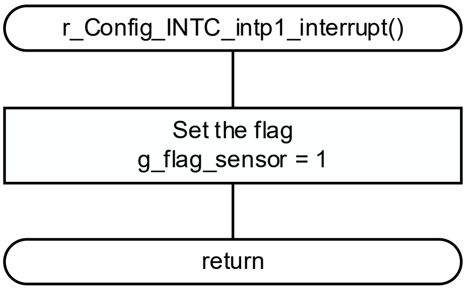
Figure 4-15 CSI Transmission Completion Interrupt



4.8.10 Motion sensor detection interrupt (external pin)

Figure 4-16 shows the flowchart of a motion sensor detection interrupt (external pin).

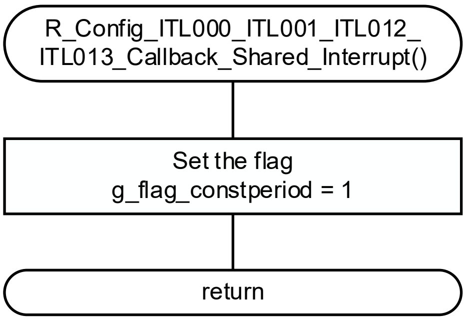
Figure 4-16 Motion Sensor Detection Interrupt (External Pin)



4.8.11 Interval timer interrupt

Figure 4-17 shows the flowchart of an interval timer interrupt.

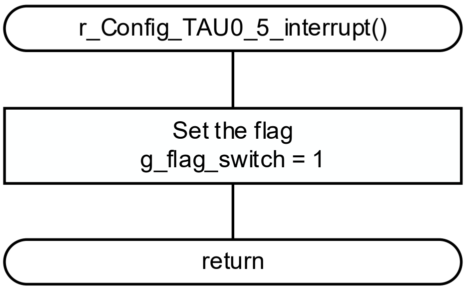
Figure 4-17 Interval Timer Interrupt



4.8.12 Switch input detection interrupt (chattering filtered)

Figure 4-18 shows the flowchart of a switch input detection interrupt (chattering filtered).

Figure 4-18 Switch Input Detection Interrupt (Chattering Filtered)



4.9 LED Control Data Format Image

Figure 4-19 shows the serial data format image used to control WS2812B (LED sheet).

Serial data is stored in the buffer variable for CSI transmission (g_buf_csi[LED_H][LED_W][RGB]).

Figure 4-19 Format Image of LED Control Data Stored in g_buf_csi[LED_H][LED_W][RGB]

Second array number [LED_W]

Third array number [RGB]

First array number [LED_H]

	0			1			2			3			4			5			6			7		
	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
0	0			1			2			3			4			5			6			7		
1	15			14			13			12			11			10			9			8		
2	16			17			18			19			20			21			22			23		
3	31			30			29			28			27			26			25			24		
4	32			33			34			35			36			37			38			39		
5	47			46			45			44			43			42			41			40		
6	48			49			50			51			52			53			54			55		
7	63			62			61			60			59			58			57			56		

LED sheet numbers

LED sheet numbers

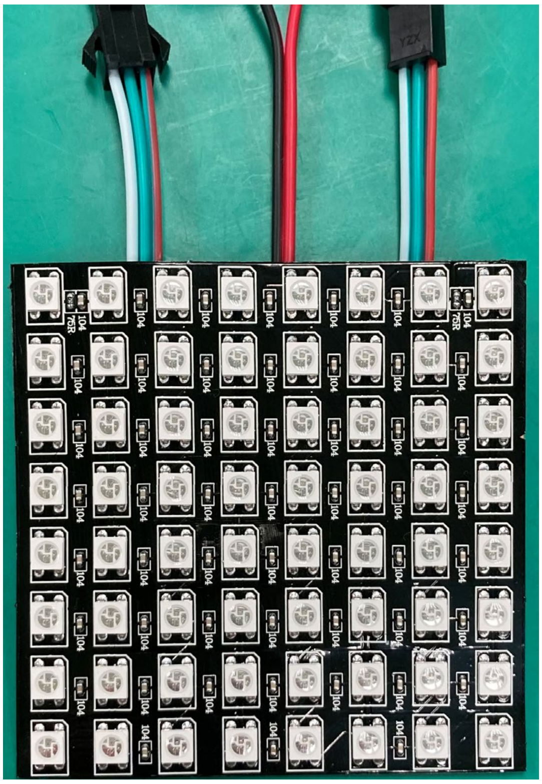
For details, refer to the WS2812B data sheet.

Figure 4-20 shows the LED sheet numbers.

Figure 4-20 LED Sheet Numbers

0

7



63

56

0	15	16	31	32	47	48	63
1	14	17	30	33	46	49	62
2	13	18	29	34	45	50	61
3	12	19	28	35	44	51	60
4	11	20	27	36	43	52	59
5	10	21	26	37	42	53	58
6	9	22	25	38	41	54	57
7	8	23	24	39	40	55	56

5. Setting Up the Smart Configurator

This application note contains the following Smart Configurator configuration file in addition to the sample code.

r01an7320_elcl.scfg

The following describes the file and provides setting examples and precautions for use.

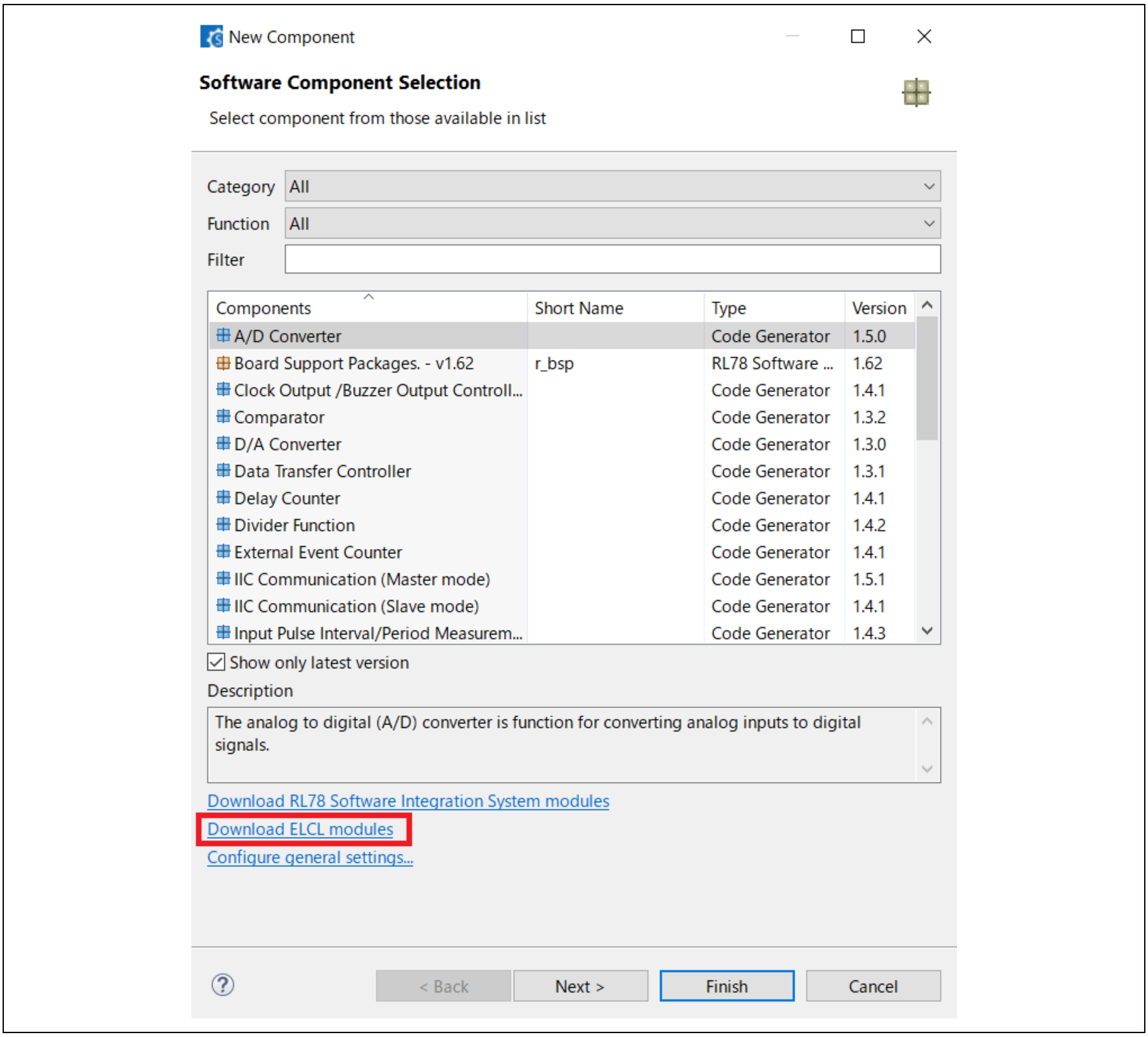
5.1 Setting the ELCL Component

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

The following describes the procedure.

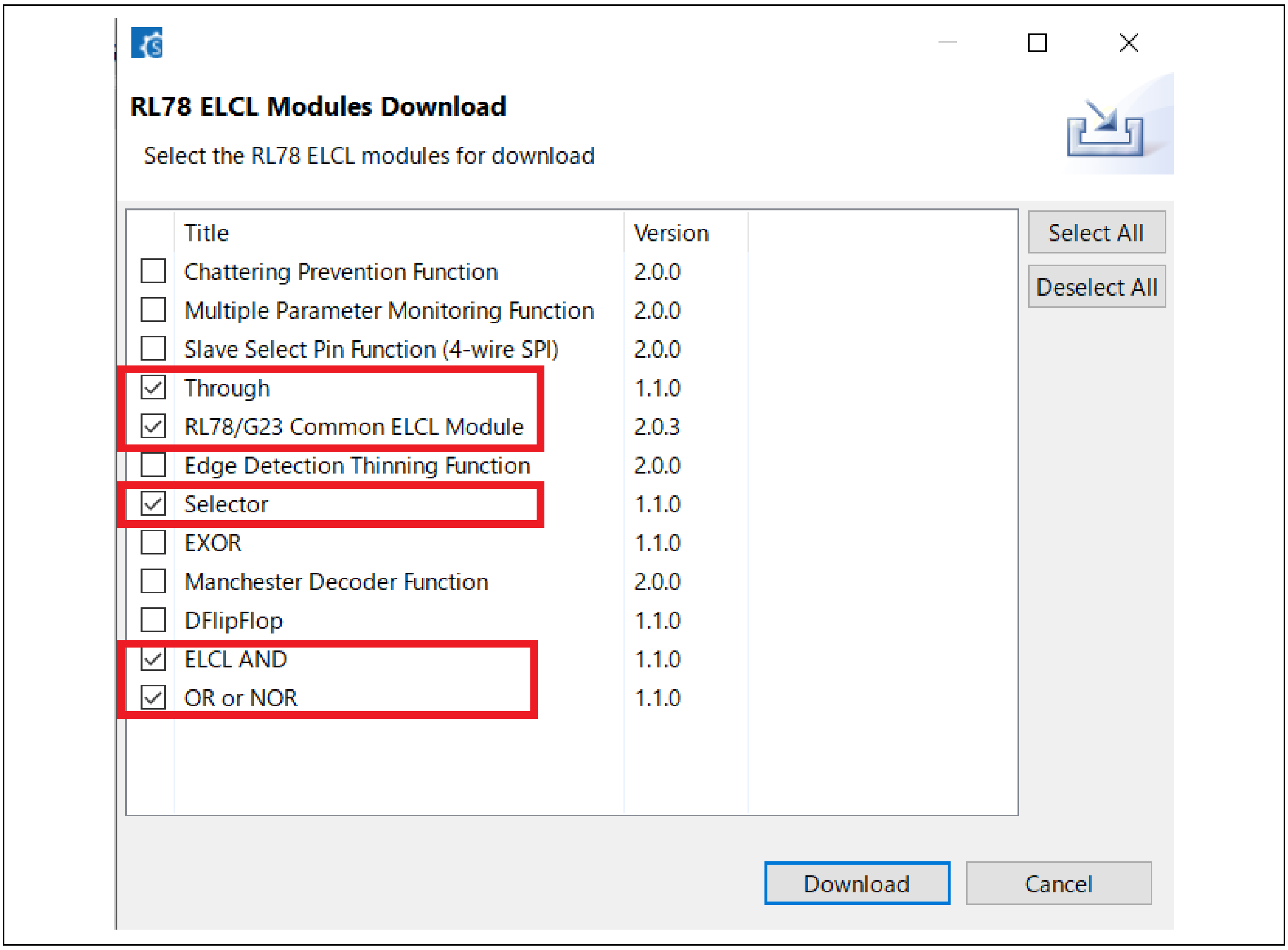
1. Start the Smart Configurator.
2. Click the [Components] tab, and then click [Add component].
3. The New Component window shown in Figure 5-1 opens. Click [Download ELCL module].

Figure 5-1 Selecting the Component



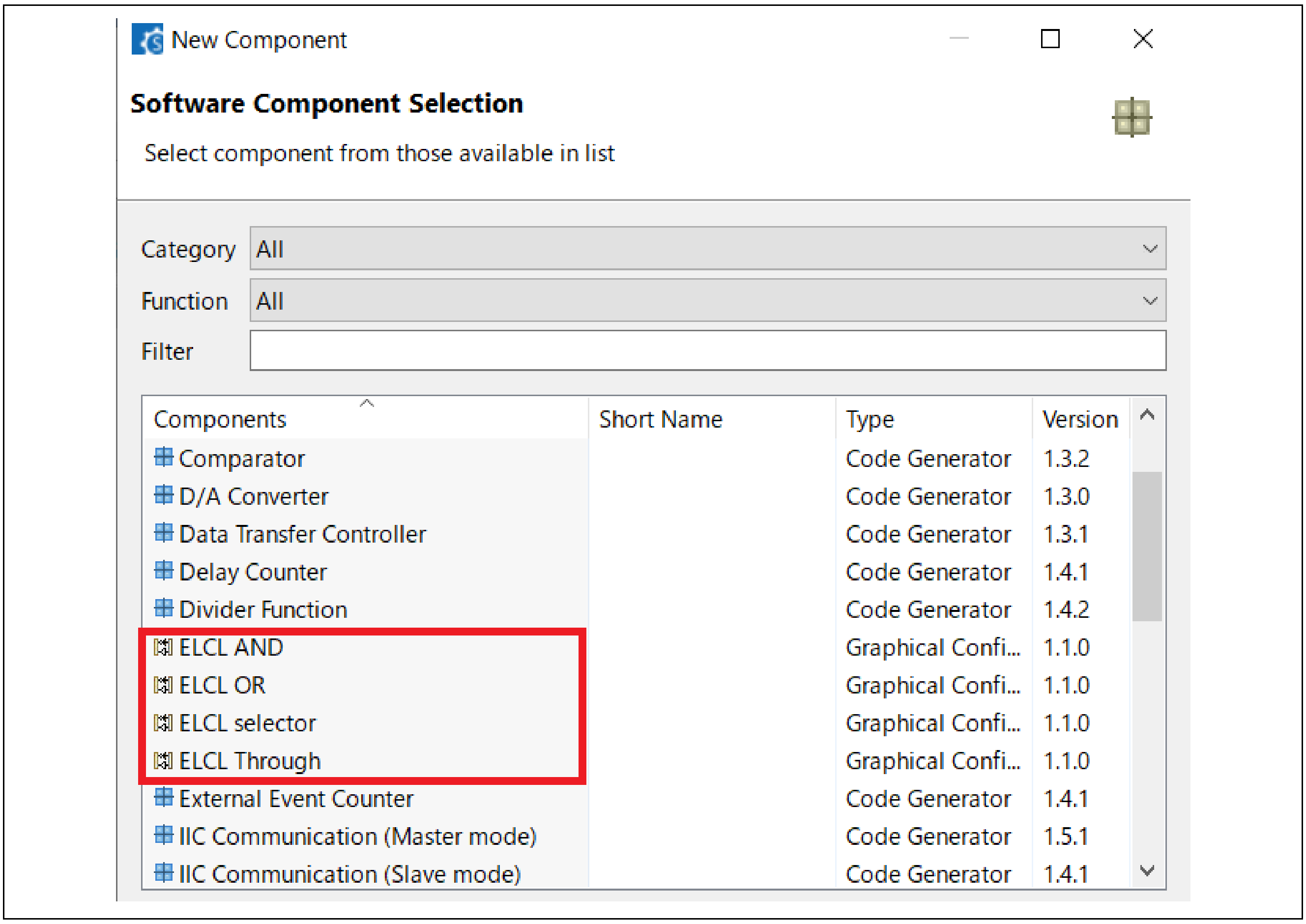
4. Select [Selector], [AND], [Through], and [OR or NOR], and then click [Download]. Also download [RL78/G23 Common ELCL Module], which is the common configuration file.

Figure 5-2 Module Download



5. After download is complete, confirm that you can select [ELCL AND], [ELCL OR], [ELCL selector], and [ELCL Through].

Figure 5-3 Selecting Modules



5.2 r01an7320_elcl.scfg

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

Table 5-1 Smart Configurator Settings

Tag name	Component name	Contents
Clock	-	Operation mode: High-speed main mode 4.0 (V) to 5.5 (V) EV_{DD} setting: 4.0 V ≤ EV_{DD0} < 5.5 V High-speed on-chip oscillator: 32 MHz f_{IHP}: 32 MHz f_{CLK}: 32000 kHz (High-speed on-chip oscillator) f_{SXP}: 32.768 kHz (Low-speed on-chip oscillator) (XT1 oscillation circuit) Operation mode: XT1 oscillation Frequency: 32.768 kHz XT1 oscillation mode: Low consumption oscillation 1 Power supply mode: Power supply enabled in STOP.HALT mode
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: Set Security ID: 0x00000000000000000000 Security ID authentication failure setting: Erase flash memory data
Component	r_bsp	Start up select : Enable (use BSP startup) Control of invalid memory access detection(IAWEN) : Disable Protected area in the RAM (GRAM0-1) : Disabled Protection of the port control registers (GPORT) : Disabled Protection of the interrupt control registers (GINT) : Disabled Protection of the clock, voltage detector, and RAM parity error detection control registers (GCSC) : Disabled Data flash memory area/extra area access control (DFLEN) : Disables Initialization of peripheral functions by Code Generator/Smart Configurator : Enable API functions disable (R_BSP_StartClock, R_BSP_StopClock) : Disable API functions disable (R_BSP_GetFclkFreqHz) : Enable API functions disable (R_BSP_SetClockSource) : Disable API functions disable (R_BSP_ChangeClockSetting) : Disable API functions disable (R_BSP_SoftwareDelay) : Disable Parameter check enable : Enable Enable user warm start callback (PRE) : Unused Enable user warm start callback (POST) : Unused Watchdog Timer refresh enable : Unused

Note: Specify the settings as follows when using IAR.

On-chip debug operation setting: Use emulator

Emulator setting: E2 emulator Lite

Table 5-2 Setting value of Smart Configurator

Tag name	Component name	Contents
Component	Config_INTC	INTP1 setting: INTP1 Valid edge: Rising edge Priority: Level 3 (low priority)
	Config_ITL000_ITL001_ITL012_ITL013	Operation clock (fITL0): f_{IHP} Clock source: f_{ITL0} Interval value: 100 ms Interrupt setting: Used Priority: Level 3 (low priority)
	Config_TAU0_0	Operation clock: CK00 Clock source: f_{CLK} Cycle setting: 1250 μs Interrupt setting: Unused PWM slave selection setting: Channel 1 slave, channel 2 slave, channel 3 slave (Slave 1) Duty: 27% Initial output value: 0 Output level: Active High Interrupt setting: Unused (Slave 2) Duty: 64% Initial output value: 0 Output level: Active High Interrupt setting: Unused (Slave 3) Duty: 100% Initial output value: 1 Output level: Active High Interrupt setting: Unused
	Config_TAU0_5	Operation clock : CK00 Clock source: f_{CLK} Input source setting: ELCL External event edger selection (TI05): Both edges Delay: 2 ms Interrupt setting: Used Priority: Level 3 (low priority)

Table 5-3 Setting value of Smart Configurator

Tag name	Component name	Contents
Component	Config_AND	Component：ELCL AND Common setting：L3L0 （Detail setting：L3L0） Input signal selector：ELISEL_8 , TO02 ELISEL_9 , TO03 Application：AND Output signal selector：P13
	Config_OR	Component：ELCL OR Common setting：L2L0 （Detail setting：L2L0） Input signal selector：ELISEL_1 , TO01 ELISEL_4 , TO00 Application：OR Output signal selector：P51
	Config_Selector	Component：ELCL Selector Common setting：L1 （Detail setting：L1） Input signal selector：ELISEL_1 , TO01 ELISEL_2 , TO02 ELISEL_3 , SO00 Application：Selector Output signal selector：P10
	Config_Through	Component：ELCL Through Common setting：L1L0,L2L1 （Detail setting：L1L0） Input signal selector：ELISEL_0 , TO01 Application：Through Output signal selector：SAU0 channel0 input clock （Detail setting：L2L1） Input signal selector：ELISEL_5 , P137 Application：Through Output signal selector：TAU0 channel 5 input
	Config_CSI00	Transfer clock mode: External clock (slave) Transfer mode setting: Continuous transfer mode Data bit length setting: 8 bits Data transfer direction setting: LSB Data transmission/reception timing: Type 2 Transfer rate setting: Unused Interrupt setting: Level 3 (low priority) Callback function setting: Transmission complete

5.2.1 Clocks

Set the clocks used in the sample code.

5.2.2 System

Specify the on-chip debug setting of the sample code.

The settings of “On-chip debug operation setting” and “Security ID authentication failure setting” affect “On-chip debugging enabled” in Table 4-3 Option Byte Setting. If you change the settings, confirm that no problems will occur.

5.2.3 r_bsp

Set the startup of the sample code.

5.2.4 Config_TAU0_0

Set TAU00 of the sample code.

In the sample code, two PWM output signals with different duty cycles are set for 0-code and 1-code data transmission, and PWM output with 100% or 0% duty cycle is set for reset code transmission. No interrupts are used.

5.2.5 Config_TAU0_5

Set TAU05 of the sample code.

In the sample code, the delay count function is used to eliminate chattering during switch input. Set the flag as a processing trigger when valid switch input is performed during interrupt vector processing.

5.2.5.1 Config_TAU0_5.c changes

In this application note, the code output by the Smart Configurator is modified as shown in the area highlighted (yellow). When the program is automatically generated by the Smart Configurator again, the changes will be overwritten with the output values of the Smart Configurator (before the change). Therefore, change the values each time the program is automatically generated.

Function name	R_Config_TAU0_5_Create
Before change	TMR05 = _0000_TAU_CLOCK_SELECT_CKM0 | _0000_TAU_CLOCK_MODE_CKS | _0100_TAU_TRIGGER_TIMN_VALID | _0080_TAU_TIMN_EDGE_BOTH_LOW | _0008_TAU_MODE_ONE_COUNT | _0001_TAU_START_INT_USED;
After change	TMR05 = _0000_TAU_CLOCK_SELECT_CKM0 | _0000_TAU_CLOCK_MODE_CKS | _0200_TAU_TRIGGER_TIMN_BOTH | _0080_TAU_TIMN_EDGE_BOTH_LOW | _0008_TAU_MODE_ONE_COUNT | _0001_TAU_START_INT_USED;

5.2.6 Config_INTC

Set INTP1 of the sample code.

In the sample code, a motion sensor signal is detected at the rising edge. Set the flag as a processing trigger when the motion sensor detects a motion during interrupt vector processing.

5.2.7 Config_ITL000_ITL001_ITL012_ITL013

Set TML32 of the sample code.

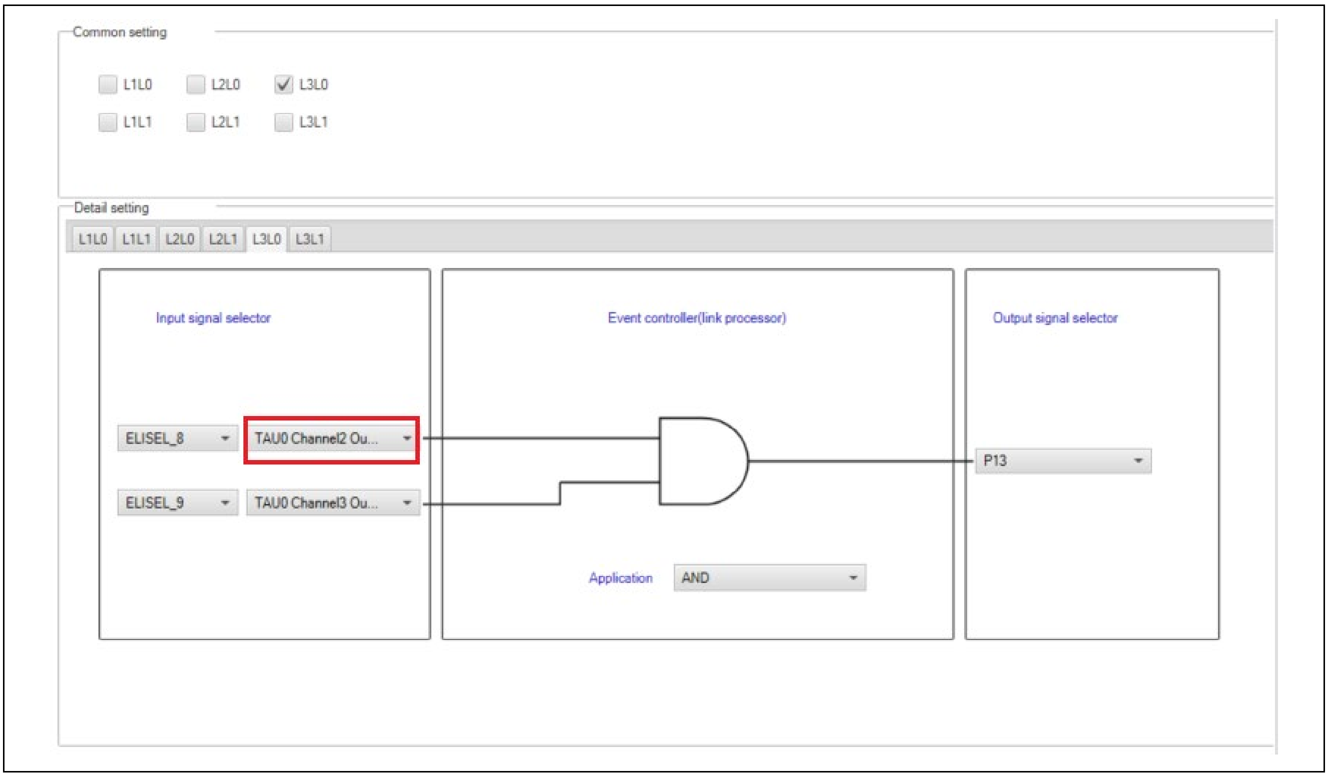
In the sample code, a 100-ms interval timer is configured for generating LED display update timing. Set the flag as a trigger of a process at regular intervals during interrupt vector processing.

5.2.8 Config_AND

Configure the AND setting of the sample code.

In this sample code, L3L0(AND) is used and LED control signal (L2L0 OR output) and PWM with 100% or 0% duty cycle are selected as input signals. Because the Smart Configurator cannot select L2L0 OR output as an input signal, either input is referred to as “TAU0 channel 2 output” for descriptive purposes.

Figure 5-4 ELCL “AND” Setting



5.2.8.1 Config_AND.c changes

In this application note, the code output by the Smart Configurator is modified as shown in the area highlighted (yellow). When the program is automatically generated by the Smart Configurator again, the change will be overwritten with the output values of the Smart Configurator (before the change). Therefore, change the value each time the program is automatically generated.

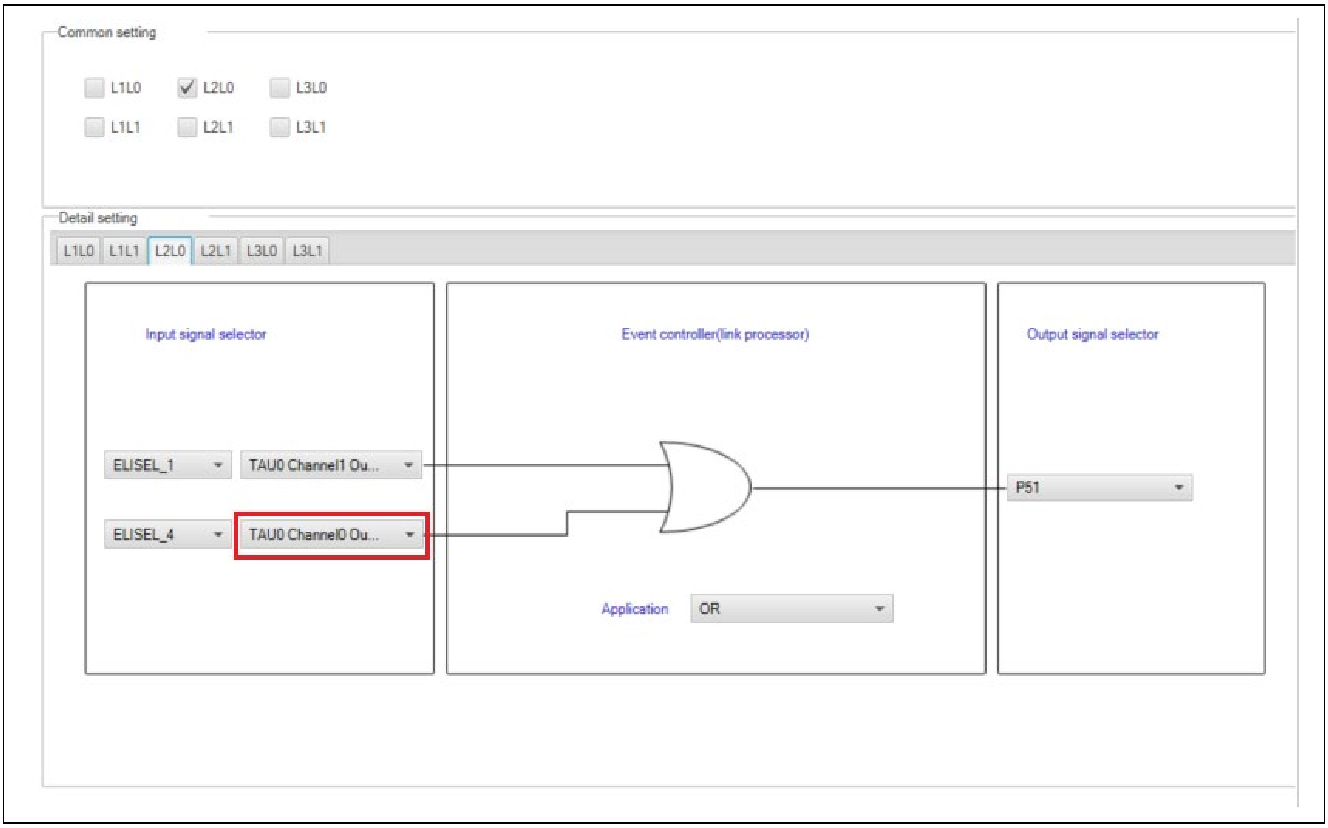
Function name	R_Config_AND_Create
Before change	PFOE0&=0xFBU; ELISEL8=0x09U; PFOE0&=0xF7U; ELISEL9=0x0AU; ELL3SEL0=0x09U; ELL3SEL1=0x0AU; ELL3LNK0=0x01U; ELL3LNK1=0x02U; ELL3CTL|=0x01U; PMCA1&=0xF7U; PM1&=0xF7U; PMCE1|=0x08U; ELOSEL3=0x0BU;
After cahnge	PFOE0&=0xFBU; ELISEL8=0x09U; PFOE0&=0xF7U; ELISEL9=0x0AU; ELL3SEL0=0x0DU; ELL3SEL1=0x0AU; ELL3LNK0=0x01U; ELL3LNK1=0x02U; ELL3CTL|=0x01U; PMCA1&=0xF7U; PM1&=0xF7U; PMCE1|=0x08U; ELOSEL3=0x0BU;

5.2.9 Config_OR

Configure the OR setting of the sample code.

In this sample code, L2L0(OR) is used, and LED control signal (L1 selector output) and PWM for 0 code are selected as input signals. Because the Smart Configurator cannot select L1 selector output as an input signal, either input is referred to as “TAU0 channel 0 output” for descriptive purposes.

Figure 5-5 ELCL「OR」setting



5.2.9.1 Config_OR.c changes

In this application note, the code output by the Smart Configurator is modified as shown in the areas highlighted (yellow). When the program is automatically generated by the Smart Configurator again, the change will be overwritten with the output value of the Smart Configurator (before the change). Therefore, change the value each time the program is automatically generated.

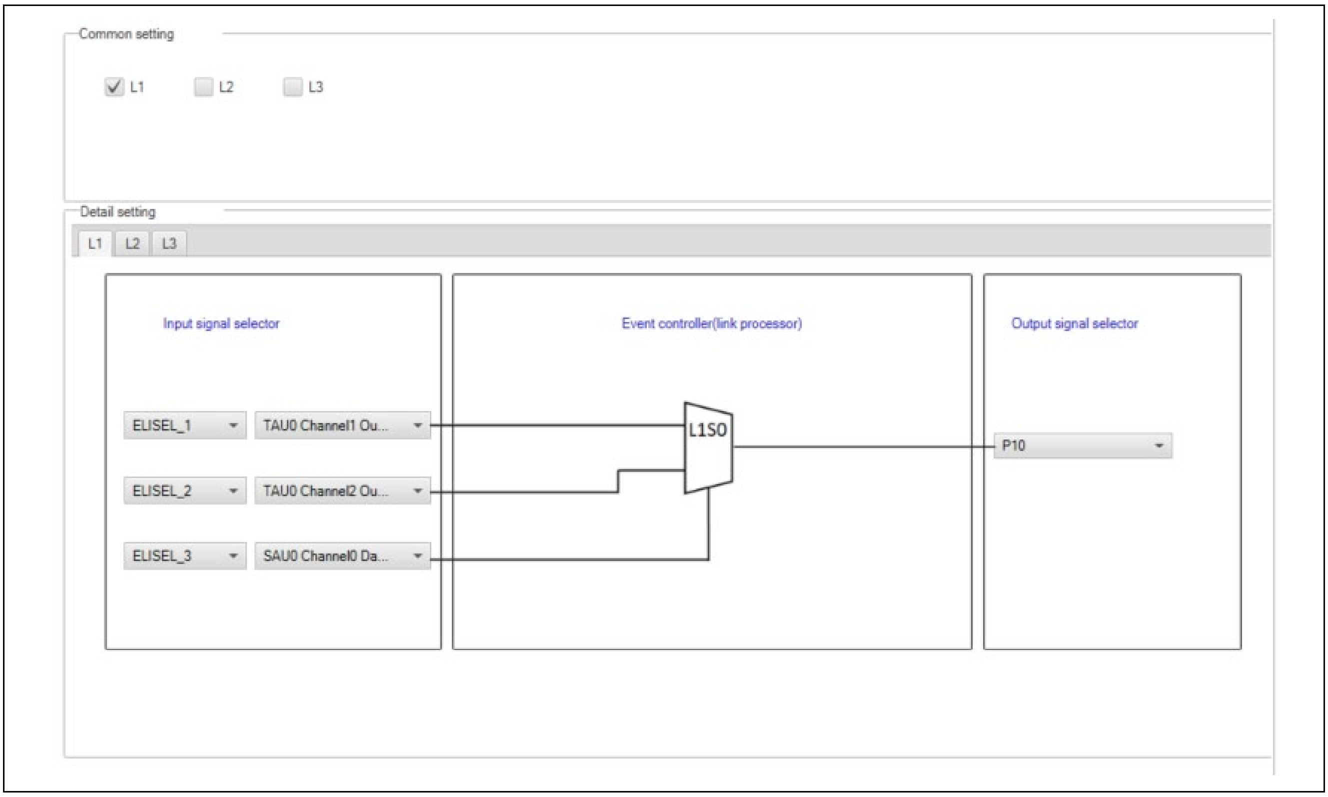
Function name	R_Config_OR_Create
Before change	PFOE0&=0xFDU; ELISEL1=0x08U; PFOE0&=0xFEU; ELISEL4=0x07U; ELL2SEL0=0x02U; ELL2SEL1=0x05U; ELL2LNK0=0x01U; ELL2LNK1=0x02U; ELL2CTL|=0x02U; PM5&=0xFDU; PMCE5|=0x02U; ELOSEL1=0x06U;
After cahnge	PFOE0&=0xFDU; ELISEL1=0x08U; PFOE0&=0xFEU; ELISEL4=0x07U; ELL2SEL0=0x02U; ELL2SEL2=0x0FU; ELL2LNK0=0x01U; ELL2LNK2=0x02U; ELL2CTL|=0x02U; PM5&=0xFDU; PMCE5|=0x02U; ELOSEL1=0x06U;

5.2.10 Config_Selector

Configure the Selector setting of the sample code.

In this sample code, L1 Selector is used and PWM for 0 code and PWM for 1 code are selected as input signals. Data output of CSI00, which outputs LED control data, is set for selection control of the Selector.

Figure 5-6 ELCL “Selector” Setting



5.2.10.1 Config_Selector.c changes

In this sample code, logical cell block L1 is also used in Through. This causes a conflict with the Through setting because the Smart Configurator cannot select an event link L1 signal select block n. Therefore, the code output by the Smart Configurator is modified as shown in the areas highlighted (yellow). When the program is automatically generated by the Smart Configurator again, the change will be overwritten with the output value of the Smart Configurator (before the change). Therefore, change the value each time the program is automatically generated.

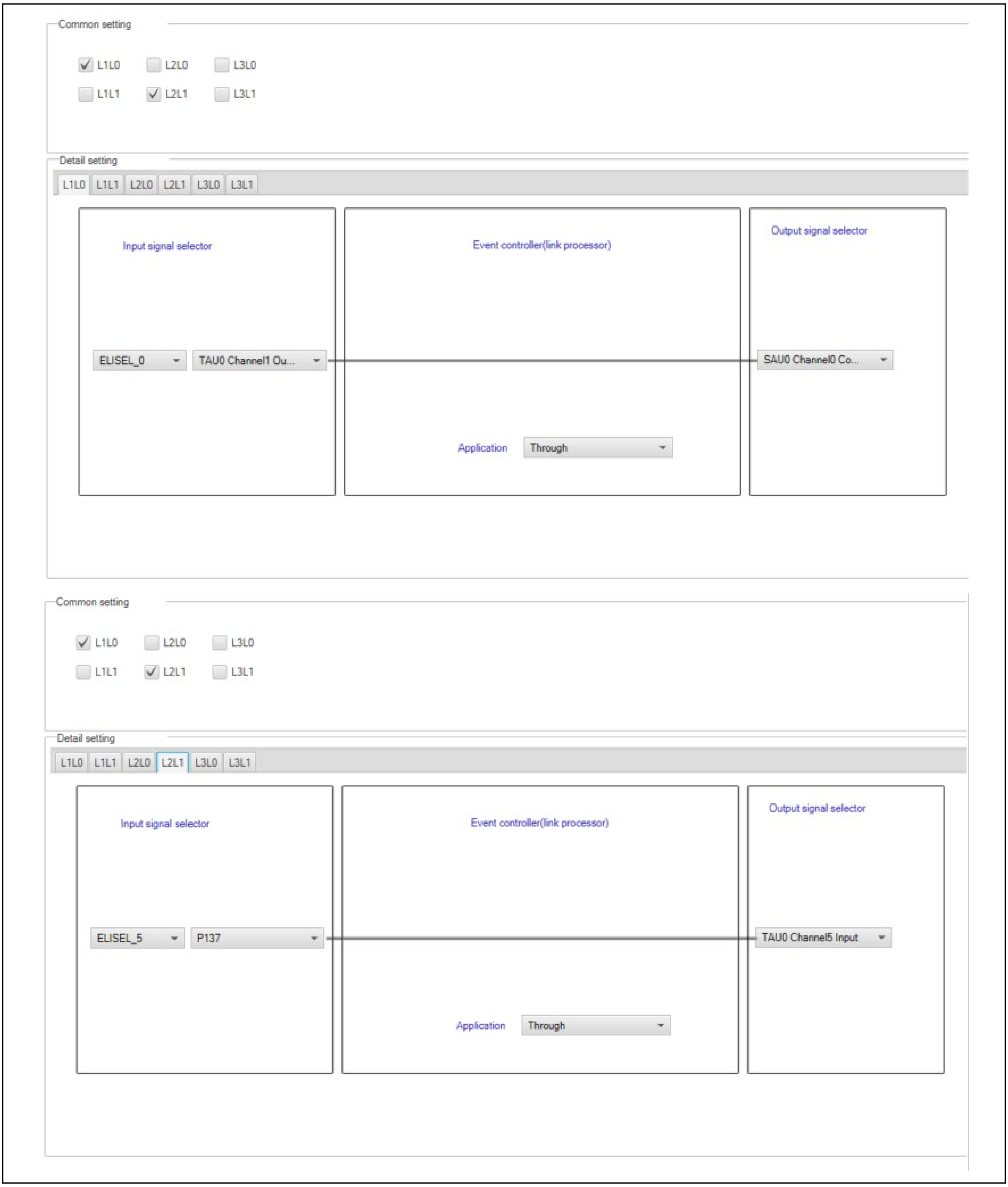
Function name	R_Config_Selector_Create
Before change	PFOE0&=0xFDU; ELISEL1=0x08U; PFOE0&=0xFBU; ELISEL2=0x09U; PFOE1&=0xFEU; ELISEL3=0x00U; ELL1SEL0=0x02U; ELL1SEL1=0x03U; ELL1SEL2=0x04U; ELL1LNK0=0x06U; ELL1LNK1=0x07U; ELL1LNK2=0x05U; PM1&=0xFEU; PMCE1|=0x01U; ELOSEL0=0x03U;
After cahnge	PFOE0&=0xFDU; ELISEL1=0x08U; PFOE0&=0xFBU; ELISEL2=0x09U; PFOE1&=0xFEU; ELISEL3=0x00U; ELL1SEL1=0x02U; ELL1SEL2=0x03U; ELL1SEL3=0x04U; ELL1LNK1=0x06U; ELL1LNK2=0x07U; ELL1LNK3=0x05U; PM1&=0xFEU; PMCE1|=0x01U; ELOSEL0=0x03U;

5.2.11 Config_Through

Configure the Through setting of the sample code sample code.

In this sample code, L1L0(Through) is used and PWM for 0 code and PWM for 1 code are selected as input signals and linked to SAU0 channel 0 communication clock. In addition, L2L1(Through) is used and P137 selected as an input signal is linked to TAU0 channel 5 input.

Figure 5-7 ELCL "Through" setting



5.2.12 Config_CSI00

Configure the CSI00 setting of the sample code.

In this sample code, the transfer clock mode is set to external clock (slave) mode and the data transmission/reception timing is set to 2. This configures the selector selection control to switch 0 or 1 code of LED control signal at the falling edge of the PWM for 0 code.

6. Sample code

Sample code can be downloaded from the Renesas Electronics website.

7. Documents for Reference

RL78/G23 User’s Manual: Hardware (R01UH0896)

RL78 family user's manual software (R01US0015)

RL78 Smart Configurator User's Guide: CS+ (R20AN0580)

RL78 Smart Configurator User's Guide: e^2 studio (R20AN0579)

RL78 Smart Configurator User's Guide: IAR (R20AN0581)

The latest versions can be downloaded from the Renesas Electronics website.

Technical update
The latest versions can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul.16.24	-	First Edition

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/.