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; anti-crime 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 majority-owned 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 and RZ family integrating the necessary tools for the development phase of software (e.g. design, implementation, and debugging) into a single platform.

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

This manual is intended for users who wish to understand the functions of the e² studio and design software and hardware application systems.

Purpose

This manual aims to provide user with the explanation of the functions provided in e² studio when they commence the development of their hardware and software systems using the targeted devices.

Organization

This manual can be broadly divided into the following units.

CHAPTER 1 GENERAL
CHAPTER 2 INSTALLATION
CHAPTER 3 PROJECT GENERATION
CHAPTER 4 BUILD
CHAPTER 5 DEBUG
CHAPTER 6 HELP

How to Read This Manual

It is assumed that the readers of this manual have general knowledge of electricity, logic circuits, and microcontrollers.

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

CHAPTER 1. GENERAL .................................................................................................................. 1
1.1. System Configuration ............................................................................................................... 1
1.2. Operating Environment ............................................................................................................ 1

CHAPTER 2. INSTALLATION ........................................................................................................ 3
2.1. Installation of e² studio IDE ................................................................................................. 3
2.2. Un-installation of e² studio IDE ............................................................................................. 4
2.3. Upgrade versions using e² studio installer ............................................................................. 5
2.4. Update plugins over Internet .................................................................................................. 5
2.5. Installation of Compiler Package .......................................................................................... 9

CHAPTER 3. PROJECT GENERATION .................................................................................... 10
3.1. New Project Generation ........................................................................................................ 10
3.2. Import Existing Projects Into Workspace ............................................................................. 16

CHAPTER 4. BUILD .................................................................................................................... 17
4.1. Build Option Settings ............................................................................................................. 17
4.2. Build A Sample Project ......................................................................................................... 20
4.3. Export Build Configuration Settings .................................................................................... 21

CHAPTER 5. DEBUG ................................................................................................................... 22
5.1. Change Existing Debug Configurations ............................................................................... 22
5.2. Create New Debug Configurations ...................................................................................... 27
5.3. Launch Bar ........................................................................................................................... 28
5.4. Basic Debugging Features .................................................................................................... 29

CHAPTER 6. HELP ..................................................................................................................... 50
CHAPTER 1. GENERAL

Renesas eclipse embedded studio (known as “e² 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 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.

![System Configuration Diagram]

**Figure 1-1 System Configuration**

1.2. Operating Environment

Below are the system requirements for this product.

1.2.1. System Requirements

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.

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 RL78 family

Gnu ARM Embedded Toolchain for RZ and Renesas Synergy

GNURX Windows Toolchain (ELF)

GNURL78 Windows Toolchain (ELF)

1.2.2.2. Supported Emulator

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

1.2.2.3. Supported Simulator

Renesas Simulator (RX), GDB Simulator (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.

e² studio installer can be used to upgrade e² studio as well as new installation. ‘Modify’ function is available if you chose an existing installation folder at step (2).
However, it does not support update between major versions such as from V5.4 to V6.0, or from V6.3 to V7.0. Please uninstall the earlier versions before installation. Alternatively, install new e² studio into a new folder if you would like to keep earlier versions.

2.1. Installation of e² studio IDE

(1) 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.
Click the [Next] button to continue.

Note1: If you would like to have multiple versions of e² studio, please specify new folder here.
Note2: Multi-byte characters cannot be used for e² studio installation folder name.

(3) Device Families
Select Devices Families to install. Click the [Next] button to continue.

(4) Extra Components
Select Extra Components (i.e. Language packs, SVN & Git support, 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.

Note: With no Internet access available, additional software installation can be skipped because software catalog cannot be downloaded. You can still continue installation, anyway.

(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.

Note: If you already have installed e² studio in another location, it is recommended to rename to distinguish from the other e² studio(s).
(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 [OK] 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) 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 e² 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 e² studio at multiple locations, you may not able to find uninstaller in “Apps & features” of Control Panel. In such cases, launch e² studio uninstaller located at:

{e² studio installed folder}/uninstall/uninstall.exe
2.3. Upgrade versions using e² studio installer

[Check for Updates] (mentioned in chapter 2.4) does not install new features. Therefore, you need e² studio installer to perform upgrade e² studio version. However, upgrading over major versions (i.e. the number increase in the major digit, for example, version up from v.4.3.0.008 to v.5.2.0.020) is not recommended because plugins compatibility and integrity may not guaranteed between different eclipse platform versions. Prior to the IDE upgrade, user must uninstall the old version IDE. To keep both old and new IDE versions, user can create new folder as installation destination for the new version IDE.

e² studio installer supports minor version update (i.e. number increase in the minor digit, for example, version up from v4.0.0.21 to v4.0.0.23). Please choose “Modify” at

2.4. Update plugins over Internet

Following procedure tells how to update existing plugins, not to install new features. Please use e² studio installer to upgrade 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] panel.

![Figure 2-1 [Check for Updates] Menu](image)

(2) By default, all the software components are selected in the [Available Updates] panel. This allows user to update them all to the latest version. (An example is shown in Figure 2-2). Click the [Next] button to proceed.
Figure 2-2 e² studio – Available Updates panel (1/3)

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

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

(4) Read and check the software license agreement. Click the [Finish] button to complete update.
(5) Click the [Help] → [About e² studio] to confirm the updated version.
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:
   http://www.renesas.com/e2studio_download
   Note: Offline version update using ‘Differential Update program’ is available with e² studio Ver3.x or older versions.

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 the upgrading.

![Figure 2-6 Upgrade e² studio from offline installer](image)

3. Follow the steps shown in Section 2.1 Installation of e² studio IDE. Step (2) Install Folder is skipped since Upgrade option will use the same destination folder as existing e² studio.
2.5. Installation of Compiler Package

V5.0 or newer e² studio installer is capable of installing compiler packages automatically during e² studio installation with valid Internet connection. However, in situation where Internet connection is not available during e² studio installation, compiler packages can be installed later from compiler package installation files from the web site shown below. This step is similar with e² studio V4.x or older.

Renesas Compiler Package download sites:

For RX Family: [http://www.renesas.com/rx_c](http://www.renesas.com/rx_c)

For RL78 Family: [http://www.renesas.com/rl78_c](http://www.renesas.com/rl78_c)

GNU Toolchain download site:

[https://gcc-renesas.com/](https://gcc-renesas.com/)

To check for compilers already installed, click ![Image](image.png) from the toolbar or click [Help] → [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.

![Image](image.png)

Figure 2-7 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 cannot be used for e² studio installation folder name, project name and its folder, and source file name.

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.

(1) Click [File] → [New] → [C/C++ Project] to open new project creation wizard.

(2) Select template for the new project (For e.g., Renesas RX: “Renesas CC-RX C/C++ Executable Project”). Click [Next] to proceed.
(3) Enter the project name. Click [Next] to proceed.

![Figure 3-3 New Project Creation Wizard (2/6)](image)

(4) Select Language, Toolchain, Toolchain Version, Target Device and Configurations. 

Note: “E2 Lite” can be selected in the same way as E1 in the Hardware Debug Configuration pull down menu.
(5) Select the Coding Assistant settings if user know how to use them, otherwise ignore this setting. Click [Next] to proceed.

Note:

- **Peripheral Code Generator** 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** not only supports the sample code to be easily embedded into a user application but also provides a common interface between the user applications and peripheral function drivers and middleware.

- **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.
(6) Keep the “Use I/O Library” unchecked and click [Next] to proceed.

(7) A project summary is displayed. Click [Finish] to generate the project.
(8) A new C project named “Tutorial” is created.
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. Import Existing Projects Into Workspace

The migration guideline between integrated development environments can be found at the following site.

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.), follow the following steps before building the project.

(1) Right click on project “Tutorial” and select [Properties] or use shortcut keys [Alt] + [Enter] to open the Properties window.

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

(2) Click [C/C++ Build] → [Change Toolchain Version] to view or change toolchain version. Refer to figure 4-2, the current version is v2.08.00 and click the “Versions” option to change toolchain version (if additional toolchain is installed).
(3) Click [C/C++ Build] → [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_8_0\doc\".
4.2. Build A Sample Project

A project can be built by one of the ways below:

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

(2) Click on the project to set focus and select [Project] → [Build Project]

(3) Click on the project to set focus and click on icon.

(4) Click on the project to set focus and press [Ctrl] + [B]

![Figure 4-4 Build a Sample “Tutorial” Project](image)

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.

![Figure 4-5 Project is built successfully.](image)
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 context menu

2. Select [Save build settings report] to save build settings report

![Figure 4-6 Project Reporter]
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.

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

![Figure 5-1 Switch to [Debug] Perspective](image)

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] → [Debug Configurations...] or ![icon](image) (downward arrow) → [Debug Configurations...] to open the “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”.

(3) Switch to the [Debugger] tab, set “E1” as the debug hardware and “R5F564ML” as the target device.
(4) 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”
  
  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”

- **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, be cautious to verify the board and emulator settings before connection.
(5) Switch to [Debug Tool Settings] sub tab, based on the RSK RX64M board to ensure

- Memory
  Endian = “Little Endian”

(6) Click [Apply] button to confirm the settings. Then click [Debug] to execute the debug launch configuration to connect to the E1 (or E2 Lite) and RSK RX64M board.
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".

![Debug View Screenshot]

**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.

2. 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.

![Figure 5-8 Duplicate A Selected Debug Launch Configuration](image)

(1) The debug launch configuration can be configured as described in chapter 5.1. For example, change the Debug Hardware to “E2 Lite (RX)”.

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

![Figure 5-9 Attach Launch Configuration to Specific Project](image)

Notes for RL78 debugging:

- Hot Plug connection is supported for RL78/F1A, F13, F14 and F15 only.
5.3. **Launch Bar**

This section explains the usage of ‘Launch Bar’, which is supported from V6.0.0 or later version. Launch Bar is located in the toolbar area of e² studio main window. Interface is very simple as shown below to build and debug for the selected launch target.

![Figure 5-10 Launch Bar interface](image)

Launch Bar buttons behave as follows:

- ![Build button](image) button builds the load module of the selected launch configuration.

  Note: There is another build button ![Build button](image) in the project management toolbar builds active build configuration of Project Explorer, while the launch bar does not reflect the active state in Project Explorer.

- ![Launch button](image) 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 → [Preferences], then click [Run/Debug] → [Launching] → [Launch Bar]

![Figure 5-11 Preferences for the Launch Bar](image)
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:

- **Step Over (F6)** can be used for stepping over the next method call (executing but without entering it) at the currently executing line of code.
- **Step Into (F5)** can be used for stepping into the next method call at the currently executing line of code.
- **Resume (F8)** can be clicked again to resume running.

To stop the debugging process, **Terminate (Ctrl+F2)** button is clicked to end the selected debug session and/or process or **Disconnect** button is clicked to disconnect the debugger from the selected process.

The other operations are as following:

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

**Note:** To demonstrate the features in following section, please use the sample code for RX64M from Renesas website as follows:

1. Download the sample code for RX64M from Renesas website:
   
Figure 5-13 Download the Sample Code

(2) After extract the package, find the project “Tutorial”

<table>
<thead>
<tr>
<th>Name</th>
<th>Date modified</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>12/21/2017 5:18 PM</td>
<td>File folder</td>
<td></td>
</tr>
<tr>
<td>Async_Serial</td>
<td>12/21/2017 5:18 PM</td>
<td>File folder</td>
<td></td>
</tr>
<tr>
<td>Low_Power_Mode</td>
<td>12/21/2017 5:18 PM</td>
<td>File folder</td>
<td></td>
</tr>
<tr>
<td>RTC</td>
<td>12/21/2017 5:18 PM</td>
<td>File folder</td>
<td></td>
</tr>
<tr>
<td>System_Input_Capture</td>
<td>12/21/2017 5:18 PM</td>
<td>File folder</td>
<td></td>
</tr>
<tr>
<td>Timer_PWM</td>
<td>12/21/2017 5:18 PM</td>
<td>File folder</td>
<td></td>
</tr>
<tr>
<td>Tutorial</td>
<td>12/21/2017 5:18 PM</td>
<td>File folder</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-14 The Sample project

(3) In e2 studio, select [File] → [Import]

Figure 5-15 Import the sample project

(4) In the [Import] dialog, select [General] → [Existing Projects into Workspace]. Click [Next]
In the [Import Projects] dialog, select “Select root directory”. Click [Browse] then select the folder “Tutorial” in the sample code package.

The project “Tutorial” will be listed in “Projects”. Check “Copy projects into workspace” then click [Finish]
(7) Right click on the imported project and select “Upgrade Legacy e2 studio Projects…”
(8) Select “Tutorial” project and click [Finish]

![Figure 5-20 Finish the upgrading](image)

(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].

![Figure 5-21 Update project toolchain](image)

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

1. 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 (Hardware breakpoint) or (Software breakpoint) is displayed in the marker bar depending on the [Breakpoint Type] selected. [Breakpoint Type] is hardware breakpoint by default.
2. Alternatively, right-click at the marker bar to choose [Toggle Hardware Breakpoint] or [Toggle Software Breakpoint] to set a hardware breakpoint or a software breakpoint.
3. Click [Windows] → [Show View] → [Breakpoints] or icon (or use shortcut key [ALT] + [Shift] + [Q], [B]) to open the [Breakpoints] view to view the corresponding software breakpoints set. Software breakpoints can be enabled and disabled in the [Breakpoints] view.

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

1. To disable a specific breakpoint, right-click on the Software breakpoint or Hardware breakpoint located in the left margin of the [C/C++ Editor] pane and select [Disable Breakpoint], or uncheck the related line in the Breakpoints view. A disabled breakpoint is displayed as a white dot ( or ).
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.

![Figure 5-22 Breakpoints view](image-url)
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.

**Figure 5-23 [Expressions] View**

To watch a global variable,

1. Click [Window] → [Show View] → [Expressions] or icon ![icon](image) 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.
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.

To view the general register “r0”,

1. Click [Window] → [Show View] → [Registers] or icon  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] → [Show View] → [Memory] or icon to open the [Memory] view.

2. Click the icon to open [Monitor Memory] dialog box. Enter the address of the variable “adc_count”.

(3) Figure 5-25 [Memory] View (1/2)
To add new renderings format (e.g. Raw Hex) for the variable “adc_count”,

(1) Click the tab [New Renderings...] 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>”.

Figure 5-26 [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.

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

1. Click [Window] → [Show View] → [Disassembly] or icon 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.

Figure 5-27 [Disassembly] View
Figure 5-28 Source Addresses Menu

Figure 5-29 Source Addresses displayed in Editor
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.

![Figure 5-30 [Variables] View]

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

1. Click [Window] → [Show View] → [Variables] or icon 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.
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.

![Figure 5-31 [Eventpoints] View (1/2)](image)

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

1. Click [Window] → [Show View] → [Eventpoints] or icon ![icon] 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.
Figure 5-32 [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.
Figure 5-33 Execution of Event Break

Figure 5-22 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.

Figure 5-34 [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 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 “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.

![Figure 5-35 [Trace] View (1/2)](image_url)

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.
- Turn on the Trace view by selecting the 🌡️ icon.
- Click icon 🇼 (?) (Acquisition) to set
  - Trace Mode: "Fill until stop"
  - Trace Type: “Branch”
  - Bus Master Of Data Access: “CPU”
- Click [OK] to proceed.
• 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”).
• Then, switch to [Stop] tab, add the 2nd event point at “R_LCD_Display()” function (by the execution address “&R_LCD_Display”).
• Next, execute the program after reset.

Figure 5-36 [Trace] View (2/2)
Figure 5-37 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.
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.

![Help Menu](image)

**Figure 6-1 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 [RenesasRulz Community Forum] to go online forum that is dedicated to topics and discussion related to e² studio IDE. Internet connection is required.
## Revision Record

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>July 20, 2018</td>
<td>-</td>
<td>First Edition issued, Supporting e² studio IDE v7.0.0</td>
</tr>
</tbody>
</table>

-
Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, 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: +86-10-8235-1155, Fax: +86-10-8235-8799

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langleo Road, Putuo District, Shanghai, P.R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852-2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886-2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B. Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Pensiarian Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9380, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

© 2017 Renesas Electronics Corporation. All rights reserved