

Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

Introduction

This application note describes the steps for creating an example of a capacitive touch sensing application on the RL78/G22 fast prototyping board (FPB), which incorporates touch electrodes.

This application note is a guide to the development of capacitive touch applications by using a combination of CS+, the standalone version of the Smart Configurator, and the standalone version of QE for Capacitive Touch. The standalone version of QE for Capacitive Touch^{Note} enables development independently of the device or integrated development environment (IDE) by using serial communications between the PC and board.

Note: QE for Capacitive Touch is a development tool for supporting initial settings and sensitivity adjustment of touch interfaces that are required in the development of embedded systems that use capacitive touch sensors.

Target Devices

RL78/G22

Other products of the RL78 family that incorporate the capacitive sensing unit (CTSU)

Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

Contents

1.	Overview	4
2.	Operating Environment	4
2.1	Functions of QE for Capacitive Touch	5
3.	Configuring the Development Environment	
3.1	Installing Development Tools	
3.1.1	Procedure for Installing the CS+ Integrated Development Environment	6
3.1.2	Procedure for Installing the Standalone Version of the Smart Configurator	6
3.1.3		
3.2	Hardware Settings	7
4.	Procedure of Application Development	8
5.	Sample Application	10
5.1	Overview of the Sample Application	
5.2	List of Pins Used	11
6.	Creating a New Project	12
7.	Settings of the Smart Configurator	13
7.1	Starting the Smart Configurator	
7.2	Setting the Clock and System	
7.3	Downloading Software Integration System (SIS) Modules	
7.4	Adding Components	17
7.5	Modifying the Component Settings in the Smart Configurator	
7.5.1	Setting the CTSU Component	
7.5.2	2 Setting the Touch Component	21
7.5.3	8 Setting the UART Communications Component	
7.5.4	Setting the LVD Component	
7.5.5	5 Setting the PORT Component	
7.5.6	Board Support Package	
7.6	Setting Unused Pins	
7.7	Generating Code	
8.	Settings of QE for Capacitive Touch	29
8.1	Starting QE for Capacitive Touch	
8.2	Preparing a Project	
8.3	Configuring the Touch Interface	
8.4	Tuning	



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping	g Board
8.5 Coding and Monitoring	
8.5.1 Monitoring	
8.6 Flowchart (Software Timer)	57
9. Another Implementation of the Sample Application	58
9.1 Touch Measurement with the Use of a Hardware Timer	58
9.1.1 Using the Smart Configurator to Make Settings (Hardware Timer)	58
9.1.2 Flowchart (Hardware Timer)	61
9.1.3 Sample Code (Hardware Timer)	62
10. Documents for Reference	64
Revision History	65



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

1. Overview

This application note describes the following procedures for using a device of the RL78 family to embed a capacitive touch function in a system.

- Creating a project for using the RL78/G22 FPB by using the standalone version of Smart Configurator
- Creating, tuning, and monitoring touch interfaces by using the standalone version of QE for Capacitive Touch

The application note describes the procedures for using the RL78/G22 FPB, but the procedures can also be applied to other devices of the RL78 family that incorporate the capacitive touch IP.

2. Operating Environment

Table 2-1 and Table 2-2 list the elements of the environment used in development for this application note.

The sample code attached to this application note was developed with the versions of tools listed in Table 2-1. This application note also supports development with the versions in parentheses () in the table.

The program generated by the standalone version of QE is written to the RL78/G22 through CS+ and then executed on the RL78/G22.

Table 2-1 Development Environment (Software)

Item	Description	Version
Integrated development	CS+ for CC	V8.13.00
environment (IDE)		(V8.09.00 or later)
Compiler	CC-RL	V1.15.00
		(V1.12.00 or later)
Development assistance tool for	Standalone version of QE for Capacitive	V4.1.0
capacitive touch sensors	Touch	
Smart Configurator	RL78 Smart Configurator	V1.12.00
		(V1.5.00 or later)
Software integration system (SIS)	Capacitive sensing unit driver (r_ctsu)	V2.10
modules	 Touch middleware (rm_touch) 	(V2.10 or later)

Caution: When the free evaluation edition of CC-RL V1.12.00 or a later version is to be used for compilation during the tuning of touch sensors, set the optimization level of the compiler for building to "Debug precedence (-onothing)".

Table 2-2 Development Environment (Hardware)

Item	Description
Target MCU	RL78/G22 (R7F102GGE2DFB)
Target board	RL78/G22 fast prototyping board (RTK7RLG220C00000BJ)

Operation of the sample code attached to this application note was verified under the following conditions.

Table 2-3 Conditions for Verifying Operation

Item	Description	
Operating voltage	5.0 V	
	Level of voltage detection by LVD0 in reset mode	
	For rising: 2.67 V typ. (2.59 V to 2.75 V)	
	For falling: 2.62 V typ. (2.54 V to 2.70 V)	
Operating frequency	High-speed on-chip oscillator clock (fiH): 32 MHz	



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

2.1 Functions of QE for Capacitive Touch

QE for Capacitive Touch is a development tool for supporting initial settings and sensitivity adjustment of touch interfaces that are required in the development of embedded systems that use capacitive touch sensors.

The following shows the main functions of QE for Capacitive Touch.

- Creating touch interface configurations This allows the visual placement of touch-interface elements such as buttons and assignment of touch sensors (electrodes) to the elements.
- Tuning

This allows automatic offset and sensitivity tuning of the touch interface.

• Monitoring operation and adjusting parameters This allows monitoring of the operation of the touch interface and the fine adjustment of parameters.

Interface gene function		Tuning function	Monitoring function	
El Denforgane el Nationale El Denforgane el Naplinguno Desser Morti	X Septer/A/cer Sector Sec Sector Sec	T Automation from homorem X/R G (24 Minor networking for (Buttont)1 1533 @ config01). N high step pileses are normal loach pressure on the sensor for ance. Press any key on the PC keyboard to accept the sensitivity measurement. Buttonshi, 1532 @ config115119	Teste Over *	<mark>.</mark>
	Ver capacitati nettati Ve Berna Salar Jentinetti Dara sensati Berna Kry pal	Con	15449	
	Et toda over Texts and Data Pe TCPe Oppehres Server	ANNA	1444	
	Canad Server Discours for Render Dach (P		13440	

Figure 2-1 Main Functions of QE for Capacitive Touch



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

3. Configuring the Development Environment

This chapter describes the procedures for installing the development environment and setting up the hardware.

3.1 Installing Development Tools

The following tools are used with this sample application.

- CS+
- Standalone version of Smart Configurator
- Standalone version of QE for Capacitive Touch

If these tools have already been installed, skip the procedures in the rest of section 0.

3.1.1 Procedure for Installing the CS+ Integrated Development Environment

- Download the installer for the latest version of the CS+ for CC integrated development environment from the following link.
 CS+ IDE and Coding Tool | Renesas
- 2. Decompress the downloaded zip file and execute the installer file.
- 3. Click on "Begin CS+ Setup".
- 4. Check that "Tools for RL78 family" is selected.
- 5. After installation has been completed, click on the [Finish] button.

3.1.2 **Procedure for Installing the Standalone Version of the Smart Configurator**

- 1. Download the installer for the latest version of the RL78 Smart Configurator from the following link. <u>RL78 Smart Conifgurator | Renesas</u>
- 2. Execute the downloaded EXE file to start the installer.
- 3. After the installer has started, follow the instructions for installation as they appear on the screen.

3.1.3 Procedure for Installing the Standalone Version of QE for Capacitive Touch

- Download the installer for the latest version of QE for Capacitive Touch, a development assistance tool for capacitive touch sensors, from the following link. <u>QE for Capacitive Touch: Development Assistance Tool for Capacitive Touch Sensors | Renesas</u>
- 2. The downloaded zip file contains both the plug-in and standalone versions of QE. Extract the contents of the zip file and install the standalone version.
- Caution: Extract the contents in a location as close as possible to the root of a drive so that the pathname does not exceed the limit on the number of characters (260) in a pathname for Windows.

Example of a suitable location: Under the "C:\Renesas" folder

Do not specify the Windows folder, the Program Files folder, or a folder that has a name which includes white space.



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

3.2 Hardware Settings

This section describes the hardware settings and connection of the target board. Table 3-1 shows the jumper settings on the target board for this sample application. Power is supplied to the target board via the USB. Connect the target board to the PC through a USB cable as shown in Figure 3-1.

The changes required in circuit connection settings differ with the target board. For details, refer to the notes on usage of QE for Capacitive Touch in the user's manual of the FPB you are using.

Jumper	Circuit Group	Setting	Description
J16 ^{Note}	QE serial connection switching jumper	Open	The serial connection function of the QE tool is used.
		Closed	The COM port debugging circuit is enabled.
J17	Power supply selection header connector	Pins 1-2 closed	5-V power is selected.

Note: In this development procedure, the J16 setting needs to be switched for tuning and monitoring. For details, see steps 3 to 5 in section 8.4, Tuning.



Figure 3-1 Connecting the PC to the Target Board



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

4. Procedure of Application Development

This chapter describes how to develop an application.

Follow the steps in the workflow of QE for Capacitive Touch.



Figure 4-1 Procedure of Application Development

Table 4-1 lists the steps in the workflow. The section numbers in the table are linked to the corresponding sections in this document. Click on each section number in the table to see how to use the corresponding function. The IDE and Smart Configurator are used in creating a project, coding a program, building a project, and starting debugging.



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

ltem			Section Number
Preparation	To Create Project	Creation of a project by using the IDE	6
		Settings made by the Smart Configurator	7
		Clock and system	7.2
		CTSU driver	7.5.1
		Touch middleware	7.5.2
		Serial interface (UART)	7.5.3
		Voltage detector (LVD)	7.5.4
		Port functions (PORT)	7.5.5
		Board support package	7.5.6
		Unused pins	7.6
	To Select a Project F	older	8.2
	To Select an MCU N	ame	
Configuration	To Prepare a Config	uration	8.3
	To Output Files for T	uning	
	To Implement Progra	am	
	To Build Project		
Tuning	To Execute Program		8.4
	To Start Tuning		
	To Output Parameter	r Files	
Coding and Monitoring	To Implement Progra	am	8.5
	To Launch Debug		
	To Connect UART		
	To Enable Monitoring	g	

Table 4-1 Workflow of Development Using QE for Capacitive Touch



5. Sample Application

5.1 Overview of the Sample Application

This application note describes an example of an application that uses two buttons and one slider. This example also involves tuning and monitoring the touch performance through serial communications.

The method of creating an application that uses two buttons and a slider and monitoring whether a button or a slider is being touched are described in chapter 6 and subsequent sections.

Remark: Communications for checking the touch performance of a touch application can also be handled by an on-chip debugging (OCD) emulator. For devices of the RL78 family, however, note that the on-chip debugging functionality of the device limits performance in monitoring. Using serial communications enables smooth monitoring of the touch performance. Serial communications can also be used in tuning for the touch sensors.





This application note is provided with the two sets of sample code listed in Table 5-1. Both have the same procedure for development but differ in terms of some settings by the Smart Configurator and the code to be added to the qe_touch_sample.c file. Chapters 6 to 8 describe the procedure for development by taking the touch application implemented with the use of a software timer as an example. For implementation of the application with the use of a hardware timer and the control of LEDs to indicate sensor operation, see chapter 9, Another Implementation of the Sample Application.

Table 5-1 Overview of the Attached Sample Code

File Name	Timer for Generating Cycles of Touch Measurement	LED Control
Capacitive_Touch_Project_Example	Software timer	None
Capacitive_Touch_Project_HardwareTimer_Example	Hardware timer	Included



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

5.2 List of Pins Used

Table 5-2 lists the pins used in this sample application.

UART communications and touch sensors should be set up according to the specifications of the target board in use.

Table 5-2 List of Pins Used in the Sample Application

Item	Pin	Description
UART communications	RxD0/P11	For tuning and monitoring
	TxD0/P12	
Touch sensor 1	TS24/P26	For detecting touching of the TS_B1 button
Touch sensor 2	TS23/P25	For detecting touching of the TS_B2 button
Touch slider	TS20/P22	For detecting the position of a finger moving
	TS21/P23	left or right on the TS_S slider
	TS22/P24	

Figure 5-2 shows the locations of the touch sensors used in this sample application.



Figure 5-2 Locations of the Touch Sensors



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

6. Creating a New Project

Start CS+ and create a new project.

For this sample application, make the following selections in the [Create Project] dialog box.

- Microcontroller
- Using microcontroller : R7F102GGExFB (48 pins)

: RL78

- Kind of project
- : Application (CC-RL) : (Desired project name)
- Project name Place
- : (Desired place)
- lace : (Desired pl

Create Project			×
Microcontroller:	RL78		~
Using microcontroller:			
(Search microcontro	ller) Update		
R7F102GBE R7F102GE R7F102GE R7F102GE R7F102GE R7F102GE R7F102GGE R7F102GGE R7F102GG2 R7F102GG2 R7F102G3 (ROM	xNP(32pin) xLA(36pin) xNP(40pin) xFP(44pin) xFP(44pin) xNP(48pin) :96KB)	R7F102GGExFB ze[KBytes]:64 ze[Bytes]:4096	
Kind of project:	Application(CC-RL)		~
Project name:	Capacitive_Touch_Project_Exam	ole 🖊 🗖	
Place:	C:\CS+_Workspace	~	Browse
	Make the project folder		
C:\CS+_Workspace\Cap	acitive_Touch_Project_Example\Capaciti	ve_Touch_Project_Exa	mple.mtpj
Pass the file composit	tion of an existing project to the new proje	at .	
Project to be passed:	(Input project file to be diverted.)	\sim	Browse
Copy composition file	es in the diverted project folder to a new p	oject folder.	
To dialog with multi-	-core option Create	Cancel	Help

Figure 6-1 Creating a New Project



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7. Settings of the Smart Configurator

This chapter describes the procedures for using the Smart Configurator to make settings. The settings required for this sample application are listed below.

- Clock and system
- CTSU driver
- Touch middleware
- Serial interface (UART communications)
- Voltage detector (LVD)
- Port functions (PORT)

7.1 Starting the Smart Configurator

Double-click on "Smart Configurator" in [Project Tree] of CS+ to start the Smart Configurator.



Figure 7-1 Starting the Smart Configurator

If the Smart Configurator does not start, check that the following two items are correctly specified.

- A correct file path is specified in [Property] of the Smart Configurator.
- "Smart Configurator for RL78 Communication Plug-in" is selected in the [Plug-in Manager] dialog box that can be opened from the [Tool] menu.

	_		
Project Tree 7 🗙		Property	* X
2 7 2 2	-	Smart Configurator Property	+ - 9
Capacitive Touch Project Example (Project)	~	Product Information	
R7F102GGExFB (Microcontroller)		Version	V1.00.10.02 [12 Nov 2024]
🎬 Smart Configurator (Design Tool)		Smart configurator setting	
🔨 CC-RL (Build Tool)		Smart Configurator for RL78 executable file path	C.*Program Files (x86)*Renesas Electronics*SmartConfigurator*RL78*eclipse*SmartConfigurator.exe
🗊 - 🗊 File			

Figure 7-2 Path to the Smart Configurator File

Plug-in Manager	×	
Checked plug-ins are loaded at the CS+ start-up. These settings are enabled at the next start-up. * You can never uncheck a check box of the grayout plugin that is recommended that the checkboxes of the plug-in for the target mic Basic Function Additional Function		
Module Name	Description	
Code Generator Plug-in	Plug in to generate the device driver automatically.for V8 Plug in to generate the device driver automatically and to	
Code Generator/PinView Plug-in	Plug-in to generate the device driver automatically and to	
Debug Console Plug-in	DebugConsole plug-in to support using standard I/O.	
Editor plug-in DLL Image: Interplug-in DLL Image: Interplug-in DLL	SEditor DLL It is a console where the IronPython commands and the C	
Pin Configurator Plug-in	Plug-in to define the device pin configuration.	
🗹 🕀 Program Analyzer Plug-in	Plug-in to analyze program.	
Quick and Effective tool solution - QE	Plug-in for application development that contains useful to	
Smart Configurator for RL78 Communication Plug-in	Plug-in to communicate with Smart Configurator for RL781	

Figure 7-3 Plug-in Manager



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7.2 Setting the Clock and System

This section describes the procedure for setting up the clock and system.

1. After starting the Smart Configurator, select the [Clocks] tab at the bottom of the Smart Configurator view and set up the clock. If the target MCU requires the use of EVDD, select an appropriate value for "EVDD setting" according to the operating mode.

Clocks configuration
Operation mode: High-speed on-chip oscillator If High-speed on-chip oscillator High-speed on-chip oscillator If HoCO tast setting: Normal If Middle-speed on-chip oscillator If HoCO tast setting: If HoCO tast setting: If HoCO tast setting: If HoCO tast setting: If HoCO tast setting:

Figure 7-4 Setting the Clocks

Operation mode:High-speed main mode $4.0(V) \sim 5.5(V)$ EVDD setting: $4.0 V \leq EVDD0 \leq 5.5 V$			
EVDD setting: $4.0 \text{ V} \leq \text{EVDD0} \leq 5.5 \text{ V}$	Operation mode:	High-speed main mode 4.0(V)~5.5(V)	-
	EVDD setting:	$4.0 \text{ V} \leq \text{EVDD0} \leq 5.5 \text{ V}$	•

Figure 7-5 Setting EVDD

2. Select the [System] tab and set up the debugging environment.

System configuration Generate Code Generate Rep	*Capacitive_Touch_Project_Example.s	cfg ×		6	
On-chip debug operation setting Unused Use emulator COM Port Emulator setting E2 © E2 Lite Pseudo-RRM/DMM function setting Used Start/Stop function setting Used Monitoring point function setting Used Start/Stop Linear Used Start/Stop L	System configuration				
On-chip debug operation setting Unused Use emulator COM Port Emulator setting E2 © E2 Lite Pseudo-RRM/DMM function setting Used Start/Stop function setting Used Monitoring point function setting Used Monitoring point function setting Used Security ID setting					
O Unused O Use emulator COM Port Emulator setting E2 E2 Ite Pseudo-RRM/DMM function setting O Unused Ised Start/Stop function setting O Unused O Used Monitoring point function setting O Used Security ID setting O Used	 On-chip debug setting 				
E2 • E2 Lite Pseudo-RRM/DMM function setting • Used Start/Stop function setting • Used • Unused • Used Monitoring point function setting • Used • Unused • Used		O Use emulator	COM Port		
Unused Image: Used Start/Stop function setting Used Image: Unused Used Monitoring point function setting Used Image: Unused Used	-	E2 Lite	•		
Unused Used Monitoring point function setting Unused Used Security ID setting		Used			
© Unused Used Security ID setting		⊖ Used			
		⊖ Used			
Deselect this box.	Use security ID	Deselect this box.			
Security ID authentication failure setting O Do not erase flash memory data © Erase flash memory data	O Do not erase flash memory data	ting			

Figure 7-6 Settings for Debugging



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7.3 Downloading Software Integration System (SIS) Modules

This section describes how to download two SIS modules, the CTSU driver and touch middleware, which are necessary to implement a touch application. If they have already been installed, skip the steps in the rest of this section.

1. Select the [Components] tab and click on the 😜 icon.



Figure 7-7 [Software component configuration] View

2. Click on "Download RL78 Software Integration System modules" at the bottom of the [New Component] dialog box.

Software Component Selection Select component from those available in list Category All Runction All Filter Image: Code Generator 16.0 Components Short Name Type Wyber Sensing Unit driver. r_ctsu RL78 Software 180 Code Generator 16.0 Code Generator 16.0 Code Generator 16.0 Code Generator 16.0 Code Generator 16.0 Code Generator 16.0 Data Transfer Controller Code Generator 16.0 Code Generator 16.0 Event Link Controller Code Generator 16.0 Code Generator 16.0 Event Link Controller Code Generator 16.0 Code Generator 16.0 Event Link Controller Code Generator 16.0 Code Generator 16.0 Event Link Controller Code Generator 16.0 Code Generator 16.0 Event Link Controller Code Generator 16.0 Code Generator 16.0 Event Link Controller Code Generator 16.0 Code Generator 16.0 Event Link Controller Code Generator 16.0 Code Generator 16.0 Event Link Controller Code Generator 16.0 Code Generator 16.0 Event Link Controll	New C	omponent				×
Select component from those available in list Category All Filter Image: Components in the second						<u></u>
Select component from those available in list Category All Function All Filter Short Name Type Version All Components Type Version No Board Support Packages v1.80 r_bsp RL78 Software 1.80 Capacitive Sensing Unit driver. r_ctsu RL78 Software 1.80 Code Generator 1.6.0 External Event Counter Code Generator Event Link Controller Code Generator II C Communication (Master mode) Code Generator Bit IC Communication (Master mode) Code Generator Bit IDC Communication (Master mode) Code Generator	Software	Component Selection			E.	
All Filter Components Short Name Type Version # A/D Converter Code Generator 1.6.0 # Board Support Packages v1.80 r_bsp RL78 Software 1.80 # Cock Output /Buzzer Output Controll F_ctsu RL78 Software 1.60 # Deata Transfer Controller Code Generator 1.6.0 # Delay Counter Code Generator 1.6.0 # Divider Function Code Generator 1.6.0 # External Event Counter Code Generator 1.6.0 # It Communication (Master mode) Code Generator 1.6.0 # It Communication (Slave mode) Code Generator 1.6.0 # Input Signal High-/Low-Level Width Code Generator 1.6.0 > Show only latest version Description Code Generator 1.6.0 The analog to digital (A/D) converter is function for converting analog inputs to digital signals. Download RL78 Software Integration System modules Configure general settings	Select co	mponent from those available in	list			-
Filter Components [^]	Category	All				~
Filter Components [^]	Function	All				~
Components Short Name Type Version If A/D Converter Code Generator 1.6.0 If Board Support Packages v1.80 r_bsp RL78 Software 1.80 If Clock Output /Buzzer Output Controll Code Generator 1.6.0 If Deata Transfer Controller Code Generator 1.6.0 If Delay Counter Code Generator 1.6.0 If Divider Function Code Generator 1.6.0 If Event Link Controller Code Generator 1.6.0 If Ukider Function Code Generator 1.6.0 If Ukider Function Code Generator 1.6.0 If Ukider Function Code Generator 1.6.0 If Ukider Function (Master mode) Code Generator 1.6.0 If It Communication (Master mode) Code Generator 1.6.0 If Input Signal High-/Low-Level Width Code Generator 1.6.0 If Input Signal High-/Low-Level Width Code Generator 1.6.0 Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. Software Integration System modules Download RL78 Software Integration System modules Configure general						_
Components Short Name Type Version Iff A/D Converter Code Generator 1.6.0 Iff Capacitive Sensing Unit driver. r_ctsu RL78 Software 1.8.0 Iff Clock Output/Buzzer Output Controll Code Generator 1.6.0 Iff Deata Transfer Controller Code Generator 1.6.0 Iff Deata Transfer Controller Code Generator 1.6.0 Iff Delay Counter Code Generator 1.6.0 Iff Delay Counter Code Generator 1.6.0 Iff Event Link Controller Code Generator 1.6.0 Iff Event Link Controller Code Generator 1.6.0 Iff Iff Communication (Master mode) Code Generator 1.6.0 Iff Input Signal High-/Low-Level Width Code Generator 1.6.0 Iff Input Signal High-/Low-Level Width Code Generator 1.6.0 ✓ Iff Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. Download RL78 Software Integration System modules Configure general settings	Filter					
# Board Support Packages v1.80 r_bsp RL78 Software 1.80 # Capacitive Sensing Unit driver. r_ctsu RL78 Software 2.10 # Clock Output / Buzzer Output Controll Code Generator 1.6.0 # Data Transfer Controller Code Generator 1.6.0 # Delay Counter Code Generator 1.6.0 # Divider Function Code Generator 1.6.0 # External Event Counter Code Generator 1.6.0 # IIC Communication (Master mode) Code Generator 1.6.0 # IIC Communication (Slave mode) Code Generator 1.6.0 # Input Signal High-/Low-Level Width Code Generator 1.6.0 > Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals.	Compor	ients	Short Name	Туре	Version	^
B Capacitive Sensing Unit driver. r_ctsu RL78 Software 2.10 Cock Generator 1.6.0 Data Transfer Controller Cock Generator 1.6.0 Description Cock Generator 1.6.0 Cock Generator 1.6.0 Event Link Controller Cock Generator 1.6.0 Event Link Communication (Master mode) Cock Generator 1.6.0 Event Link Communication (Slave mode) Cock Generator 1.6.0 Event Link Controller Input Signal High-/Low-Level Width Cock Generator 1.6.0 Event Link Controller Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. Download RL78 Software Integration System modules Configure general settings	₿A/D C	onverter		Code Generator	1.6.0	
# Clock Output /Buzzer Output Controll Code Generator 1.6.0 # Data Transfer Controller Code Generator 1.5.0 # Delay Counter Code Generator 1.6.0 # Divider Function Code Generator 1.6.0 # Event Link Controller Code Generator 1.6.0 # Event Link Controller Code Generator 1.6.0 # Event Link Controller Code Generator 1.6.0 # Ill C Communication (Master mode) Code Generator 1.6.0 # Ill C Communication (Slave mode) Code Generator 1.6.0 # Ill C Communication (Slave mode) Code Generator 1.6.0 # Ill Communication (Slave mode) Code Generator 1.6.0 # Input Pulse Interval/Period Measurem Code Generator 1.6.0 # Input Signal High-/Low-Level Width Code Generator 1.6.0 Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals.	# Board	Support Packages v1.80	r_bsp	RL78 Software	1.80	
# Data Transfer Controller Code Generator 1.5.0 # Delay Counter Code Generator 16.0 # Divider Function Code Generator 16.0 # Event Link Controller Code Generator 1.6.0 # Event Link Controller Code Generator 1.6.0 # Event Link Controller Code Generator 1.6.0 # Ilc Communication (Master mode) Code Generator 1.6.0 # Ilc Communication (Slave mode) Code Generator 1.6.0 # Ilput Pulse Interval/Period Measurem Code Generator 1.6.0 # Input Signal High-/Low-Level Width Code Generator 1.6.0 Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. V Download RL78 Software Integration System modules V Configure general settings V	H Capad	itive Sensing Unit driver.	r_ctsu	RL78 Software	2.10	
# Delay Counter Code Generator 1.6.0 Code Generator 1.6.0 Event Link Controller Code Generator 1.6.0 Code Generator 1.6.0 External Event Counter Code Generator 1.6.0 Code Generator 1.6.0 Code Generator 1.6.0 U Il IC Communication (Master mode) Code Generator 1.6.0 U Il IC Communication (Slave mode) Code Generator 1.6.0 U Il IC Communication (Master mode) Code Generator 1.6.0 U Il ID Communication (Master mode) Code Generator 1.6.0 U Il IC Communication (Master mode) Code Generator 1.6.0 U Input Pulse Interval/Period Measurem Code Generator 1.6.0 V Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. Download RL78 Software Integration System modules Configure general settings	# Clock	Output /Buzzer Output Controll		Code Generator	1.6.0	
# Divider Function Code Generator 1.6.0 # Event Link Controller Code Generator 1.3.1 # External Event Counter Code Generator 1.6.0 # IIC Communication (Master mode) Code Generator 1.6.0 # IIC Communication (Slave mode) Code Generator 1.6.0 # Input Pulse Interval/Period Measurem Code Generator 1.6.0 # Input Signal High-/Low-Level Width Code Generator 1.6.0 Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. Download RL78 Software Integration System modules Configure general settings	🖶 Data '	fransfer Controller		Code Generator	1.5.0	
# Event Link Controller Code Generator 1.3.1 # External Event Counter Code Generator 1.6.0 # ILC Communication (Master mode) Code Generator 1.6.0 # ILC Communication (Slave mode) Code Generator 1.6.0 # Input Pulse Interval/Period Measurem Code Generator 1.6.0 # Input Signal High-/Low-Level Width Code Generator 1.6.0 * Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. Download RL78 Software Integration System modules Configure general settings	# Delay	Counter		Code Generator	1.6.0	
# External Event Counter Code Generator 1.6.0 # IIC Communication (Master mode) Code Generator 1.7.0 # IIC Communication (Slave mode) Code Generator 1.6.0 # Input Pulse Interval/Period Measurem Code Generator 1.6.0 # Input Signal High-/Low-Level Width Code Generator 1.6.0 > Imput Signal High-/Low-Level Width Code Generator 1.6.0 > Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals.	# Divide	er Function		Code Generator	1.6.0	
# IIC Communication (Master mode) Code Generator 1.7.0 # IIC Communication (Slave mode) Code Generator 1.6.0 # Input Pulse Interval/Period Measurem Code Generator 1.6.0 # Input Signal High-/Low-Level Width Code Generator 1.6.0 > Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. Download RL78 Software Integration System modules Configure general settings	# Event	Link Controller		Code Generator	1.3.1	
# IIC Communication (Slave mode) Code Generator 1.6.0 # Input Pulse Interval/Period Measurem Code Generator 1.6.0 # Input Signal High-/Low-Level Width Code Generator 1.6.0 > Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. Download RL78 Software Integration System modules Configure general settings 	# Extern	al Event Counter		Code Generator	1.6.0	
# Input Pulse Interval/Period Measurem Code Generator 1.6.0 v # Input Signal High-/Low-Level Width Code Generator 1.6.0 v Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. 0 Download RL78 Software Integration System modules v v	HIC Co	mmunication (Master mode)		Code Generator	1.7.0	
Imput Signal High-/Low-Level Width Code Generator 1.6.0 > Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals.	HIC Co	mmunication (Slave mode)		Code Generator	1.6.0	
Show only latest version Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. Download RL78 Software Integration System modules Configure general settings	# Input	Pulse Interval/Period Measurem		Code Generator	1.6.0	
Description The analog to digital (A/D) converter is function for converting analog inputs to digital signals. Download RL78 Software Integration System modules Configure general settings	# Input	Signal High-/Low-Level Width		Code Generator	1.6.0	\checkmark
The analog to digital (A/D) converter is function for converting analog inputs to digital signals.	Show (only latest version				
signals. Download RL78 Software Integration System modules Configure general settings	Description	วท				
Configure general settings		og to digital (A/D) converter is fu	unction for convertion	ng analog inputs to digi	tal	
Configure general settings	Downloa	d RL78 Software Integration Syste	em modules			
(?) < Back Next > Finish Cancel	contrigue	general seconds.				
(?) < Back Next > Finish Cancel						
	?	< Back	Next >	Finish	Cancel	

Figure 7-8 [New Component] Dialog Box



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

- 3. A dialog box will open. Select the following items and click on the [Download] button.
 - RL78 Family CTSU Module Software Integration System
 - RL78 Family TOUCH Module Software Integration System

Г	Title	Document No.	Rev.	Issue date	^	Select All
	RL78 Family FS3000 Sensor Control Mod	R01AN6195EJ	Rev.1.00	2022-06-30		Deselect A
	RL78 Family FS1015 Sensor Control Mod	R01AN6198EJ	Rev.1.00	2022-06-30		Deselect A
	RL78 Family HS400X Sensor Control Mo	R01AN6446EJ	Rev.1.00	2022-06-30		
E	RL78 Family HS300x Sensor Control Mo	R01AN6194EJ	Rev.1.20	2022-05-20		
	RL78 Family CTSU Module Software Inte	R11AN0484EJ	Rev.2.10	2022-04-20		
	RL78 Family TOUCH Module Software In	R11AN0485EJ	Rev.2.10	2022-04-20		
C	RL78 Family Sensor I2C Communication	R01AN6193EJ	Rev.1.10	2022-03-02		
	RL78 Family FS2012 Sensor Control Mod	R01AN6196EJ	Rev.1.10	2022-03-02		
	RL78 Family ZMOD4410 and ZMOD4510	R01AN6197EJ	Rev.1.10	2022-03-02		
	RL78 Family Board Support Package Mo	R01AN5522EJ	Rev.1.20	2022-02-28	~	
M	odule Folder Path:					
						Browse

Figure 7-9 Downloading SIS Modules

Caution: If the TOUCH module or CTSU module does not appear in the above dialog box for downloading, download them by using the procedure below. Download the modules from the Renesas Web site and use the procedures described on the site to add the files to the folder for storing the downloaded SIS modules.

For the web pages of the individual modules and how to download and display the modules in the dialog box, refer to the following.

 Web pages for downloading the CTSU module and TOUCH module RL78 Family CTSU Module Software Integration System RL78 Family CTSU Module Software Integration System Rev.2.10 - Sample Code | Renesas

RL78 Family TOUCH Module Software Integration System RL78 Family TOUCH Module Software Integration System Rev.2.10 - Sample Code | Renesas

How to download and use the modules
 <u>How to Download and Use a SIS Module</u>



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7.4 Adding Components

1. Select the components shown below in the Smart Configurator.

Software Component Selection Select component from those available in list	
Select component from those available in list	-
Category All	
Function All	
Filter	
Components Short Name Type	Version
# A/D Converter Code Generati	
Board Support Packages v1.80 r_bsp RL78 Software	
Gapacitive Sensing Unit driver. r_ctsu RL78 Software	
Clock Output /Buzzer Output Controll Code Generation	r 1.6.0
Data Transfer Controller Code Generation	
Delay Counter Code Generate	
Divider Function Code Generate	
Event Link Controller Code Generate	
# External Event Counter Code Generate	r 1.6.0
# IIC Communication (Master mode) Code Generati	r 1.7.0
# IIC Communication (Slave mode) Code Generati	r 1.6.0
Hinput Pulse Interval/Period Measurem Code Generate	r 1.6.0
Hinput Signal High-/Low-Level Width Code Generation	r 1.6.0
H Interrupt Controller Code Generate	r 1.6.0
# Interval Timer Code Generati	r 1.6.0
# Key Interrupt Code Generati	r 1.5.0
B One-Shot Pulse Output Code Generation	r 1.6.0
Ports Code Generate	r 1.6.0
PWM Output Code Generate	r 1.8.0
Beal-Time Clock Code Generate	r 1.6.0
SNOOZE Mode Sequencer Graphical Con	i 1.3.2
BPI (CSI) Communication Code Generate	r 1.6.0
Square Wave Output Code Generati	r 1.6.0
Touch middleware. rm_touch RL78 Software	2.10
UART Communication Code Generate	r 1.8.0
By Voltage Detector Code Generate	r 1.5.0
B Watchdog Timer Code Generate	r 1.6.0
Show only latest version	
Description	
Dependency : r_bsp version(s) 1.70	
The CTSU module is a CTSU driver for the Capacitive Sensing Unit. The CTSU mod	
configured assuming access via the Touch middleware layer, but can also be acce	sed from
Download RL78 Software Integration System modules	
Configure general settings	

Figure 7-10 Software Component Selection

Note: For the RL78/G16, this component does not require addition in this way because the method of setting the voltage detector function differs from that for the other devices. For the method of voltage detector setting for the RL78/G16, see section 7.5.4.



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

2. Assign resources to the selected components. Use the following settings for this sample application.

Add new configuration for selected component	Komponent		– 🗆 X
Configuration name: Config_UART0 UART Communication Configuration name: Config_UART0 Operation: Transmission/reception Resource: UART0 Voltage Detector Configuration name: Config_LVD0 Resource: IVD0	Add new configurat	ion for selected component	
Configuration name: Config_DORT			
Resource: PORT UART Communication Configuration name: Config_UARTO Operation: Transmission/reception Resource: UARTO Voltage Detector Configuration name: Config_LVD0 Resource: LVD0		Config PORT	
Configuration name: Config_UART0 Operation: Transmission/reception Resource: UART0 Voltage Detector Configuration name: Configuration name: Config_LVD0 Resource: LVD0			~
Configuration name: Config_UART0 Operation: Transmission/reception Resource: UART0 Voltage Detector Configuration name: Configuration name: Config_LVD0 Resource: LVD0			
Resource: UARTO Voltage Detector Config.LVDO Configuration name: Config.LVDO Resource: V/DO			
Voltage Detector Config_LVD0 Resource: LVD0 ✓		Transmission/reception	~
Configuration name: Config_LVD0 Resource: LVD0 ~	Resource:	UARTO	~
Config_LVD0 Resource: LVD0	Voltage Detector		
		Config_LVD0	
	Resource:	LVD0	~
(2) Cariel Finish			
2 Pack Next > Finish			
(r) < Back INext > Finish Cancel	(?)	< Back Next > Fit	nish Cancel

Figure 7-11 Assigning Resources to Components

The components are added as shown below.

Software component configuration	Gene	🔋 👜 erate Code 🛛 Generate R	leport
Components 🚵 🖾 🖾 🗄	Configure		i
type filter text	Property v	Value	^
V 🗁 Startup	# Parameter check	Use system default	
v 🗁 Generic	# Data transfer of INTCTSUWR and INTCTSURD	Interrupt handler	
💣 r_bsp	# DTC setting	Setting in r_ctsu	
✓ ➢ Drivers	# Select auto judgement	Disable	
Power management and reset function	# Data storage address setting for CTSUWR	0xFF300	
Config_LVD0	# Variable address setting for g_ctsu_self_raw.	0xFF400	
✓ I/O port	# Variable address setting for g_ctsu_mutual_raw	0xFF500	
Config_PORT	# Data storage address setting for CTSUAJTHR.	0xFF600	
Communications	# Data storage address setting for CTSUAJMMAR.	0xFF700	
Config_UART0	# Data storage address setting for CTSUAJBLACT.	0xFF800	\sim
✓ ➢ Middleware	<	>	
🗸 🗁 Generic			\sim
💣 r_ctsu			
💣 rm_touch			~
			~

Figure 7-12 Software Component Configuration (after the Components are Added)



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7.5 Modifying the Component Settings in the Smart Configurator

Set up the components added in the previous steps.

7.5.1 Setting the CTSU Component

Click on the "r_ctsu" module and enable the TSCAP pin and five TS pins to be used by this sample application. For the correspondence between the TS pins and touch sensors, refer to the user's manual of the target board you are using.



Figure 7-13 Enabling the TSCAP Pin and TS Pins to be Used by the Application



RL78 Family Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

For the TS pins that are not to be used by the application, setting them as outputs of the low level is recommended. For the CTSU2 module, if the TS pins not to be used by the application are enabled, the Smart Configurator automatically handles them as non-measurement pins and sets them as outputs of the low level.

Therefore, in the attached two sets of sample code, all TS pins except the TS12 and TS13 pins, which are for assignment to a different multiplexed function (UART0), are enabled even if they are not to be used by the application.

Caution: In designing circuits for a user board, appropriately handle the pins such that the electrical characteristics are satisfied.



Figure 7-14 Enabling the TS Pins Not to be Used by the Application



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7.5.2 Setting the Touch Component

Click on the "rm_touch" module and make the following settings.

- Support QE monitor using UART: Enable
- Support QE tuning using UART: Enable
- UART channel: UART0

The UART channel to be used differs with the target board. For details, refer to the circuit diagram of the FPB you are using.



Figure 7-15 Setting rm_touch



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7.5.3 Setting the UART Communications Component

This section describes the procedures for setting up the UART to be used for tuning and monitoring the touch sensors.

The UART channel and port to be used differ with the target board. Click on the added UART communications module and select the operating clock and transfer rate on the [Transmission] and [Reception] tabbed pages according to the target transfer rate.

Make the specifications shown in the figure below for this sample application.

*Capacitive_Touch_Project_Example.scfg ×			
Software component configuration		Gene	🔞 👜 rate Code Generate Report
Components 🚵 🖄 🕒 🗄	Configure		
54 (g) 👘 😜	Transmission		
type filter text	UART0 clock setting		
🗸 🗁 Startup	Operation clock	СК00 ~	
✓ Generic	Clock source	fCLK/2^3 v uency: 4000 kHz)	
✓	Transfer mode setting		
 Power management and reset function Config_LVD0 	Single transfer mode	Continuous transfer mode	
✓ ⇒ I/O port	Data length setting	٩	
Config_PORT	○7 bits ● 8 bits	⊖ 9 bits	
✓	Transfer direction setting		
v 🗁 Middleware	LSB	⊖ MSB	
✓ ≥> Generic arr r_ctsu	Parity setting	0.001	
i_ctuu i rm_touch	None O parity	Odd parity Even parity	
	Stop bit length setting	🔿 2 bits	
	Transfer data level setting		
	Non-reverse	○ Reverse	
	Transfer rate setting		
	Transfer rate setting	153600 v (bps) t error: 0.16%)	
	Interrupt setting		
	Transmit end interrupt priority (INTST0)	Level 3 (Iow) V	
	Callback function setting		
	Transmission end		· · · · · · · · · · · · · · · · · · ·
Overview Board Clocks System Components Pins Inte	mpt		×
영 Smart Configurator ile Window Help 역 En III	nupr		×
Smart Configurator Window Help Configuration Configurati	nupu		
Smart Configurator Welp Capacitive_Touch_Project_Example.scfg × Software component configuration		Gene	
Image: Signature of the second se	Configure	Gene	
Smart Configurator ie Window Help Image: Second S	Configure Transmission Reception	Gene	
Smart Configurator lee Window Help Image: Second	Configure Transmission Reception UARTO clock setting	4	
Smart Configurator ile Window Help Image: Software component configuration Components Image: Software Component Configuration Components Image: Software Component Configuration Image: Software Component Configuration Image: Software Component Configuration Components Image: Software Component Configuration Image: Software Component Configuration Image: Software Configuration <td< td=""><td>Configure Transmission Reception UARTO clock setting Operation clock</td><td>CK00</td><td></td></td<>	Configure Transmission Reception UARTO clock setting Operation clock	CK00	
Smart Configurator lee Window Help Image: Software component configuration Components Image: Software Component configuration Components Image: Software Component configuration Image: Software component configuration Image: Software Component configuration Components Image: Software Component configuration Image: Software component configuration Image: Software Component configuration	Configure Transmission Reception UARTO clock setting Operation clock Clock source	4	
Smart Configurator ile Window Help Image: Component configuration Components Image: Component configuration Components Image: Component configuration I	Configure Transmission Reception UART0 clock setting Operation clock Clock source Data length setting	CK00 ~ FCLK/2^3 ~ uency: 4000 kHz)	
Smart Configurator lee Window Help Image: Software component configuration Components Image: Software Component configuration Components Image: Software Component configuration Image: Software component configuration Image: Software Component configuration Components Image: Software Component configuration Image: Software component configuration Image: Software Component configuration	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting O 7 bits	CK00	
Smart Configurator ile Window Help Image: Software component configuration Components Image: Software component configuration Components Image: Software component configuration Vipe filter text Image: Software component configuration Vipe filter text Image: Software component configuration V >> Startup Image: Software component configuration V >> Software component configuration Image: Software component configuration V >> Drivers Image: Software component configuration Image: Config_LVD0 Image: Vipert config_LVD0 Image: Vipert config_LVD0 Image: Vipert config_LVD0	Configure Transmission Reception UART0 clock setting Operation clock Clock source Data length setting	CK00 ~ FCLK/2^3 ~ uency: 4000 kHz)	
Smart Configurator lee Window Help Capacitive_Touch_Project_Example.scfg × Software component configuration Components Software component configuration Components Software component configuration Components Software component configuration Software component configur	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting O T bits	CK00 tCLK/2^3 0 9 bits	
Smart Configurator ile Window Help Image: Software component configuration Components Image: Software component configuration Components Image: Software component configuration Vipe filter text Image: Software component configuration Vipe filter text Image: Software component configuration V >> Startup Image: Software component configuration V >> Software component configuration Image: Software component configuration V >> Drivers Image: Software component configuration Image: Config_LVD0 Image: Vipert config_LVD0 Image: Vipert config_LVD0 Image: Vipert config_LVD0	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting O 7 bits	CK00 tCLK/2^3 0 9 bits	
Smart Configurator lee Window Help Capacitive_Touch_Project_Example.scfg × Software component configuration Components Software component configuration Components Software component configuration Components Software configuration Components Software configuration Components Software configuration Config_LVD0 × B/VC port Config_LVD0 × B/VC port Config_LVD0 × B/CO port Config_LVD0 × B/CO port Config_LVD0 × B/CO port Config_LVD0 × B/CO port Config_LVD0 × B/CO port Config_LVART0 × Generic	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting 7 bits B bits Transfer direction setting E LSB Parity setting None 0 parity Stop bit length setting	CK00 fCLK/2^3 v uency: 4000 kHz) O 9 bits O MSB	
Smart Configurator ie Window Help Image: Capacitive_Touch_Project_Example.scfg × Software component configuration Components Image: Imag	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting O 7 bits B bits Transfer direction setting Stab bits Parity setting None O parity Stop bit length setting 1 bit fixed	CK00 fCLK/2^3 v uency: 4000 kHz) O 9 bits O MSB	
Smart Configurator ile Window Help Image: Configuration Software component configuration Components Image: Configuration Components Image: Configuration	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting 7 bits Bata length setting LSB Parity setting None O parity Stop bit length setting bit fixed Receive data level setting	CK00 rCLK/2^3 9 bits MSB Odd parity Even parity	
Smart Configurator ile Window Help Image: Configuration Software component configuration Components Image: Configuration Components Image: Configuration	Configure Transmission Reception UART0 clock setting Operation clock Clock source Data length setting 7 bits	CK00 fCLK/2^3 v uency: 4000 kHz) O 9 bits O MSB	
Smart Configurator ile Window Help Image: Configuration Software component configuration Components Image: Configuration Components Image: Configuration	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting 7 bits Bata length setting LSB Parity setting None O parity Stop bit length setting bit fixed Receive data level setting	CK00 rCLK/2^3 9 bits MSB Odd parity Even parity	
Smart Configurator ile Window Help Image: Configuration Software component configuration Components Image: Configuration Components Image: Configuration	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting O 7 bits Batis Transfer direction setting Sab Parity setting None O parity Stop bit length setting 1 bi fixed Receive data level setting Non-reverse Transfer rate setting	CK00 FCLK/2*3 9 bits MSB Odd parity Even parity Reverse	TO Code Generate Report
Smart Configurator ile Window Help Image: Configuration Software component configuration Components Image: Configuration Components Image: Configuration	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting O 7 bits Batis Transfer direction setting Sab Parity setting None O parity Stop bit length setting 1 bi fixed Receive data level setting Non-reverse Transfer rate setting	CK00 FCLK/2^3 9 bits MSB Odd parity Even parity Reverse 153600 (bps)	TO Code Generate Report
Smart Configurator ile Window Help Image: Configuration Software component configuration Components Image: Configuration Components Image: Configuration	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting 7 h its 8 & b its Transfer direction setting 9 LSB Parity setting 9 None 0 parity 5 top bit length setting 1 bit fixed Receive data level setting 9 Non-reverse 1 Transfer rate setting 1 Transf	CK00 FCLK/2^3 9 bits MSB Odd parity Even parity Reverse 153600 (bps)	TO Code Generate Report
Smart Configurator ile Window Help Image: Configuration Software component configuration Components Image: Configuration Components Image: Configuration	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting 7 bits Barity setting ELSB Parity setting None Clock bit length setting Data length setting Clock source Data length setting	CK00 (CK02 Uuency: 4000 kHz) 9 bits 0 M5B 0 Odd parity Even parity Reverse 153600 (bps) (bps) (current error: 0.16%, the minimum is -4.49%, the maximum is 4.4	TO Code Generate Report
Smart Configurator ile Window Help Image: Configuration Software component configuration Components Image: Configuration Components Image: Configuration	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting O tata length setting O tata length setting O tata length setting O tata length setting O parity Stop bit length setting I bit fixed Receive data level setting Transfer rate setting Tr	CK00 Image: CK00 fCLK/2^3 Image: CK000 kHz 9 bits Image: CK000 kHz MSB Image: CK000 kHz Odd parity Even parity Reverse Image: CK000 kHz [153600 Image: CK000 kHz (Current error: 0.16%, the minimum is -4.49%, the maximum is 4.4 Level 3 (low) Image: CK000 kHz	TO Code Generate Report
Smart Configurator ile Window Help Image: Configuration Software component configuration Components Image: Configuration Components Image: Configuration	Configure Transmission Reception UARTO clock setting Operation clock Clock source Data length setting O tata length setting O tata length setting O tata length setting O tata length setting O parity Stop bit length setting I bit fixed Receive data level setting O non-reverse Transfer rate setting Transfer rate setting Transfer rate setting Reception error interrupt priority (INTSR0) I Reception error interrupt priority (INTSR0) I Reception error interrupt priority (INTSR0)	CK00 rCLK/2^3 9 bits O MSB O Odd parity Reverse 153600 (bps) (current error: 0.16%, the minimum is -4.49%, the maximum is 4.49%, the maximum is 4.49\%, the maximum is 4.49\%, the maximum is 4.49\%, the maximum is 4.	TO Code Generate Report

Figure 7-16 Setting the UART Communications Component (UART0)



RL78 Family Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

After that, select the [Pins] tab and set up the pins to be used for UART communications.

Assign the pins multiplexed with the TOOLTxD or TOOLRxD pin as the UART0 (SAU00) pins for this sample application. For the RL78/G22, specify the following pin numbers.

- RxD0: 21
- TxD0: 20
- Caution: These settings differ with the target device. For the pins that can be used for UART communications, refer to the circuit diagram of the FPB you are using. For the pin numbers to be specified, refer to section 1.3, Pin Configuration, in the User's Manual Hardware for the target RL78 device.

lardware Resource	🗄 🖻 🖧 🏯	Pin Func	tion					Q	: 🖪 ès e
Type filter text		type filt	er text (* = an	y string, ? = a	ny character)			All	
🖏 Real-Time Clock	^	Enabl	Function	PIOR	Assignment	Pin Number	Direction	Remarks	Comments
 Iclock Output/Buzzer Output Controller 			🙆 RxD0	PIOR1	P11/SI00/RxD0/TOOLRxD/SDA00/TS12/TII	/ 21		Multiple pin functions on the same pin	
PCLBUZ0			SCK00	PIOR1	Not assigned	Not assigne	None		
PCLBUZ1			SCL00	PIOR1	Not assigned	Not assigne	None		
强 A/D Converter			SDA00	PIOR1	Not assigned	Not assigne	None		
🗸 ٰ Serial Array Unit			SI00	PIOR1	Not assigned	Not assigne	None		
~ o SAU0			SO00	PIOR1	Not assigned	Not assigne	None		
SAU00			🔇 TxD0	PIOR1	P12/SO00/TxD0/TOOLTxD/TS13/TI05/TO0:	/ 20		Multiple pin functions on the same pin	
SAU01		1				•			
SAU02									
SAU03									
✓ SAU1									
SAU10									
SAU11									
✓ ₩ Serial Interface IICA IICA0									
 ■ IICAU > *i Serial Interface UARTA 									
UARTAO	~	<							

Figure 7-17 Assigning Pins to the UART Channel (UART0)

UART0 pin assignment errors may occur with certain versions of the tool but the errors can be ignored for the following reason.

The following pins are used for different purposes in this sample application.

(1) TOOLRxD and TOOLTxD pins:	For using the COM port to write a program created in CS+
(2) RxD0 and TxD0 pins:	For serial communications by using the standalone version of QE

Functions (1) and (2) are multiplexed and assigned to the same pins, so the Smart Configurator generates pin assignment conflict error messages. However, functions (1) and (2) are not used at the same time, so no conflict arises in actual use by the sample application. Therefore, the above assignments can be used without problems.

Sconfiguration Problems ×	
Description	Туре
✓ Ø Pin (4 items)	
8 E04010003: Pin used by RxD0 in Config_UART0 conflicts with pin used by TOOLRxD in System, pin used by TOOLRxD in Pin Allocator.	Pin
😣 E04010003: Pin used by TxD0 in Config_UART0 conflicts with pin used by TOOLTxD in System, pin used by TOOLTxD in Pin Allocator.	Pin
8 E05000010: Pin 20 cannot be used multiple times. Pin 20 is assigned to TxD0 and TOOLTxD.	Pin
8 E05000010: Pin 21 cannot be used multiple times. Pin 21 is assigned to RxD0 and TOOLRxD.	Pin





Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7.5.4 Setting the LVD Component

Set up the user option byte for voltage detector 0 (LVD0).

Click on the "LVD0" module and specify the operating mode and voltage to be detected.

Set the reset generation level (VLVD0) to 2.62 V.

Image: Smart Configurator File Window Help Image: Smart Configurator Image: Smart Configurator Image: Smart Configurator Image: Smart Co		
Software component configuration		Generate Code Generate Report
Components 🚵 🛃 🖓 🕀 🕀		0
type filter text • Startup • Generic • Lyps • Power management and reset function • Config_UND0 • Config_LARTO • Config_LARTO • Middleware • Generic • Tctsu • Tctsu	Operation mode setting ● Reset mode When setting LVD0 to reset mode, set the detection voltage of LVD1 higher than the detection voltage of Q Interrupt mode If LVD0 is set to interrupt mode and the LVD0 detection voltage is greater than the LVD1 detection voltage LVD0 becomes undefined after the LVD1 setting following release from the reset state. LVD0 Hyperbolic Level 3 (low) Voltage detection setting Reset generation level(VLVD0) Leterupt generation level(VLVD0) 1.65	
Overview Board Clocks System Components Pins Inte	mupt	

Figure 7-19 Setting the LVD Component (LVD0)

Caution: In the RL78/G16, the selectable power-on-reset circuit is used for the voltage detection function. For this function, specify the voltage for generating a reset on the [System] tabbed page as shown in Figure 7-20.

System configuration			Generate Code Generate Report
R			
 On-chip debug setting 			
On-chip debug operation setting	O Use emulator	COM Port	
Emulator setting	E2 Lite		
Pseudo-RRM/DMM function settin	🖲 Used		
Security ID setting			
Security ID	0x000000000000000000000000000000000000		
✓ Selectable Power-on-reset Circuit			
RESET pin setting	Used		
Operation mode setting Reset generation level(VSDR)	2.84 V (V)		

Figure 7-20 Setting the Voltage for Generating a Reset (RL78/G16)



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7.5.5 Setting the PORT Component

Specify the port pins that are connected to the LEDs.

Set P62 and P63 as outputs initially at the high level in this sample application.

For details of the port pins used to control the individual LEDs, refer to the circuit diagram of the target board you are using.

1. Click on the "PORT" module and select "PORT6".

*Capacitive_Touch_Project_Example.scfg ×	
Software component configuration	Generate Code Generate Repo
Components 🚵 🖄 🖓 🖯 🏵	Configure
ा सिंह 🕼 🐮 🐩 😵	Port selection PORT6
 ✓ Startup ✓ Generic 	
	PORT2 PORT3 PORT4 PORT5
Config.PORT Config.PORT Communications	
 ar Config_UART0 ✓ I Middleware 	PORT14
✓ Seneric er r_ctsu er m_touch	Port mode setting

Figure 7-21 Setting the PORT Component

2. Set up the port pins so that LED1 and LED2 are not turned on at startup. Specifically, click on the [PORT6] tab and set "P62" and "P63" as outputs initially at the high level.

*Capacitive_Touch_Proje	ct_Example.scfg ×		
Software compone			🕲 👜 Generate Code Generate Report
Components		Configure	
type filter text	60	Port selection PORT6	
 ✓ Startup ✓ Generic ✓ r_bsp ✓ B Drivers 		Apply to all Unused In Out	Output 1
 ✓ ➢ Power manager ở Config_LVD0 ✓ ➢ I/O port 	ment and reset function	P60 Unused O In O Out	Output 1
Config_POR	ns	P61 ● Unused ○ In ○ Out	Output 1
 ♂ Config_UAR ~ ➢ Middleware ~ ➢ Generic 	10	P62 O Unused O In O Out	🗹 Output 1 🗲
 r_ctsu rm_touch 		P63 O Unused O In O Out	🗹 Output 1 🦛

Figure 7-22 Setting P62 and P63 as Outputs at the High Level



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7.5.6 Board Support Package

Select the "r_bsp" module and check that "Initialization of peripheral functions by Code Generator/Smart Configurator" is set to "Enable".



Figure 7-23 Setting r_bsp



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7.6 Setting Unused Pins

Setting the unused pins as outputs at the low level is recommended.

As an example, this section describes the procedure for setting PORT41 as an output at the low level.

- Caution: In designing circuits for a user board, appropriately handle the pins so that the electrical characteristics are satisfied.
- 1. Click on the "PORT" module and select "PORT4".

<mark>1 🗁 🔚</mark> 🖗 *Capacitive_Touch_Project_I	Example.scfg ×			
Software component	t configuration			🔋 👜 Generate Code 🛛 Generate Report
Components	è 4 ¹⁸ 2 ⊟ €	Configure		·
type filter text	65	Port selection PORT	4 PORT6	
✓ Startup ✓ Generic		PORT0	DPORT1	
r_bsp✓ ₽ Drivers		PORT2	PORT3	
 Power manageme Config_LVD0 	ent and reset function	PORT4	PORT5	
✓ ➢ I/O port		PORT6	PORT7	
Config_PORT		PORT12	PORT13	
r Config_UART0 ✓ ➢ Middleware		PORT14		
✓ Generic		Port mode setting Read Pmn regist 	er values O Read digital output level	

Figure 7-24 Setting the Port Module

2. Click on the [PORT4] tab and set "P41" as an output at the low level.

Software component			ے ان اور
Components	èn 🗹 lª₂ 🖻 €	Configure	^
5H 100	5.5	Port selection PORT4	
type filter text		kommunika V	
 ✓ ➢ Startup ✓ ➢ Generic 		Apply to all Unused In Out Pull-up TTL buffer	Output 1
💣 r_bsp 🗸 🈥 Drivers			
 ✓ Power managen ✓ Config_LVD0 ✓ 	nent and reset function	P40	Output 1
Config_PORT	5	P41 Unused In Out	Output 1
r_Config_UART	0		
r_ctsu			

Figure 7-25 Setting P41 as an Output at the Low Level



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7.7 Generating Code

Click on the 🛐 icon of the Smart Configurator to generate code.

A cautionary message will appear before code generation. Ignore the message and proceed with generation.

Code Generating		×
The generated code may have runtime problem beca configuration errors.	use of configuration conflicts or	
Proceed with ignore and generate?		
Always ignore and generate without asking?		
	Proceed Cancel	
	The generated code may have runtime problem beca configuration errors. Proceed with ignore and generate?	The generated code may have runtime problem because of configuration conflicts or configuration errors. Proceed with ignore and generate? Always ignore and generate without asking?

Figure 7-26 Cautionary Message before Code Generation

Also, if the settings for on-chip debugging or option bytes have been changed, the following message may appear. Confirm the changes and click on the [OK] button.

Confirm linker option change	2		×
Setting User option byte value Option byte values for OCD Range of debug monitor area Set debug monitor area	Old value - - FE00-FFFF No	New value EFFCE8 84 0FE00-0FFFF Yes(Specify	
	OK	Cancel	

Figure 7-27 Message for Confirming Changes in Linker Options



8. Settings of QE for Capacitive Touch

8.1 Starting QE for Capacitive Touch

Start the standalone version of QE (hereafter referred to as "the QE").

- 1. Start the QE from "QE-CapTouch (QE installation folder)/eclipse/qe-captouch.exe".
- 2. Figure 8-1 shows the window after the QE has started.

Board Monitor X 🕅 🕄 🖫 🖓 🖓 ն 🏥 ՝ 🗆	Main Status Chart St	atus chart		1 ° °	Parameters × 🔤 🕼 🎲 🖓 🗖 🗖
Touch I/F:	1. Preparation Prepare a project that uses the touch interfaces.	2. Configuration Prepare a touch interface configuration.	3. Tuning QE will automatically perform tuning processing for each touch sensor.	4. Coding and Monitoring Implement a program using the touch. Then, confirm a behavior of touch interfaces and make time adjustments.	Touch I/F:
Board monitor	Development Target Show the development target. MCU 	Output mes required	A Creacute Program Bard schauging the target project for the schauging the schauging the schauging the schauging the schauging the schauging the schauging the schauging the schauging the schauging the schaugin	Implement Program Implement a program using the bouch interfaces and build the project. Implement a program using the bouch interfaces and build the project. Implement a program using the bouch interfaces. Implement a program using the bouch interface. Implement a program using the bouch interface. <	Item Value Parameters
< > > > > > > > > > > > > > > > > > > >	Tuning Touch I/F Configuration: <not selected=""></not>		interface guration	~	^
32766 Multi-status chart	Console	Co	nsole	~	

Figure 8-1 QE Window after Startup

If the layout is distorted in a full-screen display, change the "Scale and layout" for "Display" in the Windows settings to "100%".



Г

Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

8.2 Preparing a Project

Prepare a project that will use touch interfaces.

Set up the items under "Preparation" in the workflow diagram displayed across the middle of the QE window after startup.

1. Preparation	2. Configuration	3. Tuning	4. Coding and Monitoring
Prepare a project that uses the touch interfaces.	Prepare a touch interface configuration.	QE will automatically perform tuning processing for each touch sensor.	Implement a program using the touch. Then, confirm a behavior of touch interfaces and make fine adjustments.
Development Target Show the development target. MCU Create a target project using Smart Configurator. Also, Set touch sensors and add the CTSU driver with SC tool. To Select a Project Folder Select a folder that contains the target project. 	Image: Description of the section o	Conservet Program That debugging the target project in DE tool and execute the program. The tuning program written on the target baad will run. DE TOP Start Project Bollow instructions in the dialog. Start Tuning Advanced mode Dotput Parameter Files Output Parameter Files Use an external trigger Use diagnostic code	To Implement Program Implement a program using the touch interfaces and build the project. Daunch debugging for your target project and execute the program. Daunch debugging for your target project and execute the program. Daunch debugging for your target project and execute the program. Dable a monitoring function via serial communication. Baud rate 115200 Port Auto Connect Descent Connect Show monitoring views and enable a monitoring function. Implement 75 [ms]

Figure 8-2 Workflow Diagram (Preparation)

- 1. Click on [...] under "To Select a Project Folder" and select the project folder that was created by CS+.
- 2. Click on [...] under "To Select an MCU Name" and select the target MCU to be used.

L. Preparation	2. Configuration	3. Tuning	4. Coding and Monitoring
Prepare a project that uses the touch nterfaces.	Prepare a touch interface configuration.	QE will automatically perform tuning processing for each touch sensor.	Implement a program using the touch. Then, confirm a behavior of touch interfaces and make fine adjustments.
Development Target Show the development target. MCU To Create Project Create a target project using Smart Configurator. Also, Set touch sensors and add the CTSU driver with SC tool. To Select a Project Folder Select a folder that contains the target project. C:¥CS+_Workspace¥Capat To Select a MCU Name Select a name of the target MCU. R7F102GGE 	Image: Definition of the state of the s	De Execute Program Start debugging the target project in DE tool and execute the program. The tuning program written on the target basic will run. De Tool and execute the program . The tuning program written on the target basic will run. De Tool and execute the program . The tuning program written on the target basic will run. Start Tuning Advanced mode Dutput Parameter Files Output Parameter Files Use an external trigger Use diagnostic code	To Implement Program Implement a program using the touch interfaces and build the project. Description To Launch Debug Launch debugging for your target project and execute the program. Poconnect UART Table a monitoring function via serial communication. Baud rate 115200 Port Auto Connect Description Connect Description Show monitoring views and enable a monitoring function. Interval 75 [ms]

Figure 8-3 Preparing a Project



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

Product Name of Target MCL	J	×
Family Name	RL78	~
Group Name	RL78/G22	~
Pin Number	48pin:G	~
ROM Size	64KB:E	~
Product Name	R7F102GGE	
	OK Cancel	

Figure 8-4 Selecting the Target MCU

If attempting to select an MCU causes the following error, it may indicate a problem with the location or pathname of the folder where the QE has been installed. Terminate the QE, move the installation folder to another location such as under "C:\Renesas", and then restart the QE.

Internal Error	– – ×
Internal Error Reason: java.lang.NullPointerException	
	OK Details >>

Figure 8-5 Error on Attempting to Select an MCU



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

8.3 Configuring the Touch Interface

Set up the items under "Configuration" in the workflow diagram.



Figure 8-6 Workflow Diagram (Configuration)

1. Click on 🗸 under "To Prepare a Configuration" and select "Create a new configuration".

1. Preparation	2. Configuration	3. Tuning	4. Coding and Monitoring
Prepare a project that uses the touch nterfaces.	Prepare a touch interface configuration.	QE will automatically perform tuning processing for each touch sensor.	Implement a program using the touch. Then, confirm a behavior of touch interfaces and make fine adjustments.
Development Target Show the development target. MCU To Create Project Create a target project using Smart Configurator. Also, Set touch sensors and add the CTSU driver with SC tool. To Select a Project Folder Select a folder that contains the target project. C:¥CS+_Workspace¥Capac To Select a MCU Name Select a name of the target MCU. R7F102GGE 	E Prepare a Configuration Gelect or create a touch interface configuration. Image: Create a new configuration Create a new configuration Create a new configuration Output Files for Tuning Dutput files required for a tuning process. Output Files Output Files Dutput Files	 Definition Definition<	To Implement Program Implement a program using the touch Interfaces and build the project. Daunch debugging for your target project and execute the program. To Connect UARI Enable a monitoring function via serial communication. Baud rate 115200 Port Auto Connect To Enable Monitoring Show monitoring views and enable a monitoring function.

Figure 8-7 Creating a New Touch Interface Configuration



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

2. The [Create Configuration of Touch Interfaces] window will open and display the area for placing the elements of the touch interface.

Click on [Button] in the [Touch I/F] panel on the right to change the cursor to the one for use in button placement. A button can then be placed by clicking on the area where it is to be placed.

Place two buttons (Button00 and Button01) as shown below and press the [Esc] key to finish the button placement.

In the same way, click on [Slider (horizontal)] in the [Touch I/F] panel and place a slider (Slider00). Figure 8-8 shows the window after placement of the elements of the touch interface.

File Name of Touch I/F:	Capacitive_Touch_Project_Example Setup Configuration	Import / Re-edit
Description:		
Slider00		Touch I/F *
		Capacitance Type
<<<		Self-Capacitance method v
Button00 Button01	T.	Button
		Slider (horizontal)
		Slider (vertical)
		Wheel
		Key pad
		3D Gesture (AI)
		Touch pad
		Shield Pin
		TC Pin
		Capacitance Sensor
		Current Sensor
		Diagnosis Pin
Setting		Remove Touch I/F
setung	Setup Resistance Value Clear Assigned TSx	Configurations (Methods) *

Figure 8-8 Placing Buttons and a Slider

3. Assign names and touch sensors to the buttons.

Double-click on [Button00] placed in the previous step and the [Setup Touch Interface] dialog box will open. Make the following settings in the dialog box.

— Touch Sensor: TS24

- Resistance [ohm]: 560

For the resistance, refer to the user's manual or circuit diagram of the target board.

Caution: If TS pin numbers do not appear correctly in the dialog box for assigning a touch sensor to a button, the Windows settings require changing. Set the "Scale and layout" for "Display" to "100%" in the Windows system settings and then restart the QE for Capacitive Touch.

Setup Touch In	Touch Interface	×
Button(self	tton(self)	
Name	me Button00	
Touch Sens	buch Sensor Resistance[ohm] S24 V 560	~
ОК	OK Cancel Help	

Figure 8-9 Setting a Touch-Interface Element (Button)



- 4. In the same way, make the following settings for [Button01].
 - Touch Sensor: TS23
 - Resistance [ohm]: 560
- 5. Make the following settings for [Slider00].
 - Touch Sensor: TS20
 - TS21 TS22
 - Resistance [ohm]: 560

	Setup Touch Interface	×	
TS20 TS22	Number of Touch Sensor 3	0.4.11.4.	
	Reverse OK	ncel Help	

Figure 8-10 Setting a Touch-Interface Element (Slider)

6. After the settings for the elements of the touch interface are complete, the color of the electrodes changes to green as shown in Figure 8-11. Click on [Create] at this point.

File Name of Touch I/F:	Capacitive_Touch_Project_Example	Setup Configuration	Import / Re	-edit
Description:				
Slider00			Touch I/F	\$
			Capacitance Type	
T520 T521 T522			Self-Capacitance method	~
Button00 Button0	1		Button	
TS24 TS23			Slider (horizontal)	
1021			Slider (vertical)	
			Wheel	
			Key pad	
			3D Gesture (AI)	
			Touch pad	
			Shield Pin	
			TC Pin	
			Capacitance Sensor	
			Current Sensor	
			Diagnosis Pin	
Setting			Remove Touch I/F	
	Setup Resistance Value Clear Assigned TSx		Configurations (Methods)	*

Figure 8-11 Touch Interface Configuration after the Settings are Complete



RL78 Family Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

7. The touch interface configuration will be displayed in the [Tuning] panel.

iouci i i i i	configuration	Capacitive_	_Touch_Project_Exar	nple				
Method	Kind	Name	Touch Sensor	Parasitic Capacitance[pF]	Sensor Drive Pulse Frequency[MHz]	Threshold	Scan Time[ms]	Overflow
config01	Button(self)	Button00	TS24	-	-	-	-	None
config01	Button(self)	Button01	TS23	-	-	-	-	None
config01	Slider	Slider00	TS20, TS21, TS22	-	-	-	-	None
config01	Slider TS	(Slider00)	TS20	-	-	-	-	-
config01	Slider TS	(Slider00)	TS21	-	-	-	-	- · · · · · · · · · · · · · · · · · · ·
config01	Slider TS	(Slider00)	TS22	-	-	-	-	-

Figure 8-12 Touch Interface Configuration Displayed in the [Tuning] Panel

8. Create a folder for storing the output files required for tuning. For this sample application, create a new folder "qe_gen" under "Capacitive_Touch_Project_Example/src". Click on [Output Files] in the QE workflow diagram and select the folder for storing the output files.



Figure 8-13 Creating a New Folder "qe_gen"

The following shows the place of the created folder in the directory structure and the files to be output.

 Capacitive_Touch_Project_Example

 ← CS+ project folder (Project name specified in chapter 6, Creating a New Project)

 |- src_gen

 ← CS+ project folder (Project name specified in chapter 6, Creating a New Project)

 |- qe_gen

 ← New folder created

 |- qe_touch_config.c

 ← Output file

 |- qe_touch_define.h

 ← Output file

 |- qe_touch_sample.c

 ← Output file



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

 Specify the frequency of the clock to be supplied to the CPU and peripheral hardware. After the folder for storing the output files has been selected, the following dialog box will open. Set the frequency (fCLK) of the clock for the CPU and peripheral hardware and click on [OK].

Peripheral module clock frequency (PCLKB or PCLKL)[MHz] 32 OK Cancel	Frequency of Peripheral Module Clock (PCLKB or PCLKL) $ imes$
OK Cancel Help	Peripheral module clock frequency (PCLKB or PCLKL)[MHz] 32
	OK Cancel Help

Figure 8-14 Setting the Frequency of the Peripheral Module Clock

 Specify the voltage of the power supply for the MCU. After the [Power Supply Voltage of MCU] dialog box appears, specify the voltage and click on [OK].
 For the voltage to be specified, refer to the electrical characteristics of the target MCU.
 If the MCU uses EVDD, enter the EVDD value in the VDD field.

Power Supply Voltage of MCU (VDD)	×
MCU power supply voltage (VDD) [V] 5.0 • • • • • • Set the voltage within the operating range of the device. • When you use an MCU that has EVDD, replace VDD with E	VDD.
Measurement voltage setting Normal voltage v	
OK Cancel Help	

Figure 8-15 Setting the Power-Supply Voltage for the MCU

11. The [QE for Capacitive Touch] dialog box will open. The contents of this box are also displayed in the [Console] panel at the bottom of the QE window.



Figure 8-16 [QE for Capacitive Touch] Dialog Box


RL78 Family Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

Console	
2025/03/12 10:37:55	^
Succeeded to output the parameter files.	
C:¥CS+202501_Workspace¥Capacitive_Touch_Project_Example¥src¥qe_gen¥qe_touch_define.h	
C:¥CS+202501_Workspace¥Capacitive_Touch_Project_Example¥src¥qe_gen¥qe_touch_config.h	
C:¥CS+202501_Workspace¥Capacitive_Touch_Project_Example¥src¥qe_gen¥qe_touch_config.c	
C:¥CS+202501_Workspace¥Capacitive_Touch_Project_Example¥src¥qe_gen¥qe_touch_sample.c	
Launch IDE tool and open the target project. Then, perform the following operations.	
* Configure the Compiler Settings below.	
- Add "QE_TOUCH_CONFIGURATION" to "Define Preprocessor Symbol (-D)" option or similar one.	
- Add a path of the folder where you output the files to "Include Directories (-I)" option or similar one.	
* Implement a process to call the main loop for touch in the main() function.	~
	>

Figure 8-17 [Console] Panel

12. Set the compiler options. Open the CS+ window and select "CC-RL (Build Tool)" in [Project Tree]. In the [Common Options] tabbed page of [Property], select "Macro definition" under "Frequently Used Options (for Compile)" and click on [...] on the right side of the page.

0 🙎 🔳	CC-RL Property		- Q 🖬
Capacitive Touch Project Example (Project			And
Capacitive Touch Project Example (Project R7F 102GGE47B (Microcontroller) Smart Configurator (Design Tool) CCR. (Build Tool) Program Analyzer (Analyze Tool) R178 Simulator (Debug Tool) Frite Tool Build tool generated files 		DefaultBuid No RL/76-S3 core(cpu=S3) No Execute Module(Load Module File) No %BuildModeName% Perform the default optimization/No option specified) Additional include paths[13] System include paths[13] Macro definition(0) Macro definition(0) Using Iloranes(0) %BuildModeName%	
	Macro definition Specifies the macro name to be defined in the format of "(macro name)=(defined value)", with This option corresponds to the -D option of the ccrl command.	h one macro name per line. The "=(defined value)" part can be omitted, and in this case, "1" is used as the defi	ined value.

Figure 8-18 Selecting Macro Definition

13. The [Text Edit] dialog box will open. Enter "QE_TOUCH_CONFIGURATION" in the [Text] field in the dialog box and click on [OK].

Text Edit	×	
Text:		
QE_TOUCH_CONFIGURATIONS	<u>^</u>	
< 3		
OK Cancel Help		





14. Add the "qe_gen" folder to the sample project. Drag and drop the "qe_gen" folder from Explorer to the [Project Tree] of CS+.



Figure 8-20 Adding the qe_gen Folder

15. Add the path to the "qe_gen" folder as an include path. On the [Common Options] tabbed page of [Property], select "Additional include paths" under "Frequently Used Options (for Compile)" and click on [...] on the right side of the page.

The [Path Edit] dialog box will open. Check that "src\qe_gen" has been added to the [Path] field in the dialog box and click on [OK].

 Frequently Used Options(for Compile) 		
Level of optimization	Perform the default optimization(No option specified)	
> Additional include paths	Additional include paths[14]	
> System include paths	System include paths[0]	
> Macro definition	Macro definition[1]	

Figure 8-21 Adding an Include Path for the Compiler

Path Edit		×
Path(One path per o	one line): 😱	
src¥smc_gen¥r_bsp src¥smc_gen¥r_bsp src¥smc_gen¥r_bsp < Browse Permit non-exist	u nfig y¥mcu¥rl78_g22¥register_access¥ccrl y¥mcu¥rl78_g22 y¥mcu¥all y¥board¥generic_rl78_g22	~
Placeholder	Value	^
ActiveProjectDir ActiveProjectMicom ActiveProjectName BuildModeName MainProjectDir	C:¥CS+_Workspace¥Capacitive_Touch_ mName R7F102GGExFB e Capacitive_Touch_Project_Example DefaultBuild C:¥CS+_Workspace¥Capacitive_Touch_	
<		>
	OK Cancel I	Help

Figure 8-22 Checking Addition of the Include Path



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

16. Specify the language variant of the C source files.

Click on "Source" on the [Compile Options] tabbed page and click on "Language of the C source file". Click on v on the right side of the page and select "C99 (-lang=c99)".

	Source						
	anguage of the C source file			C99(-lang=c99)			
	anguage of the C++ source file		(C++14(-lang=cpp14)			
~ (Quality Improvement						
C	Detect stack smashing		1	No(No option specified)			
0	Detect illegal indirect function call		1	No			
> 1	Memory Model						
> (CLanguage						
> (Character Encoding						
> (Dutput Code						
> (Dutput File						
> 1	Assemble List						
> 1	MISRA-C Rule Check						
> 1	Message						
> (Others						~
Sele	uage of the C source file cts the language of the C source file. option corresponds to the -lang option of the ccrl command.						
Co	mmon Options A Compile Options	SMS Assemble Options / Lir	nk Options	Hex Output Options	Standard Library Generate Options	I/O Header File Generation Options	-

Figure 8-23 Selecting the C-Language Standard

- 17. Set the on-chip debugging (OCD) option byte and user option bytes. Click on "Device" on the [Link Options] tabbed page and make the following specifications. For meanings of the values of the option bytes, refer to the user's manual of the target MCU.
 - Option byte value for OCD: 84
 Set debug monitor area: Yes (Specify address range) (-DEBUG_MONITOR=<Address range>)
 User option byte value: EFFCE8

Property Main.c Mg qe_touch_sa	ple.c		* X
CC-RL Property			≥ <i>P</i> - +
> Library > Device			^
Set enable/disable on-chip debug by lin Option byte values for OCD	option	Yes(-OCDBG)	
Set debug monitor area Range of debug monitor area		Yes(Specify address range)(-DEBUG_MONITOR= <address range="">)</address>	
Set user option byte User option byte value		Yes(-USER_OPT_BYTE)	
> Output Code			~
Device			
Common Options / Compile Options	AssembleOptions / SMS Assemble Options / Link Opt	tions Dutput Options 🙏 Standard Library Generate Options 🧎 I/O Header File Generation O	Iptions -

Figure 8-24 Setting the Option Bytes



RL78 Family Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

- 18. If the free evaluation edition of CC-RL V1.12.00 or a later version is to be used for compilation, "Debug precedence (-Onothing)" should be selected as the level of optimization by the compiler before building. Double-click on "Optimization" on the [Compile Options] tabbed page and select "Debug precedence (-Onothing)" for "Level of optimization".
- Remark: This setting is only necessary for tuning. After tuning is complete, any optimization level can be specified.

V Optimization Level of optimization) Optimization(Details)) Preprocess) Source	Debug precedence(-Onothing)	× ×
Level of optimization Selects the level of the optimization for compiling. This option corresponds to the -O option of the corl command. Common Options Compile Options Common Options Compile Options	tions $ floor$ Hex Output Options $ floor$ Standard Library Generate Options $ floor$ 1/O Header File Generation Options $ floor$	-

Figure 8-25 Setting the Level of Optimization by the Compiler

19. Implement the processing for calling the main function of the touch measurement processing in the main() function.

This requires a call of the qe_touch_main() function from the main() function.

Add the following statements to the "main.c" file at the points indicated in the listing below.

- extern void qe_touch_main(void);
- qe_touch_main();



Figure 8-26 main.c



RL78 Family Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

- 20. Add the function for serial communications to the "Config_UART0_user.c" file.
 - Add the following statements to the "main.c" file at the points indicated in the listing below.
 - extern void touch_uart_callback(uint16_t event);
 - touch_uart_callback(0);
 - touch_uart_callback(1);

52 53 54	/* Start user code for global. Do not edit comment generated here */ <u>extern void touch_uart_callback(uint16_t event);</u> /* End user code. Do not edit comment generated here */
74 75 76 77 78 79	<pre>static void r_Config_UARTO_callback_sendend(void) </pre>
87 88 99 91 92	<pre>static void r_Config_UARTO_callback_receiveend(void) { /* Start user code for r_Config_UARTO_callback_receiveend. Do not edit comment generated here */ touch_uart_callback(1); /* End user code. Do not edit comment generated here */ }</pre>



21. Build the project in CS+. Click on the icon under the menu bar of CS+ to start the process of building. Check that the build process has been completed without any errors or warnings.

If the following warning (W0511187) is generated during building, change the level of optimization by the compiler to "Debug precedence (-Onothing)" as shown in Figure 8-25 and rebuild the project.



Figure 8-28 Warning during Building (W0511187)



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

8.4 Tuning

Set up the items under "Tuning" in the workflow diagram.



Figure 8-29 Workflow Diagram (Tuning)

1. Select the debugging tool to be used. Right-click on "Debug Tool" in [Project Tree] of CS+ and select "RL78 COM Port" from [Using Debug Tool].



Figure 8-30 Selecting the Debugging Tool



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

2. Set the "Communication port" in the [Property] of the debugging tool. This sample application uses COM24.

<u>لية</u>	RL78 COM Port Property		_ م
~	Internal ROM/RAM		
	Size of internal ROM[KBytes]	64	
	Size of internal RAM[Bytes]	4096	
	Size of DataFlash memory[KBytes]	2	
~	Clock		
	Main clock frequency [MHz]	Using internal clock	
	Sub clock frequency[kHz]	Using internal clock	
	Monitor clock	System	
~	Connection with Target Board		
	Communication port	COM24 COM24	
	Reset control pin	DTR	
	Baud rate	Auto	

Figure 8-31 Property of the Debugging Tool

The COM number of the port used for communications can be confirmed in the [Device Manager] window.

🛃 Device Manager	-	\times
File Action View Help		
← → 📧 📓 🖬		
 WAN Miniport (SSTP) Ports (COM & LPT) Intel(R) Active Management Technology - SOL (COM6) USB Serial Port (COM24) Print queues 		^
Processors Processors Processors Processors Sensors		v

Figure 8-32 Device Manager

- 3. Enable the COM Port debugging circuit. Check that the J16 jumper for switching the QE serial connection on the target board is closed.
- 4. Build the project and write the program. Check that the target board is connected to the PC with a USB

cable and click on the 🚳 CS+ icon. After downloading for writing the program is complete, click on the

🦲 icon to stop the program and then click on the 🔬 icon for disconnection from the debugging tool.

5. Execute the serial connection function of the QE. After disconnection from the debugging tool, remove the USB cable connecting the PC and target board and open the J16 jumper for switching the QE serial connection.

After that, reconnect the target board to the PC with the USB cable for later connection with the QE. At this time, the written program will run on the target board which is in the state of standing by for connection with the QE.

For the J16 jumper for switching the QE serial connection, refer to the user's manual of the target board. Be sure to use a USB cable that supports data transfer.



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

6. Set "Baud rate" under "To connect UART" on the QE to the value specified in section 7.5.3.



Figure 8-33 Setting "Baud rate"

7. Click on [Start Tuning] to start automatic tuning.



Figure 8-34 Automatic Tuning



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

8. A dialog box will open. Specify "Baud rate" in the dialog box and click on [Connect].

Paud rate 152600		
Baud rate 153600		
COM port Auto	~	
Connect Cancel	Help	

Figure 8-35 Setting "Baud rate"

9. In the dialog box opened in response to the previous step, specify the frequency of the clock for the CPU and peripheral hardware and click on [OK].



Figure 8-36 Setting the Frequency of the Peripheral Module Clock

10. Automatic tuning will begin. Check the messages in the [Automatic Tuning Processing] dialog box, which guides the user through the steps of tuning, and follow the instructions for proceeding through the steps.



Figure 8-37 [Automatic Tuning Processing] Dialog Box



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

After several steps of the tuning process are complete, the dialog box shown in the figure below will open. Measure the touch sensitivity at this point. Use normal pressure in touching the Button01 touch sensor as instructed by the dialog box. While the touch sensor is being touched, the bar in the dialog box will be extended to the right and the numerical value of touch counting will increase. While continuing to touch the sensor with your finger, press any key on the keyboard of the PC to confirm the measured value.

	<image/>	
I Automatic	Tuning Processing X	
In this step	l now measure touch sensitivity for (Button01, TS23 @ config01). please use normal touch pressure on the sensor for once. Press any key on the PC keyboard to accept vity measurement.	
Button01,	TS23 @ config01: 11756	
	Cancel	

Figure 8-38 Measuring the Touch Sensitivity (Button)

- 11. Proceed with measurement for the other button touch sensor in the same way.
- 12. Measure the touch sensitivity of the slider touch sensor. Use normal pressure in moving a finger up and down or left and right across the slider on the target board three or four times and then press any key on the keyboard of the PC to confirm the measured value.

Automatic Tuning Processing	×
7/8: QE will now measure touch sensitivity for all sliders in this configuration (config01) on the targe Please move your finger across all of the slider sensors 3-4 times using normal touch pressure. Whe press any key on the PC keyboard to complete the sensitivity measurement.	
	Cancel

Figure 8-39 Measuring the Touch Sensitivity (Slider)



RL78 Family Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

13. The following dialog box will open after tuning is complete and you can check the threshold values. These threshold values are used by the middleware to determine whether touch events have occurred. Click on [Continue the Tuning Process]. This is the end of the steps for automatic tuning.

			nce.						
lect the target Me	Aethod H	Kind	Name	Touch Sensor	Threshold	Overflow	Warning / Error	 	
со	onfig01	Button	Button00	TS24	2426				
со	onfig01	Button	Button01	TS23	2569				
co	onfig01 S	Slider	Slider00	TS20, TS21, TS22	2178				

Figure 8-40 Threshold Values for the Touch Sensors

14. Click on [Output Parameter Files] to output parameter files containing the results of tuning. Select the "qe_gen" folder that was newly created in section 8.3 as the folder for the output of files and the files in the folder will be overwritten.

The names of the output files are the same as those listed below that were output by [Output files] in section 8.3.

 |- qe_touch_config.c

 ← Output file

 |- qe_touch_config.h

 ← Output file

 |- qe_touch_define.h

 ← Output file

 |- qe_touch_sample.c

 ← Output file

1. Preparation	2. Configuration	3. Tuning	4. Coding and Monitoring			
Prepare a project that uses the touch nterfaces.	Prepare a touch interface configuration.	QE will automatically perform tuning processing for each touch sensor.	Implement a program using the touch. Then, confirm a behavior of touch interfaces and make fine adjustments.			
Development Target Show the development target. MCU To Create Project Create a target project using Smart Configurator. Also, Set touch sensors and add the CTSU driver with SC tool. To Select a Project Folder Select a folder that contains the target project. C:¥CS+202501_Workspace	Deference a Configuration. Select or create a touch interface configuration. Capacitive_Touch_Project_Exar Modify Configuration Modify Configuration Doubter Files for Tuning Output files required for a tuning process. Output Files Dutput Files Doubter Files Dutput Files	Concerning the target project in Dib tool and execute the program. The tuning program written on the target based will run. Destant Juning Destant Juning Destant Tuning Advanced mode Datput parameter files from a tuning Output Parameter Files Use an external trigger Use diagnostic code	To Implement Program Implement a program using the touch interfaces and build the project. To Launch Debug Launch debugging for your target project and execute the program. Defaurch Debug Launch debugging for your target project and execute the program. Defaurch Debug Launch debugging for your target project and execute the program. Defaurch Debug Inable a monitoring function via serial communication. Baud rate 153600 Port Auto Port Connect Demonitoring views and enable a monitoring function. Interval 75 [ms]			

Figure 8-41 Output of the Parameter Files



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

8.5 Coding and Monitoring

8.5.1 Monitoring

Monitor the touch interface operation by proceeding through the steps under "Coding and Monitoring" in the workflow diagram.

. Preparation	2. Configuration	3. Tuning	4. Coding a	nd Monitoria	ng
repare a project that uses the touch terfaces.	Prepare a touch interface configuration.	QE will automatically perform tuning processing for each touch sensor.	Then, confirm a	ogram using the t behavior of touc nake fine adjustr	h
Development Target Show the development target. MCU V To Create Project Create a target project using Smart Configurator. Also, Set touch sensors and add the CTSU driver with SC tool. To Select a Project Folder Select a folder that contains the target project.	To Prepare a Configuration Select or create a touch interface configuration. Capacitive_Touch_Project_Exar Modify Configuration Modify Configuration To Output Files for Tuning process. Output Files	To Execute Program Start debugging the target project in IDE tool and execute the program. The tuning program written on the target board will run. D Start Tuning Follow instructions in the dialog. Start Tuning Advanced mode D Output Parameter Files From a tuning result.	Interfaces and To Launch I Launch debug project and ex To Connect	rogram using the I build the project Debug ging for your targ ecute the progra UART toring function vi	t. get m.
C:¥CS+_Workspace¥Capac	To Implement Program Implement a process to call the main loop for touch in the main() function. To Build Project Build the target project using IDE tool.	Output Parameter Files Use an external trigger Use diagnostic code	Connect To Enable Monitoring Show monitoring views and enable a monitoring function. Interval 75 [ms]		

Figure 8-42 Workflow Diagram (Coding and Monitoring)

- 1. Enable the COM Port debugging circuit. Remove the USB cable connecting the PC and target board and close the J16 jumper for switching the QE serial connection. After that, reconnect the target board to the PC with the USB cable for connection with CS+.
- 2. Build the project and write the program by clicking on the 3 CS+ icon. After downloading for writing the program is complete, click on the \bigcirc icon to stop the program and then click on the 3 icon for disconnection from the debugging tool.
- 3. Execute the serial connection function of the QE. After disconnection from the debugging tool, remove the USB cable connecting the PC and target board and open the J16 jumper for switching the QE serial connection.

After that, reconnect the target board to the PC with the USB cable for later connection with the QE. At this time, the written program will run on the target board which is in the state of standing by for connection with the QE.



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

4. Click on [Connect] for serial connection with the target board. [Connect] shown in the red frame in Figure 8-43 will change to [Disconnect].



Figure 8-43 Serial Connection



RL78 Family Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

5. Click on [Enable Monitoring] in the [Board Monitor] panel in the top left part of the QE window. The indication "Monitoring: Disabled" will change to "Monitoring: Enabled".



Figure 8-44 Enabling Monitoring

When touching of a touch sensor is detected, the state of touching is indicated by a finger icon.



Figure 8-45 Display of the State of a Touch Sensor being Touched



- 6. Display a graph of the touch counting value in the status chart.A. Click on the [Status Chart] tab.
 - - B. Click on 💟 for "Touch I/F" on the opened [Status Chart] page and select a touch sensor of the touch interface configuration.

The chart shows the real-time value of touch counting as it is being measured. The change in the touch counting value while the touch sensor is being touched can be confirmed on the chart.

The green line shows the threshold value, which is used by the "rm_touch" middleware to judge whether operation of the touch sensor is in progress; that is, it is being touched.

The red strip at the bottom of the chart shows the duration over which the touch counting value exceeds the threshold value; that is, touching is being detected.



Figure 8-46 Graph of the Touch Counting Value (Button)



Figure 8-47 Graph of the Touch Counting Value (Slider)



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

- 7. Measure the signal-to-noise ratio (SNR) values as required.
 - A. Click on [Start Data Collection] on the [Status Chart] tabbed page.

Touch I/F: Button00 @ config01 v	Sync a selection	1
I/F Type: Button(self), TS Pin(s): TS24		
Measurement Value: 11968 Baseline: 11979 Thresh	hold: 2217 Touch ON/OFF difference: -11	
Start Data Collection		
		~
14197		
12622		
13632		
13070		
12508		
11946		

Figure 8-48 Collecting Data in the Touch-off State



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

B. Make settings for data collection as shown in the figure and click on [Start Data Collection]. Do not touch the sensor while collection of data in the touch-off state is in progress. The green bar indicates progress in data collection. When the green bar reaches the right end, the ratio of data collection is 100% so data collection in the touch-off state is completed.

	Data Collection Settings Number of data collected: Data collection target: O Touch C Collection data: O Measurement V		
	Probability:	0.1% (3.09) ~	
Touch I/F: Button00 @	S Pin(s): TS24	Sync a selection	
Measurement Value: Stop Data Collection		reshold: 2217 Touch ON/OFF difference:	4

Figure 8-49 Starting Data Collection

C. Collect data in the touch-on state in the same way. Make sure that one of your fingers is touching the sensor then click on [Start Data Collection]. When the green bar reaches the right end, data collection in the touch-on state is completed.

QE for	Capacitive Touch		×
1	Next, collect data during Touch ON. the Start data Collection button.	When you are ready to take measurements, clic	ck
		ОК	
		↓ −	
Touch I/I	F: Button00 @ config01 V	Sync a selection	
	F: Button00 @ config01 v Button(self), TS Pin(s): TS24	Sync a selection	
I/F Type:	Button(self), TS Pin(s): TS24	Sync a selection Threshold: 2217 Touch ON/OFF difference:	19

Figure 8-50 Starting Data Collection in the Touch-on State



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

D. After data collection is completed, the SNR value will be displayed.

Standard Deviation Measure	ment Result					×	
Noise Standard Deviation [NT]:	17.5 Average [NT]:	-1	Minimum:	-63	Maximum:	45	
Noise Standard Deviation [T]:	70.5 Average [T]:	4843	Signal:	4844	SNR:	17.80	
SNR is calculated with a probab	ility of 3.09.						
Measurement results are saved i	n the QE-Touch folder.						
						OK	

Figure 8-51 SNR Value



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

8. Display graphs of touch counting values for multiple touch sensors in the multi-status chart. Select the touch sensors for which values are to be displayed on the [Multi Status Chart] tabbed page in the lower left part of the QE window.



Figure 8-52 Multi-Status Chart

Manually adjust parameters as required.
 Use the [Parameters] panel in the right part of the QE window.

-
Touch parameters
ng environment.
fied in the

Figure 8-53 Adjusting Parameters



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

10. While "Monitoring: Enabled" is being displayed, click on [Enable Monitoring] to stop monitoring.

Board Monitor \times	ji (7. 17. 17. 17. 17. 17. 17. 17. 17. 17. 1	
Enable Monitoring	Monitoring: Enabled, Communication Status: Connecting via serial communication (UART / USB)	^
Touch I/F:	~	
		~
Sli	der00	^
$\langle \rangle$	\geq \geq	
Button00	Button01	
		~

Figure 8-54 Termination of Monitoring

11. Click on [Disconnect] to disconnect the serial connection.

. Preparation	2. Configuration	3. Tuning	4. Coding and Monitoring
repare a project that uses the touch terfaces.	Prepare a touch interface configuration.	QE will automatically perform tuning processing for each touch sensor.	Implement a program using the touch. Then, confirm a behavior of touch interfaces and make fine adjustments.
Development Target Show the development target. MCU To Create Project Create a target project using Smart Configurator. Also, Set touch sensors and add the CTSU driver with SC tool. To Select a Project Folder Select a folder that contains the target project.	To Prepare a Configuration Select or create a touch interface configuration. Capacitive_Touch_Project_Exar Modify Configuration To Output Files for Tuning process. Output Files Output Files To Implement Program	To Execute Program. Start debugging the target project in IDE tool and execute the program. The board will run. Data Start	To Implement Program Implement a program using the touch interfaces and build the project. Daunch Debug Launch debugging for your target project and execute the program. Daolen at Connect UART Enable a monitoring function via serial communication. Baud rate 153600 Port Auto Disconnect Image: Connect Conne
To Select a MCU Name Select a name of the target MCU. R7F102GGE	Implement a process to call the main loop for touch in the main() function. To Build Project Build the target project using IDE tool.	Use diagnostic code	To Enable Monitoring Show monitoring views and enable a monitoring function.

Figure 8-55 Disconnecting the Serial Connection



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

8.6 Flowchart (Software Timer)

Figure 8-56 is a flowchart of the touch measurement control processing with the use of a software timer.



Figure 8-56 Touch Measurement Control Processing with the Use of a Software Timer



9. Another Implementation of the Sample Application

9.1 Touch Measurement with the Use of a Hardware Timer

This section describes an example of an implementation with the use of a hardware timer to generate the cycles of touch measurement. This example uses the interval timer function of the 32-bit interval timer in 8-bit counter mode. This example also provides a function for checking the touch sensor operation by turning an LED on the target board on or off according to the results of judging the state of touching of a sensor (a button). Specifically, LED1 is turned on when a finger touches touch sensor 1 (TS_B1) and the result of judgment becomes detection of the touch-on state.

Make the settings described in the following section in addition to the settings described in chapter 7, Settings of the Smart Configurator.

Remark: The timer array unit or 12-bit interval timer can also be used instead of the 32-bit interval timer.

9.1.1 Using the Smart Configurator to Make Settings (Hardware Timer)

 Select the [Clocks] tab in the Smart Configurator view and set up the clock to be used for the interval timer. The low-speed peripheral clock (fSXP) is used in this example. In addition, deselect the XT1 oscillator.

Clocks configu	ration					Generate Code	e Generate Repo
880							
Operation mode:	High-speed main m	ode 2.7(V)~5.5(V)	•				
High-speed on-ch		* (MHz)	1				
fHOCO start setting	32 Normal	* (MP2)				RHP	
	C. O. C.	eed on-chip oscillator at	Î	N		32	(MHz)
		e and of transitions to			·	IMAIN 32	(MHz)
SNUULE model						KUK	
						32000	(kHz)
Middle-speed on-	chip oscillator		٦				(MHz)
Frequency:		= (0.01(z)					
			Divider				
X1 oscillator			- <u></u> -			MOP 🔒	
Operation mode:		*					(MHz)
Frequency:	5	(MHz)					
Stable time:	218.5	 S2018pt) 				120	
Low-speed on-chip or	cillator		7	•		flL 32.768	(kHz)
Frequency:	32.768	(KHz)					
The fiL runs while WD on-chip oscillator	T is operating or f5)	@ select Low-speed				15XP 32.768	(kHz)
XT1 oscillator			7			esa 🕢	
Operation mode:				•			(kHz)
Frequency:		(1412)					
XT1 oscillation mode:		imption 1. •					
Supply mode:		a 370P HALT mode -	8				

Figure 9-1 Setting the Clock



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

2. Add the peripheral functions that are required for touch measurement and LED control with the use of a hardware timer.

Select the [Components] tab and click on **t** to open the [New Component] dialog box. Select the "Interval Timer" and "Ports" modules and click on [Next].

After that, assign resources for the selected components. The following settings are used in this example.

Rew Component	ion for selected component	×
		#
Interval Timer		
Configuration name:	Config_ITL000	
Operation:	8 bit count mode	~
Resource:	ITL000	~
Caution:		
compare match with	mode and 16 bit count mode together when an interrupt on ITLCMP01 is not to be used as a capture trigger, add 16 bit capture and enable 16 bit count mode ITL012 ITL013 in the GLU	^
Ports		
Configuration name:	Config_PORT	
Resource:	PORT	~

Figure 9-2 Assigning Resources for the Interval Timer and Ports

3. Set up the interval timer. Select the "Config_ITL000" component and make the following specifications.

*Capacitive_Touch_Project_HardwareTimer oftware component configura	tion			Generate Code	Generate Report
Components 🚵 🖄 🖓 🖻 🕀	Configure				١
56 (g) 👘 😜	Clock setting				
type filter text	Operation clock (fITL0)	fSXP	~ 🗲		
✓ ➢ Startup	Clock source	fITL0/128	~	frequency: 0.256 kH	łz)
✓	Interval timer setting				
✓ Drivers	Interval value	20	ms v	al value: 19.53125)	
 Power management and reset fu Config_LVD0 Timers 	Interrupt setting	tch/capture completion (INTITL)			
Config_ITL000 Config_PORT Config_PORT Config_UART0 Config_UART0 Config_UART0 Config_Config_UART0 Config_UART0 Compare	Deselect	Level 3 (low)	2		
<pre>rm_touch </pre>					

Figure 9-3 Setting the Config_ITL000 Component



RL78 Family Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

4. Set up the pin to be used for LED control. Select the "PORT" module and set the P62 pin as an output initially at the high level.

Software component configuration	😼 🎃 Generate Code Generate Repor	
Components 🚵 🖄 🖓 🕀 😥	Configure	-
1 to 1	Port selection PORT4 PORT6	
type filter text		
 ✓ Startup ✓ Seneric 	Apply to all	
✓ Centenc	Unused In Out	Output 1
✓		
 Power management and reset ful 	P60	
Config_LVD0	Unused O In O Out	Output 1
V 🗁 Timers	P61	
Config_ITL000	Unused O In O Out	Output 1
Config_PORT		
✓	P62	
Config_UART0	O Unused O In Out	☑ Output 1
V 😂 Middleware	P63	
✓ Generic	O Unused O In ● Out	Output 1
e r_cisu		
< > ,		>

Figure 9-4 Setting the P62 Pin

5. Click on the 👸 icon in the top right part of the Smart Configurator view to generate code.

After this, follow the steps described in chapter 8, Settings of QE for Capacitive Touch, to complete the settings.



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

9.1.2 Flowchart (Hardware Timer)

Figure 9-5 is a flowchart of the touch measurement control processing with the use of a hardware timer.



Figure 9-5 Touch Measurement Control Processing with the Use of a Hardware Timer



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

9.1.3 Sample Code (Hardware Timer)

The following is a listing of the sample code (qe_touch_sample.c) with the use of a hardware timer in implementing touch measurement.

```
* FILE : qe sample sample.c
* DATE : 2025-02-25
* DESCRIPTION : CTSU2L Program for RL78
* NOTE: THIS IS A TYPICAL EXAMPLE.
#include "qe touch config.h"
#include "Config ITL000.h"
void R CTSU PinSetInit(void);
void qe touch main(void);
uint64 t button status;
#if (TOUCH CFG NUM SLIDERS != 0)
uint16 t slider position[TOUCH CFG NUM SLIDERS];
#endif
#if (TOUCH CFG NUM WHEELS != 0)
uint16 t wheel position [TOUCH CFG NUM WHEELS];
#endif
void qe touch main(void)
  fsp err t err;
  BSP ENABLE INTERRUPT();
  /* Initialize pins (function created by Smart Configurator) */
  R CTSU PinSetInit();
  /* Open Touch middleware */
  err = RM_TOUCH_Open(g_qe_touch_instance_config01.p_ctrl,
g qe touch instance config01.p cfg);
  if (FSP SUCCESS != err)
   {
     while (true) {}
   }
  ITLSO &= ~ 01 ITL CHANNELO COUNT MATCH DETECTE;
  R Config ITL000 Start();
  /* Main loop */
  while (true)
   {
     while ( 00 ITL CHANNELO COUNT MATCH NOT DETECTE == (ITLS0 &
01 ITL CHANNELO COUNT MATCH DETECTE)) {}
     ITLSO &= ~ 01 ITL CHANNELO COUNT MATCH DETECTE;
```



```
/* for [CONFIG01] configuration */
      err = RM_TOUCH_ScanStart(g_qe_touch_instance_config01.p_ctrl);
      if (FSP SUCCESS != err)
      {
          while (true) {}
      }
      while (0 == g_qe_touch_flag) {}
      g qe touch flag = 0;
      err = RM_TOUCH_DataGet(g_qe_touch_instance_config01.p_ctrl,
&button status, slider position, NULL);
if (FSP SUCCESS == err)
      {
          /* TODO: Add your own code here. */
          if (0 != button status)
          {
            P6 bit.no2 = 0;
          }
          else
          {
             P6 bit.no2 = 1;
          }
      }
   }
```



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

10. Documents for Reference

- User's Manual
 - RL78/G22 User's Manual: Hardware (R01UH0978)
 - RL78 Family User's Manual: Software (R01US0015)

The latest versions are available on the Renesas Electronics Web site.

- Technical Update and Technical News The latest information is available on the Renesas Electronics Web site.
- User's Manual: Development Environment
 RL78/G22 Fast Prototyping Board User's Manual (R20UT5121) The latest version is available on the Renesas Electronics Web site.
- Application Note
 - Capacitive Sensor Microcontrollers CTSU Capacitive Touch Introduction Guide (R30AN0424)
 - RL78 Family Using the Standalone Version of QE to Develop Capacitive Touch Applications (R01AN6574)
 - RL78 Debugging Functions Using the Serial Port (R20AN0632)
 - RL78 Family CTSU Module Software Integration System (R11AN0484)
 - RL78 Family TOUCH Module Software Integration System (R11AN0485)
 - Capacitive Sensor Microcontrollers CTSU Capacitive Touch Electrode Design Guide (R30AN0389)
 - ____
 - RL78 Family Using QE and SIS to Develop Capacitive Touch Applications (R01AN5512)
 - The latest versions are available on the Renesas Electronics Web site.

Web pages

- Renesas Electronics Web site
 <u>https://www.renesas.com/</u>
- Fast Prototyping Board page <u>https://www.renesas.com/fast-prototyping-board</u>
- QE for Capacitive Touch page <u>https://www.renesas.com/qe-capacitive-touch</u>
- Capacitive Sensor Unit page
 <u>https://www.renesas.com/solutions/touch-key</u>



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

Revision History

		Descriptio	Description				
Rev.	Date	Page	Summary				
1.00	Mar.20.23	-	First edition				
2.00 Mar.28	Mar.28.25	1	"Introduction" was modified.				
		1	A description of QE for Capacitive Touch was added as Note.				
		4	Chapter 1, Overview, was added.				
		4	The target devices of this document were corrected.				
				4	Table 2-1 was updated.		
		4	SIS modules were added to Table 2-1.				
		4	Table 2-3, Conditions for Verifying Operation, was added.				
		5	Section 2.1, Functions of QE for Capacitive Touch, was added. The contents are equivalent to those of chapter 1, System Overview, in Rev. 1.00.				
		5	Figure 2-1 was updated.				
		6	The procedures for installing CS+ and the Smart Configurator were added to section 3.1.				
			The title of section 3.1 was changed to "Installing Development Tools".				
		6	Cautionary notes on the folder for installing the standalone version of QE for Capacitive Touch was added.				
		7	Descriptions and a note were added to section 3.2.				
		7	Table 3-1, Jumper Settings on the Board, was added.				
		9	Table 4-1 was updated.				
		10	A statement on the method of communications for tuning and monitoring in this application note was added.				
		10	The restrictions on monitoring described in section 7.3.3 in Rev. 1.00 were moved to section 5.1 as a remark.				
		10	Figure 5-1 was updated.				
		10	An overview of the attached sample code was added to section 5.1.				
		10	Table 5-1, Overview of the Attached Sample Code, was added.				
		11	Table 5-2 was updated.				
		13	"Voltage detector (LVD)" and "Port functions (PORT)" were added as required settings to chapter 7.				
		14	Figure 7-5 and a statement of the setting of EVDD were added to section 7.2.				
		15	The descriptions in section 7.3 were updated.				
		16	A cautionary note on the procedure for downloading of SIS modules was added.				
		17	A note on the method of setting the voltage detection function of the RL78/G16 was added.				
		17-18	Section 7.4, Adding Components, was added to describe the procedures for adding the components to be used and assigning resources to them.				
		17-18	Figure 7-10 to Figure 7-12 were added.				
		19-26	Section 7.5, Modifying the Component Settings in the Smart Configurator, was added as an overview of the settings of the components.				



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

	20	The settings of the CTSU component in the attached sample code were updated.
	20	A cautionary note on the design of user circuits was added.
	20	Figure 7-14 was updated.
	21	The indications of the items to be set for the Touch component
		were modified to match the actual labels displayed in the window shown in Figure 7-15.
	22	The descriptions in section 7.5.3 were updated.
	23	A cautionary note on the settings of the pins for UART
		communications was added.
	24	Section 7.5.4, Setting the LVD Component, was added.
	24	Figure 7-19, Setting the LVD Component, was added.
	25	Section 7.5.5, Setting the PORT Component, was added.
	25	Figure 7-21, Setting the PORT Component, was added.
	25	Figure 7-22, Setting P62 and P63 as Outputs at the High Level, was added.
	26	Section 7.5.6, Board Support Package, was added.
		The contents are equivalent to those of step 1 in section 7.6, Generating Code, in Rev. 1.00.
	27	Figure 7-24 and Figure 7-25 were updated.
	28	Figure 7-26, Cautionary Message before Code Generation,
		was added.
	28	The value of the user option byte was modified in Figure 7-27.
	33	The description of step 2 in section 8.3 was updated.
	33	A description and a cautionary note were added to step 3 in section 8.3.
	33	Figure 8-8 was updated.
	34	A description was added to step 6 in section 8.3.
	34	Figure 8-11 was updated.
	35	The description of step 8 in section 8.3 was updated.
	35	Figure 8-13, Creating a New Folder "qe_gen", was added.
	35	The name of a folder was corrected.
	36	A description was added to step 9 in section 8.3.
	36	A point to note on the setting of the power-supply voltage was added to step 10 in section 8.3.
	36	Figure 8-15 was updated.
	38-40	Steps 14 to 18 of the development procedure were modified.
		• The statement on the addition of the qe_gen folder was moved to step 14.
		 The statement on the setting of the standard and mathematical libraries and Figure 8-21 (p. 36 in Rev. 1.00)
		were deleted.
	38,39	The descriptions of steps 14, 15, and 17 in section 8.3 were
	20	updated.
	39	The value of the user option byte was modified.
	39	The value of the user option byte in Figure 8-24 was modified.
	43	Figure 8-31 and Figure 8-32 were updated.
	43	The descriptions of steps 3 to 5 in section 8.4 were updated.
	45	The description of step 9 in section 8.4 was updated. Figure 8-40 was updated.
	47	The descriptions of steps 1 to 3 in section 8.5.1 were updated.
1	40	The descriptions of steps 1 to 3 in section 6.3.1 were updated.
	49	A description was added to step 4 in section 8.5.1.



Using the Standalone Version of QE to Develop Touch Applications for a Fast Prototyping Board

50	Figure 0.44 and Figure 0.45 ware undeted
50	Figure 8-44 and Figure 8-45 were updated.
51	The procedure for setting the touch interface element to be
	displayed on the status chart was modified in step 6 of section
E 4	8.5.1.
51	Figure 8-46 and Figure 8-47 were updated.
52	The description of step A was updated in step 7 of section
	8.5.1.
52	Figure 8-48 was updated.
53	A point to note on data collection was added to step B in step 7 of section 8.5.1.
53	Figure 8-49, Starting Data Collection, was added.
53	The description of step C was updated in step 7 of section 8.5.1.
53	Figure 8-50, Starting Data Collection in the Touch-on State, was added.
54-56	
54-56 56	Figure 8-51 to Figure 8-54 were updated.
20	Software Timer Sample Code
57	(described on p. 54 and p. 55 of Rev. 1.00) was deleted. CTSU port initialization and CTSU module initialization in
57	Figure 8-56 changed to initialization process
57	A title for Figure 8-56 was added.
58	The descriptions in section 9.1 were updated.
58	A point to note on the other implementation of the sample
50	application was added to section 9.1.
59	The description of step 2 in section 9.1.1 was updated.
59	Figure 9-2 was updated.
59	The description of step 3 in section 9.1.1 was updated.
60	Supplementary information regarding the procedure for
00	settings following step 5 in section 9.1.1 was added.
61	Changed Figure 9-5.
	CTSU port initialization and CTSU module initialization were
	changed to initialization process.
	Changed LED lighting processing section to LED control
61	A title for Figure 9-5 was added.
62	A description was added to section 9.1.3.
64	
64	
	pages" in chapter 10, Documents for Reference.
•••	Added reference document (R01AN5512) to chapter 10 (moved from Rev. 1.00 abstract) The page for the Fast Prototyping Board was added to "Web pages" in chapter 10. Documents for Reference.



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 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.

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 systemevaluation test for the given product.

Notice

- 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.
- 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, 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.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 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/.