RL78 Family

Capacitive Touch Low Power Application Development using SMS

Introduction

This application note describes the procedure required to create a capacitive touch low power application using the RL78 SNOOZE mode sequencer (SMS).

Automatic judgment measurement using the SMS can be used to achieve low power touch applications.

Target Device

RL78/G22

RL78/G23

Contents

1. System Overview

The RL78 Family uses the SNOOZE Mode Sequencer (SMS) to implement the auto judgement measurement operation of the capacitive touch function (CTSU2La). This application note describes the procedure required to create the auto judgement measurement using the SMS in the following two sections.

- Creating a project with the Smart Configurator using the RL78 Family CPU board
- Creating a touch interface and tuning with QE for the Capacitive Touch

2. Related Documents

This application note introduces the brief procedure required to create a workable application. For questions regarding tools used in the application example in this document or for more detailed instructions, refer to the following documents: e^2 studio / Smart Configurator, Software Integration System (SIS) Driver / Middleware, help function for Renesas Code Generator and QE for Capacitive Touch (included in e^2 studio help).

3. Auto Judgement Measurement using SNOOZE Mode Sequencer (SMS)

3.1 Module Flow for Measurements using SMS

The module flows used for SMS measurement differ for RL78/G22 and RL78/G23.

- RL78/G22 module flow
  CTSU2La uses DTC for port output.[Note] The signal output from the port is used to generate an external interrupt signal. The interrupt signal triggers the ELC to start SMS processing.

  Note. The port register (Pxx) is rewritten in 8-bit units by DTC transfer.
  Therefore, the port register (Pxx) that is the destination of DTC transfer cannot be used by other functions during auto judgement measurement by the SMS. Use a port register (Pxx) that no other system will be using.

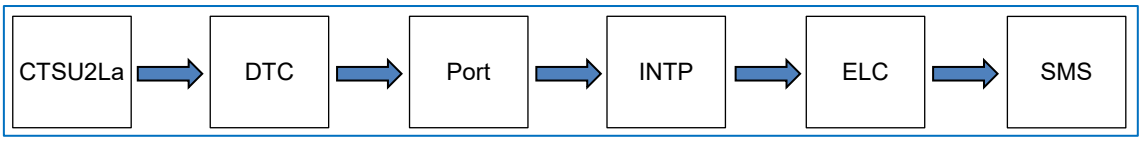


Figure 3-1 Flow of Modules Used During SMS Measurement (RL78/G22)

- RL78/G23 module flow
  CTSU2L triggers ELCL to start SMS processing.

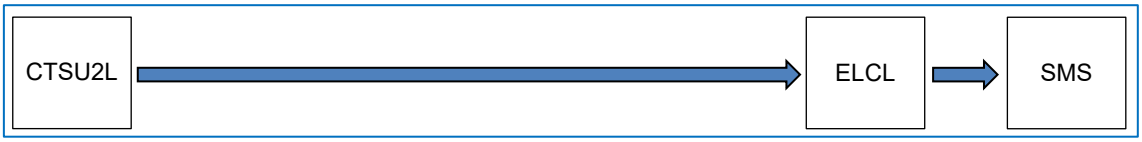


Figure 3-2 Flow of Modules Used During SMS Measurement (RL78/G23)

3.2 External Pin Wiring (RL78/G22 only)

When performing auto judgement measurement using the SMS with RL78/G22, connect a port pin (Pxx) used by the modules in the flow above and an external interrupt pin (INTPxx). This step is not required for the RL78/G23.

The following describes the wiring when using the RL78/G22 capacitive touch evaluation system (product model: RTK0EG0042S01001BJ).

Connect CN2 pin 32 (P22/TS20) and pin 16 (P16/INTP5/TS17) on the CPU board as shown in the figure below.

P22: Pin used for port output from CTSU2La using DTC.
INTP5: Pin that generates an interrupt signal using a signal output from the port pin.

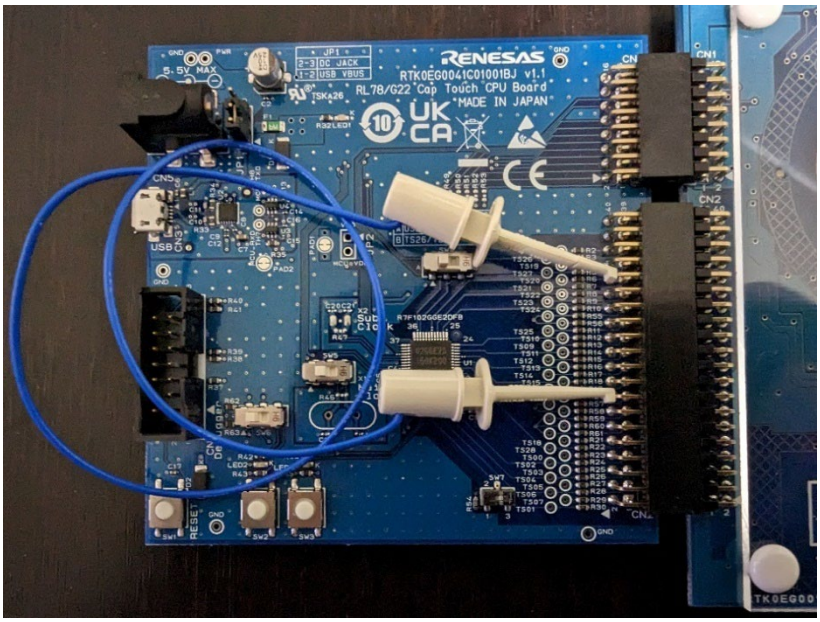


Figure 3-3 Wiring Pattern Using SMS on RL78/G22 Hardware

4. Peripheral Functions

Table 4-1 and Table 4-2 show peripheral functions used in the sample code.

Table 4-1 Peripheral Functions and Usage (RL78/G22)

Peripheral Function	Usage
Capacitive Sensor Unit (CTSU2La)	Measures the capacitance generated on the touch electrode.
32-bit Interval Timer (TML32)	Timer (measurement period: 20ms) to release the STOP mode and transit to the SNOOZE mode.
Data Transfer Controller (DTC)	Outputs a signal from a port using DTC. Generates an interrupt signal using the signal output from the port. Triggers ELC with the interrupt signal to start SMS processing.
Port	
Interrupt Controller (INTP)	
Event Link Controller (ELC)	
SNOOZE Mode Sequencer (SMS)	

Table 4-2 Peripheral Functions and Usage (RL78/G23)

Peripheral Function	Usage
Capacitive Sensor Unit (CTSU2L)	Measures the capacitance generated on the touch electrode.
32-bit Interval Timer (TML32)	Timer (measurement period: 20ms) to release the STOP mode and transition to the SNOOZE mode.
Logic & Event Link Controller (ELCL)	Triggers ELCL with measurement data transfer request (INTCTSURD) to start SMS processing.
SNOOZE Mode Sequencer (SMS)	

5. Operation Confirmation Conditions

Table 5-1 shows the peripheral environment used in the project.

Table 5-1 Operation Confirmation Conditions

Item	Description
MCU	RL78/G22(R7F102GGE2DFB)
	RL78/G23 (R7F100GSN2DFB)
Operating voltage	3.3V LVD0 detection voltage: Rising edge: TYP. 2.67V (2.59V to 2.75V) Falling edge: TYP. 2.62V (2.54V to 2.70V)
Operating frequency	High-speed on-chip oscillator clock (f_{IH}): 32 MHz Low-speed on-chip oscillator clock (f_{IL}): 32.768 kHz
Target board	RL78/G22 Capacitive touch evaluation system (Product model: RTK0EG0042S01001BJ)
	RL78/G23 Capacitive touch evaluation system (Product model: RTK0EG0030S01001BJ)
Integrated development environment	e^2 studio (2024-04)
Smart Configurator	V24.4.0
C compiler	CC-RL V1.13.00 Optimization level option: -Odefault
Development assistance tool for capacitive touch sensors	QE for Capacitive Touch V3.4.0
Emulator	E2 emulator Lite (RTE0T0002LKCE00000R)

6. Software Settings

6.1 Option Byte Setting

Table 6-1 and Table 6-2 show the option byte settings.

Table 6-1 Option Byte Settings (RL78/G22)

Address	Setting Value	Description
000C0H / 020C0H	1110 1111B (0xEF)	Disables the watchdog timer. (Counting stopped after reset)
000C1H / 020C1H	1111 1100B (0xFC)	LVD0 detection voltage: Reset mode Rising edge: 2.67V (TYP) (2.59V to 2.75V) Falling edge: 2.62V (TYP) (2.54V to 2.70V)
000C2H / 020C2H	1110 1000B (0xE8)	HS (high-speed main) mode, High-speed on-chip oscillator clock (fIH): 32 MHz
000C3H / 020C3H	1000 0100B (0x84)	Enables on-chip debugging

Table 6-2 Option Byte Settings (RL78/G23)

Address	Setting Value	Description
000C0H / 040C0H	1110 1111B (0xEF)	Disables the watchdog timer. (Counting stopped after reset)
000C1H / 040C1H	1111 1100B (0xFC)	LVD0 detection voltage: Reset mode Rising edge: 2.67V (TYP) (2.59V to 2.75V) Falling edge: 2.62V (TYP) (2.54V to 2.70V)
000C2H / 040C2H	1110 1000B (0xE8)	HS (high-speed main) mode, High-speed on-chip oscillator clock (fIH): 32MHz
000C3H / 040C3H	1000 0100B (0x84)	Enables on-chip debugging

To confirm the option byte setting, open the project property after the code is generated, then select [**C/C++Build**] - [**Settings**] - [**Tool Settings**] - [**Linker**] - [**Device**]. The setting can be confirmed in “User option byte value” and “On-chip debug control value”.

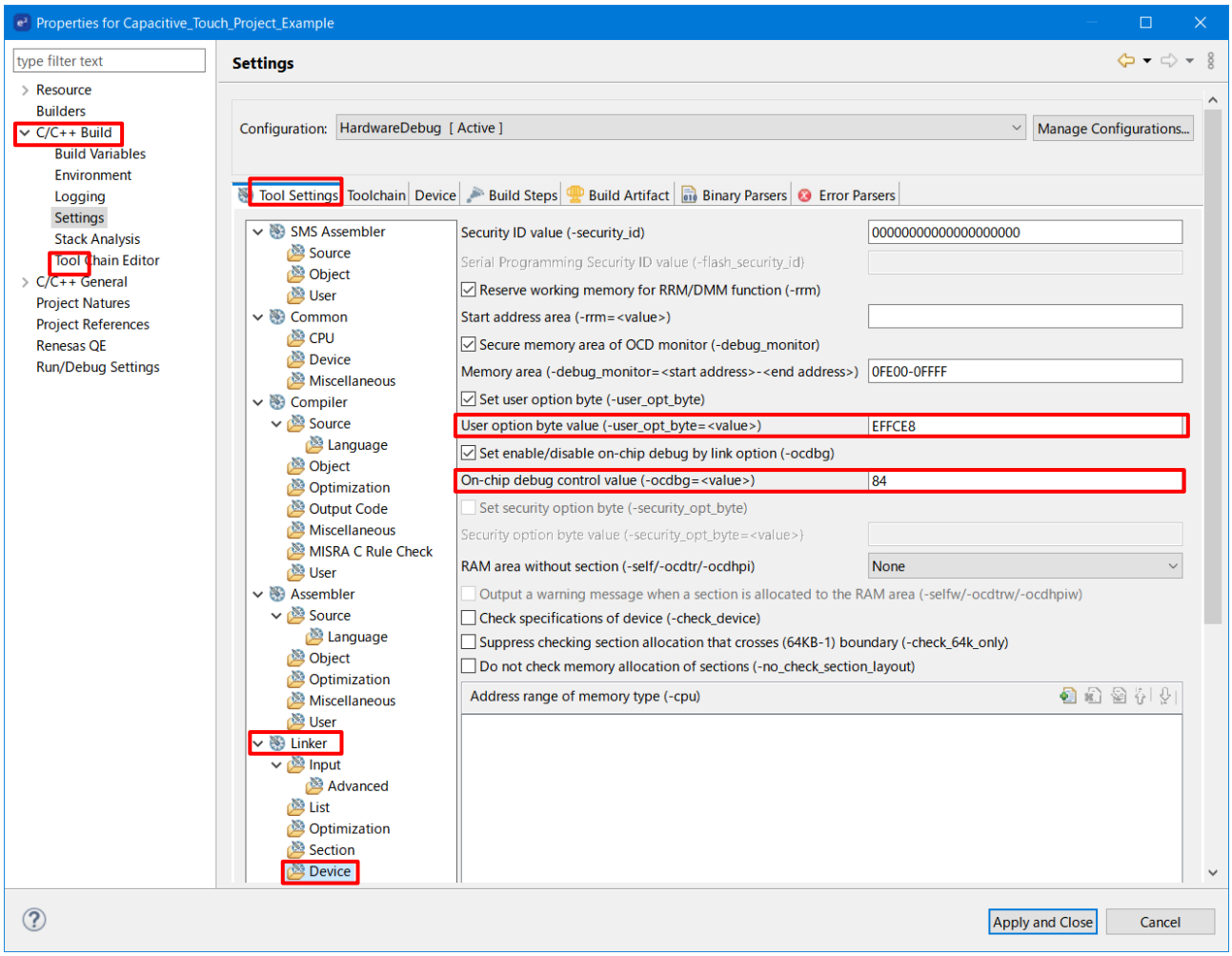


Figure 6-1 Setting of Option Bytes

6.2 Capacitive Touch Settings

This section describes the capacitive touch settings used in the application example. With RL78/G22, using the multiple electrode connection (MEC) function of the capacitive touch sensor (CTSU2La) with the SMS enables operation with even less power consumption than touch measurement operation only with the SMS.

*As the RL78/G23 embeds capacitive touch sensor CTSU2L, it does not include the MEC function.

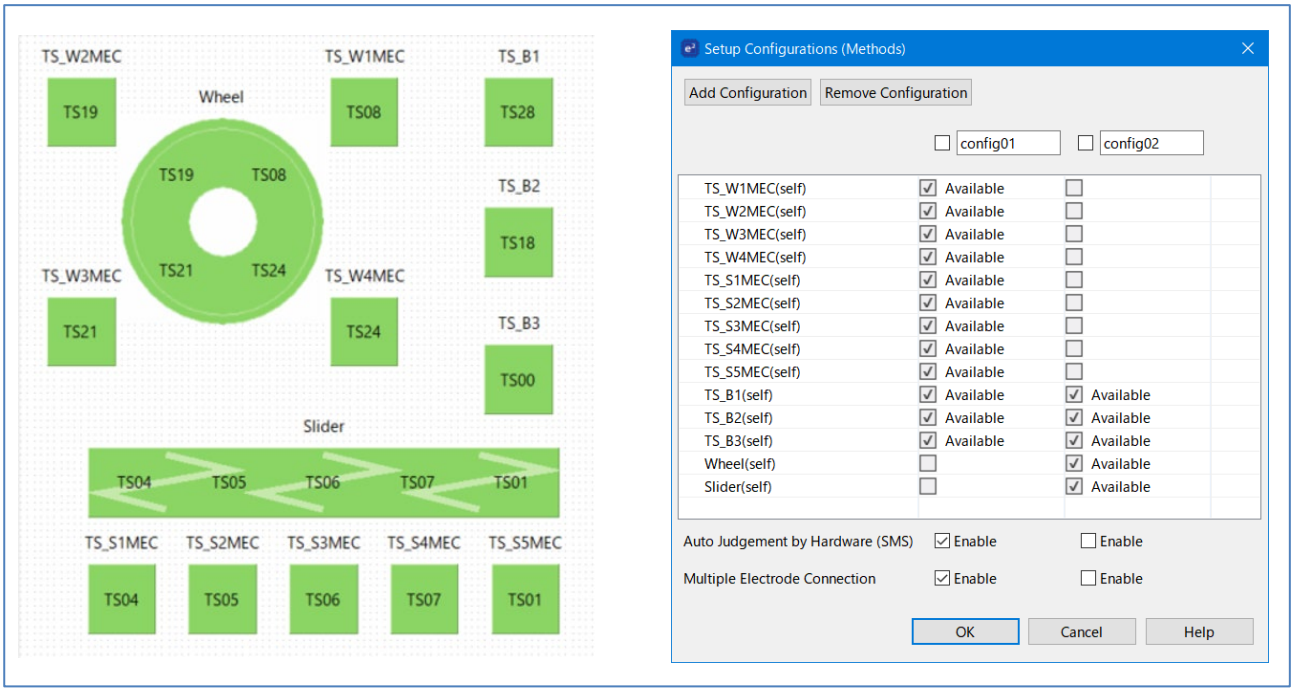


Figure 6-2 Capacitive Touch Setting for Application Example (RL78/G22)

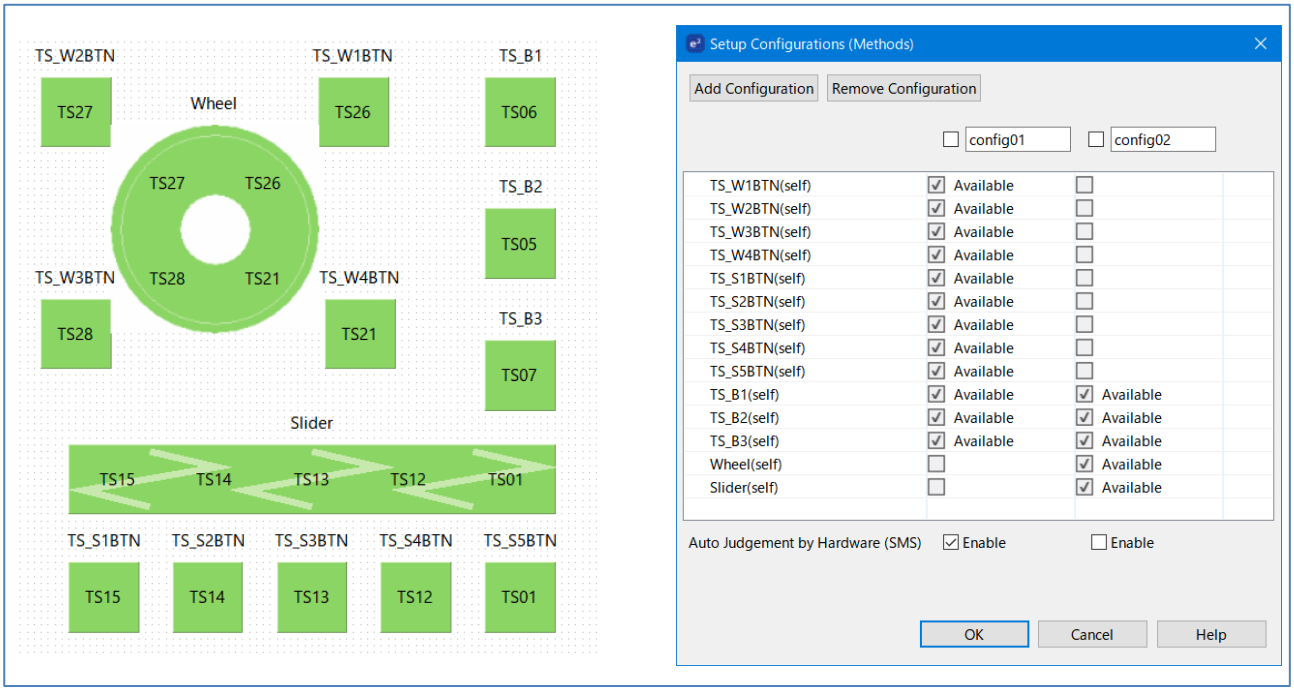


Figure 6-3 Capacitive Touch Settings for Application Example (RL78/G23)

6.3 Components

Figure 6-3 and Figure 6-4 show components and modules set by the Smart Configurator.

Component	Version	Configuration
Board Support Packages. - v1.62 (r_bsp)	1.62	r_bsp(used)
Capacitive Sensing Unit driver. (r_ctsu)	1.50	r_ctsu(used)
Interrupt Controller	1.4.0	Config_INTC(INTC: used)
Interval Timer	1.4.0	Config_ITL000(ITL000: used)
Ports	1.4.1	Config_PORT(PORT: used)
Touch middleware. (rm_touch)	1.50	rm_touch(used)
Voltage Detector	1.3.0	Config_LVD0(LVD0: used)

Figure 6-4 Components and Modules Set by Smart Configurator (RL78/G22)

Component	Version	Configuration
Board Support Packages. - v1.62 (r_bsp)	1.62	r_bsp(used)
Capacitive Sensing Unit driver. (r_ctsu)	1.50	r_ctsu(used)
Interval Timer	1.4.0	Config_ITL000(ITL000: used)
Touch middleware. (rm_touch)	1.50	rm_touch(used)
Voltage Detector	1.3.0	Config_LVD0(LVD0: used)

Figure 6-5 Components and Modules Set by Smart Configurator (RL78/G23)

7. Application Example Overview

The following describes the implementation of the application example's main loop.

===== Touch measurement (low power mode) using SMS and MEC[Note] ==========

1. Execute the RM_TOUCH_SmsSet function to specify the SMS after initial offset tuning.
2. Execute the RM_TOUCH_ScanStart function to specify the touch measurement setting, enable the SNOOZE function, and transition to the external trigger wait state.
3. Start TML32 timer count (measurement period: 20ms).
4. Execute the STOP instruction to transition to the STOP mode.
5. When a TML32 interrupt request occurs, an external trigger from the ELC (ELCL for G23) transitions the program to the SNOOZE mode and starts touch measurement.).
6. When the measurement end interrupt occurs, touch ON/OFF judgement is performed by SMS in the SNOOZE mode.
7. When the result of the judgement is touch ON, transition to the normal mode is activated; when touch OFF, the processing goes back to Step 4 above.

=====Touch measurement for button/slider/wheel (normal mode) ==========

8. Execute the RM_TOUCH_ScanStart function to enter the external trigger wait state.
9. Start TML32 timer count (measurement period: 20ms).
10. When the TML32 timer counts 20ms, an external trigger from the ELC (ELCL for G23) starts touch measurement.
11. When the measurement is completed, touch ON/OFF judgement is performed. The processing goes back to Step 1 above regardless of the result of the judgement.

Note: The RL78/G23 uses the SMS for touch measurement (low power mode).

8. Capacitive Touch Application Development Procedure

This section describes procedures required to integrate touch sensor detection into your project. These procedures are applicable to general user application development.

- Create a new project using the e^2 studio wizard.
- Add necessary modules to the created project using the Smart Configurator.
- Create a capacitive touch interface using QE for Capacitive Touch.
- Tune the project using QE for Capacitive Touch.
- Add API calls for necessary SIS modules to the project and enable capacitive touch control.

Procedures for each step above are described in the following subsections.

Unless otherwise specified, the setting procedures described are for RL78/G22.

8.1 Creating a Project

This section describes how to create a new project using the e² studio wizard.

1. Start e² studio from the Windows start menu or the shortcut on the desktop. When the dialog box appears, create a workspace in a location of your choice.

2. Select [**File**] - [**New**] - [**C/C++ Project**] in e² studio to start the new project.

3. When the dialog box appears, select “**Renesas RL78**” - ”**Renesas CC-RL C/C++ Executable Project**”, then click [**Next**].

4. In the dialog box appeared, enter your project name into the “Project name (P):” field and click [**Next**]. The project name in the application example is ”Capacitive_Touch_Project_Example”.

5. Specify the following items as shown below in the next dialog box.

- Language: C
- Toolchain: Renesas CC-RL
- Toolchain Version: v1.13.00
- Target Board: Custom
- Target Device: RL78/G22 (R7F102GGExFB)
  RL78/G23 (R7F100GSNxFB)
- Configurations: Check “**Create Hardware Debug Configuration**”, then select “E2 Lite (RL78)” from the pulldown menu.

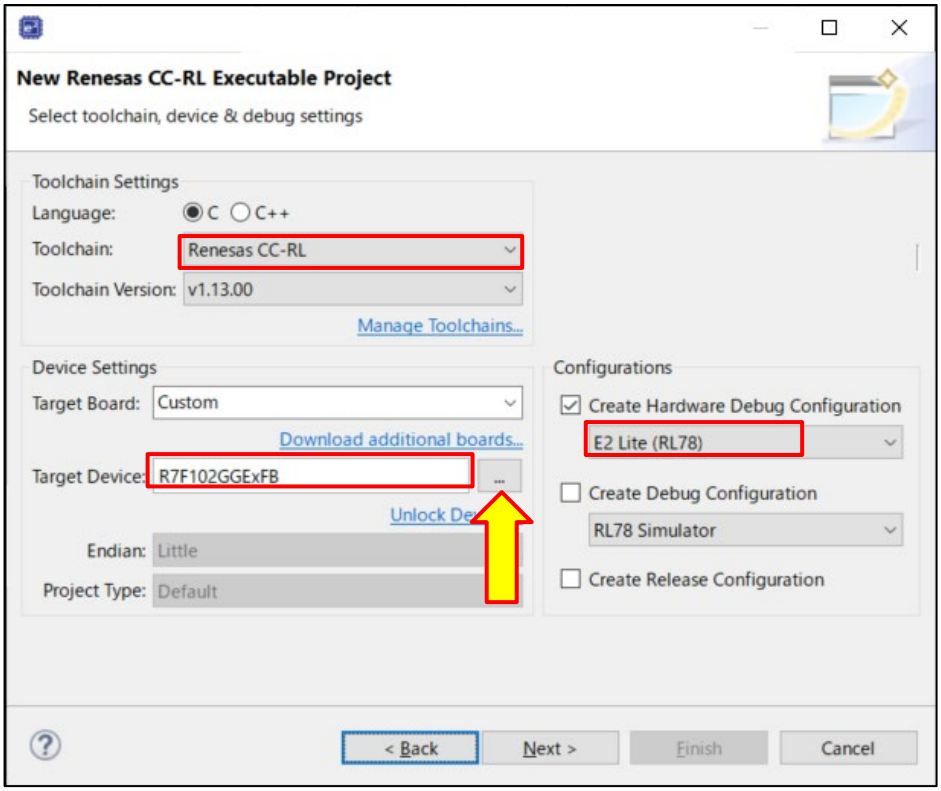


Figure 8-1 Toolchain, Device, and Debug Settings

To select the target device, click [⋯] next to the “Target Device” field. Select the device from the list shown in the "Device Selection" window.

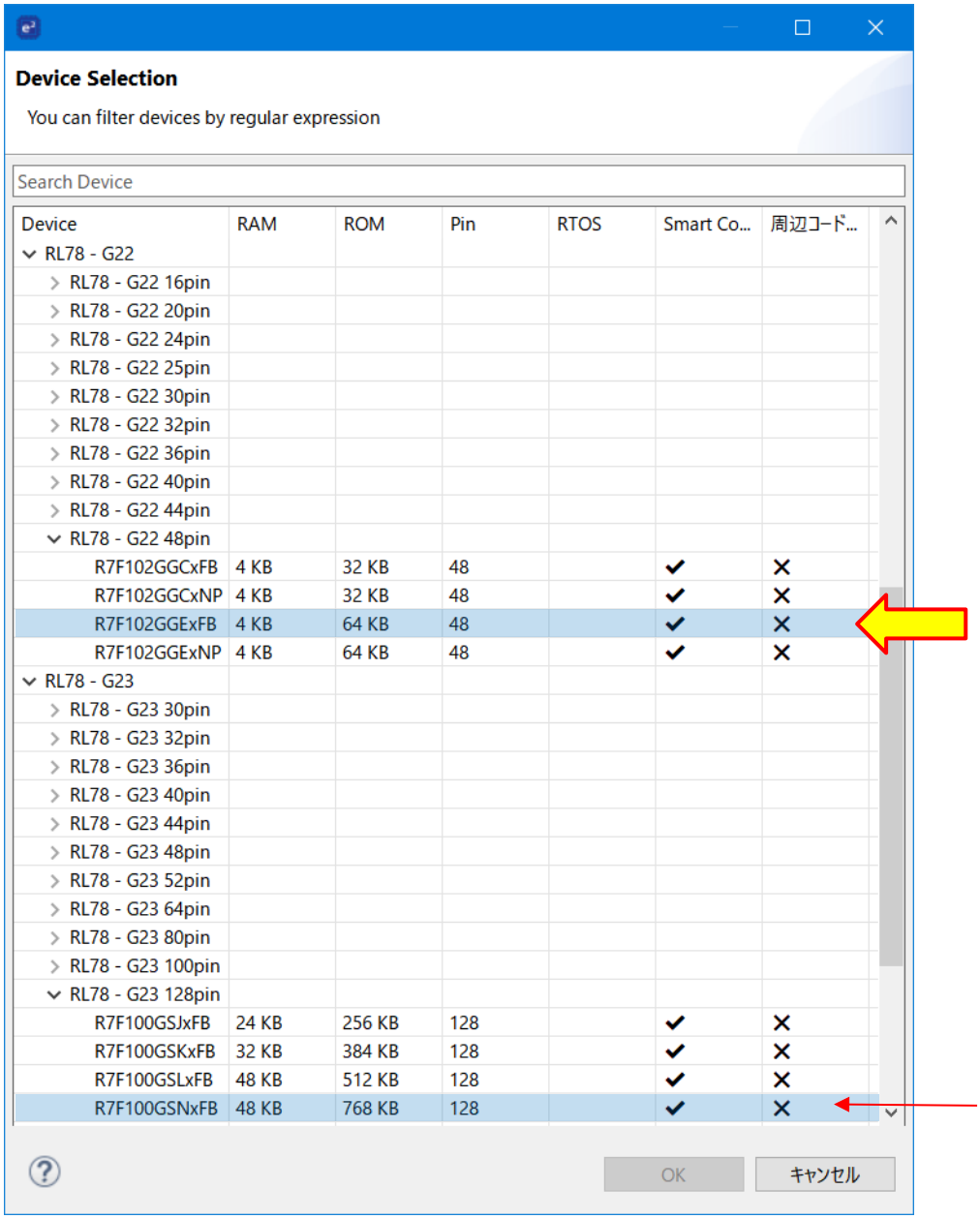


Figure 8-2 Target Device Selection

Remark: When confirming the operation with RL78/G23 RSSK (RTK0EG0030S01001BJ), select “R7F100GSNxFB” as the target device.

6. After completing your selections, click [**OK**].

7. When the next dialog box appears, check “**Use Smart Configurator**”, then click [**Finish**].

When the settings are completed, the Smart Configurator perspective will appear in the e² studio default window. The new project is now created and ready for configuration.

8.2 Creating a Module with the Smart Configurator

8.2.1 Adding Components with the Smart Configurator

This section describes how to add source files of the necessary modules using the Smart Configurator.

Some configurations differ between RL78/G22 and RL78/G23.

1. Select the [**Clocks**] tab at the bottom of e^2 studio and specify the clock setting of RL78 MCU, as shown in the figure below.
   In the application example, the low-speed on-chip oscillator is used as the low-speed peripheral clock (fSXP).
   Uncheck “XT1 oscillator” and select “Low-speed on-chip oscillator” for fSXP as shown below.

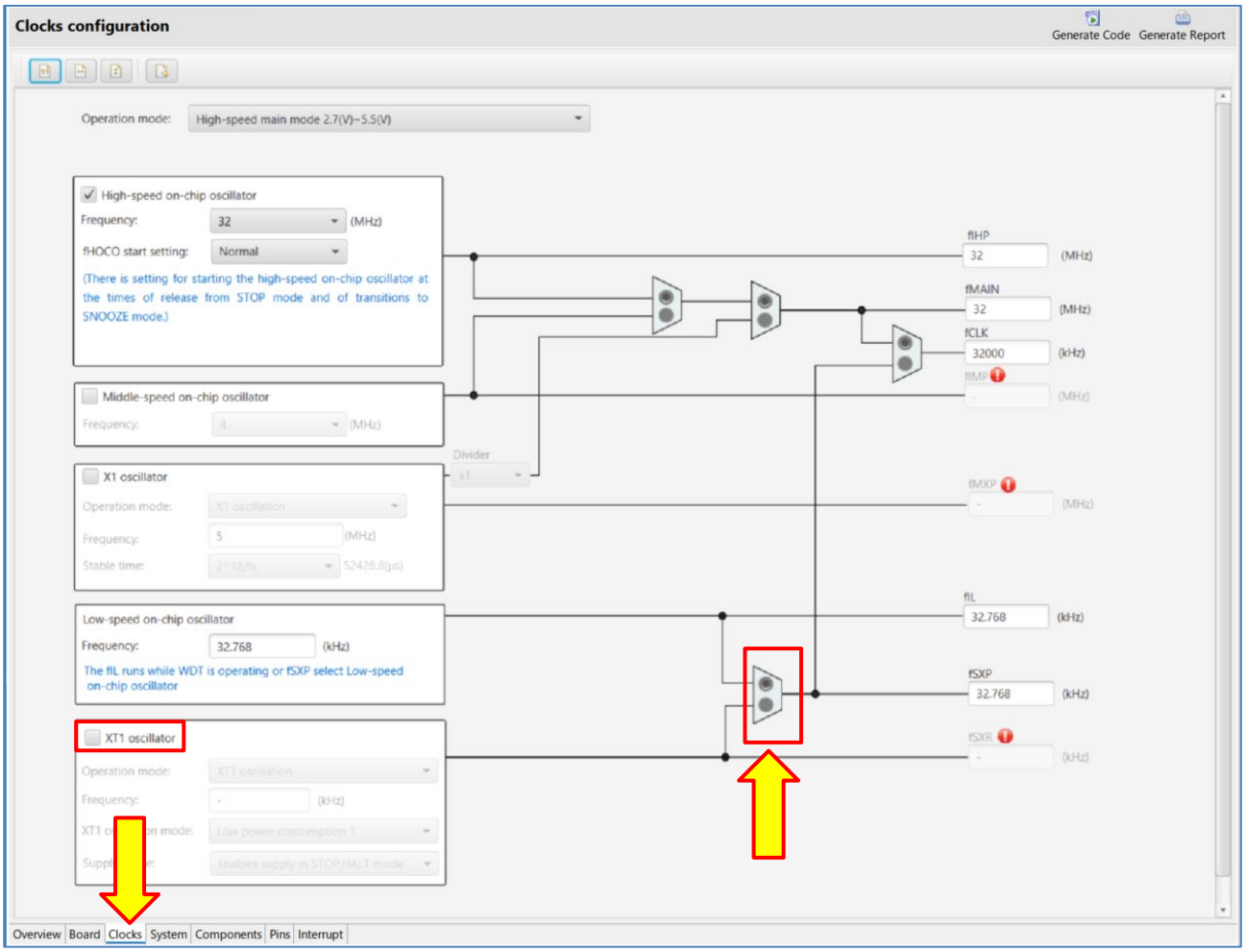


Figure 8-3 Clock Configuration

2. The application example uses the MCU in the high-speed mode (HS mode) with operation voltage between 2.7V to 5.5V. Select the appropriate “Operation mode” as shown in the figure below. When using theRL78/G23, you will also need to select the EVDD setting.

Figure 8-4 Operating Mode Setting (RL78/G22)

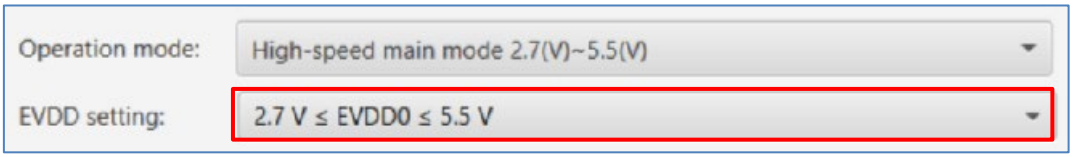


Figure 8-5 Operating Mode and EVDD Setting (RL78/G23)

3. Select the [**System**] tab.
Select "**Use emulator**” and “**E2 Lite**” and uncheck "**Use security ID**”, as shown in the figure below.

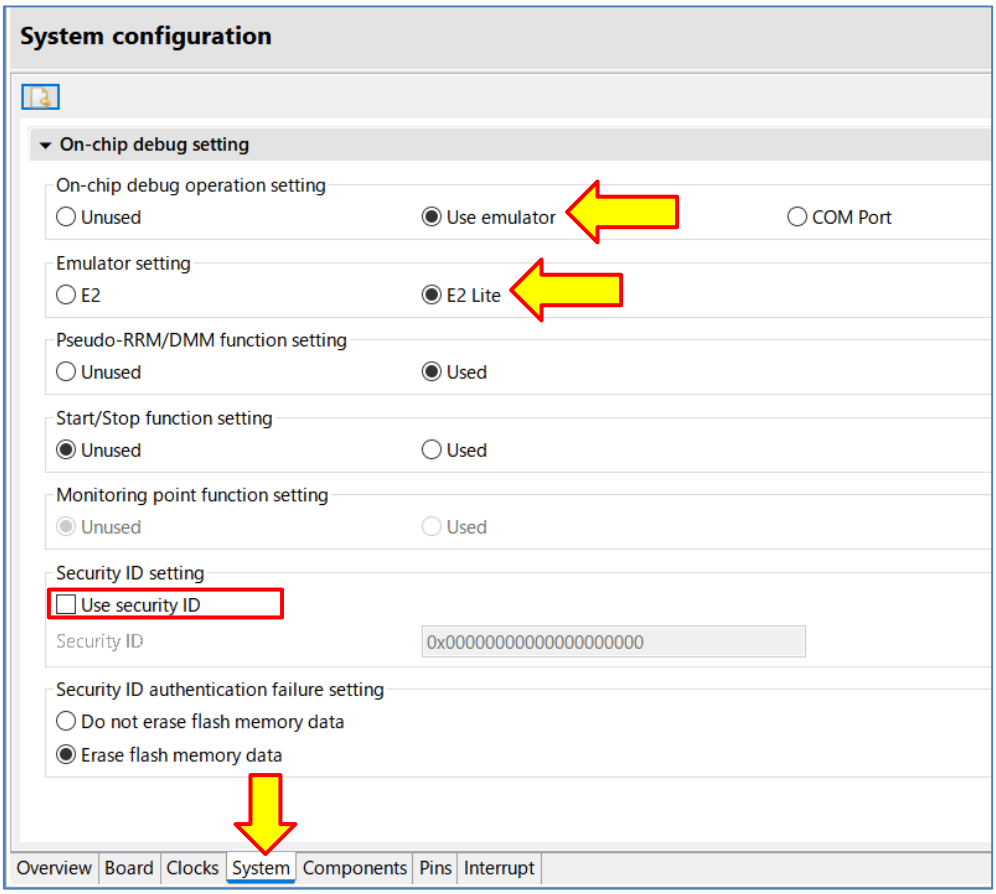


Figure 8-6 On-Chip Debug Setting

4. Select the [**Components**] tab and select and right click “r_bsp”, then click "**Change version**” to see the r_bsp version.

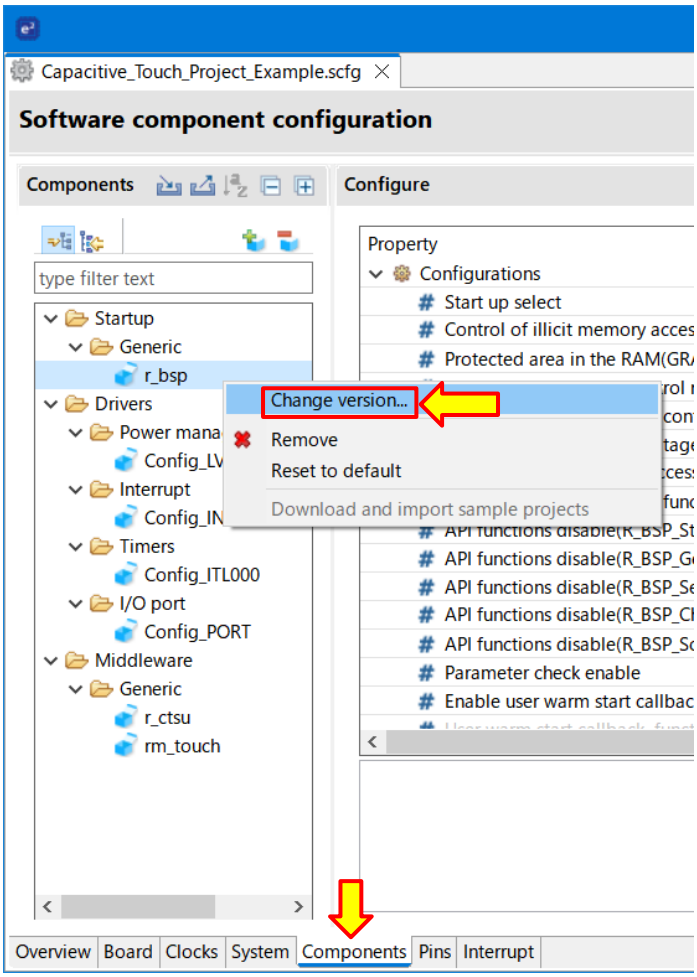


Figure 8-7 Change the r_bsp Version

When “Current version” is not “1.62” or later, select “1.62” or later version, then click [**Next**] to change the version.

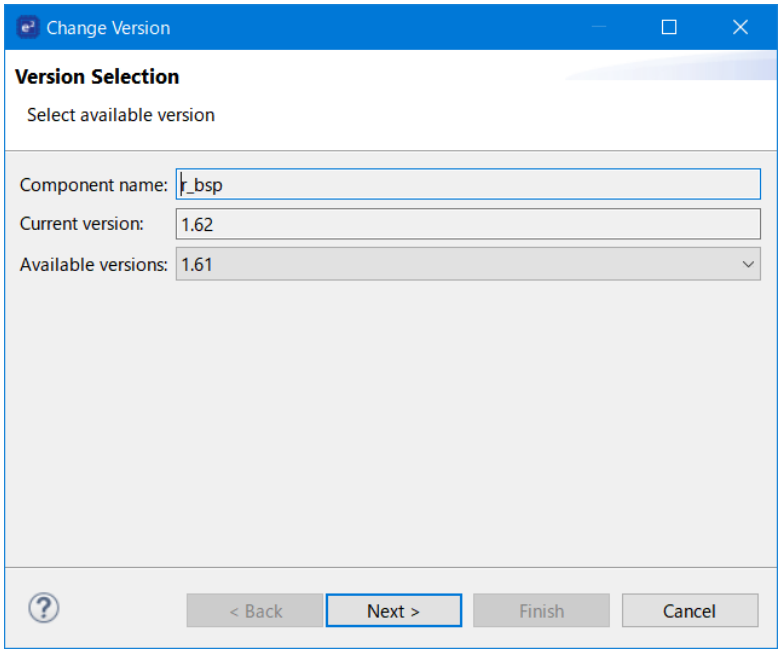


Figure 8-8 Select the r_bsp Version

5. Add the SIS modules and components to the project.
   Click the “Add” button (marked with red square in figure) and select components to be added from the list.

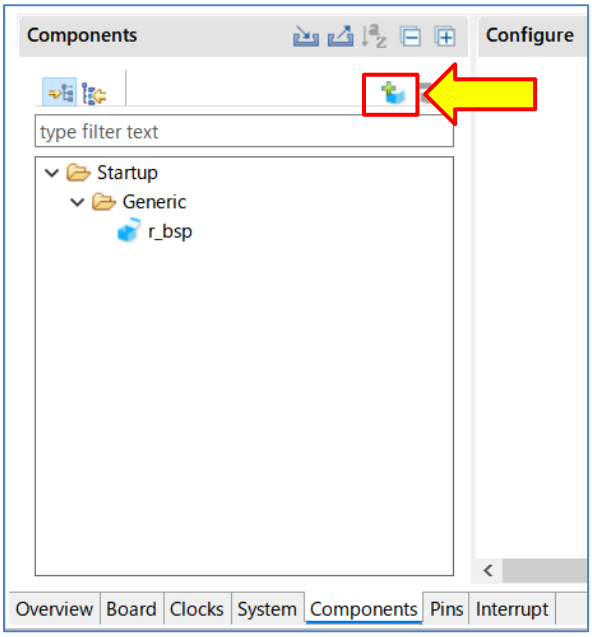


Figure 8-9 Add Components

6. The “New Component” dialog box appears with the list of available modules to add to the project.

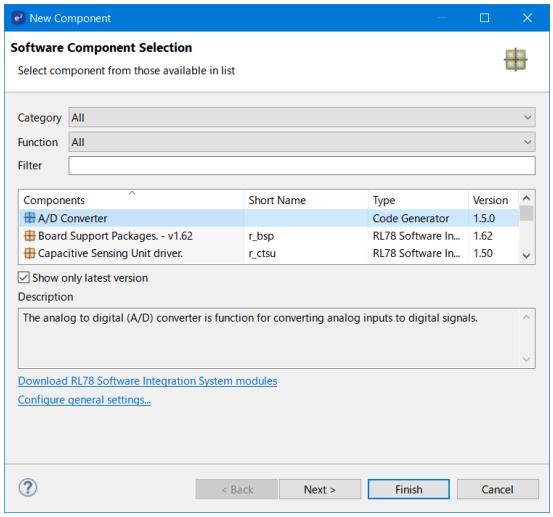


Figure 8-10 “New Component” Dialog Box

Caution: If you are using SIS (Software Integration System) modules for the first time, download the modules using steps (1) to (4) below.

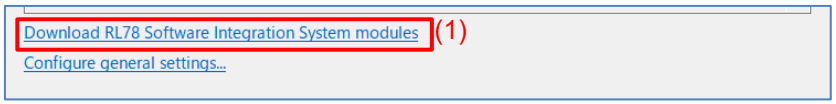


Figure 8-11 Download SIS Modules Link

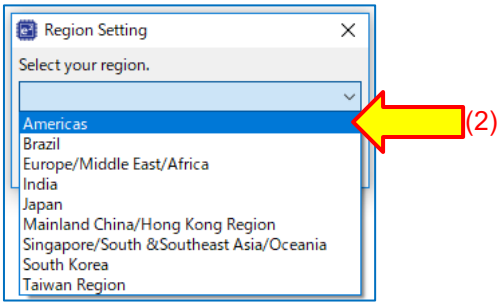


Figure 8-12 Select Region

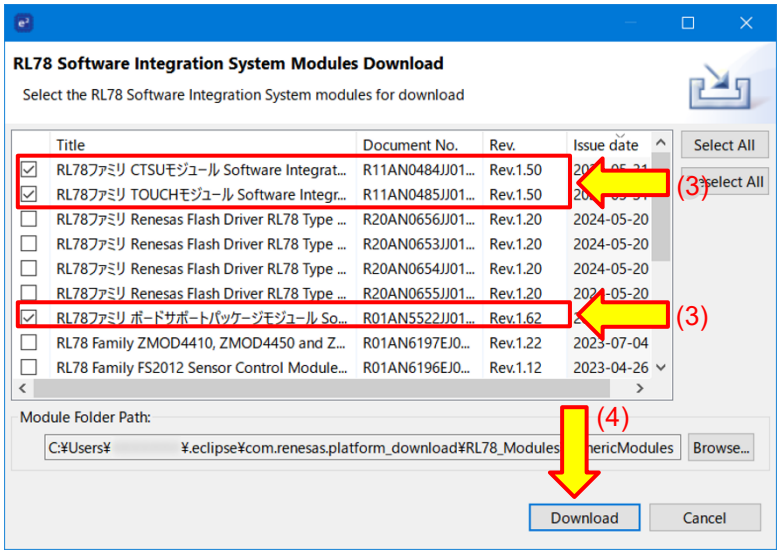


Figure 8-13 Download SIS Modules

7. Select the components shown in the figure below in the Smart Configurator.
   Note: When using the RL78/G23, “Ports” and “Interrupt Controller” are not necessary for auto judgement measurement using the SMS.

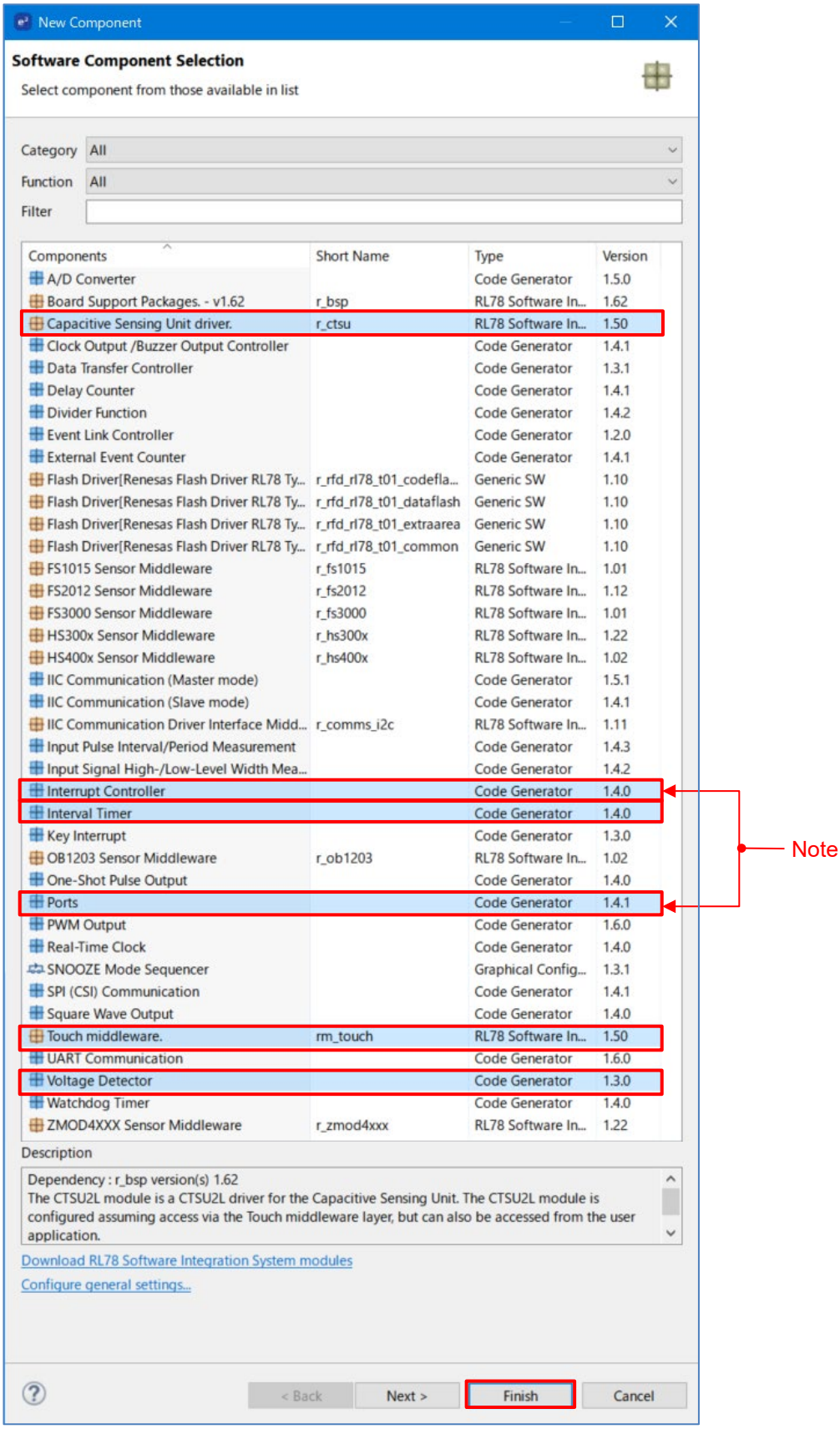


Figure 8-14 Software Component Settings

8. Specify the resources for the selected components. The settings for the application example are shown below.
   Note: When using the RL78/G23, “Ports” and “Interrupt Controller” will not appear in the dialog box below since they are not necessary for auto judgement measurement using the SMS.

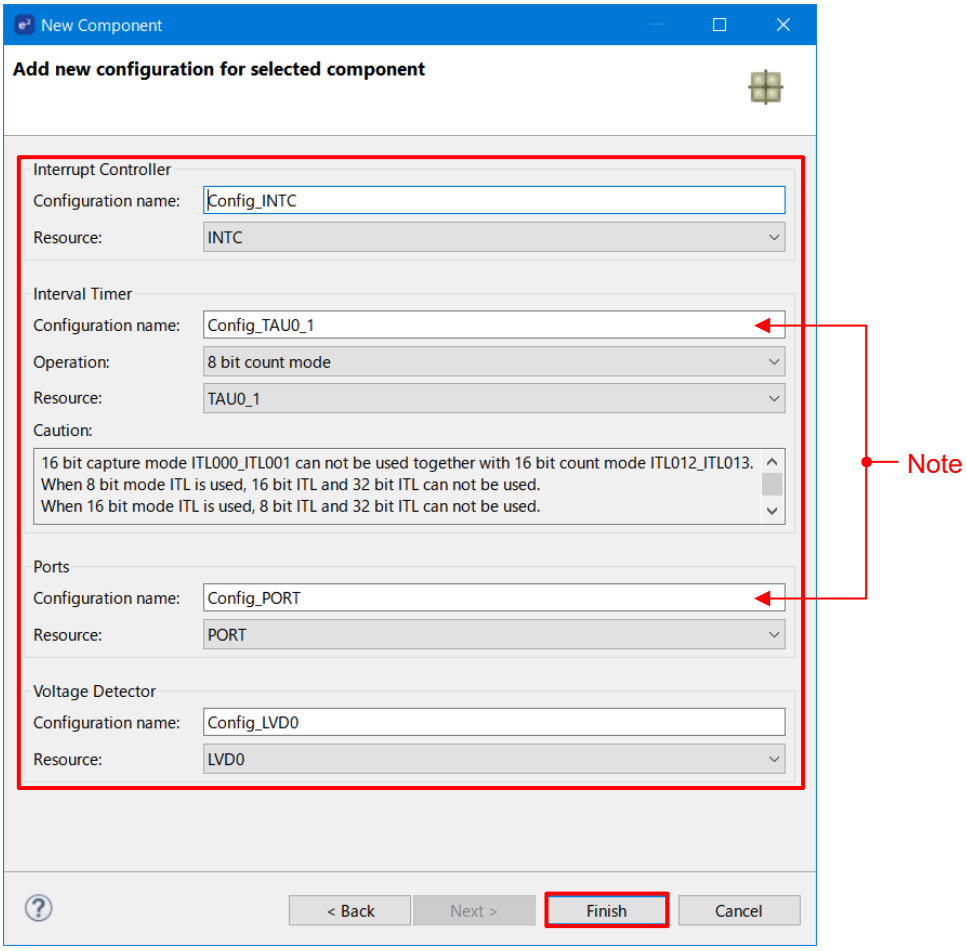


Figure 8-15 Component Resource Settings

Components are added as shown below.

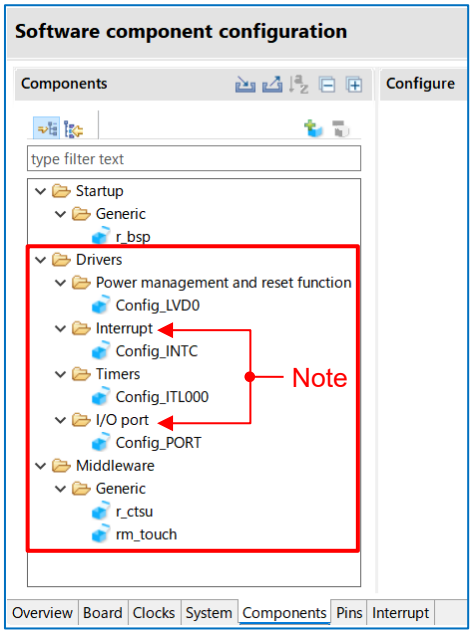


Figure 8-16 Software Component Settings (after components added)

8.2.2 Changing the Component Configuration with Smart Configurator.

Configure the added components as follows:

- CTSU Component Settings
  Note that configurations differ between RL78/G22 and RL78/G23.

  (1) Select “DTC” for “Data transfer of INTCTSUWR and INTCTSURD”.

  (2) Specify “Data storage address setting for CTSURD” according to the device used.
  RL78/G22: Set “0xFEF00”.
  RL78/G23: Set “0xFF500”.

  (3) For the RL78/G22, select the port pin to output the touch measurement complete signal transmitted from CTSU2La via the DTC. This setting is not shown for the RL78/G23 since it is not required.
  - Output port number for external trigger: Select port group.
  - Bit number for external trigger output: Select bit number from port group selected above.

  Caution: The port register (Pxx) is rewritten in 8-bit units by DTC transfer.
  Therefore, a port register (Pxx) that is the destination of DTC transfer cannot be used for other functions during auto judgement measurement by the SMS. Use a port register (Pxx) that no other system will be using.

  (4) For the RL78/G22, specify the external interrupt pin to input a signal output from the port pin. This setting is not shown for the RL78/G23 since it is not required.

  (5) Enable TS pin used for touch measurement
  Caution: For the RL78/G22, users cannot use TS pins that share functions with port pins used for automatic judgment measurement using SMS or external interrupt pins. The application example uses P22/TS20 and P16/INTP5/TS17. The user should remove checks (deselect) from pins TS17 and TS20.
  For the RL78/G23, any TS pins can be selected when using it.

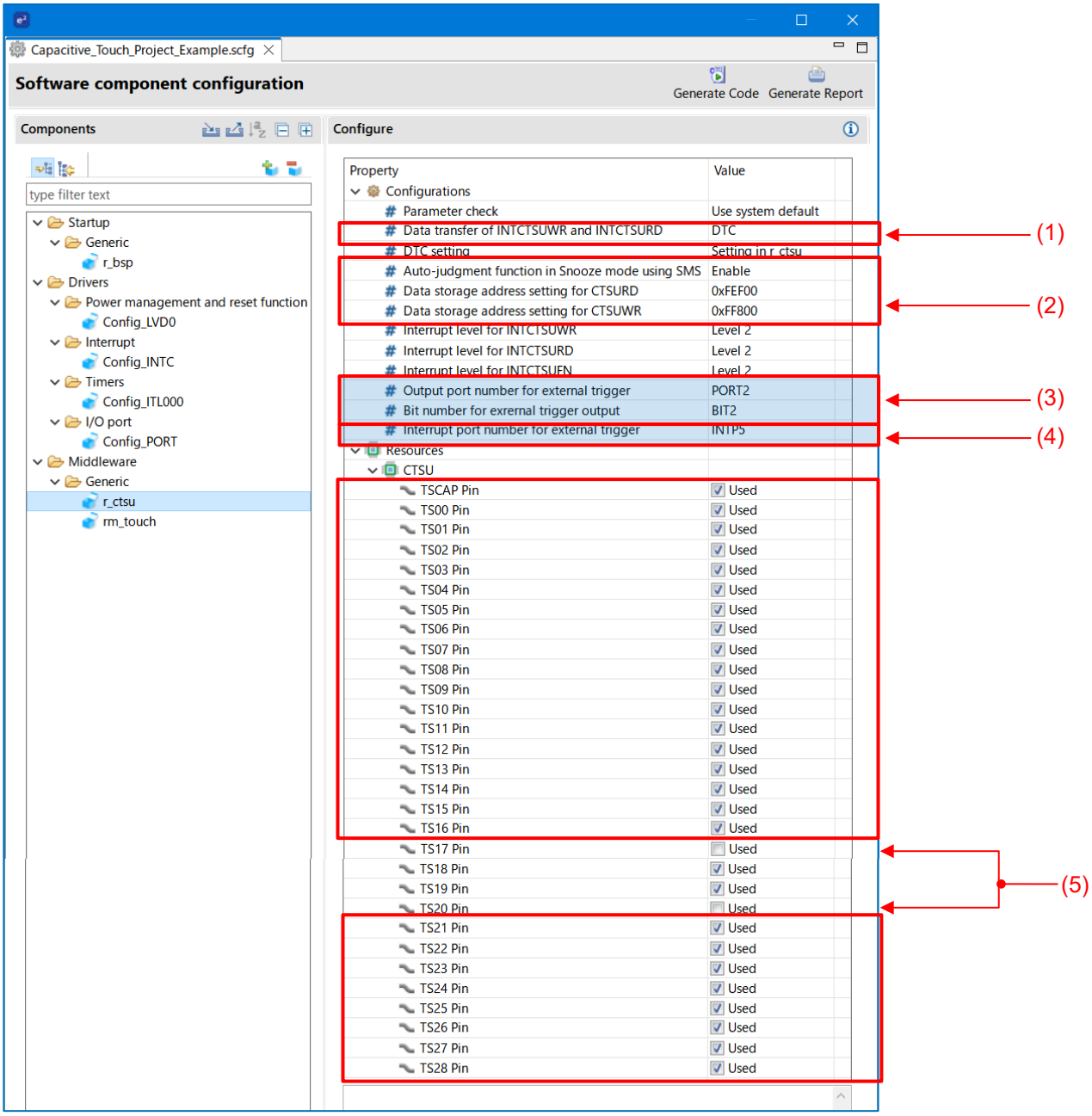


Figure 8-17 CTSU Component Settings

- Touch Component Configuration
  Use the default setting as is.

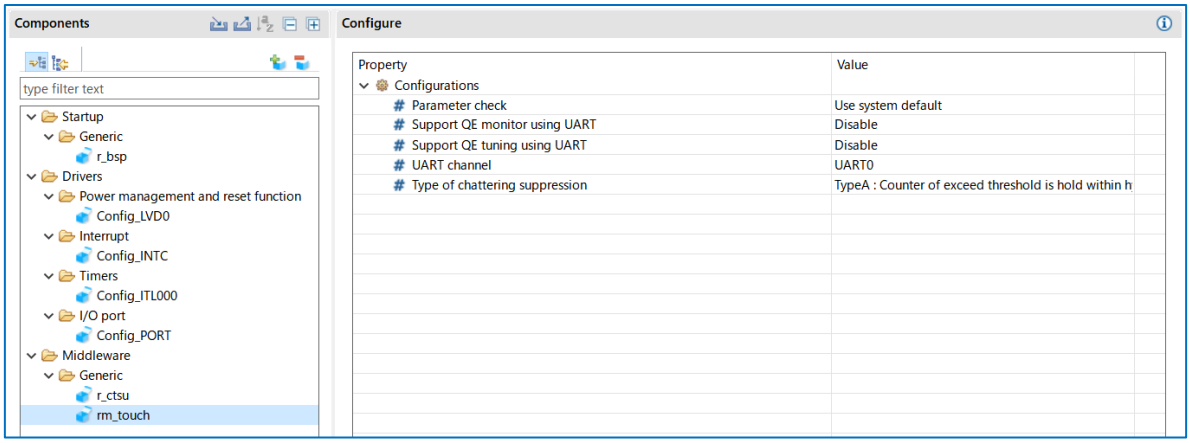


Figure 8-18 Touch Component Setting

- INTC Component Configuration
  The RL78/G23 does not require the following setting for auto judgement measurement using SMS.

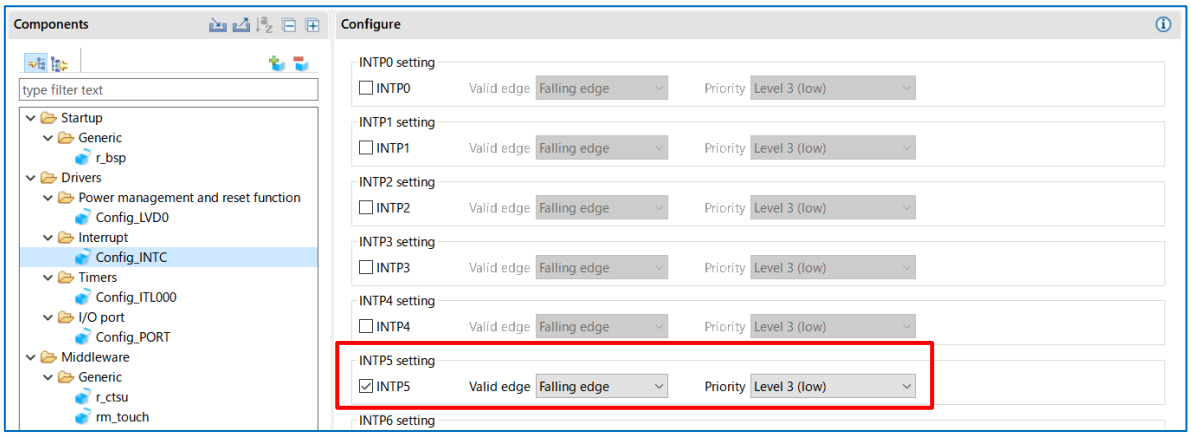


Figure 8-19 INTC Component Setting

- ITL Component Configuration

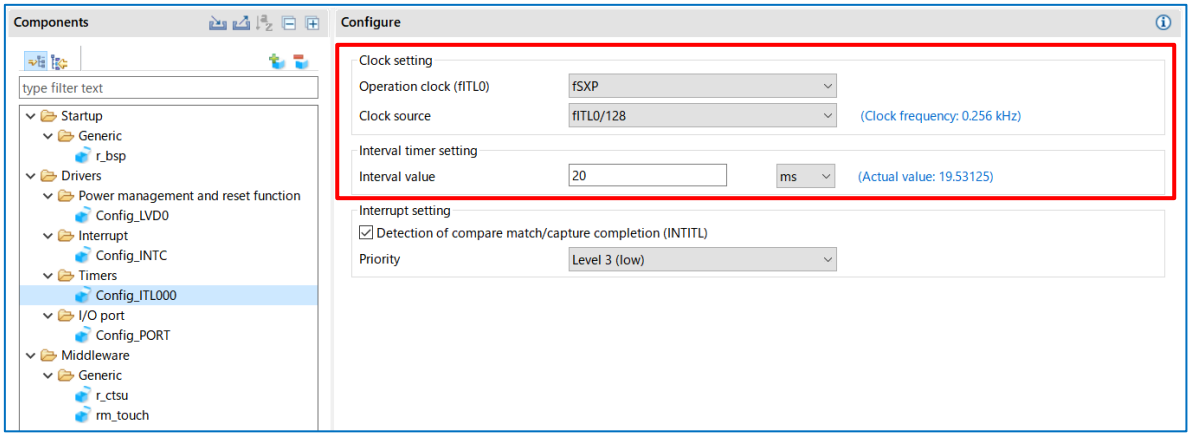


Figure 8-20 ITL Component Setting

- PORT Component Configuration
  The RL78/G23 does not require the following setting for auto judgement measurement using SMS.

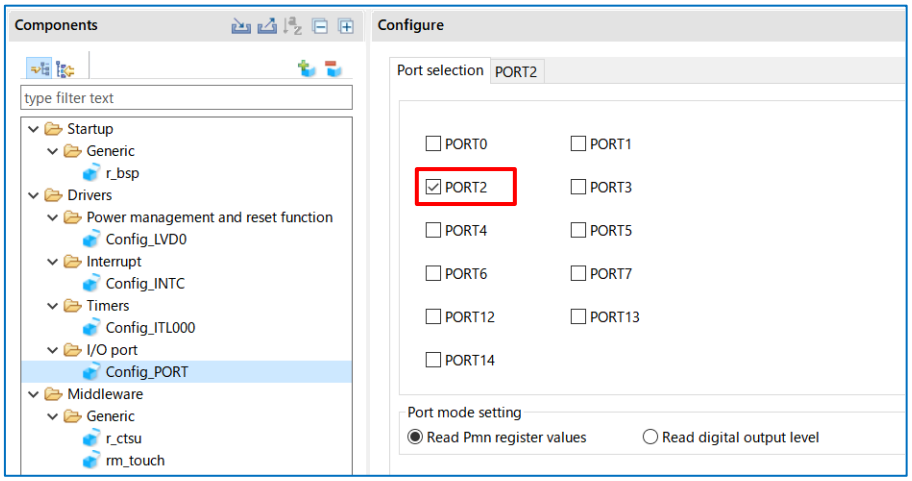


Figure 8-21 PORT Component Setting (Port selection tab)

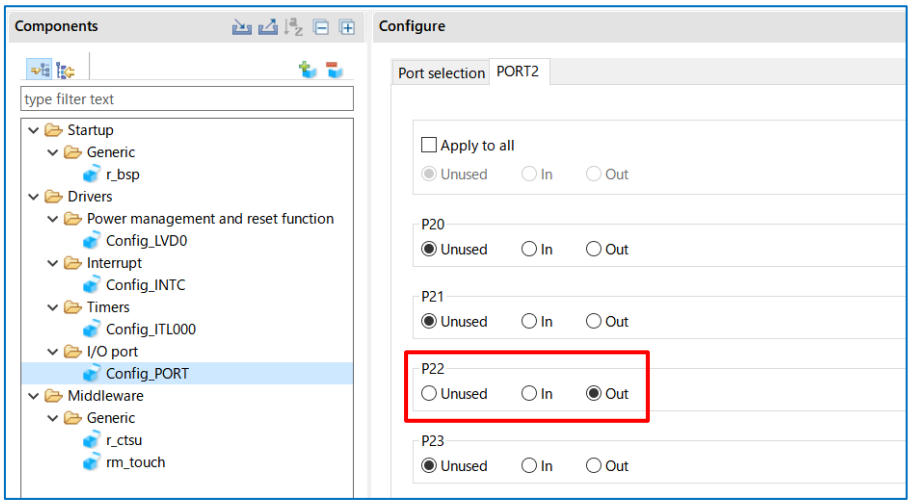


Figure 8-22 PORT Component Setting (PORT2 tab)

- LVD Component Configuration

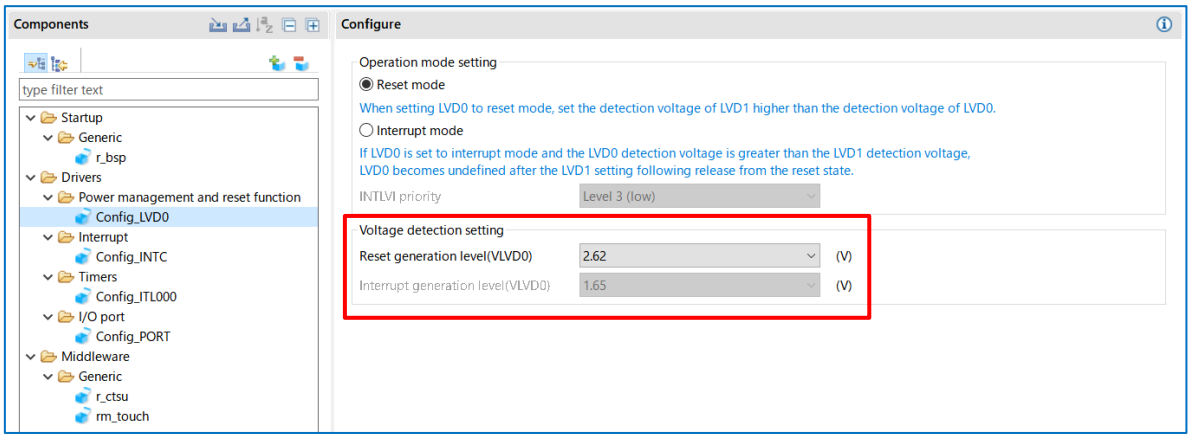


Figure 8-23 LVD Component Setting

- BSP Component Configuration
  Confirm that "Initialization of peripheral functions by Code Generator/Smart Configurator" is set to "Enable".

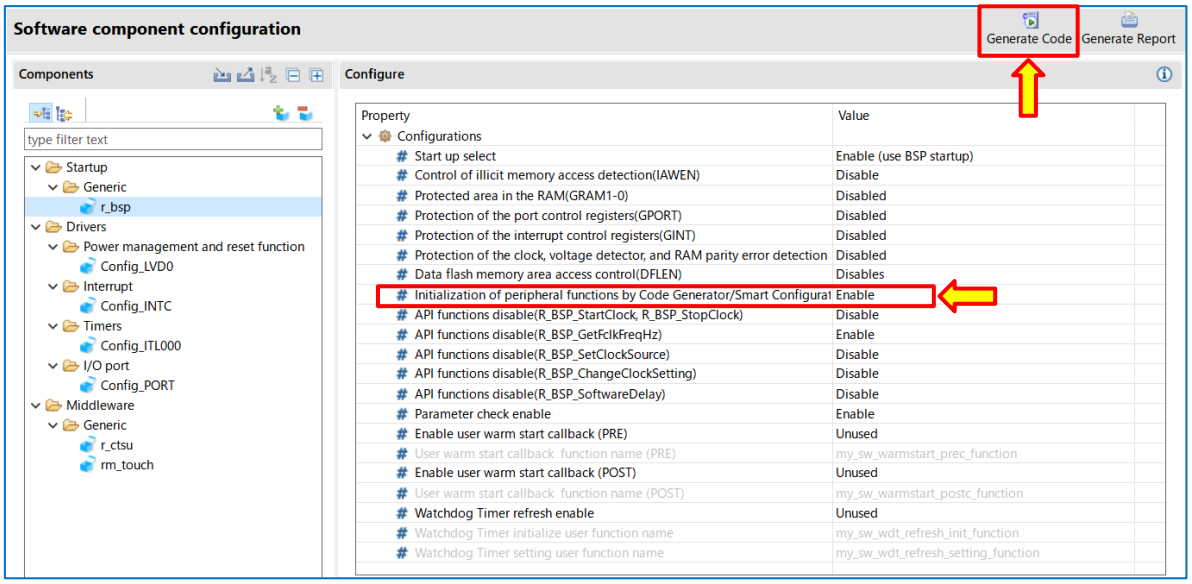


Figure 8-24 BSP Component Setting

Click the icon "Generate Code" located on the top right of the Smart Configurator to generate the code for components necessary for the project. This completes the process for adding components to the project.

8.3 Creating the Capacitive Touch Interface

This section describes how to create the configuration of the capacitive touch interface using QE for Capacitive Touch.

Some processes differ between RL78/G22 and RL78/G23.

1. Select [**Renesas Views**] - [**Renesas QE**] - [**CapTouch Main (QE)**] (or [**CapTouch Workflow (QE)**] for QE V3.2.0 or later) from the e² studio menu to open the main window and configure the capacitive settings for the project.

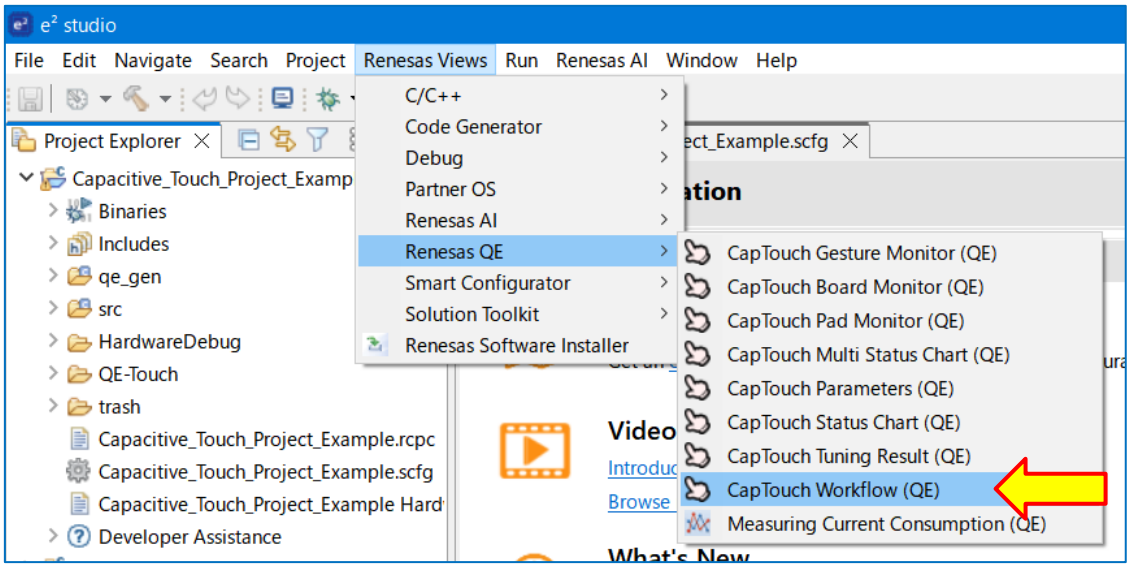


Figure 8-25 Cap Touch Workflow (QE) Selection

2. Select "**Capacitive_Touch_Project_Example**" from the pulldown menu in "Select a Project" of [**CapTouch Main (QE)**] (or [**CapTouch Workflow (QE)**] for QE V3.2.0 or later) to select the project and configure the touch interface.

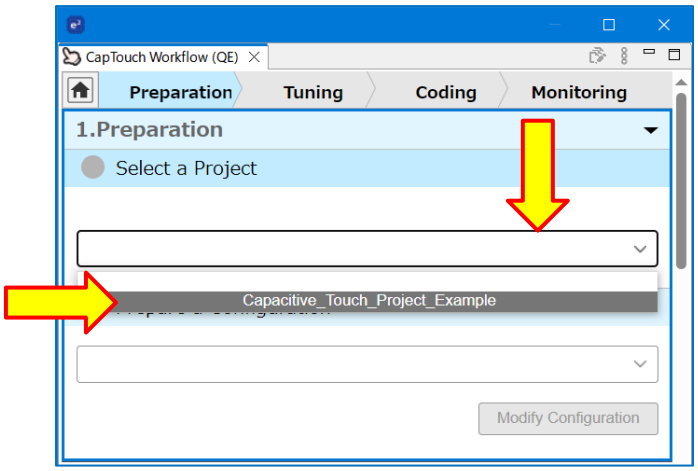


Figure 8-26 Project Selection

3. Select [**Create a new configuration**] from the pulldown menu in “Prepare a Configuration” to generate a new touch interface configuration.

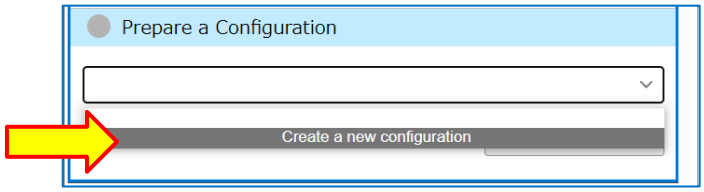


Figure 8-27 Create a new configuration

4. “Create Configuration of Touch Interfaces” window opens, providing an area to layout the touch interface.

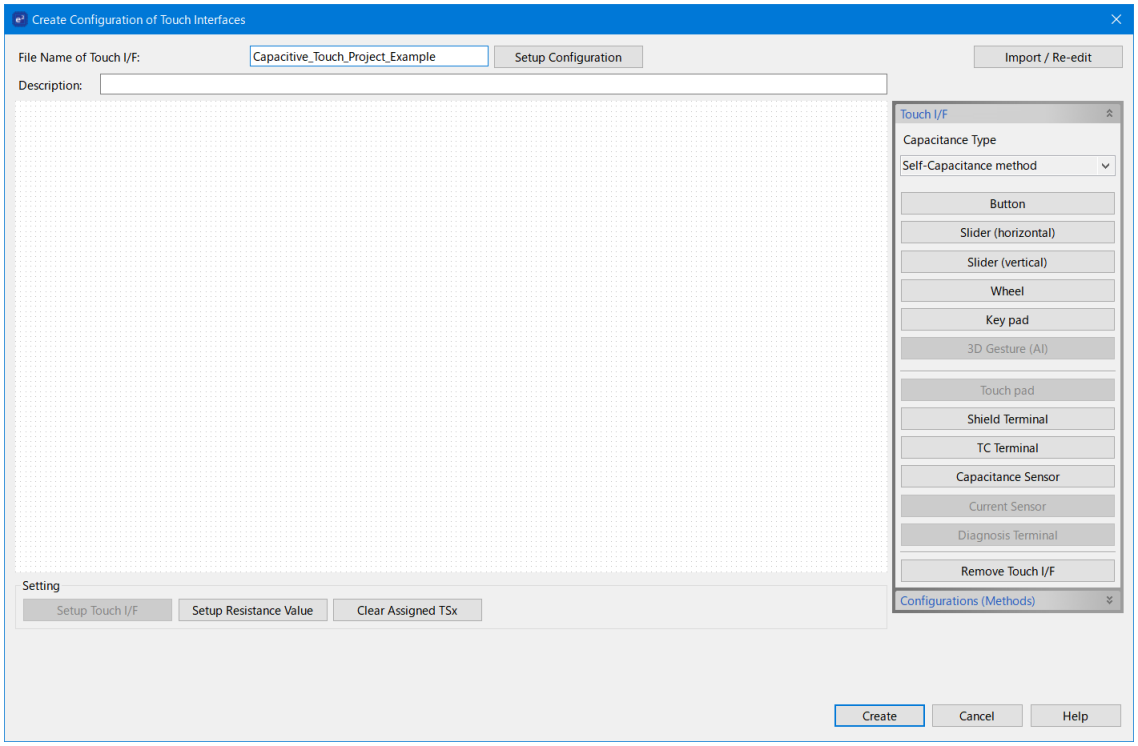


Figure 8-28 Touch Interface Layout Area

5. Select [**Button**] from “Touch I/F” on the right side of in the window. Add three buttons to the layout area. Press [**Esc**] on your keyboard to finish adding the touch interface. The layout should now look like the figure below.
   Remark: The error message “There are some problems with setting.” will appear under the layout area. Ignore this as it will disappear when you proceed to the next step and allocate a touch sensor to each button successfully.

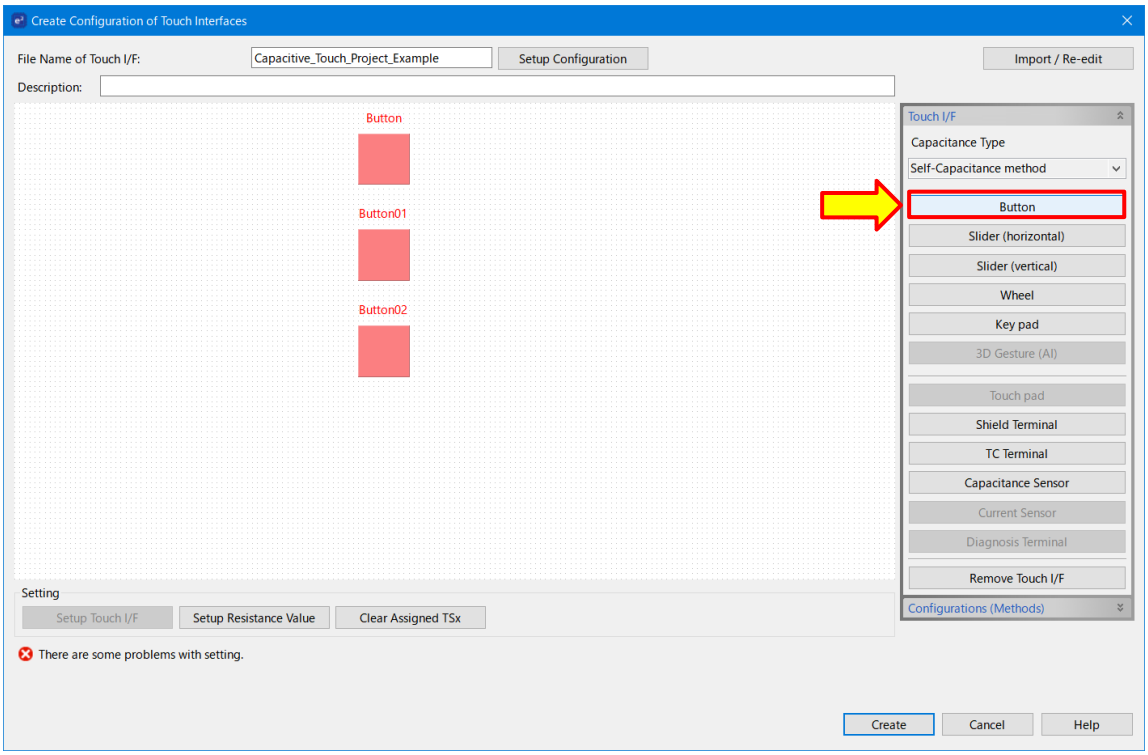


Figure 8-29 Add Touch Interface (3 buttons)

6. Allocate a name and a touch sensor to each button.
   Double click “**Button00**” to display the “Setup Touch Interface” dialog box. Change “Name” and “Touch Sensor” as marked with red in the figure, and then click [**OK**].
   For the RL78/G23, allocate TS06 to Button00.

   Caution：1. The touch sensor (TSxx) that shares a function with the port pin used as the external trigger for starting the SMS cannot be used.
   2. Any button name is acceptable, but an error message will appear if the character limit is exceeded.

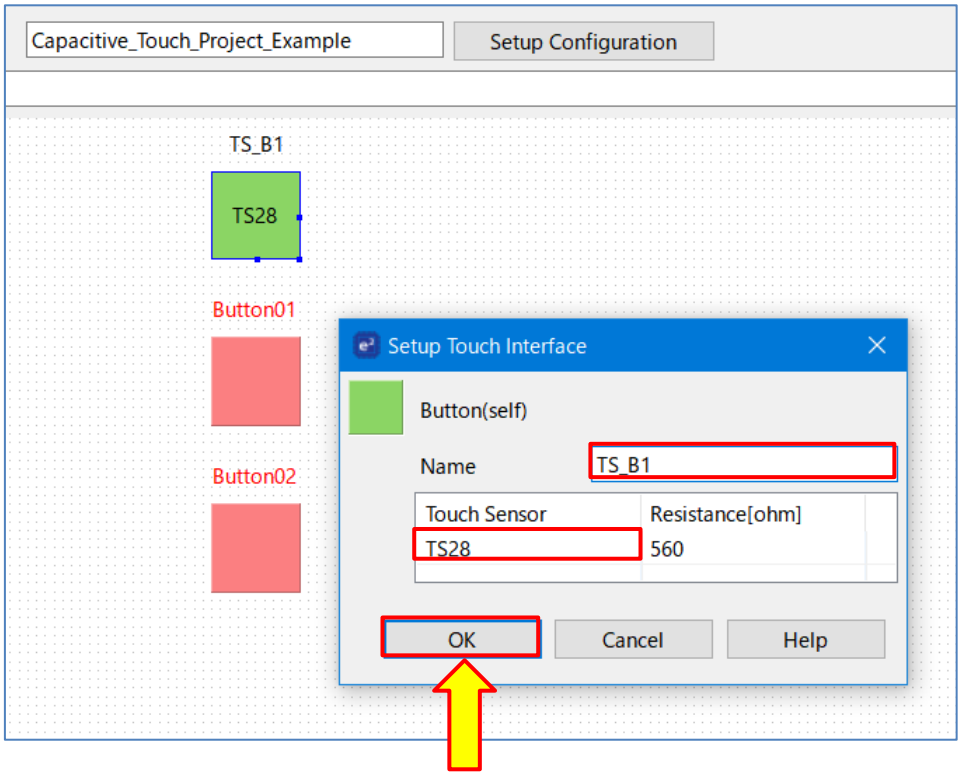


Figure 8-30 Touch Interface Setting

7. Set Button01 and Button02 in the same way as Button00(TS_B1).
   When the touch interface has been set up correctly, the error message will disappear.
   For the RL78/G23, TS05 and TS07 are allocated to Button01 and Button02, respectively.

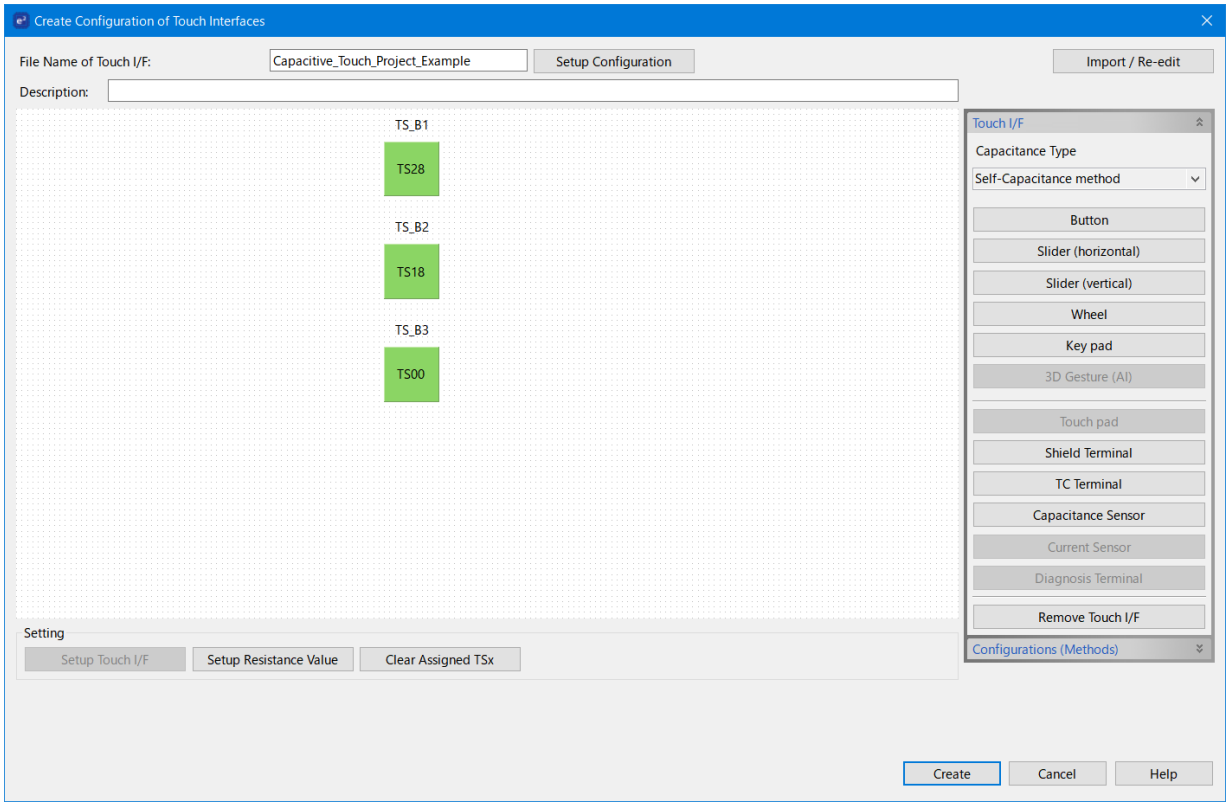


Figure 8-31 Touch Interface After Setup (3 buttons)

8. Set up wheel and slider in the same way as the buttons.
   The settings for the application example are shown in the figure below.
   When all touch interfaces are placed in the layout area, click [**Setup Configuration**].

   Caution: The error message “There are some problems with setting.” will appear under the layout area. The message will disappear when step 9 is completed.

   Remark: In the application example, the touch sensors allocated to the slider and wheel are switched to the MEC function buttons during low power mode. The name of the touch interface used for touch measurement using the SMS and MEC (low power mode) is TS_xxMEC.

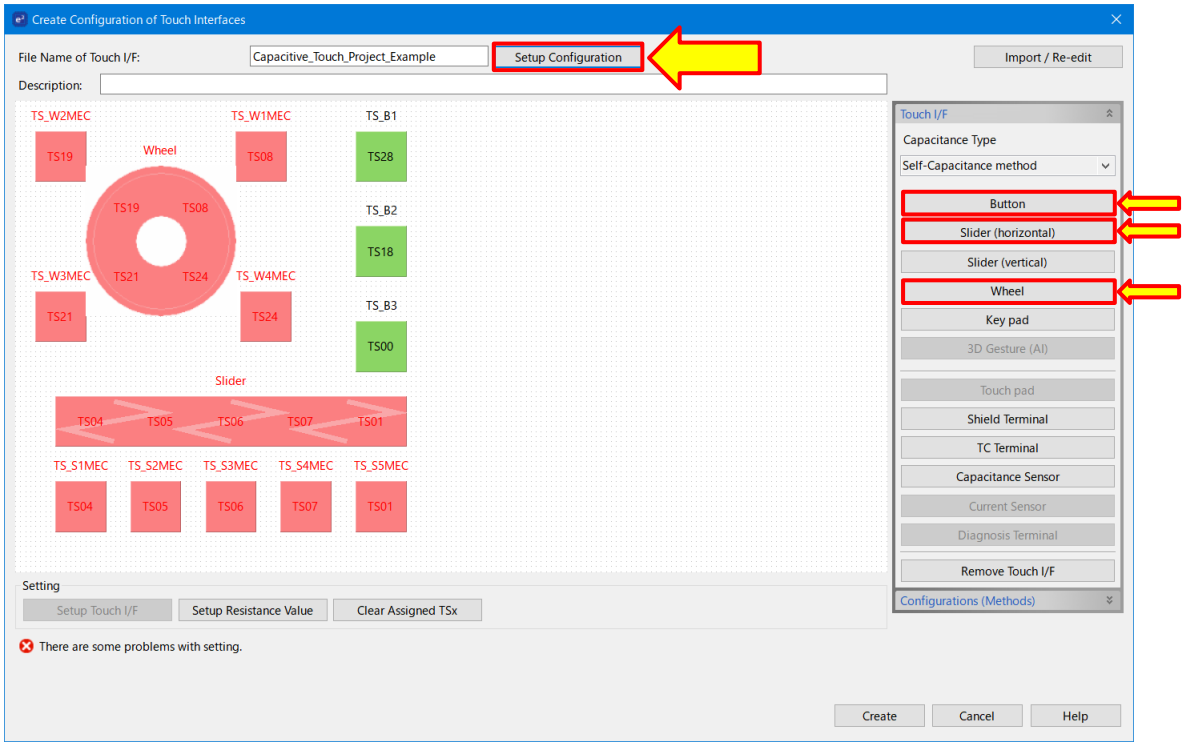


Figure 8-32 Add Touch Interface (wheel and slider) (RL78/G22)

9. Set up the auto judgement function and multiple electrode connection function in the “Setup Configurations (Methods)” window.
   When using auto judgement measurement using SMS, set "Auto Judgement by Hardware (SMS)” to "**Enable**”. When using multiple electrode connection (MEC), set "Multiple Electrode Connection” to "**Enable**”.

   The application example runs in both normal and low power mode and requires separate configurations, as shown in the figure below. Click [**Add Configuration**] to add “config02” next to “config01”. Set up each configuration, then click [**OK**].

   — config01: Function setting for touch interface configuration (method) in low power mode.
   — config02: Function setting for touch interface configuration (method) in normal mode.

   Caution: When using V1.50 of the CTSU module, the auto Judgement (SMS) can only be applied to config01.

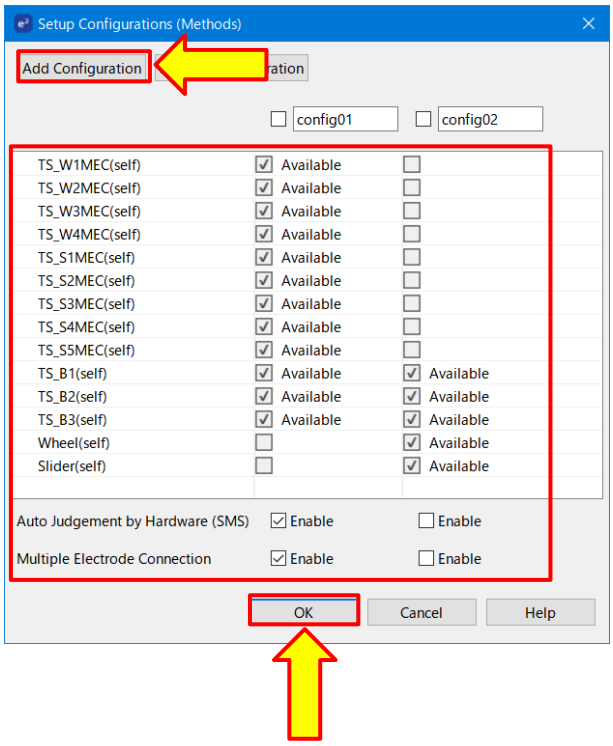


Figure 8-33 Setup Configurations (Methods) (RL78/G22)

10. Confirm there is no error displayed, then click [**Create**] in the “Create Configuration of Touch Interface” window.
    This completes the creation of the touch interface for RL78/G22.

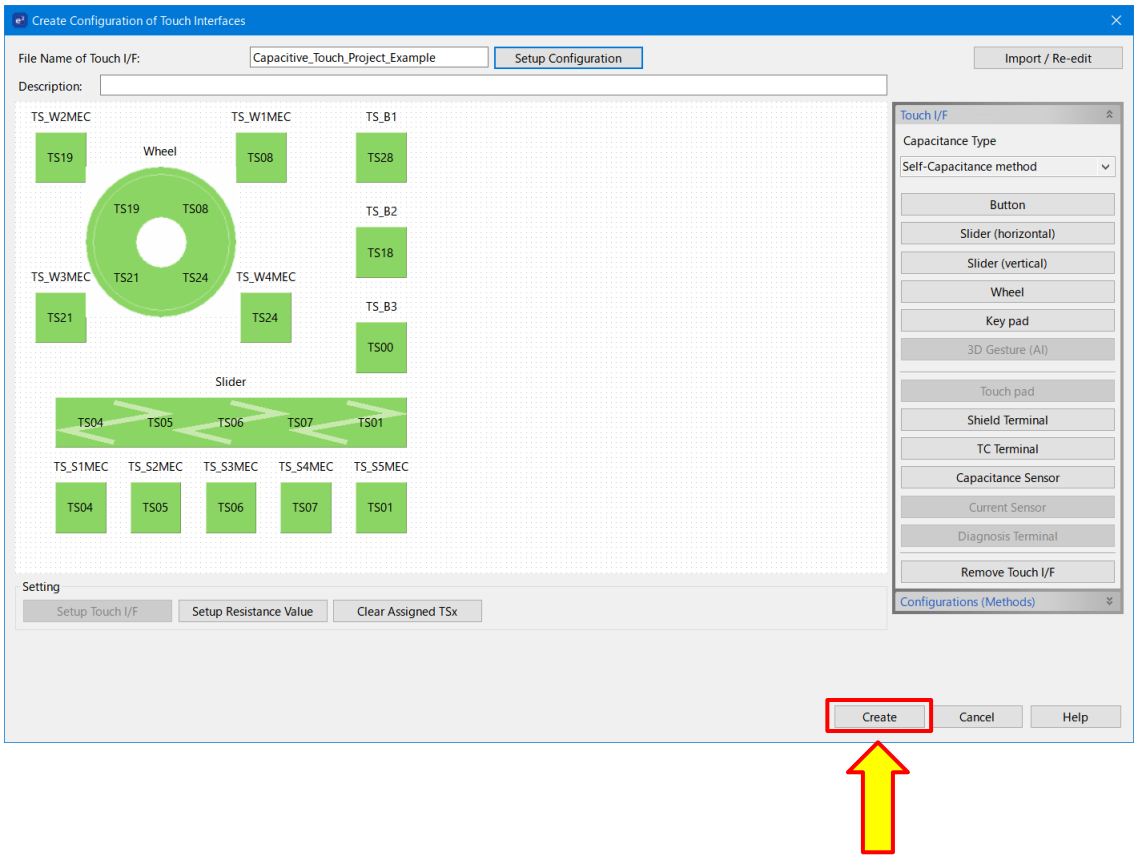


Figure 8-34 Touch Interface Configuration After Setup (RL78/G22)

11. This step on describes an example of the touch interface configuration for the RL78/G23.
    The setup procedure and notes/cautions are basically the same as those for RL78/G22.

    Caution: RL78/G22 and RL78/G23 allocate different touch sensors to electrodes.
    Remark: The name of the touch interface used for touch measurement using the SMS (low power mode) is TS_xxBTN.

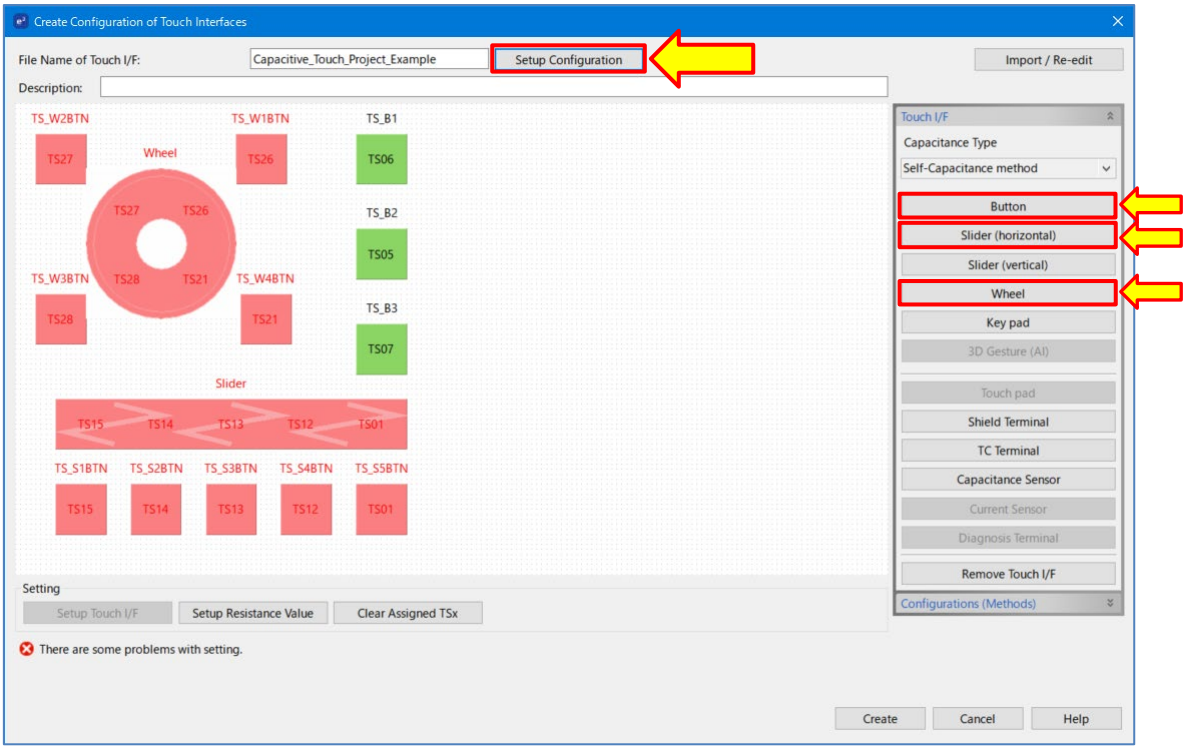


Figure 8-35 Add Touch Interface (wheel and slider) (RL78/G23)

12. Set up the auto judgement function in the “Setup Configurations (Methods)” window.
The setup for the application example is as follows:

— config01: Function setting for touch interface in low power mode.
— config02： Function setting for touch interface in normal mode.

Caution: When using V1.50 of the CTSU module, the auto Judgement (SMS) can only be applied to config01.

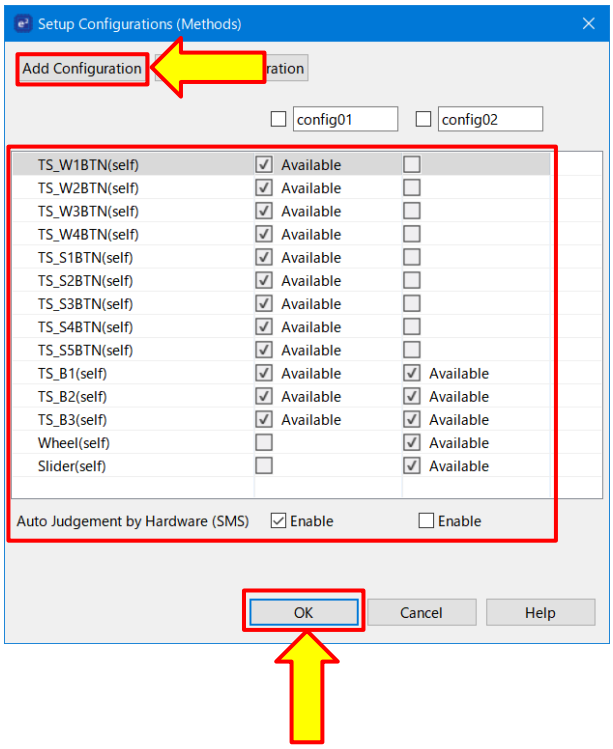


Figure 8-36 Setup Configurations (Methods) (RL78/G23)

13. Confirm there is no error displayed, then click [**Create**] in the “Create Configuration of Touch Interface” window.
    This completes the creation of the touch interface for the RL78/G23.

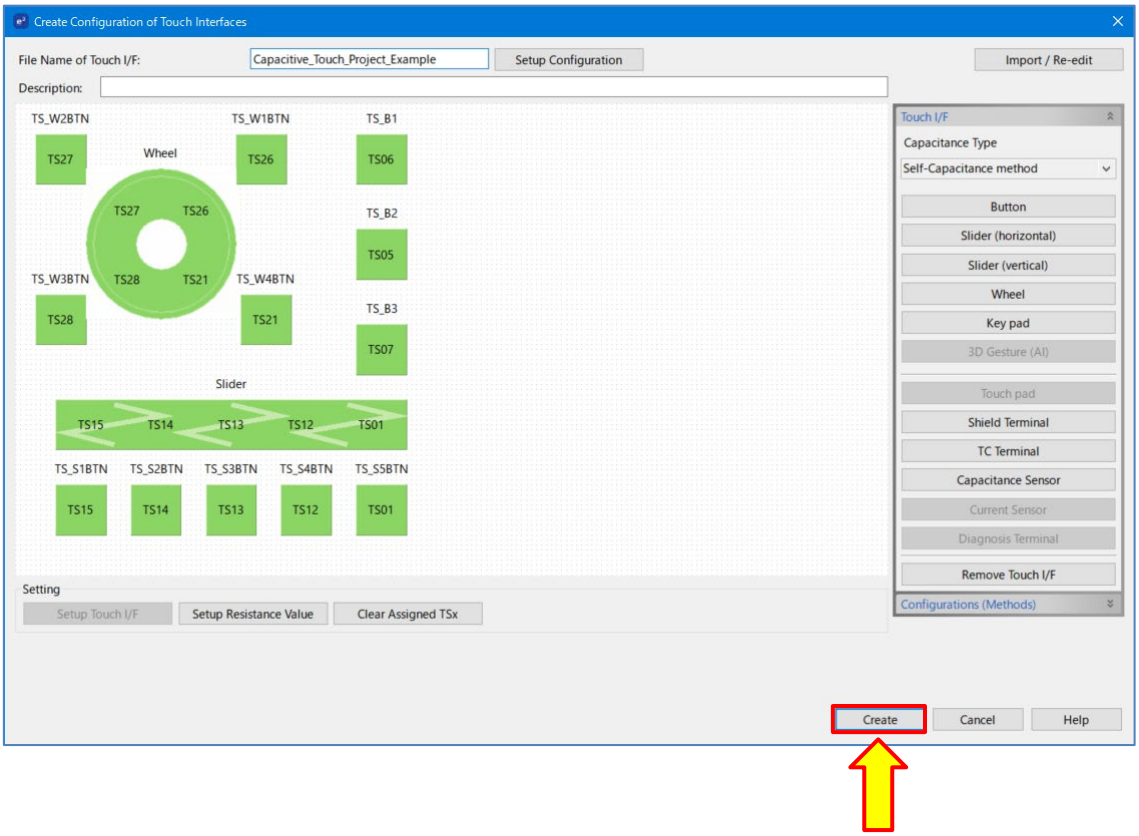
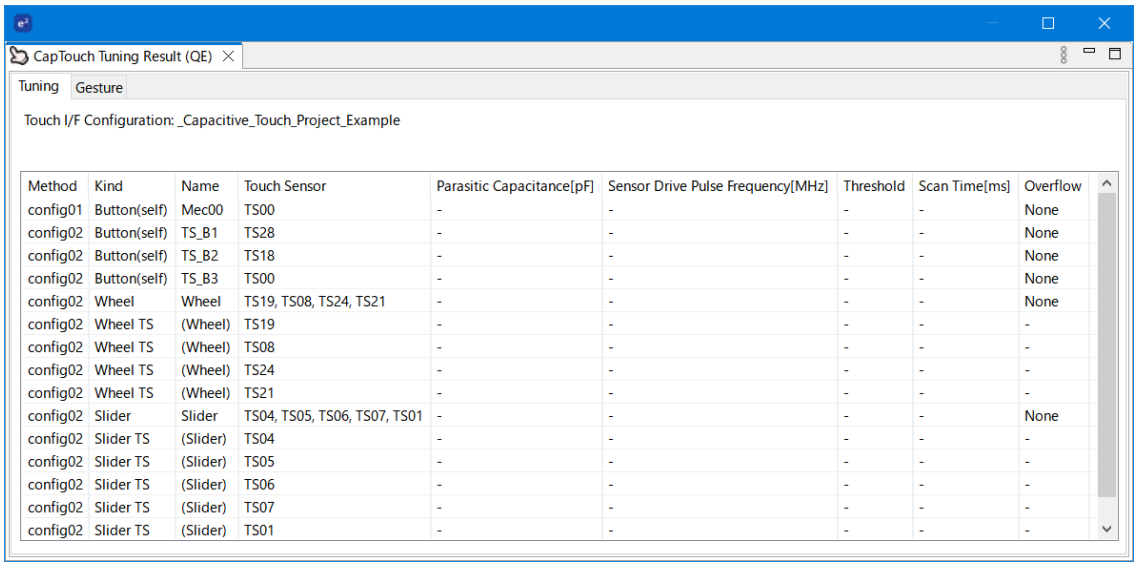


Figure 8-37 Touch Interface Configuration After Setup (RL78/G23)

14. The touch interface configuration is displayed in the “Tuning” panel of “**CapTouch Main(QE) / [CapTouch Tuning Result (QE)]** (for QE V3.2.0 or later)”.



Method	Kind	Name	Touch Sensor	Parasitic Capacitance[pF]	Sensor Drive Pulse Frequency[MHz]	Threshold	Scan Time[ms]	Overflow
config01	Button(self)	Mec00	TS00	-	-	-	-	None
config02	Button(self)	TS_B1	TS28	-	-	-	-	None
config02	Button(self)	TS_B2	TS18	-	-	-	-	None
config02	Button(self)	TS_B3	TS00	-	-	-	-	None
config02	Wheel	Wheel	TS19, TS08, TS24, TS21	-	-	-	-	None
config02	Wheel TS	(Wheel)	TS19	-	-	-	-	-
config02	Wheel TS	(Wheel)	TS08	-	-	-	-	-
config02	Wheel TS	(Wheel)	TS24	-	-	-	-	-
config02	Wheel TS	(Wheel)	TS21	-	-	-	-	-
config02	Slider	Slider	TS04, TS05, TS06, TS07, TS01	-	-	-	-	None
config02	Slider TS	(Slider)	TS04	-	-	-	-	-
config02	Slider TS	(Slider)	TS05	-	-	-	-	-
config02	Slider TS	(Slider)	TS06	-	-	-	-	-
config02	Slider TS	(Slider)	TS07	-	-	-	-	-
config02	Slider TS	(Slider)	TS01	-	-	-	-	-

Figure 8-38 [CapTouch Tuning Result (QE)] Tuning Panel

15. Click  on the top left in e² studio to build the project.

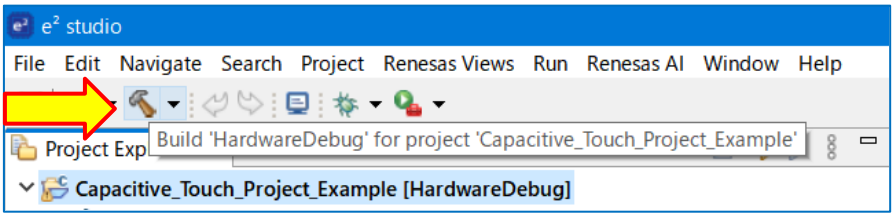


Figure 8-39 Project Build

Confirm that there is no error displayed in the “Console” window.
This completes the creation of the capacitive touch interface.

Caution: If the following error (E0562310) appears in the console after the build, open [**C/C++ Build**] - [**Settings**] - [**Linker**] - [**Input**] from the property of the created project, then check “Use runtime libraries (-library)” and rebuild the project.

```
E0562310:Undefined external symbol "__COM_lshr" referenced in ".\src\smc_gen\r_ctsu\r_ctsu.obj"
E0562310:Undefined external symbol "__COM_llshl" referenced in ".\src\smc_gen\rm_touch\rm_touch.obj"
E0562310:Undefined external symbol "__COM_llmul" referenced in ".\src\smc_gen\r_ctsu\r_ctsu.obj"
E0562310:Undefined external symbol "__COM_slldiv" referenced in ".\src\smc_gen\r_ctsu\r_ctsu.obj"
E0562310:Undefined external symbol "__COM_sldiv" referenced in ".\src\smc_gen\r_ctsu\r_ctsu.obj"
E0562310:Undefined external symbol "__COM_sldiv" referenced in ".\src\smc_gen\rm_touch\rm_touch.obj"

Renesas Optimizing Linker Abort
makefile:113: recipe for target 'G22_RSSK_SMS_MEC_test.abs' failed
make: *** [G22_RSSK_SMS_MEC_test.abs] Error 1
"make -j8 all" terminated with exit code 2. Build might be incomplete.

13:18:09 Build Failed. 8 errors, 0 warnings. (took 18s.976ms)
```

Figure 8-40 Console Window (when Error: E0562310 occurs)

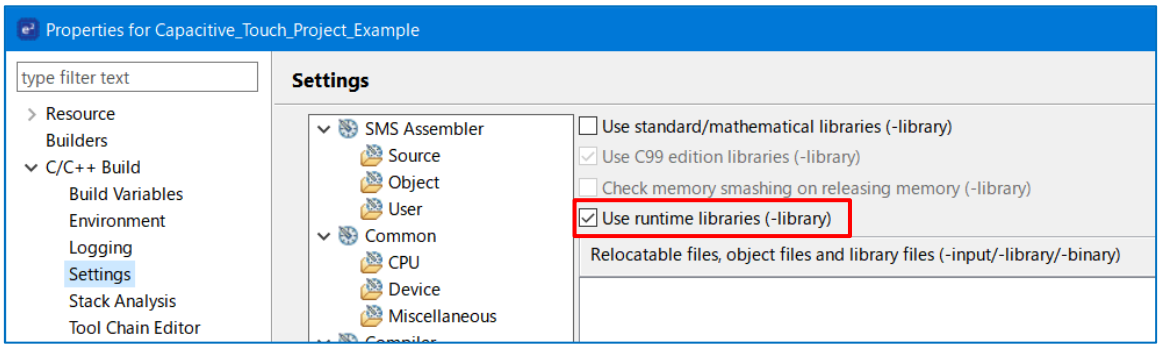


Figure 8-41 Set Runtime Libraries

8.4 Changing the Debug Configuration for Capacitive Touch Sensor Tuning

The debug configuration needs to be changed so that the tuning kernel can be downloaded to the MCU RAM after the debug session is started.

1. Click ▼ next to and select "Debug Configurations" from the pulldown menu.

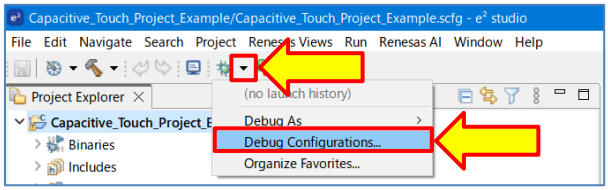


Figure 8-42 Select "Debug Configurations"

2. Select the [**Debugger**] - [**Connection Settings**] tabs in "Debug Configurations" window.
   Specify the connection settings with the target board as shown in the figure below.

   Caution: 1. In the application example, power is supplied to the target board from the emulator for easy confirmation of the operation. The power can also be supplied to the target board via E2 emulator Lite from the USB port on the PC, however we recommend using power generated by the target board.
   2. Available debugging methods vary depending on the specifications of the target board. Set up the required items by selecting "Debug hardware:" according to the debug method you are using.
   For example, COM port debugging requires a different setting.

— Debugging method: E2 Lite

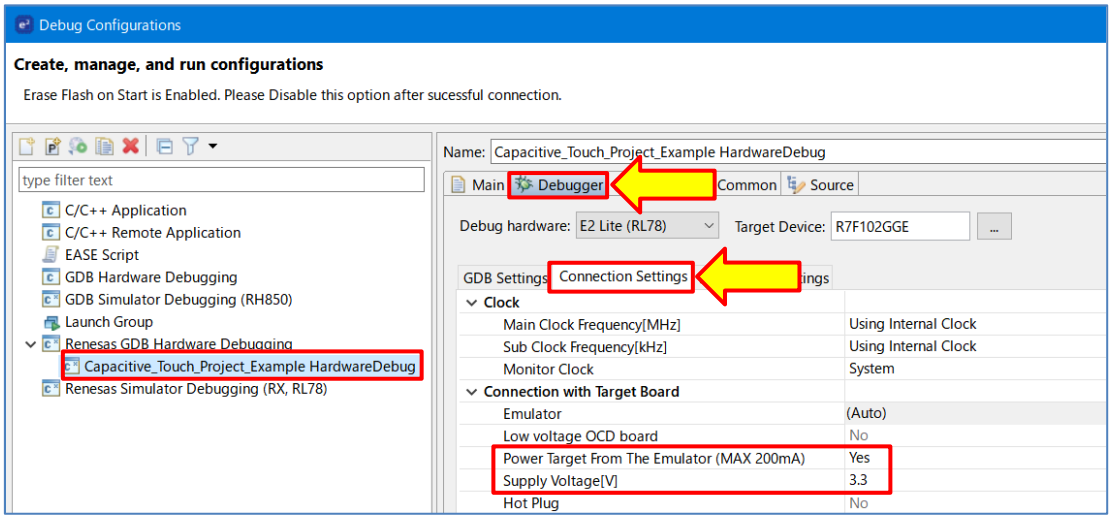


Figure 8-43 Debug Configuration (E2 Lite) Settings

— Debugging method: COM port

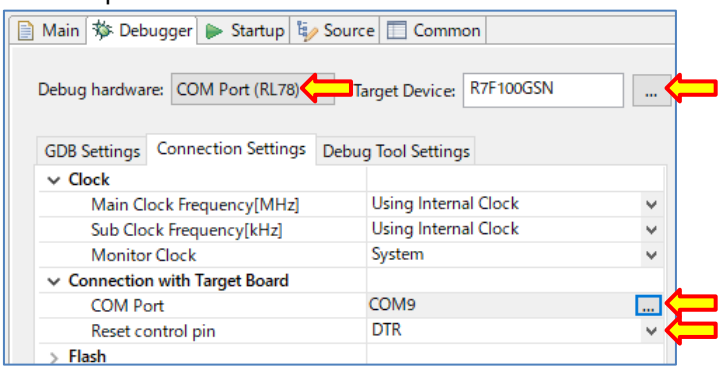


Figure 8-44 Debug Configuration (COM port) Settings

3. Select the [**Debug Tool Settings**] tab. Set "Allow to access by stopping execution while running" under "Memory" to "Yes".

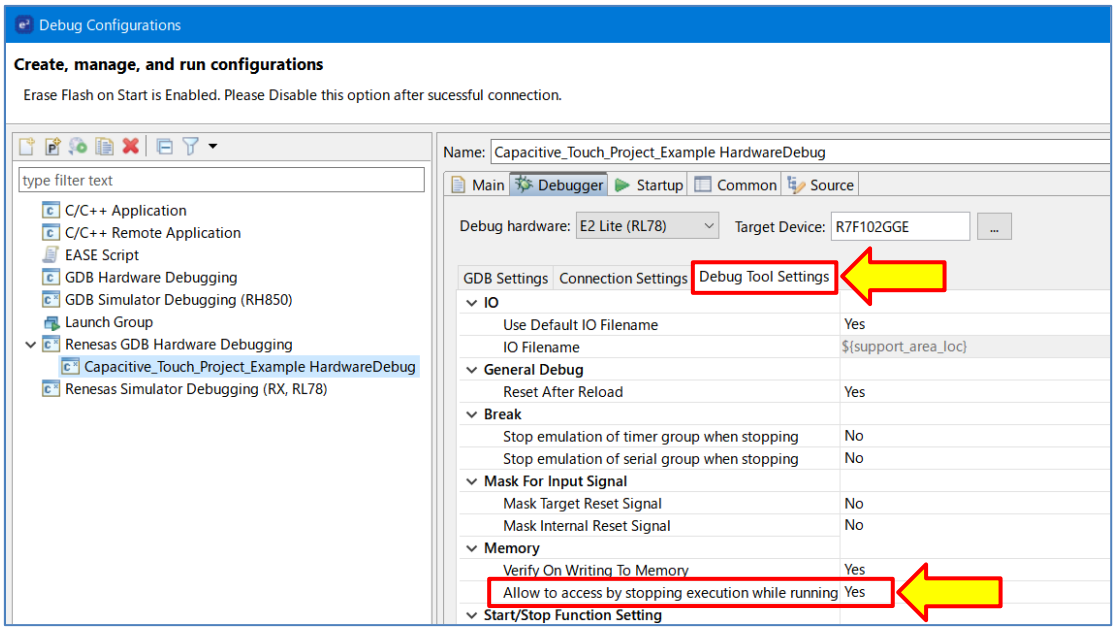


Figure 8-45 Debug Tool Settings

4. Select the “**Startup**” tab. Confirm that “Set breakpoint at:” and "Resume” are selected as shown in the figure below, then click [**Apply**].
   This completes the debug configuration for tuning.

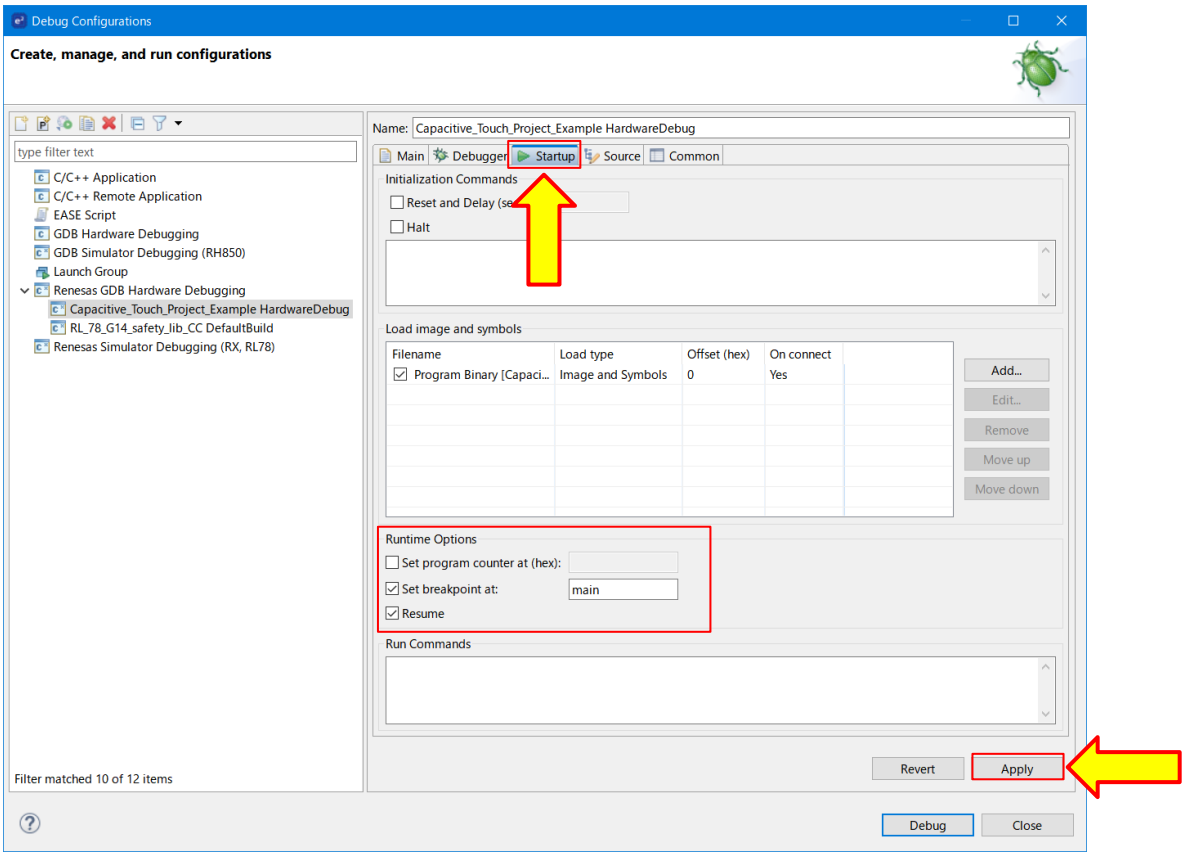


Figure 8-46 Runtime Options Settings

8.5 Capacitive Touch Sensor Tuning with QE for Capacitive Touch

This section describes how to tune the project using QE for Capacitive Touch.

1. Click [**Start Tuning**] in "[**CapTouch Main (QE)**] / [**CapTouch Workflow (QE)** (for QE V3.2.0 or later)]" to start automatic tuning.

   Note: In the application example, power is supplied to the target board from the emulator for easy confirmation of the operation. The power can also be supplied to the target board via E2 emulator Lite from the USB port on the PC, however we recommend using power generated by the target board for tuning.

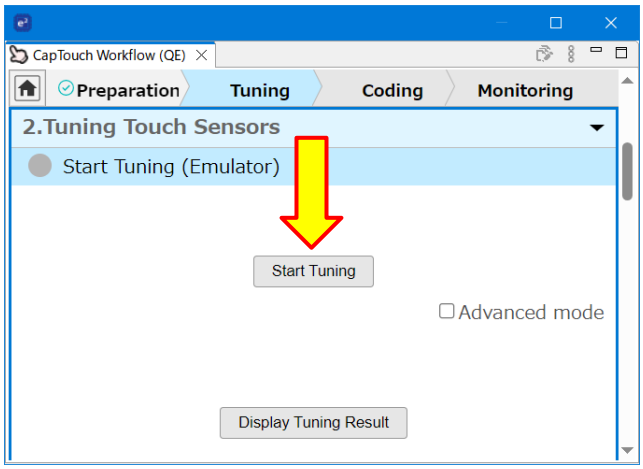


Figure 8-47 Start Automatic Tuning

2. When the debug session is started, e^2 studio may display the message to switch to the debug perspective. If the message is displayed, check [**Remember my decision(R)**], then click [**Switch**] to continue the debug session and automatic tuning for QE for Capacitive Touch.

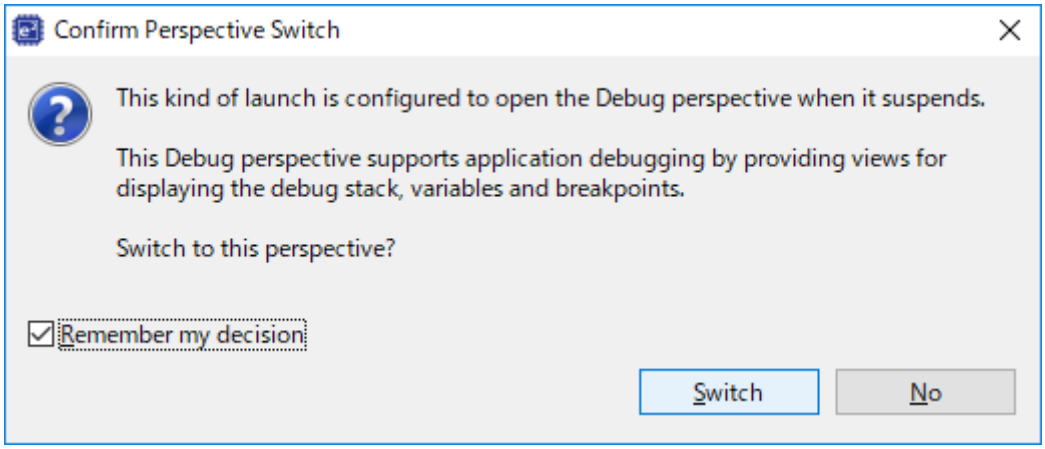


Figure 8-48 Confirm Perspective Switch

3. Automatic tuning for QE for Capacitive Touch is started. Continually confirm the “Automatic Tuning Processing” dialog box that guides the tuning process.
   The following is an example of the dialog box. Normally, the initial tuning processing will not require any action on the part of the user.

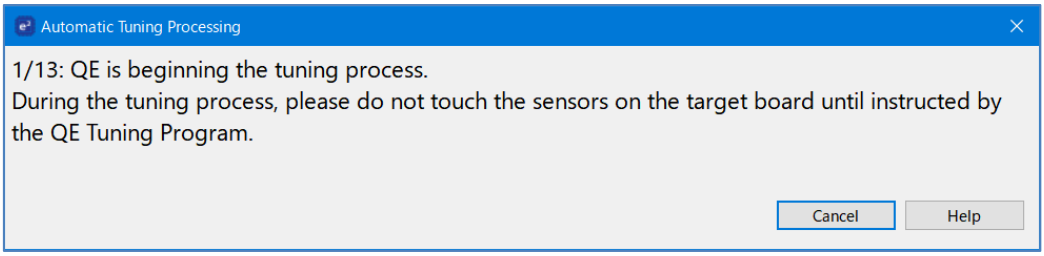


Figure 8-49 "Automatic Tuning Processing” Dialog Box (during initial tuning process)

The dialog box as shown below is displayed after several steps in the process. At this point, touch sensitivity in the tuning processing is measured. Touch the sensor (**Mec00**, **TS00**) indicated in the dialog box with normal pressure. When touching the sensor, the bar graph increases to the right and the number indicating the touch count value increases. Press any key on your PC keyboard while touching the sensor to confirm the measurement.

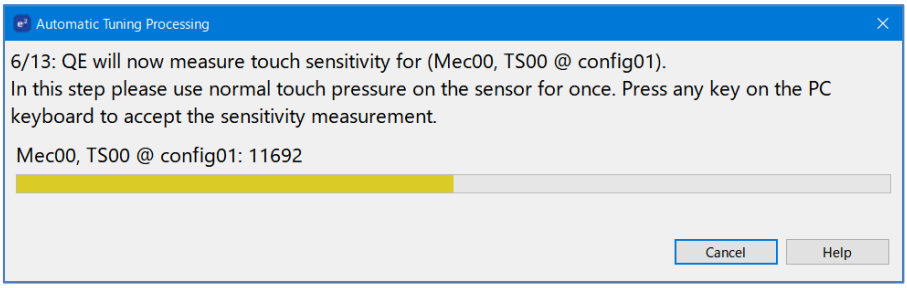


Figure 8-50 "Automatic Tuning Processing” Dialog Box (while measuring touch sensitivity)

Repeat the same procedure for all configured buttons.

4. When the tuning is completed, the following dialog box appears showing the threshold values. These threshold values are used for touch event judgement in the middleware.

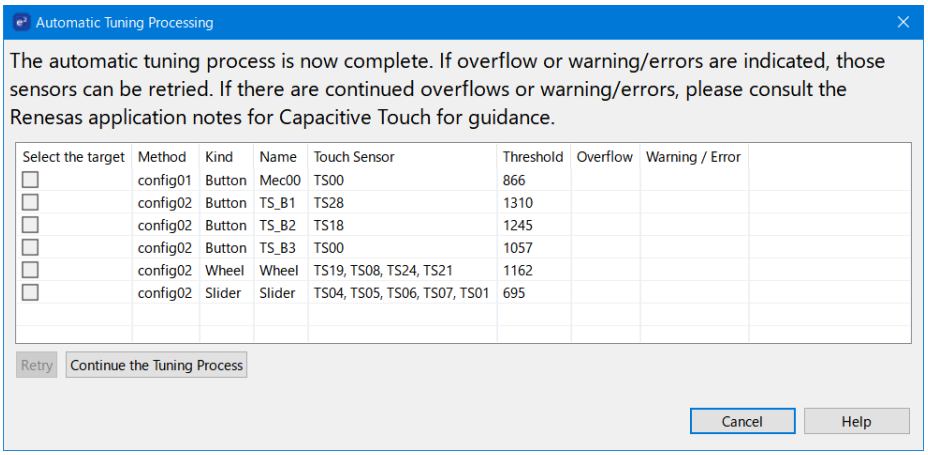


Figure 8-51 Tuning Results Display

5. Click [**Continue the Tuning Process**] in the displayed dialog box. The tuning process finishes and the target board and debugging session are disconnected. The process then goes back to "[**CapTouch Main (QE)**] / [**CapTouch Workflow (QE)** (for QE V3.2.0 or later)]".

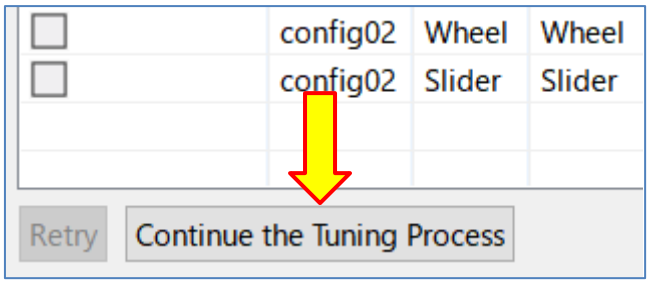


Figure 8-52 Continue the Tuning Process

6. Select "Use an external trigger" and click [**Output Parameter Files**] as shown below to output the tuned parameter files.

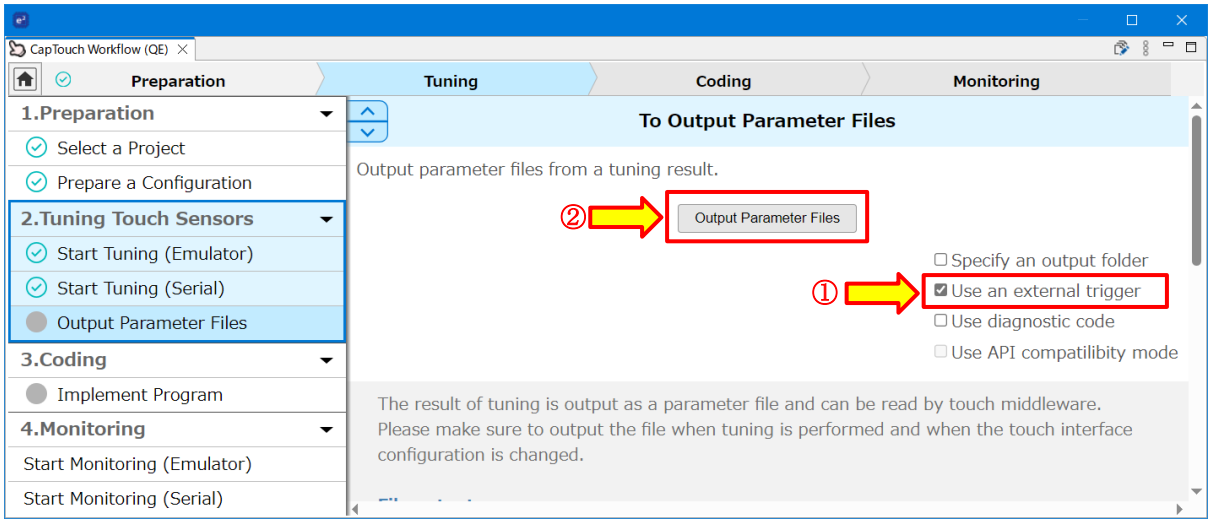


Figure 8-53 Output Parameter Files

7. Confirm that **qe_touch_config.c, qe_touch_config.h, and qe_touch_define.h** have been added in the "**Project Explorer**" window.
   These files include the tuning information required to enable touch detection using the driver.

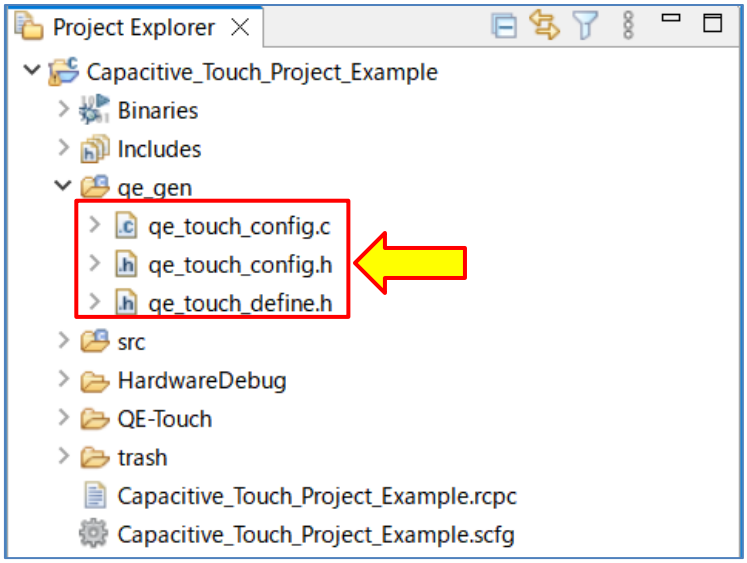


Figure 8-54 Tuning Parameter File Output

8. Click  on the top left of e² studio to build the project. In the "Console" window, confirm that no error has occurred during the build.
   This completes the tuning of the capacitive touch sensor using QE for Capacitive Touch.

8.6 Adding API Call for rm_touch Middleware

This section describes how to add the API call for the rm_touch middleware to the project and enable capacitive touch control.

1. Click [**Show Sample**] in "[**CapTouch Main (QE)**] / [**CapTouch Workflow (QE)** (for QE V3.2.0 or later)]" to implement the program that scans the touch sensor.

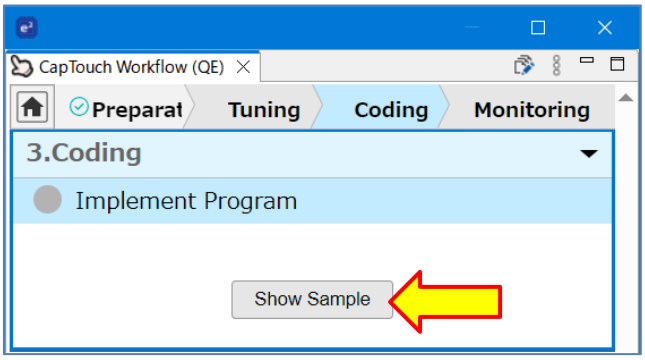


Figure 8-55 Sample Code Example Display

2. The "Show Sample Code" window opens and displays the sample code. Click [**Output to a File**] to output the sample code.

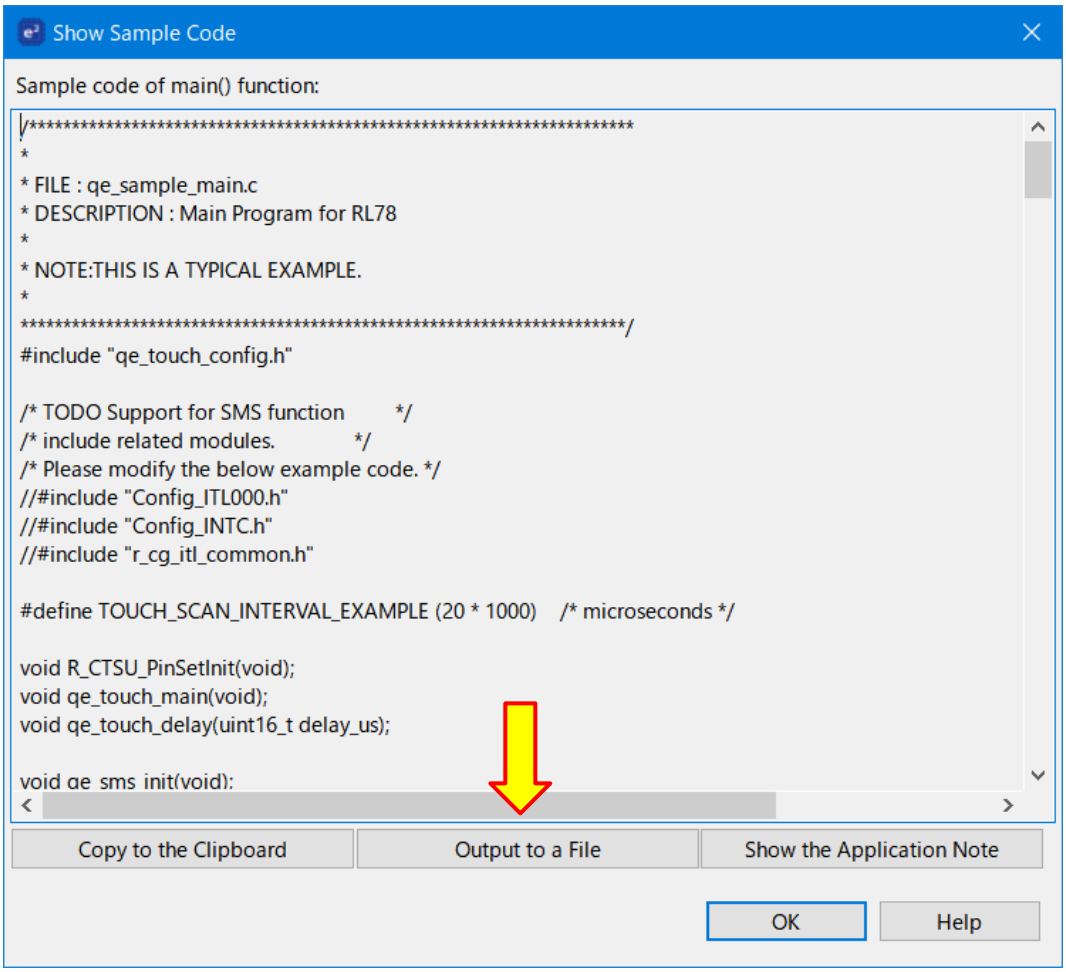


Figure 8-56 Sample Code Output to File

3. Confirm that “**qe_touch_sample.c**” file is generated in “Project Explorer”.

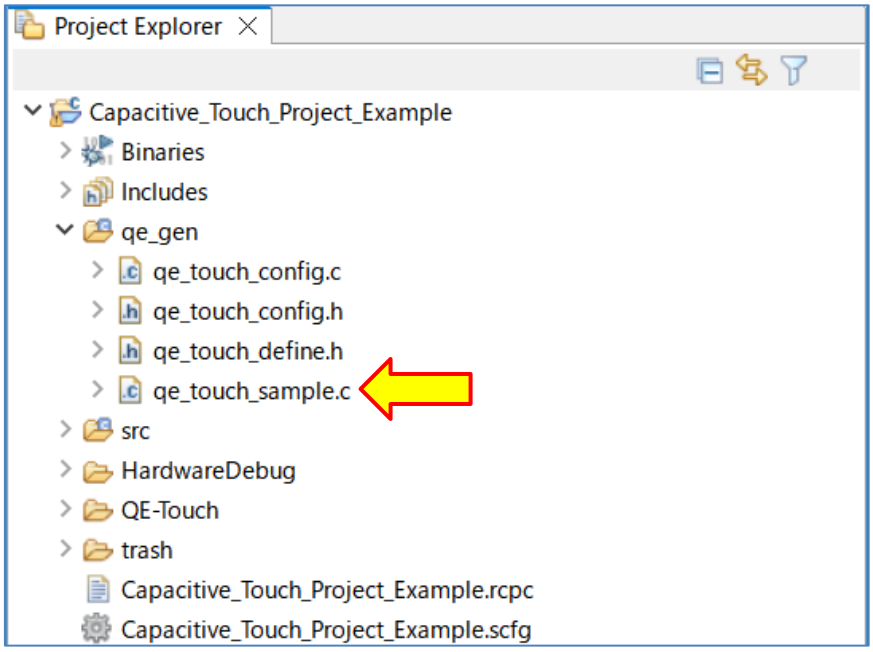


Figure 8-57 QE Output Code (qe_touch_sample.c) Generation Confirmation

4. Open “**Capacitive_Touch_Project_Example.c**”.

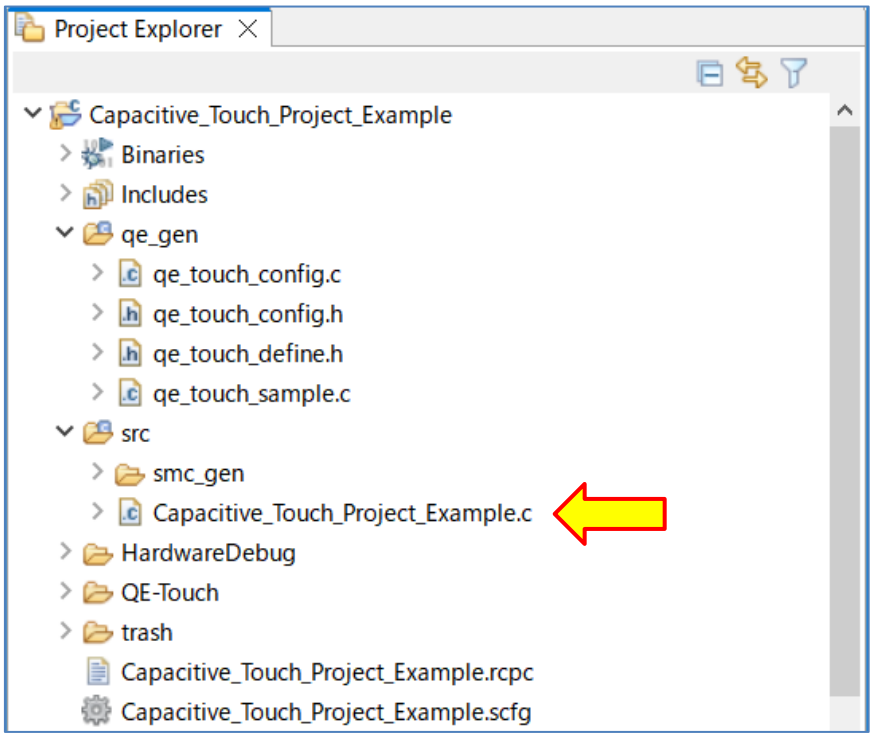


Figure 8-58 Select Capacitive_Touch_Project_Example.c

5. Add codes “**void qe_touch_main(void);**” and “**qe_touch_main();**” (marked with red in the figure) into the “**Capacitive_Touch_Project_Example.c**” file to call the qe_touch_main() function form the main() function.

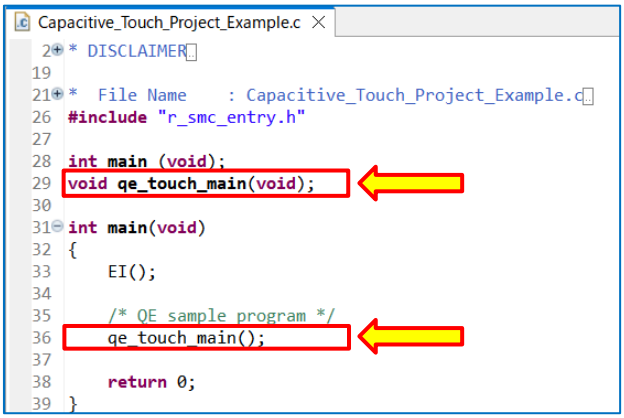


Figure 8-59 Set Call for qe_touch_main() Function

6. Specify the include file setting in the QE output code (qe_touch_sample.c).
   Change the comment settings as follows to include the necessary header files.

   Note: For RL78/G23, leave “#include "Config_INTC.h"” as commented out.

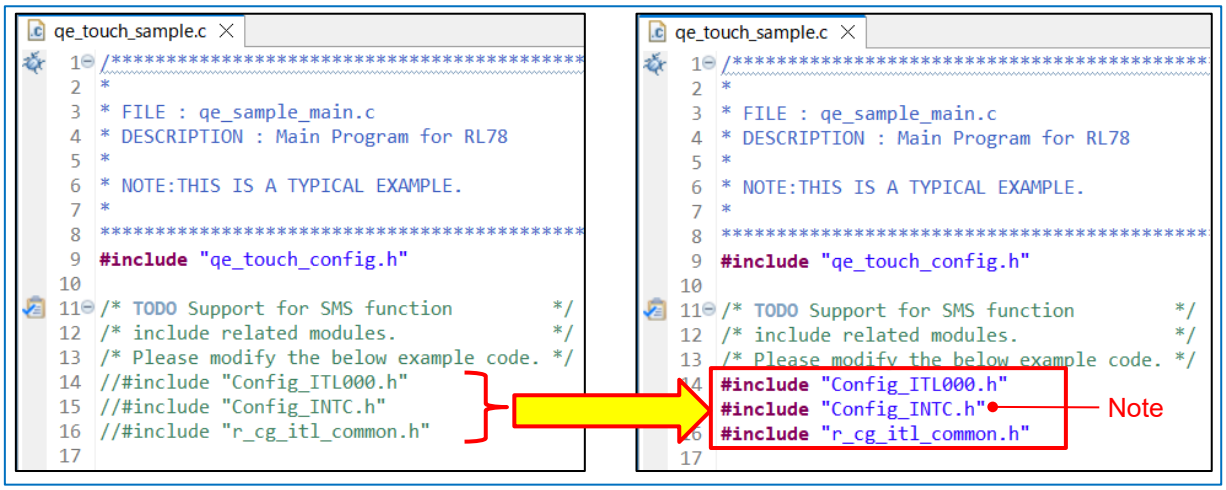


Figure 8-60 Include File Setting in QE Output Code (qe_touch_sample.c)

7. Add the initial offset tuning processing for Config2 to the qe_touch_sample.c file. Copy the Config01 code and add the code in the qe_touch_sample.c file as shown in the red boxed area in the figure below. Then change the code to specify Config02 as underlined in red in the figure.

qe_touch_sample.c

```
68    /* Initial offset tuning for [CONFIG01] */
69    {
70        err = RM_TOUCH_ScanStart(g_qe_touch_instance_config01.p_ctrl);
71        if (FSP_SUCCESS != err)
72        {
73            while (true) {}
74        }
75
76        /* Clear the flag indicating that CTSU processing is complete. */
77        g_qe_touch_flag = 0U;
78
79        while (true)
80        {
81            qe_sms_trigger_start();
82
83            while (0U == g_qe_touch_flag) {}
84            g_qe_touch_flag = 0U;
85
86            err = RM_TOUCH_DataGet(g_qe_touch_instance_config01.p_ctrl, &button_status, NULL, NULL);
87            if (FSP_SUCCESS == err)
88            {
89                qe_sms_trigger_stop();
90                break;
91            }
92        }
93    }
94
95
96    /* Initial offset tuning for [CONFIG02] */
97    {
98        err = RM_TOUCH_ScanStart(g_qe_touch_instance_config02.p_ctrl);
99        if (FSP_SUCCESS != err)
100       {
101           while (true) {}
102       }
103
104       /* Clear the flag indicating that CTSU processing is complete. */
105       g_qe_touch_flag = 0U;
106
107       while (true)
108       {
109           qe_sms_trigger_start();
110
111           while (0U == g_qe_touch_flag) {}
112           g_qe_touch_flag = 0U;
113
114           err = RM_TOUCH_DataGet(g_qe_touch_instance_config02.p_ctrl, &button_status, slider_position, wheel_position);
115           if (FSP_SUCCESS == err)
116           {
117               qe_sms_trigger_stop();
118               break;
119           }
120       }
121   }
122
123   /* Main loop */
124   while (true)
125   {
```

Figure 8-61 Initial Offset Tuning Processing for Config02

8. Add the external trigger setting for Config02 measurement in the qe_touch_sample.c file.

qe_touch_sample.c

```
128         /* SMS setting for [CONFIG01] */
129         err = RM_TOUCH_SmsSet(g_qe_touch_instance_config01.p_ctrl);
130         if (FSP_SUCCESS != err)
131         {
132             while (true) {}
133         }
134
135         /* Scan start */
136         err = RM_TOUCH_ScanStart(g_qe_touch_instance_config01.p_ctrl);
137         if (FSP_SUCCESS != err)
138         {
139             while (true) {}
140         }
141
142         qe_sms_trigger_start();
143
144         /* Wait for CTSU to finish processing */
145         while (true)
146         {
147             __stop();
148
149             if (1U == g_qe_touch_flag)
150             {
151                 break;
152             }
153         }
154
155         g_qe_touch_flag = 0U;
156
157         qe_sms_trigger_stop();
158
159         err = RM_TOUCH_DataGet(g_qe_touch_instance_config01.p_ctrl, &button_status, NULL, NULL);
160         if (FSP_SUCCESS == err)
161         {
162             /* TODO: Add your own code here. */
163         }
164
165         /* for [CONFIG02] configuration */
166         err = RM_TOUCH_ScanStart(g_qe_touch_instance_config02.p_ctrl);
167         if (FSP_SUCCESS != err)
168         {
169             while (true) {}
170         }
171
172         qe_sms_trigger_start();
173
174         while (0 == g_qe_touch_flag) {}
175         g_qe_touch_flag = 0;
176
177         qe_sms_trigger_stop();
178
179         err = RM_TOUCH_DataGet(g_qe_touch_instance_config02.p_ctrl, &button_status, slider_position, wheel_position);
180         if (FSP_SUCCESS == err)
181         {
182             /* TODO: Add your own code here. */
183         }
184
185
```

Figure 8-62 Add External Trigger Setting for Config02 Measurement

9. Change comment settings as shown in the figure below to specify the qe_sms_init() function in the qe_touch_sample.c file.

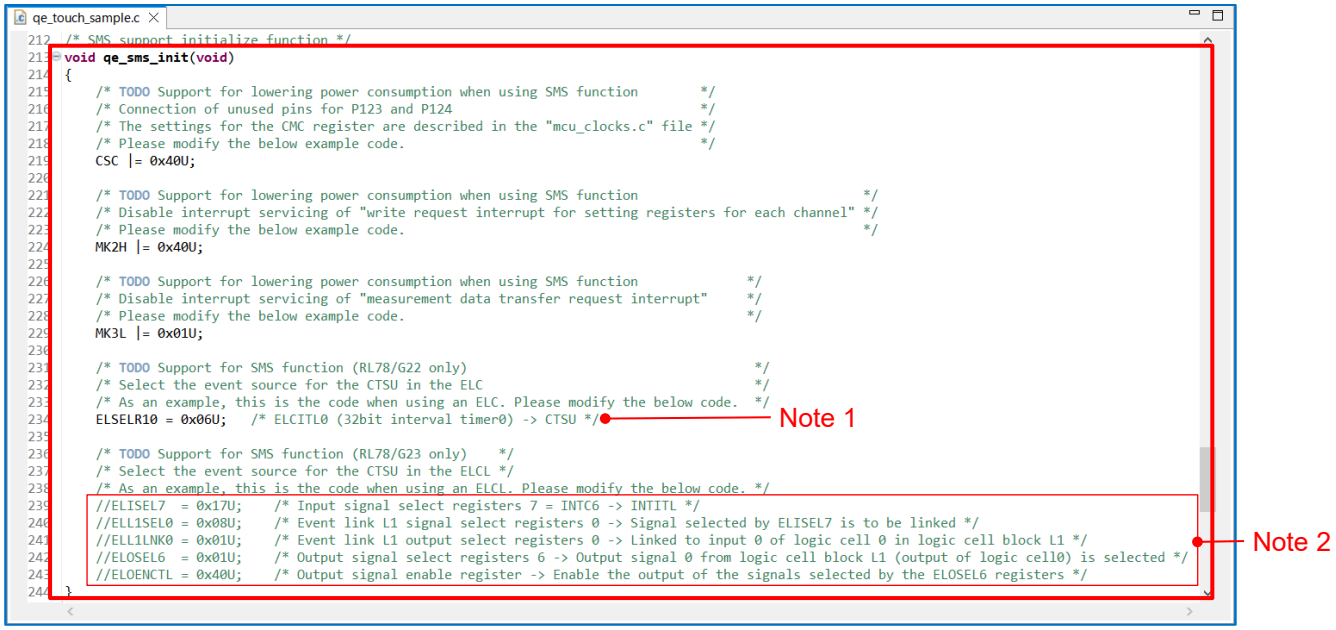


Figure 8-63 Specify qe_sms_init Function

Note: 1. For RL78/G23, leave “ELSELR10 = 0x06U;” as commented out.

Note: 2. For RL78/G23, uncomment the following code:

```
ELISEL7 = 0x17U;
ELL1SEL0 = 0x08U;
ELL1LNK0 = 0x01U;
ELOSEL6 = 0x01U;
ELOENCTL = 0x40U;
```

10. Change the comments in the qe_touch_sample.c file as shown in the figure to specify functions qe_sms_trigger_start() and qe_sms_trigger_stop().

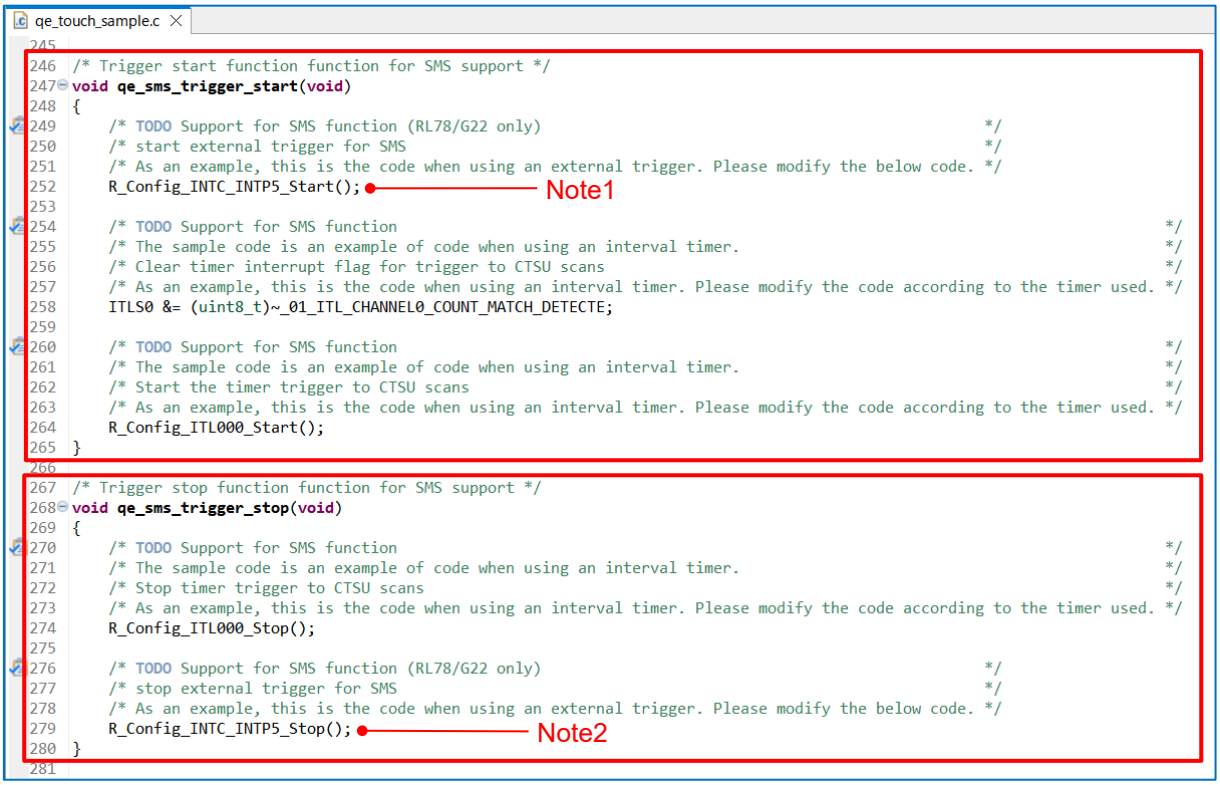


Figure 8-64 Specify Functions qe_sms_trigger_start and qe_sms_trigger_stop

Note: 1.For RL78/G23, leave “R_Config_INTC_INTP5_Start();” as commented out.
Note: 2.For RL78/G23, leave “R_Config_INTC_INTP5_Stop();” as commented out.

11. Click  on the top left of e^2 studio to build the project. Confirm that no error occurred during the build in the “Console” window.

This completes the procedure required to develop a capacitive touch low power application with auto judgement measurement using SMS.

9. Documents for Reference

RL78/Gxx User's Manual: Hardware
RL78 Family User's Manual: Software (R01US0015)
(The latest versions of the documents are available on the Renesas Electronics Website.)

Technical Updates/Technical Brochures
(The latest versions of the documents are available on the Renesas Electronics Website.)

Application Note RL78 Family Capacitive Touch Sensing Unit (CTSU2L) Operation Explanation (R01AN5744)
Application Note RL78 Family CTSU Module Software Integration System (R11AN0484)
Application Note RL78 Family TOUCH Module Software Integration System (R11AN0485)
Application Note Capacitive Sensor Microcontrollers
CTSU Capacitive Touch Electrode Design Guide (R30AN0389)
Application Note RL78/G23 Capacitive Touch Low Power Guide (SMS function) (R01AN6670)
Application Note RL78/G22 Capacitive Touch Low Power Guide (SMS / MEC function) (R01AN6847)
(The latest versions of the documents are available on the Renesas Electronics Website.)

Website and Support

Renesas Electronics Website

http://www.renesas.com/

Capacitive Sensing Unit related page
https://www.renesas.com/solutions/touch-key
https://www.renesas.com/qe-capacitive-touch

Inquiries
http://www.renesas.com/contact/

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul.22.24	-	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/.