

e^2 studio 2020-04 and e^2 studio v7.8

Integrated Development Environment RENESAS MCU RX, RL78, RH850 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 May 2020

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 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.
- 5. 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.

- 6. 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.
- 7. 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 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.
- 8. 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.
- 9. 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.
- It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party. to notify such third party in advance of the contents and conditions set forth in this document.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 Place contact a Renesas Electronics cales office if you have any questions regarding the information contained in this document or Renesas.
- 12. 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.4.0-1 November 2017)

Corporate Headquarters

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

www.renesas.com

Trademarks

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

Contact information

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

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. 0	Seneral	1
1.1 S	system Configuration	1
1.2 C	Derating Environment	1
1.2.1	System Requirements	1
1.2.2	Supported Toolchains	1
1.3 S	Supported Emulator Device	2
1.4 S	Supported Simulator	2
	nstallation	
	nstallation of e ² studio IDE (64-bit version)	
	nstallation of e2 studio IDE (32-bit version)	
	In-installation of e ² studio IDE	
	lpdate e² studio plugins	
2.4.1		
2.4.2		
2.5 lr	nstallation of Compiler Package	18
3. F	Project generation	19
3.1 N	lew Project Generation	19
3.2 N	lew Debug Only Project Generation (RH850)	24
3.3 Ir	nport Existing Projects into Workspace	27
4. E	Build	
4.1 B	uild Option Settings	32
4.2 B	uild A Sample Project	35
4.3 E	xport Build Configuration Settings	
5. C	Debug	
5.1 C	Change Existing Debug Configurations	
5.2 C	Create New Debug Configurations	42
5.3 L	aunch Bar	43
	asic Debugging Features	44
5.4.1		
5.4.2		
	Registers View	
5.4.4	5	
5.4.5		
5.4.6		
5.4.7	1	
5.4.8	0	
5.4.9		
5.4.1	0 Memory Usage View	61
6. ⊦	lelp	65

1. General

Renesas e² studio is the Integrated Development Environment for Renesas embedded microcontrollers. e² 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.

This chapter describes the system configuration and operating environment for e² studio IDE to develop applications for the RX family series microcontrollers as example.

1.1 System Configuration

Below is an example of a typical system configuration.



Figure 1-1 System configuration

1.2 Operating Environment

Below are the system requirements for this product.

1.2.1 System Requirements

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.

Operating Environment:

Windows 8.1 (32/64-bit OS) and Windows 10 (32/64-bit OS). 64-bit OS is required for e² studio 2020-04 or later.

1.2.2 Supported Toolchains

Renesas C/C++ compiler package for RX family Renesas C compiler package for RL78 family GNURX Windows Toolchain (ELF) GNURL78 Windows Toolchain (ELF)

Note1: Two types of packages (with CS+ and without IDE) for each CC-RX and CC-RL are available. Any of those can be applied to e^2 studio.

Note 2: For RH850, the e² studio can be used to debug load modules in the ELF/DWARF format which were built with the IAR Embedded Workbench from IAR Systems or the MULTI IDE from Green Hills Software.



1.3 Supported Emulator Device

E2 emulator Lite (RX, RL78) E1 (RX, RL78, RH850) E2 (RX, RL78, RH850) E20 (RX)

1.4 Supported Simulator

Renesas Simulator (RX, RL78) GDB Simulator (RH850)



2. Installation

The latest e² studio IDE installer package can be downloaded from Renesas website for free. Renesas now supports both 32-bit and 64-bit versions of e² studio.

Please check detailed information from <u>https://www.renesas.com/e2studio</u>. Note that user has to login to the Renesas account (in MyRenesas page) for the software download.

This chapter describes the installation and un-installation for the e² studio IDE.

 e^2 studio installer can be used to upgrade e^2 studio as well as new installation. However, it does not support update between major versions such as from v5.x to v6.y, or from v6.m to v7.n.

Please uninstall the earlier versions before installation. Alternatively, install new e² studio into a new folder if you would like to keep earlier versions.

The detailed information is described below.

2.1 Installation of e² studio IDE (64-bit version)

1. Double-click on e² studio installer to invoke the e² studio installation wizard page. Click [Install].

Note: If e^2 studio was installed in your PC, the option to modify, remove the existing version or install e^2 studio to a different location will be displayed.



Figure 2-1 64-bit e² studio installation wizard

2. Welcome page

The install folder can be changed by clicking [Change...]. Click [Next] to continue.

Note1: If you would like to have multiple versions of e^2 studio, please specify new folder here. **Note2:** Multi-byte characters cannot be used for e^2 studio installation folder name.





Figure 2-2 Installation of 64-bit e² studio – Welcome page

3. Device Families

Select Devices Families to install. Click [Next] to continue.

Welcome	RENESAS
Device Families	Build, Debug & Code Generation support for Renesas RA devices
	REMESAS RZ Build, Debug & Code Generation support for Renesas RZ devices
Customise Features	RENESAS RL78 RL78 Build, Debug & Code Generation support for Renesas RL78 devices
Additional Software	RENESAS RX RX Build, Debug & Code Generation support for Renesas RX devices
Licenses	RENESAS RH850
Shortcuts	RHB50 Debug support for Renesas RH850 devices
Summary	RE Duild & Dubug support for Densers DE devices
Installing	Build & Debug support for Renesas RE devices
Results	
Selec	t All

4. Extra Features

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

Click [Next] to continue.



Figure 2-4 Installation of 64-bit e² studio – Extra Features

5. Customise Features

Select the components to install and click the [Next] button to continue.

Renesas e ² studio 203 Renesas e ² studio 20	
Kenesas e studio 20	
Welcome	Select the components you want to install.
	✓ Renesas e2 studio (20.4.0.R20200412-1149)
Device Families	⇔ 🖉 Java Runtime (11.0.0)
Extra Features	OpenJDK & OpenJFX
Customise	⊕ 🖉 Renesas e2 studio Common Components (20.4.0.R20200412-1149)
Features	Common components for Renesas e2 studio
Additional Software	 ♦ I Renesas e2 studio Tools (20.4.0.R20200412-1149)
	Renesas e2 studio Tools
Licenses	⊕ □ Renesas RA Family Support (20.4.0.R20200412-1149)
Shortcuts	Renesas RA family support to allow project generation, build & debug
Summary	© 🗌 Renesas RZ Family Support (20.4.0.R20200412-1149)
Installing	Renesas RZ family support to allow project generation and build of executable projects, library projects, debug only projects and IAR projects.
Results	Car Projects.
Results	Renesas RZ family support requiring .Net Framework
	↔ 🖌 Renesas RL78 Family Support (20.4.0.R20200412-1149)
	Select All Optional Deselect All Optional
	Size of install: 904.7 MB
	Temporary space required: 1.6
	Available: 44.7 GB
<u>v202004121319</u>	< <u>B</u> ack <u>N</u> ext > <u>Install</u> Cancel

Renesas e² studio

6. Additional Software

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 software catalog cannot be downloaded. The additional software can be installed later.

Renesas e² studio 202	20-04 Setup	RENESAS
Select the additional sof	tware you wish to install	
Welcome	Renesas QE (0) Renesas Toolchains & Utilities (1) GCC Toolchains & U	tilities (0)
Device Families	🕫 Renesas CC-RX Compilers	*
Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing Results	 Renesas CCRX v2.08.01 v2.08.01 Renesas C/C++ Compiler Package for RX Family v2.08.01 Download size: 22.4 MB Requires: Renesas Tool License Manager - 2.03.01 	
		29.9 MB download required

Figure 2-6 Installation of 64-bit e² studio – Additional Software

7. License Agreement

Read and accept the software license agreement. Click the [Next] button. Please note that user must accept the license agreement, otherwise installation cannot be continued.

Welcome Please read and accept the following Software Agreements Device Families Renesas e2 studio Extra Features OpenJDK License Agreement Customise Features Additional Software Software Extra Features Customise Comparison Features Comparison Additional Comparison Software Constitutes Customse Comparison Software Constitutes
Extra Features OpenJDK License Agreement Customise Features Features Additional Software OpenStructures OpenStructures Image: Software OpenStructures Constructures Additional Constructures Software Constructions OpenStructures The Renesas IDE Software (defined below) is intended for commercial
Shortcuts 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 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. Results Installing Installing Otherwise, by clicking the "I accept" button or other button or Installing Print a

8. Shortcuts

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

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

Renesas e ² studio 2020)-04 Setup				×
Renesas e² studio 202	0-04 Setup		REN	IES/	15
Welcome	Shortcuts to important pro	ograms and files will be	created in the	e following	
Device Families	locations:				
Extra Features	🗹 In start menu group:	Renesas Electronics e2	studio		
Customise	5 1)efault
Features			L	- <u>H</u> ostore s	ordane
Additional Software					
Licenses					
→ Shortcuts					
Summary					
Installing					
Results					
<u>v202004121319</u>	< <u>B</u> ack	<u>N</u> ext > _ins	stall	Cancel	

Figure 2-8 Installation of 64-bit e² studio – Shortcuts

9. Summary

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 64-bit e² studio – Summary

10. Installing...

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

11. Results

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

Click [OK] button to complete the installation.

Renesas e² studio 2020	-04 Setup	- 🗆 X
Renesas e² studio 2020)-04 Setup	RENESAS
Welcome Device Families Extra Features Customise Features Additional Software Licenses Shortcuts Summary Installing	Installation of e2 studio is complete. Please click OK to close. Launch e2 studio? View Release Notes? View What's New?	
<u>v202004121319</u>	< <u>B</u> ack <u>N</u> ext >	OK Cancel

Figure 2-10 Installation of 64-bit e² studio – Results



2.2 Installation of e² studio IDE (32-bit version)

1. Double-click on e² studio installer to invoke the e² studio installation wizard page. Click [Install].

Note: If e^2 studio was installed in your PC, the option to modify, remove the existing version or install e^2 studio to a different location will be displayed. If you would like to have multiple versions of e^2 studio, please specify "Install to a different location" here. Click [Next] to continue.



Figure 2-11 32-bit e² studio installation wizard

2. Welcome page

Click [Next] to continue.

🛃 e² studio Setup		— 🗆 X
e² studio 7.8.0.R2020	0321-2326 Setup	RENESAS
Welcome Install Folder	Welcome to the e² studio installation wizard. This with installation process.	wizard will guide you through the
O Device Families	Click Next to proceed to the next page.	
○ Extra Components	Click Back to change something on a previous pag	10
○ Components	ciek back to change something on a previous pag	JC.
O Additional Software	Click Cancel to cancel this installation at any time.	
○ Licenses		
○ Shortcuts		
O Summary		
O Installing		
○ Results		
		Configure Proxy Settings
<u>v202003220126</u>	< <u>B</u> ack <u>N</u> ext >	Install Cancel

Figure 2-12 Installation of 32-bit e² studio – Welcome page

3. Install Folder

The default installation location is set to: "C:\Renesas\e2_studio". Enter folder name directly in the textbox or click [Browse...] button to modify it. Click [Next] to continue.

Note1: If you would like to have multiple versions of e^2 studio, please specify new folder here. **Note2:** Multi-byte characters cannot be used for e^2 studio installation folder name.

🛃 e² studio Setup		$ \square$ \times
e² studio 7.8.0.R2020	0321-2326 Setup	RENESAS
Welcome	Where would you like to install?	
🗘 Install Folder	C:\Renesas\e2_studio	
O Device Families		Bestore Default □ Browse Browse Browse
○ Extra Components		
○ Components		
O Additional Software		
O Licenses		
○ Shortcuts		
O Summary		
O Installing		
○ Results		
<u>v202003220126</u>	< <u>B</u> ack <u>N</u> ext >	Install Cancel

Figure 2-13 Installation of 32-bit e² studio – Install Folder

4. Device Families

Select Devices Families to install. Click [Next] to continue.

🛃 e² studio Setup	— 🗆 X
e ² studio 7.8.0.R20200321-2326 Setup Select the device families you wish to install components for	RENESAS
Install Folder	bug & Code Generation
Device Families Support for RZ Device Families	evices bug & Code Generation
O Extra Components O Components Support for RL78 Includes Build, De	Devices bug & Code Generation
O Additional Software O Licenses Support for RX D Includes Build, De	evices bug & Code Generation
• Shortcuts Support for RH85 RH850 Includes Debug	0 Devices
O Summary O Installing ■ Renesas Synergy™ Support for Syner Includes Build, De	gy Devices bug & Code Generation
Results Results Support for Renew Includes Build & E	
CENESAS Support for RZ/G	
Select All	
<u>v202003220126</u> < <u>B</u> ack	<u>N</u> ext > <u>I</u> nstall Cancel

5. Extra Components

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

Click the [Next] button to continue.

 e² studio Setup e² studio 7.8.0.R2020 	00321-2326 Setup	
 Welcome Install Folder 		apanese Language Pack rovides Japanese language support
 Device Families 		hinese (Simplified) Language Pack rovides Simplified Chinese language support
Components		hinese (Traditional) Language Pack rovides Traditional Chinese language support
 Additional Software Licenses 		ollaboration ncludes SVN & Git support
 Shortcuts 		TOS reeRTOS & OpenRTOS Debug Support
○ Summary ○ Installing		erminals NSI/vt102 compatible Terminal support for Serial, ssh and
○ Results	Select All Deselect	All
<u>v202003220126</u>	< <u>B</u> ack	Next > Install Cancel

Figure 2-15 Installation of 32-bit e² studio – Extra Components

6. Components

Select the components to install and click the [Next] button to continue.

e ² studio 7.8.0.R20200321-2326 Setup
 Install Folder Install Folder Device Families Extra Components Renesas e2 studio Common Core 6.12.0.v20200305-1438 Renesas e2 studio Components GCC for Renesas RL78 Build Support 7.8.0.v20200313-0148 GCC for Renesas RL78 Build Support 7.8.0.v20200310-0338 Renesas CCRL Build Support 7.8.0.v20200219-0123 Renesas RL78 Debug Support 7.8.0.v20200219-0123 Renesas RL78 Debug Support 7.8.0.v20200313-1019 Renesas e2 studio Common Build for GCC Toolchains 6.8.2.v20200312-1245 Renesas e2 studio Common Build for GCC Toolchains 6.8.2.v20200312-1245 Renesas e2 studio Common Build for Renesas Toolchains 6.11.0.v20200221-1009 Renesas e2 studio Common Build for Renesas Toolchains
Renesas e2 studio Memory Usage View Renesas e2 studio Stack Analysis View 6.9.0.v20200305-0646 Renesas e2 studio Stack Analysis View
Select All Optional Deselect All Optional Around 942.8 MB required

Figure 2-16 Installation of 32-bit e² studio – Components

7. Additional Software

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



Figure 2-17 Installation of 32-bit e² studio – Additional Software

8. License Agreement, Shortcuts, Summary, Installing... and Results

The steps for these pages are the same as installation of e^2 studio 64-bit version.

Please refer to Chapter 2.1, step 7 onwards.



2.3 Un-installation of e² studio IDE

User can uninstall e² studio program following the typical steps to uninstall a program in Window OS.

- 1. Search for [Apps & features] in Window Search Box. Click on the search result to go to [Apps & features].
- 2. From the currently installed programs list, choose "e² studio" and click the [Uninstall] button.
- 3. Click the [Uninstall] button again to confirm the deletion of e^2 studio.

At the end of the un-installation, e² studio IDE will be deleted from the installed location and Windows shortcuts menu are removed.

Note: If you have installed e^2 studio at multiple locations, you may not find the uninstaller in "Apps & features" of Control Panel. In such cases, launch e^2 studio uninstaller located at: { e^2 studio installed folder}/uninstall/uninstall.exe.



2.4 Update e² studio plugins

Following procedure describes how to update existing plugins, **not to install new features**. Please use e² studio installer to upgrade to newer versions.

2.4.1 Online Minor Version Update

This section illustrates an example on the steps to launch the online minor version update

1. From the [Help] menu, click the [Check for Updates] to display the [Available update] pane.



Figure 2-18 [Check for Updates] Menu

2. By default, all the software components are selected in the [Available Updates] panel. This allows user to update all to the latest version. (An example is shown in Figure 2-2). Click the [Next] button to proceed.

Available Updates			
Check the updates that you wish to install.			
Name	Version	ld	^
🗹 🖗 GCC for Renesas RL78 Build Support	6.1.0.v20171011-0	com.renesas.e2studio.device.rl78.build.g	
🗹 🖗 GCC for Renesas RX Build Support	6.1.0.v20171010-0	com.renesas.e2studio.device.rx.build.gcc	
🗹 🖗 GCC for Renesas RX Support Files	1.1.0.v20171010-0	com.renesas.ide.supportfiles.rx.gcc.featu	
🗹 🖗 GCC for Renesas RZ Build Support	6.1.0.v20171011-1	com.renesas.e2studio.device.rz.build.gcc	
🗹 🖗 Renesas CCRL Build Support	6.1.0.v20170927-0	com.renesas.e2studio.device.rl78.build.re	
🗹 🖗 Renesas CCRL CS+ Import/Export	6.1.0.v20170927-0	com.renesas.e2studio.device.rl78.build.re	
🗹 🖗 Renesas CCRL Support Files	1.0.1.v20170912-0	com.renesas.ide.supportfiles.rl78.ccrl.feat	
🗹 🖗 Renesas CCRX Build Support	6.1.0.v20170913-1	com.renesas.e2studio.device.rx.build.ren	
Renesas CCRX CS+ Import/Export	6.1.0.v20170913-1	com.renesas.e2studio.device.rx.build.ren	
🗹 🖗 Renesas CCRX HEW Import	6.1.0.v20170913-1	com.renesas.e2studio.device.rx.build.ren	
🗹 🖗 Renesas CCRX Support Files	1.0.1.v20170912-0	com.renesas.ide.supportfiles.rx.ccrx.featu	~
Select All	C 1 0 2017022 0	2. 1. 1	
Details			
			0
			~

Figure 2-19 e² studio – Available Updates panel (1/3)

3. Select the [Next] button to continue the update.

e ² Available Updates			— 🗆 🗙
Available Updates			
Check the updates that you wish to install.			
Name	Version	Id	^
GCC for Renesas RL78 Build Support	6.1.0.v20171011-0	com.renesas.e2studio.device.rl78.build.g	
GCC for Renesas RX Build Support	6.1.0.v20171010-0	com.renesas.e2studio.device.rx.build.gcc	
GCC for Renesas RX Support Files	1.1.0.v20171010-0	com.renesas.ide.supportfiles.rx.gcc.featu	
GCC for Renesas RZ Build Support	6.1.0.v20171011-1	com.renesas.e2studio.device.rz.build.gcc	
Renesas CCRL Build Support	6.1.0.v20170927-0	com.renesas.e2studio.device.rl78.build.re	
Renesas CCRL CS+ Import/Export	6.1.0.v20170927-0	com.renesas.e2studio.device.rl78.build.re	
🗹 🖗 Renesas CCRL Support Files	1.0.1.v20170912-0	com.renesas.ide.supportfiles.rl78.ccrl.feat	
🗹 🖗 Renesas CCRX Build Support	6.1.0.v20170913-1	com.renesas.e2studio.device.rx.build.ren	
Renesas CCRX CS+ Import/Export	6.1.0.v20170913-1	com.renesas.e2studio.device.rx.build.ren	
🗹 🏇 Renesas CCRX HEW Import	6.1.0.v20170913-1	com.renesas.e2studio.device.rx.build.ren	
🗹 🏇 Renesas CCRX Support Files	1.0.1.v20170912-0	com.renesas.ide.supportfiles.rx.ccrx.featu	
	6 1 0 20170020 0	A . P . I	•
Select All Deselect All			
Details			
			·
			\checkmark
(?)	Γ	< Back Next > Finish	Cancel
\odot		- Dock Hickey Emist	Concer

Figure 2-20 e² studio – Available Updates panel (2/3)

4. Read and check the software license agreement. Click the [Finish] button to complete update.

icenses:	License text:
A. HISTORY OF THE SOFTWARE Eclipse Foundation Software User Agreement license	Eclipse Foundation Software User Agreement April 9, 2014
License for RENESAS e2 studio	Usage Of Content
> Renesas Pin Configurator	THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU
	I do not accept the terms of the license agreements

Figure 2-21 e² studio – Available Updates panel (3/3)

5. Click the menu [Help] \rightarrow [About e² studio] \rightarrow [Installation Details] to confirm the updated version.

About e ² studio		\times	
Renesas e ² studio			
e ² Version: 2020-04 (20.4.0) Build Id: R20200319-2356			
Parts Copyright (C) 2010-2020 Renesas Electronics Corp. All rights reserved.			
e ² studio IDE is an extension of software developed for eclipse.org.			
e ² studio IDE is based on Eclipse Platform 4.14 (2019-12) and CDT version 9.10.0.			
Source code for the Eclipse Foundation plug-ins is available from www.eclipse.org, under the Eclipse Public License "EPL", see https://www.eclipse.org/org/documents/epl-2.0/EPL-2.0.html and https://www.eclipse.org/org/documents/epl-v10.html			
C C RX C ² R-70 O RZ RE 💉 💮 🚥			
Installation Details	<u>C</u> lose		

Figure 2-22 e² studio – About e² studio panel



2.4.2 Offline Minor Version Update

This section illustrates how to update e² studio with the upgrade function of the installer.

1. Download the desired new version of e² studio offline installer from the following Renesas URL: <u>http://www.renesas.com/e2studio_download</u>

Note: Offline version update using 'Differential Update program' is available with e² studio Ver3.x or older versions.

 Double-click to run the installer file downloaded in step (1). The installer will detect existing version and user can choose to upgrade or install new e² studio version to a different folder. Please choose any of the following options:

[Upgrade] - Minor version upgrade of e² studio [Modify] - Add plugins into the same version of e² studio [Delete] - Uninstall e² studio [Install] - Install e² studio to a new location

Then click [Next] to proceed.

尾 e² studio Set	up	- 🗆 X
e² studio 7.8.0	.R20200316-1101 Setup	RENESAS
e2 studio versior	n 7.7.0.R20191216-1445 is already installe	ed.
	What do you want to do?	
	Upgrade Upgrade to version 7.8.0.F	R20200316-1101.
	Install Install Install Install Install Install	on.
<u>v20200316</u>	<u>1240</u> < <u>B</u> ack <u>N</u> ext >	<u>Install</u> Cancel

Figure 2-23 Upgrade e² studio from offline installer

Note: Please do not apply "Upgrade" or "Modify" if major versions are different between installer and the installed e² studio.

3. Follow the steps shown in Section 2.1 or 2.2 to install e² studio IDE. Step (3) Install Folder is skipped since Upgrade option will use the same destination folder as existing e² studio.



2.5 Installation of Compiler Package

 e^2 studio installer could install compiler packages automatically during e^2 studio installation with valid Internet connection. However, in situation where Internet connection is not available during e^2 studio installation, compiler packages can be installed later from compiler package installation files from the web site shown below.

Compiler packages are available at each distribution sites. Please follow the installation guides at the download pages.

Renesas Compiler Package download sites:

For RX Family: <u>http://www.renesas.com/rx_c</u>

For RL78 Family: http://www.renesas.com/rl78_c

GNU Toolchain download site: https://gcc-renesas.com/

To check for compilers already installed, click \swarrow from the toolbar or click [Help] \rightarrow [Add Renesas Toolchains] to open Renesas Toolchain Management as shown below. Check the desired toolchain to integrate it in e² studio.

If desired compiler is not listed, click [Add...] and specify the installed location.

Smart C KPIT GNUARM-I v16.01 v14.02 Support v14.02 v1.09.00 v1.09.00 v1.05.00 v1.04.00 v1.04.00 v1.04.00 v1.04.00 v3.02.00 v3.01.00 v3.00.00 KPOVE KINDER VS.00 VS.00 	004 C:\Program Files (x86)\GNU Tools ARM Embedded\7 2017-q4-major\ 020 C:\Program Files (x86)\GNU Tools ARM Embedded\6 2017-q2-update\ 021 C:\Program Files (x86)\GNU Tools ARM Embedded\5 4 2016q3\ 022 C:\Program Files (x86)\GNU Tools ARM Embedded\5 4 2016q3\ 023 C:\Program Files (x86)\GNU Tools ARM Embedded\5.4 2016q3\ 029 C:\Program Files (x86)\GNU Tools ARM Embedded\4.9 2015q3\
Cogging Image: Gashing and the second se	S20 C:\Program Files (x86)\GNU Tools ARM Embedded\6 2017-q2-update\ V19 C:\Program Files (x86)\GNU Tools ARM Embedded\5.4 2016q3\ S29 C:\Program Files (x86)\GNU Tools ARM Embedded\4.9 2015q3\ NONE-EABI Toolchain C:\Program Files (x86)\KPIT\GNUARM-NONEv16.01-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\KPIT\GNUARM-NONEv16.01-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\KPIT\GNUARM-NONEv14.02-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\Renesas\RL78\1_9_0\ C:\Program Files (x86)\Renesas\RL78\1_9_0\
Wy Rene G.3.1.201706 Wenesas S.4.1.201609 S.4.1.201609 S.4.1.201609 S.4.1.201609 V.4.9.3.201505 Smart C W KPIT GNUARM-I Support V v16.01 V v16.01 V v14.02 Fracealy. W Renesas CCRL FraceX V v1.09.00 V v1.05.00 vplate D W Renesas CCRX Vpdate W v3.02.00 V v3.00.00 W v3.00.00 W KPIT GNURL78-I 	D19 C:\Program Files (x86)\GNU Tools ARM Embedded\5.4 2016q3\ 529 C:\Program Files (x86)\GNU Tools ARM Embedded\4.9 2015q3\ NONE-EABI Toolchain C:\Program Files (x86)\KPIT\GNUARM-NONEv16.01-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\KPIT\GNUARM-NONEv14.02-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\Renesas\RL78\1_9_0\
kenesas 5.4.1.201609 4.9.3.201505 imart Q 4.9.3.201505 imart Q v16.01 v16.01 v14.02 vacealy. Renesas CCRL v1.09.00 tags v1.09.00 v1.05.00 v1.04.00 v1.04.00 v3.02.00 v3.01.00 v3.00.00 v3.00.00 v KPIT GNURL78-1 	i29 C:\Program Files (x86)\GNU Tools ARM Embedded\4.9 2015q3\ NONE-EABI Toolchain C:\Program Files (x86)\KPIT\GNUARM-NONEv16.01-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\KPIT\GNUARM-NONEv14.02-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\Renesas\RL78\1_9_0\
Smart C KPIT GNUARM-I Smart V v v16.01 v v14.02 Support v v16.01 v v14.02 Tracealy. Renesas CCRL rt.05.00 v v1.05.00 v v1.04.00 v Renesas CCRX /Update v v3.02.00 v v3.01.00 v S.00.00 / Hover KPIT GNURL78-I 	NONE-EABI Toolchain C:\Program Files (x86)\KPIT\GNUARM-NONEv16.01-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\KPIT\GNUARM-NONEv14.02-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\Renesas\RL78\1_9_0\ C:\Program Files (x86)\Renesas\RL78\1_5_0\ C:\Program Files (x86)\Renesas\RL78\1_5_0\ C:\Program Files (x86)\Renesas\RL78\1_4_0\ C:\Program Files (x86)\Renesas\RX\3_2_0\ C:\Program Files (x86)\Renesas\RX\3_1_0\
Infant C Image: Viscon vis	C:\Program Files (x86)\KPIT\GNUARM-NONEv16.01-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\KPIT\GNUARM-NONEv14.02-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\Renesas\RL78\1_9_0\ C:\Program Files (x86)\Renesas\RL78\1_5_0\ C:\Program Files (x86)\Renesas\RL78\1_4_0\ C:\Program Files (x86)\Renesas\RX\3_2_0\ C:\Program Files (x86)\Renesas\RX\3_2_0\
inter tw ↓ ∨ 14.02 inter two ↓ v14.02 v14.02 inter two ↓ Renesas CCRL v1.09.00 inter two ↓ v1.09.00 v1.09.00 inter two ↓ Renesas CCRX v3.02.00 inter two ↓ v3.01.00 ↓ v3.00.00 inter two ↓ KPIT GNURL78-I ↓ KPIT GNURL78-I	C:\Program Files (x86)\KPIT\GNUARM-NONEv14.02-EABI\arm-none-eabi\arm-none-eabi\ C:\Program Files (x86)\Renesas\RL78\1_9_0\ C:\Program Files (x86)\Renesas\RL78\1_5_0\ C:\Program Files (x86)\Renesas\RL78\1_4_0\ C:\Program Files (x86)\Renesas\RX\3_2_0\ C:\Program Files (x86)\Renesas\RX\3_1_0\
Image: specific decision Image:	C:\Program Files (x86)\Renesas\RL78\1_9_0\ C:\Program Files (x86)\Renesas\RL78\1_5_0\ C:\Program Files (x86)\Renesas\RL78\1_4_0\ C:\Program Files (x86)\Renesas\RX\3_2_0\ C:\Program Files (x86)\Renesas\RX\3_1_0\
Image: Provide state Image: Provide state Image: Provide state Image: Provide state Image: Provide state Image: Provide state Image: Provide state Image: Provide state Image: Provide state Image: Provide state Image: Provide state Image: Provide state Image: Provide state Image: Providest Image: Providest <	C:\Program Files (x86)\Renesas\RL78\1_5_0\ C:\Program Files (x86)\Renesas\RL78\1_4_0\ C:\Program Files (x86)\Renesas\RX\3_2_0\ C:\Program Files (x86)\Renesas\RX\3_1_0\
Image: Second secon	C:\Program Files (x86)\Renesas\RL78\1_5_0\ C:\Program Files (x86)\Renesas\RL78\1_4_0\ C:\Program Files (x86)\Renesas\RX\3_2_0\ C:\Program Files (x86)\Renesas\RX\3_1_0\
gs ↓ v1.04.00 te D ↓ v1.04.00 ✓ Renesas CCRX ↓ v3.02.00 ↓ v3.01.00 ↓ v3.00.00 ↓ v3.00.00 ↓ v3.00.00 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	C:\Program Files (x86)\Renesas\RL78\1_4_0\ C:\Program Files (x86)\Renesas\RX\3_2_0\ C:\Program Files (x86)\Renesas\RX\3_1_0\
Image: state D Image: wide wide wide wide wide wide wide wide	C:\Program Files (x86)\Renesas\RX\3_2_0\ C:\Program Files (x86)\Renesas\RX\3_1_0\
te	C:\Program Files (x86)\Renesas\RX\3_1_0\
ver v3.01.00 ver v3.00.00 t ∨ V KPIT GNURL78-1	C:\Program Files (x86)\Renesas\RX\3_1_0\
ver v3.01.00 v3.00.00 v3.00.00 v ✓ KPIT GNURL78-	
ot Ver KPIT GNURL78-I	C:\Brogram Files (x86)\Benesas\BX\2.0.0\
✓ ✓ KPIT GNURL78-I	
	ELF Toolchain
✓ v14.03	C:\Program Files (x86)\KPIT\GNURL78v14.03-ELF\rl78-elf\
✓ ✓ GCC for Renesas	s RX
vel 🗸 8.3.0.201904	C:\Program Files (x86)\GCC for Renesas RX 8.3.0.201904-GNURX-ELF\rx-elf\rx-elf
✓ 4.8.4.201902	C:\Program Files (x86)\GCC for Renesas RX 4.8.4.201902-SP1-GNURX-ELF\rx-elf\rx-elf
✓ 4.8.4.201803	C:\Program Files (x86)\GCC for Renesas RX 4.8.4.201803-GNURX-ELF\rx-elf\rx-elf
✓ ✓ KPIT GNURX-EL	F Toolchain
✓ v14.03	C:\Program Files (x86)\KPIT\GNURXv14.03-ELF\rx-elf\rx-elf\
✓ ✓ GCC for Renesas	s RL78
4.9.2.201801	C:\Program Files (x86)\GCC for Renesas RL78 4.9.2.201801-GNURL78-ELF\rl78-elf\rl78-elf
~	Scan Add Remove
>	

Figure 2-24 Toolchain management



3. Project generation

In e² studio, "Project" is a basic unit to perform build and debug operations.

This chapter describes the creation of new project and import of existing e² studio project, High-performance Embedded Workshop IDE (described as "HEW" below) project and CS+ project to e² studio IDE.

Note: 1. To install and use the e² studio on your PC, you must install the compiler package provided separately.

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

3.1 New Project Generation

To create a new project, invoke e² studio IDE from the Windows ([Start] menu) and specify a workspace directory.

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

File	Edit	Source	Refactor	Navigate	Search	Project	Renes	as Vie	ews Ru	n Windo	w	Help	
	New					Alt+Shif	t+N>	C	RZ Linu	x C/C++	oroj	ect	
	Open	File						C.	Makefi	e Project v	vith	Existing C	ode
_	Open	Projects f	rom File Sy	stem				C	C/C++	Project			
	Close					Ctr	I+W	D	Project				



 Select template for the new project (For e.g., Renesas RX: "Renesas CC-RX C/C++ Executable Project"). If the target device family or the toolchain were not listed, you may need to run e² studio installer to add "Build/Debug support plugins" for the target device family. Click [Next] to proceed.

All Make Renesas Debug Renesas RA Renesas RE Renesas RL78 Renesas RX	 GCC for Renesas RX C/C++ Executable Project A C/C++ Executable Project for Renesas RX using the GCC for Renesas RX Toolchain. GCC for Renesas RX C/C++ Library Project A C/C++ Library Project for Renesas RX using the GCC for Renesas RX Toolchain.
Renesas RZ Renesas Synergy	Renesas CC-RX C/C++ Executable Project
	Renesas CC-RX C/C++ Library Project

Figure 3-2 New Project Creation Wizard (1/6)



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

New Renesas CC-RX Executab New Renesas CC-RX Executable	-	Ď
Project name: Tutorial		
Use <u>d</u> efault location		
Location: D:\workspa	ace\Tutorial	B <u>r</u> owse
Create Di Choose file system: default ~ Working sets	irectory for Project	
Add projec <u>t</u> to working sets		Ne <u>w</u>
Working sets:		✓ S <u>e</u> lect
?	< <u>B</u> ack <u>N</u> ext >	<u>F</u> inish Cancel

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

4. Select Language, Toolchain, Toolchain Version, RTOS, RTOS Version, Target Device and Configurations. Click [Next] to proceed.

Note: "E1" or "E2" can be selected in the same way as E2 Lite in the Hardware Debug Configuration pull down menu.

Toolchain Settings				
	● C ○ C++			
Toolchain:	Renesas CCRX	~		
Toolchain Version:	v3.02.00	~		
	Manage	Toolchains		
RTOS:	None	\sim		
RTOS Version:		~		
Device Settings			Configurations	
Target Device: R5	F564MLCxFC		✓ Create Hardware Debug Co	nfiguration
	<u>Unlo</u>	ck Devices	E2 Lite (RX)	~
Endian: Litt	le	~	Create Debug Configuration	1
Project Type: De	fault	\sim	RX Simulator	~
			Create Release Configuratio	n

Figure 3-4 New Project Creation Wizard (3/6)

5. Coding Assistant feature can be applied if necessary. Click [Next] to proceed.

Note:

- *Peripheral Code Generator* (CG) supports the generation of driver and peripheral function code based on GUI settings. Functions are provided as APIs and are not limited to initialization of peripheral function.
- *FIT* provides drivers and codes higher layer than CG, such as communication protocol stack and sample application programs using peripheral functions. All FIT modules are interchangeable because they are implemented with common interfaces.
- Smart Configurator supports a single user interface that combines the functionalities of Code Generator and FIT Configurator. Smart Configurator encompasses unified clock configuration view, interrupt configuration view and pin configuration view.

Peripheral Code Generator and Smart Configurator may not be available for some devices.

New Renesas CC-RX Executable Project Image: Control of the settings Select Coding Assistant settings Image: Control of the settings				
Smart Configurator Use Peripheral Code Gener Use FIT Module	ator Download FIT Modules			
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel		

Figure 3-5 New Project Creation Wizard (4/6)

6. Keep the "Use Renesas Debug Virtual Console" unchecked and click [Next] to proceed.

	As CC-RX Executable Project	
What kind of	initialization routine would you like to create?	
	sas Debug Virtual Console ream Buffer:	
(?)	< <u>B</u> ack <u>Next</u> > <u>F</u> inish Cancel	

Figure 3-6 New Project Creation Wizard (5/6)



7. A project summary is displayed. Click [Finish] to generate the project.

New Renesas CC-RX Ex Summary of project "Tuto	-				Ź
TOOLCHAIN NAME : TOOLCHAIN VERSION : GENERATION FILES :	Renesas CCRX v3.02.00				^
generate\resetprg.c generate\stacksct.h generate\dbsct.c generate\typedefine.h generate\vecttbl.c					L
generate\intprg.c generate\vect.h generate\iodefine.h generate\hwsetup.c generate\sbrk.h generate\sbrk.c					~
?	< <u>B</u> ack	<u>N</u> ext >	<u> </u>	Cancel	

Figure 3-7 New Project Creation Wizard (6/6)







Figure 3-8 New C Project Created

This project consists of an application file "Tutorial.c" and standard start-up files (e.g. "dbsct.c", "intprg.c", "sbrk.c" etc). All these project and source files listed in the [Project Explorer] panel reflect the folder structure of the project, just as seen on the standard file explorer.

Notes for backing up projects:

- Project properties are stored in files or folders which filenames or folder names are prefixed with a '.' (dot), for example ".project" and ".cproject". It is necessary to include these files or folders when archiving the project for back-up purpose.
- In order to restore properties shared among projects, for instance when one project makes reference to another project's files, please backup the whole workspace folder.



3.2 New Debug Only Project Generation (RH850)

Creating a debug only project allows user to debug an existing executable which user has already built. This feature will automatically create a project and debug configuration for user.

Note: Only the debugging functions of the e² studio can be used with products of the RH850 family. The e² studio can be used to debug load modules in the ELF/DWARF format which were built with the IAR Embedded Workbench from IAR Systems or the MULTI IDE from Green Hills Software.

To create a debug only project (RH850 family is supported),

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

File	Edit	Source	Refactor	Navigate	Search	Project	Renesa	as Vie	ws Run	Window	Help	
	New					Alt+Shif	't+N>	C	RZ Linux (C/C++ proj	ect	
	Open	File						C#	Makefile F	Project with	Existing C	ode
	Open	Projects f	rom File Sy	stem				C ⁰	C/C++ Pr	oject		
	Close					Ctr	l+W	Ľ	Project			



 Select template for the new project: [Renesas Debug] → "Renesas Debug Only Project". Click [Next] to proceed.

Templates for New C/C++ Project	
All CMake Make Renesas Debug Renesas RE Renesas RL78 Renesas RX Renesas RZ	
Image: Concel Image: Concel	

Figure 3-10 Specify the project template

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



Debug Only Create a Debu	a Only project		ightharpoonup
	DebugOnly_Tutorial		
└ Use <u>d</u> efault			
Location:	C:\Renesas\Workspace\GSG\DebugOnly_Tutorial		B <u>r</u> owse
	✓ Create Directory for Project		
Choose file s <u>y</u> s	tem: default \vee		
Working sets			
Add projec	t to working sets		Ne <u>w</u>
Working sets:		~	S <u>e</u> lect
?	< <u>B</u> ack <u>N</u> ext > Einis	h	Cancel

Figure 3-11 Specify project name

4. Select debug hardware (e.g. "E2 (RH850)"), target device (e.g. R7F701002xAFP). Note that these settings should be consistent with the settings to build the executable file. Then specify the location of prebuilt executable file (i.e. executable file built in other IDEs) which should be built as ELF/DWARF format to be recognized by debugger. Click [Finish] to create the project.

Debug Hardware • Hardware		Device Settings	75704000 450	
E2 (RH850)		Target Device: R		nlock Devices
○ Simulator		Endian: Lit	tle	
Device Executable				
	\Renesas\Workspace\CS	5_plus\Sample\DefaultBuil	d\Sample.abs	

Figure 3-12 Specify project settings

5. The project named "DebugOnly_Tutorial" is created. User can only modify the debug configuration of this project and start the debugging.

Renesas e² studio

<u>File Edit Source Refactor Navigate Search</u>				
🔨 🚸 🔳 🎋 Debug 🛛 🗸 🖻 DebugOnl	/_Tutorial De 🔨 🌞 📄 🔻 📓 🔞	🛞 🔻 🍕 👻 🔐 New Connection	↓ 14 14 1 14 1 19 1 10 1 1 16 1 16 10 17 16 17 17 17 17 17 17 17 17 17 17 17 17 17	# 11 / 11 11 11 1 • 6 ▼ 6 ▼ 6
> ▼ 94 ▼ 129 @ 09 ▼ 189 B 11 19 ▼ 19 ▼ 10	• <> ▼ ⇒ ▼		Qu	uick Access 🛛 📽 🔤 C/C++ 🎋 De
Project Explorer 🛛 📄 😫 🔝 🗢 🗖	🖹 DebugOnly_Tutorial DebugO)nly.launch ⊠		🗄 O 🛛 🗏 Ta 🖲 B ங D
DebugOnly_Tutorial DebugOnly_Tutorial DebugOnly.launch	2 <launchcd 3 <stringat 4 <stringat 5 <stringat 6 <boolean <br="">7 <stringat 8 <intattr: 9 <stringat 10 <stringat 11 <intattr: 12 <stringat 13 <boolean <br="">14 <boolean <br="">15 <stringat 16 <stringat 17 <boolean <br="">18 <stringat 19 <boolean <br="">19 <boolean <br="">19 <boolean <br="">19 <boolean <br="">20 <stringat 20 <string< td=""><td>tribute key="com.renesas.cdt.co tribute key="com.renesas.cdt.la tribute key="com.renesas.cdt.la</td><td><pre>dt.launch.dsf.gdbremote.launchCor re.initCommands" value="'/> re.jpAdhress" value="localhost"/> re.jpadpress" value="com.renes ore.loadImage" value="true"/> re.optionTnitCommands" value="'/> re.optionTnitCommands" value="'/> re.secondGDBExe" value="'/> re.secondGDPortNumber" value="f1237 re.serverParam" value="rg E1 -t ore.setResume" value="true"/> re.setStopAt" value="true"/></pre></td><td>An outline is not available.</td></string<></stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </stringat </boolean></boolean></boolean></boolean></stringat </boolean></stringat </stringat </boolean></boolean></stringat </intattr: </stringat </stringat </intattr: </stringat </boolean></stringat </stringat </stringat </launchcd 	tribute key="com.renesas.cdt.co tribute key="com.renesas.cdt.la tribute key="com.renesas.cdt.la	<pre>dt.launch.dsf.gdbremote.launchCor re.initCommands" value="'/> re.jpAdhress" value="localhost"/> re.jpadpress" value="com.renes ore.loadImage" value="true"/> re.optionTnitCommands" value="'/> re.optionTnitCommands" value="'/> re.secondGDBExe" value="'/> re.secondGDPortNumber" value="f1237 re.serverParam" value="rg E1 -t ore.setResume" value="true"/> re.setStopAt" value="true"/></pre>	An outline is not available.
	<		>	
	Proble [∞]	ole 🔲 Proper 👒 Smart 🖻 Stack 🔋	Memor 蹋 Periph 蹋 Device 蹋 Device 孑 C	Code P 🖋 Search 🔋 Memory 🔹
	Description	^	Resource	Location
	Description		Resource	Location
< >	<			

Figure 3-13 Debug only project is created



3.3 Import Existing Projects into Workspace

The migration guideline between integrated development environments (i.e. import CS+/HEW projects to e² studio, or export to CS+) can be found at the following site.

https://www.renesas.com/products/software-tools/tools/migration-tools/migration-tools-ide.html

To import an existing e²s project to current workspace, please follow below instructions. These steps import a sample project from Renesas website to use for demonstrating debugging features in section 5.4.

1. Download the sample code for RX64M by searching for "RX64M Renesas Starter Kit Sample Code for e² studio" from Renesas website: <u>https://www.renesas.com/search/keyword-search.html</u>.

	Keyword Searc	ch		
	RX64M Renesas Starter Ki	t Sample Code for e2 studio	Search	
	Everything Product	s Documents Downloa	ads	
	Tool Downloads (1) Sa	ample Programs (1)		
	1 result sorted by date	✓ Show Filters		
	S RX64M Renesas Starter Mar.18.15	r Kit Sample Code for e2 studio		
Comments		-		
)1an2218eg0100_rx64m.pdf) and Sample Co nesas Starter Kit+ for RX64M	odes (C Language Source Files, Project(e2 st	ludio / CC-RX)) are included.	
Download				
Download Product Name		File Name	File Size	Download Link

Figure 3-14 Download the Sample Code

2. After extract the package, find the project "Tutorial"

Name	Date modified	Туре	Size
Application	12/21/2017 5:18 PM	File folder	
Async_Serial	12/21/2017 5:18 PM	File folder	
Low_Power_Mode	12/21/2017 5:18 PM	File folder	
RTC	12/21/2017 5:18 PM	File folder	
System_Input_Capture	12/21/2017 5:18 PM	File folder	
Timer_PWM	12/21/2017 5:19 PM	File folder	
Tutorial	12/21/2017 5:19 PM	File folder	

Figure 3-15 The Sample project

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

<u>F</u> ile	<u>E</u> dit <u>S</u> ource Refactor <u>N</u> avigate	Se <u>a</u> rch <u>P</u> roject	Renesas <u>V</u> iews <u>R</u> un <u>W</u> indow <u>H</u> elp
_	New Open File Open Projects from File System		h Configurations ∨ on: 월 : Ø :
	Close Close All	Ctrl+W Ctrl+Shift+W	
	Save Save As	Ctrl+S	
Ē	Save All Revert	Ctrl+Shift+S	
	Move Rename	F2	
\$	Refresh Convert Line Delimiters To	F5	
Ē	Print	Ctrl+P	🛛 Tasks 🖳 Console 🐹 🔲 Properties
	Import Export		torial HardwareDebug [Renesas GDB Hard
	Properties	Alt+Enter	

Figure 3-16 Import the sample project

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

Select Create new projects from an archive file or directory.
Select an import wizard: type filter text
Image: Second

Figure 3-17 Select import wizard

5. In the [Import Projects] dialog, select "Select root directory". Click [Browse] then select the folder "Tutorial" in the sample code package.

Import Pro Select a dire	jects ctory to search for existing Eclipse projects.			
Select roo	t directory:	~	B <u>r</u> owse	
Select <u>a</u> rc	ive file:	~	B <u>r</u> owse	
<u>P</u> rojects:				
			<u>S</u> elect All	
			Deselect All	
			R <u>e</u> fresh	
└ Copy pr	r nested projects jects into workspace jects that already exist in the workspace			
-Working se	S			
Add pro	jec <u>t</u> to working sets		Ne <u>w</u>	
W <u>o</u> rking se	S:	~	S <u>e</u> lect	
?	< <u>B</u> ack <u>N</u> ext >	<u>F</u> inish	Cancel	
(?)	< Back Next > Figure 3-18 Select projection			

6. The project "Tutorial" will be listed in "Projects". Check "Copy projects into workspace" then click [Finish]

Import Projects Select a directory to sear	ch for existing Eclipse projects.	
 Select root directory: Select archive file: 	D:\xx_OldProj\004_e2_RX_RL_testing\Input for testing e2s v5.2\1. 2015 ~	B <u>r</u> owse
Projects:	IdProj\004_e2_RX_RL_testing\Input for testing e2s v5.2\1. 20150806_Gettin	<u>S</u> elect All Deselect All R <u>e</u> fresh
Options Options Search for nested pro Copy projects into w Hide projects that all		
Working sets Add project to work Working sets:	ing sets	Ne <u>w</u> S <u>e</u> lect
?	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Cancel

Figure 3-19 Complete project import

7. Right click on the imported project and select "Upgrade Legacy e² studio Projects..."

<u>File Edit Source Refa</u>	ā -	Move	
🍕 🏘 🔳 🐐		Rename	F2
🔒 New Connection	è	Import	
🎄 🕶 💁 👻 🔍	4	Export	
	Ŷ	Upgrade Legacy e2 studio Projects	
Project Explorer 🛛		Build Project	
C Tuto del ID el const		Clean Project	
> 🎏 Tutorial [Release]	8	Refresh	F5
		Close Project	
		Close Unrelated Projects	

Figure 3-20 Upgrade the imported project

8. Select "Tutorial" project and click [Finish]

Upgrade Legacy e2 studio Projects	
Select projects to upgrade	
<mark>✓ [©] Tutorial [Release]</mark>	
Einish Cancel	

Figure 3-21 Finish the upgrading

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



Renesas e² studio

Getting Started Guide

Properties for Tuto	orial — — >	<
type filter text	Settings $\Leftrightarrow \checkmark \Leftrightarrow$	000
 Resource Builders C/C++ Build Build Variables Environment Logging Settings Tool Chain Edir C/C++ General MCU Project Natures Project Reference: Renesas QE Run/Debug Settir Task Tags Validation 	Configuration: HardwareDebug [Active] Tool Settings Toolchain Device Puild Steps Puild Artifact Binary Parsers Forror Parsers Enable toolchain integration Current Toolchain Toolchain: Renesas CCRX Version: v2.01.00 Change Toolchain (click Apply before switching tabs) Toolchain: Renesas CCRX Version: v3.02.00 Version: v3.02.00	^
< >		~
?	Apply and Close Cancel	

Figure 3-22 Update project toolchain

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



4. Build

This chapter describes the build configurations and key build features for e² studio IDE.

4.1 Build Option Settings

A new project built with 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 project "Tutorial" and select [Properties] or click ¹ button to open the Properties window.

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

<u>F</u> ile <u>E</u> dit <u>S</u> our			Ctrl+C	
% *	Ē	Paste	Ctrl+V	
	×		Delete	
		Source	>	
Project Explo		Move		
> 😂 Tutorial [J	Rename	F2	
	2	Import		
	പ്പ	Export		
		Build Project		
		Clean Project		
	8	Refresh	F5	
		Close Project		
		Close Unrelated Project		
		Build Targets	>	
		Index	>	
		Build Configurations	>	
		Run As	>	
	*	Debug As	>	
		Profile As	>	
		Team	Open Properties Dialog	
		Compare With	>	
		Restore from Local Histo	ory	
		MISRA-C	>	
		Save build settings repo	ort	
		Change Device		
	\$	C/C++ Project Settings	Ctrl+Alt+P	
	*	Run C/C++ Code Analys		
		System Explorer		
	EN., 1	Command Prompt		
		Validate		
		Configure	>	
		Source	>	
		Properties	Alt+Enter	
		·		

Figure 4-1 Open the Properties window

2. Click $[C/C++ Build] \rightarrow [Settings] \rightarrow [Toolchain]$ to view or change toolchain version.

Click the "Versions" option to change toolchain version (if additional toolchain is installed).
Renesas e² studio

Getting Started Guide

type filter text	Settings $\diamond * \diamond *$	ŏ
 > Resource Builders > C/C++ Build > Build Variables Environment > Logging > Settings > Tool Chain Edir > C/C++ General > MCU > Project Natures > Project Reference: > Renesas QE Run/Debug Settir Task Tags > Validation 	Configuration: HardwareDebug [Active] Tool Settings Toolchain Device Puild Steps Build Artifact Binary Parsers Forror Parsers Enable toolchain integration Current Toolchain Toolchain: Renesas CCRX Version: v3.02.00 Change Toolchain (click Apply before switching tabs) Toolchain Renesas CCRX Version: v3.02.00 v3.02.00 v3.01.00 v3.00.00	
< >		~
?	Apply and Close Cancel	

Figure 4-2 Change Toolchain Version

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

Builders C/C++ Build Build Variables	Configuration:	lardwareDebug [Active]	✓ Manage Co	nfigurations
Environment Logging	Environment vari	ables to set		Add
Settings	Variable	Value	Origin	Select
Tool Chain Edi	ArtifactName	Tutorial	BUILD SYSTEM	
C/C++ General	CWD	C:\Renesas\Workspace\v2020_04\Tutorial\HardwareDebug	BUILD SYSTEM	Edit
MCU	PATH	C:\Program Files (x86)\Renesas\RX\3_2_0\bin\;C:\Users\duyp	BUILD SYSTEM	Delete
Project Natures	PWD	C:\Renesas\Workspace\v2020_04\Tutorial\HardwareDebug	BUILD SYSTEM	Undefine
Project Reference:	TCINSTALL	C:\Program Files (x86)\Renesas\RX\3_2_0\	BUILD SYSTEM	Ondenne
Renesas QE	TC_VERSION	v3.02.00	BUILD SYSTEM	
Run/Debug Settir Task Tags				
Validation	Appand varial	bles to native environment		
		e environment with specified one		
	0.116.000			
>		R	estore <u>D</u> efaults	<u>A</u> pply

Figure 4-3 Build Environment Settings



Renesas e² studio

4. Setting Build Options

Right click on a project in the Project Explorer and select [Properties] or click ³⁰ button to open the Properties window.

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

User could set all build settings under 'Tool Settings' tab.

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

Click [Apply and Close] to save the build setting changes.



Figure 4-4 Build Option Settings

The detail of build option is described in compiler user manual which is stored at "{Compiler installation directory}\doc". For example, it can be found in "C:\Program Files (x86)\Renesas\RX\3_2_0\doc\".

Note: There is "*Toolchain Editor*" under "C/C++ Build", **please do not change the configuration**. The Toolchain editor is used for toolchains which are NOT supported by Renesas build support plugins.



4.2 Build A Sample Project

A project can be built by the steps below:

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

Project Explorer 🔀	9 4	Export	
↓ 🕞 🗠 🖓 🖉 🗠 🗠		Build Project	
✓ j) Includes		Clean Project	
🕟 📴 C:/Program Files (x8 🖇	8	Refresh F5	
🕞 Tutorial/generate		Close Project	
V 😕 generate		Close Unrelated Projects	
> 🖻 dbsct.c			1

Figure 4-5 Build a Sample "Tutorial" Project

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

At the end of this build, files output to the \${CONFIGDIR} directory consists of "makefile", "Tutorial.abs", "Tutorial.map", "Tutorial.mot", "Tutorial.x" etc.

"Tutorial.abs" is a Renesas standard load module in ELF/DWARF format (*.abs) used for the debugging. Because GDB supports a load module format with different ELF/DWARF specification (*.x or *.elf), hence "Tutorial.abs" has to be converted to "Tutorial.x" for the debugging in e² studio IDE.

					÷ ÷ 5		🗐 🕶 📑 👻
CDT Build Console	[Tutorial]]					
Extracting sup	port fi	les					~
			l of configura	ation HardwareDe	bug for project	Tutorial ****	^
			l of configura	ation HardwareDe	bug for project	Tutorial ****	^
17:35:27 ****	Increme		l of configura	ation HardwareDe	bug for project	Tutorial ****	^
17:35:27 **** make -j4 all	Increme		l of configura	ation HardwareDe	bug for project	Tutorial ****	

Figure 4-6 Project is built successfully.

3. In some cases, the build can be unsuccessful. The console window will show error messages, please check it and revise the source code or configuration and rebuild the project

🕄 Problems 🖳 Console 🕺 🗆 Properties 🗈 Stack An 🧏 Peripher 🏂 Device To 🛒 Code Pre 🖋 Search 🛛 Memory 👘 🗖
In the second and the second s
CDT Build Console [Tutorial] -ut+8 -nomessage
-output=obj -obj_path="src" -debug
-outcode=utf8 -nologo
<pre>ccrx -subcommand="src\cSubCommand.tmp" "/src/TutoriaI.c" /src/TutoriaI.c(26):E0520020:Identifier "uint8 t" is undefined</pre>
make: *** [src/TutoriaI.obj] Error 1
<pre>src/subdir.mk:17: recipe for target 'src/TutoriaI.obj' failed</pre>
17:02:06 Build Failed. 2 errors, 0 warnings. (took 707ms)
×
< >

Figure 4-7 Unsuccessful build shows error messages



4.3 Export Build Configuration Settings

The Project Reporter feature can export project and build configuration settings from e² studio IDE to a file for easy checking and comparison of project/build environment settings.

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

<u>F</u> ile <u>E</u> dit <u>S</u> ourc		Сору	Ctrl+C	
		Paste	Ctrl+V	
% * = :		Delete	Delete	
		Source	>	
陷 Project Explor		Move		
> 😂 Tutorial [l		Rename	F2	
			12	
		Import		
		Export		
		Build Project		
		Clean Project		
	8	Refresh	F5	
		Close Project		
		Close Unrelated Project		
		Build Targets	>	
		Index	>	
		Build Configurations	>	
	•	Run As	>	
		Debug As	>	
	vr	Profile As	>	
		Team	>	
		Compare With	>	
		Restore from Local History		
		MISRA-C	>	
	L	Save build settings report		
		Change Device		
		C/C++ Project Settings	Ctrl+Alt+P	
	*	Run C/C++ Code Analysis		
		System Explorer		
		Command Prompt Validate		
		Validate		

Figure 4-8 Project Reporter



5. Debug

This chapter describes the usage of debug configuration and key debugging features for e² studio. The following illustration refers to "Tutorial" project built (in Chapter 4.2) and based on hardware configuration: E1 emulator or E2 emulator Lite and RSK RX64M board.

Firstly, open "Tutorial" project workspace in e² studio IDE and click [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 user to perform specific task.

For instance, [C/C++] perspective has views that help user to develop C/C++ programs and [Debug] perspective has views that enable user to debug the program. If user attempts to connect up the debugger in the [C/C++] perspective, IDE will then prompt users to switch to the [Debug] perspective.

One or more perspectives can exist in a single Workbench window. User can customize them or add new perspective.

Note: For more information on debug, please refer to "e² studio Debug Help" as described in chapter 6.

5.1 Change Existing Debug Configurations

The debug configuration has to 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 "Tutorial" Project in [Project Explorer] pane to set focus.

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



Figure 5-2 Open Debug Configurations Window

2. In "Debug Configurations" windows, go to [Renesas GDB Hardware Debugging] → [Tutorial HardwareDebug]. Click on the [Main] tab to ensure the load module is "Tutorial.x".



Image: Image	Name: Tutorial HardwareDebug Main 梦 Debugger ► Startup ♥ So	urce 🗖 <u>C</u> ommon		
 C/C++ Application C/C++ Remote Application 	Project: Tutorial			Drawa
 EASE Script GDB Hardware Debugging GDB OpenOCD Debugging 	C/C++ Application: HardwareDebug/Tutorial.x			<u>B</u> rowse
GDB Simulator Debugging Java Applet Java Application	Build (if required) before launching	<u>V</u> ariables	Searc <u>h</u> Project	B <u>r</u> owse
록 Launch Group ▶ Launch Group (Deprecated) Remote Application	Build Configuration: Select Automatic	O Disable au		~
 Remote Debugger Remote Java Application Renesas GDB Hardware Det 	Use workspace settings	<u>Configure Wo</u>	orkspace Settings	
Tutorial HardwareDebug Renesas Linux Application Renesas Simulator Debuggi				
Target Communication Fran			Revert	Apply

Figure 5-3 Select Load Module

3. Switch to the [Debugger] tab, set "E1" as the debug hardware and "R5F564ML" as the target device.

Main Startup Source Common Debug hardware: E2 Lite (RX) Target Device: R5F564ML	Name: Tutorial HardwareDebug	
Debug hardware: E2 Lite (RX) V Target Device: R5F564ML	📄 Main 🕸 Debugger 🕨 🕨 Startup 🧤 Source 🔲 <u>C</u> ommon	
	Debug hardware: E2 Lite (RX) V Target Device: R5F564ML	^

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 E1 emulator and RSK RX64M board:
 - Clock
 - Main Clock Source = "EXTAL"
 - Extal Frequency(MHz) = "24.0000"

Note: Extal frequency is the value printed on the oscillator device on your board.

- Connection with Target Board
 - Connection Type = "JTag"
 - JTag Clock Frequency [MHz] = "16.5"
 - Hot plug = "No"

Hot plugin feature is only available with the device which has the capability. Please refer to device Hardware manual for "On chip debugger" specifications for the details.

- Power
 - Power Target From The Emulator (MAX 200mA) = "No"

Choose "Yes" if you would like to supply power through an emulator, when external power is unplugged. Choose "No" if external power is plugged.

- Communication Mode
 - Mode = "Debug Mode"

Another communication mode "Write On Chip Flash Memory" is used for flashing codes including ID code area, however debugger will be disconnected after flash.

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.

	🔲 <u>C</u> ommon	
GDB Settings Connection Settings Debug Tool S	Settings	1
⊿ Clock		
Main Clock Source	EXTAL	v
Extal Frequency[MHz]	24.0000	
Permit Clock Source Change On Writing Int	terna Yes	¥
Emulator	(Auto)	
Connection Type	JTag	¥
JTag Clock Frequency[MHz]	16.5	¥
Fine Baud Rate[Mbps]	2.00	×
Hot Plug	No	×
⊿ Power		
Power Target From The Emulator (MAX 200)mA) No	¥
Supply Voltage	3.3V	×
▲ CPU Operating Mode		
Register Setting	Single Chip	¥
Mode pin	Single-chip mode	¥
Communication Mode		
Mode	Debug Mode	¥
Execute The User Program After Ending The	a Deb No	×
⊿ Flash		
ID Code	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	

Figure 5-5 Change Connection Setting

 Switch to [Debug Tool Settings] sub tab which is related to debugger behavior, please refer to the e² studio Help content at "e² studio User Guide" → "Debugging Projects" for the details.



📑 🛤 🗖 🔿 🗸	Name: Tutorial HardwareDebug		
type filter text	Main Debugger Startup Source	Common	
C/C++ Application			
C/C++ Remote Application	Debug hardware: E1 (RX) \checkmark Targ	get Device: R5F564ML	
EASE Script			
GDB Hardware Debugging	GDB Settings Connection Settings Debug	Tool Settings	
GDB OpenOCD Debugging	~ IO		^
🖻 GDB Simulator Debugging	Use Default IO Filename	Yes	~
Java Applet	IO Filename	\${support_area_loc}	
Java Application	✓ General Debug		
🕏 Launch Group	Reset After Reload	Yes	~
Launch Group (Deprecated)	✓ Memory		
👒 Remote Application	Endian	Little Endian	~
🕸 Remote Debugger	Internal Flash Memory Overwrite	[1158]	
Remote Java Application	External Memory Areas	[0]	
👻 🖻 Renesas GDB Hardware Del	Work RAM Start Address	0x1000	
Tutorial HardwareDebug	Work RAM Size (Bytes)	0x500	
Renesas Linux Application	✓ System		
🖻 Renesas Simulator Debuggi	Debug the program re-writing the on-o		~
Target Communication Fran	Debug the program re-writing the on-o	hip DA No	~ ~
< >			
Filter matched 18 of 20 items		Revert	Apply

Figure 5-6 Change Debug Tool Settings

- Memory
 - Endian = "Little Endian"

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

"Internal Flash Memory Overwrite", "External Memory Areas"

These configuration control to allow/deny flashing blocks upon module downloads. Uncheck specific memory blocks if you would like to reserve the contents.

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



7. For a successful connection, [Debug] view to show target debugging information in a tree hierarchy. The program entry point is set at "PowerON_Reset() in "resetprg.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 "Debug Configurations" window.
- Select a debug configuration (e.g. "Tutorial HardwareDebug") and then click icon (Duplicates the currently selected launch configuration). A new debug launch configuration (e.g. "Tutorial HardwareDebug (1)") is created. User can rename it to identify the settings by typing in the "Name" textbox then click [Apply] button.

type filter text C/C++ Application EASE Script GDB Hardware Debugging GDB OpenOCD Debugging GDB Simulator Debugging (RH85 Java Applet Java Applet Java Application Launch Group Launch Group	Name: [Tutorial E2 Lite Debug] Main * Debugger Startup Common Source Debug hardware E2 Lite (RX) Target Device: R5F564ML GDB Settings Connection Settings Debug Tool Settings GDB Connection Settings O Autostart local GDB server Host name or IP address: localhost Connect to remote GDB server GDB port number: 61234 GDB Command:
 Launch Group (Deprecated) Remote Application Remote Debugger Remote Java Application Renesas GDB Hardware Debuggir Tutorial E2 Lite Debug Tutorial HardwareDebug Renesas Linux Application Renesas Simulator Debugging (R. 	rx-elf-gdb -rx-force-v2 Browse Variables Additional GDB Server Arguments ^
< >> Filter matched 19 of 21 items	Revert Apply

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 (RX)".
- 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: Tutorial E2 Lite Debug	
🗈 Main 🕸 Debugger 🕨 Startup 💷 <u>C</u> ommon 🛛 🐓 Source	
Save as C Local file	^
	wse

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 located in the toolbar area of e^2 studio main window.

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

File Edit Source Refactor Navigat	e Search Project Ren	nesas Views Run	Window Help
🐔 🔅 🔳 🕸 Debug	✓ C™ myprj Hard	rdwareDebug	✓ ♦ ↓
Buttons to perform build/launch debug for	r selected target	Select build/debu	g target first

Figure 5-10 Launch Bar interface

Launch Bar buttons behave as follows:

• K button builds the load module of the selected launch configuration.

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

• 🎄 📄 buttons are trigger 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]



5.4 Basic Debugging Features

This section explains the typical Debug views supported in e² 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 exist in the [Debug] view:



Figure 5-11 Useful Toolbars in Debug Views

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

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

- 🔁 button or [F5] can be used for stepping into the next method call at the currently executing line of code.
- 🐼 button or [F6] can be used for stepping over the next method call (executing but without entering it) at the currently executing line of code.
- Ib 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 🕅 button is clicked to disconnect the debugger from the selected process.

The other operations are as following:

- 👌 button can be clicked to start new debug session.
- \delta button can be clicked to reset the program to entry point at the PowerOn Reset.
- U₅ button is used for re-downloading the binary file to target system.

Note: To demonstate the features in following section, please use the sample code for RX64M from Renesas website as instruction in section 3.3.



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. e² 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] → [e² studio Breakpoint]. For a software breakpoint, select [Breakpoint Types] → [C/C++ Breakpoints].

To set a breakpoint:

- 1. Open "r_cg_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 <a>? (Hardware breakpoint) or <a>? (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 20 or a software breakpoint 20.
- Click [Windows] → [Show View] → [Breakpoints] or icon ⁹
 (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 (o or 0).
- 2. To skip all breakpoints, click on the kicon in the Breakpoints view. A blue dot with a backslash will appear in the editor pane as well as in the Breakpoints view.



Figure 5-12 [Breakpoints] view



5.4.2 Expressions View

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

Expression	Туре	Value	Address	Name : adc_count	^
R adc_count	uint8_t	0 '\0'	0x485	Details:0 '\0' Default:0 '\0'	
				Decimal:0 Hex:0x0 Binary:0 Octal:0	~
c r_cg_resetprg.c	💼 r_cg_main.c 🛛	c r_okaya_lcd.c			
67 68 69 71 72 73 74 75	/* Prototype de static void lcc /* Prototype de static void uar /* Variable to	: get_adc (void); claration for lcd_display [_display_adc (const uint) cclaration for uart_displa ct_display_adc (const uint store the A/D conversion	.6_t adc_result); ay_adc */ :8_t adc_count, con		
76 77 78	static uint8_t	<pre>adc_count = 0; claration for led display</pre>	(count */		

Figure 5-13 [Expressions] View

To watch a global variable,

- 1. Click [Window] \rightarrow [Show View] \rightarrow [Expressions] or icon $\frac{66}{2}$ 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 refresh the expression value in real-time when 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.



Renesas e² studio

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



Figure 5-14 Add local variable to Expression view



5.4.3 Registers View

Registers view lists the information about the general registers of the target device. Changed values are highlighted when the program stops.

· · · · · · · · · · · · · · · · · · ·			
🔤 Registers		🐌 🕫 🖂 📑 🖆 🏟 🎽 🖻	
Name	Value	Description	
▲ ## General Registers		General Purpose and FPU Register Group	
8889 rO	0x1518		-
888 r1	0x0		=
888 r2	0x0		
888 r3	0x0		
888 r4	0x10		
888 r5	0x80		
888 r6	0x100b		
888 r7	0x1010		
888 r8	0x0		
888 r 9	0x0		_
1919 -10	0.00	4	
Name : r0 Hex:0x1518 Decimal:5400 Octal:012430 Binary:10101000 Float:7.5670112 Default:5400			^

Figure 5-15 [Registers] View

To view the general register "r0",

- 1. Click [Window] \rightarrow [Show View] \rightarrow [Registers] or icon ¹⁰⁰/₁₀₀ 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 memory of a variable (e.g. "adc_count"),

- 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 "adc_count".



Figure 5-16 [Memory] View (1/2)



To add new renderings format (e.g. Raw Hex) for the variable "adc_count",

🐈 New Renderings...) 1. Click the tab

to select "Raw Hex" to add the rendering

This creates a new tab named "&adc_count < Raw Hex>" next to the tab "&adc_count<Hex Integer>".

Monitors	🕂 🗶 🔶	&adc_count <hex integ<="" th=""><th>er> 🚺 👍</th><th>New Renderings</th><th></th><th></th><th></th><th></th><th></th></hex>	er> 🚺 👍	New Renderings					
&adc_count		Memory Monitor: &ado					 		
 ccauc_count 		1 1	-	460					
		Select rendering(s) to cr	eate:						
		Waveform					~ [Add Render	ng(s)
		Hex Integer							-
		Fixed Floating Point Fixed Point							
		Image							
		Raw Image							
		Floating Point							
		Traditional							
		Raw Hex ASCII							
							\sim		
	2 Sourch	Cine of Internet				0 1010 .40 🔫			
Console Memory S			. 6			1 1010 🤣 📑		• •	° 🗆
Monitors	3 🔗 Search 💠 🗶 🔆	&adc_count <hex integr<="" td=""><td>er> &a</td><td>adc_count : 0x485 < F</td><td></td><td>💠 New Rend</td><td></td><td>• ~ -</td><td></td></hex>	er> &a	adc_count : 0x485 < F		💠 New Rend		• ~ -	
			er> &a 0 - 3	adc_count : 0x485 < P				• •	· •
Monitors		&adc_count <hex integ<="" td=""><td></td><td>4 - 7</td><td>Raw Hex> 😒</td><td>💠 New Rend</td><td></td><td>• •</td><td></td></hex>		4 - 7	Raw Hex> 😒	💠 New Rend		• •	
Monitors		&adc_count <hex integr<br="">Address</hex>	0 - 3	4 - 7 00 00000041	Raw Hex> 😒 8 - B	<mark> - New Rend</mark> C - F		• •	
Monitors		&adc_count <hex integr<br="">Address 0000000000000000480</hex>	0 - 3 000000	4 - 7 00 00000041 3A 20787878	Raw Hex> 🛛 8 - B 44432078	New Rend C - F 48205661		• ~ -	
Monitors		&adc_count <hex integr<br="">Address 0000000000000480 0000000000000490</hex>	0 - 3 000000 6C7565	4 - 7 00 00000041 3A 20787878 00 0000000	Raw Hex> 🛛 8 - B 44432078 480D0A00	New Rend C - F 48205661 0012022C			
Monitors		&adc_count <hex integr<br="">Address 00000000000000480 0000000000000490 0000000000</hex>	0 - 3 000000 6C7565 000000	4 - 7 00 00000041 3A 20787878 00 00000000 00 00000000	Raw Hex> 8 - B 44432078 480D0A00 00000000	New Rend C - F 48205661 0012022C 00000000		80 ▼ ⊽ □	
Monitors		&adc_count <hex integr<br="">Address 0000000000000480 0000000000000490 0000000000</hex>	0 - 3 000000 6C7565 000000 000000	4 - 7 00 00000041 3A 20787878 00 00000000 00 00000000 00 00000000 00 00000000	Raw Hex> 8 8 - 8 44432078 480D0A00 00000000 00000000	New Rend C - F 48205661 0012022C 00000000 00000000			
Monitors		&adc_count < Hex Integr Address 0000000000000480 0000000000000490 0000000000	0 - 3 000000 6C7565 000000 000000	4 - 7 00 00000041 3A 20787878 00 00000000 00 00000000 00 00000000 00 00000000 00 00000000 00 00000000	Raw Hex> 8 8 - 8 44432078 480D0A00 00000000 00000000 00000000	New Rend C - F 48205661 0012022C 00000000 00000000 00000000 00000000			
Monitors		&adc_count <hex integr<br="">Address 0000000000000480 0000000000000490 0000000000</hex>	0 - 3 000000 6C7565 000000 000000 000000	4 - 7 000000241 3A 20787878 00 00000000 00 00000000 00 00000000 00 00000000 00 00000000 00 00000000	Raw Hex> 23 8 - B 44432078 480D0A00 00000000 00000000 00000000 00000000	New Rend C - F 48205661 0012022C 00000000 00000000 00000000 00000000			

Figure 5-17 [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 a specific instruction in the program.

r_cg_resetpr	g.c 💽 r_cg_ma	ain.c 🛛				
98						^
99	/* In	itialise t	he switch	module */		
100 ffc00b		TCH Init()				
101	-	- 0				
102	/* Se	t the call	back fun	ction when	SW1 or SW2 is pressed *	·/ 🗧
103 ffc00b	0c R_SWI	TCH_SetPre	ssCallbac	k(cb_switch	_press);	
104						
105	2000 C	itialise t	he debug:	LCD */		
106 ffc00b	16 R_LCD	_Init();				
107	14					
108					the debug LCD */	
109 ffc00b 110 ffc00b				t *)" RSK+R		
110 ffc00b				t *)" Tutor	Any Switch ");	
112	52 K_CCU	_DISPINU(2	, (uinco_	(C) Press	Any Switch);	
113	/* St	art the A/	D convert	er */		
	<	are enery	b convert	c. ,	This allows the assembly	>
	-				source to be linked with the	-
🗄 Outline 🔛	Disassembly 🛛 🖡	Project Exp	lorer	_		🗆 🗖
	Opcodes	Functi	on Offsets	inter location	here 🗸 💽 🟠 😫	
55 001 05						
100	0x00000039fd0	a main+2	bsr.w		<r_main_userinit></r_main_userinit>	^
VO	0x00000565100	a main+5	_	<pre>TCH_Init();</pre>	<r init="" switch=""></r>	
103	0X00000303100				sCallback(cb switch pre	(cc).
	0xfb12030cc0f	f main+9		#0xffc00c0		33/,
	0x00000561100		bsr.a		<r setpresscall<="" switch="" td=""><td>.back></td></r>	.back>
106			R LCD	<pre>Init();</pre>		
ffc00b16	0x0000053b0b0	0 main+19	bsr.a	0xffc01651	<r init="" lcd=""></r>	
109			R_LCD	_Display(0,	(uint8_t *)" RSK+RX64M	");
	0xfb22e006c0f		mov.l	#0xffc006e	0, r2	
££-00-00	0x00000000660	1 main+29	mov.l			
ffc00b22	0x000005420b0	0 main+31	bsr.a		<r_lcd_display></r_lcd_display>	
ffc00b22 110	0x000005420b0		R_LCD	_Display(1,	(uint8_t *)" Tutorial	");
ffc00b22 110				_Display(1,	(uint8_t *)" Tutorial	");

Figure 5-18 [Disassembly] View

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

- 1. Click [Window] \rightarrow [Show View] \rightarrow [Disassembly] or icon $\stackrel{\text{IIII}}{=}$ to open the [Disassembly] view
- 2. Click icon ⁴ 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 displayed in Editor

Renesas e² studio



5.4.6 Variables View

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

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

Name	Туре	Value	Name : position Details:0 '\0'	^
(×)= position	const uint8_t	0 '\0'	Default:0 \0'	
> 🕈 string	uint8_t * const	0xffc006e0 " RSK+RX64M "	Decimal:0 Hex:0x0 Binary:0 Octal:0	~
98	<pre>r_cg_main.c r_okaya End of function R LCD I</pre>	-		
103 * * 112 ffc01664 • v 113 { 114 ffc0166a 115 ffc01671 116 } 117	Function Name : R_LCD_D oid R_LCD_Display (uint8 R_LCD_DisplaySetCurso R_LCD_DisplayStr(stri	<pre>Display[B_t const position, uint8_t * cons pr(0, position); ing);</pre>	st string)	
103 * * 112 ffc01664 • v 113 { 114 ffc0166a 115 ffc01671 116 } 117	Function Name : R_LCD_D oid R_LCD_Display (uint8 R_LCD_DisplaySetCurso	<pre>Display[B_t const position, uint8_t * cons pr(0, position); ing);</pre>	st string)	

Figure 5-21 [Variables] View

To observe a local variable (e.g. "position" for function "R_LCD_Display()"),

- 1. Click [Window] \rightarrow [Show View] \rightarrow [Variables] or icon \bigotimes to open the [Variables] view.
- 2. Step into the function "R_LCD_Display()" to view the value of local variable "position".

Note:

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

By disabling optimization, variables would become visible in most of the cases. However, it means to give up all 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 differ 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 **before** execution of the instruction at the specified address) or other events (e.g. with event condition is satisfied immediately **after** 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.

Eventpoints 🖾 💊	Breakpoints	1000 Registers	🔺 Modules	ော် Expressi	ons 📘 IO Regis	sters (x)= Var	riables			8	
					🔌 🗙 🙀		🖕 РС	C: 0/2 OA	: 0/2 🗣	8 🖻	
ype	A	Address		Data	Count	Timer	Channel	Com	ment		
🗌 🎁 Trace Start											
🗌 🖀 Trace Stop											
Trace Record											
Timer Start	J										
Timer Stop											
		Cg Edi	t Event Break								
									T : 0		
									Trigger: Of	<	- ₽
		Туре		Address	Data	Count	Timer	Channel	Comment		
		Ad	d Edit.	. Delete	PC: 0/8 OA: 0/4 A	II: 0					
								_			
									OK		Cancel

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

To set an event break for a global variable when address/data is matched (e.g. when adc_count = "0x6"),

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



		Add Eventpoint				×	
	t	Eventpoint Type: Data	Access		\sim		
		Address Settings Dat	a Access	s Setti	ngs		
		Address:	&_\$a	dc_co	unt		
		Add Eventpoint				×	
		Eventpoint Type: Data	Access				
		Address Settings Dat	a Acces	s Sett	ings		
		Data Settings:					
		Read/Write:	R	lead/V	Vrite	\sim	
		Size:	N	lot Sp	ecified	~	
		Bus Master:	C	:PU		\checkmark	
		Compare Settings:					
		Compare:	C	0x6			
		Mask Value:	C)			
		Comparison:	E	quals		\checkmark	
		Triaaer Count:		Time	r:	~	
C Edit Event Break	k					×	
						Trigger: OR 🗸 🗘 🗘	
Туре	Address	Data	Count	Timer	Channel	Comment	
🗹 👍 OR	&_\$adc_count	Read/Write All Compare (0x6)					
Add	Edit Dele	te PC: 0/8 OA: 1/4 All: 1					
						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 "_\$adc_count". (The address of this global variable is "&_\$adc_count")

- 6. Next, switch to the [Data Access Settings] tab, enable the [Compare Settings] checkbox and set the compare value equals to "0x6". Click [OK] to proceed.
- 7. Ensure that the event break for "adc_count = 0x6" is set and enabled in the [Eventpoints] view. Reset to execute the program from the start. Press SW1 6 times.

(x)= Variables 💁 Breakpoir	nts 1919 Registers 🚱	Expressions 🛛	Eventpoints		
				約 🔩 🕞 🕂 🗙 🔌 📑 🛤	r∣ 🏟 🔻
Expression	Туре	Value		Name : adc_count Details:6 '\006'	^
R adc_count	uint8_t	0хб (Нех)		Default:6 '\006'	
🐈 Add new expression				Decimal:6	
				Hex:0x6 Binary:110	
				Octal:06	
					1
					\sim
<			>	<	>
c r_cg_resetprg.c	r_cg_main.c 🛿 🚺	r_okaya_lcd.c			
134					^
135 136 ffc00b5f ⊖		ement the adc == (++ <mark>adc cou</mark>		ay using the LEDs */	
137	{	== (++auc_cou	nc))		
138 ffc00b71	adc	_count = 0;			
139	}	1		_	
◆ 140 ffc00b7a 141	led_dis	play_count(ad	c_count);		
141	14 - 1		11 11057 ¥/		× .
<					2

Figure 5-24 Execution of Event Break

Figure 5-24 shows that when adc_count reaches the value of 6 (or 0x6), the program stops at code line No.140 (right after the line of code increasing adc_count).



5.4.8 IO Registers View

IO Registers is 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 own [IO registers] view by adding IO registers selectively to the [Selected Registers] pane.

Name	Value (Hex)	Value (Bin)	Add	ress	Access
✓ ○ PORT0					
> 🛛 PDR	0x00	000	00x00 0x00	08c000	RW
> PODR	0x00	000	00000 0x00	08c020	RW
> PIDR	0x00	000	00x00 0x00	08c040	RW
> PMR	0x00	000	00000 0x00	08c060	RW
> 🛛 PCR	0x00	000	00000 0x00	08c0c0	RW
> PORT1					
All Registers Selver LPegi	sters				
📘 IO Registers 🕱		Ŧ.	\$ 🗞 🎪 🔍	. 🚍 🛅 🖂	
Name	Value (Hex)	Value (Bin)	Address	Access	
V O PORTO					
> PDR	0x00		0x0008c000	RW	
> • PCR	0x00	0000000	0x0008c0c0	RW	

Figure 5-25 [IO Registers] View

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

- 1. Click [Windows] → [Show View] → [Others…]. In "Show View" dialog, click [IO Registers] under [Debug] or icon icon icon 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 "PDR" and "PCR" to the [Selected Registers] pane. A green dot besides the IO register indicates the status of being the selected register(s).
- 4. Switch to the [Selected Registers] tab to view "PDR" 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. It helps user to track the program execution flow to search for and examine the points where problems arise.

The trace buffer is limited (with size of 1 to 32 Mbytes), oldest trace data is overwritten with the new data after the buffer has become full.

Cons	ole 🧟	Tasks 🖹 Pro	oblems 🍕		_						
No reco	ords			1		⇒ Q, 19°	H 😂 🕈	; 🚖 🖄 🖗	₩ ∀ @	÷; (U)	\$20 12 12
Cycle	Label	Addr	Sourc	Desti	Data	Size	R/W	BUS	Туре	BCN	Branc
	C	Trace Acqu	isition						×		
	Tra	Trace Mode:				ntil stop			\sim		
	Tra	ace Output	:		Do n	ot outpu	t(Internal	Buffer Us	sed) 🗠		
	Tra	ace Type:			Branc	'n			~		
	Tra	ace Capacit	y (frames	5):					~		
	Ti	mestamp F	requency	Divider:	No fr	equency	division		~		
	Er	able Times	tamp Dis	splay:	\checkmark						
	Bu	ıs Master C)f Data A	ccess:	CPU				\sim		
		art address									
		ccess(witho d address									
		id address ccess(witho									
						OK		Canc	el		
<]	>

Figure 5-26 [Trace] View (1/2)

To set a point-to-point trace between the two (2) functions (e.g. tracing from function "main()" to "R_LCD_Display()"),

- Click [Windows] → [Show View] → [Others...]. In "Show View" dialog, click [Trace] under [Debug] or icon
 to open the [Trace] view.
- 2. Turn on the Trace view by selecting the icon.
- 3. Click icon 🗐 (Acquisition) to set
 - Trace Mode: "Fill until stop"
 - Trace Type: "Branch"
 - Bus Master Of Data Access: "CPU"
- 4. Click [OK] to proceed.



Cycle	Label	Address	Source A	Destinati	Da	ata	Size	t/w	BUS N
								Ļ	
	📴 Trace Ever	ntpoints							×
(🇯 Start 🔳 S	Stop 🧉 Record							
						Triç	gger: OR	×	Û
	Туре	Address	Data	Count	Timer	Channel	Comment		
	<mark>⊠ ≜</mark> OR	&main							
	Add	dit Delete P	C: 1/8 OA: 0/4 A	All: 1					
					Г	Ok	(Cancel	
[🔄 Trace Eve	ntnoints				_			×
		Stop 📽 Record							
						Trie	gger: OR	~	Ŷ
	Turne	Address	D	ata	Count		Channel Con	amont	
	Type ☑ ▲ OR	&R LCD Disp		ala	Count	niner (liment	
	<								>
	Add	Edit Delete	C: 2/8 OA: 0/4 /	UI. 2					

Figure 5-27 [Trace] View (2/2)

- 5. Click Sector (Edit Trace Event Points) to open [Trace Eventpoints] dialog box
- 6. Under the [Start] tab, add the 1st event point at "main()" function (by the execution address "&main").
- Then, switch to [Stop] tab, add the 2nd event point at "R_LCD_Display()" function (by the execution address "&R_LCD_Display").
- 8. Next, execute the program after reset.





Figure 5-28 Point-to-Point Trace between Two Functions

The figure above shows the trace result from function "main()" to "R_LCD_Display()". The trace result can be filtered by the key trace parameters (e.g. branch type, address range) and saved to the .xml format (with the inclusion of bus, assembly and source information).

Note:

External trace feature of RX device with E20 emulator works only through Mictor-38pin interface. However, it is not available through 14pin JTAG/FINE interface, even with E20 emulator. RX emulator interface specifications can be downloaded at the following site.

https://www.renesas.com/search/keyword-search.html#q=R20UT0399

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 project.

To view the memory usage of a project,

- 1. Click [Window] \rightarrow [Show View] \rightarrow [Other...] \rightarrow [Debug] \rightarrow [Memory Usage] to open the memory usage view.
- 2. The default display of memory usage view is different for each kind of project.
 - a. The GUI of memory usage view for executable project which uses Renesas Toolchain includes 3 regions: (1) Group size region, (2) RAM/ROM Usage region/Device Memory Usage region, (3) Detail table region. The map file location is shown at bottom bar.

Size:	(1)]	RAM/ROM U	sage Device Me	emory Usage		
Progra	m: (1)			5819 by	rte(s)	RAM/ROM	Usage:			
Consta	int:			1915 by	rte(s)	RAM	-	552KB		
 Initialized Data: Uninitialized Data: 			48 byte(s) 1176 byte(s)				1 % used	JJZKD	(2)	
									(2)	
Data:				48 by		ROM	1 % used	4096KB		
Stack:				1024 by		/ ND	usea			
Others					te(s)					
				,						
Section										
Section	Group	Start address	End address	Size (byte)	Align	Attribute	Load address			1
SU	Stack	0x00000000	0x000000FF	256	4					
SI	Stack	0x00000100	0x000003FF	768	4					
	Uninitialized D	0x00000400	0x00000483	132	1					
B_1		0x00000484	0x000004A1	30	1					
	Data			1000	2				(\mathbf{n})	
B_1	Data Uninitialized D	0x000004A2	0x000008A9	1032	2					
B_1 R_1		0x000004A2 0x000008AA	0x000008A9 0x000008AF	1032	2				(3)	
B_1 R_1 B_2	Uninitialized D								(3)	
B_1 R_1 B_2 R_2	Uninitialized D Data	0x000008AA	0x000008AF	6	2				(3)	

Figure 5-29 Executable Project - Renesas Toolchain

b. For executable project which uses GCC Toolchain, "Memory region usage" region (2) will be displayed instead of RAM/ROM usage region:

Size:	(1)				`	Memory Regi	ion Usage Device Men	nory Usage		
Program	יי 🔽 🗌			14432 by	rte(s)	Memory Re	gion Usage:			1
Constan	t:			1520 by	rte(s)	ROM	1 %	4096KB		
Initialize	d Data:			12 by	rte(s)		5KB used	4096KB	(2)	
Uninitialized Data:			2140 byte(s)							
Data:				0 by	rte(s)	RAM 1	1 % KB used	512KB		
Stack:				0 by	rte(s)	/	KB useu			
Others:				5376 by	rte(s)	OFS	15 % OB used	256B		
Section	Group	Start addres			Align		Load addre			1
.ustack	Stack	0x00001D68	End address	Size (byte)	Align	Attribute				, i i
.tors	Constant	0xFFC03DA8		0						
.text	Program	0xFFC00000	0xFFC0385F	14432						
.rvectors	Constant	0xFFC03860	0xFFC03C5F	1024				(3)		
1.1	Constant	0xFFC03C60	0xFFC03DA7	328						
.rodata	Others	0x00000D68	0x00001D67	4096			0xFFC04B10			
.rodata .r_bsp_ustack	Others	0x00000868	0x00000967	256			0xFFC04610			
	Others		0x00000D67	1024			0xFFC04710			
.r_bsp_ustack		0x00000968	0.00000000	1024						

Figure 5-30 Executable Project - GCC Toolchain



c. For library project which uses Renesas Toolchain, "Library information view" will be displayed instead of "Group size view"

Library Info	ormation:			RAM/ROM Usage Device Memory Usage				
Library Information: Library name: CPU: Endian: Attribute: Number of module:		CCRX_Lib.lib RX Little user 3	RAM/ROM Usage: Library project is not supported to show RAM/ROM usage information. (2)					
Module								
		C						
Module	Section	Symbol						
Module sample1	Section							
·								
sample1								
sample1 sample2				(3)				

Note: Only available for Renesas CCRX or CCRL toolchains.

Figure 5-31 Library Project - Renesas Toolchain

d. Memory usage view is not available for library project which uses GCC Toolchain.

Group size view:

Displays the total size of Program, Constant, 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.

Library information view:

Displays the information of selected library list file. The information to be visualized on this region consists of:
The name of the selected library

- The type of CPU specified by the project
- Endian
- Attribute
- Number of module.

RAM/ROM usage region:

Shows percentage of RAM/ROM usage by numerical value and status of bar. Color of bar is based on percentage value.

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

Memory region usage region:

Display feature of this region is similar to RAM/ROM usage region.



Device memory usage region:

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

📲 Registers 🗈 Problems 🍕	🗞 Smart Browser 👒 Trace 🔋 Memory Usage 🛛 0 Mem	ory 눱 Project Explo	orer 🤌 🤣 🖓 🍕 🕆 🗖
Size:		RAM/ROM Usage	Device Memory Usage
Program:	5819 byte(s)	• 0x00000000	InternalRam
Constant:	1915 byte(s)	0x0007FFFF	2248/524288 byte(s) (0.43%)
Initialized Data:	48 byte(s)	 0x00080000 	
Uninitialized Data:	1176 byte(s)	- 0x00080000	lonegister
Data:	48 byte(s)	0x00083FFF	16384 byte(s)
Stack:	1024 byte(s)	0x00084000	Non-map area
Others:	0 byte(s)	0x00085FFF	
		0x00086000	loRegister
		0x000A3FFF	122880 byte(s)
		0x000A4000	InternalRam
		0.00045555	0/8192 byte(s) (0.00%)

Figure 5-32 Device memory usage region

Expand memory area to see all sections. Color of sections corresponds with that of Group Size Region.

Registers 🖹 Problems 🤇	🏶 Smart Browser 👒 Trace 🔋 Memory Usage 🖄 🛚 M	1emory 陷 Project Explorer		🔊 🗞 🕅 🖓 🦉
ize:		RAM/ROM Usage Device	e Memory Usage	
Program:	5819 byte(s)	0x00000000	nalRam	
Constant:	1915 byte(s)	Ux0007FFFF	2248/524288	3 byte(s) (0.43%)
Initialized Data:	48 byte(s)			
Uninitialized Data:	1176 byte(s)	0x00000000	SU	256 byte(s)
Data:	48 byte(s)	0x00000100	SI	768 byte(s)
Stack:	1024 byte(s)	0 x00000400	B_1	132 byte(s)
Others:	0 byte(s)	0 x00000484	R_1	30 byte(s)
		0 x000004A2	B_2	1032 byte(s)
		0 x000008AA	R_2	6 byte(s)
		Ox00008B0	В	12 byte(s)
		0 x000008BC	R	12 byte(s)
		■ 0x000008C8	Unused	522040 byte(s)
		• 0x00080000 loRec	gister	
		0x00083FFF	1638	34 byte(s)

Figure 5-33 Expand memory area

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.
- "Vector" tab: displays the vector table information that is retrieved from map file. This tab is only available for executable project that is configured to work with these toolchains: Renesas CCRX, Renesas CCRL.

- "Cross Reference" tab: displays the cross reference information that is retrieved from map file. This tab is only available for executable project.
- "Module" tab: contains "Module" table. "Module" tab is only available for library project that is configured to work with these toolchains: Renesas CCRX, Renesas CCRL.

Map file location:

Memory Usage will display the information of (*.map) file or library list file (*.lbp) from project. User can see the relative path of 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 an online forum dedicated to the e² studio from here.

Click on [Help] tab to open the Help menu.

<u>H</u> el	р		
2	Welcome	(1)	
? %	Help Contents Search Show Contextual Help	(2) (3)	
	Show Active Keybindings Cheat Sheets	(-)	Ctrl+Shift+L
R Ø %	RenesasRulz Community Forum Add Renesas Toolchains Perform Setup Tasks	(4)	
 𝔅 𝔅 𝔅 𝔅 𝔅 𝔅 	Check for Updates Install New Software Renesas e2 studio feedback IAR Embedded Workbench plugin n	nanager	
•	About e ² studio		

Figure 6-1 Help – Help Menu

Quick Help Tips:

- (1) Click [Welcome] for an overview of the e^2 studio and to view Release Notes.
- (2) Click [Help Contents] to open a separate Help window with a search function.

There are many useful topics under the [Help Contents]. Example, the "Debugging Projects" topic which provides useful information such as debug configuration, supported number of breakpoints, etc. It can be launched by clicking on the [Help] menu \rightarrow [Help Contents] \rightarrow "e² studio User Guide".

(3) Click [Show Contextual Help] to open the Help view within the workbench.

(4) Click [RenesasRulz Community Forum] to go an online forum that is dedicated to topics and discussions related to the e² studio (Internet connection is required).



Rev.	Date	Description
		·
Re	vision History	Renesas e ² studio User's Manual: Getting Started Guide

Rev.	Date	Description			
		Page	Summary		
1.00	May 30, 2020	-	First edition for RX, RL78, RH850 based on e ² studio 2020-04 and v7.8.0		

Renesas e^2 studio User's Manual: Getting Started Guide for RX, RL78, RH850 Family

Publication Date: Rev.1.00 May.30.2020

Published by: Renesas Electronics Corporation

RX/RL/RH Family



R20UT4819EJ0100