e² studio

Integrated Development Environment

User's Manual: Getting Started Guide

Target Device RX, RL78, RH850 and RZ Family



Rev.4.00 May 2016

Notice

- 1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
- 2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
- 3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
- 5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anticrime systems; and safety equipment etc.

Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.

- 6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
- 7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
- 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
- 11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
- 12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majorityowned subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

How to Use This Manual

This manual describes the role of the e² studio integrated development environment for developing applications and systems and provides an outline of its features.

e² studio is an integrated development environment (IDE) for RX family, RL78 family, RZ family and Renesas Synergy integrating the necessary tools for the development phase of software (e.g. design, implementation, and debugging) into a single platform. About Synergy development platform, please refer to the Synergy Gallery documents found at : <u>https://synergygallery.renesas.com/</u>

By providing an integrated environment, it is possible to perform all development using just this product, without the need to use many different tools separately.

Readers		or users who wish to understand the functions of the vare and hardware application systems.
Purpose	•	de user with the explanation of the functions provided in e^2 nce the development of their hardware and software systems s.
Organization	This manual can be broad	Ily divided into the following units.
	CHAPTER 1 GENERA CHAPTER 2 INSTALL CHAPTER 3 PROJEC CHAPTER 4 BUILD CHAPTER 5 DEBUG CHAPTER 6 HELP	ATION
How to Read This Manual	It is assumed that the read circuits, and microcontroll	ders of this manual have general knowledge of electricity, logic ers.
Conventions	Data significance:	Higher digits on the left and lower digits on the right
	Active low representation:	XXX (overscore over pin or signal name)
	Note:	Footnote for item marked with Note in the text
	Caution:	Information requiring particular attention
	Remark:	Supplementary information
	Numeric representation:	Decimal XXXX Hexadecimal 0xXXXX

TABLE OF CONTENTS

CHA	APTER 1.	GENERAL	1
1.1.	System Cor	nfiguration	1
1.2.	Operating E	Environment	1
CHA	APTER 2.	INSTALLATION	
2.1	Installation	of e ² studio IDE	
2.2.	Un-installat	tion of e ² studio IDE	4
2.3.	Upgrade of	e ² studio IDE	4
2.4.	Installation	of Compiler Package	5
CHA	APTER 3.	PROJECT GENERATION	7
3.1.	New Projec	et Generation	7
3.2.	Import Exis	sting Projects Into Workspace	
3.3.	HEW Proje	ect Import	
3.4.	CS+ Projec	t Import	
CHA	APTER 4.	BUILD	
4.1.	Build Optic	on Settings	21
4.2.	Build A Sa	mple Project	23
4.3.	Build Confi	iguration Report	
CHA	APTER 5.	DEBUG	
5.1.	Change exi	sting debug configuration	
5.2.	-	Debug Configurations	
5.3.	Basic Debu	gging Features	
CHA	APTER 6.	HELP	

CHAPTER 1. GENERAL

Renesas eclipse embedded studio (known as " e^2 studio") is a complete, state of the art development environment supporting Renesas embedded micro-controllers. It is developed based on a popular open-source Eclipse CDT (C/C++ Development Tooling) project that covers build (e.g. editor, compiler and linker control) and debug phase with an extended GDB interface support.

This chapter describes the system configuration and operating environment for e^2 studio IDE to develop applications for the RX family series microcontrollers as example. The descriptions e^2 studio in this document is based on e^2 studio V5.0, unless otherwise noted.

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

PC Hardware Environment:

Processor:	At least 1GHz (support hyper-threading/multi-core CPU)
Main Memory:	At least 1GB (2GB or larger is recommended, especially for Windows 64-bit OS)
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.

PC Software Environment:

Windows 7 (32/64-bit OS), Windows 8.1 (32/64-bit OS) and Windows 10 (32/64-bit OS)

1.2.2. Supported Toolchain

1.2.2.1. Supported Compiler

Renesas C/C++ compiler package for RX family

Renesas C compiler package for RL family

Renesas GNUARM-NONE Windows Toolchain (ELF)

GCC for Renesas GNURX Windows Toolchain (ELF)

GCC for Renesas GNURL78 Windows Toolchain (ELF)

1.2.2.2. Supported Emulator

E2 emulator Lite (RX, RL78), E1 (RX, RL78, RH850), E20 (RX), IECUBE(RL78) and Segger J-Link (RX, RZ)

1.2.2.3. Supported Simulator

Renesas Simulator (RX, RL78)

CHAPTER 2. INSTALLATION

The latest e² studio IDE installer package can be downloaded from Renesas website for free. 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.

Notes:

- 1. Two types of installers are available and Web installer (smaller one) requires Internet connection to perform installation.
- 2. Note 1: e² studio does not support updating between major versions such as from V4.x (any of V4.0.0 to V4.3.1) to V5.x (V5.0.0 or later). Please uninstall the earlier versions before installation. Alternatively, install a new e² studio into a new folder.
- 3. Note 2: e² studio installer (V4 or later) has a 'Modify' function to add or remove features in the existing installation. Please update e² studio with [Help], [Check for Updates] before using the 'Modify' function.

2.1 Installation of e² studio IDE

- Double-click on e² studio installer to invoke the e² studio installation wizard page. Click the [Next] button to continue.
- (2) Install Folder

The default installation location is set to: "C:\Renesas\e2_studio". Input install folder directly to textbox or click [Browse...] button to modify it.

Select Windows users that e² studio is installed for.

Click the [Next] button to continue.

(3) Device Families

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

(4) Extra Components

Select Extra Components (i.e. language pack, SVN & Git support, Micrium, RTOS support) to install. Click the [Next] button to continue.

(5) Components

Select Components and click the [Next] button to continue.

(6) Additional Software

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

(7) License Agreement

Read and accept the software license agreement to proceed with the [Next] button.

Please note that user has to accept the license agreement, otherwise installation cannot be continued.

(8) Shortcuts

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

(9) Summary

Click the [Install] button to install the Renesas e² studio IDE.

(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

Click [Finish] button to complete the installation.

2.2. Un-installation of e² studio IDE

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

- (1) Click on [Start] \rightarrow [Control Panel] \rightarrow [Programs and Features]
- (2) From the currently installed programs list, choose "e2 studio" and click the [Uninstall] button.
- (3) Click the [Uninstall] button to confirm the deletion in the "Uninstall" dialog.

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

2.3. Upgrade of e² studio IDE

This section illustrates how to update e^2 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 1: Two types of installers are available but Web installer requires Internet connection to perform installation.

Note 2: Offline version update using 'Differential Update program' is only applicable for e² studio Ver3.X and below only.

(2) 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.

Click [Upgrade], [Next] to begin upgrading, or choose [Install] to install into the new location.

	What do you want to do? Upgrade Upgrade to version 4.1.0.018. Install to a different location.
<u>v201510021331</u>	< Back Next > Install Cancel

Figure 2-1 Upgrade e² studio with installer

(3) Follow the steps shown in Section 2.1 Installation of e² studio IDE. When Upgrade was chosen, the step (2) Install Folder will be skipped since using the same location of existing e² studio.

e ² About e2 studio		- • •
e ²	e2 studio Version: 4.1.0.018 Parts Copyright (C) 2010-2015 Renesas Electronics Corp. All rights reserved. e2 studio IDE is an extension of software developed for eclipse.org.	× III
<u>/</u> © (e² 🗾 🕑 🗾 🧭 🌌 🚱	
Installation E	Details	ОК

Figure 2-2 e² studio – About e² studio panel

(4) Click the [Help] \rightarrow [About e2 studio] to confirm the updated version.

2.4. Installation of Compiler Package

 e^2 studio installer (V4.0 or later) is capable to 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. This procedure is common with older e^2 studio than V3.1.

Renesas Compiler Package download sites:

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

For RL78 Family: <u>http://www.renesas.com/rl78_c</u>

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

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.

pe filter text	Renesa	as Toolchain Manager	ment	⇔ - ⇒ -
General	Scar	n for installed toolchains o	on startup	
C/C++		ble warning if no toolcha	•	
Appearance		-		
Build Code Analysis	loolch	hain Type	Installation Path	
Code Style	⊳	KPIT GNUARM-NONE-	•	
Debug	4	Renesas CCRL		
⊳ Editor		v1.02.00	C:\Program Files\Renesas\RL78\1_2_0\	
File Types		V1.01.00	C:\Program Files\Renesas\RL78\1_1_0\	
Indexer	4	Renesas CCRX		
Language Mappings		v2.04.00	C:\Program Files\Renesas\RX\2_4_0\	
New C/C++ Project Wi:		V2.03.00	C:\Program Files\Renesas\RX\2_3_0\	
 Property Pages Settings Renesas 		v2.02.00	C:\Program Files\Renesas\RX\2_2_0\	
⊿ Renesas Device add-ins Sup;		Renesas SHC	C. (Frogram Files (Kenesas (IVX (2_2_0)	
Emulator				
Language Settings	⊳	KPIT GNURL78-ELF Too		
Launch Settings	\triangleright	KPIT GNURX-ELF Toolo		
Renesas Toolchain I		KPIT GNUSH-ELF Tool	c	
Smart Browser				
Smart Manual				
Task Tags				
Template Default Value				
Help Install/Update				
Library Hover				
Mylyn				
Run/Debug				
Team				
Terminal				
µC/Probe Proxy Preference			Scan Add Remove	
4				

Figure 2-3 Toolchain Management

CHAPTER 3. PROJECT GENERATION

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 should be used for e² studio installation folder name, workspace folder name, project name and its folder, and source file name. File path to any source files also should not contain any Multi-byte characters.

3.1. New Project Generation

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

 Click [File] → [New] → [C Project] to create a new C Project. New project creation Wizard as shown below will start.

e ²	C Project	
C Project Create C project of selected typ	2	
Project name: Tutorial ✓ Use default location Location: E:\workspace\Tutor ✓ Create Directory f		B <u>r</u> owse
Project type: Executable (Renesas) Sample Project Static Library (Renesas) Sample Project Debug-Only Project Debug-Only Project Debug-Only Project Debug-Only Project	Toolchains: KPIT GNUARM-N KPIT GNURL78-E	Toolchain FToolchain oolchain olchain olchain
?	Back <u>N</u> ext > <u>F</u> init	sh Cancel

Figure 3-1 New Project Creation Wizard (1/4)

(2) Enter the project name and select toolchains: "Renesas RXC Toolchain". Click [Next] to continue. If "Renesas RXC Toolchain" is not available, please follow the steps in Section 2.5 to install 'RX Compiler Package'.

Toolchain Version : Debug Hardware:	▼2.04.00 ▼ E1 (RX) ▼	* texts
Data endian : Select Target:	Little-endian data ▼ R5F564MLCxFC	
Select Configuration	: Debug using hardware ulator : Debug using simulator	nation
options, however by first configuration se (RX600) the debug ha	will be created in the project only for the se default the project will be built for the activ lected from group. Based on the device sele ardware (E1 (RX)) and debug target (R5F564 automatically created for you.	ve configuration i.e., ection you made

Figure 3-2 New Project Creation Wizard (2/4)

(3) Select Toolchain Version, Debug Hardware and Target Device.
 (For e.g., Toolchain Version: "v2.04.00", Debugger Hardware: "E1" and Target Device: "RX64M [176 pin device, part number: R5F564MLCxFC)). Click [Next] to proceed.

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



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

(4) Check the "Use Peripheral code Generator" or "Use FIT modules" options if available (depends on target device, FIT is available for RX family), otherwise ignore this setting. Click [Finish] to complete it. FIT modules can be downloaded at this dialog, or you can also add modules later.

Note: Please refer to "Renesas device support" tab in "Installation	details" dialog.
---	------------------

Summary
Project Summary: PROJECT GENERATOR PROJECT NAME : Tutorial PROJECT NAME : Tutorial CPU SERIES : RX600 CPU TYPE : RX64M TOOLCHAIN NAME : Renesas RXC Toolchain TOOLCHAIN VERSION : v2.03.00 GENERATION FILES : E\workspace\Tutorial\src\Tutorial.c Main Program E\workspace\Tutorial\src\dbsct.c Setting of B and R sections E\workspace\Tutorial\src\typedefine.h Aliases of Integer Type E\workspace\Tutorial\src\typedefine.h
Click OK to generate the project or Cancel to abort.
OK Cancel

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

(5) A project summary is displayed. Click [Ok] to generate the project.

	C/C++ - Tutorial/src/Tutorial.c - e2 studio Search Project Renesas Views Run Window Help N N N N N N N N N N N N N N N N N N N	
 Project Explorer III I HardwareDebug Tutorial [HardwareDebug] Includes Includes	Iog Iutorialc ⊠ 3 # /* 12 * 13 * 14 * 15 * 16 * 17 * 18 * * * 18 * * * 18 * * * 18 * * * 18 * * * 18 * * * 19 * 10 * 21 * 22 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * <th>void void</th>	void void
< >>		S C

Figure 3-5 New C Project Created

(6) A brand new C project named "Tutorial" is created as shown above.

This project consists of an application file "Tutorial.c" and standard start-up files (e.g. "dbsct.c", "interrupt_handlers.c", "sbrk.c" etc.). All these project and source files listed in the [Project Explorer] panel reflects 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". 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 make reference to another project's files, please backup the whole workspace folder.

3.2. Import Existing Projects Into Workspace

This section explains how to import existing projects from a directory or an archive into workspaces.



Figure 3-6 Import Existing Projects Wizard

 In e² studio IDE, click [File] → [Import] to open the HEW Project import wizard. Select "Existing Projects into Workspace" and click [Next] button to open "Import Projects" window.

e ² Import Projects	Import		
	h for existing Eclipse projects.		
• Select roo <u>t</u> directory:	E:\SampleProjects	~	B <u>r</u> owse
○ Select <u>a</u> rchive file:		×	B <u>r</u> owse
<u>P</u> rojects:			
	SampleProjects\ADC_Oneshot) ampleProjects\ADC_Repeat)	^	<u>S</u> elect All
✓ Application (E:\Sa	mpleProjects\Application)		<u>D</u> eselect All
CAC (E:\SamplePr	-		R <u>e</u> fresh
CRC (E:\SamplePr	ojects\CRC)	~	
Searc <u>h</u> for nested proj	ects		
✓ <u>C</u> opy projects into wo Hide projects that alre	orkspace ady exist in the workspace		
Working sets			
Add projec <u>t</u> to worki	ng sets		
W <u>o</u> rking sets:		~	S <u>e</u> lect
?	< Back Next >	Finish	Cancel

Figure 3-7 Import Projects Window In e² studio IDE

(2) Browse for a directory or an archive that stores the projects. All existing project are shown. Select projects to be imported and click [Finish] button to complete importing the projects.

		C/C++ - Tutorial/src/r_cg_r		tudio	
e <u>E</u> dit <u>S</u> ource Refactor <u>N</u> aviga					
/ 🕶 🔛 🕒 🛛 💥 🖛 🔦 🖛 🐔	🔜 🔌 💾 💋 🎗	🛯 🔂 🕶 🔂 🕶 🔂 🕶 🥱	🌾 🕶 🜔 🕶 🍳	L 🕈 😂 🗁 🖋 🖬 🗾	▋ 🖬 🧶 ▾ 🏹 ▾ 🌤 🗢 ▾ ⇒ ▾
				Quick Ad	ccess 📄 🖻 📴 C/C++ 🔅 Debug
Project Explorer 🔀 📃 🗖	🖻 r_cg_main.c ⊠				🗄 O 🖾 🖲 M 🗐 T 🖓 🗖
🖻 🔄 💱 🗢	52			A	💱 🖃 🎼 😿 🖋 🔍 🦈 🗢
😂 DTC	53	/* Contains functions ar	nd macro def	initions for RSK	r cg macrodriver.h
😂 I2C Master	54	<pre>#include "switch.h"</pre>			r cq cqc.h
I2C_Muster	55	/* Contains functions ar	d macro dof	initions for LCD	r_cq_icu.h
	57	#include "lcd.h"	iu macro dei	Inferons for LCD	r_cg_port.h
😂 LIN Master	58	winclude learn			r cg mtu2.h
S LIN Slave	59	/* Flashing LED function	definition	is */	- 3-
	60	<pre>#include "flashled.h"</pre>			r_cg_cmt.h
Power_Down	61				r_cg_sci.h
😂 PWM	62	/* End user code. Do not		nt generated here	r_cg_s12adb.h
😂 SPI	63	<pre>#include "r_cg_userdefing)</pre>	ne.h"		switch.h
Sync_Serial	64 66	⊕ Global variables and fur	stions.		Icd.h
Timer_Capture	68	/* Start user code for g		ot edit comment a	flashled.h
Timer_Compare	69	, Start user code for g	,10001.00 1	loc cure connerre 6	r_cg_userdefine.h
Timer_Event	70	/* Define the DSK chart	name */	¥	# NICKNAME
😂 Timer_Mode		<		>	++ ^{\$} timer_adc(void) : void
Tutorial [HardwareDebug]	& Conflicts View 2	2	□ □ □		S 🚳 S 🛷 S 📄 I 🎇 🗖 🖻
😂 USB_Basic		2			
SB_HMSC	0 items				🖹 🔜 🚽 🖬 🚽 🖬 🤹
SB_PCDC	Description		Resource	Import wizard messages	
USB_PHID					
Voltage_Detect					
>			>	<	>
	Writab	le Smart Insert 2 :	1		

Figure 3-8 Projects Import In e² studio IDE

(3) The project has been successfully imported to the e^2 studio IDE.

Instead of import project with the existing project name, e^2 studio allows the project to be renamed. With this option, only one project can be imported at a time.



Figure 3-9 Import And Rename Project Wizard

In e² studio IDE, click [File] → [Import] to open the HEW Project import wizard. Select "Rename & Import Existing C/C++ Project into Workspace" and click [Next] button to open "Rename & Import Project" window.

Rename & Import Project Select a directory to search for existing Eclipse projects. Project name: Tutorial_NewName Select root directory: E\SampleProjects Select archive file: Browse Projects: Browse Timer_Capture (E:\SampleProjects\Timer_Capture) mer_Compare (E:\SampleProjects\Timer_Compare) Timer_Compare (E:\SampleProjects\Timer_Compare) Timer_Event (E:\SampleProjects\Timer_Lowname) Tutorial (E:\SampleProjects\Timer_Mode) Tutorial (E:\SampleProjects\Timer_Mode) UsB_Basic (E:\SampleProjects\USB_Basic) USB_PAINSC (E:\SampleProjects\USB_HMSC) USB_PCDC (E:\SampleProjects\USB_PHID) Voltage_Detect (E:\SampleProjects\Voltage_Detect) Voltage_Detect (E:\SampleProjects\Voltage_Detect) v	e ²	Import	
Select root directory: E:\SampleProjects Browse Browse Projects: Timer_Capture (E:\SampleProjects\Timer_Capture) Timer_Compare (E:\SampleProjects\Timer_Compare) Timer_Event (E:\SampleProjects\Timer_Event) Timer Mode (E:\SampleProjects\Timer Mode) Tutorial (E:\SampleProjects\Tutorial) USB_Basic (E:\SampleProjects\USB_Basic) USB_HMSC (E:\SampleProjects\USB_HMSC) USB_PCDC (E:\SampleProjects\USB_PCDC) USB_PHID (E:\SampleProjects\USB_PHID)	-	-	
Select archive file: Browse Projects: Timer_Capture (E:\SampleProjects\Timer_Capture) Timer_Compare (E:\SampleProjects\Timer_Compare) Timer_Event (E:\SampleProjects\Timer_Event) Timer Mode (E:\SampleProjects\Timer Mode) Tutorial (E:\SampleProjects\Timer Mode) Tutorial (E:\SampleProjects\USB_Basic) USB_Basic (E:\SampleProjects\USB_Basic) USB_HMSC (E:\SampleProjects\USB_HMSC) USB_PCDC (E:\SampleProjects\USB_PCDC) USB_PHID (E:\SampleProjects\USB_PHID) USB_PHID	<u>P</u> roject name: Tutoria	al_NewName	
Projects: Timer_Capture (E:\SampleProjects\Timer_Capture) Timer_Compare (E:\SampleProjects\Timer_Compare) Timer_Event (E:\SampleProjects\Timer_Event) Timer_Mode (E:\SampleProjects\Tutorial) USB_Basic (E:\SampleProjects\USB_Basic) USB_HMSC (E:\SampleProjects\USB_HMSC) USB_PCDC (E:\SampleProjects\USB_PCDC) USB_PHID (E:\SampleProjects\USB_PHID)	• Select roo <u>t</u> directory:	E:\SampleProjects	B <u>r</u> owse
Timer_Capture (E:\SampleProjects\Timer_Capture) Timer_Compare (E:\SampleProjects\Timer_Compare) Timer_Event (E:\SampleProjects\Timer_Event) Timer_Mode (E:\SampleProjects\Timer_Mode) Tutorial (E:\SampleProjects\Tutorial) USB_Basic (E:\SampleProjects\USB_Basic) USB_HMSC (E:\SampleProjects\USB_HMSC) USB_PCDC (E:\SampleProjects\USB_PCDC) USB_PHID (E:\SampleProjects\USB_PHID)	○ Select <u>a</u> rchive file:		✓ Browse
Timer_Compare (E:\SampleProjects\Timer_Compare) Timer_Event (E:\SampleProjects\Timer_Event) Timer_Mode (E:\SampleProjects\Timer_Mode) Tutorial (E:\SampleProjects\Tutorial) USB_Basic (E:\SampleProjects\USB_Basic) USB_HMSC (E:\SampleProjects\USB_HMSC) USB_PCDC (E:\SampleProjects\USB_PCDC) USB_PHID (E:\SampleProjects\USB_PHID)	<u>P</u> rojects:		
	Timer_Compare (E Timer_Event (E:\Sa Timer Mode (E:\Sa Tutorial (E:\Sample USB_Basic (E:\Sam USB_HMSC (E:\Sar USB_PCDC (E:\Sam USB_PHID (E:\Sam	:\SampleProjects\Timer_Compare) mpleProjects\Timer_Event) impleProjects\Timer_Mode) eProjects\Tutorial) pleProjects\USB_Basic) npleProjects\USB_HMSC) ipleProjects\USB_PCDC) pleProjects\USB_PHID)	~

Figure 3-10 Rename & Import Project Window In e² studio IDE

(2) Browse for a directory or an archive that stores the projects. All existing project are shown. Select a project, input its new name, and click [Finish] button to import this project.

Γ

e			rial_NewName,			- e2 studio	
<u>File Edit Source Refactor Navigate</u>							
🗂 ▼ 🗄 🧠 🖕 🗞 ▼ 🐔 🖡	1) : X : 🖆 : 🖉 : X	i 🖸 🔻	oĭ + [c] + (o'	▼ : - ²	- 0 - 4		
						Quick Ac	ccess 🖹 🖻 📴 C/C++ 🎄 Debug
🎦 Project Explorer 🔀 📃 🗖	💽 r_cg_main.c 🛛						EO X ® M ■T □
🖻 🔄 🝃 🔻	52	/ Jta	art user coue i	101 11	ictude. Do i	NOT COTT CONNIENT	🐉 🖻 🚏 🚿 🖋 🛛 🗰 🛆
✓ Iutorial_NewName [HardwareD ▷ 前 Includes	53 54		ntains functior u <mark>de "switch.h</mark> "	ns and	macro def:	initions for RSK	I r_cg_macrodriver.h ∧ I r_cg_cgc.h
→ HardwareDebug → Release	55 56 57		ntains functior ude "lcd.h"	ns and	i macro def	initions for LCD	r_cg_jcu.h r_cg_port.h
 > c=> src Description.txt makefile.init 	58 59 60		ashing LED fund u de "flashled. H		definition	s */	<pre> r_cg_mtu2.h r_cg_cmt.h r cq sci.h </pre>
RSK_Header.txt Tutorial_NewName Release.la 	61 62 63		d user code. Do u de "r_cg_user o			nt generated here	r_cg_s12adb.h
Tutorial_NewName.launch Tutorial.cgp	64 66 68		l variables and art user code f			ot edit comment g	Icd.h flashled.h r_cq_userdefine.h
	69 70		fine the RSK sł	0		v	<pre># r_cg_userderine.n # NICKNAME + ^s timer_adc(void) : void</pre>
	Conflicts View 🔀				~	🖻 С 🖾 🗆 Р 🔋 М 🖫	s 🛶 s 🚀 s 🗖 i 🍡 🗖 🗖
	0 items						🖹 📑 📑 🛨 🖬 🗸
	Description				Resource	Import wizard messages	
							¥
× >	<			1.0.0	>	<	>
	Writable	2	Smart Insert	2:1			

Figure 3-11 Project Rename And Import In e² studio IDE

(3) The project has been successfully renamed and imported to the e^2 studio IDE.

3.3. HEW Project Import

This section explains HEW import feature to migrate existing project workspace to the e² studio IDE. This enables code re-usability for application codes and workspace created in HEW IDE previously.



Figure 3-12 HEW Project Import Wizard

- (1) In e² studio IDE, click [File] → [Import] to open the HEW Project import wizard. Select "HEW Project" and click [Next] button to open 'Import HEW Project(s)' window.
- (2) Browse for the HEW Project file (.hwp) and click [Finish] button to import this project.

Eile Edit Source Refactor Naviga					29 (2) 🖋 🕶 🗾 🔳 🗊	
Project Explorer 🛛 🖳 🗖	🗋 📄 .log 🛛 🔂 ma	ain.c 🛙			- 8	🔚 O 😒 💿 M 🗐 T 🖵 E
 Application [Debug] Mincludes Mincludes<td>7 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 24 7</td><td>* intended software all appl: THIS SOF LIMITED AND NON- TO THE M. FOR ANY I ANY REAS BEEN ADV. RERESAS and to d you agre followin http://m</td><td>tware is suppli for use with § is owned by Rg icable laws, ir TNARE IS PROVID TNARE, WHETHER TO WARRANTIES C TORPORTING TO COMPORTING TO COMPORTING TO COMPORTING TO COMPORT TO COMPORT TO</td><td>enssas produce nessas Electron cluding copyr ED "AS IS" ANN EEVRESS, IMPL F MERCHAITABI I SUCH WARRAN EEMITTED NOT I NOR AINY OF I T, SPECIAL, I HIS SOFTWARE, SIBLITY OF SI ght, without availability onal terms an disclaimer ssas Electron</td><td>D RENESAS MAKES NO IED OR STATUTORY, I LITY, FITNESS FOR A MITIES ARE EXPRESSLY PROHIBITED BY LAW, TS AFFILIATED COMPA NCIDENTAL OR CONSEQ EVEN IF RENESAS OR</td><td> ● □ Ⅰ²2 液 % ● 株 ● main(void): void </td>	7 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 24 7	* intended software all appl: THIS SOF LIMITED AND NON- TO THE M. FOR ANY I ANY REAS BEEN ADV. RERESAS and to d you agre followin http://m	tware is suppli for use with § is owned by Rg icable laws, ir TNARE IS PROVID TNARE, WHETHER TO WARRANTIES C TORPORTING TO COMPORTING TO COMPORTING TO COMPORTING TO COMPORT TO	enssas produce nessas Electron cluding copyr ED "AS IS" ANN EEVRESS, IMPL F MERCHAITABI I SUCH WARRAN EEMITTED NOT I NOR AINY OF I T, SPECIAL, I HIS SOFTWARE, SIBLITY OF SI ght, without availability onal terms an disclaimer ssas Electron	D RENESAS MAKES NO IED OR STATUTORY, I LITY, FITNESS FOR A MITIES ARE EXPRESSLY PROHIBITED BY LAW, TS AFFILIATED COMPA NCIDENTAL OR CONSEQ EVEN IF RENESAS OR	 ● □ Ⅰ²2 液 % ● 株 ● main(void): void
 ▶ In vect.h ▶ Ic vecttbl.c ■ Application Debug.launch 	Conflicts View 3 0 items Description	2		Resource	💽 P 🧔 T 📮 C 😒	🛄 P 🔋 M 🐚 S 🚸 S 🖵 E
Application.hwp DefaultSession.hsf Description.txt RSK_Header.txt	v .			>		C_v1.02.00' is missing for the p
· · ·		Writable	Smart Insert	1:1		

Figure 3-13 Project Migration from HEW to e² studio IDE

(3) The HEW project has been successfully imported to the e^2 studio IDE.

After conversion, all the original project and source files are kept with the newly generated project workspace in e² studio IDE. In addition, ".cproject", ".*linker", ".info" and ".project" are created and added. Both the HEW and e² studio project workspaces share the same files in the physical file location.

If HEW project import fails, please check the following two (2) pre-requisite conditions:

(i) HEW project workspace must be of the version, v4.07 and above

(ii) Files e.g. ".cproject", ".*linker", ".info" and ".project" must be deleted manually for HEW project re-import.

Tips: In order for HEW project workspace older than v4.07 to be imported, please update this workspace by using HEW v4.07 and above version first.

3.4. CS+ Project Import

For code re-usability purpose, this section explains the CS+ import feature to migrate existing project workspace to the e² studio IDE.

Figure 3-14 CS+ Project Import Wizard (1/2)

In e² studio IDE, click [File] \rightarrow [Import]. Select "Renesas Common Project File" and click [Next] button to proceed. "Import Projects" window will open.

(1) Browse for the CS+ Project file (.rcpe) and click [Finish] button to import this project.

e ²		C/C++ - Tutorial/ma			
<u>File Edit Source Refactor Navigate</u>					
📑 🕶 📙 🕒 📄 😁 🕶 🐔 🖬) e 🔌 e 🛄 e 🖉 e 💭 e 🔂 e 🔂 🔻	63 ▼ C ▼ C ▼ F	🌾 🕶 🜔 🖛 💁 ·	* 1 😂 🗁 🔗 🥆 🗾 🗏 🛽	∐:⋭ ▼ 🖗 ▼ ⊱ 🔶 ▼ → ▼
					Quick Access
🎦 Project Explorer 🛛 🗖 🗖	📄 .log 🛛 🔂 main.c 🔀				🗄 O 🕱 🛞 M 🗐 T 🖳 🗖
🗖 🔩 📴 🔻		SCLAIMER.		^	🔋 🖻 🎼 😿 🖋 🗕 🗰 🗵
a 😂 Tutorial [Debug]				onics Corporation.	stdint.h
Includes		le Name : main storv : 13 A		1.00 First Releas	jodefine.h
Software	67	story . 15 g	NB. 2012 XEL.	1.00 TILSE RELEAS	Lcd.h
i → src	69 ⊕* Sy	stem Includes 🗌			timeradc.h
b dbsct.c			le provides C9	9 type definitions	flashLED.h
flashLED.c		<pre>lude <stdint.h></stdint.h></pre>			# NICKNAME
hwsetup.c	73 75 ⊕*Us	er Includes (Proje	st Loval Inclu	doc).	++ Static Test(void) : void
id_code.c		efines MCU registe		ues)	Timer_Delay(uint32_t, int8_t)
intprg.c		lude "iodefine.h"			gConstStr : const uint8 t[]
lcd.c			ction prototyp	es & macro definit	gReplaceStr: uint8 t[]
⊳ 💽 main.c		<pre>lude "lcd.h" imer & ADC functio</pre>		* /	main(void) : void
resetprg.c		imer & ADC TUNCTIO	n definitions	~/ ¥	Static Test(void) : void
⊳ 🔊 sbrk.c	۲.			>	< >
b is switch.c	🐣 Conflicts View 🔀		~ - 8	🖹 Р 🧟 Т 📃 С 🕱	🗖 P 📓 M 🕞 S 🛞 S 🖵 🗖
ic timeradc.c	0 items				
k lc vecttbl.c	Description		Resource		
DefaultSession.hsf	Description		Resource	Import wizard messages	
Description.txt 🗸				Tellers	on on! is since for the V
< >	<		>	<	>
	Writable	Smart Insert	2:1	:	1

Figure 3-15 Project Migration from CS+ to e² studio IDE

(2) The CS+ project has been successfully imported to the e^2 studio IDE.

After conversion, all the original project and source files are kept with the newly generated project workspace in e² studio IDE. In addition, ".cproject", ".*linker", ".info" and ".project" are created and added. Both the CS+ and e² studio project workspaces share the same files in the physical file location.

CHAPTER 4. BUILD

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

4.1. Build Option Settings

The default build option is generated when a project is created and it can usually be used to build the project. However, if changing build option is necessary (e.g. Toolchain version, Optimization options, etc.), please follow the following steps before building the project.



Figure 4-1 Properties for Tutorial Project and Properties for Tutorial.c Source File

Build option can be accessed in the properties window of a project or a source file.

- (1) (1) Set the focus at the project "Tutorial" or (2) set the focus at the source file Tutorial.c
- (2) Click the icon e² (or right-click to select [Properties], or use shortcut keys [Alt]+[Enter] or [Alt]+[T]) to open properties dialog.

(3) Click "C/C++ Build" option to view or edit the configuration settings.

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

type filter text Change Toolchain Version Resource Builders C/C++ Build Project Name : Tutorial Change Toolchain Version Dependency Scan Device Environment Logging Settings Tool Chain Editor C/C++ General Project References Run/Debug Settings Task Repository V2.03.00 V2.02.00 Kestore Defaults Apply

Figure 4-2 Change Toolchain Version

 Click [C/C++ Build] → [Change Toolchain Version] to view or change toolchain version. You can choose toolchain among the versions listed in "Available Versions" drop down list, based on the configuration of Toolchain management view as mentioned at Figure 2-3.

type filter text	Environment			¢	• = -
Resource Builders C/C++ Build Build Variables	Configuration: Hard	wareDebug [Active]		✓ Manage Cor	nfigurations
Change Toolchain Version Dependency Scan	Environment variable	s to set			Add
Environment	Variable	Value	Origin	^	Select
Logging	BIN_RX	C:\PROGRA~2\Renesas	BUILD SYSTEM		Jelect
Settings	CONFIGDIR	\${workspace_loc:/\${Proj	BUILD SYSTEM		Edit
Tool Chain Editor	CWD	E:\workspace\Tutorial\	BUILD SYSTEM		Delete
C/C++ General	INC_RX	C:\PROGRA~2\Renesas			Delete
Project References Run/Debug Settings	PATH	C:\PROGRA~2\Renesas	BUILD SYSTEM		Undefine
Task Repository	PWD	E:\workspace\Tutorial\	BUILD SYSTEM		
Tusk nepository	RXC_LIB	C:\PROGRA~2\Renesas	BUILD SYSTEM		
	TCINSTALL	C:\PROGRA~2\Renesas	BUILD SYSTEM	×	
		to native environment vironment with specified one			
< >			н	Restore <u>D</u> efaults	<u>A</u> pply
?				ОК	Cancel

Figure 4-3 Build Settings for Compiler: Environment

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

Build option allows user to retain all the toolchain configuration settings, including path name specified by using the environment variables. The current build configuration is "HardwareDebug [Active]", as shown in Figure 4-3.

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\Renesas\RX\2_3_0\doc\".

4.2. Build A Sample Project

 iile Edit Source Refactor Navigate So 3 ▼	earch Proj	Open Project Close Project		ndow Help	♀ ▼ ≷ ∅	ê 😂 😂 🕂 🕶 🌛	🔲 👖 🐓 🗸 Quick Acc		
Project Explorer ⊠ □ Project Explorer ⊠ □ Tutorial [HardwareDebug] Tu	T 12 12 14 15 15 18 27 28 29 30	Open Synergy Build All Build Configur Build Project Build Working Clean Build Automat Make Target Renesas Quick	Set iically	Ctrl+B	: Tutoria .c or Main ;::init_cn	.cpp // Rem		Image: Second system Image: Second system Image: Second system Image: Second system Image: Head system Image: Second system <td< th=""><th>oid oid</th></td<>	oid oid
 A sbrk.h A stacksct.h C Tutorial.c A typedefine.h A vect.h C vector_table.c c vector_table.d Tutorial Debug.launch Tutorial HardwareDebug.launch 	31 0° 33 34 35 37 38 39 40 41 42 43	Update All Dep C/C++ Index Properties	endif roid main(vo	Alt+D			~		
	Proble 0 items Descriptio		ks 📮 Console	Properties	Resource	Jsage 🛓 Stack Analy Path	rsis 🛞 Smart Brov	vser 👘 🗢	
			Writable	Sma	rt Insert 3	: 1 Installi	ing Smart Manuals	: (73%)	

Figure 4-4 Build a Sample "Tutorial" Project

- (1) Under e² studio IDE environment, create a new project named "Tutorial" (or open any existing project).
- (2) In [Project Explorer], click at the project to set focus.
- (3) Click [Project] \rightarrow [Build Project] or $\sqrt[6]{}$ icon to build this project.

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), hence "Tutorial.abs" is converted to "Tutorial.x" for the debugging in e² studio IDE.

4.3. Build Configuration Report

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.



Figure 4-5 Project Reporter

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

CHAPTER 5. DEBUG

This chapter describes the usage of debug configuration and key debugging features for e² studio IDE. 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.

□ □ ♦	😰 🔤 C/C++ 株 Debug]
🤣 🔐 🗶 🍇 🔂 🖆 🔻		
	🍫 🎳 🗶 🔂 🗹 🗸	

Figure 5-1 Switch to [Debug] Perspective

(1) Open "Tutorial" project workspace in e² studio IDE and click [Debug] perspective.

Perspective defines purpose-specific window layout of Workbench. Each perspective consists of a combination of views, menus, and toolbars. By switching perspective, views are laid out optimized for each purposes.

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 the debugger in the [C/C++] perspective, IDE will then prompt users to switch to the [Debug] perspective.

Workbench can have multiple perspectives and user can customize them, or add even more perspectives.

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

5.1. Change existing debug configuration

For the first time of launch debugger of the project, Debug Configuration should be adjusted. Default configuration can be changed as following operations:

le Edit Source Refactor Navigate Sear	ch Project Renesas Views	Run	Window Help			
? • 🔚 🕼 🛞 • 🔦 • 🐔 🗟 💽 🛀 .				>	🌙 🗉 🖬 💈	• 🖗 • 🀤 🔶 • 🔿 •
		Q	Run	Ctrl+F11	Qu	iick Access 🛛 😭 🛛 🔂 C/C-
🛛 Project Explorer 🔀 📃 🗖	🖻 Tutorial.c 🛛	첺	Debug	F11		₽0 🖾 🖲 M 🌇 🗖
	3 ⊕ /*		Run History	>	^	💱 🖂 🗦 🔁 🙀 😼 🗣 🗰
🜮 Tutorial [HardwareDebug]	12		Run As	>		\bigtriangledown
> 🐝 Binaries	14		Run Configurations			🕂 main(void) : void
✓ (29 src	15		Debug History	>		main(void) : void
> 尾 dbsct.c		1 6	Debug As	>		
> 🔂 interrupt_handlers.c	28 ⊖//#:1		Debug Configurations		Remove	
> .c dbsct.c	18 ⊕ /* 1 27 ⊖ #ifg			>	Remove	● main(void ǿ abort(void

Figure 5-2 Open Debug Configurations Window

(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.

Create, manage, and run configurations	
Image: Second system Image: Second system type filter text Image: Second system Image: Second system <t< th=""><th>Name: Tutorial HardwareDebug</th></t<>	Name: Tutorial HardwareDebug
 GDB Simulator Debugging (SH, RL78, GHS Local C/C++ Launch Renesas GDB Hardware Attach Renesas GDB Hardware Debugging Tutorial HardwareDebug Tutorial Release Renesas Simulator Debugging (RX onl 	Tutorial Browse C/C++ Application: HardwareDebug\Tutorial.x HardwareDebug\Tutorial.x Search Project Build (if required) before launching Browse Build configuration: Use Active C Enable auto build Disable auto build Image: Our search project of the search proje
Filter matched 9 of 13 items	Apply Re <u>v</u> ert

Figure 5-3 Select Load Module

- (2) In "Debug Configurations" windows, expand the "Renesas GDB Hardware Debugging" debug configuration and click on existed debug configuration (e.g. "Tutorial HardwareDebug").
- (3) Go to the [Main] tab and browse to add the load module "Tutorial.x" located in the project build folder.

Main * Debugger Startup Source Common Debug hardware: E1 (RX) Target Device: R5F564ML	Name: Tutorial HardwareDebug		
	📄 Main 🕸 Debugger 🕒 St	artup 💱 Source 🔲 <u>C</u> ommon	
			^

Figure 5-4 Select Target Device

- (4) Switch to the [Debugger] tab, set "E1" as the debug hardware and "R5F564ML" as the target device.
 - Debug Hardware: "E1"
 - Target Device: "R5F564ML"

🕽 Main 🕸 Debugger 🛛 🕨 Startup 🦆 Source 🔲 🖸	ommon		
GDB Settings Connection Settings Debug Tool Settin	igs		
⊿ Clock			
Main Clock Source	EXTAL	× .	
Extal Frequency[MHz]	24.0000		
Permit Clock Source Change On Writing Interna	Yes	×	
Connection with Target Board			
Emulator	(Auto)		
Connection Type	JTag	× .	
JTag Clock Frequency[MHz]	16.5	× .	
Fine Baud Rate[Mbps]	2.00	× 1	
Hot Plug	No	¥ .	
⊿ Power			
Power Target From The Emulator (MAX 200mA)	No	× 1	
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 Deb	No	× 1	
⊿ Flash			
ID Code	FFFFFFFFFFFFFFFFFFFFFFFFFFF		

Figure 5-5 Change Connection Setting

- (5) Under the [Debugger] tab, go to the [Connection Settings] sub tab to configure the following based on the settings in E1 emulator and RSK RX64M board:
 - Clock
 - Main Clock Source = "EXTAL"
 - Extal Frequency(MHz) = "24.0000"
 - Connection with Target Board
 - Connection Type = "JTag"
 - JTag Clock Frequency [MHz] = "16.5"
 - Power
 - Power Target From The Emulator (MAX 200mA) = "No"
 - Communication Mode
 - Mode = "Debug Mode"

When "Power Target From The Emulator (MAX 200mA)" is set to "Yes", the emulator will power up (with current up to 200mA) the target board without an external power source

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, do be cautious to verify the board and emulator settings before connection.

In addition, e^2 studio also provides the function of duplicating existing project debug configuration for a new project. This is applicable for the projects using the same device and debugger settings.

Name: Tutorial HardwareDebug		
📄 Main 🕸 Debugger 🛛 🕨 Startup 🦆 Source 🔲 Common		
Debug hardware: E1 (RX)		
✓ 10		
Use Default IO Filename	Yes	
IO Filename	<pre>\${eclipse_home}\internal\loFiles\RX\RX64M.sfrx</pre>	
✓ General Debug		
Reset After Reload	No	
V Memory		_
Endian	Little Endian	
Internal Flash Memory Overwrite	[1158]	
External Memory Areas	[0]	
Work RAM Start Address	0x1000	
Work RAM Size (Bytes)	0x500	
✓ System		
Debug the program re-writing the on-chip PROGRAM ROM	No	
Debug the program re-writing the on-chip DATA FLASH	No	
 Performance Timer 		
Operating Frequency [MHz]		
 Start/Stop Function Setting 		
Execute function before running user program	No	
Address for start function	0x0	
Execute function after stopping user program	No	

Figure 5-6 Change Debug Tool Settings

- (6) Switch to [Debug Tool Settings] sub tab, based on the RSK RX64M board to ensure
 - Memory Endian = "Little Endian"
- (7) Click [Apply] button to confirm the settings.
- (8) Click [Debug] to execute the debug launch configuration to connect to the E1 (or E2 Lite) and RSK RX64M board.

Tutorial [Renesas GDB Hardware Debugging]
 Tutorial.x [1] Thread #1 1 (single core) (Suspended : Signal : SIGINT:Interrupt) PowerON_Reset() at r_cg_resetprg.c:71 0xfffe0000 E:/Software/e2_studio_4.0.0.16/DebugComp/rx-elf-gdbrx-force-64bit-double (7.7.1) GDB server

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

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 "r_cg_resetprg.c".

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.



Figure 5-8 Open Debug Configurations Window

(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.

reate, manage, and run configurations		The second
type Duplicates the currently selected launch configure © Debug-only © GDB Hardware Debugging © GOB Simulator Debugging (SH, RL78, RH850) Image: Select and Se	Name: Tutorial HardwareDebug (1) Irration Image: Debugger Image	Browse
< >> > > > > > > > > > > > > > > > > >	Use workspace settings Configure Workspace Settings Apply Debug	Revert

Figure 5-9 Duplicate A Selected Debug Launch Configuration

- (2) In "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.
- (3) The debug configuration can be configured as described in chapter 5.1.

Notes for RL78 debugging:

• Hot Plug connection is supported for RL78/F1A, F13, F14 and F15 only.

5.3. 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.

To open "Debug Toolbar", click the pull down menu button and check on [Show Debug Toolbar]. The following are some useful toolbars exist in the [Debug] view:



Figure 5-10 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**. While program is paused, user can perform the following operations:

- \mathbb{R}_{2} button or [F5] can be used for stepping into the next method call at the currently executing line of code.
- Solution but the 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, **b** 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:

- $\sqrt[3]{5}$ button can be clicked to restart program from entry point (same as to press $\sqrt[3]{5}$ then \mathbb{D}).
- \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.

5.3.1. Working with Breakpoints

Breakpoints can be placed at source code line or at specific address while debugger connection established. Breakpoint markers can be placed on editor views or on disassembly views, as shown below. When program counter reached to the corresponding instruction address of an enabled breakpoint, the execution suspends (so called break) before execution of the instruction. e² studio allows software and hardware breakpoint to be set explicitly in IDE. Default type of breakpoints are placed by double-clicking the left garter of editor view or disassembly view. If the hardware resources are not available, debugger console shows a warning that the breakpoint type has been replaced to "Software".

There are 2 methods to set breakpoints.

Method 1:



Figure 5-11 [Breakpoint] View – Breakpoint Setting Method 1

To set a breakpoint by method 1,

 Right-click at the source gutter to choose [Toggle Software Breakpoint] or [Toggle Hardware Breakpoint] to set hardware breakpoint 20 or software breakpoint 20.
Method 2:

0 95 fffe000e 96 97 110	Toggle Software Breakpoint Toggle Hardware Breakpoint	you use <u>strtok()</u>
0111 fffe0017	Toggle Breakpoint Ctrl+Shift+B	
112	Add Breakpoint Ctrl+Double Click	you use global class object
113	Add Dynamic Printf	
114	Disable Breakpoint Shift+Double Click	you need to change PSW PMbit
118	Breakpoint Properties Ctrl+Double Click	
119	Breakpoint Types	C/C++ Breakpoints
120 fffe001b	Switch Default e2 studio Breakpoint type to Software	C/C++ Dynamic Printf
122	Toggle Relocated Breakpoint Enabled	e2 studio Breakpoint
125 124 fffe0021	set_psw(PSW_init); // Set Ubit & Ibit for	
125 126	<pre>// chg_pmusr(); // Remove the comment of // Remove the comment of</pre>	
127 fffe0024 28 129	Multiple markers at this line - Line breakpoint: reset_program.c [line: 127] [type: Hardware] - Breakpoint installation failed: Hardware breakpoints used exceeds limit.	
130 131 132	// _CALL_END(); // Remove the comment w	
133 fffe0028	<pre>brk(); }</pre>	

Figure 5-12 [Breakpoint] View – Breakpoint Setting Method 2

To set a breakpoint by method 2,

- (1) Right-click to pop up context menu to choose [Breakpoint Types] → [e2 studio Breakpoint] (hardware breakpoint by default) or [Breakpoint Types] → [C/C++ Breakpoints] (software breakpoint)
- (2) Double-click at the source gutter to set software or hardware breakpoint
- (3) Click [Show View] → [Breakpoint] or icon open [Breakpoints] view to view the corresponding hardware/software breakpoint set.

To disable breakpoints, user can choose to disable breakpoints selectively or to skip all breakpoints.

- Breakpoints can be enabled and disabled in [Breakpoints] view. A disabled software breakpoint is displayed as a white o. A disabled hardware breakpoint is displayed as a white o.
- (2) To skip all breakpoints, click on the 🔌 icon in the Breakpoints view. A blue dot with a backslash will appear in the editor pane as well as in the Breakpoints view.

5.3.2. Expressions View

Expressions view monitors the value of global variable, static variable or local variable during debugging. The view is refreshed at break and background will be colored in yellow when the value has been changed from last time. Real-time refresh enabled variables ('R' marker is shown) are periodically updated at each configured time while running debugger.

D : 54		Expression	Туре	Value
🖻 main.c 🛛	resetprg.c	R gPeriodic_Delay	volatile uint32_t	40
 108	const uint8 t ucR	e 🚽 Add new expression		
109				
110		*		
111	* Externally Decl	ared Global V les		
112	**********	******	*****	
113	/* Declare an ext	ernal variable */		
ö 114	extern volatile u	int32 t gPeriodic Delay	j,	
115	/* Declare an ext	ernal variable †lag */		
å 116	and a second sec	<pre>int8 t gPeriodic Flag;</pre>		

Figure 5-13 [Expression] View

To watch a global variable,

- (1) Click[Show View] \rightarrow [Expressions] or icon $\oint_{\mathbb{R}}$ to open the [Expressions] view
- (2) In "main.c" at line 114, drag and drop a global variable (e.g. "gPeriodic_Delay") 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.

5.3.3. Registers View

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

IIII Registers 🖾		🗄 🕫 🖂 📫 🖶 👘 🎽 👘	
Name	Value	Description	*
4 🛗 General Registers		General Purpose and FPU Register Group	
888 rO	0x1518		Ξ
888 r1	0x0		=
888 r2	0x0		
888 r3	0x0		
800 r4	0x10		
888 r5	0x80		
888 r6	0x100b		
1999 r7	0x1010		
888 r8	0x0		
888 r9	0x0		
1010 -10	0.00	4	-
Name : r0 Hex:0x1518 Decimal:5400 Octal:012430 Binary:1010100 Float:7.567011 Default:5400			*

Figure 5-14 [Registers] View

To view the general register "r0",

- (1) Click [Show View] \rightarrow [Registers] or icon $\frac{100}{1000}$ 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.

5.3.4. 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).

Monitor Memory	X						
Enter address or expression to mo	nitor:						
&Data1							X
	ch	Project	Run Window	/ Help			
	3	ð. 👁 🗲	: i> 🗟 🗷 🏁	🛤 🔏 🐑 💊	i 🎄 🔻 🖸 👻 🛛	L 🛨 🙋 😥 🛷 🕶]	
ОК Са	ancel			The global va		L/L++ 1% De	ebu
				presented in model in model in model in model in the second second second second second second second second se			, E
🔋 Memory 🛛 🏁 Debug 🕬 Var	riables 🛛 Nod	lules 📲 l	Eventp 🔳 🤇			Registers	
Monitors	&Data1: 141	0 × Hoy In	nteger> 🛛 🕂	New Rendering		🎫 🗄 🖬 🔻	•
			-				
	Address		0 - 3	4 - 7	8 - B	C - F	1
							_
↑	0000000000	001410	00000000	00000000	00000000	00001014	ן
	0000000000 000000000		00000000 00000000	00000000 00000000	00000000 00000000	00001014 73CA7729	
Memory Monitor for "Data1"		001420					
Memory Monitor for "Data1" is specified by the address	0000000000	001420 001430	00000000	00000000	00000000	73CA7729	
-	0000000000 000000000	001420 001430 001440	00000000 F3FD6215	00000000 DD7706CD	00000000 E43BBE64	73CA7729 A8B77F59	

Figure 5-15 [Memory] View (1/2)

To view memory of a variable (e.g. "Data1"),

- (1) Click [Show View] \rightarrow [Memory] or icon [] to open the [Memory] view.
- (2) Click the icon 🕂 to open [Monitor Memory] dialog box. Enter the address of the variable "Data1".

					kn 60° Evnre	ss IIII Registers	- 6
🔋 Memory 🛛 🕸	Debug (×)=	Variables 🛋 Modules 📲 🛙	eventp 🔲 IC) Keg 🤏 Brea	ikh rybie		
					1019 1010 🚸 🤘	🎄 📧 🔠 🖽 🕯	. • `
Monitors	+ × 🤅	& &Data1 <hex integer=""></hex>	🕂 New Ren	derings			
♦ &Data1		Memory Monitor: &Data	1:0x1418				
		Select rendering(s) to cre	eate:				
		Hex Integer				Add Rende	ering(s
		Floating Point				Add Rende	ching(s)
		Fixed Point					
		Image				=	
		Raw Image					
		Floating Point					
		Traditional					
		Raw Hex				~	
		Raw Hex				T	
Memory 🛙 🅸	Debug (x)= '	Raw Hex /ariables ➡ Modules ●	Event 🗖 IG	D Reg 💊 Bre		ress IIII Register	rs
	Debug 🕬 '	/ariables 🛋 Modules 👓		D Reg ⁰₀ Bre ℃ (1418 <raw he<="" td=""><td>😵 009 Eur 😓</td><td>ress IIII Register</td><td>rs</td></raw>	😵 009 Eur 😓	ress IIII Register	rs
		/ariables 🛋 Modules 👓 I		_	😵 009 Eur 😓	<u>م</u>	rs
onitors		/ariables ➡ Modules ● & &Data1 <hex integer=""></hex>	&Data1 : 0>	1418 <raw he<="" td=""><td>×> ∞ + Ne</td><td>🔹 🎫 🖩 🔄 ew Renderings</td><td>rs</td></raw>	×> ∞ + Ne	🔹 🎫 🖩 🔄 ew Renderings	rs
onitors		Variables ➡ Modules ● & & Data1 < Hex Integer> Address	&Data1:0> 0 - 3	(1418 <raw he<br="">4 - 7</raw>	×> ≈ + Ne 8 - B	ew Renderings)	rs
onitors		Variables ➡ Modules ● & &Data1 <hex integer=""> Address 00000000001410</hex>	&Data1:0) 0 - 3 00000000	1418 <raw he<br="">4 - 7 00000000</raw>	x> ≥ 000000000	Image: Second	rs
onitors		Variables ➡ Modules ● & &Data1 <hex integer=""> Address 00000000001410 00000000001420</hex>	&Data1:0x 0 - 3 00000000 00000000	(1418 <raw he<br="">4 - 7 00000000 00000000</raw>	x> 20 + Ne 8 - B 00000000 00000000	Image: C - F 14100000 2977CA73	rs
onitors		Variables ➡ Modules ● & &Data1 <hex integer=""> Address 000000000001410 000000000001420 000000000001430</hex>	&Data1:0x 0 - 3 00000000 00000000 1562FDF3	(1418 <raw he<br="">4 - 7 00000000 00000000 CD0677DD</raw>		Image: C - F 14100000 2977CA73 597FB7A8	rs
onitors		Variables Modules ● & & Data1 < Hex Integer> Address 000000000001410 000000000001420 00000000001430 00000000001440	&Data1:0 0 - 3 00000000 00000000 1562FDF3 7841EFDF	(1418 <raw he<br="">4 - 7 00000000 0000000 CD0677DD 8F71503A</raw>	Image: Non-Section 1 Image: Non-Section 1 8 - B 000000000 00000000 64BE3BE4 5FC62F64	Image: Second	rs
onitors		Variables Modules ● & Data1 < Hex Integer> Address 00000000001410 00000000001420 00000000001430 00000000001440 0000000001450	&Data1:0 0 - 3 00000000 00000000 1562FDF3 7841EFDF 230164F5	(1418 <raw he<br="">4 - 7 00000000 0000000 CD0677DD 8F71503A 0216E9DC</raw>		Image: C - F 14100000 2977CA73 597FB7A8 DBBF6B20 4F9C3524	rs
lonitors		Variables Modules ● & & Data1 < Hex Integer> Address 000000000001410 000000000001420 000000000001430 00000000001430 00000000001450 00000000001450 0000000001450	&Data1:0x 0 - 3 00000000 00000000 1562FDF3 7841EFDF 230164F5 320AF822	(1418 < Raw Hereinstein) 4 - 7 00000000 00000000 CD0677DD 8F71503A 0216E9DC 80E0489B		Image: C - F 14100000 2977CA73 597FB7A8 DBBF6B20 4F9C3524 EE1B0737	

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

To add new renderings format (e.g. Raw Hex) for the variable "Data1",

(1) Click the tab 🖶 New Renderings...) to select "Raw Hex" to add the rendering

This creates a new tab named "&Data1 < Raw Hex>" next to the tab "&Data1 <Hex Integer>".

5.3.5. 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.

<pre>89</pre>			
<pre>94 {</pre>	፪ r_cg_main.c ⊠	- 1	
96 fffe8948 R_NAIN_UserInit(); 97 /* Start user code. Do not edit comment generated here */ 98 /* Display Splash Screen */ 99 /* Display LCD(0, (uint8 t *)"Renesas"); 101 fffe894b Display_LCD(1, (uint8_t *)NICKNAME); 102 /* Begins the initial LED flash sequence */ 103 /* Begins the initial LED flash sequence */ 104 fffe8963 fffe8963 Flash_LED(); ** Display_LCD(0 main+0 105 /* Begins the initial LED flash sequence */ ffffe8946: Opcodes p1 Function Offsets rtsd #4 main: bsr.w 0xfffe896f <r_main_userinit> 0 Display_LCO(0, (uint8_t *)"Renesas"); fffe894b: 0x000000000392700 main+0 bsr.w 0xfffe896f <r_main_userinit> 0 0 Display_LCO(0, (uint8_t *)"Renesas"); fffe894b: 0x00000000000000000000000000000000000</r_main_userinit></r_main_userinit>	94 Jutoga/src/r.cg.main.cj	^	^
97 /* Start user code. Do not edit comment generated here */ 98 /* Display Solash Screen */ 99 /* Display LCD(0, (uint8 t *)"Renesas"); 101 ffe894b Display LCD(1, (uint8 t *)"Renesas"); 102 /* Begins the initial LED flash sequence */ 104 fffe8963 rffe8963 Flash_LED(); 105 /* Begins the initial LED flash sequence */ rtsd #4 main: bsr.w 0xfffe896f <r_main_userinit> 0100 Display_LCD(0, (uint8_t *)"Renesas"); 101 fffe894b: 0x00000000392700 main+0 0100 Display_LCD(0, (uint8_t *)"Renesas"); fffe894b: 0x00000000392700 main+0 Display_LCD(0, (uint8_t *)"Renesas"); fffe894b: 0x000000000392700 main+0 Display_LCD(0, (uint8_t *)"Renesas"); fffe894b: 0x000000000392700 main+0 Source to be linked with the</r_main_userinit>	L		
99 /* Display Splash Screen */ 100 fffe894b Display LCD(0, (uint8 t *)"Renesas"); 101 ffe8957 Display_LCD(1, (uint8_t *)NICKNAME); 102 /* Begins the initial LED flash sequence */ 104 fffe8963 Flash_LED(); 105 /* Display_LCD(I, (uint8_t *)"Renesas"); 106 fffe8963 107 /* Begins the initial LED flash sequence */ 108 Flash_LED(); 109 fffe8948: 0x00000000392700 main+0 0100 Display_LCD(0, (uint8_t *)"Renesas"); 101 Display_LCD(0, (uint8_t *)"Renesas"); 102 Display_LCD(0, (uint8_t *)"Renesas"); 108 mov.1 #0, r1 109 Display_LCD(0, (uint8_t *)"Renesas"); 100 Display_LCD(0, (uint8_t *)"Renesas"); 111 Display_LCD(0, (uint8_t CD) 111 bsr.a 0xfffe806f <display lcd=""> 111 bsr.a 0xfffe806f <display lcd=""> 111 bsr.a 0xfffe806f <display lcd=""></display></display></display>	97 /* Start user code. Do not edit com	ment generated here */	
Display LCD(0, (uint8 t *)"Renesas"); 101 ffe8957 102 Display_LCD(1, (uint8_t *)NICKNAME); 103 /* Begins the initial LED flash sequence */ 104 fffe8963 105 Flash_LED(); ** Disassembly X 104 fffe8963 105 Function Offsets 106 fffe8946: 0 107 Function Offsets 108 fffe8948: 0x00000000392700 main+0 109 Display_LCD(0, (uint8_t *)"Renesas"); 100 Display_LCD(0, (uint8_t *)"Renesas"); 116 fffe894b: 0x00000000000000000000000000000000000			
102 103 104 104 fffe8963 /* Begins the initial LED flash sequence */ Flash_LED(); ** Disassembly % Enter location here ** Disassembly % Function Offsets fffe8946: 0 Opcodes fffe8948: 0x00000000392700 main+0 bsr.w 0xfffe896f <r_main_userinit> o 100 fffe894b: 0x00000000000000000000000000000000000</r_main_userinit>):	
<pre>103 104 fffe8963 /* Begins the initial LED flash sequence */ Flash_LED(); * * * * * * * * * * * * * * * * * * *</pre>	101 #fe8957 Display_LCD(1, (uint8_t *)NICKNAME)		
104 fffe8963 Flash_LED(); 105 Flash_LED(); 106 Enter location here 107 Disassembly % 108 Opcodes 109 Function Offsets 100 Display_LCD(0, (uint8_t *)"Renesas"); 100 Display_LCD(0, (uint8_t *)"Renesas"); 101 fffe894b: 0x00000000000000000000000000000000000			
Ins Ins Image: Construction of the set of	,	quence */	
<pre> Control Contro Control Control Control Control Control Control Control Contr</pre>		· · · · · · · · · · · · · · · · · · ·	4
ffffa8946: 0 Opcodes 01 Function Offsets rtsd #4 main: bsr.w 0xfffe896f <r_main userinit=""> bsr.w 0xfffe896f <r_main userinit=""> 0 100 Display_LCD(0, (uint8_t *)"Renesas"); fffe894b: 0x000600000000000000000000000000000000</r_main></r_main>		>	
<pre>fff48946: 0 Opposed p1 final field in the field in t</pre>		Enter location here 🗸 🖌 🐑 🛱 🏹 🕄 🖄 🔀	
fff 8948: 0x0000000392700 main+0 bsr.w 0xfffe896f <r_main_userinit> 100 Display_LCD(0, (uint8_t *)"Renesas"); fffe894b: 0x000fb22bc16feff main+3 fffe8951: 0x0000000051cf2ff main+11 fffe806f <display_lcd> this allows the assembly source to be linked with the</display_lcd></r_main_userinit>	fffe8946: 0 Opcodes 01 Function Offsets rts	d #4	A
<pre>Display_LCD(0, (uint8_t *)"Renesas"); fffe894b: 0x00fb22bc16feff main+3 mov.l #0xfffe16bc, r2 fffe8951: 0x00000000006b01 main+9 mov.l #0, r1 bsr.a 0xfffe806f <display lcd=""> This allows the assembly source to be linked with the</display></pre>			
fffe894b: 0x00fb22bc16feff main+3 mov.1 #0xfffe16bc, r2 fffe8951: 0x00000000000000000000000000000000000			
fffe8951: 0x000000006601 main+9 mov.1 #0, r1 fffe8053: 0x000000051cf7ff main+11 bsr.a 0xfffe806f <display lcd=""> This allows the assembly source to be linked with the</display>	VO		
fffe8053: execonogene51cf2ff main+11 bsr.a 0xfffe806f <display lcd=""> This allows the assembly source to be linked with the</display>			
This allows the assembly source to be linked with the			-
source to be linked with the		>	
	C source (active debug		
	, Ç		
	context).		

Figure 5-17 [Disassembly] View

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

- (1) Click[Show View] \rightarrow [Disassembly] or icon \blacksquare to open the [Disassembly] view
- (2) Click icon 🔄 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.

1	Toggle Software Breakpoint	
-	Toggle Hardware Breakpoint	
	Toggle Breakpoint	Ctrl+Shift+B
	Add Breakpoint	Ctrl+Double Click
	Add Dynamic Printf	
	Enable Breakpoint	Shift+Double Click
	Breakpoint Properties	Ctrl+Double Click
	Breakpoint Types	>
	Switch Default e2 studio Breakpoint type to Software	
	Toggle Relocated Breakpoint Enabled	
	Delete Relocated Breakpoint	
±.	Go to Disassembly	
ø	Toggle Timer Start Eventpoint	
ø	Toggle Timer Stop Eventpoint	
	Edit Eventpoint	
	Disable Timer Stop Eventpoint	
	Disable Timer Start Eventpoint	
D.	Clear Coverage Markers	
	Go to Annotation	Ctrl+1
	Add Bookmark	
3	Add Task	
- -	Show Source Addresses	
~	Show Eventpoints	
\checkmark	Show Quick Diff	Ctrl+Shift+Q
1	Show Line Numbers	
	Folding	>
	Preferences	





Figure 5-19 Source Addresses displayed in Editor

5.3.6. Variables View

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

524			
526	⊕* Outline	StartD <u>ebounceTimer</u>	
534 fffc88ba	⊖void StartDeboun	ceTimer(uint16_t compare_match)	
535	{		
536	⊖ /* Declare l	ocal static variable to track if the C	MT timer has been
537		sed yet */	
538	static bool	<mark>timer_initialised</mark> = false;	
539			
540	⊖ /* Check if	the CMT timer is not initialised (firs	t time function has bee
541	called).	۴/	
542 fffc88be	⊖ if (!timer_in	itialised)	
543	{		
544	/* Disab	le register protection */	
545 fffc88d1	SYSTEM.P	RCR.WORD = $0 \times A50B$;	
546			
	•		
x)= Variables 🛛			
	Туре	Value	
Name		2048	
	atch uint16 t		
Name ⋈⁼ compare_m ⋈= timer initial	_	true	

Figure 5-20 [Variables] View

To observe a local variable (e.g. "compare_match" for function "StartDebounceTimer()"),

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

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

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

				X X 3	k 📴	0	PC: 0/2	2 OA: 0/2 🛛 🎭
Туре	Classifier Edit Event	Break						X
Trace Start						Т	rigger: OR	▼ ♣ ↔
Trace Record	Туре	Address	Data	Count	Timer	Channel	Comment	
Timer Stop								
	Add	Edit Delete P	C: 0/8 OA: 0/4 All: 0)				
							OK	Cancel

Figure 5-21 [Eventpoints] View (1/2)

To set an event break for a global variable when address/data is matched (e.g. when gFlashCount = "0xB0"),

- (1) Click [Show View] \rightarrow [Eventpoints] or icon \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	X			
	Eventpoint Data /	Access 🔹			
	Address Settings Da	ta Access Settings			
	Address:	&_\$gFlashCount			
	Address Settings Da	ta Access Settings	_		
	Data Settings:				
	Read/Write:	Read/Write 🔻			
	Size:	Not Specified 🔻			
	Compare Settings	:			
	Compare:	0xB0			
	Mask Value:	0			
	Comparison:	Equals 🔻			
	1				
● Eventpoints 🛛		X X 🗱 🖬 😣	🐁 PC	: 0/8 OA	: 1/4 🗞
Туре	Address	Data	Count	Timer	Chann
Trace Start					
Trace Stop					
▲ ✓ ^c ! Event Break					
I OR	&_\$gFlashCount (0	Read/Write All Compare (0xb0)			0

Figure 5-22 [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 "_\$gFlashCount". (The address of this global variable is "&_\$gFlashCount")
- (6) Next, switch to the [Data Access Settings] tab, enable the [Compare Settings] checkbox and set the compare value equals to "0xB0". Click [Ok] to proceed.
- (7) Ensure that the event break for "gFlashCount = 0xB0" is set and enabled in the [Eventpoints] view. Reset to execute the program from the start.

Debug - C:\Users\ting	.huang.uw\My Documents	\e2_studio\work	space\Tutorial	\src\flashLED.c - e2	stu 🗆 🗆 💥
	actor Navigate Search				
📬 🕶 🛛 🖪 🕹 🛛 💌). 🔿 . R 🖬 🚟	R 🕄 📆 🖌	1 🔁 🖉 🌣 🗸 () + 9 <u></u> +
😂 😂 🛷 🕶 🍠 🖢 👻	$[\bullet \bullet \bullet \bullet \bullet \bullet] =$	1	Quick Ac	cess 🕴 🖻	曀 C/C++ 🕸 Debug
🏇 Debug 🔎 Eventpoin	its 🖾		🗙 🗙 🔆 🛙	6 🙉 🖕 PC: 0/	'8 OA: 1/4 🗞 🗖 🗖
Project Saved Temple	ates				
ৰু Expressions প্ল			<u>بة</u> +	×i ⊡ + × 🔆	[] ːː () ◊ ▽ □ □
Expression	Туре	Value			
R gFlashCount	uint16_t	176			
🕂 Add new expression	on				
Hex:0xb0 Binary:101100 Octal:0260	00				• •
lashLED.c 💿 0xf	fff8000 🗖 resetprg.c	🗖 main.c	flashLED.c	23	- 8
72 73 74 fffc8002 75	/* Flash the LED while((gSwitchF]				tch is presse
➡ 76 fffc8014	for (ulLed_[)elay = 0; u	Led_Delay	< 4000000; ++1	ulLed_Delay)
77	{				
78 79	}				=
80	/* Tl +	1 150 61			
81 fffc801d	Toggle_LED()		er a specif	ic delay. */	~

Figure 5-23 Execution of Event Break

Figure 5-23 shows that when gFlashCount reaches the value of 176 (or 0xB0), the program stops at code line no. 76.

5.3.8. 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.

IO Registers 🛛			🖻 🍣 🔕 🏌	🍃 🔍 🚍 💾 🖻 💎
Name	Value (Hex)	Value (Bin)	Address	Access
V O PORTO				
> 👄 PDR	0x00	0000	0000 0x0008c0	00 RW
> PODR	0x00	0000	0000 0x0008c0	20 RW
> PIDR	0x00	0000	0000 0x0008c0	40 RW
> PMR	0x00	0000	0000 0x0008c0	60 RW
⊳ ⊚ PCR	0x00	0000	0000 0x0008c0	c0 RW
> PORT1				
All Registers Selected Reg	isters	Ē 🛱 🗲	🕯 🍲 🔍 🚍	
	Value (Hex)	14 L (B) 1	Address	Access
Name	value (Hex)	Value (Bin) A	address .	
Name	0x00	Value (Bin) A		RW
V O PORTO)x0008c000	RW RW

Figure 5-24 [IO Registers] View

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

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

5.3.9. 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.

😂 Trace	x	S 🗐 🗄	₽Q. B°	빛 🗲 🗢 🖄 🖗	▼ 🖻 🍫 🚺 ▽ 🗆 🗖
No record	5				
Cycle	Label	Address	Source A	e ² Tra	ce Acquisition
				Trace Mode: Trace Output:	Fill until stop V Do not output(Internal Buffer Used) V
				Тгасе Туре:	Branch V
				Trace Capacity:	×
				Timestamp Frequency Divider:	✓
				Enable Timestamp Display:	
					OK Cancel

Figure 5-25 [Trace] View (1/2)

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

- Click [Windows] → [Show View] → [Others...]. In "Show View" dialog, click [Trace] under [Debug] or icon to open the [Trace] view.
- Turn on the Trace view by selecting the [U] icon.
- Click icon 📰 (Acquisition) to set
 - Trace Mode: "Fill until stop"
 - Trace Type: "Branch"
- Click [OK] to proceed.

Trace			FY 94 10	H 🕞 🕯		P" [###		1 🍫 🛛] ~ -	
recon cle	Label	Address Source	Destin	Data	Size	R/W	/	BUS M	Туре	
Trace Eventpoints										
🎼 Start 🗃 Stop 🧉 Record										
	_				Trigg	er: Of	R	•	} f	
Typ	e	Address		Data Count			Timer Channel Comment			
V	OR	&main (0xfffc8228)	J					0		
			1						57	
	ace Eventpoi	nts						l	X	
e ² Tra		🖉 🍧 Record								
e² Tra	tart 📓 Stop				Trigge	er: OR	{	• 0	, [f]	
	tari 🛎 Stor				mgge					
		Address		Data		ount 1	limer	Channel	Comn	

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

- Click 😂 (Edit Trace Event Points) to open [Trace Eventpoints] dialog box
- Under the [Start] tab, add the 1st event point at "main()" function (by the execution address "&main", or 0xFFFC8228).
- Then, switch to [Stop] tab, add the 2nd event point at "Display_LCD()" function (by the execution address "&Display_LCD", or 0xFFFC816F).
- Next, execute the program after reset.



Figure 5-27 Point-to-Point Trace between Two Functions

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

CHAPTER 6. HELP

The help system allows user to browse, search, bookmark and print help documentation from a separate Help window or Help view within the workbench. User can also access online forum dedicated to e² studio from here.

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

Figure 7-1 Help Menu

Quick Help Tips

- ① Click [Welcome] for Overview of e² studio, link to access IDE tutorial and sample, and to view Release Notes.
- ^② Click [Help Contents] to open a separate Help window with search function.
- ③ Click [Dynamic Help] to open Help view within the workbench.
- ⊕ Click [RenesasRulz Community Forum] to go online forum that is dedicated to topics and discussion related to e² studio IDE. Internet connection is required.

Under the [Help Contents] window, there are many useful topics. One of them is the "e² studio Debug Help" topic, which provides useful information such as debug configuration, supported number of breakpoints, usage of emulator etc. It can be launched by clicking on [Help] menu \rightarrow [Help Contents] \rightarrow "e² studio Debug Help".

Revision Record

	Date		Description				
Rev.		Page	Summary				
1.00	Sep 30, 2013	-	First Edition issued				
2.00	Apr 15, 2014	-	Supporting e ² studio IDE v3.0.0.022				
2.10	Dec 08, 2014	-	Supporting e ² studio IDE v3.1.2.009				
3.00	June 16, 2015	-	Supporting e ² studio IDE v4.0.0.023				
4.00	May 27, 2016	-	Supporting e ² studio IDE v5.0.0.043				



Renesas Electronics Corporation

e² studio User's Manual: Getting Started Guide

Publication Date: Rev.4.00 May 27, 2016

Published by: Renesas Electronics Corporation



SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information. Renesas Electronics America Inc. 2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A. Tel: +1-408-588-6000, Fax: +1-408-588-6130 Renesas Electronics Canada Limited 9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3 Tel: +1-408-588-7000 Renesas Electronics Canada Lourine End, Buckinghamshire, SL8 5FH, U.K Tel: +44-1828-585-100, Fax: +44-1228-585-900 Renesas Electronics Europe Limited Dukes Meadow, Milboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K Tel: +44-1828-585-100, Fax: +49-211-6503-01, Fax: +49-211-6503-1327 Renesas Electronics (China) Co., Ltd. Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China Tel: +49-11-5603-0, Fax: +49-211-6503-1327 Renesas Electronics (China) Co., Ltd. Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China Tel: +49-12-226-0880, Fax: +49-21-1250-099 Renesas Electronics (Shanghai) Co., Ltd. Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P.R. China 200333 Tel: +49-21-2226-0880, Fax: +486-21-2226-099 Renesas Electronics Taiwan Co., Ltd. Unit 101-1611, 16F-, Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong Tel: +852-256-6888, Fax: +865-2886-9022 Renesas Electronics Singapore Plac. Ltd. 30 Bendemeer Road, Unit #00-02 Hyllux Innovation Centre, Singapore 339949 Tel: +656-213-0200, Fax: +66-213-0300 Renesas Electronics Singapore Plac. Ltd. 30 Bendemeer Road, Unit #06-02 Hyllux Innovation Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia Tel: +00-3756-3990, Fax: +46-6213-0300 Renesas Electronics India Pvt. Ltd. No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India Tel: +01-05729700, Fax: +46-6213-0777 Renesas Electronics Korae Co., Ltd. 127, 293 Teheran-ro, Gamam-Gu, Seoul, 135-080, Korea Tel: +91-80-672087777 Renesas Flectronics Korae Co., Ltd. 127, 293 Teheran-ro, Camama-G

e² studio