

## Using QE (standalone ver.) to Develop Touch Applications for FPB board

### Introduction

This application note explains the steps to create an application example that uses capacitive touch sensing using the RL78/G22 FPB (Fast Prototyping Board) (product name: RTK7RLG220C00000BJ) with mounted touch electrodes.

This application note is capacitive touch application development guide using "CS+, standalone version Smart Configurator and standalone version QE for Capacitive Touch".

Using standalone version QE can develop application regardless of device or IDE.

If you are using the RL78/G22 Capacitive Touch Evaluation System (RTK0EG0042S01001BJ) with "CS+, standalone version Smart Configurator and standalone version QE for Capacitive Touch" as an alternative development environment, see the following application note.

• RL78 Family Using the standalone version of QE to Develop Capacitive Touch Applications (R01AN6574)

If you don't use standalone version QE but "e<sup>2</sup> studio, plug-in version Smart Configurator and plug-in version QE for Capacitive Touch" as development environment, see the following application note.

• RL78 Family Using QE and SIS to Develop Capacitive Touch Applications (R01AN5512)

### **Target Device**

RL78/G22 RL78 family with Capacitive Sensing Unit (CTSU)



## Using QE (standalone ver.) to Develop Touch Applications for FPB board

## Contents

1.	System Overview	4
2.	Operating Environment	5
3.	Building the Development Environment	6
3.1	Installation of the Standalone Version QE for Capacitive Touch	6
3.2	Connection of the Target Board	7
4.	Workflow for Developing an Application	8
5.	Application Example	.10
5.1	Application Example Overview	. 10
5.2	List of Used Pins	. 11
6.	Project Creation	.12
7.	Setup of Smart Configurator	.13
7.1	Launching Smart Configurator	. 13
7.2	Setup of Clock and System	. 15
7.3	Setup of SIS (software Integration System) Modules	. 16
7.3.1	Download of SIS Modules	. 16
7.3.2	Setup of CTSU Driver	. 18
7.3.3	Setup of Touch Middleware	. 20
7.4	Setup of Serial Interface (UART)	. 21
7.5	Setting Unused Pins to Low-level Output	. 24
7.6	Generating Code	. 26
8.	Setup of QE for Capacitive Touch	.27
8.1	Launching QE for Capacitive Touch	. 27
8.2	Preparation	. 28
8.3	Configuration	. 30
8.4	Tuning	. 41
8.5	Coding and Monitoring	. 47
8.5.1	Monitoring	. 47
8.6	Sample Code	. 54
8.7	Flowcharts	. 56
9.	Appendix	.57
9.1	Touch Measurement by Hardware Timer	. 57
9.1.1	Setup of Smart Configurator	. 57
9.1.2	Sample Code	. 60
9.1.3	Flowcharts	. 62



## Using QE (standalone ver.) to Develop Touch Applications for FPB board

10. Documents for Reference	64
Revision History	65



#### 1. System Overview

QE for Capacitive Touch is a development tool that supports initial setup and adjusting sensitivity of the touch interface for a development of embedded system using capacitive touch sensors.

The main functions of QE for Capacitive Touch are as follows.

- Creating touch interface configurations It is possible to set visually assignments of touch sensor and positions of touch interface such as button.
- Tuning

It is possible to tune automatically offset and sensitivity of touch interface.

• Monitoring and parameter adjustment It is possible to monitor the performance of touch interface and adjust details of parameters.

Interface gene function	eration	Tuning function	Monitoring function	
Direc Collapsions of Technolous Take New York (See Collapsion) Force (See Collapsion) Force (See Collapsion) Force (See Collapsion) Force (See Collapsion)	X Ingori / Re-stil Kado (K. K. K	Automate Targe Processo     6/7: QE will now measure touch sensitivity for (Button00, TS06 © config01).     In this step please use normal touch pressure on the sensor for once. Press any     key on the PC keyboard to accept the sensitivity measurement.     Button01, TS05 © config01: 15328     Caset Help	Nor Toxa Corr 11 And Reading Savet B. Jacob B. and B. and B. and B. Stark Street B. B. Corr Mark, 1920 Mercen Mark (1920, Benefit), B. B. Barrow Mark (1920) Mercen Mark (1920, Benefit), B. B. Mark (1920) Mercen Mark (1920, Benefit), B. Mark (1920) Mercen Mark (1920) Mercen Ma	Ranas.
	See service Week Soy peet 30 General Viet Shaked Pro Catcherer Server Catenerer Catenerer		1654	
The function in the free face take Cher Adapted 'Sa	Cause Cause New		1114	

Figure 1-1. Main Functions of QE for Capacitive Touch



## 2. Operating Environment

Table 2-1 and Table 2-2 show the operating environment for this application note.

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

This application note can be utilized for other devices from the "RX/RA/RL78 family and Renesas Synergy <sup>™</sup> platform" with capacitive touch IP.

Table 2-1.	Operating	Environment	(Software)
------------	-----------	-------------	------------

Items	Contents	Version
IDE	CS+ for CC	8.09.00 or later
Toolchains	CC-RL	1.12.00 or later
QE	Standalone Version QE for Capacitive Touch	3.2.0 or later
Smart Configurator	RL78 Smart Configurator	1.5.0 or later

Caution When using the CC-RL free evaluation edition V1.12.00 or later for tuning of touch sensors, select "debug precedence(-onothing)" as the optimization levels.

#### Table 2-2. Operating Environment (Hardware)

Items	Contents
Microcontroller used	RL78/G22 (R7F102GGE2DFB)
Target Board	RL78/G22 Fast Prototyping Board (RTK7RLG220C00000BJ)



### 3. Building the Development Environment

This chapter explains how to install tools and connect the board to PC.

This application example uses the following tools.

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

This chapter will not explain how to install CS+ and Smart Configurator. If you haven't installed them yet, install them according to their procedure.

### 3.1 Installation of the Standalone Version QE for Capacitive Touch

Install standalone version QE for Capacitive Touch by taking the following steps.

If you have already installed, this section is not necessary.

- 1. Download "QE for Capacitive Touch" from Renesas Electronics website.
- The downloaded zip file has plugin version and standalone version. Extract the downloaded zip file. Then choose a folder for extraction which windows file path is not over the character limit (260 characters). For example, in the directory of "C:\Renesas".



## 3.2 Connection of the Target Board

Connect the target board to the PC.

Following Figure 3-1, connect the target board to PC. via USB.

In this application example, power is supplied to the target board via USB. Confirm the circuits on the target board, and then set switches or jumpers as necessary.

For the application example, set the jumpers of the target board as follows.

- JP16 : Open when performing the QE serial connection function
  - : Closed when performing the COM PORT debug connection function
- JP17 : 1-2 short



Figure 3-1. Target Board and PC Connection



Using QE (standalone ver.) to Develop Touch Applications for FPB board

## 4. Workflow for Developing an Application

This chapter explains how to create an application.

Follow the steps in the workflow of QE for Capacitive Touch to develop an application.



Figure 4-1. Workflow for Developing an Application

Table 4-1 shows each step within the workflow. Chapter numbers in the table are linked to the corresponding chapter page. Click each chapter number in the table to see how to use each function. IDE and Smart Configurator is used for project creation and coding, project build, and debug.



Using QE (standalone ver.) to Develop Touch Applications for FPB board

## Table 4-1. Items of QE for Capacitive Touch

Items				Capture
Preparation	Project Creation	reation Creating Project Using IDE		6
		Setup of	Smart Configurator	7
			Setup of Clock and System	7.2
			Setup of CTSU Driver	7.3.2
			Setup of Touch Middleware	7.3.3
			Setup of Serial Interface (UART)	7.4
			Setting Unused Pins to Low-level Output	7.5
	To Select a Proje	ct Folder		8.2
	To Select a MCU	Name		
Configuration	To Prepare a Cor	figuration		8.3
	To Output Files for	or Tuning		
	To Implement Pro	gram		
	To Build Project			
Tuning	To Execute Progr	am		8.4
	To Start Tuning			
	To Output Param	eter Files		
Coding and Monitoring	To Implement Pro	ogram		8.5
	To Launch Debug	1		
	To Connect UAR	Г		
	To Enable Monito	ring		



## 5. Application Example

## 5.1 Application Example Overview

This application note provides an example of an application which uses two buttons and one slider.

From chapter 6 onward, the application note explains how to create the application and monitor whether the either of the buttons or the slider is touched.



Figure 5-1. Application Example



## 5.2 List of Used Pins

Table 5-1 shows the pins used in this application example.

UART communication and touch sensors in the application depend on the target board you are using

Table 5-1.	List of Used Pins	for Application	Example
------------	-------------------	-----------------	---------

Items	Pins	Uses
UART Communication	RxD0/P11	Tuning
	TxD0/P12	Monitoring
Touch Sensor 1	TS24/P26	Button (TS_B1)
Touch Sensor 2	TS23/P25	Button (TS_B2)
Touch Slider	TS20/P22	Slider (TS_S)
	TS21/P23	
	TS22/P24	

Figure 5-2 shows positions of the touch sensors used for this application example.



Figure 5-2. Position of Touch Sensors



### 6. Project Creation

Launch CS+ and create new project.

In "Create Project" dialog, select the following.

- Microcontroller
- : RL78 : R7F102GGExFB (48pin)
- Using microcontroller
   Kind of project
- Kind of project : Application (CC-RL) : (Any project name)
- Project name Place
- : (Any place)

Using microcontroller:	
(Search microcontrol	ller) Update
RL78/G22 (ROM R7F 102G6Ex: R7F 102GGEx R7F 102GGEx	t64KB) (SP(20pin) (SP(20pin) (SPE(48pin)) (NP(48pin)) (NP(48pin))
Kind of project	Application(CC-RL)
Project <u>n</u> ame:	Capacitive_Touch_project_Example Capacitive_Touch_Project_Ex
Place:	C#CS+_Workspace
	Make the project folder
0.W00, WL I	acitive_Touch_project_Example¥Capacitive_Touch_project_Example.mtpj
C#US+_Workspace#Uapa	ition of an existing project to the new project
□ Pass the file composit	that of all existing project to the new project
Project to be passed	(Input project file to be diverted)
Pags the file composition file	(Input project file to be diverted) Browse
0.V00. W. I	acitive_Touch_project_Example¥Capacitive_Touch_project_Example.mtpj

Figure 6-1. Creating New Project



## 7. Setup of Smart Configurator

This chapter explains how to set by Smart Configurator. Necessary setup for this application example is the following.

- Clock and system
- CTSU driver
- Touch middleware
- Serial interface (UART Communication)
- Unused pins to low-level output

## 7.1 Launching Smart Configurator

Double-click "Smart Configurator" in "Project Tree" of CS+, and launch Smart Configurator.



Figure 7-1. Launching Smart Configurator

If Smart Configurator cannot be launched, confirm the following.

- Whether file path in the property of Smart Configurator is correct.
- Whether "Smart Configurator for RL78 Communication Plug-in" is selected in "Tool" -> "Plug-in Manager" of Menu.

Project Tree 4 X		Property			* x
202 2	9	Smart Configurator Property		٩	- +
Capacitive Touch project Example (Project)	~	Product Information			
— R7F102GGExFB (Microcontroller)		Version	V1.00.04.03 [16 Nov 2022]		
— Smart Configurator (Design Tool)	×	Smart configurator setting		-	
- K CC-RL (Build Tool)		Smart Configurator for RL78 executable file path	C:¥Program Files (x86)¥Renesas Electronics¥SmartConfigurator¥RL78¥eclipse¥SmartConfigurator.exe	< C	
				-	
- Program Analyzer (Analyze Tool)					
	L				

Figure 7-2. File Path of Smart Configurator



	Plug-in Manager	×	
	Checked plug-ins are loaded at the CS+ start-up. These settings are enabled at the next start-up. <sup>+</sup> You can never uncheck a check box of the grayout plugin that i recommended that the checkboxes of the plug-in for the target mix Down F and the function	s required by the CS+. Also, on the [Basic Function] tab, it is rocontroller of the development are not cleared.	
	Module Name	Description	
_	Code Generator Plug in Gode Generator Plug in Gode Generator /Plu/ew Plug in Gode Generator /Plu/ew Plug in Editor plug in DLL Gode Longingutor Plug in Plug	Plug in to generate the device driver automatically ifor V8 Plug in to generate the device driver automatically and to DebugConcole plug in to support using standard I/O. SEditor DLL It is a console where the IronPython commands and the C Plug in to define the device pin configuration. Plug in to analyze program. Plug in for application development that contains useful to Plug in for application development that contains useful to Plug in to communicate with Smart Configurator for RL781 Utility to display and adjust stack usage of each functions Plug in to communicate with CS+ Update Manager.	
		UK Cancel Help	

Figure 7-3. Plug-in Manager



#### 7.2 Setup of Clock and System

This section explains how to set the clocks and system.

Clocks configuration	Generate Code: Generate Rep
(acia) ao ità	
Operation model . Right speed mate mans 2,705-1500	High-speed main mode 2.7(V)~5.5(V)
N fige cases on ridg on face fragments 0000 new white the same and Other is uniting to dark the last same and of secondary is 00000 new of source the last fige and during multiples at 000000 new of source the last same and of secondary is	32 Min
Low speed at chip software Response <u>30,700</u> ding	
If the section           Operations mode           Department           Programs           Vith and interest mode           Vith and interest mode	

1. Select "Clocks" tab in lower-middle menu, and set clocks.

Figure 7-4. Setup of Clocks

2. Select "System" tab and set the debug environment.

System configuration				Gene	Tate Code Generate	Report
(2)						
* On-chip debug setting						
On-chip debug operation setting O Unused	O Use emulator	COM Port	-	Use COM po	ort	
Emulator setting	(i) E2 Lite					
Pseudo-RRM/DMM function setting	9 • Used					
Start/Stop function setting Unused	⊖ Used					
Monitoring point function setting	O Used					
Security ID setting Use security ID Security ID	Uncheck "Use se	curity ID″				
Security ID authentication failure se	tting					

Figure 7-5. Setup of Debug



## 7.3 Setup of SIS (software Integration System) Modules

This section explains how to add two SIS modules which are "CTSU Driver" and "Touch Middleware" used for QE for Capacitive Touch and set them.

#### 7.3.1 Download of SIS Modules

Download "CTSU Driver" and "Touch Middleware" by Smart Configurator.

If you have already installed, this section is not necessary.

1. Select "Components" tab and click the 🖆 icon.



Figure 7-6. Software Component Configuration



2. Click "Download RL78 Software Integration System modules" at lower of "New Component" dialog.

New Co	omponent		-		×
Software	Component Selection				
Salact cor	monont from those quailable in	list		1	
Select cor	nponent from those available in	list			
Category	All				~
Function	۵۱				~
runcuon					-
Filter					
Compon	ents	Short Name	Туре	Version	^
#A/D C	onverter		Code Generator	1.2.0	
# Board	Support Packages v1.40	r_bsp	RL78 Software	1.40	
# Capaci	itive Sensing Unit driver.	r_ctsu	RL78 Software	1.30	
Clock	Output /Buzzer Output Controll		Code Generator	1.3.0	
# Data T	ransfer Controller		Code Generator	1.2.0	
# Delay	Counter		Code Generator	1.3.0	
# Divide	r Function		Code Generator	1.3.0	
# Event I	Link Controller		Code Generator	1.1.0	
# Extern	al Event Counter		Code Generator	1.3.0	
# IIC Cor	mmunication (Master mode)		Code Generator	1.4.0	
# IIC Cor	mmunication (Slave mode)		Code Generator	1.3.0	
🖶 Input F	Pulse Interval/Period Measurem		Code Generator	1.3.0	
# Input S	ignal High-/Low-Level Width		Code Generator	1.3.0	~
Show o	only latest version				
Descriptio	n				
The analo	og to digital (A/D) converter is fu	Inction for converting a	analog inputs to digi	tal	^
signals.					
			_		~
Download	RL78 Software Integration Syste	m modules			
Configure	general settings				
?	< Back	Next >	Finish	Cancel	

Figure 7-7. Software Component Selection Dialog Box

- 3. Select the following, and click "Download".
  - RL78 Family CTSU Module Software Integration System
  - RL78 Family TOUCH Module Software Integration System

	Title	Document No.	Rev.	Issue date	^	Select All
	RL78 Family FS3000 Sensor Control Mod	R01AN6195EJ	Rev.1.00	2022-06-30		Developed All
	RL78 Family FS1015 Sensor Control Mod	R01AN6198EJ	Rev.1.00	2022-06-30		Deselect Al
	RL78 Family HS400X Sensor Control Mo	R01AN6446EJ	Rev.1.00	2022-06-30		
	RL78 Family HS300x Sensor Control Mo	R01AN6194EJ	Rev.1.20	2022-05-20		
	RL78 Family CTSU Module Software Inte	R11AN0484EJ	Rev.1.20	2022-04-20		
	RL78 Family TOUCH Module Software In	R11AN0485EJ	Rev.1.20	2022-04-20		
1 0	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 7-8. Download SIS Modules



#### 7.3.2 Setup of CTSU Driver

This subsection explains how to set "CTSU Driver".

1. Select "Components" tab and click 🖆 icon. In the displayed dialog, select "r\_ctsu" module and click "Finish".

Capacitive Sensing Unit driver.	r_ctsu	RL78 Software Integration System	1.30



 Click "r\_ctsu" module and enable TS pins used for this application example. In this application example, five TS pins are used. Please check user's manual of your target board in order to confirm assignment between TS pins and touch sensor.

Components	2 2 2 1 2 E	Configure		
No 15	2.2	Property	Value	
type filter text		<ul> <li>Configurations</li> </ul>		
On Stadue		# Parameter check	Use system default	
- Startup		# Data transfer of INTCTSUWR and INTCTSURD	Interrupt handler	
v er Generic		# DTC setting	Setting in r. ctsu	
Middleware		# Auto-judgment function in Snooze mode using SMS	Disable	
<ul> <li>Middlewate</li> <li>Cononic</li> </ul>		# Data storage address setting for CTSURD		
v e denenc		Data storage address setting for CTSUWR		
1,000		# Interrupt level for INTCTSUWR	Level 2	
		# Interrupt level for INTCTSURD	Level 2	
		# Interrupt level for INTCTSUFN	Level 2	
		Output port number for external trigger	PORT14	
		# Bit number for external trigger output	BITO	
		Interrupt port number for external trigger		
		✓ ◎ Resources		
		✓		
		TSCAP Pin     TSCAP Pin     TSCAP Pi	N 🖉 Used	
		🛰 TS00 Pin	🖾 Used	
		🕆 TS01 Pin	🖾 Used	
		🛰 TS02 Pin	🖂 Used	
		🛰 TS03 Pin	🖾 Used	
		🛰 TS04 Pin	🛄 Used	
		➤ TS05 Pin	🛄 Used	
		∽ TS06 Pin	🖾 Used	
		∽ TS07 Pin	🖂 Used	
		🛰 TS08 Pin	🖾 Used	
		∽ TS09 Pin	🖾 Used	
		∽ TS10 Pin	🖾 Used	
		STS11 Pin	🖾 Used	
		TS12 Pin     TS12 Pin	🖾 Used	
		∼ TS13 Pin	Used	
		∼ TS14 Pin	🗇 Used	
		™ TS15 Pin	🗆 Used	
		► 1516 Pin	Used	
		∼ 151/ Pin	E Used	
		T 1518 Pin	E Used	
		TC00 Dia	( Build	
		5 1520 PIN	IN Used	
		~ 1521 Pm	i Used	
		1522 Pin	12 Used	
		IS24 Pin	in Used	
		<ul> <li>1524 Pm</li> <li>TC25 Pm</li> </ul>	V Used	
		T 1363 MR	E Used	
		~ 1500 PIN	E Used	
		5 TC10 Dia	E Used	
		™ 12C0 PIR	El Usea	
	~			0
L				

Figure 7-10. Enable Used TS Pins



3. It is recommended to set unused TS pins to low-level output. In CTSU2, when TS pins not used in the application are enabled, the TS pins are set to low-level output as non-measurement pins.

In this application example, enable all TS pins, including unused pins. Note that pins TS12/TS13 pins are excluded as their dual functions are used in the application.

In designing your circuit, make sure to perform sufficient pin processing and satisfy electrical characteristic requirements.

Components	한 년 1일 문 문	Configure			
51 1d	2.2	Droparty		Jalue	
type filter text		× @ Configurations		value	
type inter text	1000	# Darameter check	1	ke ovtem default	
<ul> <li>Startup</li> </ul>	^	# Data transfer of INTCTSLIMP and INTCTSLIPD		nternint handler	
- Generic		# DTC setting		etting in r.ctsu	
e r_bsp		# Auto-judgment function in Spooze mode using SMS	-	)isable	
Middleware		# Data storage address setting for CTSLIRD		VEESOO	
Generic		# Data storage address setting for CTSUWR			
r_ctsu		# Interrupt level for INTCTSUWR		evel 2	
		# Interrupt level for INTCTSURD	L	evel 2	
		# Interrupt level for INTCTSUFN	L	evel 2	
		# Output port number for external trigger		ORT14	
		# Bit number for external trigger output			
		# Interrupt port number for external trigger			
		✓ <sup>I</sup> Resources			
		✓ ⓓ CTSU	1		
		∼ TSCAP Pin	1	Used	
		🛰 TS00 Pin		Used	
		🛰 TS01 Pin	6	Used	
		🛰 TS02 Pin	6	/ Used	
		🛰 TS03 Pin		Used	
		🛰 TS04 Pin		Used	
		∼ TS05 Pin	6	Used	
		🛰 TS06 Pin		/ Used	
		∼ TS07 Pin		l Used	
		∼ TS08 Pin		! Used	
		∼ TS09 Pin	6	l Used	
		™ TS10 Pin		l Used	
		∼ TS11 Pin		l Used	
		∼ TS12 Pin		Used	
		∼ TS13 Pin		Used	
		∼ TS14 Pin		Used	
		∼ TS15 Pin		Used	
		™ 1516 Pin		Used	
		∼ 1517 Pin		Used	
		~ 1518 Pin		Used	
		~ 1519 Pin		Used	
		T 1520 Pin		Used	
		5 1521 Pin		Used	
		T 522 Pin		Used	
		TS24 Pin		Used	
		TS25 Pin		/ Used	
		TS26 Pin	1	/ Used	
		TS27 Pin		/ Used	
		TS28 Pin		/i Used	

Figure 7-11. Enable TS Pins Unused in the Application



#### 7.3.3 Setup of Touch Middleware

This subsection explains how to set "Touch Middleware".

Monitoring touch performance for touch applications is possible by communication via the OCD (On-Chip Debugging) emulator. However, in RL78 family case, monitoring performance is limited by the OCD function of the RL78 family.

Monitoring touch performance using serial communication enable smooth monitoring. Also it is possible to tune using serial communication.

1. Click 🔽 icon and select "rm\_touch" module in the displayed dialog, and click "Finish".

touch mudieware.	ini_toden	RE70 Software integration System	1.50
	Figure 7-12. "rm_t	ouch" Module	

- 2. Click "rm\_touch" module and set the following.
  - Enable to support QE monitor using UART
  - Enable to support QE tuning using UART
  - UART channel UART0

UART channel to set depends on your target board.

Components	à 4 l²₂ ⊟ ⊞	Configure		
Type filter text	10 T	Property		Value
✓ ➢ Startup	^	# Parameter check # Support QE monitor using UART	Enable	Use system default
r_bsp		# Support QE tuning using UART # UART channel	Enable	Enable UART0
<ul> <li>✓ Generic</li> <li></li></ul>	n	# Type of chattering suppression	UART0	TypeA : Counter of exceed threshold is ho
	ý			

Figure 7-13. Setup of "rm\_touch" Module



## Using QE (standalone ver.) to Develop Touch Applications for FPB board

#### 7.4 Setup of Serial Interface (UART)

This section explains how to set UART for tuning and monitoring of touch sensors.

The UART channel and port to be set depend on your target board.

- 1. Click 🔽 icon. In the displayed dialog, select "UART Communication" module and click "Next". Then set as follows and click "Finish".
  - Operation : Transmission/reception
  - Resource : UART0

 UART Communication
 Code Generator
 1.4.0

Figure 7-14. "UART Communication" Module

UART Communication Configuration name: Config_UART0 Operation: Transmission/reception Transmission/reception Resource: UART0 VART0	New Component	on for selected component	— C	×
Operation: Transmission/reception Transmission/reception Resource: UARTO VARTO V	UART Communication	Config_UART0	_	Î
	Operation: Resource:	Transmission/reception	Transmission/rece	eption

Figure 7-15. Select UART Channel



2. Click the added "UART Communication" module and set the operation clock and transfer rate (baud rate) in the transmit and receive sections.

Components da să 15 11 18	Configure				6
No. 65	Transmission Reception				
type filter text	UART0 clock setting				
<ul> <li>Startup</li> </ul>	Operation clock	CK00	÷	CK00, fCL	K/2^3
e r bsp	Clock source	fCLK/2^3	¥ 1	Clock frequency: 4000 kHz)	
<ul> <li></li></ul>	Data length setting				
<ul> <li>Communications</li> <li>Condition UKBTD</li> </ul>	⊖7 bits ●8 bits	O9 bits			
✓ ar Middleware	Transfer direction setting				
👻 👄 Generic	158	O M58			
e r_ctsu	Parity setting				
•	None     O parity	Odd parity	Ofv	en parity	
	Stop bit length setting 1 bit fixed				
	Receive data level setting				
	Non-reverse	OReverse			
	Transfer rate setting				
	Transfer rate setting	153600		(bps) 153600	
		(Current error: 0.169	6, the min	imum is -249%, the maximum is 4.42%	
	Interrupt setting				
	Reception end interrupt priority (INTSR0)	Level 3 (low)	~		
	Reception error interrupt priority (INTSRE0)	Level 3 (low)			
	Callback function setting				
	Reception end	Reception error			
Overview Board Clocks System Components Software component configura	Pina Interrupt			Gene	13 Binnate Code Generate Report
Overview Board Clocks System Components Software component configura Components as ab its III	Pris Interrupt tion Configure			Gene	19 @ rate Code: Generate Repor
Overview Board Clocks System Components Software component configura Components au cb /5 III II Ni cf & Components	Pens Interrupt tion Configure Transmission Reception			Gene	Tate Code Generate Repor
Overview Board Clocks System Components Software component configura Components as ub 1% III II Ni the U Component Configuration Ni the U Component Configuration Ni the U Component Configuration Ni the U Configuration	Pres Interrupt tion Configure Reception UART0 clock setting			Gene	G rate Code Generate Repor
Overview Board Clocks System Component Software component configura Components as ub 1% III II Ni the total Type Filter text * © Startup * © Generic	Pins Interrupt tion Configure Bansmission Reception UART0 clock setting Operation clock	CK00	×	Gene CK00, fCL	rate Code Generate Rapor
Overview Board Clocks System Component Software component configura Components as ub 1% III II Ni the to the total Type Filter text * © Startup * © Generic @ c_bup	Pres Interrupt tion Configure Bansmission Reception UART0 clock setting Operation clock Clock source	CK00 FCLK/2^3	< <	Gene CK00, fCL Cock frequency: 4000 kHz;	rate Code Generate Rapor
Overview Board Clocks System Component Software component configura Components as ab in 8 in No 10 in 10 in 10 in Type Filter text * Startup * Generic * c.bsp * Drivers * Drivers	Pres Interrupt tion Configure Bansmission Reception UAR10 clock setting Operation clock Clock source Data length setting	CK00 FCLK/2^3	× ×	Cerve CK00, fCL Cock frequency: 4000 kHz	Rate Code Generate Rapor
Overview Board Clocks System Component Software component configura Components as ab in the in Ni the Software test Type Filter test * Startup * Generic * Configurations * Drivers * Communications * Configurations	Pres Interrupt tion Configure UART0 clock setting Operation clock Clock source Data length setting O thes	CK00 FCLK/2^3 O 9 bits	< <	Gene CK00, fCL Cock frequency: 4000 kHz)	Rate Code Generate Rapor
Overview Board Clocks System Component Software component configura Components as ab in the in Ni the Startup * Startup * Seenric * Communications * Config UARTO * Middleware	Pres Interrupt tion Configure UART0 clock setting Operation clock Clock source Data length setting O 7 bits Transfer direction setting Transfer direction setting	CK00 FCLK/2^3 () 9 bits	< <	Gene CK00, fCL Cock frequency: 4000 kHz)	rate Code Generate Rapor
Overview Board Clocks System Component Software component configura Components as ab in the in No of the interview of the	Pres Interrupt tion Configure Bansmission Reception UAR10 clock setting Operation clock Clock source Data length setting O 7 bits Transfer direction setting ELS8 Data term	Скоо FCLK/2^3 () 9 bits () MS8	× ×	Gene CK00, fCL Cock frequency: 4000 MHz)	Rote Code Generate Report
Overview Board Clocks System Components Software component configura Components Supporting that Supporting tha	Pine Interrupt tion Configure Transmission Reception UAR10 clock setting Operation clock Clock setting Operation clock Clock setting O 7 bis Transfer direction setting O 158 Panity setting Panity setting O nacity	CK00 fCLK/2^3 0 9 bits 0 M58	* *	Gene Cock frequency: 4000 kHz)	Rate Code Generate Report
Overview Board Clocks System Components Software component configura Components data 5 11 11 11 11 12 12 12 12 12 11 12 12 12 12 12 12 12 12 12 12 12 12 13 12 12 12 12 12 12 14 12 12 12 12 12 12 12 12 15 12 12 12 12 12 12 12 12 12 12 12 12 12	Pine Interrupt tion Configure UAR10 clock setting Operation clock Clock setting Operation clock Clock setting Operation setting O 7 bits Baits Transfer direction setting O 158 Parity setting O parity Stop bit length settine	CKD0 fCLK/2^3 9 bits MS8 Odd parity	v v	Gene Cock frequency: 4000 kHz) en parity	Rate Code Generate Report
Overview Board Clocks System Components Software component configura Components and 15 III II III III IIII Viat Components IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Pine Interrupt  tion  Configure  Itansmission Reception UAR10 clock setting Operation clock Clock source Data length setting O bits Transfer direction setting ELS8 Parity setting Name O parity Stop bit length setting I bit flead	CK00 FCLK/2^3 9 bits MSB Odd parity	~ ~	Gene Clock frequency: 4000 kHz) en parity	Rober Generate Report
Overview Board Clocks System Components Software component configura Components and 15 III III III IIII V De filter text V De Startup V De Computations Config LUARTO V De Config LUARTO V De Generic Config LUARTO V De Generic	Pins Interrupt tion Configure UAR10 dock setting Operation clock Clock source Data length setting 7 bits % 8 bits Transfer direction setting % IS8 Parity setting % None 0 parity Stop bit length setting 1 bit fileed Receive data level setting	CK00 FCLK/2^3 9 bits MS8 Odd parity	v v O Ev	Gene CK00, fCL Clock frequency: 4000 kH2 en parity	Rote Code Generate Report
Overview Board Clocks System Components Software component configura Components and N III III III E IIII View Computer text Components and N IIII IIIII Components IIIIII IIIIII IIIIII IIIIII IIIIII	Pine Interrupt  tion  Configure  UART0 dock setting Operation clock Clock source Data length setting O this Transfer direction setting ELS8 Parity setting None Data length setting Data l	CK00 FCLK/2^3 9 bits MS8 Odd parity	v v	Gene CK00, fCL Clock frequency: 4000 kH2 en parity	K/2 <sup>n</sup> 3
Overview Board Clocks System Component Software component configura Components and N III N IIII N III N IIII N III N IIII N IIIII N IIII N IIIII N IIII N IIIIIII N IIIII N IIIII N IIIII N IIIII N IIIII N IIIIII N IIIII N IIIII N IIIIII N IIIII N IIIII N IIIII N IIIII N IIIII N IIIIII N IIIIII N IIIII N IIIIII N IIIIII N IIIIII N IIIIII N IIIIII N IIIIII N IIIIII N IIIIII N IIIIII N IIIIIII N IIIIII N IIIIII N IIIII N IIIIII N IIIIII N IIIIII N IIIIII N IIIIII N IIIIII N IIIII N IIIIII N IIIIII N IIIII N IIIIIIII N IIIII N IIIII N IIIIIII N IIIIII N IIIIIIIII N IIIIIIII N IIIIIIIIII	Pres Interrupt  tion  Configure  UART0 clock setting Operation clock Clock source Data length setting O tata length setting ELS8 Parity setting EL	CK00 FCLK/2^3 9 bits MSB Odd parity Reverse	v v	CK00, fCL Cox frequency: 4000 kHz en parity	K/2^3
Overview Board Clocks System Component Software component configura Components and A III III IIII IIIIIIIIIIIIIIIIIIIIII	Pine Interrupt  tion  Configure  UART0 clock setting Operation clock Clock source Data length setting O that length setting O to a bits Transfer direction setting O parity Stop bit length setting I bit fixed Receive data level setting Data setting Transfer rate setting Transfer rate setting Transfer rate setting	CK00 fCLX/2^3 9 bits MS8 Odd parity Reverse 153600	v v Otv	CK00, fCL Cook frequency: 4000 kHz en parity	K/2^3
Overview Board Clocks System Component Software component configura Components and A IIII Nited Component V Diverse V Drives V Drives V Communications Config LUNITO V Drives V Communications Components V Communications Communicat	Pres Interrupt  tion  Configure  UART0 clock setting Operation clock Clock source Data length setting O to bia Transfer direction setting E LSB Parity setting E None Data length setting D to fixed Receive data level setting D to fixed Receive rate setting Transfer rate setting Transfer rate setting	CK00 FCLK/2^3 9 bits MS8 Odd parity Reverse 153600 (Current error 0.163	с Обл	CERN CK00, fCL Clock frequency: 4000 kH29 en parity (bps) 153600 inum is -449%, the treasmont is 44500	Rate Code Generate Report
Overview Board Clocks System Component Software component configura Components and A IIII Nited a Configurations * © Startup * © Generic * Configurations * © Computations * © Configurations * © Configurations	Pres Interrupt  tion  Configure  UART0 clock setting Operation clock Clock source Data length setting O to bits Transfer direction setting E IS8 Parity setting None Data length setting D	CK00 FCLK/2^3 9 bits MS8 Odd parity Reverse 153600 (Current error: 0.169		CERN CK00, fCL Clock frequency: 4000 kH29 en parity (bpt) 153600 inum is -49%, the reasonant is 4.4240	K/2^3
Overview Board Clocks System Component Software component configura Components and A II II N at Software Type Titler test	Pres Interrupt  tion  Configure  UAR10 clock setting Operation clock Clock source Data length setting O to bits Transfer direction setting O to bits Transfer direction setting O parity Stop bit length setting I bit fixed Receive data level setting Data fransfer rate setting Transfer rate setting Transfer rate setting Interrupt setting Reception end interrupt priority (INTSR0)	CK00 (CLK/2^3 9 bits MS8 Odd parity Reverse 153600 (Current error: 0.169 (Lervet 3 (low)	<ul> <li>✓</li> <li>✓</li> <li>✓</li> <li>✓</li> <li>✓</li> <li>✓</li> <li>✓</li> <li>✓</li> <li>✓</li> </ul>	Cene CK00, fCL Cock frequency: 4000 KH2 en parity (bps) 153600 inum is -49%, the resonant is 4.42%	K/2 <sup>A</sup> 3
Overview Board Clocks System Components Software component configura Components duil 17 III II IIII Components IIIII Components IIIII Components IIIII Components IIIIII Components IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Pine Interrupt  tion  Configure  Tansmission Reception UAR10 clock setting Operation clock Clock source Data length setting Opinity @ 8 bits Transfer direction setting ③ 1 bits Receive data level setting 1 bit fixed Receive data level setting Transfer rate setting Transfer rate setting Transfer rate setting Reception error interrupt priority (INTSRE()  Reception error interrupt priority (INTSRE()	CK00 (CLK/2^3 ) 9 bits MS8 Odd parity Reverse 153600 (Current error: 0.169 Level 3 (low) Level 3 (low)	<ul> <li>✓</li> <li>✓</li></ul>	Cent CKOO, fCL Cock frequency: 4000 kHz) en parity (bps) 153600 imum is -449%, the recomum is 4.42%	K/2 <sup>A</sup> 3
Overview Board Clocks System Components Software component configura Components duild 15 II	Pine Interrupt  tion  Configure  Tansmission Reception UAR10 clock setting Operation clock Clock source Data length setting O 1 bits  Transfer direction setting Stop bit length setting 1 bit flaed Receive data level setting Transfer rate sett	CK00 (CLK/2^3 ) 9 bits MS8 Odd parity Reverse [153600 (Current error: 0.169 Level 3 (low) Level 3 (low)	<ul> <li>✓</li> <li>✓</li></ul>	Cock frequency: 4000 kHz) en parity (bpk) 153600 insum is -4.49%, the maximum is 4.424	K/2 <sup>A</sup> 3
Overview Board Clocks System Component Software component configura Components duild 15 II II The II The III Startup Startup Communications Communicati	Pine       Interrupt         tion       Interrupt         Configure       Interrupt         Tansmission       Reception         UAR10 clock setting       Operation clock         Clock source       Data length setting         0 7 bits       ® 8 bits         Transfer direction setting       0 parity         Stop bit length setting       1 bit flad         Receive data level setting       None         Tansfer rate setting       Transfer rate setting         Transfer rate setting       Transfer rate setting         Interrupt setting       Reception end interrupt priority (NTSR0)         Interrupt setting       Reception end interrupt priority (NTSR0)	CKD0 fCLK/2^3 9 bits M58 Odd parity Reverse 153600 (Current error: 0.169 Level 3 (low) Level 3 (low)	о обл обл	Cock frequency: 4000 kHz) en parity (bps) 153600 incurn is -1.9%, the maximum is 4.000	K/2 <sup>A</sup> 3

Figure 7-16. Setup of "UART Communication" Module (UART0)

- 3. Select "Pins" tab and assign the following pins to the UART (SAU00) channel.
  - RxD0 : 21
  - TxD0 : 20



Type filter text       type filter text (* = any string, ? = any character)       All	Hardware Resource 🛛 🕀 🛱 🏯	Pin Fun	ction							1	
▲ All       Pin NumDirecti       Remarks       Comme <sup>©</sup> Clock Generator <sup>©</sup> RxD0        PioR1 <sup>PIDR</sup> <th>Type filter text</th> <th>type fi</th> <th>lter text (* =</th> <th>any string,</th> <th>? = any character)</th> <th></th> <th></th> <th></th> <th></th> <th>All</th> <th>,</th>	Type filter text	type fi	lter text (* =	any string,	? = any character)					All	,
is UO Ports          PixD0       PIOR1          P11/5100/RxD0/TOOLRxD/SD400/TS12/TI06/TOO6          / 21         Multiple pin functions on the same pin          is Clock Generator          SCK00       PIOR1          Not assigned           Not assign None          is Clock Gutput/Buzzer Output Controller           SCK00       PIOR1          Not assigned           Not assign None          is AnD Converter           SD00       PIOR1          Not assigned           Not assign None          is SAU0           SD00       PIOR1          Not assigned           Not assign None          is SAU0           SD00          PIOR1          Not assigned           Not assign None          is SAU0           SO00          PIOR1           Not assigned           Not assign None          is SAU00           SO00           PIOR1           Not assign None           Not assign None          is SAU01           SAU0           SO00           PIOR1           Not assign None           Not assign None          is SAU01           SAU0           SAU0	∆ All	Ena	Function	PIOR	Assignment	Pin N	um Dire	ecti	Remarks		Comments
Clock Generator SCK00 PIOR1 * Not assigned * Not assign None   0. Timer Array Unit SCL00 PIOR1 * Not assigned * Not assign None   0. Real-Timer Clock PIOR1 * Not assigned * Not assign None   • Clock Output/Buzzer Output Controller SDA00 PIOR1 * Not assigned * Not assign None   • Clock Output/Buzzer Output Controller SDA00 PIOR1 * Not assigned * Not assign None   • Clock Output/Buzzer Output Controller SD00 PIOR1 * Not assigned * Not assign None   • Clock Output/Buzzer Output Controller SD00 PIOR1 * Not assigned * Not assign None   • o SAU00 • SAU00 • SAU00 • Not assigned * Not assign None   • SAU01 • SAU02 • SAU02 • SAU01 • Interrupt None   • SAU10 • SAU10 • SAU10 • SAU11 • SAU11   • SAU11 • Sau11	te I/O Ports		@ RxD0	PIOR1	P11/SI00/RxD0/TOOLRxD/SDA00/TS12/TI06/TO06	1 21	-		Multiple pin functions on the same	pin	
<ul> <li>Q. Timer Array Unit</li> <li>Q. Real-Time Clock</li> <li>A/D Converter</li> <li>A/D Converter</li> <li>S000</li> <li>PIOR1</li> <li>Not assigned</li> <li>Not assign None</li> <li>S000</li> <li>PIOR1</li> <li>Not assigned</li> <li>Not assign None</li> <li>Not assign None</li> <li>S000</li> <li>PIOR1</li> <li>Not assigned</li> <li>Not assign None</li> <li>S000</li> <li>PIOR1</li> <li>Not assigned</li> <li>Not assign None</li> <li>Not assign None</li> <li>S000</li> <li>PIOR1</li> <li>Not assigned</li> <li>Not assign None</li> <li>Not assign None</li> <li>S000</li> <li>PIOR1</li> <li>Not assigned</li> <li>Not assign None</li> <li>S000</li> <li>PIOR1</li> <li>Not assign None</li> <li< td=""><td>Clock Generator</td><td></td><td>SCK00</td><td>PIOR1</td><td>Not assigned</td><td>/ Not</td><td>assigi Nor</td><td>ne</td><td></td><td></td><td></td></li<></ul>	Clock Generator		SCK00	PIOR1	Not assigned	/ Not	assigi Nor	ne			
<sup>0</sup> Real-Time Clock <sup>1</sup> Clock Output/Buzzer Output Controlle <sup>1</sup> Not assigned <sup>1</sup> Not assign None <sup>4</sup> Clock Output/Buzzer Output Controlle <sup>1</sup> Not assigned <sup>1</sup> Not assigned <sup>1</sup> Not assign None <sup>4</sup> Serial Array Unit <sup>1</sup> Sou <sup>1</sup> Not assigned <sup>1</sup> Not assigned <sup>1</sup> Not assign None <sup>4</sup> Serial Array Unit <sup>1</sup> Sou <sup>1</sup> Not assigned <sup>1</sup> Not assign None <sup>4</sup> Serial Array Unit <sup>1</sup> Not assigned <sup>1</sup> Not assigned <sup>1</sup> Not assign None <sup>4</sup> Serial Array Unit <sup>1</sup> Not assigned <sup>1</sup> Not assigned <sup>1</sup> Not assign None <sup>4</sup> Serial Interface UAA <sup>1</sup> Not assign None <sup>1</sup> Not assign None <sup>1</sup> Not assign None <sup>4</sup> Serial Interface IICA <sup>1</sup> Not assign None <sup>1</sup> Not assign None <sup>1</sup> Not assign None <sup>4</sup> Serial Interface IICA <sup>1</sup> Not assign None <sup>1</sup> Not assign None <sup>1</sup> Not	Timer Array Unit		SCL00	PIOR1	Not assigned	/ Not	assigi Nor	ne			
<ul> <li>Interrupt Function</li> <li>Interrupt Functi</li></ul>	@ Real-Time Clock		SDA00	PIOR1	/ Not assigned	/ Not	assigi Nor	ne			
Q. A/D Converter     • & SAU0     • • SAU0     • SAU01   • SAU02   • SAU03   • SAU10   • S	A Clock Output/Buzzer Output Controller		\$100	PIOR1	/ Not assigned	/ Not	assigi Nor	ne			
<ul> <li>Serial Array Unit</li> <li>SAU0</li> <li>SAU1</li> <li>SAU0</li> <li>SAU0</li> <li>SAU0</li> <li>SAU1</li> <li>SAU1</li></ul>	💁 A/D Converter		SO00	PIOR1	Not assigned	/ Not	assigi Nor	ne			
• © SAU00         • • • • • • • • • • • • • • • • • • •	🕆 📽 Serial Array Unit		O TxD0	PIOR1	P12/SO00/TxD0/TOOLTxD/TS13/TI05/TO05	/ 20	-		Multiple pin functions on the same	pin	
SAU00     SAU01     SAU02     SAU03     SAU03     SAU0     SAU03     SAU1     SAU10     SAU10     SAU10     SAU11     SAU10     SAU11     Serial Interface IICA     Serial Interface UARTA     Interrupt Function     Reset Function     Reset Function     Con-chip Debug     Serial Interrupt     Power Supply	~ o SAU0	7 -					-				
• SAU01         • SAU02           • SAU03         • • • • • • • • • • • • • • • • • • •	SAU00										
• SAU02         • SAU03         • SAU03         • SAU10         • SAU10         • SAU10         • SAU10         • SAU11         • SAU11         • SAU10         • SAU11         • SAU10         • SAU10 <t< td=""><td>SAU01</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>	SAU01										
• SAU03        • SAU1        • SAU1        • SAU1        • SAU10        • SAU11        • SAU11        • Saul Interface IICA        • Serial Interface UARTA        • Interrupt Function        • Key Interrupt        • Reset Function        • Capacitive Sensing Unit        • On-Chip Debug        • Power Supply	SAU02										
SAU1     SAU1       SAU10     SAU10       SAU10     SAU10       SAU10     SAU10       Sau11     Sau10       Sexial Interface IICA     Sau10       Interrupt Function     Sau10       Reset Function     Sau10       Reset Function     Sau10       Capacitive Sensing Unit     Sau10       On-Chip Debug     Sau10       Power Supply     Sau10	SAU03										
SAU10     SAU11       SAU11     Seial Interface IICA       Seial Interface UARTA     Seial Interface UARTA       Interrupt Function     Seial Interface UARTA       Reset Function     Seial Interface UARTA       Ocapacitive Sensing Unit     Seial Interface UARTA       On-Chip Debug     Seial Interface UARTA	✓ SAU1										
** SAU11            ** Serial Interface IICA            ** Serial Interface UARTA            ** Interrupt Function            ** Key Interrupt            ** Reset Function            ** Capacitive Sensing Unit            ** On-Chip Debug            ** Power Supply	SAU10										
	SAU11										
<ul> <li>* Serial Interface UARTA</li> <li>Ø Interrupt Function</li> <li>Key Interrupt</li> <li>Reset Function</li> <li>Ø Capacitive Sensing Unit</li> <li>Ø On-Chip Debug</li> <li>Ø Power Supply</li> </ul>	Serial Interface IICA										
Interrupt Function     Image: Comparison of Co	Serial Interface UARTA										
Wey Interrupt         Image: Comparison of Comparison	Interrupt Function										
œ Reset Function               œ Capacitive Sensing Unit                 œ On-Chip Debug               œ                 œ Power Supply               œ	Key Interrupt	_									
Capacitive Sensing Unit On-Chip Debug Power Supply	Reset Function										
On-Chip Debug Power Supply	Capacitive Sensing Unit										
Power Supply	3 On-Chip Debug										
	Power Supply										
▲ Voltage Regulator	🚣 Voltage Regulator										_

Figure 7-17. Assignment of Pins for UART Channel (UART0)

RxD0/TxD0 UART0 pin assignment error may occur depending on the tool version used, but the error should be ignored.

In this application, the program generated by the COM port debug function is written to RL78/G22 using CS+. The pins used to write the program (TOOLRxD/TOOLTxD) also function as UART0 RxD0/TxD0 pins, which may cause pin conflicts in the Smart Configurator. However, conflicts will not occur in actual use since the CS+ and standalone version of QE are not used at the same time.

When using CS+ (writing a program): operates as TOOLRxD/TOOLTxD pins

When using standalone version of QE: operates as RxD0/TxD0 pins

Description	Туре
<ul> <li>✓ 9 Pin (4 items)</li> </ul>	
e8 E04010003: Pin used by RxD0 in Config_UART0 conflicts with pin used by TOOLRxD in Pin Allocator, pin used by TOOLRxD in System.	Pin
8 E04010003: Pin used by TxD0 in Config_UART0 conflicts with pin used by TOOLTxD in System, pin used by TOOLTxD in Pin Allocator.	Pin
E05000010: Pin 20 cannot be used multiple times. Pin 20 is assigned to TxD0 and TOOLTxD.	Pin
E05000010: Pin 21 cannot be used multiple times. Pin 21 is assigned to RxD0 and TOOLRxD.	Pin

Figure 7-18. Pin Assignment Error for UART0



## Using QE (standalone ver.) to Develop Touch Applications for FPB board

#### 7.5 Setting Unused Pins to Low-level Output

It is recommended to set ports unused in the application to low-level output.

In designing your circuit, make sure to perform sufficient pin processing and satisfy electrical characteristic requirements.

Please see user's manual of your target board in order to confirm ports which you need to set to low-level output.

As example, this section explains how to set "PORT63" to low-level.

1. Select "Components" tab and click 🖆 icon. In the displayed dialog, select "Port" module and click "Finish".

# Ports	Code Generator	1.3.0	
---------	----------------	-------	--

Figure 7-19. "Ports" Module

2. Select "Port" module and check "PORT6".

Software component con	iguration	Generate Code Generate Report
Components 🔤 🖬 🗄 🗷	Configure	·
type filter text	Port selection PORT6	
<ul> <li>Startup</li> <li>Generic</li> </ul>	PORTO PORT1	
<ul> <li>in r_bsp</li> <li>in privers</li> </ul>	PORT2 PORT3	
✓ ≥ I/O port	PORT4 PORT5	
<ul> <li>✓ inition</li> <li>✓ inition</li> <li>✓ inition</li> </ul>	PORT6 PORT7	
Config_UART0     Middleware	PORT12 PORT13	
✓ 🍅 Generic	PORT14	
rm_touch	Port mode setting	
	Read Pmn register values     Read digital output level	
	(	

Figure 7-20. Setup of "Ports" Module



Using QE (standalone ver.) to Develop Touch Applications for FPB board

3. Click "PORT6" tab and set "P63" to output.

oftware component configu	ration	🗐 🚔 Generate Code Generate Report
Components 🖮 🖬 🖄 🖻 🕅	Configure     Port selection PORT6	í
ype filter text → 😓 Startup → 🗠 Generic → r_bsp	Apply to all Unused In Out	Output 1
<ul> <li></li></ul>	P60 ● Unused ◯ In ◯ Out	Output 1
Config_UART0     Middleware     Generic	P61 Unused In Out P62	Output 1
<ul> <li>e<sup>2</sup> r_ctsu</li> <li>e<sup>3</sup> rm_touch</li> </ul>	Unused O In O Out	Output 1
	OUnused OIn Out	Output 1
	<	>

Figure 7-21. Setting "P63" to Output



## 7.6 Generating Code

Perform generating code.

1. Select "r\_bsp" module and confirm that "Initialization of peripheral functions by Code Generator/Smart Configurator" is set to "Enable".

nponents in 🖬 🖱 🖻 🖲	Configure	
	Dente	14.5
se filter text	Property	Value
(h) Etachan	Conigurations     Exact in select	Enable (use PEP startup)
Startup	<ul> <li>Start up seed.</li> <li>Control of invalid memory access datastion</li> </ul>	Dirable
- Ceneric	Consider and thereby access detection     PAM grant conservation	Disabled
e Dubarr	<ul> <li>Guard of control sensitives of port function/GROPT)</li> </ul>	Disabled
e unvers	Grand of consistent of internet intercention Grand of consistent of internet intercention(SINT)	Disabled
Config PORT	<ul> <li>Guard of registers of interrupt inscription (interrupt and addention and RAM parity error detection function/GCSC)</li> <li>Big Guard of control parity is a clock control function voltage detector and RAM parity error detection function/GCSC)</li> </ul>	Disabled
<ul> <li>Communications</li> </ul>	Data flack access postso (VEE)     Data flack access postso (VEE)	Disables
Coofia LIARTO	Provide instructions considerations by Code Generator/Smart Configurator	Enable
Middleware	# ADI functions disable R EDS Start Carls & DS Stort Carls	Enable
x D Generic	# ADI (include disability Cost (Cost approximation)	Enable
e r etsu	API functions disable(R BSP SetClockSource)	Enable
a no touch	# API functions disable(R_RSP ChangeClockSetting)	Enable
C THIL BOACH	# Parameter check enable	Enable
	Setting for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOO7E mode	High-speed
	# Enable user warm start callback (PRE)	Unused
	User warm start callback function same (PRE)	my sw warmstart prec function
	# Enable user warm start callback (POST)	Unused
	<ul> <li>User warm start callback. function name (POST)</li> </ul>	my sw warmstart postc function
	Watchdog Timer refresh enable	Unused
	<ul> <li>Watchdog Timer initialize user function name</li> </ul>	my sw wdt refresh init function
	Watchdog Timer setting user function name	my_sw_wdt_refresh_setting_function

Figure 7-22. Setup of "r\_bsp"

2. Click 随 icon on Smart Configurator to perform generating code.

When setting of on-chip debugging or option byte is changed, "Confirm linker option change" dialog may be displayed. Confirm the changes and click "OK".

Confirm linker option change		- 0	×
Setting User option byte value Option byte values for OCD Set debug monitor area Control allocation to self RA	Old value - - No No	New value EF3AE8 84 Yes(Specify Yes(Error m	
	OK	Cancel	

Figure 7-23. Confirm Linker Option Change



Using QE (standalone ver.) to Develop Touch Applications for FPB board

#### 8. Setup of QE for Capacitive Touch

#### 8.1 Launching QE for Capacitive Touch

Launch standalone version QE for Capacitive Touch (QE).

- 1. Launch QE by "QE-CapTouch (install folder of QE) / eclipse / qe-captouch.exe".
- 2. Figure 8-1 shows the window of QE after launching.

A QE for Capacitive Touch					– 6 ×
File View Help Board Monitor	Main Status Chart	Status C	hart		Parameters 22 12 12 12 12 12 12 12 12
Enable Monitoring Monitoring: Disabled, Communication Status: Disconnect	1 Program	2. Configuration	2 Tuning	4. Coding and Manifester	
Touch VF:	1. Preparation Prepare a project that uses the touch interfaces.	2. Configuration Prepare a touch interface configuration.	<ol> <li>Luning QE will automatically perform tuning processing for each touch sensor.</li> </ol>	4. Coding and Monitoring Implement a program using the touch. Then, confirm a behavior of touch interfaces and make fine adjustments.	Touch I/F:
Board Monitor	Create Project     Consets Straft Doyled using Grant     Consets     Cons	Encourse a Configuration     Generate about interface     Configuration     Monocharacter     Mon	Encurrence of the second	Indexemptions       Processing         Indexemptions       Processing         Index of the spectra state       Processing         Index of the spectra state <t< td=""><td>Parameters</td></t<>	Parameters
<ul> <li>49149</li> <li>Multi Status Chart</li> <li>16383</li> </ul>	Turing Touch UF Configurations «Not Selected» Method Kind Name Touch Sensor	Peradic Capacitie Tou	ch I/F Configurati	ion	

Figure 8-1. QE Window after Launching

If the layout in full window collapses, set layout of Windows to 100% by Windows setting.



### 8.2 Preparation

Set items according to "Preparation" of Workflow Diagram at middle of QE window.

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.
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.         Image: Select a MCU Name Select a name of the target MCU.         Select a name of the target MCU.	Image: Select or create a touch interface configuration.         Select or create a touch interface configuration.         Image: Modify Configuration         Modify Configuration         Image: Modify Configuration <t< td=""><td><ul> <li><b>D</b> Exercise Program</li> <li><b>S</b> That debugging the target project in Utuning program written on the target board will run.</li> <li><b>D</b> Exercise Program written on the target board will run.</li> <li><b>D</b> Exercise Provide the target of target o</li></ul></td><td>To Implement Program         Implement a program using the touch interfaces and build the project.         To Launch Oebug         Launch debugging for your target project and execute the program.         To Connect UART         Enable a monitoring function via serial communication.         Baud rate       115200         Port       Auto         Connect         Descent         Show monitoring views and enable a monitoring function.         Interval       75 [ms]</td></t<>	<ul> <li><b>D</b> Exercise Program</li> <li><b>S</b> That debugging the target project in Utuning program written on the target board will run.</li> <li><b>D</b> Exercise Program written on the target board will run.</li> <li><b>D</b> Exercise Provide the target of target o</li></ul>	To Implement Program         Implement a program using the touch interfaces and build the project.         To Launch Oebug         Launch debugging for your target project and execute the program.         To Connect UART         Enable a monitoring function via serial communication.         Baud rate       115200         Port       Auto         Connect         Descent         Show monitoring views and enable a monitoring function.         Interval       75 [ms]

Figure 8-2. Workflow Diagram (Preparation)

- 1. Click "..." under "To Select a Project Folder" and select your project folder created by CS+.
- 2. Click "…" under "To Select a MCU Name" and select your using microcontroller.

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.
In Create Project         Create a target project using Smart         Configurator.         Also, Set touch sensors and add the         CTSU driver with SC tool.         In Select a Project Folder         Select a folder that contains the target project.         CWCS+_WorkspaceWCapac         To Select a MCU Name         Select a nume of the target MCU.	Io Prepare a Configuration Belect or create a touch interface configuration.	To Execute Program Start debugging the target project in IDE tool and execute the program. The board will run. To Start Tuning Follow instructions in the dialog. Start Tuning Date advanced tuning Dutput parameter files from a tuning	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.       To Connect UART Enable a monitoring function via serial communication.       Baud rate     115200       Port     Auto
R7F102GGE	To Implement Program beent 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 <u>To Enable Monitoring</u> Show monitoring views and enable a monitoring function. <u>1</u> Interval T5 [ms]

Figure 8-3. Preparation



Using QE	(standalone	ver.) to	Develop	Touch	Applications	for FPB I	board
----------	-------------	----------	---------	-------	--------------	-----------	-------

Product Name of Targ	et MCU	×
Family Name	RL78	~
Group Name	RL78/G22	~
Pin Number	48pin:G	*
ROM Size	64KB:E	~
Deadwet Manua	P7E103GGE	

Figure 8-4. To Select a MCU Name

If the following error is occurred by "To Select a MCU Name", the place of QE install folder may be incorrect. Stop QE, move the install folder to other place such as in the directory of "C:\Renesas" and launch QE.

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

Figure 8-5. Internal Error



## 8.3 Configuration

Set items according to "configuration" of Workflow Diagram.

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.
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         CWCS+_WorkspaceWCapac         To Select a MCU Name         Belect a name of the target MCU.	Io Prepare a Configuration Select or create a touch interface configuration. Modify Configuration Modify Configuration Io Output Files for Tuning process. Output Files	To Execute Program         Start debugging the target project in 10E tool and execute the program. The toring program wilten on the target board will run.         Descent Program         To Estart Tuning         Follow instructions in the dialog.         Start Tuning         Enable advanced tuning         Output Parameter files from a tuning result.	To Implement Program         Implement a program using the touch interfaces and build the project.         To Launch Debsug         Launch debugging for your target project and execute the program.         To Connect UART         Enable a monitoring function via serial communication.         Baud rate       115200         Port       Auto
R7F102GGE	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	Connect <u>To Enable Monitoring</u> Show monitoring views and enable a monitoring function. Interval 75 [ms]

Figure 8-6. Workflow Diagram (Configuration)

1. Click 🔽 icon 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 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.
In Create Project         Cireate 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.         CMCS+_WorkspaceWCapac         Image: Select a norm of the target MCU	To Propare a Configuration Select or create a touch interface configuration.	<b>De Execute Program</b> Start debugging the target project in IDE tool and execute the program. The turning program written on the target board will run. <b>De Start Turning</b> Start Start Turning         Start Turning         De the advanced turning <b>De Start Parameter Files</b> Output parameter files from a turning	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.       To Connect Unity       Enable a monitoring function via serial communication.       Baud rate     115200       Port     Auto
R7F102GGE	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 IOE tool.	Output Parameter Files Use an external trigger Use diagnostic code	Connect <u>To Enable Monitoring</u> Show monitoring views and enable a monitoring function. (Interval 75 [ms] views)

Figure 8-7. Create a New Configuration



2. "Create Configuration of Touch Interfaces" window appears and displays the area for setting touch interface.

Click "Button" in the "Touch I/F" panel on the right to enable the cursor for button placement, then click wherever you wish to place a button.

Set two buttons as shown below, and then press the "Esc" key to cancel the button positioning function. In the same manner, click "Slider (horizontal)" in the "Touch I/F" panel to use the cursor to place a slider; click anywhere within the area you wish to place the slider. To cancel the slider positioning function, press the "ESC" key.

File Name of Touch Uf:	Capacitive, Touch, project, Exemple Setup Configuration	import / Re-acia
Description		
		Touch UP R
Burnando		Capacitorice Type
		Self Gapacitance v
Buttonit		Domas Burnas
	<u></u>	Sider (horizontal)
		Sider (versical)
SuberCO		Wed
>>		Key pad
< <		10 Gesture (Al)
		Souchiesa button (Al)
		Truch ped
		Shinid Pm
		% Per
		Capacitance Sensor
		Correct Second
Setting		Disprime Fin
Song South 15 Set	ap Resistance Value Clase Assigned TSr	Remove Touch UF
		Configurations (Methods) 2

Figure 8-8. Adding Buttons and Slider

- 3. Double click the "Button00" created in the previous step and set as follows in "Setup Touch Interface" dialog.
  - Touch Sensor : TS24
  - Resistance[ $\Omega$ ] : 560

For the resistance value, please see user's manual or circuit diagram of the target board.

🔳 s	etup Touch Interface	2	×
	Button(self)		
	Name	Button00	
	Touch Sensor	Resistance[ohm]	
TS24	TS24	560 560	
E	OK	Cancel <u>H</u> elp	

Figure 8-9. Setup of Touch Interface (Button)



- 4. Set "Button01" as follows.
  - Touch Sensor : TS23
  - Resistance[ $\Omega$ ] : 560
- 5. Set "Slider00" as follows.
  - Touch Sensor : TS20
    - : TS21
      - : TS22
  - Resistance[ $\Omega$ ] : 560

🔳 S	etup Touch Interface		×
222	Slider		
	Name Number of Touch Sensor	Slider00 3	~
TS20         	Touch Sensor TS20 TS21	Resistance[ohm] 560 ← 27 560 560	
	TS22	560	
	Reverse		
	ОК	Cancel <u>H</u> el	p

Figure 8-10. Setup of Touch Interface (Slider)

6. After setting touch interface, the area should look as follows. To complete the settings, click "Create".

File Name of Touch Uf: Capacitive, Touch, project, frample Setup Configuration	import / Re-adit
Decipion	
Butters30	Touch UP A
	Capacitance Type
	iser Lapacheria v
h-moli	Button
-	Sider (horizontal)
750	Sider (verScal)
Sider00	Wheel
10 10 10 10 10 10 10 10 10 10 10 10 10 1	Key pad
	ID-Gesture 180
	Touchkess button (All
	Buch par
	Shield Pin
	TC Pm
	Capacitance Sensor
	Current Service
50510	Depters for
Simp Touch 17 Setup Resistance Value Clear Assigned TSs	Remove Touch UV
	Conference Methods II

Figure 8-11. Touch Interface Configuration after Setting



7. "Touch I/F Configuration" is displayed in "Tuning" panel.

uning ouch I/F (	Configuration:	Capacitive_T	ouch_project_Exan	nple				
Method	Kind	Name	Touch Sensor	Parasitic Capacitance[pF]	Sensor Drive Pulse Frequency[MHz]	Threshold	Scan Time[ms]	Overflow
config01	Button(self)	Button00	TS24	11.118	2.0	2281	0.576	None
config01	Button(self)	Button01	T\$23	12.306	2.0	2162	0.576	None
config01	Slider	Slider00	TS20, TS21, TS22		*	2293		None
config01	Slider TS	(Slider00)	T\$20	11.778	2.0		0.576	
config01	Slider TS	(Slider00)	TS21	13.007	2.0		0.576	•
confie01	Slider TS	(Slider00)	TS22	13.236	2.0		0.576	22



 Click "Output Files" and select folder for the output files. Create new folder "qe\_gen" under "Capacitive\_Touch\_Project\_Example/src" and output them to the folder. The following is the configuration of the folder including output files.

The felletting is the semigaration of the fela	ter melaamg eatpat mee.
Capacitive_Touch_Project_Example	← CS+ Project Folder

- |- src |- src\_gen |- qe\_gen |- qe\_touch\_config.c |- qe\_touch\_config.h |- qe\_touch\_define.h |- qe\_touch\_sample.c ← New Folder ← Output File ← Output File
- 9. After selecting folder for output files, the following dialog appears. Set clock and click "OK".

equency of Peripheral Module Clock (PCLKB or PCLKL)	×
eripheral module clock frequency (PCLKB or PCLKL)[MHz]	32 🚺 32
p	

Figure 8-13. Setting Frequency of Peripheral Module Clock



In the following dialog, set power supply voltage and click "OK".
 Please confirm the electric characteristics of the microcontroller you are using.
 When using the RL78/G22, set the power supply voltage of VDD.



Figure 8-14. Setting Power Supply Voltage of MCU

11. Next, "QE for Capacitive Touch" dialog appears. Follow the instructions of the dialog. Also the contents of the dialog is displayed in "Console" panel at lower of QE window.



Figure 8-15. QE for Capacitive Touch Dialog



Figure 8-16. Console



A. Set compiler option.

Select "CC-RL (Build Tool)" in Project Tree of CS+.

Select "Macro definition" of "Frequency Used Options(for Compile)" in property and click "…" at right side.

Control for the hole     Control     Contr	Turkey Turkey Topik OwenDest	• 3
transfer Protects     voir proceedings     voir to all black     voir proceedings     voir proceedi	CO-RL Property	
Notice     Notice     Notice       Constrained     Notice <th>ratect Example Project: V Build Mode</th> <th></th>	ratect Example Project: V Build Mode	
Checker property value for all and/or and once     No       Name     R. Th-Git contribution       Name     R. Th-Git contribution       Name     Encode Model State Mode       Name     Encode Model State Mode       Name     Encode Model State Mode       Name     Encode Model State       Name     Encode Name       Name     State Name	Microcov (Solida) Datal mode	Livia Pouro
Op/Sec         CPU         Multi-classifie           Nalyse Tool         V         Original Tail Type and Path         Encode ModerSignal Moder Fale           Exclusion Interaction Interaction Interaction         Nal         Nal         Nal           Sec         Original Tail Type and Path         Secure ModerSignal Moder Fale         Secure ModerSignal Moder Fale           Secure Interaction Interaction Interaction         Nal         Nal         Nal         Secure ModerSignal Moder Fale           Secure Interaction	Change property value for all build modes at once	No
May be Tools     South CVD Gate     RLP-3 devicepable       ed files     Output file type     Except ModelSale Model File       ed files     Output file type     Except ModelSale Model File       or     Vertregath Mad Spinsber Internation     No       or     Additional incluing prints     Additional incluing paths[14]       0     Sciete incluing prints     Additional incluing paths[16]       0     Mass definite     Sciete incluing paths[16]       0     Mass definite<	Defaul V CPU	
State State         • Output File Type and Public State           Ideal File         Output File Type and Public State           Ideal Output File         State Computer           Ideal Output File         Poters the default again against muture and the public State of the Output File           Ideal Output File         Additional muture public State           Ideal Output File         Mains state           IdEal State         State	Specify CPU core	PL78-G3 conf-courses
Output file type     Exclute Model/Each       of Note     Output file type       of     Distribution       of     Potom the objective of a structure internation       0     South include poting       0     Masson structure [0]       0     Masson structure [0]       0     Masson structur	<ul> <li>Output File Type and Path</li> </ul>	
1985     Output creas information     No       1985     Output creas information     No       1986     International table     No       1987     Trequarity listed (bisculture)     Poteon the default aptingstandho option specified)       1987     Output defaultation     Additional incluing pathic       199     Output defaultation     Additional incluing pathic       199     Output defaultation     Additional incluing pathic       199     Output defaultation     Output defaultation       199     Output defaultation     Additional incluing pathic       199     Output defaultation     Additional incluing pathic       199     Output defaultation     Additional incluing pathic       199     Additional incluing pathic     Additional incluing pathic       199     Output defaultation     Output defaultation       199     Terreparatify Undefaultation     Output defaultation       199     Terreparatify Undefaultation     Output defaultation       199     Terreparatify Undefaultation     Stational incluing pathic       199     Terreparatify Undefaultation     Stational incluing pathic       199     Terreparatify Undefaultation     Stational incluing pathic       199     Terreparatify Undefaultation     No       199     Terreparatify Undefaultation	Output file type	Execute Hodule(Load Hodule File)
Ibe module like output table     Versume     Vers	les Output croce reference into reation	Sin
Not  Version of the default set of the default	Intermediate file output tokler	#Build/lightsetRane#
Bits     Potom fe default spinication/option specified)       Additional include paths     Additional include paths       Bits     Spitten include paths     Exitem include paths       V     Fragmently Used Optionality Assemble)     More include paths       V     Fragmently Used Optionality Intell Optionality Intell Spitter include paths     [2]       V     Fragmently Used Optionality Intell Spitter include paths     [2]       V     Fragmently Used Optionality Intell Spitter include paths     [3]       V     Fragmently Used Optionality Intell Spitter include paths     [4]       V     Vergenetly Used Optionality Intell Spitter include paths     [4]       V     Vergenetly Used Optionality Intell Spitter include paths     [5]       V     Vergenetly Used Optionality Intell Spitter include paths     [6]       V     Vergenetly Used Optionality Intell Spitter include paths     [6]       V     Vergenetly Used Optionality Intell Spitter include paths     [6]       V     Vergenetly Used Optionality Intell Spitter include paths     [6]       V     Vergenetly Used Optionality Intell Spitter include paths     [6]	<ul> <li>Frequently thad Optimuther Compile)</li> </ul>	
Additional inclusion parties     Additional inclusion parties     Additional inclusion parties     Support include parties     Additional inclusion     Additional     Additional inclusion     Adi	Level of optimization	Porto-in the default aptimization/No option specified?
Soften include pathe     Soften include	Additional include method	Additional exclusion patho [14]
2      2	<ol> <li>System include paths</li> </ol>	Eviters include participal
Property Used (pisodle Assemble)     Additional include paths (2)     Additional include paths (2)     Additional include paths (2)     Southen include paths (2)     Southen include paths (2)     Very and the mean     Coupt field     Very and advected advecte	March Infectory	Marris detectory[1]
Additional incluios pathe     Additional incluios pathe [2]       Lipited     5 System blaids pathe     System incluios pathe [2]       System blaids pathe     System incluios pathe [3]       System pathe pathe     Maxes antinices [3]       System pathe file	Frequently (ked Optimullar Assemble)	
Justici Splatning Markagenia     Splatning Markagenia       Splatning Markagenia     Splatning Markagenia       Markagenia     Markagenia       Opport Inde	Additional projets safes	Additional methods with [2]
	K4 Suther include parts	Sectors include and a 10
Index Section     Version Section     Version Section     Version     Ver	a cysers include point	Arrest in the parts of
	Francisco detection	serve measure (d)
Corper, file ranke     Corper, for an end     Corper, for an en	<ul> <li>Frequency there optimizers two</li> </ul>	1
Organ Mode         ModeWiseName           Organ Mode         Bit section           Organ Mile revise         Bit section           Organ Mile revise         Bit section           Organ Mile revise         ModeWiseName	3 Using Kit wind.	(Ling Howsell)
Cupy the name Upe standardsheartstall Branes Upe standardstall	Ourput roldar	Now KITIGO Normal
Use reform Streams         YestLikerary tar C910           Use reform Streams         Yes           V Frequently Used Quirentile Hen Output         Yes           Orgot Ine Taile         Yes           Output Ine Taile         Yes           Output Ine Taile         Yes           Output Ine Taile         Where Grant Streams           Output Ine Taile         Where Grant Streams           Output Ine Taile         Stream Output Itele           Stream Output Itele         Stream Output Itele           Output Ine Taile         Stream Output Itele           Stream Output Itele         Stream Output Itele </td <td>Output his name</td> <td>Bringson States Aug</td>	Output his name	Bringson States Aug
Use nabre ibrains	Over standard/leathematical libraries	Yes(Likeary for C910
V Frequently Used QuintedAr Hes Output     Output here this     Vec.	Use runtime libraries	Yes
Organ ban risk         Yes           Her file format         Mutucki S-record file(-POrm-Stype)           Organ risk         Mutucki S-record file(-POrm-Stype)           > Device output file(POrm-Stype)         Owners output file(PO           > Marring Message         Image Message           > Bunkel Mertond         Owners           > Merking Message         Image Mertond	Frequently Used Optionality Hex Output)	
Here for format         Miturals Smooth Bits-POn=Stypel           Curpor Inform         Situat/StochandS           Curpor Curport         Situat/StochandS           Director curport         Situat/StochandS           Director curport         Onicon subul file[0]           Image StochandS         Onicon subul file[0]           Warring Bitstam         Onicon subul file[0]           Director curport         Site StochandS           Director curport         Onicon subul file[0]           Director curport         Site StochandS           Director curport         Director curport           Director curport         Site StochandS	Output here file	Yes
Output noise         Stilluittinoilousetti           Output noise         Strippit charactionettinoilousetti           > Draston output line         Draston output line[0]           > Erran Output         Branch Output           > Branch Output         Draston output line[0]           > Warning Managan         Branch Output           > Draston output line[0]         Branch Output           > Branch Output         Branch Output           > Draston output line[0]         Branch Output           > Draston output line[0]         Branch Output           > Draston output line[0]         Branch Output           > Marine Status         Branch Output           > Version Robert         Branch Output	Hes. file format	Muturala Smooth Ref. FOrestStype)
Cupie file rates Miniperfilement Director output file Director output file Miniming Montage Device Device Device Version Select Director	Output folder	Silu Minute Narwall
Orwann organ blan     Orwann organ blan     Orwann organ blan     Working Mennam     Morring Mennam     Orwann     O	Output file name	30Project/Name/Kanot
Erver Output     Working Message     Device     Isuald Method     Vervis In Debet     Moles     Moles     Molese	> Orvision output file	Division subsit title]
Warning Hessage     Working Hessage     Device     Build Herbord     Build Herbord     Morise Solution     Working Solution	> Error Output	
Device     Device     Device     Device     Device     Device     Notion     Notion     Notion	) Warning Meanage	
Social Method     Montain Select     Montain Select     Montain     Montain     Montain	> Device	
Version Select     Notes     Notes	) Duild Nethod	
> Notes	1 Version Select	
1 Others	Notes	
	Other	
	Processing the many party is the defined in the house of Time	an analytic for the West was an an an in the West (dated of the second or the second of the second of the second or the dated of the
Marra de fedias	spectres the macro have to be defeed in the tornal of "Ina-	re-name/relativest valuer, with one makes partition, the "Highrand valuer" partical be onlyted, and in this case, "IT is used as the defined value.
Mapped of Findum. Specifies the many name to be defined in the bornet of "(suppo name)-defined value)", with one many name per line. The "-defined value)" part can be orbited, and in this case, "I" is used as the defined value.	the option conseptions to the "D option of the con-constants	
Macro-defection Specifies for many name to be defined in the terms of "(many name)-(defined value)", with one many name per line. The "-(defined value)" part can be unitted, and in this case, "1" is used as the defined value. The option compounds in the -D option of the cast contented.		
Marredefinition Specifics the mars made to be defined in the tornal of "(suppowers)-(defined value)", with one marso name per line. The "+(defined value)" part can be onlined, and in this case, "I" is used as the defined value. This option corresponds to the -D option of the cast command.	Company Options / Compile Options / Assemble Options	

Figure 8-17. Selecting Macro Definition

Add "QE\_TOUCH\_CONFIGURATION" to text field in "Text Edit" dialog and click "OK".

Text Edit		×
Igat	-	
QE_TOUCH_CONFIGURATION	-	^
c		>
	OK Cancel	Help

Figure 8-18. Edit Macro Definition



Next, select "Additional include path" of "Frequency Used Options (for Compile)" and click "…" at right side.

Add "src¥qe\_gen" to path field in "Path Edit" dialog and click "OK".



Figure 8-19. Additional Include Paths

Path Edit	
Path(One path per one line):	
sroVsmc.genWr.bspWboardVg sroVsmc.genWr.bsp sroVsmc.genWrm.touch sroVsmc.genWcontig.UART0 sroVsmc.genWContig.PORT sroVsmc.genWContig.PORT sroVsg.gen DefaultBuild	seneric_f178_g22 0
Browse_	
Permit non-existent path	
Include subfolders autom	atically
Placeholder:	
Placeholder	Value
ActiveProjectDir ActiveProjectMicomName ActiveProjectName BuildModeName MainProjectDir	CiVCS*_WorkspaceVCapacitive_Touch_pro R7F102GGExFB Capacitive_Touch_project_Example DefaultBuild CiVCS*_WorkspaceVCapacitive_Touch_pro
1	>

Figure 8-20. Path Edit

Next, click "Use Standard / Math Library" under "Frequently Used Options (links)".

Click  $\checkmark$  (drop-down arrow) on the right side, and select "Yes (Library for C99)".

1	<ul> <li>Frequently Used Options(for Link)</li> </ul>		
	> Using libraries	Using libraries[0]	
11	Output folder	%BuildModeName%	
	Output file name	%ProjectName%abs	
	Use standard/mathematical libraries	Yes(Library for C99)	
11	Use runtime libraries	Yes	
	> Frequently Used Options(for Hex Output)		
	> Error Output		
	> Warning Message		~
LS	Use standard/mathematical libraries Select whether to use the standard/mathematical libraries which the compiler provides.		
$\setminus$	Common Options / Ille Options / Assemble Options / SMS Assemble Options	Link Options / Hex Output Options / I/O Header File Generation Options /	•

Figure 8-21. Selecting / Standard Math Library



Next, double-click "Source" in "Compile Options", and then click "Language of the C source file".

Click  $\checkmark$  (drop-down arrow) on the right and select "C99 (-lang=c99)".

Y	Source	1000				^
	Language of the C source file	C99(-la	ng=c99)			~
	Language of the C++ source file	C++14(-	ang=cpp14)			
>	Quality Improvement					
>	Memory Model					
>	CLanguage					
>	Character Encoding					
>	Output Code					
>	Output File					
3	Assemble List					
3	MISRA-C Rule Check					
5	Message					
>	Others					~
						-
La	nguage of the C source file					
Sel	ects the language of the C source file.					
111	s option corresponds to the stang option of the con command.					
	ammon Options Compile Options	SMS Assemble Options / Link Or	tions / Hex Output Options	I/O Header File Generation Options	/	-
1.	and a second sec			generation options		

Figure 8-22. Selecting C Language Standard

Next, double-click "Device" in the "Link Options" tab and enter "84" as the "On-Chip Debug Option Byte Control Value"

Similarly, for the "Set debug monitor area", select "Yes (range setting) (-

DEBUG\_MONITOR=<address range>)".

For the "User Option Byte Value, "enter "EFFFE8".

For the option byte value setting, refer to the user's manual of the microcontroller you are using.

	Device		^
	Set enable/disable on-chip debug by link option	Yes(-QCDBG)	~
	Option byte values for OCD	<b>HERE 84</b>	
	Set debug monitor area	Yes(Specify address range)(-DEBUG_MONITOR= <address range="">)</address>	
	Range of debug monitor area	FE00-FFFF	
	Set user option byte	Yes(-USER_OPT_BYTE)	
	User option byte value	EF3AE8	
	Output Code		
È.	List		
,	Variables/functions information		
5	Section		
	Varity		
>	VCINY		
2	Message		

Figure 8-23. Selecting Option Byte



If you use the CC-RL free evaluation edition V1.12.00 or later, select "debug precedence(-onothing)" as the optimization levels.

- Double-click "Optimization" of "Compile Options", and then click "Level of optimization".
- Click <sup>✓</sup> (drop-down arrow) on the right, and select "debug precedence(-onothing)".

Remark This optimization setting need only for tuning of touch sensors. After tuning is complete, you can use this application with any optimization settings.

Level Optin Prepa Source	el of optimization imization(Details) process roe	Debug precedence(-	Onothing)	~
Optin Prepa Source	imization(Details) process rce			
Prep	process			
Sour	rce			
Qual	ality Improvement			
Mem	nory Model			
CLar	anguage			
Char	racter Encoding			
Outp	put Code			
Outp	put File			

Figure 8-24. Selecting level of optimization



- B. Perform coding of touch main function in main() function.
  - If "qe\_gen" folder is not in project tree of CS+, add "qe\_gen" folder to project tree from Windows Drag and drop from Explorer.

RL78 COM Port (Debug Tool)     Q Program Analyzer (Analyze Tool)     File     File	main.c     Drag and Drop
--	--------------------------

Figure 8-25. Adding "qe\_gen" Folder to Project Tree

Call "qe\_touch\_main()" function in main() function. Add the following code to main.c.

- extern void qe\_touch\_main(void);
- qe\_touch\_main();

<pre>30 void main(void); 31 extern void qe_touch_main(void); 32 void main(void) 33 void main(void) 34 □{ 35 qe_touch_main(); 36 }</pre>
---

Figure 8-26. main.c



12.Add function for serial communication to "Config\_UART0 user.c".

Add the following code.

- 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 */ extern void touch_uart_callback(uint16_t event); /* End user code. Do not edit comment generated here */
74 75 76 77 78 79	<pre>static void r_Config_UART0_callback_sendend(void)  {     /* Start user code for r_Config_UART0_callback_sendend. Do not edit comment generated here */     touch_uart_callback(0);     /* End user code. Do not edit comment generated here */ }</pre>
87 88 89 90 91 92	<pre>static void r_Config_UART0_callback_receiveend(void)  {     /* Start user code for r_Config_UART0_callback_receiveend. Do not edit comment generated here */     touch_uart_callback(1);     /* End user code. Do not edit comment generated here */ }</pre>

Figure 8-27. Config\_URAT0\_user.c

13. Build the project by CS+. Click icon on CS+ and start build. Confirm that build finished without any errors or warning.

If the following warning (W0511187) is occurred when build the project, change the optimization levels to "debug precedence(-onothing)" as shown in Figure 8-24 on page 36. And then, rebuild the project.



Figure 8-28. Warning (W0511187)



## 8.4 Tuning

Set according to "Tuning" of Workflow Diagram.

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.
To Create Project         Create a target project using Smart         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.         CtYCS+_Workspace¥Capac         To Select a MCU Name         Select a name of the target MCU.         R7F102GGE	In Prepare a Configuration         Select or create a touch interface configuration.         Lapacitive_Touch_project_Exa ↓         Modify Configuration         Modify Configuration         Output Files for Tuning process.         Output Files required for a tuning process.         Output Files         Implement Program         Implement a process to call the main toop for touch in the main() function.         Incolution in the main() function.         Dated Project         Build the target project using IDE tool.	<b>De Execute Program</b> Start debugging the target project in IDE tool and execute the program. The touring program written on the target board will run. <b>De Start Puning</b> Follow instructions in the dialog. <b>Start Tuning</b> Chable advanced tuning <b>Dutput Parameter Files</b> 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.         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 8-29. Workflow Diagram (Tuning)

1. Right-click "Debug Tool" in "Project Tree" of CS+, and click "Using Debug Tool". Select RL78 COM Port" as the debug tool you intend to use.



Figure 8-30. Selecting Debug Tool



2. Set the "Communication port" in the "Debug Tool" properties. This application example uses COM3 as the communication port.

<ul> <li>Internal ROM/RAM</li> </ul>			
Size of internal ROM[KByte	es]	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	
<ul> <li>Connection with Target Be</li> </ul>	oard		
Communication port		COM3	~
Reset control pin		DTR	
✓ Flash			
Security ID		HEX 0000000000000000000	
Permit flash programming		Yes	
Use wide voltage mode		Yes	
Erase flash ROM when star	rting	No	
Using the flash self program	mming	No	
Communication port			
Select the communication port.			

Figure 8-31. Property of Debug Tool

Confirm the communication port setting in the Device Manager.

File       Action       View       Help         Image: Constraint of the second	🚦 Device Manager	<u></u>	×
<ul> <li>WAN Miniport (SSTP)</li> <li>Ports (COM &amp; LPT)</li> <li>Intel(R) Active Management Technology - SOL (COM6)</li> <li>USB Serial Port (COM3)</li> <li>Print queues</li> <li>Print queues</li> <li>Processors</li> <li>Security devices</li> <li>Sensors</li> </ul>	File Action View Help		_
<ul> <li>WAN Miniport (SSTP)</li> <li>Ports (COM &amp; LPT)</li> <li>Intel(R) Active Management Technology - SOL (COM6)</li> <li>USB Serial Port (COM3)</li> <li>Print queues</li> <li>Processors</li> <li>If Security devices</li> <li>Servirty devices</li> </ul>	+ + II II II		
<ul> <li>Ports (COM &amp; LPT)</li> <li>Intel(R) Active Management Technology - SOL (COM6)</li> <li>USB Serial Port (COM3)</li> <li>Print queues</li> <li>Processors</li> <li>Y Security devices</li> <li>Serial Serial Serial</li></ul>	💭 WAN Miniport (SSTP)		^
Intel(R) Active Management Technology - SOL (COM6) USB Serial Port (COM3)  Print queues  Processors  Processors  Y Security devices  Y	Ports (COM & LPT)		
USB Serial Port (COM3)  The Print queues  Processors  Processors  Security devices  Sensors  V	Intel(R) Active Management Technology - SOL (COM6)		
Print queues     Processors	💭 USB Serial Port (COM3)		
Processors     P	> 🚍 Print queues		
Pecurity devices     Final Sensors	Processors		
> En Sensors *	If Security devices		
	> 🔄 Sensors		~

Figure 8-32. Device Manager

3. Confirm that the QE serial connection switching jumper (J16) on the target board is shorted and that the PC and target board are connected with a USB cable, then click the CS+ kine icon to build and write the program. When the download is complete after writing the program, click the stop the program, and, finally, click the kine icon to disconnect.

After disconnection, remove the USB cable connecting the PC and target board, and open the QE serial connection switching jumper (J16).

Next, reconnect the USB cable between the PC and target board so that you can connect QE. At this time, the target board will be in standby state for connection with QE, while it runs the written program.

For details regarding the QE serial connection switching jumper (J16), refer to the target board user's manual. Always use a USB cable that supports data transfer.



4. On QE, set "Baud rate" of "To Connect UART" to the value which is set in chapter 7.4.



Figure 8-33. Setting Baud Rate

5. Click "Start Tuning", and start tuning.

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.
To Create Project         Create a target project using Smart         Also, Set touch sensors and add the         CTSU driver with SC tool.         To Select a Project Folder         Select a folder that contains the target         CréCS+_Workspace¥Capac         Ct         Select a name of the target MCU.         R7F102GGE	Image: Design of the second secon	Image: Second	Implement Program         Implement a program using the touch interfaces and build the project.         Data Debugg         Launch debugging for your target project and execute the program.         D Connect UART         Jable a monitoring function via serial communication.         Baud rate       153600         Port       Auto         Connect         Show monitoring function.         Show monitoring function.         Interval         75 [ms]

Figure 8-34. Tuning



Using QE (standalone ver.) to Develop Touch Applications for FPB board

6. Set baud rate and click "Connect" on the displayed dialog.

Conne	ect to COM port (Serial Communication)	×
Baud rate COM port	153600 <b>A</b> uto	>
Conn	nect Cancel <u>H</u> elp	

Figure 8-35. Setting Baud Rate

7. In the next dialog, set clock and click "OK".



Figure 8-36. Setting Frequency of Peripheral Module Clock

In the next dialog, set power supply voltage and click "OK".
 Please confirm electric characteristic of your using microcontroller.
 When using the RL78/G22, set the power supply voltage of VDD.

Power Supply Voltage of MCU (EVDD)		×
MCU power supply voltage (EVDD) [V]	5.0V: (4.5V ≅ EVDD ≅ 5.5V)	~
ОК	Cancel <u>H</u> elp	

Figure 8-37. Setting Power Supply Voltage of MCU

9. Tuning start. Confirm the contents of "Automatic Tuning Processing" dialog which shows guidelines for tuning process and follow the instructions of the dialog.

Automatic Tuning Processing	X
1/8: QE is beginning the tuning process. During the tuning process, please do not touch the sense the QE Tuning Program.	ors on the target board until instructed by
	Cancel <u>H</u> elp

Figure 8-38. Automatic Tuning Processing Dialog



After some steps, the following dialog appears.

This step is for measuring touch sensitivity. Touch with normal pressure the touch sensor indicated in the dialog. While touching the touch sensor, the bar graph will extend to the right and touch counts will increase.

While touching, press any key on the PC keyboard to confirm the sensitivity measurement.



Figure 8-39. Measuring Touch Sensitivity (Button)

- 10. The touch sensitivity of the other touch sensor can be measured in the same manner
- 11. Touch sensitivity can also measured for the slider touch sensor. After tracing the slider on the target board 3 or 4 times up and down or left and right with normal pressure, keep your finger on the slider and press any key on the PC keyboard to confirm the measurement.



Figure 8-40. Measuring Touch Sensitivity (Slider)



- 12. The threshold can be confirmed in the following dialog, which appears when tuning is completed. This threshold is used to determine touch events in the middleware.
  - After confirming the threshold, click "Continue the Tuning Process". This completes automatic tuning.

sensors can b application ne	c tuning e retriec otes for	) proce d. If the Capaci	ss is nov ere are c tive Toue	v complete. If ontinued ove ch for guidan	rflows or ce.	v or wari warning	ning/errors a g/errors, plea	re indicated, tho se consult the Re	se enesas
Select the target	Method	Kind	Name	Touch Sensor	Threshold	Overflow	Warning / Error		
	config01	Button	Button00	TS24	1691				
	config01	Button	Button01	TS23	1837				
	config01	Slider	Slider00	TS20, TS21, TS22	1898				
Retry Continue	the Tuning	Process	-					Cancel <u>H</u> e	elp

Figure 8-41. Threshold of Touch Sensor

13. Click "Output Parameter Files" and output parameter files including result of tuning. Choose "qe\_gen" folder created at chapter 8.3 as the folder for output files and overwrite the files.

The output files are same as the following files that is outputted at "Output files" of chapter 8.3.

- qe touch config.c ← Output File
- |- qe touch config.h ← Output File
- |- ge touch define.h ← Output File ← Output File - qe touch sample.c Workflow Diagram 1. Preparation 2. Configuration 3. Tuning 4. Coding and Monitoring 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. Prepare a project that uses the touch interfaces. Prepare a touch interface configuration. To Implement Program Implement a program using the touch interfaces and build the project. To Create Project Create a target project using Smart To Prepare a Configuration 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. Select or create a touch inte configuration. Configurator Also, Set touch sensors and add the CTSU driver with SC tool. To Launch Debug Launch debugging for your target project and execute the program. Capacitive\_Touch\_project\_Exa 🗸 To Start Tuning Follow instructions in the dialog. To Select a Project Folder Select a folder that contains the target Modify Configuration To Connect UART Enable a monitoring function via serial project. To Output Files for Tuning Output files required for a tuning Start Tuning communication. C:¥CS+\_Workspace¥Capac ... Enable advanced tuning process. 153600 Baud rate To Output Parameter Files Output parameter files from a tuning result. To Select a MCU Name Select a name of the target MCU. **Output Files** Auto Port ~ To Implement Program Implement a process to call the main loop for touch in the main() function. Connect R7F102GGE Output Parameter Files .... Use an external trigger To Enable Monitoring To Build Project Build the target project using IDE tool. nd enable a Show monitoring view monitoring function. Use diagnostic code Interval 75 [ms] v

Figure 8-42 To Output Parameter Files



## 8.5 Coding and Monitoring

## 8.5.1 Monitoring

Set according to "Coding and Monitoring" of Workflow Diagram.

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.
In Create Project         Greate a target project using Smart Configurator.         Also. Set touch sensors and add the CTSU driver with SC tool.         Image: Set of the Set o	In Prepare a Configuration         Select or create a touch interface configuration.         Capacitive_Touch_project_Exa ↓         Modify Configuration         Modify Configuration         Output Files for Juning process.         Dutput files required for a tuning process.         Dutput Files         Implement Program         Implement a process to call the main loop for touch in the main() function.         Inglement a project using IDE tool.	To Execute Program.         Statt debugging the target project in IDE tool and execute the program. The tuning program written on the target board will run.         Data Control C	<b>D</b> Implement Program.         Implement a program using the touch interfaces and build the project. <b>D</b> Launch Debug         Launch debugging for your target project and execute the program. <b>D</b> Connect UARI         Baud rate       153600         Port       Auto         Connect <b>D</b> Connect </td

Figure 8-43. Workflow Diagram (Coding and Monitoring)

1. Remove the USB cable connecting the PC and target board, and short the QE serial switching jumper (J16). Next, reconnect the USB cable between the PC and target board so that you can connect CS+.

Click the CS+ kicon to build and write the program. When the download is complete after writing the program, click the icon to stop the program, and, finally, click the kicon to disconnect.

After disconnection, remove the USB cable connecting the PC and target board, and open the QE serial connection switching jumper (J16).

Next, reconnect the USB cable between the PC and target board so that you can connect QE. At this time, the target board will be in standby state for connection with QE, while it runs the written program.



2. Click "Connect". "Connect" changes to "Disconnect".



Figure 8-44. To Connect UART



3. Click "Enable Monitoring" of "Board Monitor" panel at top left of QE window. "Monitoring: Disabled" changes to "Monitoring: Enabled".



Figure 8-45. Enable Monitoring

4. While touching the touch sensor, the finger icon shows the state of touch sensor.

Board Monitorix ×     Image: Ima	
Button00 Button01 Slider00	~

Figure 8-46. Display of the Condition while Touching



- 5. Represent a graph of the touch counts
  - A. Click "Status Chart" tab at the panel including "Workflow Diagram".
  - B. Click <sup>▶</sup> icon of "Touch I/F" at "Status Chart" window and select touch interface.
  - C. The Graph shows real-time value of the touch sensor. When touching the touch sensor, touch counts change on the graph.

The green line shows the threshold, which "rm\_touch" middleware uses to determine whether the touch sensor is actuated/touched.

The red belt at the bottom of the graph shows that touch counts is over the threshold and the touch sensor is being touched.



Figure 8-47. Graphical Representation of Touch Counts (Button)



Figure 8-48. Graphical Representation of Touch Counts (Slider)



6. As necessary, measure standard deviation.

A. Click "Start Data Collection" without touching. Don't touch the touch sensor while measuring the value in the state of touch-off.

The green bar shows the rate of the data collection. When the green bar goes all the way to the right, the data collection for touch-off state is done.



Figure 8-49. Data Collection of Touch-off State

B. Click "Stop Data Collection", when the green bar goes all the way to the right.

Count Value:	15544	Reference Value:	15508	Threshold:	2281	Difference:	36
Stop Data Col	lection						
Noise [NT]:	19	Average [NT]:	15508	Minimum:	15436	Maximum:	15560
Noise [T]:		Average [T]:		Signal:		SNR:	

Figure 8-50. Stop Data Collection

- C. Next, in the same way, start data collection in the state of touch-on.
- D. After finishing data collection, SNR value appears.



Figure 8-51. SNR value



Represent a graph of the touch counts for multiple touch sensors.
 Select the touch sensors in "Multi Status Chart" panel at the lower-left of QE window.



Figure 8-52. Multi Status Chart

8. As necessary, adjust parameters manually. Adjust parameters in "Parameters" panel at the right side of QE.

Item	Value			Touch Parameters
Drift Correction Interval	255			
Long Touch Cancel Cycle	0			
Positive Noise Filter Cycle	3			
Negative Noise Filter Cycle	3			
Moving Average Filter Depth	4			
Touch Threshold	2281			
Hysteresis	114			
Set a drift correction interval. Drift Correction is a function to a environment. Input a value between 0 and 655 - The value is 1 or more: The ref specified in the [Drift Correction - The value is 0: No correction. This setting item will be applied	nake the reference value 35. erence value will be corre Interval] item. for each method.	follow the surround	ling	Explanation of the Selected Parameter

Figure 8-53. Adjustment of Parameters



9. Click "Enable Monitoring" in the state of "Monitoring: Enabled" to stop monitoring.



Figure 8-54. Stop Monitoring

10. Click "Disconnect" to disconnect the connection of UART.

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.
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 CtVCS+_Workspace¥Capac CtVCS+_Workspace¥Capac CtVCS+_Workspace¥Capac CtVCS+_Workspace¥Capac CtVCS+_Workspace¥Capac CtVCS+_Workspace¥Capac CtVCS+_Workspace¥Capac CtVCS+_Workspace¥Capac CtVCS+_Workspace¥Capac CtVCS+_UNDE Select a name of the target MCU. CtVCS+_UNDE CtVCS	Image: A configuration         Select or create a touch interface configuration.         Capacitive_Touch_project_Exa         Modify Configuration         Modify Configuration         Output Files for Tuning process.         Output Files         Duput Files         Modify Configuration         Duput files required for a tuning process.         Duput Files         Modify Configuration         Duput files         Point a process to call the main loop for touch in the main function.         Description:         Diagnement Program         Implement a process to call the main loop for touch in the main function.         Description:         Diagnement project using IDE tool.	<b>Detectore Program</b> Start debugging the target project in 1DE tool and execute the program. The board will ren. <b>Detectore Detectore Detectore</b>	<b>D: Implement Program.</b> Implement a program using the touch interfaces and build the project. <b>D: Connect Debug</b> Launch debugging for your target project and execute the program. <b>D: Connect UARI</b> Baud rate       153600         Port       Auto         Disconnect <b>Disconnect Show monitoring views and enable a monitoring views andenable a monitoring views and enable a monitoring views</b>

Figure 8-55. Disconnect UART



## 8.6 Sample Code

The sample code (qe\_touch\_sample.c) outputted by QE for Capacitive Touch is as follows.

In this sample code, a touch measurement cycle is created by a software timer.

```
* FILE : qe_sample_main.c
* DATE : 2022-02-14
* DESCRIPTION : Main Program for RL78
* NOTE: THIS IS A TYPICAL EXAMPLE.
#include "ge touch config.h"
#define TOUCH SCAN INTERVAL EXAMPLE (20 * 1000) /* microseconds */
void R CTSU PinSetInit(void);
void ge touch main(void);
void qe_touch_delay(uint16_t delay_us);
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) {}
 }
```



```
/* Main loop */
  while (true)
  {
    /* for [CONFIG01] configuration */
    err = RM_TOUCH_ScanStart(g_qe_touch_instance_config01.p_ctrl);
    if (FSP SUCCESS != err)
    {
       while (true) {}
    }
    while (0 == g \text{ 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. */
    }
    /* FIXME: Since this is a temporary process, so re-create a waiting process yourself. */
    qe_touch_delay(TOUCH_SCAN_INTERVAL_EXAMPLE);
  }
}
void qe_touch_delay(uint16_t delay_us)
{
  uint32_t i;
  uint32 t loops required;
  uint16_t clock_mhz;
  clock_mhz = (uint16_t)(R_BSP_GetFclkFreqHz() / 1000000);
  if (0 == clock_mhz)
  {
    clock_mhz = 1;
  }
  loops_required = ((uint32_t)delay_us * (uint32_t)clock_mhz);
  loops_required /= 20;
  for (i = 0; i < loops required; i++)
  {
    BSP_NOP();
  }
}
```



#### 8.7 Flowcharts





## 9. Appendix

### 9.1 Touch Measurement by Hardware Timer

This section explains the program using hardware timer (32-bit interval timer channels in 8-bit counter mode) to create a touch measurement cycle.

In addition, operations can be confirmed by turning on/off the LED on the target board depending on the sensor (button) touch state.

#### 9.1.1 Setup of Smart Configurator

1. Select the "Clocks" tab on the Smart Configurator and set the fSXP clock to be used as the interval timer. Also, uncheck the XT1 oscillator.



Figure 9-1. Setting Clock

2. Select "Components" tab and click 🖆 icon to open "New Component" dialog. Select "Interval Timer" module and click "Next".

Set configuration of "Interval Timer" to the follow and click "Finish".



Using QE (standalone ver.) to Develop Touch Applications for FPB board

Figure 9-2. Configuration of Interval Timer



3. Select "Interval Timer" module and set such as clocks.

omponents 🚵 🖄 🛱 🖽	Configure			C
si 😥 🐮 🐌	Clock setting	fSXP	~	fSXP
<ul> <li>Startup</li> <li>Generic</li> </ul>	Clock source	fITL0/128	~ 7	fITL0/128
♂ r_bsp	Interval timer setting			
<ul> <li>✓ B Drivers</li> <li>✓ B Timers</li> </ul>	Interval value	20	ms ~	20 ms
Config_ITL000	Detection of compare ma	atch/capture completion (INTITL)		
<ul> <li>Config_PORT</li> <li>Communications</li> <li>Config_UART0</li> </ul>	rity	Level 3 (low)	~	
<ul> <li>✓ implication</li> <li>✓ implication</li> <li>✓ implication</li> <li>✓ implication</li> <li>✓ implication</li> </ul>	Uncheck			

Figure 9-3. Setting "Config\_ITL000"

4. Set the Pin for LED. Set "P62" to high-level output in "Ports" module.

Software component configura	tion	Generate Code Generate Report
Components 🔤 🖬 🗄	Configure	
type filter text	Port selection PORT2 PORT4 PORT6	
<ul> <li>✓ Startup</li> <li>✓ Seneric</li> <li>✓ r_bsp</li> </ul>	Apply to all Unused In Out	Output 1
	P60 O Unused O In O Out	Output 1
Config_PORT	P61 O Unused O In O Out	Output 1
<ul> <li>✓ Middleware</li> <li>✓ Beneric</li> </ul>	Unused In Out	🗹 Output 1
● r_ctsu ● rm_touch	Unused In Out	Output 1
	<	>

Figure 9-4. Setting "P62"

5. Click 📴 icon on Smart Configurator to perform generating code.



## 9.1.2 Sample Code

The sample code (qe\_touch\_sample.c) outputted by QE for Capacitive Touch is as follows.

In this sample code, a touch measurement cycle is created using a hardware timer.

```
*
* FILE : qe_sample_main.c
* DATE : 2022-12-15
* 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) {}
 }
```



```
ITLS0 &= ~_01_ITL_CHANNEL0_COUNT_MATCH_DETECTE;
  R_Config_ITL000_Start();
  /* Main loop */
  while (true)
  {
    while ( 00 ITL CHANNELO COUNT MATCH NOT DETECTE == (ITLS0 &
_01_ITL_CHANNEL0_COUNT_MATCH_DETECTE)) {}
    ITLS0 &= ~ 01 ITL CHANNEL0 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 ge 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;
      }
    }
  }
}
```



#### 9.1.3 Flowcharts









### **10. Documents for Reference**

- RL78/G22 User's Manual: Hardware (R01UH0978)
- RL78 Family User's Manual: Software (R01US0015)
- RL78/G22 Fast Prototyping Board User's Manual (R20UT5121)
- RL78/G23 Capacitive Touch Evaluation System User's Manual (R12UZ0095) (The latest versions of the documents are available on the Renesas Electronics Website.)
- Application Note RL78 Family
  - Using the standalone version of QE to Develop Capacitive Touch Applications (R01AN6574)
- Application Note RL78 Debugging Functions Using the Serial Port (R20AN0632)
- Application Note RL78 Family Using QE and SIS to Develop Capacitive Touch Applications (R01AN5512)
   Application Note RL78 Family Capacitive Touch Sensing Light (CTSLI2L) Operation Figure 1.
- Application Note RL78 Family Capacitive Touch Sensing Unit (CTSU2L) Operation Explanation (R01AN5744)
- Application Note RL78 Family CTSU Module Software Integration System (R11AN0484)
- Application Note RL78 Family TOUCH Module Software Integration System (R11AN0485)
- Application Note Capacitive Sensor Microcontrollers CTSU Capacitive Touch Electrode Design Guide (R30AN0389)
- Application Note RL78 Family RL78/G23 Capacitive Touch Low Power Guide (SNOOZE function) (R01AN5886)
- RL78/G23 Capacitive Touch Low Power Guide (SMS function) (R01AN6670) (The latest versions of the documents are available on the Renesas Electronics Website.)
- Technical Updates/Technical Brochures (The latest versions of the documents are available on the Renesas Electronics Website.)

#### Website

- Renesas Electronics Website
   <u>http://www.renesas.com/</u>
- QE for Capacitive Touch related page <u>https://www.renesas.com/ge-capacitive-touch</u>
- Capacitive Sensing Unit related page
   <u>https://www.renesas.com/solutions/touch-key</u>



## Using QE (standalone ver.) to Develop Touch Applications for FPB board

## **Revision History**

		Description	
Rev.	Date	Page	Summary
1.00	Mar.20.23	-	First edition



# General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

#### 1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

#### 2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

#### 6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a 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.
- 6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

- 7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
- 8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- 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: <a href="http://www.renesas.com/contact/">www.renesas.com/contact/</a>.