RL78/L23

LCD display (Clock demo)

Abstract

This document describes how to control the LCD panel using the RL78/L23 LCD controller/driver and how the sample code works.

The sample code uses the RL78/L23 LCD controller/driver to display a clock in 24-hour mode. The sample code stores the time measured by the real-time clock (RTC) in the LCD display data memory area and changes the time each time an RTC constant-period interrupt occurs (once a second).

Additionally, by pressing the user switch, the time can be adjusted. By touch detection of the touch buttons, the adjusted time is displayed on the LCD.

Products

RL78/L23

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Specifications

This application describes using the RL78/L23 LCD controller/driver to display a clock in 24-hour mode on an LCD.
The sample code stores the time measured by the RTC in the LCD display data memory area, and changes the time each time an RTC constant-period interrupt occurs (once a second).

When the user switch (SW) is pressed, the hour indicator, minute indicator, and second indicator can be adjusted (hour setting mode, minute setting mode, or second setting mode), the LCD controller/driver adjusts the time, and the adjusted time is displayed on the LCD by touch detection of the touch buttons.

When adjusting the time using hour setting mode and minute setting mode and second setting mode, the corresponding digits on the LCD are blinking.

Table 1-1 lists the peripheral functions and their applications. Figure 1.1 shows an operation overview.

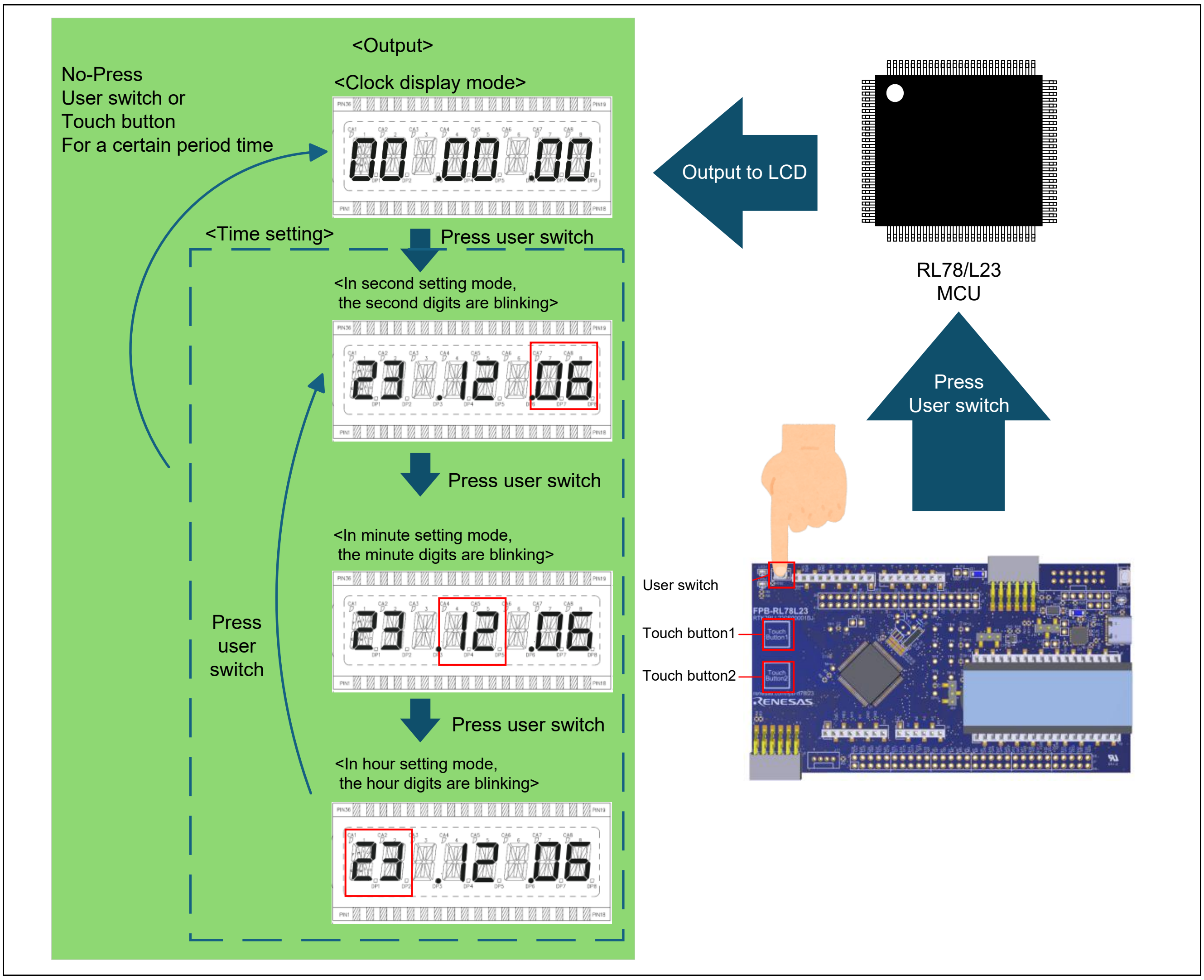
Table 1-1 Peripheral Functions and Their Applications

Peripheral Function	Application
LCD controller/driver	Controls the LCD panel.
RTC	Counts the time.
Channel 1 of timer array unit 0 (TAU0_1)	Blinks the selected part of the LCD.
Channel 3 of timer array unit 0 (TAU0_3)	Counts wait time.
Channel 4 of timer array unit 0 (TAU0_4)	Counts time for preventing chattering.
Channel 5 of timer array unit 0 (TAU0_5)	Determines continuous touch (long press) on touch button.
Capacitive Touch Sensing Unit (CTSU2La)	Measures capacitance generated on touch electrode.
External interrupt INTP0	Detects input from the SET switch and enters hour setting mode, minute setting mode and second setting mode

The RL78/L23 LCD controller/driver can use external resistance division, internal voltage boosting, or capacitor split to generate LCD drive voltage. For details, refer to 3.3 LCD Drive Voltage Generator.

The sample code uses internal voltage boosting for the LCD drive voltage generator.

Figure 1.1 Operation Overview



Number of touch button presses	State
0	Clock display mode
1	Second setting mode
2	Minutes setting mode
3	Hour setting mode

Touch detection of touch button	State
Touch button 1	Adds 1 to the blinking digit
Touch button 2	Subtracts 1 from the blinking digit

Note: From the fourth press on the user switch, the state transition repeats from the state of first press.

2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2-1 Operation Confirmation Conditions

Item	Contents
MCU used	RL78/L23(R7F100LPL3CFB)
Operating frequencies	・High-speed on-chip oscillator clock (f_{HOCO}) : 32MHz(typ.) ・CPU/peripheral hardware clock (f_{CLK}) : 32MHz ・Subsystem clock XR (f_{SXR}) : 32.768kHz
Operating voltage	3.3V LVD operation(V_{LVD0}) : in reset mode is 2.97 V at the rising edge or 2.91 V at the falling edge.
Integrated development environment (CS+)	CS+ for CC V8.14.00 from Renesas Electronics Corp.
C compiler (CS+)	Renesas CC-RL V1.15.01 from Renesas Electronics Corp.
Integrated development environment (e^2 studio)	e^2 studio Version 2025-07 from Renesas Electronics Corp.
C compiler (e^2 studio)	Renesas CC-RL V1.15.01 from Renesas Electronics Corp.
Integrated development environment (IAR)	IAR Systems IAR Embedded Workbench for Renesas RL78 V 5.20.1
C compiler (IAR)	
Smart Configurator	RL78 Smart Configurator V1.14.0 from Renesas Electronics Corp.
Development support tool for the capacitive touch sensor	QE for Capacitive Touch V4.2.0
Board support package (BSP)	V1.91 Smart Configurator.
TOUCH Driver	V2.20 from Renesas Electronics Corp.
CTSU Driver	V2.20 from Renesas Electronics Corp.
Board used	RL78/L23 Fast Prototyping Board (RTK7RLL230S00001BJ)
LCD module	Varitronix VIM-878-DP-FC-S-LV 16 segments x 8 digits Header: 36-pin (18 x 2 rows) x 1 Operating voltage condition: 3V to 4.6V (When using the module with operating voltage other than 3.3V, remove the LCD panel.)

3. Peripheral Function

This chapter describes the LCD controller/driver.

For more information on the operation, refer to section 22 LCD Controller/Driver in RL78/L23 User's Manual: Hardware (R01UH1082).

3.1 Basic Features of RL78/L23 LCD Controller/Driver

RL78/L23 LCD controller/driver includes the following features:

- Waveform A or B selectable
- LCD driver voltage generator can be switched between internal voltage boosting, capacitor split, or external resistance division
- Segment and common signals can be output automatically by reading the LCD display data register automatically Reference voltage generated when the voltage boost circuit is operating can be selected from 23 levels (contrast adjustment)
- LCD blinking is available

3.2 LCD Controller/Driver Display Mode

LCD controller/driver display modes are combinations of the LCD drive waveform and LCD voltage generator. Table 3-1,Table 3-2 and Table 3-3 list the maximum number of pixels in each display mode.

Table 3-1 Maximum Number of Pixels for an 100-pin Package(waveform A)(1/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
waveform A	External resistance division		-	Static	56 (56 segment signals × 1 common signals)
			1/2	2	112 (56 segment signals × 2 common signals)
				3	168 (56 segment signals × 3 common signals)
			1/3	3	
				4	224 (56 segment signals × 4 common signals)
				6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)
			1/4	8	
	Internal voltage Boosting	V_{L1} Reference	1/3	3	168 (56 segment signals × 3 common signals)
				4	224 (56 segment signals × 4 common signals)
				6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)
			1/4	6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)
		V_{L2} Reference	1/3	3	168 (56 segment signals × 3 common signals)
				4	224 (56 segment signals × 4 common signals)
				6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)
				6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)

Table 3-2 Maximum Number of Pixels for an 100-pin Package(waveform A)(2/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
waveform A	Capacitor split	V_{DD} Reference	1/3	3	168 (56 segment signals × 3 common signals)
				4	224 (56 segment signals × 4 common signals)
				6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)
			1/4	6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)
		V_{L4} Reference	1/3	3	168 (56 segment signals × 3 common signals)
				4	224 (56 segment signals × 4 common signals)
				6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)

Table 3-3 Maximum Number of Pixels for an 100-pin Package(waveform B)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
waveform B	External resistance division		1/3	3	168 (56 segment signals × 3 common signals)
				4	224 (56 segment signals × 4 common signals)
				6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)
			1/4	8	
	Internal voltage boosting	V_{L1} Reference	1/3	3	168 (56 segment signals × 3 common signals)
				4	224 (56 segment signals × 4 common signals)
				6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)
			1/4	8	
		V_{L2} Reference	1/3	3	168 (56 segment signals × 3 common signals)
				4	224 (56 segment signals × 4 common signals)
				6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)
	Capacitor split	V_{DD} Reference	1/3	3	168 (56 segment signals × 3 common signals)
				4	224 (56 segment signals × 4 common signals)
				6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)
			1/4	8	
		V_{L4} Reference	1/3	3	168 (56 segment signals × 3 common signals)
				4	224 (56 segment signals × 4 common signals)
				6	324 (54 segment signals × 6 common signals)
				8	416 (52 segment signals × 8 common signals)

3.3 LCD Drive Voltage Generator

The RL78/L23 LCD controller/driver can use external resistance division, internal voltage boosting, or capacitor split to generate LCD drive voltage. This chapter covers the features of each method.

Table 3-4 LCD Drive Method and Its Application

LCD Drive Method	Feature/Usage			Application
	Drive capacity	Operating current	Drive voltage	
External resistance division	High Supports large LCDs	Standard	V_{DD} - dependent	**Suitable for large format LCDs or AC power supply sets** The LCD drive capacity is high and the drive voltage is generated by a resistor divider, which contributes to cost reduction. This method generates the LVD drive voltage by an external resistor divider. As the voltage is applied externally, the operating current and drive capacity can be adjusted by an external resistor.
Internal voltage boosting	Standard	Small	Constant As the drive voltage is constant, the LCD display does not change with the battery voltage decreased	**Suitable for battery sets** The operating current is small and the LCD display does not become dim as the drive voltage is constant even when the battery voltage is reduced. This method generates the reference voltage internally and boosts it by an external capacitor. As the reference voltage is adjusted by software, the LCD contrast can be adjusted from 23 levels in RL78/L23.
Capacitor split	Standard	Much Smaller	Depends on V_{DD} when the reference voltage is V_{DD} LCD display becomes dim with the supply voltage decreased	**Suitable for battery sets** This method has the smallest operating current among three LCD drive modes, and thus the LCD display becomes dim with decreasing the supply voltage. Use this method to allow the screen to be dim according to the battery level. If you do not want the screen to be dim when the battery voltage is decreased, change the LCD drive method to internal voltage boosting. It works in an external circuit of the capacitor split method.

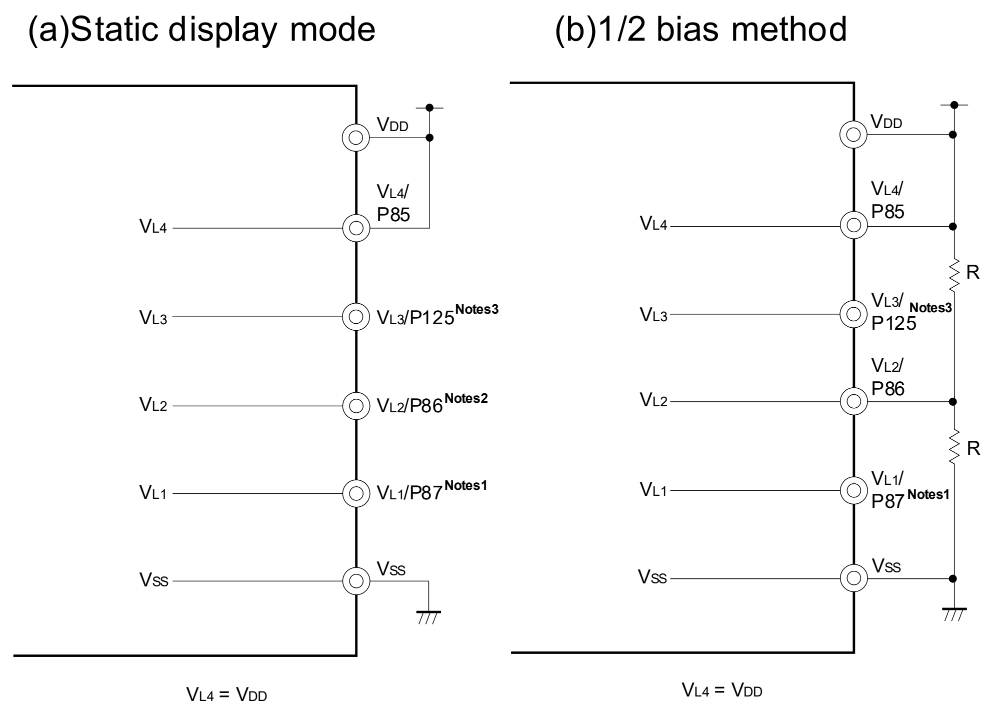
3.3.1 External Resistance Division Method

This is suitable for large format LCDs or AC power sets. As it has a large drive capacity and generates the drive voltage by a resistor divider, which contributes to cost reduction.

To be more specific, this method generates an LCD drive voltage using an external resistor divider. As the voltage is applied externally, the operating current or the drive capacity can be adjusted by the external resistor.

Figure 3.1 and Figure 3.2 show connection examples of external resistance division methods.

Figure 3.1 Connection Example of External Resistance Division Method (1/2)



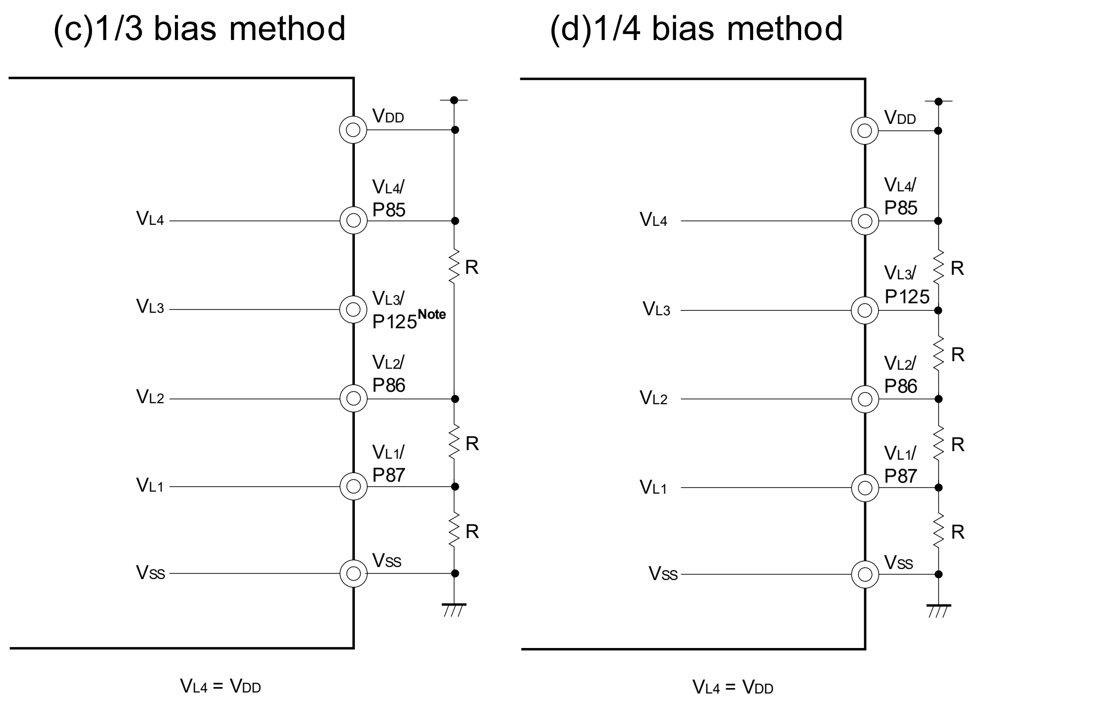
Notes for Figure 3.1 (a) and (b).

Notes 1. V_{L1} can be used as a port(P87).

Notes 2. V_{L2} can be used as a port(P86).

Notes 3. V_{L3} can be used as a port(P125).

Figure 3.2 Connection Example of External Resistance Division Method (2/2)



Note. V_{L3} can be used as a port(P125).

Caution 1. CAPL can be used as port-pin P126 and CAPH can be used as port-pin P127.

Caution 2. The reference resistance “R” value for external resistance division is 10 kΩ to 1 MΩ.
In addition, to stabilize the potential of the V_{L1} to V_{L4} pins, connect a capacitor between each of pins V_{L1} to V_{L4} and the GND pin as needed. The reference capacitance is about 0.47 μF but it depends on the LCD panel used, the number of segment pins, the number of common pins, the frame frequency, and the operating environment. Thoroughly evaluate these values in accordance with your system and adjust and determine the capacitance.

3.3.2 Internal Voltage Boosting Method

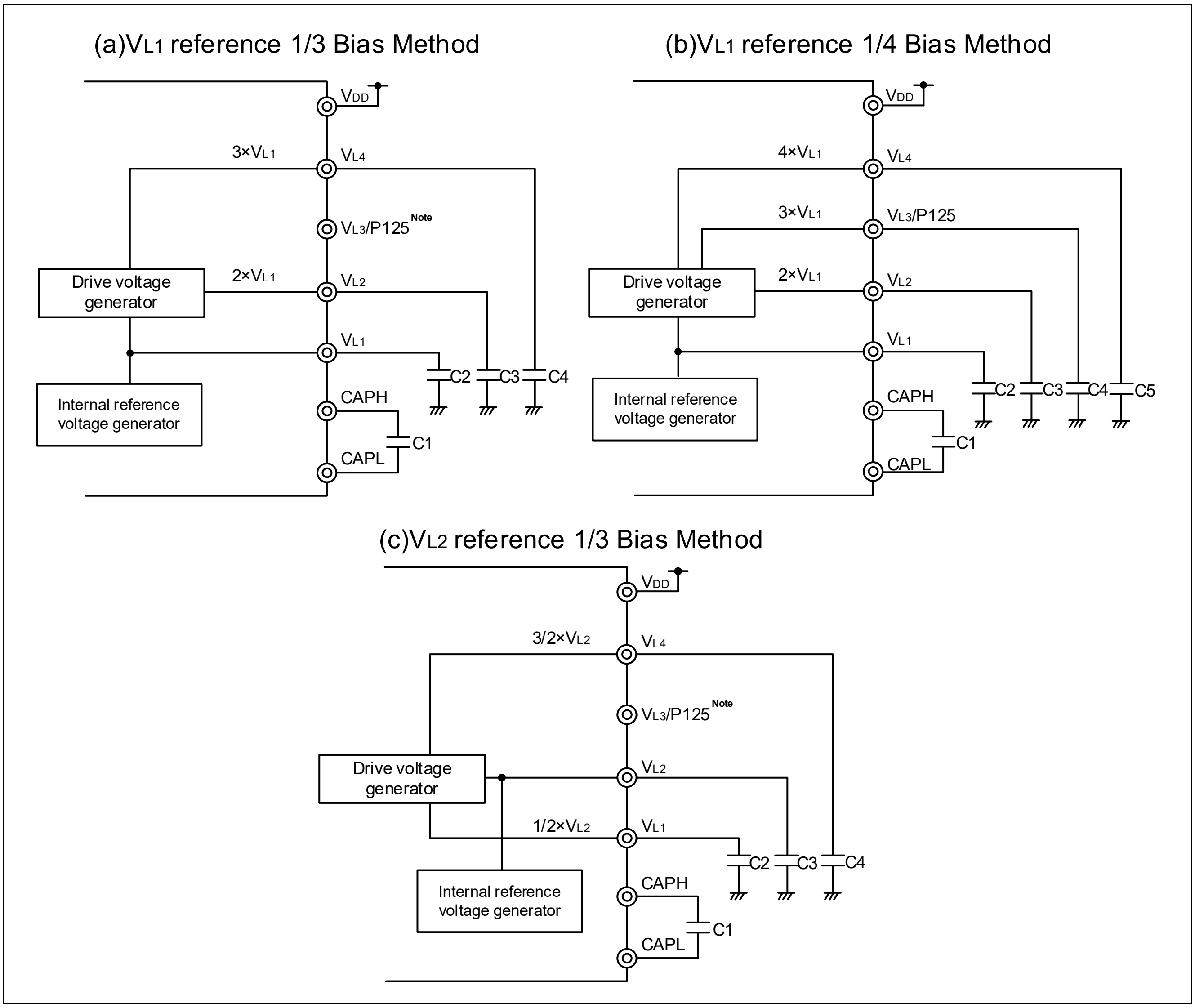
This is suitable for a battery set.

The operating current is small, and the LCD display does not become dim as the drive voltage is constant even when the battery voltage is reduced.

This method generates the reference voltage internally and boosts it by an external capacitor. As the reference voltage is adjusted by software (the LCD boost level control register, VLCD), the LCD contrast can be adjusted from 23 levels in RL78/L23.

Figure 3.3 shows a connection example of an internal voltage boosting method.

Figure 3.3 Connection Example of Internal Voltage Boosting Method



Note: V_{L3} can be used as a port(P125)

Remark: Use a capacitor with as little leakage as possible. Make sure to use a non-polar capacitor for C1.

3.3.3 Capacitor Split Method

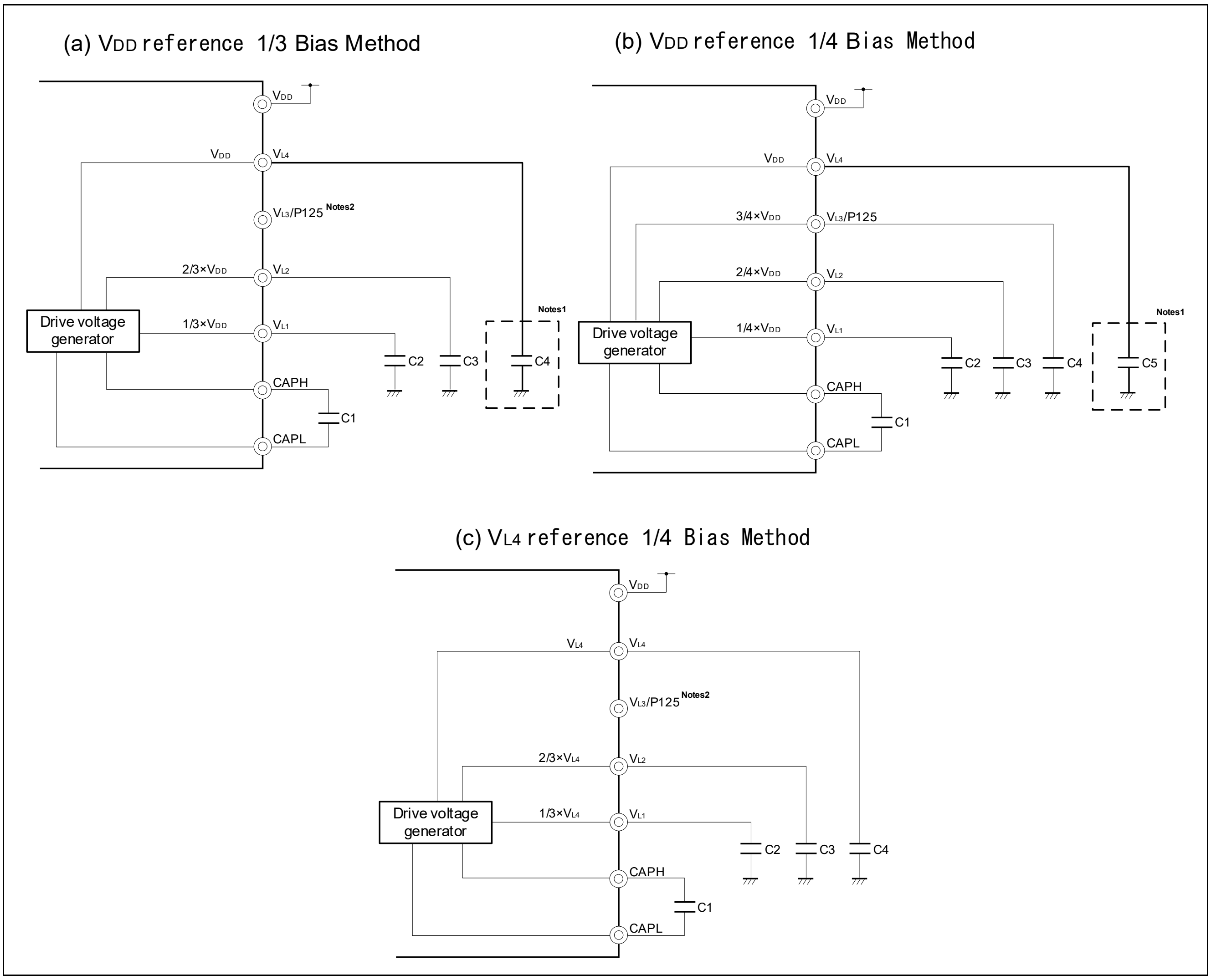
This is also suitable for a battery set.

This method has the smallest operating current among three LCD drive modes, and thus the LCD display becomes dim with the supply voltage decreased.

If you do not want the screen to be dim when the battery voltage is decreased, change the LCD drive method to internal voltage boosting method. The method works in an external circuit of the capacitor split method.

Figure 3.4 shows a connection example of capacitor split method.

Figure 3.4 Connection Example of Capacitor Split Method



Notes 1. As the V_{L4} pin is internally connected to the V_{DD} pin, the capacitor is not always essential. Note, however, that selection of the internal voltage boosting method requires connection of this capacitor. In addition, when only capacitor split method is in use, the capacitor can be connected as a means for stabilizing the voltage supplied to the LCD.

Notes 2. V_{L3} can be used as port-pin P125

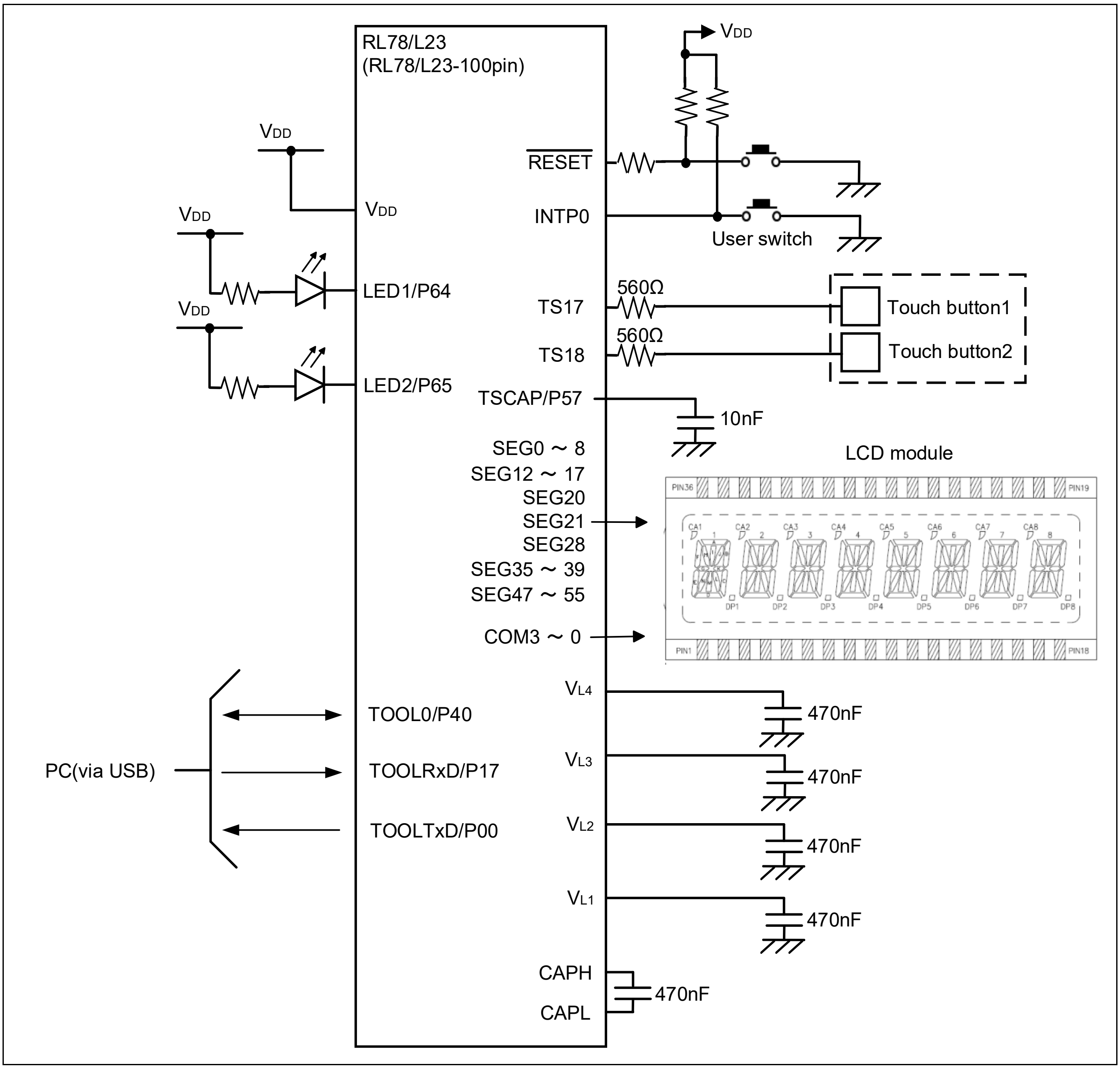
Remark: Use a capacitor with as little leakage as possible. Make sure to use a non-polar capacitor for C1.

4. Hardware

4.1 Hardware Example

Figure 4.1 shows the hardware configuration used in this application note.

Figure 4.1 Hardware Configuration



Notes 1. The above figure is simplified to show an overview of the hardware connection. When designing application circuits, make sure to handle unused pins appropriately to satisfy the electrical characteristics (connect input only ports independently to either V_{DD} or V_{SS} via resistors).

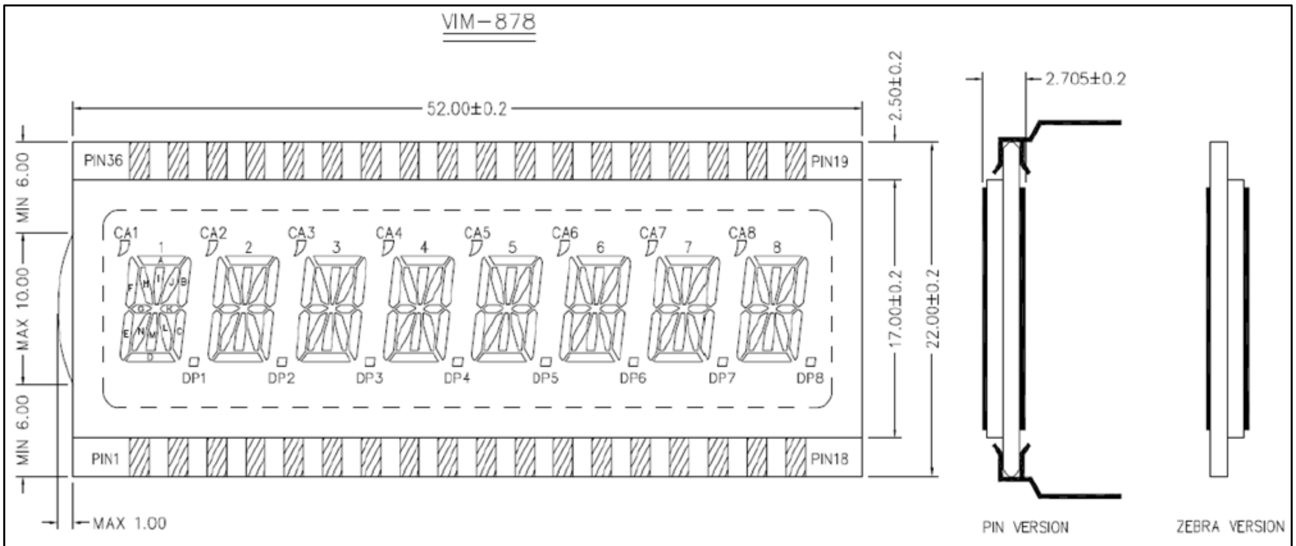
Notes 2. Make sure to set V_{DD} greater than the detection voltage (V_{LVD}) specified by the LVD.

4.2 LCD Module

This section describes the LCD module used in the sample code accompanying this application note.

The RL78/L23 Fast Prototyping Board is equipped with the LCD panel (16 segments, 8 digits) with the sockets. Figure 4.2 shows the panel image and the pin allocation table. Table 4-1 and Table 4-2 show the connections between the LCD panel and RL78/L23.

Figure 4.2 LCD Panel and Pin Allocation Table



PIN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
COM0	1D	1L	2D	2L	3D	3L	4D	4L	5D	5L	6D	6L	7D	7L	8D	8L	COM0	---
COM1	1E	1K	2E	2K	3E	3K	4E	4K	5E	5K	6E	6K	7E	7K	8E	8K	---	COM1
COM2	1F	1J	2F	2J	3F	3J	4F	4J	5F	5J	6F	6J	7F	7J	8F	8J	---	---
COM3	CA1	1I	CA2	2I	CA3	3I	CA4	4I	CA5	5I	CA6	6I	CA7	7I	CA8	8I	---	---

PIN	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
COM0	---	---	DP8	8M	DP7	7M	DP6	6M	DP5	5M	DP4	4M	DP3	3M	DP2	2M	DP1	1M
COM1	---	---	8C	8N	7C	7N	6C	6N	5C	5N	4C	4N	3C	3N	2C	2N	1C	1N
COM2	COM2	---	8B	8G	7B	7G	6B	6G	5B	5G	4B	4G	3B	3G	2B	2G	1B	1G
COM3	---	COM3	8A	8H	7A	7H	6A	6H	5A	5H	4A	4H	3A	3H	2A	2H	1A	1H

(Source: Datasheet for VIM-878-DP-FC-S-LV Varitronix Optoelectronics | Octopart)

Table 4-1 J5 Socket and LCD Module Connection Table

J5 socket pin number	LCD panel pin number	Symbol	port	Symbol and port	Pin number	LCD Header
1	LCD_1	SEG8	P54	P54(SEG8)	54	J1_21
2	LCD_2	SEG7	P53	P53(SEG7)	55	J1_22
3	LCD_3	SEG6	P52	P52(SEG6)	56	J1_23
4	LCD_4	SEG5	P51	P51(SEG5)	57	J1_24
5	LCD_5	SEG4	P50	P50(SEG4)	58	J1_25
6	LCD_6	SEG3	P97	P97(SEG3)	59	J1_26
7	LCD_7	SEG2	P96	P96(SEG2)	60	J1_27
8	LCD_8	SEG1	P95	P95(SEG1)	61	J1_28
9	LCD_9	SEG0	P94	P94(SEG0)	62	J1_29
10	LCD_10	SEG50	P07	P07(SEG50)	69	J1_34
11	LCD_11	SEG49	P06	P06(SEG49)	70	J1_35
12	LCD_12	SEG48	P05	P05(SEG48)	71	J1_36
13	LCD_13	SEG47	P04	P04(SEG47)	72	J1_37
14	LCD_14	SEG39	P14	P14(SEG39)	83	J1_45
15	LCD_15	SEG38	P13	P13(SEG38)	84	J1_46
16	LCD_16	SEG37	P12	P12(SEG37)	85	J1_47
17	LCD_17	COM0	P90	P90(COM0)	66	J1_33
18	LCD_18	COM1	P91	P91(COM1)	65	J1_32

Table 4-2 J6 Socket and LCD Module Connection Table

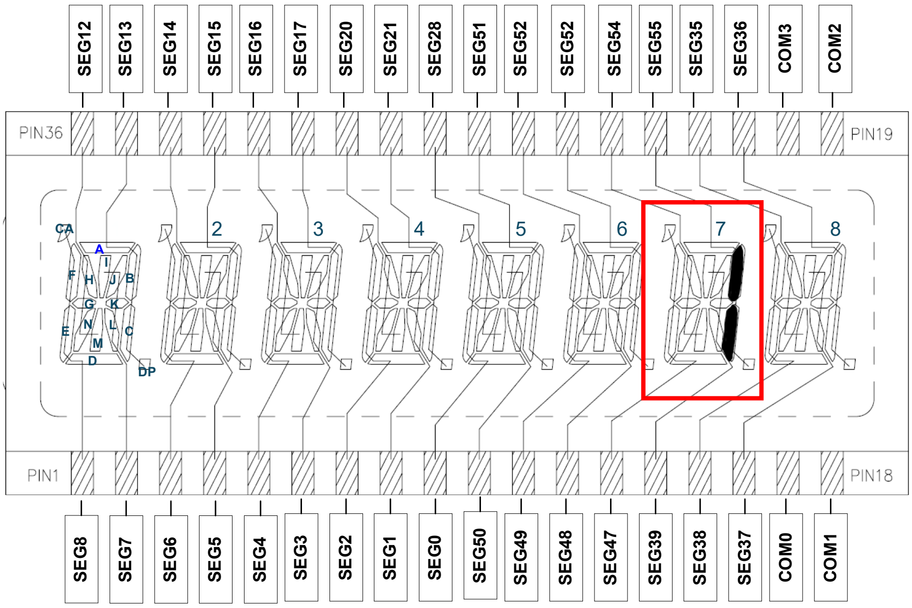
J6 socket pin number	LCD panel pin number	Symbol	port	Symbol and port	Pin number	LCD Header
1	LCD_19	COM2	P92	P92(COM2)	64	J1_31
2	LCD_20	COM3	P93	P93(COM3)	63	J1_30
3	LCD_21	SEG36	P11	P11(SEG36)	86	J1_48
4	LCD_22	SEG35	P10	P10(SEG35)	87	J1_49
5	LCD_23	SEG55	P145	P145(SEG55)	88	J1_50
6	LCD_24	SEG54	P144	P144(SEG54)	89	J1_51
7	LCD_25	SEG53	P143	P143(SEG53)	94	J1_56
8	LCD_26	SEG52	P142	P142(SEG52)	95	J1_57
9	LCD_27	SEG51	P141	P141(SEG51)	96	J1_58
10	LCD_28	SEG28	P130	P130(SEG28)	2	J1_1
11	LCD_29	SEG21	P31	P31(SEG21)	39	J1_8
12	LCD_30	SEG20	P30	P30(SEG20)	40	J1_9
13	LCD_31	SEG17	P75	P75(SEG17)	43	J1_12
14	LCD_32	SEG16	P74	P74(SEG16)	44	J1_13
15	LCD_33	SEG15	P73	P73(SEG15)	45	J1_14
16	LCD_34	SEG14	P72	P72(SEG14)	46	J1_15
17	LCD_35	SEG13	P71	P71(SEG13)	47	J1_16
18	LCD_36	SEG12	P70	P70(SEG12)	48	J1_17

Figure 4.3 shows an example of the 8-digit LCD panel display and the segments and their corresponding segment signals used to achieve the display.

Figure 4.3 Segments and Corresponding Signals

Example of displaying “1” on DISIT 7

The following processing is implemented based on the pin assignments indicated in Tables 4.1 and 4.2.



Segment signal SEG55 is connected to PIN23 of the LCD module.

From the pin assignment table:

PIN23 7C is connected to the COM1 signal

PIN23 7B is connected to the COM2 signal

When 06H is set in the SEG55 register, 7C and 7B will turn on and DISIT 7 will display "1".

PIN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	2
COM0	1D	1L	2D	2L	3D	3L	4D	4L	5D	5L	6D	6L	7D	7L	8D	8L	COM0	---	---	---	DP8	8M	DP7	7M	DP6	6
COM1	1E	1K	2E	2K	3E	3K	4E	4K	5E	5K	6E	6K	7E	7K	8E	8K	---	COM1	---	---	8C	8N	7C	7N	6C	6
COM2	1F	1J	2F	2J	3F	3J	4F	4J	5F	5J	6F	6J	7F	7J	8F	8J	---	---	COM2	---	8B	8G	7B	7G	6B	6
COM3	CA1	1I	CA2	2I	CA3	3I	CA4	4I	CA5	5I	CA6	6I	CA7	7I	CA8	8I	---	---	---	COM3	8A	8H	7A	7H	6A	6

(Additional info: VIM-878.pdf)

Table 4-3 Segments and Corresponding Commons (DISIT 7)

LCD Module PIN No. Note	LCD Display Data Register Note	COM3	COM2	COM1	COM0
		bit3	bit2	bit1	bit0
PIN23	SEG55	A	B	C	DP
PIN13	SEG47	CA	F	E	D
PIN14	SEG39	I	J	K	L
PIN24	SEG54	H	G	N	M

Note: For other DISITs, replace with the corresponding PIN number and segment signal register name.

4.3 Capacitive Touch Sensing Unit

This section describes the capacitive touch sensing unit used in this sample code.

RL78/L23 Fast Prototyping Board is equipped with two electrodes: touch buttons 1 and 2.

The touch buttons can be enabled by using the CTSU module, the TOUCH module, and QE for Capacitive Touch (development support tool for the capacitive touch sensor). For details on how to use the touch buttons, refer to Capacitive Sensor Microcontrollers CTSU Capacitive Touch Introduction Guide (R30AN0424).

4.4 Pins Used

Table 4-4 Pins Used and Their Functions(1/2)

Pin name	I/O	Function
P137/INTP0	Input	Detects input from the user switch and enters hour, minute or second setting mode
P57/TSCAP	-	Secondary power supply capacitor connection pin for measurement
P56/TS17	Input-Output	Detects input from the touch button 1 and increments hours, minutes and seconds displayed on the LCD
P55/TS18	Input-Output	Detects input from the touch button 2 and decrements hours, minutes and seconds displayed on the LCD
P94/SEG0	Output	LCD controller/driver common signals
P95/SEG1		
P96/SEG2		
P97/SEG3		
P50/SEG4		
P51/SEG5		
P52/SEG6		
P53/SEG7		
P54/SEG8		
P70/SEG12		
P71/SEG13		
P72/SEG14		
P73/SEG15		
P74/SEG16		
P75/SEG17		
P30/SEG20		
P31/SEG21		
P130/SEG28		
P10/SEG35		
P11/SEG36		
P12/SEG37		
P13/SEG38		
P14/SEG39		
P04/SEG47		
P05/SEG48		
P06/SEG49		
P07/SEG50		
P141/SEG51		
P142/SEG52		
P143/SEG53		
P144/SEG54		
P145/SEG55		
P90/COM0	Output	LCD controller/driver common signals
P91/COM1		
P92/COM2		
P93/COM3		
P87/V_{L1}	-	LCD drive voltage
P86/V_{L2}		
P85/V_{L4}		

Table 4-5 Pins Used and Their Functions(2/2)

Pin name	I/O	Function
P126/CAPL	-	Capacitor connection for LCD controller/driver
P127/CAPH		
P40/TOOL0	Input-Output	COM Port debugging
P17/TOOLRxD	Input	COM Port debugging
P00/TOOLTxD	Output	COM Port debugging

5. Software

5.1 Operation Overview

This sample code uses the RL78/L23 LCD controller/driver to display a clock in 24-hour mode. It stores the time measured by the RTC in the LCD display data memory area to change the time at each time an RTC constant-period interrupt occurs (once a second).

The sample code switches the time unit in the order of second, minute, and hour by pressing the user switch.

The time is adjusted by touch detection of the touch buttons, and the set time is displayed. When adjusting second, minute, or hour, the LCD display blinks on the digit of the adjusting time unit.

In the initial settings, the clock frequency, I/O ports, RTC, IT, and LCD controller/driver are configured.

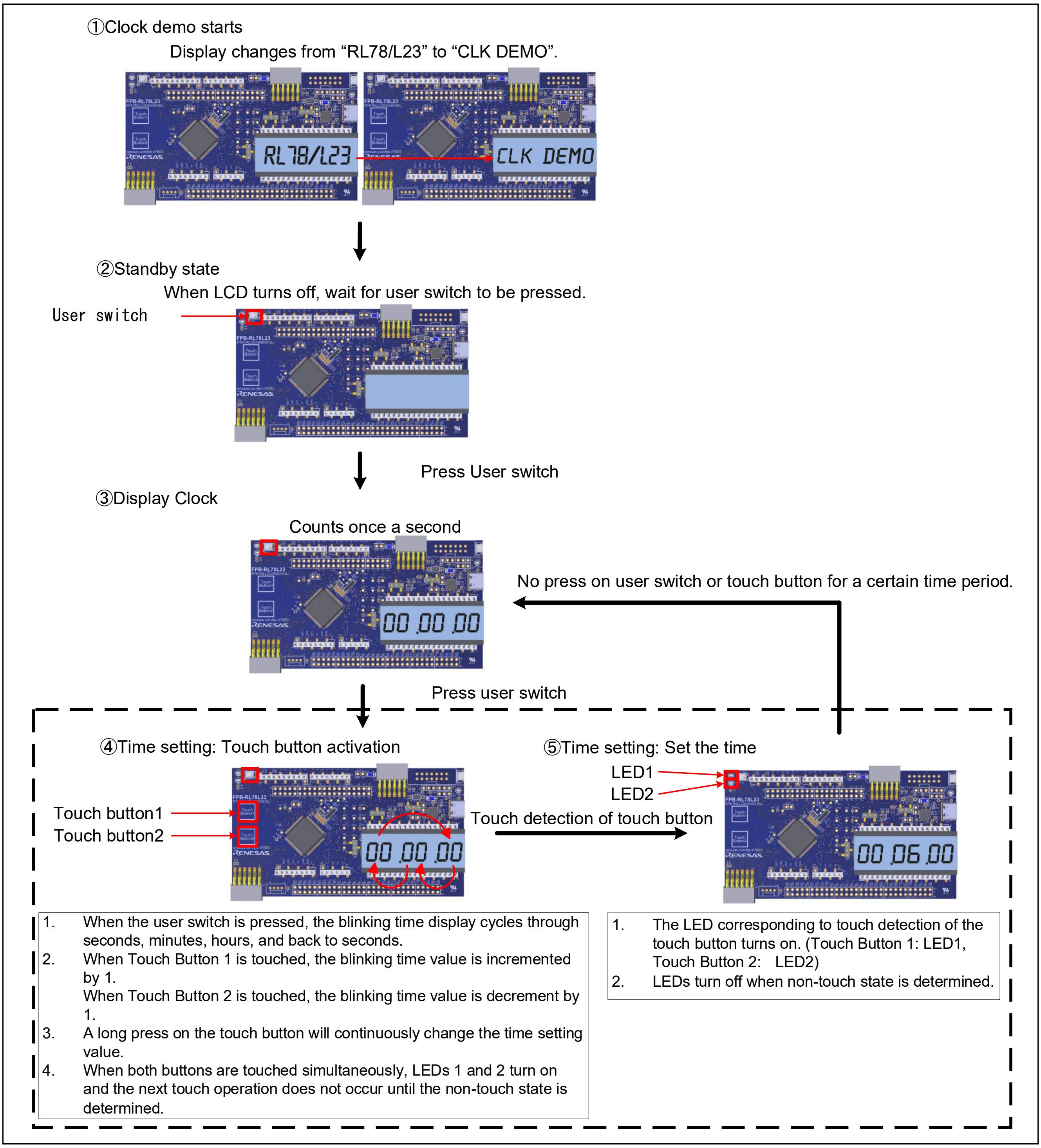
After the initial setting is complete, the sample code displays “RL78/L23” and “CLK DEMO” on the LCD in order, and transitions to the wait state. The wait state is released by a short press on the user switch (both edges of INTP0 detected) and the time is displayed. When the RTC constant-period interrupt occurs, the time display changes by 1 second. You can set any time to display by pressing the user switch.

The touch buttons are disabled in the time display state, and available only in second, minute, and hour adjustment states. The measurement cycle is 20ms.

The first, second, and third presses on the user switch allow you adjust second, minute, and hour, respectively. And the fourth press returns to the second adjustment state. If the user switch is not pressed and no touch detection of the touch buttons occurs within a certain time, the sample code returns to the time display state to display the set time. In the time adjustment state, touch detection of touch button 1 increases the value by 1 second, 1 minute, or 1 hour according to the selected time unit. Touch detection of touch button 2 decreases the value by 1 second, 1 minute, or 1 hour. The adjusted value changes continuously when touch detection of a touch button is held.

For more information, refer to the state transition diagram on the next page.

Figure 5.1 State Transition Diagram



5.2 File Composition

Table 5-1 lists the files used in the sample code. Files generated by the integrated development environment are not included in this table.

Table 5-1 Files Used in the Sample Code(1/2)

Folder and file name	Description	Smart Configurator Usage
¥clock_demo<DIR>	Sample code folder	
¥src<DIR>	Source folder	
¥clock_demo.c	Clock demo source folder	
¥clock.c	Clock source file	
¥clock.h	Clock header file	
¥lcd_segdata.c	LCD display source file	
¥smc_gen<DIR>	Smart configurator generation components storage folder	✓
¥Config_INTC<DIR>	External interrupt components folder	✓
¥Config_LCD<DIR>	LCD components folder	✓
¥Config_PORT<DIR>	PORT components folder	✓
¥Config_RTC<DIR>	RTC components folder	✓
¥Config_TAU0_1<DIR>	TAU0_1 components folder	✓
¥Config_TAU0_3<DIR>	TAU0_3 components folder	✓
¥Config_TAU0_4<DIR>	TAU0_4 components folder	✓
¥Config_TAU0_5<DIR>	TAU0_5 components folder	✓
¥r_ctsu<DIR>	CTSU driver folder	✓
¥rm_touch<DIR>	TOUCH driver folder	✓

Table 5-2 Files Used in the Sample Code(2/2)

Folder and file name	Description	Smart Configurator Usage
¥clock_demo<DIR>	Sample code folder	
¥qe_gen<DIR>	QE for Capacitive Touch generation file	
¥qe_touch_config.c	QE generation configuration source file	
¥qe_touch_config.h	QE generation configuration header file	
¥qe_touch_define.h	QE generation definition header file	
¥qe_touch_sample.c	Main function source file including touch operation	
¥QE-Touch<DIR>	QE configuration folder	

Precaution regarding Table 5-1 and Table 5-2.

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

5.3 Smart Configurator Settings

Figure 5.2 shows the clock settings of the Smart Configurator.

Figure 5.2 Clock Settings

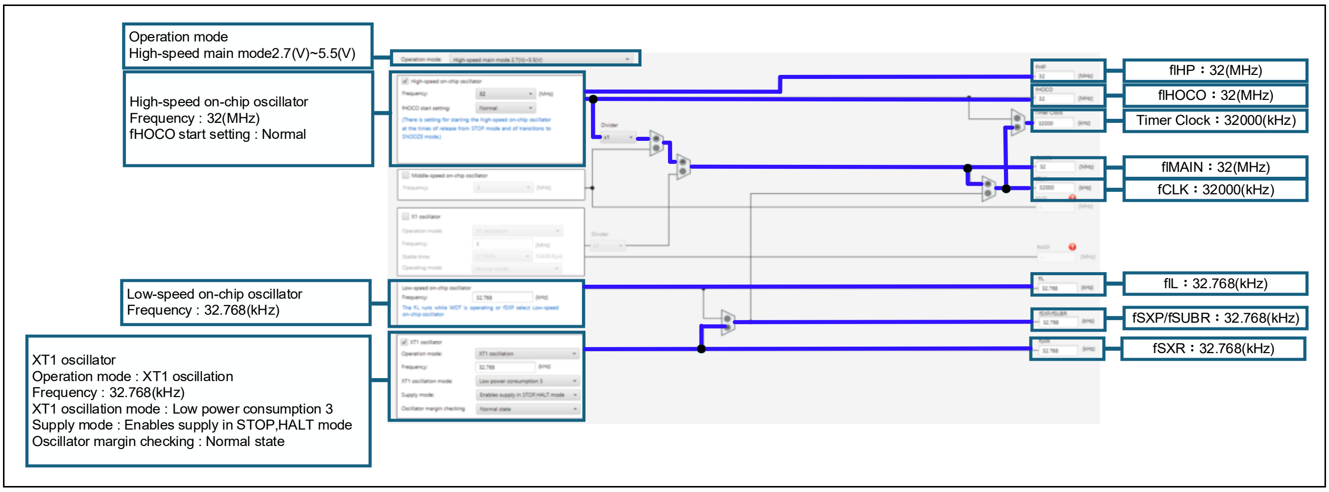


Figure 5.3 shows the system settings of the Smart Configurator.

Figure 5.3 System Settings

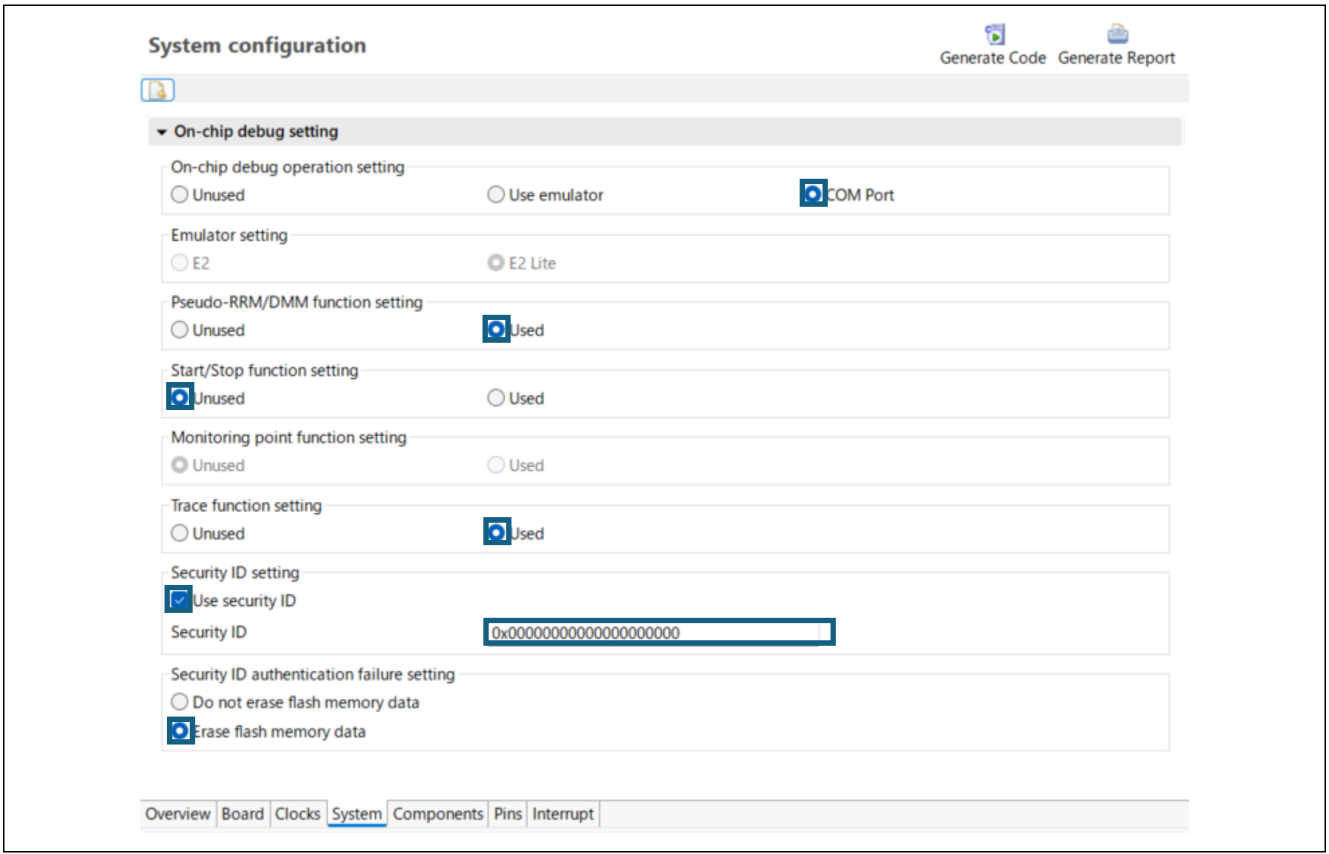


Figure 5.4 shows the LVD components (Config_LVD0) settings.

Figure 5.4 LVD Components (Config_LVD0) Settings

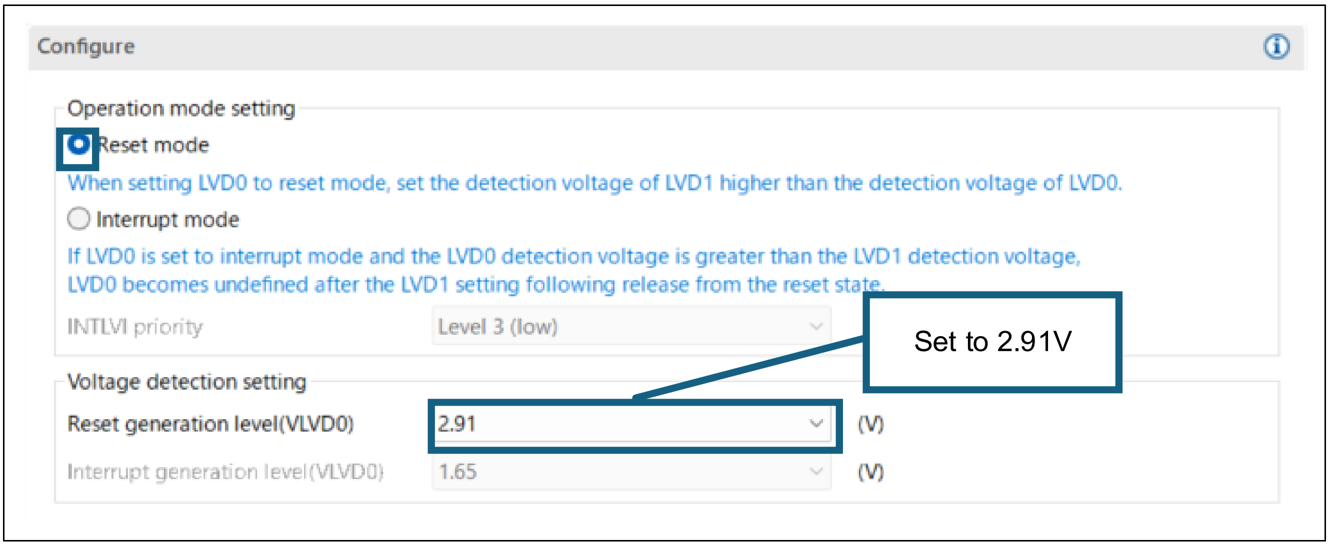


Figure 5.5 shows the settings of external interrupt components (Config_INTC).

Figure 5.5 External Interrupt Components (Config_INTC) Settings

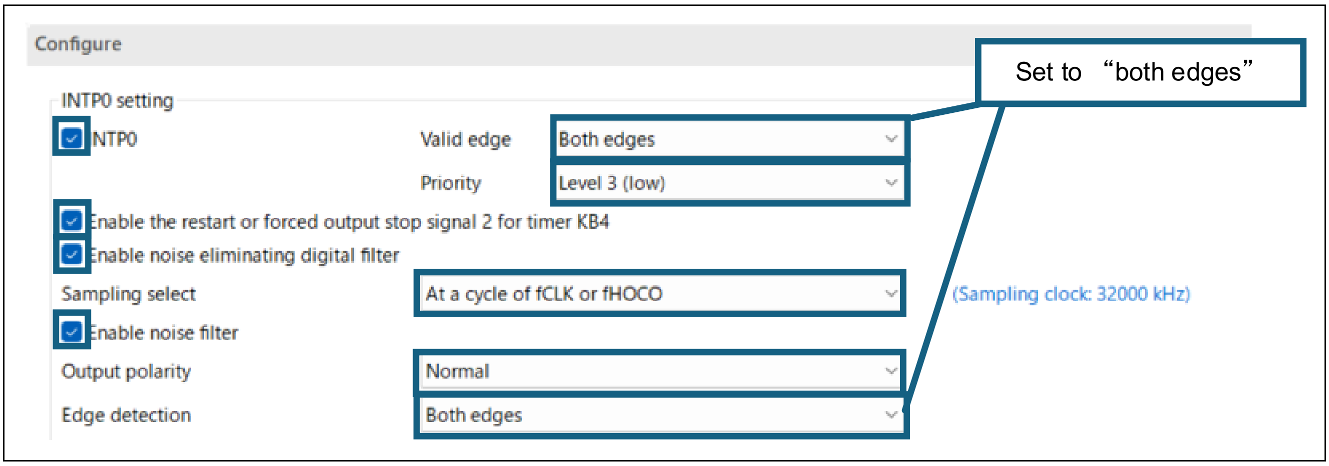


Figure 5.6 shows the settings of RTC components (Config_RTC).

Figure 5.6 RTC Components (Config_RTC)Settings

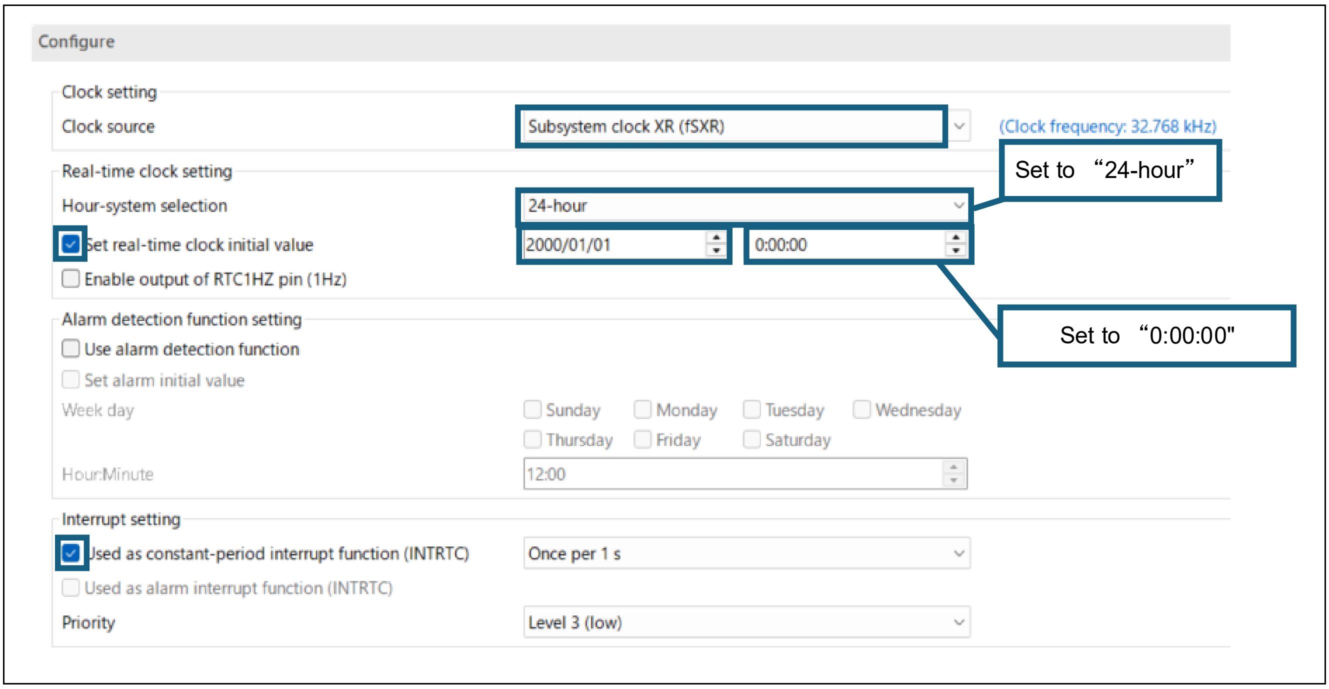


Figure 5.7 shows settings of TAU0_1 components (Config_TAU0_1).

Figure 5.7 TAU0_1 Components (Config_TAU0_1) Settings

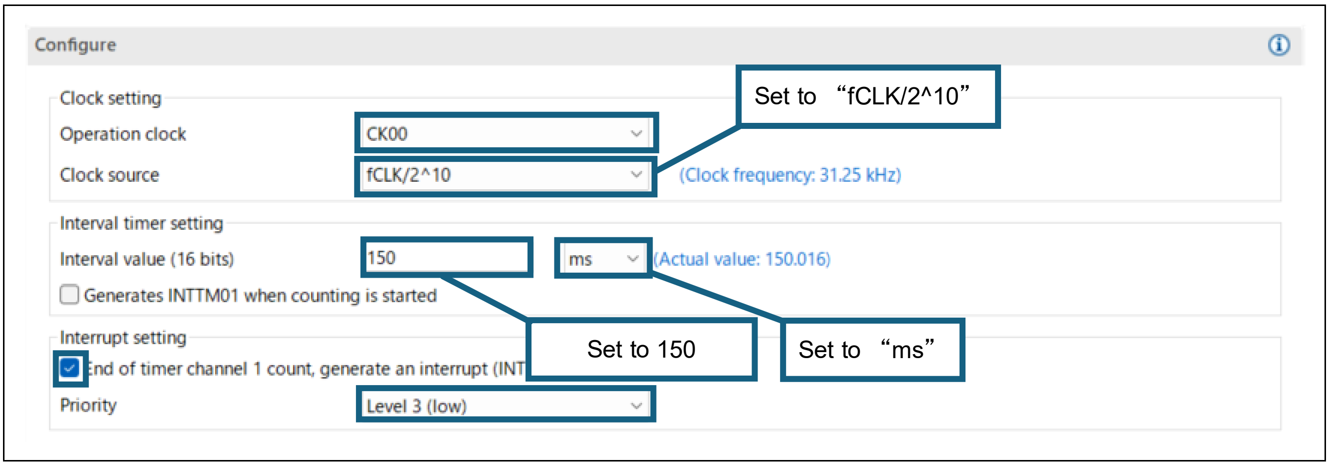


Figure 5.8 shows the settings of TAU0_3 components (Config_TAU0_3).

Figure 5.8 TAU0_3 Components (Config_TAU0_3) Settings

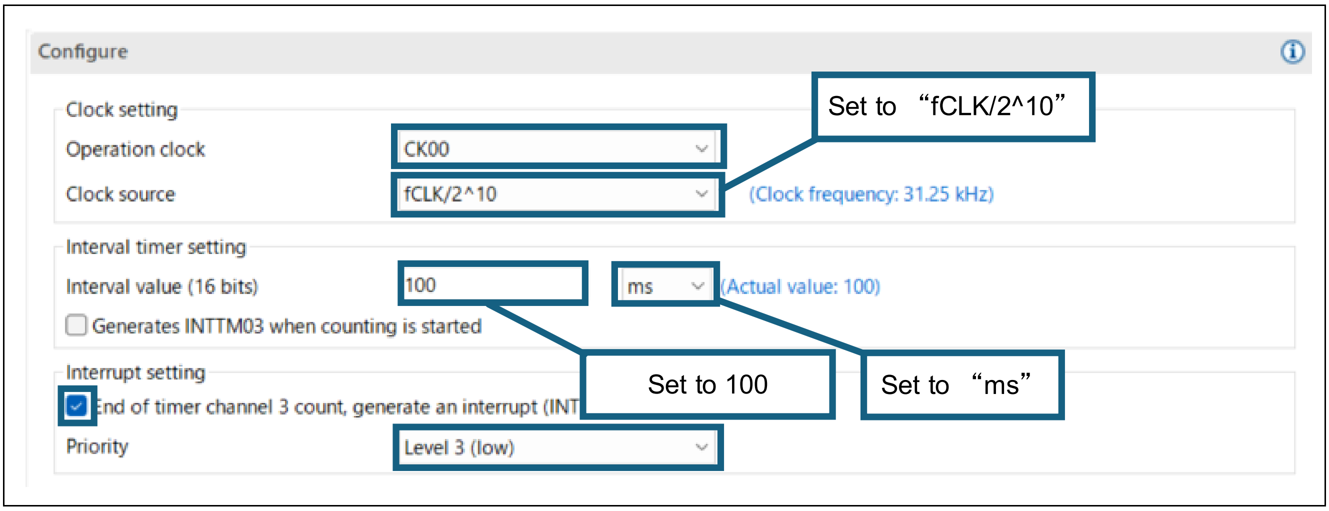


Figure 5.9 shows the settings of TAU0_4 components (Config_TAU0_4).

Figure 5.9 TAU0_4 Components (Config_TAU0_4) Settings

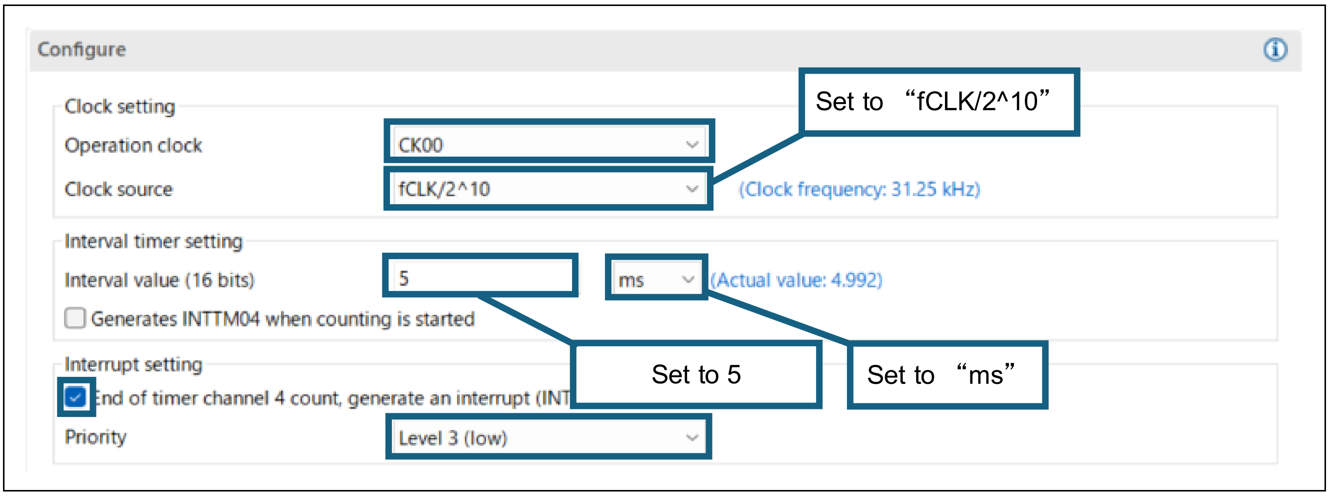


Figure 5.10 shows the settings of TAU0_5 components (Config_TAU0_5).

Figure 5.10 TAU0_5 Components Settings

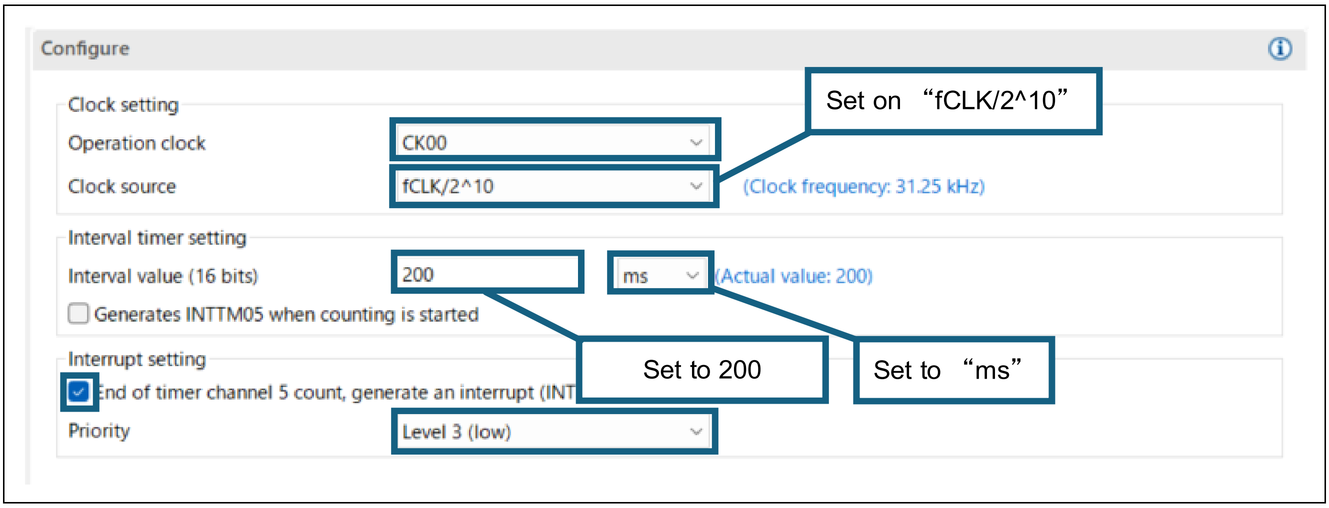


Figure 5.11 shows the settings of LCD components (Config_LCD).

Figure 5.11 LCD Components (Config_LCD) Setting

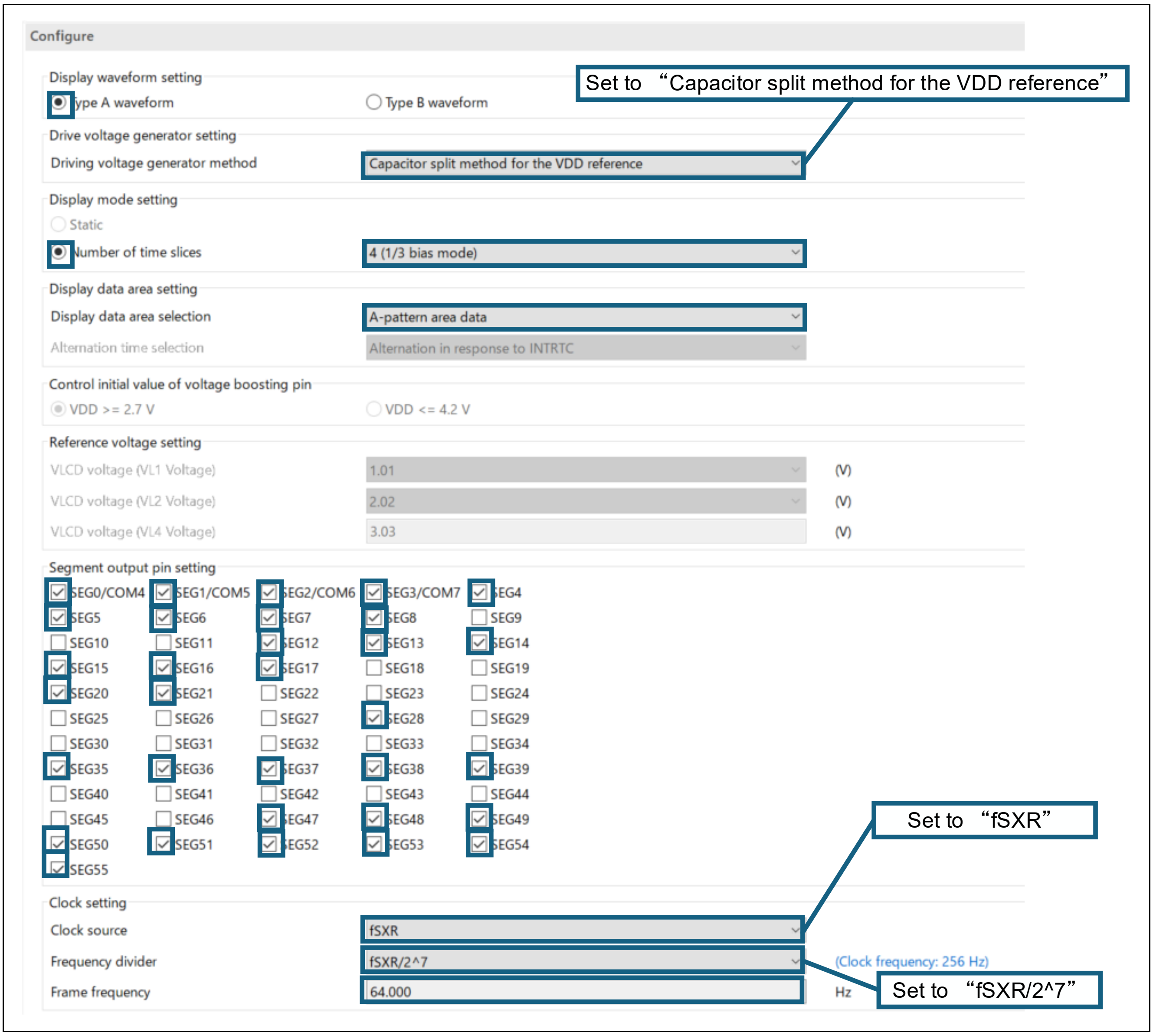
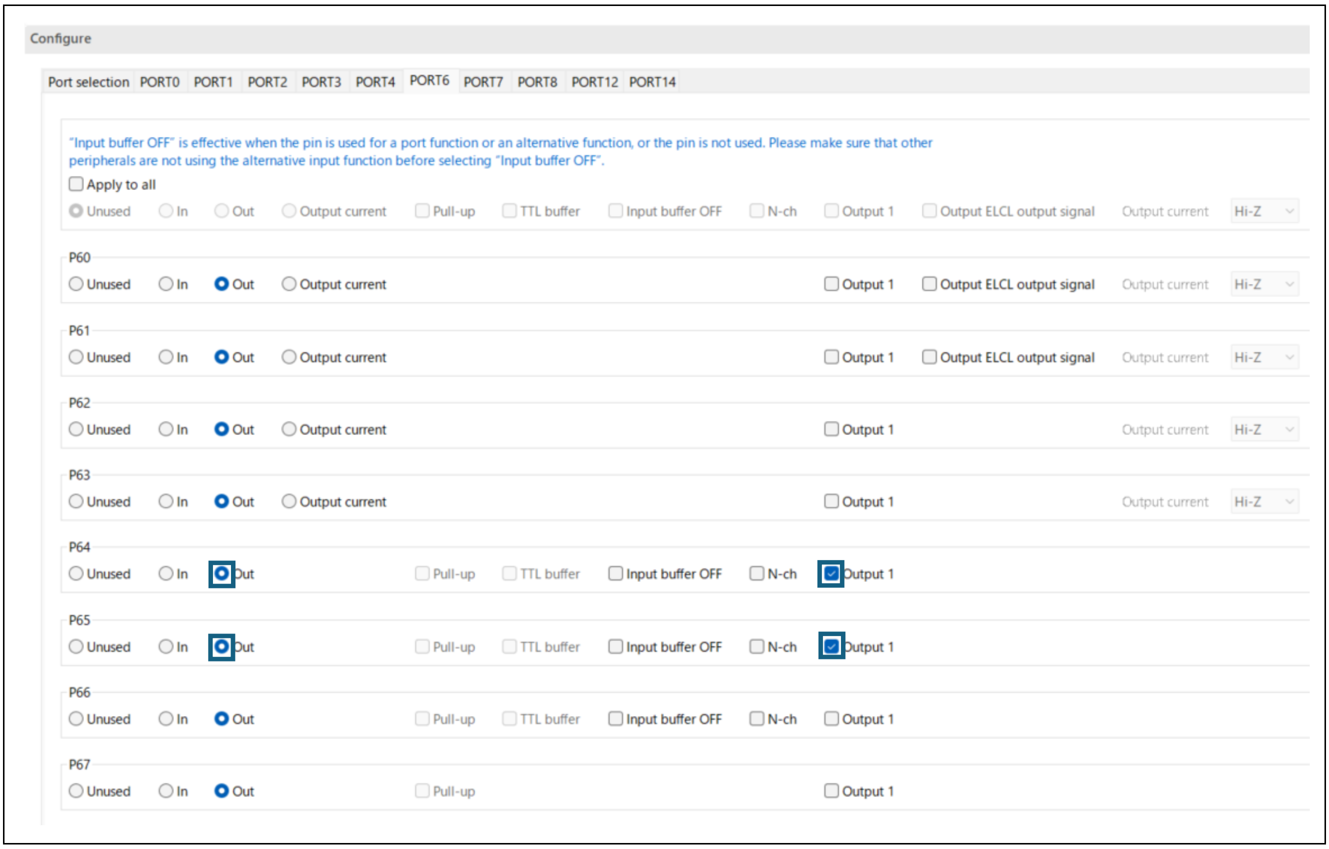


Figure 5.12 and Figure 5.13 show the settings of PORT components (Config_PORT).

Figure 5.12 PORT Components (Config_PORT) Settings

Figure 5.13 PORT6 Settings



Note: Only P64 and P65 are specified as LED ports in this document. Other ports are set to prevent unnecessary current consumption caused by the floating state. Settings to prevent unnecessary current consumption by ports floating are applied to ports other than these. Refer to “2.3 Connection of Unused Pins” in RL78/L23 User’s Manual: Hardware (R01UH1082) for details on how to properly handle the application’s unused ports and design your application to meet the electric characteristics.

Figure 5.14 shows the CTSU driver (r_ctsu) settings

Figure 5.14 CTSU Driver (r_ctsu) Settings

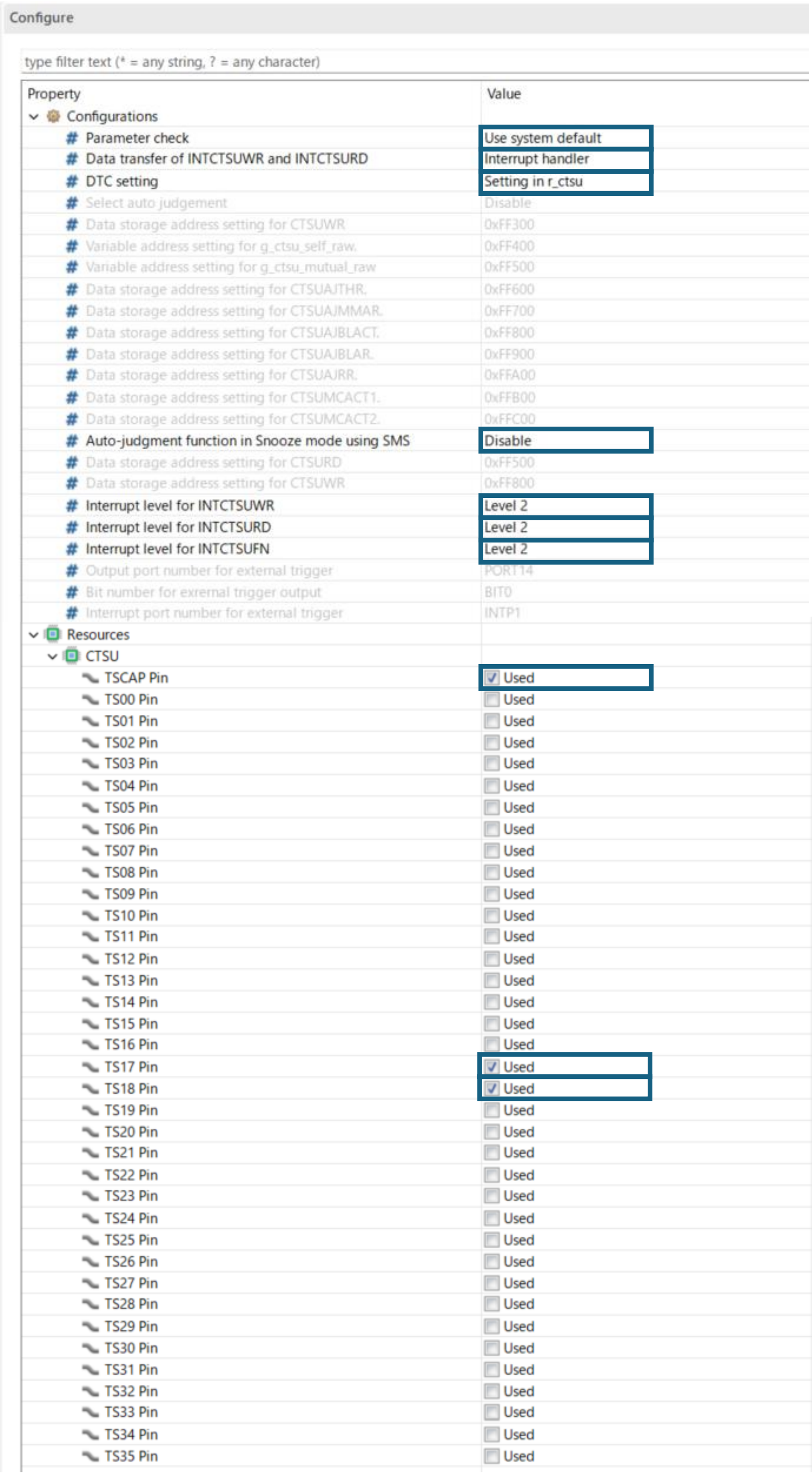


Figure 5.15 shows the settings of the TOUCH driver (rm_touch).

Figure 5.15 TOUCH Driver (rm_touch) Settings

Configure

type filter text (* = any string, ? = any character)

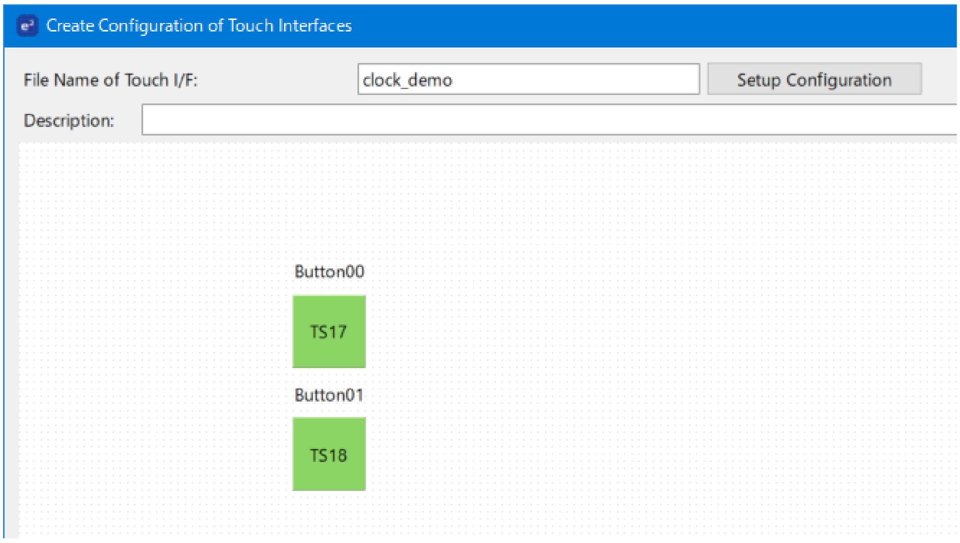
Property	Value
Configurations	
# Parameter check	Use system default
# Support QE monitor using UART	Disable
# Support QE tuning using UART	Disable
# UART channel	UART0
# Type of chattering suppression	TypeA : Counter of exceed threshold is hold within hysteresis range.

5.4 Capacitive Touch Settings

5.4.1 Touch Interface Configuration

Figure 5.16 shows the touch interface configuration. TS17 and TS18 are measured with the self-capacitance method.

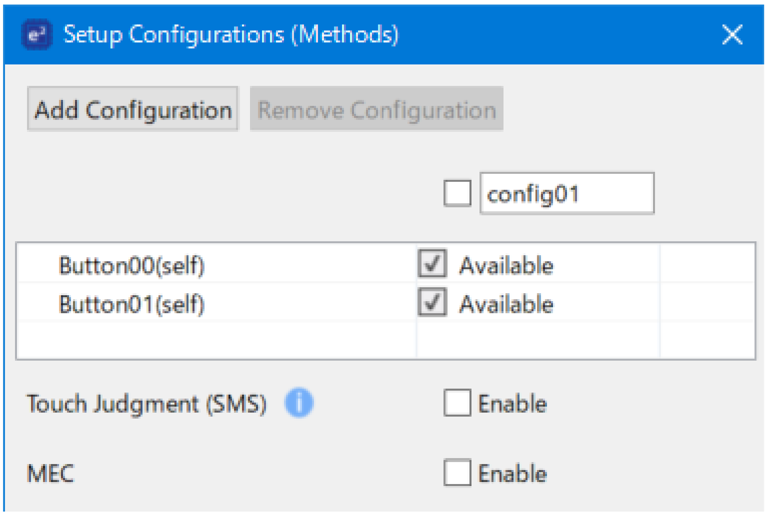
Figure 5.16 Touch Interface Configuration



5.4.2 Configuration (Methods) Settings

Figure 5.17 shows the touch interface configuration (methods) settings.

Figure 5.17 Configuration (Methods) Settings



5.4.3 Tuning Results

Figure 5.18 shows the QE tuning results of the touch interface. This sample code operates with the setting values shown in the figure below.

The tuning results depend on the operating environment when QE tuning is performed. Thus, if QE tuning is performed again, these values may change.

Figure 5.18 QE Tuning Results

Tuning | Gesture

Touch I/F Configuration: clock_demo

Method	Kind	Name	Touch Sensor	Parasitic Capacitance[pF]	Sensor Drive Pulse Frequency[MHz]	Threshold	Scan Time[ms]	Overflow
config01	Button(self)	Button00	TS17	7.5	5.436	3990	0.576	None
config01	Button(self)	Button01	TS18	7.312	5.547	3967	0.576	None

5.5 Constants

Table 5-3 lists the constants used in the sample code.

Table 5-3 Constants Used in the Sample Code

Constant Name	Setting Value	Contents
MAX_SEC_MIN	59	Clock maximum minute and second values
MAX_HOUR	23	Clock maximum hour value
MIN_SEC_MIN_HOUR	0	Clock minimum values (0) of hour, minute and second
UNIT_SEC	0	Clock unit of seconds
UNIT_MIN	1	Clock unit of minutes
UNIT_HOUR	2	Clock unit of hours
TIME_SETTING_FINISH	5	Time to end time setup (sec)
LONG_PRESS_SEC	3	Continuous touch detection time (sec)
COUNTINUOUS_START_NUM	3	Continuous touch detection count

5.6 Variables

Table 5-4 and Table 5-5 lists the static variables.

Table 5-4 Static Variables Used in the Sample Code(1/2)

Type	Variable Name	Contents	Function Used
uint8_t	g_rtc_interrupt_flag	RTC interrupt flag	r_rtc_callback r_change_time
uint8_t	g_show_segdata[40][4]	Array to store characters displayed on the LCD	r_lcd_show
uint8_t	g_digit_segdata[8][4]	Array to store the LCD display digits	r_lcd_show
uint8_t	g_touch_button_flag	Flag to enable the touch function	qe_touch_main r_rtc_callback, r_time_unit_change
uint8_t	s_lcd_seconds	Variable to store the clock second value	r_rtc_callback r_time_show r_change_time
uint8_t	s_lcd_minutes	Variable to store the clock minute value	r_rtc_callback r_time_show r_change_time
uint8_t	s_lcd_hours	Variable to store the clock hour value	r_rtc_callback r_time_show r_change_time
uint8_t	sp_time_address[3]	Array to store clock second, minute, and hour variables	r_touch_plus_set_time r_touch_minus_set_time
uint8_t	s_unit	Variable to store clock unit	r_touch_plus_set_time r_touch_minus_set_time r_blink_set_time r_time_unit_change
uint8_t	s_clock_mode	Variable to store clock mode	r_rtc_callback r_time_unit_change
uint8_t	s_blink_flag	Flag to blink the selected unit on the LCD	r_tau0_1_blink_callback
uint8_t	s_set_time_limit_count	Flag to count time to end of time setup state	r_userswitch_callback r_rtc_callback r_touch_plus_set_time r_touch_minus_set_time r_touch_both_buttons
uint8_t	s_clock_start_flag	Flag to indicate start of LCD time display	r_clock_demo_start r_userswitch_release r_userswitch_callback
uint8_t	s_100ms_count	Flag to count a wait time every 100ms	r_tau0_3_delay_callback r_delay_100ms
uint8_t	s_chattering_flag	Flag to count time of long press on the user switch	r_userswitch_callback r_tau0_4_chattering_callback
uint8_t	s_prev_button1_flag	Flag to store the previous state of touch button 1	r_touch_plus_set_time r_touch_both_buttons r_no_touch

Table 5-5 Static Variables Used in the Sample Code(2/2)

Type	Variable Name	Contents	Function Used
uint8_t	s_prev_button2_flag	Flag to store the previous state of touch button 2	r_touch_minus_set_time r_touch_both_buttons r_no_touch
uint8_t	s_continuous_touch_flag	Flag to determine an interval of continuous touch on the touch button	r_tau0_5_continuous_callback r_touch_plus_set_time r_touch_minus_set_time r_touch_both_buttons r_no_touch
uint8_t	s_continuous_touch_count	Flag to determine continuous touch start count on the touch button	r_tau0_5_continuous_callback r_touch_plus_set_time r_touch_minus_set_time
uint8_t	s_userswitch_on_off_flag	Flag to determine ON/OFF state of the user switch	r_userswitch_release r_userswitch_callback

5.7 Functions

Table 5-6 lists the functions used in the sample code (clock.c).

Table 5-6 Functions Used in the Sample Code (clock.c)

Function Name	Outline
void r_clock_demo_start(void);	Clock demo start processing
void r_userswitch_release(void);	User switch detection processing
void r_change_to_sub(void);	Transition to the sub-system clock operation
void r_change_to_main(void);	Transition to the main system clock operation
void r_userswitch_callback(void);	Callback for user switch press interrupt
void r_rtc_callback(void);	Callback for RTC interrupt
void r_tau0_1_blink_callback(void);	Callback for TAU0_1 interrupt
void r_tau0_3_delay_callback(void);	Callback for TAU0_3 interrupt
void r_tau0_4_chattering_callback(void);	Callback for TAU0_4 interrupt
void r_tau0_5_continuous_callback(void);	Callback for TAU0_5 interrupt
void r_touch_plus_set_time(void);	Processing for touch detection of touch button 1
void r_touch_minus_set_time(void);	Processing for touch detection of touch button 2
void r_touch_both_buttons(void);	Processing for simultaneous touch detection of touch buttons 1 and 2
void r_no_touch(void);	Touch release detection processing
static void r_lcd_show(uint8_t value, uint8_t digit);	Display of LCD values
static void r_time_show(void);	Display of time on the LCD
static void r_change_time(void);	Acquisition of time from the RTC register
static void r_blink_set_time(void);	Blinking of the selected unit during time setup
static void r_array_show(uint8_t* array, uint8_t num, uint8_t delay);	Sequential display processing of array data on LCD
static void r_clear_show(void);	Clearing of LCD display
static void delay_100ms(uint8_t num);	Delay processing
static void r_time_unit_change(void) ;	Change of time unit to be set

5.8 Function Specifications

The following tables list the sample code (clock.c) function specifications.

r_clock_demo_start

Outline	Clock demo start processing
Header	clock.h
Declaration	void r_clock_demo_start(void)
Description	Displays "RL78/L23" and "CLK DEMO" on the LCD when the demo starts.
Arguments	None
Return Value	None
Remarks	None

r_userswitch_release

Outline	User switch detection processing
Header	clock.h
Declaration	void r_userswitch_release(void)
Description	Changes the corresponding flag depending on release of the user switch.
Arguments	None
Return Value	None
Remarks	None

r_change_to_sub

Outline	Transition to the sub-system clock operation
Header	clock.h
Declaration	void r_change_to_sub(void)
Description	Switches the current clock from the main system clock to the sub-system clock.
Arguments	None
Return Value	None
Remarks	None

r_change_to_main

Outline	Transition to the main system clock operation
Header	clock.h
Declaration	void r_change_to_main (void)
Description	Switches the current clock from the sub-system clock to the main system clock.
Arguments	None
Return Value	None
Remarks	None

r_userswitch_callback

Outline	Callback for interrupt when the user switch is pressed.
Header	clock.h
Declaration	void r_userswitch_callback(void)
Description	With short press, starts time setup and calls r_time_unit_change.
Arguments	None
Return Value	None
Remarks	None

r_rtc_callback

Outline	Callback for RTC interrupt
Header	clock.h
Declaration	void r_rtc_callback(void)
Description	Counts time up every time an interrupt occurs. Counts to return to clock when performing time setup by pressing the user switch.
Arguments	None
Return Value	None
Remarks	None

r_tau0_1_blink_callback

Outline	Callback for TAU0_1 interrupt
Header	clock.h
Declaration	void r_tau0_1_blink_callback(void)
Description	Blinks the selected unit of time on the LCD when setting time.
Arguments	None
Return Value	None
Remarks	None

r_tau0_3_delay_callback

Outline	Callback for TAU0_3 interrupt
Header	clock.h
Declaration	void r_tau0_3_delay_callback(void)
Description	Increments the delay count variable with the timer (TAU0 channel 3).
Arguments	None
Return Value	None
Remarks	None

r_tau0_4_chatering_callback

Outline	Callback for TAU0_4 interrupt
Header	clock.h
Declaration	void r_tau0_4_chattering_callback(void)
Description	Disables external interrupts for a certain time period to prevent chattering.
Arguments	None
Return Value	None
Remarks	None

r_tau0_5_continuous_callback

Outline	Callback for TAU0_5 interrupt
Header	clock.h
Declaration	void r_tau0_5_continuous_callback(void)
Description	Determines continuous touch on the touch button
Arguments	None
Return Value	None
Remarks	None

r_touch_plus_set_time

Outline	Processing for touch detection of touch button 1
Header	clock.h
Declaration	void r_touch_plus_set_time(void)
Description	Adds 1 to the LCD display of the currently set time unit when touch detection of touch button 1 occurs.
Arguments	None
Return Value	None
Remarks	None

r_touch_minus_set_time

Outline	Processing for touch detection of touch button 2
Header	clock.h
Declaration	void r_touch_minus_set_time(void)
Description	Subtracts 1 from the LCD display of the currently set time unit when touch detection of touch button 2 occurs.
Arguments	None
Return Value	None
Remarks	None

r_touch_both_buttons

Outline	Processing for simultaneous touch detection of touch buttons 1 and 2
Header	clock.h
Declaration	void r_touch_both_buttons(void)
Description	Stops the timer and resets control variables when simultaneous touch detection of touch buttons 1 and 2 occurs.
Arguments	None
Return Value	None
Remarks	None

r_no_touch

Outline	Touch release detection processing
Header	clock.h
Declaration	void r_no_touch(void)
Description	Initializes the internal state and stops controls such as the timer when no touch detection of either touch button occurs.
Arguments	None
Return Value	None
Remarks	None

r_lcd_show

Outline	Display of LCD values	
Header	None	
Declaration	static void r_lcd_show(uint8_t value. Uint8_t digit)	
Description	Displays values on the LCD by setting the numeral/character code and digit position (LCD digit number) stored in the global variables.	
Arguments	uint8_t value	Numeral and character to be displayed
	uint8_t digit	Digit to be display
Return Value	None	
Remarks	None	

r_time_show

Outline	Display of time on the LCD
Header	None
Declaration	static void r_time_show(void)
Description	Displays values stored in the global variables s_lcd_hours, s_lcd_minutes, and s_lcd_seconds on the LCD.
Arguments	None
Return Value	None
Remarks	None

r_change_time

Outline	Acquisition of time from the RTC register
Header	None
Declaration	static void r_change_time(void)
Description	Increments the global variable value by 1 second each time the RTC interrupt occurs.
Arguments	None
Return Value	None
Remarks	None

r_blink_set_time

Outline	Blinking of the selected unit during time setup
Header	None
Declaration	static void r_blink_set_time(void)
Description	Blinks the selected part of the LCD when setting time.
Arguments	None
Return Value	None
Remarks	None

r_array_show

Outline	Sequential display processing of array data on LCD	
Header	None	
Declaration	static void r_array_show(uint8_t * array, uint8_t num, uint8_t delay)	
Description	Displays the numerals and characters on the LCD that correspond to the numbers stored in the array.	
Arguments	uint8_t * array	Pointer to the value to be displayed
	uint8_t num	Number of elements to be displayed
	uint8_t delay	Wait time after each display (every 100ms)
Return Value	None	
Remarks	None	

r_clear_show

Outline	Clearing of LCD display
Header	None
Declaration	static void r_clear_show(void)
Description	Displays blank for all 8 digits on the LCD to clear all displayed contents.
Arguments	None
Return Value	None
Remarks	None

r_delay_100ms

Outline	Delay processing	
Header	None	
Declaration	static void r_delay_100ms(uint8_t num)	
Description	Generates 100ms delay.	
Arguments	uint8_t num	Wait time in 100ms intervals, e.g. 300ms wait when num = 3
Return Value	None	
Remarks	None	

r_time_unit_change

Outline	Change of time unit to be set
Header	None
Declaration	static void r_time_unit_change(void)
Description	Changes unit of time to be set (sec -> minute -> hour -> sec).
Arguments	None
Return Value	None
Remarks	None

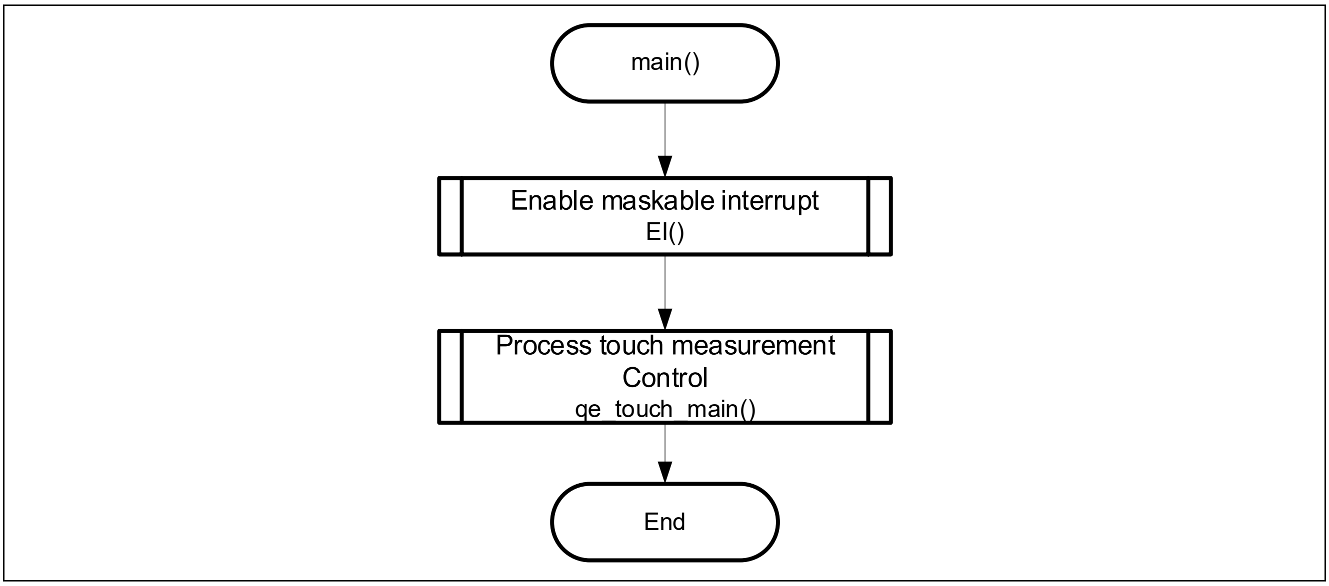
5.9 Flowcharts

The flowchart of the main functions showing the flow of this sample code is shown below.

5.9.1 main Function

Figure 5.19 shows a flowchart of the main function.

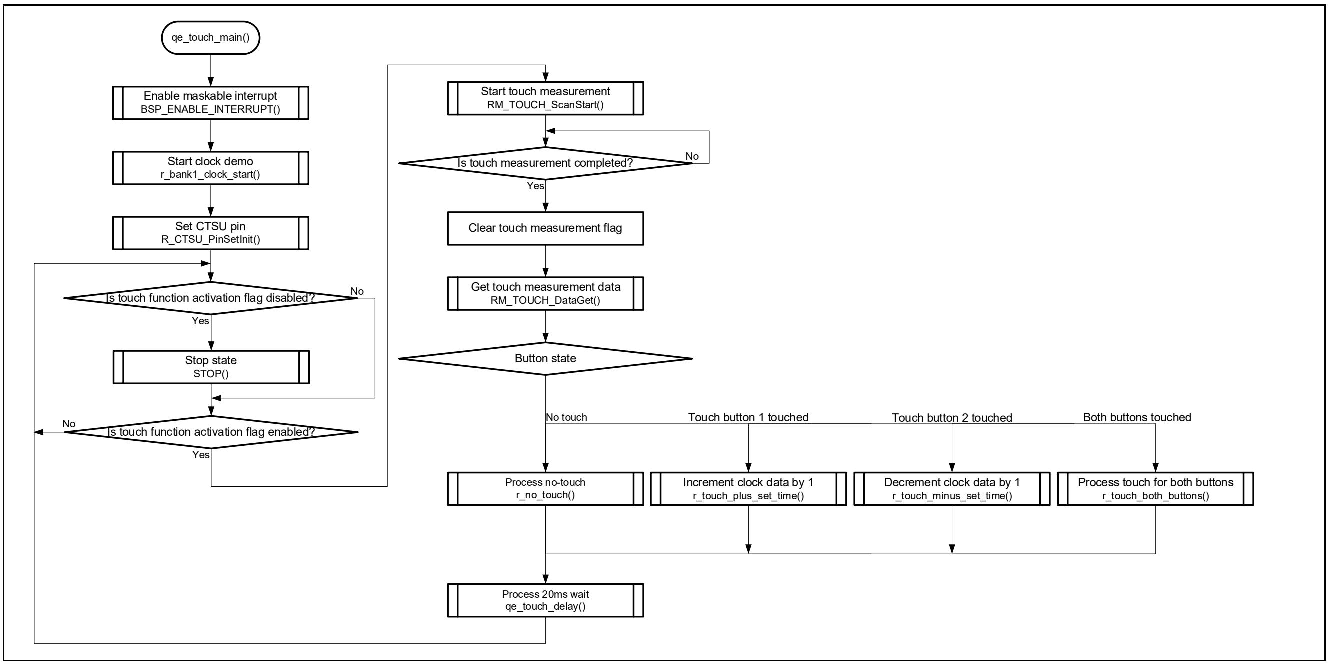
Figure 5.19 main Function Flowchart



5.9.2 qe_touch_main Function

Figure 5.20 shows the qe_touch_main function flowchart.

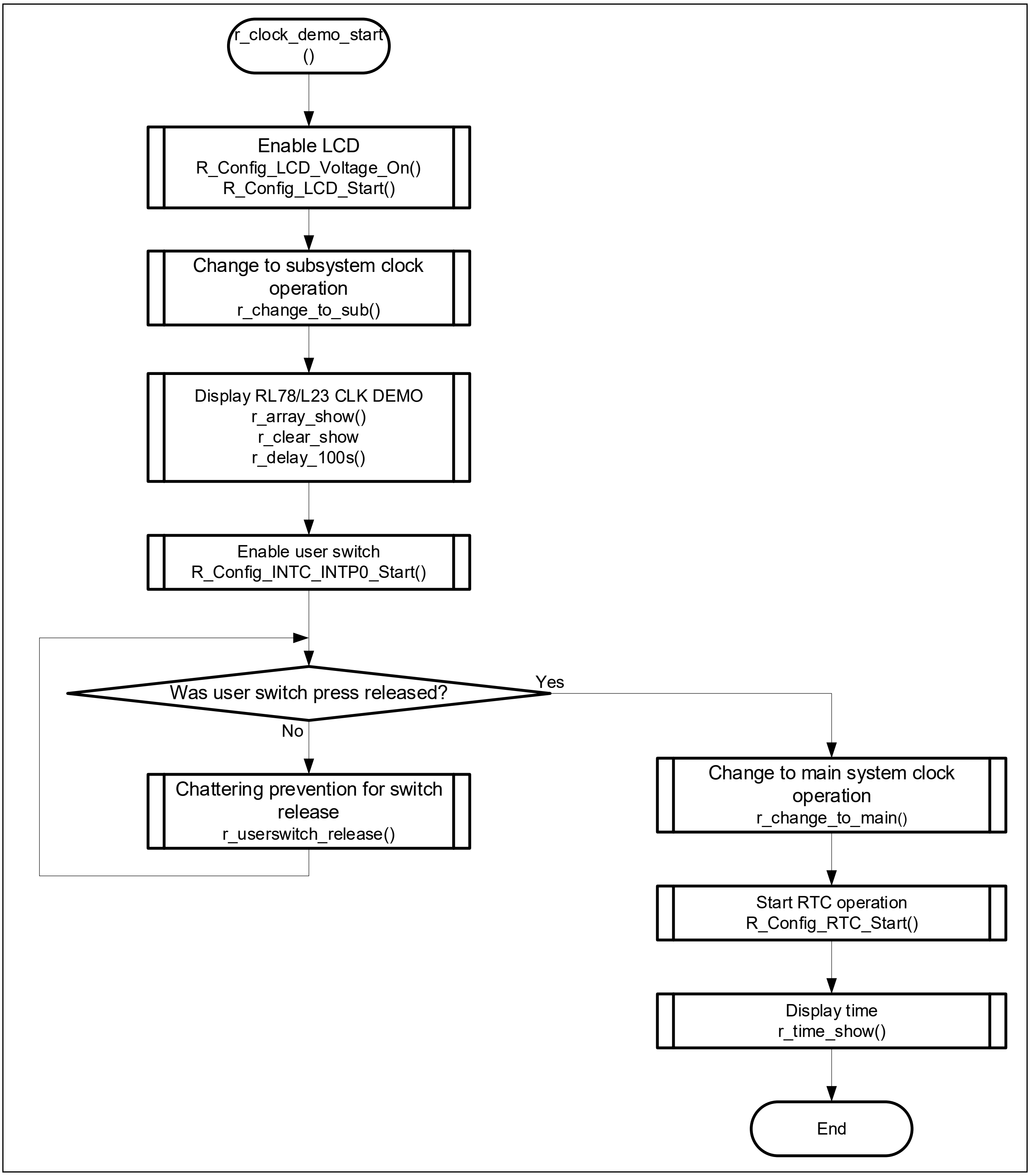
Figure 5.20 qe_touch_main Function Flowchart



5.9.3 r_clock_demo_start Function

Figure 5.21 shows the r_clock_demo_start function flowchart.

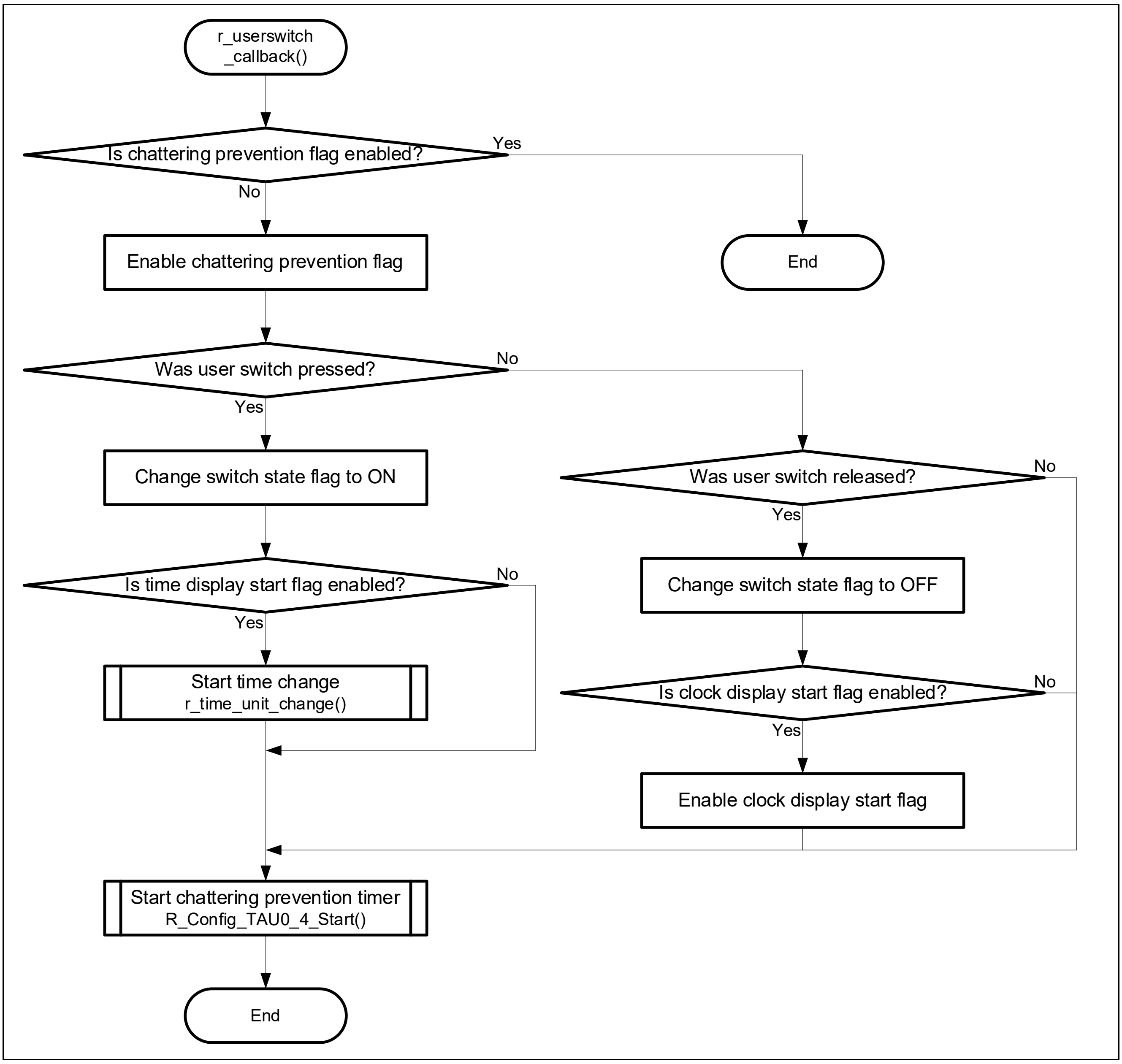
Figure 5.21 r_clock_demo_start Function Flowchart



5.9.4 r_userswitch_callback Function

Figure 5.22 shows the r_userswitch_callback function flowchart.

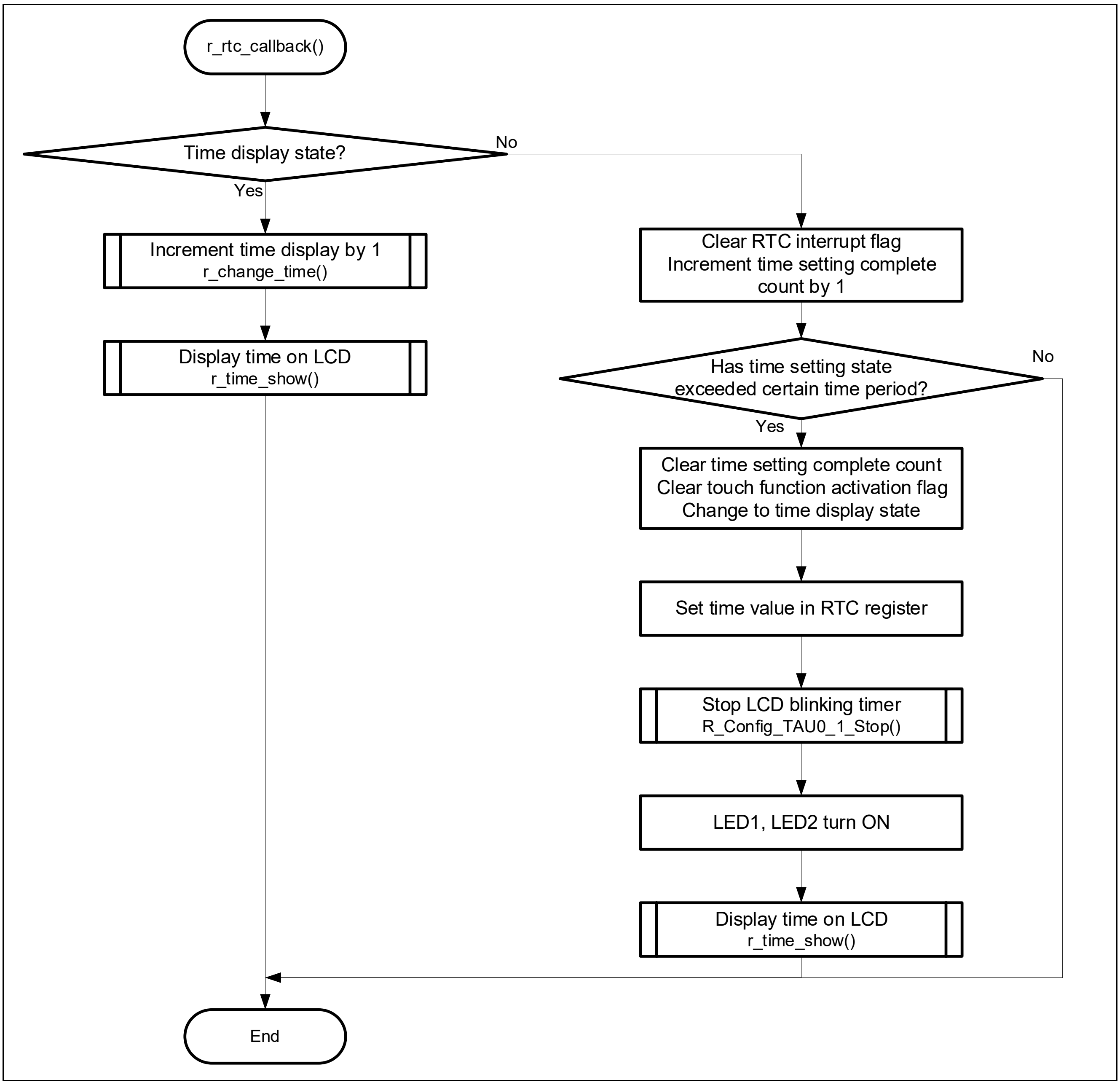
Figure 5.22 r_userswitch_callback Function Flowchart



5.9.5 r_rtc_callback Function

Figure 5.23 shows the r_rtc_callback function flowchart.

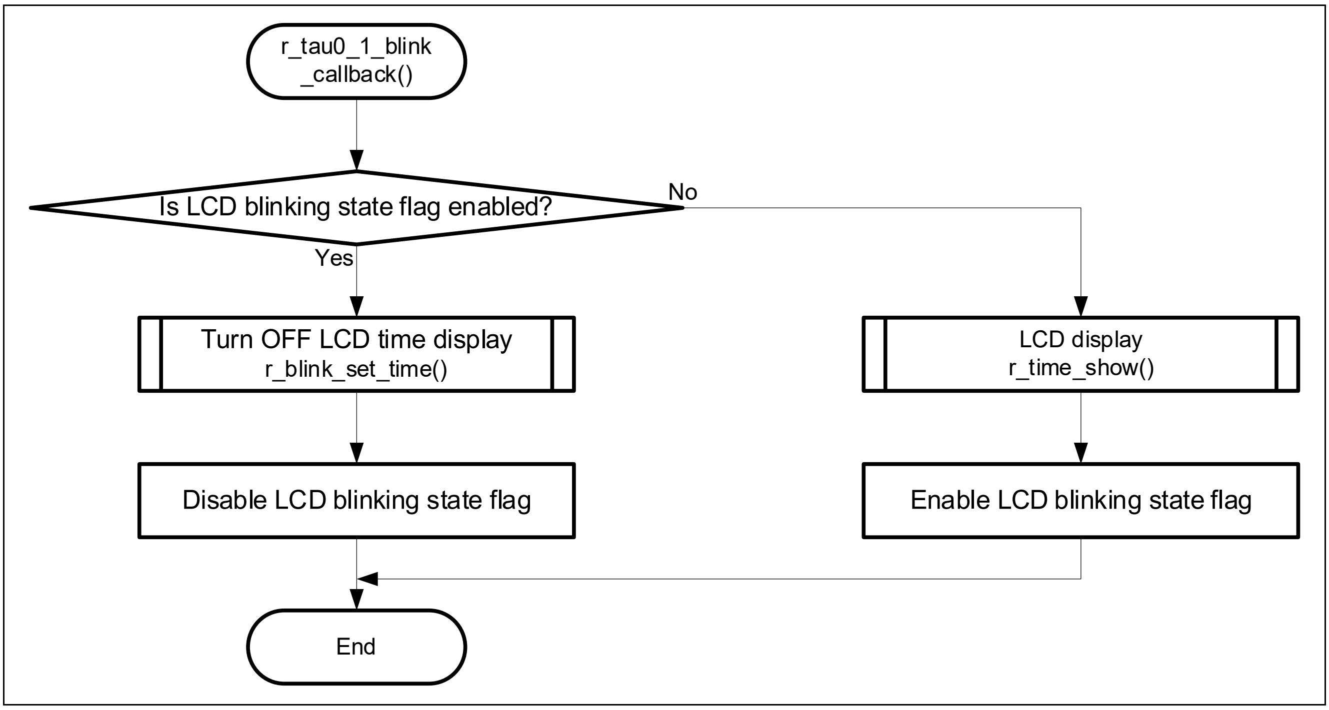
Figure 5.23 r_rtc_callback Function Flowchart



5.9.6 r_tau0_1_blink_callback Function

Figure 5.24 shows the r_tau0_1_blink_callback function flowchart.

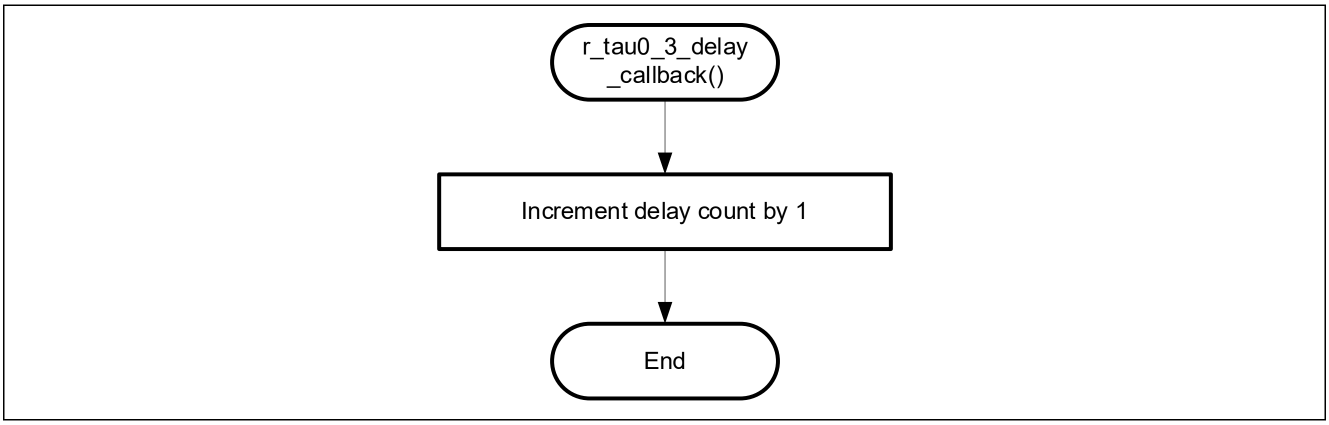
Figure 5.24 r_tau0_1_blink_callback Function Flowchart



5.9.7 r_tau0_3_delay_callback Function

Figure 5.25 shows the r_tau0_3_delay_callback function flowchart.

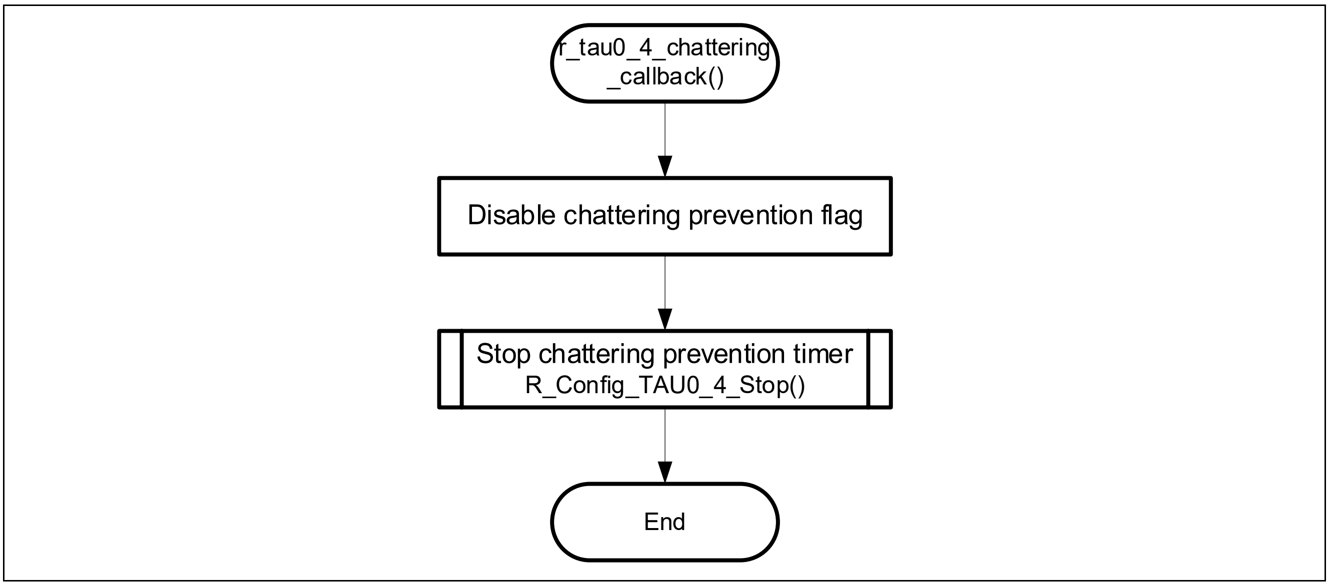
Figure 5.25 r_tau0_3_delay_callback Function Flowchart



5.9.8 r_tau0_4_chattering_callback Function

Figure 5.26 shows the r_tau0_4_chattering_callback function flowchart.

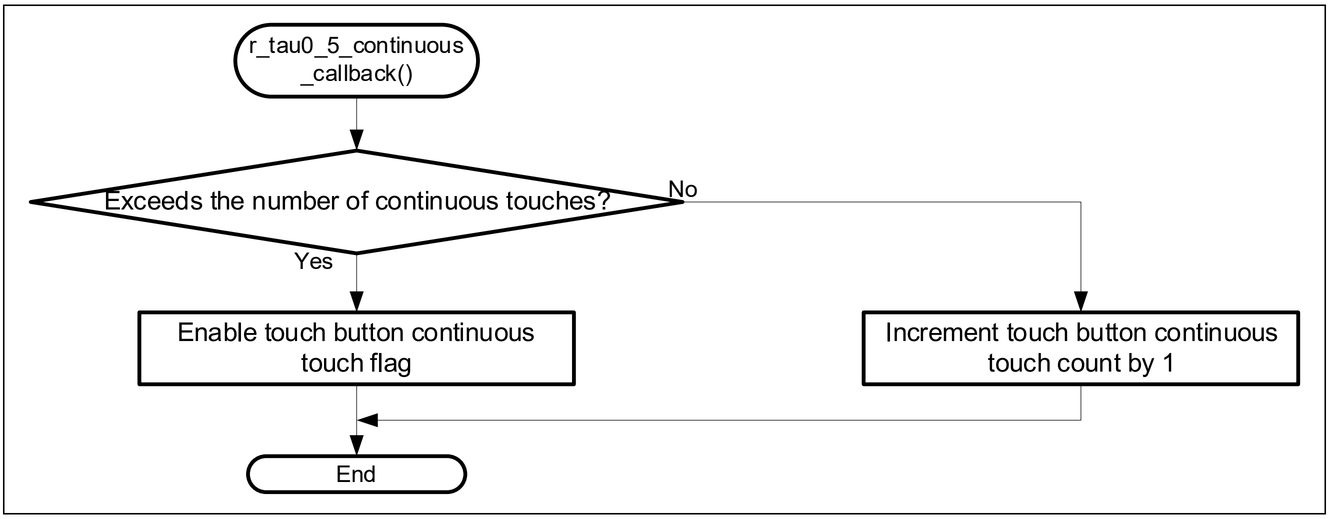
Figure 5.26 flowchart of r_tau0_4_chattering_callback function



5.9.9 r_tau0_5_continuous_callback Function

Figure 5.27 shows a flowchart of r_tau0_5_coutinuous_callback function.

Figure 5.27 flowchart of r_tau0_5_continuous_callback function



6. Importing the Project

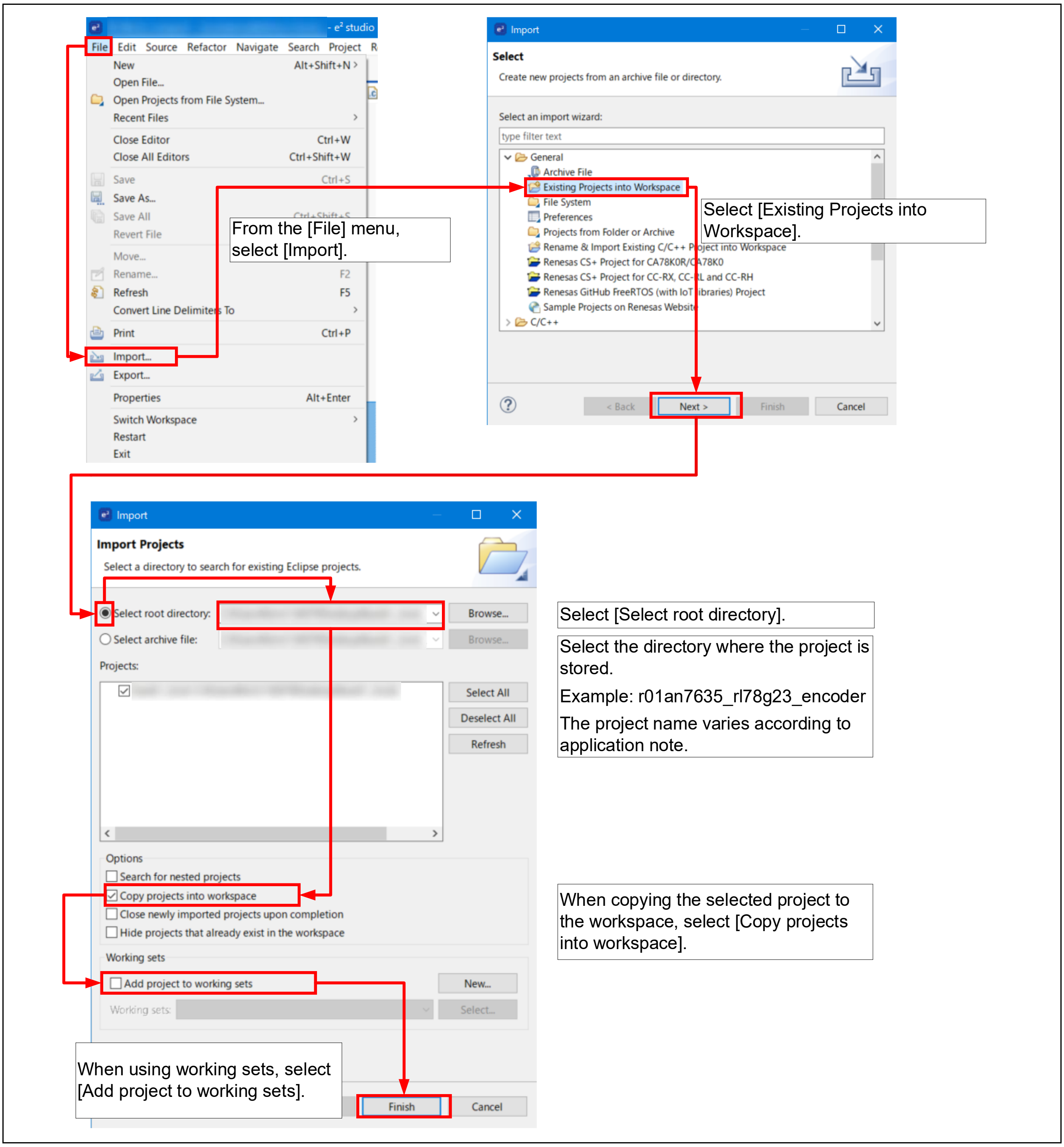
6.1 Importing with e² studio

When using the sample code with e² studio, use the following procedure to import the project in e² studio.

Note that spaces and symbols (especially $, #, and %) cannot be included in e² studio project folder names or path to these folders.

(Dialog boxes shown in the figure may differ depending on the e² studio version used.)

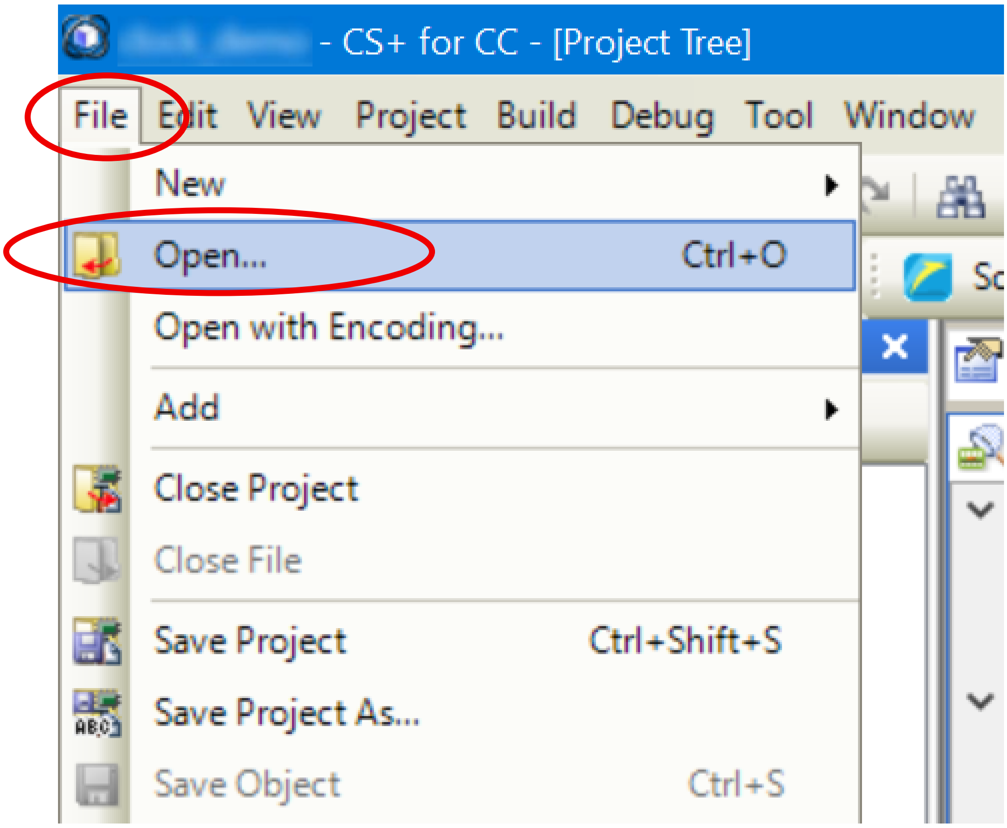
Figure 6.1 Importing the Project into e² studio



6.2 Importing with CS+

When using the sample code with CS+, open the mtpj file of the project by selecting [File] – [Open…] menu. (The image shown in the figure may differ depending on the CS+ version used.)

Figure 6.2 How to Import a Project to CS+

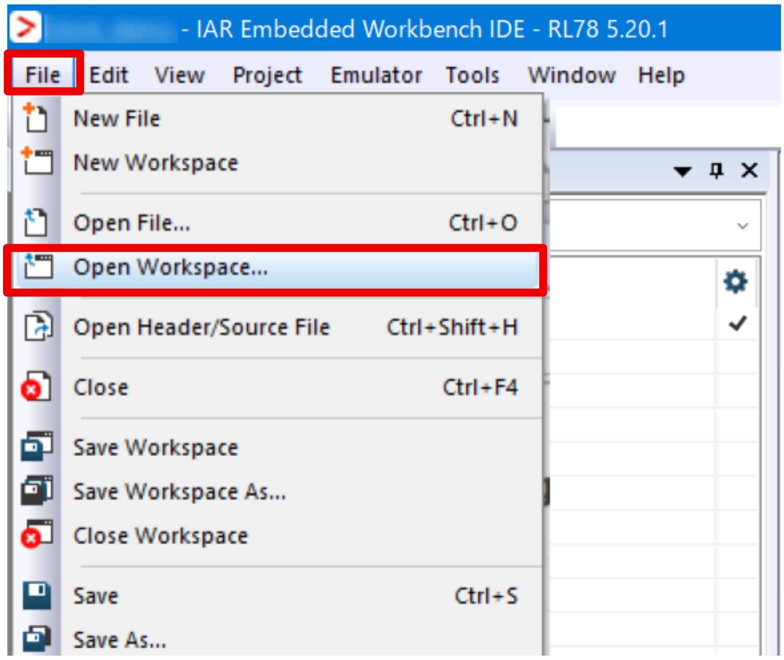


6.3 Importing with IAR

When using the sample code with IAR, open the eww file of the project by selecting [File] – [Open Workspace…] menu.

(The image shown in the figure may differ depending on the IAR version used.)

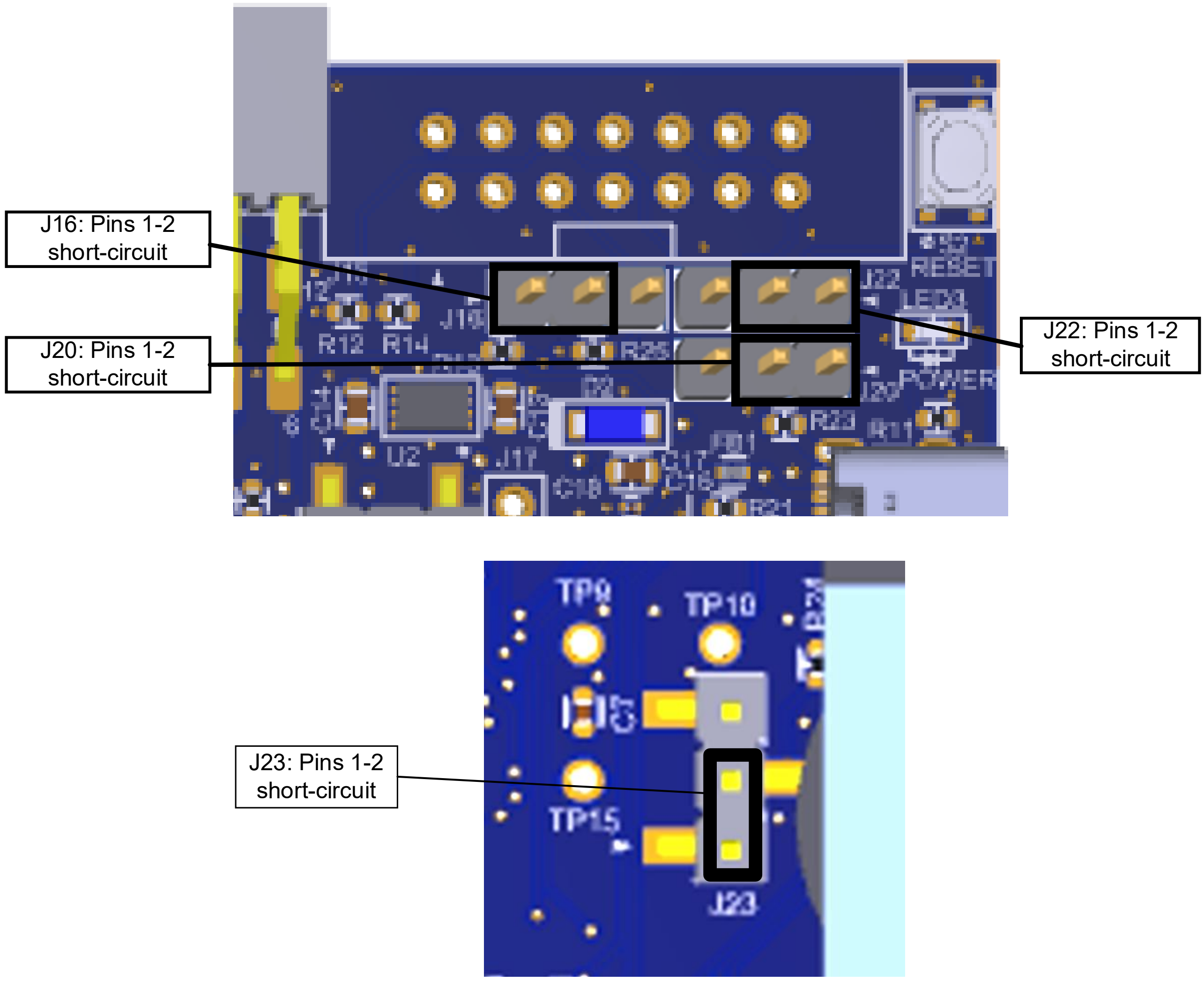
Figure 6.3 How to Import a Project to IAR



7. Setting the Debug Tool

In the sample code, the USB-C port of RL78/L23 Fast Prototyping Board (RTK7RLL230S000001BJ) is used to perform debugging via the COM Port. When debugging via the COM port, make sure that jumper pins of RL78/L23 are configured as shown in Figure 7.1. For details, refer to 5.15 USB-to-Serial Converter and 5.21 Emulator Connector in RL78/L23 Fast Prototyping Board User's Manual (R20UT5544).

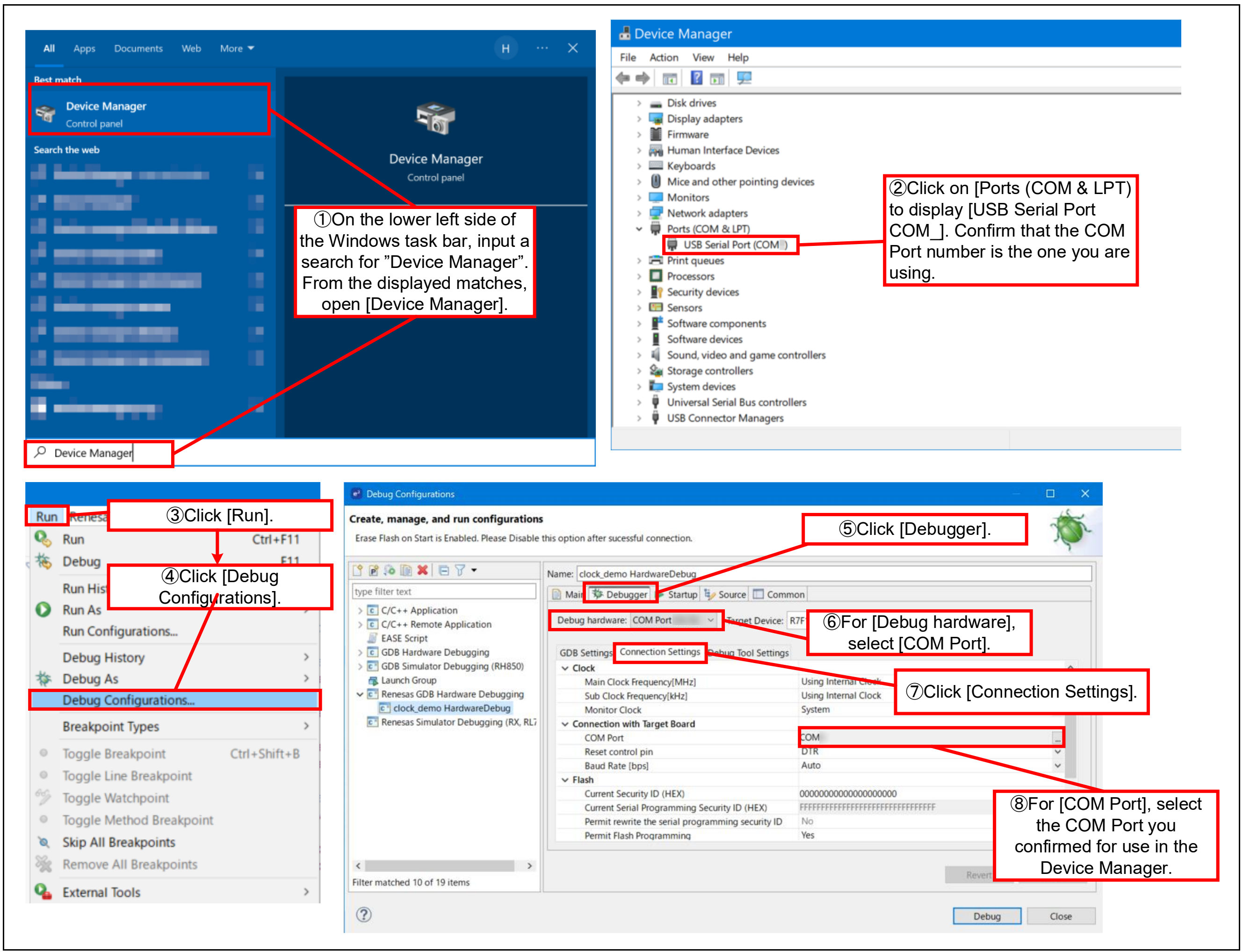
Figure 7.1 COM Port Setting When Using Debug



7.1 COM port Setting with e² studio

Connect the PC and RL78/L23 with a USB-C cable, then select a COM Port for debugging as shown in Figure 7.2.

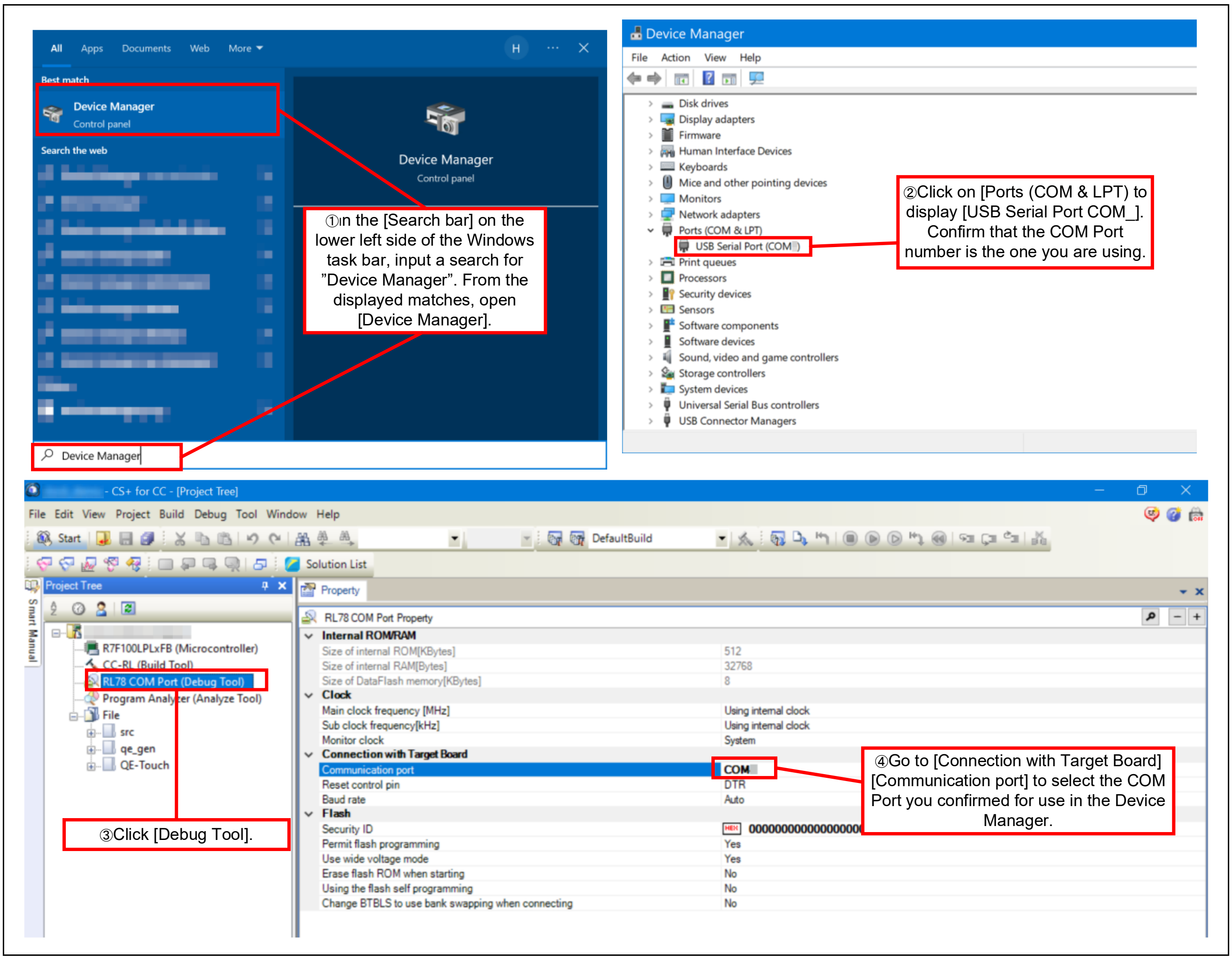
Figure 7.2 COM Port Setting with e² studio



7.2 COM Port Setting with CS+

Connect the PC and RL78/L23 with a USB-C cable, then select a COM Port for debugging as shown in Figure 7.3.

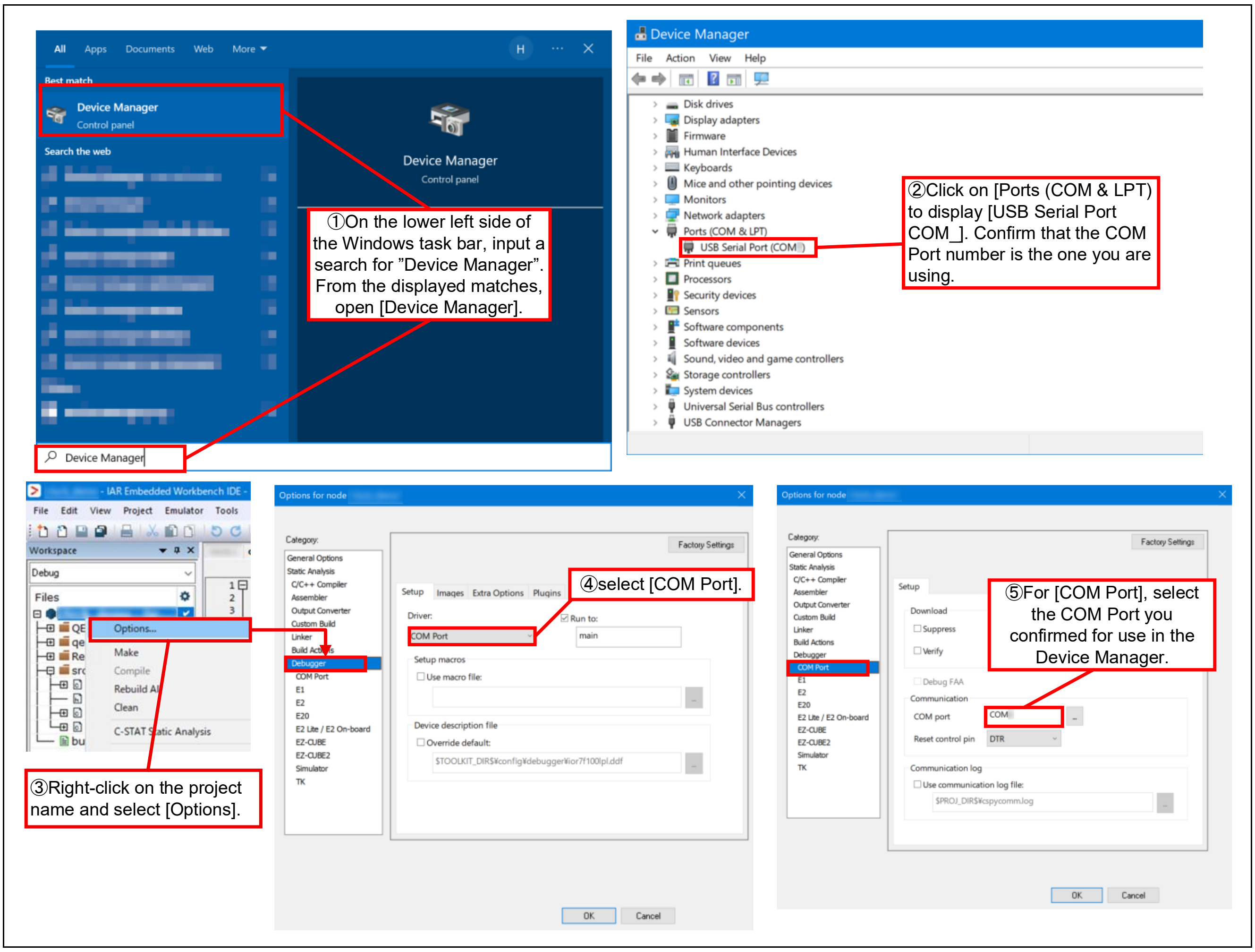
Figure 7.3 COM Port Setting with CS+



7.3 COM Port Setting with IAR

Connect the PC and RL78/L23 with a USB-C cable, then select a COM Port for debugging as shown in Figure 7.4.

Figure 7.4 COM Port Setting with IAR



8. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

9. Reference Documents

RL78/L23 User's Manual: Hardware (R01UH1082)

RL78 Family User's Manual: Software (R01US0015)

RL78/L23 Fast Prototyping Board User's Manual (R20UT5544)

Capacitive Sensor Microcontrollers CTSU Capacitive Touch Introduction Guide (R30AN0424)

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

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

Technical Update/Technical News

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

REVISION HISTORY

Rev.	Date	Description	
		Page	Summary
1.00	Aug.18.25	-	First edition issued
1.10	Oct.26.25	6	2. Operation Confirmation Conditions Added description of the IAR environment.
		26-27	Split Table 5-1 to create Table 5-2. Added the note about File Composition of IAR version.
		50	Modified flowchart in Figure 5-20.
		58	6.3 Importing with IAR Added description.
		62	7.3 COM Port Setting with IAR Added description.

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/.