

## Using QE for Capacitive Touch for IAR EWRX

#### Introduction

This application note explains the steps involved in automatically adjusting and monitoring a touch sensor in a project of the IAR Embedded Workbench for Renesas RX (IAR EWRX) generated by the Smart Configurator with the use of the standalone version of QE for Capacitive Touch (standalone-version QE), which is a tool for development involving capacitive touch sensors.

#### **Target Evaluation Kit**

RSK-RX140 (RTK551406BS00000BE): CPU board (R5F51406BDFN)

#### **Operating Environment**

IDE	IAR Embedded Workbench for Renesas RX version 4.20.3 or later				
Configuration tool	RX Smart Configurator V2.15.0 or later				
	RX family RX Driver Package Ver.1.36 or later				
	Firmware Integration Technology (FIT) for RX family				
FIT module	<ul> <li>CTSU QE API (r_ctsu_qe): V2.10 (or later)</li> </ul>				
	<ul> <li>SCI Driver (r_sci_rx): V4.5 (or later)</li> </ul>				
	<ul> <li>Touch QE API (rm_touch): V2.10 (or later)</li> </ul>				
Toolchain	IAR EWRX Toolchain				
QE	Standalone version of QE for Capacitive Touch V3.1.0 or later				

Note: Download tools from the following URLs in advance.

- URL for downloading the IAR Embedded Workbench for Renesas RX
   <u>https://www.iar.com/products/architectures/renesas/iar-embedded-workbench-for-renesas-rx/</u>
- URL for downloading QE for Capacitive Touch
   <u>https://www.renesas.com/us/en/software-tool/qe-capacitive-touch-development-assistance-tool-capacitive-touch-sensors#download</u>
- URL for downloading the RX Smart Configurator
   <u>https://www.renesas.com/jp/en/software-tool/rx-smart-configurator#download</u>
- URL for downloading the RX driver package
   <u>https://www.renesas.com/jp/en/software-tool/rx-driver-package</u>
- RX Smart Configurator User's Guide: IAREW https://www.renesas.com/us/en/document/mat/rx-smart-configurator-users-guide-iarew
- Renesas Starter Kit for RX140 User's Manual
   <u>https://www.renesas.com/us/en/document/mat/renesas-starter-kit-rx140-users-manual</u>
- RX140 Group User's Manual: Hardware Rev.1.10 <u>https://www.renesas.com/us/en/document/mah/rx140-group-users-manual-hardware-rev110</u>



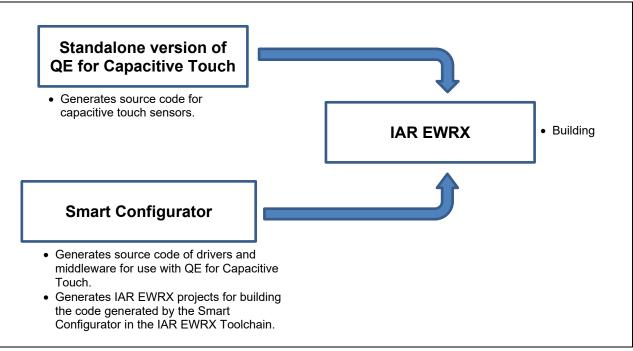


Figure 1 Configuration for Using QE for Capacitive Touch with IAR EWRX



## Contents

1.	Installation	4
1.1	Installing the IAR EWRX	4
1.2	Installing the Smart Configurator	4
1.3	Installing the Standalone Version of QE for Capacitive Touch	4
2.	Preparing a Project	4
2.1	Creating a Configuration File	5
2.2	Setting the Smart Configurator	8
2.3	Generating Code	15
2.4	Setting the Project Folder and Target MCU	19
3.	Preparing the Touch Interfaces	20
3.1	Creating a Touch Interface Configuration	20
3.2	Outputting Files for Tuning Touch Sensors	23
3.3	Implementing Programs for Tuning	25
3.4	Building a Project	27
4.	Tuning Touch Sensors	31
4.1	Connecting the RSK-RX140 Board with a PC	31
4.2	Starting Tuning	32
4.3	Acquiring the Tuning Result	35
5.	Coding and Monitoring	37
5.1	Monitoring	37
5.2	Disconnecting Serial Communications	40
Rev	/ision History	42



#### 1. Installation

#### 1.1 Installing the IAR EWRX

After starting the downloaded installer, install the IAR EWRX according to the procedure prompted by the installer.

Log in with administrator rights to proceed with installation.

#### **1.2 Installing the Smart Configurator**

After starting the downloaded installer, install the Smart Configurator according to the procedure prompted by the installer.

Log in with administrator rights to proceed with installation.

### **1.3 Installing the Standalone Version of QE for Capacitive Touch**

The downloaded zip file has both the plugin version of QE for Capacitive Touch and the standalone version of QE for Capacitive Touch.

- 1. Expand the downloaded zip file.
- 2. Since the standalone-version QE for Capacitive Touch is stored in the QE-CapTouch folder, move that folder to a desired location. The QE-CapTouch folder must be moved to a folder that is close to the root of the drive so that the Windows path name does not exceed the limit on the number of characters (260 characters).

— Examples of folder name: C:¥Renesas, C:¥QeTouch

Note that the tool can be launched from "qe-captouch.exe" that is in the "QE-CapTouch¥eclipse" folder.

### 2. Preparing a Project

Using the Smart Configurator, create an IAR EWRX project that uses the standalone-version QE. The Smart Configurator has a simple wizard for creating projects. An IAR EWRX project can be created by specifying the location, file name, platform, toolchain, and project type of a project, making the RTOS setting, and setting the locations of FIT modules. Follow the procedure below to create a new IAR EWRX project.

Sections 2.1 to 2.3 explain the procedure of <u>To Create Project</u> in the workflow diagram of the standalone-version QE.

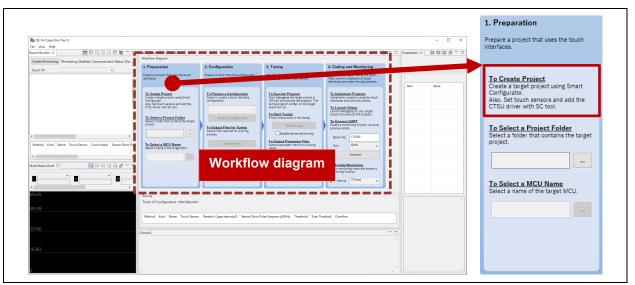


Figure 2-1 Standalone-Version QE - Workflow Diagram: To Create Project



#### 2.1 Creating a Configuration File

 From the Windows start menu, select [Renesas Electronics Smart Configurator] and then [Smart Configurator for RX Vx.x.x]. The main window of the Smart Configurator will be started. Note: Vx.x.x should be replaced with the version in use.

Smart Configurator	– 🗆 X
File Window Help	
	E   🖉
Console ×	
No consoles to display at this time.	

#### Figure 2-2 Starting the Smart Configurator

2. Select [New...] from the [File] menu or click on the 📩 [New Configuration File] button on the main toolbar.

<b>₹</b> Ŝ Si	Smart Confi	igurator
File	Window	Help
<u> 1</u>	New	
C	Open	
	Save	Ctrl+S
F	Restart	
I	Exit	

Figure 2-3 Creating a New Smart Configuration File



- 3. The dialog box for making Smart Configurator settings is displayed.
  - Set the following items.
    - (1) [Platform:] Enter "RX140" and select "RSKRX140 (R5F51406BxFN)" under "Board" that is displayed after filtering has finished.
    - (2) [Toolchain:] Select "IAR EWRX Toolchain".
    - (3) [File name:] Enter the name of the project to be created.
    - (4) [Location:] Confirm the folder in which the project is to be created. To change the project folder, click on [Browse...] and select the folder for saving the project.

--- Click on [Next >] and then set the items for the use of FIT modules and middleware.

Kew Si	mart Configuration	File		-	- C	) ;	×
Smart Co	nfiguration Setti	ngs					
Select pla	tform and toolchai	n settings for the	e new configura	ation file			
Category:	RX						$\sim$
Platform:			Toolchain:				
RX140	(1)	×		RXC Toolcha	<sup>in</sup> (2)		
Tarı ✓ Device ✓ RX	KRX140 (R5F51406B getBoardRX140 (R5 e			RX Toolchain			
	more boards						
ROM size	: 256KB, RAM size: (	54KB, Pin count:					
File name:	IAR_Touch_projec	t	(3)				
Location:	C:¥Work¥IAR_EW	RX_Touch¥RSK_F	X140	(4)		Browse	
	< ]	<u>B</u> ack <u>N</u>	ext >	<u>F</u> inish	C	ancel	

Figure 2-4 Smart Configurator Settings (1)



 Confirm that [RTOS:] in [RTOS Settings] is "None" and set the location of the FIT modules. Unzip the downloaded FIT module zip file. Click on [Browse...] and select the folder containing the unzipped FIT modules.

	New Smart Configuration File	– 🗆 X
Sma	rt Configuration Settings	
Cor	figure FIT module location and RTOS settings for the new	
cor	figuration file	
RT	OS Settings	
RT	OS: None	~
RT	OS Version:	$\sim$
		Manage RTOS Versions
- FIT	module location	
	¥Renesas¥RX¥r01an6515xx0136-rx-fit¥FITModules	Browse
	#Reflesas#RA#f01af16515xx0156-fx-fit#F11W0dules	browse
	< <u>B</u> ack <u>N</u> ext >	Einish Cancel

Figure 2-5 Smart Configurator Settings (2)

— Clicking on [Finish] creates a configuration file (\*.scfg) and displays the main window.

File Window Help					E 🛃
			🔓 MCU/MI	PU Package 🗙	
Overview information		1	<b>a</b>		
		Generate Code	Generate Report	N 🔎 🔎 🖻 🖻	>>
✓ General Information			0		
Get an <u>overview</u> Get an <u>overview</u> of the features provided by S Configurator.	inart	components			UI,
Videos Introduction to Smart Configurator Browse related videos	RTOS	Components Middleware & Drivers Device Drivers			
What's New Check out <u>what's new</u> in the latest release.		ardware			P           P
✓ Current Configuration					
Selected board/device: R5F51406BxFN (ROM size: 256KB, R	AM size: 64KB, Pin count: 80)				
Generated location (PROJECT_LOC¥): src¥smc_gen		Edit			
Selected components:					
Component	Version Configur				
Board Support Packages. (r_bsp)     Overview Board Clocks System Components Pins Interru	7.20 r_bsp(use	ed)	✓ Legend		
			, cegena		
Console ×  Smart Configurator Output				🗎 🔝 🛃 📑	! * <mark>™</mark> * <sup>™</sup> E
Smart Configurator Souput MoS9000001 Pin 12 is assigned to EXTAL MOS900001 Pin 7 is assigned to XCIM MOS9000016: Pin 8 is assigned to XCUM MOS9000016: Pin 7 is assigned to KCIM MOS900016: Pin 7 is assigned to XCIM					ŕ
M05000016: Pin 8 is assigned to XCOUT M05000016: Pin 10 is assigned to XTAL M05000016: Pin 12 is assigned to XTAL					

Figure 2-6 Main Window of Smart Configurator



#### 2.2 Setting the Smart Configurator

When a tab at the bottom of the Smart Configurator view is selected, the displayed contents are switched. Select a tab as required and set the necessary items.

Overview Board Clocks System Components Pins Interrupts
---

#### Figure 2-7 Tabs of the Smart Configurator View

#### 1. Setting clocks

Select the [Clocks] tab.

	Overview Board	Clocks	System	Components	Pins	Interrupts	
--	----------------	--------	--------	------------	------	------------	--

Figure 2-8 Selecting the [Clocks] Tab

- Main clock: Tick the checkbox ( $\sqrt{}$ ).

Enter values so that the frequency of FCLK, ICLK, and PCLDK is 48 MHz, and the frequency of PCLKB is 24 MHz, as shown in Figure 2-9.

AR_Touch_project.scfg ×		ت این این میں
CC: 5.0 (V) (Actual value: 5)		
Main clock PLL circuit Frequency Division: x1		
requency: 8 (MHz) Vait time: 8192 V 2048 (µs)	SCKCR (FCK[3:0])	FlashiF clock (FCLK)
	x1 SCKCR (ICK[3:0]) x1	48 (MHz) System clock (ICLK) 48 (MHz)
Sub-clock           requency:         32.768	SCKCR (PCKB[3:0]) x1/2	Peripheral module clock (PCLKB)
scillator drive capacity: Low CL	SCKCR (PCKD[3:0])	Peripheral module clock (PCLKD) 48 (MHz)
HOCO clock requency: 48 (MHz)		×
Equency. (mix)	x 1/4	Low power timer clock (LPTCLK)
LOCO clock		
requency: 4 (MHz)	CKOCR (CKODIV[2:0])	CLKOUT pin
CTSU output clock		
IWDT-dedicated low-speed clock		IWDTCLK/CACILCLK

#### Figure 2-9 Setting Clocks



## 2. Setting the system

Select the [System] tab.

Overview Board C	locks System	Components	Pins	Interrupts	

Figure 2-10 Selecting the [System] Tab

#### - Confirm that "FINE" is selected in [Debug interface setting].

*IAR_Touch_project.scfg × System configuration		Generate Code Generate Report	
<ul><li>☑</li><li>✓ On-chip debug setting</li></ul>			
Debug interface setting O Unused	● FINE		

Figure 2-11 Setting the System

3. Setting software components Select the [Components] tab.



Figure 2-12 Selecting the [Components] Tab



— Add a driver, middleware, and serial interface and set them up.

Figure 2-13 Setting Software Components

— To add a component, click on the 🔪 [Add component] button to open the "New Component" dialog box and select the component to be added.

Software component configuration       Components       Components       Components       Type       Version       Image: Components       <	Smart Configurator         File       Window         Help         Image: Smart Configurator         Im	New Component Software Component Selection Select component from those available in list Category All Function All Filter		- • ×	
	type filter text       v > Startup       v > Generic	B-Bit Timer     Board Support Packages.     Bayes     Board Support Packages.     Byte-based circular buffer library.     Clock Frequency Accuracy Measurement.     Clock Synchronous Control Module for     Clock Synchronous Control Module for     Comparator     Show only latest version     Hide items that have duplicated functionali     Description     This software component generates two units     that comprise two 8-bit counter channels, tot     Download the latest FIT drivers and middleware     Configure general settings	Code Gene r_bsp Firmware I Code Gene r_byteq Firmware I Code Gene r_eeprom_spi Firmware I r_flash_spi Firmware I Code Gene y (unit 0, unit 1) of an on-chip 8-bit time aling four channels.	erator 1.10.0 Integr 7.20 erator 1.11.0 Integr 2.00 erator 1.11.0 Integr 3.02 Integr 3.03 Integr 5.20 erator 1.5.0 ¥	e (

Figure 2-14 Displaying the "New Component" Dialog Box



2.10

#### a. Adding a CTSU driver

Open the "New Component" dialog box, select "r\_ctsu\_qe", and click on [Finish].

r\_ctsu\_qe

```
🕀 CTSU QE API
```

Firmware Integration Technology

#### Figure 2-15 Selecting "r\_ctsu\_qe"

• Click on the added "r\_ctsu\_qe" module and enable the TSCAP pin and the two TS pins to be used. For the assignment between the TS pins and touch sensors, confirm the user's manual of the target board in use.

		Genera	ate Code Generate Repo
Components 🚵 🗳 📮 🗄 茸	Configure		(i
۵	Property	Value	^
type filter text	# Interrupt level for INTCTSURD	Level 2	
V 🗁 Startup	# Interrupt level for INTCTSUFN	Level 2	
V C Generic	✓ I Resources		
er bsp	✓ I CTSU		
✓ ➢ Drivers	🛰 TSCAP Pin	🗹 Used 🛛 🦟 🖊	2)
✓ ➢ Capacitive Touch	🛰 TSO Pin	Used .	
clipitate issue	🛰 TS1 Pin	Used Used	
15contde	🛰 TS2 Pin	Used Used	
	🛰 TS3 Pin	Used 📃	
	🛰 TS4 Pin	Used 📃	
	🛰 TS5 Pin	Used Used	
	🛰 TS6 Pin	Used Used	
	🛰 TS7 Pin	Used 📃	
	🛰 TS8 Pin	Used 📃	
	🛰 TS9 Pin	Used Used	
	🛰 TS10 Pin	Used	_
	🛰 TS11 Pin	🔽 Used	3)
	🛰 TS12 Pin	Used	<b>)</b>
	🛰 TS13 Pin	Used	
	🛰 TS14 Pin	Used Used	
	🛰 TS15 Pin	Used 📃	
	🛰 TS16 Pin	Used 📃	~
	Macro definition: CTSU_CFG_PARAM_CHECKING_ENABL Setting to BSP_CFG_PARAM_CHECKING_ENABLE utilizes : Setting to 1 includes parameter checking; 0 compiles out	he system default setting	^

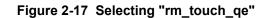
Figure 2-16 Enabling the Pins in Use ((1): r\_ctsu\_qe, (2): TSCAP Pin, (3): Two TS Pins)



#### b. Adding touch middleware

Open the "New Component" dialog box, select "rm\_touch\_qe", and click on [Finish].

H Touch QE API	rm_touch_qe	Firmware Integr	2.10	
----------------	-------------	-----------------	------	--



 Click on the added "rm\_touch\_qe" module and set up serial communications monitoring through a UART.

Set items in [Value] as shown below.

# Support for QE monitoring using UART:Include code to update sensor data for monitor# Support for Serial tuning using UART:Include code to serial tuning# UART channel:UART1

Components <sup>2</sup>	
type filter text     v @ Configurations       v @ Startup     *       g Generic     # Support for QE monitoring using UART	i
# Support for Serial tuning using UART Include code to serial tuning # UART channel # UART channel # UART boadrate # UART priority # UART priority Level 1 Include code to serial tuning # UART priority Include code to serial tuning # UART channel # UART priority UART priority Include code to serial tuning # UART channel # UART priority UART priority Include code to serial tuning # UART priority Include code to serial tuning Include code to serial tuning # UART priority Include code to serial tuning # UART priority Include code to serial tuning	hitor

Figure 2-18 Selecting the Property of "rm\_touch\_qe" ((1): rm\_touch\_qe)



# c. Adding a serial interface

Open the "New Component" dialog box, select "r\_sci\_rx", and click on [Finish].

🖶 SCI Driver	r_sci_rx	Firmware Integration Technology	4.40



• Click on the added "r\_sci\_rx" module to display the related ports on the configuration panel and set items in [Value] as shown below.

# Use ASYNC mode:Include# Include software support for channel 1:Include# Transmit end interrupt:EnableResources > SCI > SCI1: Tick the checkbox ( $\sqrt{}$ ).RXD1, TXD1: Tick the "Used" checkbox ( $\sqrt{}$ ).

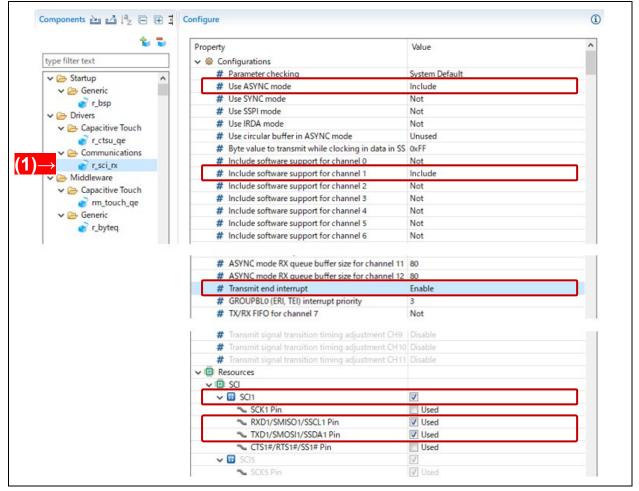


Figure 2-20 Setting "r\_sci\_rx" ((1): r\_sci\_rx)



• Assign the pin functions for SCI1. Select the [Pins] tab.

Figure 2-21 Selecting the [Pins] Tab

Expand "Serial communications interface" in the tree next to the [Pin Function] panel on the [Pin configuration] tabled page and select SCI1.

Assign the "RXD1" function to "P30" and the "TXD1" function to "P26".

Hardware Resource ⊞ ⊟ ↓ <sup>a</sup> z a			ing, ? = any character)	2 🖬	<b>[]</b>   22 23 ~
TMR0     TMR1     TMR2     TMR2     TMR3     Will Serial compunications interface     SCI5     SCI6     SCI6     SCI8     SCI9     SCI9     SCI12     Viji I2C bus interface     RIC0     Viji Serial peripheral interface     RSPI0     CAN module (RSCAN)     RSCAN0     SPin Function     Pin Number	Enabled     Enabled     Comparison     Compari	Function CTS1# RTS1# RXD1 SCK1 SMISO1 SM0S11 SS01 SS01 SSCL1 SSDA1 TXD1	Assignment Not assigned Not assigned P30/MTIOC48/TMRI3/POE8#/RXD1/SMISO1/SSCL1/TS2/IRQ0 Not assigned P26/MTIOC2A/TMO1/LPTO/TXD1/SMOSI1/SSDA1/TS4	Pin Number / Not assigned / 20	I None None None None

Figure 2-22 Assigning Pin Functions for SCI1 ((1): [Pin Function] Tab, (2): SCI1)



#### 2.3 Generating Code

Generate code.

1. Click on the Generate Code button of the Smart Configurator and generate code.

Components 🚵 🛃 🛱 🖡	± 🐳 *	Configure			
	😧 🐨 👘	Property	Value		
type filter text		✓			
✓ → Startup		# User stack setting	2 stacks		
✓ → Startup ✓ → Generic	^	# User stack size	0x400		
r bsp		# Interrupt stack size	0x100		
V > Drivers		# Heap size	0x400		
<ul> <li>Capacitive Touch</li> </ul>		# Initializes C input and output library functions	Disable		
r ctsu ge		# Enable user stdio charget function	Use BSP charget() function		
✓ Communications		# User stdio charget function name	my_sw_charget_function		
r_sci_rx		# Enable user stdio charput function	Use BSP charput() function		
✓ → Middleware		# User stdio charput function name	my_sw_charput_function		
<ul> <li>Capacitive Touch</li> </ul>		# Processor Mode	Stay in Supervisor mode		
rm_touch_ge		# ID code 1	OXFFFFFFF		
✓ (⇒ Generic		# ID code 2	0xFFFFFFF		
r_byteq		# ID code 3	OxFFFFFFF		
<ul> <li>Participation</li> </ul>		u		_	

Figure 2-23 Generating Code

When the following message box is displayed, click on [Proceed].

Configuration must be saved before generating code. Proceed with save and generate? Always save and generate without asking? Proceed Cancel	Code Generating ×

Figure 2-24 Message at Code Generation

2. Confirm that "Code generation is successful" is displayed in the Console view.

⊆ Console ×	🖹 🛃 🛃 🚽 🚽 🗂 🗖 🗖 🗖
Smart Configurator Output	
THOMODODOT. IIIE génerateu. Sitt. Seit (E Dyted (E Dyted II.II	•
M04000001: File generated: <u>src\smc gen\r byteq\src\r byteq.c</u>	
M04000001: File generated: <u>src\smc gen\r byteq\src\r byteq private.h</u>	
M04000001: File generated: <u>src\smc gen\r config\r byteq config.h</u>	
M04000001: File generated:src\smc gen\general\r cg macrodriver.h	
M04000001: File generated: <u>src\smc gen\general\r cg userdefine.h</u>	
M04000001: File generated: <u>src\smc gen\general\r smc entry.h</u>	
M04000001: File generated: <u>src\smc gen\general\r cg hardware setup.c</u>	
M04000001: File generated:src\smc gen\r pincfg\r pinset.h	
M05000012: File generated:src\smc gen\r pincfg\Pin.h	
M05000012: File generated: <u>src\smc gen\r pincfg\Pin.c</u>	
M06000002: File generated: <u>src\smc gen\general\r smc interrupt.c</u>	
M06000002: File generated: <u>src\smc gen\general\r smc interrupt.h</u>	
M0300004: <u>File modified costers gonte con</u> figtr bsp config.h	
M0000002: Code generation is successful:C \Work\IAR EWRX Touch\RSK RX140\src\smc gen	
	J

Figure 2-25 Termination of Code Generation



3. Terminate the Smart Configurator by selecting [Exit] from the [File] menu.

Γ	💰 Sma	t Configui	rator					
	ile Wi	ndow H	lelp					
1	9 Ne	<i></i>						
6	🍃 Ор	:n						
	Sav	2	Ctrl+S	<				
	Res	art		nt con				
-	Exit	n (**) (		Œ				
				10				
	~ (≥	ilter text Startup Ə Gene		^				

Figure 2-26 Terminating the Smart Configurator

4. In Windows Explorer, confirm that an IAR IDE Workspace file (\*.eww) with the ③ icon has been created in the folder with the location specified in section 2.1.3 (3).

$\leftarrow$ $\rightarrow$ $\checkmark$ $\uparrow$ $\bigcirc$ $\sim$ Work $\rightarrow$ IAR_EV	NRX_Touch > RSK_RX140	✓ Ö	(140
Name	Date modified	Туре	Size
.settings	2022/11/30 12:41	File folder	
src	2022/11/30 12:41	File folder	
main.c	2022/11/30 12:41	C File	1 KB
IAR_Touch_project.ewd	2022/11/30 12:41	EWD File	28 KB
IAR_Touch_project.ewp	2022/11/30 12:41	EWP File	72 KB
IAR_Touch_project.eww	2022/11/30 12:41	IAR IDE Workspace	1 KB
buildinfo.ipcf	2022/11/30 12:41	IPCF File	17 KB
IAR_Touch_project.scfg	2022/11/30 12:41	SCFG File	55 KB

Figure 2-27 IAR EWRX Workspace Folder

5. Set items to allow launching the Smart Configurator from the [Tools] menu in the IAR EWRX window. Double-click on the IAR IDE Workspace file created in Windows Explorer to start the IDE. In the opened IAR EWRX window, select [Configure Tools...] from the [Tools] menu to open the [Configure Tools] dialog box.

r					
[	IAR_Touch_project - IAR Embedded W	/ork	bench IDE - RX 4.20.3		
	File Edit View Project Simulator	Too	ols Window Help		
	1 1 🗈 🖴 🕋 🔚 🛛 🗶 🛍 🖸 🗄	٥	Options	Ξ <	
	Workspace		Filename Extensions		
	Debug		Configure Viewers		
	Files		Configure Custom Argument Variables		
	🗆 🌒 IAR_Touch_project - Debug	6	Configure Tools		
	⊢⊞ ■ Renesas_AP ├── D buildinfo.ipcf		Convert To IAR for RX		
	Here in a main.c	_	•		
	L 🕀 💼 Output				
	IAR_Touch_project				

Figure 2-28 Tools



- In the [Configure Tools] dialog box, click on [New] and enter values for items (2) to (5) shown below.
  - (1) Menu Content: The contents entered in [Menu Text] are displayed.
  - (2) Menu Text: Smart Configurator for RX
  - (3) Command: [Installation folder]¥eclipse¥SmartConfigurator.exe
  - (4) Argument: \$PROJ\_FNAME\$.scfg
  - (5) Initial Directory: \$PROJ\_DIR\$

Click on [OK] to close the "Configure Tools" window.

Configure Tools
Menu Content: Smart Configurator for RX Cancel New Delete
Meru Text: Smatt Configurator for RX Command: figurator/RX\cclipse\SmattConfigurator.exe Argument: \$PRDJ_FNAME\$.scfg Initial Directory: \$PRDJ_DIR\$ Redirect to Qutput Window Prompt for Command Line Tool Agailable: Always

Figure 2-29 Setting Tools

— The Smart Configurator can now be selected from the [Tools] menu.

IAR_Touch_project - IAR Embedded V File Edit View Project Simulator	/orkbench IDE - RX 4.20.3 Tools Window Help
Pile     Color     Color     Color       Workspace     Debug       Files       Image: IAR_Touch_project - Debug       Image: IAR_Touch_p	Options     E       Filename Extensions     Configure Viewers       Configure Custom Argument Variables     E
IAR_Touch_project	

Figure 2-30 Settings Reflected in the [Tools] Menu



6. Next, right-click on the project name in the [Workspace] window, then select [Rebuild All] to perform building. After building has finished, confirm that no errors are indicated in the [Build] window or status bar at the bottom of the main window.

id Messages	efore bui	ld		L	a x J	i Status bar	
cinyoprocetty	Set as Active					Ready Errors 0, Warnings 0 CAP NUM	OVR 📑
Workbench 4203 (CVProgram File: Systems)Embedded Workbench fo birlysprac.dl)	Open Containing Foldes File Properties						
Log Wed Nov 30, 2022 0855:51: IAR Em	Version Control System	>				Total number of errors: 0 Total number of warnings: 0	
buglog	Rename			•	· a ×	Messages A ft or the util of	L
AR_Touch_project	Remove					Build	•
	Add						
	Stop Build	-					
	C-STAT Static Analysis	-					
	Rebuild All Clean						
-E 🖬 main.c	Compile						
KA Touch project - Debug     Fenesas_AP     Duidinta.jpcf	Options Make	- 1					
les IAR Touch project-Debug	Options						
lebug	~						
nkspace	+ a x						
· دا 🖸 🖬 🗶 🚔 ا 😫 🖆 🖞	c • <	् > ቱ	⊨≝ < 📮 > ≷ [≥	 a O	• .		
le Edit View Project Simulator Too	is Window Help						

Figure 2-31 Building

7. Select [Exit] from the [File] menu. In the [laeldePm] dialog box that is displayed, click on [Yes] to save the project and terminate the IAR EWRX workspace.

_	IAR_Touch_project - IAR Em					
1 1 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	Edit View Project S New File New Workspace Open File Open Workspace Open Header/Source File	mulator Tools Ctri Ctri	+N +O	dow Help		
	Save Workspace Save Workspace As Close Workspace Save Save As Save All	Ctrl-	+F4 I+S		IarldePm       Save project "IAR_Touch_project"?       Yes	X Cancel
	Page Setup Print Recent Files Recent Workspaces	Ctr	I+P			
8	Exit bin\rxproc.dll)			/x/		

Figure 2-32 Terminating the IAR EWRX Workspace



#### 2.4 Setting the Project Folder and Target MCU

After creating a project, launch the standalone-version QE. The folder containing the created project and MCU product information should be set, according to "1. Preparation" in the workflow diagram.

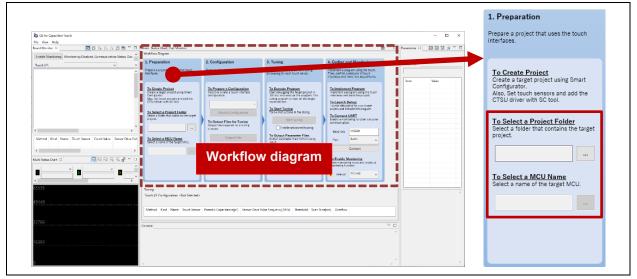


Figure 2-33 Standalone-Version QE - Workflow Diagram: <Step 1> Preparation

1. Select the project folder.

Click on [...] to the right of the input area under "<u>To Select a Project Folder</u>" to select the folder storing the project.

2. Select product information on the microcontroller.

Click on [...] to the right of the input area under "<u>To Select a MCU Name</u>" to select an MCU by clicking on the <u>select a MCU Name</u> is labeled in the "Product Name of Target MCU" dialog box that is displayed.

- (1) Family Name Select "RX".
- (2) Group Name Select "RX140".
- (3) Pin Number Select "LQFP/80pin:FN".
- (4) ROM Size Select "256KB:6".
- (5) Product Name "R5F51406xxFN" will be displayed when the above items (1) to (4) have been selected correctly.

Click on [OK] to close the dialog box.

Product Name of Target MCU		×
Family Name	RX	¥
Group Name	RX140	¥
Pin Number	LQFP/80pin:FN	✓
ROM Size	256KB:6	<b>v</b>
Product Name	R5F51406xxFN	
0	K Cancel Help	

Figure 2-34 "Product Name of Target MCU" Dialog Box



#### 3. Preparing the Touch Interfaces

Set items according to "2. Configuration" in the workflow diagram.

File View Help								2. Configuration
Board Monitor 🖾 🔲 🗔 🗔 🗔 🖬 🛱 🖤	Main Status Chart Pad Monitor				Parameters 8	13 13 13 (% °		
Enable Monitoring Monitoring: Disabled, Communication Status:	Workflow Diagram							Prepare a touch interface configuration.
Touch I/F:	1. Preparation	2. Configuration	3. Tuning	4. Coding and Monitoring	Touch VF:		-	
	Prepare a project that uses the touch     Interfaces	Prepare a touch internet guration.	QE will automatically perform tuning	Implement a program using the touch.				
	Interfaces.						- 2	
					Item	Value		
	To Create Project Create a target project using Smart	To Prepare a Configuration Select or create a touch interface	To Execute Program Start debugging the target project in	To Implement Program Implement a program using the touch				To Prepare a Configuration
	Configurator. Also, Set touch sensors and add the	configuration.	IDE tool and execute the program. The turning program written on the target	interfaces and build the project.				Select or create a touch interface
	CTSU driver with SC tool.	~	board will run.	To Launch Debug Launch debugging for your target project and execute the program.				configuration.
	To Select a Project Folder	Modify Configuration	To Start Tuning Follow instructions in the dialog.					
	Select a folder that contains the target project.		Start Tuning	To Connect UART Enable a monitoring function via serial				¥ .
	CHWorkHAR EWRX Toucl	To Output Files for Tuning Output files required for a tuning	Enable advanced tuning	communication.				
٢		process.		Baud rate 115200				M RECEIPT
Method Kind Name Touch Sensor Count Value Sensor D	we To Select a MCU Name	Output Files	To Output Parameter Files Output parameter files from a tuning	Port Auto v				Modify Configuration
	Select a name of the target MCU.		result.					
د	> RSF51406xxFN			Connect				To Output Files for Tuning
Multi Status Chart 💠 🚺 🔂 🖂 🖂 🖉 "		Workflow	diagram	To Enable Monitoring				Output files required for a tuning
			anagram	Show monitoring views and enable a monitoring function.				process.
				A Interval 75 [ms]				P
	- <b>*</b>			A marva				
55535								Output Files
05555	Tuning						~	
10110	Touch I/F Configuration: <not selected<="" td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td></not>							
49149								To Implement Program
	Method Kind Name Touch Sens	r Parasitic Capacitance[pF] Sensor Drive R	Pulse Frequency[MHz] Threshold Scan Ti	me[ms] Overflow				Implement a process to call the main
								loop for touch in the main() function.
32766	Console							
								To Build Project
1000								Build the target project using IDE tool.
16383								

Figure 3-1 Standalone-Version QE - Workflow Diagram: <Step 2> Configuration

#### 3.1 Creating a Touch Interface Configuration

Create a new touch interface configuration.

1. Click on under "<u>To Prepare a Configuration</u>" and select "Create a new configuration". The "Create Configuration of Touch Interfaces" window appears and displays the area for placing touch interfaces.

. Configuration	File Name of Touch I/F:	RSK_RX140	Setup Configuration	Import / Re-edit
	Description:	131_101140	Setup Configuration	import/ ne-eut
repare a touch interface configuration.				Touch I/F *
				Capacitance Type
				Self Capacitance V
To Develop a Configuration				Sen capacitance V
To Prepare a Configuration Select or create a touch interface				Button
configuration.				Slider (horizontal)
				Slider (vertical)
~				Wheel
				Key pad
Create a new configuration				3D Gesture (AI)
To Output Files for Tuning				So destare (A)
To Output Files for Tuning Output files required for a tuning				Touch pad
process.				Shield Pin
				TC Pin
Output Files				Capacitance Sensor
				Current Sensor
To Implement Program				Diagnosis Pin
Implement a process to call the main loop for touch in the main() function.				
	Setting			Remove Touch I/F
To Build Project Build the target project using IDE tool.	Setup Touch I/F	Setup Resistance Value Clear Assigned TSx		Configuratoins (Methods)
build the target project USINg IDE tool.				
				Create Cancel Help

Figure 3-2 Creating a Touch Interface Configuration



2. Clicking on [Button] in the "Touch I/F" panel on the right side of the "Create Configuration of Touch Interfaces" window enables the placement of buttons at the cursor position. Clicking on a position in the area causes a button to be placed there.

After having placed two buttons as shown below, press the [ESC] key to stop adding touch interfaces.

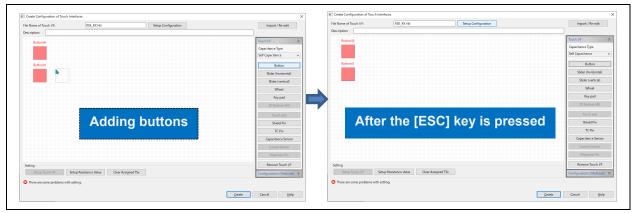


Figure 3-3 Adding Buttons

3. Double-click on the placed buttons and set the following items in the displayed "Setup Touch Interface" dialog box.

Button00: [Name] TS\_B1 [Touch Sensor] TS11 [Resistance] 560 Button01: [Name] TS\_B2 [Touch Sensor] TS12 [Resistance] 560 The color of the buttons will change to green if the items of the touch interface have been set correctly.

File Name of Touch I/F:	F	SK_RX140	
Description:			
Button00			
	Setup Touch Interfa		×
Button01	Button(self)		
	Name	TS_B1	
	Touch Sensor	Resistance[o	hm]
	TS11	♥ 560	~
	ОК	Cancel	Help

Figure 3-4 Setting the Touch Interface Items for Button00



4. Confirm that the color of the two buttons that have been placed is green and click on [Create] to close the window.

File Name of Touch I/F:	RSK_RX140	Setup Configuration	Import /	Re-edit
Description:				
TS_B1			Touch I/F	\$
			Capacitance T	ype
TS11			Self Capacitan	ce 🗸
TS_B2			Bu	tton
TS12			Slider (h	orizontal)
			Slider	(vertical)
			W	heel
			Kej	/ pad
			3D Ges	ture (Al)
			Touc	h pad
				ld Pin
				Pin
			Capacita	nce Sensor
				t Sensor
			Diagn	osis Pin
Cattlera			Remove	Touch I/F
Setting Setup Touch I/F	Setup Resistance Value Clear Assign	ed TSx	Configuratoin	
			Configuration	s (Methods) 🔹

Figure 3-5 Stopping Creation of Touch Interface Configurations

5. The touch interface configurations are displayed on the [Tuning] panel in the "Main" window.

oard Monitor 🛛 🗾 🗐 🕄 🛄 🖓 🛄 💭 🛄 " 🗆	Main Status Chart Pad Monitor			<b>u</b> - 0	Parameters 🛛 🟥 🕼 🕼 🗇 " 🗆
Enable Monitoring Monitoring: Disabled, Communication Status: I	Workflow Diagram				
Touch I/F: v v	1. Preparation	2. Configuration	3. Tuning	4. Coding and Monitoring	Touch I/F:
TS_B1	Prepare a project that uses the touch interfaces.	Prepare a touch interface configuration.	QE will automatically perform tuning processing for each touch sensor.	Implement a program using the touch. Then, confirm a behavior of touch interfaces and make fine adjustments.	
TS.B2 Method Kind Name Touch Sensor Count Value Sensor Drive	In Create Project Credge at urget project using Smart Configuration Methods at urget project using Smart Configuration Internet and the Configuration Internet (Configuration Internet) Configuration Internet Description Configuration Internet Description Configuration Internet Description Configuration Conf	Technoses a Configuration Beet or easts a taxoh interface configuration. SSK, KY-40.01rf // Modify Configuration To Ottos Tiles for Turning Cator The reaction of a taxing Cator The reaction of the taxon Cator The reaction of the taxing Cator The reaction of the taxon Cator The reaction of the taxing Cator The reaction of the taxing Cator The taxon of the taxon Cator The taxon of the taxing Cator The taxon of the taxon of the taxon The Build Preisel Build the taxon of the taxon of the taxon Cator The taxon of the taxon of the taxon of the taxon Cator The taxon of taxon of the taxon of the taxon Cator The taxon of taxon of taxon of taxon Cator The taxon of taxon of taxon of taxon Cator The taxon of taxon of taxon of taxon of taxon Cator The taxon of taxon of taxon of taxon of taxon of taxon of taxon Cator The taxon of	A Create Program Description of the Larger for the State of the State	Indicators Program using the torust interfaces and object the spletcher interface and	Item Volue
< > 55535					
	Tuning			^	^
19149	Touch I/F Configuration: RSK_RX140				
12766	Method         Kind         Name         Touch           config01         Button(self)         TS_B1         TS11           config01         Button(self)         TS_B2         TS12	Sensor Parasitic Capacitance[pF] Sensor 	Drive Pulse Frequency[MHz] Threshold - -	- None - None v	
6383	Console			- 8	

Figure 3-6 Touch Interface Configurations



#### 3.2 Outputting Files for Tuning Touch Sensors

After the files for tuning touch sensors have been output to the specified folder, set the frequency of the peripheral module clock of the MCU and the voltage supplied to the microcontroller.

1. Click on [Output Files] to display the "The output folder of parameter files" dialog box. Click on the folder to which the files are to be output and then click on [Select Folder].

Prepare a touch interface configuration.	a The output folder of parameter	files			
	$\leftarrow$ $\rightarrow$ $\checkmark$ $\uparrow$ $\square$ « Work »	$IAR_EWRX_Touch \rightarrow RSK_RX140$	, 5 v	Search RSK	_RX140
To Prepare a Configuration Select or create a touch interface configuration.	Organize 🔻 New folder				== - (
RSK_RX140.tifcfg	Name	Date modified	Туре	Size	
Modify Configuration	.settings	2022/11/30 6:59	File folder		
	Debug	2022/11/30 8:40	File folder		
Output Files for Tuning put files required for a tuning	QE_FirstAdjustments	2022/11/30 10:06	File folder		
ess.	QE-Touch	2022/11/30 9:46	File folder		
Output Files	settings	2022/11/30 8:54	File folder		
	src src	2022/11/30 8:41	File folder		
o Implement Program nplement a process to call the main op for touch in the main() function.	Folder: QE	FirstAdjustments			
To Build Project Build the target project using IDE tool.			S	elect Folder	Cancel

Figure 3-7 Outputting Files for Tuning

In the example shown in this figure, the "QE\_FirstAdjustments" folder is created under the project folder and selected as the folder for output of the parameter files.

2. When the folder for outputting files is selected, the "Frequency of Peripheral Module Clock (PCLKB or PCLKL)" dialog box appears. Enter "48" for the frequency and click on [OK].

Frequency of Peripheral Module Clock (PCLKB or PCLKL)	×
Peripheral module clock frequency (PCLKB or PCLKL)[MHz] 48	
OK Cancel	Help

Figure 3-8 Setting the Frequency of the Peripheral Module Clock (PCLKB or PCLKL)

In the "Operating Voltage of MCU (VCC)" dialog box that appears next, click on . Then select [5.0V : (4.5V ≤ VCC ≤ 5.5V)] as the voltage value and click on [OK]. Since an external power supply is used, confirm the electrical characteristics of the microcontroller in use.

Operating Voltage of MCU (VCC)		×
	5.0V : (4.5V ≦ VCC ≦ 5.5V)	~
	3.3V : (2.4V ≤ VCC ≤ 5.5V) 1.8V : (1.6V ≤ VCC ≤ 5.5V)	

Figure 3-9 Setting the Operating Voltage (VCC) of the MCU



4. Finally, the "QE for Capacitive Touch" dialog box appears. Follow the instructions in the dialog box. The same contents are displayed in the "Console" panel at the bottom of the window.

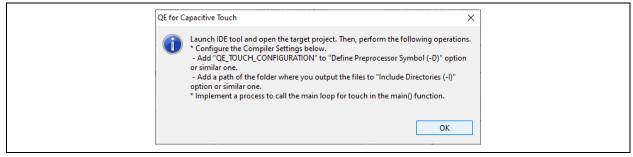


Figure 3-10 "QE for Capacitive Touch" Dialog Box

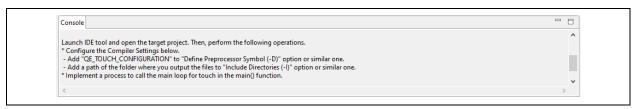


Figure 3-11 "Console" Panel



#### 3.3 Implementing Programs for Tuning

In the IAR EWRX project, modify the main program to operate the programs for tuning, add the programs for tuning, add compiler options, and make settings for debugging.

1. In Windows Explorer, double-click on 'S <workspace file name>.eww" to open the IAR EWRX window.

Image: Settings   Image: Settings <th>File Home Share View</th> <th></th> <th>~ 📀</th> <th></th>	File Home Share View		~ 📀	
Name     Date modified     type     Size       Settings     Size     Size       Obug     Ital: Touch_project - IAR Embedded Workbench IDE - RX 4.20.3     ×       Settings     Size     Ital: Touch_project - IAR Embedded Workbench IDE - RX 4.20.3     ×       Settings     Size     Ital: Touch_project - IAR Embedded Workbench IDE - RX 4.20.3     ×       Settings     Size     Ital: Touch_project - IAR Embedded Workbench IDE - RX 4.20.3     ×       Settings     Size     Ital: Touch_project - IAR Embedded Workbench IDE - RX 4.20.3     ×       Size     Size     Ital: Touch_project - IAR Embedded Workbench IDE - RX 4.20.3     ×       Size     Size     Ital: Size     Ital: Size       Size     Ital: Size     Ital: Size     Ital: Size       Size     Ital: Size     Ital: Size     Ital: Size       Size     Ital: Size     Ital: Size     Ital: Size       Ital: Size	$\leftarrow$ $\rightarrow$ $\checkmark$ $\Uparrow$ IAR_EWRX_Touch $\rightarrow$	RSK_RX140 O Search RSK_RX140		
Debug       Impleter Jak Embeddeed Workbench for PK 4.20.1 (sk)         G. E. Finst Adjustments       G. E. Touch         G. E. Touch       File         Settings       Settings         Sr       Debug         D. R. Touch project.       Settings         I. Settings       Vebuldinto.         I. Settings       Vebuldinto.         I. Settings       Settings         Settings       Vebuldinto.         Settings       Vebuldinto.         Settings       Vebuldinto.         Settings       Vebuldinto.	Name	Date modified Vpe	Size	
15 items 1 item selected 182 bytes  Debug Log  Wed Nov 30, 2022 10:17:29: IAR Embedded  Workbench 42:0.3 (CV/Program Files (x86)(IAR Systems(Embedded Workbench for FX 4:20.1(xx) bin(xxproc.dll)	Debug GE, FirstAdjustments GE-Touch settings src buildinfo.ipcf IAR_Touch_project.ew IAR_Touch_project.ewp IAR_Touch_project.ewp IAR_Touch_project.ewp IAR_Touch_project.eww IAR_Touch_project.eww IAR_Touch_project.eww	File Edit View Project Simulator Tools Window Help         Workspace         Workspace         Files         Files         B Reneses_AP         B mein c         B mein c	<ul> <li>&lt; Q &gt; \$ +Ξ &lt; Q &gt;</li> </ul>	
Log Wed Nov 30, 2022 10:17:29: IAR Embedded Workbench 4:20.3 (C:Program Files (x86)(IAR Systems)Embedded Workbench for FX 4:20.1(x), bin(xproc.dll)	15 items 1 item selected 182 bytes	p		
Build 👻 🗸 🗸		Log Wed Nov 30, 2022 10:17:29: IAR Embedded Workbench 4:20.3 (CVProgram Files (x66)/IAR System/Embedded Workbench for 9X 4:20.1 (vA		¥¥X
		Build		<b>→</b> ‡ ×
Messages File Li		Messages	F	File Li

Figure 3-12 IAR EWRX Window

- 2. Change the main function.
  - Implement processing for calling the touch main function in the main() function.
  - Double-click on main.c to open the file. Add the declaration of the touch main function and code for calling it, then save main.c and close the file.

Line 5: extern void qe\_touch\_main(void);

Line 9: qe\_touch\_main();

<ul> <li>IAR_Touch_project - IAR Embedded Workl</li> <li>File Edit View Project Simulator Too</li> </ul>				- 🗆 X
it t 🗈 🗳 🍯 🖶 i X 🗈 🗂 i 5 (	C			🗉 < Q > ⇆ म्म < 📮 > 🕢 🖻 📓 🖷 💽 🕨
Workspace	•	‡Χ	main.c *	x
Debug		~		f
Files  Image: Gradient Content of the series	<b>*</b>	•	1 2 3 4 5 6 7 8 9 10 11	<pre>#include "r_smc_entry.h" int main(void); extern void qe_touch_main(void);</pre>
IAR Touch project				

Figure 3-13 Modifying the main() Program



- 3. Add the files for tuning.
  - Copy the files for tuning to the "qe\_gen" folder and add the files for tuning to the project.
  - a. Create a "qe\_gen" folder in the folder for storing the workspace file. Copy all files in the folder where the files for tuning were saved to the "qe\_gen" folder.

File Home Share View	"			~ 😨	File Home Share View	N			~ (
← → ~ ↑ 📙 « RSK_RX140	> QE_FirstAdjustments	5 v	⊘ Search QE_FirstAdjustments		← → × ↑ <mark> </mark>	> qe_gen	ч <mark>0</mark> , Р	Search qe_gen	
Name	Date modified	Туре	Size		Name	Date modified	Туре	Size	
ge_touch_config.c	2022/11/30 10:13	C File	7 KB		ge_touch_config.c	2022/11/30 10:13	C File	7 KB	
ge_touch_sample.c	2022/11/30 10:13	C File	4 KB		qe_touch_config.h	2022/11/30 10:13	H File	4 KB	
qe_touch_config.h	2022/11/30 10:13	H File	4 KB		qe_touch_define.h	2022/11/30 10:13	H File	5 KB	
qe_touch_define.h	2022/11/30 10:13	H File	5 KB		ge_touch_sample.c	2022/11/30 10:13	C File	4 KB	

Figure 3-14 Copying the Files for Tuning

B. Right-click on the project name in the [Workspace] window and select in the order of [Add] → [Add Group...]. In the [Add Group] dialog box, enter "qe\_gen" and click on [OK].

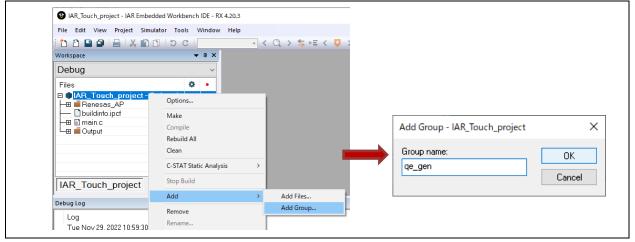


Figure 3-15 Adding a Group

c. Right-click on the added group name and select in the order of [Add] → [Add Files...]. In the [Add Files] dialog box, select all files and click on [Open].

	- IAR Embedded Workb							
File Edit View Pr	oject Simulator Tool	ls \	Vindow	Help				
i 🗅 🗅 🔛 🕋 🔒	1 X 🗋 🗋 1 5 0			- < Q > ⇆ >Ξ				
Workspace		-	ąх					
Debug			~					
Files		٥	•		🔮 Add Files - qe_gen			
□ ● IAR_Touch_p 	project - Debug *	~						0.0.1
⊞ <b>≣</b> Renesas	Options				$\leftrightarrow \rightarrow \uparrow \uparrow \sqcup w RSK_RX$	140 > qe_gen	~ Ö	,○ Search qe_gen
- Duildinfo.i	Make				Organize 👻 New folder			800 -
⊨⊞ 🗟 main.c	Compile				Name	Date modified	Туре	Size
L-⊞ i Output	Rebuild All				ge_touch_config.c	2022/11/30 10:13	C File	7 KB
					qe_touch_config.h	2022/11/30 10:13	H File	4 KB
	Clean				qe_touch_define.h	2022/11/30 10:13	H File	5 KB
	C-STAT Static Analys	sis			ge_touch_sample.c	2022/11/30 10:13	C File	4 KB
IAR_Touch_	Stop Build				File name:	"ge_touch_config.c" "ge_touch_confi	q.h" "qe_to ~	Source Files (*.c;*.cpp;*.cc;*
Debug Log	Add			Add Files				Open Can
Log				Add Group				

Figure 3-16 Adding Files



#### 3.4 Building a Project

1. Add the compiler options.

Add the include paths and symbol definitions of the files for tuning to the preprocessor settings.

- Right-click on the project name in the [Workspace] window and select [Options...] to open the [Options] dialog box.
- Select "C/C++ Compiler" in [Category:] and then select the [Preprocessor] tab.

IAR_Touch_project - IAR Embedded W	/orkbench IDE - RX 4.20.3	Category: General Options Static Analysis	Multi-file Compilation	Factory Settings
File Edit View Project Simulator		Runtime Checking <b>C/C++ Compiler</b> Assembler Output Converter Custom Build Build Actions Linker Debugger E1 / E20 E2 E2 Lite / E2-CUBE2 J-Link Simulator	MISRA-C:2004 MISRA-C:1988 Encodings Language 1 Language 2 Dode Output List Preprocessor Isnore standard include directories Additional include directories: (one per line) SFROU_DIRS stores more anitrm, touch, get stor SFROU_DIRS stores more anitrm, touch, get stor SFROU_DIRS stores anitrm, touch, get stores and SFROU_DIRS stores anitrm, touch, get store SFROU_DIRS stores anitrm, touch, get stores and SFROU_DIRS stores anitrm, touch, get stores and SFROU_DIRS stores anitrm, touch, get stores and SFROU_DIRS stores	

Figure 3-17 Displaying the [Preprocessor] Tab of "C/C++ Compiler"

- a. Add the include paths.
  - (1) On the [Preprocessor] tabbed page, click on [...] in [Additional include directories: (one per line)] to open the [Edit Include Directories] dialog box.
  - (2) Scroll down to the bottom of the [Edit Include Directories] dialog box and click on <Click to add>.
  - (3) The [Add Include Directory] dialog box appears. Select the folder where the files for tuning were saved and click on [Select Folder].

MISRA-C:2004	MISRA-0:1998	Encodines	Extra Options	Edit Include Dire	ctories			
Language 1	Language 2	Code	Optimizations					
Output		Preprocessor	Disgnostics	Include director	, c\smc_gen\rm_touc			
_	l include directories lirectories: (one per			\$PROJ_DIR\$\s \$PROJ_DIR\$\s	c\smc_gen\rm_touc c\smc_gen\r_sci_rx c\smc_gen\r_sci_rx	h_qe\wrapper		
\$PROJ_DIR\$¥src \$PROJ_DIR\$¥src \$PROJ_DIR\$¥src	¥smo_gen¥general ¥smo_gen¥rm_touci ¥smo_gen¥r_bsp ¥smo_gen¥r_byteq ¥smo_gen¥r_config	h_qe	×	<click add="" to=""></click>				OK Cancel
		g	Add Include Directory - → × ↑ 📙 « IAR_EWRX	Touch > RSK_RX140 >	▼ <b>で</b>	Search RSK_RX140	×	
			Organize 🔻 New folder				?	
		1	Vame ^	Date modified	Туре	Size	^	
			.settings Debug	2022/11/30 6:59 2022/11/30 8:40	File folder File folder			
			QE_FirstAdjustments	2022/11/30 10:13	File folder			
			qe_gen	2022/11/30 10:34	File folder			
			QE-Touch	2022/11/30 9:46	File folder		~	
			Folder: qe_g					
			1-5		Salas	t Folder Cancel		

Figure 3-18 Adding an Include Directory



(4) Change the path specification mode of the added include directory from absolute path to relative path.

Click on since the relative directory that starts with "\$PROJ\_DIR\$". After confirming that the path has been changed to a relative path in the [Edit Include Directories] dialog box, click on [OK].

clude directory		^		
ROJ_DIR\$\src\smc_gen\rm_touch_qe\inc				
ROJ_DIR\$\src\smc_gen\rm_touch_ge\wrapper			Edit Include Directories	
ROJ_DIR\$\src\smc_gen\r_sci_rx\src				
ROJ_DIR\$\src\smc_gen\r_byteq\src			Include directory	
C:\Work\IAR_EWRX_Touch\RSK_RX140\qe_gen			\$PR0J_DIR\$\src\smc_gen\rm_touch_ge\inc	
		C:\Work\IAR_EWRX_Touch\RSK_RX140\qe_gen	PDJ_DIR\$\src\smc_gen\rm_touch_ge\wrapper	
	OK	SPROJ_DIRS\ge_gen	\$P: DIR\$\src\smc_gen\r_sci_rx\src	
		\$TOOLKIT DIR\$\\\\Work\IAR EWRX Touch\RSK RX140\ge gen	\$PRt	
		anoockin_bina (a	\$PR0J_DIR\$\qe_gen	

Figure 3-19 Changing the Path to the Include Directory to a Relative Path

b. Enter "QE\_TOUCH\_CONFIGURATION" in [Defined symbols: (one per line)].

Language 1 Lar
Output List
Ignore standard include Additional include directori     ROJDIR\$#sro¥smo.g     PROJDIR\$#sro¥smo.g     PROJDIR\$#sro¥smo.g     \$PROJDIR\$#sro¥smo.g     \$PROJDIR\$#sro¥smo.g

Figure 3-20 Adding a Symbol Definition

- 2. Next, set up the debugger.
  - Select "Debugger" in [Category:] and select the [Setup] tab.
  - Click on vin [Driver:] and select "E2 Lite/EZ-CUBE2".
  - Enter "main" in [Run to:] and tick the checkbox ( $\sqrt{}$ ).
  - Click on [OK] to close the [Options] dialog box.

General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger E1 / E20 E2 E2 Ltte / E2-CUBE2 J-Link Simulator	Setup       nages       Extra Options       Plagins         Driver:       Plagins       Plagins         Simulator       Plagins       Plagins         E7 AED       Plagins       Plagins	Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger E1/E20 E2 E2 Lite / E2-CUBE2 J 3-Link Simulator	Setup       Images       Exite Options       Fluctures         Priver:       Province:       Images       Images         S2 Life / E2-OUBE2       Images       Images       Images         System macros       Images       Images       Images         System macros       Images       Images       Images         Device default       Images       Images       Images         STOOLKUT_DIREWoon figHdebugger/lor5 f51 406/ ddf       Images       Images
E2 Lite / EZ-CUBE2 J-Link	\$TOOLKIT,DIR\$WoonfeWdebugger/ior5/51406.ddf	E2 Lite / EZ-CUBE2 J-Link	\$TOOLKIT_DIR\$Woon fgWdebugger/lor5/61 406 ddf

Figure 3-21 Debugger - [Setup] Tab



#### 3. Set up the hardware.

a. The [E2/E2 Lite] menu will become selectable from the menu bar. Select [Hardware Setup...] from the [E2/E2 Lite] menu.

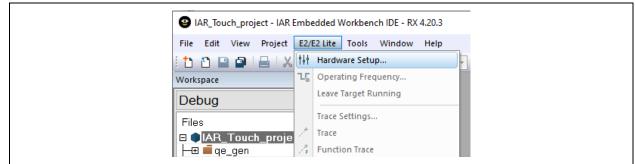


Figure 3-22 Selecting [Hardware Setup...]

- b. Select the [MCU] tab in the [Hardware Setup] dialog box and set the following items.
  - [ICLK frequency:] in [MCU] Enter "48.000".
  - [Power supply] "Power target from the emulator (MAX 200mA)" If the checkbox is ticked (√), cancel the selection by clicking on the checkbox.
     Click on [OK].

MCU		Communication		
Mode pin setting: Single-chip mode	EXTAL frequency: 22.0000 MHz	O JTAG clock:		MHz
	IVII IZ	FINE baud rate:	1.5M	✓ bps
Byte order: Little-endian	ICLK frequency: 48.0000 MHz			
	48.0000 MHz	System		
Register setting: Single-chip mode		Debug the program re	e-writing the PROGRAM R	MO
		Debug the program re	e-writing the DATA FLASH	
Change startup bank		Emulator mode:		
Startup bank		Trace		$\sim$
Bank 0				
E-t		Download		
External memory areas: A. Byte order BU	Edit	Erase flash ROM before	ore download	
A. Dyle order DO	Edit	Erase data flash ROM	I before download	
		Power supply		
		Power target from the	emulator (MAX 200mA)	
		○ 1.8 V		_
		⊚ 3.3 V		
		0 5.0 V		
Allow clock source change when writing internal to	flash	ID code		
Allow clock source change when whiling internal in				
Work RAM start address (0x500 bytes used):	0x1000	FFFFFFFFFFFFFFFFFF	FFFFFFFFFFFFFFFF	

Figure 3-23 Hardware Setup - [MCU] Tab



4. Build the project with the files for tuning included.

Right-click on the project name in the [Workspace] window, then select [Rebuild All] to perform building. After building has finished, confirm that no error has occurred in the [Build] window or status bar at the bottom of the main window.

🎦 🗋 🔛 🗃 🔚 🗶 🛍 ( Vorkspace	) 5C  ▼#×	< Q > \$ ₩ < §	> e >     • • • •	•				
Debug	~							
Files B <b>IAR_Touch_project -</b>	۰ ،							
—⊞ 🛋 qe qen	Options							
- Renesas AP	Make							
Buildinfo.ipcf	Compile							
- El Cutput	Rebuild All							
	Clean							
	C-STAT Static Analysis	>						
	Stop Build							
IAR Touch project	Add	>		Build				<b>-</b>
	Remove				essages			L L
ebug Log	Rename		*		otal number of errors: 0		After build	d
Log Wed Nov 30, 2022 11:04:10	Version Control System	>		T	otal number of warnings: 0			
Workbench 4.20.3 (C\Prog	Open Containing Folder			<				
Systems\Embedded Work bin\pproc.dll)	File Properties			Ready		Errors 0, Warn		System CAP NUM OVR
binyxproc.uij	Set as Active					1 St	atus bar	
uild			-	×				
Messages			File Li					
	Before bui	Id						

Figure 3-24 Building Including Files for Tuning



#### 4. Tuning Touch Sensors

Connect RSK-RX140 with the host PC. Start debugging of the IAR EWRX project and perform automatic tuning processing for touch sensors in the standalone-version QE.

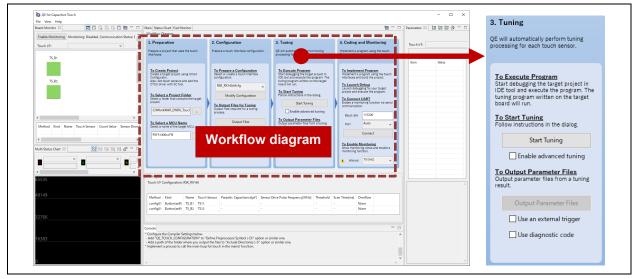


Figure 4-1 Standalone-Version QE - Workflow Diagram: <Step 3> Tuning

#### 4.1 Connecting the RSK-RX140 Board with a PC

Connect the RSK-RX140 board with a host PC as shown in Figure 4-2. A center-positive power supply with a stabilized DC output (min. 10 W) must always be used as the power supply for connection to the board.

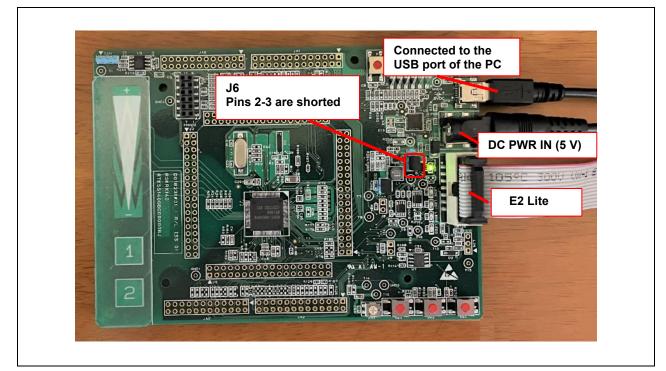


Figure 4-2 Connecting the RSK-RX140 Board



## 4.2 Starting Tuning

Execute the tuning program with IAR EWRX to start tuning in the standalone-version QE.

1. Execute the tuning program with IAR EWRX.

- (1) Select [Download and Debug] from the [Project] menu and start debugging.
- (2) After debugging has started, execution is stopped in the main() program.
- (3) Select [Go] from the [Debug] menu and start execution.

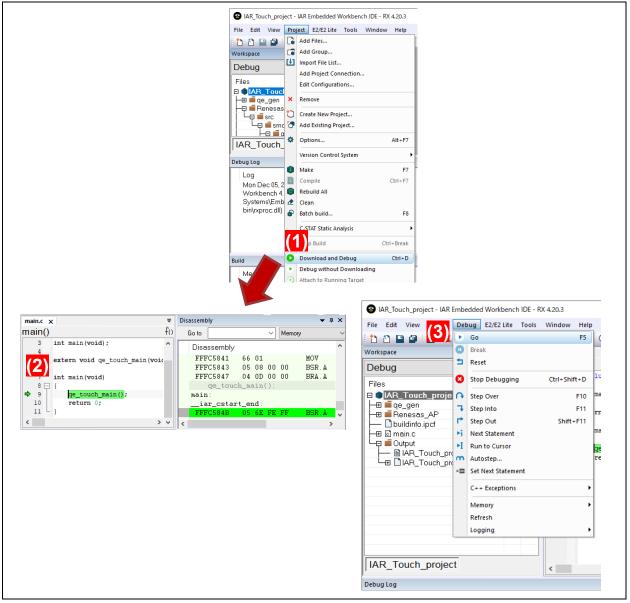


Figure 4-3 <Step 3> Tuning: Executing the Program



2. Start automatic tuning in the standalone-version QE.

Click on [Start Tuning] in the workflow to start automatic tuning in the standalone-version QE. For the interactive steps in automatic tuning processing, follow the instructions in the dialog box that provides guidance.

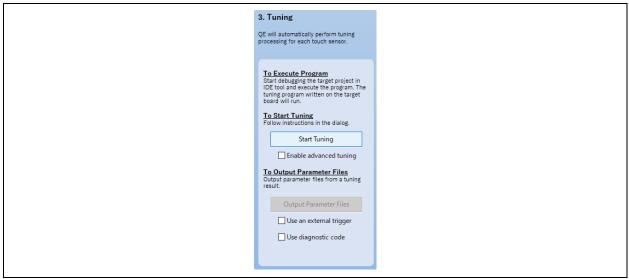


Figure 4-4 Automatic Tuning Processing

a. Confirm the "Automatic Tuning Processing" dialog box for guidance in the automatic tuning process as required. Normally, no manipulations are required during the initial stages of tuning.

E Automatic Tuning Processing	×
1/7: QE is beginning the tuning process. During the tuning process, please do not touch the se the QE Tuning Program.	ensors on the target board until instructed by
	Cancel Help

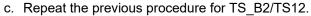
Figure 4-5 "Automatic Tuning Processing" - Initial Tuning

- b. If the touch sensitivity has been found to be measurable after a number of automated steps, a dialog box with a notification similar to that shown in the figure below will appear. This is the first interactive step in the tuning process.
  - Touch the sensor (TS\_B1/TS11) that is indicated in the dialog box.
  - While touching the sensor, the bar graph will grow to the right and the counted numerical value will increase.
  - While continuing to touch the sensor, press any key on the PC keyboard to secure the measured value.

Automatic Tuning Processing	×
5/7: QE will now measure touch sensitivity for (TS_B1, TS11 @ config01). In this step please use normal touch pressure on the sensor for once. Press any key on the PC keyboard to accept the sensitivity measurement. TS_B1, TS11 @ config01: 15252	
Cancel Help	

Figure 4-6 "Automatic Tuning Processing" - Measuring the Touch Sensitivity (TS\_B1)





Automatic Tuning Processing	×
6/7: QE will now measure touch sensitivity for (TS_B2, TS12 @ config01). In this step please use normal touch pressure on the sensor for once. Press any key on the PC keyboard to accept the sensitivity measurement. TS_B2, TS12 @ config01: 15382	:
Cancel	Help

Figure 4-7 "Automatic Tuning Processing" - Measuring the Touch Sensitivity (TS\_B2)

d. Once automatic tuning has been completed for both buttons, the thresholds can be checked in a dialog box like that shown below. These thresholds are used by the middleware to determine if touch events have occurred.

	c tuning e retriec	proce d. If the	ere are	continued	overflow			rs are indicate please consult	
lect the target	Method	Kind	Name	Touch Sensor	Threshold	Overflow	Warning / Error		
	config01	Button	TS_B1	TS11	1968				
	config01	Button	TS_B2	TS12	2134				
Retry Continue	the Tuning	Process	1		1			Cancel	Help

Figure 4-8 "Automatic Tuning Processing" - Displaying the Measurement Results

3. Stop tuning.

Click on the [Continue the Tuning Process] button in the dialog box that shows the thresholds of the touch sensors. The automatic tuning process is finished at this point and the debug session with the target board is cut off.

	c tuning e retried	proce d. If the	ere are	continued	overflow			ors are indicated, those please consult the Renes
Select the target	Method config01 config01	Button	TS_B1		Threshold 1968 2134	Overflow	Warning / Error	
Retry Continue	the Tuning	Process						Cancel Help

Figure 4-9 "Automatic Tuning Processing" - Continuing the Tuning Process



## 4.3 Acquiring the Tuning Result

Acquire the tuning result in the standalone-version QE.

1. Stop debugging of IAR EWRX.

Select [Stop Debugging] from the [Debug] menu.

IAR_Touch_project - IAR	Emb	edded Workbench IDF - F	RX 4 20 3		
File Edit View Project				Help	
		Go		F5	C
Workspace	0	Break			F
Debug	1	Reset			
Files	8	Stop Debugging	Ctrl+Shift	+D	lu
🗆 🌒 IAR_Touch_proje	•	Step Over		10	ma
-⊞ ■ qe_gen  -⊞ ■ Renesas_AP	7	Step Into		11	rn
- Duildinfo.ipcf		Step Out	Shift+I		
-⊞ ioi main.c		Next Statement			ma
	_	Run to Cursor Autostep			qe re
		Set Next Statement			10
		C++ Exceptions			
	-			-	
		Memory Refresh			
		Logging		•	
IAR_Touch_project	دل t				
, '			<		_
Debug Log					

Figure 4-10 Tuning Related

 Click on the [Output Parameter Files] button in the standalone-version QE. In the "The output folder of parameter files" dialog box, select a folder and click on [Select Folder]. In the example below, the "qe\_tuning\_r1" folder was created in advance so it could be selected here.

QE will automatically perform tuning processing for each touch sensor.	🗞 The output folder of parameter file	es			>
	$\leftrightarrow$ $\rightarrow$ $\checkmark$ $\uparrow$ $\blacksquare$ « (C:) Window	ws > Work > IAR_EWRX_Touch :	> RSK_RX140 >	۹ 5 ۲	Search RSK_RX140
To Execute Program	Organize 🔻 New folder				BEE 👻 ?
Start debugging the target project in IDE tool and execute the program. The	Name	Date modified	Туре	Size	
tuning program written on the target	settings	2022/12/05 15:07	File folder		
board will run.	Debug	2022/12/05 13:58	File folder		
To Start Tuning	QE_FirstAdjustments	2022/12/05 14:08	File folder		
Follow instructions in the dialog.	qe_gen	2022/12/05 14:10	File folder		
	📙 qe_tuning_r1	2022/12/05 15:34	File folder		
Start Tuning	QE-Touch	2022/12/05 15:26	File folder		
	settings	2022/12/05 14:01	File folder		
Enable advanced tuning	src	2022/12/05 13:57 2022/12/05 15:16	File folder File folder		
To Output Parameter Files Output parameter files from a tuning result. Output Parameter Files	trash	LULU 12/03 13/10	The folder		
Use an external trigger	Folder: qe_tur	ning_r1			
Use diagnostic code				Sele	ect Folder Cancel

Figure 4-11 Outputting the Tuning Result



3. Reflect the values from tuning of the touch interfaces in the IAR EWRX project. Open the folder where the files are saved and copy the following three files to the "qe\_gen" folder in the workspace folder. If the file contents before tuning are unnecessary, overwrite the files. If the file contents are necessary, save the files before performing copy.

	← → × ↑ 🔤 « Work > IAF	R_EWRX_Touch > RSK_RX140 > 0	qe_tuning_r1	5 V	⊘ Searce	h qe_tuning_r1
	Name	Date modified	Туре	Size		
Files to be copied>	ge_touch_sample.c	2022/12/05 16:51	C File		4 KB	
_touch_define.h	ge_touch_define.h	2022/12/05 16:51	H File		5 KB	
touch_config.h	qe_touch_config.h	2022/12/05 16:51	H File		4 KB	
	ge_touch_config.c	2022/12/05 16:51	C File		7 KB	
e_touch_config.c	4 items 3 items selected 13.9 KB	Touch\RSK_RX140\qe_gen			_	
}_touch_config.c	4 items 3 items selected 13.9 KB			<b>č</b>	Search qe_ge	□ × ~ 0
_touch_config.c	4 items 3 items selected 13.9 KB	_Touch\RSK_RX140\qe_gen		ق ب Size	-	□ × ~ 0
_touch_config.c	4 items 3 items selected 13.9 KB 4 items C:\Work\IAR_EWRX, File Home Share View $\leftarrow \rightarrow \checkmark \uparrow \square \iff Work > IAR$	_Touch\RSK_RX140\qe_gen	je_gen ∽		-	□ × ~ 0
e_touch_config.c	4 items 3 items selected 13.9 KB 4 items $\rightarrow$ C:\Work\IAR_EWRX File Home Share View $\leftarrow \rightarrow \sim \uparrow$ $\checkmark$ $\forall$ Work > IAR Name	_Touch\RSK_RX140\qe_gen _EWRX_Touch > RSK_RX140 > o 	re_gen ∽ Type		 Search qe_ge	□ × ~ 0
_touch_config.c	4 items 3 items selected 13.9 KB 4 items $\neg$ $\neg$ $\neg$ C:\Work\IAR_EWRX. File Home Share View $\leftarrow \rightarrow \checkmark \uparrow \square \ll Work > IAR$ Name $\square qe_touch_sample.c$	Touch\RSK_RX140\qe_gen	<b>je_gen ↓</b> Type C File		Search qe_ge	□ × ~ 0

Figure 4-12 Copying the Files with Tuned Values

4. Build the IAR EWRX project and generate modules in which the tuned values of touch interfaces are reflected. For the method of building, refer to section 3.4.4.



#### 5. Coding and Monitoring

Using the module of section 4.3 4 that was built with the parameter files obtained in automatic tuning processing, perform monitoring in the standalone-version QE.

A facility for scanning the state of touch sensors is implemented in a function included in "qe\_touch\_sample.c", which is a tuning file obtained in "2. Configuration" of the workflow diagram of the standalone-version QE. Operation can be checked through monitoring without any particular requirements to change the program code.

### 5.1 Monitoring

Execute the program built with IAR EWRX, according to the procedure in section 4.2.1.

1. Start monitoring.

Click on the [Connect] button under "<u>To Connect UART</u>" in the workflow diagram. Wait for a while until connection is completed.

ard Monitor 12 🛛 😨 🗈 🗔 🗔 🖬 😁 🗖	Workflow Diagram				Parameters 22 1 1 1 1 1 1 1 1 1		Implement a program using the touch. Then, confirm a behavior of touch
ouch l/F:	1. Preparation	2. Configuration	3. Tuning	4. Coding and Margaret	Terret I/E		interfaces and make fine adjustments.
TS_B1	Prepare a project that uses the touch interfaces.	Prepare a touch interface configuration.	QE will automatically perform tuning processing for each touch sensor.	Implement a program Then, confirm a behavior or noch interfaces and make fire adjustments.	Item Value		
TS,82 Method End Name Touch Sensor Court Water Sensor Diver 20 Status David III O CO CO CO Sensor Sensor Diversity 20 Sensor David III O CO CO CO Sensor Sensor Diversity	Laber 1 for each of the set of th	Research Contention the Content of the Mark Content o	Hender Stand and Stan	Image: Section of the section of t	Image: sector		To Implement Program Implement a program using the touch interfaces and build the project. To Launch Debugging for your target project and execute the program. To Connect UART Enable a monitoring function via serial communication.
							Baud rate 115200
	Touch VF Configuration: RSK_RX140					~	Port Auto V
	Method Kind Name Touc config01 Button(self) TS_B1 TS11 config01 Button(self) TS_B2 TS12	h Sensor Parasitic Capacitance[pF] Sensor 	r Drive Pulse Frequency(MHz) Threshold - -	Scan Time(ms) Overflow - None - None			Connect
	Console				_		To Enable Monitoring
	* Configure the Compiler Settings below. - Add *QE_TOUCH_CONFIGURATION* to	"Define Preprocessor Symbol (-D)" option or s ut the files to "Include Directories (-I)" option p for touch in the main() function.	similar one. or similar one.		^		Show monitoring views and enable a monitoring function.

Figure 5-1 <Step 4> Coding and Monitoring: Connect

2. When connection is established, "Connecting via serial communication (UART/USB)" is displayed for "Communication Status" in the Board Monitor view.

Board Monitor 🖾			∎ □ □
Enable Monitoring	Monitoring: Disabled	Communication Status: Connecting via serial communication (UART /	USB)
Touch I/F:		~	¥
TS_B1			^

Figure 5-2 State of Serial Communications



3. Click on [Enable Monitoring]. The display is switched between "Monitoring: Disabled" and "Monitoring: Enabled".

Board Monitor 🛙 🗖 🗖	G C 🗈 🛅 🗖 🗖	QE for Capacitive	Touch	
Enable Monitoring Monitoring: Disabled, Communication Status: Connecting via serial communicatication Status:	cation (UART / USB)	Board Monitor 🛛	Monitoring: Enabled, Communication Status: Connecting via serial communication (	
TS_B1	^	Touch I/F: TS_B1	v	~

Figure 5-3 Enabling the Monitoring Function

4. Touch the button TS\_B2 on the target board. In the Board Monitor view, a finger icon on the button like that in the image below indicates that the button is being touched.

Board Monitor 🛙 🔲 📮 📮 📮 📮 🗮 🗖
Enable Monitoring Monitoring: Enabled, Communication Status: C
Touch I/F:
TS_B1
<:

Figure 5-4 Board Monitoring: Finger Icon

5. To see a graphical representation of the counted values, click on the [Status Chart] tab as shown below.

Touch I/F:	✓ Sync a selection		
Coloretation to use 1/E to use			
Select the touch I/F to me	nitor		

Figure 5-5 Selecting the [Status Chart] Tab

— Select "TS\_B1@ config01" from the pull-down menu.

Mai	Status Cl	art 🛛 🏼 P	ad Mon	itor			
	uch I/F:				¥	Sync a sele	ection
Se	ect the tr	_B1 @ cor _B2 @ cor	nfig01 nfig01				
Т	ouch Positio	on:		Reference	Value:		Threshold:

Figure 5-6 Selecting "Touch I/F"



— The graph will begin to display a running depiction of the values. The increases in the counted values while the TS\_B1 button on the target board is being touched can be confirmed from the graph. The green line indicates the threshold, which is used by the "rm\_touch\_qe" FIT module to determine whether the button is being actuated, that is, touched. The red belt at the bottom of the graph indicates the periods over which the counted value was exceeding the threshold so that the button is judged to have been touched.

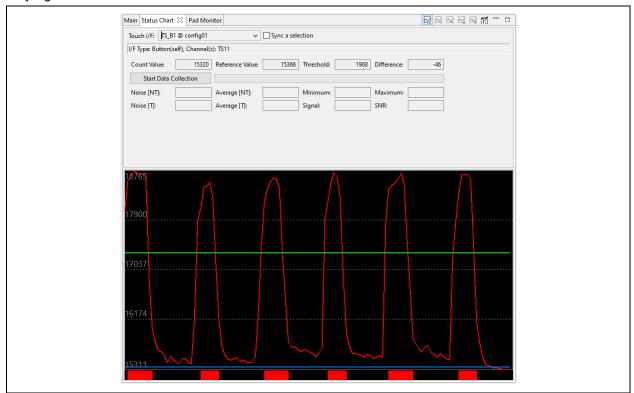


Figure 5-7 [Status Chart] Tab



6. To see a graphical representation of counted values that indicate multiple touch sensors being touched, use the [Multi Status Chart] tab. It will be displayed when the touch interface is selected.

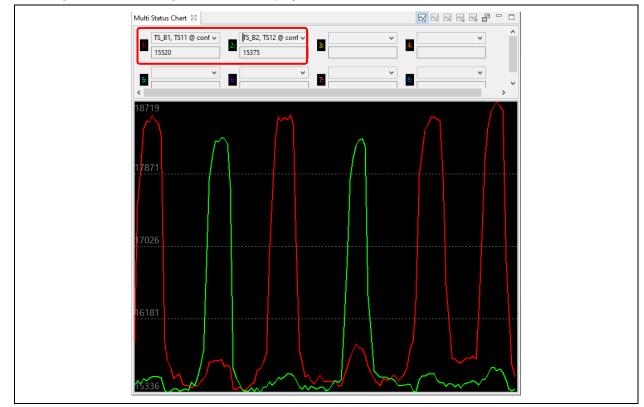


Figure 5-8 [Multi Status Chart] Tab

#### 5.2 Disconnecting Serial Communications

#### 1. Stop monitoring.

When the monitoring function is enabled, click on [Enable Monitoring] and confirm that the monitoring function has become disabled.

Board Monitor 🛛		
Enable Monitoring	Monitoring: Disabled, Communication Status: Connecting via serial communication (UART / USB)	^
Touch I/F:	v	¥
TS_B1		^

Figure 5-9 State of Serial Communications



2. Disconnect the serial connection.

Click on the [Disconnect] button under "<u>To Connect UART" under</u>" 4. Coding and Monitoring" of the workflow diagram. When disconnection is completed, "Disconnected from ¥¥.¥COMx." is displayed in the Console view, and the label on the [Disconnect] button changes to [Connect].

4. Coding and Monitoring		4. Coding and Monitoring
mplement a program using the touch. Then, confirm a behavior of touch nterfaces and make fine adjustments.		Implement a program using the touch. Then, confirm a behavior of touch interfaces and make fine adjustments.
To Implement Program Implement a program using the touch interfaces and build the project.		To Implement Program Implement a program using the touch interfaces and build the project.
To Launch Debug Launch debugging for your target project and execute the program.		To Launch Debug Launch debugging for your target project and execute the program.
To Connect UART Enable a monitoring function via serial communication.		To Connect UART Enable a monitoring function via serial communication.
Baud rate 115200	Console 🗖 🗖	Baud rate 115200
Port Auto 🗸	Failed to connect to the serial port. Connected to \\.COM5 Disconnected from \\.COM5.	Port Auto v
Disconnect		Connect
To Enable Monitoring Show monitoring views and enable a monitoring function.	<	To Enable Monitoring Show monitoring views and enable a monitoring function.
🔒 Interval 75 [ms] 🗸		🚹 Interval 75 [ms] 🗸

Figure 5-10 <Step 4> Coding and Monitoring: Disconnect



## **Revision History**

		Description		
Rev.	Date	Page	Summary	
1.00	Dec.22.22	—	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 is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

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

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

#### Notice

- Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- 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 products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
   Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
- (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

#### **Corporate Headquarters**

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

#### Trademarks

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

## **Contact information**

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