# RL78/L23

## QE for Segment LCD Application Development Guide


### Summary

QE for Segment LCD[RL78] is a solution toolkit that operates on the e$^2$ studio integrated development environment. This is an assistance tool for segment LCD solution development, providing full-flow support for configuring LCD pin settings, verifying pin settings and wiring, designing display patterns, generating sample application programs, and performing software debugging. Using this tool simplifies and accelerates the development of segment LCD solutions, and it more efficient.


### Target Devices

RL78/L23

### Assumed Operating Environment

The procedure described in this application note assume operation in one of the following environments.

- RL78/L23 Fast Prototyping Board

When applying this application note to a different device or board, adjust the settings as appropriate and thoroughly evaluate the results.

## Contents

# 1. Overview

## 1.1 System Overview

QE for Segment LCD[RL78] is a solution toolkit that operates on the e$^2$ studio integrated development environment. QE for Segment LCD [RL78] can be used in conjunction with the Smart Configurator, which simplifies the integration of the Renesas drivers. This combination of tools simplifies the process from segment LCD pin configuration to software debugging.

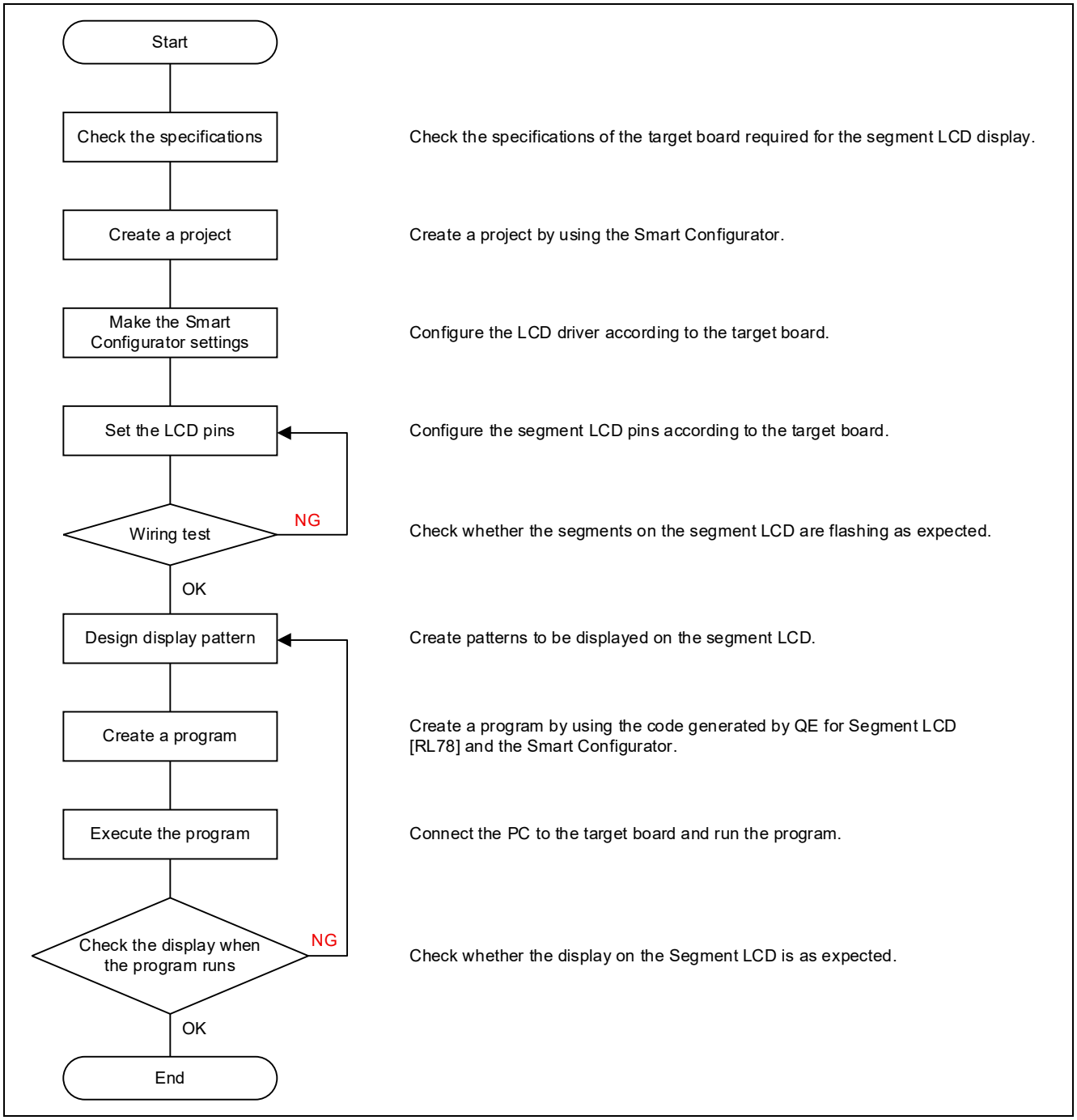
The main functions are listed below:

- Create LCD components and configure COM and SEG pins for each segment
- Verify LCD wiring by blinking the target segment.
- Prepare design patterns and pattern API to turn patterns on and off.
- Generate a sample program that displays design patterns for each component sequentially.
- Perform software debugging by designing stop condition and setting checkpoints.

## 1.2 System Development Flow Using QE for Segment LCD

Figure 1-1 shows the system development flow using QE for Segment LCD[RL78].

**Figure 1-1 System Development Using QE for Segment LCD[RL78]**



## 2. Operating Environment

The procedures described in this application note assume operations on the “RL78/L23 Fast Prototyping Board”.

Table 2-1 shows the development environment used in this application note.

**Table 2-1 Operating Environment**

| Item | Contents |
|---|---|
| PC OS | Windows 11(64-bit version) |
| Integrated development environment (IDE) | Renesas e$^2$ studio 2025-12 from Renesas Electronics Corp. |
| Toolchain | Renesas CC-RL v1.16.00 from Renesas Electronics Crop. |
| QE for Segment LCD | V1.1.0 |

## 3. Related materials

Table 3-1 shows the documents related to this application note, and Table 3-2 shows the related videos.

**Table 3-1 Documents Related**

| Title | Document No. |
|---|---|
| RL78/L23 LCD display (Clock demo) Application Note | R01AN7834 |
| QE for Segment LCD[RL78] V1.1.0 Release Note | R20UT5779 |

**Table 3-2 Video Related**

| Title | Overview |
|---|---|
| e² studio Quick Start Guide (2/3) - Project Generation for RL78 \| Renesas | Explanation of how to create a project for RL78. |
| Getting Started with QE for Segment LCD \| Renesas | Introduction to the usage of the QE for Segment LCD tool and the configuration and debugging of segment LCD systems. |

# 4. Procedures for Execution

## 4.1 Installation, Update, and Uninstallation of QE for Segment LCD

### 4.1.1 Installing This Product

Use either of the following procedures to install this product.

#### 4.1.1.1 Install from the "Renesas software installer" menu of e² studio

1. Start e² studio.
2. Select the “Renesas Views” - “Renesas Software Installer” menu of e² studio to open the “Renesas Software Installer” dialog box.
3. Select the “Renesas QE” and click the “Next >” button.
4. Select the “QE for Segment LCD (V1.1.0)” check box and click the “Finish” button.
5. Check that the “Renesas QE for Segment LCD[RL78]” check box is selected in the “Install” dialog box and click the “Next >” button.
6. Check that the “Renesas QE for Segment LCD[RL78]” check box is selected as the target of installation and click the “Next>” button.
7. After confirming the license agreements, if you agree to the license, select the “I accept the terms of the license agreements” radio button, and click the “Finish” button.
8. When the dialog box for selecting a trusted certificate appears, check the displayed certificate, and then click on the “Trust Selected” button to continue installation.
9. Restart the e² studio by following the instructions on the screen.
10. Start this product from the “Renesas Views” - “Renesas QE” menu of e² studio.

#### 4.1.1.2 Install using QE (zip file) downloaded from the Renesas website

1. Extract the downloaded zip file.
2. Start the e² studio.
3. Click on “Help”, and then click on the “Install New Software...” menu item to open the “Install” dialog box.
4. Click on the “Add...” button to open the “Add Repository” dialog box.
5. Click on the “Archive...” button, select the installation file (zip file under the QE-Segment LCD folder) in the opened file selection dialog box, and then click on the “Open” button.
6. Click on the “Add” button in the “Add Repository” dialog box.
7. Expand the “Renesas QE” item shown in the “Install” dialog box, select the “Renesas QE for Segment LCD[RL78]” check box, and then click on the “Next>” button. *If you check off the “Contact all update sites during install to find required software” checkbox, you can shorten the installation time.
8. Confirm that the installation target is “Renesas QE for Segment LCD[RL78]”, and then click on the “Next>” button.
9. After checking the license, select the “I accept the terms of the license agreements” radio button if you agree it, and then click on the “Finish” button.
10. When the dialog box for selecting a trusted certificate appears, check the displayed certificate, and then click on the “Trust Selected” button to continue the installation.
11. Restart the e² studio by following the instructions on the screen.
12. Start this product from the “Renesas Views” - “Renesas QE” menu of the e² studio. For details about how to use this product, see the “Help” menu of e² studio.

### 4.1.2 Updating This Product

If you have already used this product, you can update it in the same way as the procedure for installation.

### 4.1.3 Uninstalling This Product

Follow the procedure below to uninstall this product.

1. Start the e$^2$ studio.
2. Select “Help -> About e$^2$ studio” to open the “About e$^2$ studio” dialog box.
3. Click the “Installation Details” button to open the “e$^2$ studio Installation Details” dialog box.
4. Select “Renesas QE for Segment LCD[RL78]” displayed on the “Installed Software” tabbed page and click on the “Uninstall...” button to open the “Uninstall” dialog box.
5. Check the displayed information and click on the “Finish” button.
6. Restart the e$^2$ studio by following the instructions on the screen.

## 4.2 Setting LCD According to the Workflow of QE for Segment LCD

### 4.2.1 Preparation

Make the settings for “1. Preparation” in the workflow view of QE for Segment LCD[RL78].

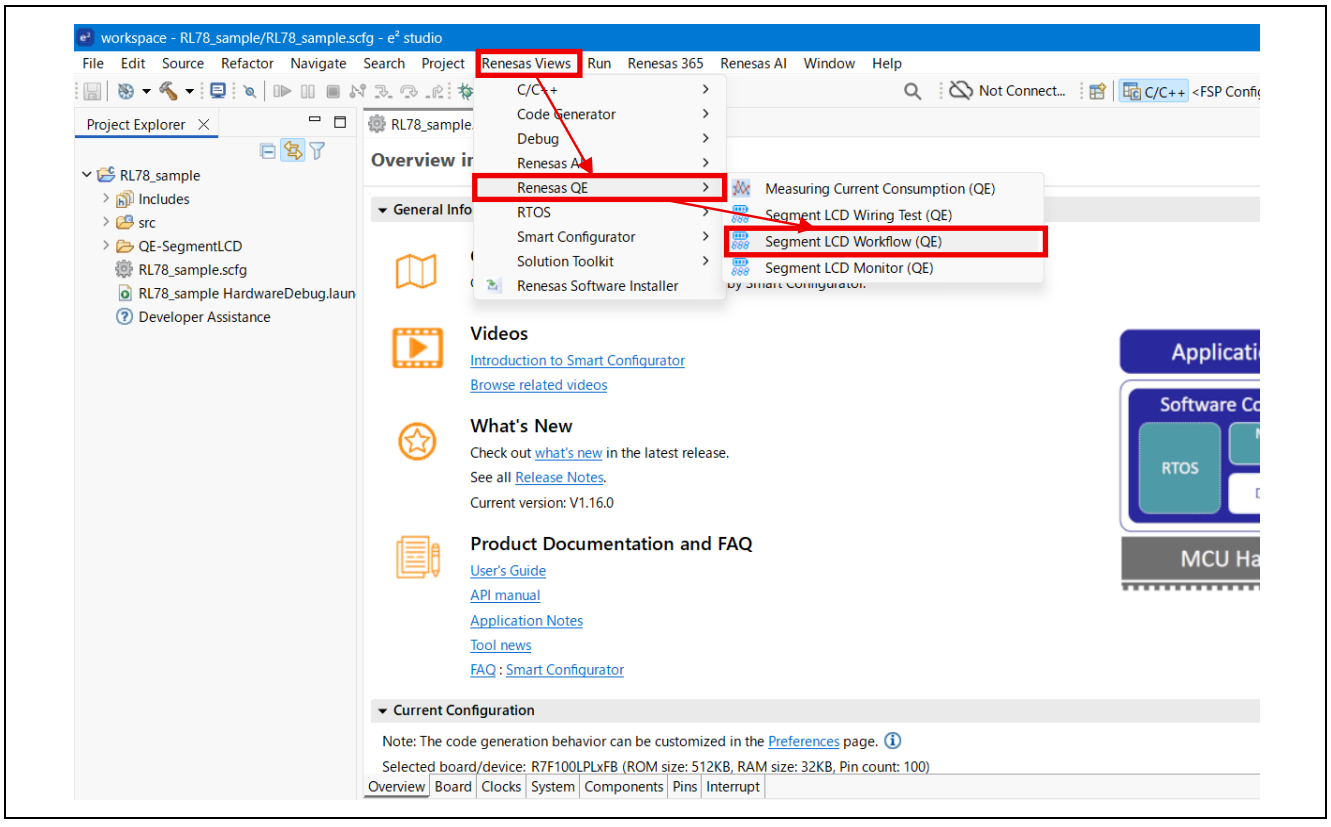
To use QE for Segment LCD, you must select a new or existing project.
For information on how to create a new project, refer to "e² studio Quick Start Guide (2/3) – Project Generation for RL78".

Note that some procedures differ from those shown in the video. Select the following items for this application note.

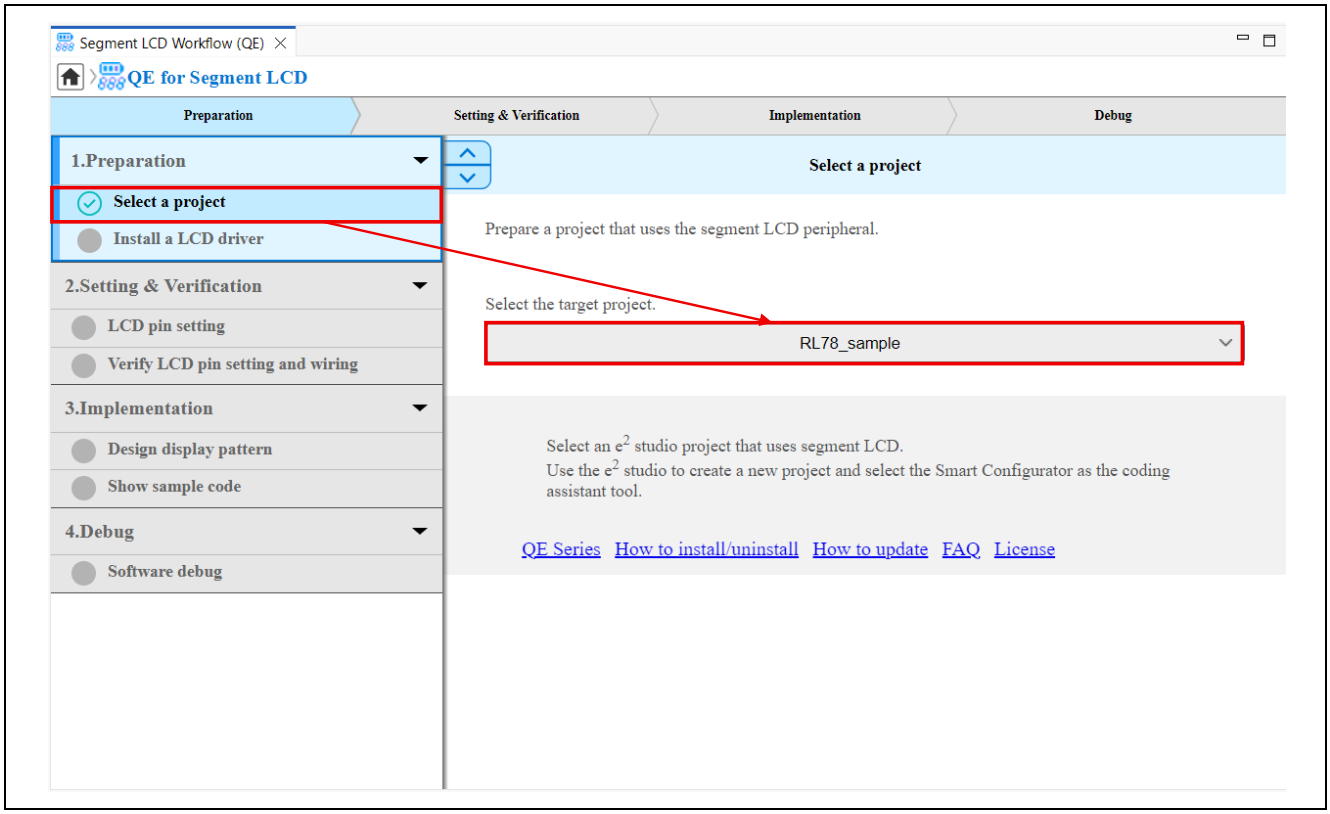
- The “Select toolchain, device & debug settings” dialog
  — Target Board: RL78/L23_Fast Prototyping Board

- The “Choose a project template” dialog
  — Project template selection: Minimal

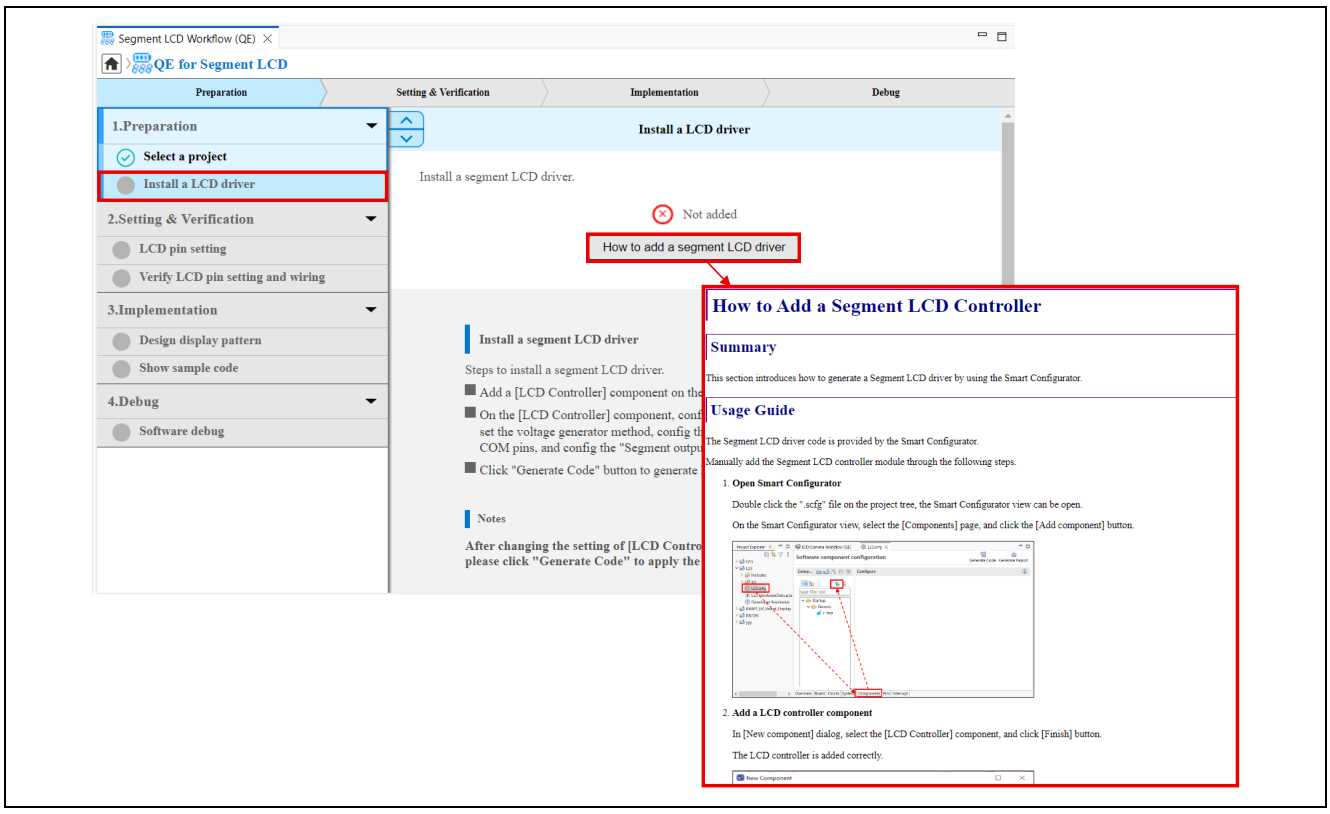
1. Launch the “Segment LCD Workflow”.



2. In “Select a project”, select the project.

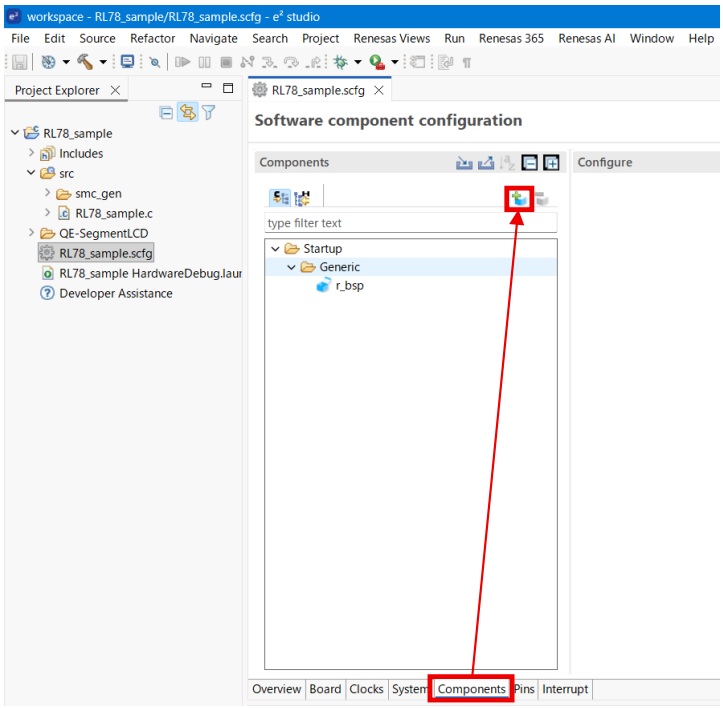



3. Check how to add the driver in “How to add a Segment LCD driver”.

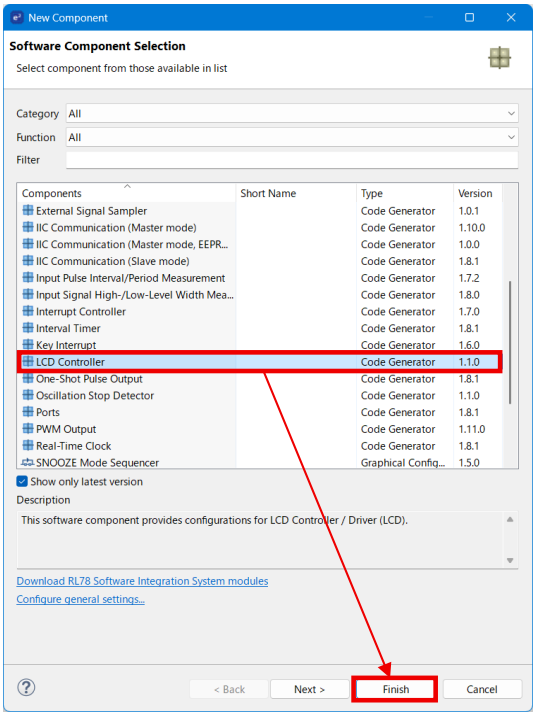


4. Double-click the ".scfg" file in the Project Explorer view to open the Smart Configurator.

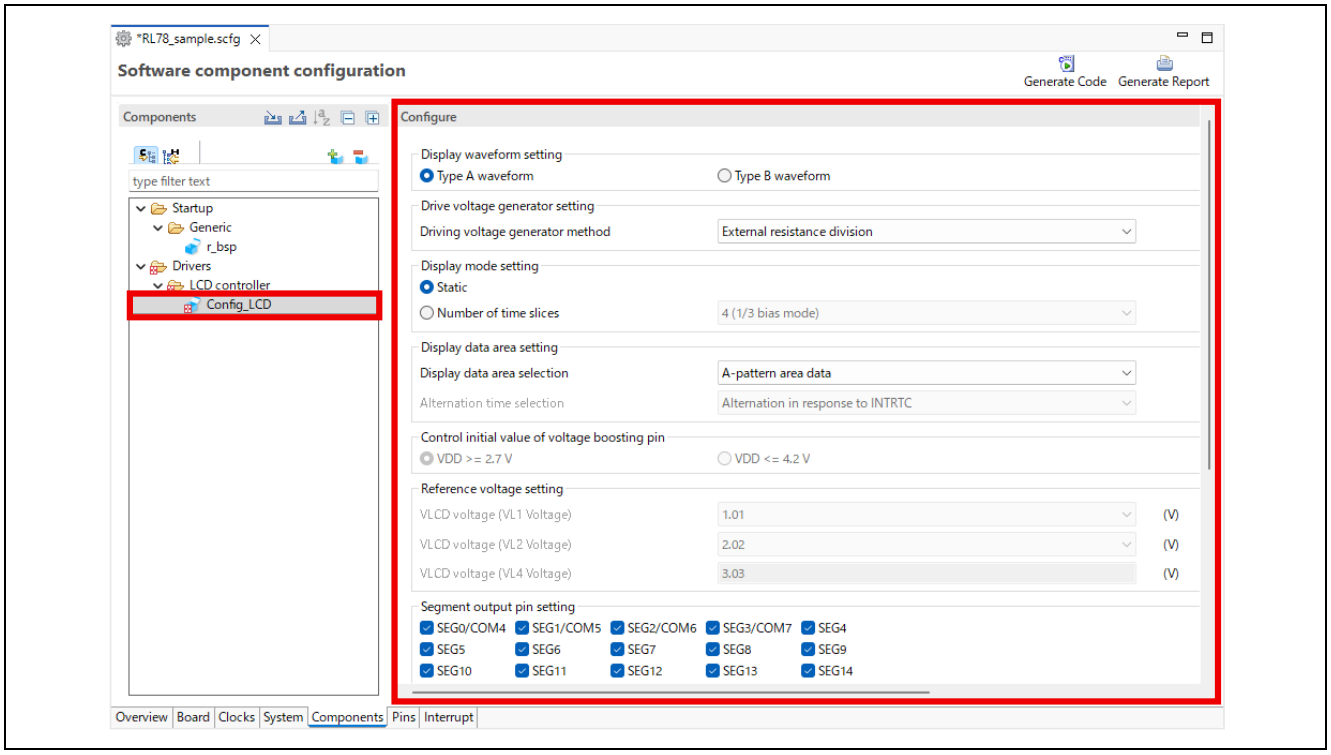
5. Select the "Components" tab and click the "Add component" button.



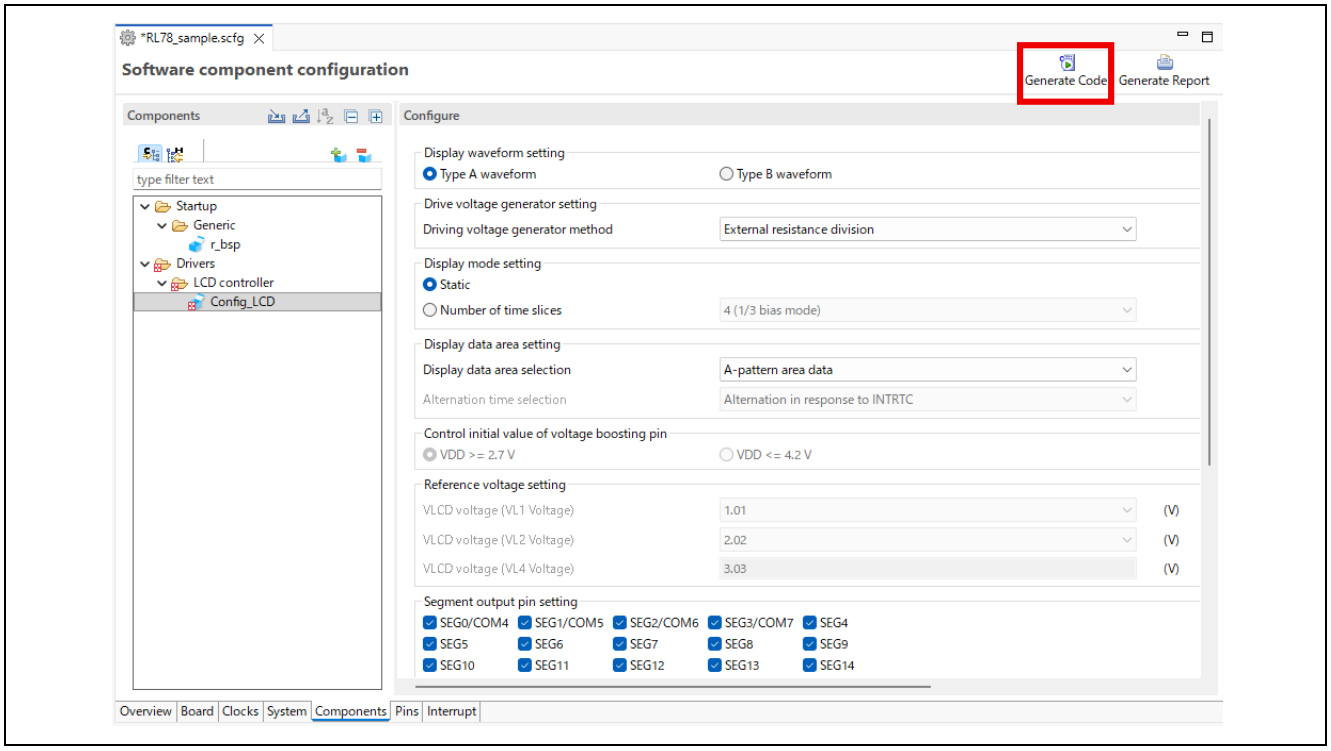

6. In the "New Component" dialog, select the "LCD Controller" component, and click the "Finish" button.



7. Open “Config_LCD” in the component tree, adjust the “Driving voltage generator method” under “Driving voltage generation setting”, as well as “Display mode setting” and “Segment output pin setting”.
   For the settings on the RL78/L23 Fast Prototyping Board, refer to Figure 6-1 in “6.1 Smart Configurator Settings”.



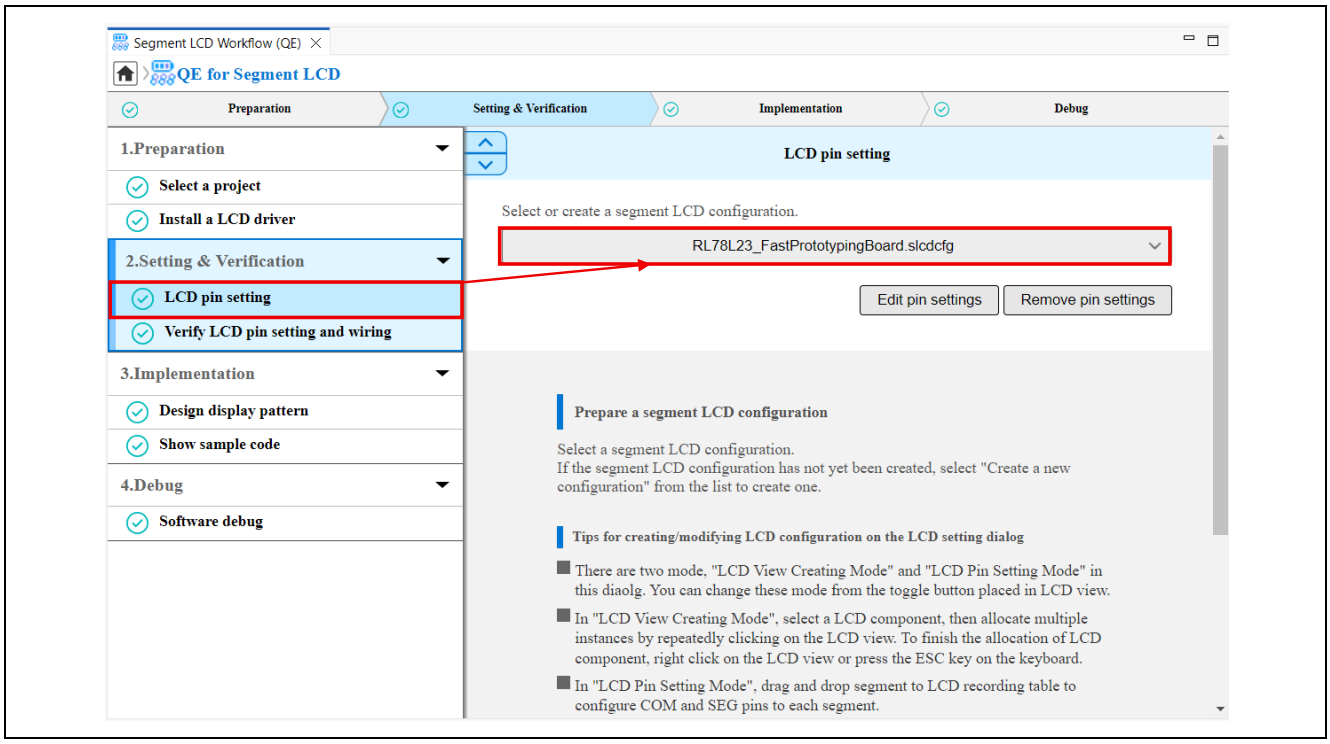

8. Click the “Generate Code” button in the Smart Configurator to generate the LCD driver source code.



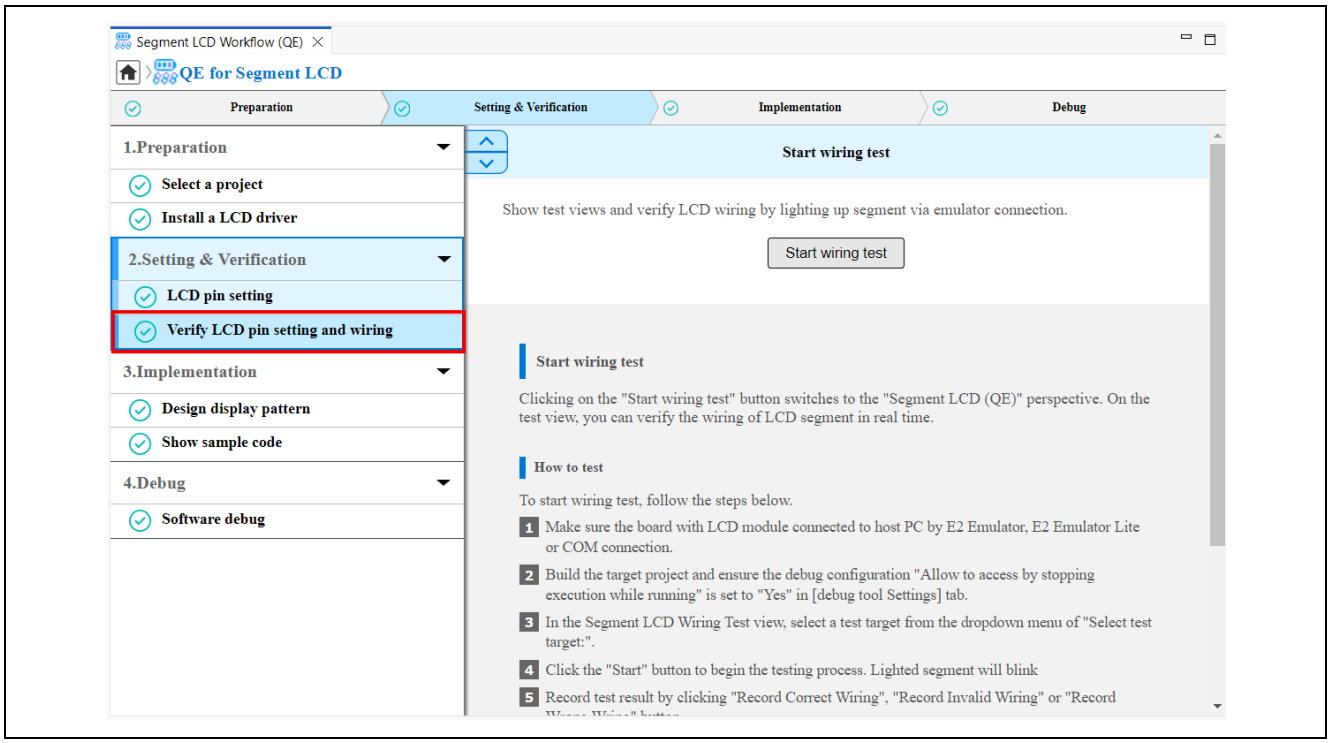
### 4.2.2 Setting & Verification

Make the settings for “2. Setting & Verification” in the workflow view of QE for Segment LCD[RL78].

1. In “LCD pin setting”, select a segment LCD configuration.



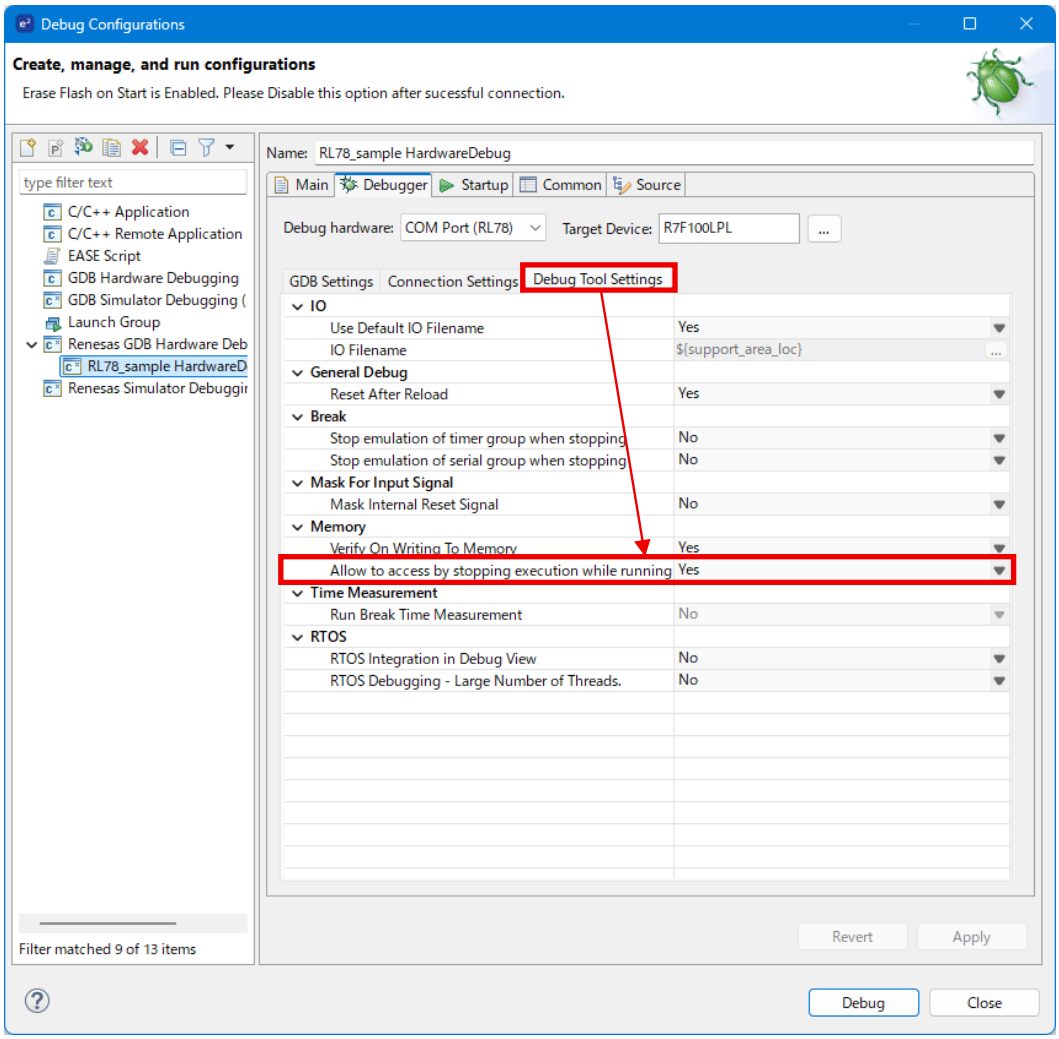

2. Open “Verify LCD pin setting and wiring”.



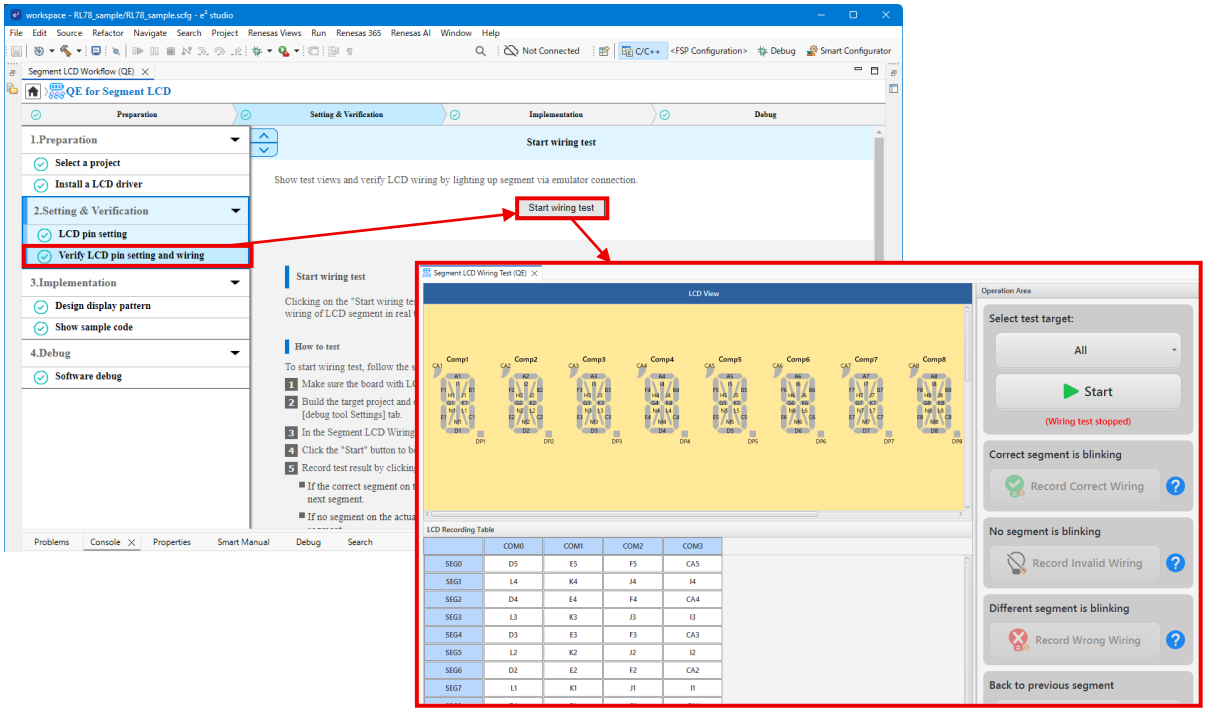
3. Make sure the board with the LCD module is connected to the host PC via an E2 Emulator, E2 Emulator Lite, or COM connection.

4. Build the target project and confirm that there are no errors.

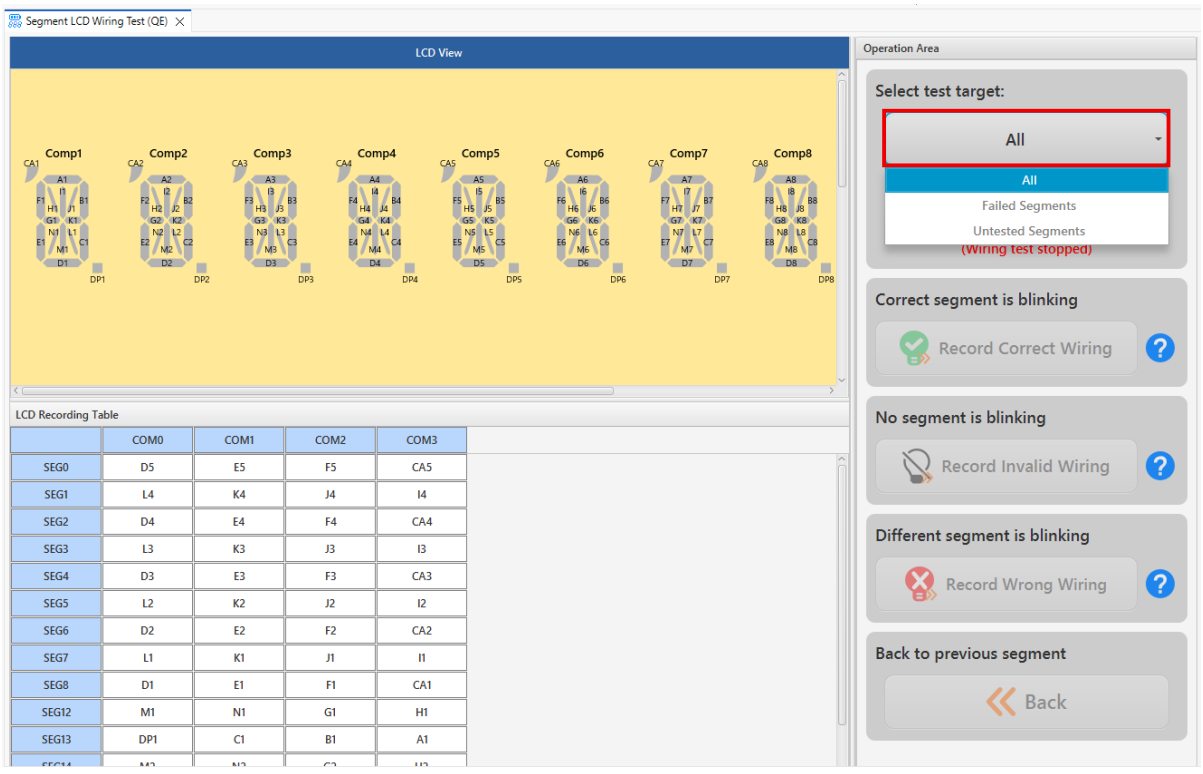
5. In the “Debug Tool Settings” tab of the “Debug Configurations”, set “Allow to access by stopping execution while running” to “Yes”.



6. Click the “Start wiring test” button to display the “Segment LCD Wiring Test (QE)” view.

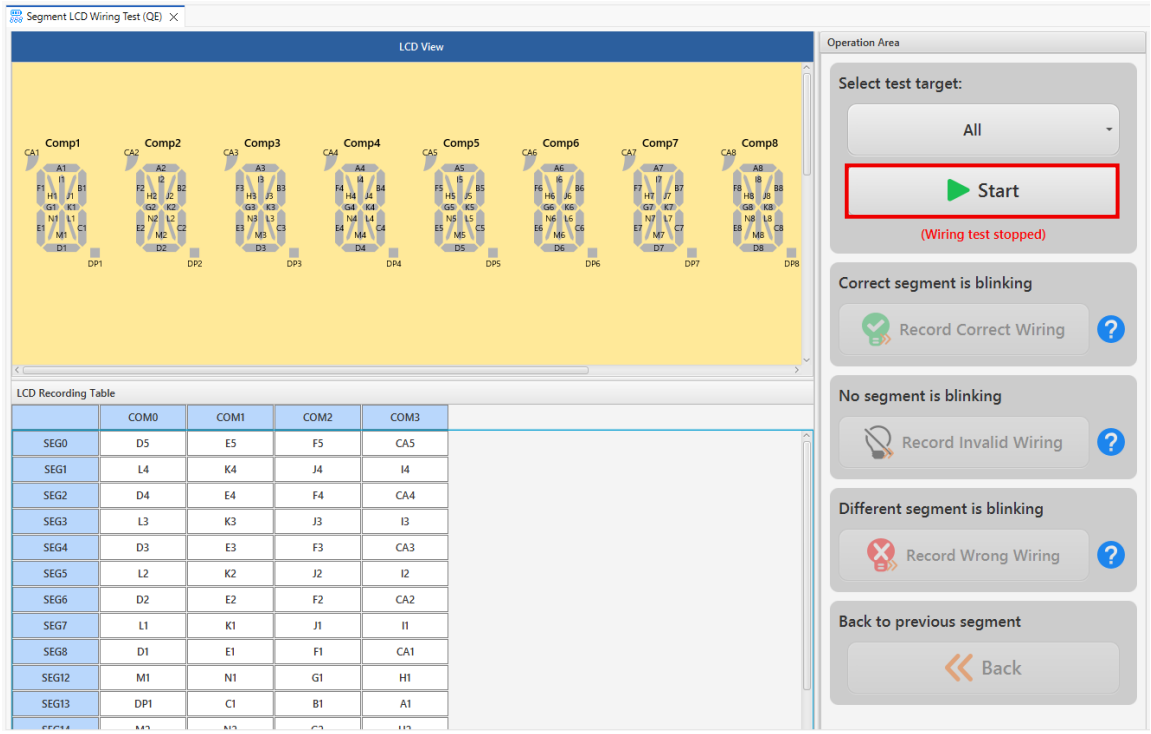



7. In the “Segment LCD Wiring Test (QE)” view, select a test target from the “Select test target:” dropdown menu.



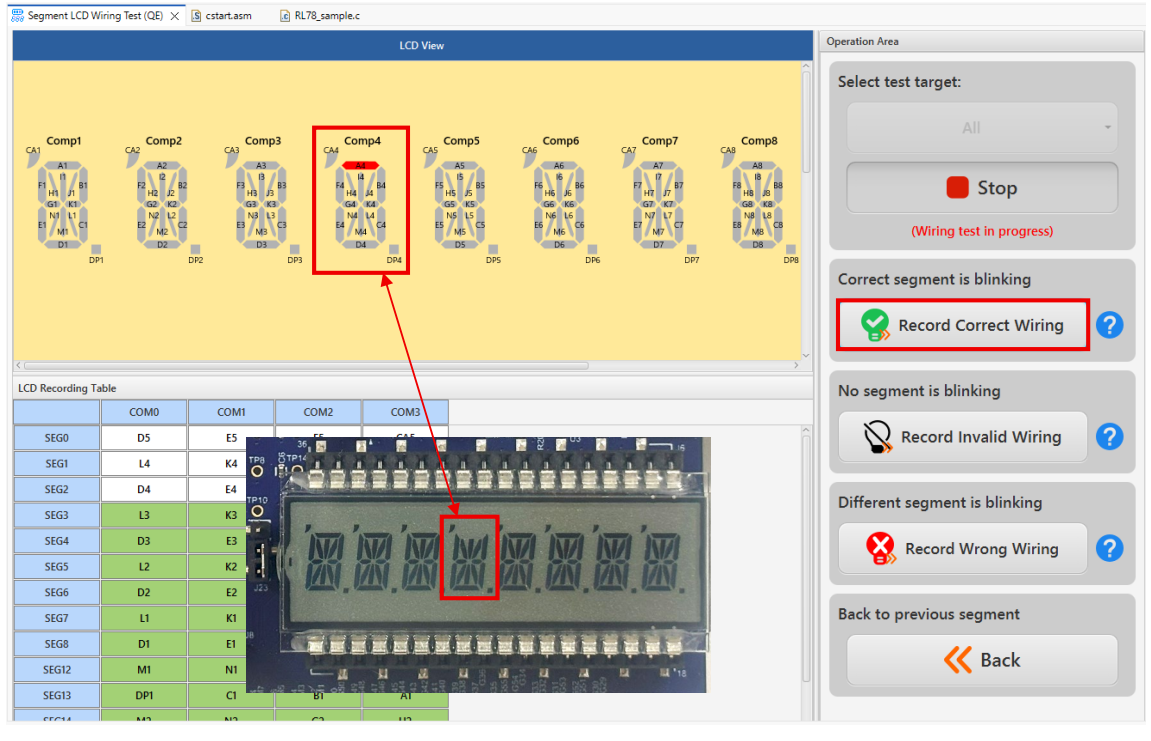

| | COM0 | COM1 | COM2 | COM3 |
|---|---|---|---|---|
| SEG0 | D5 | E5 | F5 | CA5 |
| SEG1 | L4 | K4 | J4 | I4 |
| SEG2 | D4 | E4 | F4 | CA4 |
| SEG3 | L3 | K3 | J3 | I3 |
| SEG4 | D3 | E3 | F3 | CA3 |
| SEG5 | L2 | K2 | J2 | I2 |
| SEG6 | D2 | E2 | F2 | CA2 |
| SEG7 | L1 | K1 | J1 | I1 |
| SEG8 | D1 | E1 | F1 | CA1 |
| SEG12 | M1 | N1 | G1 | H1 |
| SEG13 | DP1 | C1 | B1 | A1 |

8. Click the “Start” button to start the test.

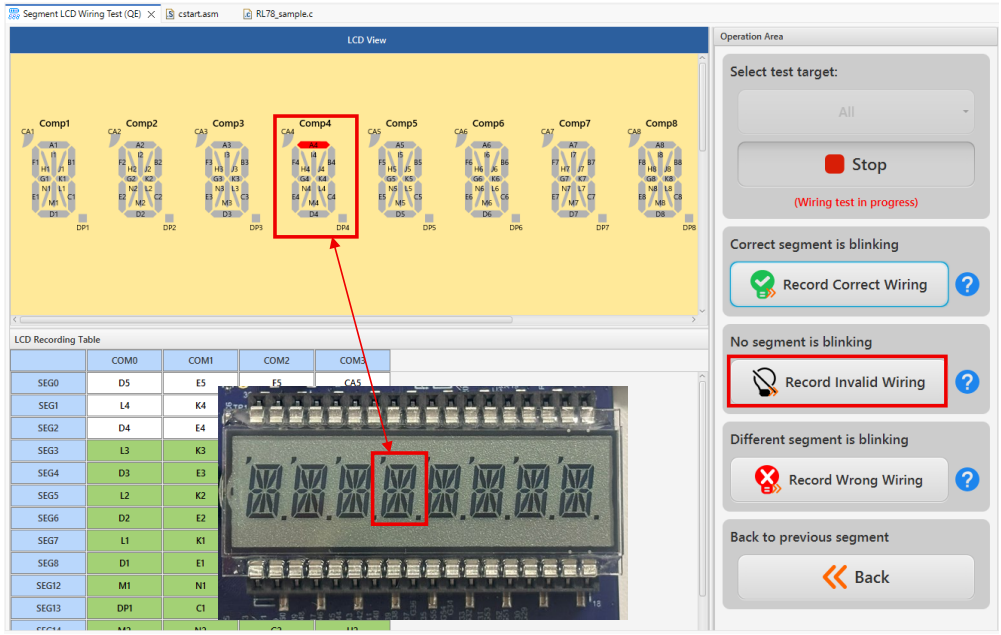



9. Check the display status on the actual LCD and click one of the following buttons to record the result.

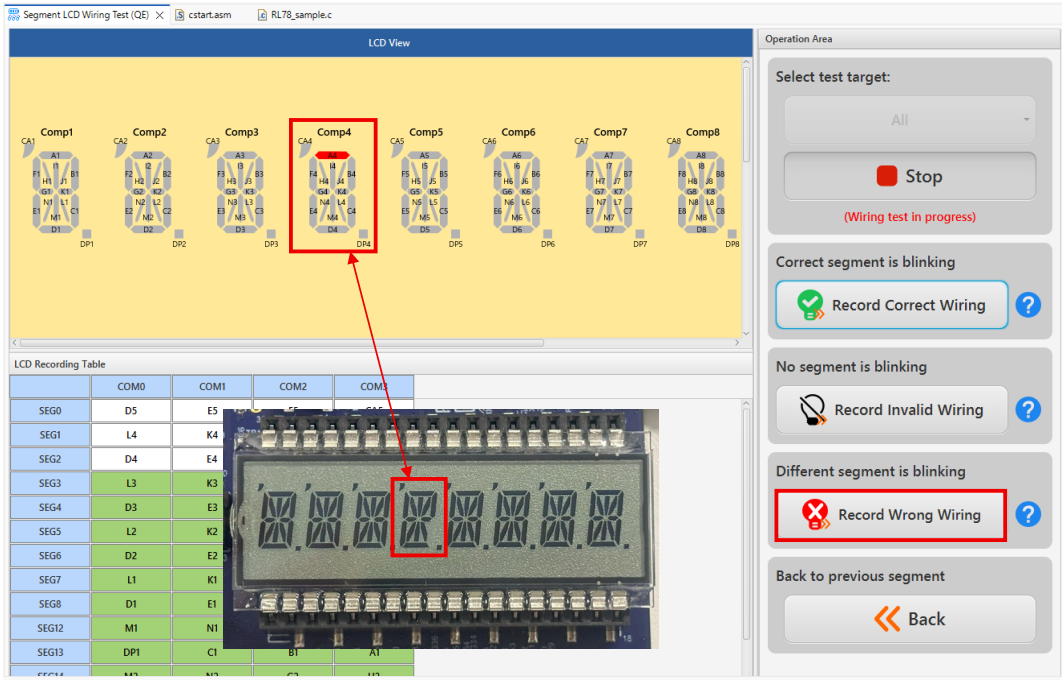
— If the correct segment on the actual LCD is blinking
Click the “Record Correct Wiring” button to record the status and proceed to the next test.



— If no segment on the actual LCD is blinking
Click the “Record Invalid Wiring” button to record the status.
If no segment is blinking, correct the settings in “Edit pin settings” of “LCD pin setting”, and repeat the wiring test until you can confirm that the wiring is correct.



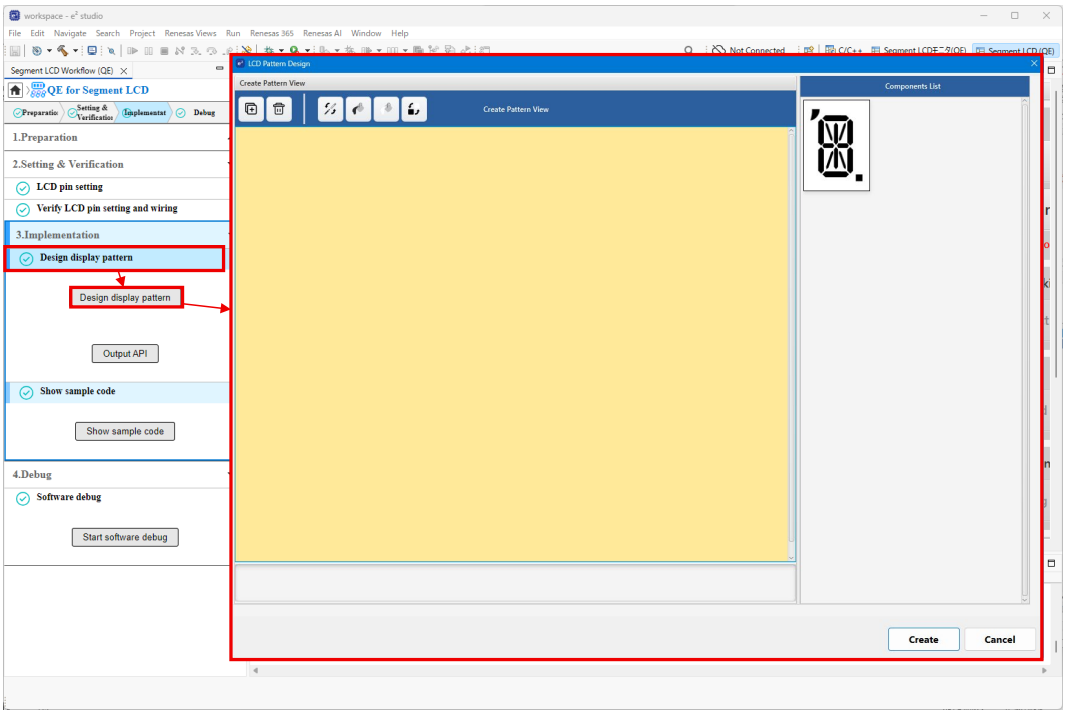

— If a segment other than the expected one on the actual LCD is blinking
Select the segment blinking on the actual LCD in the LCD View and click the “Record Wrong Wiring” button to record the status.
If the wiring is wrong, correct the settings in “Edit pin settings” of “LCD pin setting”, and repeat the wiring test until you can confirm that the wiring is correct.



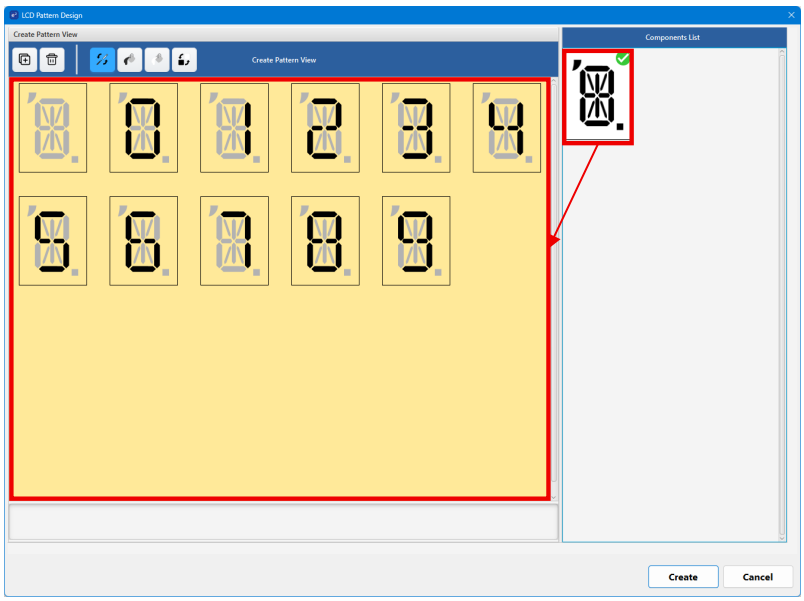
### 4.2.3 Implementation

Make the settings for “3. Implementation” in the workflow view of QE for Segment LCD[RL78].

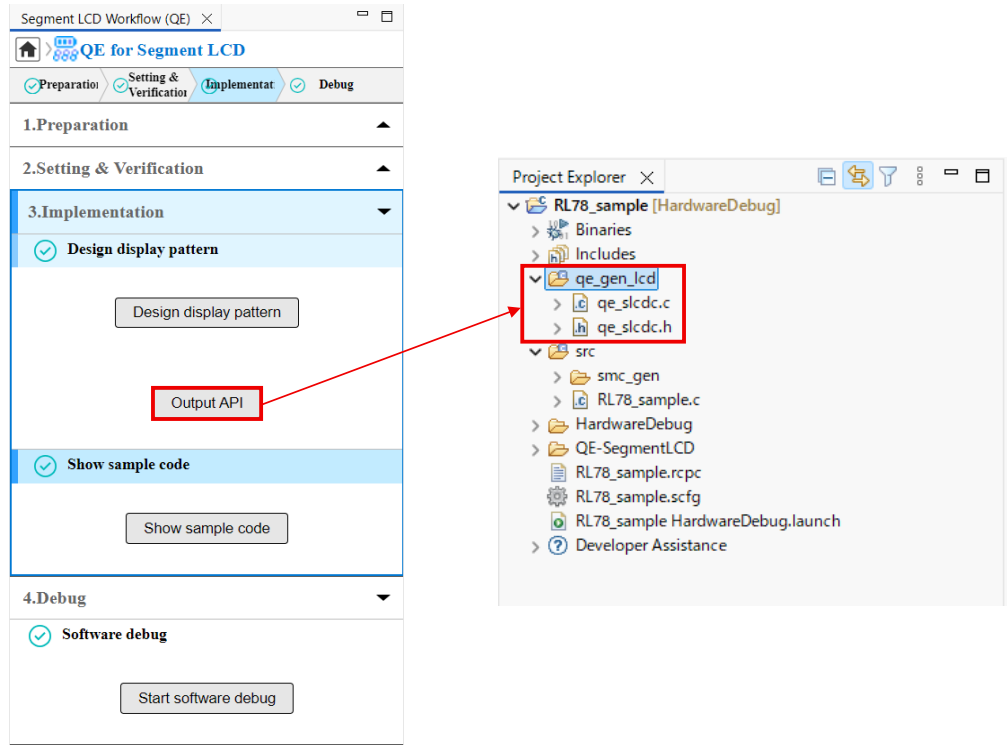
1. In “Design display pattern”, click “Design display pattern”.



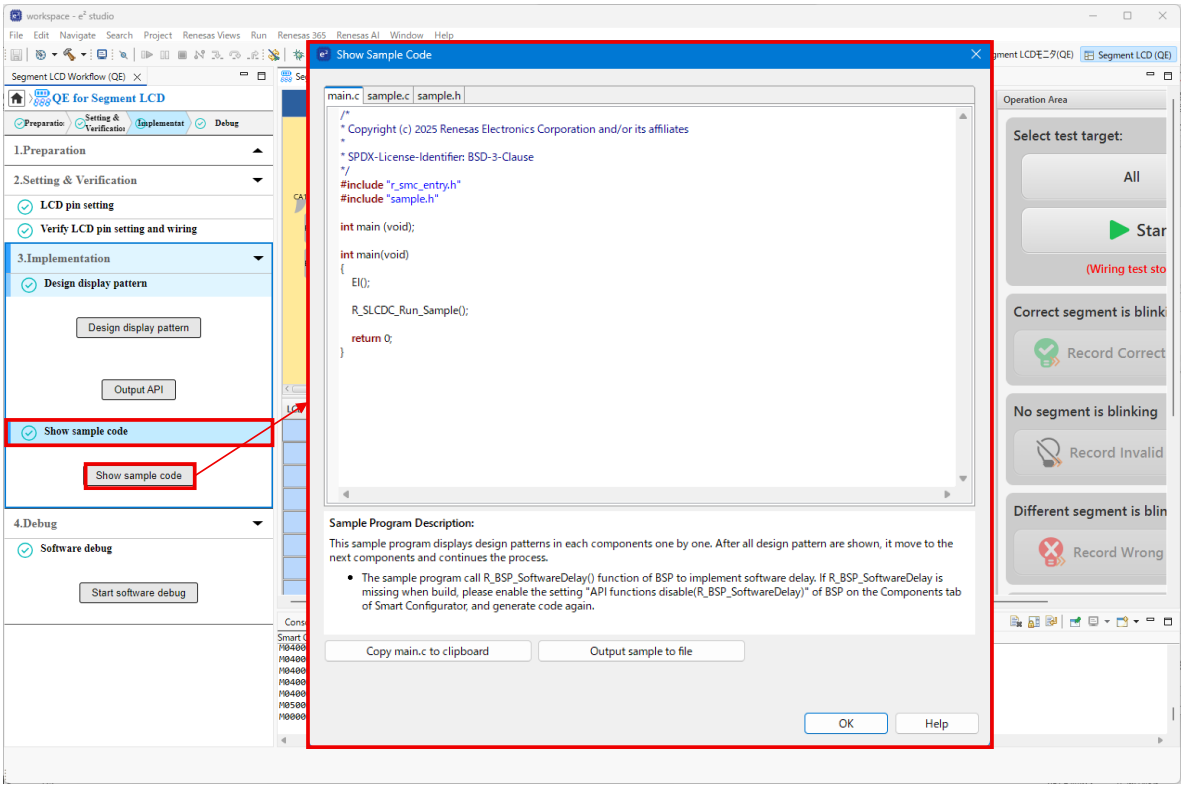

2. When you select a component to create design patterns, the default design pattern for that component is displayed in the “Create Pattern View”.
   - To create design patterns other than the default ones, use the tools in the toolbar.
   - If identical design patterns exist, an error will occur. You cannot generate design patterns or edit other components until the error is resolved.



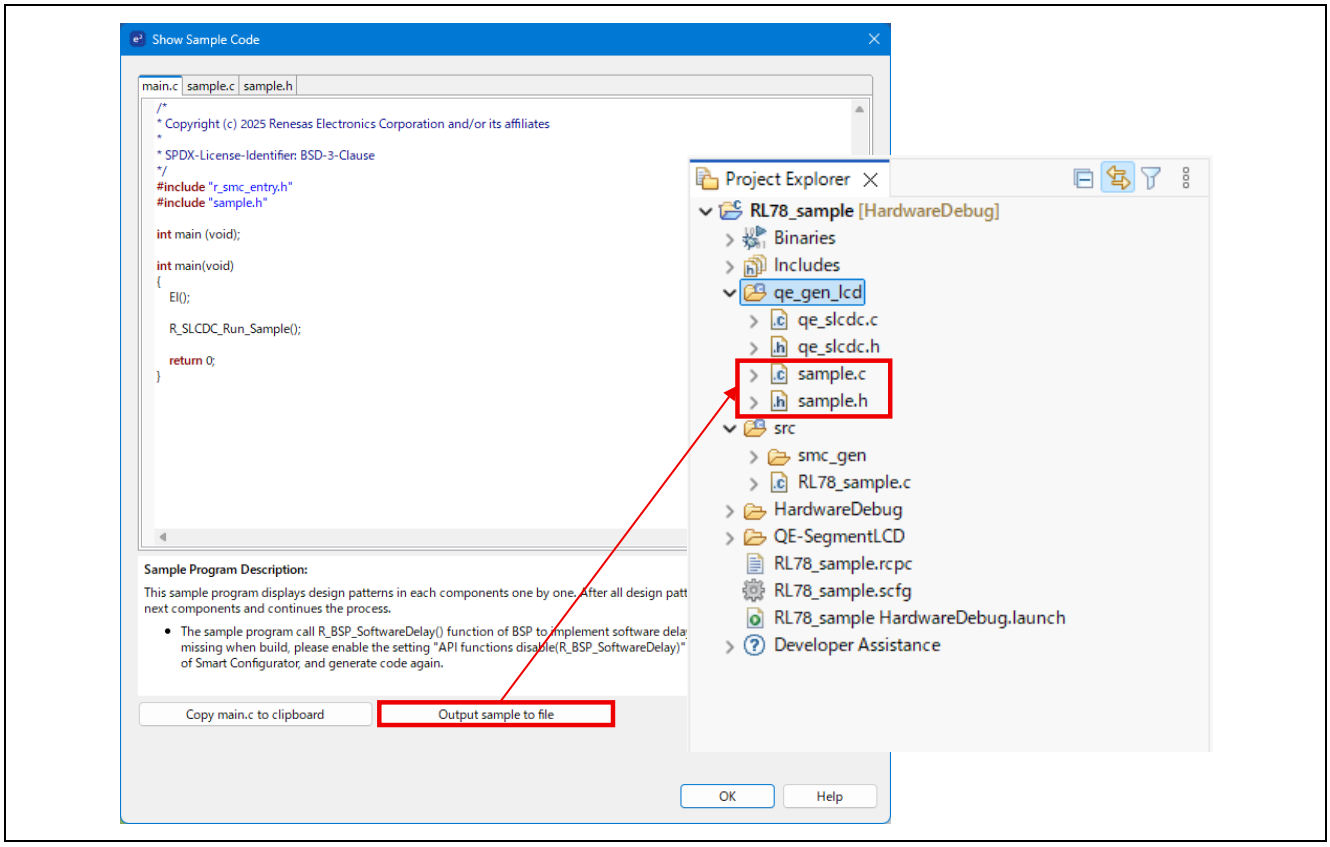
3. Click “Output API” to output the “qe_gen_lcd” folder.



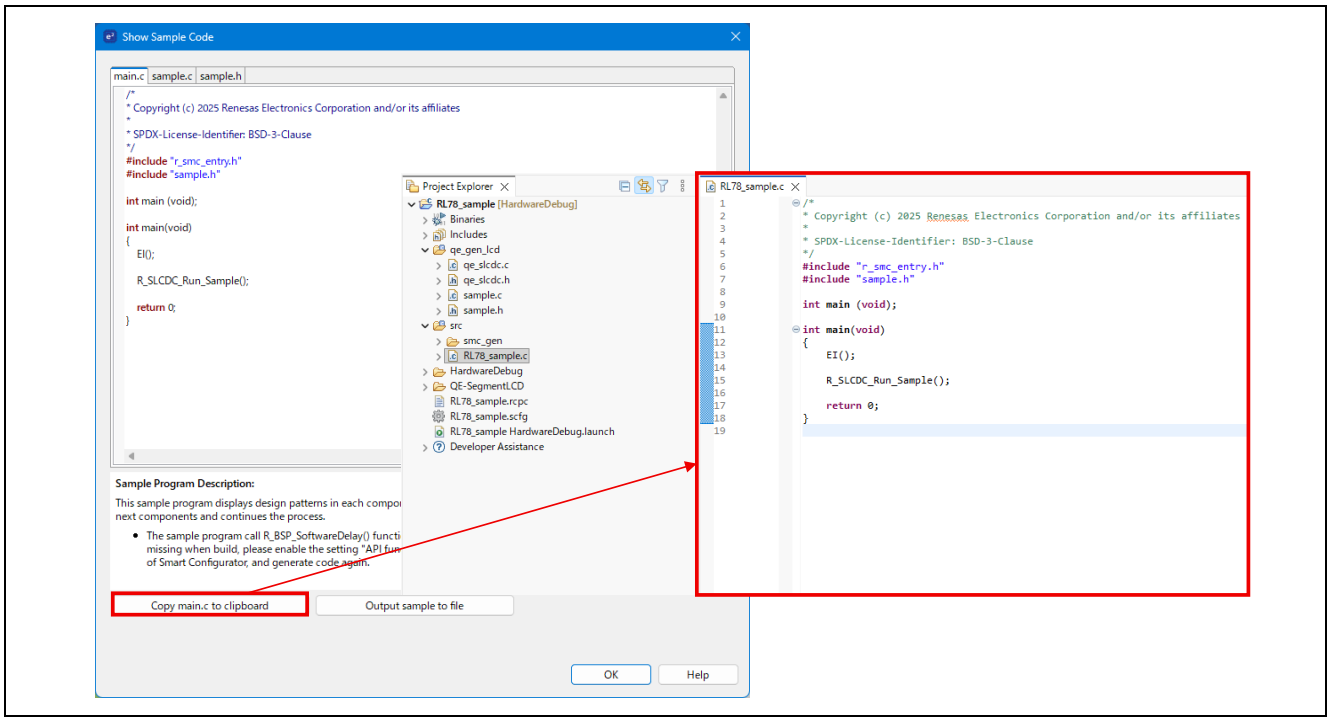

4. In “Show sample code”, click “Show sample code” to display the dialog.



5. In the dialog, click “Output sample to file” to output the “sample.c” and “sample.h” to the “qe_gen_lcd” folder.




6. Click the “Copy main.c to clipboard” button and paste the content into the source file that serves as the entry point for the project.

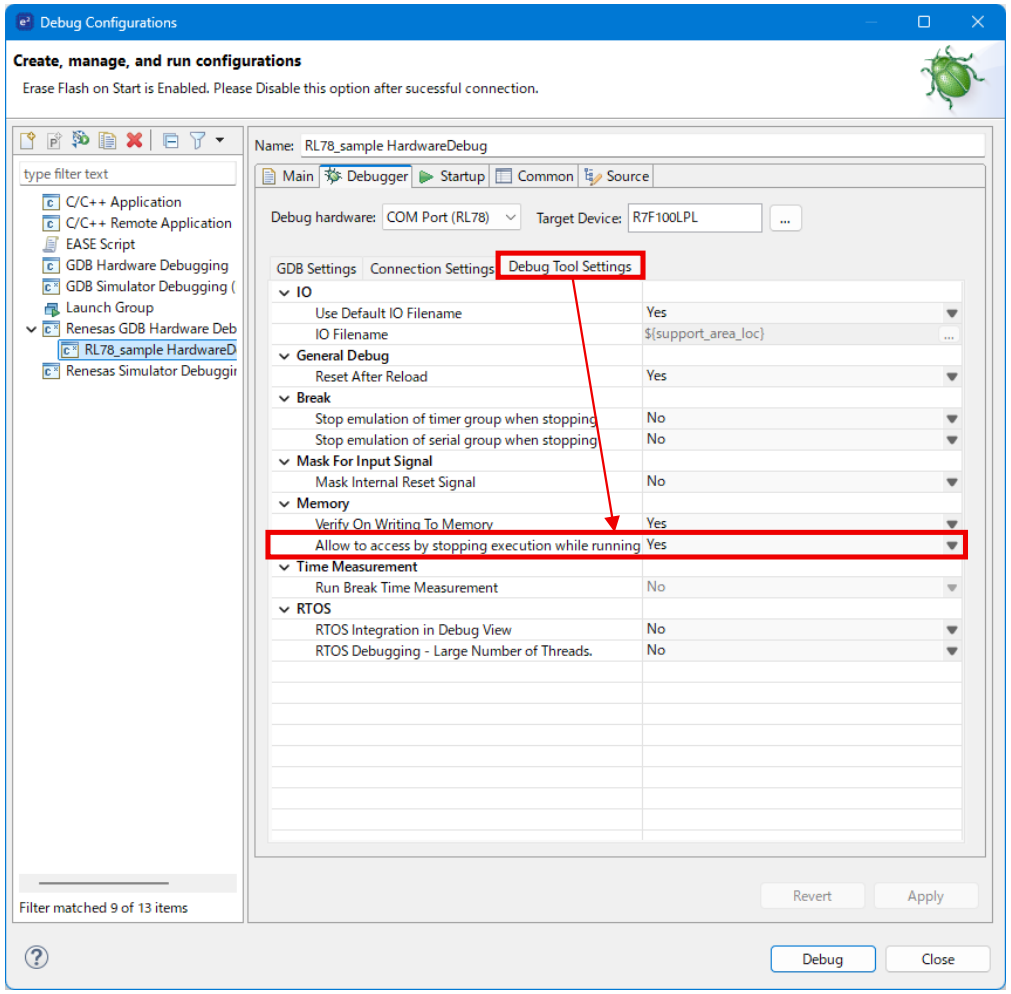


### 4.2.4 Debug

Make the settings for “4. Debug” in the workflow view of QE for Segment LCD[RL78].

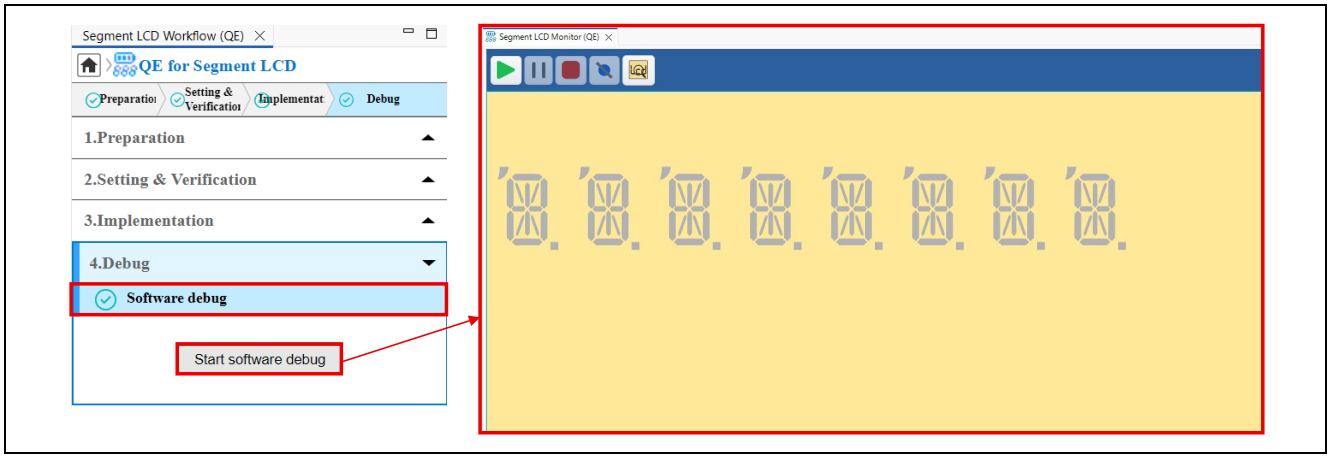
1. Make sure the target board is connected to host PC via E2 Emulator, E2 Emulator Lite or COM connection.

2. In the “Debug Tool Settings” tab of the “Debug Configurations”, set “Allow to access by stopping execution while running” to “Yes”.

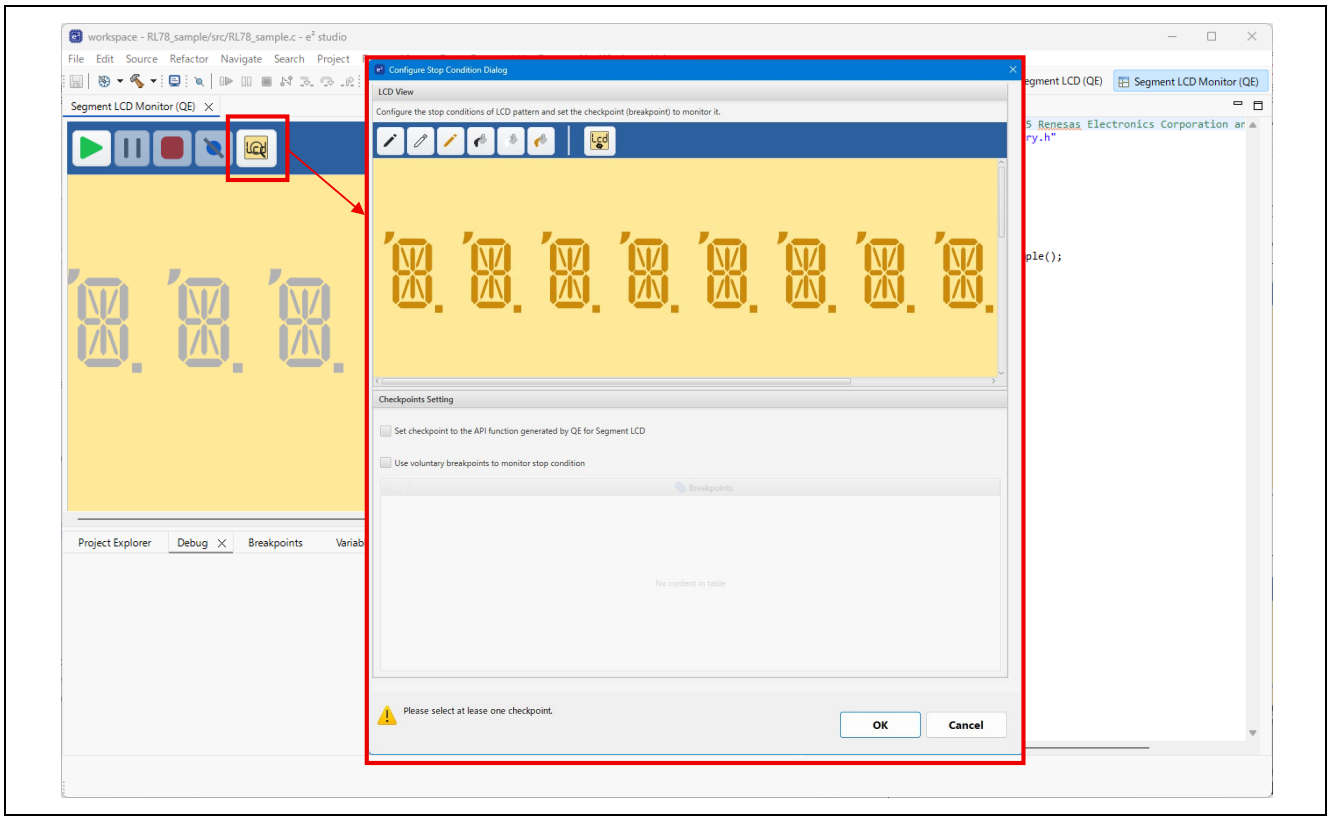



3. Build the target project and confirm that there are no errors.

4. Click “Start software debug”, to display the “Segment LCD Monitor (QE)” view.

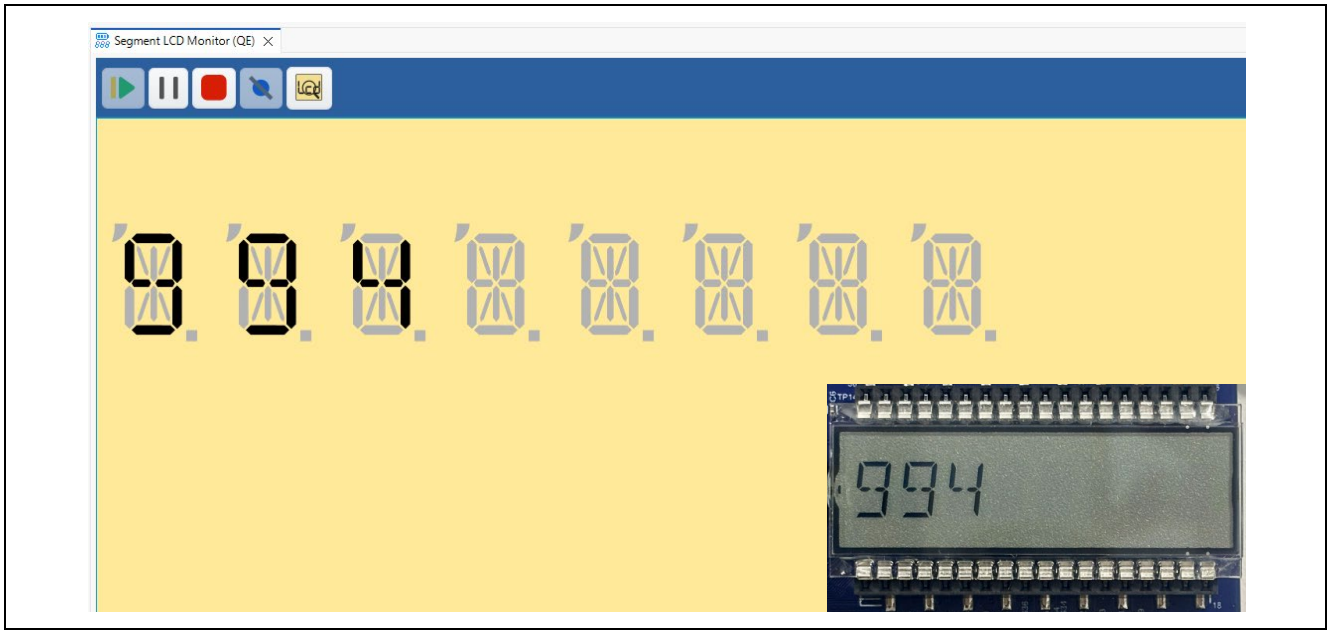



5. In the “Segment LCD Monitor (QE)” view, open the Configure Stop Condition Dialog, and configure the LCD stop conditions and checkpoint where you want the program to stop.

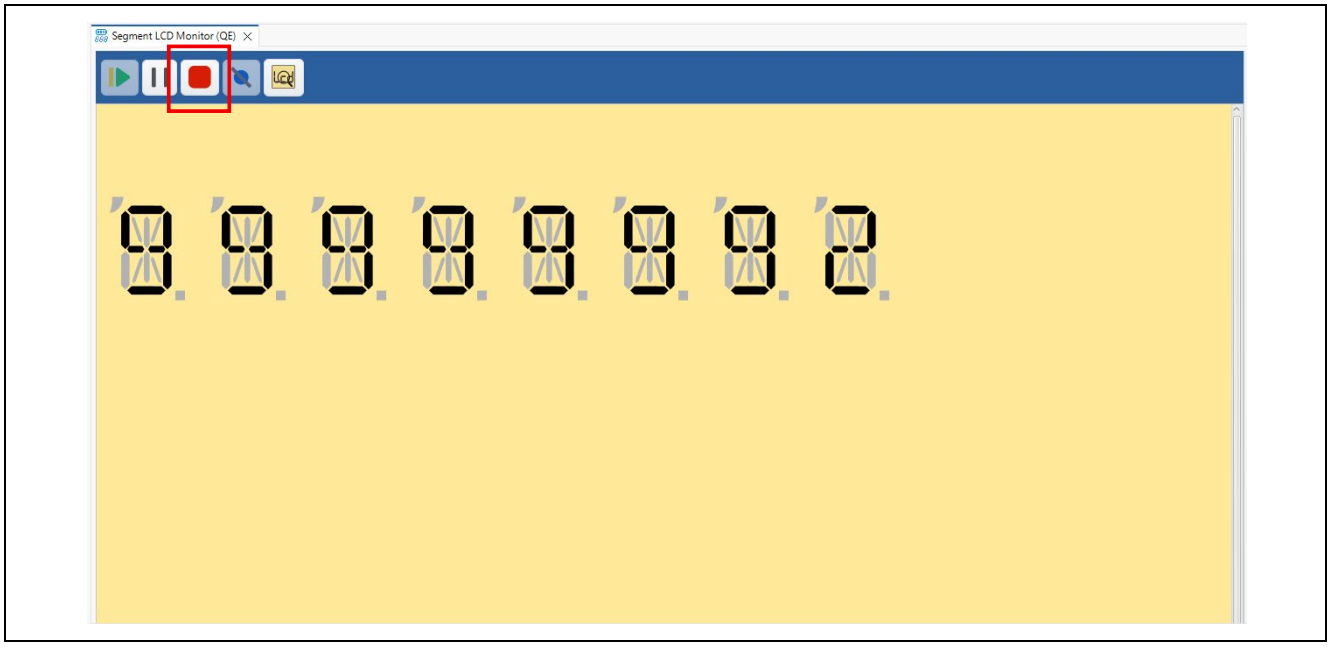


6. Click “Start the debug” button in the “Segment LCD Monitor (QE)” view to start debugging. The program pauses at checkpoint when the LCD display pattern matches the stop condition.


7. Confirm that the registered design patterns are displayed in order starting from the leftmost digit of the segment LCD.



8. To stop the debugging, click the “stop the debug” button.

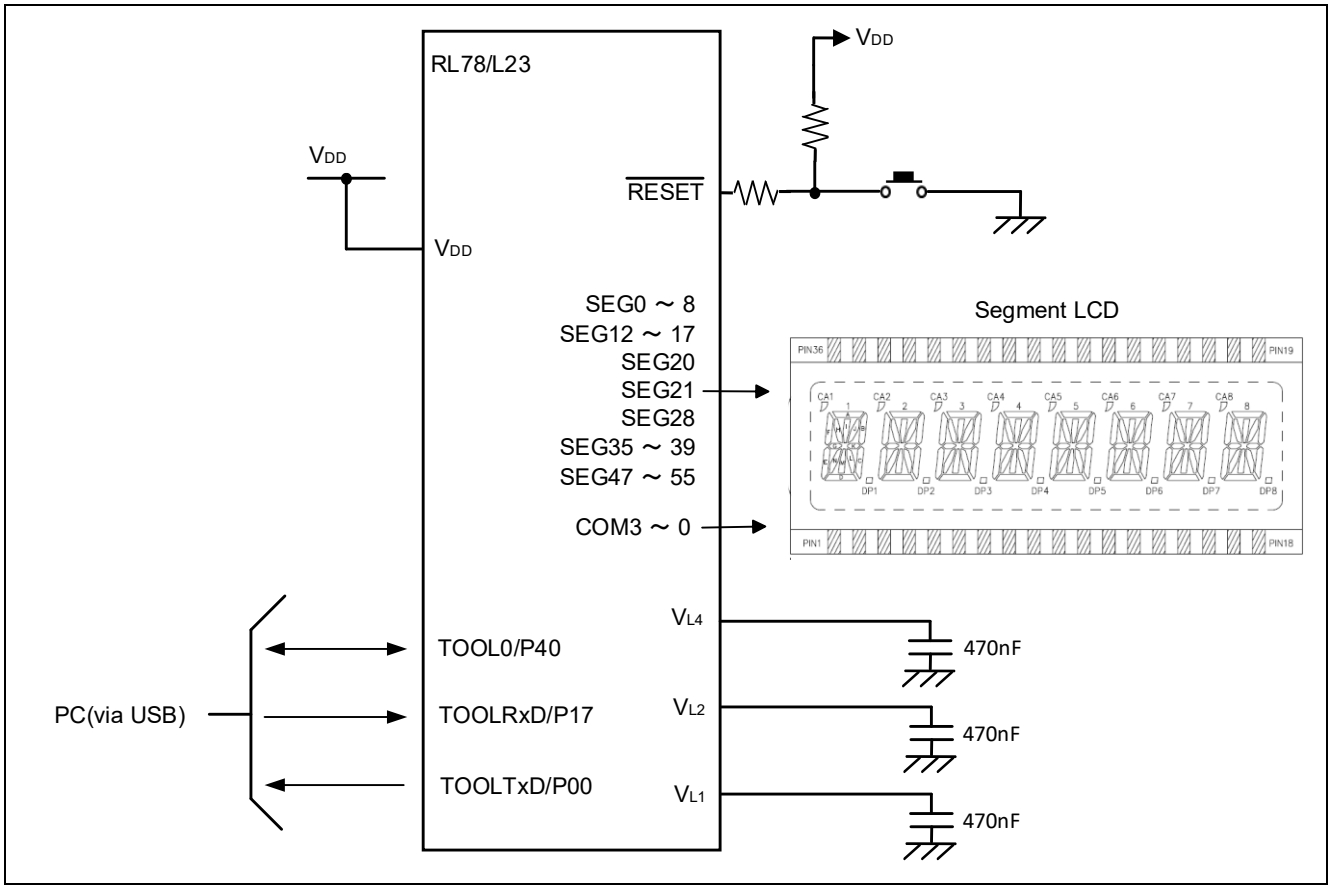


# 5. Hardware

## 5.1 Hardware Example

Figure 5-1 shows the hardware configuration used in this application note.

**Figure 5-1 Hardware Configuration**




Note

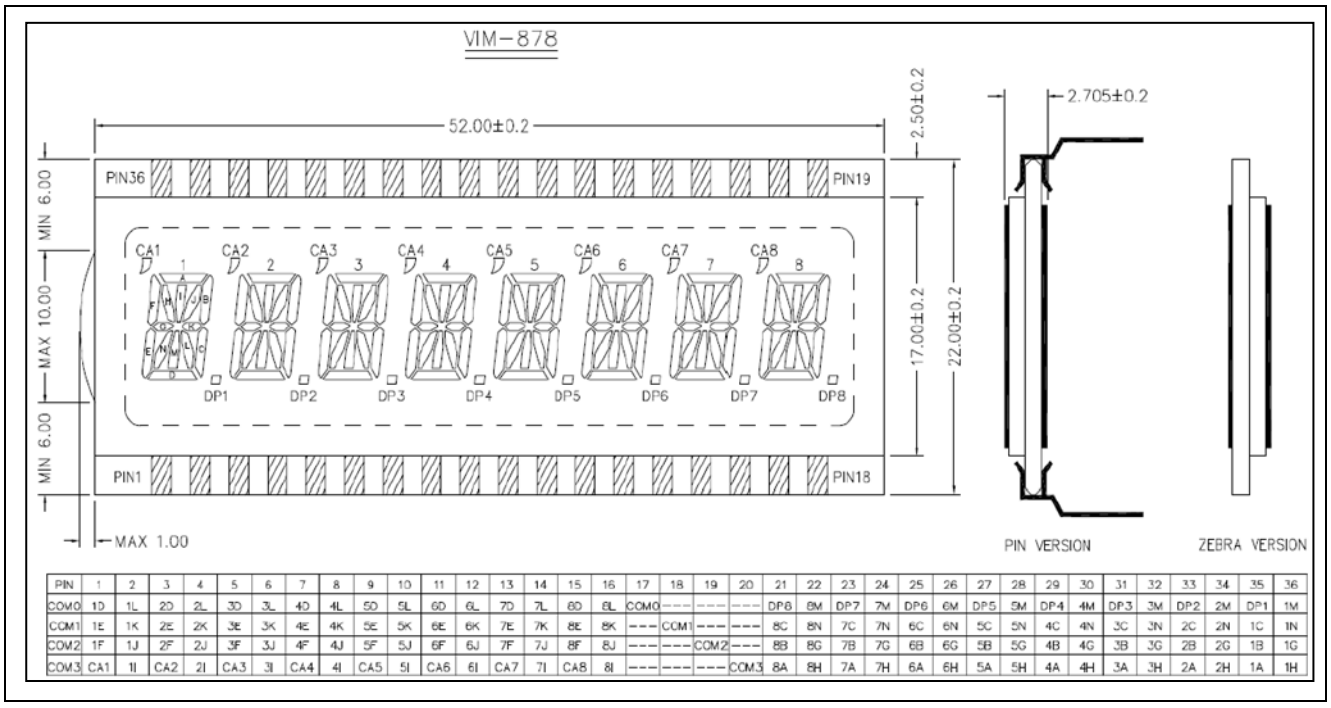
1. The above figure is simplified to show an overview of the hardware connection. When designing application circuits, make sure to handle unused pins appropriately to satisfy the electrical characteristics (connect input only ports independently to either $V_{DD}$ or $V_{SS}$ via resistors).
2. Make sure to set $V_{DD}$ greater than the detection voltage ($V_{LVD}$) specified by the LVD.

## 5.2 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 5-2 shows the panel image and the pin allocation table.

**Figure 5-2 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 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COM0 | 1D | 1L | 2D | 2L | 3D | 3L | 4D | 4L | 5D | 5L | 6D | 6L | 7D | 7L | 8D | 8L | COM0 | --- | --- | --- | DP8 | 8M | DP7 | 7M | DP6 | 6M | DP5 | 5M | DP4 | 4M | DP3 | 3M | DP2 | 2M | DP1 | 1M |
| COM1 | 1E | 1K | 2E | 2K | 3E | 3K | 4E | 4K | 5E | 5K | 6E | 6K | 7E | 7K | 8E | 8K | --- | COM1 | --- | --- | 8C | 8N | 7C | 7N | 6C | 6N | 5C | 5N | 4C | 4N | 3C | 3N | 2C | 2N | 1C | 1N |
| COM2 | 1F | 1J | 2F | 2J | 3F | 3J | 4F | 4J | 5F | 5J | 6F | 6J | 7F | 7J | 8F | 8J | --- | --- | COM2 | --- | 8B | 8G | 7B | 7G | 6B | 6G | 5B | 5G | 4B | 4G | 3B | 3G | 2B | 2G | 1B | 1G |
| COM3 | CA1 | 1I | CA2 | 2I | CA3 | 3I | CA4 | 4I | CA5 | 5I | CA6 | 6I | CA7 | 7I | CA8 | 8I | --- | --- | --- | 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)

## 5.3 Pins Used

**Table 5-1 Pins Used and Their Functions**

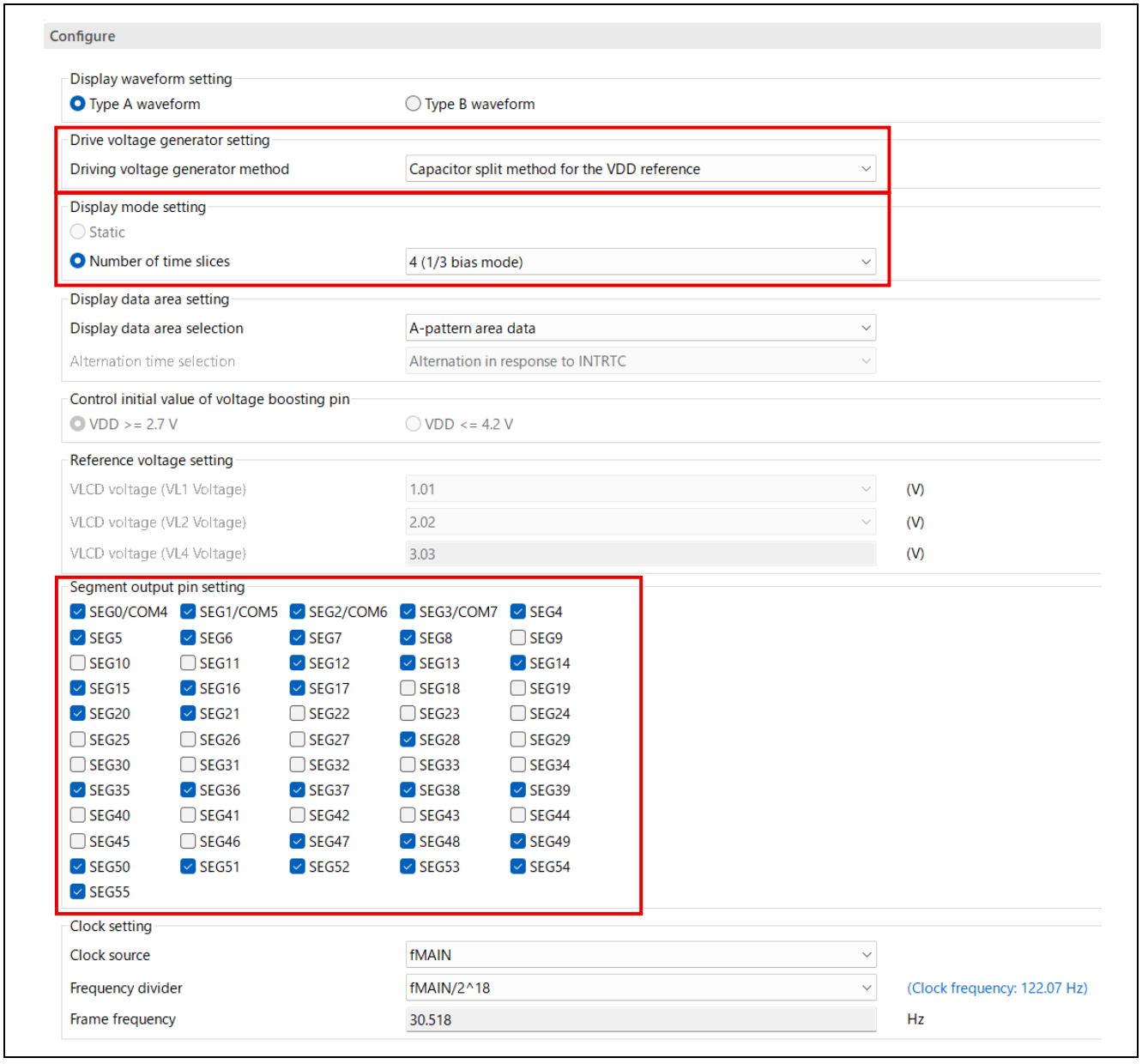
| Pin name | I/O | Functions |
|---|---|---|
| P94/SEG0 | Output | LCD controller/driver common signals |
| P95/SEG1 | | |
| P96/SEG2 | | |
| P97/SEG3 | | |
| P50/SEG4 | | |
| P51/SEG5 | | |
| P52/SEG6 | | |
| P53/SEG7 | | |
| P54/SEG8 | | |
| P70/SEG12 | | |
| P71/SEG13 | | |
| P72/SEG14 | | |
| P73/SEG15 | | |
| 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/VL1 | - | LCD drive voltage |
| P86/VL2 | | |
| P85/VL4 | | |
| P40/TOOL0 | Input-Output | COM Port debugging |
| P17/TOOLRxD | Input | |
| P00/TOOLTxD | Output | |

# 6. Software

## 6.1 Smart Configurator Settings

Figure 6-1 shows the settings of LCD components (Config_LCD).

**Figure 6-1 LCD Components (Config_LCD) Setting**



## 6.2 Design Pattern API Specifications

The API specifications for the design patterns output to the “qe_gen_lcd” folder are shown below.

R_SLCDC_Set_Pattern

| | | |
|---|---|---|
| **Outline** | Sets a design pattern for the specified component | |
| **Header** | qe_slcdc.h | |
| **Declaration** | MD_STATUS R_SLCDC_Set_Pattern(uint16_t component_id, uint16_t pattern_index) | |
| **Description** | Sets the display data corresponding to the pattern index for the specified component ID. | |
| **Arguments** | uint16_t component_id | Component ID |
| | uint16_t pattern_index | Index of the design pattern to be set |
| **Return Value** | MD_OK | Normal completion |
| | MD_ARGERROR | Argument error |
| **Remarks** | None | |

# 7. Application Example of QE for Segment LCD (Clock)

This section explains a clock implementation example using QE for Segment LCD.

The workflow settings and sample code for updating the LCD display every second using the APIs generated by QE for Segment LCD are shown below.

## 7.1 Setting LCD According to the Workflow of QE for Segment LCD

### 7.1.1 Preparation

For “1. Preparation” in the QE for Segment LCD[RL78] workflow, refer to “4.2.1 Preparation”.

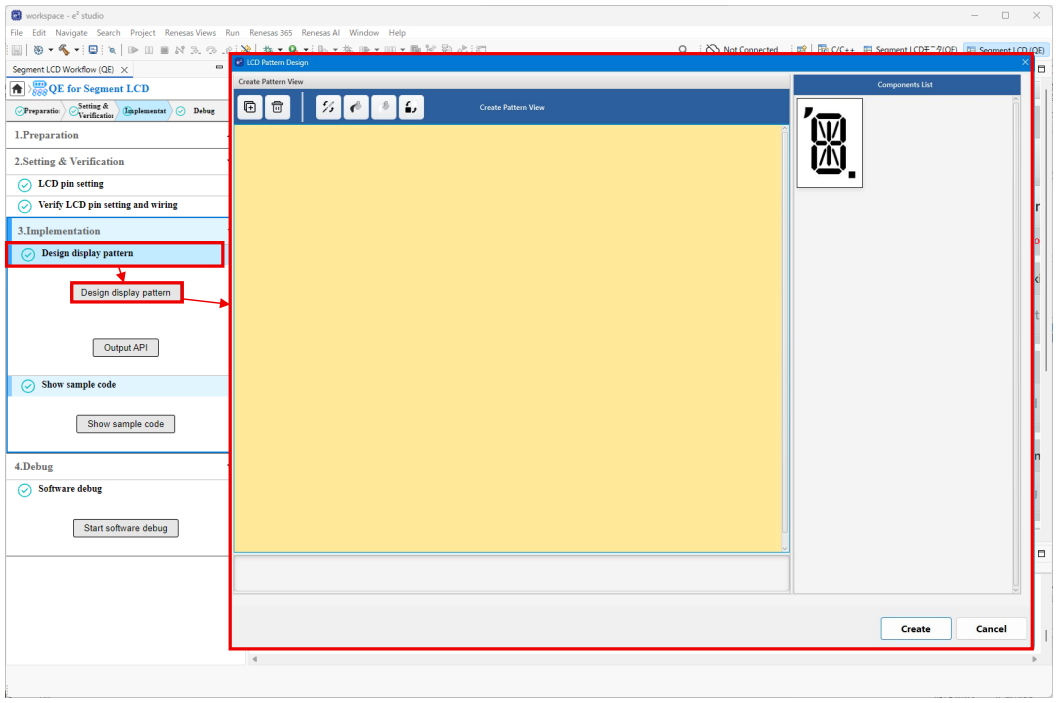
### 7.1.2 Settings & Verification

For “2. Settings & Verification” in the QE for Segment LCD[RL78] workflow, refer to “4.2.2 Setting & Verification”.

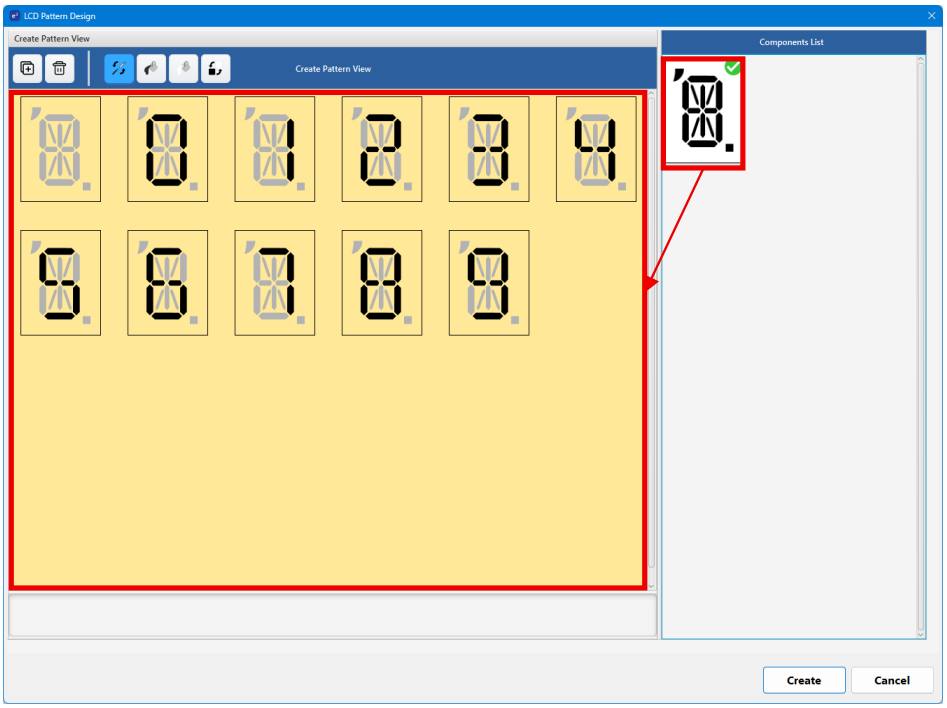
### 7.1.3 Implementation

Make the settings for “3. Implementation” in the workflow view of QE for Segment LCD[RL78].

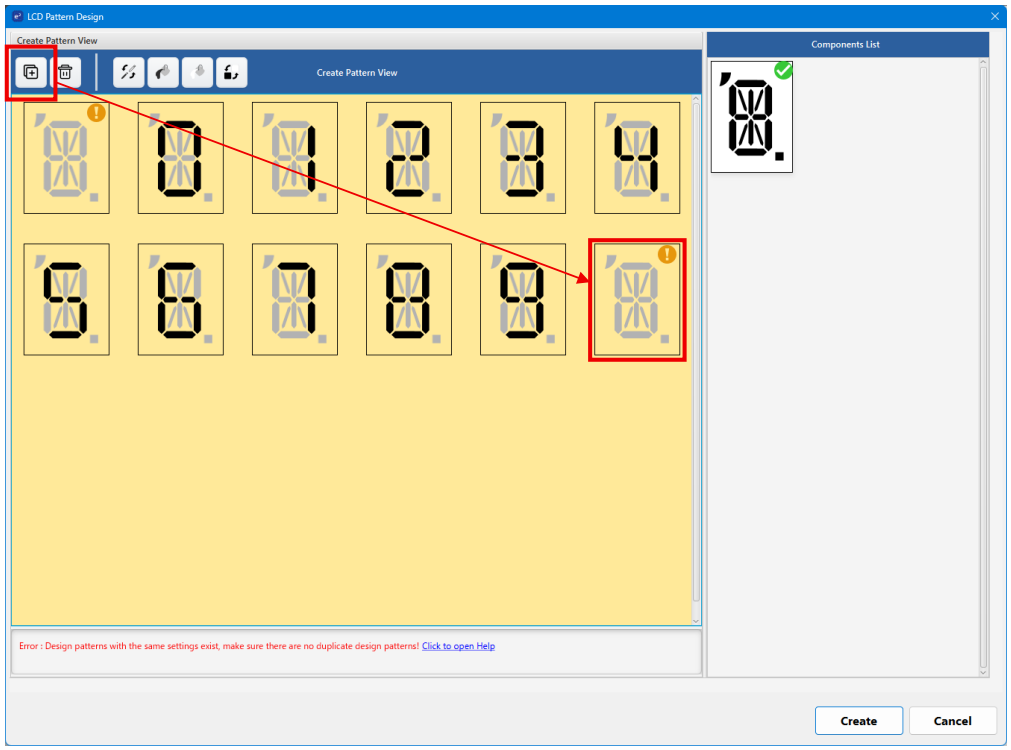
1. In “Design display pattern”, click “Design display pattern”.



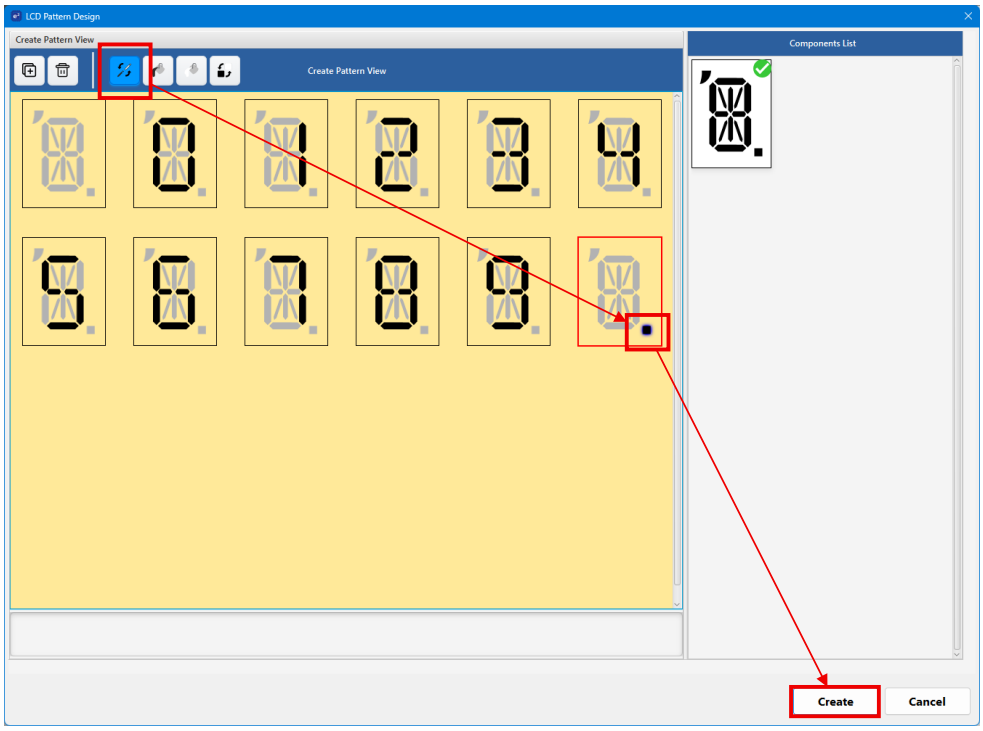

2. When you select a component to create design patterns, the default design pattern for that component is displayed in the “Create Pattern View”.



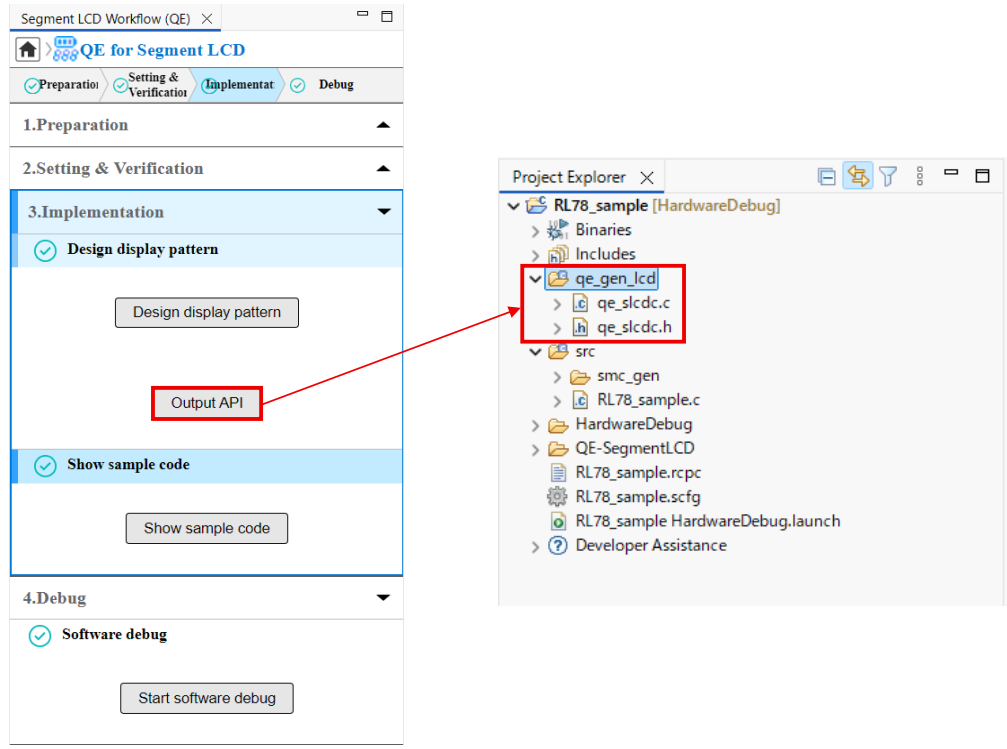
**3.** Add a design pattern using the button on the toolbar.



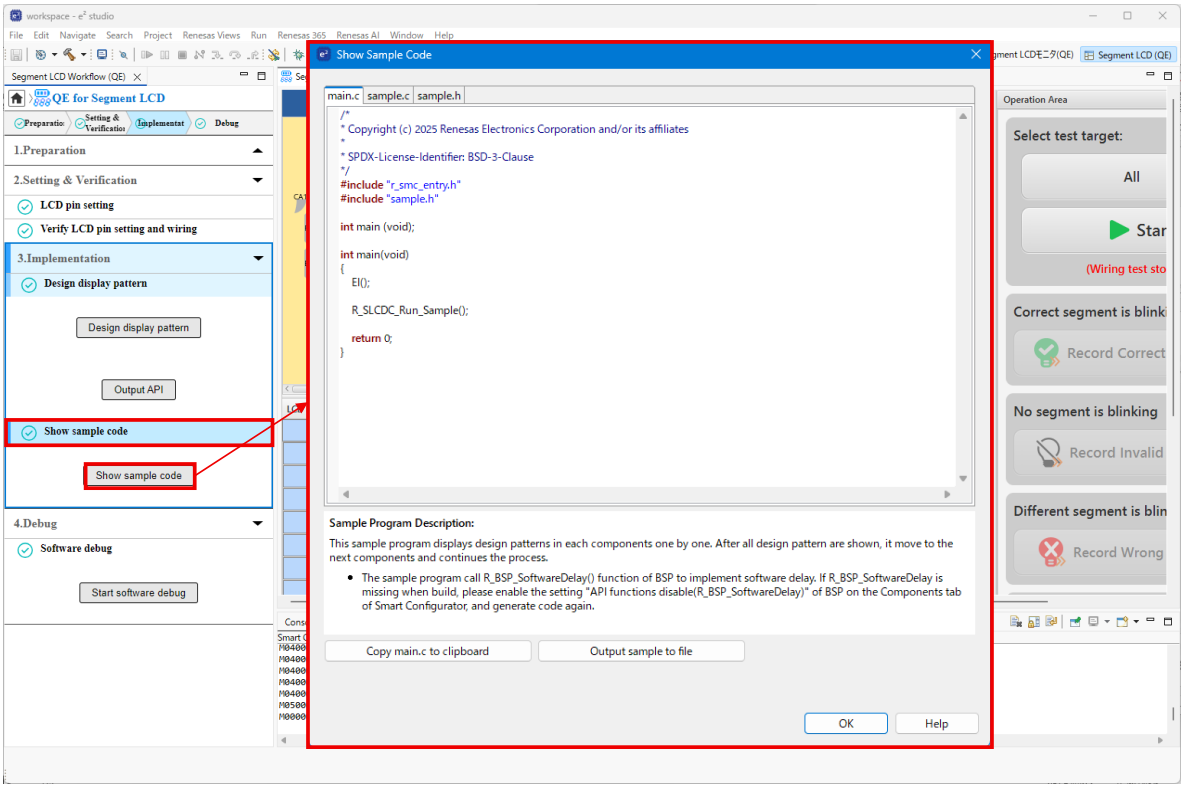

**4.** Use the tools to change the pattern to a ".(period)" and click the "Create.



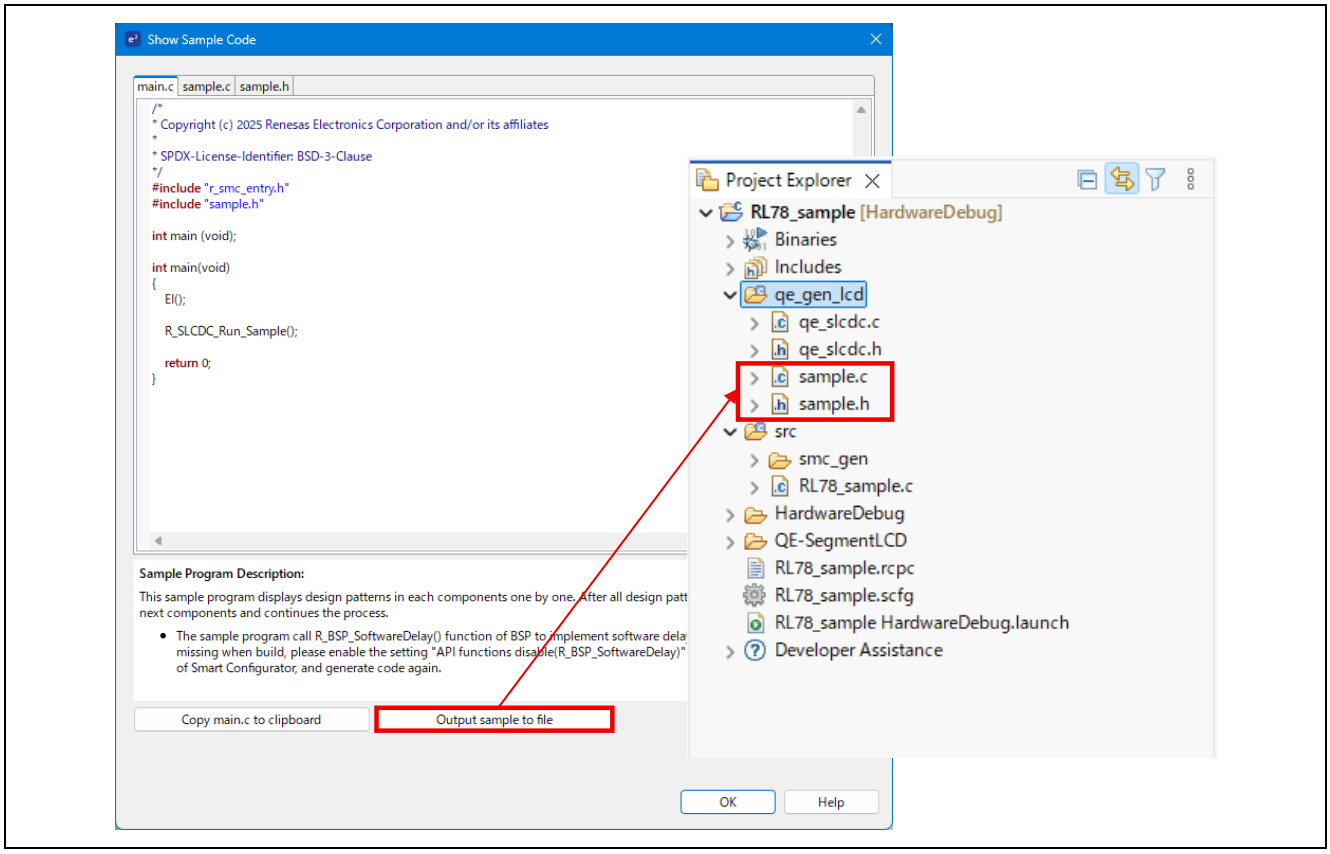
5. Click “Output API” to output the “qe_gen_lcd” folder.



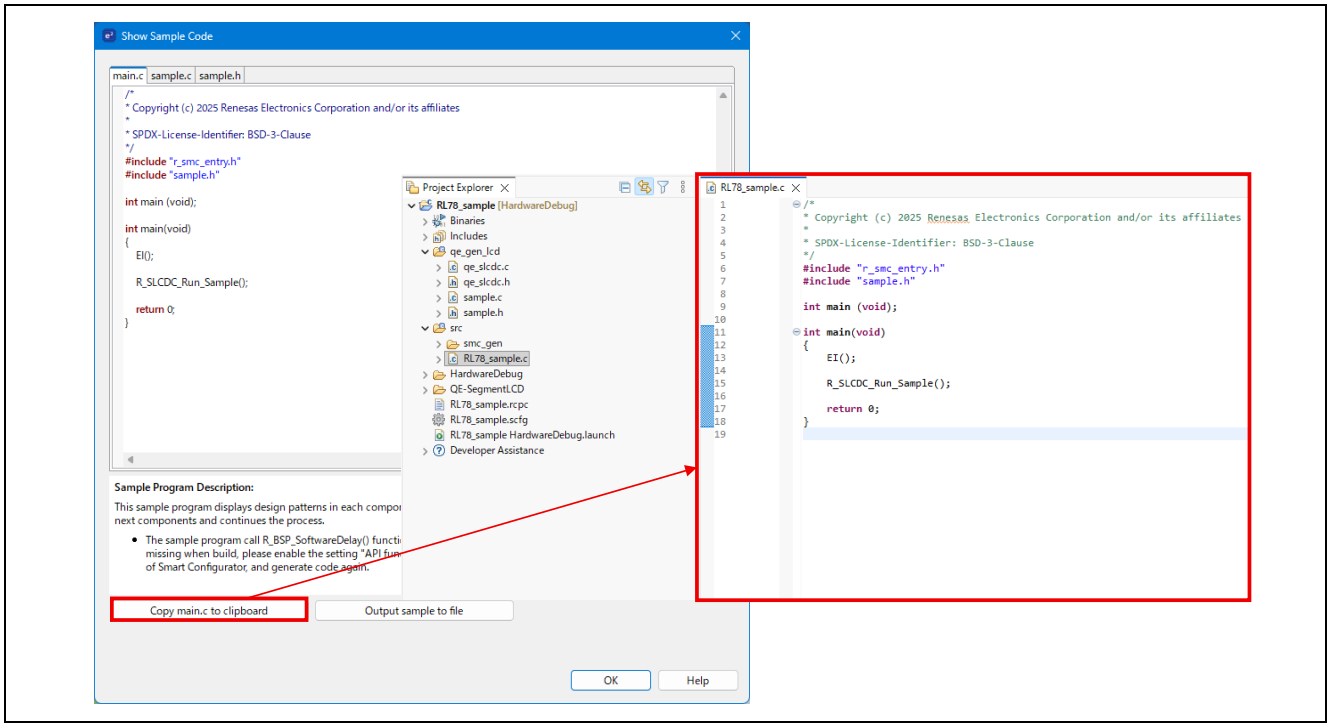

6. In “Show sample code”, click “Show sample code” to display the dialog.



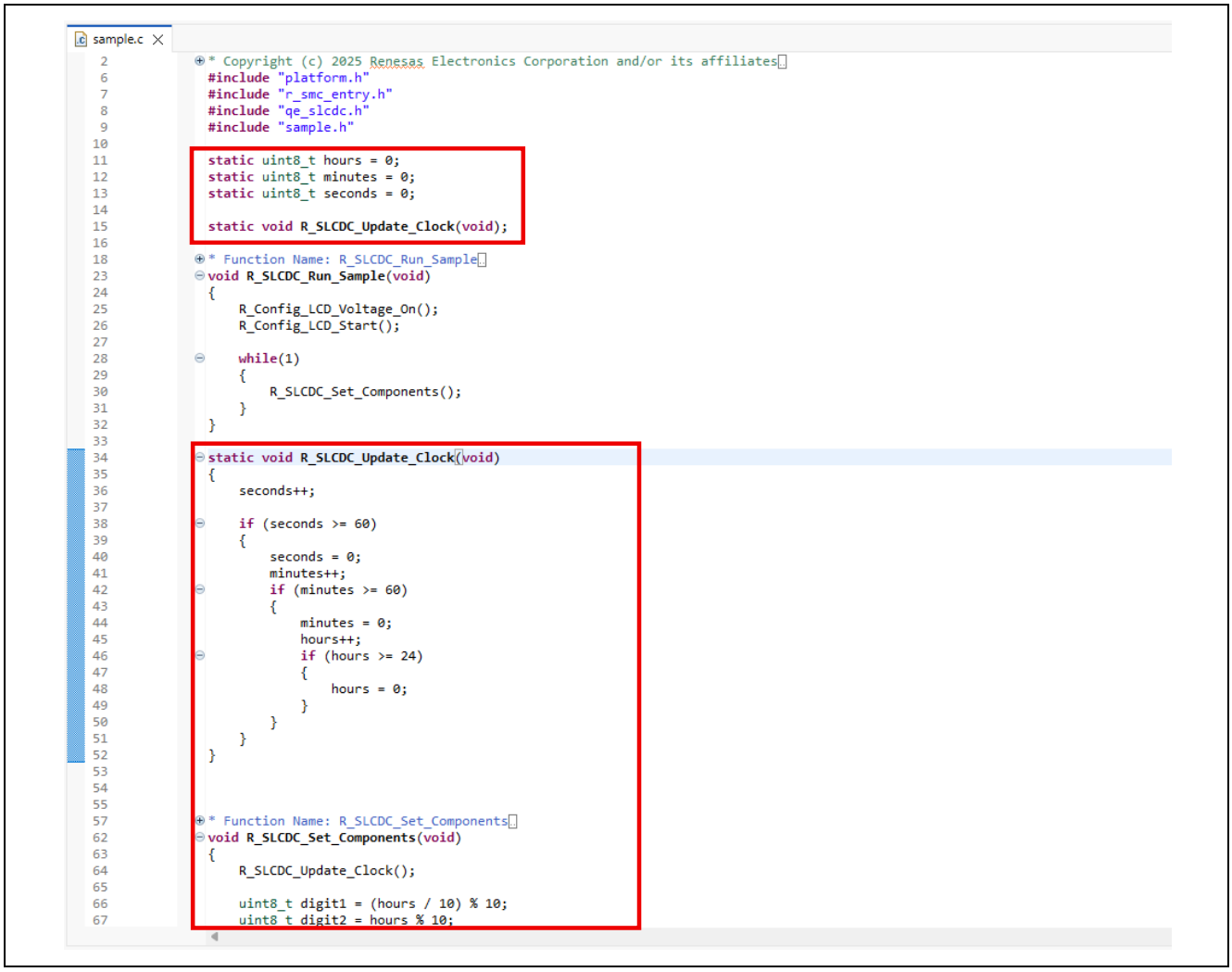
7. In the dialog, click “Output sample to file” to output the “sample.c” and “sample.h” to the “qe_gen_lcd” folder.




8. Click the “Copy main.c to clipboard” button and paste the content into the source file that serves as the entry point for the project.



**9.** Paste the variable definitions and functions described in “7.2 Sample Code” into the appropriate locations in “sample.c”.

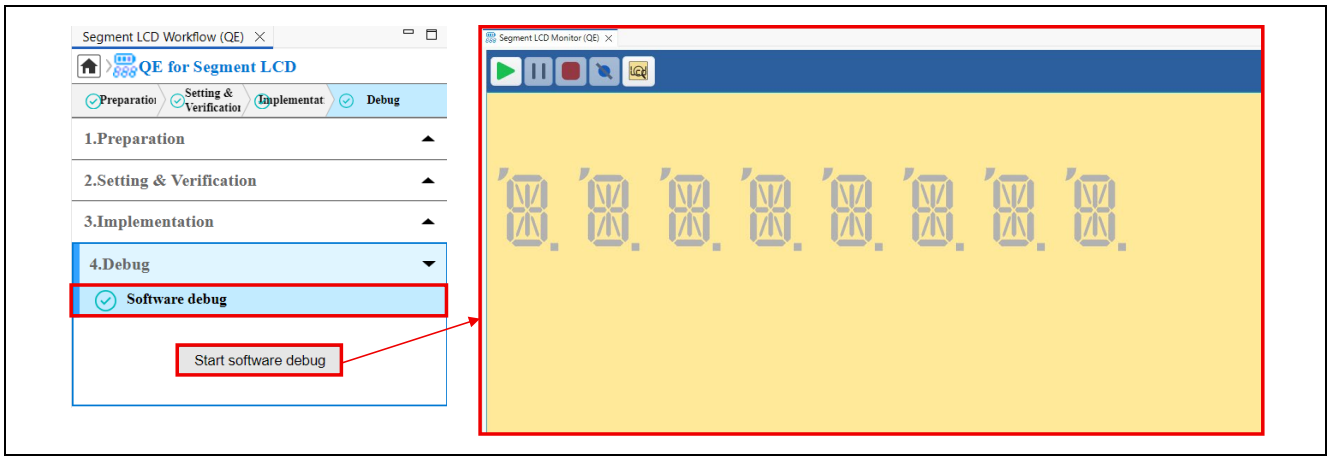


```c
/*
* Copyright (c) 2025 Renesas Electronics Corporation and/or its affiliates
*/
#include "platform.h"
#include "r_smc_entry.h"
#include "qe_slcdc.h"
#include "sample.h"

static uint8_t hours = 0;
static uint8_t minutes = 0;
static uint8_t seconds = 0;

static void R_SLCDC_Update_Clock(void);

/*
* Function Name: R_SLCDC_Run_Sample
*/
void R_SLCDC_Run_Sample(void)
{
    R_Config_LCD_Voltage_On();
    R_Config_LCD_Start();

    while(1)
    {
        R_SLCDC_Set_Components();
    }
}

static void R_SLCDC_Update_Clock(void)
{
    seconds++;

    if (seconds >= 60)
    {
        seconds = 0;
        minutes++;
        if (minutes >= 60)
        {
            minutes = 0;
            hours++;
            if (hours >= 24)
            {
                hours = 0;
            }
        }
    }
}


/*
* Function Name: R_SLCDC_Set_Components
*/
void R_SLCDC_Set_Components(void)
{
    R_SLCDC_Update_Clock();

    uint8_t digit1 = (hours / 10) % 10;
    uint8_t digit2 = hours % 10;
```

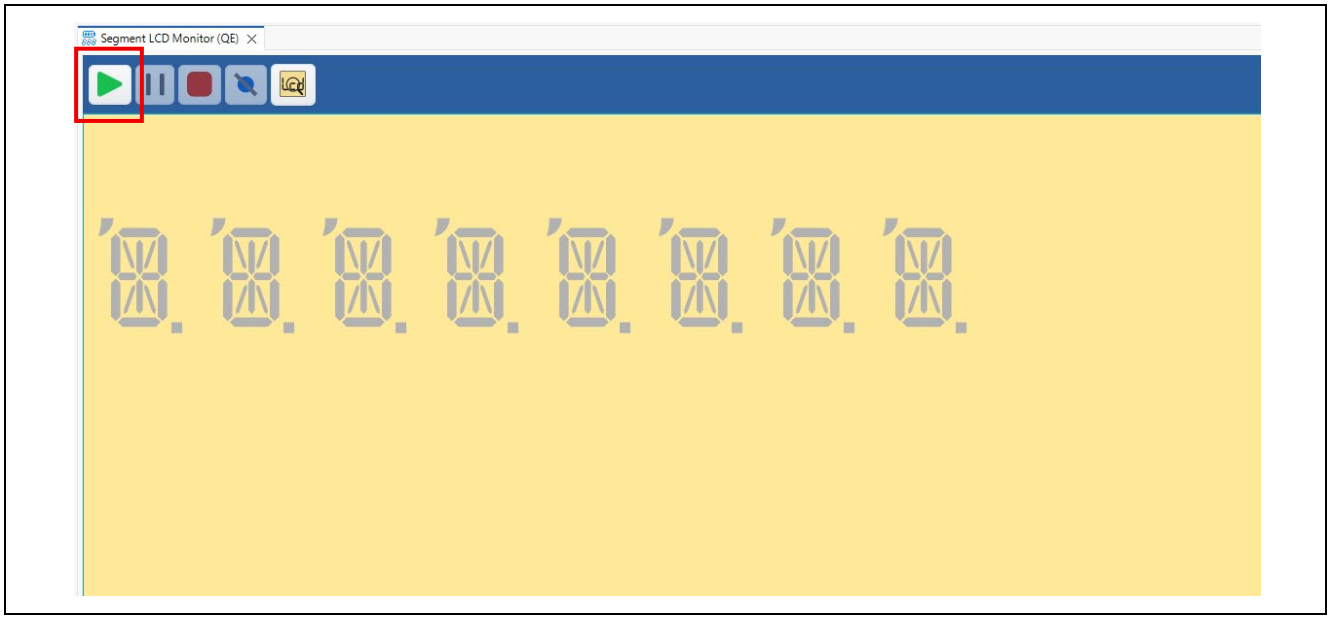
### 7.1.4 Debug

For steps 1 to 3 of “4. Debug” in the QE for Segment LCD[RL78] workflow, refer to “4.2.4 Debug”.

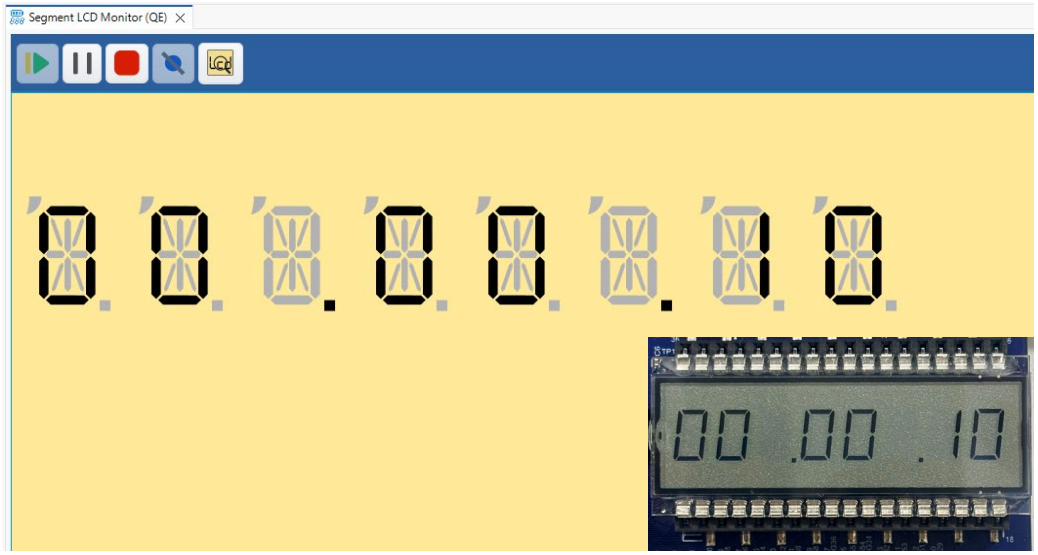
**4.** Click “Start software debug”, to display the “Segment LCD Monitor (QE)” view.



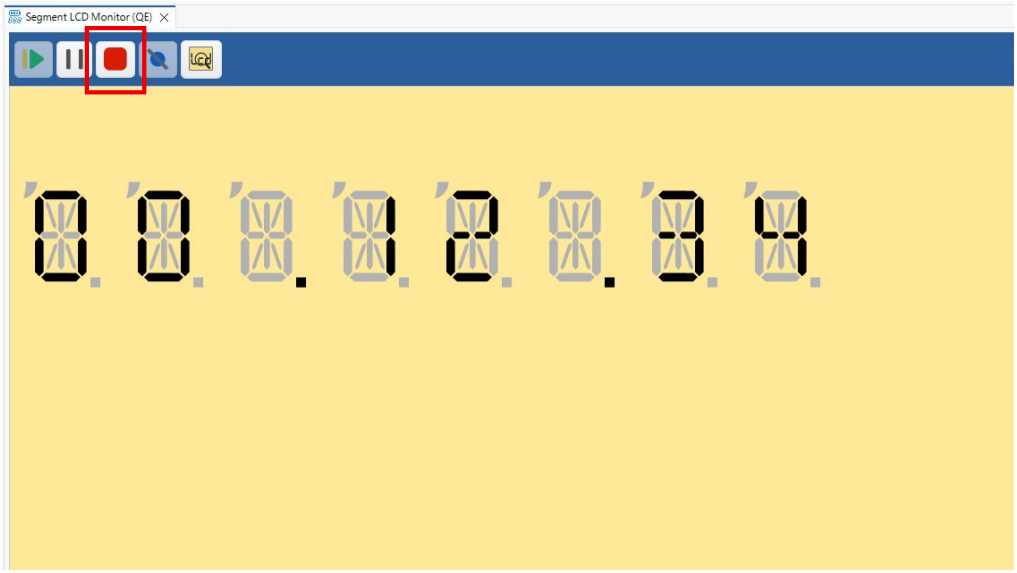

**5.** Click “Start the debug” button in the “Segment LCD Monitor (QE)” view to start debugging.



6. Confirm that the display on the segment LCD is counting up every second.




7. To stop the debugging, click the “stop the debug” button.



## 7.2 Sample Code

The function descriptions and sample code for displaying the clock are shown below.

R_SLCDC_Update_Clock

| | |
|---|---|
| **Outline** | Count-up process for the clock (seconds, minutes, and hours) |
| **Header** | None |
| **Declaration** | static void R_SLCDC_Update_Clock(void) |
| **Description** | Increments the variables for seconds, minutes, and hours, and updates the time data in a 24-hour cycle. |
| **Arguments** | None |
| **Return Value** | None |
| **Remarks** | None |

R_SLCDC_Set_Components

| | |
|---|---|
| **Outline** | Display update for the clock demo and setting of design patterns |
| **Header** | sample.h |
| **Declaration** | void R_SLCDC_Set_Components(void) |
| **Description** | Updates the time, sets the numerical values corresponding to each LCD component as design patterns, and then waits for one second. |
| **Arguments** | None |
| **Return Value** | None |
| **Remarks** | None |

```
static uint8_t hours = 0;
static uint8_t minutes = 0;
static uint8_t seconds = 0;

static void R_SLCDC_Update_Clock(void);


static void R_SLCDC_Update_Clock(void)
{
  seconds++;

    if (seconds >= 60)
    {
        seconds = 0;
        minutes++;
        if (minutes >= 60)
        {
            minutes = 0;
            hours++;
            if (hours >= 24)
            {
                hours = 0;
            }
        }
    }
}
```

```
void R_SLCDC_Set_Components(void)
{
    R_SLCDC_Update_Clock();

    uint8_t digit1 = (hours / 10) % 10;
    uint8_t digit2 = hours % 10;
    uint8_t digit3 = (minutes / 10) % 10;
    uint8_t digit4 = minutes % 10;
    uint8_t digit5 = (seconds / 10) % 10;
    uint8_t digit6 = seconds % 10;

    for (uint16_t component_index = 0; component_index <
QE_SLCDC_NUM_COMPONENT; component_index++)
    {
        uint16_t pattern_index = 0;

        switch (component_index)
        {
            case 0:
                pattern_index = digit1 + 1;
                break;
            case 1:
                pattern_index = digit2 + 1;
                break;
            case 2:
                pattern_index = 11;
                break;
            case 3:
                pattern_index = digit3 + 1;
                break;
            case 4:
                pattern_index = digit4 + 1;
                break;
            case 5:
                pattern_index = 11;
                break;
            case 6:
                pattern_index = digit5 + 1;
                break;
            case 7:
                pattern_index = digit6 + 1;
                break;
            default:
                pattern_index = 0;
                break;
        }

        R_SLCDC_Set_Pattern(components[component_index].component_id,
pattern_index);
    }

    R_BSP_SoftwareDelay(1000, BSP_DELAY_MILLISECS);
}
```

# 8. Notes

Please pay attention to the following items.

## 8.1 Notes on opening [Segment LCD Workflow (QE)] view

The workflow cannot be opened if WebView2 Runtime is not installed on your PC.

[Workaround]

Download and install WebView2 (x64 version) from the Microsoft web page.

## 8.2 Notes on sample code license

The license applied to the sample code generated by this tool is BSD-3-Clause.

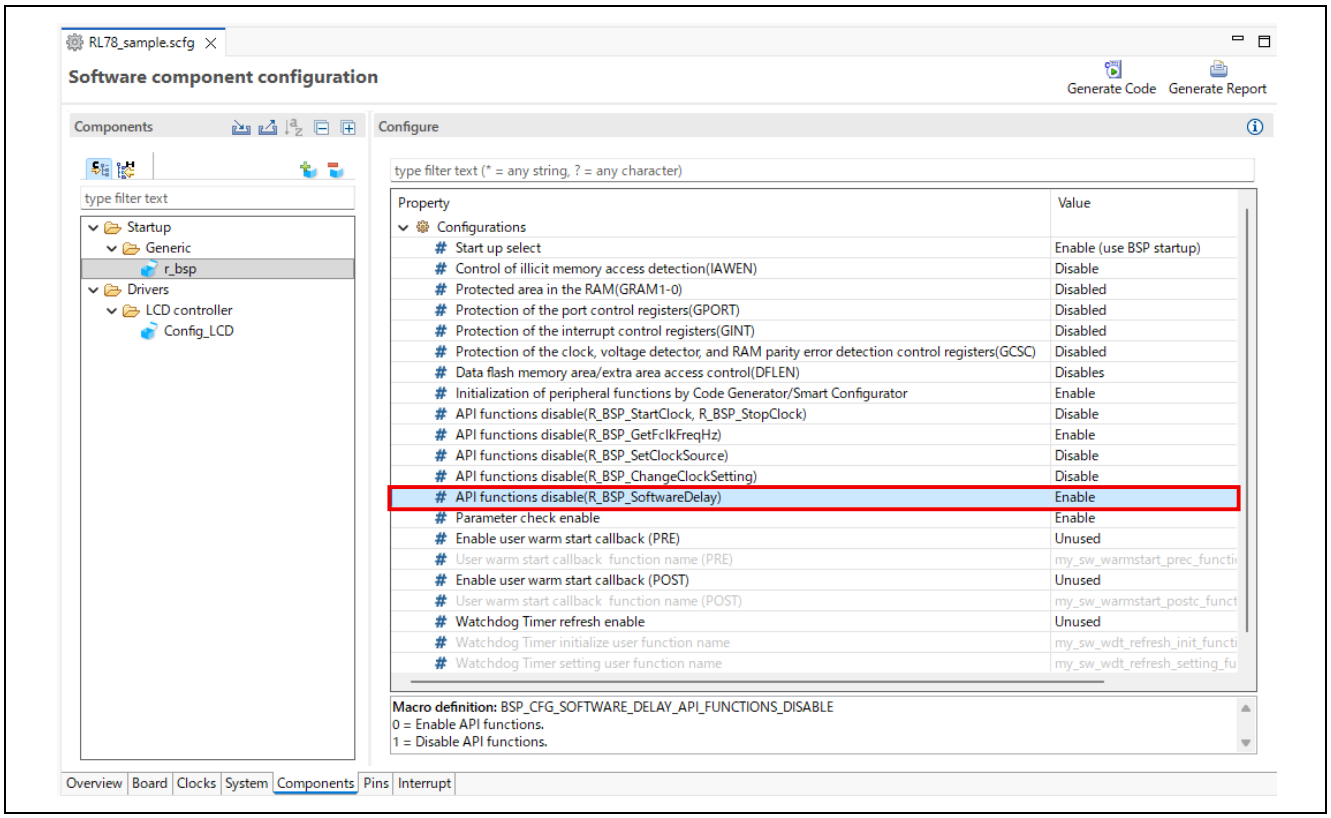
## 8.3 Notes on sample code build error

The sample code will have build error “undeclared function R_BSP_SoftwareDelay” if R_BSP_SoftwareDelay is disabled on BSP.

[Workaround]

As shown in Figure 8-1, set the "API functions disable(R_BSP_SoftwareDelay)" item to “Enable” on the “Components” tab of the Smart Configurator, generate the code, and then build the project again.

**Figure 8-1 Enable R_BSP_SoftwareDelay function**



## 8.4 Notes on qe_breakpoint_trap calling

Please don't remove the calling of qe_breakpoint_trap in R_SLCDC_Set_Pattern, or [Set checkpoint to the API function generated by QE for Segment LCD] in Configure Stop Condition Dialog cannot work normally.

**Figure 8-2 Calling of qe_breakpoint_trap**

qe_slcdc.c

```
}

static int qe_breakpoint_trap(void);

#pragma noinline qe_breakpoint_trap
static int qe_breakpoint_trap(void){
    volatile int a = 0; //To prevent the compiler from optimizing this function away
    return a;
}

* Function Name: R_SLCDC_Set_Pattern
MD_STATUS R_SLCDC_Set_Pattern(uint16_t component_id, uint16_t pattern_index) {
    if(component_id >= QE_SLCDC_NUM_COMPONENT)
    {
        return MD_ARGERROR;
    }

    const st_component * component = get_component(component_id);
    if(pattern_index >= component->p_patterns->patterns_length)
    {
        return MD_ARGERROR;
    }

    uint16_t segment_len = component->segments_length;
    uint16_t pattern_width = ((segment_len + 7) / 8);
    const uint8_t * byte_index = component->p_patterns->p_pattern_datas + pattern_index * pattern_width;
    uint16_t bit_offset = pattern_width * 8 - segment_len;

    for (uint16_t segment_index = 0; segment_index < segment_len; segment_index++) {
        uint16_t bit_index = segment_index + bit_offset;
        uint16_t byte_offset = bit_index / 8;
        uint8_t onoff = *(byte_index + byte_offset) >> (7 - (bit_index % 8));
        onoff &= 0x01;

        set_display_data(component->p_segments[segment_index], onoff);
    }

    qe_breakpoint_trap();  //Please don`t remove this calling of qe_breakpoint_trap function, or QE checkpoint cannot work normally.

    return MD_OK;
}
```

## Revision History

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.00 | Jun.30.26 | - | First edition issued. |
| | | | |

# 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/.