# Renesas e<sup>2</sup> studio 2021-07 or Higher User's Manual: Quick Start Guide

Renesas MCU RE Family

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics

Rev.1.00 Jul 2021

#### 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.
- 13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
- 14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
- (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

## **Corporate Headquarters**

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

#### www.renesas.com

#### Trademarks

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

## **Contact information**

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

## 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 system-evaluation test for the given product.

## Table of Contents

1. Over	view	1
1.1 Sy	stem Configuration	2
1.2 Sy	stem Requirements	2
1.2.1	Hardware Environment	2
1.2.2	Operating Environment	2
1.3 Su	pported Toolchain	2
1.4 Su	pported Emulator	2
2. Instal	lation	3
2.1 Ins	stalling e <sup>2</sup> studio and the Software Package	4
2.1.1	Installing e <sup>2</sup> studio	4
2.1.2	Installing GNU Arm Embedded Toolchain	
2.1.3	Installing the Software Package for RE	
2.2 Un	-installation of e <sup>2</sup> studio IDE	
2.3 Up	dating e <sup>2</sup> studio	
2.4 Up	dating Software Package	
	ct generation	
	enerating new Project with Smart Configurator	
	enerating new Executable Project without Smart Configurator	
3.3 Ge	enerating and Using a RE Static Library	
3.3.1	Creating a Static Library Project	
3.3.2	Creating an Executable Project using the existing Static Library	
	port Existing Projects into Workspace	
	nfiguration Editor	
3.5.1	Summary Page	
3.5.2	BSP Page	
3.5.3	Clocks Page	
3.5.4	Pins Page	
3.5.5	Stacks Page	
3.5.6	Interrupts Page	56
3.5.7	Components Configuration Page	60
	ing	61
	ild Option Settings	
4.1 Bu 4.1.1		
	Recommended Build settings	
	ild a Sample Project	
4.3 EX	port Build Configuration Settings	
5 Debu	g	70
	ange Existing Debug Configurations	
	eate New Debug Configurations	
	unch Bar	
	sic Debugging Features	
5.4.1	Breakpoints View	
5.4.1	Expressions View	
5.4.2	Registers View	
5.4.3 5.4.4	Memory View	
5.4.4 5.4.5	Disassembly View	
5.4.5 5.4.6	Variables View	
5.4.0 5.4.7	Eventpoints View	
5.4.7	IO Registers View	
5.4.0		

93
96
99

## 1. Overview

Renesas e<sup>2</sup> studio is the Integrated Development Environment for Renesas embedded microcontrollers. e<sup>2</sup> studio is based on the industry-standard open-source Eclipse IDE framework and the C/C++ Development Tooling (CDT) project, covering build (editor, compiler, and linker control) and debug phases with an extended GNU Debug (GDB) interface support.

e<sup>2</sup> studio IDE includes a configurator for Software Package which is an optimized software package designed to provide easy to use, scalable, high-quality software for embedded system design.

e<sup>2</sup> studio IDE with the RE configurator provides multiple Graphical User Interface (GUI) wizards to autogenerate code, configure drivers, configure build and debug options, and run the applications created. Driver documentation is integrated in the form of tooltips, which are available in the code editor view.

This chapter describes the system configuration and operating environment for e<sup>2</sup> studio IDE to develop applications for the RE family series microcontrollers.

The outline of this document is shown in the following table. It is recommended to read part of chapter 3, 4 and 5 to easily create or import a project and run it.

In this document, Software Development Kit is referred to as SDK. The SDK can be configured using the FSP Configuration of e<sup>2</sup> studio 2021-07.

Chapter	What user would learn	Guide to create and run a project
1. <u>Overview</u>	Development environment requirement	Δ
2. Installation	Steps to install IDE, especially RE development $\Delta$ environment and toolchain	
3. Project generation	Creating project with/without Software Package	O (sections 3.1 and 3.4)
	and configuring Software Package for RE MCU family	$\Delta$ (Other sections)
4. <u>Building</u>	Configuring build options and building the project	O (sections 4.2)
		$\Delta$ (Other sections)
5. <u>Debug</u>	Using basic debugging features of e <sup>2</sup> studio	O (section 5.1)
		$\Delta$ (Other sections)
6. <u>Help</u>	Using some important items of Help menu $\Delta$	

#### Table 1-1. Content of this document

O: Recommended reading.

 $\Delta$ : Optional reading.



## 1.1 System Configuration

Below is an example of a typical system configuration.



Figure 1-1 System Configuration

## 1.2 System Requirements

#### 1.2.1 Hardware Environment

Processor: At least 1GHz (support hyper-threading/multi-core CPU	
Main Memory: At least 2GB of free memory space	
Harddisk Capacity:	At least 2GB of free space
Display:	Resolution at least 1,024 x 768; at least 65,536 colors
Interface:	USB 2.0 (High-speed/Full-speed). High-speed is recommended.

#### 1.2.2 Operating Environment

Architecture	Windows	e² studio
64-bit version	Windows 8.1, Windows 10	2021-07

## 1.3 Supported Toolchain

GNU Arm Embedded Toolchain (version: GCC V.6 GNU 6-2017-q2-update)

IAR C/C++ Compiler for ARM (version: 8.50.x or higher).

## 1.4 Supported Emulator

Segger J-Link, Segger J-Link OB, Renesas E2, Renesas E2 Lite



## 2. Installation

The table below shows the outline of this chapter. To set up the environment, it is recommended to read chapter 2.1. Please read the other sections when you need to update or uninstall the environment.

#### Table 2-1. Outline of this chapter

Chapter	Guide for Installation
2.1 Installing e <sup>2</sup> studio and the Software Package	0
2.2 Un-installation of e <sup>2</sup> studio IDE	Δ
2.3 Updating e <sup>2</sup> studio	Δ
2.4 Updating Software Package	Δ

O: Recommended reading,

 $\Delta$ : Optional reading.



#### 2.1 Installing e<sup>2</sup> studio and the Software Package

This section describes the installation of the following components.

- e<sup>2</sup> studio 2021-07
- GCC ARM embedded compiler
- Software Package for RE.

#### 2.1.1 Installing e<sup>2</sup> studio

- 1. Download e<sup>2</sup>studio 2021-07 offline installer from <u>https://www.renesas.com/e2studio</u>
- 2. Run the e<sup>2</sup> studio installer to invoke the e<sup>2</sup> studio installation wizard page.
- If e<sup>2</sup> studio was installed on your PC, the options to modify, remove the existing version and install e<sup>2</sup> studio to a different location will be shown. It is possible to install multiple versions of e<sup>2</sup> studio by selecting Install to a different location. Click the [Next] button to continue.

🗟 Renesas e² studio 2021-07 Setup	— 🗆 X
Renesas e <sup>2</sup> studio 2021-07 Setup	RENESAS
e2 studio is already installed.	
What do you want to do?         Modify         Add or remove features in the existing installati         Location:         Image: Second Sec	ion.
<u>v202106172157</u> User: All Users < <u>B</u> ack	Next > Install Cancel

Figure 2-1 Installation of e<sup>2</sup> studio – Install Type

#### 4. Welcome page:

Users may use the default folder or change the folder by clicking [Change...]. Click [Next] to continue.

**Note**: Multi-byte characters cannot be used for  $e^2$  studio installation folder name, project name and its folder, and source file name

🗟 Renesas e <sup>2</sup> studio 2021-0	7 Setup	_		×
Renesas e <sup>2</sup> studio 2021-0	7 Setup	RENE	5 <b>/</b>	S
() Welcome	Install directory ready Install Location: C:\Renesas\e2_studio			^
Device Families	[Change]	,		
Extra Features				
Customise Features	Prerequisite software already installe	:a		
Additional Software	Internet connection available			
Licenses	Change Proxy Settings			
Shortcuts	Change Proxy Searings			
Summary	Ready to install			
Installing Results	Software to install: • Renesas e2 studio • Java Runtime v11.0.0 • IAR Plugin Manager v1.1.0.2020072 • OpenJDK v11.0.8.202009102051 • openjfx.media.feature v11.0.0.2019 • openjfx.standard.feature v11.0.0.2019 • openjfx.swing.feature v11.0.0.201901	901231300 01901231300 901231300		~
<u>v202106172157</u>	User: All Users < <u>B</u> ack N	ext >Install	Cance	el

Figure 2-2 Installation of e<sup>2</sup> studio – Welcome page

#### 5. Device Families page:

Select the checkbox for "RE". Checkboxes of other device families are optional. Click [Next] to continue.



Figure 2-3 Installation of e<sup>2</sup> studio – Device Families page

#### 6. Extra Features page:

Select Extra Features (i.e. Language packs, SVN & Git support, RTOS support...) to install. For the non-English language menu, please select Language packs at this step.

Click [Next] to continue.

🔜 Renesas e <sup>2</sup> studio 2021-0	Renesas e² studio 2021-07 Setup — 🗆 🗙				
Renesas e <sup>2</sup> studio 2021-0 Select the extra features you		RENESAS			
Welcome Device Families		Japanese Language Support			
Extra Features		Chinese (Simplified) Language Support			
Customise Features Additional Software	□ ♀	Chinese (Traditional) Language Support			
Licenses Shortcuts		Git Integration Git SCM Support			
Summary Installing Results		Terminals ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet			
		RTOS FreeRTOS & OpenRTOS Debug Support			
	Select All				
<u>v202106172157</u>	User: All Users	< <u>B</u> ack <u>N</u> ext > <u>Install</u> Cancel			

Figure 2-4 Installation of e<sup>2</sup> studio – Extra Features page

#### 7. Customise Features page:

Ensure that "Renesas RE Family Support" and "Renesas FSP Smart Configurator" is checked. Click [Next] to continue.

Renesas e² studio 2021-07 Setup — 🗆 🗙				
Renesas e <sup>2</sup> studio 2021-	07 Setup	ΔS		
Welcome	Select the components you want to install. Renesas KL/8 tamily support requiring .Net Framework			
Device Families Extra Features	Renesas RX Family Support (Renesas RX Family Support to allow project generation and build of executable projection in the projects, debug only projects and IAR projects.	ects,		
	Renesas RX Family Support (requiring .Net Framework)     Renesas RX family support requiring .Net Framework			
Geatures	⊕ Renesas RH850 Family Support     Renesas RH850 family support to debugging and GHS toolchain projects.			
Additional Software Licenses	유 Renesas RE Family Support ( Renesas RE family support to allow project generation and build of executable projection in the project of the projects of the projects and IAR projects.	ects,		
Shortcuts	⊕ Renesas Synergy Family Support      Renesas Synergy family support to allow project generation, build & debug			
Summary				
	TM Terminal (4.6.0) ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet	~		
	Select All Optional			
	Size of install: 958.4 N	MB		
<u>v202106172157</u>	User: All Users < <u>B</u> ack <u>N</u> ext > <u>I</u> nstall	Cancel		

Figure 2-5 Installation of e<sup>2</sup> studio – Customize Features page

#### 8. Additional Software page:

Select additional software (i.e. compilers, utilities, QE...) and click [Next] to continue.

**Note:** With no Internet access available, additional software installation can be skipped because the software catalog cannot be downloaded. The additional software can be installed later.

🛃 Renesas e <sup>2</sup> studio 2021-0	🗟 Renesas e² studio 2021-07 Setup — 🗆 🗙			
Renesas e <sup>2</sup> studio 2021-0 Select the additional softwa	7 Setup re you wish to install	<b><i>ENESAS</i></b>		
Welcome	Renesas QE (0) Renesas Toolchains & Utilities (0) GCC	Foolchains & Utilities (1)		
Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Summary	GNU ARM Embedded CNU ARM Embedded 6 2017q2 6.0.0.2017q3 GNU ARM Embedded 6 2017q2 Download size: 82.6 MB			
Installing Results		82.6 MB download required		
<u>v202106172157</u>	User: All Users < <u>B</u> ack <u>N</u> ext >	Install Cancel		

Figure 2-6 Installation of e<sup>2</sup> studio – Additional Software page



#### 9. License page:

Read and accept the software license agreement. Click [Next] to continue. Please note that user must accept the license agreement, otherwise installation cannot proceed.

🗟 Renesas e² studio 202	1-07 Setup	— D X
Renesas e <sup>2</sup> studio 202	I-07 Setup	RENESAS
Welcome	Please read and accept the	
Device Families       Renesas e2 studio         Device Families       OpenJDK License Agreem         Customise       ARM DS-5 Toolchain Inter         Features       Additional         Software       Software         Dicenses       Licenses         Shortcuts       Summary         Installing       Contract		License Terms and Conditions for RENESAS e2 studio This Renesas e2 studio license agreement ("Agreement") is between the entity on whose behalf you are entering into this Agreement ("Client") and Renesas Electronics Corporation, a Japanese company with its registered office at 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan ("Renesas"). YOU SHOULD READ THIS AGREEMENT CAREFULLY, AS IT CONSTITUTES A BINDING CONTRACT BETWEEN CLIENT AND RENESAS. The Renesas IDE Software (defined below) is intended for commercial use by a company or corporation only and is not designed, developed or produced for any private use or purpose. If you are an individual, or you intend to install the Renesas IDE Software on behalf of an individual, or the Renesas IDE Software is expected to be used for a private purpose directly or indirectly, you should click "No" on the installer. Otherwise, hu clicking the "I accent" button or other button or
Results	✓ accept the terms of the	Software Agreements Print all
v202106172157	User: All Users	< <u>B</u> ack <u>N</u> ext > <u>Install</u> Cancel

Figure 2-7 Installation of e<sup>2</sup> studio – Licenses page

#### 7. Shortcuts

Select shortcut name for the start menu and click [Next] to continue.

**Note:** If  $e^2$  studio was installed in another location, it is recommended to rename it to distinguish it from the other  $e^2$  studio(s).



							×
studio 2021-	07 Setup		R	ENE	ES	S	5
ome	Shortcuts to important pro	grams and files v	vill be created in t	he following loca	ations:		
e Families Features	🗹 ln start menu group:	Renesas Electro	nics e2 studio202	1-07 FSP100			
mise res					<u>€</u> <u>R</u>	estore De	afaul
ional are							
ses							
tcuts							
nary							
ing							
5							
06172157	User: All Users	< <u>B</u> ack	<u>N</u> ext >	<u>I</u> nstall		Cance	el .
	e Families Features mise res ional are ses tcuts hary ing	e Families Features mise res ional are ses tcuts hary ing s 200172157 User: All Users	e Families Features mise res ional are ses tcuts hary ing s 00172157 User: All Users < <u>Back</u>	ome Shortcuts to important programs and files will be created in t   a Families ☑ In start menu group:   Features Renesas Electronics e2 studio202   mise are   ional are   ses tcuts   nary ing   is User: All Users	Shortcuts to important programs and files will be created in the following location of the features in the following location of the features in the following location of the features is a set of the feature of the	Shortcuts to important programs and files will be created in the following locations:   e Families   Features   mise   res   ional   are   ses   tcuts   nary   ing   is     06172157   User: All Users     Shortcuts to important programs and files will be created in the following locations:     Renesas Electronics e2 studio2021-07 FSP100     Renesas Electronics e3 studio2021-07 FSP100     Renesas Electro	e Families Features mise res ional are ses tcuts nary ing ts D6172157 User: All Users VI N start menu group: Renesas Electronics e2 studio2021-07 FSP100  Renesas Electronics e2 studio2021  Renesas



#### 8. Summary page:

The components list to be installed is shown. Please confirm the contents and click the [Install] button to install the Renesas  $e^2$  studio IDE.



Figure 2-9 Installation of e<sup>2</sup> studio – Summary page

9. Installing... page:

The installation is now performed. Based on the selected items of Additional Software, new dialogs are opened to proceed with installation for these software items.

#### 10. Results page:

Installation results are listed here. Please note if any errors are shown.

Click [OK] to complete the installation.

	🔜 Renesas e² studio 2021-	07 Setup			_		×
ľ	Renesas e <sup>2</sup> studio 2021-(	07 Setup		RE	NE	50	S
	Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing	Installation of <b>e2 studio</b> is complete Please click <b>OK</b> to close. Launch e2 studio? View Release Notes? View What's New?					
	<u>v202106172157</u>	User: All Users <	<u>B</u> ack <u>N</u>	Next >	ОК	Canc	el

Figure 2-10 Installation of e<sup>2</sup> studio – Results



#### 2.1.2 Installing GNU Arm Embedded Toolchain

The GNU Arm Embedded Toolchain can be installed during e<sup>2</sup> studio installation. Or after e<sup>2</sup> studio has been installed, the GNU Arm Embedded Toolchain can be installed separately.

To install GNU Arm Embedded Toolchain, follow these steps:

- 1. Download version 6-2017-q2-update of the GNU Arm Embedded Toolchain supported by Renesas RE (gcc-arm-none-eabi-6-2017-q2-update-win32.exe) from <u>https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads</u>
- 2. Run the installer to install the GNU Arm Embedded Toolchain on the host machine.
- 3. Select the installation language. Click [Yes] in the installation confirmation dialog.
- 4. Keep all default settings in the installation wizard.
- 5. When the "Install wizard Complete" dialog appears, check the box "Add path to environment variable", "Add registry information" and "Launch gccvar.bat". Then click [Finish] to complete the installation.

#### 2.1.3 Installing the Software Package for RE

The Software Package can be installed during e<sup>2</sup> studio installation. Alternatively, after e<sup>2</sup> studio has been installed, the Software Package can be installed separately.

To install the Software Package, follow these steps:

- (1) Visit the https://www.renesas.com/software-tool/re-software-development-kit
- (2) Download the latest SDK (e.g. "r01an5970xx0110-re-sdk.zip"). Unzip this file to get the latest software pack installer (e.g. "RE\_SDK\_Packs\_<version>.exe"). The Software Package Installer includes the driver, HTML User's Manual.
- (3) Make sure that a compatible  $e^2$  studio was installed and closed during this installation.
- (4) Run the Software packs installer and click [Next] to continue.
- (5) Click [I Agree] to accept the agreement.
- (6) Browse to the folder where e<sup>2</sup> studio is installed (e.g. C:\Renesas\e2\_studio) and click [Install].
- (7) Click [Finish] to finish the installation.

## 2.2 Un-installation of e<sup>2</sup> studio IDE

Users can uninstall e<sup>2</sup> studio program following the typical steps to uninstall a program in Window OS.

- 1. Click on [Start]  $\rightarrow$  [Control Panel]  $\rightarrow$  [Programs]  $\rightarrow$  [Programs and Features]
- 2. From the currently installed programs list, right-click on "Renesas e<sup>2</sup> studio" and select the [Uninstall] button.
- 3. Click [Uninstall] to confirm the deletion in the "Uninstall" dialog.

At the end of the un-installation, e<sup>2</sup> studio IDE will be deleted from the installed location and Windows shortcuts menu are removed. **Note:** If you have installed e<sup>2</sup> studio at multiple locations, you may not be able to find the uninstaller in "Program and Features" of Control Panel. In such cases, launch e<sup>2</sup> studio uninstaller located at: {e2 studio installed folder}/uninstall.exe

## 2.3 Updating e<sup>2</sup> studio

To update  $e^2$  studio, run the new version of  $e^2$  studio installer (either *Software Package with e<sup>2</sup> studio* installer or standard  $e^2$  studio installer). Please download the installer according to the instructions in Chapter 2.

Please note that you should not overwrite an existing installation. Before the IDE upgrade, users must uninstall the old version of e<sup>2</sup> studio. However, to keep both old and new e<sup>2</sup> studio versions, users can install the new e<sup>2</sup> studio version to a different location.

## 2.4 Updating Software Package

To update Software Package, run the new version of the Software Package installer. Download the installer according to Chapter 2.1.3.



## 3. Project generation

This chapter describes the creation of a new RE project. e<sup>2</sup> studio includes a wizard to help the quick creation of a new RE project. This is achieved by the ability of the wizard to match the project to a particular RE device and board.

The project generator can set up the pin configurations, interrupts, clock configurations and the necessary driver software.

As a pre-requisite, the Software Package and the toolchain must be installed on the host machine as described in chapter 2.

The following table shows the outline of this chapter. It is recommended to read chapters 3.1 and 3.4 to create or import a RE project in e<sup>2</sup> studio. Please read the other sections when you need information about creating project with Smart Configurator, creating a project without Smart Configurator and creating a static library project.

#### Table 3-1. Outline of this chapter

Chapter	What user would learn	Guide for project generation
3.1. <u>Generating new Project with Smart</u> <u>Configurator</u>	Creating a new project with Smart Configurator	0
3.2. <u>Generating new Executable Project</u> without Smart Configurator	Creating a new project without Smart Configurator	Δ
3.3. <u>Generating and Using a RE Static</u> Library	Creating a static library project with Smart Configurator and using it from an executable project	Δ
3.4. Import Existing Projects into Workspace	How to import an existing project into e <sup>2</sup> studio	0
3.5. <u>Configuration Editor</u>	Configuring the project with Smart Configurator	Δ

O: Recommended reading.

 $\Delta$ : Optional reading.



## 3.1 Generating new Project with Smart Configurator

1. Click [File]  $\rightarrow$  [New]  $\rightarrow$  [Renesas C/C++ Project]  $\rightarrow$  [Renesas RE] to open new project creation wizard.

File	Edit Navigate	Search	Project	Renesas Views	Run	Window Help		
	New			Alt+Shift+N >		Renesas C/C++ Proje	ct >	Renesas Debug
	Open File				<b>C</b> <sup>⁰</sup>	C/C++ Project		Renesas RE
۵,	Open Projects fro	om File Sy	stem		2	Project		
	Recent Files			>		Example		
	Close Editor			Ctrl+W		Other	Ctrl+N	
	Close All Editors			Ctrl+Shift+W	-			1

Figure 3-1 Open new project creation wizard

2. Select "All" "Renesas RE C/C++ SDK Project" template. Click [Next] to continue.



Figure 3-2 Project generation – Select template

3. Enter the project name. Click [Next] to continue.

	C/C++ SDK Project
Toject Nam	
<u>P</u> roject nar	ne
Tutorial	
🗹 Use <u>d</u>	efault location
Location:	D:\WorkSpace_for_RE\workspace_2021-07_FSP110\Tutorial
	Choose file system: default 🗸
You can dov	vnload more Renesas packs here

Figure 3-3 Project generation – Specify project name



- 4. In the device selection dialog, enter device and tool information:
  - Board: EK-RE01 1500KB
  - Toolchain version: Latest GNU Arm Embedded Toolchain approved for use with Renesas RE (e.g. GCC ARM Embedded 6.3.1.20170620)

Keep other fields as default. Click [Next] to continue.

🗐 Renesas RE	C/C++ SDK Project		- D X
Renesas RE C, Device and Too	/C++ SDK Project ols Selection		
Device Selection DEV Version: Board: Device: Language:	1.1.0 EK-RE01 1500KB	<ul> <li>Board Description</li> <li>Evaluation Kit REG</li> <li>Visit <u>https://www</u> manual, quick sta projects, etc.</li> <li>Device Details</li> <li>TrustZone</li> <li>Pins</li> <li>Processor</li> </ul>	
Toolchains GNU ARM Er ARM Compil	ler 6.12	Debugger J-Link ARM	~
?		< <u>B</u> ack	Next > Einish Cancel

Figure 3-4 Project generation – Select device and tools

5. In the build artifact dialog, select the build artifact: executable or library. Click [Next] to continue.

Note: FreeRTOS is not available for RE now

Build Artifact Selection	RTOS Selection		
<ul> <li>Executable</li> <li>Project builds to an executable file</li> <li>Static Library</li> <li>Project builds to a static library file</li> </ul>	No RTOS		

#### Figure 3-5 Project generation – Select build artifact

- 6. In the project template selection dialog, select the project template to be created:
  - Bare Metal Blinky: RE project with BSP and LED blinking sample program.
  - Bare Metal Minimal: RE project with BSP and no sample program.

Renesas RE C/C++ SDK Project
Renesas RE C/C++ SDK Project
Project Template Selection
Project Template Selection
<ul> <li>Bare Metal - Blinky Bare metal RE project that includes BSP and will blink LEDs if available. This project will initialize clocks, pins, stacks, and the C runtime environment. [Renesas.RE_sdk_re01_1500kb.1.1.0.pack]</li> <li>Bare Metal - Minimal Bare metal RE project that includes BSP. This project will initialize clocks, pins, stacks, and the C runtime environment. [Renesas.RE_sdk_re01_1500kb.1.1.0.pack]</li> </ul>
Code Generation Settings ☐ Use Renesas Code Formatter
< Back     Next >     Einish     Cancel

Figure 3-6 Project generation – Select project template

7. Click [Finish] to create the project.

You may be prompted to open the FSP Configuration perspective. Click [Yes] to open the perspective.

(In Eclipse, a 'perspective' is a predetermined arrangement of panes and views.)

e<sup>2</sup> studio creates a new project with various views, among them are the Project Explorer view, the Configuration editor and the Package view.

note: You must select the Pins tab in the Configuration Editor to see the package view.

Project Explorer 🐹 📄 🔄 🏹 🖇 🖳 🗖	[Tutorial] FSP Configuration	1 🖾		🎲 FSP Visualization 🗙 📮 Package 👘 🗖
✓ <sup>™</sup> Tutorial → <sup>™</sup> Includes	Pin Configuration		Generate Project Content	<b>E A P P B B &gt;</b>
>	Select Pin Configuration RE01-1500KB-EK.pincfg Generate data: g_bsp_pi	Manage configurations	Configure Pin Driver Warnings	Package view
🛐 R7F0E015D2CFB.pincfg 🥘 Tutorial Debug_Flat.launch	Pin Selection 🔢 🕀 🗔	Pin Configuration	😲 Cycle Pin Group	•
	Type filter text	Name Value	Link	II Renesas II
Project Explorer view	<ul> <li>&gt; P2</li> <li>&gt; P3</li> <li>&gt; P4</li> <li>&gt; P5</li> <li>&gt; P6</li> <li>&gt; P7</li> <li>&gt; P8 ▼</li> <li></li> <li>Yin Function     </li> </ul>	Configuratio	n Editor	RED_150008
	Summary BSP Clocks Pins In	nterrupts Event Links Stacks Components		▶ Legend

Figure 3-7 Project Generation – New Project Creation View



## 3.2 Generating new Executable Project without Smart Configurator

*Note:* It is recommended to use Software Package for projects using RE MCU Family. Smart Configurator enable easy project configuration and avoid the potential problems typically encountered during customized or manual configuration of the device.

Start the e<sup>2</sup> studio application and choose a workspace folder in the Workspace Launcher. To configure a new RE project, follow these steps:

1. Select [File] → [New] → [Renesas C/C++ Project] → [Renesas RE] to open the New Project creation wizard.



Figure 3-8 Open the New Project creation wizard

2. Select "All" "Renesas RE C/C++ Executable Project" template. Click [Next] to continue.

Sew C/C++ Pro	oject —		×
Templates for Re	enesas RE Project		
All C/C++	Renesas RE C/C++ Executable Project A C/C++ Executable Project for Renesas RE.		
	Renesas RE C/C++ Library Project A C/C++ Library Project for Renesas RE.		
	Renesas RE C/C++ SDK Project Create an executable or static library C/C++ SDK project for Renesas RE.		
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cance	el

Figure 3-9 New Project Creation Wizard (1/6)

3. Enter the project name (e.g. Tutorial). Click [Next] to proceed.

	cutable Project for Renesas RE	
<u>P</u> roject name:	Tutorial	
<mark>⊘ U</mark> se <u>d</u> efault	location	
Location:	D:\WorkSpace_for_RE\workspace_2021-07_FSP110\Tutorial	owse
	Create Directory for Project	
Choose file s <u>y</u> st	tem: default 🗸	
Working sets		
Add projec	to working sets Ne	2 <u>w</u>
Working sets:	→ S <u>e</u> l	ect

Figure 3-10 New Project Creation Wizard (2/6)



- 4. Enter device and tool information:
  - Target device: R7F0E015D2CFB (e.g.)
  - Toolchain version: Latest GNU Arm Embedded Toolchain approved for use with Renesas RE (e.g. GCC ARM Embedded 6.3.1.20170620)
  - Debugger: J-Link (ARM)
  - Keep all other fields as default and click [Finish] to generate the project

**Note**: "E2" or "E2 Lite" can be selected in the same way as "J-Link" in the Hardware Debug Configuration pull-down menu.

8		– 🗆 X
	++ Executable Project device _debug settings	
Device Settings Target Device: Project Type: Toolchain Setti Language: Toolchain: Toolchain Versi	R7F0E015D2CFB Unlock Devices Default	Configurations Create Hardware Debug Configuration J-Link ARM Create Release Configuration
?	< <u>B</u> ack	<u>N</u> ext > <u>F</u> inish Cancel

#### Figure 3-11 New Project Creation Wizard (3/6)

5. A new C project named "Tutorial" is created.



Figure 3-12 New C Project Created



## 3.3 Generating and Using a RE Static Library

This section describes how to generate a RE static library project and an executable project that references to the library project.

#### 3.3.1 Creating a Static Library Project

The following steps show an example of how to create a RE static library project.

- 1. Generate a new SDK project as described in chapter 3.1 and use the following option:
  - Name the project (e.g. "RE\_Lib") at step 3,
  - Select build artifact as "Static Library" at step 5,

🗐 Renesas RE C/C++ SDK Project				_		×
Renesas RE C/C++ SDK Project Build Artifact and RTOS Selection						2
Build Artifact Selection		RTOS Selection				
<ul> <li>Executable</li> <li>Project builds to an executable file</li> </ul>		No RTOS				$\sim$
<ul> <li>Static Library         <ul> <li>Project builds to a static library file</li> </ul> </li> </ul>						
?	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish		Cance	ł

Figure 3-13 Select "Static Library" as Build Artifact

• Select project template as "Bare Metal – Blinky" at step 6.

2. After the project is created, click [Generate Project Content] in the Configuration editor.

ummary			Generate Project	t Content
Project Summary	/	Rem	NESAS	^
Board:	EK-RE01 1500KB			
Device:	R7F0E015D2CFB			
Toolchain:	GCC ARM Embedded			
Toolchain Version:	6.3.1.20170620			
FSP Version:	1.1.0			
Project Type:	Flat			
Selected software co				
Renesas Core	that blinks an LED. No RTOS included.	v1.1.0 v1.1.0		
Low Power Mode		v1.1.0		
System		v1.1.0		
Arm CMSIS Versio	n 5 - Core (M)	v5.7.0+sdk.1	10	
Arm Driver Header		v5.7.0+sdk.1		
	kage for R7F0E015D2CFB	v1.1.0		
Board support pac	v1.1.0		•	
Support				

Figure 3-14 Generate library project content

3. From the project explorer window, open "hal\_entry.c" under RE\_Lib\src\



Figure 3-15 Original "hal\_entry.c"

4. Rename the function hal\_entry() to hal\_entry\_lib(), and add a declaration for hal\_entry\_lib().

德	FSP C	on	figuration 🛛	🚺 *hal_entry.c 🗙	
	39				
	40		/* End us	ser code. */	
	41				
	42		/* Functi	ions */	
	43		void led	<pre>run( void );</pre>	
	44				attribute ((sect
	45				attribute ((secti
	46		-	` ^	
	48		⊕ * Function	on Name: hal entr	<b>V</b>
	54			_entry_lib(void);	
	55			,.	
	56		⊖void hal	_entry_lib(void)	
	57		{		
			• • • • • • •		and the second

Figure 3-16 Modified "hal\_entry.c"

5. Build the Library Project. The build outputs a static library file RE\_Lib\Debug\"libRE\_Lib.a".



Figure 3-17 The built static library



#### 3.3.2 Creating an Executable Project using the existing Static Library

This chapter shows how to use the static library created in the previous chapter (3.3.1) in a RE executable project by performing the following steps,

- Create a RE executable project
- Modify the source code to call a function hal\_entry\_lib() declared in the static library project
- Modify the build settings to add the static library
- Build the RE executable project.

Detailed steps are described below.

- 1. Create an executable project as described in chapter 3.2 and name the project (e.g. "RE\_App").
- 2. From project explorer window, open "RE\_App.c" under RE\_App\src\.
- 3. Add a declaration for library function "hal\_entry\_lib()", and call this function in main() function.



Figure 3-18 Update "RE\_App.c"

4. Right-click on "RE\_App" project and select [C/C++ Project Settings].

5. In the Project Properties dialog, select [C/C++ Build] → [Settings], then select [Tool Settings] tab. Select [GNU ARM Cross C Linker] → [Libraries] and click [Add...] at "User defined archive (library) files (-I)" to add "RE\_Lib" as the library file.



Figure 3-19 Add reference to library file
6. Click [Add...] at "User defined archive search directories (-L)" to add "\${workspace\_loc:/RE\_Lib/Debug}"



Figure 3-20 Add reference to library location



7. Verify that the new path has been updated. Go to [Cross ARM C Linker]  $\rightarrow$  [Libraries] to check.

Tool Settings 🛞 Toolchain 🛞 Devic	e 🎤 Build Steps	🚇 Build Artifact	Binary Parsers	8	Error P	arsers	
🖄 Target Processor	Standard arch	ive (library) files (-l)		Ð	a 8	취	θı
🖄 Optimization							-
🖄 Warnings	m c						
🖄 Debugging	gcc						
🗸 🛞 Cross ARM GNU Assembler	13						
Preprocessor							
Includes							
👰 Warnings							
Miscellaneous	Standard arch	ive search directories	; (-L)		<b>N</b> 2	) 쥰l	신
Cross ARM C Compiler							
Preprocessor							
실 Includes							
Optimization							
🖉 Warnings							
Miscellaneous							
🗸 🛞 Cross ARM C Linker	Search archi	ves repeatedly for un	defined references	;			
🖄 General						- 以 I	0.2
🖄 Libraries	User defined a	archive (library) files (	-0	•			핟
🖄 Miscellaneous	RE_Lib						
🖄 Other							
🗸 🛞 Cross ARM GNU Create Flash Image							
🖄 General							
Cross ARM GNU Print Size							
🖄 General							
With the second seco	User defined a	archive search directo	ories (-L)	Ð	<b>8</b> 😪	신	Ϋ́
🖄 Settings	"\${workspace	loc:/RE_Lib/Debug}"					
	s[worksbucc_	ioe, ne_eib, bebudj					
	11						

Figure 3-21 Cross ARM C Linker settings are already updated

8. In the Properties dialog, select [Project References] in the left pane, then tick on "RE\_Lib" to mark the executable project as depending on the static library project. Click [Apply and Close].

Properties for RE_App	Project References $\diamond \bullet \bullet \bullet \circ$
<ul> <li>Resource Builders</li> <li>C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor</li> <li>C/C++ General</li> <li>MCU Project Natures</li> <li>Project References</li> <li>Renesas QE Run/Debug Settings Task Tags</li> <li>Validation</li> </ul>	Project references of 'RE_App':
?	Apply and Close Cancel

Figure 3-22 Mark executable project as depending on the static library project

- 9. Build the application project.
- 10. Set a breakpoint where the library function hal\_entry\_lib() is called. Run RE\_App project.
- 11. When the program stops at the breakpoint, resume it. Confirm that the library function which blinks the LEDs (e.g. hal\_entry\_lib()) is executed.





Figure 3-23 Application project invokes library function



# 3.4 Import Existing Projects into Workspace

To import an existing e<sup>2</sup> studio project to the current workspace, follow the steps below. These steps import a sample project from Renesas website and will be used for demonstrating debugging features in section 5.4.

1. Download the LED Blinker sample code for RE01 from Renesas website: <u>https://www.renesas.com/sg/en/document/scd/led-blinker-sample-code-re01-1500kb-and-256kb-group-application-note-sample-code</u>.

# LED Blinker Sample Code for REO1 1500KB and 256KB Group Application Note - Sample Code

By clicking on the "I accept" button or other button or mechanism designed to acknowledge agreement to the terms of an electronic copy of the Disclaimer002 (the "Agreement"), or by downloading, installing, accessing, or otherwise copying or using all or any portion of the licensed software described in the Agreement (the "Licensed Software"), (a) you accept the Agreement on behalf of the licensee for whom you are authorized to act (the "Licensee"), and acknowledge that the Licensee is legally bound by the Agreement, and (b) you represent and warrant that you have the right, power, and authority to act on behalf of and bind the Licensee. IF THE LICENSEE DOES NOT AGREE TO THE TERMS CONTAINED IN THIS AGREEMENT, OR IF YOU DO NOT HAVE THE RIGHT, POWER, AND AUTHORITY TO ACT ON BEHALF OF AND BIND THE LICENSEE, DO NOT SELECT THE "I ACCEPT" BUTTON OR OTHER BUTTON OR MECHANISM DESIGNED TO ACKNOWLEDGE ACCEPTANCE OF THE AGREEMENT, AND DO NOT DOWNLOAD, INSTALL, ACCESS, OR OTHERWISE COPY OR USE ALL OR ANY PORTION OF THE LICENSED SOFTWARE. RENESAS PERMITS THE LICENSEE TO DOWNLOAD, INSTALL, ACCESS, OR OTHERWISE COPY OR USE THE LICENSED SOFTWARE (INCLUDING THE FUNCTIONALITY OR FEATURES THEREOF) ONLY IN ACCORDANCE WITH THE AGREEMENT.

ACCEPT AND DOWNLOAD >

# Figure 3-24 Download the Sample Code

#### 2. After downloading the package, extract the zip file.

Name	Date modified	Туре	Size
📳 an4950_gpio_re_256kb.zip	7/12/2021 1:43 PM	Compressed (zipp	71,450 KB
🕌 an4950_gpio_re_1500kb.zip	7/12/2021 1:43 PM	Compressed (zipp	73,644 KB
👃 r01an4950ej0101-re.pdf	7/12/2021 1:43 PM	Adobe Acrobat D	191 KB
👃 r01an4950jj0101-re.pdf	7/12/2021 1:43 PM	Adobe Acrobat D	316 KB

Figure 3-25 The Sample code package



3. In  $e^2$  studio, select [File]  $\rightarrow$  [Import]

<u>F</u> ile	<u>E</u> dit	<u>N</u> avi <u>o</u>	jate	Se <u>a</u> rch	י <u>P</u>	roject	Ren	nesas <u>V</u> i	iews	<u>R</u> u
	New						ļ	Alt+Shi	ift+N⊃	>
	Open	n File								
È,	Open	n Project	ts fro	om File S	Syste	:m				
	Recer	nt Files							2	>
	Close	e Editor						Ct	rl+W	
	Close	e All Edi	tors				C	trl+Shi	ft+W	
	Save							C	trl+S	
	Save	As								
R	Save	All					0	Ctrl+Sh	ift+S	
	Rever	rt								
	Move	e								
	Renar	me							F2	
39	Refre	sh							F5	
	Conv	ert Line	2 Delir	imiters T	ō				2	>
₽	Print.							C	trl+P	
è	Impo	irt								
4	Expo	rt								
	Prope	erties						Alt+I	Enter	
	Swite	h Work	space	e					3	>
	Resta	irt								
	Exit									

Figure 3-26 Import the sample project

4. In the [Import] dialog, select [General] → [Existing Projects into Workspace]. Click [Next]

📴 Import —		
Select Create new projects from an archive file or directory.	Ľ	
Select an import wizard:		
<ul> <li>✓ Seneral</li> <li>Archive File</li> <li>✓ CMSIS Pack</li> <li>✓ Existing Projects into Workspace</li> <li>✓ File System</li> <li>✓ Preferences</li> <li>✓ Projects from Folder or Archive</li> <li>✓ Renewa &amp; Import Existing C/C++ Project into Workspace</li> <li>✓ Renesas CS+ Project for CA78K0R/CA78K0</li> <li>✓ Renesas CS+ Project for CC-RX and CC-RL</li> <li>✓ Sample Projects on Renesas Website</li> <li>✓ C/C++</li> </ul>	~	
? < <u>B</u> ack <u>Next &gt;</u> <u>Finish</u>	Cancel	

Figure 3-27 Select import wizard

 In the [Import Projects] dialog, select "Select archive file". Click [Browse] then select the zip file named "an4950\_gpio\_re\_1500kb.zip" in the sample code package. Select the project listed in "Projects:" and click [Finish].

📴 Import		_		×
Import Projects Select a directory to searc	h for existing Eclipse projects.			
○ Select roo <u>t</u> directory:		~	B <u>r</u> owse	2
Select <u>archive file</u> :	D:\WorkSpace_for_RE\r01an4950;	xx0101-re\rl ~	B <u>r</u> owse	2
<u>P</u> rojects:				
an4950_gpio_re_1	500kb (an4950_gpio_re_1500kb/)		Select /	All
			<u>D</u> eselect	t All
			R <u>e</u> fres	h
	orkspace d projects upon completion eady exist in the workspace		Ne <u>w</u> S <u>e</u> lect	

# Figure 3-28 Select project to import

6. Open the project properties, select [C/C++ Build] → [Settings] in the left pane. Select tab [Toolchain] and select the correct toolchain for the project. Click [Apply and Close].

Properties for an4950_gpio_	re_1500kb	— 🗆 X
	Settings	<> ▼ ⇒ ▼ 8
<ul> <li>Resource</li> <li>Builders</li> <li>C/C++ Build</li> <li>Build Variables</li> </ul>	Configuration: HardwareDebug [Active] ~	A Manage Configurations
Environment Logging Settings Tool Chain Editor > C/C++ General > MCU Project Natures Project References Renesas QE Run/Debug Settings Task Tags > Validation	<ul> <li>Tool Settings Toolchain So Device P Build Steps P Build Artifact</li> <li>Enable toolchain integration</li> <li>Current Toolchain</li> <li>Toolchain: GNU ARM Embedded</li> <li>Version: 6.3.1.20170620</li> <li>Change Toolchain (click Apply before switching tabs)</li> <li>Toolchain: GNU ARM Embedded</li> <li>Version: 6.3.1.20170620</li> <li>Additional Tools</li> <li>Create Library generator</li> <li>Create Flash image</li> </ul>	t 🖬 Binary Parsers ( )
?	Аррђ	y and Close Cancel

Figure 3-29 Update project toolchain (without Smart Configurator)



If the imported project uses Smart Configurator, the setting GUI may be different.

Properties for Tutorial		— 🗆 X
	Settings	⟨¬ ▼ ¬ ↓
> Resource		
Builders C/C++ Build Build Variables Environment	Configuration: Debug [Active]	✓ Manage Configurations
Logging	🛞 Tool Settings 🛞 Toolchain	🎤 Build Steps  🚇 Build Artifact 🛛 🗟 Binary Parsers 🛛 😣 Error Parsers
Settings Tool Chain Editor	Name	GNU Tools for ARM Embedded Processors (arm-none-eabi-gcc) 🗸 🗸
> C/C++ General > MCU	Architecture	ARM (AArch32) ~
Project Natures	Prefix	arm-none-eabi-
Project References Renesas OE	Suffix	
Run/Debug Settings	C compiler	gcc
Task Tags Validation	C++ compiler	g++
validation	Archiver	ar
	Hex/Bin converter	objcopy
	Listing generator	objdump
	Size command	size
	Build command	make
	Remove command	rm
	✓ Create flash image	
	Create extended listing	
	Print size	
	Use integrated toolchain versior	n: 6.3.1.20170620 V
	Per project path	Browse
<b>?</b> )		Apply and Close Cancel

Figure 3-30 Update project toolchain (with Smart Configurator)

7. Build the project and make sure that it is successful.

# 3.5 Configuration Editor

The Configuration editor view displays the current project configuration settings. The settings are saved in the file 'configuration.xml'. The project configuration settings are grouped into multiple pages that allow you to set several configurable aspects of the project, such as how pins and clocks are set up, and which drivers are included.

To edit the project configuration, make sure that:

- RE Configuration perspective is open by clicking [Window] → [Perspective] → [Open Perspective] → [Other…] → [FSP Configuration] and,
- The 'configuration.xml' file is opened.

🏊 Project Explorer 🗙 📄 🛱 🏹 🖇 🖳 🗖	🔅 [Tutorial] FSP Configurat	tion 🔀		
<ul> <li>an4950_gpio_re_1500kb</li> <li>RE_App</li> <li>RE_Lib</li> </ul>	Summary	Click here to generate project content	Generate Project	Content
✓ (ﷺ Tutorial > ) ∭ Includes	Project Summary		RENESAS	^
> 😕 re > 😕 re_gen		EK-RE01 1500KB		
> 2 src		R7F0E015D2CFB		
> 🗁 re_cfg	Toolchain: Toolchain Version:	GCC ARM Embedded		
> 🕞 script		1.1.0		
configuration.xml R7F0E015D2CFB.pincfg		Flat		
Tutorial Debug_Flat.launch	Selected software cor	nponents		
	Simple application the	nat blinks an LED. No RTOS included.	v1.1.0	
Double-click .xml file to	Renesas Core		v1.1.0	
open Configuration	Low Power Mode		v1.1.0	
open comiguration	System		v1.1.0	
	Arm CMSIS Version	5 - Core (M)	v5.7.0+sdk.1.1.0	
	Arm Driver Header		v5.7.0+sdk.1.1.0	
		age for R7F0E015D2CFB age for RE01 1500KB	v1.1.0	$\sim$
	Doard Subbort backa		v1.1.0	
	D D Support	Page selection		
	Summary   BSP   Clocks   Pin	s Interrupts Event Links Stacks Compon	ents	

Figure 3-31 Configuration Editor view

There are 8 pages (or tabs) in the Configuration editor.

The Summary page contains project-specific summary information.

The BSP page allows users to select the FSP version, the type of RE board, and the device.

The configuration steps and options for the Clocks, Pins, Interrupts, Event Links, Stacks and Components pages are discussed in the following chapters.

# 3.5.1 Summary Page

The summary page contains a project-specific summary which includes details of the currently selected device, board and RE software components, etc. There are also useful links to the 'Renesas Presents' YouTube channel and the Software Package's user manual.

If user adds new threads and modules/objects to a thread, this information will be also shown on the Summary page.

[Tutorial] FSP Configu	ration 🛛		
ummary		Generate P	oject Content
Project Summary	/	RENESAS	^
Board: Device: Toolchain: Toolchain Version: FSP Version: Project Type: Selected software c	1.1.0 Flat	Information about board, device, toolchain and FSP Software compon included in the pro	
Simple application	that blinks an LED. No R	TOS included. v1.1.0	
Renesas Core		v1.1.0	
Low Power Mode		v1.1.0	
System		v1.1.0	
Arm CMSIS Versio	on 5 - Core (M)	v5.7.0+sdk.1.1.0	
Arm Driver Heade	r	v5.7.0+sdk.1.1.0	
Board support pac	kage for R7F0E015D2CF	B v1.1.0	~
Board support pac	kage for RE01 1500KB	v1.1.0	
Support	Useful lini	ks	
ummary BSP Clocks P	ins Interrupts Event Links	Stacks Components	

Figure 3-32 Summary page

# 3.5.2 BSP Page

The BSP Page allows users to select the FSP version, board and device. Users can also import the CMSIS pack from this page.

🔅 [Tutorial] FSP Configuration	×	
Board Support Package	e Configuration	Generate Project Content
		Restore
Device Selection Se	elect FSP version	Import CMSIS
FSP version: 1.1.0 Board: EK-RE01 1500KI Device: R7F0E015D2CF RTOS: No RTOS Select board		Board Details Evaluation Kit RE01 1500KB Visit <u>https://www.renesas.com/ek-re01-1500kb</u> to kit user's manual, quick start guide, errata, design package, example projects, etc. Select device
<		>

Figure 3-33 BSP page



## 3.5.3 Clocks Page

The Clocks page sets up the initial clocking for the application. Clock sources, PLL settings, and clock divider settings can be selected for each of the output clocks.

For details on the Clock Generation Circuit (CGC), see the RE hardware user's manual. To update the project, follow these steps:

#### 1. Select a value in the drop-down list for the clock setting on GUI.

🔅 *[Tutorial] FSP Configuration 🗙		
Clocks Configuration	Generate Project (	Content
	Restore	Defaults
Normal mode (High-s $\vee$	Disabled $\checkmark$	
↓ Drive Capability: 7 ↓ PL ↓ Oscillator ↓	Div /2 ~ Uiv 16MHz Mul x2.0 ~ UCLK 32MHz UCLK 32MHz	
HOCO Disabled $\rightarrow$ HOCO 32MHz         MOCO Enabled $\rightarrow$ MOCO 2MHz         LOCO Enabled $\rightarrow$ LOCO 32768Hz         SOSC Disabled $\rightarrow$ Standard CL $\downarrow$ Noise filter Enabled         SOSC 32768Hz $\downarrow$	Clock Src: MOCO Clock Src: HOCO Clock Src: MOCO Clock Src: MOSC Clock Src: SOSC Clock Src: PLL	
Summary BSP Clocks Pins Interrupts Event Links Stacks Cou	ponents	

## Figure 3-34 Clock configuration page

- 2. Save the Project Configuration Settings, for example by hitting Ctrl-S.
- 3. Click the Generate Project Content button Generate Project Content
- 4. The file bsp\_clock\_cfg.h is updated with the selected clock configuration.

# Renesas e<sup>2</sup> studio 2021-07 or Higher

Project Explorer 🐹 🛛 🖨 🏷 🖇 🖵 🗖	💮 [Tutorial] FSP Co	onfiguration 🚺 bsp_clock_cfg.h 🗙 🗁 🗖	
an4950_gpio_re_1500kb	1	/* generated configuration header file - do not edit */ 🔨 🔺	
RE_App	2	⊖#ifndef BSP_CLOCK_CFG_H_	
The Lib	3	#define BSP_CLOCK_CFG_H_	
V 😤 Tutorial [Debug]	4	<pre>#define BSP_CFG_CLOCKS_SECURE (0)</pre>	
	5	<pre>#define BSP_CFG_CLOCKS_OVERRIDE (0)</pre>	
> 🔊 Includes	6	<pre>#define BSP_CFG_MOSC_ENABLE (0) /* MOSC Disabled */</pre>	
> 📇 re	7	#define BSP_CFG_MOSC_HZ (32000000) /* MOSC 3200000Hz */	
✓ <sup>™</sup> re_gen	8	<pre>#define BSP_CFG_MOSC_DRIVE (7) /* Drive Capability: 7 */</pre>	
> h bsp_clock_cfg.h	9	<pre>#define BSP_CFG_MOSC_CLOCK_SOURCE (0) /* Oscillator */</pre>	
> c common_data.c	10	#define BSP_CFG_MOSC_LOW_POWER_ENABLE (0) /* Low Consumption Oscillat	
> h common_data.h	11 12	<pre>#define BSP_CFG_MOSC_WAIT_TIME (5) /* Stabilization Wait Time: 61 cycl #define BSP_CFG_MOSC_ENABLE (0) /* MOSC_Disabled */</pre>	
	13	<pre>#define BSP_CFG_HOCO_ENABLE (0) /* HOCO Disabled */ #define BSP_CFG_HOCO_FREQUENCY (1) /* HOCO_32MHz */</pre>	
> c hal_data.c	14	#define BSP CFG MOCO ENABLE (1) /* MOCO Enabled */	
> h hal_data.h	15	#define BSP CFG LOCO ENABLE (1) /* LOCO Enabled */	
> 🖻 main.c	16	#define BSP CFG SOSC ENABLE (0) /* SOSC Disabled */	
> 🖻 pin.c	17	#define BSP CFG SOSC DRIVE (0) /* Standard CL */	
> h pin.h	18	#define BSP CFG SOSC NF (0) /* Noise filter Enabled */	
> c vector_data.c	19	#define BSP_CFG_PLL_ENABLE (0) /* PLL Disabled */	
> h vector_data.h	20	#define BSP_CFG_PLL_DIV (1) /* PLL Div /2 */	
> 🛱 src	21	<pre>#define BSP_CFG_PLL_MUL (1) /* PLL Mul x2.0 */</pre>	
	22	<pre>#define BSP_CFG_CLOCK_SOURCE (1) /* Clock Src: MOCO */</pre>	
✓ ▷ re_cfg	23	<pre>#define BSP_CFG_ICLK_PCLKA_DIV (0) /* ICLK &amp; PCLKA Div /1 */</pre>	
> 🗁 fsp_cfg	24	<pre>#define BSP_CFG_PCLKB_DIV (0) /* PCLKB Div /1 */</pre>	
🖻 r_core_cfg.h	25	<pre>#define BSP_CFG_POWER_CONTROL_MODE (1) /* Normal mode (High-speed mode)</pre>	
💼 r_lpm_cfg.h	26	<pre>#endif /* BSP_CLOCK_CFG_H_ */</pre>	
r_system_cfg.h	27		
> 🗁 script			
ig configuration.xml			
R7F0E015D2CFB.pincfg			
re_cfg.txt			
Tutorial Debug_Flat.launch			
		×	
		< >>	

Figure 3-35 bsp\_clock\_cfg.h is updated



# 3.5.4 Pins Page

The Pins page provides a graphical user interface for generating the pin configuration settings for the project.

			Generate Project Conter
Select Pin Configuration		Export to CSV file 🛛 🖺 Confi	gure Pin Driver Warnings
RE01-1500KB-EK.pincfg		nfigurations (1)	
Pin Selection 🗄 🕀 🖡	Pin Configuration		🤨 Cycle Pin Group
Type filter text	Name (3)	Value	Link

Figure 3-36 Pin configuration GUI

he Pin Configuration window consists of 3 parts:

- 1. Select Pin Configuration: Selects pin-configuration file and specifies the name for the associated data structure. Multiple pin configurations can be set as follows:
  - Create a new .pincfg file (e.g. NewName.pincfg) in Project Explorer by copying an existing one.
  - Select the new .pincfg file (e.g. NewName.pincfg) in the "Select Pin Configuration" dialog box.
  - Check the "Generate data" checkbox and give the new pin configuration a unique data structure name in the text field.
  - The multiple pin configurations will be created in different data structures.
- 2. Pin Selection: Selects pin or peripheral that will be set up.
- 3. Pin Configuration: Set up for function/property of the selected pin/peripheral.

Follow below steps to configure pins of the peripherals to be used in the project:

- 1. Select a peripheral in the "Pin Selection" pane, e.g. [Connectivity: SCI] → [SCI4]. The configuration for this peripheral will be shown in the "Pin Configuration" pane.
- 2. Select an Operation Mode for the peripheral, e.g. "Simple SPI".
- 3. Select the pins you would like to use for the Input/Output functions of the selected peripheral in the selected mode.

🔅 *[Tutorial] FSP Configuration 🗙 🔥	bsp_clock_cfg.h			
Pin Configuration				Generate Project Content
Select Pin Configuration		📑 Export to CSV f	ile 🖺 C	onfigure Pin Driver Warnings
RE01-1500KB-EK.pincfg	Manage configurations			
Pin Selection $\exists \exists \exists \exists \exists \exists z \end{bmatrix}$	Pin Configuration			😲 Cycle Pin Group
Type filter text  Connectivity:LPG Connectivity:MIP Connectivity:QSPI Connectivity:SCI SCI0 SCI1 SCI2 SCI3 SCI3 SCI4 (1) SCI5 SCI5	Name Pin Group Selection Operation Mode V Input/Output SS4 MISO4 SCK4 - MOSI4 (3)	Value Mixed Simple SPI (2) None ✓ P112 ✓ P108 None None ✓ P113		Link
SCI9 Connectivity:SPI Connectivity:USB FS Debug:JTAG/SWD Interrupt:IRQ Interrupt:KR *		I2C mode, ensure port pins tween I2C and other modes,		
Pin Function Pin Number Summary BSP Clocks Pins Interrupts	Event Links Stacks Components			

Figure 3-37 Pin Configuration Setting (by Peripheral)

A single pin can also be set up following the steps below:

- 1. Select a pin in the "Pin Selection" pane, e.g. [Ports]  $\rightarrow$  [P0]  $\rightarrow$  [P003]. The configuration for this pin will be shown in the "Pin Configuration" pane.
- 2. Enter properties for this pin as an example in the figure below.

			Gen	erate Project Conte
Select Pin Configuration		🔛 Export to CSV	file 🔚 Configure	Pin Driver Warning
RE01-1500KB-EK.pincfg	✓ Manage configurations			
Generate data: g_bsp_pin_cfg				
or Generate data: [g_bsp_pin_cig]				
Pin Selection 📋 🕀 🖃 🖡	Pin Configuration			😲 Cycle Pin Group
Type filter text	Name	Value	Link	
V V Ports	Symbolic Name			
V V POILS	Comment	GPIO1		
P000	Mode	Input mode		
P000	Pull up/down	None (2	2)	
P001	Output Type	CMOS		
✓ P003 (1)	Drive Capacity	Н		
P004	✓ Input/Output			
P005	P003	V GPIO	$\Rightarrow$	
P005				
✓ P007				
✓ P008	<			3
✓ P009				
P010	Module name: P003			
P011	Port Capabilities: ADC0: AN003			
P012				
P012				
< >>				

Figure 3-38 Pin Configuration Setting (by single pin)

# 3. The MCU Package view shows this pin change.



Figure 3-39 MCU Package View (Connection Status)

#### 3.5.5 Stacks Page

The Stack page allows users to:

- Configure threads within a RE project.
- Add RE modules and objects to a thread.
- Modify module and object properties in the Properties View.

🔅 *[Tutorial] FSP Configuration 🗙 🕞 bsp_clock	k_cfg.h		- 8
Stacks Configuration			Generate Project Content
Threads New Thread Remove HAL/Common Renesas Core Low Power Modes Driver on r_Ipm System Driver on r_system (1) Objects New Object > Remove	HAL/Common Stacks  Renesas Core  Add Energy Harvesting(EHC) Driver	New Stack Low Power Modes Driver on r_lpm (1)	<ul> <li>&gt; Extend Stack &gt; Remove</li> <li>              System Driver on r_system      </li> </ul>
(3) Summary BSP Clocks Pins Interrupts Event Link	ks Stacks Components		
Properties ※ 訳問題      Smart Browser			rt 8 - E
Renesas Core			
Settings Property Common Energy Harvesting (EHC) Option Function Select 0 (OFS0)	Value (4)		^
> Option Function Select 1 (OFS1) > Security MPU			~

## Figure 3-40 Stacks Configuration GUI

The Stacks page consists of 3 panes:

- 1. Threads pane: Add/remove threads.
- 2. Stacks pane: Add/remove Software module instances, i.e. IO port, SCI, UART, etc.
- 3. Objects pane: Add/remove kernel objects.

In addition, the Properties view supports the Threads Configuration and is used to modify module/object properties.

A module can be added to the existing project following the steps below:

- 1. Select a thread, i.e. HAL/Common. The modules and objects in this thread are shown.
- In the Stacks pane, click <sup>4</sup> "New Stack" to add a module to the thread, i.e. "New Stack" → [Driver] → [Connectivity] → [CMSIS Driver for USART on r\_usart ch(SCI)].
- 3. Click the Generate Project Content Generate Project Content button to generate the source code content.
- 4. Users can change the properties of the selected module in the Properties view according to their requirements.

🔅 *[Tutorial] FSP Configuration 🗙 🔓 bsp_clock_cfg.h				
Stacks Configuration				(3) Generate Project Content
Threads 🐑 New Thread 🔊 Remove 📄	HAL/Common Stacks			(2) 🔄 New Stack > 🚔 Extend Stack > 👔 Remove
HAL/Common (1)	🕀 Renesas Core	Low Power Modes Driver on r_lpm	System Driver on r_system	CMSIS Driver for USART on r_usart ch0(SCI0)
<ul> <li>System Driver on r_system</li> <li>CMSIS Driver for USART on r_usart ch0(SCI0)</li> </ul>	(i) •	<b>(i)</b>	<b>(i)</b>	0
	Add Energy Harvesting(EHC) Driver			Add DMAC or DTC Driver for Transmission
Objects 🕢 New Object > 🙀 Remove				
Summary BSP Clocks Pins Interrupts Event Links Stac	ks Components			
🔲 Properties 🛛 🔝 問題 🛞 Smart Browser				📑 🕴 🗖
CMSIS Driver for USART on r_usart ch0(SCI0	)			
Settings Property V Common	Value			^
> Common (4) > SCI0				
> SCI1 > SCI2				~

Figure 3-41 Add New Module to Thread



An added module may require dependent modules or configuration settings. Necessary dependent modules will be added automatically. Optional dependent modules are suggested to be added manually by the user. In this case, users should click on the suggested modules to add and configure their properties.



**Figure 3-42 Dependent Modules** 



A module or a module stack can also be added by performing a "copy and paste" operation on the Threads page. Right-click on a module and select "Copy" to copy it. Right-click in the stack pane of the same or a different thread in the same project and select "Paste".

A "cut and paste" operation is also available.

Threads	Diew Thread Remove (=)	HAL/Common Stacks			New Stack >	Estand Stack > R Rem	ove
	AL/Common			1			
	Renesas Core Low Power Modes Driver on r. Ipm	Renesas Core	Low Power Modes Driver on r_Ipm	System Driver on r_system	CMSIS Driver for US	SART on r_usart ch0(SCI0)	
	System Driver on r_system	0	(I)	(I)	1) Team	>	
	CMSIS Driver for USART on r_usart ch0(SCI0)	1			Resource C	Configurations >	E
		Add Energy Harvesting(EHC)			Nalidate		
٢	>	Driver			Tr of Cut	Ctrl+X	
Objects	New Object > R Remove	-			Paste	CM+C CM+V	H
					X Delete	Delete	
					Non-secur	re Callable	
					in Import		
		¢			Export     Module Re	BOURG	>
	BSP Clocks Pins Interrupts Event Links Stac	ks Components			O Run As	>	
Propert					Debug As		
CMSIS D	river for USART on r_usart ch0(SCI0)				Compare V		
Settings	Property	Value			Replace W	ith >	
	✓ Common						
	> Common						
	> Common > SCI0						
	> SCI0 > SCI1						
	> SCI0						
	> SCI0 > SCI1 > SCI2 > SCI2 > SCI3 W[FSP Configuration X @ bsp_clock_clp.h						
Stacks (	> \$CI0 > \$CI1 > \$CI2 > \$CI2	HAL/Common Stacks			New Stack >	Generate Project Co Se Estend Stack + 20 Rem	ntent
Stacks ( Threads	> SCI0 > SCI1 > SCI2 > SCI3 with TSP. Configuration X @ bsp_clock_clip.h Configuration () Time Thread () Remove () AL/Common	HAL/Common Stacks	(de Lou Dever Moder	) (de Sature Driveren		Generate Project Co	ntent
Stacks ( Threads	SCI0     SCI0     SCI1     SCI2     SCI2     SCI2     SCI2     SCI3		Low Power Modes     Driver on r_Ipm	System Driver on r_system		O Generate Project Co	ntent
Stacks ( Threads	SCI0 SCI1 SCI2 SCI2 SCI2 SCI2 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3	HAL/Common Stacks				Generate Project Co	ntent
Stacks ( Threads	SCI0     SCI0     SCI1     SCI2     SCI2     SCI2     SCI3  all FSP Configuration X     Bop_clock_cfip.h  Configuration      Trine Thread    Sci Remove      AL/Common     Remeas Core     Low Power Modes Driver on r_lipm	HAL/Common Stacks	Driver on r_lpm	r_system	CMSIS Driver for US	Cemerate Project Co Estend Stack + Rem SART on r_usert ch0(SO0)	ntent
Stacks ( Threads	SCI0 SCI1 SCI2 SCI2 SCI2 SCI2 SCI3 Configuration X @ bsp_clock_cfp.h Configuration	HAL/Common Stacks	Driver on r_lpm	r_system	CMSIS Driver for US	Generate Project Co Estend Stack + R Rem SART on r_usart ch0(SOO)	ntent
Stacks ( Threads	SCI0 SCI1 SCI2 SCI2 SCI2 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3	HAL/Common Stacks	Driver on r_lpm	r_system	CMSIS Driver for U	Cemerate Project Co Estend Stack + Rem SART on r_usert ch0(SO0)	ntent
Stacks ( Threads	SCI0 SCI1 SCI2 SCI2 SCI2 SCI2 SCI3 Configuration X @ bsp_clock_cfp.h Configuration	HAL/Common Stacks  Renesas Core  Renesas Core Renesas	Driver on r_lpm	r_system Team Resource Configu	CMSIS Driver for U	Generate Project Co Estend Stack + R Rem SART on r_usart ch0(SOO)	ntent
Stacks (	SCI0 SCI1 SCI2 SCI2 SCI2 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3	HAL/Common Stacks  Renesas Core  Renesas Core Renesas	Driver on r_lpm	Team Resource Configur	CMSIS Driver for U	Generate Project Co Estend Stack + R Rem SART on r_usart ch0(SOO)	ntent
Stacks (	SCI0 SCI1 SCI2 SCI2 SCI2 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3	HAL/Common Stacks  Renesas Core  Renesas Core Renesas	Driver on r_lpm	r_system ③ Team Resource Configur ♥ Validate ♥ Cut	CMSIS Driver for U	Generate Project Co Estend Stack + R Rem SART on r_usart ch0(SOO)	ntent
Stacks (	SCI0 SCI1 SCI2 SCI2 SCI2 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3 SCI3	HAL/Common Stacks	Driver on r_lpm	Cysystem Team Resource Configur Validate Cut Copy Paste	ChtSis Driver for U	Generate Project Co Estend Stack + R Rem SART on r_usart ch0(SOO)	n
Stacks ( Threads ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) )	SOD	HAL/Common Stacks	Driver on r_lpm	r_system ③ Team Resource Configue ⊘ Validate ⊘ Cot © Copy ⑦ Paste ≫ Delete	CMSIS Driver for U  CMSIS Driver for U  CMSIS Driver for U  CMSIS  Add DMAC or DTC  Privat for  rations  Cell+X  Cell+X  Cell+C  Delete  CMSIS  Cell+C  CMSIS  CMSIS CMSIS CMSIS  CMSIS CMS CMSIS CMSIS CMS CMS CMS CMS CMS CMS CMS CMS CMS CM	Generate Project Co Estend Stack + R Rem SART on r_usart ch0(SOO)	ntent
Stacks ( Threads	SCI0 SCI1 SCI2 SCI2 SCI2 SCI2 SCI2 SCI3 Configuration Con	HAL/Common Stacks	Driver on r_lpm	Cisystem Team Resource Configur Validate Cut Copy Copy Copy Delete Non-secure Callab	CMSIS Driver for U  CMSIS Driver for U  CMSIS Driver for U  CMSIS  Add DMAC or DTC  Privat for  rations  Cell+X  Cell+X  Cell+C  Delete  CMSIS  Cell+C  CMSIS  CMSIS CMSIS CMSIS  CMSIS CMS CMSIS CMSIS CMS CMS CMS CMS CMS CMS CMS CMS CMS CM	Generate Project Co Estend Stack + R Rem SART on r_usart ch0(SOO)	n
Stacks ( Threads	SCI0 SCI1 SCI2 SCI2 SCI2 SCI2 SCI2 SCI3 Configuration Con	HAL/Common Stacks	Driver on r_lpm	r_system ③ Team Resource Configur ⊘ Validate ⊘ Cut ⊙ Copy ① Paste Non-secure Callab ≥ Import	CMSIS Driver for U  CMSIS Driver for U  CMSIS Driver for U  CMSIS  Add DMAC or DTC  Privat for  rations  Cell+X  Cell+X  Cell+C  Delete  CMSIS  Cell+C  CMSIS  CMSIS CMSIS CMSIS  CMSIS CMS CMSIS CMSIS CMS CMS CMS CMS CMS CMS CMS CMS CMS CM	Generate Project Co Estend Stack + R Rem SART on r_usart ch0(SOO)	n
Stacks ( Threads	SOD	HAL/Common Stacks	Driver on r_lpm	Cisystem Team Resource Configur Validate Cut Copy Copy Copy Delete Non-secure Callab	Ctd+X Ctd+V Delete Del	Generate Project Co Estend Stack + R Rem SART on r_usart ch0(SOO)	n
Stacks ( Threads	SCI0 SCI1 SCI2 SCI2 SCI2 SCI2 SCI2 SCI3 Configuration Con	HAL/Common Stacks	Driver on r_lpm	Cisystem Team Resource Configur Validate Cut Copy Paste Non-secure Callab Minport Export	Ctd+X Ctd+V Delete Del	Generate Project Co Estend Stack + R Rem SART on r_usart ch0(SOO)	n
Stacks ( Threads	SOD	HAL/Common Stacks	Driver on r_lpm	r_system ③ Team Resource Configue ⊘ Validate ⊘ Cot © Copy ⑦ Paste Non-secure Callab Module Resource	CMSIS Driver for U  CMSIS Driver for U  CMSIS Driver for U  CMSIS Add DMAC or DTC  Driver for  CMSIS CEF+X CMSIS CEF+X CMSIS Delete Delete	Generate Project Co Estend Stack + R Rem SART on r_usart ch0(SOO)	n

Figure 3-43 Copy & Paste operation

There will be a name conflict between the old module instance and the new one. Renaming one of the module instances will solve the problem. If there is hardware conflict, one of the modules should change to a different hardware channel.



Figure 3-44 Module Instance name conflict



## Renesas e<sup>2</sup> studio 2021-07 or Higher

A module or a module stack can also be added by performing the export and import operation on the Stacks page. Right-click on a module and select "Export..." to export the configuration of the module to an XML file. Right-click in the stack pane of the same or a different thread in the same project and select "Import..." to import the configuration from the exported XML file.

V 🏠 HAL/Cor	🐑 New Thread 🔬 Remove 📋	HAL/Common Stacks		Ne	w Stack > 🚔 Extend Stack	Remove
		CMSIS Driver for USART on r_usart ch0(S	iCI0)	+ CMSIS D	river for USART on r_usart ch Team	
	S Driver for USART on r_usart ch0(SCl0) S Driver for USART on r_usart ch1(SCl1)	Add DMAC or DTC	C or DTC Reception		Resource Configurations Validate Cut	Ctrl+X
Dbjects	🐑 New Object >  Remove				<b>Copy</b> Paste <b>Delete</b> Non-secure Callable	Ctrl+C Ctrl+V Delete
		د		-	import Export	
mmary BSP C	locks Pins Interrupts Event Links Stack	s Components		0	Module Resources	
Properties 🔀	1			0	Run As	>
					Debug As	>
ASIS Driver	for USART on r_usart ch1(SCI1)				Compare With	>
ettings Prop	API Allocation		Value		Replace With	>
~ N	Iodule CMSIS Driver for USART on r_usart	ch1(SCI1)				
	Instance Name Channel		g_usart1	8		
	Transmit Buffer Empty Interrupt Handler	Registration	Enabled			
	Receive data full Interrupt Handler Regis		Enabled			
	Export Stack Export Export a stack to a con	figuration fragment on the local file system	- 1	· ×		
	To file: D:\WorkSpace Options Include common	for_RE\workspace_2021-07_FSP110\sci.xml		Browse		

# Figure 3-45 Export the RE stack



# Renesas e<sup>2</sup> studio 2021-07 or Higher

# Quick Start Guide

eads 💿 New Thread 😥	Remove 😑 HAL/Common Stacks		🔊 New Stack > 🚊	Extend Stack >	Remo
<ul> <li>HAL/Common</li> <li>Renesas Core</li> <li>Low Power Modes Driver on r_Jg</li> <li>System Driver on r_system</li> <li>CMSIS Driver for USART on r_us;</li> <li>CMSIS Driver for USART on r_us;</li> </ul>	art ch0(SCI0)	Low Power Modes Driver on r_Ipm	<ul> <li>System Driver on r_system</li> </ul>	<ul> <li>CMSIS D</li> <li>CMSIS D</li> </ul>	river for US/
ects	Add Energy Harvesting(EHC) Driver		Team Resource Configurations ✓ Validate ✓ Cut	Ctrl+X	AC or DTC
	<		Copy Paste Paste Collecte Non-secure Callable	Ctrl+C Ctrl+V Delete	
roperties 23			dule Resources As ug As	>	
			ace With	>	]
Stack Import Import stacks from a configuration			ace With		]
Stack Import Import stacks from a configuration From file: D:\WorkSpace_for_RE\wo Stacks:		5Clmodule.driver.re01.1500kb.usart	Ce With		]
Stack Import Import stacks from a configuration From file: D:\WorkSpace_for_RE\wo Stacks:	orkspace_2021-07_FSP110\scixml	5Clmodule.driver.re01.1500kb.usart	ace With		

# Figure 3-46 Import the RE stack

## 3.5.6 Interrupts Page

The Interrupt page allows the management of Event (interrupts) and ISR (Interrupt Service Routines) for use with the RE interrupt framework.

The interrupt Page consists of 2 panes:

- 1. The "User Events" pane shows a list of events that have been created manually by user.
- 2. The "Allocations" pane shows a list of events that have been provided by instantiated RE modules on the Stacks Configuration page.

In each pane, the "Event" column contains event names. The "ISR" column contains subscriber for the corresponding event in the "Event" column.

terrupts	Configuration	Ge	enerate Project Conter
ser Events		New User	Event > 🙀 Remove
Event		ISR	(1)
	Front	100	
Illocations	Event SCI0 RXI (Receive data full)	ISR R_DRIVER_INITIAL_CALL	(2)
			BACK
Interrupt 4	SCI0 RXI (Receive data full)	R_DRIVER_INITIAL_CALL	BACK BACK
Interrupt 4 5	SCI0 RXI (Receive data full) SCI0 TXI (Transmit data empty)	R_DRIVER_INITIAL_CALL R_DRIVER_INITIAL_CALL R_DRIVER_INITIAL_CALL	BACK BACK BACK
Interrupt 4 5 7	SCI0 RXI (Receive data full) SCI0 TXI (Transmit data empty) SCI0 ERI (Receive error)	R_DRIVER_INITIAL_CALL R_DRIVER_INITIAL_CALL	BACK BACK BACK BACK

Figure 3-47 Interrupts Page

A user event and its ISR can be created manually by clicking on the button [New User Event], then select an event to create.

**Note:** ISR registration is required even when using interrupts as triggers for DTC or DMAC. Prepare a dummy function for ISR

🌞 [Tutorial] FSP	Configuration 🗙 💽 bsp_clock_cfg.h						
Interrupts C	onfiguration		O Generate Projec	ct Content			
User Events		Nev	ACMP	,			
Event		ISR	ADC	>			
			AGT	>			
			CAC	>			
			ccc	>			
			CGC	>			
			DIV	>			
			DMAC	>	DMAC0	>	DMAC0 INT (DMAC transfer end 0)
Allocations			DOC	>	DMAC1		
Interrupt	Event	ISR	DTC	>	DMAC2	,	
4	SCI0 RXI (Receive data full)	R_DRIVER_INITIAL_CALLBACK	ELC	>	DMAC3	,	
5	SCI0 TXI (Transmit data empty)	R_DRIVER_INITIAL_CALLBACK	FCU	> П	5111105	_	
7	SCI0 ERI (Receive error)	R_DRIVER_INITIAL_CALLBACK	GDT	>			
8	SCI1 RXI (Receive data full)	R_DRIVER_INITIAL_CALLBACK	GPT	· ·			
9	SCI1 TXI (Transmit data empty)	R_DRIVER_INITIAL_CALLBACK	ICU				
11	SCI1 ERI (Receive error)	R_DRIVER_INITIAL_CALLBACK		2			
Summany BSP (	Clocks Pins Interrupts Event Links Stacks C	omnonents	IIC				
Summary DSF C		omponents	INTEGRATE				
	🖾 New User Event		×				
			~				
	Enter the name of the ISR for	the new user event:					
	myISR						
		ОК	Cancel				
		<u> </u>	Cancel				

Figure 3-48 Adding a new user event



The newly created event will be displayed in the "User Events" pane.

	Configuration × 🕞 bsp_clock_cfg.h				i -
nterrupts Co	onfiguration			Generate F	Project Conten
Jser Events				🗿 New User Event >	Remove
Event		ISR			
DMAC0 INT (DM	MAC transfer end 0)	myISR			
			-		
llocations					
Interrupt	Event		ISR		^
0	SCI1 RXI (Receive data full)		R_DRIVER_INITIAL_CALLBACK		
4	SCI0 RXI (Receive data full)		R_DRIVER_INITIAL_CALLBACK		
5	SCI0 TXI (Transmit data empty)		R_DRIVER_INITIAL_CALLBACK		
7	SCI0 ERI (Receive error)		R_DRIVER_INITIAL_CALLBACK		
8	DMAC0 INT (DMAC transfer end 0)		myISR		
9	SCI1 TXI (Transmit data empty)		R_DRIVER_INITIAL_CALLBACK		~
mmary BSP CI	locks Pins Interrupts Event Links Stacks Com	nonents			

Figure 3-49 User event created

To remove a user event, select the event and click instantiated RE modules in the "Allocations" pane cannot be removed).

nterrupts Co	onfiguration			Generate F	Project Content
User Events				🗿 New User Event >	🔊 Remove
Event		ISR			
DMAC0 INT (DM	/AC transfer end 0)	myIS	R		
<u> </u>					
Allocations					
	Event		ISR		~
Interrupt	Event				
Interrupt 0	SCI1 RXI (Receive data full)		R_DRIVER_INITIAL_CALLBACK		
			R_DRIVER_INITIAL_CALLBACK R_DRIVER_INITIAL_CALLBACK		
0	SCI1 RXI (Receive data full)				
0 4	SCI1 RXI (Receive data full) SCI0 RXI (Receive data full)		R_DRIVER_INITIAL_CALLBACK		
0 4 5	SCI1 RXI (Receive data full) SCI0 RXI (Receive data full) SCI0 TXI (Transmit data empty)		R_DRIVER_INITIAL_CALLBACK R_DRIVER_INITIAL_CALLBACK		

Figure 3-50 Remove user events

Removing a user event can also be done by specifying "Disabled" for it in the Properties window.

cks C	onfigurat	ion					Generate Project C	onte
eads	🐑 New 1	hread 🔊 Remove 😑	HAL/Common Stacks		🔊 New Sta		nd Stack > 🔬 Re	
\$ \$ \$ \$	System Driver CMSIS Driver	odes Driver on r_Ipm on r_system for USART on r_usart ch0(S for USART on r_usart ch1(S	ower Modes on r_lpm	System Driver on r_system	CMSIS Driver     Add DMAC o Driver for	<b>^</b>	Add DMAC or DTC	
		>			Transmission		Driver for Reception	on
ects	<u>e</u> 100	w Object > 🙀 Remove						
			¢					
		ins Interrupts Event Links	Stacks Components					
ropertie	es 🗙						<b>1</b> 8	•
	iver for US	ART on r_usart ch0(S	(10)					
313 01		ART OTT_usart cito(s	(10)					
		ART OTT_USATCON(S	(10)	Value				
tings	Property		(10)	Value				
	Property	te Name		Value g_usart0 0				
	Property Instand Chann	te Name		g_usart0	~			
	Property Instand Chann Transm	el Name	ndler Registration	g_usart0 0	· • • • • • • • • • • • • • • • • • • •			
	Property Instand Chann Transer	el it Buffer Empty Interrupt Ha	ndler Registration	g_usart0 0 Disabled				>
tings	Property Instance Chann Transre	el it Buffer Empty Interrupt Ha	ndler Registration	g_usart0 0 Disabled				>
tings	Property Instance Chann Transre	el it Buffer Empty Interrupt Ha	ndler Registration	g_usart0 0 Disabled			~ 0	>
tings	Property Instanc Chann Transm <	te Name el nit Buffer Empty Interrupt Ha dets full Internet Wandler Configuration × ⓒ bsp_clo	ndler Registration	g_usart0 0 Disabled			0	>
tings	Property Instanc Chann Transm <	el it Buffer Empty Interrupt Ha	ndler Registration	g_usart0 0 Disabled		Gener		>
tings Ir	Property Instanc Chann Transrr Chann (Transrr Channel (Channel) (C	te Name el nit Buffer Empty Interrupt Ha dets full Internet Wandler Configuration × ⓒ bsp_clo	ndler Registration	g_usart0 0 Disabled			ate Project Content	>
tings Ir	Property Instanc Chann Transm <	Configuration X (c) bsp_clo	ndler Registration Prointration ck_cfg.h	g_usart0 0 Disabled Combined	" setting		0	>
tings ti tr	Property Instanc Chann Transm Chann Channel Ch	Configuration X & bsp_clo onfiguration SCI TXI w	ndler Registration Paraistration ck_cfg.h /ill be removed v	g_usart0 0 Disabled Combined	" setting		ate Project Content	>
tings ti tr	Property Instanc Chann Transm Chann Channel Ch	Configuration X (c) bsp_clo	ndler Registration Prointration ck_cfg.h	g_usart0 0 Disabled Combined	" setting		ate Project Content	>
tings ti tr	Property Instanc Chann Transm Chann Channel Ch	Configuration X & bsp_clo onfiguration SCI TXI w	ndler Registration Paraistration ck_cfg.h /ill be removed v	g_usart0 0 Disabled Combined	" setting		ate Project Content	>
tings ti tr	Property Instanc Chann Transm Chann Channel Ch	Configuration X & bsp_clo onfiguration SCI TXI w	ndler Registration Paraistration ck_cfg.h /ill be removed v	g_usart0 0 Disabled Combined	" setting		ate Project Content	>
tings ti tr	Property Instanc Chann Transm Chann Channel Ch	Configuration X & bsp_clo onfiguration SCI TXI w	ndler Registration Paraistration ck_cfg.h /ill be removed v	g_usart0 0 Disabled Combined	" setting		ate Project Content	>
tings ti tr	Property Instanc Chann Transm Chann Channel Ch	Configuration X & bsp_clo onfiguration SCI TXI w	ndler Registration Paraistration ck_cfg.h /ill be removed v	g_usart0 0 Disabled Combined	' setting		ate Project Content	>
tings tings tr	Property Instanc Chann Transm Chann Channel Ch	Configuration X & bsp_clo onfiguration SCI TXI w	ndler Registration Paraistration ck_cfg.h /ill be removed v	g_usart0 0 Disabled Combined	" setting		ate Project Content	>
tings Ir	Property Instanc Chann Transm Chann Channel Ch	Configuration X & bsp_clo onfiguration SCI TXI w	ndler Registration Paraistration ck_cfg.h /ill be removed v	g_usart0 0 Disabled Combined	' setting		ate Project Content	>
tings Ir	Property Instanc Chann Transm Channel Fritunnal FSP Interrupts Co User Event Event DMACO INT (D	Configuration X (c) bsp_clo onfiguration SCI TXI w MAC transfer end 0)	ndler Registration Parity time ck_cfg.h /ill be removed v	g_usart0 0 Disabled Combined	' setting		ate Project Content	2
tings Ir L	Property Instanc Chann Transm Chann Friturial FSF event DMACO INT (D MACO INT (D	te Name el nit Buffer Empty Interrupt Ha c dete full Internet Liandler Configuration X @ bsp_clo onfiguration SCI TXI w MAC transfer end 0)	ndler Registration Prointention ck_cfg.h rill be removed to my/SR	g_usat0 0 Disabled  with "Disabled"	" setting		ate Project Content	>
tings Ir L	Property Instanc Chann Transm Channel Fritunnal FSP Interrupts Co User Event Event DMACO INT (D	Event SCI TXI (Receive data full)	ndler Registration Resistantion ck_cfg.h vill be removed v my/SR	g_usat0 0 Disabled Excision	" setting		ate Project Content	>
tings Ir	Property Instanc Chann Transer Chann Channel Become Composition Co	te Name el nit Buffer Empty Interrupt Ha c dete full Internet Liandler Configuration X @ bsp_clo onfiguration SCI TXI w MAC transfer end 0)	ndler Registration Prointention ck_cfg.h vill be removed v my/SR	g_usat0 0 Disabled Euchilad with "Disabled"	" setting		ate Project Content	2
tings Ir	Property Instanc Chann Transer Chann Channel Become Composition Co	Event SCI TXI (Receive data full) SCI RXI (Receive data full)	ndler Registration Protection kk_cfg.h vill be removed v myrSR ISR R_DR R_DR M/2 R_BR	g_usat0 0 Disabled Euchilad with "Disabled"	" setting		ate Project Content	2
tings Ir	Property Instanc Chann Transer Chann Channel Become Composition Co	Event SCII RXI (Receive data full) SCIO RXI (Receive data full)	ndler Registration Penintration ck_cfg.h /ill be removed v myISR ISR R_DR R_DR R_DR	g_usat0 0 Disabled  Excision with "Disabled"	" setting		ate Project Content	>
tings Ir	Property Instanc Chann Transm Channel Channel Comment Comment Comment DMACO INT (D Allocations Interrupt C 4 5 7	E Name el int Buffer Empty Interrupt Ha Configuration X (c) bsp_clo onfiguration SCI TXI w MAC transfer end 0) Event SCI RXI (Receive data full) SCI RXI (Receive data full) SCI DXI (Receive data full) SCI DXI (Receive error)	ndler Registration Penintention ck_cfg.h rill be removed v myISR ISR R_DR R_DR R_DR N/************************************	g_usat0 0 Disabled  Excision with "Disabled"	" setting		ate Project Content	>

Figure 3-51 Remove user event in Properties window

# 3.5.7 Components Configuration Page

The Components Configuration page enables the individual modules required by the application to be included or excluded.

Modules common to all RE projects are preselected (for example HAL Drivers  $\rightarrow$  re01\_1500kb  $\rightarrow$  r\_adc).

All modules that are necessary for the drivers selected in the Stacks page are included automatically. Users can include or exclude additional modules by checking the box next to the required component.

*Note:* The primary way of adding modules to an application is by using the Stacks page. The Components page is primarily used as a list of components available in the installed Software Package.

Components Configuration			0	
components comgutation			Generate Project Con	tent
		Filter Al	I v Search	
Component	Version	Description	Variant	1
🗸 🚕 BSP				
🗸 🥥 Board				
custom	1.1.0	Custom Board Support Files		
🔽 re01_1500kb_ek	1.1.0	Evaluation Kit RE01 1500KB Support Files		
re01_256kb_ek	1.1.0	Evaluation Kit RE01 256KB Support Files		
✓ ♀ re01_1500kb				
V device	1.1.0	Board support package for R7F0E015D2CFB	R7F0E015D2CFB	
device	1.1.0	Board support package for RE01 1500KB		
device	1.1.0	Board support package for R7F0E014D2CFB	R7F0E014D2CFB	
device	1.1.0	Board support package for R7F0E014D2CFP	R7F0E014D2CFP	
device	1.1.0	Board support package for R7F0E015D2CFP	R7F0E015D2CFP	
device	1.1.0	Board support package for R7F0E016D2DBN	R7F0E016D2DBN	
device	1.1.0	Board support package for R7F0E017D2DBN	R7F0E017D2DBN	
🔽 sdk	1.1.0	Board support package for RE01		
> 💡 re01_256kb				
🗸 🍌 CMSIS				

## Figure 3-52 Components page

#### Note: Coding for User Application

User must NOT add user's application code in main.c. Please note that main.c is overwritten each time the [Generate] button is clicked in the configuration window. This is the characteristic of the Smart Configurator. As such, please implement user's application code in hal\_entry.c or other \*.c files appropriately.



# 4.Building

This chapter describes the build configurations and key build features for e<sup>2</sup> studio IDE.

# 4.1 Build Option Settings

A new project built with the default option can work properly. However, if user would like to change build options (e.g. toolchain version, optimization options, etc.), please follow the following steps before building the project.

1. Right-click on the sample project (e.g. "an4950\_gpio\_re\_1500kb") and select [Properties] or use shortcut keys [Alt] + [Enter] to open the Properties window.

Properties window is supported at workspace, project and source level. Properties window for project supports more configurations that apply across all the files within the same project workspace.

Project E	xplorer 🗙 🛛	p 4	<b>x</b> 77 8 <b>- -</b>	
	0_gpio_re_1500kb		New	>
> 🔂 Inc			Go Into	
> 😕 Device		Open in New Window		
> 🔁 CN			Show In	Alt+Shift+W >
> 🗁 coi	-		Show in	AIL+Shirt+W 7
	cuments		Сору	Ctrl+C
	sh Debug rdwareDebug	Ē	Paste	Ctrl+V
> 🗁 scr	-	×	Delete	Delete
> 👝 SV			Source	>
> 💽 ma			Move	
	4950_gpio_re_1500kb		Rename	F2
	4950_gpio_re_1500kb			
	4950_gpio_re_1500kb		Import	
	4950_gpio_re_1500kb 4950_gpio_re_1500kb		Export	
-	4950_gpio_re_1500kb 4950_gpio_re_1500kb		Build Project	
	4950_gpio_re_1500kb		Clean Project	
TRE_Ap		KD §	Refresh	F5
📋 RE_Lik		<b>~</b>	Close Project	
📁 Tutorial				
		Close Unrelated Project		
			Build Targets	>
		Index		>
Propertie	ち 🛛 🔝 問題 🦓		Build Configurations	>
an4950_g	pio_re_1500kb	0	Run As	>
	Property	*	Debug As	>
Resource Property v Info derived		Team	>	
		Compare With	>	
	editable		Restore from Local History	
	<		MISRA-C	>
		<b>S</b>	C/C++ Project Settings	Ctrl+Alt+P
📂 an4950_g	ipio_re_1500kb		Save build settings report	

Figure 4-1 Open the Properties window

 Click [C/C++ Build] → [Settings] → [Toolchain] tab to view or change toolchain version. Refer to figure 4-2, the current version is 6.3.1.20170620 and click the "Versions" option to change the toolchain version if necessary.



Figure 4-2 Change Toolchain Version



3. Click  $[C/C++ Build] \rightarrow [Environment]$  to set build option and add or edit the environment variables.

pe filter text	Environment			← ▼ ⊂	> -
Resource Builders C/C++ Build Build Variables Environment	Configuration: Hardw	vareDebug [Active]		✓ Manage Configura	tions
Logging Settings	Environment variables	to set		A	dd
Tool Chain Editor C/C++ General MCU Project Natures Project References Renesas QE Run/Debug Settings Task Tags Validation	Variable CWD GCC_VERSION PATH PWD TCINSTALL TC_VERSION	Value D:\WorkSpace_for_RE\w 6.3.1 C:\Program Files (x86)\ D:\WorkSpace_for_RE\w C:\Program Files (x86)\ 6.3.1.20170620	Origin BUILD SYSTEM BUILD SYSTEM BUILD SYSTEM BUILD SYSTEM BUILD SYSTEM	E	lect dit elete define
	<ul> <li>Append variables to</li> <li>Replace native envir</li> </ul>	native environment ronment with specified one		Restore <u>D</u> efaults <u>A</u> p	ply

Figure 4-3 Build Environment Settings



#### 4. Setting Build Options

Build options for compiler and linker, etc. can be set on "C/C ++ Build"  $\rightarrow$  "Settings"  $\rightarrow$  "Tool Setting" tab.

All settings are summarized in "All options:" which shows the commands use in the build process.

The "Build configuration" can be switched via the "Configuration:" dropdown list at the top of the window. Each build configuration manages a set of build options.

Properties for an4950_gpio_re	1500kb	— 🗆 X
type filter text	Settings Build configuration	← ←
<ul> <li>&gt; Resource</li> <li>Builders</li> <li>✓ C/C++ Build</li> <li>Build Variables</li> <li>Environment</li> </ul>	Configuration: HardwareDebug [Active]	✓ Manage Configurations
Logging Settings Tool Chain Editor > C/C++ General > MCU Project Natures Project References Renesas QE Run/Debug Settings Task Tags > Validation	<ul> <li>Target Processor</li> <li>Optimization</li> <li>Warnings</li> <li>Debugging</li> <li>Cross ARM GNU Assembler</li> <li>Preprocessor</li> <li>Includes</li> </ul>	IS: -I"D:\WorkSpace_for_RE\workspace_2021- O7_FSP110\an4950_gpio_re_1500kb\Device" -I"D: WorkSpace_for_RE\workspace_2021- O7_FSP110\an4950_gpio_re_1500kb\Device\Driver \Include" -I"D:\WorkSpace_for_RE\workspace_2021- tings: Confirm settings in command line format
List of settings	<ul> <li>Structure</li> <li>Preprocessor</li> <li>Includes</li> <li>Optimization</li> <li>Warnings</li> <li>Miscellaneous</li> <li>Cross ARM C Linker</li> <li>General</li> <li>Libraries</li> <li>Miscellaneous</li> <li>Other</li> <li>Structure</li> <li>General</li> <li>Cross ARM GNU Create Flash Image</li> <li>General</li> <li>Scross ARM GNU Print Size</li> <li>General</li> </ul>	SCOMMAND' Scross toolchain flags SELAGS - c SOUTPU
?		Apply and Close Cancel

## Figure 4-4 Build Option Settings

The detail of build option is described in the compiler user manual which is stored at "{Compiler installation directory}\share\doc". For example, it can be found in "C:\Program Files (x86)\GNU Tools ARM Embedded\6 2017-q2-update\share\doc".



#### 4.1.1 Recommended Build settings

- As you see, there is "*Toolchain Editor*" under "C/C++ Build", **please do not change the configuration**. The Toolchain editor is used for toolchains that are NOT supported by Renesas build support plugins.
- Please remove all checked options in "Tool Settings" → "Warnings" for now.



Figure 4-5 All "Warning" options should be unchecked

• This driver has been evaluated without "newlib-nano" option. If there is an issue with using the standard library, please disable this option (it is enabled by default when project is created).

Properties for Tutorial	— 🗆 X
type filter text	Settings $\langle \neg \neg \neg \rangle$
<ul> <li>&gt; Resource</li> <li>Builders</li> <li>✓ C/C++ Build</li> </ul>	Configuration: Debug [Active]
Build Variables Environment Logging Settings Tool Chain Editor C/C++ General MCU Project Natures Project References Refactoring History Renesas QE Run/Debug Settings Task Tags Validation	Tool Settings Toolchain   Target Processor   Optimization   Warnings   Debugging   Signu ARM Cross Assembler   Signu ARM Cross Compiler   Signu ARM Cross Clinker   Signu ARM Cross Print Size   Other objects
	Generate map "\${BuildArtifactFileBaseName}.map" Cross reference (-Xlinkercref) Print link map (-Xlinkerprint-map) Use newlib-nano (specs=nano.specs) Use float with nano printf (-u_printf_float) Use float with nano scanf (-u_scanf_float)
	Oser Not With Halo Scaling ( d_Scaling hold)       Do not use syscalls (specs=nosys.specs)       Verbose (-v)       Other linker flags
	Restore <u>D</u> efaults <u>A</u> pply
?	Apply and Close Cancel

Figure 4-6 "newlib-nano" option

# 4.2 Build a Sample Project

A project can be built by steps below:

1. Right-click on the project and select [Build Project]

✓ Sanappio_re_1500kb [HardwareDebug]			
> 🔊 Includes		New	、 、
> 🔑 Device			/
> 👝 CMSIS		Go Into	
> 🔁 config		Open in New Window	
> 👝 Documents		Show In	Alt+Shift+W >
> 👝 Flash Debug		Show in	AIL+Shirt+W /
> 🔁 HardwareDebug		Сору	Ctrl+C
> 🗁 script	Ē	Paste	Ctrl+V
> 🧁 SVD	×	Delete	Delete
> 🖻 main.c	~		Delete
an4950_gpio_re_1500kb.dep		Source	>
an4950_gpio_re_1500kb.ewd		Move	
an4950_gpio_re_1500kb.ewp		Rename	F2
an4950_gpio_re_1500kb.ewt			
an4950_gpio_re_1500kb.eww	è	Import	
an4950_gpio_re_1500kb HardwareDebug.jlini		Export	
an4950_gpio_re_1500kb HardwareDebug.laur	- F	Build Project	
RE_App		build Project	

Figure 4-7 Build a Sample Project

2. The [Console] pane shows 'Build complete.' message to indicate a successful build.



Figure 4-8 Project is built successfully.
3. If the build is unsuccessful, the console window will show error messages which contain details about the errors (location where the error occurs (file, column, and line), reason...). Please check these errors and revise the source code or configuration and rebuild the project.



Figure 4-9 Unsuccessful build shows error messages



# 4.3 Export Build Configuration Settings

Project build settings in e<sup>2</sup> studio IDE can be saved to a file using the Project Reporter feature.

- 1. Right-click at [Project Explorer] to pop up the context menu
- 2. Select [Save build settings report] to save build settings report

🍋 Project Explorer 🗙		🖻 🕏 🍸 🕴 🗖	
✓ ⅔ an4950_gpio_re_1500kb [HardwareDebug]			1
> 🔊 Includes > 😂 Device > 🕞 CMSIS		Paste Delete	Ctrl+V Delete
<ul> <li>&gt; Config</li> <li>&gt; Documents</li> <li>&gt; Flash Debug</li> <li>&gt; HardwareDebug</li> <li>&gt; Device</li> <li>&gt; Construction</li> </ul>	* 2	Source Move Rename Import Export	F2
<ul> <li>Makefile</li> <li>makefile.init</li> <li>objects.mk</li> <li>sources.mk</li> <li>subdir.mk</li> <li>&gt; ⊱ script</li> <li>&gt; SVD</li> </ul>	۶	Build Project Clean Project Refresh Close Project Close Unrelated Project	F5
main.c an4950_gpio_re_1500kb.dep an4950_gpio_re_1500kb.ewd an4950_gpio_re_1500kb.ewp		Build Targets Index Build Configurations	> >
an4950_gpio_re_1500kb.ewt an4950_gpio_re_1500kb.eww an4950_gpio_re_1500kb HardwareDebug.jli an4950_gpio_re_1500kb HardwareDebug.la		Run As Debug As Team	> >
<ul> <li>an4950_gpio_re_256kb</li> <li>RE_App</li> <li>RE_Lib</li> <li>Tutorial</li> </ul>		Compare With Restore from Local History MISRA-C	>
	<b>\$</b>	C/C++ Project Settings Save build settings report	Ctrl+Alt+P

Figure 4-10 Project Reporter

# 5. Debug

This chapter describes the usage of debug configuration and key debugging features for e<sup>2</sup> studio. The following illustration refers to the sample project which is downloaded and imported in chapter 3.4 (e.g. "r01an4950ej0101", hereinafter called "sample project") and based on hardware configuration: J-Link ARM and EK-RE01 1500KB board.

1. Open the sample project workspace in e<sup>2</sup> studio IDE and click [Debug] perspective.

|--|

Figure 5-1 Switch to [Debug] Perspective

Perspective defines the layout views (related to development tools) in the Workbench window. Each perspective consists of a combination of views, menus and toolbars that enable users to perform specific tasks. For instance,

- The [Debug] perspective has views that enable the user to debug the program
- The [RE Configuration] perspective together with "configuration.xml" in the editor window will open the RE configuration, as well as the Package and Properties views for project configuration settings
- The [C/C++] perspective has views that help the user to develop C/C++ programs.

If a user attempts to connect the debugger when not in the [Debug] perspective, e<sup>2</sup> studio will prompt the user to switch to the [Debug] perspective. One or more perspectives can exist in a single Workbench setup. Users can customize them or add new perspectives.

# 5.1 Change Existing Debug Configurations

The debug configuration must be configured when debugging for the first time and it just needs to be done once. An existing debug configuration can be changed as follows.

1. Click the sample project in the [Project Explorer] pane to set focus.

Click [Run]  $\rightarrow$  [Debug Configurations...] or *is* icon (downward arrow)  $\rightarrow$  [Debug Configurations...] to open the "Debug Configurations" window.



#### Renesas e<sup>2</sup> studio 2021-07 or Higher

<u>File Edit Source Refactor Navigate Search Project Renes</u>	as <u>V</u> iews <u>R</u> un	<u>W</u> indow <u>I</u>	Help		
🐔 🔅 🔳 🎋 Debug 🗸 💽 an4950_gpio	re_1500ki	Renesas Devi	ice Partition Manager	🕸 💸 🐐 🗣 🗣 🕫 🔹 🚸 🐲	III 😭 🖏 🕹 😻 🥔 🥭 🛷 🕶 🌛
½ × {i + <b>*&gt; + *&gt; +   =</b>	T	TraceX	>	(no launch history)	Q i 🖻   कि C/C++ 🔅 FSP Configuration
Project Explorer 🐹 🛛 🖹 🕏 🌱 🔋 🗖 🚺 🔂 mair	.c 🛛 🖻	Tracealyzer	>	Debug As	Visualization X 🔂 Package
	<b>Q</b>	Run Debug	Ctrl+F11 F11	Debug Configurations Organize Favorites	ve editor element does not use this view
> 🔊 Includes 29 30	<i>1</i> 00	Run History	F11		
> 🔁 CMSIS 32	0	Run As	>		
> > Documents 34		Run Configu			
> 🔑 Flash Debug > 🍃 HardwareDebug 43	格	Debug Histo Debug As	ry >		
> 🔁 script 44 > 🤁 SVD 445		Debug Confi	gurations		
> C main.c 46 47	<b>9</b>	External Tool			
an4950_gpio_re_1500kb.ewd         48           an4950_gpio_re_1500kb.ewp         59		/* If the	recommends use this part as t e API functions are allocated pies Driver codes, which chang	to RAM, this function must be	
an4950_gpio_re_1500kb.ewt 51 an4950_gpio_re_1500kb.ewt 52		/* Attent	tion: Every code Pin.c is alwa		
an4950_gpio_re_1500kb HardwareDebug.jlink 53		/* This f	alize System Function Driver. function needs to be called af		
an4950_gpio_re_1500kb HardwareDebug.launc		R_SYS_Ini	itialize();	×	
Properties      I      B     B     We Smart Browser	1 <b>-</b> 7 -	8 - 0	E Console 🔀	<b>X</b>   4	û 😫 📰 📲 = 🗟 🖃 🛃 🖬 マ 🗅 マ 🗆
Property	Value		CDT Build Console [an4950_gpio_re_1] Extracting support files		
				ild of configuration HardwareDebu	g for project an4950_gpio_re_1500kb ****
				ex-m0plus -march=armv6-m -mthumb x "an4950 gpio re 1500kb.elf" "a	-mlittle-endian -mfloat-abi=soft -O2 -fmess; an4950 gpio re 1500kb.hex"
			text data bss	berkeley "an4950_gpio_re_1500kb.e dec hex filename	
			-	312 c488 an4950_gpio_re_1500kb	
			16:31:14 Build Finished. 0 e	rrors, 0 warnings. (took 1s.521ms	·)
					~
<	1	>	<		>

Figure 5-2 Open Debug Configurations Window

 In "Debug Configurations" windows, go to [Renesas GDB Hardware Debugging] → [an4950\_gpio\_re\_1500kb.elf HardwareDebug]. Click on the [Main] tab to ensure the load module is "an4950\_gpio\_re\_1500kb.elf".

Debug Configurations			- 🗆 X
Create, manage, and run configurations			Ť.
C 🖻 🐢 🗎 🗙 🖻 🏹 🗸	Name: an4950_gpio_re_1500kb Hard	dwareDebug	
type filter text	📄 Main 🕸 Debugger 🕟 Startu	p 🔲 <u>C</u> ommon 🧤 Source	
<ul> <li>C/C++ Application</li> <li>C/C++ Remote Application</li> </ul>	Project:		
EASE Script	an4950_gpio_re_1500kb		<u>B</u> rowse
C GDB Hardware Debugging	C/C++ Application:		
CDB OpenOCD Debugging	HardwareDebug/an4950_gpio_re_1	500kb.elf	
문핵 GDB Simulator Debugging (RH850) ୭୦ Java Applet		Variables Search Project	Browse
Java Application	Build (if required) before launchin		
🚭 Launch Group		-	
🖳 Remote Java Application	Build Configuration: Select Auto	omatically	$\sim$
✓ C <sup>™</sup> Renesas GDB Hardware Debugging	○ Enable auto build	O Disable auto build	
c <sup>™</sup> an4950_gpio_re_1500kb HardwareDebug c <sup>™</sup> Renesas Simulator Debugging (RX, RL78)	Use workspace settings	Configure Workspace Settings.	
C Refesas simulator Debugging (KX, RE76)			
		Deveet	Annha
Filter matched 13 of 15 items		Re <u>v</u> ert	Apply
?		<u>D</u> ebug	Close

#### Figure 5-3 Select Load Module

3. Switch to the [Debugger] tab, set "J-Link ARM" as the debug hardware and "R7F0E015D2CFB" as the target device.

📄 Main 🏇 Debugger ⊳ Startup 🔲 <u>C</u> ommon 🦃 Source
Debug hardware: J-Link ARM 🗸 Target Device: R7F0E015D2CFB

Figure 5-4 Select Target Device

- 4. Under the [Debugger] tab, go to the [Connection Settings] sub-tab which is related to emulator connection. The following example is based on the environment with J-Link ARM emulator and RSK RE01 board:
  - J-Link
    - Type = "USB".
    - Low Power Handling = "Yes"
  - Interface
    - Type = "SWD"
    - Speed = "320"
  - Connection
    - Reset before run = "**Yes**"
    - Reset before download = "Yes"

*Note:* This debug configuration in Figure 5-5 is shown as an example. The wrong settings may cause malfunction or damage to the hardware. So, be cautious to verify the board and emulator settings before connection.



Debug Configurations		— 🗆 X
Create, manage, and run configurations		- Tom
📑 🖻 🕫 🔚 🗶 🗖 🗖 🗸	Name: an4950_gpio_re_1500kb HardwareDebug	
type filter text	📄 Main 🎊 Debugger 🕨 Startup 🔲 <u>C</u> ommo	n 🖏 Source
<ul> <li>C/C++ Application</li> <li>C/C++ Remote Application</li> <li>EASE Script</li> <li>GDB Hardware Debugging</li> <li>GDB OpenOCD Debugging</li> </ul>	Debug hardware: J-Link ARM V Target De GDB Settings Connection Settings Debug To	evice: R7F0E015D2CFB
GDB Simulator Debugging (RH850)	✓ J-Link	USB
Java Applet	Type J-Link Serial	(A.1-)
Java Application	Settings File	(Auto) \${workspace_loc:\\${ProjName}}\\${Launcl
🛃 Launch Group		
Remote Java Application	Script File	Yes 🗸 🗸
✓ Renesas GDB Hardware Debugging	Low Power Handling Host Name/IP Address[:port number]	TES V
an4950_gpio_re_1500kb HardwareDebug	Interface	
c * Kenesas Simulator Debugging (KX, KL/8)	Type	SWD
	Speed (kHz)	320
	JTAG Scan Chain	520 V
	Multiple Devices	No
	IRPre	0
	DRPre	0
	Connection	~
	Register initialization	Yes 🗸
	Reset on connection	Yes
	Reset before run	Yes
	ID Code (Bytes)	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
	Hold reset during connect	Yes ¥
	Reset before download	No
	Prevent Releasing the Reset of the CM3 C	ore No 🗸
	→ SWV	
	Core clock (MHz)	0
	✓ TrustZone	
	Set TrustZone secure/non-secure bounda	ries No 🗸
Filter matched 13 of 15 items		Re <u>v</u> ert Appl <u>v</u>
?		<u>D</u> ebug Close

## Figure 5-5 Change Connection Settings

Switch to [Debug Tool Settings] sub-tab which is related to debugger behavior. Please refer to the e<sup>2</sup> studio Help content at "e<sup>2</sup> studio User Guide" → "Debugging Projects" for the details.

b HardwareDebug Startup  Common  Source Target Device: R7F0E015D2CFB Settings Debug Tool Settings	
Startup Common Source RM V Target Device: R7F0E015D2CFB	
Settings Debug Tool Settings	
	^
Yes	×
Yes	~
Little Endian	×
ts No	×
No	~
	×
	×
am flash before download No	×
	× ×
	its No No am flash before download No

#### Figure 5-6 Change Debug Tool Settings

- Memory
  - Endian = "Little Endian"

Endian setting of debugger memory reference. This configuration does not affect the target program behavior.

- Break
  - Use Flash Breakpoints = "No"

Setting software breakpoints to flash memory area is restricted, so the hardware breakpoints must be used.

6. Click [Apply] button to confirm the settings. Then click [Debug] to launch the debugger.

7. For a successful connection, [Debug] view to show target debugging information in a tree hierarchy. The program entry point is set at "Reset\_Handler() in "startup\_RE01\_1500KB.c".



Figure 5-7 User Target Connection in the [Debug] View



## 5.2 Create New Debug Configurations

The simplest way to create a new debug configuration is by duplicating an existing one. It can be done by the following steps.

- 1. Repeat step 1 in section 5.1 to open the "Debug Configurations" window.
- Select a debug configuration (e.g. "an4950\_gpio\_re\_1500kb HardwareDebug") and then click icon (to duplicates the currently selected launch configuration). A new debug launch configuration (e.g. "an4950\_gpio\_re\_1500kb HardwareDebug (1)") is created. User can rename it to identify the settings by typing in the "Name" textbox then click [Apply] button.

Debug Configurations		_	— 🗆 X
Create, manage, and run configurations			The
			25
C/C++ Application C/C++ Application EASE Script GDB Hardware Debugging GDB OpenOCD Debugging GDB Simulator Debugging (RH850) Java Applet	Name:       an4950_gpio_re_1500kb HardwareDebug (1)         Main       Debugger         Debug hardware:       J-Link ARM         Target Device:       R7F0E015D2CFB         GDB Settings       Connection Settings         GDB Connection Settings       Debug Tool Settings         GDB Connection Settings       Host name or IP address:		
<ul> <li>Java Application</li> <li>Launch Group</li> <li>Remote Java Application</li> <li>Renesas GDB Hardware Debugging</li> <li>an4950_gpio_re_1500kb HardwareDebug</li> <li>an4950_gpio_re_1500kb HardwareDebug (1)</li> <li>Tutorial Debug_Flat</li> <li>Renesas Simulator Debugging (RX, RL78)</li> </ul>	Connect to remote GDB server GDB port number: 61234 GDB GDB Command: arm-none-eabi-gdb Step Mode	Browse	Variables
	Additional GDB Server Arguments		
			^
Filter matched 15 of 17 items		Re <u>v</u> ert	Apply
?		<u>D</u> ebug	Close

Figure 5-8 Duplicate A Selected Debug Launch Configuration

3. The debug launch configuration can be configured as described in chapter 5.1. For example, change the Debug Hardware to "E2 Lite (ARM)".

4. If the launch configuration was added with [local] and \* (red star) marker, it is not yet attached to any project. Then please specify the project name in the Common tab.

Name: an4950_gpio_re_1500kb HardwareDel	bug
📄 Main 🕸 Debugger 🕨 Startup 🔲 🖸	ommon 🧤 Source
Save as O Local file Shared file: \an4950_gpio_re_1500k	b <u>B</u> rowse

Figure 5-9 Attach Launch Configuration to Specific Project



#### 5.3 Launch Bar

This section explains the usage of 'Launch Bar', which is supported from V6.0.0 or later version. Launch Bar is located in the toolbar area of  $e^2$  studio main window.

The interface is very simple as shown below to build and debug for the selected launch target.

File	Edit	Source	Refactor	Navigate	Search	Project	Renesas Views	Run	Window	Help
5	*		🎋 Deb	ug	~	💽 myprj	HardwareDebug		✓ ∛	F   X
			~						. <u> </u>	
But	tons to	perform b	uild/launch d	lebug for sele	cted target		Select build/de	bug tar	get	

#### Figure 5-10 Launch Bar interface

Launch Bar buttons behave as follows (please select the build/ debug target in advance):

• Solution builds the load module of the selected launch configuration.

Note: There is another build button in the "File toolbar" that builds the active build configuration of Project Explorer, while the launch bar does not reflect the active state in Project Explorer.

• 🛛 🎋 📔 buttons are triggers of debugger launch and terminate the selected launch target.

Launch Bar and build button can be hidden through the following dialog.

• Click [Window] menu  $\rightarrow$  [Preferences], then click [Run/Debug]  $\rightarrow$  [Launching]  $\rightarrow$  [Launch Bar].

Preferences				$\times$
type filter text		Launch Bar	() <b>v</b>	⇒ ▼ 8
Renesas QE Run/Debug Console External Tools GDB-Servers Launching Default Launchers Launch Bar Launch Configurations MCU Peripherals Views Perspectives String Substitution View Management View Performance	>	Preferences for the Launch Bar ✓ Enable the Launch Bar ✓ Enable the Build button Always show the target selector Restore Defaults	Appl	У
? 🗠 🖆 🔘		Apply and Close	Cancel	

Figure 5-11 Show/hide Launch Bar

## 5.4 Basic Debugging Features

This section explains the typical Debug views supported in e<sup>2</sup> studio IDE.

- Standard GDB Debug (supported by Eclipse IDE framework): Breakpoints, Expressions, Registers, Memory, Disassembly and Variables
- Renesas Extension to Standard GDB Debug: Eventpoints, IO Registers and Trace.

The following are some useful buttons in the [Debug] view:



Figure 5-12 Useful Toolbars in Debug Views

The program is run by clicking IP button or pressing [F8].

The program can be paused by breakpoint or by clicking button. When the program is paused, user can perform the following operations:

- Note: button or [F5] can be used for stepping into the next method call at the currently executing line of code.
- Description by the security of the stepping over the next method call (executing but without entering it) at the currently executing line of code.
- IP button can be clicked again to resume running.
- To stop the debugging process, button is clicked to end the selected debug session and/or process or process.

The other operations are as following:

- 🥸 button can be clicked to start new debug session.
- button can be clicked to reset the program to entry point at the PowerOn Reset.
- Use button is used for re-downloading the binary file to target system.

*Note:* To demonstrate the features in the following section, please use the sample code for RE01 from Renesas website as instruction in chapter 3.4.

#### 5.4.1 Breakpoints View

The Breakpoints view stores the breakpoints that were set on executable lines of a program. If a breakpoint is enabled during debugging, the execution suspends before that line of code executes. e<sup>2</sup> studio allows software and hardware breakpoints to be set explicitly in the IDE. Any breakpoints added via double click on the marker bar are by default hardware breakpoints. If the hardware resources are not there then the breakpoint setting will fail. In case of a hardware breakpoint setting failure, an error message will prompt the user to switch to a software breakpoint.

To select a default Hardware or Software breakpoint type:

Right-click on the marker bar to pop up the context menu. For a hardware breakpoint, select [Breakpoint Types] → [e<sup>2</sup> studio Breakpoint]. For a software breakpoint, select [Breakpoint Types] → [C/C++ Breakpoints].

**Note:** Software breakpoint is prohibited in flash memory area, hardware breakpoint is not supported in RAM area.

To set breakpoints in flash memory area, please use hardware breakpoint.

To set breakpoints in RAM area, please use software breakpoint.

To set a breakpoint:

- 1. Open "main.c", double-click on the marker bar located in the left margin of the [C/C++ Editor] pane to set a breakpoint. A dot 🔐 (Hardware breakpoint) or 🐭 (Software breakpoint) is displayed in the marker bar depending on the [Breakpoint Type] selected. [Breakpoint Type] is hardware breakpoint by default.
- 2. Alternatively, right-click at the marker bar to choose [Toggle Hardware Breakpoint] or [Toggle Software Breakpoint] to set a hardware breakpoint or a software breakpoint .
- Click [Windows] → [Show View] → [Breakpoints] or icon <sup>So</sup> (or use shortcut key [Alt] + [Shift] + [Q], [B]) to open the [Breakpoints] view to view the corresponding software breakpoints set. Software breakpoints can be enabled and disabled in the [Breakpoints] view.



To disable breakpoints, users can choose to disable specific breakpoints or to skip all breakpoints:

- 1. To disable a specific breakpoint, right-click on the Software breakpoint or Hardware breakpoint located in the left margin of the [C/C++ Editor] pane and select [Disable Breakpoint], or uncheck the related line in the Breakpoints view. A disabled breakpoint is displayed as a white dot ( or or ).
- 2. To skip all breakpoints, click on the kick icon in the Breakpoints view. A blue dot with a backslash will appear in the editor pane as well as in the Breakpoints view.



Figure 5-13 [Breakpoints] view



#### 5.4.2 Expressions View

Expressions view monitors the value of global variable, static variable or local variable during debugging. These variables (including the local variables in scope) can be set for real-time refresh.

Expression	Туре	Value	Address	Name : gs_lpm_init_flg		
		1 '\001'	0x2000a4c0	Details:1 '\001' Default:1 '\001' Decimal:1 Hex:0x1 Binary:1 Octal:01		
				<		
main.c startup_F  278	& (MSTP->MS	r_system_api.c log r_	_ipm_api.c ∞			
279	&(MSTP->MS					
280		,				
281						
282	** Flag to pr	went the calling	<u>more th</u> an once from app	plication. */		
283	tatic volatil	e uint8 t gs lpm	init flg = R LPM PRV FL	AG NOT INITILIZED;		
284						
285	tatic volatil	e uint8_t gs_lpm_	_snz_flag = R_LPM_PRV_FL	AG_SNZ_DISABLE;		
286	tatic volatil	e uint16_t gs_lpm	1_sbycr_ssbymp_retpm = 0	J;		
207	tatic volatil	e uint8_t gs_lpm_	snzcr_snzdtcen_rxdregen	= 0U;		
287 .		ARAM CHECKING ENA	ABLE == 1)			
	fit (LPM_CFG_P/					
288 😁				_start_ope = LPM_POWER_SUPPLY_M		

#### Figure 5-14 [Expressions] View

#### To watch a global variable,

- 1. Click [Window]  $\rightarrow$  [Show View]  $\rightarrow$  [Expressions] or icon  $\stackrel{\text{formula}}{=}$  to open the [Expressions] view
- 2. Drag and drop a global variable over to the [Expressions] view. (Alternatively, right-click at the global variable to select "Add Watch Expression..." menu item to add it to the [Expressions] view).
- 3. In the [Expressions] view, right-click to select "Real-time Refresh" menu item. This refreshes the expression value in real-time when the program is running. The character "R" indicates that this global variable will be updated in real-time.
- 4. To disable the "Real-time Refresh", simply right-click to select "Disable Real-time Refresh" menu item.



Local variables can be added in the same way. However, the watch is not available when the program is running out of the scope of the variable.

Expression	Type	Value	Address	Multiple errors reported.
R gs_lpm_init_flg	volatile uint8 t	1 '\001'	0x2000a4c0	
R data		Error: Multiple errors		1) Failed to execute MI command:
<ul> <li>              Add new express      </li> </ul>				<ul> <li>-var-create - * data</li> <li>Error message from debugger back end:</li> <li>-var-create: No symbol "data" in current context.</li> <li>2) Failed to execute MI command:</li> <li>-data-evaluate-expression &amp;(data)</li> <li>Error message from debugger back end:</li> <li>No symbol "data" in current context.</li> <li>3) Failed to execute MI command:</li> <li>-var-create - * data</li> <li>Error message from debugger back end:</li> <li>-var-create: No symbol "data" in current context.</li> <li>4) Failed to execute MI command:</li> <li>-var-create: No symbol "data" in current context.</li> <li>4) Failed to execute MI command:</li> <li>-data-evaluate-expression data</li> <li>Error message from debugger back end:</li> <li>No symbol "data" in current context.</li> </ul>
🖻 main.c 😣 🖻 startu		r_system_api.c	☑ r_lpm_api.c	- 6
42				^
44 e 45 00001b3c 46	<pre>int main() {     uint32_t d</pre>	ata <mark>;</mark>		
47 48 49 50 51	/* If the /* It copi	API functions are es Driver codes,	allocated t which change	emplate this device's start-up code. ****/ to RAM, this function must be executed first */ e the allocation area from ROM to RAM, by using r_[1 /s allocated to RAM. These functions are called from

Figure 5-15 Add local variable to Expression view



#### 5.4.3 Registers View

Registers view lists the information about the general registers of the target device.

Name	Value	Description
🖌 🛗 General Registers		General Purpose and FPU Register Group
1010 rO	0x20000800	
<sup>1010</sup> r <b>1</b>	0xc	
1010 r2	0x10	
1010 r3	Охс	
<sup>1010</sup> r <b>4</b>	0x1bc8	
1010 r5	0x20000824	
1010 r6	0x20000824	
Name : r0 Hex:0x20000800 Decimal:5368729 Octal:040000040 Binary:10000000 Float:1.0844668 Default:5368729 Raw:0x20000800	000 00000000000010000000 87e-019	10000

#### Figure 5-16 [Registers] View

To view the general register "r0",

- 1. Click [Window]  $\rightarrow$  [Show View]  $\rightarrow$  [Registers] or icon <sup>110</sup> to open the [Registers] view.
- 2. Click "r0" to view the values in different radix format.

Values that have been changed are highlighted (e.g. in yellow) in the [Registers] view when the program stops.



#### 5.4.4 Memory View

Memory view allows users to view and edit the memory presented in "memory monitors". Each monitor represents a section of memory specified by its location called "base address". The memory data in each memory monitor can be presented in different "memory renderings", which are the predefined data formats (e.g. Hex integer, signed integer, unsigned integer, ASCII, image, etc.).

To view the memory of a variable (e.g. "gs\_lpm\_init\_flg"),

- 1. Click [Window]  $\rightarrow$  [Show View]  $\rightarrow$  [Memory] or icon  $\bigcirc$  to open the [Memory] view.
- 2. Click the icon to open [Monitor Memory] dialog box. Enter the address of the variable "gs\_lpm\_init\_flg".

e <sup>2</sup> Monitor Memory	<					
Enter address or expression to monitor:						
&gs_lpm_init_flg ~	]	The global va	• -	0	•	
⑦ OK Cancel		in memory re	nderings of "H	Hex Integer" f	ormat.	
🖙 Console 🧟 Tasks 🗳 Renesas De	Problems 🏽 Smart B	row 🗟 Debug	jger C 🟮 Me	mory 🛛 🖪 Sta	ck Analy 🛛 🔓	eripherals 🗖 🗖
				se 1019	uu 🔗 📑 🛃 🎫	
Monitors 🕂 🕈 🕷	&gs_lpm_init_flg : 0x20		nteger> 🛛 🦂	New Renderir	ngs	
&gs lpm init flg	Address	0 - 🛪	4 - 7	8 - B	C - F	~
↑ ↑	000000002000A4C	<mark>0 0000000 0</mark>	00000000	00000000	00000000	
	000000002000A4D	00000000	00000000	00000000	00000000	
	000000002000A4E	00000000	00000000	00000000	00000000	
	000000002000A4F	00000000	00000000	00000000	00000000	
lemory Monitor for "gs_lpm_init_flg"	000000002000A50	00000000	00000000	00000000	00000000	
is specified by the address	000000002000A51	00000000	00000000	00000000	00000000	
"&gs lpm init flg"	000000002000A52	00000000	00000000	00000000	00000000	
0 _ 1 0	000000002000A53	00000000	00000000	00000000	00000000	

Figure 5-17 [Memory] View (1/2)



To add new renderings format (e.g. Raw Hex) for the variable "gs\_lpm\_init\_flg",

1. Click the tab <u>• New Renderings...</u> to select "Floating Point" to add the rendering

This creates a new tab named "&gs\_lpm\_init\_flg <Floating Point>" next to the tab "&gs\_lpm\_init\_flg <Hex Integer>".

					19 Eu 😵	" 🖑 🗋 🛃 📲 👪	•
Monitors	🕂 🗙 🖗	&gs_lpm_init_flg <hex inte<="" td=""><td>ger&gt;</td><td>🕈 New Re</td><td>enderings</td><td></td><td></td></hex>	ger>	🕈 New Re	enderings		
&gs_lpm_init_flg		Memory Monitor: &gs_lpn	n_init_flo	g : 0x2000/	\4C0		
		Select rendering(s) to creat	e:	-			
		Waveform Hex Integer				∧ Add Renderi	ng(s)
		Fixed Floating Point Fixed Point					
		lmage Raw Image					
		Floating Point					
🖻 Console 🦃 Tasks 📮 Re	anasas Da	Traditional	R D	ehi er C	Memory 8 1 Stad	▼	
□ Console	enesas De 💠 🗙 🔆	🗈 Problems 🤏 Smart Brow			in اس 🖏	k Analy ⅔ Peripherals " � î ថ d	- □ •
Monitors		Problems & Smart Brow	eger>	&gs_lpm_	init_flg : 0x2000A4C0 <	k Analy ⅔ Peripherals " 🔗 🖀 🛋 📾 🖩 🖏 Flo 🛛 🗣 New Rendering	□ □ ▼ ▼ gs]
		🗈 Problems 🤏 Smart Brow	eger> 0.00		in اس 🖏	k Analy ⅔ Peripherals " � î ថ d	□ □ gs]
Monitors		Problems Smart Brow <u> &amp;gs_lpm_init_flg</u> <hex inte<br=""><del> 0x0000000200004C0</del></hex>	eger> 0.000 0.000	&gs_lpm_ 0000E0	init_flg : 0x2000A4C0 < 0.00000E0	k Analy 🔀 Peripherals • 🚸 🗈 🖻 🐟 🖩 😨 📲 Flo 🕺 🗣 New Renderin 0.000000E0	□ □ gs
Monitors		Problems  Smart Brow &gs_lpm_init_flg <hex inte<br=""> 0x000000002000A4C0 0x000000002000A4CC</hex>	eger> 0.000 0.000 0.000	&gs_lpm_ 0000E0 0000E0	په ۱۱۰۵ په ۱۱۰۵ په ۱۱۰۵ په ۱۱۰۵ په ۱۱۰۵ په ۱۱۰۵ پې ۱۱۰۵ ۵.000000E0 ۵.000000E0	k Analy 등 Peripherals •	▼ ▼ gs]
Monitors		Problems Smart Brow &gs_lpm_init_flg <hex inte<br="">bx000000002000A4C0 bx000000002000A4CC bx000000002000A4D8</hex>	eger> 0.000 0.000 0.000 0.000	&gs_lpm_ 0000E0 0000E0 0000E0	(init_flg : 0x2000A4C0 < 0.000008E0 0.000008E0 0.000008E0 0.000008E0	k Analy 등 Peripherals •	□ □ gs]
Monitors		Problems	eger> 0.000 0.000 0.000 0.000 0.000	&gs_lpm_ 0000E0 0000E0 0000E0 0000E0	(init_flg : 0x2000A4C0 < 0.00000E0 0.00000E0 0.00000E0 0.00000E0 0.00000E0	k Analy 등 Peripherals •	□ □ gs
Monitors		Problems <sup>∞</sup> Smart Brow &gs_lpm_init_flg <hex inte<br="">0x000000002000A4C0 0x000000002000A4CC 0x000000002000A4D8 0x000000002000A4E4 0x000000002000A4F0</hex>	eger> 0.000 0.000 0.000 0.000 0.000	&gs_lpm_ 0000E0 0000E0 0000E0 0000E0 0000E0	init_flg:         0x2000A4C0           0.00000000         0.00000000           0.00000000000         0.00000000000000000000000000000000000	k Analy Se Peripherals •	▼ ▼ gs]
Monitors		Problems	eger> 0.000 0.000 0.000 0.000 0.000 0.000	&gs_lpm_ 0000E0 0000E0 0000E0 0000E0 0000E0 0000E0	<pre>init_flg: 0x2000A4C0 &lt; 0.00000E0 0.00000E0</pre>	k Analy Se Peripherals •	▼ ▼ gs

Figure 5-18 [Memory] View (2/2)



#### 5.4.5 Disassembly View

Disassembly view shows the loaded program as assembler instructions mixed with the source code for the comparison. Current executing line is highlighted by an arrow marker in the view. In the [Disassembly] view, user can set breakpoints at the assembler instruction, enable or disable these breakpoints, step through the disassembly instructions and even jump to specific instruction in the program.

	int main()	^ L		^
45 00001b3c			00001b24: 0x00000000 SystemInit+140 b.n 0x1b28 <systeminit+144> 00001b26: 0x00000140 SystemInit+142 ands r1.r0</systeminit+144>	
46	uint32_t data;		00001b28: 0x000000 (adr r5, 0x1b38 <system< th=""><th></th></system<>	
47 48	the second s			
48	<pre>/**** It recommends use this part as template this /* If the API functions are allocated to RAM. this</pre>		<sup>00001b2a:</sup> <sup>0xfffff</sup> This allows the assembly <sup>f1</sup>	
50	/* It copies Driver codes, which change the allocated		99991b39 · evenee	
50	/* Attention: Every code Pin.c is always allocated		e eeeelb32; execeed source to be linked with the C	
S₂ 00001b3e	R SYS CodeCopy();		00001b34: 0x00001	
8 5 0000105E	/* initialize System Function Driver. */		eeeelb36: exeeeee source (active debug context).	
54	/* This function needs to be called after R SYS Co	ode(	00001b38: 0x00000 (dottvo dobdy context). (adr r5, 0x1b3c <main></main>	
55 00001b42	R SYS Initialize();		00001b3a: 0xffff10b5 SystemInit+162 vsli.32 d27, d0, #31	
56	/* Initialize LPM Function Driver. */		<pre>re 52 R_SYS_CodeCopy();</pre>	
57	/* This function needs to be called before call R	LPN *	• 00001b3e: 0xfef721fd main+2 b1 0x584 <r_sys_codecopy></r_sys_codecopy>	
58 00001b46	R LPM Initialize();		55 R_SYS_Initialize();	
59	/* Set Power Supply Open Control Register (VOCR re	gis	00001b42: 0xfef7fbfc main+6 bl 0x53c <r_sys_initialize></r_sys_initialize>	
60	/* If you doesn't clear bits which corresponding y	/our	58 R_LPM_Initialize();	
61	/* correctly value of input signal.		00001b46: 0xfef765fd main+10 bl 0x614 <r_lpm_initialize></r_lpm_initialize>	
62	/* The VOCR register is used for control such that		69 R_LPM_IOPowerSupplyModeSet(0x00);	
63	/* the AVCC0, AVCC1, IOVCC0, IOVCC1, IOVCC2, IOVCC3		00001b4a: 0x00000020 main+14 movs r0, #0	
64	/* the operation of circuits that operate with the		<pre>000001b4c: 0x00f034f8 main+16 bl 0x1bb8 &lt;_R_LPM_IOPowerSupplyModeSet_vene</pre>	
65	/* does not affect the circuits that operate with		78 PORTO->PODR = 0x0380;	
66	/* Thie feature is mainly usign when the device is		00001b50: 0x0000e021 main+20 movs r1, #224 ; 0xe0 00001b52: 0x0000064a main+22 ldr r2. [pc. #24] : (0x1b6c <main+48>)</main+48>	
67	/* Default value is 0xFF which is not propagated s		00001b52: 0x0000064a main+22 ldr r2, [pc, #24] ; (0x1b6c <main+48>) 00001b54: 0x00008999 main+24 ls1s r1, r1, #2</main+48>	
68	/* Please change the value of argument with your 1	ang	00001054 000001186 main+26 strh r1, [r2, #0]	
69 00001b4a	<pre>R_LPM_IOPowerSupplyModeSet(0x00); /**** End of template code. ****/</pre>		83 PORTO->PDR = 0x0380;	
	/ code		1 0 m 2, #2]	
Source	address	***	Opcodes [ Function Offsets 2, **2]	
	user code for user init and system ope	tenet	(0x1b70 <main+52>)</main+52>	
74	/*************************************	***	00001b5c: 0x0000013b main+32 subs r3, #1	
75	'		87 for(data = 0; data < 50000;data++)	
76			00001b5e: 0x0000002b main+34 cmp r3, #0	
77	/*set P007, 008, and 009 output to high (LED off)'	41		

Figure 5-19 [Disassembly] View

To view both C and assembly codes in a mixed mode,

- 1. Click [Window]  $\rightarrow$  [Show View]  $\rightarrow$  [Disassembly] or icon  $\blacksquare$  to open the [Disassembly] view
- 2. Click icon 5 to enable the synchronization between assembly source and the C source (active debug context).
- 3. In [Disassembly] view, right-click at the address column to select "Show Opcodes" and "Show Function Offsets".
- 4. You can enable source addresses within the editor using the context menu.



Figure 5-20 Source Addresses Menu

#### 5.4.6 Variables View

Variables view displays all the valid local variables in the current program scope.

Please refer to the 'Expressions' view to watch global variables or external variables out of current program scope.



#### Figure 5-21 [Variables] View

To observe a local variable (e.g. "data" in function "main()"),

- 1. Click [Window]  $\rightarrow$  [Show View]  $\rightarrow$  [Variables] or icon<sup>[M]=</sup> to open the [Variables] view.
- 2. Step into the function "main()" to view the value of local variable "data".

#### Note:

The variables which are optimized out or temporarily allocated to accumulator registers may not appear in this view. Please refer to the Disassembly view if necessary.

By disabling optimization, variables will become visible in most cases. However, this will give up all the benefits of optimization such as memory efficiency, code size reduction and performance improvement.



#### 5.4.7 Eventpoints View

An event refers to a combination of conditions set for executing break or trace features during program execution. [Eventpoints] view enables user to set up or view defined events of different category e.g. trace start, trace stop, trace record, event break, before PC, performance (timer) start and performance (timer) stop.

The number of events that can be set and the setting conditions differs with each MCU. These are two (2) types of events:

- Execution address: The emulator detects execution of the instruction at the specified address by the CPU. It can be a "before PC" break (e.g. with event condition is satisfied immediately <u>before</u> execution of the instruction at the specified address) or other events (e.g. with event condition is satisfied immediately <u>after</u> execution of the instruction at the specified address).
- Data access: The emulator detects access under a specified condition to specified address or specified address range. This allows to setup complex address and data matching criteria.

Event combination (e.g. OR, AND (cumulative) and Sequential) can be applied to two (2) or more events.

				6	<b>x ‰</b> §		/4 OA: 0/2   🗞	<b>1</b>				
							4 0/2					
/pe	Address	Data	Cou	limer	Han	Comment						
Trace Start												
Trace Stop			_	_				_				
📽 Trace Record												
😲 Event Break								_				
🔊 Timer Start												
🔊 Timer Stop												
								>				
ject Saved Templates												
						┛						
Jool			C9 Edi	t Event	Break	➡						
Jeet			C <sup>®</sup> Edi	it Event	Break	ł				Trig	gger: OR	~ <del>†</del>
			<b>C</b> g Edi Type		Break Add	ress	Data	Count	Timer		gger: OR Comment	~ ¥
						ress	Data	Count	Timer			~ ÷
						ress	Data	Count	Timer			~ <b>↓</b>
						ress	Data	Count	Timer			~ <b>↓</b>
						ress	Data	Count	Timer			~ ¥
						ress	Data	Count	Timer			~ ¥
					Add		Data )/0 OA: 0/2 All:		Timer			~ <b>U</b>

Figure 5-22 [Eventpoints] View (1/2)

To set an event break for a global variable when address/data is matched (e.g. when gs\_lpm\_init\_flg is written),

- 1. Click [Window]  $\rightarrow$  [Show View]  $\rightarrow$  [Eventpoints] or icon  $\bullet$  to open the [Eventpoints] view.
- 2. Double-click on the "Event Break" option to open [Edit Event Break] dialog box
- 3. Click [Add...] button to continue.

	e <sup>2</sup> Add Eventpoint	×
	Eventpoint Type: Da	ata Access
	Address Settings D	
	Address:	&gs_lpm_init_flg ~
	Address Settings	Data Access Settings
	Data Settings:	
	Read/Write:	Write
	Size:	Not Specified $\checkmark$
	Bus Master:	CPU ~
	Compare Setting	ngs:
	Compare:	0
	Mask Value:	0
	Comparison:	Equals
Clear Edit Even	t Drook	×
s cuit even	Греак	Trigger: OR V V 1
-		
Type ☑ ▲ OR	Address Data &gs_lpm_init_flg Write All	Count Timer Handle Comment
Add	dit Delete PC: 0/0 OA: 1/2	2 All: 1/2
		OK Cancel

Figure 5-23 [Eventpoints] View (2/2)

- 4. Select "Data Access" as the eventpoint type.
- 5. Go to the [Address Settings] tab, click the icon to browse for the symbol "gs\_lpm\_init\_flg". (The address of this global variable is "&gs\_lpm\_init\_flg")
- 6. Next, switch to the [Data Access Settings] tab, select "Write" from "Read/Write" dropdown list. Click [OK] to proceed.

7. Ensure that the event break for "gs\_lpm\_init\_flg Write All" is set and enabled in the [Eventpoints] view. Reset to execute the program from the start.

Expression	Туре	Value	Address	Name : gs_lpm_init_flg	
R gs_lpm_init_flg	volatile uint8_t	1 '\001'	0x2000e250	Details:1 '\001' Default:1 '\001'	
				Decimal:1 Hex:0x1 Binary:1 Octal:01	
				<	
I main.c startu 319			i.c	ction.	^
319 321 322 000007a0 323 000007a4 324 325 000007ae	<pre>* * @brief T} void R_LPM_1 {     if (R_LF</pre>	his function : Initialize (vo PM_PRV_FLAG_NO lpm_init_flg	initialize r_lpm fun oid) OT_INITILIZED == gs_ = R_	lpm_init_flg) LPM_PRV_FLAG_INITILIZED;	•
<ul> <li>317</li> <li>319</li> <li>321</li> <li>322 000007a0</li> <li>323 000007a4</li> <li>324</li> <li>325 000007ae</li> <li>326 000007b4</li> </ul>	<pre>* @brief Th •void R_LPM_1 { • if (R_LP { gs_ gs_</pre>	his function : Initialize (vo PM_PRV_FLAG_No lpm_init_flg lpm_snz_flag	initialize r_lpm fun oid) OT_INITILIZED == gs_ = R_ = R_	lpm_init_flg) LPM_PRV_FLAG_INITILIZED LPM_PRV_FLAG_SNZ_DISABLE;	
319 321 322 000007a0 323 000007a4 324 325 000007ae	<pre>* @brief Th void R_LPM_1 { if (R_LF</pre>	his function Initialize (vo PM_PRV_FLAG_No <u>lpm_init_flg</u> lpm_snz_flag lpm_sbycr_ssby lpm_snzcr_snzo	initialize r_lpm fun oid) OT_INITILIZED == gs_ = R_ = R_	lpm_init_flg) LPM_PRV_FLAG_INITILIZED LPM_PRV_FLAG_SNZ_DISABLE; ;	
<ul> <li>⇒1/ 319</li> <li>⇒21</li> <li>⇒321</li> <li>⇒323</li> <li>⊗00007a0</li> <li>⇒325</li> <li>⊗00007a0</li> <li>⇒326</li> <li>⊗00007ba</li> <li>⇒327</li> <li>⊗00007ba</li> <li>⇒328</li> <li>⊗00007c0</li> </ul>	<pre>* * @brief Th = void R_LPM_1 {</pre>	his function Initialize (vo PM_PRV_FLAG_No <u>lpm_init_flg</u> lpm_snz_flag lpm_sbycr_ssby lpm_snzcr_snzo	<pre>initialize r_lpm fun oid) OT_INITILIZED == gs_ = R_ = R_ ymp_retpm = 0U dtcen_rxdregen = 0U ING_ENABLE == 1)</pre>	lpm_init_flg) LPM_PRV_FLAG_INITILIZED LPM_PRV_FLAG_SNZ_DISABLE; ;	

Figure 5-24 Execution of Event Break

Figure 5-24 shows that when gs\_lpm\_init\_flg is assigned with value R\_LPM\_PRV\_FLAG\_INITILIZED, the program stops at code line No.326 (right after the line of code writing to gs\_lpm\_init\_flg).



#### 5.4.8 IO Registers View

IO Registers are also known as the Special Function Registers (SFR). The [IO Register] view displays all the registers set defined in a target-specific IO file, including their address, hex and binary value. User can further customize their own [IO registers] view by adding IO registers selectively to the [Selected Registers] pane.

🗱 Variables 🔎 Eventpoints 🌯 E	Breakpoints 🔤 R	egisters 🛋 Mo	odules 🖋 Exp	oressions 🔲 IO	Registers 🛛
Name	Value (Hex)	Value (Bin)	Address	Access	
✓ PORT0					
> PCNTR1	0x03800380	000000111	0x400400	00 RW	
> PODR	0x0380	000000111	0x400400	00 RW	
> PDR	0x0380	000000111	0x400400	02 RW	
> PCNTR2	0x0000000	00000000	0x400400	04 R	
> PIDR	0x0000	00000000	0x400400	06 R	
> PCNTR3			0x400400	08 W	
> PORR			0x400400		
> POSR			0x400400	0a W	
> PORT1					
> PORT2					
All Registers Selected Register	S				
🗱 Variables 🔲 IO Reg	isters 🛛 👓 Event	points 🔏 Break	points 🕮 Reg	isters 🍽 Module	es
Name	Value (Hex)	Value (Bin) /	Address	Access	
<ul> <li>✓ ● PORT0</li> </ul>					
> • PODR	0x0380	000000111 0	0x40040000	RW	
> ● PDR	0x0380	000000111 0	0x40040002	RW	
All Registers	Desisters				
All Registers Selected	a Registers				

Figure 5-25 [IO Registers] View

To view selected IO registers (e.g. PODR and PDR in PORT0),

- Click [Windows] → [Show View] → [Others...]. In "Show View" dialog, click [IO Registers] under [Debug] or icon to open the [IO Registers] view
- 2. Under the [All Registers] tab, locate [PORT0] in the [IO Registers] view. Expand the PORT0 IO register

list. You could also use Search button \_\_\_\_\_ in the IO Register toolbar to quickly search by name.

- 3. Drag and drop the "PODR" and "PDR" to the [Selected Registers] pane. A green dot <a>beside the IO register indicates the status of being the selected register(s).</a>
- 4. Switch to the [Selected Registers] tab to view "PODR" and "PCR" of the "PORT0" IO register.

The expanded IO register list may take a longer time to load in the [All Registers] pane. Hence, it is advisable to customize and view multiple selected IO registers from the [Selected Registers] pane.

#### 5.4.9 Trace View

Tracing means the acquisition of bus information per cycle from the trace memory during user program execution. The acquired trace information is displayed in the [Trace] view. This helps users to track the program execution flow to search for and examine the points in the program where problems arise.

The trace buffer is limited, the oldest trace data is overwritten with the new data after the buffer has become full.

To set a trace until the program is suspended, users can do as following:

- 1. Click [Renesas Views]  $\rightarrow$  [Debug]  $\rightarrow$  [Trace] or icon  $\bowtie$  to open the [Trace] view.
- 2. Turn on the Trace view by selecting the icon.



#### Figure 5-26 Turn on Trace view

3. Execute the program and stop program execution by using a breakpoint or by pressing the [Suspend] button on the Debug Toolbar. The content stored in trace memory at that point in time is displayed as trace result.



4. Select the display mode by clicking on the corresponding button.

The below figure shows the trace result before the main() function is executed.

🖻 main.c	startup_RE01_1500k	⟨B.c ¤ 🖻 r_syste	m_api.c 🖻 r_lpm_api.c	- 8
385	{	-		~
	0005a44 *	pdest = *pso	urce;	
387	}			
388	0005a66 main(	\ ·		
390	<pre></pre>	/ 5		
391	#endif			
392 0	0005a6a } /* End	of function	Reset_Handler() */	
393				~
205		d addtognoup	ann amaid cono.	>
		0 C I D		
🕒 Conso	ole 🤏 Irace 🛛 📶 Problems	Smart Brows	er 🗟 Debugger Console 🛛 Memory 🍣 Call Hierarchy	
Listing fr	ions record 1 of 940		31 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	88 🛃 🗸
	om record 1 of 849			
Record	Label	Address	Source	^
0		5A64		
	startup_RE01_1500KB.c	384	for (uint32_t index = 0U; index < bytes; index++, pdest++, psource++)	
		5A64	bcc.n0x5a44 <reset_handler+6< td=""><td></td></reset_handler+6<>	
1		5A62	Bus Disassembly Source	
	startup_RE01_1500KB.c	384	for (uint32_t index = 0U; index < bytes; index++, pdest++, psource++)	
		5A62	cmpr2, r3	
2		5A60		
	startup RE01 1500KB.c	384	for (uint32 t index = 0U; index < bytes; index++, pdest++, psource++)	
		5A60	ldrr3, [r7, #20]	
3		5A5E	iono, [iii, "Eo]	
0		JAJL		
	startup RE01 1500KB.c	384	for (uint32 t index = 0U; index < bytes; index++, pdest++, psource++)	

Figure 5-27 Select display mode in Trace view

5. The trace records are displayed from oldest data to latest data by default. The display order can be changed by clicking button.

Consc	ole 👒 Trace 🛛 🕄 Problems	s 🏽 Smart Brows	ser 🗟 Debugger Console 🏮 Memory 🍣 Call Hierarchy		
			調 🕨 🖉 😂 😂 🖓 🐘 💷	🐺   🏢 🗸 🖾 🔌	0 🕹 🖻
Listing f	rom record 1 of 849				
Record	Label	Address	Source		~
848		5A34			
	startup_RE01_1500KB.c	376	while (bytes > (uint32_t)0) /* @suppress("Cast comment") */ /	/* @sup	
		5A34	bne.n0x5a20 <reset_handler+32></reset_handler+32>		
847		5A20		Click t	0
	startup_RE01_1500KB.c	378	bytes;	chang	-
		5A20	ldrr3, [r7, #20]	display o	rder
846		5A22			
	startup_RE01_1500KB.c	378	bytes;		
		5A22	subsr3, #1		
845		5A24			
	startup_RE01_1500KB.c	378	bytes;		
	startup_RE01_1500KB.c	378	bytes;		$\sim$

Figure 5-28 The display order is changed

6. The trace result can be filtered by clicking on <sup>→</sup> button. Users can select to filter by "Record" and/or "Address".

🗳 Consc	ole 🛰 Trace 🛛 🕄 Problems			
Listina f	rom record 1 of 849	Þ II 😫 🍳 🐨 日	i 😂   🗢 🕾 🖄 🐺   🇰 🛛 🖗	<mark>ତ Ů</mark> ‰ ≝ ▽
			1	
Record	Label	Address	Source	<u>^</u>
848		5A34		
	startup_RE01_1500KB.c	376	while (bytes > (uint32_t)0	)) /* @
	/	5A34	bne.n0x5a20 <reset_han< th=""><th>ndler+32&gt;</th></reset_han<>	ndler+32>
847	/	5A20		
	e <sup>2</sup> Filter		×	
	Record         Address	Address 0 Range 0 Exclud		
		ОК	Cancel	

Figure 5-29 Filter trace result

7. Trace result can be saved to a .csv file (with the inclusion of bus, assembly and source information). Trace view also allows to load trace results from a .csv file.



Figure 5-30 Save and load trace result

#### 5.4.10 Memory Usage View

Memory view allows users to view the total memory size, usage of ROM and RAM ratio and detailed information of sections, objects, symbols, module, vector and cross-reference used in the project.

To view the memory usage of a project,

- 1. Click [Window] → [Show View] → [Other...] → [Debug] → [Memory Usage] to open the memory usage view.
- 2. The default display of the Memory Usage view for executable project (which uses GNU ARM Embedded Toolchain) has 3 regions: (1) Group size region, (2) Memory Region Usage/ Device Memory Usage region, and (3) Detailed table region.

:e:				Memory Region	Usage Device M	lemory Usage			
			7112 byte(s)						
Program:				Memory Regi	on Usage:				
Constant			0 byte(s)						
Initialized			12 byte(s)	RAM		16 %	2	55KB	
Uninitiali	zed Data:		1620 byte(s)		42KB used				
Data:			0 byte(s)						
Stack:			1024 byte(s)	FLASH		1%	1	536KB	
Others:			40544 byte(s)	1 EAST	6KB used	1.70		550105	
					OVP used				
									(2)
	(1)			OFS		83 %	24	4B	
	(-)				20B used				
				QSPI_FLASH	1	0 %	1:	31072KB	
					0B used				
					00 0000				
				E2S_TRACE_		0 %	11	KB	
					0B used				
				<					
on Object Sy	mbol							1	
tion	Group	Start address (VMA)	End address	Size (byte)	Align	Attribute	Load address (LMA)		
11_veneer	Constant	0x00001BC8		0					
bx	Constant	0x00001BC8		0				(2)	
	Program	0x00000000	0x00001BC7	7112				(3)	
t	Stack	0x2000A8B0	0x2000ACAF	1024					
	SLOCK								
t ck_dummy .dyn	Constant	0x2000A25C		0			0x0000B620		

#### Figure 5-31 Memory Usage view

#### (1) Group size region (for executable project):

Displays the total size of Program, Constants, Initialized Data, Uninitialized Data, Data, Stack and Other according to the selected map file.

Note: This view only displays for executable project of supported toolchains.

#### (2) Memory Region Usage and Device Memory Usage region:

Memory Region Usage shows percentage of RAM/ROM/flash usage by numerical value and status of bar. Color of the bar is based on the percentage value.

- If percentage < 75%: Green.
- If percentage >= 75% and percentage < 90%: Orange.
- If percentage >= 90%: Red.

Device Memory Usage region shows device memory of the selected project's device. Each memory area shows the name, start address, end address, used size and size.

🚍 Console 👒 Trace 🧃 Mem	ory Usage 🗙 📋 Memory					😕 🗞 🗐 🕼 😋 🤹 🖻	
Size:			Memo	ry Region Usag	e Device Memory Usage		
Program:	7112	byte(		0x0000000	InternalRom		1
Constant:		byte(		0x0017FFFF		7112/1572864 byte(s) ( 0.45% )	
Initialized Data:		byte(		0x00180000			
Data:	1620	byte(: byte(:	•		Non-map area		
Stack:	1024	-		0x0100A14E			
Others:	40544	byte(		0x0100A14F	InternalRom		
				0x0100A167		20/25 byte(s) ( 80.00% )	
				0x0100A168	Non-map area		
				0x07FFFFFF			
				0x0800000	InternalRam		
					Internalitam	0/134217728 byte(s) ( 0.00% )	
				0x0FFFFFFF			
				0x10000000	Non-map area		
				0x1FFFFFFF			
				0x20000000	InternalRam		
<		>	L				× 1

Figure 5-32 Device memory usage region

Expand memory area to see section inside. Color of sections corresponds with that of Group Size region.

🛛 Console 👒 Trace 🎁 Merr	nory Usage 🗙 📲 📲 🗛 🗛	1500kb/	Device	/Driver/Src/r_lp	m/r_lpm_api.c		😕 🗞 🗐 🐺 🖓 🕰 🖻
Size:			Memo	ory Region Usag	e Device Memo	ory Usage	
Program:	7112	byte(		0x0000000	InternalRom		
Constant:	0	byte(:		0x0017FFFF		7112/1572864 byte(s)	(0.45%)
Initialized Data:		byte(:		UXUU1/FFFF		• • •	
Uninitialized Data:		byte(:		0x000000	00	.text	7112 byte(s)
Data:		byte(		0x00001E	C8	.glue_7	0 byte(s)
Stack:		byte(:		_		-	
Others:	40544	byte(:		0x00001E	68	.glue_7t	0 byte(s)
				0x00001E	C8	.vfp11_veneer	0 byte(s)
				0x00001E	C8	.v4_bx	0 byte(s)
				0x00001E	C8	.iplt	0 byte(s)
				0x00001E	C8	Unused	1565752 byte(s)
				0x00180000	Non-map area		
				0x0100A14E			
				0x0100A14F	InternalRom		
		>		0x0100A167		20/25 byte(s) ( 80.	00% )

Figure 5-33 Expand memory area

#### (3) Detail table region:

Display the map file information of an active project or the opened map file.

- "Section" tab: contains "Linkage map" table which displays the list of Sections analyzed from map file and its detailed information.
- "Object" tab: contains "Object" table which displays the list of Objects analyzed from map file and its detailed information.
- "Symbol" tab: contains "Symbol" table which displays the list of Symbols analyzed from map file and its detailed information.
- "Cross Reference" tab: displays the cross-reference information that is retrieved from map file. This tab is only available for executable project.

#### Map file location:

Memory Usage will display the information of (\*.map) file or library list file (\*.lbp) from the project. The user can see the relative path of the selected map file or library list file at the bottom of Memory Usage view.



# 6.Help

The help system allows users to browse, search, bookmark and print help documentation from a separate Help window or Help view within the workbench. Users can also access online forum dedicated to e<sup>2</sup> studio from here.

<u>H</u> elp	•		
6	Welcome	(1)	
?	Help Contents	(2)	
2	Search		
	Show Contextual Help	(3)	
	Show Active Keybindings		Ctrl+Shift+L
	Cheat Sheets		
	CMSIS Packs Management		>
R	RenesasRulz Community Forum	(4)	
Ø	Add Renesas Toolchains		
~??	Perform Setup Tasks		
2	Check for Updates		
<b>6</b> -	Install New Software		
	Renesas e2 studio feedback		
a	IAR Embedded Workbench plugin m	anager	
•	About e <sup>2</sup> studio		

Click on [Help] tap to pull down the Help menu.

Figure 6-1 Help Menu

#### Quick Help Tips

- Click [Welcome] for Overview of e<sup>2</sup> studio, link to access IDE tutorial and sample, and to view Release Notes.
- (2) Click [Help Contents] to open a separate Help window with search function.
- (3) Click [Show Contextual Help] to open Help view inside the workbench.
- (4) Click [RenesasRulz Community Forum] to go online forum that is dedicated to topics and discussions related to e<sup>2</sup> studio IDE. Internet connection is required.

Revision History			Renesas e <sup>2</sup> studio 2021-07 or Higher User's Manual : Quick Start Guide				
	Date		Description				
Rev.		Page	Summary				
1.00	Jul.20.2021	-	Initial release.				

Renesas e² studio 2021-07 or higher<br/>User's Manual: Quick Start GuidePublication Date:Rev.1.00Jul 20, 2021Published by:Renesas Electronics Corporation

# Renesas e<sup>2</sup> studio 2021-07 or Higher User's Manual: Quick Start Guide

