

RL78 Family

Capacitive Touch Low Power Application Development using SMS

Introduction

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

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

Target Device

RL78/G22 RL78/G23



Contents

1.	System Overview	3
2.	Related Documents	3
3.	Auto Judgement Measurement using SNOOZE Mode Sequencer (SMS)	4
3.1	Module Flow for Measurements using SMS	4
3.2	External Pin Wiring (RL78/G22 only)	5
4.	Peripheral Functions	6
5.	Operation Confirmation Conditions	6
6.	Software Settings	7
6.1	Option Byte Setting	7
6.2	Capacitive Touch Settings	9
6.3	Components	10
7.	Application Example Overview	10
8.	Capacitive Touch Application Development Procedure	11
8. 8.1	Capacitive Touch Application Development Procedure	11
8. 8.1 8.2	Capacitive Touch Application Development Procedure Creating a Project Creating a Module with the Smart Configurator	11
8. 8.1 8.2 8.2.7	Capacitive Touch Application Development Procedure Creating a Project Creating a Module with the Smart Configurator Adding Components with the Smart Configurator	11
8. 8.1 8.2 8.2.2	Capacitive Touch Application Development Procedure Creating a Project Creating a Module with the Smart Configurator Adding Components with the Smart Configurator Changing the Component Configuration with Smart Configurator	11
8. 8.1 8.2 8.2.7 8.2.2 8.3	Capacitive Touch Application Development Procedure Creating a Project Creating a Module with the Smart Configurator Adding Components with the Smart Configurator Changing the Component Configuration with Smart Configurator Creating the Capacitive Touch Interface	
8. 8.1 8.2 8.2. 8.2. 8.3 8.4	Capacitive Touch Application Development Procedure Creating a Project Creating a Module with the Smart Configurator Adding Components with the Smart Configurator Changing the Component Configuration with Smart Configurator Creating the Capacitive Touch Interface Changing the Debug Configuration for Capacitive Touch Sensor Tuning	
8. 8.1 8.2 8.2.2 8.2.2 8.3 8.4 8.5	Capacitive Touch Application Development Procedure Creating a Project Creating a Module with the Smart Configurator Adding Components with the Smart Configurator Changing the Component Configuration with Smart Configurator Creating the Capacitive Touch Interface Changing the Debug Configuration for Capacitive Touch Sensor Tuning Capacitive Touch Sensor Tuning with QE for Capacitive Touch	
8. 8.1 8.2 8.2.7 8.2.7 8.3 8.4 8.5 8.6	Capacitive Touch Application Development Procedure Creating a Project Creating a Module with the Smart Configurator Adding Components with the Smart Configurator Changing the Component Configuration with Smart Configurator Creating the Capacitive Touch Interface Changing the Debug Configuration for Capacitive Touch Sensor Tuning Capacitive Touch Sensor Tuning with QE for Capacitive Touch Adding API Call for rm_touch Middleware	
8. 8.1 8.2 8.2. 8.3 8.4 8.5 8.6 9.	Capacitive Touch Application Development Procedure Creating a Project Creating a Module with the Smart Configurator Adding Components with the Smart Configurator Changing the Component Configuration with Smart Configurator Creating the Capacitive Touch Interface Changing the Debug Configuration for Capacitive Touch Sensor Tuning Capacitive Touch Sensor Tuning with QE for Capacitive Touch Adding API Call for rm_touch Middleware	



1. System Overview

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

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

2. Related Documents

This application note introduces the brief procedure required to create a workable application. For questions regarding tools used in the application example in this document or for more detailed instructions, refer to the following documents: e² studio / Smart Configurator, Software Integration System (SIS) Driver / Middleware, help function for Renesas Code Generator and QE for Capacitive Touch (included in e² studio help).



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

3.1 Module Flow for Measurements using SMS

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

• RL78/G22 module flow

CTSU2La uses DTC for port output.^{Note} The signal output from the port is used to generate an external interrupt signal. The interrupt signal triggers the ELC to start SMS processing.

Note. The port register (Pxx) is rewritten in 8-bit units by DTC transfer.

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



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

• RL78/G23 module flow

CTSU2L triggers ELCL to start SMS processing.



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



3.2 External Pin Wiring (RL78/G22 only)

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

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

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

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



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



4. Peripheral Functions

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

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

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

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

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

5. Operation Confirmation Conditions

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

Table 5-1 Operation Confirmation Conditions

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



6. Software Settings

6.1 Option Byte Setting

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

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

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

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

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



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



Figure 6-1 Setting of Option Bytes



6.2 Capacitive Touch Settings

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

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



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



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



6.3 Components

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

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

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

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

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

7. Application Example Overview

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

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

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

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



8. Capacitive Touch Application Development Procedure

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

- Create a new project using the e² studio wizard.
- Add necessary modules to the created project using the Smart Configurator.
- Create a capacitive touch interface using QE for Capacitive Touch.
- Tune the project using QE for Capacitive Touch.
- Add API calls for necessary SIS modules to the project and enable capacitive touch control.

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

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



8.1 Creating a Project

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

- 1. Start e² studio from the Windows start menu or the shortcut on the desktop. When the dialog box appears, create a workspace in a location of your choice.
- 2. Select [File] [New] [C/C++ Project] in e² studio to start the new project.
- 3. When the dialog box appears, select "Renesas RL78" "Renesas CC-RL C/C++ Executable Project", then click [Next].
- 4. In the dialog box appeared, enter your project name into the "Project name (P):" field and click [**Next**]. The project name in the application example is "Capacitive_Touch_Project_Example".
- 5. Specify the following items as shown below in the next dialog box.
- Language: C
- Toolchain: Renesas CC-RL
- Toolchain Version: v1.13.00
- Target Board: Custom
- Target Device: RL78/G22 (R7F102GGExFB)
 - RL78/G23 (R7F100GSNxFB)
- Configurations: Check "Create Hardware Debug Configuration", then select "E2 Lite (RL78)" from the pulldown menu.

		— — ×
ew Renesas	CC-RL Executable Project	
Select toolcha	in, device & debug settings	
Toolchain Set	tings	
Language:		
Toolchain:	Renesas CC-RL ~	
Toolchain Ver	sion: v1.13.00 ~	
	Manage Toolchains	
Device Setting	gs	Configurations
Target Board:	Custom ~	Create Hardware Debug Configuration
	Download additional boards	E2 Lite (RL78) ~
Target Device	R7F102GGExFB	
		PL79 Simulator
Endian	: Little	
		Create Release Configuration
Endian	Little	Create Debug Configuration RL78 Simulator Create Release Configuration

Figure 8-1 Toolchain, Device, and Debug Settings



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

							×
Device Selection You can filter devices by r	egular expre	ession					-
Search Device							
Device	RAM	ROM	Pin	RTOS	Smart Co	周辺コード	^
✓ RL78 - G22							
> RL78 - G22 16pin							
> RL78 - G22 20pin							
> RL78 - G22 24pin							
> RL78 - G22 25pin							
> RL78 - G22 30pin							
> RL78 - G22 32pin							
> RL78 - G22 36pin							
> RL78 - G22 40pin							
> RL78 - G22 44pin							
✓ RL78 - G22 48pin							
R7F102GGCxFB	4 KB	32 KB	48		✓	×	
R7F102GGCxNP	4 KB	32 KB	48		✓	×	Λ
R7F102GGExFB	4 KB	64 KB	48		~	×	$(\neg$
R7F102GGExNP	4 KB	64 KB	48		✓	×	
✓ RL78 - G23							
> RL78 - G23 30pin							
> RL78 - G23 32pin							
> RL78 - G23 36pin							
> RL78 - G23 40pin							
> RL78 - G23 44pin							
> RL78 - G23 48pin							
> RL78 - G23 52pin							
> RL78 - G23 64pin							
> RL78 - G23 80pin							
> RL78 - G23 100pin							
✓ RL78 - G23 128pin							
R7F100GSJxFB	24 KB	256 KB	128		✓	×	
R7F100GSKxFB	32 KB	384 KB	128		✓	×	
	48 KB	512 KB	128		✓	×	
R7F100GSLxFB							

Figure 8-2 Target Device Selection

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

- 6. After completing your selections, click [**OK**].
- 7. When the next dialog box appears, check "Use Smart Configurator", then click [Finish].

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



8.2 Creating a Module with the Smart Configurator

8.2.1 Adding Components with the Smart Configurator

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

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

1. Select the [**Clocks**] tab at the bottom of e² studio and specify the clock setting of RL78 MCU, as shown in the figure below.

In the application example, the low-speed on-chip oscillator is used as the low-speed peripheral clock (fSXP).

Uncheck "XT1 oscillator" and select "Low-speed on-chip oscillator" for fSXP as shown below.

A A					
Operation mode:	High-speed main mode 2.7(V)-5.5(V)	•			
High-speed on-ct	hip oscillator	7			
Frequency:	32 * (MHz)			840	
fHOCO start setting:	Normal 👻	•		32	(MHz)
(There is setting for	starting the high-speed on-chip oscillator a			fMAIN	
the times of release SNOOZE mode.)	e from STOP mode and of transitions to			9 32	(MHz)
				ICLK RCLK	(kda)
L				100 BIMP	(KHZ)
Middle-speed on	-chip oscillator	┣━━━			(MHz)
Frequency:	4. 👻 (MHz)				
		Divider			
X1 oscillator				mxp 🕕	
Operation mode:	X1 oscillation ·				(MHz)
Frequency:	5 (MHz)				
Stable time:	2118/(x 👻 52428.8(µs)				
				fiL	0.000
Low-speed on-chip o	22.750 //Hz)		Ī	32.106	(KF12)
The fill runs while WE	DT is operating or fSXP select Low-speed			#SVD	
on-chip oscillator				32.768	(kHz)
		-			
XT1 oscillator				TSXR U	(kHz)
Operation mode:	XT1 oscillation		<u></u>		
Frequency:	- (kHz)				
XT1 or on mode:	Low power consumption 1.				
Suppl e:					

Figure 8-3 Clock Configuration



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

Operation mode:	High-speed main mode 2.7(V)~5.5(V)	-<
	Figure 8-4 Operating Mode Setting (RL78/G22)	

Operation mode:	High-speed main mode 2.7(V)~5.5(V)	•
EVDD setting:	2.7 V ≤ EVDD0 ≤ 5.5 V	•

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

3. Select the [System] tab.

Select "Use emulator" and "E2 Lite" and uncheck "Use security ID", as shown in the figure below.

System configuration		
▼ On-chip debug setting		
On-chip debug operation setting O Unused	Use emulator	COM Port
Emulator setting C E2	E2 Lite	
Pseudo-RRM/DMM function setting O Unused	Used	
Start/Stop function setting Unused 	⊖ Used	
Monitoring point function setting Unused 	OUsed	
Security ID setting		
Security ID	0x000000000000000000000000000000000000	
Security ID authentication failure setting		
O Do not erase flash memory data		
Erase flash memory data		
Ļ		
Overview Board Clocks System Components	Pins Interrupt	

Figure 8-6 On-Chip Debug Setting



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



Figure 8-7 Change the r_bsp Version



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

Change Version						×
Version Selection	n					
Component name:	r_bsp					
Current version:	1.62					
Available versions:	1.61					\sim
\odot	< Back	Next >	Fini	sh	Cance	

Figure 8-8 Select the r_bsp Version

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



Figure 8-9 Add Components



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

💽 New Co	mponent					
Software	Component Selection				ta i	
Select con	nponent from those available	e in list		t		
Category	All				\sim	
Function	All				\sim	
Filter						
Compone	ents	Short Name	Туре	Version	^	
H A/D C	onverter		Code Generator	1.5.0		
H Board	Support Packages v1.62	r_bsp	RL78 Software In	1.62		
Capacitive Sensing Unit driver. r_ctsu RL78 Software In 1.50						
Show o	inly latest version					
Descriptio	n					
The analo	og to digital (A/D) converter	is function for convert	ing analog inputs to digital sig	als	~	
	ig to argital (100) contenter		ing analog inputs to argital sig			
×						
Download	RL78 Software Integration S	ystem modules				
Configure	general settings					
~						
?		< Back	lext > Finish	Cancel	1	

Figure 8-10 "New Component" Dialog Box

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



Figure 8-11 Download SIS Modules Link



Figure 8-12 Select Region

e								
RL7 Sel	RL78 Software Integration System Modules Download Select the RL78 Software Integration System modules for download							
	Title	Document No.	Rev.	Issue date ^	Select All			
\checkmark	RL78ファミリ CTSUモジュール Software Integrat	R11AN0484JJ01	Rev.1.50	201 05 21				
\checkmark	RL78ファミリ TOUCHモジュール Software Integr	R11AN0485JJ01	Rev.1.50	20	(3 ^{select All}			
	RL78ファミリ Renesas Flash Driver RL78 Type	R20AN0656JJ01	Rev.1.20	2024-05-20				
	RL78ファミリ Renesas Flash Driver RL78 Type	R20AN0653JJ01	Rev.1.20	2024-05-20				
	RL78ファミリ Renesas Flash Driver RL78 Type	R20AN0654JJ01	Rev.1.20	2024-05-20				
	RL78ファミリ Renesas Flash Driver RL78 Type	R20AN0655JJ01	Rev.1.20	202-05-20				
\checkmark	RL78ファミリ ボードサポートパッケージモジュール So	R01AN5522JJ01	Rev.1.62		(3)			
	RL78 Family ZMOD4410, ZMOD4450 and Z	R01AN6197EJ0	Rev.1.22	2023-07-04				
	RL78 Family FS2012 Sensor Control Module	R01AN6196EJ0	Rev.1.12	2023-04-26 ~				
<				>				
Мо	dule Folder Path:			(4)				
	C:¥Users¥ ¥.eclipse¥com.renesas.platform_download¥RL78_Modules hericModules Browse							
			De	ownload	Cancel			

Figure 8-13 Download SIS Modules



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

Software Component Solutions Set component from those available in its	😢 New Co	omponent				×	
Peter component from those available in list	Software	Component Selection				-]
Category All Image: Category Filter Image: Category All Filter Image: Category All Image: Category All Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Category Image: Catego	Select con	popent from those available in list					
Audio Image: Control and the second seco	Select con	iponent nom those available in rist					
Curryon and mathematical and the second s	Catagoni	41					
Finter Image: Short Name Type Version AD/Converter Short Name Type Version Based Support Packages - v1.62 r.bgo R178 Software In. 152 Converter Code Generator 1.4.1 Code Generator 1.4.1 Delay Counter Code Generator 1.4.1 Delay Counter Code Generator 1.4.1 Delay Counter Code Generator 1.4.1 Bash Driver (Riness Bash Driver R178 Typ. r.fd1/178.101.codelba Generator 1.4.1 Bash Driver (Riness Bash Driver R178 Typ. r.fd1/178.101.codelba Generator 1.4.1 Bash Driver (Riness Bash Driver R178 Typ. r.fd1/178.101.codelba Generator 1.4.1 Bash Driver (Riness Bash Driver R178 Typ. r.fd1/178.101.common Generator 1.1.1 Brites Driver (Riness Bash Driver R178 Typ. r.fd1/178.101.common Generator 1.1.1 Brites Driver (Riness Bash Driver R178 Typ. r.fd1/178.101.common Generator 1.1.1 Brites Driver (Riness Bash Driver R178 Typ. r.fd1/178.101.common Generator 1.5.1 1.5.1 Brites Driver M140leware r.fs10	Category	All				~	
There Image: Control of the second	Function	All				~	
Components Short Name Type Version 18 bard Support Raktages, + v1.62 r, byp R178 Software In 1.62 19 Cock Output / Buzzer Output Controller Code Generator 1.41 19 Data Transfer Gode Generator 1.41 19 Data Transfer Gode Generator 1.41 19 Data Driver RU28 Ty, r. (rfd. rfl.8, 01, catranse Generic SW 1.10 19 Data Driver RU28 Ty, r. (rfd. rfl.8, 01, catranse Generator 1.41 19 Data Driver RU28 Ty, r. (rfd. rfl.8, 01, catranse Generator 1.41 19 Data Driver RU28 Ty, r. (rfd. rfl.8, 01, catranse Genereator 1.41	Filter						
Who converter pice r. bsp RL78 Software In. 162 Beard Support Packages v1.62 r. bsp RL78 Software In. 162 Capacitive Sensing Und driver. r. ctsu RL78 Software In. 162 Delay Counter Code Generator 14.1 Beach Driver (Reness Rab Driver RL78 ty., r.rfd, r78, 101, codella Generic SW 110 Flash Driver (Reness Rab Driver RL78 ty., r.rfd, r78, 101, codella Generic SW 110 Flash Driver (Reness Rab Driver RL78 ty., r.rfd, r78, 101, codella Generic SW 110 Flash Driver (Reness Rab Driver RL78 ty., r.rfd, r78, 101, codella Generic SW 110 Flash Driver (Reness Rab Driver RL78 ty., r.rfd, r78, 101, codella Generic SW 110 Flash Driver (Reness Rab Driver RL78 ty., r.rfd, r78, 101, codella Generic SW 110 Flash Driver (Reness Rab Driver RL78 ty., r.rfd, r78, 101, codella Generic SW 110 Flash Driver (Reness Rab Driver RL78 ty., r.rfd, r78, 101, codella Generic SW 110 Flash Dri	Compone	entr ^	Short Name	Tune	Version		
Bard Software In. 162 r. bsp RJR Software In. 162 Bord Software In. 150 Gode Generator 1.11 Didy Gounter Gode Generator 1.12 Didy Gounter Gode Generator 1.12 Didy Gounter Gode Generator 1.12 Didy Gounter Gode Generator 1.22 Didy Gounter Gode Generator 1.21 Didy Grammer Bernston International State Internation Internatintere International State Internation Internati		onverter	Short Name	Code Generator	150		
Equative Sensing Unit drive: r_ctou RL78 Software In. 150 Clock Output /Buzzer Output Controller Code Generator 1.1 Data Transfer Controller Code Generator 1.1 Divider Function Code Generator 1.1 Event Link Controller Code Generator 1.4.1 Enab Driver[Reness Flash Driver RL78 Ty r.fd. /178_01_codefla. Generic SW 1.10 Flash Driver[Reness Flash Driver RL78 Ty r.fd. /178_01_codefla. Generic SW 1.10 Flash Driver[Reness Flash Driver RL78 Ty r.fd. /178_01_cotdefla. Generic SW 1.10 Flash Driver[Reness Flash Driver RL78 Ty r.fd. /178_01_cotdefla. Generic SW 1.10 Flash Driver[Reness Flash Driver RL78 Ty r.fd. /178_01_cotdefla. Generic SW 1.10 Flash Driver[Reness Flash Driver RL78 Ty r.fd. /178_01_cotdefla. Generic SW 1.10 Flash Driver[Reness Flash Driver RL78 Ty r.fd. /178_01_cotdefla. Generator 1.41	Board	Support Packages v1.62	r_bsp	RL78 Software In	1.62		
Cock Output /Buzzer Output Controller Code Generator 1.4.1 Data Transfer Controller Code Generator 1.4.1 Divider Function Code Generator 1.4.1 Divider Function Code Generator 1.4.1 Brash Driver Rizes Flash Driver Rizes Frid, dri28,101,codedla. Generic SW 1.10 Brash Driver(Renessa Flash Driver Rizes r,rfd, dri28,101,codmala Generic SW 1.10 Brash Driver(Renessa Flash Driver Rizes r,rfd, dri28,101,common Generic SW 1.10 Brash Driver(Renessa Flash Driver Rizes r,rfd, dri28,101,common Generic SW 1.10 Brash Driver(Renessa Flash Driver Rizes r,rfd, dri28,101,common Generic SW 1.10 Brash Driver(Renessa Flash Driver Rizes r,rfd, dri28,101,extraares Generic SW 1.10 Brash Driver(Renessa Flash Driver Rizes r,rfd, dri28,101,extraares Generic SW 1.10 Brash Driver(Renessa Flash Driver Rizes r,rfd, dri28,101,extraares Generic SW 1.10 Brash Driver(Renessa Flash Driver Rizes r,rfd, dri28,101,extraares Loce Loce Loce Brash Driver(Renessa Flash Driver Rizes r,rfd, dri28,101,extraares Loce Loce	H Capac	tive Sensing Unit driver.	r_ctsu	RL78 Software In	1.50		
Beta Transfer Controller Code Generator 1.3.1 Delay Counter Code Generator 1.4.1 Divider Function Code Generator 1.4.2 External Event Counter Code Generator 1.4.2 Generator 1.4.1 Generator 1.4.2 Flash Driver[Renessa Flash Driver RL78 Ty r.rfd.r178.01_codefla. Generic SW 1.10 Flash Driver[Renessa Flash Driver RL78 Ty r.rfd.r178.01_staflash Generic SW 1.10 Flash Driver[Renessa Flash Driver RL78 Ty r.rfd.r178.01_staflash Generic SW 1.10 Flash Driver[Renessa Flash Driver RL78 Ty r.rfd.r178.01_extmare Generic SW 1.10 Flash Driver[Renessa Flash Driver RL78 Ty r.rfd.r178.01_extmare I.12 Flash Driver[Renessa Flash Driver RL78 Dy r.rfd.r178.01_extmare I.12 Flash Driver[Renessa Flash Driver RL78 Dy r.rfd.r178.00 RL78 Software In	# Clock	Output /Buzzer Output Controller		Code Generator	1.4.1		
Belay Counter Code Generator 1.4.1 Buider Function Code Generator 1.4.2 Betent Link Controller Code Generator 1.2.0 Brad Driver[Reness Flash Driver RL78 Ty, r.rfd,r178_t01_contella. Generic SW 1.10 Brads Driver[Reness Flash Driver RL78 Ty, r.rfd,r178_t01_common Generic SW 1.10 Brads Driver[Reness Flash Driver RL78 Ty, r.rfd,r178_t01_common Generic SW 1.10 Brads Driver[Reness Flash Driver RL78 Ty, r.rfd,r178_t01_common Generic SW 1.10 Brads Driver[Reness Flash Driver RL78 Ty, r.rfd,r178_t01_common Generic SW 1.10 Brads Driver[Reness Flash Driver RL78 Ty, r.rfd,r178_t01_common Generic SW 1.10 Brads Driver[Reness Flash Driver RL78 Ty, r.rfd,r178_t01_common Generic SW 1.10 Brads Driver[Reness Flash Driver RL78 Ty, r.rfd,r178_t01_common Generic SW 1.10 Brads Driver[Reness Flash Driver RL78 Ty, r.rfd,r178_t01_common Generic SW 1.10 Brads Driver[Reness Flash Driver RL78 Ty, r.rfd,r178_t01_common Generator 1.4.1 Brads Driver RL78 Software In. 1.12 Generator 1.4.1	Data 1	Transfer Controller		Code Generator	1.3.1		
Event Link Controller Code Generator 1.4.2 Event Link Controller Code Generator 1.4.1 Ef Rab Driver(Reness Flash Driver RL78 Ty r.rfd.,178,101.codfallaß Generic SW 1.10 Endab Driver(Reness Flash Driver RL78 Ty r.rfd.,178,101.commo Generic SW 1.10 Endab Driver(Reness Flash Driver RL78 Ty r.rfd.,178,101.commo Generic SW 1.10 Endab Driver(Reness Flash Driver RL78 Ty r.rfd.,178,101.commo Generic SW 1.10 Endab Driver(Reness Flash Driver RL78 Ty r.rfd.,178,101.commo Generic SW 1.10 Endab Driver(Reness Flash Driver RL78 Ty r.rfd.,178,101.commo Generic SW 1.10 Endab Driver(Reness Flash Driver RL78 Ty r.rfd.,178,101.commo Generic SW 1.10 Endab Driver(Reness Flash Driver RL78 Ty r.rfd.,178,101.commo Generic SW 1.10 Endab Driver(Reness Flash Driver RL78 Ty r.rfd.,178,101.commo Generic SW 1.10 Endab Driver(Reness Flash Driver RL78 Ty r.rfd.,178,101.commo Generic SW 1.10 Endab Driver Interface Midd r.comms.j2c RL78 Software In. 1.02 Code Generator 1.41 E	Delay	Counter		Code Generator	1.4.1		
External Lownoiser Code Generator 1.4.1 # Bash Driver RL78 Ty, r_rfd_r178_t01_codefla Generator 1.4.1 # Hash Driver[Reness Flash Driver RL78 Ty, r_rfd_r178_t01_codefla Generator 1.4.1 # Hash Driver[Reness Flash Driver RL78 Ty, r_rfd_r178_t01_codefla Generator 1.4.1 # Hash Driver[Reness Flash Driver RL78 Ty, r_rfd_r178_t01_codefla Generator 1.4.1 # Hash Driver[Reness Flash Driver RL78 Ty, r_rfd_r178_t01_codefla Generator 1.4.1 # Hash Driver[Reness Flash Driver RL78 Ty, r_rfd_r178_t01_codefla Generator 1.4.1 # Hash Driver[Reness Flash Driver RL78 Ty, r_rfd_r178_t01_codefla Generator 1.4.1 # Hash Driver[Reness Flash Driver RL78 Ty, r_rfd_r178_t01_codefla Generator 1.4.1 # Hash Driver[Reness Flash Driver RL78 Ty, r_rfd_r178_t01_codefla Flash Driver[Reness Flash Driver RL78 Ty, # Hash Driver[Reness Flash Driver RL78 Ty, r_rfd_r178_t01_codefla Flash Driver[Reness Flash Driver RL78_Ty, Flash Driver RL78_Ty, # Hash Driver[Reness Flash Driver RL78_Ty, r_rfd_r178_t01_codefla Flash Driver RL78_Ty, Flash Driver RL78_Ty, Flash Driver RL78_Ty, Generator 1.4.1 # Hash Driver RL78_Ty, r_rfd_r178_t01_code Generator 1.4.1 Flash Driver RL78_	# Divide	er Function		Code Generator	1.4.2		
Image: Second	Event	Link Controller		Code Generator	1.2.0		
# Rash Driver(Reness Rash Driver RL/28 Ty, # RS1000 Sensor Middleware r, fs1015 RL/28 Software In. 1.01 # FS1015 Sensor Middleware r, fs1015 RL/28 Software In. 1.01 # FS3000 Sensor Middleware r, fs0000 RL/28 Software In. 1.02 # FS3000 Sensor Middleware r, fs0000 RL/28 Software In. 1.02 # IC Communication (Master mode) Code Generator 1.4.1 # IC Communication (Slave mode) Code Generator 1.4.1 # Input Puise Interval/Period Measurement Code Generator 1.4.0 # Interval Timer Code Generator 1.4.1 # Not Nutre Controller r, coh Sonson Middleware r, coh Zode Generator 1.4.0 # Not Nutre Controller r, coh Zode Generator 1.4.0 1.0.2 Code Generator 1.4.0 # Not Nutre Controller r, coh Zode Generator 1.4.0 1.0.2 Code Generator 1.4.0 1.0.2 1.0.2 1.0.2 1.0.2 1.0.2 1.0.2 1.0.2	# Flash	Driver Renesas Flash Driver RI 78 Tu	r rfd rl78 t01 codefla	Generic SW	1.10		
# Rah Driver(Reness Rah Driver RL78 Ty, # Rah Driver(Reness Rah Driver RL78 Ty, # F51015 Sensor Middleware r,frd, r/78_t01_extraares Generic SW 1.10 # F51015 Sensor Middleware r,fs1015 RL78 Software In., 1.01 # F51015 Sensor Middleware r,fs2012 RL78 Software In., 1.01 # F52012 Sensor Middleware r,fs2010 RL78 Software In., 1.02 # F53000 Sensor Middleware r,fs400x RL78 Software In., 1.02 # Hash Driver(Reness Rah Driver RL78 Ty, r fs400x RL78 Software In., 1.02 # F53001 Sensor Middleware r,fs400x RL78 Software In., 1.02 # Hash Driver(Reness Rah Driver RL78 Ty, r fs400x RL78 Software In., 1.02 # Hash Driver(Reness Rah Driver RL78 Ty, r fs400x RL78 Software In., 1.02 # Hash Driver(Reness Rah Driver RL78 Ty, r fs400x RL78 Software In., 1.02 # Hash Driver RL78 Tyme r, fs400x RL78 Software In., 1.02 # IC Communication Driver Interface Midd. r, cold Generator 1.41 1.41 # Ord Notabut Code Generator 1.40 1.41 1.41 1.41 # Key Interrupt Code Generator 1.40 1.41	Elash I	Driver Renesas Flash Driver RL78 Ty	r rfd rl78 t01 dataflash	Generic SW	1.10		
# Flash Driver (Renesas Flash Driver RL78 Tyr., r_rfd;r78,tb1_common_Generic SW 1.10 # F51015 Sensor Middleware r_r51015 RL78 Software In., 1.01 # F52012 Sensor Middleware r_r52012 RL78 Software In., 1.01 # F53003 Sensor Middleware r_r53000 RL78 Software In., 1.02 # F5400X Sensor Middleware r_rb400x RL78 Software In., 1.02 # F5400X Sensor Middleware r_rb400x RL78 Software In., 1.02 # IC Communication (Master mode) Code Generator 1.41 # Interval/Period Measurement Code Generator 1.42 # Interval Signal High-fLow-Level Width Mea r_comms_12c RL78 Software In., 1.02 # Interval Timer Code Generator 1.40 # put Signal High-fLow-Level Width Mea Code Generator 1.40 # put Signal High-fLow-Level Width Mea Code Generator 1.40 # ports Code Generator 1.40 <td>H Flash</td> <td>Driver[Renesas Flash Driver RL78 Ty</td> <td>r rfd rl78 t01 extraarea</td> <td>Generic SW</td> <td>1.10</td> <td></td> <td></td>	H Flash	Driver[Renesas Flash Driver RL78 Ty	r rfd rl78 t01 extraarea	Generic SW	1.10		
# FS1015 Sensor Middleware r,fs1015 RL78 Software In. 1.01 # FS2012 Sensor Middleware r,fs2012 RL78 Software In. 1.02 # FS3000 Sensor Middleware r,fs3000 RL78 Software In. 1.02 # HS400x Sensor Middleware r,hs400x RL78 Software In. 1.02 # HS400x Sensor Middleware r,hs400x RL78 Software In. 1.02 # IC Communication (Master mode) Code Generator 1.51 # IC Communication Driver Interface Middr.comms,J2c Code Generator 1.43 # Input Pulse Interval/Period Measurement Code Generator 1.42 # Interrupt Controller Code Generator 1.40 # Netrupt Code Generator 1.40 # Netrupt Code Generator 1.41 # Netrupt Code Generator 1.40 # Netrupt Code Generator 1.40 # Netrupt Code Generator 1.41 # Network Niddleware r_ob1203 RU78 Software In. 1.02 # One-Shot Pulse Output Code Generator 1.60 Note # SoNOCXE Mode Sequencer m.touch RU78 Software In. 1.50	H Flash I	Driver[Renesas Flash Driver RL78 Ty	r_rfd_rl78_t01_common	Generic SW	1.10		
# F52012 Sensor Middleware r, fs2012 RL78 Software In 1.12 # F53000 Sensor Middleware r, fs3000 RL78 Software In 1.01 # H5300x Sensor Middleware r, hs300x RL78 Software In 1.02 # H5300x Sensor Middleware r, hs300x RL78 Software In 1.02 # IC Communication (Master mode) Code Generator 1.51 # IC Communication (Very Interface Midd r_comms_j2c RL78 Software In 1.11 # Interupt Controller Code Generator 1.43 # Interupt Controller Code Generator 1.40 # Key Interrupt Code Generator 1.40 # New Interrupt Code Generator 1.40 # Notobust Code Generator 1.41 # Notobust Code Generator 1.40 # Notobust Code Generator 1.41 # Notobust Code Generator 1.41 # Notobust Code Generator 1.40 <tr< td=""><td>⊕ FS101</td><td>5 Sensor Middleware</td><td>r_fs1015</td><td>RL78 Software In</td><td>1.01</td><td></td><td></td></tr<>	⊕ FS101	5 Sensor Middleware	r_fs1015	RL78 Software In	1.01		
# F53000 Sensor Middleware r_fs3000 RL78 Software In. 1.01 # H5300x Sensor Middleware r_fs300x RL78 Software In. 1.02 # H5400x Sensor Middleware r_fs400x RL78 Software In. 1.02 # IIC Communication (Master mode) Code Generator 1.4.1 # IIC Communication (Master mode) Code Generator 1.4.1 # Input Pulse Interval/Period Measurement Code Generator 1.4.2 # Interval Signal High-/Low-Level Width Mea. Code Generator 1.4.0 # Interval Signal High-/Low-Level Width Mea. Code Generator 1.4.0 # Interval Signal High-/Low-Level Width Mea. Code Generator 1.4.0 # Rel Interval Timer Code Generator 1.4.0 # Rel Time Clock Code Generator 1.4.0 # Rel Time Clock Code Generator 1.4.0 # Spl (CSI) Communication Code Generator 1.4.0 #	⊕ FS201	2 Sensor Middleware	r_fs2012	RL78 Software In	1.12		
# H3300x Sensor Middleware r_h3300x RL78 Software In 1.22 # H5400x Sensor Middleware r_h400x RL78 Software In 1.02 Code Generator 1.5.1 Code Generator 1.5.1 FILC Communication (Slave mode) Code Generator 1.4.1 FILC Communication (Slave mode) Code Generator 1.4.1 FILC Communication Driver Interface Midd r_comms_j2c RL78 Software In 1.11 Input Signal High-/Low-Level Width Mea Code Generator 1.4.0 Code Generator 1.4.0 Code Generator 1.4.0 # Interval Timer Code Generator 1.4.0 # Code Generator 1.4.0 Code Generator 1.4.0 # Revel Interrupt Code Generator 1.4.0 Code Generator 1.4.0 # One-Shot Pulse Output Code Generator 1.4.1 1.0.2 Code Generator 1.4.1 # PWM Output Code Generator 1.4.0 Code Generator 1.4.1 1.0.2 Code Generator 1.4.1 # Spl (CSI) Communication Code Generator 1.4.1 Code Generator 1.4.1 1.0.0 1.0.0 1.0.0	⊕ FS300	0 Sensor Middleware	r_fs3000	RL78 Software In	1.01		
# H5400x Sensor Middleware r_h5400x RL78 Software In 1.02 # IIC Communication (Master mode) Code Generator 1.4.1 # IIC Communication Driver Interface Midd r_comms_j2c RL78 Software In 1.11 # Input Signal High-/Low-Level Width Mea Code Generator 1.4.3 # Interrupt Controller Code Generator 1.4.0 # Interrupt Controller Code Generator 1.4.0 # Interrupt Controller Code Generator 1.4.0 # Rey Interrupt Code Generator 1.4.0 # OB1203 Sensor Middleware r_ob1203 RL78 Software In 1.02 # One-Shot Pulse Output Code Generator 1.4.0 0.0 0.0 # Ports Code Generator 1.4.0 0.0	HS300	0x Sensor Middleware	r_hs300x	RL78 Software In	1.22		
If IC Communication (Master mode) Code Generator 1.5.1 If IC Communication (Slave mode) Code Generator 1.4.1 If IC Communication Driver Interface Midd r_comms_J2c R.178 Software In 1.11 Input Signal High-/Low-Level Width Mea Code Generator 1.4.2 Interrupt Controller Code Generator 1.4.0 Immer Code Generator 1.4.1 Immer Code Generator 1.4.0 Immer Code Generator 1.4.0 Immer <	HS400	0x Sensor Middleware	r_hs400x	RL78 Software In	1.02		
Code Generator 1.4.1 In C. Communication Driver Interface Midd, r_comms_i2c RL78 Software In., 1.11 Code Generator 1.4.3 Code Generator 1.4.3 Code Generator 1.4.0 Code Generator 1.4.0 Interrupt Controller Code Generator 1.4.0 Code Generator 1.4.1 Ports Code Generator 1.4.1 Ports Code Generator 1.4.1 Code Generator 1.4.1 Code Generator 1.4.1 Code Generator 1.4.0 Code Generat	IIC Co	mmunication (Master mode)		Code Generator	1.5.1		
In Loommunication Driver Interface Wilds f_Comms_J2c R.CrossOftware Int 1.11 Input Pulse Interval/Period Measurement Code Generator 1.4.3 Code Generator 1.4.2 Code Generator 1.4.2 Interval Timer Code Generator 1.4.0 Key Interrupt Code Generator 1.4.0 One-Shot Pulse Output Code Generator 1.4.0 One-Shot Pulse Output Code Generator 1.4.1 Ports Code Generator 1.4.1 Ports Code Generator 1.4.1 PWM Output Code Generator 1.4.0 Square Wave Output Code Generator 1.4.1 Ports Code Generator 1.4.1 Touch middleware r_ob1203 Code Generator 1.4.1 Square Wave Output Code Generator 1.4.1 Code Generator 1.4.1 Watchdog Timer m_touch RL78 Software In 1.50 Code Generator 1.4.0 ZMOD4XXX Sensor Middleware r_zmod4xox RL78 Software In 1.22 Code Generator 1.4.0 Devinition Code Generator 1.4.0 1.4.2 </td <td></td> <td>mmunication (Slave mode)</td> <td></td> <td>Code Generator</td> <td>1.4.1</td> <td></td> <td></td>		mmunication (Slave mode)		Code Generator	1.4.1		
Input Signal High-/Low-Level Width Mea Code Generator 14.2 Interrupt Controller Code Generator 14.0 Interval Timer Code Generator 14.0 Key Interrupt Code Generator 13.0 Oble Oble Oble Oble Output Code Generator 14.0 Ports Code Generator 16.0 PWM Output Code Generator 16.0 Real-Time Clock Code Generator 14.1 PWM Output Code Generator 16.0 Real-Time Clock Code Generator 16.0 Real-Time Clock Code Generator 14.1 Square Wave Output Code Generator 16.0 Watchdog Timer m_touch RL78 Software In 15.0 UART Communication Code Generator 16.0 16.0 Watchdog Timer Code Generator 16.0 16.0 16.0 ZMOD4XXX Sensor Middleware r_zmod4xox RL78 Software In 15.0 Description Code Generator 16.0 16.0 16.0 Download RL78 Software Integration System modules configure assuming access via the Touch middleware layer, but		mmunication Driver Interface Midd	r_comms_I2c	KL/8 Software In	1.11		
Interrupt Controller Code Generator 14.0 Interval Timer Code Generator 14.0 Key Interrupt Code Generator 13.0 OB1203 Sensor Middleware r_ob1203 RL78 Software In 1.02 One-Shot Pulse Output Code Generator 14.0 Ports Code Generator 14.0 Ports Code Generator 14.0 Real-Time Clock Code Generator 14.0 Synopside Code Generator 14.1 PWM Output Code Generator 14.0 Synopside Code Generator 14.0 Real-Time Clock Code Generator 14.0 Synopside Code Generator 14.0 Synopside Code Generator 14.0 Synopside Code Generator 14.0 Watchdog Timer Code Generator 14.0 ZMDO4XXX Sensor Middleware r_zmod4xxx RL78 Software In 122 Description Interval Code Generator 14.0 Interval Timer Download RL78 Software Integration System modules Configure dessuming access via the Touch middleware layer, but can also be accessed		Signal High-/Low-Level Width Mea		Code Generator	142		
Interval Timer Code Generator 14.0 Key Interrupt Code Generator 1.3.0 OB1203 Sensor Middleware r_ob1203 RL78 Software In 1.02 One-Shot Pulse Output Code Generator 1.4.0 Ports Code Generator 1.4.1 PMM Output Code Generator 1.4.1 Real-Time Clock Code Generator 1.4.0 Ar SNOOZE Mode Sequencer Graphical Config 1.3.1 SPI (CSI) Communication Code Generator 1.4.0 Square Wave Output Code Generator 1.4.0 Touch middleware. m_touch RL78 Software In 1.50 UART Communication Code Generator 1.6.0 Voltage Detector Code Generator 1.6.0 Voltage Detector Code Generator 1.4.0 ZMOD4XXXX Sensor Middleware r_zmod4xxx RL78 Software In 1.22 Description Inter CTSU2L module is a CTSU2L driver for the Capacitive Sensing Unit. The CTSU2L module is configured assuming access via the Touch middleware layer, but can also be accessed from the user application. v Download RL78 Software Integration System modules Configure general settings	H Intern	upt Controller		Code Generator	1.4.0	٦◀	
Key Interrupt Code Generator 1.3.0 OB1203 Sensor Middleware r_ob1203 RL78 Software In 1.02 One-Shot Pulse Output Code Generator 1.4.0 Ports Code Generator 1.4.0 Real-Time Clock Code Generator 1.6.0 Are Real-Time Clock Code Generator 1.4.0 SNOOZE Mode Sequencer Graphical Config 1.3.1 SSINO CZE Mode Sequencer Graphical Config 1.3.1 Sysuer Wave Output Code Generator 1.4.0 Toch middleware. m_touch RL78 Software In 1.50 UART Communication Code Generator 1.4.0 ZMOD4XXX Sensor Middleware r_zmod4xxx RL78 Software In 1.22 Description Image: Code Generator 1.4.0 Image: Code Generator 1.4.0 Dependency : r_bsp version(s) 1.62 The CTSU2L module is a CTSU2L driver for the Capacitive Sensing Unit. The CTSU2L module is configured assuming access via the Touch middleware layer, but can also be accessed from the user application. v Download RL78 Software Integration System modules Configure general settings	H Interva	al Timer		Code Generator	1.4.0		
Image: Construction of the construc	🖶 Key In	iterrupt		Code Generator	1.3.0		
One-Shot Pulse Output Code Generator 14.0 Ports Code Generator 14.1 PWM Output Code Generator 16.0 Code Generator 14.0 14.0 Real-Time Clock Code Generator 14.0 SNOOZE Mode Sequencer Graphical Config 13.1 SQuare Wave Output Code Generator 14.1 Touch middleware. rm_touch RL78 Software In 1.50 UART Communication Code Generator 1.4.0 Watchdog Timer Code Generator 1.4.0 ZMOD4XXX Sensor Middleware r_zmod4xxx RL78 Software In 1.22 Description Dependency: r_bsp version(s) 1.62 The CTSU2L module is a CTSU2L driver for the Capacitive Sensing Unit. The CTSU2L module is configured assuming access via the Touch middleware layer, but can also be accessed from the user application. v Download RL78 Software Integration System modules Configure general settings v	0B120	03 Sensor Middleware	r_ob1203	RL78 Software In	1.02		
Ports Code Generator 14.1 PWM Output Code Generator 1.6.0 Real-Time Clock Code Generator 1.4.0 SNOZE Mode Sequencer Graphical Config 1.3.1 SPI (CSI) Communication Code Generator 1.4.1 Square Wave Output Code Generator 1.4.1 Output Milleware. rm_touch RL78 Software In 1.50 UART Communication Code Generator 1.6.0 Voltage Detector Code Generator 1.4.0 Watchdog Timer Code Generator 1.4.0 ZMOD4XXX Sensor Middleware r_zmod4xxx RL78 Software In 1.22 Description Dependency : r_bsp version(s) 1.62 The CTSU2L module is a CTSU2L driver for the Capacitive Sensing Unit. The CTSU2L module is configured assuming access via the Touch middleware layer, but can also be accessed from the user application. v Download RL78 Software Integration System modules Configure general settings v	H One-S	hot Pulse Output		Code Generator	1.4.0		
PWM Output Code Generator 1.6.0 Real-Time Clock Code Generator 1.4.0 SNOOZE Mode Sequencer Graphical Config 1.3.1 SPI (CSI) Communication Code Generator 1.4.1 Square Wave Output Code Generator 1.4.1 Touch middleware. rm_touch RL78 Software In 1.50 UART Communication Code Generator 1.6.0 Voltage Detector Code Generator 1.6.0 Voltage Detector Code Generator 1.4.0 ZMOD4XXX Sensor Middleware r_zmod4xxx RL78 Software In 1.22 Description Code Generator 1.4.0 1.22 Download RL78 Software Integration System modules Configure general settings	Ports	-		Code Generator	1.4.1	┛]
Image: Real-Lime Clock Code Generator 1.4.0 Image: SNOOZE Mode Sequencer Graphical Config 1.3.1 Image: SNOOZE Mode Sequencer Image: SNOOZE Mode Sequencer 1.4.0 Image: SNOOZE Mode Sequencer Image: SNOOZE Mode Sequencer 1.4.0 Image: SNOOZE Mode Sequencer Image: SNOOZE Mode Sequencer 1.4.0 Image: SNOOZE Mode Sequencer Image: SNOOZE Mode Sequencer 1.4.0 Image: SNOOZE Mode Sequencer Image: SNOOZE Mode Sequencer 1.4.0 Image: SNOOZE Mode Mode Mode Mode Mode Mode Mode Mode	PWM	Output		Code Generator	1.6.0		
SNOUZE Wobe Sequencer Graphical Config 1.3.1 Sprincal Config Code Generator 1.4.1 Square Wave Output Code Generator 1.4.1 Touch middleware. rm_touch RL78 Software In 1.50 UART Communication Code Generator 1.6.0 Voltage Detector Code Generator 1.4.0 Watchdog Timer Code Generator 1.4.0 ZMOD4XXX Sensor Middleware r_zmod4xxx RL78 Software In 1.22 Description Dependency : r_bsp version(s) 1.62 Image: Code Generator 1.4.0 Dependency : r_bsp version(s) 1.62 Image: Code Generator Image: Code Generator Image: Code Generator Dependency : r_bsp version(s) 1.62 Image: Code Generator Image: Code Generator Image: Code Generator Image: Code Generator Download RL78 Software Integration System modules Image: Configure general settings Image: Configure general settings Image: Configure general settings	the Real-T	ime Clock		Code Generator	1.4.0		
Square Wave Output Code Generator 1.4.0 Touch middleware. rm_touch RL78 Software In 1.50 UART Communication Code Generator 1.6.0 Voltage Detector Code Generator 1.4.0 Watchdog Timer Code Generator 1.4.0 ZMOD4XXX Sensor Middleware r_zmod4xxx RL78 Software In 1.22 Description Dependency : r_bsp version(s) 1.62 Image: Configure d assuming access via the Touch middleware layer, but can also be accessed from the user application. Image: Configure d assuming access via the Touch middleware layer, but can also be accessed from the user application. Download RL78 Software Integration System modules Configure general settings Image: Configure general settings		SI) Communication		Graphical Config	1.3.1		
Image: Control of the second secon	Square	e Wave Output		Code Generator	1.4.0		
UART Communication Code Generator 1.6.0 Voltage Detector Code Generator 1.3.0 Watchdog Timer Code Generator 1.4.0 ZMOD4XXX Sensor Middleware r_zmod4xxx RL78 Software In 1.22 Description Dependency : r_bsp version(s) 1.62 The CTSU2L module is a CTSU2L driver for the Capacitive Sensing Unit. The CTSU2L module is configured assuming access via the Touch middleware layer, but can also be accessed from the user application. Download RL78 Software Integration System modules Configure general settings	H Touch	middleware.	rm touch	RL78 Software In	1.50		
Woltage Detector Code Generator 1.3.0 Watchdog Timer Code Generator 1.4.0 ZMOD4XXX Sensor Middleware r_zmod4xxx RL78 Software In 1.22 Description Dependency : r_bsp version(s) 1.62 The CTSU2L module is a CTSU2L driver for the Capacitive Sensing Unit. The CTSU2L module is configured assuming access via the Touch middleware layer, but can also be accessed from the user application. Download RL78 Software Integration System modules	UART	Communication		Code Generator	1.6.0	-	
Watchdog Timer Code Generator 1.4.0 ZMOD4XXX Sensor Middleware r_zmod4xxx RL78 Software In 1.22 Description Dependency : r_bsp version(s) 1.62 1.4.0 1.22 The CTSU2L module is a CTSU2L driver for the Capacitive Sensing Unit. The CTSU2L module is configured assuming access via the Touch middleware layer, but can also be accessed from the user application. Download RL78 Software Integration System modules Configure general settings Configure general settings 	H Voltag	ge Detector		Code Generator	1.3.0		
Image: Tripping the second	H Watch	ndog Timer		Code Generator	1.4.0	-	
Description Dependency : r_bsp version(s) 1.62 The CTSU2L module is a CTSU2L driver for the Capacitive Sensing Unit. The CTSU2L module is configured assuming access via the Touch middleware layer, but can also be accessed from the user application. Download RL78 Software Integration System modules Configure general settings	# ZMOD	04XXX Sensor Middleware	r_zmod4xxx	RL78 Software In	1.22		
Dependency : r_bsp version(s) 1.62 The CTSU2L module is a CTSU2L driver for the Capacitive Sensing Unit. The CTSU2L module is configured assuming access via the Touch middleware layer, but can also be accessed from the user application. Download RL78 Software Integration System modules Configure general settings	Descriptio	n					
application. Download RL78 Software Integration System modules Configure general settings 	Depende The CTSU configure	ncy : r_bsp version(s) 1.62 J2L module is a CTSU2L driver for the ed assuming access via the Touch mid	Capacitive Sensing Unit. T Idleware layer, but can also	The CTSU2L module i b be accessed from t	s he user	î	
Download RL/8 Software Integration System modules Configure general settings	applicatio	on.				~	
	Download Configure	I RL78 Software Integration System m general settings	nodules				
O < Back Next > Finish Cancel	?	< Ba	ck Next >	Finish	Cance	4	

Figure 8-14 Software Component Settings



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

New Component			×	
dd new configurati	on for selected component			
				_
Interrupt Controller				
Configuration name:	Config_INTC			
Resource:	INTC		\sim	
Interval Timer				
Configuration name:	Config_TAU0_1			<u> </u>
Operation:	8 bit count mode		\sim	
Resource:	TAU0_1		\sim	
Caution:				
16 bit capture mode I When 8 bit mode ITL When 16 bit mode ITI	TL000_ITL001 can not be used together with 16 bit count mode I is used, 16 bit ITL and 32 bit ITL can not be used. . is used, 8 bit ITL and 32 bit ITL can not be used.	L012_ITL0	13. ^	• No
Ports				
Configuration name:	Config_PORT			
Resource:	PORT		~	
Voltage Detector				
Configuration name:	Config_LVD0			
Resource:	LVD0		\sim	
?	< Back Next > Finish	Ca	ncel	

Figure 8-15 Component Resource Settings

Components are added as shown below.

Software component configuration					
Components			•	Config	ure
yee filter text yee filter text y ≥ Startup y ≥ Startup y ≥ Drivers y ≥ Powe CC y ≥ Inter CC y ≥ Inter CC y ≥ Inter CC y ≥ Inter CC y ≥ Gene y ⊂ CC y ≥ Gene y ⊂ CC y ⊂ CCC y ⊂ CCC y ⊂ CCC y ⊂ CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	ric bsp onfig_LVD0 upt onfig_INTC rs onfig_ITL000 ort vare ric ctsu o_touch	nt and reset funct	te		
Overview Board	Clocks Syste	m Components	Pins	Interrupt	

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



8.2.2 Changing the Component Configuration with Smart Configurator.

Configure the added components as follows:

- CTSU Component Settings Note that configurations differ between RL78/G22 and RL78/G23.
 - (1) Select "DTC" for "Data transfer of INTCTSUWR and INTCTSURD".
 - (2) Specify "Data storage address setting for CTSURD" according to the device used. RL78/G22: Set "0xFEF00". RL78/G23: Set "0xFF500".
 - (3) For the RL78/G22, select the port pin to output the touch measurement complete signal transmitted from CTSU2La via the DTC. This setting is not shown for the RL78/G23 since it is not required.
 - Output port number for external trigger: Select port group.
 - Bit number for external trigger output: Select bit number from port group selected above.

Caution: The port register (Pxx) is rewritten in 8-bit units by DTC transfer.

Therefore, a port register (Pxx) that is the destination of DTC transfer cannot be used for other functions during auto judgement measurement by the SMS. Use a port register (Pxx) that no other system will be using.

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

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

For the RL78/G23, any TS pins can be selected when using it.



		— 🗆	×
Capacitive_Touch_Project_Example.scfg $ imes$			- 0
oftware component configuration		🔁 🕒	
	Gener	rate Code Generate R	eport
omponents 🚵 🛃 📮 🕀	Configure		í
	Property	Value	
type filter text	V @ Configurations	Line most and defends	_
✓	# Parameter cneck	Use system default	
🗸 🗁 Generic	# Dice atting	DIC Sotting in r stau	
💣 r_bsp	# Dic setting	Setund III Cisu	
✓ ➢ Drivers	Auto-Judgment function in Shooze mode using SMS		
Power management and reset function	Data storage address setting for CTSUMP	OvEE800	\exists
Config_LVD0	# Interrupt level for INTCTSUWR	Level 2	
V 🗁 Interrupt	# Interrupt level for INTCTSURD	Level 2	\vdash
Config_INTC	# Interrupt level for INTCTSUEN	Level 2	
V 🗁 Timers	# Output port number for external trigger	PORT2	
Config_ITL000	# Bit number for external trigger output	BIT2	
✓ ⇒ I/O port	# Interrupt port number for external trigger	INTP5	_
Config_PORT	V II Resources		
✓ ➢ Middleware			
V 🦻 Generic	🔨 TSCAP Pin	V lsed	
er_ctsu	🛰 TS00 Pin	Used	
rm_touch	🛰 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	
	TS11 Pin	Used	
	TS12 Pin	Used	
	IS13 Pin	Used	\vdash
	TS15 Pin	V Used	\vdash
	TS16 Pin		
	TS17 Din		
	TS18 Din		\exists
	TS19 Pin	Used	\vdash
	TS20 Pin	Used	
	TS21 Pin	V Used	
	TS22 Pin	Vised	
	TS23 Pin	Vised	
	🔨 TS24 Pin	Used	
	🛰 TS25 Pin	Used	
	🛰 TS26 Pin	🔽 Used	
	🥆 TS27 Pin	🔽 Used	
	🛰 TS28 Pin	🔽 Used	

Figure 8-17 CTSU Component Settings



• Touch Component Configuration Use the default setting as is.

Components $\bowtie \bowtie \bowtie \lg_2 = \boxdot$	Configure	٩
Ype filter text ✓ Image: Startup ✓ Image: Startup	Property	Value Use system default Disable UARTO TypeA : Counter of exceed threshold is hold within h

Figure 8-18 Touch Component Setting

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

Components 🚵 🗳 🖧 🖹 🕀	Configure	(i)
type filter text	INTPO setting INTPO Valid edge Falling edge Priority Level 3 (low)	
✓ 🧽 Startup ✓ 🧁 Generic 💣 r_bsp	INTP1 setting INTP1 Valid edge Falling edge Priority Level 3 (low)	
Config_LVD0	INTP2 setting Valid edge Falling edge Priority Level 3 (low) Valid edge	
Config_INTC Config_INTC Config_INTC	INTP3 setting INTP3 Valid edge Falling edge Priority Level 3 (low)	
Config_ITL000 V Config_PORT	INTP4 setting INTP4 Valid edge Falling edge V Priority Level 3 (low)	
 ✓ ➢ Middleware ✓ ➢ Generic ✓ Ț_ctsu 	INTPS setting INTPS Valid edge Falling edge Priority Level 3 (low)	
Trm_touch	[INTP6 setting	

Figure 8-19 INTC Component Setting

• ITL Component Configuration

Components $\ge a \ge a \ge a \ge a$	Configure		()
type filter text	Clock setting Operation clock (fITL0)	fSXP ~	
✓ 🧁 Startup ✓ 🗁 Generic	Clock source	fITL0/128	equency: 0.256 kHz)
 ♂ r_bsp ✓ ⊘ Drivers ✓ ⊘ Power management and reset function 	Interval timer setting Interval value	20 ms V (Actual va	alue: 19.53125)
 Config_LVD0 Interrupt 	Interrupt setting Detection of compare match	/capture completion (INTITL)	
Config_INTC	Priority	Level 3 (low) \checkmark	
Config_ITL000			
Config_PORT			
 ✓ I Middleware ✓ I Generic 			
erctsu erm_touch			

Figure 8-20 ITL Component Setting



PORT Component Configuration

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

Components	èn da lª₂ ⊟ ⊞	Configure			
▼∎ Iss type filter text ✓ ➢ Startup ✓ ➢ Generic ○ ⑦ r_Dsp	t t	Port seler	ORT2		
 Drivers Drivers Config_LVD0 Config_LVD0 Interrupt Config_INTC Timers Config_INTC 	and reset function	□ P □ P	ORT4 ORT6 ORT12	PORT5 PORT7 PORT13	3
✓ → I/O port ✓ → Middleware		P	ORT14		
 ✓		Port mo	ode setting d Pmn register	r values	O Read digital output level

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

Components 🚵 🛃 📮 🕀	Configure
ve filter text	Port selection PORT2
✓ ➢ Startup ✓ ➢ Generic ✓ ௺ Losp	Apply to all Unused In Out
 Config_LVD0 Interrupt 	P20
Config_INTC	P21
	P22 O Unused O In Out
	P23

Figure 8-22 PORT Component Setting (PORT2 tab)

• LVD Component Configuration

Components 🚵 🖾 🖾 🗉 🕀	Configure	í
Image: second secon	Operation mode setting Reset mode When setting LVD0 to reset mode, set the detection voltage of LVD1 higher than the detection voltage of LVD0. O Interrupt mode If LVD0 is set to interrupt mode and the LVD0 detection voltage is greater than the LVD1 detection voltage, LVD0 becomes undefined after the LVD1 setting following release from the reset state. INTLVI priority	
Config_LVD0 Config_INTC Config_INTC Config_INTC Config_INTC Config_ITL000 Config_ORT Config_PORT Config_PORT Config_PORT Config_Contig_PORT Config_PORT Config_POR	Voltage detection setting Reset generation level(VLVD0) 2.62 (V) Interrupt generation level(VLVD0) 1.65 (V)	





• BSP Component Configuration

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

Software component configuration		Genera	ite Code Generate Report
Components 🚵 🛃 🖓 🕞 🕀	Configure	1	1
Image: Second control of the second	Property ▼ Start up select # Control of illicit memory access detection(IAWEN) # Protected area in the RAM(GRAM1-0) # Protection of the port control registers(GPORT) # Protection of the interrupt control registers(GPORT) # Protection of the interrupt control registers(GINT) # Protection of peripheral functions by Code Generator/Smart Configural # API functions disable(R, BSP_StartClock, R, BSP_StopClock) # API functions disable(R, BSP_SetTClockSource) # API functions disable(R, BSP_SchrwareDelay) # API functions disable(R, BSP_SoftwareDelay) # Parameter check enable # Lable user warm start callback (PRE) # Lable user warm start callback (PRE) # Lable user warm start callback (PCST) # User warm start callback function name (PCST) # Watchdog Timer initialize user function name # Watchdog Timer setting user function name	Value Enable (use BSP startup) Disabled Disabled Disabled Disabled Disable Enable Disable Disable Enable Disable Enable Usable Enable E	On

Figure 8-24 BSP Component Setting

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



8.3 Creating the Capacitive Touch Interface

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

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

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

🚱 e ² studio						
File Edit Navigate Search Project	Renesas V	ïews Run	Renesas	AI W	Vindo	ow Help
📓 🛛 ▼ 🗞 ▼ 🖓 🗣 🖗 •	C/C+	+ Concreter		>		
陷 Project Explorer 🛛 🖻 🕏 🍸 💈	Debu	Generator		<i>`</i>	ect_	Example.scfg $ imes$
✓	Partn	er OS		>	atio	on
> 🐝 Binaries	Rene	sas Al		>		
> 🗊 Includes	Rene	sas QE		>	C	CapTouch Gesture Monitor (QE)
> 😂 qe_gen	Smar	t Configurat	or	>	B	CapTouch Board Monitor (QE)
> 📇 src	Solut	ion Toolkit		>	S	CapTouch Pad Monitor (QE)
> 🔁 HardwareDebug	🔌 Rene	sas Software	e Installer		S	CapTouch Multi Status Chart (QE)
> 🦻 QE-Touch					S	CapTouch Parameters (QE)
> 🧀 trash				deo	S	CapTouch Status Chart (QE)
Capacitive_Touch_Project_Exar	nple.rcpc		Int	ueu	S	CapTouch Tuning Result (QE)
Capacitive_louch_Project_Example.scfg			Dre	TOQUC	b	CapTouch Workflow (QE)
Capacitive_Iouch_Project_Example	npie Haro		DIC	Jwse	٨x	Measuring Current Consumption (QE)
> O Developer Assistance			\//	hat'	c N	ew/

Figure 8-25 Cap Touch Workflow (QE) Selection

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

•						—		×
😂 Ca	pTouch Workflow (QE) $ imes$						ē 8	
	Preparation	Tuning	\rangle	Coding	\rangle	Monit	toring	
1.F	Preparation				1			•
	Select a Project							
					く	ح		
$\left[\right]$						v	,	\sim
		10 T						
		acitive_roud	n_Pro	ject_Examp	le			
								~
					Mod	ify Con	figuratior	۱

Figure 8-26 Project Selection



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



Figure 8-27 Create a new configuration

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

Create Configuration of Touch Interfaces	
File Name of Touch I/F: Capacitive_Touch_Project_Example Setup Configuration	Import / Re-edit
Description:	
	Touch I/F *
	Capacitance Type
	Self-Capacitance method v
	Button
	Slider (horizontal)
	Slider (vertical)
	Wheel
	Key pad
	3D Gesture (AI)
	Touch pad
	Shield Terminal
	TC Terminal
	Capacitance Sensor
	Current Sensor
	Diagnosis Terminal
	Remove Touch I/F
Setting	Configurations (Methods) *
Scop rounty Scop resistance value Creat Assigned Tax	
Create	e Cancel Help

Figure 8-28 Touch Interface Layout Area



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

😨 Create Configu	uration of Touch Interfaces	
File Name of Tou	ch I/F: Capacitive_Touch_Project_Example Setup Configuration	Import / Re-edit
Description:		
	Button	Touch I/F *
		Capacitance Type
		Self-Capacitance method 🗸
	Button01	Button
		Slider (horizontal)
		Slider (vertical)
	Rutton02	Wheel
		Key pad
		3D Gesture (AI)
		Touch pad
		Shield Terminal
		TC Terminal
		Capacitance Sensor
		Current Sensor
		Diagnosis Terminal
Setting		Remove Touch I/F
Setup Tou	uch I/F Setup Resistance Value Clear Assigned TSx	Configurations (Methods) *
There are son	me problems with setting	
	ne protono manocangi	
		Create Cancel Help

Figure 8-29 Add Touch Interface (3 buttons)



6. Allocate a name and a touch sensor to each button.

Double click "**Button00**" to display the "Setup Touch Interface" dialog box. Change "Name" and "Touch Sensor" as marked with red in the figure, and then click [**OK**]. For the RL78/G23, allocate TS06 to Button00.

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

TS_B1 TS28 Button01		
TS_B1 TS28 Button01		
TS28 Button01		
Button01		
e Se	etup Touch Interface	X
	Button(self)	
Button02	Name TS_B1	
	Touch SensorResistance[ohm]TS28560]
	OK Cancel H	lelp
	\uparrow	

Figure 8-30 Touch Interface Setting



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

Create Configuration of Touch Integration	erfaces		×
File Name of Touch I/F:	Capacitive_Touch_Project_Example	Setup Configuration	Import / Re-edit
Description:			
	TS_B1		Touch I/F *
	тсэр		Capacitance Type
	1320		Self-Capacitance method v
	TS_B2		Button
			Slider (horizontal)
	TS18		Slider (vertical)
	TS B3		Wheel
			Key pad
	TS00		3D Gesture (AI)
			Touch pad
			Shield Terminal
			TC Terminal
			Capacitance Sensor
			Current Sensor
			Diagnosis Terminal
Catting			Remove Touch I/F
Setup Touch I/F Setu	up Resistance Value Clear Assigned TSx		Configurations (Methods) *
	· · · · · · · · · · · · · · · · · · ·		
			Create Cancel Help

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



- Set up wheel and slider in the same way as the buttons.
 The settings for the application example are shown in the figure below.
 When all touch interfaces are placed in the layout area, click [Setup Configuration].
 - Caution: The error message "There are some problems with setting." will appear under the layout area. The message will disappear when step 9 is completed.
 - Remark: In the application example, the touch sensors allocated to the slider and wheel are switched to the MEC function buttons during low power mode. The name of the touch interface used for touch measurement using the SMS and MEC (low power mode) is TS_xxMEC.

Name of Touch I/F:	Capacitive_Touch_F	oject_Example Setup Configuration	Impor	rt / Re-edit
cription:		N		
S_W2MEC	TS_W1MEC	TS_B1	Touch I/F	*
TS19 Wheel	TS08	T528	Capacitance Type	
			Self-Capacitance method	· · · ·
TS19 TS	308	TS_B2	Button	
			Slider (horizor	ntal)
	TO WALKS	TS18	Slider (vertic	al)
S_W3MEC IS21 IS	IS_W4IMEC	77.00	Wheel	
TS21	TS24	IS_B3	Key pad	
		T500	3D Gesture (/	AI)
	Slider			
>		>	Iouch pad	
TS04 TS05	TS06 TS07	TS01		
TS_S1MEC TS_S2MEC	TS_S3MEC TS_S4MEC	S_S5MEC	IC lermina	1
			Capacitance Se	insor
TS04 TS05	TS06 TS07	TS01	Current Sens	or
			Diagnosis Terr	hinal
***			Remove Touch	n I/F
Eatup Touch I/F	Cotum Registance Value	Clear Accimed TSV	Configurations (Methods)	*
Setup Touch I/F	Setup Resistance Value	Clear Assigned TSx		

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



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

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

- config01: Function setting for touch interface configuration (method) in low power mode.
- config02: Function setting for touch interface configuration (method) in normal mode.
- Caution: When using V1.50 of the CTSU module, the auto Judgement (SMS) can only be applied to config01.

Setup Configurations (Methods)			×
Add Configuration	ration		
N	config01	config02	
TS_W1MEC(self)	✓ Available		
TS_W2MEC(self)	✓ Available		
TS_W3MEC(self)	✓ Available		
TS_W4MEC(self)	✓ Available		
TS_S1MEC(self)	✓ Available		
TS_S2MEC(self)	✓ Available		
TS_S3MEC(self)	✓ Available		
TS_S4MEC(self)	✓ Available		
TS_S5MEC(self)	✓ Available		
TS_B1(self)	✓ Available	✓ Available	
TS_B2(self)	✓ Available	✓ Available	
TS_B3(self)	✓ Available	✓ Available	
Wheel(self)		✓ Available	
Slider(self)		✓ Available	
Auto Judgement by Hardware (SMS)	🗹 Enable	Enable	
Multiple Electrode Connection	🗹 Enable	Enable	
	ОК	Cancel Help	

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



10.Confirm there is no error displayed, then click [**Create**] in the "Create Configuration of Touch Interface" window.

This completes the creation of the touch interface for RL78/G22.

le Name of Touch I/F:	Capacitive_Touch_Project_Example Setup Configuration	Import / Re-edit
escription:		
TS_W2MEC	TS_W1MEC TS_B1	Touch I/F
TS19 Wheel	T508 T528	Capacitance Type
		Self-Capacitance method
TS19 TS0	8 TS_B2	Button
		Slider (horizontal)
S WRMEC TOOL TO	TS18	Slider (vertical)
	TS B3	Wheel
TS21	TS24	Key pad
	TS00	3D Gesture (AI)
	Slider	Touch pad
TS04 TS05	T506 T507 T501	Shield Terminal
\sim		TC Terminal
TS_S1MEC TS_S2MEC	TS_S3MEC TS_S4MEC TS_S5MEC	Capacitance Sensor
TS04 TS05	TS06 TS07 TS01	Current Sensor
		Diagnosis Terminal
		Remove Touch I/F
Setup Touch I/F	atun Resistance Value Clear Assigned TSy	Configurations (Methods)
Setup Touch I/F S	tup Resistance Value Clear Assigned TSx	Create Cancel

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



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

Caution: RL78/G22 and RL78/G23 allocate different touch sensors to electrodes.

Remark: The name of the touch interface used for touch measurement using the SMS (low power mode) is TS_xxBTN.

ile Name of Touch I/F: Capacitive_Touch_Project_Example Setup Configuration	Import / Re-edit
escription:	
TS_W2BTN TS_W1BTN TS_B1	Touch I/F *
Wheel	Capacitance Type
1527	Self-Capacitance method v
T527 T526 T5_82	Button
	Slider (horizontal)
TS W2RTN TC 20 TC 21 TS W4RTN	Slider (vertical)
	Wheel
T528 T521	Key pad
TS07	3D Gesture (Al)
Slider	Touch pad
T\$15 T\$14 T\$13 T\$12 T\$01	Shield Terminal
	TC Terminal
TS_STBIN TS_S2BIN TS_S3BIN TS_S4BIN TS_S5BIN	Capacitance Sensor
TS15 TS14 TS13 TS12 TS01	Current Sensor
	Diagnosis Terminal
a dia mandri ang	Remove Touch I/F
Satur Touch 1/5 Satur Becictarse Value Clear Accioned TSv	Configurations (Methods) *

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



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

- config01: Function setting for touch interface in low power mode.
- config02: Function setting for touch interface in normal mode.
- Caution: When using V1.50 of the CTSU module, the auto Judgement (SMS) can only be applied to config01.

Setup Configurations (Methods)			×
Add Configuration	ration		
	Config01	config02	
TS_W1BTN(self)	✓ Available		
TS_W2BTN(self)	✓ Available		
TS_W3BTN(self)	✓ Available		
TS_W4BTN(self)	✓ Available		
TS_S1BTN(self)	✓ Available		
TS_S2BTN(self)	✓ Available		
TS_S3BTN(self)	✓ Available		
TS_S4BTN(self)	✓ Available		
TS_S5BTN(self)	✓ Available		
TS_B1(self)	✓ Available	✓ Available	
TS_B2(self)	✓ Available	✓ Available	
TS_B3(self)	✓ Available	✓ Available	
Wheel(self)		✓ Available	
Slider(self)		✓ Available	
uto Judgement by Hardware (SMS)	🗹 Enable	Enable	
_			
l	ОК	Cancel Hel	р

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



13.Confirm there is no error displayed, then click [**Create**] in the "Create Configuration of Touch Interface" window.

This completes the creation of the touch interface for the RL78/G23.

New York Int	Connective Touch Proj	ant Furnalis	
Name of Touch I/F:	Capacitive_Ioucn_Proj	Setup Configuration	Import / Re-edit
cription:			
5_W2BTN	TS_W1BTN	IS_B1	Touch I/F
TS27 Wheel	TS26	TS06	Capacitance Type
			Self-Capacitance method
TS27 TS26		rs_B2	Button
			Slider (horizontal)
	/	T505	Slider (vertical)
_W3BIN TS28 TS21	IS_W4BIN		Wheel
TS28	TS21	IS_B3	Key pad
		TS07	3D Gesture (Al)
	Slider		Touch as d
	>	>	fouch pad
1515 1514	1513 1512 15	.01	
TS_S1BTN TS_S2BTN TS	S_S3BTN TS_S4BTN TS	_S5BTN	
			Capacitance Sensor
TS15 TS14	TS13 TS12	1501	Current Sensor
			Diagnosis Terminal
ing			Remove Touch I/F
Setup Touch I/F Setu	p Resistance Value CI	ear Assigned TSx	Configurations (Methods)

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

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

e								—		×
bc	apTouc	h Tuning Resu	It (QE) $ imes$						8	- 0
Tuni	ng Ge	esture								
Тог	ich I/E (Configuration	Capaciti	vo Touch Project Example						
100		configuration.	_capaciti	ve_louch_project_example						
Me	ethod	Kind	Name	Touch Sensor	Parasitic Capacitance[pF]	Sensor Drive Pulse Frequency[MHz]	Threshold	Scan Time[ms]	Overflow	^
со	nfig01	Button(self)	Mec00	TS00	-	-	-	-	None	
со	nfig02	Button(self)	TS_B1	TS28	-	-	-	-	None	
со	nfig02	Button(self)	TS_B2	TS18	-	-	-	-	None	
со	nfig02	Button(self)	TS_B3	TS00	-	-	-	-	None	
со	nfig02	Wheel	Wheel	TS19, TS08, TS24, TS21	-	-	-	-	None	
со	nfig02	Wheel TS	(Wheel)	TS19	-	-	-	-	-	
со	nfig02	Wheel TS	(Wheel)	TS08	-	-	-	-	-	
со	nfig02	Wheel TS	(Wheel)	TS24	-	-	-	-	-	
со	nfig02	Wheel TS	(Wheel)	TS21	-	-	-	-	-	
со	nfig02	Slider	Slider	TS04, TS05, TS06, TS07, TS01	-	-	-	-	None	
со	nfig02	Slider TS	(Slider)	TS04	-	-	-	-	-	
со	nfig02	Slider TS	(Slider)	TS05	-	-	-	-	-	
со	nfig02	Slider TS	(Slider)	TS06	-	-	-	-	-	
со	nfig02	Slider TS	(Slider)	TS07	-	-	-	-	-	
со	nfig02	Slider TS	(Slider)	TS01	-	-	-	-	-	\sim

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



15.Click $\overset{\frown}{}$ on the top left in e² studio to build the project.



Figure 8-39 Project Build

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

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





Properties for Capacitive_Touch_Project_Example							
type filter text	Settings						
 Resource Builders C/C++ Build Build Variables Environment Logging Settings Stack Analysis Tool Chain Editor 	 SMS Assembler Source Object User Common CPU Device Miscellaneous 	 Use standard/mathematical libraries (-library) Use C99 edition libraries (-library) Check memory smashing on releasing memory (-library) Use runtime libraries (-library) Relocatable files, object files and library files (-input/-library/-binary) 					

Figure 8-41 Set Runtime Libraries



8.4 Changing the Debug Configuration for Capacitive Touch Sensor Tuning

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

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



Figure 8-42 Select "Debug Configurations"

- 2. Select the [**Debugger**] [**Connection Settings**] tabs in "Debug Configurations" window. Specify the connection settings with the target board as shown in the figure below.
 - Caution: 1. In the application example, power is supplied to the target board from the emulator for easy confirmation of the operation. The power can also be supplied to the target board via E2 emulator Lite from the USB port on the PC, however we recommend using power generated by the target board.
 - 2. Available debugging methods vary depending on the specifications of the target board. Set up the required items by selecting "Debug hardware:" according to the debug method you are using.

For example, COM port debugging requires a different setting.

— Debugging method: E2 Lite

Debug Configurations					
Create, manage, and run configurations					
Erase Flash on Start is Enabled. Please Disable this option after s	ucessful connection.				
	Name: Capacitive_Touch_Project_Example HardwareDebug				
type filter text	📄 Main 🕸 Debugger Common 🧤 Sou	ırce			
C/C++ Application		275102.005			
C/C++ Remote Application	Debug hardware: E2 Lite (RL78) Target Device: R7F102GGE				
EASE Script					
GDB Hardware Debugging	GDB Settings Connection Settings				
💽 GDB Simulator Debugging (RH850)	V Clock				
🚭 Launch Group	Main Clock Frequency[MHz]	Using Internal Clock			
✓ C [™] Renesas GDB Hardware Debugging	Sub Clock Frequency[kHz]	Using Internal Clock			
Capacitive_Touch_Project_Example HardwareDebug	Monitor Clock	System			
Renesas Simulator Debugging (RX, RL78)	 Connection with Target Board 				
	Emulator	(Auto)			
	Low voltage OCD board	No			
	Power Target From The Emulator (MAX 200mA)	Yes			
	Supply Voltage[V]	3.3			
	Hot Plug	No			

Figure 8-43 Debug Configuration (E2 Lite) Settings



— Debugging method: COM port

Main 🕸 Debugger 🕨 Startur	o 🎼 Sou	rce 🔲 Common	
ebug hardware: COM Port (RL	78)	Target Device: R7F100GSN	
GDB Settings Connection Setti	ngs Deb	ug Tool Settings	
✓ Clock			
Main Clock Frequency[MI	Hz]	Using Internal Clock	~
Sub Clock Frequency[kHz]	Using Internal Clock	~
Monitor Clock		System	~
 Connection with Target Boar 	d		
COM Port		COM9	
Reset control pin		DTR	~
. Flash			

Figure 8-44 Debug Configuration (COM port) Settings

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

Debug Configurations		
Create, manage, and run configurations		
Erase Flash on Start is Enabled. Please Disable this option after s	sucessful connection.	
🗅 🖻 🐌 🗎 🗶 🖻 🏹 🗸	Name: Capacitive_Touch_Project_Example HardwareDebug	
type filter text	📄 Main 🏇 Debugger 🕨 Startup 🔲 Common 🦆 So	urce
C/C++ Application C/C++ Remote Application	Debug hardware: E2 Lite (RL78) V Target Device:	R7F102GGE
C GDB Hardware Debugging	GDB Settings Connection Settings Debug Tool Settings	
💽 GDB Simulator Debugging (RH850)	✓ IO	- 1
🚭 Launch Group	Use Default IO Filename	Yes
✓ C Renesas GDB Hardware Debugging	IO Filename	\${support_area_loc}
Capacitive_Touch_Project_Example HardwareDebug	✓ General Debug	
Renesas Simulator Debugging (RX, RL78)	Reset After Reload	Yes
	✓ Break	
	Stop emulation of timer group when stopping	No
	Stop emulation of serial group when stopping	No
	✓ Mask For Input Signal	
	Mask Target Reset Signal	No
	Mask Internal Reset Signal	No
	✓ Memory	
	Verify On Writing To Memory	Yes
	Allow to access by stopping execution while running	ng Yes
1	✓ Start/Stop Function Setting	

Figure 8-45 Debug Tool Settings



4. Select the "**Startup**" tab. Confirm that "Set breakpoint at:" and "Resume" are selected as shown in the figure below, then click [**Apply**].

This completes the debug configuration for tuning.

Debug Configurations	—
Create, manage, and run configurations	TO T
Image: Constraint of the second se	Name: Capacitive_Touch_Project_Example HardwareDebug Main
Filter matched 10 of 12 items	Runtime Options Set program counter at (hex): Set breakpoint at: main Resume Run Commands Revert Apply
0	Debug Close

Figure 8-46 Runtime Options Settings



8.5 Capacitive Touch Sensor Tuning with QE for Capacitive Touch

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

- 1. Click [Start Tuning] in "[CapTouch Main (QE)] / [CapTouch Workflow (QE) (for QE V3.2.0 or later)]" to start automatic tuning.
 - Note: In the application example, power is supplied to the target board from the emulator for easy confirmation of the operation. The power can also be supplied to the target board via E2 emulator Lite from the USB port on the PC, however we recommend using power generated by the target board for tuning.

•						×	
SapTouch Workflow (QE) $ imes$,		ê :		
	Tuning	Coding	\rangle	Monit	oring		
2.Tuning Touch S	ensors					•	
Start Tuning (End	nulator)						
	Start Tur Display Tunin	ning g Result		dvanc	ed mo	de	

Figure 8-47 Start Automatic Tuning

 When the debug session is started, e² studio may display the message to switch to the debug perspective. If the message is displayed, check [Remember my decision(R)], then click [Switch] to continue the debug session and automatic tuning for QE for Capacitive Touch.

💽 Conf	Confirm Perspective Switch					
2	This kind of launch is configured to open the Debug perspective when it suspends.					
	This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints.					
	Switch to this perspective?					
☑ <u>Remember my decision</u>						
		<u>S</u> witch <u>N</u> o				

Figure 8-48 Confirm Perspective Switch



3. Automatic tuning for QE for Capacitive Touch is started. Continually confirm the "Automatic Tuning Processing" dialog box that guides the tuning process.

The following is an example of the dialog box. Normally, the initial tuning processing will not require any action on the part of the user.



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

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



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

Repeat the same procedure for all configured buttons.



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

Automatic Tuning Processing							×		
The automatic tuning process is now complete. If overflow or warning/errors are indicated, those sensors can be retried. If there are continued overflows or warning/errors, please consult the Renesas application notes for Capacitive Touch for guidance.									
Select the target	Method	Kind	Name	Touch Sensor	Threshold	Overflow	Warning / Error		
	config01	Button	Mec00	TS00	866				
	config02	Button	TS_B1	TS28	1310				
	config02	Button	TS_B2	TS18	1245				
	config02	Button	TS_B3	TS00	1057				
	config02	Wheel	Wheel	TS19, TS08, TS24, TS21	1162				
	config02	Slider	Slider	TS04, TS05, TS06, TS07, TS01	695				
Retry Continue t	the Tuning	Process							
							Cano	cel Help	

Figure 8-51 Tuning Results Display

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

	config02	Wheel	Wheel		
	config02	Slider	Slider		
Retry Continue the Tuning Process					

Figure 8-52 Continue the Tuning Process

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

2			- 🗆	×
CapTouch Workflow (QE) ×			🔊 i =	, 🗆
🖻 🥝 Preparation	Tuning	Coding	Monitoring	
1.Preparation		To Output Parameter F	iles	â
📀 Select a Project		·		-1
Prepare a Configuration	Output parameter files from	a tuning result.	_	
2.Tuning Touch Sensors	2	Output Parameter Files		
Start Tuning (Emulator)			Specify an output folder	
✓ Start Tuning (Serial)		(1)	Use an external trigger	
Output Parameter Files			Use diagnostic code	
3.Coding	·		Use API compatilibity mode	э
Implement Program	The result of tuning is out	put as a parameter file and can b	be read by touch middleware.	
4.Monitoring	Please make sure to output the file when tuning is performed and when the touch interface			
Start Monitoring (Emulator)	configuration is changed.			
Start Monitoring (Serial)				×

Figure 8-53 Output Parameter Files



7. Confirm that **qe_touch_config.c**, **qe_touch_config.h**, **and qe_touch_define.h** have been added in the "**Project Explorer**" window.

These files include the tuning information required to enable touch detection using the driver.

陷 Project Explorer 🛛 🕹		🖻 🕏 🍸	000	' 🗆			
✓ ☐ Capacitive_Touch_Project_Example							
> 🐝 Binaries	> 🞇 Binaries						
> 🗊 Includes	> 🔊 Includes						
Ƴ 😕 qe_gen							
> 🖻 qe_touch_c	onfig.c						
> 🖻 qe_touch_config.h							
> h qe_touch_d	lefine.h						
> 冯 src							
> 🔁 HardwareDeb	ug						
> 🗁 QE-Touch							
> 🗁 trash							
Capacitive_Touch_Project_Example.rcpc							
🌼 Capacitive_Tou	uch_Project_Example.s	scfg					

Figure 8-54 Tuning Parameter File Output

8. Click so on the top left of e² studio to build the project. In the "Console" window, confirm that no error has occurred during the build.

This completes the tuning of the capacitive touch sensor using QE for Capacitive Touch.



8.6 Adding API Call for rm_touch Middleware

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

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

•	_		×	
SapTouch Workflow (QE) $ imes$		r 🖏		
OPreparat Tuning Coding	Мо	nitorir	ng	•
3.Coding			•	
Implement Program				
Show Sample				

Figure 8-55 Sample Code Example Display

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

💽 Show Sample Code	×				
Sample code of main() function:					
V*************************************	^				
* FILE : ge sample main.c					
* DESCRIPTION : Main Program for RL78					
* * NOTE:THIS IS A TYPICAL EXAMPLE.					
*					

#include de_touch_conlig.n					
/* TODO Support for SMS function */					
/* include related modules. */					
//#include "Config ITI 000.h"					
//#include "Config_INTC.h"					
//#include "r_cg_itl_common.h"					
#define TOUCH_SCAN_INTERVAL_EXAMPLE (20 * 1000) /* microseconds */					
void R_CTSU_PinSetInit(void);					
void qe_touch_main(void);					
void qe_touch_delay(uint16_t delay_us);					
void ae sms init(void):	>				
Copy to the Clipboard Output to a File Show the Application	Note				
OK H	elp				

Figure 8-56 Sample Code Output to File



3. Confirm that "qe_touch_sample.c" file is generated in "Project Explorer".



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

4. Open "Capacitive_Touch_Project_Example.c".



Figure 8-58 Select Capacitive_Touch_Project_Example.c



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

Capacitive_Touch_Project_Example.c ×
2⊕ * DISCLAIMER.
19
21⊕ * File Name : Capacitive_Touch_Project_Example.c.
<pre>26 #include "r_smc_entry.h"</pre>
27
28 int main (void);
<pre>29 void qe_touch_main(void);</pre>
30
31⊖ int main(void)
32 {
33 EI();
34
35 /* QE sample program */
36 qe_touch_main();
37
38 return 0;
39 }

Figure 8-59 Set Call for qe_touch_main() Function

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

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



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



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



Figure 8-61 Initial Offset Tuning Processing for Config02



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



Figure 8-62 Add External Trigger Setting for Config02 Measurement



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



Figure 8-63 Specify qe_sms_init Function

Note: 1. For RL78/G23, leave "ELSELR10 = 0x06U;" as commented out. Note: 2. For RL78/G23, uncomment the following code: ELISEL7 = 0x17U; ELL1SEL0 = 0x08U; ELL1LNK0 = 0x01U; ELOSEL6 = 0x01U;

ELOENCTL = 0x40U;



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

🚺 qe_	touch_sample.c ×	
245		
246	/* Trigger start function function for SMS support */	
247	void qe sms trigger start(void)	
248		
249	/* TODO Support for SMS function (RL78/G22 only) */	
250	/* start external trigger for SMS */	
251	/* As an example, this is the code when using an external trigger. Please modify the below code. $*/$	
252	R_Config_INTC_INTP5_Start();	
253		
254 🖉	/* TODO Support for SMS function	*/
255	/* The sample code is an example of code when using an interval timer.	*/
256	/* Clear timer interrupt flag for trigger to CTSU scans	*/
257	/* As an example, this is the code when using an interval timer. Please modify the code according to the timer used.	*/
258	ITLS0 &= (uint8_t)~_01_ITL_CHANNEL0_COUNT_MATCH_DETECTE;	
259		
260	/* TODO Support for SMS function	*/
261	/* The sample code is an example of code when using an interval timer.	*/
262	/* Start the timer trigger to CISU scans	*/
263	/* As an example, this is the code when using an interval timer. Please modify the code according to the timer used.	*/
264	R_Config_IIL000_Start();	
265	}	
266	/* Tringer shere furshing for CMC surgerst */	
207	/* Intgger stop function function for SmS support */	
208	vola de_sms_trigger_stop(vola)	
209	1 /* TODO Support for SMS function	*/
270	/* Tobo Support for SmS function	*/
271	/ The sample code is an example of code with using an interval time.	*/
273	/* As an example this is the code when using an interval timer. Please modify the code according to the timer used	*/
274	R Config T1190 Story).	<i>'</i>
275	·····-==···	
276	/* TODO Support for SMS function (RL78/G22 only) */	
277	/* stop external trigger for SMS */	
278	/* As an example, this is the code when using an external trigger. Please modify the below code. $*/$	
279	R_Config_INTC_INTP5_Stop(); Note2	
280	}	
281		

Figure 8-64 Specify Functions qe_sms_trigger_start and qe_sms_trigger_stop

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

11.Click so the top left of e² studio to build the project. Confirm that no error occurred during the build in the "Console" window.

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



9. Documents for Reference

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

Technical Updates/Technical Brochures

(The latest versions of the documents are available on the Renesas Electronics Website.)

Application Note RL78 Family Capacitive Touch Sensing Unit (CTSU2L) Operation Explanation (R01AN5744)

Application Note RL78 Family CTSU Module Software Integration System (R11AN0484) Application Note RL78 Family TOUCH Module Software Integration System (R11AN0485) Application Note Capacitive Sensor Microcontrollers

CTSU Capacitive Touch Electrode Design Guide (R30AN0389) Application Note RL78/G23 Capacitive Touch Low Power Guide (SMS function) (R01AN6670) Application Note RL78/G22 Capacitive Touch Low Power Guide (SMS / MEC function) (R01AN6847) (The latest versions of the documents are available on the Renesas Electronics Website.)

Website and Support

Renesas Electronics Website

http://www.renesas.com/

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

Inquiries

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



Revision History

Description		Description	
Rev.	Date	Page	Summary
1.00	Jul.22.24	-	First edition issued



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

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

1. Precaution against Electrostatic Discharge (ESD)

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

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

- 1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- 2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
- 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
- 5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
- Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
 - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

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

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

- 7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
- 8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas
- Electronics products. (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled
- (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.